

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ СИСТЕМ

Бубенчиков Максим Александрович

Выпускная квалификационная работа бакалавра

Сравнительный анализ методов нахождения
особых точек на изображении

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент

Ковшов А.М.

Санкт-Петербург
2016

Содержание

Введение.....	3
Обзор литературы.....	5
Постановка задачи.....	6
Глава 1. Детекторы особых точек.....	7
1.1. Детектор Моравца.....	7
1.2. Детектор Харриса.....	8
1.3. Детектор Shi-Tomasi.....	10
1.4. Детектор FAST.....	10
1.5. Детектор SIFT.....	13
1.6. Детектор SURF.....	16
1.7. Детектор BRISK.....	16
Глава 2. Практическое сравнение работы детекторов.....	18
2.1. Детектор Харриса.....	18
2.2. Детектор Shi-Tomasi.....	23
2.3. Детектор FAST.....	27
2.4. Детектор SIFT.....	32
2.5. Детектор SURF.....	37
2.6. Детектор BRISK.....	41
2.7. Сравнение результатов работы методов.....	45
Выводы.....	48
Заключение.....	50
Список литературы.....	50

Введение

Сопоставление изображений является одним из фундаментальных аспектов многих задач компьютерного зрения, таких как распознавание объектов, построение трёхмерной сцены на основе нескольких изображений, создание стереопары, создание панорамных изображений, motion tracking (слежение за объектом в видеопоследовательности) и так далее. По причине того, что область знаний, решающая подобные проблемы, ещё молода (интенсивное изучение началось лишь в 1970-х годах), не существует определённого универсального метода, идеально подходящего для всех задач такого рода. Однако, существуют методы решения более узких задач, каждый из которых будет показывать лучшие результаты в зависимости от задачи.

Человеку несложно сравнить два изображения и выделить на них объекты, всё это происходит на интуитивном уровне и не вызывает затруднений. Но для компьютера изображение — лишь набор данных, не несущий информации об объектах, изображенных на нём. Можно описать два основных подхода к наделению машины таким умением.

Первый подход заключается в сравнении изображений на основе информации полученной от анализа всех пикселей изображения [3]. В общем случае алгоритм такого типа выглядит следующим образом: для каждого пикселя вычисляется значение некоторой функции, и на основе этих значений получается определенная характеристика всего изображения в целом. И тогда задача сводится к сравнению этих характеристик. И хотя такие алгоритмы просты и не требуют больших вычислительных затрат, результат их работы оставляет желать лучшего. Причина заключается в их основной идее, в характеристику изображения вносит вклад каждый пиксель, несмотря на ценность этого вклада. Метод будет выдавать неприемлемый результат в случае появления новых объектов, перекрытия одних объектов другими, появления шума, изменение положения объекта в трёхмерном пространстве и так далее.

Второй подход, о котором и пойдёт речь в данной работе, основан на идее использования лишь тех пикселей, вклад которых в общую характеристику будет значительным. Изображение заменяется на набор точек, называемых особыми.

Определение. Особой точкой изображения называется точка изображения, окрестность которой $o(m)$ отличима от окрестности любой другой точки изображения $o(n)$ в некоторой другой окрестности особой точки $o_2(m)$. [1]

В большинстве алгоритмов в качестве окрестности точки используется прямоугольное окно размером 5×5 пикселей. Харалик и Шапир [2] определили следующие требования, которым должны удовлетворять особые точки:

1. Отличимость: Особая точка должна выделяться на фоне и быть уникальной в своей окрестности.
2. Инвариантность: Выбор особых точек должен быть независимым от аффинных преобразований.
3. Стабильность: Выбор особых точек должен быть устойчив к шуму и ошибкам.
4. Уникальность: Помимо локальной отличимости, особая точка должна также обладать свойством глобальной уникальности для того, чтобы улучшить различимость повторяющихся паттернов.
5. Интерпретируемость: Особые точки должны определяться таким образом, чтобы их можно было использовать для анализа соответствий и лучшей интерпретации изображения.

Процесс сравнения изображений разделяется на три этапа. Первый этап — нахождение множества особых точек с помощью методов, называемых детекторами. Данные методы обеспечивают инвариантность нахождения одних и тех же точек относительно преобразований изображения. Однако, недостаточно использования лишь детектора, так как результатом его работы

является множество координат особых точек, которые на каждом изображении различны. Для этого на втором этапе происходит построение дескрипторов. Дескриптор — это описание точки, уникально идентифицирующее её среди множества всех точек. Дескриптор должен обеспечивать инвариантность нахождения соответствий между точками относительно преобразований изображения. Некоторые методы выполняют сразу обе задачи, детектирование особых точек и построение дескрипторов. И третий этап заключается в сравнении дескрипторов и поиске точек, совпадающих на обоих изображениях.

Обзор литературы

В 1981 году с выходом работы Моравеца [4] начались исследования сопоставления изображений с помощью нахождения особых точек. Моравец предложил простой алгоритм поиска углов на изображении, который впоследствии был улучшен Харрисом и Стефенсом в 1988 году в работе «A combined corner and edge detector» [5], и был назван детектором Харриса (Harris and Stephens detector или Plessey operator). Его преимуществом была инвариантность детектора к поворотам. в 1994 году Ши и Томаси внесли небольшую модификацию в детектор Харриса, получив более точные результаты при детектировании в работе «Good Features to Track» [6]. В 2005 году Ростен и Драммонд предложили метод детектирования особых точек, который они назвали Features from Accelerated Segment Test (особенности ускоренных испытаний сегмента) в [7]. Главной целью его создания было значительное ускорение работы детектора. Однако FAST обладал рядом недостатков. С целью улучшить алгоритм, Э. Ростен, Р. Портер и Т. Драммонд предложили использовать машинное обучение для определения особых точек в работе [8]. Этот алгоритм был назван FAST-ER (ER — Enhanced Repeatability, улучшенная повторяемость). И хотя FAST-ER оказался медленнее, качество детектирования повысилось. Все предыдущие методы не

были инвариантны к изменению масштаба. Для решения этой проблемы, в 2004 году, Д. Лоув предложил новый алгоритм детектирования и вычисления дескрипторов особых точек, названный SIFT (Scale Invariant Feature Transform) в работе «Distinctive Image Features from Scale-Invariant Keypoints» [9], основывающийся на рассмотрении разностей Гауссианов. Метод SURF был представлен Г. Бэем в 2006 году в работе [10], он был создан с целью сохранить качества SIFT, повысив при этом скорость работы. Метод BRISK был предложен С. Леутнеггером, М. Чли и Р. Сигвартом в 2011 году в [11]. Целью его создание было достижение высокой скорости работы, сопоставимой с FAST и инвариантности к масштабу.

Постановка задачи

Задача данной работы состоит в сравнении методов детектирования особых точек. Процесс её выполнения можно разделить на следующие этапы:

- Изучение принципов функционирования методов.
- Разработка инструментария, позволяющего обнаруживать особые точки каждым из рассмотренных детекторов.
- Анализ результатов работы методов на наборе тестовых изображений.
- Получение выводов о возможных сценариях использования каждого из них, основанных на выявленных во время анализа достоинствах и недостатках.

Глава 1. Детекторы особых точек

В данной главе будут рассмотрены наиболее популярные детекторы особых точек, принцип их работы и основные преимущества и недостатки каждого из них.

1.1. Детектор Моравеца

Одним из самых распространённых детекторов особых точек являются детекторы углов на изображении. Удобство их использования заключается в том, что углы на изображении можно однозначно сопоставить, в отличие от рёбер.

Метод, предложенный Моравецом [4] является самым простым среди детекторов углов.

Автор предлагает измерять изменение яркости пикселя (x, y) посредством смещения квадратного окна с центром в (x, y) на один пиксель в каждом из восьми направлений (вверх, вниз, вправо, влево, и в четырёх направлениях по диагонали). Размер окна чаще всего выбирается равным 3×3 , 5×5 или 9×9 пикселей.

Детектор работает следующим образом:

1. Для каждого из направлений $(u, v) \in \{(1, 0), (1, 1), (0, 1), (-1, 0), (-1, -1), (0, -1), (1, -1)\}$ вычисляется изменение яркости:

$$V_{u,v}(x, y) = \sum_{\forall a, b} (I(x+u+a, y+v+b) - I(x+a, y+b))^2$$

, где $I(x, y)$ – яркость пикселя (x, y) в изначальном изображении.

