

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Пастушков Александр Викторович

МЕТОД И АЛГОРИТМЫ ПОИСКА ОБЪЕКТА В ВИДЕОПОТОКЕ

05.13.17 – Теоретические основы информатики

Диссертация на соискание учёной степени кандидата технических наук

Научный руководитель:

Калайда Владимир Тимофеевич, д.т.н., профессор

Томск – 2017

Оглавление

Введение.....	5
1 Анализ методов и алгоритмов обнаружения объекта и слежения за объектом в видеопотоке.....	11
1.1 Классификация методов и алгоритмов обнаружения объекта и слежения за объектом в видеопотоке	11
1.2 Детерминированные методы.....	11
1.2.1 Методы поиска по шаблону.....	12
1.2.2 Методы поиска оптического потока	15
1.2.3 Методы поиска особенных точек	16
1.3 Вероятностные методы.....	31
1.4 Нейросетевые методы.....	32
1.4.1 Классическая нейронная сеть	33
1.4.2 SNoW – разреженная просеивающая сеть.....	34
1.5 Комбинированные методы.....	34
1.5.1 Метод Виолы–Джонса.....	35
1.5.2 Метод TLD	38
1.6 Проблемы обнаружения и слежения за объектом в видеопотоке.....	41
1.7 Сравнение алгоритмов и методов выделения объекта в видеопотоке	42
1.8 Выводы	48
2 Метод и алгоритм поиска объекта в видеопотоке	49
2.1 Определение требований.....	49
2.2 Функциональная модель.....	50
2.2.1 Диаграмма верхнего уровня.....	52
2.2.2 Диаграмма первого уровня	53
2.2.3 Диаграммы второго уровня.....	55

2.3	Выбор процедуры нахождения ключевых точек и вычисления дескрипторов	60
2.4	Алгоритмы и методы нахождения пересечения дескрипторов.....	62
2.4.1	RANSAC.....	63
2.4.2	Алгоритм Куна-Манкреса	64
2.4.3	Алгоритм ограничения области поиска объекта в кадре	65
2.5	Алгоритм идентификации области изображения	67
2.5.1	Градиентный спуск	68
2.6	Алгоритм поиска объекта в видеопотоке	71
2.7	Структурная модель.....	74
2.7.1	Структура модуля вычисления вектора дескрипторов объекта.....	75
2.7.2	Структура модуля извлечения кадра	76
2.7.3	Структура модуля вычисления вектора ключевых точек кадра	77
2.7.4	Структура модуля поиска областей претендентов	77
2.8	Выводы	78
3	Проектирование и реализация программы поиска объекта в видеопотоке	79
3.1	Проектирование классов системы.....	79
3.2	Сценарии работы системы	79
3.2.1	Сценарий инициализации изображения объекта.....	81
3.2.2	Сценарий поиска кадров объекта в видеопотока.....	81
3.3	Реализация программы поиска объекта в видеопотоке	85
3.4	Описание технологий	85
3.5	Выводы	88
4	Тестирование системы.....	89
4.1	Схема проведения тестирования	89

4.2 Результаты тестирования	92
4.2.1 Тестирование инвариантности к проективным изменениям.....	92
4.2.2 Сравнение результатов методов поиска по вероятности обнаружения объекта.....	101
4.2.3 Сравнение быстродействия разработанного метода с методом SURF	102
4.2.4 Тестирование производительности	104
4.2.5 Применение метода для видео, полученных в реальных условиях.....	111
4.3 Выводы	115
Заключение	116
Список литературы	118
Приложение А (справочное) Справки и акты об использовании результатов диссертационной работы.....	129
Приложение Б (справочное) Свидетельство о регистрации программы для ЭВМ	131
Приложение В (справочное) Текст скрипта программы для вычисления значений ROC кривой.....	132

Введение

Актуальность темы исследования. Информация о многих физических процессах представляется в виде скалярного поля. Системы безопасности, наблюдения, навигационные, ограниченного доступа, метеорологии являются областями анализа видеопотока. Одной из важных задач является выделение объекта в видеопотоке. С этой задачей связаны задачи слежения за объектом, сопоставления изображения с базой данных, поиск дубликатов изображений, соединения кадров.

Анализ научных работ по данному направлению исследований показал, что задача поиска объекта изучается многими исследователями (Т. Анштедт, И. Келлер, Х. Лутц [1], Guoshen Yu [2], P. Viola, M.J. Jones [3], Соколов С. М., Богуславский А. А. [4], Потапов А. С. [5] и др.), но, несмотря на это, на сегодняшний день задача не является полностью решённой. Процесс поиска объекта осложняется аффинными, проективными искажениями, перекрытием объекта другими объектами и шумом приёмника (датчика). Для реальных практических приложений задача должна обрабатывать видеопоследовательность в реальной скорости получения потока данных.

Методы выделения объекта можно разделить на следующие группы:

- нейросетевые методы;
- комбинированные методы;
- вероятностные методы;
- детерминированные методы.

Нейросетевые [6] и часть комбинированных методов с предварительным обучением [3] используются только на классах объектов, вошедших в обучаемые выборки. При этом сам процесс обучения системы является трудоёмким. В комбинированных методах с динамическим обучением [7] предполагается, что проективные искажения изображения объекта между соседними кадрами незначительны, и параметры поиска адаптируются к изменению проективных свойств изображения объекта между соседними кадрами в процессе слежения,

поэтому эта группа методов неприменима для поиска. Для вероятностных методов [8] необходимо задать начальное положение на первом кадре, поэтому они в основном используются только для слежения в видеопотоке. Из детерминированных методов самыми устойчивыми к проективным искажениям являются методы, основанные на поиске особенных точек [9], [2]. Однако их недостатком является высокая вычислительная сложность.

В связи с вышеизложенным объектом исследования в работе является задача поиска объекта в видеопотоке без предварительного обучения. В связи с чем, предметом исследования являются методы цифровой обработки изображений, используемые для решения задач по поиску объекта и слежению за объектом в видеопотоке.

Существующие в настоящее время решения этой проблемы, такие как Google Glass [10], Microsoft HoloLens [11], Kinect SDK [12], Metaio SDK [13], Vuforia SDK [14], Kudan SDK [15], OpenCV [16], продукты компании «Синезис» [17] обладают рядом недостатков, которые ограничивают область использования продуктов. Среди них следует отметить:

- ориентированность на узкий круг задач (в решении задачи используются ограничения, применяемые в определённой области);
- необходимость дополнительного специального оборудования (сенсоры, датчики);
- длительный процесс предварительного обучения.

Актуальность данной задачи в Российской Федерации подтверждается включением задачи «Информационно-телекоммуникационные системы» в список приоритетных направлений развития науки, технологии и техники в Российской Федерации [18] и внесением «Технологии информационных, управляющих, навигационных систем» в перечень критических технологий Российской Федерации [18].

Цели и задачи исследования. Цель исследования: разработка инвариантных к проективным преобразованиям и условиям съёмки алгоритмов,

обеспечивающих поиск заданного объекта в видеопотоке в реальном масштабе времени.

Для достижения этой цели необходимо решить следующие задачи:

1. Определение набора требований к системе поиска объекта в видеопотоке.
2. Разработка метода поиска объекта в видеопотоке.
3. Определение эффективного набора функций системы.
4. Разработка функциональной и структурной модели системы.
5. Разработка алгоритмов поиска объекта в видеопотоке, обеспечивающих поиск заданного объекта в видеопотоке в реальном масштабе времени.
6. Реализация алгоритмов поиска объекта в видеопотоке в программной системе.
7. Экспериментальное исследование эффективности работы системы – оценка качества процесса поиска объекта в видеопотоке.

Научная новизна.

1. Предложен метод поиска объекта в видеопотоке, включающий шаги вычисления дескрипторов ключевых точек на основе предварительно проективных искажений образца, и оценки меры схожести дескрипторов кадра и дескрипторов объекта, позволяющий уменьшить время поиска объекта.

2. Предложена модель формализованного описания параметров объекта, используемых для идентификации на базе совмещения интегральных и локальных признаков изображения объекта, обеспечивающая уменьшение ошибок поиска.

3. Разработан новый алгоритм поиска объекта в видеопотоке, основанный на обнаружении и сопоставлении ключевых точек изображения и использующий цветовую гистограмму для идентификации изображения объекта, позволяющий повысить вероятность обнаружения.

4. Предложен новый алгоритм сопоставления образца и области кадра, основанный на выборе масштаба сравниваемой области по масштабу сопоставляемого набора ключевых точек, обеспечивающий уменьшение количества анализируемых областей кадра.

Практическая значимость работы.

1. Основные результаты диссертации внедрены в компании ООО «Универсальные терминал системы» в виде программного модуля, предназначенного для осуществления поиска объекта в видеопотоке, и используемого в коммерческом продукте игрового движка GameX, поставляемого компанией ООО «Универсальные терминал системы». Копия акта внедрения результатов диссертационной работы приведена в Приложении А.

2. Разработанный алгоритм поиска объекта в видеопотоке послужил основой для создания программы слежения за объектом в видеопотоке – «Object tracking PS». Копия свидетельства о регистрации программы представлена в Приложении Б.

3. Разработанный в диссертации алгоритм поиска объекта в видеопотоке использован при выполнении научно-исследовательской работы №2013-1.5-14-515-0036-109 «Разработка технологий активного и пассивного зондирования атмосферы земли в оптическом и радио диапазонах для создания распределенной информационно-вычислительной системы комплексной обработки, передачи и использования экспериментальных данных» (2013 г.), проведенной на кафедре оптико-электронных систем и дистанционного зондирования Томского государственного университета.

4. Предложенные автором метод и алгоритмы использованы при разработке алгоритмов кластеризации облачных полей по спутниковым изображениям земной поверхности по теме: «Разработка радиофизических, оптических и ультразвуковых методов, аппаратуры и программных средств для комплексных исследований окружающей среды», выполняемой НИ ТГУ в рамках программы «Государственная поддержка ведущих университетов Российской Федерации в целях повышения их конкурентной способности среди ведущих мировых научно-образовательных центров» (проект № 1.42.2014).

5. Программно-техническая система слежения за объектом в видеопотоке «Object tracking PS», созданная в рамках диссертационного исследования, используется в Федеральном государственном бюджетном учреждении науки в

Институте оптики атмосферы им. В.Е. Зуева Сибирского отделения Российской академии наук (ИОА СО РАН) для решения задачи определения скорости и направления ветра по слежению за облачными полями. Копия акта внедрения результатов диссертационной работы приведена в Приложении А.

Методология и методы исследования. В качестве основных методов исследования выбраны методы обработки цифровых изображений, системного анализа, метод функционального проектирования по методологии IDEF0: метод структурного проектирования; методы объектно-ориентированного проектирования; методы объектно-ориентированного программирования.

Защищаемые положения.

1. Метод поиска объекта в видеопотоке, включающий шаги вычисления дескрипторов ключевых точек по предварительно искаженному проективными преобразованиями изображению образца и оценки меры схожести дескрипторов кадра и объекта, позволяет повысить скорость обработки до ≈ 10 раз.

Соответствует пункту 7 паспорта специальности: Разработка методов распознавания образов, фильтрации, распознавания и синтеза изображений, решающих правил.

2. Методика поиска объекта в видеопотоке, основанная на анализе ключевых точек и использовании алгоритмов идентификации, инвариантных к проективным искажениям изображения, позволяет повысить вероятность нахождения объекта с $\approx 0,8$ до уровня $\approx 0,9$.

Соответствует пункту 5 паспорта специальности: Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечения разработка и исследование методов и алгоритмов анализа текста, устной речи и изображений.

3. Алгоритм ограничения области поиска объекта в кадре, основанный на оценке масштабов ключевых точек проективно искаженных изображений объекта, обеспечивает уменьшение времени поиска объекта до $\approx 10\%$.

Соответствует пункту 5 паспорта специальности: Разработка и исследование моделей и алгоритмов анализа данных, обнаружения

закономерностей в данных и их извлечениях разработка и исследование методов и алгоритмов анализа текста, устной речи и изображений.

4. Программная система поиска объекта в видеопотоке, использующая все вычислительные ресурсы ЭВМ, обеспечивает решение задачи в условиях проективных искажений со скоростью обработки 25 кадров в секунду для размеров картинки до 960×540 точек.

Соответствует пункту 1 паспорта специальности: Исследование, в том числе с помощью средств вычислительной техники, информационных процессов, информационных потребностей коллективных и индивидуальных пользователей.

Достоверность результатов. Степень обоснованности результатов, изложенных в диссертации, обеспечивается корректностью постановки задачи, тщательным анализом предложенных методов и алгоритмов, а также подтверждается удовлетворительным согласием результатов численных расчётов и экспериментальных данных. Основные научные результаты работы докладывались и обсуждались на следующих конференциях: международная научно-практическая конференция «Современные направления теоретических и прикладных исследований '2014» (г. Одесса, 2014); международная научно-практическая конференция «Актуальные проблемы радиофизики» (г. Томск, 2013, 2015); международная научно-практическая конференция «Перспективные инновации в науке, образовании, производстве и транспорте '2013» (г. Одесса, 2013); международный симпозиум «Оптика атмосферы и океана. Физика атмосферы» (г. Томск, 2015). Основное содержание диссертации отражено в 11 научных работах, в том числе в 3 статьях в журналах, входящих в перечень ВАК, в 1 статье в журнале, входящем в систему Web Of Science. Получено свидетельство о государственной регистрации программы для ЭВМ (№2014662190). Работа выполнена при поддержке Министерства образования и науки РФ в соответствии с государственным заданием ТУСУР 2.8172.2017/8.9.

Структура и объем работы. Диссертация изложена на 135 страницах, содержит 47 рисунков, 14 таблиц, и состоит из введения, четырех разделов, заключения, списка литературы, включающего 84 наименования, 3 приложений.

1 Анализ методов и алгоритмов обнаружения объекта и слежения за объектом в видеопотоке

1.1 Классификация методов и алгоритмов обнаружения объекта и слежения за объектом в видеопотоке

Видеоаналитика – технология, использующая методы компьютерного зрения для автоматизированного получения различных данных на основании анализа последовательности изображений, поступающих с видеокамер в режиме реального времени или из архивных записей [1]. Под задачей обнаружения динамических объектов понимается задача обнаружения и выделения изменяющихся участков изображения в последовательности кадров [19]. Соответственно, под обнаружением определённого объекта понимается выбор одного или нескольких обнаруженных динамических объектов, которые имеют некоторые схожие признаки с заданным объектом поиска. Признаки выбираются согласно алгоритму. Трекинг (от англ. tracking – слежение) называется определение местоположения движущегося объекта (нескольких объектов) во времени с помощью камеры [20].

Методы обнаружения и слежения можно разбить на следующие группы:

- детерминированные методы;
- вероятностные методы;
- нейросетевые методы;
- комбинированные методы.

В этих методах объект слежения в последовательности кадров воспринимается по-разному: объект с неизменяющимися, либо с изменяющимися признаками.

1.2 Детерминированные методы

Детерминированные методы выдают уникальный и предопределённый результат для заданных входных данных. Детерминированные методы

рассматривают объект слежения, как объект с неизменяющимися признаками. Эти методы можно разделить на группы: *методы поиска оптического потока*, *методы поиска особенных точек* и *методы поиска по шаблону*.

Методы поиска оптического потока основаны на вычислении разреженного оптического потока. Эти методы строят векторное поле скоростей выделенных точек (пикселей изображения).

Методы поиска особенных точек основаны на вычислении характерных особенностей на изображении и на нахождении соответствия между ними в видеопоследовательности.

Методы поиска по шаблону не имеют этапа обучения (методы без учителя [21]). Эти методы вычисляют набор признаков на одном заданном изображении с объектом для поиска. Методы поиска по шаблону имеют сложный этап обнаружения объекта.

1.2.1 Методы поиска по шаблону

В методах поиска по шаблону используются детекторы особенностей. Детекторы обнаруживают особенные, отличительные участки изображения. Можно выделить самые распространённые детекторы – это детекторы рёбер, детекторы углов, детекторы окружностей. Методы поиска по шаблону применяют в основном в качестве вспомогательных методов, так как эти методы позволяют обнаружить некоторые геометрические примитивы (прямая, круг, прямоугольник), а как их сравнивать на изображениях – это отдельная задача. Для сопоставления обнаруженных участков изображения может применяться алгоритм сопоставления дескрипторов особенностей, в котором особенными точками будут каждые точки найденных примитивов.

Детектор прямых линий

С точки зрения детектора прямых линий преобразование можно представить как суммирование яркостей точек на контурном изображении вдоль

всевозможных направлений. Направления однозначно задаются перпендикулярными им векторами, проведёнными из центра картинной плоскости. Вектора задаются в полярной системе координат длиной и углом с вертикалью. На выходе преобразования получается функция, зависящая от двух аргументов угла и расстояния. По значениям функции можно определить количество точек, лежащих вдоль определённой прямой линии.

Первый этап предполагает выделение контуров. На втором этапе происходит суммирование яркостей точек вдоль прямой, которая задается углом и длиной. Результатом суммирования по всем прямым линиям является двумерная функция, зависящая от угла и расстояния. Полученная функция не несёт информации о расположении отрезков на линии, она лишь говорит, что он есть, поэтому, в дополнение к описанным операциям, потребуется реализация алгоритма локализации отрезка на прямой.

К достоинствам преобразования следует отнести высокую надёжность детектирования прямых линий. Разрывы контурной линии вдоль прямой оказывают незначительное влияние на работу алгоритма.

К недостаткам следует отнести необходимость проведения операции нахождения контуров, поиска областей пересечения траекторий отдельных точек в многомерном пространстве параметров и необходимость дополнительных алгоритмов, для локализации отрезков на найденных прямых линиях.

Детекторы углов

Один из первых алгоритмов предложил Бедет [22]. Он определяет положения углов по максимумам определителя Гессиана от функции яркости изображения:

$$H = I_{xx}I_{yy} - I_{xy}^2$$

Этот метод работает хорошо с углами равными 90. И так, как в методе используются вторые производные от функции яркости, то результат сильно подвержен влиянию шума. В свою очередь Форстнер [23] предложил детектор

углов, использующий только первые производные от функции яркости, и определил углы, как локальные максимумы:

$$F(x, y) = \frac{\bar{I}_x^2 \cdot \bar{I}_y^2 - (\overline{I_x \cdot I_y})^2}{\bar{I}_x^2 + \bar{I}_y^2},$$

где чёрточки над переменными обозначают среднее значение в некоторой области точки (x,y).

Недостатком детекторов углов, использующих компоненты градиента яркости, является то, что определение самих компонент градиента основано на дифференциальных масках, моделях горизонтального и вертикального контрастного перепадов. Они плохо работают в местах расположения углов, т. к. маска предполагает, что контрастный перепад может быть продлён по прямой до бесконечности.

Детекторы окружностей

Очевидным методом нахождения окружностей на изображении является прослеживание кривизны контурных линий. Али Айдариди Рад и др. предложили алгоритм быстрого поиска окружностей на изображении [24], используя противоположную направленность пары векторов градиента, лежащих на противоположных концах окружности, а также тот факт, что их базы лежат на прямой параллельной им [25].

Метод превосходит по скорости метод Хука для поиска окружностей и является более устойчивым к наличию шума типа «соль и перец» (шум в виде случайных белых и чёрных точек).

Методы поиска по шаблону позволяют обнаружить некоторые геометрические примитивы на изображении. Процесс обнаружения быстр по скорости, но алгоритм сопоставления примитивов может быть очень вычислительно сложным, потому что необходимо учитывать взаимное расположение примитивов на изображении с объектом на изображении, на котором осуществляется поиск.

Таким образом, детерминированные методы воспринимают объект с практически неизменяющимися свойствами в видеопоследовательности, методы основаны на обнаружении графических примитивов: точка, прямая, круг... Методы являются вычислительно сложными, имеют большую популярность в настоящее время благодаря своему качеству обнаружения.

1.2.2 Методы поиска оптического потока

Методы поиска оптического потока основываются на вычислении направления характерных участков изображения. В этих методах процесс обнаружения делится на два этапа: вычисление векторного поля скоростей и определение смещения объекта.

Метод Лукаса–Канаде

Метод Лукаса–Канаде – локальный метод вычисления оптического потока, имеющего линейную вычислительную сложность. Основное уравнение оптического потока содержит две неизвестных и не может быть однозначно разрешено. Метод Лукаса–Канаде обходит неоднозначность за счет использования информации о соседних пикселях в каждой точке.

Метод основан на предположении, что в локальной окрестности каждого пикселя значение оптического потока одинаково. Рассмотрим пиксель p , тогда, по методу Лукаса–Канаде, оптический поток должен быть одинаков для всех пикселей, находящихся в окне с центром в точке p . А именно, вектор оптического потока (V_x, V_y) в точке p определяется по формуле:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i \omega_i I_x(q_i)^2 & \sum_i \omega_i I_x(q_i) I_y(q_i) \\ \sum_i \omega_i I_x(q_i) I_y(q_i) & \sum_i \omega_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i \omega_i I_x(q_i) I_t(q_i) \\ -\sum_i \omega_i I_y(q_i) I_t(q_i) \end{bmatrix},$$

где q_1, q_2, \dots, q_n – пиксели внутри окна, $I_x(q_i)$, $I_y(q_i)$, $I_t(q_i)$ – частные производные изображения I по координатам x , y и времени t , вычисленные в точке q_i , ω_i – вес,

присвоенный пикселу q_i . В качестве весов ω_i обычно используется нормальное распределение расстояния между q_i и p [26].

Метод Лукаса–Канаде является сугубо локальным и не может определить направление движения пикселей внутри однородных областей. Некоторые изображения могут давать вырожденную матрицу A , для которой не может быть найдена обратная матрица, соответственно для таких изображений невозможно определить смещение.

На сегодняшний день метод Лукаса–Канаде имеет множество модификаций. В методе Томаси–Канаде движением считается смещение и рассчитывается путём итеративного решения построенной системы линейных уравнений. Метод Ши–Томаси–Канаде учитывает аффинные искажения. Метод Джин–Фаваро–Соатто учитывает аффинные изменения освещённости.

1.2.3 Методы поиска особенных точек

Алгоритм работы методов поиска особенных точек можно разделить на два этапа: обнаружение особенных точек, сопоставление особенных точек. Для сопоставления обнаруженных особенностей используются дескрипторы особенностей. Дескриптор особенности – вектор числовых характеристик окрестности особенности $D(x) = [f_1(w(x))..f_n(w(x))]$, где $w(x)$ – некоторая окрестность точки x , а $f(w_1, w_2)$ – мера, используемая для сравнения окрестностей особых точек. При сопоставлении особенностей, для принятия решений о том, соответствуют ли друг другу особенности или нет, сравниваются именно дескрипторы особенностей.

Метод Харриса–Лапласа

Метод Харриса–Лапласа находит особенные точки на изображении. Классический метод Харриса–Лапласа не устойчив к масштабированию объектов на изображении, метод не находит особые точки при сильном изменении

масштаба. Опишем метод Харриса–Лапласа, учитывающий масштабирование объектов на изображении.

1. Для начала необходимо вычислить значения адаптированной к масштабированию функции Харриса для масштабов $\sigma_n = \xi^n \cdot \sigma_0$

$$H(x, \sigma_1, \sigma_D) = \det(\mu(x, \sigma_1, \sigma_D)) + 0.04 \cdot \text{trace}^2(\mu(x, \sigma_1, \sigma_D)),$$

где $\mu(x, \sigma_1, \sigma_D) = g(\sigma_1) \cdot \begin{bmatrix} L_{x,norm}^2(x, \sigma_D) & L_{xy,norm}(x, \sigma_D) \\ L_{x,norm}(x, \sigma_D) & L_{y,norm}^2(x, \sigma_D) \end{bmatrix}, \sigma_1 = \sigma_n, \sigma_D = s\sigma_1, s = 0.7.$

2. Количество слоев и значение шага масштаба ξ следует выбирать в зависимости от того, насколько большим может быть изменение масштаба между двумя изображениями.

3. Для каждого уровня масштаба найти локальные максимумы вычисленной функции Харриса, это и есть особые точки для данного масштаба изображения. Обычно таким образом получается достаточно много точек и часть из них можно отбросить. Например, можно отбросить все точки, для которых значение функции Харриса не превосходит некоторого значения H_{thr} , т.к. максимумы с небольшим значением функции Харриса менее устойчивы.

4. Для каждой найденной таким образом особенности установить, достигается ли в ней максимум функции $LoG(x, \sigma_n) = |L_{xx,norm}(x, \sigma_n) + L_{yy,norm}(x, \sigma_n)|$ по переменной n , т.е. $LoG(x, \sigma_{n-1}) < LoG(x, \sigma_n), LoG(x, \sigma_{n+1}) < LoG(x, \sigma_n)$. Если локальный максимум не достигается, либо значение функции не превосходит порога LoG_{thr} , то точка отбрасывается.

5. Все оставшиеся точки являются особенностями изображения, с каждой точкой ассоциирован масштаб σ_n , на котором она была обнаружена [27].

Инвариантный дескриптор к изменению масштаба

При использовании scale–space детектора особенностей добиться инвариантности к изменению масштаба очень просто. Для этого достаточно перед вычислением дескриптора провести нормировку в соответствии с локальным масштабом особенности, например, если с особенностью ассоциирован масштаб

2, то окрестность особенности следует масштабировать с коэффициентом 0,5 и т.д. Если дескриптор состоит из выражений, в которых используются исключительно нормированные производные, то масштабировать окрестность не обязательно. Достаточно рассчитывать значения производных для значения масштаба σ , который ассоциирован с особенностью.

Инвариантный дескриптор к повороту

Самый простой метод добиться инвариантности к повороту при сопоставлении особенностей – использовать дескрипторы, компоненты которых инварианты к повороту. Все производные в выражении – нормированные производные.

Серьезным недостатком этого метода является то, что в дескрипторе нельзя использовать компоненты, которые не инвариантны к повороту, а число операторов, которые инвариантны к повороту, и при этом применимы на практике, ограничено. Еще один способ добиться инвариантности к повороту – предварительно нормировать окрестность точки особым образом, чтобы скомпенсировать поворот, и лишь потом вычислять дескрипторы для особенности. Для того чтобы нормировать окрестность по повороту требуется оценить т.н. ориентацию особенности (см. рисунок 1.1). Существует много методов оценки локальной ориентации особенности, все они так или иначе используют направление векторов градиентов в окрестности особенности.

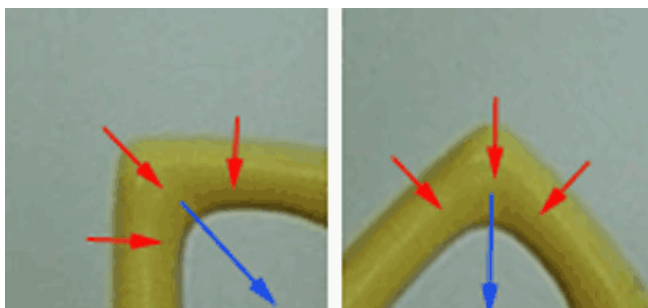


Рисунок 1.1 – Окрестность особой точки. Красным указаны направления градиентов, а синим показана ориентация особенности [27]

Разделим множество углов от 0 до 360 градусов, например, на 36 одинаковых участков, каждый по 10 градусов. Каждому из участков сопоставим ячейку в гистограмме H . Для каждой точки x_0 из некоторой окрестности a особенности необходимо вычислить фазу и модуль вектора градиента

$$\text{grad}(x_0, \partial) = (L_{x, \text{norm}}(x, 0, \partial) \ L_{y, \text{norm}}(x, 0, \partial)),$$

$$\theta = \frac{L_{y, \text{norm}}(x, 0, \partial)}{L_{x, \text{norm}}(x, 0, \partial)},$$

$$A = |\text{grad}(x_0, \partial)|,$$

$$H[i_\theta] = H[i_\theta] + Aw,$$

где i_θ – индекс ячейки, которая соответствует фазе градиента, а w – вес точки x_0 . В качестве веса можно использовать, например, $w = 1$ (это простейший случай), но более качественных и устойчивых результатов оценки удаётся добиться при использовании, в качестве веса, значения Гауссиана с центром в точке a .

После этого для каждой точки окрестности особенности в качестве ориентации особенности следует выбрать $\varphi = i \cdot 10^\circ$, где i – индекс элемента гистограммы с наибольшим значением. Затем необходимо повернуть особенность на угол $(-\varphi)$ вокруг центра особенности. К сожалению, для некоторой части особенностей ориентация оценивается неверно (обычно 10–20%) и дескрипторы этих особенностей оказываются абсолютно непригодны к сопоставлению. Именно это является основным недостатком данного подхода.

Инвариантный дескриптор к изменению освещенности

Для того чтобы сделать дескриптор инвариантным к изменению освещенности нужно для начала ввести модель этого изменения. Обычно применяют т. н. аффинную модель, которая подразумевает, что значения пикселей окрестности при изменении освещенности меняется по следующему закону: $\hat{I} = a \cdot I(x) + b$. Это модель не очень точно соответствует действительности и реально процессы, происходящие в пикселях при изменении освещенности намного сложнее, но в силу того, что особенности локальны и имеют небольшой

размер такой модели более чем достаточно. Учитывая принятую модель можно использовать следующий алгоритм, устраняющий влияние освещенности на значение пикселей в окрестности особенности:

$$I_{mean}(w(x_0)) = I(w(x)) - \text{mean}(I(w(x))), I_{result}(w(x)) = \frac{I_{mean}(w(x))}{\text{std}(I(w(x)))},$$

где $\text{mean}(I(w(x)))$ и $\text{std}(I(w(x)))$ обозначают выборочное среднее и среднее квадратичное отклонения в окрестности w , $I_{mean}(w(x))$ – окрестность, скомпенсированная по переносу, а I_{result} – результирующая окрестность, скомпенсированная по освещенности. Именно по особенности I_{result} следует вычислять дескриптор особенности, чтобы добиться инвариантности к изменениям освещенности. Также можно составлять дескриптор из функций, которые инвариантны к аффинным изменениям освещенности.

Алгоритм сопоставления

После того как для каждой особенности на паре изображений рассчитаны дескрипторы можно непосредственно приступить к сопоставлению особенностей. Существует множество вариантов алгоритмов сопоставления особенностей по их дескрипторам. Приведём один из них [27].

Обозначим наборы особых точек первого и второго изображения $A = \{a_i, i=1..n\}$, $B = \{b_i, i=1..m\}$, а их дескрипторы $D_A = \{D_{A,i}, i=1..n\}$, $D_B = \{D_{B,i}, i=1..m\}$. Будем считать что дескрипторы – вектора столбцы.

Для каждой точки первого изображения $H[i_\theta] = H[i_\theta] + A w$ найти точку второго изображения b_i , такую что $j = \arg \min_k (f(D_{A,j}, D_{B,k}))$, $f(D_1, D_2) = (D_1 - D_2) K^{-1} (D_1 - D_2)^T$, где K – матрица ковариации дескриптора. Обычно считают что дескриптор – случайная величина с многомерным нормальным распределением. При этом мера f соответствует расстоянию Махаланобиса.

Если значение $f(D_{A,i}, D_{B,j})$ превосходит значение некоторого порога f_{thr} , следует отбросить данное соответствие. Значение порога сильно зависит от дескрипторов, которые используются для сопоставления.

Все пары соответствующих точек следует упорядочить, используя некоторую меру качества. Как меру качества можно использовать неоднозначность соответствия, предложенную в [28], которая имеет следующий вид:

$$Amb(D_{A,i}, D_{B,j}) = \frac{f(D_{A,i}, D_{B,j})}{\min_{k \neq j} (f(D_{A,i}, D_{B,k}))}$$

Чем меньше значение меры Amb , тем более качественным является соответствие. Выбрать необходимо N первых (самых качественных) соответствий.

