|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | ИУ «Информатика и системы управления» |

|  |  |
| --- | --- |
| КАФЕДРА | ИУ-1 «Системы автоматического управления» |

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

|  |
| --- |
|  |
|  |
| ***Распознавание и наблюдение за объектом*** |
| ***на видеофрагменте*** |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ1-61Б |  | 21/05/2021 |  | Д.И. Юдаков |
|  | (Группа) |  | (Подпись, дата) |  | (И.О. Фамилия) |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель | 21/05/2021 |  | К.В. Парфентьев |
|  | (Подпись, дата) |  | (И.О. Фамилия) |

*2020 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заведующий кафедрой | | | | | | | ИУ-1 | | |
|  | | | | | | | (Индекс) | | |
|  |  | | | |  | К.А. Неусыпин | | | |
|  | | | | | | (И.О. Фамилия) | | | |
|  | « | 11 | » | сентября | | | 20 | 20 | г. |

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| по теме | | Распознавание и наблюдение за объектом на видеофрагменте | | |
|  | | | | |
|  | | | | |
| Студент группы | ИУ1-61Б | |  | |
|  | | | | |
| (Фамилия, имя, отчество) | | | | |
| Направленность НИР (учебная, исследовательская, практическая, производственная, др.) | | | | |
| исследовательская | | | | |
| Источник тематики (кафедра, предприятие, НИР) | | | | НИР |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| График выполнения НИР: | | 25% к |  | н., 50% к |  | н., 75% к |  | н., 100% к |  | н. |
|  | | | | | | | | | | |
| ***Техническое задание*** | Исследовать современные подходы и методики к решению задачи | | | | | | | | | |
| распознавания и наблюдения за объектом на видеофрагменте. | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |

***Оформление научно-исследовательской работы:***

|  |  |  |
| --- | --- | --- |
| Расчетно-пояснительная записка на | ***29*** | листах формата А4. |
| Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.) | | |
| оформление графического материала в НИР не предусмотрено | | |
|  | | |
|  | | |
|  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | « | 08 | » | февраля | 20 | 21 | г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель НИР** |  | 08/02/2021 |  | К.В. Парфентьев |
|  |  | (Подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |
| **Студент** |  | 08/02/2021 |  | Д.И. Юдаков |
|  |  | (Подпись, дата) |  | (И.О. Фамилия) |

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc73296595)

[ВВЕДЕНИЕ 4](#_Toc73296596)

[1. ОБНАРУЖЕНИЕ И СОПОСТАВЛЕНИЕ ПРИЗНАКОВ 5](#_Toc73296597)

[2. МЕТОДЫ НАХОЖДЕНИЯ ОСОБЫХ ТОЧЕК И ДЕСКРИПТОРОВ. 7](#_Toc73296598)

[2.1. SURF. 7](#_Toc73296599)

[2.2. BRISK. 9](#_Toc73296600)

[2.3. Harris Corner Detector. 10](#_Toc73296601)

[2.4. FAST 11](#_Toc73296602)

[2.5. Minumum Eigenvalue 12](#_Toc73296603)

[2.6. SIFT 12](#_Toc73296604)

[2.7. BRIEF 15](#_Toc73296605)

[2.8. ORB 16](#_Toc73296606)

[3. СРАВНЕНИЕ МЕТОДОВ ПОИСКА ОСОБЫХ ТОЧЕК 18](#_Toc73296607)

[ЗАКЛЮЧЕНИЕ 25](#_Toc73296608)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 26](#_Toc73296609)

[ПРИЛОЖЕНИЯ 27](#_Toc73296610)

# ВВЕДЕНИЕ

На сегодняшний день учёные из разных стран занимаются разработкой алгоритмов, позволяющих научить компьютер видеть так же, как и видит сам человек. Если для людей получать необходимую информацию посредством зрительного канала является чем-то простым и само собой разумеющимся, то обучить компьютер подобным вещам является и по сей день не выполнимой задачей. Множество IT корпораций работают над её решением, но это требует большого вложения человеческого труда, финансовых затрат и вычислительных мощностей.

Но существуют методы, которые позволяют, хоть и при использовании в узконаправленных задачах, получить желаемый результат при меньших затратах. Они основаны не на структуре человеческого аппарата анализа и интерпретации изображений, а непосредственно на особенностях самого изображения. Одни из таких методов основаны на нахождении особых точек и их численного описания. Основываясь только на наборе таких данных цифрового изображения можно с достаточно высокой точностью позволить компьютеру работать с визуальными образами подобно человеку.

В данной работе сравниваются несколько наиболее популярных и современных методов поиска особых точек и расчёта их дескрипторов, по результатам распознавания объекта на видеофрагменте.

Поставленной целью является создание программы для исследования методов поиска особых точек на кадрах видеофрагмента с последующим трекингом желаемого объекта с помощью языка высокого уровня MATLAB.

Для достижения цели будет необходимо выполнять ряд задач. Поставленные задачи:

1. Сделать обзор теоретического материала по обнаружению и сопоставлению признаков (ключевых точек).
2. Найти наиболее популярные и эффективные методы поиска особых точек и расчёта их дескрипторов и описать их;
3. Провести исследование этих методов, сравнивая их по скорости работы, эффективности и точности.

# ОБНАРУЖЕНИЕ И СОПОСТАВЛЕНИЕ ПРИЗНАКОВ

Обнаружение и сопоставление признаков является важной задачей во многих применениях компьютерного зрения, таких как автоматизация отслеживания объектов, программной стабилизации изображения (стереокалибровка), навигация роботов, поиск и индексация изображений и многих других. Признак – это часть информация, которая является релевантной для решения вычислительной задачи, связанной с определённым применением. Особенностями или признаками могут быть специфические структуры на изображении, такие как точки, края или объекты. Признаки могут быть разделены на две основные категории:

1. Те, которые в конкретных местах изображений, например, горные вершины, углы зданий, дверные проёмы или участки снега с интересной формой. Эти виды локализованных признаков часто называются *ключевыми особенностями или ключевыми точками* (или углами) и описываются появлением кусков из пикселей, окружающих местоположение точки.
2. Те, которые могут быть сопоставлены в зависимости от их ориентации и локального внешнего вида. Эти виды называются *рёбрами* и они могут быть хорошими индикаторами границ объекта и перекрытия с другими объектами в последовательности изображений.

Тремя основными компонентами обнаружения и сопоставления признаков являются:

1. Детектирование. Необходимо определить *точку интереса*.
2. Описание. Локальный внешний вид вокруг каждой ключевой точки описывается каким-то образом, который инвариантен при изменениях освещения, масштаба и вращения в плоскости. Обычно выводится вектор-дескриптор для каждой характерной точки.
3. Соответствие. Дескрипторы сравниваются по изображениям, чтобы определить аналогичные признаки. Например, для двух изображений можно получить набор пар , где – признаки на одном изображении, а – признаки на другом изображении.

Точка интереса – это выделяющаяся по текстуре точка изображения или точка, в которой резко меняется направление границы объекта (например, угол) или точка пересечения между двумя или более сегментами рёбер.