2. На втором шаге строится карта вероятности нахождения углов путём вычисления величины $C(x, y)$ для каждого пикселя (x, y) :
$$C(x, y) = \min(V_{u,v}(x, y))$$

3. Определяется порог значений T карты вероятности. Для всех $C(x, y) < T$ значение $C(x, y)$ обнуляется.

4. С помощью процедуры NMS (non-maximal suppression) находятся локальные максимумы.

Результатом выполнения данного алгоритма является карта вероятностей. Все точки, значения в карте для которых не равно нулю считаются углами, то есть особыми точками.

Метод Моравеца обладает следующими недостатками: большое количество ошибок из-за наличия шума на изображении, анизотропия всего в 8 направлениях, отсутствие свойства инвариантности к преобразованию типа «поворот».

1.2. Детектор Харриса

Метод Харриса основан на детекторе Моравеца и является его улучшением, так как для него характерна анизотропия по всем направлениям. Харрис и Стефенс предложили рассмотреть производные по множеству направлений.

Для данного изображения I рассмотрим окно W (обычно его размер выбирается равным 5×5 пикселей, но может быть другим в зависимости от размера изображения) с центром в точке (x, y) , а также сдвиг этого окна на (u, v) (изображено на рисунке 1.2.1).

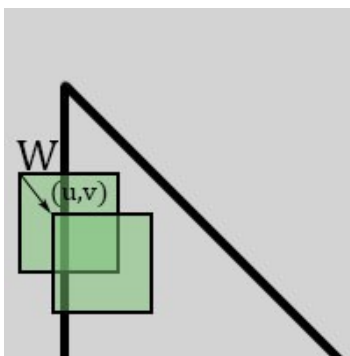


рис 1.2.1. Используемое окно

Тогда взвешенная сумма квадрата разностей между окном W и окном

W , сдвинутым на (u, v) (т.е. изменение окрестности точки (x, y) при сдвиге на (u, v)) равна:

$$E(u, v) = \sum_{(u, v) \in W} w(x, y) (I(x+u, y+v) - I(x, y))^2 \approx \sum_{(u, v) \in W} w(x, y) (I_x(x, y)u - I_y(x, y)v)^2$$

где $w(x, y)$ – весовая функция (обычно используется функция Гаусса или бинарное окно). M – автокорреляционная матрица:

$$M = \sum_{(u, v) \in W} w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Угол характеризуется большими изменениями функции $E(x, y)$ по множеству всех направлений (x, y) , что эквивалентно большим по модулю собственным значениям матрицы M . Расположение собственных значений приведено на рисунке 1.2.2.

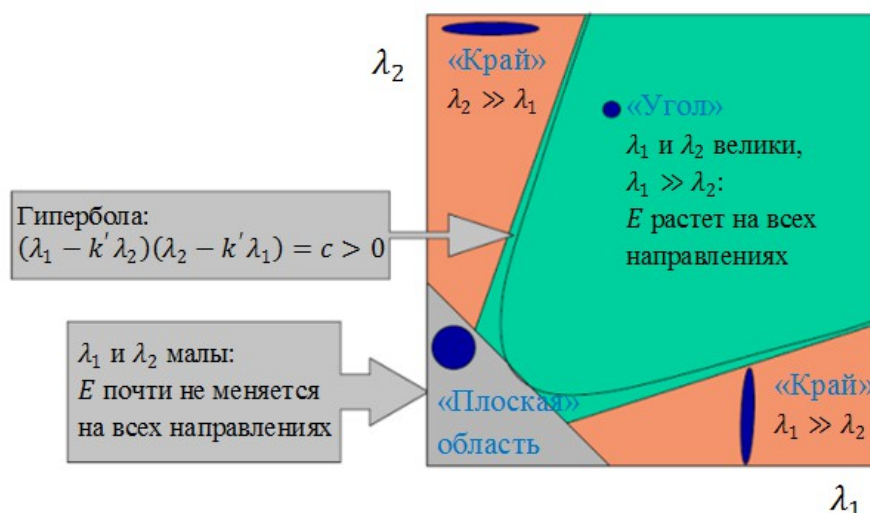


рис. 1.2.2. Расположение собственных значений

Так как считать собственные значения напрямую затратно с точки зрения ресурсов, Харрис и Стефен предложили меру отклика:

$$R = \det M - k(\text{tr} M)^2 > k \quad \text{где } k \text{ – эмпирическая константа, } k \in [0,04, 0,06] .$$

Таким образом, значение $R \geq 0$ для угловых особых точек. Затем производится фильтрация точек по R (т.е. те точки, у которых значение R меньше определённого порога, исключаются из рассмотрения). Далее

находятся локальные максимумы функции отклика по окрестности заданного радиуса и выбираются в качестве особых точек.

Детектор Харриса инвариантен к преобразованию типа «поворот», однако чувствителен к изменению масштаба и шуму.

1.3. Детектор Shi-Tomasi

Детектор Shi-Tomasi основан на детекторе Харриса. Различие заключается в том, что в детекторе Shi-Tomasi мера отклика вычисляется следующим способом:

$$R = \min(\lambda_1, \lambda_2)$$

Если это значение выше определённого порога, то точка считается углом и соответственно особой (рисунок 1.3.1).

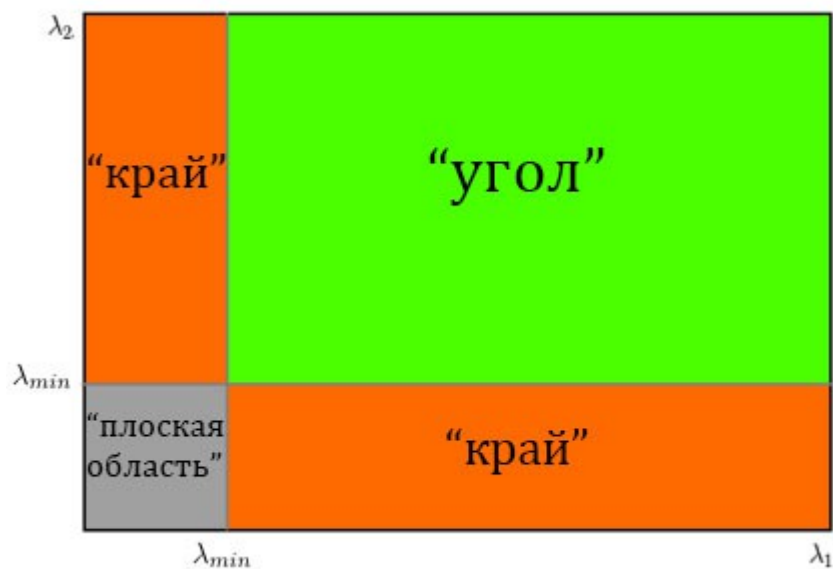


рис 1.3.1. Расположение собственных значений.

1.4. Детектор FAST

Принцип работы метода FAST [7] заключается в следующем:

1. Выбирается точка изображения p , для которой будет решаться, является ли она особой или нет. Пусть I_p - яркость точки.

2. Выбирается значение порога t .
3. Рассматривается окрестность из 16 пикселей вокруг выбранной точки (рисунок 1.4.1).

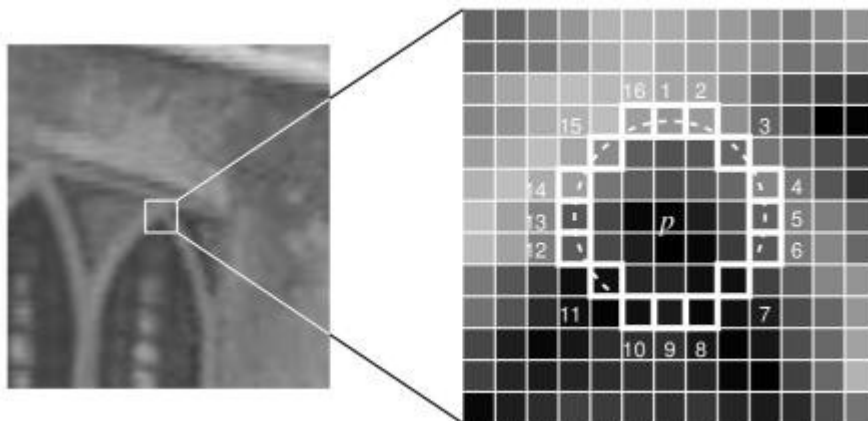


рис. 1.4.1. Рабочая окрестность пикселя при использовании FAST детектора.