Scale-Invariant Feature Transform (SIFT)

Идея метода SIFT сводится к следующему: нахождение изображения объекта через уникальные локальные признаки объекта. Метод SIFT можно разделить на следующие этапы:

- определение локальных особенностей (точек интереса или ключевых точек);
- локализация особенностей;
- вычисление ориентаций особенностей;
- описание локальных особенностей через дескриптор;
- сопоставление дескрипторов.

Для определения особенностей используется Гауссово масштабируемое пространство. Построение масштабируемого пространства представляет операцию разложения исходного сигнала в семейство постепенно сглаживаемых, упрощающихся версий сигнала, удовлетворяющих требованиям линейности, инвариантности к сдвигу, к масштабу и обладающих свойством полугруппы и положительности [29]. Доказано, что гауссово масштабируемое пространство является линейным, инвариантным к сдвигу, к масштабу, не смещающим локальные экстремумы [30]. В методе SIFT область на изображении является особенной, если она является локальным экстремумом в масштабируемом пространстве (пирамиде) разности гауссиан. На этапе определения характерных

точек строятся пирамиды гауссианов и разности гауссианов (*DoG – Difference of Gaussian*):

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

где L – значение гауссиана в точке с координатами (x, y) , σ – радиус размытия, G – гауссово ядро, I – значение исходного изображения, $*$ – операция свертки.

Гауссово ядро вычисляется по следующей формуле:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

Разность гауссиан *DoG* вычисляется следующим образом:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma),$$

где k – коэффициент изменения радиуса размытия σ .

После вычисления разности гауссиан происходит передискретизация исходного изображения: уменьшается частота дискретизации в 2 раза по вертикали и горизонтали. Затем над уменьшенным изображением вычисляют разности гауссиан.

Таким образом, строится пирамида разности гауссиан до тех пор, пока размеры изображения не будут меньше определённого порога по ширине или по высоте, либо пока передискретизация не произведётся максимально допустимое количество раз R .

После построения пирамиды находятся экстремумы путём сравнения каждой точки с восьмью соседями на текущем изображении *DoG* (если такие имеются) и с девятью соседями изображений *DoG* выше и ниже в пирамиде (см. рисунок 1.2) [9]. Этап построения масштабируемого пространства можно описать при помощи уравнения теплопроводности:

$$\frac{\partial u}{\partial t} = a^2 \Delta u,$$

где $u = u(t, x)$, $t \in R$, $x \in R^n$, Δ – оператор Лапласа по x , $a > 0$, $u(t, x)$ – температура среды в точке x в момент времени t [31]. В случае Гауссово масштабируемого пространства получается следующее уравнение:

$$\partial L_\sigma = \sigma^2 \Delta L,$$

где $L(x, y, 0) = I(x, y)$. После аппроксимации производной ∂L_σ получается следующая формула:

$$L(x, y, k\sigma) - L(x, y, \sigma) \approx (k-1)\sigma^2 \Delta L,$$

где k – шаг изменения масштаба σ . Таким образом, для моделирования изображения в некотором масштабе необходимо найти разность гауссиан.

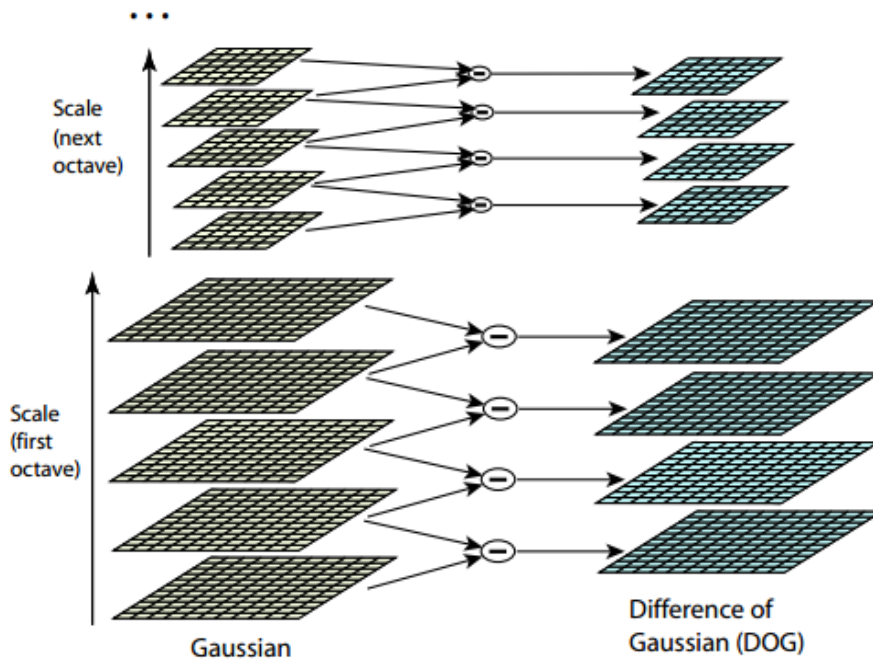


Рисунок 1.2 – Процесс нахождения локальных особенностей

После определения особенностей производится локализация и проверка особенностей на контрастность и на принадлежность к границе между областями изображения, т. е. исключаются из дальнейшего рассмотрения особенности, которые имеют контрастность ниже определённого порога и которые находятся на линейной протяжённой границе между цветовыми объектами (см. рисунок 1.3).

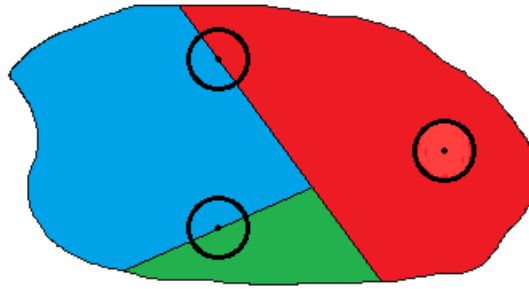


Рисунок 1.3 – Примеры особенностей, исключаемых из рассмотрения

Вначале этапа локализации определяются координаты ключевых точек с субпиксельной точностью, для этого аппроксимируется функция DoG многочленом Тейлора второго порядка в найденном экстремуме [32]:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x ,$$

где x – смещение, $x = (x, y, \sigma)$. Вычислив производную и приравняв равенство к нулю, получим экстремум \hat{x} :

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} .$$

Если смещение \hat{x} больше, чем 0.5, то это означает, что экстремум лежит ближе к другой точке выборки. В этом случае точка выборки изменяется, и выполняется интерполяция заново для изменённой точки. Итоговое смещение \hat{x} добавляется к месту точки, чтобы получить интерполированную оценку для расположения экстремума.

После вычисления экстремума с субпиксельной точностью, вычисляется значение DoG в точке \hat{x} :

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} .$$

Значение $D(\hat{x})$ сравнивается с граничным значением $D_{min} = 1$. Если $|D(\hat{x})| < D_{min}$, то особенность считается с малой контрастностью и извлекается из дальнейшего рассмотрения.

Проверка особенностей на принадлежность к границе между областями изображения осуществляется через вычисление матрицы Гессе H :

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}.$$

Пусть $Tr(H)$ – след матрицы, $Det(H)$ – определитель матрицы H , тогда точка исключается из дальнейшего рассмотрения, если имеет большой изгиб α вдоль границы и малый β в перпендикулярном, тогда:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$Det(H) = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta.$$

Точка извлекается из дальнейшего рассмотрения, если не удовлетворяет условию:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r},$$

где $r = \alpha / \beta$.

Для нахождения ориентации ключевой точки вычисляются направления и величины градиентов в окне с радиусом 3σ в центре особенности:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2},$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right),$$

где m – величина градиента, θ – направление градиента. После вычисления направлений градиентов строится гистограмма направлений $Hist$, состоящая из 36 компонент:

$$w_k = m(x, y) * G(x, y, 1.5\sigma),$$

где направление градиента выбирается из ближайшего направления k гистограммы $Hist$, w_k – вес градиента в точке (x, y) для k -ой компоненты $Hist$. Направление особенности находится в промежутке, покрываемом максимальной компонентой гистограммы $Hist$. Максимальное значение и два соседние значения интерполируются параболой, точка максимума параболы берётся в качестве направления особенности. Если в гистограмме есть ещё компоненты со

значениями не меньше 80% от максимального направления, то они аналогично интерполируются и дополнительные направления добавляются к ключевой точке.

После определения ориентации поворачивают окно с радиусом 3σ в центре особенности в направление ориентации – этим достигается инвариантность относительно поворота. Затем вся область в окне делится на 16 квадратных одинаковых блока (4×4), в которых вычисляются гистограммы направлений (см. рисунок 1.4).

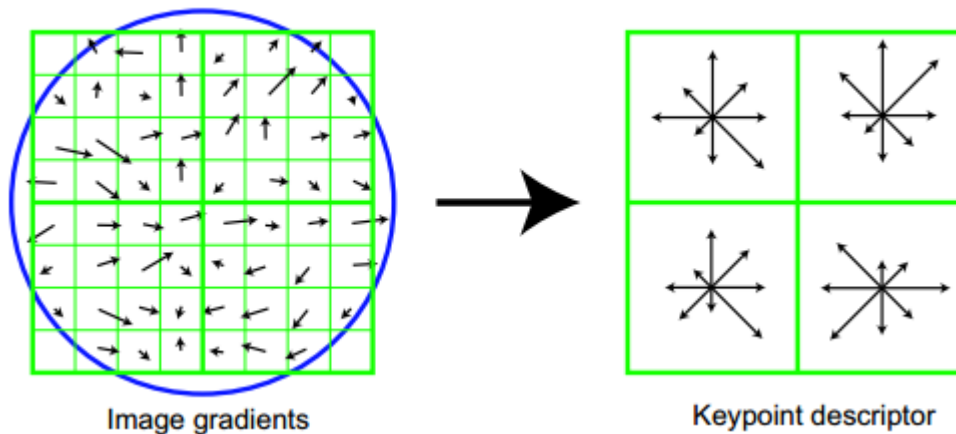


Рисунок 1.4 – Построение дескриптора размерностью $2 \times 2 \times 8$

Дескриптор особенности состоит из полученных гистограмм. Размерность дескриптора оригинального метода SIFT составляет $4 \times 4 \times 8 = 128$ элементов. После описания дескриптора через вектор, состоящего из 128 элементов, дескриптор нормализуется.

Affine-Scale-Invariant Feature Transform (ASIFT)

Метод ASIFT является усложнением метода SIFT и является полностью аффинно-инвариантным методом за счёт моделирования всех изменений угла наклона камеры.

Цифровое изображение I можно разложить на составляющие по следующей формуле:

$$I = H_{\lambda} R_1(\psi) T_t R_2(\phi) = \lambda \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix},$$

где λ – определитель матрицы I , R_1 , R_2 – повороты, $\varphi \in [0, \pi)$, T_t – наклон, $t > 1$. Геометрическое представление аффинной декомпозиции представлено на рисунке 1.5.

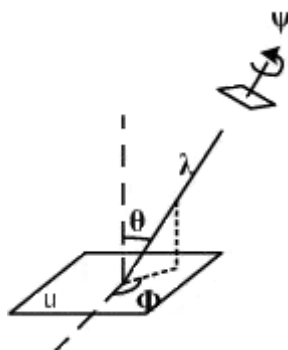


Рисунок 1.5 – Геометрическое представление аффинной декомпозиции

Угол θ предлагается изменять по геометрической прогрессии: $1, a, a^2, \dots, a^n$, где $n = 5$, $a = \sqrt{2}$, а угол φ для каждого наклона изменять по арифметической прогрессии: $0, b/t, \dots, kb/t$, где $b \simeq 72^\circ$ и $kb/t < 180^\circ$. На каждой смоделированной картинке находятся ключевые точки с дескрипторами методом SIFT. Все найденные дескрипторы сравниваются с дескрипторами оригинального изображения [2].

Speed Up Robust Feature (SURF)

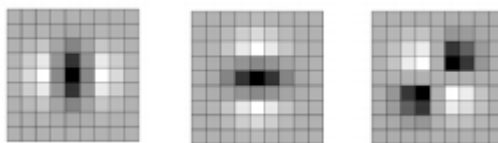
Метод SURF является аналогом метода SIFT. Метод ищет ключевые точки и строит описание найденных ключевых точек через дескрипторы особенностей.

Ключевая точка в SURF – это локальный экстремум детерминанта матрицы Гессе. Для двумерного случая детерминант матрицы Гессе определяется следующим образом:

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2,$$

где H – матрица Гессе, $f(x,y)$ – функция изменения градиента. На рисунке 1.6 показаны фильтры компонент D_{xx} , D_{yy} , D_{xy} матрицы Гессе H размером 9×9 . Метод SURF использует аппроксимацию этих фильтров, изображённых на рисунке 1.7.

Такие фильтры устойчивы к вращению и их можно эффективно вычислить при помощи интегральной матрицы.



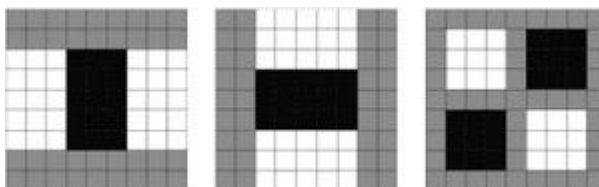
Слева направо: D_{xx} , D_{yy} , D_{xy} . Серый регион соответствует значению 0.

Рисунок 1.6 – Фильтры компонент D_{xx} , D_{yy} , D_{xy} матрицы Гессе

Детерминант матрицы Гессе для фильтров, изображённых на рисунке 1.7 вычисляется по следующей формуле:

$$\det(H_{approx}) = D_{xx} D_{yy} - (0,9 D_{xy})^2,$$

где коэффициент 0,9 взят на основании результатов в [33]. Гессеиан инвариантен относительно поворота, но не относительно масштаба, поэтому используются фильтры с разным масштабом для определения гессеианов.



Слева направо: D_{xx} , D_{yy} , D_{xy} . Серый регион соответствует значению нуля, белый – плюс один, чёрный – минус два (на третьем фильтре минус один).

Рисунок 1.7 – SURF фильтры компонент D_{xx} , D_{yy} , D_{xy} , D_{xy} матрицы Гессе

Для определения ориентации особенности используются фильтры Хаара (см. рисунок 1.8), размеры которых равны $4s$, где s – масштаб особой точки. Значения вейвлета Хаара dX и dY для каждой точки умножаются на значение гауссианы с центром в особой точке и сигмой равной $2,5s$. Благодаря такому взвешиванию, отсеиваются случайные помехи на далёких расстояниях от ключевой точки. Затем

при помощи углового окна выбирается направление особенности, при котором длина суммарного вектора для попавших в окно точек будет максимальна.



Чёрные области соответствуют значению -1, белые +1.

Рисунок 1.8 – SURF фильтры компонент D_{xx} , D_{yy} , D_{xy} матрицы Гессе

Для вычисления дескриптора особенности формируется прямоугольная область (квадрат) вдоль приоритетного направления размера $20s$, где s – масштаб в котором была найдена особенность. Квадрат разбивается на 16 квадратных подобластей, в каждом из которых берется регулярная сетка 5×5 . В каждой подобласти вычисляется градиент, с помощью фильтра Хаара. Размер фильтра Хаара берется равным $2s$.



Рисунок 1.9 – Поведение величин $\sum dX$, $\sum |dX|$, $\sum dY$, $\sum |dY|$ при различных ситуациях

После нахождения 25 точечных градиент подобластей, вычисляются четыре величины: $\sum dX$, $\sum |dX|$, $\sum dY$, $\sum |dY|$. На рисунке 1.9 приведены примеры поведений этих величин для разных участков изображений. В случае, когда область однородна, то все коэффициенты относительно малы. Для вертикальных

повторяющихся полосок все величины кроме $\sum |dX|$ малы. В случае увеличения яркости вдоль оси x , величины $\sum dX$, $\sum |dX|$ имеют большие значения относительно $\sum dY$, $\sum |dY|$.

Таким образом, для 16 областей получается 64 значения величин, которые и являются компонентами дескриптора. Все значения величин взвешиваются на гауссиану, с центром в особой точке и с сигмой $3,3 s$.

Методы ASIFT, SURF и SIFT на сегодняшний день являются базовыми методами поиска ключевых точек на изображении. Существует огромное количество модификаций этих методов, уменьшающие ошибки поиска для определённого класса задач. Например, метод MODS соотносит корректно от 5 до 20% больше ключевых точек по сравнению с методом ASIFT [34].

В таблице 1.1 представлены вычислительные сложности методов SIFT, ASIFT и SURF согласно исследованиям [35], [36], [2], где N^2 – размер изображения, на котором осуществляется поиск, w – размер окна функции Гаусса, s – количество октав, x – количество точек в окрестности для вычисления ориентации, k – количество проективно искажённых изображений.

Таблица 1.1 – Вычислительная сложность методов SIFT, ASIFT, SURF

Метод	Вычислительная сложность
ASIFT	$O(k(sw^2+x^2)N^2)$
SIFT	$O((sw^2+x^2)N^2)$
SURF	$O(sN^2)$

Существуют ряд работ по объединению различных детекторов ключевых точек для уменьшения ошибки поиска [37], однако при этом увеличиваются вычислительные затраты.

Таким образом, к достоинствам методов поиска особенных точек можно отнести их высокую устойчивость к масштабированию, к незначительному повороту изображения объекта для поиска; к недостаткам можно отнести

зависимость обнаружения объекта от фона, высокую сложность, неустойчивость к специфичным изображениям объекта, на которых нельзя определить направления дескрипторов.

1.3 Вероятностные методы

Вероятностные методы воспринимают объект с изменяющимися признаками в последовательности кадров. Эти методы используют подход, основанный на понятии пространства состояний. Считается, что движущийся объект имеет определенное внутреннее состояние, которое измеряется на каждом кадре. В простейшем случае под состоянием понимается положение объекта на изображении. Чтобы оценить следующее состояние объекта, требуется максимально обобщить полученные измерения, т.е. определить новое состояние при условии, что получен набор измерений для состояний на предыдущих кадрах. Типичными примерами таких методов являются методы на базе фильтра Кальмана и фильтра частиц [38].

Фильтр Калмана применяют при известном начальном состоянии объекта на изображении, иными словами применяют для решения задачи сопровождения объекта. Задачу сопровождения можно рассматривать как хорошо изученную проблему теории управления, которая состоит в том, чтобы оценить состояние системы на основании последовательности зашумленных измерений.

Алгоритм состоит из двух повторяющихся фаз: предсказание и корректировка. На первом рассчитывается предсказание состояния в следующий момент времени (с учетом неточности их измерения). На втором, новая информация с датчика корректирует предсказанное значение (также с учетом неточности и зашумленности этой информации) [39].

Фильтр Калмана имеет линейную вычислительную сложность [40] и применяется для предсказания, определения признаков объекта. Из полученных результатов в [8] можно сделать вывод, что применение фильтра Калмана позволяет увеличить скорость обработки и повышает устойчивость метода

обнаружения объекта в видеопотоке, при этом результаты обнаружения практически не изменяются по сравнению с итерационными алгоритмами нахождения центра местоположения объекта.

Для некоторых практических задач, чтобы получить более точную оценку состояния системы, необходимо уйти от предположения, что шум имеет Гауссово распределение. В этом случае вводится понятие мультимодального распределения шума, а для моделирования подобных систем используются фильтры частиц. Фильтры частиц являются более общим подходом к решению задачи сопровождения с применением вероятностных методов. Мультимодальным распределением называется распределение, имеющее несколько мод или локальных максимумов. Мультимодальное распределение зачастую представляется смесью нескольких распределений.

Алгоритм воспроизведения условной плотности (CONDitional DENSity propAGATION, CONDENSATION) – базовый алгоритм фильтрации частиц, на основании которого строится большинство алгоритмов данной группы, применяемых в компьютерном зрении [54].

Таким образом, вероятностные методы слежения позволяют предсказывать состояние объекта на изображении, не сохраняя все данные о предыдущих состояниях; позволяют воспринимать объект с изменяющимися признаками в видеопоследовательности; устойчивы к зашумлению изображения, к изменению ряда характеристик изображения объекта: яркость, поворот, масштабирование. Вероятностные методы применяют на практике в качестве дополнительных методов для повышения устойчивости к изменению изображения объекта.

1.4 Нейросетевые методы

В нейросетевых методах объект в видеопотоке может рассматриваться одновременно с изменяющимися и неизменяющимися признаками. В этих методах неотъемлемой частью является этап обучения нейронной сети. Обучать сеть необходимо под каждый тип задач.

1.4.1 Классическая нейронная сеть

Основная идея, лежащая в основе нейронных сетей – это последовательное преобразование сигнала, параллельно работающими элементарными функциональными элементами, нейронами. Основным принципом настройки нейронной сети заключается в применении оптимизационных методов к минимизации среднеквадратичной ошибки, как следствие – склонность к переобучению. Главное преимущество нейронных сетей – гибкость.

Геометрически, разделяющая классы поверхность представляет собой множество гиперплоскостей. Каждая из областей, на которые гиперплоскости разбивают пространство признаков X относится к одному из классов. Существует множество методов обучения нейросетей, однако все они сводятся к минимизации среднеквадратичной ошибки. Важно отметить, что найденный минимум, будет локальным. Также следует отметить, что верно классифицированные прецеденты не вносят никаких изменений в оптимизируемый функционал. Таким образом, найденная разделяющая поверхность не будет являться ни единственным, ни оптимальным решением.

Системы распознавания объектов на изображениях основанные на нейронных сетях используют иерархическую архитектуру. Вначале вектор признаков обрабатывается грубой сетью с высоким уровнем ошибок второго рода, далее, если вектор не был классифицирован как не объект, решение корректируется более точной и более медленной сетью. В целом нейронные сети склонны к переобучению, хотя и существуют некоторые методы, которые в частном случае могут решить эту проблему. Устойчивость к шуму сильно зависит от конкретной архитектуры сети. В общем случае, нейросеть чувствительна к шуму. Вычислительная сложность квадратично зависит от числа нейронов в скрытом слое. Каждый нейрон требует вычисления функции активации. Для задач распознавания объектов на изображениях скорость обработки является недостаточной для применения в решении задач в реальной скорости потока данных [41].

1.4.2 SNoW – разреженная просеивающая сеть

SNoW (Sparse network of Winnows) – особый вид нейронной сети [6]. Вектор признаков полагается бинарным. Сеть состоит из двух (по числу возможных классов) линейных нейронов, связанных с компонентами вектора признаков. Классификация проходит по принципу победитель забирает всё.

Геометрически, SNoW представляет собой две гиперплоскости в пространстве векторов признаков. Вектор относится к тому классу, соответствующая гиперплоскость которого ближе всего. Таким образом, результирующая разделяющая поверхность является гиперплоскостью в исходном пространстве X . Одно из достоинств данной архитектуры – возможность «прореживать» вектор признаков, на основе обучающей выборки – компоненты вектора, не несущие информации, отбрасываются.

SNoW считается достаточно эффективным методом для решения задачи обнаружения объектов на изображениях. За счёт просеивания компонент вектора признаков достигается высокая скорость – сложность линейна относительно количества эффективных компонент вектора признаков [6].

Таким образом, нейронные сети имеют высокий процент распознавания объекта на изображении, низкий процент ложного распознавания, но необходимо учитывать, что нейронную сеть необходимо обучать под каждый тип задач. Методы поиска объекта с применением классических нейронных сетей имеют низкую скорость, в отличие от разреженных сетей.

1.5 Комбинированные методы

Особенность комбинированных методов заключается в том, что они состоят из нескольких методов, комбинируя методы по наивысшим показателям разных критериев. Такие методы более устойчивы к шуму, к различным видам искажений объекта. Комбинированные методы могут сочетать в себе детерминированные, вероятностные, нейросетевые методы. Комбинированные методы можно разбить на две группы: методы с учителем и методы без учителя.

1.5.1 Метод Виолы–Джонса

Метод был разработан и представлен в 2001 году Полом Виолой и Майклом Джонсом [3]. Этот метод относится к методам с учителем. Этап обучения происходит очень медленно, но зато результаты поиска очень быстры. Метод Виолы–Джонса использует признаки Хаара. Быстрое вычисление признаков достигается при помощи интегрального представления изображения. Алгоритм бустинга используется для выбора признаков. Метод Виолы–Джонса использует подход на основе сканирующего окна (scanning window): сканируется изображение окном поиска (так называемое, окно сканирования), а затем применяется классификатор к каждому положению. Система обучения и выбора наиболее значимых признаков полностью автоматизирована и не требует вмешательства человека, поэтому данный подход работает быстро.

Интегральное представление изображения

Интегральное представление позволяет быстро рассчитывать суммарную яркость произвольного прямоугольника на данном изображении, причем время расчета не зависит от размеров прямоугольника.

Интегральное представление изображения – это матрица, совпадающая по размерам с исходным изображением. В каждом элементе ее хранится сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента. Элементы матрицы рассчитываются по следующей формуле:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j),$$

где $I(i, j)$ – яркость пикселя исходного изображения.

Каждый элемент матрицы $L[x, y]$ представляет собой сумму пикселей в прямоугольнике от $(0, 0)$ до (x, y) , т.е. значение каждого пикселя (x, y) равно сумме значений всех пикселей левее и выше данного пикселя (x, y) . Расчет матрицы занимает линейное время, пропорциональное числу пикселей в изображении, поэтому интегральное изображение просчитывается за один проход.

Расчет матрицы производится по следующей формуле:

$$L(x, y) = I(x, y) - L(x-1, y-1) + L(x, y-1) + L(x-1, y).$$

По интегральной матрице можно очень быстро вычислить сумму пикселей произвольного прямоугольника, произвольной площади.

Признаки Хаара

Признак – это отображение $f: X \Rightarrow D_f$, где D_f – множество допустимых значений признака. Если заданы признаки f_1, \dots, f_n , то вектор признаков $x = (f_1(x), \dots, f_n(x))$ называется признаковым описанием объекта $x \in X$. Признаковые описания допустимо отождествлять с самими объектами. При этом множество $X = D_{f_1} * \dots * D_{f_n}$ называют признаковым пространством [42].

Признаки делятся на следующие типы в зависимости от множества D_f :

- бинарный признак, $D_f = \{0, 1\}$;
- номинальный признак: D_f – конечное множество;
- порядковый признак: D_f – конечное упорядоченное множество;
- количественный признак: D_f – множество действительных чисел.

Для каждого типа задач необходимо подбирать признаки индивидуально – от этого зависит качество поиска.

В стандартном методе Виолы – Джонса используются прямоугольные признаки (см. рисунок 1.10). Эти признаки называются примитивами Хаара.

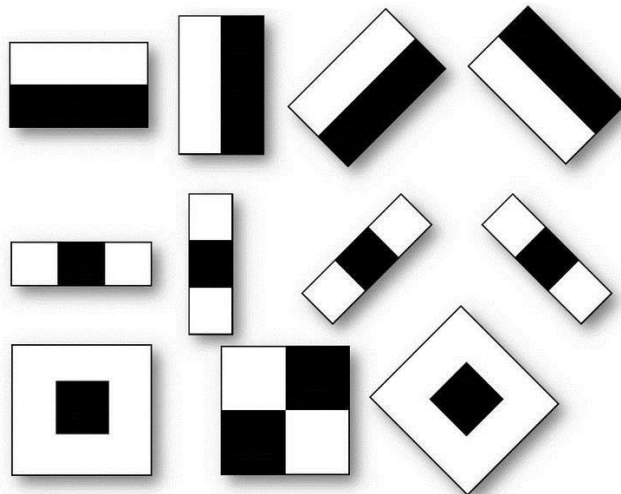


Рисунок 1.10 – Примитивы Хаара

Вычисляемым значением такого признака будет $F = x - Y$, где X – сумма значений яркостей точек закрываемых светлой частью признака, а Y – сумма значений яркостей точек закрываемых темной частью признака. Признаки Хаара дают точечное значение перепада яркости по оси X и Y соответственно [42].

Алгоритм бустинга

Алгоритм бустинга был опубликован в 1996 и послужил основой для всех последующих исследований в данной области. На данный момент наиболее распространёнными вариантами базового алгоритма являются Gentle AdaBoost и Real AdaBoost, превосходящие базовый алгоритм по своим характеристикам, но сохраняют все основные принципы. К основным достоинствам AdaBoost и его вариантов можно отнести высокую скорость работы, высокую эффективность распознавания, простоту реализации, общность.

Пусть требуется построить классифицирующую функцию $F : X \rightarrow Y$, где X – пространство векторов признаков, Y – пространство меток классов. Пусть в распоряжении имеется обучающая выборка $(x_1, y_1), \dots, (x_N, y_N)$. Где $x_i \in X$ вектор признаков, а $y_i \in Y$ метка класса, к которому принадлежит $x_i \in X$. Далее будем рассматривать задачу с двумя классами, то есть $Y = \{-1; +1\}$. Также есть семейство простых классифицирующих функций $H : X \rightarrow Y$. Будем строить финальный классификатор в следующей форме:

$$F(x) = \text{sign} \left[\sum_{m=0}^M \alpha_m h_m(x) \right],$$

где $\alpha_m \in R, h_m \in H$. Построим итеративный процесс, где на каждом шаге будем добавлять новое слагаемое $f_m = \alpha_m h_m(x)$, вычисляя его с учётом работы уже построенной части классификатора. На каждом шаге для каждого примера (x_i, y_i)

из обучающей выборки вычисляется его вес: положим $D_0(i) = \frac{1}{N}$, тогда

$$D_{m+1}(i) = \frac{D_m(i) \exp(-y_i f_m(x_i))}{Z_i},$$

где Z_i – нормализующий коэффициент, такой что $\sum_{i=1}^N D_{m+1}(i) = 1$.

Вес каждого элемента обучающей выборки на текущем шаге задает «важность» этого примера для очередного шага обучения алгоритма. Чем больше вес, тем больше алгоритм будет «стараться» на данном шаге классифицировать этот пример правильно. Варьируем веса таким образом, чтобы классификатор, включенный в комитет на текущем шаге, «концентрировался» на примерах, с которыми предыдущие шаги «не справились». Таким образом, на каждом шаге работаем с какой-то частью данных, плохо классифицируемой предыдущими шагами, а в итоге комбинируем все промежуточные результаты.

Очередной простой классификатор выбирается исходя из взвешенной с распределением D_m ошибки. Выбирается $h_m \in H$, минимизирующий взвешенную ошибку классификации:

$$e_m = \sum_{i=1}^N D_m(i) \cdot (h_m(x_i) \neq y_i).$$

Далее вычисляется вклад текущего слагаемого классифицирующей функции:

$$\alpha_m = \log \left(\frac{1 - e_m}{e_m} \right).$$

Процесс продолжается до некоторого шага M , номер которого определяется вручную [43].

1.5.2 Метод TLD

Метод надёжного длительного сопровождения заранее неизвестных объектов в естественной среде. Метод выдерживает разрывы между кадрами, быстрое движение камеры, полное исчезновение, а затем появление объекта. Подход, который использован в данном методе называется Сопровождение–Моделирование–Обнаружение (Tracking–Modeling–Detection (TMD)), он сочетает

адаптивное сопровождение объекта с обучением детектора объекта в процессе распознавания. После того как объект был захвачен при помощи какого-либо метода захвата, траектория объекта начинает наблюдаться двумя процессами (расширяющее и урезающее события). Они строят детектор объекта на лету. Оба процесса делают ошибки, стабильность системы достигается отменой событий. Метод TLD является методом без учителя, но процесс обучения всё же имеется. Обучение происходит «на лету», т. е. в процессе обнаружения объекта, и классификация производится при помощи рандомизированного леса.