Свойства точки интереса:

1. Имеет чётко определённое положение в пространстве изображения или хорошо локализована.
2. Стабильно при локальных и глобальных возмущениях в области изображения, так что при изменении освещённости или яркости точки интереса могут быть рассчитаны с высокой степенью повторяемости.
3. Обеспечивают эффективное обнаружение.

Основными подходами к определению точек интереса являются методы, основанные на яркости изображении и методы, основанные на извлечении границ. Алгоритмы для идентификации точек интереса: Детектор Харриса Корнера (Harris Corner Detector), SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), BRISK (Binary Robust Invariant Scalable Keypoints), FAST (Features from Accelerated Segment Test), ORB (Oriented FAST and Rotated BRIEF).

Дескриптор признаков – это алгоритм, который находим векторы признаков на изображении. Он кодирует важную информацию в ряд чисел и выступают в качестве числового «отпечатка пальца», который можно использовать для отличия одних признаков от других. В идеале эта информация должна быть инвариантной при преобразовании изображения, поэтому те же векторы признаков можно найти и в преобразованном изображении. После обнаружения точки интереса вычисляется дескриптор для каждой из них. Дескрипторы можно разделить на два класса:

1. Локальный дескриптор. Это компактное представление локальной окрестности точки. Локальные дескрипторы пытаются иметь сходную форму и внешний вид только в локальной окрестности точки и поэтому очень подходят для её представления с точки зрения сопоставления.
2. Глобальный дескриптор. Глобальный дескриптор описывает всё изображение. Как правило, он не очень надёжен, поскольку изменение части изображения может привести к его сбою.

Алгоритмы для определения дескрипторов признаков: SIFT, SURF, BRISK, BRIEF (Binary Robust Independent Elementary Features), ORB.

Сопоставление признаков или, как правило, сопоставление изображений является задачей установления соответствия между двумя изображениями одной и той же сцены или объекта. Основной подход к сопоставлению изображений заключается в обнаружении набора точек интереса, каждая из которых связана с дескрипторами изображения из данных изображения. После того, как признаки и их дескрипторы будут извлечены из двух или более изображений, следующим шагом будет установление некоторых предварительных совпадений признаков между этими изображениями.

Как правило, производительность методов сопоставления на основе точек интереса зависит от выбора связанных с ними дескрипторами изображений. Таким образом, в приложениях должны использоваться детекторы и дескрипторы, соответствующие содержанию изображений. Например, если изображение содержит клетки бактерий, следует использовать детектор каплей, а не угловой. Но, если изображение представляет собой вид на город сверху, то угловой детектор подходит для поиска большинства рукотворных конструкций. Кроме того, очень важен выбор детектора и дескриптора, который решает проблему деградации изображения. Основными алгоритмами для сопоставления признаков являются Brute-Force Matcher и FLANN Matcher (Fast Library for Approximate Nearest Neighbors).

Исходя из вышенаписанного можно составить алгоритм обнаружения и сопоставления признаков:

1. Найти набор отличительных ключевых точек.
2. Определить область вокруг каждой ключевой точки.
3. Извлечь и нормализовать содержимое окрестности ключевых точек.
4. Вычислить локальный дескриптор из нормализованной окрестности.
5. Сопоставить локальные дескрипторы.

# МЕТОДЫ НАХОЖДЕНИЯ ОСОБЫХ ТОЧЕК И ДЕСКРИПТОРОВ.

## SURF.

Алгоритм метода SURF:

1. Выполняется поиск особых точек с помощью матрицы Гессе

где матрица Гессе, функция изменения градиента яркости.

1. Для найденных точек вычисляется направление наибольшего изменения яркости.
2. Перебираются масштабы матрицы Гессе по октавам.
3. Формируются дескрипторы. Для метода SURF как правило дескриптор состоит из 64 чисел.

Матрица Гессе имеет инвариантность относительно вращения изображения. Однако она не инвариантна к изменению масштаба (в многократных размерах), поэтому метод SURF сначала использует фильтры различных масштабов для вычисления Гессианов и затем уже для каждой особой точки вычисляется градиент и масштаб. Градиент в точке вычисляется при помощи фильтров Хаара. Размерность фильтра берётся равной где масштаб точки.

После того как были найдены особые точки метод SURF вычисляет их дескрипторы. Дескриптор представляет собой набор из 64 чисел для каждой ключевой точки. Каждое число отображает разницу градиента вокруг особой точки, а каждая особая точка представляет собой максимум Гессиана, что гарантирует факт того, что в окрестности этой точки будут участки с разными градиентами. Таким образом, обеспечивается различие дескрипторов для разных особых точек, вследствие чего образуется инвариантность дескриптора относительно вращения. Размер области, на которой считается дескриптор, определяется масштабом матрицы Гессе, что обеспечивает инвариантность относительно масштаба.

Алгоритм построения дескриптора следующий:

1. Вокруг точки строится квадратная окрестность размером , где масштаб, на котором получено максимальное значение детерминантна матрицы Гессе.
2. Полученная квадратная область разбивается на блоки, в результате область будет разбита на 4x4 региона (рис. 1).

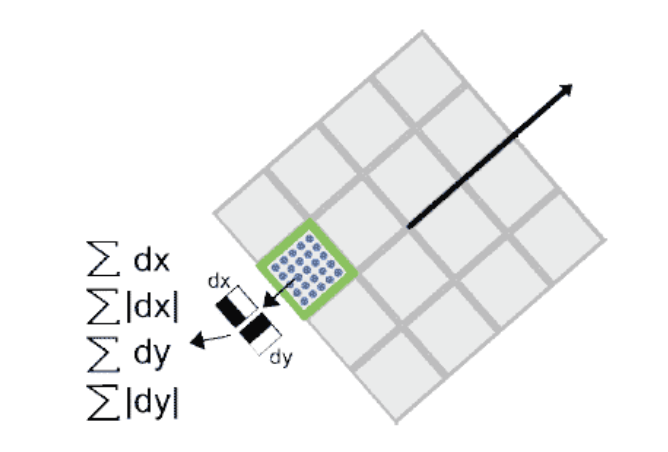


Рис. 1. Область, разбитая на 4x4 региона.

1. Для каждого блока вычисляются более простые признаки. Как следствие, получается вектор, содержащий 4 компоненты: 2 – это суммарный градиент по квадранту, 2 – сумма модулей точечных градиентов.
2. Дескриптор формируется в результате склеивания взвешенных описаний градиента для 16 квадрантов вокруг особой точки. Элементы дескриптора взвешиваются на коэффициенты Гауссова ядра. Веса необходимы для большей устойчивости к шумам в удалённых точках.
3. Дополнительно к дескриптору заносится след матрицы Гессе. Эти компоненты необходимы, чтобы различать тёмные и светлые пятна. Для светлых точек на тёмном фоне след отрицателен, а для тёмных точек на светлом фоне – положителен.

Метод SURF также используется для поиска объектов. Тем не менее, дескриптор никак не использует информацию об объектах. SURF рассматривает изображение как единое целое и выделяет особенности всего изображения, поэтому он плохо работает с объектами простой формы.

## BRISK.

Основное отличие метода BRISK от метода SURF: метод оценивает истинный масштаб каждой ключевой точки в непрерывном масштабе пространства.

К круговым окрестностям потенциальных ключевых точек применяется Гауссово сглаживание. Для определения направления ключевой точки используется сумма локальных градиентов.