4. Точка p считается углом, если среди 16-ти пикселей окрестности существует n пикселей, каждый из которых ярче, чем $I_p + t$, или n пикселей, каждый из которых темнее, чем $I_p - t$ (такие пиксели отмечены пунктирной линией на рисунке 1.4.1.). n обычно выбирается равным 12. Эксперименты показали, что наименьшее значение n , при котором точки начинают стабильно обнаруживаться, равно 9.

5. Высокая скорость работы данного алгоритма обусловлена тем, что сначала проверяется интенсивность лишь четырёх точек из окрестности, под номерами 1, 5, 9 и 13. Если хотя бы для трёх из них выполняется условие $I_{p_i} > I_p + t$ или $I_{p_i} < I_p - t, i=1, \dots, 4$, тогда производится проверка всех оставшихся 12-ти пикселей. В ином случае выбирается следующая точка и алгоритм повторяется для неё.

Данный алгоритм обладает рядом недостатков, например его эффективность зависит от порядка обработки изображения и распределения пикселей, вблизи некоторой окрестности может обнаружиться несколько особых точек. В [8] было предложено использование машинного обучения

для исправления этих недостатков, полученный алгоритм был назван FAST-ER:

1. Выбирается набор изображений для обучения.
2. Алгоритм FAST применяется для поиска особых точек на каждом из изображений.
3. Для каждой особой точки сохраняется 16 пикселей вокруг неё в виде вектора. После выполнения данного шага для каждого изображения, все полученные вектора объединяются в один вектор P .

4. Каждый пиксель из 16-ти может находиться в одном из трёх состояний:

$$S_{p \rightarrow x} = \begin{cases} d, I_{p \rightarrow x} \leq I_p - t & (\text{темнее}) \\ s, I_p - t < I_{p \rightarrow x} < I_p + t & (\text{похож}) \\ b, I_p + t \leq I_{p \rightarrow x} & (\text{светлее}) \end{cases}$$

5. В зависимости от состояний, приведённых выше, P разделяется на три подмножества, P_d, P_s, P_b .

6. В рассмотрение вводится булева переменная K_p , которая равна 1, если p — угол и 0 в противном случае.

7. Используется алгоритм ID3 (классификатор деревьев решений) [9]. Он выбирает x , который даёт больше всего информации о том, является ли пиксель углом или нет, вычисленной с помощью энтропии K_p .

8. Эта процедура повторяется рекурсивно, до тех пор пока энтропия K_p не станет равной нулю.

9. Созданное таким образом дерево решений используется для более быстрого детектирования в других изображениях.

Для того, чтобы избежать нахождения нескольких особых точек вблизи некоторой окрестности, используется подход, называемый non-maximal suppression, то есть поиск локальных максимумов. Вычисляется значение функции V для каждой из найденных особых точек. V — это сумма разностей интенсивностей p и каждого из 16-ти окружающих пикселей.

Рассматриваются две соседних особых точки и сравниваются их значения V . Точка с меньшим значением V исключается из рассмотрения.

1.5. Детектор SIFT

Выше были рассмотрены детекторы углов, инвариантные к преобразованию типа «поворот» (кроме детектора Моравеца). Это логично, так как углы остаются углами при повороте, однако, при изменении масштаба, угол может быть не распознан алгоритмом (рисунок 1.5.1).

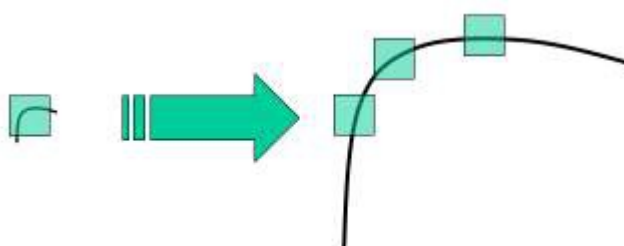


рис. 1.5.1. Угол при изменении масштаба.

В [9] предлагается алгоритм, названный SIFT для решения этой проблемы. Далее будет рассмотрен принцип работы той части алгоритма, в которой происходит нахождение особых точек.

Основной идеей детектирования точек в данном методе является построение пирамиды Гауссианов и пирамиды разностей Гауссианов.

Гауссианом называется изображение:

$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$, где L – значение гауссиана в точке (x, y) , σ – радиус размытия. G – гауссово ядро, I – значение интенсивности точки (x, y) в исходном изображении.

Разностью гауссианов называется изображение, полученное путём попиксельного вычитания двух гауссианов с разными значениями радиуса размытия. Процесс поиска разностей гауссианов повторяется для различных октав изображения в пирамиде гауссианов (рисунок 1.5.2).

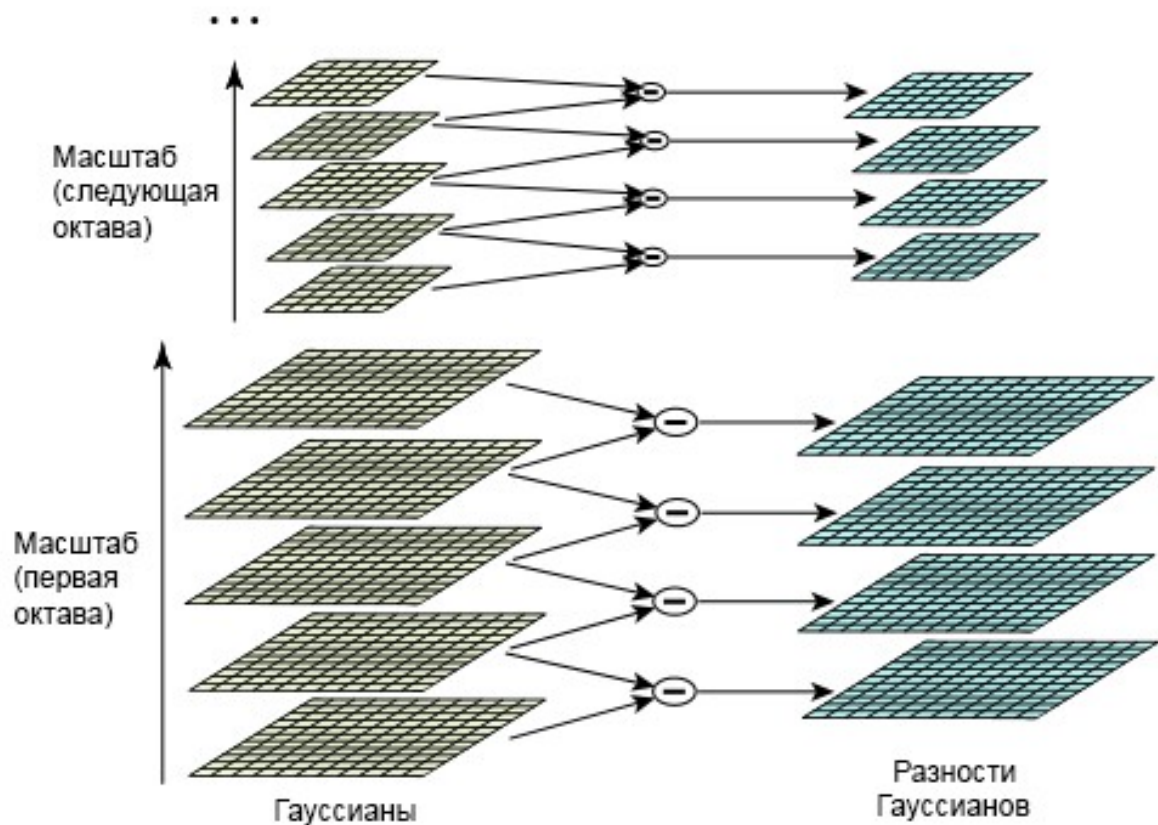


рис. 1.5.2. Пирамиды гауссианов и разностей гауссианов.

Как только разности гауссианов найдены, производится поиск локальных экстремумов. Например, точка изображения сравнивается с восемью соседними точками и с девятью точками из следующей разности гауссианов и с девятью точками из предыдущей разности гауссианов (рисунок 1.5.3). Если точка является локальным экстремумом, то она считается потенциальной особой точкой.