Объект сопровождается при помощи краткосрочного трекера. Траектория в пространстве признаков анализируется двумя событиями, которые непрерывно пытаются расширить или уменьшить пространство, описываемое моделью. L_t расширяется измерениями, которые вероятнее всего содержат сопровождаемый объект. Эти измерения определяются при помощи расширяющего события. Из L_t удаляются измерения, которые определены как не содержащие объект при помощи обрезающих событий. События работают параллельно, стремясь достичь $L_t \rightarrow L^*$ (см. рисунок 1.11).

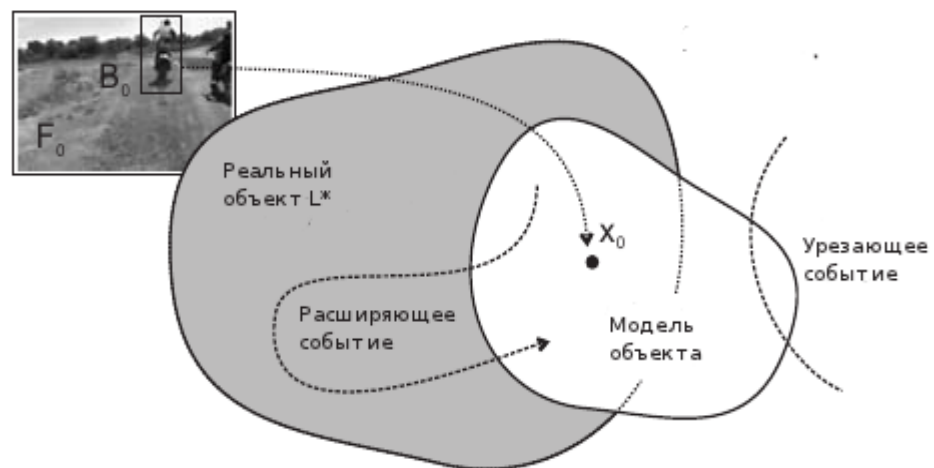


Рисунок 1.11 – Модель объекта инициализируется первым кадром. Она увеличивается расширяющими событиями, уменьшается урезающими событиями.

Постепенно она приближается к реальному объекту

Расширяющие события в одиночку приводят к высокому уровню примеси и, следовательно, к детектору низкой точности. Обрезающие события служат отрицательной обратной связью: чем выше уровень примесей, тем больше измерений в модели определяются как неверные и удаляются. Динамическое взаимодействие расширяющих и обрезающих событий является решающим в придании системы стабильности.

Краткосрочный трекер основывается на методе Лукаса–Канаде (ЛК) [44]. Вначале множество ключевых точек извлекается из прямоугольной решётки внутри описанного вокруг сопровождаемого объекта прямоугольника. Затем трекер сопровождает точки от одного кадра до другого, строя разреженное поле движения. Основываясь на поле движения, смещение и изменение масштаба ограничивающего прямоугольника могут быть надёжно оценены как средние значения по распределению. В каждом новом кадре сопровождается новый набор точек, это делает трекер адаптивным.

Детектор объекта основан на двухбитных бинарных шаблонах. Эти признаки измеряют ориентацию градиента внутри определённой зоны, квантуют её и выдают 4 возможных кода (см. рисунок 1.12) [7].

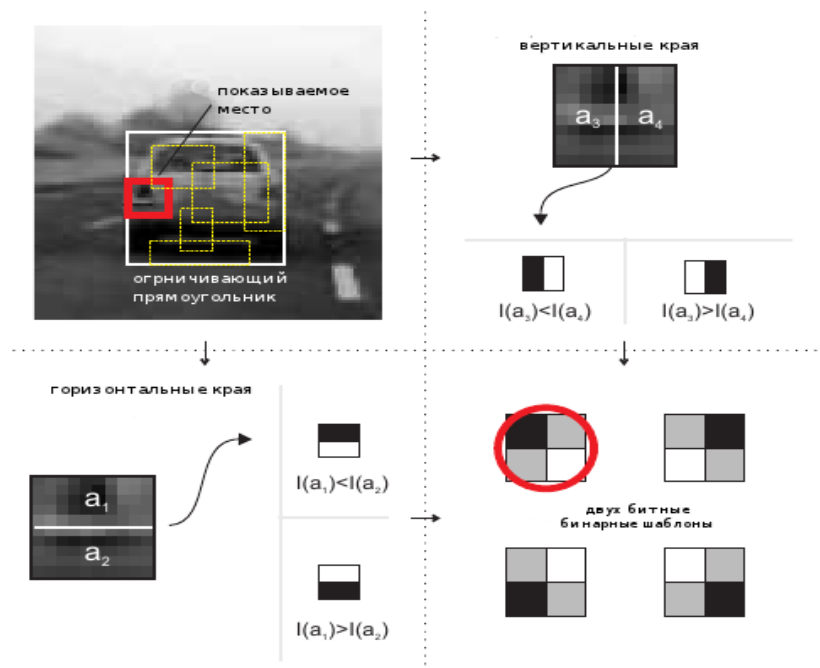


Рисунок 1.12 – Двухбитные бинарные шаблоны – признаки, используемые в детекторе объекта.

Таким образом, комбинированные методы имеют высокий уровень обнаружения, позволяют объединять достоинства разных методов поиска. В комбинированных методах сложной частью является соединение методов, т. к. необходимо минимизировать следующее:

$$\min_{j \in C} (O(Method_{combj}) - \max_{i \in N} (O(Method_i))),$$

где $O(x)$ – сложность метода x , $Method_{combj}$ – комбинированный j метод, $Method_i$ – i -ый «элементарный» метод (составная часть комбинированного), N – количество «элементарных» методов (составных частей комбинированного), C – количество всевозможных комбинаций объединения методов по сложности.

1.6 Проблемы обнаружения и слежения за объектом в видеопотоке

Можно выделить основные проблемы процесса обнаружения и слежения за объектом в видеопотоке:

1. Изменение яркости. Яркость изображения может меняться в пространстве и в течение времени: некоторые части объекта могут быть ярче других; яркие объекты на текущем изображении могут стать тёмными на следующем.

2. Масштабируемость объекта. Объект может приближаться и отдаляться от камеры.

3. Повороты объекта. Объект может относительно камеры поворачиваться по всем трём осям трёхмерного пространства.

4. Заслонение объекта. Искомый объект может перекрываться другими объектами.

5. Шум на изображении. Шум на изображении может возникнуть по разным причинам: естественным и искусственно созданным. К естественным можно отнести некорректную работу фото (видео) камеры и все природные условия, при которых происходит получение изображения (видео): снегопад, дождь... К искусственно созданным причинам можно отнести все те действия человека, которые направлены на изменение качества изображения: перекодирования

изображения в другое цветовое пространство; сжатия изображения с потерями качества, размытие изображения и т. д.

6. Изменение геометрической формы объекта. Наблюдаемый объект может деформироваться, повернуться к камере той частью, которая по своей форме является отличительной от формы, наблюдаемой прежде. Объект может состоять из мелких, связанных объектов, которые могут перемещаться в пространстве по разным траекториям. Например, искомый объект – человек, у которого руки могут двигаться в разные стороны.

7. Изменение цветовой гистограммы объекта. Объект может иметь на разных сторонах своего тела разные цветовые гистограммы [45].

Масштабируемость и повороты относятся к аффинным преобразованиям [46]. Согласно [47] проективные преобразования включают в себя аффинные преобразования и учитывают искажения, связанные с наклоном камеры.

1.7 Сравнение алгоритмов и методов выделения объекта в видеопотоке

Рассмотренные методы можно классифицировать на четыре группы: детерминированные, вероятностные, нейросетевые и комбинированные методы (см. рисунок 1.13).



Рисунок 1.13 – Классификация методов выделения объекта в видеопотоке

Согласно постановленной цели и описанным проблемам обнаружения и слежения за объектом в видеопотоке, можно выделить основные критерии для сравнения:

- устойчивость к изменению яркости;
- устойчивость к аффинным преобразованиям.
- устойчивость к заслонению объекта другими объектами;
- устойчивость к шуму на изображении;
- обучаемость к новой геометрической форме объекта;
- обучаемость к изменению цветовой гистограммы объекта.

Имеется ещё немало важный критерий сравнения алгоритмов и методов поиска объекта – быстродействие.

В таблице 1.2 приведены сравнительные характеристики рассмотренных методов и алгоритмов поиска объекта в видеопотоке, где **Y** – устойчивость к изменению яркости, **Sc** – устойчивость к масштабированию, **Rot** – устойчивость к поворотам, **Ovl** – устойчивость к заслонению другими объектами, **Ns** – шумоустойчивость, **Frm** – устойчивость к изменению геометрической формы, **G** –

устойчивость к изменению цветовой гистограммы, O_1 – сложность этапа обучения (если этап присутствует), O_f – сложность этапа обнаружения, n – количество точек на изображении, на котором осуществляется поиск, w – размер окна функции Гаусса, s – количество октав, x – количество точек в окрестности для вычисления ориентации, k – количество проективно искажённых изображений.

В таблице 1.2 не приведены методы поиска по шаблону, т. к. эти методы не осуществляют поиск определённого объекта, а только заточены под поиск определённых примитивов, поэтому в сравнении с другими методами методы поиска по шаблону не участвуют.

Таким образом, все рассмотренные методы в некоторых пределах устойчивы к шуму на изображении и к перекрытию объекта другими объектами. Из детерминированных методов самый устойчивый метод к рассмотренным критериям является метод ASIFT, однако он имеет большую вычислительную сложность. Из детерминированных методов стоит отметить методы SURF, Харриса Лапласа, учитывающий масштабирование изображения объекта, применённый совместно с дескриптором освещённости, имеют меньшую вычислительную сложность, чем метод ASIFT, однако они не являются полностью устойчивыми к аффинным преобразованиям. Вероятностные методы необходимо применять в случаях, когда необходимо добиться устойчивости по какому-то одному критерию. Можно применить несколько фильтров, устойчивых к изменению разных свойств объекта, но тогда и сложность этапа обнаружения геометрически возрастает. Методы, которые устойчивы к наиболее большому числу рассмотренных критериев, являются нейросетевые и комбинированные методы, но методы, основанные на классических и разреженных нейронных сетях, и метод Виолы–Джонса имеют длительный процесс обучения, гораздо сложнее этапа обнаружения [48].

Таблица 1.2 – Сравнение методов и алгоритмов поиска объекта в видеопотоке (+ – проблема решается полностью, +/- – проблема решается частично, – – проблема не решается)

Метод / алгоритм поиска		Критерии сравнения								
		Y	Sc	Rot	Ovl	Ns	Frm	G	Ol	Of
Метод Лукаса–Канаде (классический)		–	–	+/-	+/-	+/-	–	–	0	<i>n</i>
Ши–Томаси–Канаде		–	+	+	+/-	+/-	–	–	0	<i>n</i>
Джин–Фаваро–Соатто		+	–	+/-	+/-	+/-	–	+/-	0	<i>n</i>
Метод Харриса–Лапласа (классический)	Масштабирующий дескриптор	–	+	–	+/-	+/-	+/-	–	0	<i>n</i> ²
	Дескриптор поворота	–	–	+/-	+/-	+/-	+/-	–	0	<i>n</i> ²
	Дескриптор освещённости	+	–	–	+/-	+/-	+/-	–	0	<i>n</i> ²
Метод Харриса–Лапласа, учитывающий масштабирование объекта	Масштабирующий дескриптор	–	+	–	+/-	+/-	+/-	–	0	<i>n</i> ²
	Дескриптор поворота	–	+	+/-	+/-	+/-	+/-	–	0	<i>n</i> ²
	Дескриптор освещённости	+	+	–	+/-	+/-	+/-	–	0	<i>n</i> ²

Продолжение таблицы 1.2

Метод / алгоритм поиска		Критерии сравнения								
		Y	Sc	Rot	Ovl	Ns	Frm	G	Ol	Of
SIFT		+/-	+	+/-	+/-	+/-	+/-	-	0	$(sw^2+x^2)n$
SURF		+/-	+	+/-	+/-	+/-	+/-	-	0	sn
ASIFT		+/-	+	+	+/-	+/-	+/-	-	0	$k(sw^2+x^2)n$
Фильтр Калмана	Учитывающий изменения яркости	+	-	-	+/-	+/-	-	-	0	n^2
	Учитывающий изменения цветности	-	-	-	+/-	+/-	-	+	0	n^2
	Учитывающий изменение масштаба	-	+	-	+/-	+/-	+/-	-	0	n^2
	Учитывающий повороты объекта	-	-	+/-	+/-	+/-	+/-	-	0	n^2
Фильтр частиц	Учитывающий изменения яркости	+	-	-	+/-	+/-	-	-	0	n^2
	Учитывающий изменения цветности	-	-	-	+/-	+/-	-	+	0	n^2
	Учитывающий изменение масштаба	-	+	-	+/-	+/-	+/-	-	0	n^2
	Учитывающий повороты объекта	-	-	+/-	+/-	+/-	+/-	-	0	n^2

Продолжение таблицы 1.2

Метод / алгоритм поиска	Критерии сравнения								
	Y	Sc	Rot	Ovl	Ns	Frm	G	OI	Of
Методы, основанные на классических нейронных сетях	+/-	+/-	+/-	+/-	+/-	-	+/-	$n^3 \div n^5$	n^2
Методы, основанные на разреженных нейронных сетях	+/-	+/-	+/-	+/-	+/-	-	+/-	$n^3 \div n^5$	$n \log(n)$
Метод Виолы–Джонса	+/-	+	+/-	+/-	+/-	+/-	+	$n^5 \div n^7$	n
TLD	+/-	+	+/-	+/-	+/-	+/-	+/-	0	n^2

1.8 Выводы

На основании проведённого аналитического обзора можно сделать следующие выводы:

1. В настоящее время не реализованы помехоустойчивые, пригодные для практического использования алгоритмы (и методы) поиска объекта в видеопотоке.

2. Методы, основанные на поиске ключевых точек объекта, более всего удовлетворяют требованиям практики. Алгоритмы, построенные на базе этих методов, инвариантны к проективным преобразованиям, устойчивы к шуму, изменению яркости и характеризуются меньшими вычислительными затратами по сравнению с альтернативными.

3. Анализ условий использования методов поиска объекта показывает, что эти методы должны удовлетворять следующим требованиям:

- инвариантность к проективным преобразованиям изображения объекта;
- вычислительная сложность должна быть минимально достижимой для применения решения задач в режиме реального времени.

2 Метод и алгоритм поиска объекта в видеопотоке

2.1 Определение требований

На основании сформулированной цели и проведённого аналитического обзора можно определить требования к разрабатываемой системе поиска объекта в видеопотоке. Эти требования можно разделить на две группы: основные (функциональные) и дополнительные (структурные). К основным можно отнести требования, предъявляемые к технологии поиска и алгоритмам системы, а к дополнительным – требования, предъявляемые к структуре системы.

Основные требования к системе поиска объекта в видеопотоке формулируются на основании результатов аналитического обзора и потребностей практического использования:

1. Алгоритм системы должен быть основан на поиске ключевых точек объекта.

2. Вычислительная сложность метода поиска должна быть минимально достижимой для применения в решении задач реального времени.

3. Алгоритм работы системы не должен содержать шаг предварительного обучения.

4. Система должна позволять управлять способом обработки результата: сохранять результат в виде текстовой информации, в виде набора изображений с выделенным найденным объектом, либо выводить на экран набор кадров с выделенными объектами.

Исходя из требований скорости обработки видеопотока и простоты реализации, структурные требования могут быть сведены к следующему: модель структуры системы должна быть с минимальным количеством статических связей. Из этого следует, что желаемая структура системы должна быть статической и сетевой.

2.2 Функциональная модель

Согласно сформулированным требованиям алгоритм системы должен быть основан на поиске ключевых точек. Из проведённого обзора технической литературы следует, что методы поиска ключевых точек имеют следующие особенности, которые необходимо учитывать при использовании:

1. Ключевая точка может не являться частью изображения только одного объекта, она может встречаться и на изображениях других объектов, так как ключевая точка является локальным участком на изображении сравнительно небольшого размера и на изображении может иметь «схожие» дубликаты за счёт малой информативности.

2. Ключевые точки могут находиться не на всей области изображения объекта, а только в некоторых областях (см. рисунок 2.1 [49]), т. е. по найденным ключевым точкам нельзя судить о местонахождении всего изображения объекта, поэтому определение место положения объекта на изображении, основываясь только на положениях ключевых точек, является некорректным.

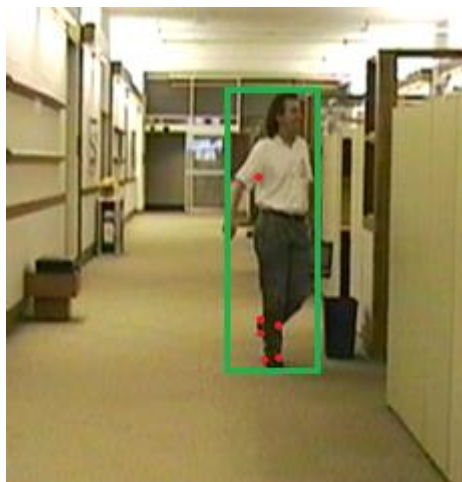


Рисунок 2.1 – Пример расположения найденных локальных особенностей

Наличие локальных признаков не гарантирует надёжную идентификацию из-за возможности повторяемости признаков на других объектах и возможности искажения. В этой связи для повышения надёжности идентификации необходимо расширить вектор локальных признаков одним из глобальных, позволяющего

идентифицировать объект. Из глобальных признаков следует выделить цветовую гистограмму, так как цветовая гистограмма вычисляется быстро и не зависит от поворота и масштаба изображения.

Предложена модель представления изображения объекта в виде совокупности интеральных и локальных признаков изображения объекта:

$$F(x, y) \rightarrow \begin{cases} K^{ref} = \{K_0^{ref}, \dots, K_{m-1}^{ref}\} \\ H^{ref} = \{H_0^{ref}, \dots, H_{n-1}^{ref}\} \end{cases},$$

где $F(x, y)$ – прямоугольная матрица значений интенсивности пикселей, K^{ref} – вектор ключевых точек изображения объекта размерности m , H^{ref} – цветовая гистограмма изображения объекта, состоящей из n элементов.

Для повышения быстродействия предлагается перед началом анализа видеопотока осуществить следующие действия:

1. Создать набор изображений с проективными искажениями образца.
2. Найти на каждом созданном изображении ключевые точки.
3. Вычислить дескрипторы каждой ключевой точки.

Описанные шаги по вычислению дескрипторов ключевых точек на предварительно проективно искаженных изображениях образца позволяют использовать алгоритмы для поиска и описания ключевых точек, которые являются неустойчивыми к проективным преобразованиям.

Таким образом, общая схема алгоритма поиска может быть представлена в виде двух этапов [50]:

1. Быстрый поиск областей претендентов с изображением объекта с помощью ключевых точек.
2. Идентификация объекта на изображении на основе глобального признака – цветовой гистограммы.

На основании этой схемы, сформулированных требований к системе и предложенному методу вычисления дескрипторов на проективно искажённых изображениях образца можно определить набор функций, которыми должна

обладать система, провести функциональное проектирование. Функциональная модель описывается на основании нотации IDEF0 [51].

2.2.1 Диаграмма верхнего уровня

При рассмотрении системы с точки зрения функции верхнего уровня выделены следующие положения (см. рисунок 2.2):

- базовой функцией системы является нахождение объекта в видеопотоке;
- входными потоками системы являются изображение объекта и видеопоток;
- механизмом, при помощи которого функционирует система, является программа;
- выходными данными являются набор изображений объекта в видеопотоке.



Рисунок 2.2 – Контекстная диаграмма верхнего уровня

2.2.2 Диаграмма первого уровня

В результате первичной декомпозиции диаграммы верхнего уровня А-0 получена диаграмма А0 (см. рисунок 2.3).

Декомпозиция функции верхнего уровня системы выявляет следующие основные функции:

- функция вычисления вектора дескрипторов по многомерной матрице проективно искажённых изображений объекта (А1);
- функция вычисления гистограммы изображения объекта (А2);
- функция извлечения кадра из видеопотока (А3);
- функция вычисления вектора ключевых точек кадра видеопотока (А4);
- функция нахождения областей претендентов объекта на кадре (А5);
- функция идентификации объекта в найденных областях претендентов на основании вычисления цветowych гистограмм (А6).

На диаграмме А0 выделены основные информационные потоки:

1. Поток, содержащий изображение искомого объекта.
2. Видеопоток. Данный поток включает в себя набор кадров, на которых осуществляется поиск объекта.

В функции идентификации по выделенным областям кадра находится область с ближайшим соответствием гистограммы изображения искомого объекта.

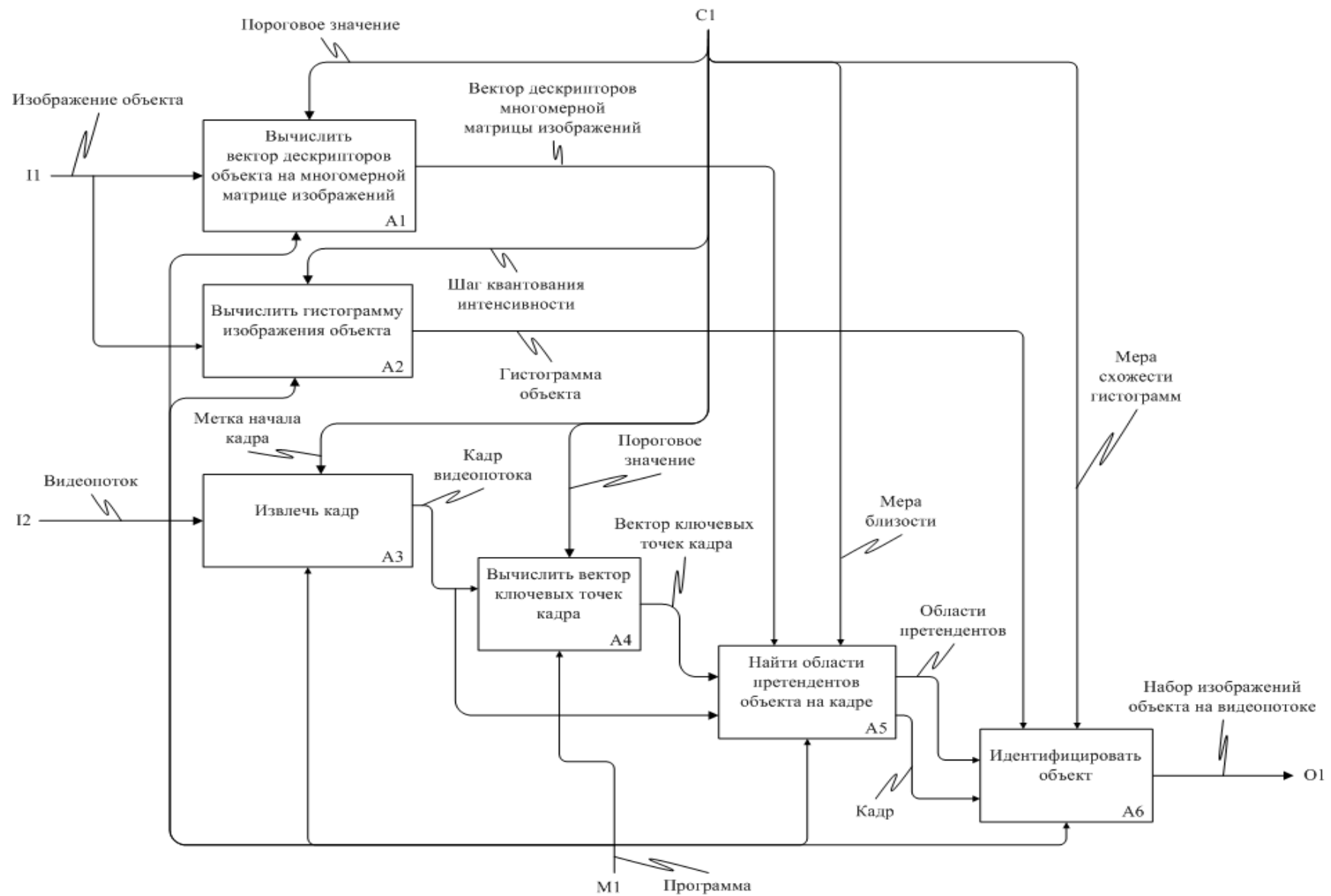


Рисунок 2.3 – Контекстная диаграмма первого уровня

2.2.3 Диаграммы второго уровня

Функция вычисления вектора дескрипторов многомерной матрицы изображений объекта A1 декомпозирована на этапы (см. рисунок 2.4), механизмом которых является программа:

1. Преобразование изображения в полутоновое (A11). Управлением осуществляется формулой преобразования изображения в полутоновое.

2. Генерация изменений масштаба (A12) согласно управлению – параметрам масштаба. На этом этапе создаётся вектор масштабированных изображений объекта.

3. Генерация изменений наклона камеры (A13) согласно параметрам наклона. Этот этап создаёт матрицу изображений из вектора масштабированных изображений изменением наклона камеры.

4. Генерация изменения поворота объекта согласно параметрам поворота (A14). Этот этап создаёт трёхмерный массив изображений из матрицы масштабированных изображений изменением угла поворота изображения объекта. Выходом функции является трёхмерная матрица изображений объекта.

5. Нахождение координат ключевых точек изображения объекта (A15). Ключевая точка описывается координатами областей изображения и дескриптором. На этом шаге находятся координаты уникальных ключевых точек. Выходом функции является вектор координат с радиусом окружностей ключевых точек.

6. Вычисление дескрипторов ключевых точек (A16) на трёхмерной матрице изображений объекта. Дескриптор ключевой точки – это вектор параметров, вычисленных на изображении в области ключевой точки. Ключевая точка описывается координатами центра и радиусом окружности. Управление осуществляется вектором координат центра и радиусом окружностей ключевых точек, и пороговым значением. Выходом является вектор дескрипторов изображений объекта.

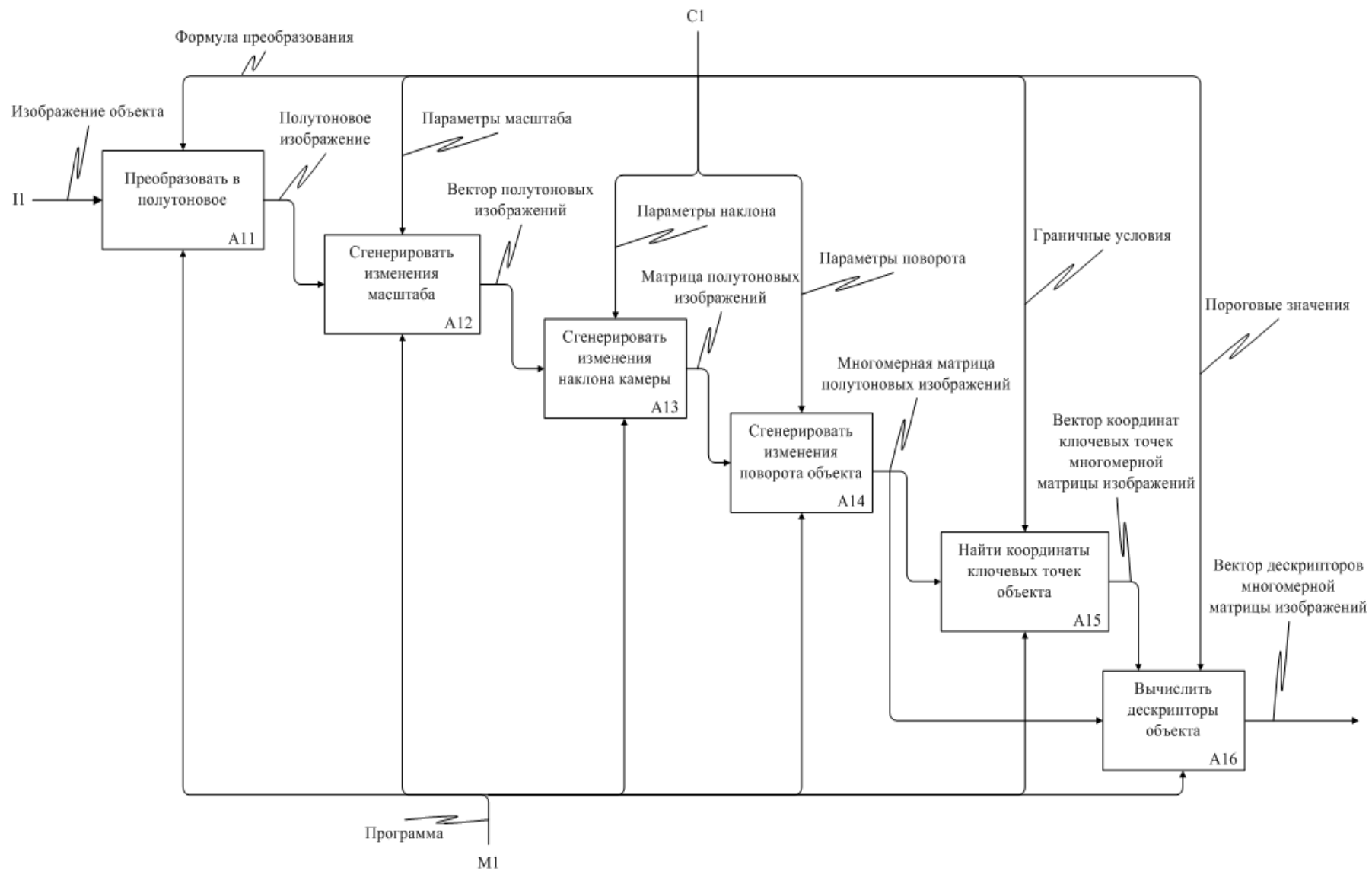


Рисунок 2.4 – Диаграмма функции вычисления вектора дескрипторов изображения объекта (A1)

Описанные этапы A11, A12, A13, A14, A15 и A16 – это шаги метода вычисления дескрипторов на проективно искажённых изображениях образца.

Результат декомпозиции функции извлечения кадра из видеопотока A3 представлен на рисунке 2.5. В процессе выполнения функции видеопоток проходит через следующие этапы:

1. Демультимпликация. Этот этап необходим для извлечения закодированного видеосигнала из контейнера (A31).

2. Декодирование. Этап необходим для представления закодированного видеосигнала в набор изображений, удобных для обработки в функции поиска объекта в видеопотоке (A32).

3. Выделение кадра. На этом этапе согласно параметрам кадра из декодированного видеопотока, извлекается кадр.