Для построения дескриптора область вокруг особой точки разбивается на 60 участков (рис. 2):

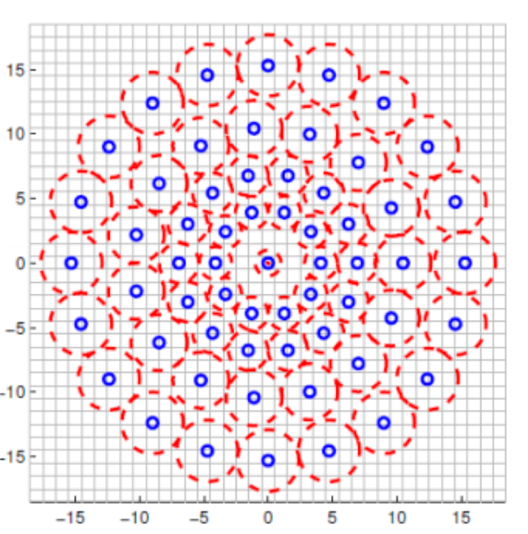


Рис. 2. Область вычисления дескриптора.

Множество разбивается на два подмножества:

где размер особой точки.

Вычисляется среднее значения градиента множества :

Дескриптор состоит из бинарной строки длиной 512, заполненной результатами проведённых тестов в множестве :

где интенсивность окрестности радиуса точки ,

угол направления градиента

## Harris Corner Detector.

По результатам исследований детектор Харриса является одним из самых оптимальных детекторов углов. Данный метод является улучшением метода Моравеца, авторы которого рассматривают производные изменения яркости изображения для исследования изменений яркости по всем направлениям.

Для изображения рассмотрим некоторое окно (как правило, размер окна , но может зависеть от размера изображения) с центром в точке , а также его сдвиг на (рис. 3).

Сумма квадратов разностей между исходным и сдвинутым окном будет равна:

где весовая функция,

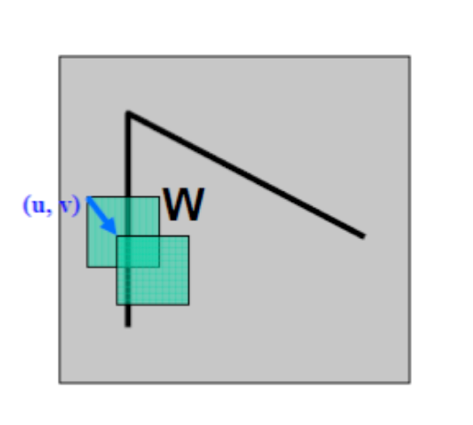


Рис. 3. Окно с центром в (x,y) и его сдвиг на (u,v).

автокорреляционная матрица:

При больших изменениях функции по всевозможным направлениям определяется угол, следовательно получаются большие по модулю собственные значения матрицы . Ниже приведено расположение собственных значений (рис. 4).

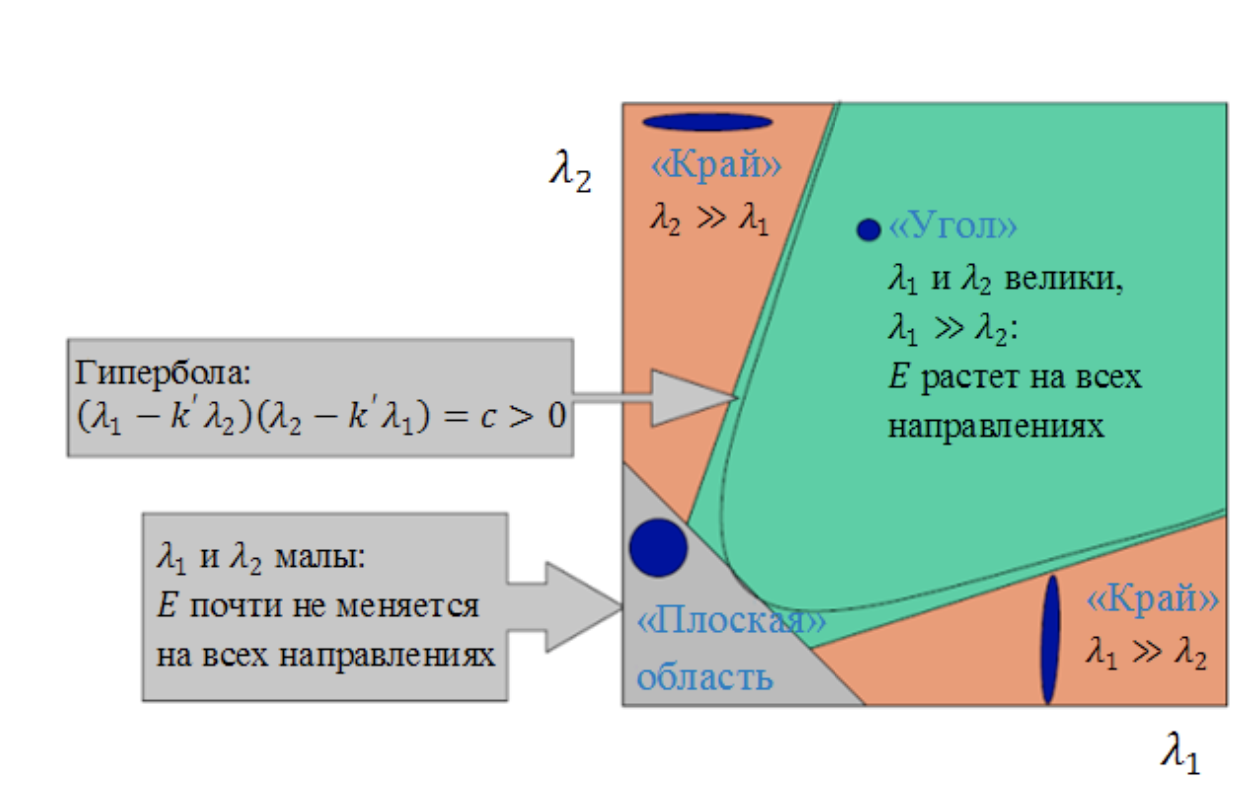


Рис. 4. Расположение собственных значений.

В связи с трудоёмкостью вычисления собственных чисел матрицы Харрис и Стефан предложили ввести меру отклика:

где эмпирическая константа, . В таком случае, значение будет положительно для угловых особых точек. Затем убираются точки, которые меньше некоторого порога . Далее вычисляются локальные максимумы функции в окрестности заданного радиуса и выбираются в качестве угловых особых точек.

Метод Харриса инвариантен к вращению, а также к аффинным изменениям яркости, однако он чувствителен к шумам и зависит от масштаба изображения (в инос случае используют многомасштабный метод Харриса (Multi-scale Harris Detector)).

## FAST

Вышеописанные методы определяют особые точки, применяя свои алгоритмы напрямую к пикселям исходного изображения. Существует альтернативный подход, заключающийся в использовании алгоритмов машинного обучения для тренировки классификатора точек на некотором множестве изображений. Метод FAST строит деревья решения для классификации пикселей.

Для каждого пикселя рассматривается окружность радиусом 4, вписанная в квадратную область со стороной 7 пикселей (рис. 5).