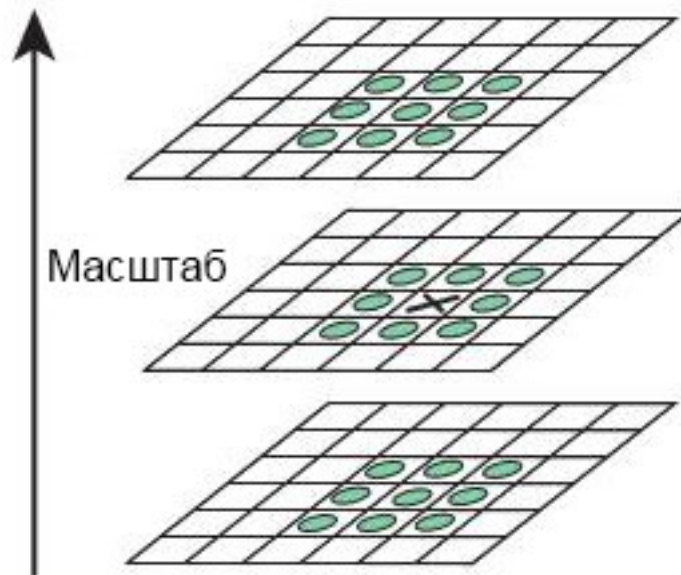


рис. 1.5.3. Поиск локальных экстремумов.

На основе экспериментов были получены оптимальные значения: количество октав = 4, количество различных значений масштаба в октаве = 5, начальное значение $\sigma = 1.6$.

Далее происходит локализация особых точек. На этом этапе проверяется, подходят ли точки экстремума на роль особых. Используется разложение масштабируемого пространства в ряд Тейлора, чтобы получить более точное положение экстремума, и если его интенсивность меньше чем пороговое значение (автором метода предложено значение 0.03), то точка не является особой. После этого происходит проверка, не лежит ли точка на границе какого-либо объекта. Такие точки имеют большой изгиб (одна из компонент второй производной) вдоль границы и малый в перпендикулярном направлении. Этот большой изгиб определяется матрицей Гессе, размером 2×2 .

Алгоритм SIFT обладает рядом преимуществ, таких как инвариантность к поворотам, смещению, а также частичная инвариантность к изменению масштаба, яркости и положения камеры. Из недостатков можно отметить невысокую скорость работы алгоритма.

1.6. Детектор SURF

Метод SURF так же как и SIFT принадлежит к числу методов, производящих детектирование особых точек и построение дескрипторов. Он основан на тех же принципах что и SIFT, однако имеются различия, благодаря которым SURF работает быстрее.

В данном методе используются фильтры квадратной формы для аппроксимации размытия Гаусса. Фильтрация изображения с помощью квадрата происходит гораздо быстрее, если использовать интегральное представление изображения, которое выглядит следующим образом:

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

SURF использует детектор каплей (blob detector) основанный на использовании матрицы Гессе. Определитель матрицы Гессе используется как мера локальных изменений вокруг точки, и, выбираются те точки, для которых этот определитель максимален. Также, SURF использует определитель матрицы Гессе для выбора масштаба. Матрица Гессе выглядит следующим образом:

$$H(p, \sigma) = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix}, \text{ где } p - \text{ точка изображения с}$$

координатами (x, y) , σ - масштаб фильтра, $L_{xx}(p, \sigma)$ — свертки аппроксимации второй производной Гауссова ядра с исходным изображением.

1.7. Детектор BRISK

Целью создания данного метода было достижение инвариантности к изменению масштаба, при этом сохранив высокую скорость работы.

Авторы метода создали свой метод на основе детектора AGAST, который, по сути, является расширением метода FAST. Однако, для того, чтобы добиться инвариантности к изменению масштаба, нахождение

максимумов происходит не только на исходном изображении, но и в масштабируемом пространстве. В методе BRISK масштабируемое пространство состоит из n октав c_i и n внутренних октав d_i , $i=\{0,1,\dots,n-1\}$. n обычно выбирается равным 4. Октавы составляются путём уменьшения масштаба исходного изображения в два раза. Каждая внутренняя октава d_i располагается между октавами c_i и c_{i+1} . Первая внутренняя октава получается путём уменьшения масштаба исходного изображения в 1.5 раза, а каждая последующая — путём уменьшения масштаба предыдущей внутренней октавы в два раза.

Метод FAST применяется к каждой из октав и каждой из внутренних октав по отдельности с одинаковым пороговым значением, с целью найти потенциально особые области. Затем среди точек, содержащихся в этих областях производится поиск локальных максимумов во всём масштабируемом пространстве.

Глава 2. Практическое сравнение работы детекторов

Автором данной работы было разработано программное обеспечение, реализующее поиск особых точек одним из методов (Детектор Харриса, Shi-Tomasi, FAST, SIFT, SURF и BRISK), подсчитывающее количество найденных точек и время, затраченное каждым из алгоритмов на поиск. Детектор Моравеца не был реализован в программе, по причине его низкой эффективности и редкости применения. В данной главе будет рассмотрено сравнение результатов работы каждого из детекторов на изображениях. Для сравнения выбраны изображения, обладающие некоторыми отличительными свойствами, например, наличие шума.

2.1. Детектор Харриса

В данном разделе будут рассмотрены результаты работы детектора Харриса на пяти различных изображениях: созданном с помощью средств компьютерной графики изображении простых фигур, фотографии здания и трёх её вариаций (повёрнутое, масштабированное и с добавленным шумом).

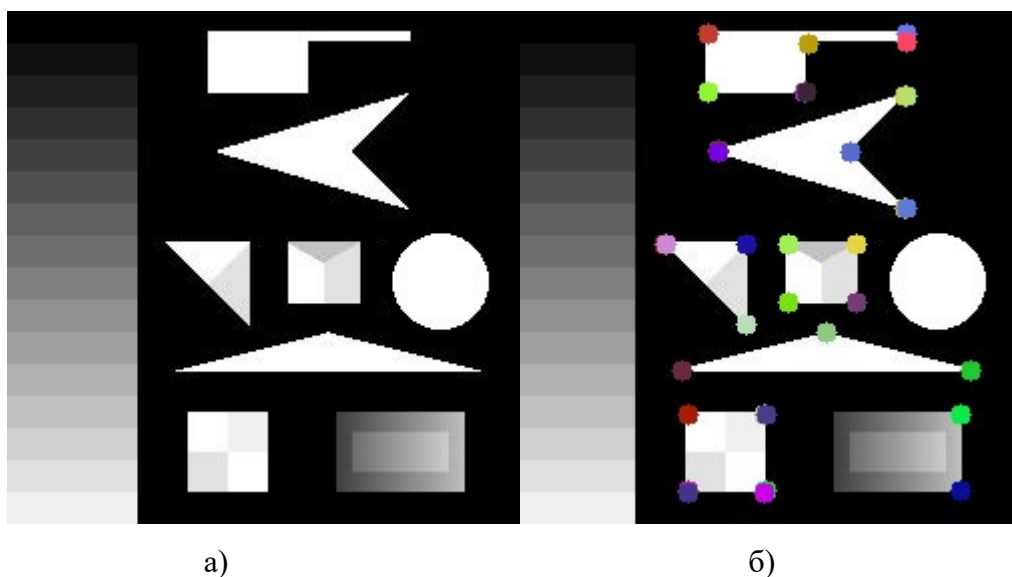


рис 2.1.1. а) Тестовое изображение. б) Особые точки, найденные методом Харриса.

На рисунке 2.1.1 показано тестовое изображение и изображение с отмеченными на нём особыми точками, найденными детектором Харриса. На изображении с разрешением 256x256 пикселей было найдено 35 особых точек за 0.053 секунд. Так как детектор Харриса является детектором углов, он не обнаружил особых точек на окружности и рёбрах объектов. Однако, можно заметить, что в окрестностях некоторых углов было найдено больше одной особой точки, что можно считать недостатком работы данного алгоритма. Также, детектор игнорировал углы, яркость которых не очень высока (прямоугольники разной яркости в левой части изображения и углы внутри некоторых фигур). Благодаря свойству анизотропии по всем возможным направлениям, детектор Харриса обнаружил углы разной величины (не только 45 и 90 градусов).



рис. 2.1.2. Особые точки, найденные методом Харриса на фотографии здания.

Следующим изображением для теста была выбрана фотография здания. Можно заметить, что с фотографией детектор Харриса справился хуже (рисунок 2.1.2). Многие углы, легко находимые человеком, не были найдены алгоритмом, в то время как в отдельных частях изображения было

обнаружено очень много особых точек близко друг к другу. Всего было найдено 127 точек за 0.405 секунд на изображении 800x600 пикселей.