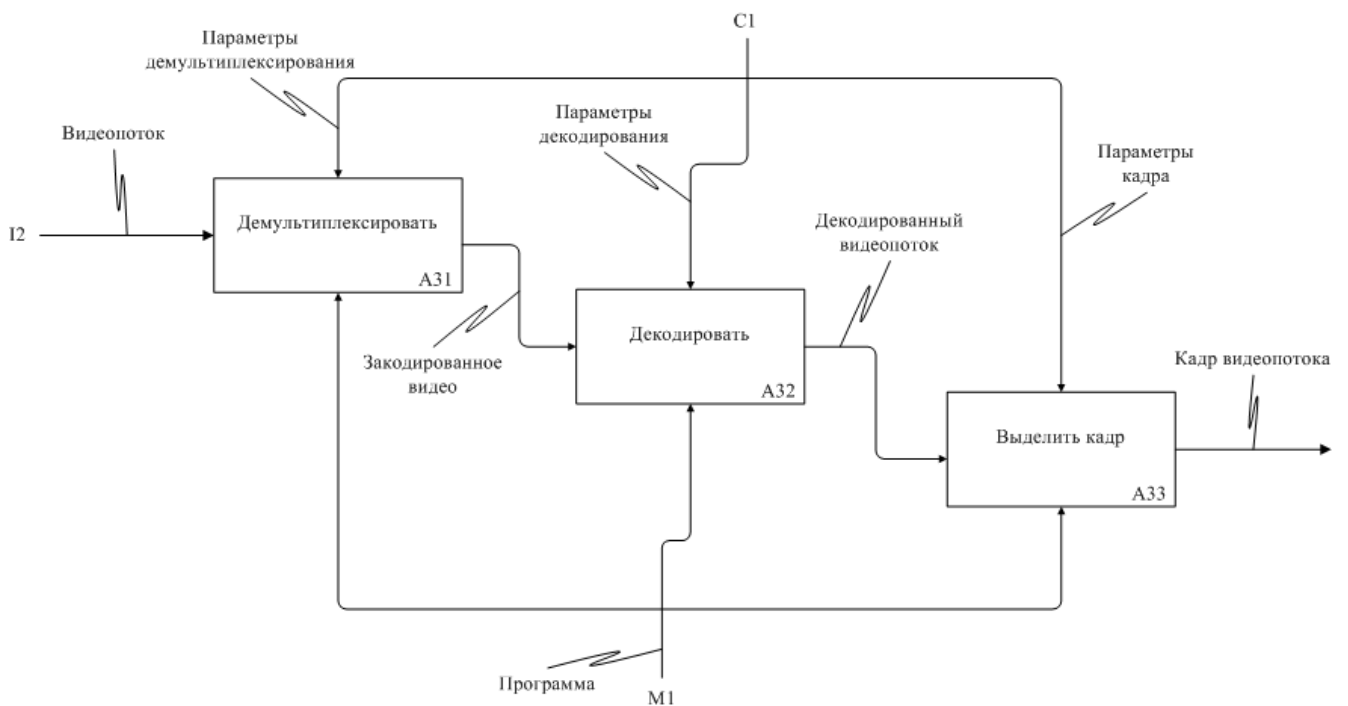


Рисунок 2.5 – Диаграмма функции извлечения изображения из видеопотока (A3)

Функция вычисления вектора ключевых точек кадра A4 декомпозирована на следующие этапы (см. рисунок 2.6):

1. Преобразование в полутоновое (A41);
2. Нахождение координат ключевых точек кадра (A42);
3. Вычисление дескрипторов в найденных координатах кадра (A43).

Вектор координат ключевых точек необходим для вычисления дескрипторов (A43). Выходом функции является вектор ключевых точек с информацией о номере кадра, координатах и дескрипторе по каждой ключевой точке.

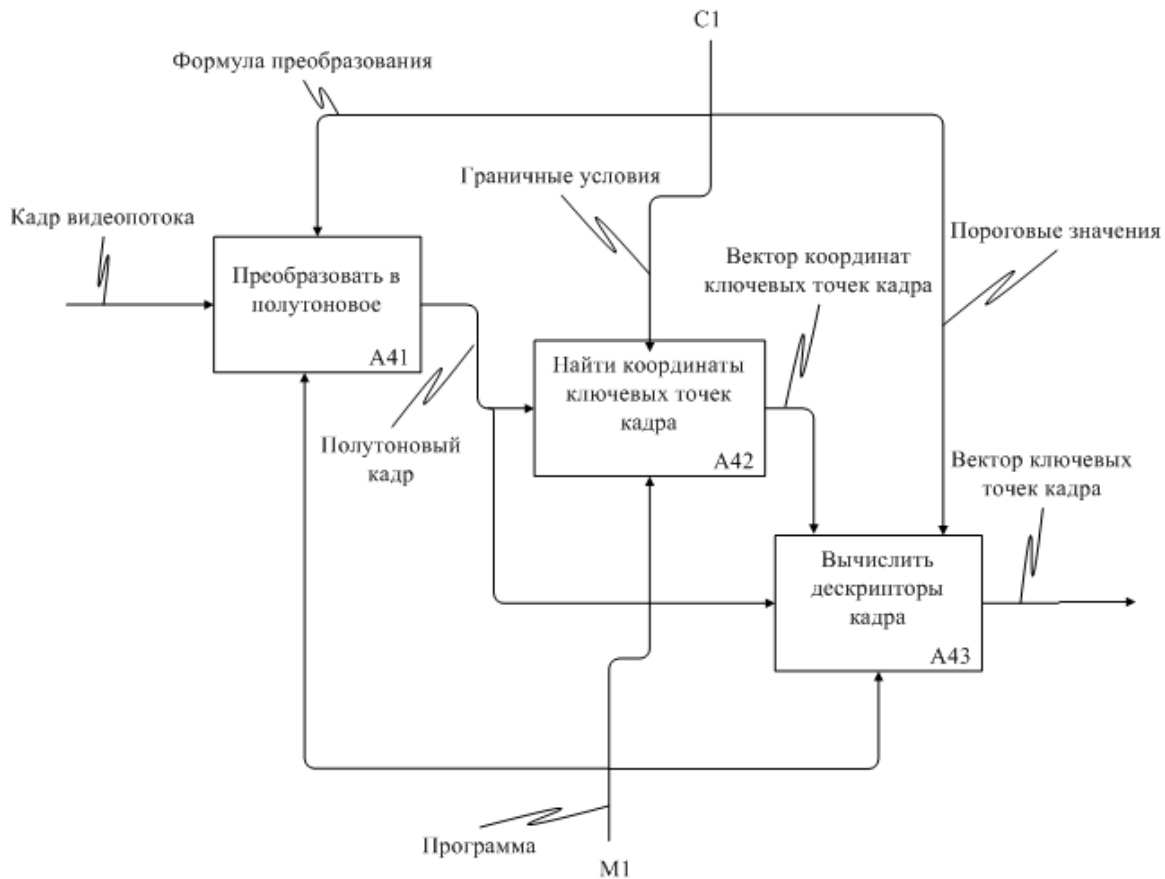


Рисунок 2.6 – Диаграмма функции вычисления вектора ключевых точек кадра (A4)

Функция нахождения областей претендентов объекта на кадре (A5) декомпозирована на составляющие (см. рисунок 2.7):

1. Нахождение общих ключевых точек (A51). На вход поступает вектор ключевых точек кадра. Управление осуществляется вектором дескрипторов объекта. Используя меру близости, определяются ключевые точки, дескрипторы которых считаются одинаковыми с дескрипторами изображения объекта, т. е. изображения областей, на которые указывают ключевые точки, считаются

«похожими». В случае получения непустого множества после пересечения, выводятся общие ключевые точки и информация о номере кадра.

2. Пропускание кадра (A52). Согласно информации о номере кадра принимается решение о пропуске входного кадра видеопотока для дальнейшей обработки.

3. Получение областей претендентов (A53). Входной вектор координат ключевых точек кадра объединяется согласно размеру окна в области, которые подаются на выход функции.

Центральными функциями системы являются функции нахождения ключевых точек изображения объекта (A15) и кадра (A42), вычисления дескрипторов ключевых точек изображения объекта (A16) и кадра (A43), нахождения общих ключевых точек по дескрипторам (A51) и идентификации кадра на предмет наличия изображения объекта (A6). От результата выполнения этих функций зависит качество и скорость обнаружения объекта в видеопотоке. Далее рассматриваются подробнее алгоритмы нахождения ключевых точек совместно с вычислением дескрипторов, алгоритмы нахождения пересечения дескрипторов и алгоритмы, позволяющие идентифицировать изображение объекта в области кадра по вычислению гистограммы.

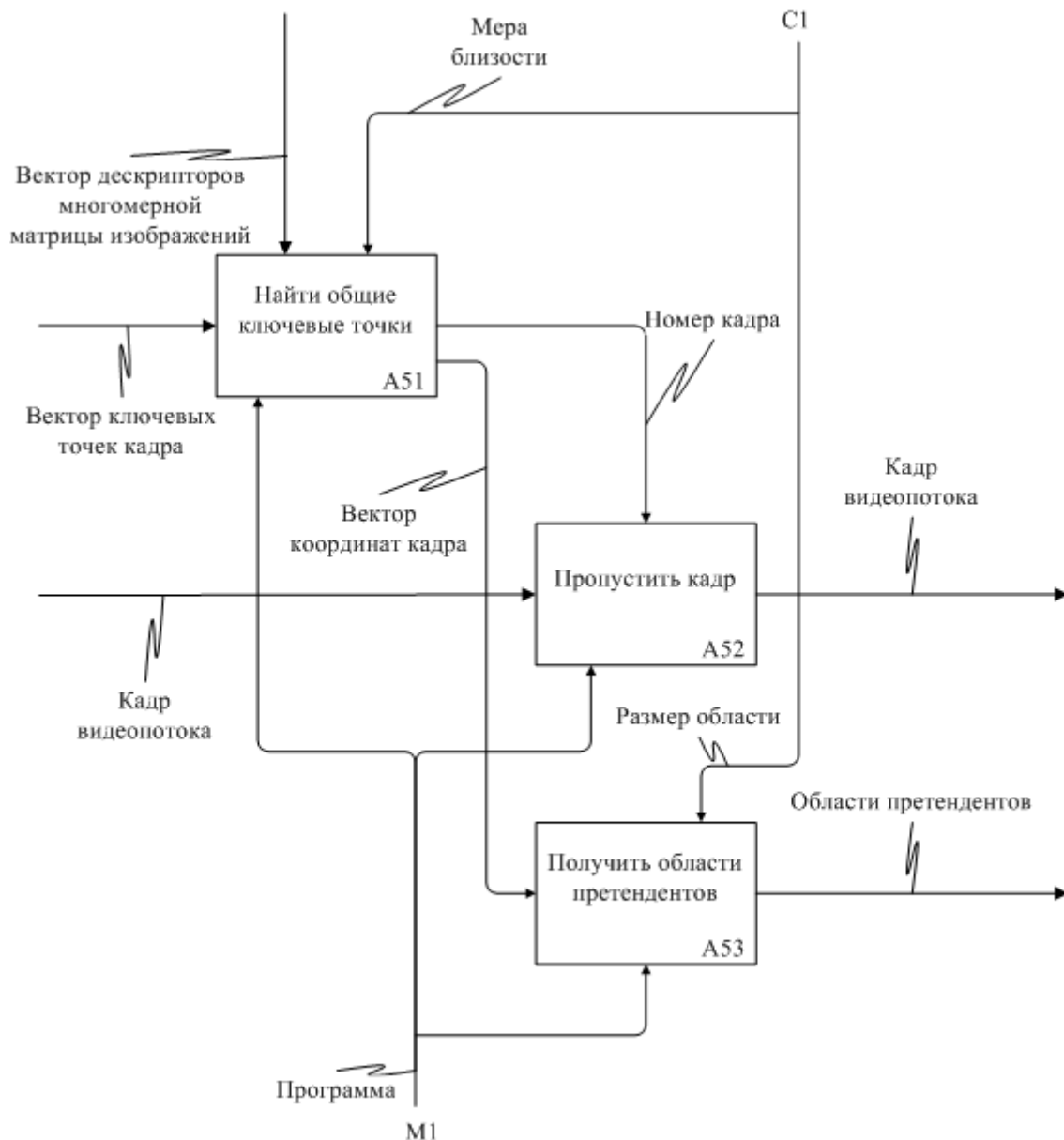


Рисунок 2.7 – Диаграмма функции нахождения областей претендентов объекта на кадре (A5)

2.3 Выбор процедуры нахождения ключевых точек и вычисления дескрипторов

Согласно требованию к системе, алгоритм должен быть основан на поиске ключевых точек объекта. Проведенные исследования показали, что метод ASIFT является наиболее устойчивым к рассмотренным критериям. ASIFT основан на методе SIFT, у которого существует быстроедейственный аналог – метод SURF.

На рисунке 2.8 представлен пример обнаружения объекта методами ASIFT, SIFT и SURF. Метод ASIFT является наиболее устойчивым к рассмотренным критериям, но обладает высокой вычислительной сложностью. Алгоритм ASIFT есть усложнение метода SIFT, позволяющее достичь устойчивости ко всем аффинным преобразованиям благодаря моделированию изменений наклона камеры. Согласно построенной функциональной модели генерацию изменений наклона камеры осуществляет функция A12, тем самым метод ASIFT становится избыточным по нахождению особых точек.

В SIFT ключевой точкой считается локальный экстремум в масштабируемом пространстве разности гауссиан. В быстросейственном аналоге – методе SURF [52] ключевой точкой является локальный экстремум детерминанта матрицы Гессе. На практике алгоритм SURF выделяет меньше ключевых точек на изображении объекта, но имеет высокую скорость обработки кадра по сравнению с методом SIFT [53]. Для быстрого нахождения ключевых точек и вычисления дескрипторов за основу предложено использовать метод SURF.

Учитывая построенную функциональную модель, устойчивость к масштабированию в системе достигается за счёт выполнения функции A13 над изображением искомого объекта, поэтому метод SURF модернизирован: поиск ключевых точек выполняется только на одной октаве. В связи с этим вычислительная сложность модернизированного метода уменьшается в s раз (см. таблицу 1.1), где s – количество октав.

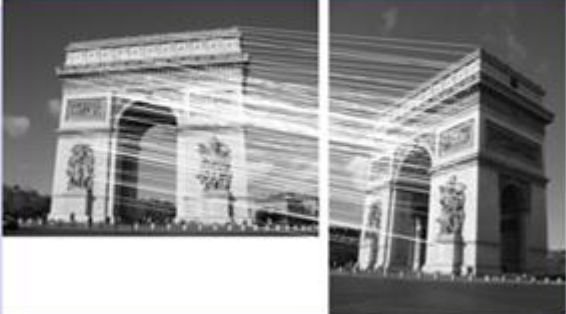


Метод	Пример обнаружения	
ASIFT		
SIFT		
SURF		

Рисунок 2.8 – Пример обнаружения объекта методами ASIFT, SIFT, SURF

2.4 Алгоритмы и методы нахождения пересечения дескрипторов

Для нахождения пересечения двух дескрипторных множеств на сегодняшний день активно используются следующие подходы:

- метод RANSAC;
- алгоритм Куна – Манкреса.

2.4.1 RANSAC

RANSAC – это общий метод, который используется для оценки параметров модели на основании случайных выборок. При сопоставлении модель представляет собой матрицу преобразования (гомография). На входе алгоритма имеется два множества дескрипторов. Схема работы RANSAC состоит из многократного повторения трех этапов:

1. Выбор точек и построение параметров модели. Из входных множеств дескрипторов выбираются случайным образом без повторений наборы фиксированного размера. На основании полученных наборов строится матрица преобразования.

2. Проверка построенной модели. Для каждого дескриптора изображения объекта находится проекция на текущем кадре и выполняется поиск наиболее близкого дескриптора из множества дескрипторов текущего кадра. Дескриптор помечается как выброс, если расстояние между проекцией и соответствующим дескриптором текущего изображения больше некоторого порога.

3. Замещение модели. После проверки всех точек проверяется, является ли построенная модель лучшей среди набора предшествующих моделей. В результате применения RANSAC строится наилучшая матрица гомографии. Вычислив перспективную проекцию набора дескрипторов изображения объекта, достаточно выполнить проход по всем соответствиям, полученным в процессе перебора, и проверить, является ли соответствующий дескриптор текущего кадра достаточно близким к проекции дескриптора изображения объекта. Если не является, то пара отбрасывается [54].

Согласно [55] для одной модели вычислительная сложность составит $O(n)$, однако на практике результаты являются не приемлемыми для использования за счёт большого количества возможных ошибок [56]. Существуют модификации метода RANSAC. Например, алгоритм Джи-Линкейджа и алгоритм адаптации ядра, позволяющие находить пары с меньшим количеством ошибок, но обладающие вычислительной сложностью $O(n^2)$ [56]. Алгоритм масштабно

сжатого вектора Фишера с RANSAC (SCFV-RANSAC) [57] аналогичным образом засчёт дополнительной обработки набора дескрипторов для сопоставления имеет меньше количество ошибок.

2.4.2 Алгоритм Куна-Манкреса

Задачу сопоставления дескрипторов можно представить в виде задачи о назначениях. Интерпретируем в графовый вид. Пусть параметры масок (дескрипторы) – вершины графа, а значения меры схожести вершин – рёбра этого графа. Сложность оригинального алгоритма составляет $O(n^4)$. Для решения задачи методом Куна-Манкреса необходимо добавить новые виртуальные вершины графа, которые будут бесконечно удалены от других вершин. Тогда $K_{n,n}[W]$ – взвешенный граф с долями X и Y . Выходом метода является множество ребер оптимального паросочетания P в данном графе.

Метод Куна – Манкреса можно представить в виде следующих последовательных операций:

1. Задать в $K_{n,n}[W]$ произвольную допустимую разметку f и найти подграф равенств $G_{W,f}$.
2. Венгерским алгоритмом найти максимальное паросочетание P в графе $G_{W,f}$ и множество F свободных относительно P вершин доли X .
3. Если $F = 0$, закончить работу.
4. Найти все чередующиеся цепи в графе $G_{W,f}$, начинающиеся в F , положить S и T равными множеству всех вершин доли X (соответственно, доли Y), встретившихся в этих цепях.
5. Если в T нет свободных вершин, положить

$$\Delta = \min_{x_i \in S, y_j \in Y \setminus T} \{f(x_i) + f(y_j) - w_{ij}\},$$

где $f(x) = f(x) - \Delta$ для всех $x \in S$, $f(y) = f(y) + \Delta$ для всех $y \in T$, найти новый граф $G_{W,f}$ и вернуться на шаг 4.

6. Увеличить P , перекрасив найденную увеличивающую цепь, и вернуться на шаг 3 [58].

2.4.3 Алгоритм ограничения области поиска объекта в кадре

Алгоритм ограничения области поиска объекта оценивает масштаб изображения объекта по дескрипторам ключевых точек по следующей схеме:

1. Найти для каждой ключевой точки кадра ближайшее соответствие из набора проективно искажённых изображений образца.

2. Убрать из дальнейшего рассмотрения ключевые точки кадра, имеющие значение меры близости ниже порога Thr .

3. Для каждой оставшейся ключевой точки кадра построить прямоугольную область. Координаты выделенной области на изображении определяются по координатам соответствующей ключевой точки на искажённом изображении образца.

4. Из множества ключевых точек для дальнейшего анализа оставить только те, прямоугольные области которых имеют площади пересечений с другими прямоугольными областями меньше, чем половина площади прямоугольной области рассматриваемой ключевой точки.

Более подробно алгоритм представлен в виде блок-схемы на рисунках 2.9 и 2.10, построенных согласно стандарту ЕСПД 19.701-90 «Схемы алгоритмов, программ, данных и систем» [60], где Thr – порог меры близости дескрипторов.



Рисунок 2.9 – Блок-схема алгоритма ограничения области поиска объекта в кадре

Мера близости между дескрипторами кадра и изображением образца вычисляется по коэффициенту Бхаттачария [59]:

$$p = \sum_{i=0}^{n-1} \sqrt{a_i \cdot b_i},$$

где a, b – векторы размерности n , p – мера близости, $p \in [0, 1]$.

Предложенный алгоритм имеет меньше вычислительную сложность, чем алгоритмы RANSAC и Куна-Манкреса. Однако недостатком такого подхода является определение значения порога Thr .

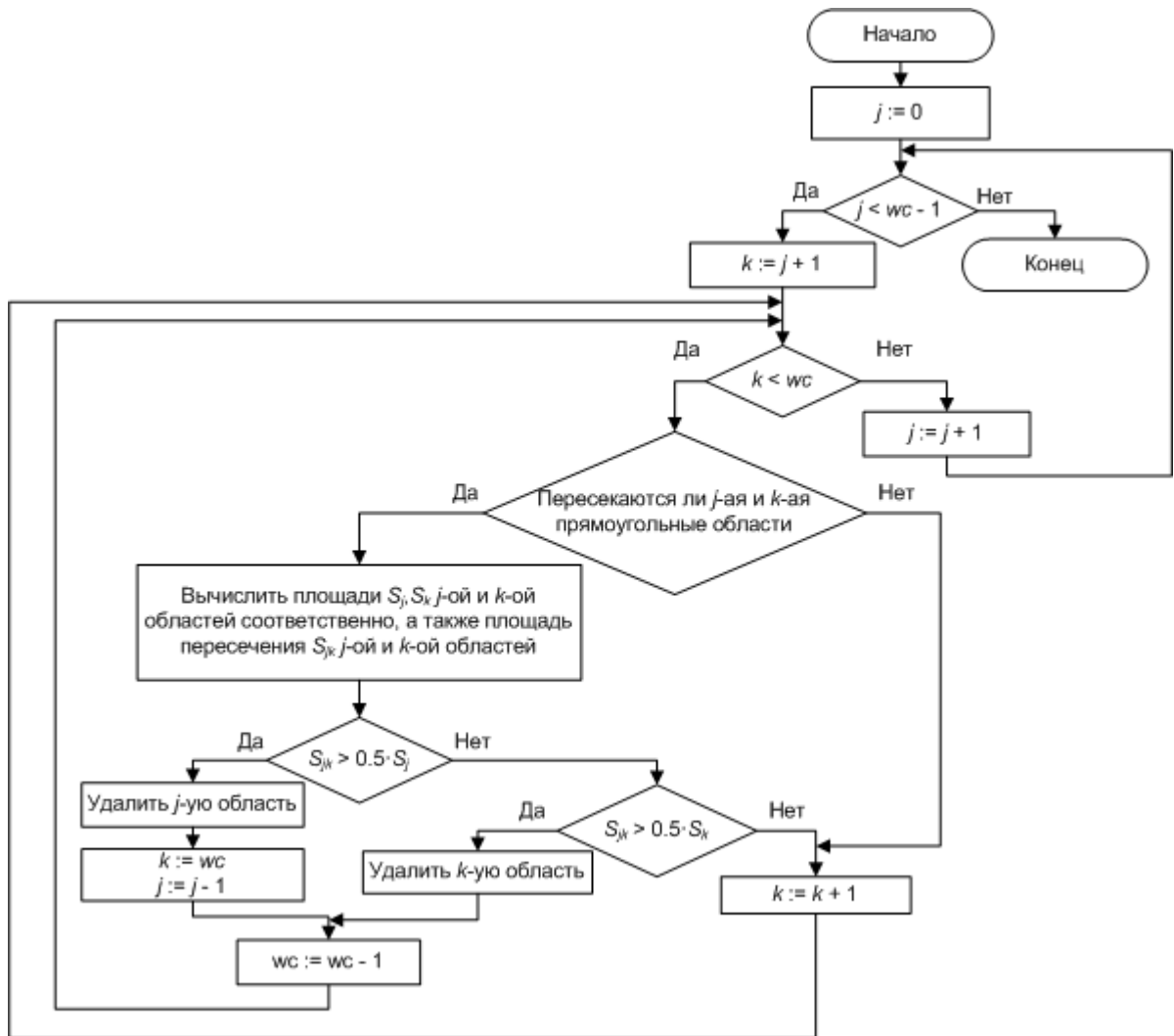


Рисунок 2.10 – Блок-схема процесса удаления прямоугольных областей поиска объекта

2.5 Алгоритм идентификации области изображения

Алгоритм идентификации должен определять, является ли область на кадре изображением или частью изображения объекта. Для этого алгоритм должен найти параметры окна на кадре по найденным областям, полученными на основании сопоставления локальных признаков изображения – ключевых точек. Пусть алгоритм идентификации находит объект эллиптическим окном. В алгоритме предлагается использовать метод, основанный на глобальном свойстве изображения. Одной из самых распространённых глобальных характеристик

является цветовая гистограмма [61]. Цветовая гистограмма вычисляется быстро, однако при вычислении, не учитывается пространственное расположение пикселей. Предлагается значения цвета точек вносить с определённым весом: чем ближе точка к центру окна, тем больше её вес. Это необходимо и для того, чтобы небольшие смещения окна приводили к небольшим изменениям ошибки сопоставления. Такому условию соответствует ядро Епонечникова [62]:

$$K(x) = \begin{cases} 1 - x^2, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases}.$$

Таким образом, цвет пикселя x будет внесён в цветовую гистограмму с определённым весом $K(x)$.

В основу идентификации объекта предлагается использовать технику Mean Shift [63]. Mean Shift основан на поиске максимума плотности вероятности некоторой функции, которая описывает дискретные данные.

Для локализации объекта предлагается использовать градиентный спуск. В качестве критерия схожести предлагается использовать коэффициент Бхаттачария [59].

2.5.1 Градиентный спуск

Градиентный спуск применяется для решения задачи нахождения локального минимума [64]. Предлагается использовать четырёхпараметрический поиск окна изображения объекта (см. рисунок 2.11) [65]. Для устойчивого процесса идентификации к незначительным изменениям цвета и для уменьшения размера гистограммы производится квантование значений цветовой гистограммы.

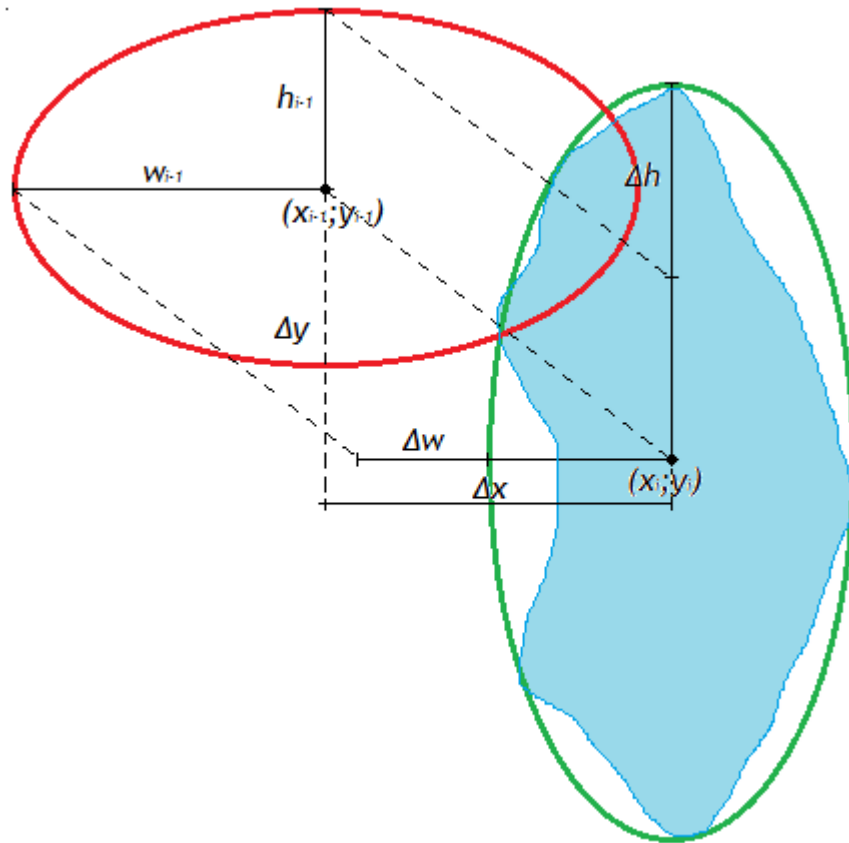


Рисунок 2.11 – Четырёх параметрическая модель поиска

Четырёх параметрический поиск окна изображения объекта методом градиентного спуска состоит из следующих шагов:

1. Задать ρ_{th} (минимальное значение порога схожести), i_{max} (максимальное количество итераций), w_0 и h_0 (длины полуосей эллипса).
2. Вычислить нормированный вектор частот значений интенсивностей (гистограмму) $hist_0$ в эллиптической области (x_0, y_0, w_0, h_0) .
3. Вычислить меру схожести при помощи коэффициента Бхаттачария:

$$\rho_0 = \sum_{b \in B} \sqrt{hist_{refb} \cdot hist_{0b}},$$

где $hist_{ref}$ – гистограмма изображения объекта для поиска, b – шаг гистограммы, $b \in B$.

4. $i = 1$.
5. Пока $\rho_{i-1} < \rho_{th}$ и $i < i_{max}$ выполнить:
 - 5.1 Вычислить градиент $grad\rho_{i-1}$:

$$\text{grad } \rho_{i-1} = \left(\frac{\partial \rho_{i-1}}{\partial x}, \frac{\partial \rho_{i-1}}{\partial y}, \frac{\partial \rho_{i-1}}{\partial w}, \frac{\partial \rho_{i-1}}{\partial h} \right)$$

5.2 Вычислить k_{i-1} :

$$k_{i-1} \approx \frac{1 - \rho_{i-1}(x_{i-1}, y_{i-1}, w_{i-1}, h_{i-1})}{|\text{grad } \rho_{i-1}|^2}.$$

5.3 Вычислить величину шага:

$$(\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i) = k_{i-1} \cdot \text{grad } \rho_{i-1}.$$

5.4 Изменить параметры эллипса:

$$(x_i, y_i, w_i, h_i) = (\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i) + (x_{i-1}, y_{i-1}, w_{i-1}, h_{i-1}).$$

5.5 Рассчитать относительную гистограмму $hist_i$ эллиптической области (x_i, y_i, w_i, h_i) .

5.6 Вычислить меру схожести ρ_i :

$$\rho_i = \sum_{b \in B} \sqrt{hist_{refb} \cdot hist_{ib}}.$$

5.7 $i = i + 1$, перейти на шаг 5.

6. Стоп.

Начальные значения w_0 и h_0 берутся из параметров рассматриваемой области P , полученной после алгоритма ограничения области поиска объекта в кадре.

Цветовая гистограмма вычисляется по цветоразностным компонентам U и V цветового пространства YUV [66]:

$$\begin{aligned} U &= -0.14713 \cdot R - 0.28886 \cdot G + 0.436 \cdot B + 128 \\ V &= 0.615 \cdot R - 0.51499 \cdot G - 1.0001 \cdot B + 128 \end{aligned}$$

где R, G, B – 8-ми битные значения цвета.

Такая гистограмма имеет меньше размер в отличие от гистограммы, состоящей из трёх компонентов пространства RGB и такая гистограмма устойчивее к изменению яркостной составляющей на изображении [66].

Градиентный спуск, применённый для определения параметров четырёхпараметрической модели, имеет вычислительную сложность $O(n^2)$ [67]. Как только в методе идентификации значение критерия соответствия цветовой

гистограммы становится выше определённого порога ρ_{th} , либо количество итераций превышено максимального значения i_{max} , процесс идентификации прекращается. Изображение объекта считается найденным на кадре, если превышено минимальное значение порога схожести ρ_{th} .

2.6 Алгоритм поиска объекта в видеопотоке

Декомпозиция функции поиска объекта в видеопотоке позволяет описать алгоритм поиска:

1. Вычислить параметры изображения объекта для поиска.

1.1 Создать набор Q^{ref} проективно искажённых изображений объекта Y^{ref} .

1.2 Найти ключевые точки изображений объекта для поиска:

$$K^{ref} = \{K^{ref}_0, \dots, K^{ref}_{l-1}\}.$$

1.3 Вычислить дескрипторы ключевых точек K^{ref} изображений объекта:

$$D^{ref} = \begin{bmatrix} D^{ref}_0^0 & \dots & D^{ref}_{m-1}^0 \\ \vdots & \ddots & \vdots \\ D^{ref}_0^{k-1} & \dots & D^{ref}_{m-1}^{k-1} \end{bmatrix},$$

где D^{ref} – матрица значений дескрипторов объекта, $D^{ref}_j^i$ – j -ое значение i -ого дескриптора, m – размер дескриптора, k – количество дескрипторов.