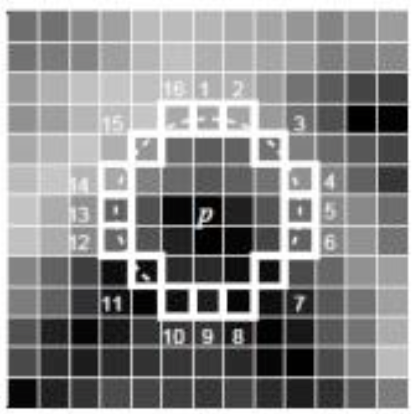


Рис. 5. Рабочая окрестность пикселя при использовании FAST детектора.

Окружность проходит через 16 пикселей. Каждый из пикселей окружности относительно пикселя может находиться в одном из 3х состояний:

Для каждого и найденного для каждого ( множество всех пикселей тренировочного набора изображений) разделим на 3 подмножества точек, которые темнее, схожи либо светлее точки соответственно. Затем строится дерево решений. По результатам построенного дерева определяются углы на тестовых изображениях.

## Minumum Eigenvalue

Метод Ши-Томаси (Shi-Tomasi или Kanade-Tomasi) очень схож с методом Харриса, но отличается при вычислении меры отклика: алгоритм напрямую вычисляет значение . Делается предположение, что при использовании этого алгоритма поиск углов будет более стабильным.

## SIFT

Основные преимущества метода SIFT:

1. Локальность. Признаки локальные, поэтому они устойчивы к перестановкам и перекрытиям.
2. Отличительность. Отдельные признаки могут быть сопоставлены с большой базой данных объектов.
3. Количество. Множество признаков могут быть сгенерированы даже для небольших объектов.
4. Эффективность. Близкая к производительности в режиме реального времени.
5. Расширяемость. Может быть легко расширен до широкого спектра различных типов признаков, каждый из которых добавляет устойчивость.

Алгоритм SIFT метода:

1. Выбор пика масштабного пространства. Потенциальное расположение для поиска признаков.
2. Локализация ключевых точек. Точное расположение ключевых точек признаков.
3. Назначение ориентации. Назначение ориентации ключевым точкам.
4. Дескриптор ключевых точек. Описание ключевых точек как вектора большой размерности.
5. Соотнесение ключевых точек.

Подробно рассмотрим каждый пункт алгоритма. Выбор пика масштабного пространства связан с тем, что объекты реального мира имеют смысл только в определённом масштабе. Например, если посмотреть на кубик сахара на столе в масштабе планеты или даже всего Млечного Пути, то его попросту не существует. Масштабное пространство изображения – это функция , которая создаётся в результате свёртки гауссового ядра (размытия) в разных масштабах со входным изображением. Масштабное пространство разделено на октавы, а количество октав и масштаб зависит от размера исходного изображения. Таким образом создаётся несколько октав исходного изображения. Размер изображения каждой октавы составляет половину предыдущего

В октаве изображения постепенно размываются с помощью оператора Гаусса. С математической точки зрения «размывание» является свёрткой гауссовского оператора и изображения. Результатом является размытое изображение:

где это гауссовский оператор размытия ( масштаб):

а изображение.

Затем, полученные размытые изображения используются для создания другого набора изображений – разницы гауссианов. Они получаются вычитанием гауссовского размытия изображения с двумя разными (масштабами). Этот процесс выполняется для различных октав изображения в гауссовской пирамиде.

Разница гауссианов используется для расчёта лаплациана гауссовских аппроксимаций, которые являются инвариантными масштабу. Один пиксель в изображении сравнивается с его 8 соседями, а также с 9 пикселями в предыдущем и в следующем масштабе. Таким образом, в общем сложности проводится 26 проверок. Если точка является локальным экстремумом, то это потенциальная ключевая точка.

Ключевых точек, сгенерированных на предыдущем шаге, оказывается слишком много, и часть из них лежат вдоль края или у них недостаточно контраста. В обоих случаях они не так полезны, как признаки, поэтому существует необходимость в избавлении от таких точек. Подход аналогичен подходу, используемому в детекторе углов Харриса для удаления краевых элементов. Для признаков с низкой контрастной мы проверяем их интенсивность.

До сих пор уже были определены легитимные ключевые точки, протестированные на стабильность. Также уже известен масштаб, в котором была обнаружена ключевая точка, таким образом существует масштабная инвариантность. Следующее, что необходимо – назначить ориентацию каждой ключевой точке, чтобы сделать её инвариантной вращению. Для этого в окрестности каждой ключевой точки в зависимости от масштаба рассчитываются величина и направление градиента. Затем выбирается наибольшее значение среди всех градиентов для данной точки. Таким образом создаются ключевые точки с одинаковым расположением и масштабом, но разными направлениями.

Далее необходимо вычислить дескриптор для локальной области изображения о каждой ключевой точке, которая является очень отличительной и инвариантной, насколько это возможно, к таким вариациям, как вращение и освещённость. Для этого берётся окно размером 16х16 вокруг ключевой точки, разделённое на 16 подблоков размером 4х4. Для каждого подблока создаётся 8 гистограмм ориентации. Таким образом 128 значений представляются в виде вектора признаков для формирования дескриптора ключевой точки. Этот дескриптор имеет несколько осложнений:

1. Зависимость от вращений. Вектор признаков используется ориентации градиентов, поэтому при повороте изображения, всё поменяется. Для достижения независимости вращения поворот ключевой точки вычитается из каждой ориентации. Таком образом, каждая ориентация градиента становится зависимой от ориентации ключевой точки.
2. Зависимость от освещённости. Если взять большие пороговые значения, то можно достичь независимости от освещённости. Таким образом, любое число (из 128), которое больше 0,2, меняется на 0,2. Нормализация результирующего вектора даёт независимый от освещённости вектор признаков.

Ключевые точки между двумя изображениями сопоставляются путём идентификации ближайших соседей. Но в некоторых случаях второе ближайшее совпадение может быть очень близко к первому, например, по причине шума. В этом случае берётся отношение ближайшего расстояния к второму ближайшему расстоянию. Если это отношение больше 0,8, то сопоставление отклоняется. Эта методика позволяет устранить 90% ложных совпадения и в то же время отбрасывает 5% правильных совпадений.

## BRIEF

Метод BRIEF работает с изображениями на пиксельном уровне, поэтому он очень чувствителен к шуму. За счёт предварительного сглаживания участка изображения эту чувствительность можно снизить, там самым повысив стабильность и повторяемость дескрипторов. Для сглаживания изображения используется гауссово ядро.

После сглаживания необходимо создать двоичный вектор признаков. Для этого создаётся вектор двоичных признаков двоичных тестовых ответов:

где интенсивность в точке . Выбор множества пар составляет множество двоичных ответов, длина двоичного вектора признаков,

Создание вектора длины бит оставляет много вариантов выбора пары тестовых мест . Пара также называется случайной парой, которая расположена внутри куска изображения. Существует пять подходов выбора. Рассмотрим кусок изображения размером и предположим, что ключевая точка находится внутри этого куска (рис. 6).