рис. 2.1.3. Особые точки, найденные методом Харриса на повернутой фотографии.

Далее тест был проведён на том же самом изображении здания, но повернутом примерно на 10 градусов (рисунок 2.1.3). Несмотря на некоторые незначительные различия в результате, детектор Харриса обнаружил особые точки там же где и до этого, в примерно таком же количестве (110 точек за 0.42 секунд), что говорит о инвариантности данного метода к поворотам.



рис. 2.1.4. Особые точки, найденные методом Харриса на увеличенной фотографии.

При поиске особых точек на увеличенном изображении здания, была подтверждена теория о том, что детектор Харриса не инвариантен к изменению масштаба (рисунок 2.1.4). Алгоритм обнаружил множество точек, которые не были обнаружены в исходном изображении, а также не обнаружил некоторые точки, которые на исходном изображении были найдены, хотя таких точек оказалось немного. Всего было найдено 224 особых точки за 0.413 секунды.



рис. 2.1.5. Особые точки, найденные методом Харриса на фотографии с шумом.

На зашумлённом изображении детектор Харриса допустил небольшое количество ошибок (рисунок 2.1.5). Несколько особых точек было найдено там, где на самом деле нет углов и не были найдены некоторые из точек, которые были найдены в исходном изображении. Алгоритм показал не идеальные, хоть и достаточно хорошие результаты. Было найдено 96 особых точек за 0.352 секунд.



рис. 2.1.6. Особые точки, найденные методом Харриса на затемнённой фотографии.

На затемнённом изображении метод Харриса обнаружил 81 точку за 0.376 секунд (рисунок 2.1.6). По сравнению с оригинальным изображением, было найдено значительно меньше точек, но в то же время уменьшилось количество дубликатов и были обнаружены некоторые из углов, которые не были обнаружены до этого. Из различий в результате следует, что детектор Харриса лишь частично инвариантен к изменениям яркости изображения.

Из полученных результатов можно сделать вывод, что метод Харриса хорошо работает для изображений с ярко выраженными углами, хорошо выделяющимися на фоне, инвариантен к поворотам, обладает довольно высокой скоростью работы, однако плохо находит особые точки на изображениях разного масштаба, при наличии шума и различиях в яркости.

2.2. Детектор Shi-Tomasi

Для оценки результатов работы детектора Shi-Tomasi и всех остальных детекторов будут использоваться те же изображения, что и в предыдущем разделе. Это позволит наиболее точно сравнить результаты работы методов.

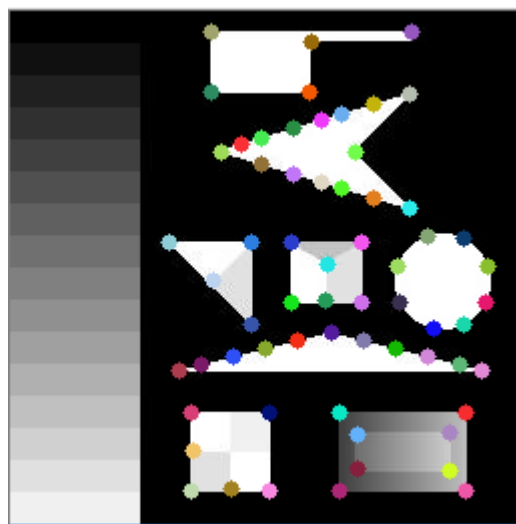


рис. 2.2.1. Особые точки, найденные методом Shi-Tomasi на тестовом изображении.

На первом изображении детектором Shi-Tomasi было найдено 63

особых точки за 0.151 секунд (рисунок 2.2.1). Алгоритм обнаружил некоторые из углов, которые не обнаружил детектор Харриса, однако допустил множество ошибок, найдя особые точки на диагональных рёбрах и окружности. Плюсом Shi-Tomasi можно считать отсутствие дубликатов точек в окрестностях углов. Так же как и детектор Харриса, Shi-Tomasi игнорировал углы недостаточной яркости (например, прямоугольники в левой части изображения), хоть таких и оказалось меньше.



рис. 2.2.2. Особые точки, найденные методом Shi-Tomasi на фотографии здания.

Результат поиска особых точек на фотографии здания очень похож на результат детектора Харриса (рисунок 2.2.2). В целом, алгоритмом Shi-Tomasi найдены и пропущены те же самые углы, но преимуществом является отсутствие дубликатов точек в окрестностях углов. Было найдено 98 точек за 2.1 секунды.



рис. 2.2.3. Особые точки, найденные методом Shi-Tomasi на повернутой фотографии.

Результат поиска особых точек на повернутой фотографии так же похож на результат метода Харриса (рисунок 2.2.3), за исключением отсутствия дубликатов. Найдены почти те же самые углы, что и на исходном изображении, что говорит о инвариантности метода Shi-Tomasi к поворотам. Было найдено 95 точек за 1.77 секунд.



рис. 2.2.4. Особые точки, найденные методом Shi-Tomasi на увеличенной фотографии.

На увеличенной фотографии метод Shi-Tomasi показал лучший результат, по сравнению с детектором Харриса (рисунок 2.2.4). Особых точек найдено меньше, но большинство из них действительно находятся на углах объектов на изображении. Как и в предыдущих случаях, отсутствуют дубликаты. Всего было обнаружено 82 точки за 1.77 секунд.

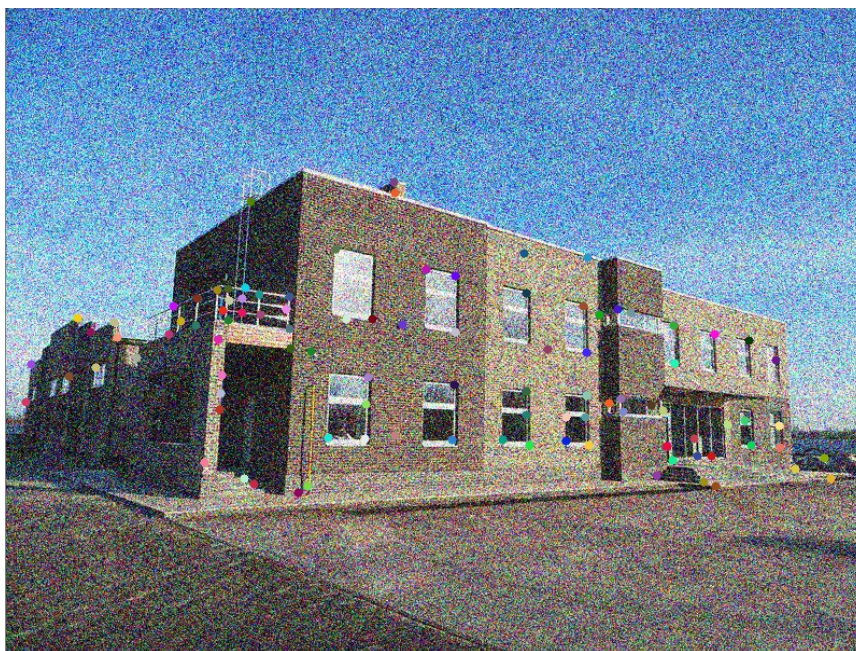


рис. 2.2.5. Особые точки, найденные методом Shi-Tomasi на фотографии с шумом.

Метод Shi-Tomasi обнаружил 93 особых точки за 1.97 секунд на изображении с наложенным шумом (рисунок 2.2.5). Так же как и метод Харриса, он допустил ошибки в детектировании, найдя точки там где на самом деле нет углов и не найдя множество из углов, которые легко различимы для человека. Результат поиска выглядит заметно хуже чем у детектора Харриса. Метод Shi-Tomasi является частично инвариантным к зашумлённости изображения.



рис. 2.2.6. Особые точки, найденные методом Shi-Tomasi на затемнённой фотографии.

На затемнённом изображении метод Shi-Tomasi обнаружил 83 особых точки за 2.53 секунд (рисунок 2.2.6). Результат поиска гораздо лучше, чем на исходном изображении, было найдено множество углов, которые не были найдены ранее. Это говорит о том, что метод частично инвариантен к изменению яркости изображения.