1.4 Вычислить цветовую гистограмму изображения объекта:

$$H^{ref} = \{H_0^{ref}, \dots, H_{n-1}^{ref}\}.$$

2. Пока $i < N_{max}$ выполнить:

2.1 Найти ключевые точки i -ого кадра:

$$K^i = \{K_0^i, \dots, K_{m-1}^i\}.$$

2.2 Вычислить дескрипторы ключевых точек i -ого кадра:

$$D^i = \{D_0^i, \dots, D_{m-1}^i\}.$$

2.3 Сравнить дескрипторы D^i кадра со строками матрицы D^{ref} и выбрать отклоняющиеся выше определённого порога Thr .

2.4 Найти набор прямоугольных областей P^i на кадре алгоритмом ограничения области поиска объекта.

2.5 Пока $j < Sp^i$ выполнить:

2.5.1 Проверить область P_j^i на наличие изображения объекта четырёх параметрическим градиентным спуском.

2.5.2 Если мера схожести гистограмм ниже порога ρ_{th} , делать:

2.5.2.1 $j = j + 1$, перейти на шаг 2.5.

2.5.3 Иначе делать:

2.5.3.1 Выделить прямоугольным окном изображение объекта и сохранить кадр.

2.5.3.2 $i = i + 1$, перейти на шаг 2.

3. Стоп.

На рисунке 2.12 представлен разработанный алгоритм более подробно в виде блок-схемы. На рисунке 2.13 представлена функция вычисления дескрипторов на проективно искажённых изображениях объекта. Блок-схемы построены согласно стандарту ЕСПД 19.701-90 «Схемы алгоритмов, программ, данных и систем» [60].

Функция преобразования изображения $I\{R, G, B\}$ в полутоновое изображение Y взята из рекомендации ВТ.601 [66], [68]:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B,$$

где R, G, B – матрицы цветности изображения I красного, зелёного и синего цвета соответственно.

SURF ищет ключевые точки в масштабе до 10 раз. Предлагается масштабировать изображение образца до 20 раз с шагом равным 2 в связи с тем, что ключевые точки вычисляются только на одном масштабе. Согласно [2] потребуется 6 значений долготы и 3 значения широты для генерации изменений наклона камеры. SURF устойчив к поворотам до 30° [69], следовательно, предлагается использовать 12 значений коэффициентов изменения угла поворота с шагом 30° .

Таким образом, общее количество проективно-искажённых изображений составит $20 \cdot 6 \cdot 3 \cdot 12 + 1 = 4321$ штука.

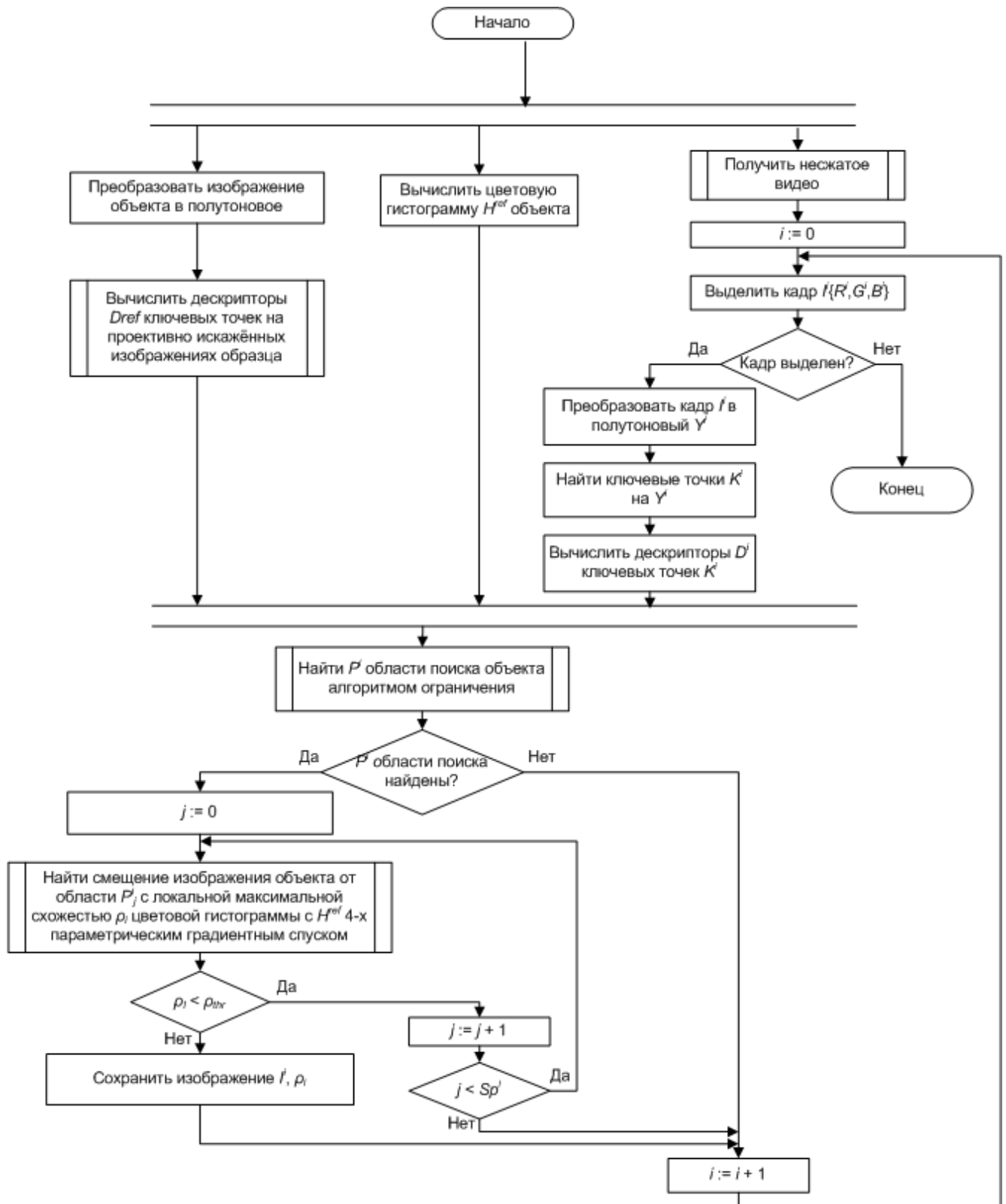


Рисунок 2.12 – Блок-схема алгоритма поиска объекта в видеопотоке

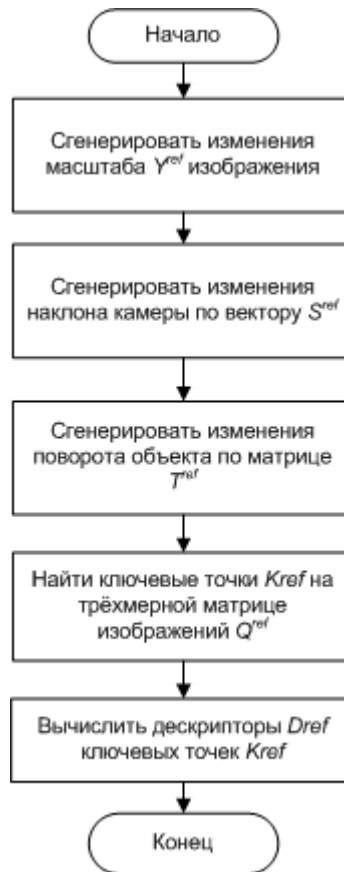


Рисунок 2.13 – Блок-схема функции вычисления дескрипторов на проективно искажённых изображениях объекта

2.7 Структурная модель

На основании построенной функциональной модели можно определить основные модули системы:

- модуль вычисления вектора дескрипторов объекта;
- модуль вычисления гистограммы;
- модуль извлечения кадров;
- модуль вычисления вектора дескрипторов кадра;
- модуль поиска областей претендентов;
- модуль идентификации.

На рисунке 2.14 представлена диаграмма основных компонентов системы. Модуль вычисления вектора дескрипторов объекта выполняет функции A11, A12,

A13, A14, A15, A16, представленные в IDEF0-диаграммах. Модуль извлечения кадров выполняет функции A31, A32, A33, кадры из которого поступают в модуль вычисления вектора дескриптора кадра и в модуль поиска областей претендентов. Функция A2 выполняется модулем вычисления гистограммы. Модуль вычисления вектора дескрипторов кадра выполняет функции A41, A42, A43. Модуль поиска областей претендентов реализует функции A51, A52, A53. Идентификатор выполняет функцию A6.

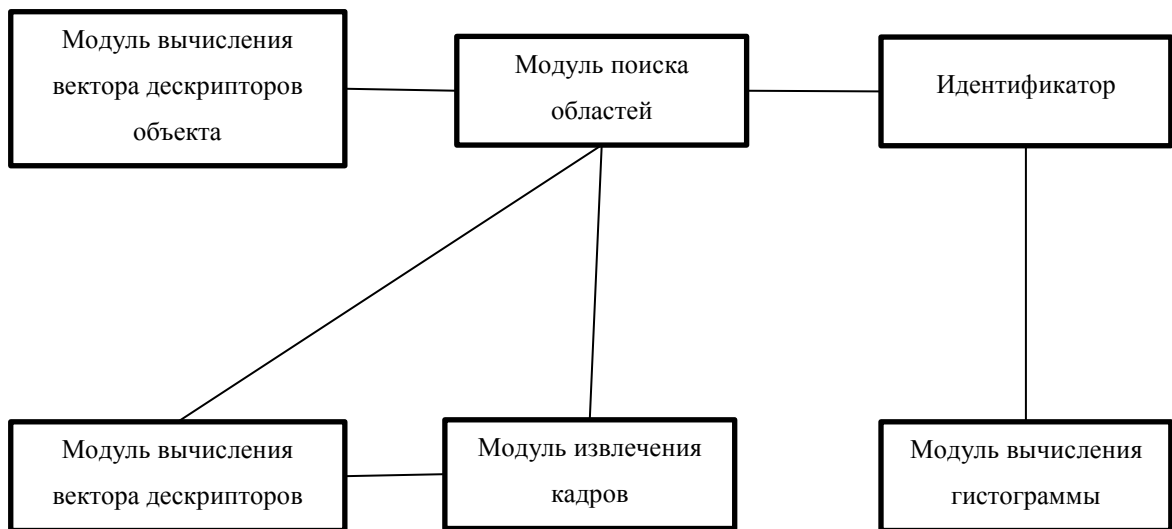


Рисунок 2.14 – Структурная диаграмма

Структуры модулей вычисления многомерной матрицы дескрипторов объекта, извлечения кадров, вычисления вектора дескрипторов кадра и поиска областей претендентов представлены в последующих подразделах.

2.7.1 Структура модуля вычисления вектора дескрипторов объекта

На рисунке 2.15 представлена структура модуля вычисления вектора дескрипторов объекта. Модуль состоит из следующих элементов:

1. Полутоновый преобразователь. Выполняет функцию A11.
2. Преобразователь масштаба. Выполняет функцию A12.
3. Преобразователь наклона. Моделирует изменения наклона камеры на заданные углы. Реализует функцию A13.

4. Генератор поворота. Выполняет функцию A14.
5. Поисквик ключевых точек. Выполняет функцию A15.
6. Вычислитель дескрипторов. Выполняет функцию A16.



Рисунок 2.15 – Структура модуля вычисления вектора дескрипторов объекта

2.7.2 Структура модуля извлечения кадра

Структура модуля извлечения кадра представлена на рисунке 2.16. Модуль состоит из следующих элементов:

1. Демультимплексер. Выполняет A31 функцию демультимплексирования видеопотока из контейнера в сжатый формат видео.
2. Декодер. Выполняет A32 функцию декодирования сжатого формата видео.
3. Компонента выделения кадра. Выполняет A33 функцию выделения кадра из декодированного видеопотока.



Рисунок 2.16 – Структура модуля извлечения кадра

2.7.3 Структура модуля вычисления вектора ключевых точек кадра

Структура модуля вычисления вектора ключевых точек кадра представлена на рисунке 2.17. Модуль состоит из следующих элементов:

1. Полутоновый преобразователь. Выполняет функцию А41.
2. Поисквик ключевых точек. Находит ключевые точки на кадре. Выполняет функцию А42.
3. Вычислитель дескрипторов. Вычисляет дескрипторы ключевых точек. Выполняет функцию А43.

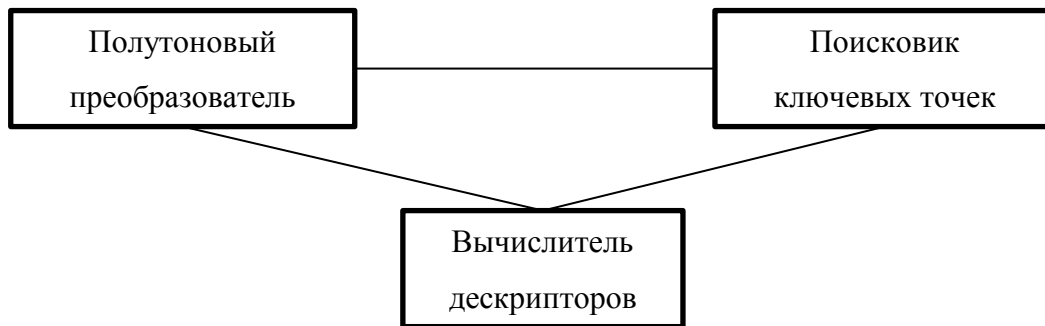


Рисунок 2.17 – Структура модуля вычисления вектора ключевых точек кадра

2.7.4 Структура модуля поиска областей претендентов

Структура модуля поиска областей претендентов представлена на рисунке 2.18. Модуль состоит из следующих элементов:

1. Определитель пересечения дескрипторов. Выполняет функцию А51.
2. Кадровый фильтр. Пропускает кадр с определённым номером, полученным из определителя пересечения дескрипторов. Выполняет функцию А52.
3. Определитель областей. Выполняет функцию А53.

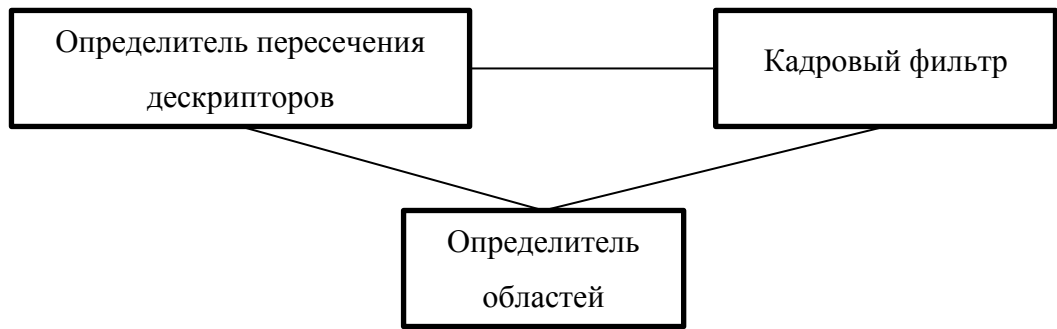


Рисунок 2.18 – Структура модуля поиска областей претендентов

2.8 Выводы

1. Функциональный анализ системы поиска объекта показал, что в структуре программной системе должны быть включены в качестве базовых функций следующие:

- нахождение ключевых точек;
- вычисление дескрипторов ключевых точек;
- нахождение общих ключевых точек по дескрипторам;
- идентификация кадра на предмет наличия изображения объекта.

2. Алгоритм поиска объекта должен строиться на основе вычисления дескрипторов, так как такой подход позволяет избежать влияния проективных искажений.

3. Для увеличения скорости работы алгоритма необходимо перед началом поиска подвергнуть изображение объекта проективными искажениями для нахождения ключевых точек и вычисления дескрипторов объекта.

4. Для повышения вероятности обнаружения дескрипторный поиск ключевых точек необходимо дополнить шагом сравнения цветных гистограмм.

3 Проектирование и реализация программы поиска объекта в видеопотоке

Описанные в разделе 2 функции и структурные элементы системы позволяют провести проектирование этой системы. Диаграммы построены с помощью языка UML [70].

3.1 Проектирование классов системы

Структурная модель позволяет перейти к описанию классов системы, к описанию взаимосвязей. Для детального описания предметной области добавлены следующие элементы:

- область;
- ключевая точка;
- кадр.

На рисунке 3.1 представлена диаграмма классов системы. Структура классов полностью согласуется со структурной моделью системы.

3.2 Сценарии работы системы

Функциональная модель системы позволяет определить основные сценарии функционирования:

- инициализация изображения объекта;
- поиск кадров объекта в видеопотоке.

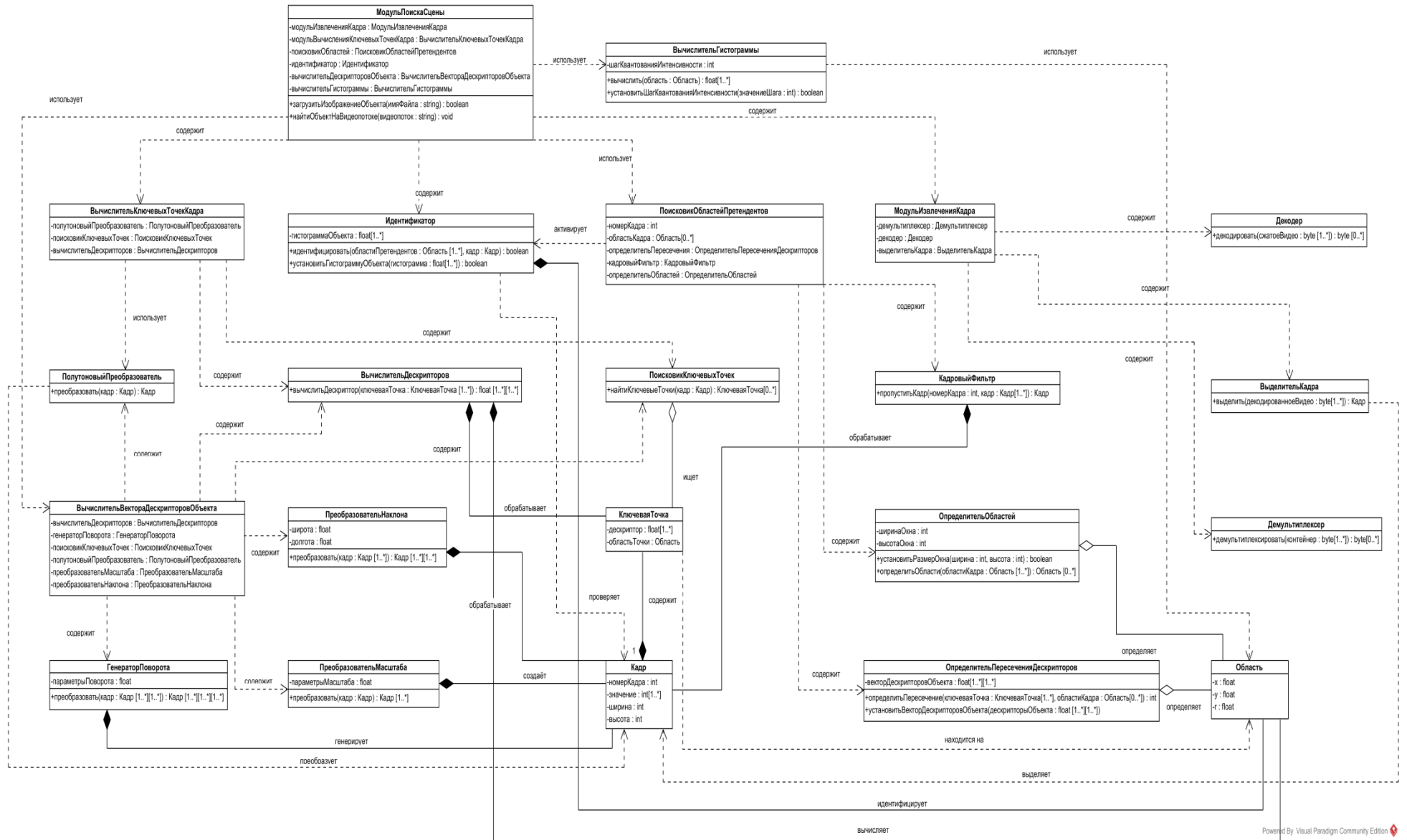


Рисунок 3.1 – Диаграмма классов системы

Для иллюстрации событий, связывающих внешних исполнителей с системой, в языке UML существуют специальные обозначения, позволяющие создавать диаграммы последовательностей, являющиеся отправной точкой для проектирования взаимодействия объектов [71]. Построенная диаграмма классов позволяет описать выделенные сценарии. В следующих подразделах представлены диаграммы последовательностей сценариев функционирования.

3.2.1 Сценарий инициализации изображения объекта

На рисунке 3.2 представлен сценарий инициализации изображения объекта.

Вектор дескрипторов ключевых точек объекта вычисляется параллельно с вычислением гистограммы изображения. Вычислитель вектора дескрипторов объекта генерирует трёхмерную матрицу проективно искажённых полутоновых изображений образца при помощи преобразователя масштаба, преобразователя наклона, генератора поворота и полутонового преобразователя. Вычисленная гистограмма передаётся в идентификатор, а вектор дескрипторов объекта – в поисковик областей претендентов. Инициализация изображения объекта завершается после получения сообщения об установке дескрипторов объекта и значений гистограммы.

3.2.2 Сценарий поиска кадров объекта в видеопотоке

На рисунке 3.3 представлен сценарий поиска кадров объекта видеопотоке. Входной видеопоток поступает в демуксер для извлечения закодированного видео из контейнера. Демуксированное видео поступает в декодер для представления видео в набор байт, который поступает в выделитель кадра. Согласно параметрам, пока выделитель кадра успешно извлекает из последовательности декодированных байт кадр, выполняется процесс поиска объекта в видеопотоке. Полученный кадр передаётся в вычислитель ключевых точек кадра и в поисковик области претендентов. Вычислитель ключевых точек кадра преобразует кадр в полутоновое изображение, на котором осуществляет поиск ключевых точек с

вычислением дескрипторов. Если ключевые точки найдены, то поисковик ключевых точек передаёт ключевые точки вычислителю дескрипторов. Если вектор дескрипторов кадра не является пустым, то дескрипторы передаются поисковику областей претендентов для нахождения общих дескрипторов P с дескрипторами изображения поискового объекта. Если пересечение дескрипторов является не пустым множеством ($P \neq \{0\}$), то это множество отправляется определителю областей и подаётся информация о номере кадра кадровому фильтру, который пропускает соответствующий кадр в идентификатор. Из определителя областей поступают области кадра в идентификатор. Если объект локализуется на кадре, то идентификатор сохраняет текущий кадр.

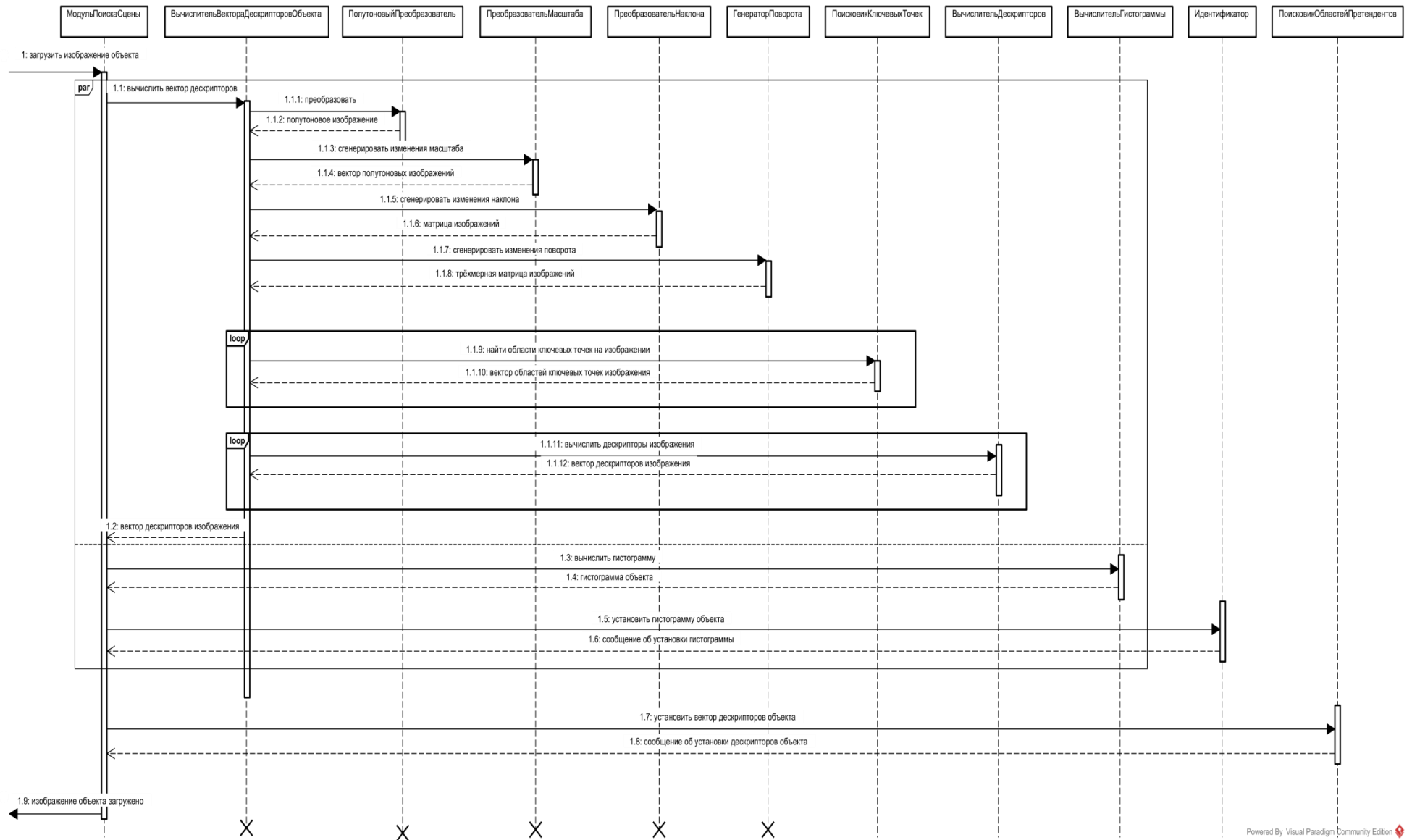


Рисунок 3.2 – Сценарий инициализации изображения объекта

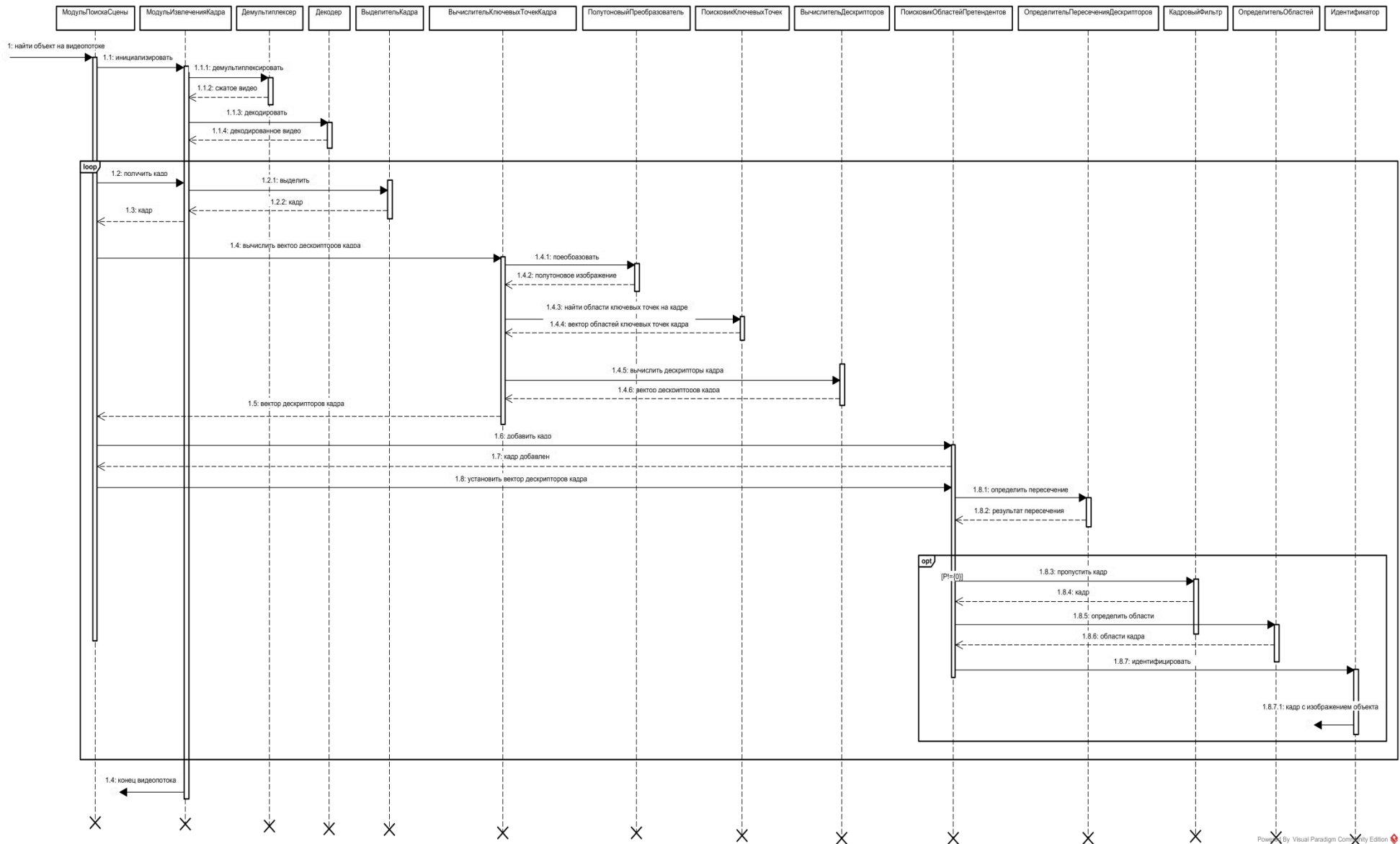


Рисунок 3.3 – Сценарий поиска кадров объекта в видеопотоке

3.3 Реализация программы поиска объекта в видеопотоке

Спроектированные классы и взаимодействия системы позволяют осуществить реализацию программы поиска объекта в видеопотоке. Реализация программы должна удовлетворять требованиям к программе и соответствовать построенным функциональной и структурной моделям. Далее описываются технологии разработки программы для эффективного выполнения поиска объекта.

3.4 Описание технологий

Программа работает в двух режимах согласно описанным сценариям работы системы:

- режим инициализации изображения объекта;
- режим поиска кадров объекта в видеопотоке.

Поиск кадров с изображением объекта в видеопотоке требует гораздо большего количества выполнения вычислительных операций, чем инициализация изображения объекта. Реализация программной системы должна быть ориентирована на эффективное использование вычислительных ресурсов для удовлетворения требования к скорости обработки. Предлагается использовать следующие технологии:

- конвейерная обработка данных;
- пакетная обработка данных;
- CUDA для параллельных вычислений в видеокарте.

Конвейерная архитектура позволит выполнять обработку видео кадров одновременно на всех компонентах системы [72]. Тем самым уменьшая среднее время обработки кадра до максимального времени обработки одного из компонентов системы. В идеальном разбиении на n компонент время обработки кадров практически уменьшится в n раз.

Решение об отсутствии объекта на кадре может быть принято на этапе определения пересечения дескрипторов кадра и изображения объекта, поэтому предлагается выделять больше вычислительных ресурсов операциям поиска ключевых точек, вычисления дескрипторов, определения пересечения дескрипторов.