1. Равномерное (G I): и пиксели , и пиксели в случайной паре взяты из равномерного распределения вокруг ключевой точки. Пара может лежать близко к границе куска.
2. Гауссиан (G II): и пиксели , и пиксели в случайной паре взяты из гауссовского (нормального) распределения вокруг ключевой точки.
3. Гауссиан (G III): первый пиксель в случайной паре взят из гауссовского распределения, сосредоточенного вокруг ключевой точки с отклонением , второй пиксель – . Это заставляет тестовую пару быть более локальной. Тестовые пары вне куска изображения прижимаются к краю.
4. Грубая полярная сетка (G IV): и пиксели , и пиксели в случайной паре отбираются из дискретных мест грубой полярной сетки, вводя пространственное квантование.
5. Грубая полярная сетка (G V): Первый пиксель в случайной паре находится в , а второй пиксель в случайной паре взят из дискретных мест грубой полярной сетки.

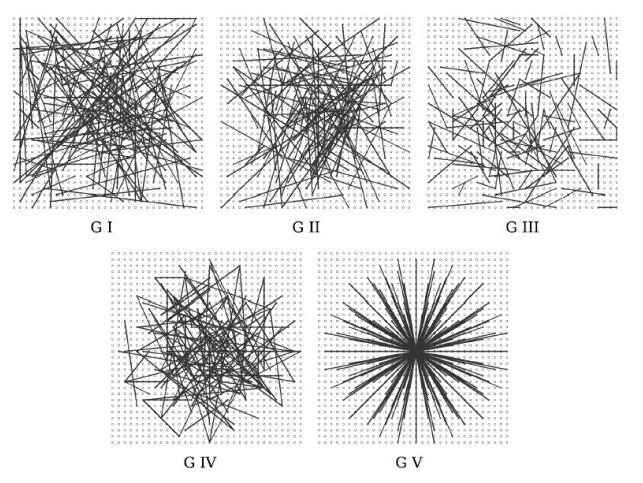


Рис. 6. Различные подходы к выбору тестовых пар.

Наконец, дескриптор BRIEF выглядит следующим образом:

Преимуществами BRIEF являются скорость и точность распознавания, если инвариантность к сильным вращениям в плоскости не является обязательным требованием.

## ORB

ORB справляется с задачей обнаружения так же хорошо, как и SIFT, и даже лучше, чем SURF, будучи на два порядка быстрее. ORB основан на детекторе ключевых точек FAST и дескрипторе BRIEF. Обе эти техники привлекательны из-за хорошей производительности. Основные отличия от этих техник:

1. Добавление быстрого и точно компонента ориентации в FAST.
2. Эффективные расчёт ориентированных признаков BRIEF.
3. Анализ дисперсии и корреляции ориентированных признаков BRIEF.
4. Метод обучения декорреляции признаков BRIEF при ротационной инвариантности, что приводит к лучшей производительности в применениях ближайшего соседа.

Поскольку BRIEF неинвариантен вращению, то ORB использует rBRIEF аналог, алгоритм которого выглядит следующим образом:

1. Запуск каждого теста (тестовой пары) со всеми тренировочными кусками изображения.
2. Упорядочивание тестов по их расстоянию от среднего значения 0,5, образовывая вектор .
3. Жадный поиск:
   1. Первый тест запоминается в вектор и удаляется из .
   2. Следующий тест сравнивается со всеми тестами в . Если его абсолютная корреляция больше порогового значения, тест удаляется, иначе добавляется к .
   3. Предыдущий шаг повторяется, пока в не будет 256 тестов. Если их оказалось меньше, необходимо повысить порог и прогнать поиск заново.

# СРАВНЕНИЕ МЕТОДОВ ПОИСКА ОСОБЫХ ТОЧЕК

Сравнение методов поиска особых точек и построения дескрипторов будем производить на основе программы, код которой расположен в приложении А. На вход данной программы подаётся видеофрагмент, на первом кадре которого выделяется необходимый для отслеживания объект. Затем на каждом последующем кадре ищется тот же объект и выделяется прямоугольником. Результатом выполнения программы является видео с выделенным объектом на кадрах и соотношение количества кадров, на которых объект был найден ко всему количеству кадров.

Поскольку некоторые из детекторов являются детекторами углов, то тестирование методов будем производить на видео с кубиком Рубика, первый кадр которого показан на рисунке 7 (именно на первом кадре будет выделяться объект для исследования).

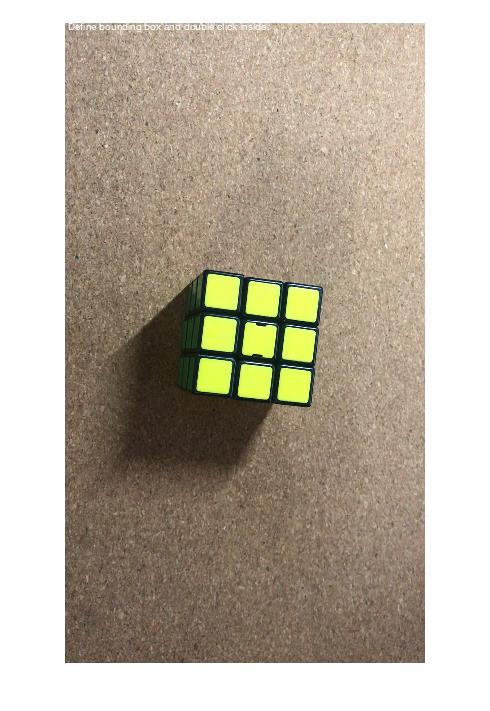


Рис. 7. Первый кадр из видео.

Ключевые точки, найденные с помощью BRISK метода, изображены на рисунке 8.

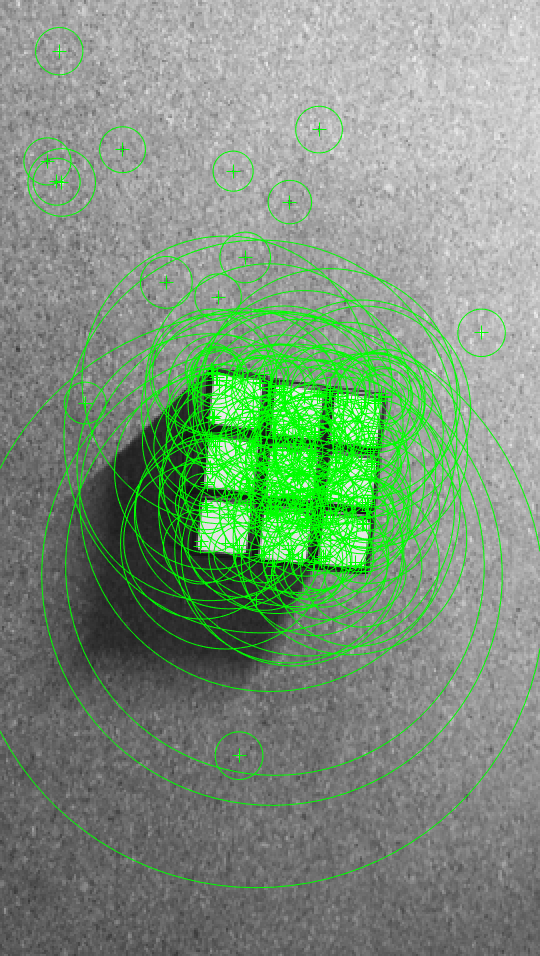


Рис. 8. Ключевые точки, найденные с помощью BRISK метода.