Детектор Shi-Tomasi создан на основе детектора Харриса и это сходство заметно при практическом сравнении их результатов. Он тоже является не инвариантным к изменению масштаба и появлению шума, частично инвариантен к изменению яркости и не зависит от поворотов. В целом он показывает лучшие результаты, хотя и совершает ошибки, несвойственные

методу Харриса, а именно, находит особые точки на диагональных рёбрах и окружностях.

2.3. Детектор FAST

В данном разделе будет рассмотрен результат работы алгоритма FAST на наборе тестовых изображений. Детектор FAST является детектором углов, так же как и детектор Харриса и Shi-Tomasi, поэтому найденные им особые точки будут сравнены с найденными точками предыдущих методов.



рис. 2.3.1. Особые точки, найденные методом FAST на тестовом изображении.

На тестовом изображении детектором FAST было обнаружено 46 особых точек (рисунок 2.3.1). Алгоритм нашел несколько точек в тех местах, где углов на самом деле нет, а так же игнорировал углы недостаточной яркости, как и оба предыдущих детектора. К недостаткам можно отнести наличие небольшого количества дубликатов особых точек в окрестностях некоторых углов. Необходимо отметить крайне высокую скорость работы. На изображении с разрешением 256x256 пикселей алгоритм закончил свою работу всего за 0.004 секунды.



рис. 2.3.2. Особые точки, найденные методом FAST на фотографии здания.

На фотографии здания было найдено 207 особых точек (рисунок 2.3.2). Метод показал хороший результат на данном изображении, обнаружив множество углов, малое количество лишних точек, не принадлежащих углам и малое количество дубликатов. Скорость работы снова оказалась крайне высокой. Точки были найдены всего за 0.008 секунд.



рис. 2.3.3. Особые точки, найденные методом FAST на повернутой фотографии.

На повернутом изображении алгоритм FAST обнаружил 202 особые точки за 0.008 секунд (рисунок 2.3.3). Почти все точки, найденные на исходном изображении, были найдены и на повернутом. Их количество также примерно одинаково. Это говорит о инвариантности алгоритма к преобразованию типа «поворот».



рис. 2.3.4. Особые точки, найденные методом FAST на увеличенной фотографии.

На увеличенной фотографии метод FAST обнаружил 136 точек за 0.008 секунд (рисунок 2.3.4). Не были найдены некоторые из точек, которые были найдены до этого, а также появились новые особые точки там где их не было раньше. Такой результат говорит о том, что метод FAST не инвариантен к изменению масштаба.

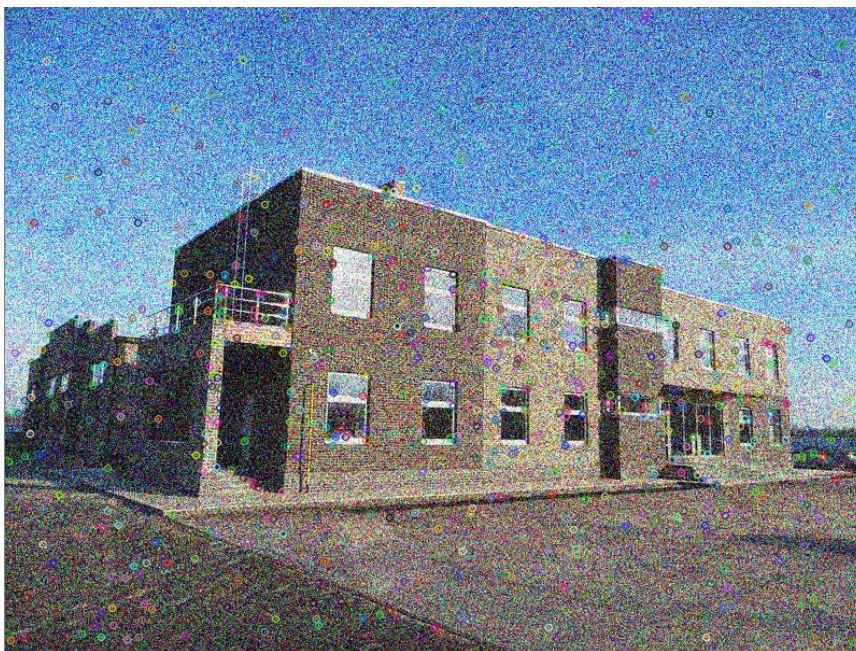


рис. 2.3.5. Особые точки, найденные методом FAST на фотографии с шумом.

На фотографии с шумом было найдено гораздо больше особых точек, чем на исходном изображении (710 штук) (рисунок 2.3.5). Можно увидеть, что большинство из найденных точек находятся в местах, в которых нет углов, а настоящие углы почти не были найдены. В целом, точки расположены хаотично и совершенно не подходят для последующего анализа. Это говорит о том, что метод FAST совершает очень много ошибок на зашумлённом изображении и не инвариантен к появлению шума.



рис. 2.3.6. Особые точки, найденные методом FAST на затемнённой фотографии.

Детектор FAST показал хорошие результаты поиска особых точек на затемнённой фотографии (рисунок 2.3.6). Были обнаружены почти те же самые углы, что и на исходном изображении, за небольшими исключениями. Из этого можно сделать вывод, что детектор FAST инвариантен к изменению яркости. Всего было найдено 159 точек за 0.008 секунд.

2.4. Детектор SIFT

В данном разделе будут рассмотрены результаты работы алгоритма SIFT на тестовых изображениях. Так как SIFT не относится к детекторам углов, тест не будет проводиться на изображении простых фигур, так как это не даст никакой точной информации о качестве работы детектора. Сравнение будет проводиться лишь на фотографии здания и её модифицированных версиях (повёрнутой, увеличенной, зашумлённой и затемнённой).

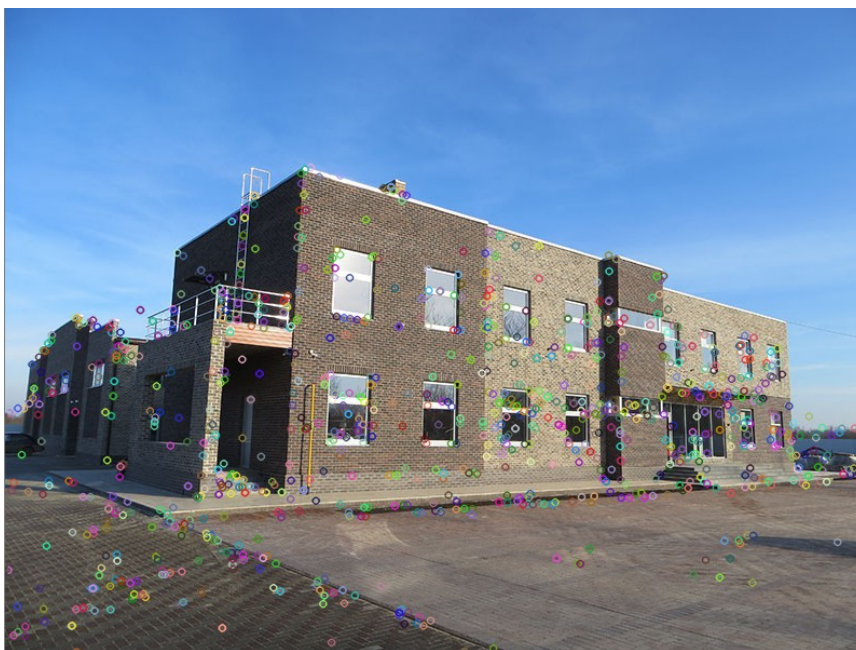


рис. 2.4.1. Особые точки, найденные методом SIFT на фотографии здания.

На фотографии здания была обнаружена 1081 особая точка за 4.23 секунды (рис. 2.4.1). Найденные точки в основном расположены в окрестности здания, что говорит о том, что метод хорошо выявил особенности изображения и объектов на нём. Ни одной точки не было найдено в той части фотографии, где изображено небо, то есть метод игнорирует монотонные части изображения. Однако были допущены ошибки, такие как наличие особых точек на текстуре самого здания и поверхности пола. Такие точки впоследствии не подойдут для сравнения изображений, так как текстура повторяется и точки нельзя будет однозначно соотнести, то есть было нарушено свойство уникальности, описанное в начале данной работы.