Для минимизации времени поиска объекта в видеопотоке предлагается реализовать основные функции с использованием пакетной обработки данных: технология SSE2 (Streaming SIMD Extensions 2). Её назначение – повысить эффективность операций со 128-разрядными данными в формате плавающей точки с двойной точностью и с целочисленными данными [73]. Все современные процессоры поддерживают технологию SSE2, в отличие от технологий SSE4, AVX и AVX2.

Для оптимального использования ресурсов предлагается при наличии видеокарты с поддержкой технологии CUDA использовать вычислительные ресурсы видеокарты. CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU (графических процессоров). [74]. На основании результатов вычисления алгоритмической сложности модифицированного метода SURF, процесс поиска ключевых точек требует большое количество вычислительных операций и является легко распараллеливаемым в отличие процесса вычисления дескриптора. Пусть поиск ключевых точек осуществляется в видеокарте при её наличии. Тогда для операции вычисления дескриптора будет отдаваться больше процессорных вычислительных ресурсов.

На рисунке 3.4 представлена диаграмма развёртывания. Представленная диаграмма развёртывания позволяет увеличивать экземпляры компонент при увеличении количества вычислительных ресурсов за счёт интерфейсов взаимодействия между компонентами, которые управляют доступом на запись и чтение необходимой информации для обмена.

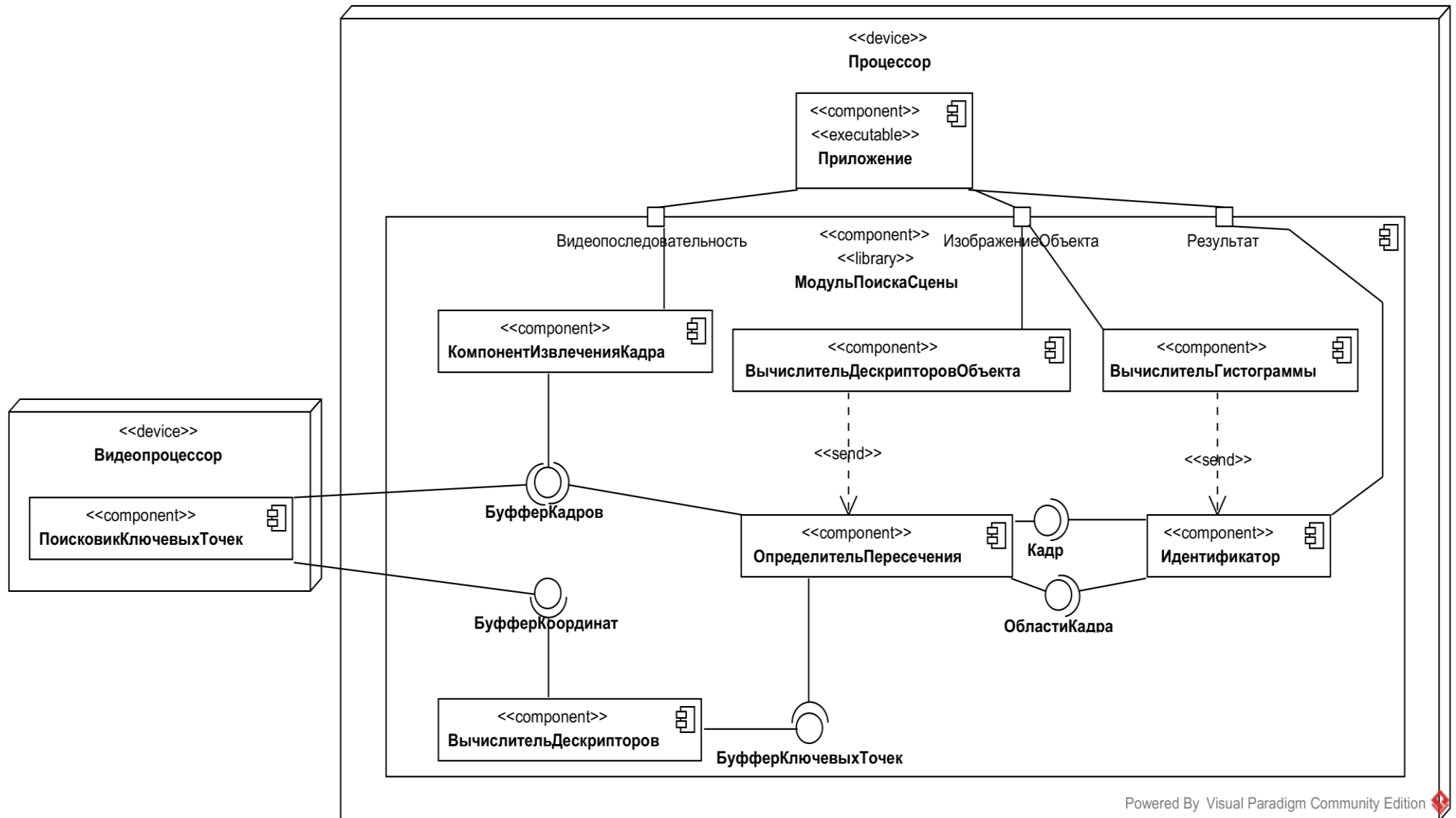


Рисунок 3.4 – Диаграмма развёртывания

Для реализации программ использованы следующие программные средства:

- языки программирования C++ и Assembler с использованием SSE2.
- библиотека OpenCV 2.4 содержит функции демуксирования, декодирования видеопотока, вычисление SURF дескрипторов, алгоритм RANSAC [16].
- набор инструментов CUDA SDK.
- операционная система Microsoft® Windows 7.
- среда разработки: Microsoft® Visual Studio 2010.
- система контроля версий: Tortoise SVN.

Копия свидетельства о регистрации программы «Object tracking PS» [75] (программа слежения за объектом в видеопотоке) представлена в Приложении Б.

3.5 Выводы

На основе проведенных исследований можно сделать следующие выводы:

1. Для реализации программы поиска объекта на изображении и повышения скорости обработки кадров необходимо использовать конвейерную обработку данных; пакетную обработку данных, CUDA для параллельных вычислений в видеокарте и объектно-ориентированное программирование.
2. Разработанный набор классов с выделенными методами позволяет полностью реализовать все функции системы.
3. Спроектированная структура компонентов предоставляет возможность разворачивать программную систему на все доступные процессоры, включая видеопроцессор.

4 Тестирование системы

Реализованная программа позволяет осуществить автоматическое тестирование разработанного алгоритма поиска объекта в видеопотоке с использованием текстового формата вывода результатов обнаружения. Необходимо разработать схему тестирования и написать исполняемые скрипты для проведения тестов.

4.1 Схема проведения тестирования

Согласно поставленным требованиям система должна быть инвариантна к проективным преобразованиям, устойчива к шуму и находить объект в видеопотоке в режиме реального времени, поэтому предлагается исследовать каждую программу на удовлетворение этим требованиям. Для этого подготавливаются тесты с видеопотоками, на которых изображение определённого объекта подвергается проективным преобразованиям. В качестве сравнения результатов тестирования с аналогами используются готовые решения от Vuforia [14], Kudan AR [15] и реализации методов SURF, ASIFT совместно с методом RANSAC.

В тестированиях предлагается использовать метрику ROC [76] для численного измерения качества нахождения объекта в видеопотоке различными методами. ROC-кривая имеет особую популярность в области анализа классификаций и отображает зависимость доли верно классифицированных положительных событий в общем числе положительных событий от доли неверно классифицированных отрицательных событий в общем числе отрицательных событий, поэтому на графике точка $(0;1)$ соответствует идеальному классификатору. Для построения ROC-кривых необходимо вычислить:

- сумму истинно-положительных значений (true-positive, TP),
- сумму ложно-положительных значений (false-positive, FP),
- сумму истинно-отрицательных значений (true-negative, TN),

– сумму ложно-отрицательных значений (false-negative, FN).

TP , FP , FN , TN определяются по следующим формулам 4.1, 4.2, 4.3, 4.4.

$$TP_i = \sum_{\rho_j \geq Th_i} \rho_j \quad (4.1)$$

$$FP_i = \sum_{\rho_j \geq Th_i} (1 - \rho_j) \quad (4.2)$$

$$FN_i = \sum_{\rho_j < Th_i} \rho_j \quad (4.3)$$

$$TN_i = \sum_{\rho_j < Th_i} (1 - \rho_j) \quad (4.4)$$

График ROC-кривой есть график зависимости доли истинно-положительных значений (TPR , формула 4.5) от доли ложно-положительных значений (FPR , формула 4.6) [77].

$$TPR_i = \frac{TP_i}{TP_i + FN_i} \quad (4.5)$$

$$FPR_i = \frac{FP_i}{FP_i + TN_i} \quad (4.6)$$

Численный показатель площади под ROC кривой (AUC) определяется по формуле 4.7 [76].

$$AUC = \frac{1}{2} \sum_i (TPR_{i+1} + TPR_i) \cdot (FPR_{i+1} - FPR_i) \quad (4.7).$$

Программная система работает эффективно, если использует все предоставленные аппаратно-программные средства в течение всего времени работы. В качестве критериев эффективности предлагается вычислять среднюю загрузку процессоров и среднюю загрузку процессоров на обработку видеопоследовательности в секунду.

Средняя загрузка процессоров вычисляется по следующей формуле:

$$L = \frac{1}{n} \sum_{i=0}^{n-1} CPU_i,$$

где n – количество процессоров, CPU_i – средний процент загрузки i -ого процессора в секунду.

Средняя загрузка процессоров на обработку видеопоследовательности в секунду вычисляется по формуле:

$$L_s = \frac{L}{T},$$

где L_s – средняя загрузка процессоров, T – время выполнения программы на обработку видеопоследовательности. Чем ближе значение L_s к 0, тем дольше по времени необходимо использовать вычислительных ресурсов для обработки видеопоследовательности.

Схема тестирования приведена на рисунке 4.1. Тест содержит изображение объекта для поиска, специально подготовленный видеопоток и информацию о номерах кадров с изображением объекта. Выполнение теста осуществляется одним реализованным методом специальной программой. Результат выполнения теста – значение вычисленной метрики. Задача скрипта – получить результат метрики по выходным данным отработанной программы на определённом видеофайле с изображением объекта с использованием информации об ожидаемых результатах (список номеров кадров видеофайла с изображением объекта).

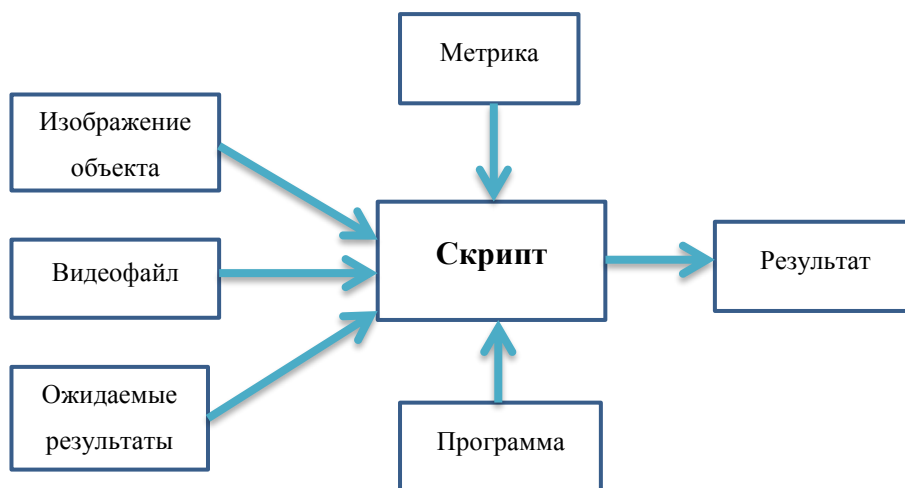


Рисунок 4.1 – Схема проведения тестирования

По знанию системы тестирование производится «чёрного ящика» через специальные интерфейсы взаимодействия с программой [79]. Для проведения тестирования написан скрипт на языке python [80]. Скрипт выполняет

последовательный запуск реализованных программ поиска объекта в видеопотоке с пороговым значением меры близости и вычисляет значения ROC кривой (*FPR*, *TPR*), используя информацию о номерах кадров с изображением объекта для каждого видеопотока. Текст скрипта представлен в Приложении В. Все рассматриваемые методы имеют критерии сходства, значения которых изменяются от нуля до единицы. Пороговое значение меры близости в скрипте изменяется от нуля до единицы с шагом 0,05.

4.2 Результаты тестирования

Тестирование проводилось на машине HP Z420 с восьми ядерным процессором Intel Xeon E5-1620 v2 с частотой 3,7 ГГц, объёмом памяти равным 8 Гб и операционной системой MS Windows 7 x64 с видеокартой GIGABYTE GeForce GTX 750 Ti частотой 1033 МГц и объёмом памяти 2048 Мб с отключенным Turbo Boost.

Поиск при помощи Vuforia SDK и Kudan SDK осуществлялся на мобильной платформе Android 4.1.2 с четырёх ядерным процессором Qualcomm частотой 1,2 ГГц, объёмом памяти 1 Гб с графическим ускорителем Adreno 203.

Реализации приведены к максимально одинаковым условиям, т. е. все программы написаны на языках C++ и Assembler, используют параллельные вычисления с технологиями CUDA и SSE2. Использованы Vuforia SDK 5.5.9 версии 2016 года [14] и Kudan SDK 1.2.3 [15] для Android платформы. Реализация метода ASIFT взята из [2]. Использована библиотека OpenCV 2.4 [16] для выполнения поиска объекта методом SURF и процедуры сопоставления дескрипторов методом RANSAC.

4.2.1 Тестирование инвариантности к проективным изменениям

Для проведения тестирования инвариантности к проективным преобразованиям подготовлены видеофайлы с изменённым поворотом,

масштабом объекта и наклоном камеры к объекту. В последовательности принимали участие кадры, не содержащие изображения объекта.

Для проверки инвариантности к поворотам взято десять изображений различных объектов для поиска, из которых смоделированы видеопоследовательности изменением угла поворота. Для первого изображения каждого видеофайла угол изменён на 5° от оригинальной картинке. Это сделано для того, чтобы изменение угла поворота происходило с другим шагом, отличным от шагов в сравниваемых методах. Остальные изменения угла поворота происходят на 10° . Пример кадров одной из видеопоследовательности изображен на рисунке 4.2.



Рисунок 4.2 – Пример кадров видеофайла с изменением угла поворота (слева направо: изображение для поиска, изображение, повёрнутое на угол 25° , 115° и 285° и изображение, не содержащее изображение объекта).

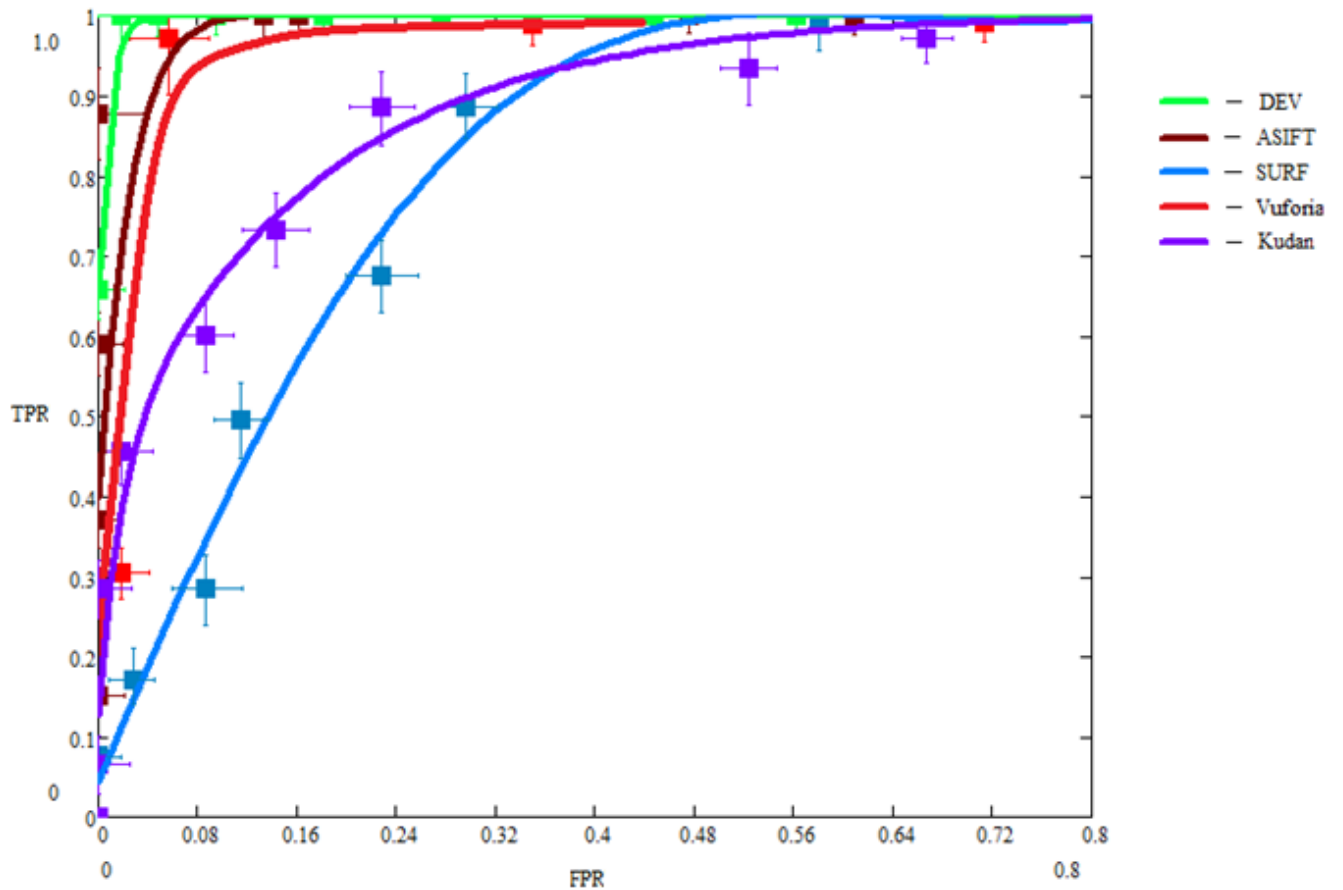


Рисунок 4.3 – ROC кривые результатов выполнения теста на проверку инвариантности к повороту объекта

Значения результатов тестов на различных входных видеопотоках для FPR и TPR усреднены и представлены в таблице 4.1, где DEV – разработанный метод. На рисунке 4.3 приведены графики ROC кривых (доверительный интервал равен 95%).

График разработанного метода (DEV), ASIFT и метода Vuforia SDK проходят всех ближе к точке (0;1), значит успешно классифицировали практически все кадры видеопоследовательностей с наличием изображения объекта и с его отсутствием.

Аналогичным образом проводились тесты с изменением масштаба на десяти видеопотоках. Масштаб картинки уменьшается от 2 до 20 раз по ширине и высоте, и затем увеличивается от 2 до 20 раз с шагом равным 2. Результаты приведены в таблице 4.2 и на рисунке 4.4.

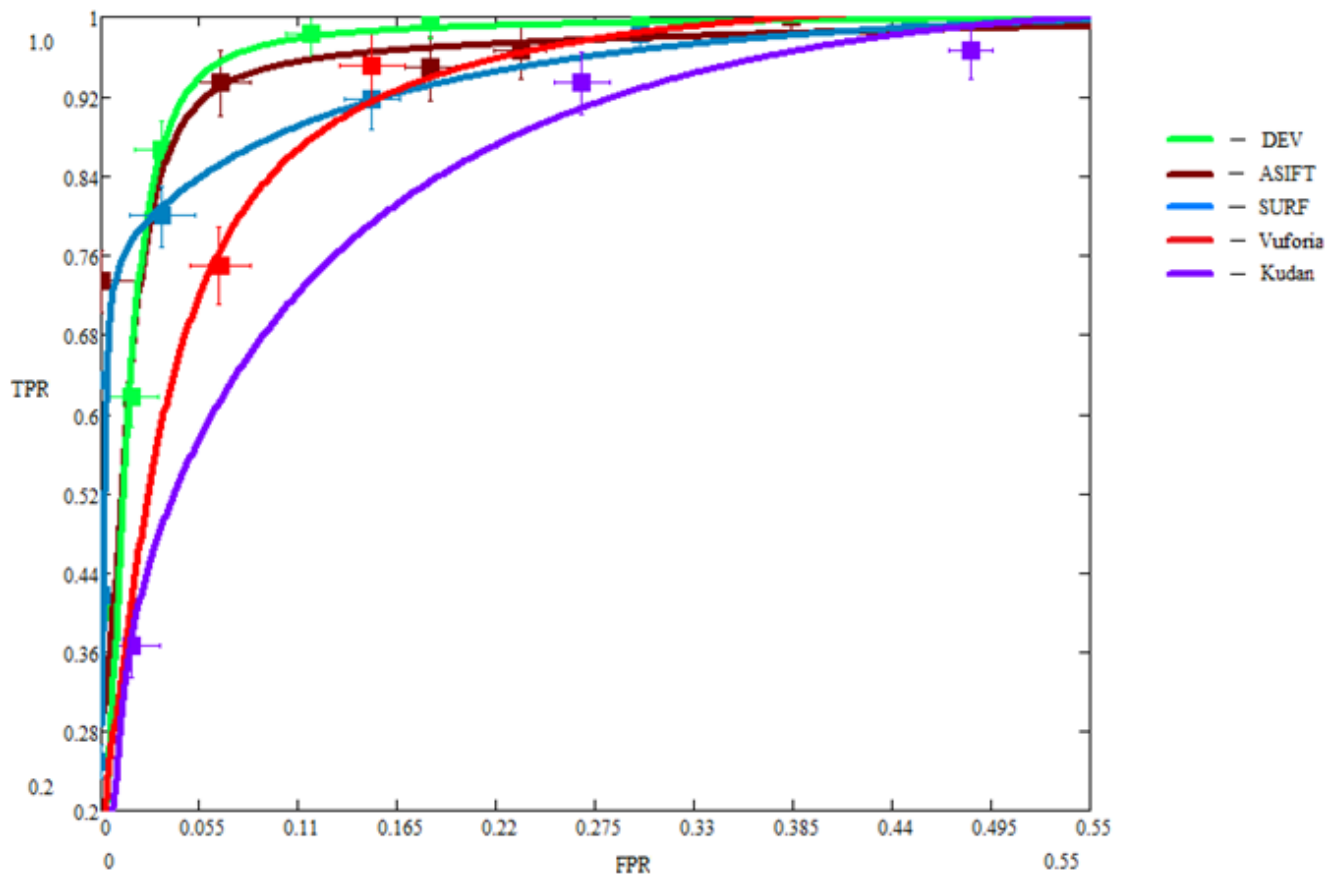


Рисунок 4.4 – ROC кривые результатов выполнения теста на проверку инвариантности к изменению масштаба объекта

Результаты классификаций для теста с изменением масштаба выглядят менее убедительными, чем результаты теста на проверку инвариантности к повороту. Это связано с тем, что большинство методов основаны на дескрипторе, устойчивым к повороту на 180° , а для инвариантности к масштабу методы изменяют размеры картинки до определённых установленных максимальных и минимальных значений.

Графики ROC кривых методов DEV, ASIFT являются всех ближе к точке (0; 1) и при $\approx 0,1$ вероятности ошибки классификации кадров без изображения объекта достигается 0,92 (и больше) вероятности распознавания кадров с изображением объекта.

Для проведения тестов на проверку инвариантности к изменениям наклона камеры подготовлены аналогичным образом 10 видеофайлов. Для изменения наклона использовалось 3 значения широты (0° , 45° , 170°) и 6 значений долготы

(0°, 36°, 72°, 108°, 144°, 180°). Результаты теста приведены в таблице 4.3, графики ROC кривых отображены на рисунке 4.5. График разработанного метода на области FPR от 0,06 до 0,275 имеет значения TPR выше, либо равные значениям TPR остальных графиков. Это значит, что при одной и той же вероятности ошибки классификации кадров без изображения объекта вероятность распознавания кадров с изображением объекта выше у разработанного метода DEV на промежутке FPR от 0,06 до 0,275.

Значения показателей AUC представлены в таблице 4.4 для численного сравнения результатов теста. Разработанный метод имеет самые большие значения AUC в проведённых тестах.

Таблица 4.3 – Вычисленные средние значения FPR и TPR результатов теста на проверку инвариантности к изменению наклона камеры

Порог меры схожести	Dev		Vuforia SDK		Kudan SDK		ASIFT		SURF	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR
0,95	0	0,2222	0	0,0186	0	0,2222	0	0,1481	0	0,2037
0,90	0	0,3704	0	0,4074	0	0,3704	0	0,3333	0	0,3148
0,85	0	0,5370	0,0185	0,6481	0,0185	0,5370	0	0,5185	0	0,5185
0,80	0	0,7593	0,1481	0,8333	0,0741	0,7222	0,0185	0,7963	0,0556	0,6481
0,75	0,0741	0,9815	0,2407	0,9259	0,1111	0,9074	0,1481	0,9444	0,1481	0,8148
0,70	0,1481	1	0,2963	1	0,2037	0,9444	0,2037	1	0,2593	0,9259
0,65	0,2222	1	0,3704	1	0,2778	1	0,2407	1	0,3148	1
0,60	0,4444	1	0,5185	1	0,4444	1	0,3889	1	0,4074	1
0,55	0,6667	1	0,6481	1	0,6667	1	0,6667	1	0,6111	1
0,50	0,7778	1	0,7963	1	0,7778	1	0,7778	1	0,7407	1
0,45	0,8519	1	0,8889	1	0,8519	1	0,8333	1	0,8148	1
0,40	0,9815	1	0,9815	1	0,9815	1	1	1	1	1
0,35	0,9815	1	1	1	0,9815	1	1	1	1	1

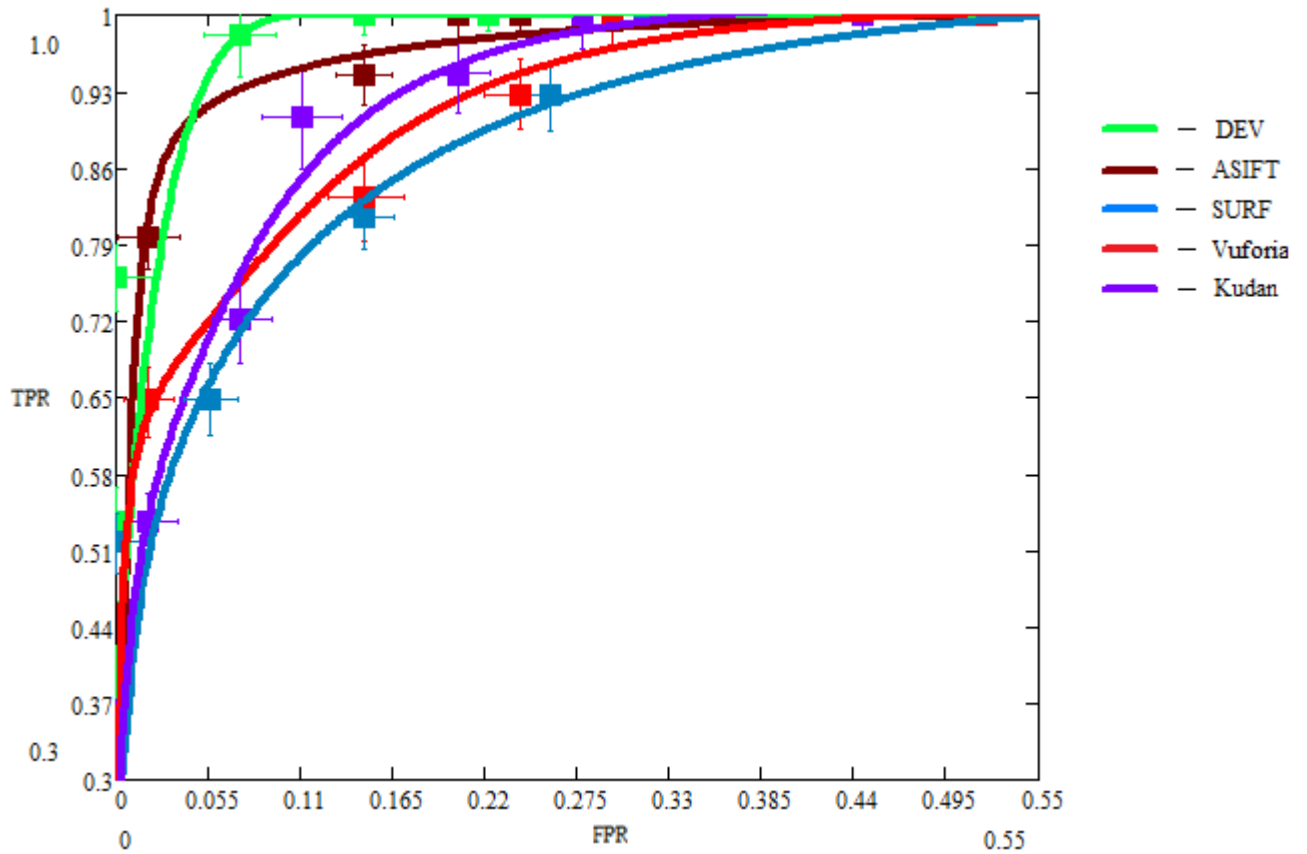


Рисунок 4.5 – ROC кривые результатов выполнения теста на проверку инвариантности к изменению наклона камеры

На основании анализа результатов, приведённых на графиках и в таблице, можно сказать, что классификация разработанного метода является сопоставимой с классификацией методов Vuforia SDK, ASIFT и разработанный метод позволяет находить проективно искажённое изображение объекта.

Таблица 4.4 – AUC значения тестов

Тест	Метод				
	DEV	Vuforia SDK	Kudan SDK	ASIFT	SURF
Изменение поворота	0,9967	0,9678	0,8863	0,9917	0,8332
Изменение масштаба	0,9807	0,9457	0,9264	0,9775	0,9652
Изменение наклона	0,9897	0,9444	0,9535	0,9753	0,9355

4.2.2 Сравнение результатов методов поиска по вероятности обнаружения объекта

Под вероятностью обнаружения объекта p понимается отношение суммы количества верно обнаруженных кадров объекта (TP) и количества верно классифицированных кадров с отсутствием объекта (TN) к общему количеству кадров (N):

$$p = \frac{TP + TN}{N}.$$

При использовании этой формулы необходимо, чтобы рассматриваемая видеопоследовательность имела равное количество кадров с наличием и с отсутствием объекта.

Для вычисления вероятности обнаружения каждого метода суммированы значения результатов TP и TN тестов. Пороговое значение меры близости для каждого метода выбраны на основании результатов тестирования инвариантности к проективным изменениям (см. таблицы 4.1, 4.2, 4.3), при которых достигается наименьшее число ошибок. В таблице 4.5 приведены значения вероятностей обнаружения каждого из методов на тестируемых видеофайлах. Общее количество кадров $N = 944$.

Таблица 4.5 – Вероятности обнаружения объектов методами

Метод	<i>TP</i> (количество кадров)	<i>TN</i> (количество кадров)	Вероятность обнаружения
SURF	333	321	0,69
Kudan SDK	345	341	0,73
Vuforia SDK	427	335	0,81
ASIFT	432	319	0,80
DEV (разработанный)	471	351	0,87

Значения результатов вероятностей обнаружения методов показывают, что разработанная методика позволяет увеличить вероятность обнаружения объекта с уровня $\approx 0,81$ до $\approx 0,87$.