Ключевые точки, найденные с помощью FAST метода, изображены на рисунке 9.

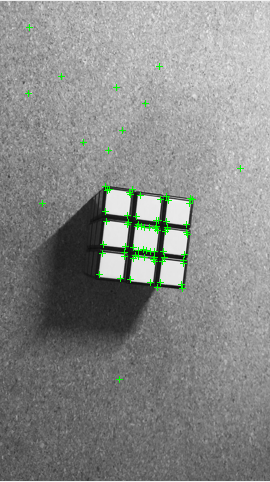


Рис. 9. Ключевые точки, найденные с помощью FAST метода.

Ключевые точки, найденные с помощью метода Харриса, изображены на рисунке 10.

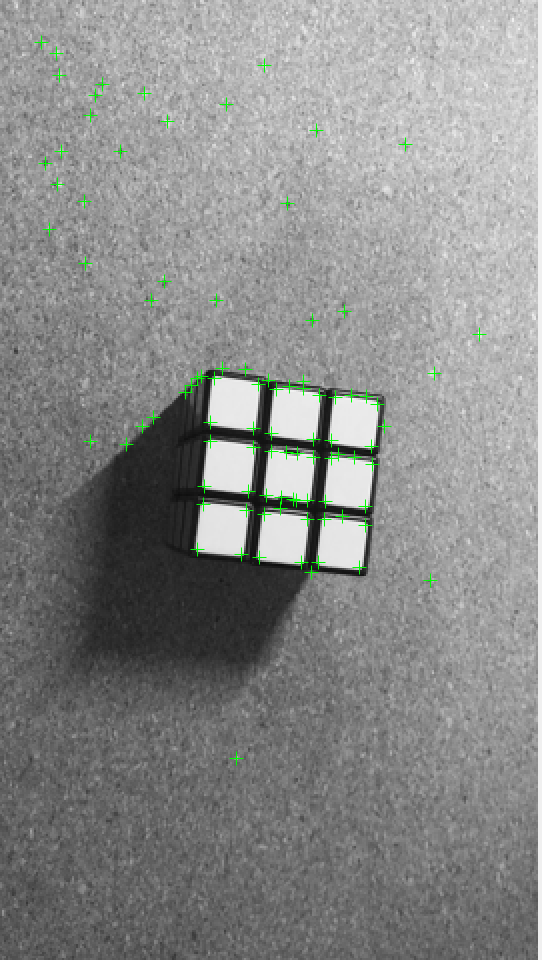


Рис. 10. Ключевые точки, найденные с помощью метода Харриса.

Ключевые точки, найденные с помощью метода Ши-Томаси (Minumum Eigenvalue), изображены на рисунке 11.

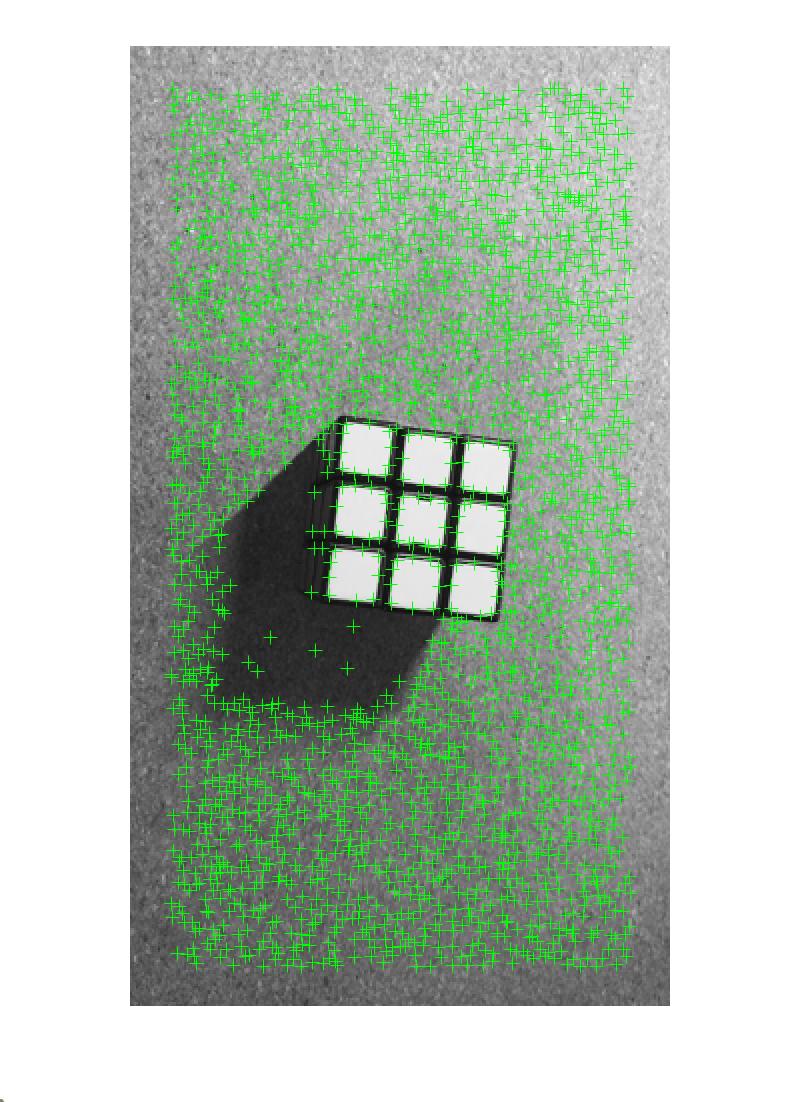


Рис. 11. Ключевые точки, найденные с помощью метода Ши-Томаси.

Ключевые точки, найденные с помощью метода ORB, изображены на рисунке 12.



Рис. 12. Ключевые точки, найденные с помощью метода ORB.

Ключевые точки, найденные с помощью метода SURF, изображены на рисунке 13.