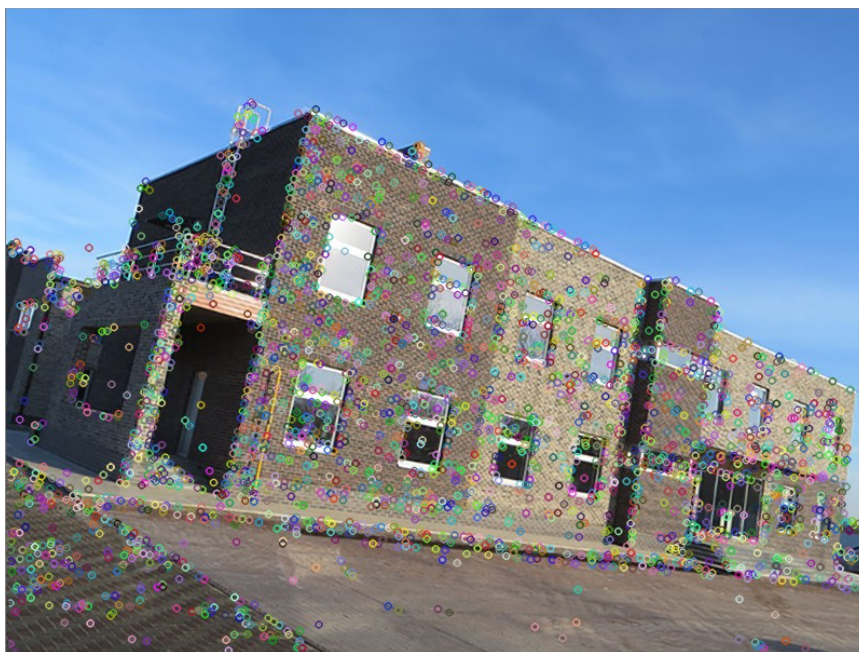


рис. 2.4.2. Особые точки, найденные методом SIFT на повёрнутой фотографии.

На повёрнутой фотографии детектор SIFT допустил те же ошибки, что и на исходном, обнаружив особые точки на повторяющихся текстурах здания и пола, но уже в большем количестве (рисунок 2.4.2). И хотя такие точки можно исключить из рассмотрения на стадии построения дескрипторов, их большое количество сказывается на скорости работы метода. Всего была найдена 3441 особая точка, что более чем в три раза превышает количество точек на исходном изображении.



рис. 2.4.3. Особые точки, найденные методом SIFT на увеличенной фотографии.

Количество точек найденных на текстуре здания при увеличении изображения стало ещё больше (всего 5077 особых точек), что заметно увеличило время работы алгоритма (5.98 секунд) (рисунок 2.4.3). Несмотря на теоретическую инвариантность метода к изменениям масштаба, практический результат показывает обратное. Стоит заметить, что в монотонных частях изображения особые точки по-прежнему не находятся, что говорит о том, что при использовании данного метода выполняется свойство отличимости, описанное в начале работы.



рис. 2.4.4. Особые точки, найденные методом SIFT на фотографии с шумом.

На фотографии с шумом метод SIFT показал крайне плохой результат (рисунок 2.4.4). Особые точки были найдены там где их не было до этого, в том числе на монотонных частях изображения, что делает их непригодными для последующей работы. При появлении шума нарушается как свойство локальной отличимости, так и глобальной уникальности.



рис. 2.4.5. Особые точки, найденные методом SIFT на затемнённой фотографии.

Детектор SIFT хорошо справляется с поиском особых точек на затемнённом изображении (рисунок 2.4.5). Большинство точек, найденных на исходном изображении было найдено и на затемнённом. Некоторые точки не были найдены, хотя таких немного. Метод SIFT частично инвариантен к изменениям яркости. Было обнаружено 508 точек за 3.67 секунд.

2.5. Детектор SURF

В данном разделе будут рассмотрены результаты поиска особых точек с помощью метода SURF. Так же как и SIFT, он не относится к детекторам углов, поэтому тест не будет проводиться на изображении простых фигур.



рис. 2.5.1. Особые точки, найденные методом SURF на фотографии здания.

На фотографии здания метод SURF показал результат, похожий на результат метода SIFT, точки найдены в той части где располагается здание, не обнаружены на монотонных частях изображения, что говорит о выполнении свойства глобальной уникальности (рисунок 2.5.1). Точек, найденных на повторяющихся текстурах мало, из чего можно сделать вывод, что свойство локальной отличимости тоже выполняется на достаточно

хорошем уровне. Стоит отметить, что метод SURF работает заметно быстрее чем SIFT (1337 особых точек было обнаружено за 1.08 секунд).



рис. 2.5.2. Особые точки, найденные методом SURF на повернутой фотографии.

Результат поиска на повернутом изображении почти не отличается от результатов поиска на исходном (рисунок 2.5.2). Алгоритм по-прежнему нашел точки, обладающие свойством глобальной и локальной отличимости, примерно в тех же местах, что и до этого. Это говорит о инвариантности метода к преобразованию типа «поворот». Всего было найдено 1604 особых точки за 1.15 секунд.

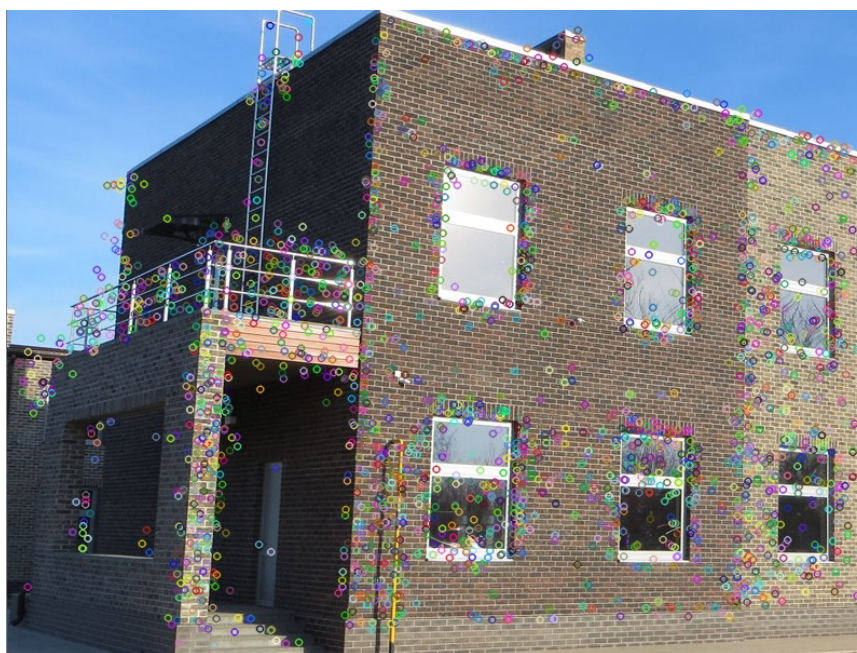


рис. 2.5.3. Особые точки, найденные методом SURF на увеличенной фотографии.

На увеличенной фотографии здания детектор SURF показал лучшие результаты, чем детектор SIFT (рис. 2.5.3). Хотя и было обнаружено некоторое количество точек, не обладающих свойством глобальной уникальности (на повторяющихся текстурах), их заметно меньше, чем у предыдущего алгоритма. Большинство найденных точек соответствуют точкам, найденным на исходном изображении. Можно сделать вывод, что теоретическая инвариантность метода SURF подтверждается на практике. Алгоритмом было найдено 2392 особых точки за 1.35 секунды.



рис. 2.5.4. Особые точки, найденные методом SURF на фотографии с шумом.

На фотографии с шумом алгоритм SURF показал плохой результат (рисунок 2.5.4). Точки, найденные алгоритмом хаотично разбросаны по всему изображению без каких-либо закономерностей. Точки были найдены как на монотонных частях изображения (небо, поверхность пола), так и на повторяющихся текстурах. Можно сделать вывод, что результат работы метода SURF сильно зависит от появления шума. Всего была обнаружена 3621 особая точка за 1.63 секунд.



рис. 2.5.5. Особые точки, найденные методом SURF на затемнённой фотографии.

Детектор SURF как и SIFT оказался почти независимым от изменений яркости (рисунок 2.5.5). На затемнённом изображении особые точки были найдены в тех же местах, что и на исходном. Точек не обнаружено на монотонных и повторяющихся частях изображения. Результат говорит о инвариантности данного метода к изменениям яркости.

2.6. Детектор BRISK

Для детектора BRISK будут также проанализированы результаты работы лишь на фотографии здания и её модификациях.



рис. 2.6.1. Особые точки, найденные методом BRISK на фотографии здания.