4.2.3 Сравнение быстродействия разработанного метода с методом SURF

Для сравнения быстродействия методов SURF с разработанным реализована программа, выполняющая последовательный поиск заданного объекта в видеопотоке разработанным или SURF совместно с RANSAC методом. Оба метода имеют общий код нахождения ключевых точек и вычисления дескрипторов. Использована реализация метода SURF из библиотеки OpenCV 2.4 [16]. Для разработанного метода выставляется 1 октава с 1 масштабом ключевой точки, а для метода SURF выставляется 6 октав с 4 масштабами ключевой точки. Реализация метода RANSAC взята также из библиотеки OpenCV 2.4 [16]. Программы написаны без использования параллельной обработки.

На рисунках 4.6 и 4.7 представлены результаты выполнения методов в видеопотоках, содержащих изображение объекта и без изображения объекта соответственно.

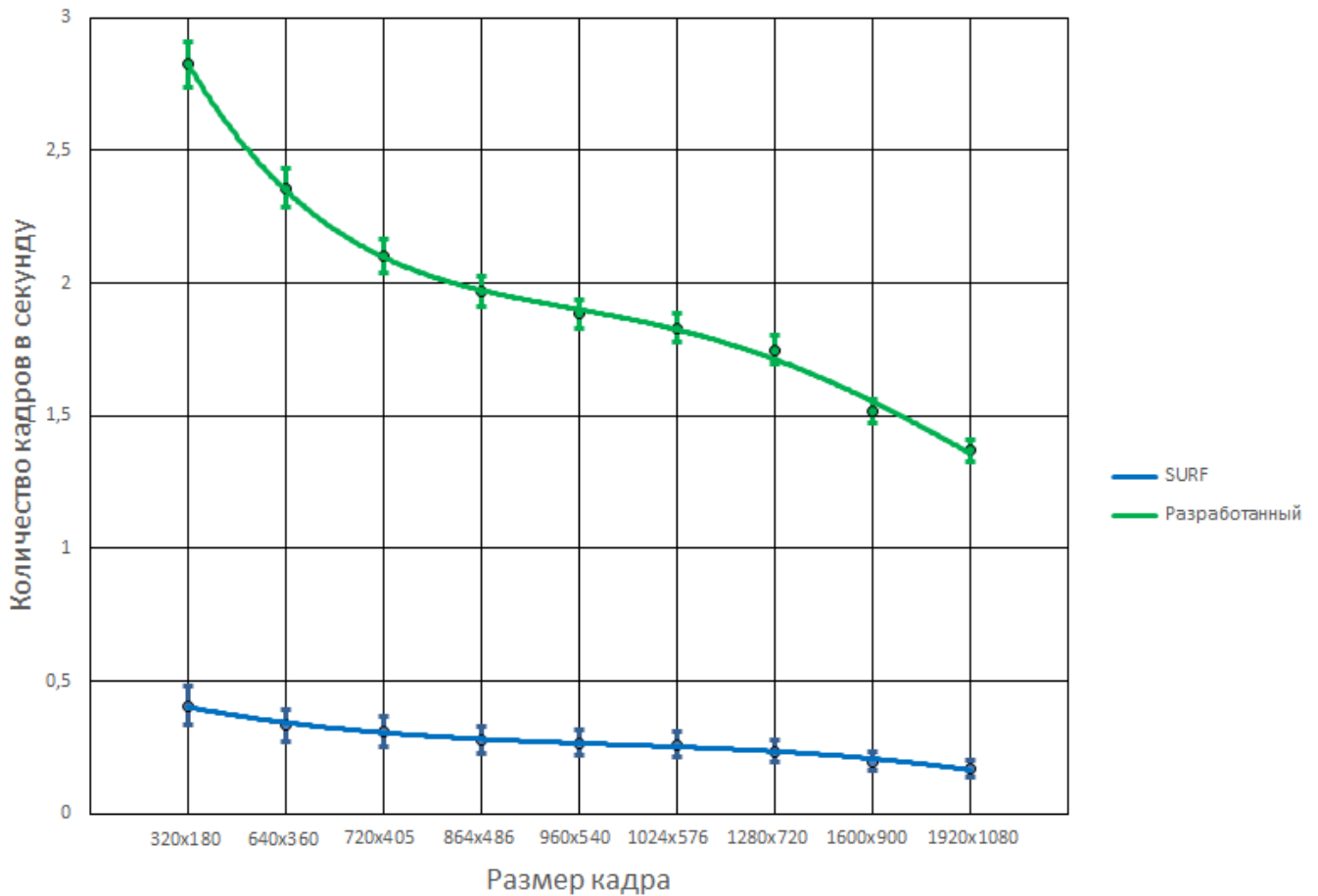


Рисунок 4.6 – Результаты выполнения SURF и разработанного методов в видеопотоках, содержащих изображение объекта

Существующие методы имеют скорость обработки примерно в 8–10 раз меньше разработанного. Прирост скорости в разработанном методе вызван уменьшением вычислительной сложности оригинального метода SURF в s раз (s – количество октав) и заменой алгоритмов RANSAC и Куна-Манкреса для поиска пересечения дескрипторных множеств на упрощённый алгоритм с линейной вычислительной сложностью.

На основании анализа результатов, приведённых на рисунках 4.6 и 4.7, можно сказать, что разработанный метод позволяет обрабатывать кадры с изображением объекта до ≈ 8 раз быстрее аналога SURF и до ≈ 10 раз – кадры без изображения объекта.

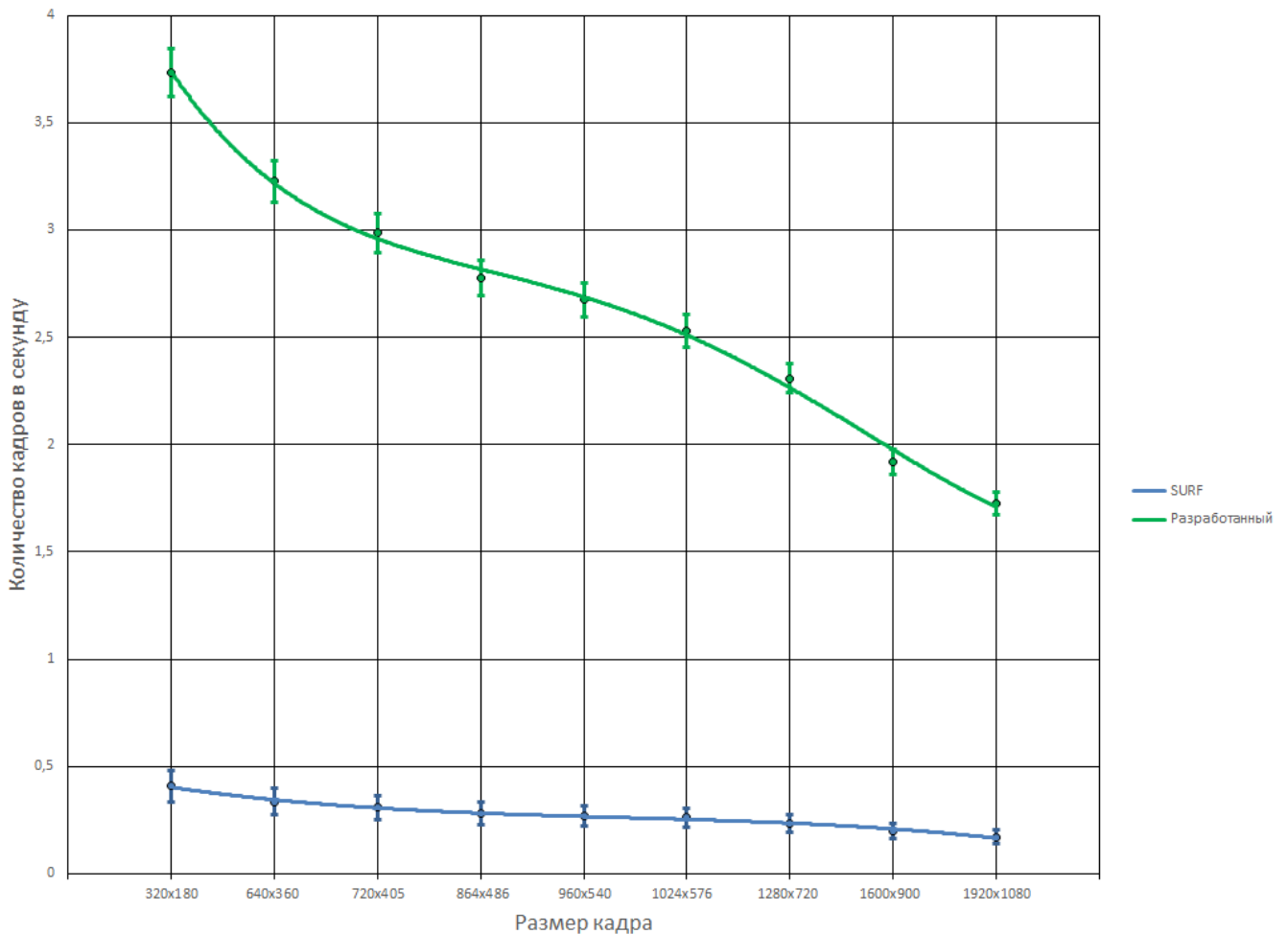


Рисунок 4.7 – Результаты выполнения SURF и разработанного методов в видеопотоках, не содержащих изображения объекта

При обработке кадра, содержащего изображение объекта, необходимо дополнительное время на выполнение функции идентификации объекта (A6), в то время как при отсутствии изображения объекта эта функция может быть не выполнена в связи с принятием решения об остановке процесса поиска в функции нахождения общих ключевых точек кадра и объекта (A51). С этим связано различие скоростей обработки видеопоследовательностей с изображением объекта (см. рисунок 4.6) и с его отсутствием (см. рисунок 4.7)

4.2.4 Тестирование производительности

Для проверки производительности системы выбраны видеофайлы (см. таблицу 4.6) из открытого доступа [49]:

Таблица 4.6 – Тестовые видеопотоки

Номер видеофайла	Название файла	Электронный адрес ссылки
1	big_buck_bunny	https://media.xiph.org/video/derf/y4m/big_buck_bunny_1080p24.y4m.xz
2	pedestrian_area	ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/1080p/pedestrian_area.yuv
3	Netflix_SquareAndTimelapse	https://media.xiph.org/video/derf/ElFuente/Netflix_SquareAndTimelapse_4096x2160_60fps_10bit_420.y4m
4	Netflix_RitualDance	https://media.xiph.org/video/derf/ElFuente/Netflix_RitualDance_4096x2160_60fps_10bit_420.y4m
5	ice	ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/ICE_704x576_60_orig_02.yuv.zip
6	mobile	ftp://ftp3.itu.int/video-site/sequences/mobile_cif.gz
7	stefan	https://media.xiph.org/video/derf/y4m/stefan_sif.y4m
8	shields	ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/720p/720p50_shields_ter.yuv
9	Netflix_Tango	https://media.xiph.org/video/derf/ElFuente/Netflix_Tango_4096x2160_60fps_10bit_420.y4m
10	Netflix_Narrator	https://media.xiph.org/video/derf/ElFuente/Netflix_Narrator_4096x2160_60fps_10bit_420.y4m
11	Netflix_Crosswalk	https://media.xiph.org/video/derf/ElFuente/Netflix_Crosswalk_4096x2160_60fps_10bit_420.y4m

На каждом видеофайле выбран объект для поиска. Пример объектов для поиска приведён на рисунке 4.8. Стоит заметить, что объектом для поиска может служить только изображение с наличием особенных (ключевых) точек объекта, т. е. монотонное изображение без ярко выделенных линий, точек не может быть

принято в качестве объекта для поиска. В таблице 4.7 приведена информация о видеопотоках, на которых осуществлялся поиск различных объектов.



Рисунок 4.8 – Объекты для поиска в видеопоследовательностях pedestrian_area и Netflix_SquareAndTimelapse

Получены новые видеопоследовательности путём масштабирования в разрешениях: 640×360 , 720×405 , 848×480 , 960×540 , 1024×576 , 1280×720 , 1600×900 , 1920×1080 . Каждый видео файл обработан методами поиска объекта ASIFT, SURF и DEV. Результаты производительности усреднены между десяти запусками программной системы с различными видеопотоками. На рисунке 4.9 представлены графики зависимости частоты обработки кадров от разрешения видеопоследовательности. Vuforia SDK и Kudan SDK выпускаются только на мобильные платформы и не позволяют организовать поиск объекта на стационарной вычислительной машине для сравнения в одинаковых условиях с другими методами. Методы поиска объекта из библиотеки Vuforia SDK и Kudan SDK перед выполнением поиска объекта уменьшают размер изображения до 640×480 точек и во время поиска могут принимать решения о пропуске некоторых кадров из процесса анализа для достижения реальной скорости обработки [78]. Следовательно, размер 640×480 – является максимальным для выполнения поиска библиотеками Vuforia SDK и Kudan SDK в реальном масштабе времени (минимум 25 кадров в секунду).

Таблица 4.7 – Информация о видеопотоках

Номер видеофайла (см. таблицу 4.6)	Размер кадра	Относительная площадь целевого объекта от площади кадра	Количество перекрытий объекта	Общее количество кадров	Количество кадров с изображением объекта	Длительность видеопотока, сек.
1	1920× 1080	<20%	9	14315	326	596,46
2	1920× 1080	< 2%	5	375	155	15
3	4096× 2160	< 2%	36	600	284	10,01
4	4096× 2160	<20%	132	600	408	10
5	704× 576	< 10%	7	480	187	16
6	720× 480	< 10%	4	300	300	10
7	720× 480	< 10%	0	300	300	10
8	1280× 720	< 10%	0	504	504	10,08
9	4096× 2160	< 5%	102	294	192	4,9
10	4096× 2160	< 5%	0	600	600	10,01
11	4096× 2160	< 4%	112	300	112	5

Средние ошибки первого и второго рода результатов обработки видеопотоков с различными разрешениями разработанной программной системы представлены в таблице 4.8.

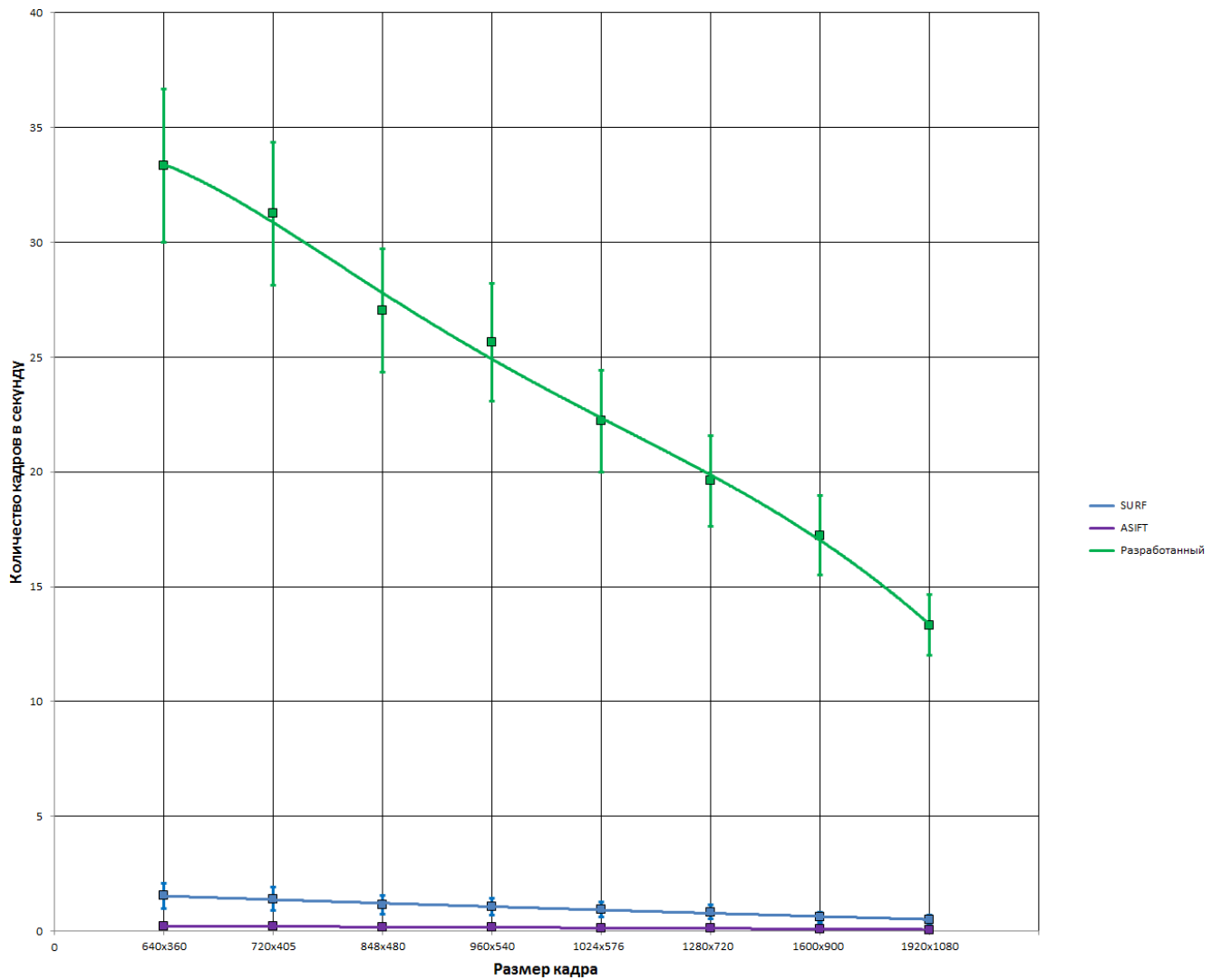


Рисунок 4.9 – Результаты сравнения скорости обработки на различных размерах кадров

Таблица 4.8 – Ошибки первого и второго рода результатов выполнения программной системы, основанной на разработанном методе и алгоритмах

В процентах

Вид ошибки	Номер видеофайла (см. таблицу 4.6)										
	1	2	3	4	5	6	7	8	9	10	11
I-ого рода	4,85	6,35	5,86	5,82	4,46	4,35	5,28	3,61	4,34	2,42	3,87
II-ого рода	2,37	2,22	3,41	4,56	2,67	3,72	2,49	3,95	2,52	2,86	2,65

На основании анализа результатов можно сказать, что разработанную программную систему поиска объекта можно применять в режиме реальной скорости обработки данных (25 кадров в секунду) на разрешении до 960×540 точек с ошибкой первого рода до $\approx 7\%$ и второго рода до $\approx 5\%$. Добавление функции идентификации увеличивает время обработки лишь первого кадра, благодаря конвейерной архитектуре системы (время выполнения функции идентификации меньше времени выполнения поиска ключевых точек или вычисления дескрипторов). Стоит отметить, что время выполнения процесса инициализации изображения объекта в разработанном методе занимает от 20 секунд до 4 минут в зависимости от количества найденных ключевых точек.

В таблице 4.9 представлена производительность программной системы с предложенным алгоритмом нахождения областей претендентов и с использованием алгоритма RANSAC. Видеопотоки имеют размеры кадра 1920×1080 точек. На основании результатов таблицы можно сказать, что программная система с разработанным алгоритмом в отличие от использования алгоритма RANSAC выполняется быстрее до $\approx 10\%$.

Таблица 4.9 – Скорости обработки программной системы с алгоритмами сопоставления дескрипторных множеств

В кадрах в секунду

Алгоритм	Номер видеофайла (см. таблицу 4.6)										
	1	2	3	4	5	6	7	8	9	10	11
RANSAC	13,54	14,21	13,48	14,18	12,75	13,87	13,34	13,04	14,16	13,42	12,51
Разработанный	14,42	14,87	14,67	14,23	13,45	15,24	14,61	13,41	14,94	14,24	13,16
Различие, %	6,50	4,64	8,83	0,35	5,50	9,88	9,52	2,84	5,51	6,11	5,20

Значения средней загрузки процессоров и средней загрузки процессоров в секунду на обработку видеопоследовательности, состоящей из 944 кадров (отрезок от видео pedestrian_area), представлены в таблице 4.10 и в таблице 4.11 соответственно.

Таблица 4.10 – Средняя загрузка процессоров

В процентах

SURF	ASIFT	Vuforia SDK	Kudan SDK	DEV
83,57	40,46	61,25	52,50	94,02

Таблица 4.11 – Средняя загрузка процессоров на обработку видеопоследовательности из 944 кадров в секунду

В процентах в секунду

SURF	ASIFT	Vuforia SDK	Kudan SDK	DEV
0,09	0,04	1,57	1,31	1,84

SURF, Vuforia SDK и разработанный метод DEV имеют среднюю загрузку процессоров выше 60%. Показатели средней загрузки процессоров у методов Vuforia SDK, Kudan SDK и DEV при обработке видеопоследовательности из 944 кадров в секунду являются намного выше показателей методов SURF и ASIFT. Таким образом, на основании приведённых значений в таблицах 4.10 и 4.11 можно сказать, что методы Vuforia SDK и DEV имеют самые высокие показатели эффективности из рассмотренных методов, а значит, эффективнее используют все предоставленные вычислительные ресурсы. Однако разработанный метод в отличие от метода Vuforia SDK обрабатывает видеопоследовательность без уменьшения размеров. Исходя из результатов, разработанный метод обрабатывает кадр размером 640×480 быстрее, чем 30 кадров в секунду, что является выше скорости обработки методом из Vuforia SDK, однако не учитывается время на выполнение процесса масштабирования кадра до размера 640×480.

Не всякое изображение может быть принято в качестве целевого изображения объекта. В таблице 4.12 приведены ограничения к объекту для поиска.

Таблица 4.12 – Ограничения к целевому объекту для поиска

Ограничение	Значение
Минимальный размер изображения объекта	20×20 точек
Максимальное изменение (увеличение/уменьшение) размера изображения объекта в кадре	До 20 раз
Цветовая характеристика	Цветовые гистограммы изображений образца и объекта в кадре должны соответствовать по мере Бхаттачария > 80%
Изменение яркости	Значение детерминанта матрицы Гессе для ключевой точки должно быть не меньше 300

4.2.5 Применение метода для видео, полученных в реальных условиях

Реализованную систему можно применить в различных областях. В области метеорологии можно применить систему для нахождения скорости и направления ветра по перемещению областей на изображениях.

Мощным источником информации о различных параметрах атмосферы и подстилающей поверхности (в том числе и об облачных образованиях) являются данные спутникового пассивного зондирования Земли. Одним из открытых источников является набор инструментов API Google карт. Спутниковые данные над участком с заданными координатами предоставляются с интервалом в 1 час. В этом случае по данным о перемещении облачных образований можно оценить важные метеорологические параметры: скорость и направление ветра. Эта информация необходима для метеорологических прогнозов [81]. Оценку этих параметров можно представить в виде следующей последовательности:

1. Выбор области облачного поля.

2. Поиск выбранного облачного поля на кадре разработанным методом.
3. Определение смещения центра найденной области.
3. Оценка скорости и направления ветра по перемещению контролируемых (кандидатов) областей на изображениях.

Смещение центра найденной области определяется по координатам x и y (см. рисунок 4.10). Имея информацию о расстоянии от спутника до Земли, частоту снимков и смещение облачного поля (\bar{x}, \bar{y}) , рассчитывается скорость ветра.

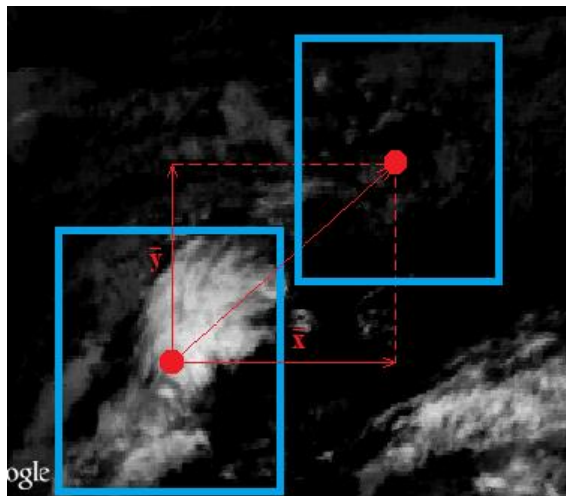
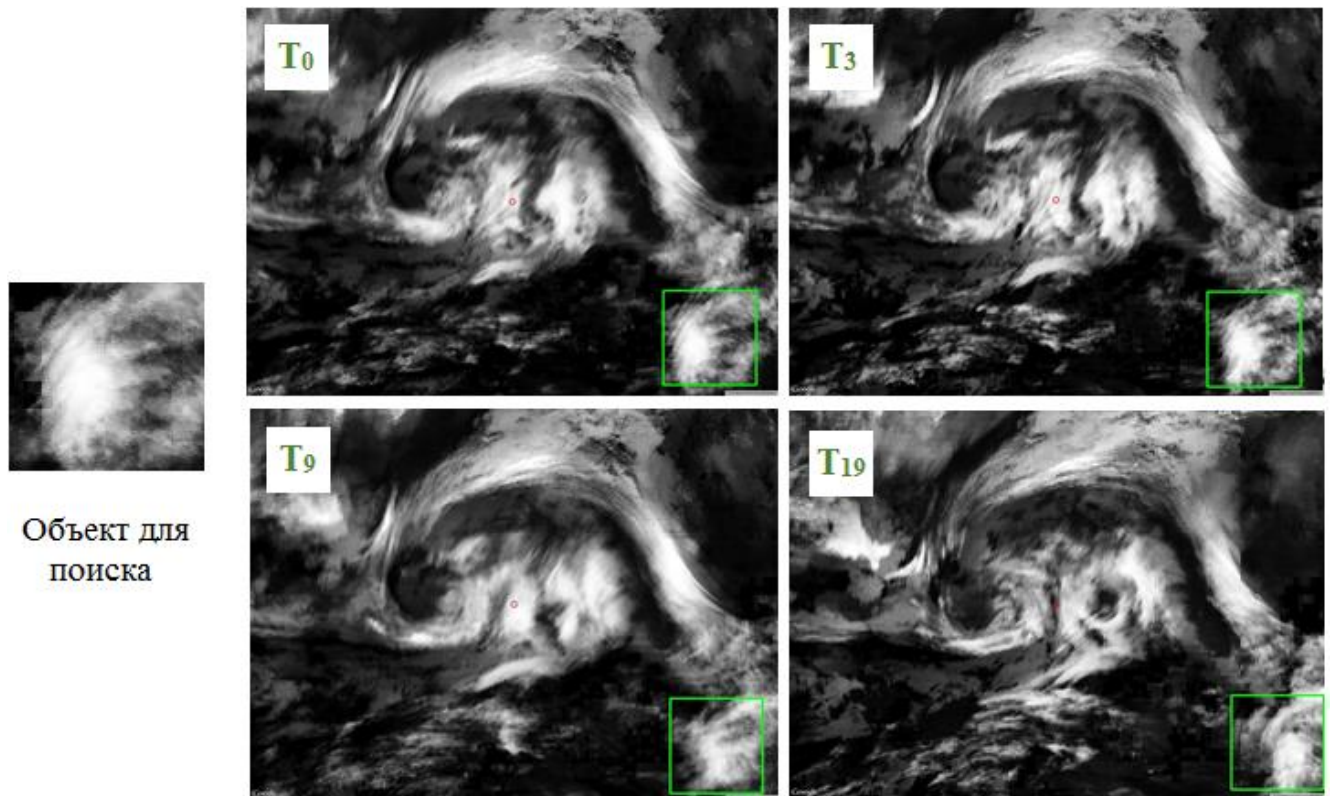


Рисунок 4.10 – Определение смещения центра найденной области

Во всех проведённых тестах слежение за выбранным облачным полем успешно осуществляется до 3 часов. После 3 часов слежения облачное поле может существенно изменять свою форму, отличительные признаки.

Результаты одного из эксперимента представлены на рисунке 4.11 [82]. Объект успешно обнаруживается в течение 23 снимков, после чего объединился с другим облачным полем и приобрёл новые свойства, а, следовательно, изображение объекта стало иметь новые ключевые точки.



T_0 – снимок, полученный в момент времени t_0 , T_3 – снимок, полученный в момент времени $t_0+3\Delta t$, T_9 – снимок, полученный в момент времени $t_0+9\Delta t$, T_{19} – снимок, полученный в момент времени $t_0+19\Delta t$

Рисунок 4.11 – Поиск и слежение за облачным полем

Разработанная система используется в Федеральном государственном бюджетном учреждении науки Институт оптики атмосферы им. В.Е. Зуева Сибирского отделения Российской академии наук (ИОА СО РАН) для вычисления скорости и направления движения крупных (50×50 км и более) облачных образований на период до 3 часов. Копия акта внедрения программно-технической системы «Object tracking PS» приведена в Приложении А.

Система может быть применена в системах безопасности для нахождения предмета или живого существа в видеопотоке, который может поступать напрямую с камеры или из архивных данных.

Для проведения эксперимента использованы видео контенты из открытого источника Xiph.org [49] (см. таблицы 4.6, 4.7). На каждом видеофайле выбран

объект для поиска. Шаги выполнения поиска изображения объекта в архивных данных могут быть представлены в следующей последовательности:

1. Загрузка изображения объекта для поиска (инициализация объекта).
 2. Инициализация видео архива, на котором будет осуществляться поиск (загрузка пути до директории с видео файлами).
 3. На каждом видео файле архива выполнить поиск загруженного объекта.
- Результатом работы описанных шагов является набор кадров с выделенными участками найденного объекта из архива с видео данными.

На рисунке 4.12 представлен пример работы программы поиска объекта [83], взятого из видео контента pedestrian_area.



Рисунок 4.12 – Поиск человека в видеопотоке

В результате выполнения программы выделены участки кадров видеопоследовательности с найденными изображениями объекта (см. рисунок 4.12). В течение всей видео последовательности искомый объект перекрывается 5 раз другими объектами, после которых успешно локализуется [84]. Метод

устойчив к частичному перекрытию, к временному отсутствию объекта, к изменению общей яркости изображения.

В приложении А приведена копия акта внедрения программного модуля, основанного на разработанном методе и алгоритме поиска объекта в видеопотоке, в коммерческий продукт игрового движка GameX, поставляемого компанией ООО «Универсальные терминал системы». Включение программного модуля позволило повысить скорость обработки на 5% заменой существующего функционала и добавить новую функциональность: слежение и поиск объекта в реальном масштабе времени в видео кадрах размером 640×360 с частотой поступления до 30 кадров в секунду.

4.3 Выводы

1. Использование разработанной методики, основанной на выделении ключевых точек и анализе цветовой гистограммы при идентификации объекта, позволяет обнаруживать проективно искажённый объект с вероятностью $\approx 0,9$.

2. Реализованные алгоритм и на его основе программная система позволяют повысить быстродействие системы приблизительно в 8 – 10 раз по сравнению с существующими аналогами.

3. Реализованная программная система обеспечивает поиск объекта с реальной скоростью потока данных (25 кадров в секунду) на кадрах с размером до 960×540 пикселей, что позволяет применять её для практических задач в различных функциональных приложениях.

4. Результаты экспериментального исследования системы поиска объекта в видеопотоке показывают, что за счёт комбинированного поиска локальных (ключевых точек) и глобальных (цветовой гистограммы) признаков разработанный метод по скоростным характеристикам и вероятности распознавания превосходит существующие «промышленные» системы.

Заключение

В диссертации решена задача разработки инвариантного к проективным преобразованиям и условиям съёмки алгоритма поиска заданного объекта в видеопотоке. Практическая реализация подтвердила теоретическое обоснование разработанной системы для применения в задачах с реальной скоростью потока видеопоследовательности (25 кадров в секунду). Исследование выполнено полностью: по проекту системы создан программный продукт «Object tracking PS», и на реальных задачах подтверждена его эффективность.