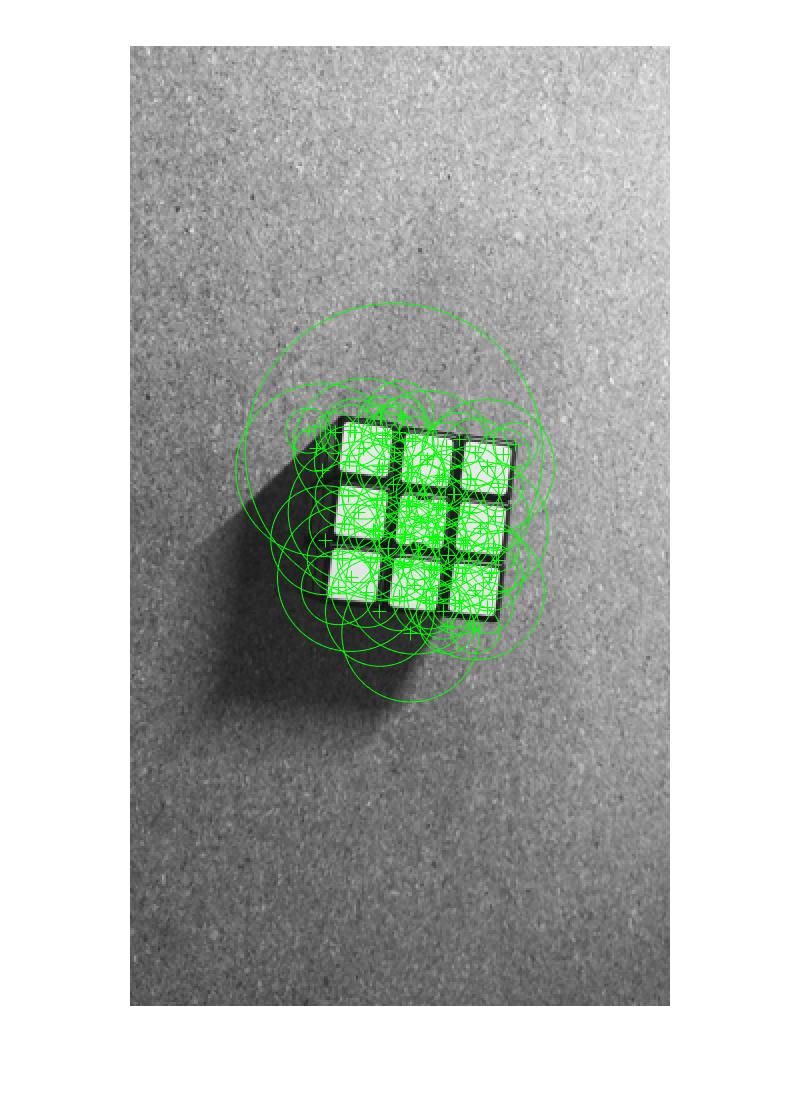


Рис. 13. Ключевые точки, найденные с помощью метода SURF.

Исследование будем проводить на первых 1000 кадрах видеофрагмента с заранее заданным положением выделения объекта для исключения погрешности. Результаты исследования представлены в таблице 1. Здесь точность выступает в значении процентного отношения количества кадров, на которых объект распознался неверно, ко всем распознанным кадрам.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Время, с | Количество кадров, на которых распознался объект | Точность, % |
| SURF | 56,5 | 1000 | 80 |
| ORB | 40,9 | 425 | 50 |
| Minimum Eigenvalue | 84,8 | 194 | 70 |
| Харриса | 63,9 | 100 | 90 |
| FAST | 44,5 | 240 | 60 |
| BRISK | 167,7 | 103 | 70 |

Таблица 1. Результаты исследования.

Из таблицы видно, что самым быстрым оказался метод ORB, но он также является наименее точным из всех. Наиболее оптимальным вариантом среди всех методов можно выделить SURF, который распознал объект на всех кадрах за приемлемое время и точность. Алгоритм Харриса нашёл объект лишь на 100 кадрах, но сделал это довольно точно. Примеры распознанного объекта на рисунках 14, 15, 16, 17.

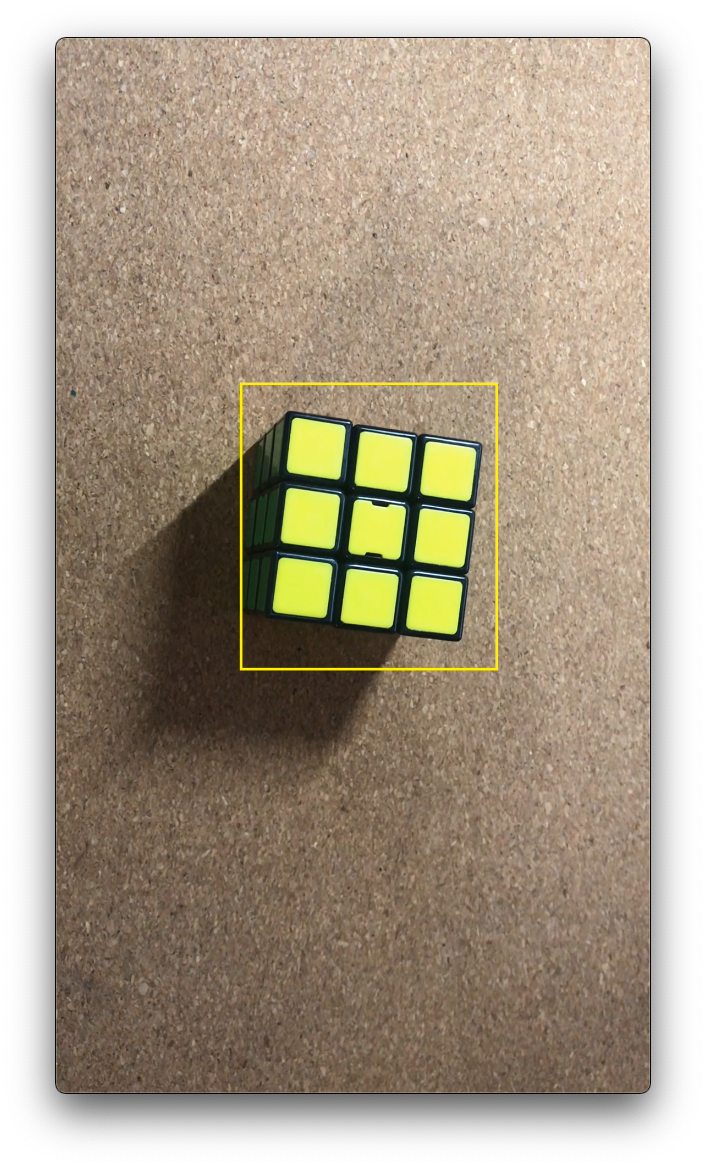


Рис. 14. Фото 1 с распознанным объектом.



Рис. 15. Фото 2 с распознанным объектом.

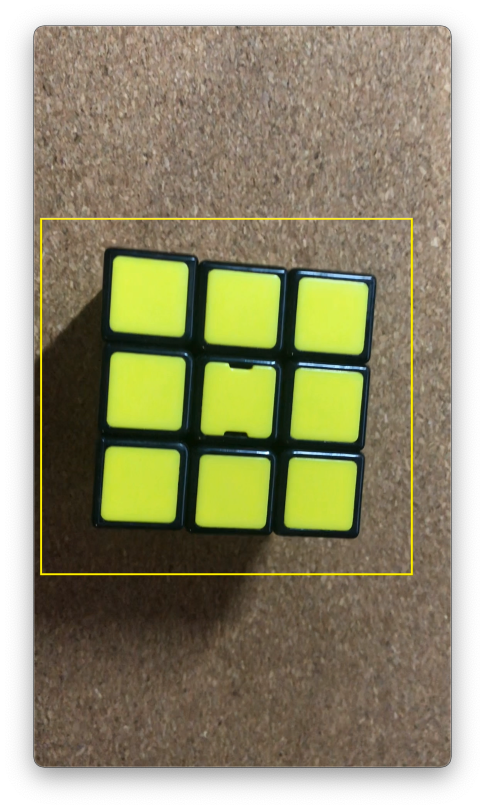


Рис. 16. Фото 3 с распознанным объектом.