На фотографии здания детектор BRISK обнаружил меньшее количество точек, чем SIFT и SURF (249 штук), однако все они сосредоточены около выделяющихся частей изображения (рисунок 2.6.1). Это говорит о том, что метод хорошо обнаруживает особенности и объекты. Скорость работы алгоритма оказалась выше, чем SIFT, но меньше чем все остальные из ранее рассмотренных методов (для данного изображения точки были найдены за 2.57 секунд).



рис. 2.6.2. Особые точки, найденные методом BRISK на повернутой фотографии.

За небольшими исключениями, на повернутом изображении (рисунок 2.6.2) были обнаружены те же самые точки, что на исходном и в похожем количестве (293 точки за 2.59 секунд). Это говорит о инвариантности метода к поворотам.



рис. 2.6.3. Особые точки, найденные методом BRISK на увеличенной фотографии.

На увеличенной фотографии алгоритм допустил небольшое количество ошибок, найдя новые точки и не обнаружив некоторые из найденных ранее (рисунок 2.6.3). Однако таких ошибок немного и в целом результат для увеличенной фотографии похож на результат для исходной. Следовательно, метод BRISK частично инвариантен к изменениям масштаба.



рис. 2.6.4. Особые точки, найденные методом BRISK на фотографии с шумом.

Результат поиска особых точек на изображении с шумом с помощью метода BRISK представлен на рисунке 2.6.4. Обнаружено большое количество точек на монотонных частях изображения, таких как небо. Точки хаотично разбросаны по всей плоскости фотографии, что говорит о неинвариантности метода к появлению шума. Количество найденных точек заметно превышает предыдущие результаты для данного метода (1506 точек за 2.94 секунды).



рис. 2.6.5. Особые точки, найденные методом BRISK на затемнённой фотографии.

Из рисунка 2.6.5. можно сделать вывод, что метод BRISK инвариантен к изменениям яркости. Было найдено приблизительно похожее количество точек, что и для исходного изображения (168 штуки за 2.53 секунд), и располагаются они в тех же местах. Новых лишних точек почти не появилось, а найденные до этого были найдены и сейчас.

2.7. Сравнение результатов работы методов

Автором работы были проведены дополнительные тесты для каждого метода для изображений разного разрешения с целью сравнения скорости работы детекторов. Результаты тестов представлены в таблице 2.7.1.

	800x600	1280x960	1920x1440	4000x3000
Harris	0.385	0.982	2.127	10.344
Shi-Tomasi	2.449	9.223	17.562	75.422
FAST	0.012	0.034	0.071	0.237
SIFT	4.399	10.109	24.506	94.746
SURF	1.511	2.975	6.168	28.529
BRISK	2.633	2.770	3.023	4.630

Таблица 2.7.1. Время работы методов на изображениях разного разрешения (в секундах).

Из таблицы 2.7.1 можно сделать вывод, что время поиска особых точек всеми методами пропорционально количеству пикселей на изображении. Исключением является метод BRISK, время работы которого не так сильно зависит от разрешения. Время работы для изображения из 12000000 (4000x3000) пикселей всего в 1.75 раза выше, чем для изображения из 480000 (800x600) пикселей.

В следующей таблице (таблица 2.7.2) представлено сравнение алгоритмов по следующим параметрам: инвариантность к изменениям масштаба, инвариантность к поворотам, независимость от наличия шума, независимость от изменений яркости, скорость работы для изображений низкого и высокого разрешений, а также общее качество поиска точек по шкале от 1 до 5, где 1 — плохо, 5 — отлично.

	Инвариантность				Скорость работы		Качество найденных точек
	масштаб	поворот	шум	изменение яркости	Низкое разрешение	Высокое разрешение	
Harris	2	5	4	3	4	4	3
Shi-Tomasi	3	5	4	3	3	2	4
FAST	3	5	2	4	5	5	3
SIFT	4	3	2	5	1	1	4
SURF	5	4	2	4	4	3	5
BRISK	4	4	2	5	3	5	5

Таблица 2.7.2. Сравнение методов детектирования.

Выводы

На основе полученных результатов, автор работы может заявить, что среди рассмотренных в работе алгоритмов нет единственного, который подойдет для любой задачи, в которой используется поиск особых точек на изображении. Каждый из рассмотренных методов обладает своими достоинствами и недостатками и может быть наиболее подходящим для решения какой-либо задачи.

Основными достоинствами метода Харриса является инвариантность к повороту и шуму, а также достаточно высокая скорость работы. Он лучше всего подойдет для задач, при решении которых используется поиск особых точек на изображениях с ярко-выраженными углами, наличием шума. Он не подойдет для задач, в которых производится сравнение изображений разного масштаба и с большими различиями в яркости. Также стоит обратить внимание на детектор Харриса, если важна скорость работы метода. Примером задач, в которых детектор Харриса покажет хорошие результаты могут быть задачи, в которых используется поиск простых фигур на изображениях, созданных с помощью средств компьютерной графики, так как в таком случае все углы этих фигур будут хорошо заметны для алгоритма.

Детектор Shi-Tomasi очень похож по результатам на детектор Харриса, поэтому может использоваться в задачах похожего вида. Однако стоит учитывать меньшую скорость работы. По сравнению с методом Харриса он обладает более высоким качеством детектирования. Как результат, этот метод станет лучшим решением, если в задаче более важна точность, и не так важна скорость работы.

Детектор FAST обладает несравнимой скоростью работы, однако уступает предыдущим методам в качестве детектирования и показывает гораздо худший результат при анализе изображений с присутствующим на них шумом. Детектор FAST лучше всего подходит для задач, в которых время

детектирования играет ключевую роль, например слежение за объектом в видео или поиске объектов на видео. Его высокая скорость позволит работать с видео высокого разрешения в реальном времени.

Метод SIFT показал худший результат среди исследованных детекторов по многим характеристикам, однако обладает хорошим качеством детектирования и устойчивостью к изменениям яркости. SIFT оказался самым медленным из рассмотренных детекторов и не подойдёт для большинства задач, однако если в задаче совершенно не важна скорость работы и в сравниваемых изображениях нет шума и изображенные объекты не повёрнуты, но необходима высокая точность, то SIFT может быть выбран в качестве наиболее подходящего метода.

Метод SURF работает не так быстро как FAST или Harris, но обладает крайне высокой степенью устойчивости к афинным преобразованиям, изменениям яркости и качеством детектирования в целом. SURF будет лучшим выбором для задач, в которых критически важна точность детектирования. Примерами таких задач может быть построение стереопары, панорам и поиск объектов снятых с разных ракурсов и при разном освещении.

Последний из рассмотренных детекторов, BRISK отличается высокой скоростью работы на изображениях высокого разрешения. При качестве детектирования, сравнимом с качеством детектора SURF и независимостью от поворотов и изменения масштаба, он подойдёт для работы с большими изображениями, например при поиске совпадений на фотографиях со спутников, так как для таких фотографий характерно большое разрешение.

Заключение

Задачей данной выпускной квалификационной работы было исследование принципов работы детекторов особых точек на изображении и их сравнение на практике. Для этого было разработано программное обеспечение, реализующее методы детектирования и использовано для сравнения результатов их работы. На основе полученных результатов были сделаны выводы о сильных и слабых сторонах алгоритмов и возможных сценариях применения каждого из них.

Список литературы

- [1] Конушин А. Слежение за точечными особенностями сцены // Компьютерная графика и мультимедиа, Вып. №1(5)/2003.
- [2] Rodehorst V., Koschan A. Comparison and evaluation of feature point detectors, 2006.
- [3] Построение SIFT дескрипторов и задача сопоставления изображений <https://habrahabr.ru/post/106302/>
- [4] Moravec, H. Rover visual obstacle avoidance // In International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981, pp. 785-790.
- [5] Harris, C. and Stephens, M. A combined corner and edge detector // In Fourth Alvey Vision Conference, Manchester, UK, 1988, pp. 147-151.
- [6] Shi T. Good Features to Track, 1994.
- [7] E. R. a. T. Drummond. Fusing Points and Lines for High Performance Tracking, 2005.
- [8] R. P. a. T. D. Edward Rosten. Faster and better: a machine learning approach to corner detection, 2008.
- [9] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, 2004
- [10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool,

"Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven

[11] Stefan Leutenegger, Margarita Chli and Roland Siegwart: BRISK: Binary Robust Invariant Scalable Keypoints. ICCV 2011: 2548-2555.