По результатам исследования сформулированы основные результаты:

1. Предложенная методика поиска объекта в видеопотоке, основанная на анализе ключевых точек и использовании алгоритмов идентификации, инвариантных к проективным искажениям изображения, позволяет повысить вероятность нахождения объекта с $\approx 0,8$ до уровня $\approx 0,9$.

2. Разработанный метод поиска объекта в видеопотоке, включающий шаги вычисления дескрипторов ключевых точек по предварительно искаженному проективными преобразованиями изображению образца и оценки меры схожести дескрипторов кадра и дескрипторов объекта, позволяет повысить скорость обработки до ≈ 10 раз по сравнению с быстросейственным аналогом SURF.

3. Разработанный алгоритм ограничения области поиска объекта в кадре, основанный на оценке масштабов ключевых точек проективно искаженных изображений объекта, обеспечивает уменьшение времени поиска объекта до $\approx 10\%$ по сравнению с алгоритмом RANSAC.

4. Разработанные в диссертации функциональная и структурная модели системы позволяют полностью удовлетворить основные требования, выдвигаемые к системе поиска объекта в видеопотоке, а, следовательно, существенно повысить эффективность вычислений и использования имеющихся вычислительных ресурсов.

5. Предложенный метод поиска объекта в видеопотоке в отличие от существующих позволяет осуществлять инвариантный к проективным

искажениям поиск объекта в режиме реальной скорости потока данных (25 кадров в секунду).

6. Разработанная программная система поиска объекта в видеопотоке позволяет использовать аппаратно-программные средства со средней загрузкой процессора на обработку видеопотока в секунду $L_s \approx 1,84\%$.

7. Реализованная программная система поиска объекта в видеопотоке, ориентированная на использовании всех ресурсов компьютера, включая видеопроцессор, позволяет решать широкого круга задачи в различных функциональных областях.

8. Результаты испытаний программной системы «Object tracking PS», основанной на разработанной архитектуре, подтвердили эффективность заложенных алгоритмов и технологий.

Рекомендации.

Разработанная система может быть применена для решения широкого круга задач (слежение за объектом, обнаружение объектов, поиск дубликатов изображений, соединения кадров) в различных предметных областях (зондирование земли, системы безопасности, обработка видео). На основе разработанного алгоритма и построенной структуре создана программная система «Object tracking PS», направленная для выполнения научных вычислений и ориентированная на применение в научно-исследовательском процессе в исследовательских и конструкторских организациях.

Дальнейшее развитие системы возможно с добавлением одновременного поиска нескольких объектов на одном кадре, что расширит круг потенциальных потребителей.

Список литературы

1. Анштедт, Т. Видеоаналитика: мифы и реальность [Текст] / Т. Анштедт, И. Келлер, Х. Лутц. – М.: Security Focus, 2012 г. – 176 с. – 1500 экз. – ISBN 978-5-9901176-5-5.
2. Guoshen, Yu. ASIFT: An Algorithm for Fully Affine Invariant Comparison, Image Processing On Line [Электронный ресурс] / Yu. Guoshen, M. Jean-Michel // Image Processing On Line. – 2011 – №1. – Электрон. текст. дан. – Режим доступа: <http://www.ipol.im/pub/art/2011/my-asift/article.pdf> (дата обращения: 07.07.2015).
3. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features [Текст] / P. Viola // Accepted Conference on Computer Vision and Pattern Recognition (CVPR 2001). – 2001. – С. 511–518.
4. Соколов, С. М. Система технического зрения для информационного обеспечения автоматической посадки и движения по ВПП летательных аппаратов [Текст] / С. М. Соколов, А. А. Богуславский, Н. Г. Фёдоров, П. В. Виноградов // Известия ЮФУ. Технические науки. – 2015. – № 1(162). – С. 96–109.
5. Филатов, В. И. Система обучения визуальным понятиям на основе соотнесения лексем и ключевых точек [Текст] / В. И. Филатов, А. С. Потапов // Научно-технический вестник информационных технологий, механики и оптики. – 2016. – Т. 16, № 4. – С. 689–696.
6. Roth, D. The SNoW Learning Architecture [Текст] / D. Roth // Technical Report UIUCDCS–R–99–2102, UIUC Computer Science Department. – 1999. – С. 8–10.
7. Суворов, Д. А. Алгоритм сопровождения TLD (aka Predator) [Электронный ресурс] / Д. А. Суворов // Разработка роботов. – 2012. – Электрон. дан. – Режим доступа: <http://robot-develop.org/archives/4463> (дата обращения: 15.09.2016).
8. Дязитдинов, Р. Р. Применение фильтра Калмана для оценки параметров освещённости [Текст] / Р. Р. Дязитдинов, Н. Н. Васин; под ред. С. А. Прохорова // Перспективные информационные технологии в научных исследованиях, проектировании и обучении (ПИТ 2012): труды научно–технической конференции с международным участием и элементами научной школы

молодёжи, посвящённой 40-летию кафедры информационных систем и технологий СГАУ. – Самара: Издательство Самарского научного центра РАН. – 2012. – С. 21–25.

9. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints [Текст] / D. G. Lowe // International of computer vision. – 2004. – С. 28–29.

10. Mann, S. «GlassEyes»: The Theory of EyeTap Digital Eye Glass [Электронный ресурс], / S. Mann // IEEE Technology and Society. – 2012. – Электрон. дан. – Режим доступа: <http://wearcam.org/glass.pdf> (дата обращения: 05.02.2016).

11. Development Edition [Электронный ресурс] / Официальный сайт Microsoft. – 2016. – Электрон. дан. – Режим доступа: <https://www.microsoft.com/microsoft-hololens/en-us> (дата обращения: 19.08.2013).

12. Meet Kinect for Windows [Электронный ресурс] / Официальный сайт Microsoft. – 2016. – Электрон. дан. – Режим доступа: <https://dev.windows.com/en-us/kinect> (дата обращения: 21.08.2013).

13. Metaio [Электронный ресурс] / Официальный сайт Metaio. – 2016. – Электрон. дан. – Режим доступа: <http://www.metaio.com/> (дата обращения: 24.05.2014).

14. Vuforia 5.5 SDK [Электронный ресурс] / Vuforia Developer Portal. – 2016. – Электрон. дан. – Режим доступа: <https://developer.vuforia.com/downloads/sdk> (дата обращения: 12.02.2016).

15. Kudan SDK 1.2.3 version [Электронный ресурс] / Kudan Augmented Reality. – 2016. – Электрон. дан. – Режим доступа: <https://www.kudan.eu/download/> (дата обращения: 17.03.2016).

16. Kenneth, D.-H. A Practical Introduction to Computer Vision with OpenCV [Текст] / D.-H. Kenneth. – Ireland: Trinity College Dublin, Ireland, 2014, – 234 с. – ISBN 978-1-118-84845-6.

17. Продукты Kipod для интеллектуального видеонаблюдения [Электронный ресурс] / Официальный сайт Синезис. Видеоаналитика. – 2015. – Электрон. дан. – Режим доступа: <http://synesis.ru/products> (дата обращения: 14.10.2015).

18. Об утверждении приоритетных направлений развития науки, технологии и техники в Российской Федерации и перечня критических технологий Российской Федерации: [Указ Президента РФ №899 от 7 июля 2011 г.] // Российская газета. – 2011. №6018. – С. 42.

19. Алпатов, Б. А. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление [Текст] / Б. А. Алпатов, П. В. Бабаян, О. Е. Балашов, А. И. Степашкин, – М.: Радиотехника, 2008 г., – 176 с. – ISBN 978-5-88070-201-5.

20. Трекинг (компьютерная графика) [Электронный ресурс]: Материал из Википедии – свободной энциклопедии / Авторы Википедии // Википедия, свободная энциклопедия, – Электрон. дан. – Сан-Франциско: Фонд Викимедиа, 2014. – Режим доступа: <http://ru.wikipedia.org/?oldid=67077116>.

21. Уоссермен, Ф. Нейрокомпьютерная техника: Теория и практика – Neural Computing. Theory and Practice [Текст] / Ф. Уоссермен – М.: Мир, 1992. – 240 с. – ISBN 5-03-002115-9.

22. Beaudet, P. Rationally Invariant Image Operations [Текст] / P. Beaudet // International Joint Conference on Pattern Recognition. – 1978. – С. 579–583.

23. Forstner, W. A feature based correspondence algorithm for image matching [Текст] / W. Forstner // International Archives of Photogrammetry and Remote Sensing. – 1986. – № 5. – 150–166.

24. Rad, A. A. Fast Circle Detection Using Gradient Pair Vectors [Текст] / A. A. Rad, K. Faez, N. Qaragozlou // Proc. VIIth Digital Image Computing: Techniques and Applications. – 2003. – С. 879–887.

25. Краснобаев, А. А. Обзор алгоритмов детектирования простых элементов изображения и анализ возможности их аппаратной реализации [Электронный ресурс] / А. А. Краснобаев // Институт прикладной математики им. М. В. Келдыша Российской академии наук. – 2005. – Электрон. текст. дан. – Режим доступа: http://www.keldysh.ru/papers/2005/prep114/prep2005_114.html (дата обращения: 17.06.2015).

26. Lucas, B. D. An iterative image registration technique with an application to stereo vision [Текст] / B. D. Lucas, T. Kanade // Proceedings of Imaging Understanding Workshop. – 1981. – С. 121–130.

27. Гаганов, В. А. Инвариантные алгоритмы сопоставления точечных особенностей на изображениях [Электронный ресурс] / В. А. Гаганов // Сетевой журнал - Компьютерная графика и мультимедия. – 2009. – Электрон. текст. дан. – Режим доступа: http://cgm.computergraphics.ru/issues/issue17/invariant_features (дата обращения: 17.05.2015).

28. Vaumberg, A. Reliable Feature Matching Across Widely Separated Views [Текст] / A. Vaumberg // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2000). – 2000. – С. 774–781.

29. Скурихин, А. В. Применение методов масштабируемого пространства в обработке сигналов [Электронный ресурс] / А. В. Скурихин // Лаборатория АНИ, СПИИРАН. – 2014. – Электрон. текст. дан. – Режим доступа: <http://www.docme.ru/doc/541964/primenenie-metodov-masshtabiruemogo-prostranstva-v> (дата обращения: 23.04.2016).

30. Bauhad, J. Uniqueness of the Gaussian kernel for scale-space filtering [Текст] / J. Bauhad, A. P. Witkin, M. Baudin, R. O. Duda // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1986. – №1 – С. 26–33.

31. Шубин, М. А. Лекции об уравнениях математической физики [Текст]: 2-е изд., испр. / М. А. Шубин. – М.: МЦНМО, 2003. – 303 с. – ISBN 5-900916-97-9.

32. Кострикин, А. И. Линейная алгебра и геометрия [Текст] / А. И. Кострикин, Ю. И. Манин. – М.: Наука, 1986. – 304 с.: ил.

33. Bay, H. SURF: Speeded Up Robust Features [Текст] / H. Bay, A. Ess, T. Tuytelaars, L. Van Gool // Computer Vision and Image Understanding (CVIU), Vol. 110. – 2008. – № 3. – С. 346–359.

34. Mishkin, D. MODS: Fast and Robust Method for Two-View Matching [Текст] / D. Mishkin, M. Perdoch, J. Matas // Computer Vision and Image Understanding (CVIU), Vol. 141, – 2015. – С. 81–93.

35. Vinukonda, P. A study of the scale – invariant feature transform on a parallel pipeline [Электронный ресурс] / P. Vinukonda // В. TECH., JNTU University. – 2011. – Электрон. Текст. дан. – Режим доступа: http://etd.lsu.edu/docs/available/etd-04272011-105721/unrestricted/Vinukonda_Phaneendra_Thesis.pdf (дата обращения: 06.04.2016).

36. Pele, O. A Linear Time Histogram Metric for Improved SIFT Matching [Текст] / O. Pele, M. Werman // Proceedings of the 10th European conference on computer vision. – 2008. – №3 – С. 495–508.

37. Sivic, J. Efficient Visual Search for Objects in Videos [Текст] / J. Sivic, A. Zisserman // Proceedings of the IEEE, Vol. 96. – 2008. – № 4. – С. 548 – 566.

38. Золотых, Н. Ю. Обзор методов поиска и сопровождения транспортных средств на потоке видеоданных [Текст] / Н. Ю. Золотых, В. Д. Кустикова, И. Б. Мееров // Информационные технологии, Вестник Нижегородского университета им. Н. И. Лобачевского. – 2012. – №5 (2). – С. 348–358.

39. Решмин, Б. И. Имитационное моделирование и системы управления [Текст] / Б. И. Решмин. – М.: Инфра-Инженерия, 2016. – 74 с. – ISBN 978-5-9729-0120-3.

40. Джиган, В. И. Прикладная библиотека адаптивных алгоритмов [Текст] / В. И. Джиган // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. – 2006. – № 1. – С. 60–65.

41. Вежневцев, А. П. Методы классификации с обучением по прецедентам в задаче распознавания объектов на изображениях [Электронный ресурс] / А. П. Вежневцев // Лаборатория Компьютерной Графики и Мультимедиа факультета ВМиК, Московский государственный университет им. М.В.Ломоносова, Москва, Россия. – 2006. – Электрон. текст. дан. – Режим доступа: http://www.graphicon.ru/2006/fr10_34_VezhnevetsA.pdf (дата обращения: 14.10.2014).

42. Гурьянов, А. В. Метод Виолы–Джонса (Viola–Jones) как основа для распознавания лиц [Электронный ресурс] / А. В. Гурьянов. – 2011. – Электрон.

дан. – Режим доступа: <http://habrahabr.ru/post/133826/> (дата обращения: 05.07.2014).

43. Вежневцев, А. П. Boosting – Усиление простых классификаторов [Электронный ресурс] / А. П. Вежневцев, В. П. Вежневцев // Компьютерная графика и мультимедия, Сетевой журнал. – 2006. – Электрон. текст. дан. – Режим доступа: <http://cgm.computergraphics.ru/content/view/112> (дата обращения: 29.09.2015).

44. Garibotto, G. B. Object Detection and Tracking from Fixed and Mobile Platforms [Текст] / G. B. Garibotto, F. Buemi // Image Analysis and Processing ICIAP 2015: 18th International Conference Genoa. – Springer, 2015. – Part 2. – С. 631 – 642.

45. Пастушков, А. В. Программная система слежения за объектом в видеопотоке [Текст] / А. В. Пастушков, В.Т. Калайда // Известия высших учебных заведений. Физика. – 2013. – Т. 56, №8/3. – С. 334–335.

46. Потапов, А. А. Новейшие методы обработки изображений [Текст] / А. А. Потапов, Ю. В. Гуляев, С. А. Никитов, А. А. Пахомов, В. А. Герман; под ред. А. А. Потапова. – М.: ФИЗМАТЛИТ, 2008 г. – 496 с. – ISBN 978-5-9221-0841-6.

47. Грузман, И. С. Цифровая обработка изображений в информационных системах [Текст]: учебное пособие / И. С. Грузман, В. С. Киричук, В. П. Косых, Г. И. Перетягин, А. А.Спектор. – Новосибирск: Изд-во НГТУ, 2002 г. – 352 с. – ISBN 5-7782-0330-6.

48. Пастушков, А. В. Методы и алгоритмы поиска объекта на видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Сборник научных трудов SWorld. – 2013. – Т. 6, № 3. – С. 38–42.

49. Xiph.org Video Test Media [derf's collection] [Электронный ресурс] / The Xiph.Org Foundation. – 2016. – Электрон. дан. – Режим доступа <https://media.xiph.org/video/derf/> (дата обращения: 02.03.2016).

50. Пастушков, А. В. Метод поиска объекта на изображении [Текст] / А. В. Пастушков, В. Т. Калайда // Известия высших учебных заведений. Физика. – 2015. – Т. 58, № 10/3. – С. 117–119.

51. System Engineering Fundamentals. Department of defense [Текст]. – Systems Management College. – Defense acquisition university press, 2001. – 216 с.

52. Panchal, P. M. A Comparison of SIFT and SURF [Электронный ресурс] / P. M. Panchal, S. R. Panchal, S. K. Shah // International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2. – 2013. – Электрон. текст. дан. – Режим доступа: http://www.ijircce.com/upload/2013/april/21_V1204057_A%20Comparison_H.pdf (дата обращения: 01.02.2016).

53. Luo, J. A Comparison of SIFT, PCA-SIFT and SURF [Электронный ресурс] / J. Luo, G. Oubong // International Journal of Image Processing (IJIP) Volume(3), Issue(4). – 2009. – Электрон. текст. дан. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.7041&rep=rep1&type=pdf> (дата обращения: 26.10.2015).

54. Кустикова, В. Д. Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP. Учебный курс. Лекция «Отслеживание движения и алгоритмы сопровождения ключевых точек» [Электронный ресурс] / В. Д. Кустикова // Нижегородский государственный университет им. Н.И. Лобачевского. Факультет вычислительной математики и кибернетики. – 2013. – Электрон. текст. дан. – Режим доступа: <http://www.hpcc.unn.ru/file.php?id=735> (дата обращения: 19.09.2016).

55. Shan, Y. Linear Model Hashing and Batch RANSAC for Rapid and Accurate Object Recognition [Электронный ресурс] / Y. Shan, B. Matei, H. S. Sawhney, R. Kumar // Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2004. – №2. – Электрон. дан. – Режим доступа: http://ri.cmu.edu/pub_files/pub4/shan_y_2004_1/shan_y_2004_1.pdf (дата обращения: 12.11.2013).

56. David, F. Multi-model Estimation in the Presence of Outliers [Электронный ресурс] / F. David // The Faculty of the Computer Science Department of Middlebury College. – 2011. – Электрон. дан. – Режим доступа: http://www.cs.cmu.edu/~dfouhey/thesis/dfouhey_thesisPresentation.pdf (дата обращения: 14.05.2015).

57. Meng, J. Object Instance Search in Videos via Spatio-Temporal Trajectory Discovery [Текст] / J. Meng, J. Yuan, J. Yang, G. Wang, Y.-P. Tan // IEEE Transactions on multimedia, Vol. 18. – 2016. – № 1. – С. 116 – 127.

58. Верников, Б. М. Лекция 8: Алгоритмы для задач о паросочетаниях [Электронный ресурс] / Б. М. Верников, А. М. Шур // Уральский федеральный университет, Институт математики и компьютерных наук, кафедра алгебры и дискретной математики, – Электрон. дан. – Режим доступа к файлу: <http://kadm.imkn.urfu.ru/files/tgr08.pdf> (дата обращения: 06.08.2015).

59. Дуда, Р. Распознавание образов и анализ сцен [Текст] / Р. Дуда, П. Харт. – М.: Мир, 1976. – 511 с.

60. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Текст]. – М.: Стандартинформ, 2005. – 116 с.

61. Гонсалес, Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное [Текст] / Р. Гонсалес, Р. Вудс. – Москва: Техносфера, 2012. – 1104 с. – ISBN 978-5-94836-331-8.

62. Деврой, Л. Непараметрическое оценивание плотности. L1-подход [Текст] / Л. Деврой, Л. Дьёрфи; пер. с англ. А. Б. Цыбакова; под ред. М. Б. Малютова. – М.: Мир, 1988. – 408 с. – ISBN 5-03-000445-0.

63. Comaniciu, D. Mean Shift: A Robust Approach Toward Feature Space Analysis [Текст] / D. Comaniciu, P. Meer // IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5), 2002. – С. 603–619.

64. Конушин, А. С. Сопровождение объектов и распознавание событий / А. С. Конушин [Электронный ресурс] // Computer science club, Екатеринбург, – 2012. – Электрон. текст. дан. – Режим доступа: http://logic.pdmi.ras.ru/csclub/sites/default/files/slides/20120414_videorecognition_konushin_lecture04.pdf (дата обращения: 03.03.2015).

65. Пастушков, А. В. Программная система слежения за объектом в видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Южно-Сибирский научный вестник. – 2013. – №1. – С. 90–91.

66. Recommendation ITU-R BT.601-7. Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios [Текст], – International Telecommunication Union, 2011 – 18 с.

67. Шапошников, А. И. Компьютерное слежение с масштабированием, основанное на градиентном спуске [Электронный ресурс] / А. И. Шапошников, Е. В. Шапошникова // 21st International Conference on Computer Graphics and Vision, Moscow, Russia, September 26-30, – Электрон. версия печатн. публ. – 2011. – Режим доступа: <http://www.graphicon.ru/proceedings/2011/conference/gc2011shaposhnikov.pdf> (дата обращения: 14.09.2015).

68. Пастушков, А. В. Инвариантность компонент цветковых пространств к изменению естественной яркости на видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Сборник научных трудов SWorld. – 2014. – Т. 6, № 1. – С. 54–56.

69. Chidambaram, C. Intelligent Data Engineering and Automated Learning [Текст] / C. Chidambaram, S. M. Marlon, B. D. Leyza, V. N. Hugo, S. L. Heitor // 13th International Conference. Natal. Brazil. – Springer, 2012. – С. 143–151.

70. Буч, Г. UML. Руководство пользователя [Текст] / Г. Буч, Д. Рамбо, А. Якобсон. – М.: ДМК пресс, 2001. – 432 с. – ISBN 5-94074-144-4.

71. Крэг, Л. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание [Текст] / Л. Крэг; пер. с англ. А.Ю. Шелестова. – М.: ООО «И. Д. Вильямс», 2013. – 736 с. : ил. – 1000 экз. – ISBN 978-5-8459-1185-8.

72. Степанов, А. Н. Архитектура вычислительных систем и компьютерных сетей [Текст] / А. Н. Степанов. – СПб.: Питер, 2007. – 509 с.: ил. – ISBN 978-5-469-01451-5.

73. Магда, Ю. С. Ассемблер для процессоров Intel Pentium [Текст] / Ю.С. Магда. – СПб.: Питер, 2006. – 410 с.: ил. – ISBN 5-469-00662-X.

74. Боресков, А. В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA [Текст]: учеб. пособие / А. В. Боресков, А. А. Харламов и др.; предисл.: В. А. Садовничий. – М.: МГУ, 2012. – 336 с. – ISBN 978-5-211-06340-2.

75. Программа слежения за объектом в видеопотоке «Object tracking PS»: свидетельство о государственной регистрации программы для ЭВМ № 2014662190 Российская Федерация / Пастушков А. В.; правообладатель – Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет» (RU). Зарегистрировано в Реестре программ для ЭВМ 25 ноября 2014 г.

76. Obuchowski, N. A. Receiver operating characteristic curves and their use in radiology [Текст] / N. A. Obuchowski // Radiology 229, 2003. – №1. – С. 3–8.

77. Пастушков, А. В. Сравнение качеств методов поиска объекта на видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Международный союз учёных «НАУКА. ТЕХНОЛОГИИ. ПРОИЗВОДСТВО». – 2014. – №2 / 2014. – С. 21–24.

78. Maximum marker size [Электронный ресурс] / Vuforia Developer Portal. – 2016. – Электрон. дан. – Режим доступа: <https://developer.vuforia.com/forum/creating-ar-trackables/maximum-marker-size> (дата обращения 07.04.2016).

79. Орлов, С. А. Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. Стандарт третьего поколения [Текст] / С. А. Орлов, Б. Я. Цилькер. – СПб.: Питер, 2012. – 608 с.: ил. – 2000 экз. – ISBN 978-5-459-01101-2.

80. Лутц, М. Программирование на Python, том II, 4-е издание [Текст] / М. Лутц; пер. с англ. А. Киселева. – СПб.: Символ-Плюс, 2011. – 992 с., ил. – ISBN 978-5-93286-211-7.

81. Пастушков, А. В. Метод поиска и слежения за облачными полями на изображении [Текст] / А. В. Пастушков, А. И. Елизаров, В. Т. Калайда // Оптика атмосферы и океана. Физика атмосферы: тезисы докладов XXI международного симпозиума. Томск, 22–26 июня 2015 г. – Томск, 2015. – С. 156–158.

82. Pastushkov, A. V. Search and tracking method of cloud fields on image / A. V. Pastushkov, V. T. Kalayda // SPIE Proceedings. – 2015. – Vol. 9680. – 96805V. – 4 p. – DOI: 10.1117/12.2205433.

83. Пастушков, А. В. Метод поиска человека на видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Мир науки и инноваций. – 2015. – Вып. 1(1), т. 3. – С. 39–42.

84. Пастушков, А. В. Программа поиска и слежения за человеком на видеопотоке [Текст] / А. В. Пастушков, В. Т. Калайда // Наука и образование в жизни современного общества: сборник научных трудов по материалам международной научно-практической конференции. Тамбов, 30 апреля 2015 г. – Тамбов, 2015. – Т. 12. – С. 108–109.

Приложение А
(справочное)

Справки и акты об использовании результатов диссертационной работы

ООО "Универсальные терминал
системы»

Юридический адрес:

634021, г. Томск, ул. Сибирская 31-39

Фактический адрес:

634045 г. Томск, ул. Мокрушина 9,
стр. 40 оф. 19

Почтовый адрес:

г. Томск, ул. Мокрушина 9, стр. 40
оф. 21

ИНН 7017252707/ КПП 701701001


ОГРН 1097017021991

ОКПО 64100549

ОКАТО 69401372000

УТВЕРЖДАЮ

Директор компании ООО
«Универсальные терминал
системы»,

 Куимов А. А.
« 20 » 05 2016 г.

Акт

о внедрении

Комиссия в составе управляющего директора компании ООО «Универсальные терминал системы» Куимова Алексея Анатольевича и руководителя it-отдела Чернявского Сергея Юрьевича составили настоящий акт о том, что:

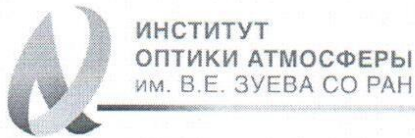
1. Включение программного модуля, основанного на разработанном в диссертационной работе Пастушкова А. В. методе и алгоритме поиска объекта на видеопотоке, в коммерческий продукт игрового движка GameX, поставляемого компанией ООО «Универсальные терминал системы», позволило повысить скорость обработки кадра на 5%.
2. Внедрённый программный модуль позволил добавить новую функциональность в игровой движок GameX, за счёт поиска целевого изображения объекта на видео кадрах размером 640×360 с частотой поступления до 30 кадров в секунду.

Члены комиссии:



Куимов А.А.

Чернявский С.Ю.



**ИНСТИТУТ
ОПТИКИ АТМОСФЕРЫ**
им. В.Е. ЗУЕВА СО РАН

ФАНО России

Федеральное государственное бюджетное
учреждение науки
Институт оптики атмосферы им. В.Е. Зуева
Сибирского отделения
Российской академии наук
(ИОА СО РАН)

Пл. Академика Зуева, д.1, г. Томск, 634055
тел.: (3822) 492 738, факс: (3822) 492 086
e-mail: mgg@iao.ru, www.iao.ru
ОКПО 03534050, ОГРН 1027000880268
ИНН/КПП 7021000893/701701001

УТВЕРЖДАЮ

Заместитель директора по научной
работе,
д.ф.-м.н., профессор
Белан Б.Д.



« 15 » 05 2016 г.

А К Т

ВНЕДРЕНИЯ ПРОГРАММНО-ТЕХНИЧЕСКОЙ СИСТЕМЫ «Object tracking PS»

Комиссия в составе: председатель Белов В. В. и член комиссии: Елизаров А. И. составили настоящий акт о том, что программно-техническая система слежения за объектом в видеопотоке «Object tracking PS», разработанная при выполнении диссертационной работы Пастушкова А. В. (научный руководитель разработки Калайда В. Т.) используется в Федеральном государственном бюджетном учреждении науки Институт оптики атмосферы им. В.Е. Зуева Сибирского отделения Российской академии наук (ИОА СО РАН) для решения вычислительных задач атмосферной оптики.

Использование указанной системы позволяет:

- вычислять скорость и направление движения крупных (50×50 км и более) облачных образований;
- прогнозировать перемещение облачных образований на период до 3 часов.

Внедрение системы позволило сократить погрешность оценки скорости и направления ветра до $\pm 10\%$.

Председатель комиссии,
зав. лаб. распространения оптических
сигналов, д.ф.-м.н.

 Белов В. В.

с.н.с. лаб. лидарных методов,
к.т.н.

 Елизаров А. И.

Приложение Б
(справочное)

Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ
№ 2014662190

**«Object tracking PS». Программа слежения за объектом в
видеопотоке**

Правообладатель: *Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский государственный
университет» (RU)*

Автор: *Пастушков Александр Викторович (RU)*

Заявка № **2014619989**
Дата поступления **06 октября 2014 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **25 ноября 2014 г.**



*Врио руководителя Федеральной службы
по интеллектуальной собственности*

 Л.Л. Кирий

Приложение В
(справочное)

Текст скрипта программы для вычисления значений ROC кривой

```
import sys
import subprocess
import re
import os

def execute(params, log_file_name="") :
    if isinstance(params, str):
        params = params.split()

    # execute
    _p = subprocess.Popen(params, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, shell=False)

    # write log
    if log_file_name != "":
        _f = open(log_file_name, "w")
        _out, _err = _p.communicate()
        _f.write(str(_out))
        _f.write(str(_err))
        _f.close()

    return _p.returncode

def find_result_in_file(log_file, expression):
    try:
        _file = open(log_file, 'r')
```

```

    _data = _file.read()
    _file.close()
except (IOError, OSError):
    print "find_result_in_file() can not open file " + log_file
    return ""

_data = _data.replace('\r', ")
_find_result = re.findall(expression, _data)
return _find_result

help_line = "Correct command line is following:\nr\nrun.py <method
(DEV/ASIFT/SIFT/SURF)> <image with refference object> <input stream>
<refference results file> <count of stream frames>"
if(len(sys.argv) != 6) :
    print help_line
    quit()

method = sys.argv[1]
if(method != "DEV" and method != "ASIFT" and method != "SURF" and method !=
"SIFT") :
    print "ERROR: Incorrect method name!\n" + help_line
    quit()

ref_object = sys.argv[2]
video = sys.argv[3]
result_file = sys.argv[4]
count_of_frames = int(sys.argv[5])
ref_results = find_result_in_file(result_file, "(\\d+)")
if(len(ref_results) == 0) :
```

```
print "ERROR: " + result_file + " doesn't contain numbers of files with refference  
object!"
```

```
threshold = 0.0
```

```
index = 0
```

```
while threshold <= 1.0 :
```

```
    TP = 0
```

```
    TN = 0
```

```
    FP = 0
```

```
    FN = 0
```

```
    method_results_log = "results" + str(index) + ".txt"
```

```
    execute(["object_searcher_ps.exe",    method,    ref_object,    video,    "0"],  
method_results_log)
```

```
    method_results_list = find_result_in_file(method_results_log, "(\\d)")
```

```
for i in range(0, count_of_frames) :
```

```
    if(str(i) in ref_results) :
```

```
        manual_result_flag = True
```

```
    else :
```

```
        manual_result_flag = False
```

```
    if(str(i) in method_results_list) :
```

```
        method_result_flag = True
```

```
    else :
```

```
        method_result_flag = False
```

```
    if(manual_result_flag == method_result_flag) :
```

```
        if(manual_result_flag) :
```

```
            TP += 1
```

```
        else :
```

```
TN += 1
else :
    if(manual_result_flag) :
        FN += 1
    else :
        FP += 1

FPR = FP / float(FP + TN)
TPR = TP / float(TP + FN)
print ("%f\t%f" % (FPR, TPR))

threshold += 0.05
index += 1
```