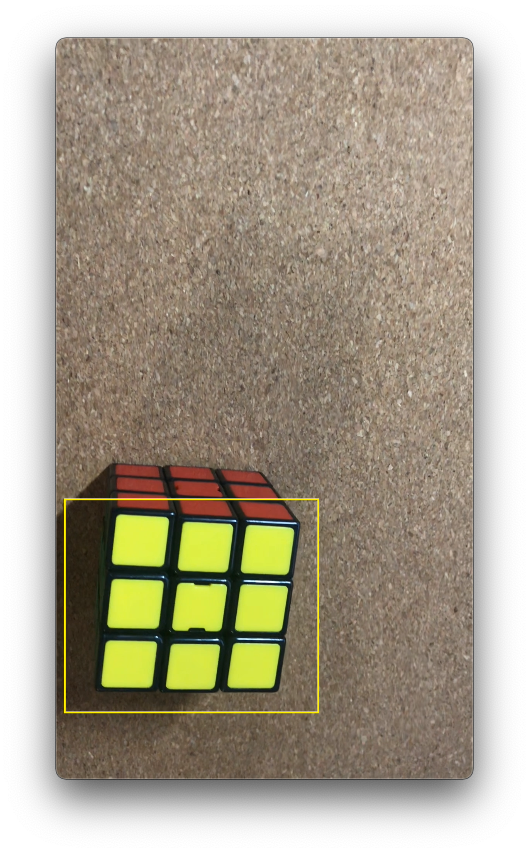


Рис. 17. Фото 4 с распознанным объектом.

# ЗАКЛЮЧЕНИЕ

В данной работе был произведён обзор теоретического материала по обнаружению и сопоставлению признаков изображения. Подробно рассмотрены наиболее современные и популярные методы поиска особых точек и построения их дескрипторов. Были детально показаны этапы алгоритма каждого из методов, указаны отличия их от других с указанием достоинств и недостатков и их области применимости. Для достижения поставленной цели была разработана программа, с помощью которой можно провести анализ методов и на практике убедиться в легитимности теоретических сведений.

Первым этапом в отслеживании объекта на видеофрагменте является выделение объекта должным образом на кадре, нахождение особых точек образа объекта и построение их дескрипторов. Эта задача является достаточно важной, поскольку от правильного выбора объекта зависит точность отслеживания, а количество и качество особых точек позволяет выделять и следить за объектом с разных ракурсов. Для нахождения особых точек на объекте и последующих кадров используются методы SURF, Харриса, ORB, BRISK, Ши-Томаси и FAST.

В дальнейшем есть возможность продолжения работы над следующим этапом – отслеживанием объекта в реальном времени или даже нескольких объектов.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski: “ORB: an efficient alternative to SIFT or SURF”, Computer Vision (ICCV), IEEE International Conference on. IEEE, pp. 2564 – 2571, 2011.
2. Stefan Leutenegger, Margarita Chli, Roland Siegwart: “BRISK: Binary Robust Invariant Scalable Keypoints". Computer Vision (ICCV), pp. 2548 – 2555, 2011.
3. Michael Calonder, Vincent Lepedit, Christoph Strecha, Pascal Fua, “BRIEF: Binary Robust Independent Elementary Features”, 11th European Conference on Computer Vision (ECCV), pp. 430 – 443, 2006.
4. Herbert Bay, Tinne Tuyletaars, Luc Van Gool, “SURF: Speeded Up Robust Features”, Proceedings of the ninth European Conference on Computer Vision, pp. 404 – 417, 2006.
5. Harris C., Stephens M.: “A Combined Corner and Edge Detector”. Proceedings of the 4th Alvey Vision Conference, pp. 147 – 151, 1988.
6. R. Szelinski. Computer Vision: Algorithms and Applications. Springer, 2010.
7. Viswanathan, Deepak. “Features from Accelerated Segment Test (FAST)”, 2011.

# ПРИЛОЖЕНИЯ

**Приложение А.**

**Код исследования методов поиска особых точек.**

beep off;

clc, clear all, close all;

warning off all;

tic;

global sc

global method

sc = 0.25;

methods = ["BRISK", "FAST", "Harris", "MinEigen", "ORB", "SURF"];

methodNum = 6; % Choose method to solve

method = methods(methodNum);

source.vid = VideoReader('video.mov');

N = source.vid.NumFrames;

out = struct('cdata',zeros(source.vid.Height, source.vid.Width, 3, 'uint8'),...

'colormap', []);

researchMethods = true;

startImage = read(source.vid, 1);

if (researchMethods)

source.bb = [332, 689, 468, 513];

else

source.bb = bb\_click(startImage);

end

boxImage = imresize(imcrop(startImage, source.bb), sc);

boxImageGray = rgb2gray(boxImage);

[input.boxFeatures, input.boxPoints] = extractFeatures(boxImageGray, ...

eval("detect" + method + "Features(boxImageGray);"));

v = VideoWriter('outputVideo', 'MPEG-4');

v.FrameRate = round(source.vid.FrameRate);

open(v);

showFeatures = false;

recognized = 0;

for i = 1 : 1000

i

frameImage = read(source.vid, i);

output = processFrame(rgb2gray(frameImage), boxImageGray, input);

if (showFeatures)

showFeatures = false;

imshow(output.sceneImage);

hold on

output.boxPoints.Location = output.boxPoints.Location + ...

[source.bb(1) source.bb(2)];

plot(output.boxPoints);

hold on

plot(output.scenePoints);

hold off

end

if ~isempty(output.boundingBox)

recognized = recognized + 1;

b = output.boundingBox / sc;

frameImage = insertShape(frameImage, 'rectangle',...

[b(1), b(2), b(3), b(4)], 'LineWidth', 5);

end

writeVideo(v, frameImage);

hold on

end

close(v)

recognized

toc

disp("The video is located in " + v.Path + "/" + v.Filename);

%% Functions

function bb = bb\_click(img)

imshow(img);

text(10, 10, 'Define bounding box.','color','white');

h = drawrectangle;

p = h.Position;

bb = [p(1); p(2); p(3); p(4)];

clf;

close all

end

function output = processFrame(sourceImage, boxImage, output)

global sc

global method

output.boundingBox = [];

output.sceneImage = imresize(sourceImage, sc);

[output.sceneFeatures, output.scenePoints] = extractFeatures(output.sceneImage, ...

eval("detect" + method + "Features(output.sceneImage);"));

boxPairs = matchFeatures(output.boxFeatures, output.sceneFeatures);

matchedBoxPoints = output.boxPoints(boxPairs(:, 1), :);

matchedScenePoints = output.scenePoints(boxPairs(:, 2), :);

if size(matchedBoxPoints, 1) > 5

[tform, inlierIndex, status] = ...

estimateGeometricTransform2D(matchedBoxPoints, matchedScenePoints, 'affine');

boxPolygon = [1, 1;...

size(boxImage, 2), 1;...

size(boxImage, 2), size(boxImage, 1);...

1, size(boxImage, 1);...

1, 1];

newBoxPolygon = transformPointsForward(tform, boxPolygon);

x1 = newBoxPolygon(1, 1);

y1 = newBoxPolygon(1, 2);

width1 = abs(newBoxPolygon(3, 1) - newBoxPolygon(1, 1));

height1 = abs(newBoxPolygon(3, 2) - newBoxPolygon(1, 2));

output.boundingBox = [x1, y1, width1, height1];

end

end