

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

И МНОГОПРОЦЕССОРНЫХ СИСТЕМ

Ивашечкин Александр Павлович

Выпускная квалификационная работа бакалавра
Методы нахождения особых точек изображения
и их дескрипторов

Направление 010400

Прикладная математика

и информатика

Научный руководитель,

Кандидат тех. наук, доцент

Гришкин В.М.

Санкт-Петербург

2016

Содержание

Введение	3
Постановка задачи	5
Глава 1. Методы нахождения особых точек	6
1.1. SURF	6
1.2. BRISK	7
1.3. Harris	7
1.4. FAST	10
1.5. MSER	11
1.6. Minimal eigenvalue	12
Глава 2. Дескрипторы	13
2.1. SURF	13
2.2. BRISK	14
2.3. Block	15
Глава 3. Реализация	16
3.1. Структура программы	16
3.2. Результат	17
Выводы	18
Заключение	19
Список литературы	20
Приложение	21

Введение

Задача сопоставления изображений используется следующих целей: Создание панорам, стереопары, воссоздание 3D-модели какого-либо объекта по его нескольким двумерным изображениям, распознавание изображений и нахождение на них объектов по каким-либо критериям слежение за движущимися объектами. Для их сопоставления необходимо применить методы поиска точек, которые являются общими на обоих изображениях, а также поиск дескрипторов (описаний) этих точек. Нужно учитывать, что на сегодняшний день нет полноценного универсального метода, который подошел бы под текущие задачи. Нельзя забывать о том, что существуют разные методы для разных изображений: характер сцены, разные предпочтения, тип объектов и т.д.

Человек по изображению может интуитивно понять, что на нем находится. Для компьютера же любое изображение – ничем не говорящий набор данных. Необходимо определить, как же все-таки компьютер определяет изображение.

У каждого изображения есть некие особые точки. Что это такое? Особые точки – такие точки, по которым можно классифицировать изображение, распознать его, некая особенность изображения, уникальность. Как правило – это угловые точки, либо те, где резко меняется цвет, яркость, и т.д. Нужно выбирать такие точки, которые вносят некий вклад в характеристику изображения, также необходимо считать особыми такие точки, которые с большой вероятностью будут найдены на другом изображении. Каждый метод обнаружения особых точек должен гарантировать инвариантность относительно любых преобразований изображения.

Осталось понять, каким образом компьютер понимает – какие ключевые точки разных изображений соответствуют друг другу. Ведь у

каждой точки на разных изображениях разные координаты, каким образом идет их сопоставление? Для этого каждой особой точке необходимо присвоить описание, которое будет одинаковым на разных изображениях.

Дескриптор – идентификатор особой точки, который делает её уникальной относительно остальных особых точек. Разумеется, используя дескрипторы, нельзя забывать об инвариантности относительно преобразования изображений.

В итоге получается следующая инструкция по сопоставлению изображений:

1. На изображениях находятся особые точки и их дескрипторы
2. По дескрипторам среди этих точек выявляются пары соответствующих точек
3. По этим парам идет построение модели преобразования изображений

В нашем случае будет подробно рассматриваться первый пункт инструкции. Ниже перечислено несколько преобразований, относительно которых нужно получить инвариантность:

1. Изменение яркости
2. Изменение размера (масштабирование)
3. Изменение положения камеры
4. Вращение
5. Смещение

Постановка задачи

В текущей работе был использован программный пакет «MATLAB 2014a», а также набор функций «Computer Vision System Toolbox», рассматривались следующие методы поиска особых точек:

1. SURF
2. BRISK
3. Harris
4. FAST
5. MSER
6. Minimal Eigenvalue

Для этих методов есть дескрипторы, которые наиболее быстро и точно позволяют сопоставить особые точки. Но насколько эти дескрипторы хорошо подходят к каждому методу? Для каждого ли типа изображения они будут хорошо работать?

Для исследования было взято 3 типа изображений: пейзаж, портрет и текстовый документ. Необходимо было проанализировать: какая пара «метод/дескриптор» наилучшим образом с минимальными ошибками и максимальным количеством пар особых точек подходит для того или иного типа изображений.

Дескрипторы же были использованы следующие:

1. SURF
2. BRISK
3. FREAK
4. Block

Следует ознакомиться с методами и дескрипторами в подробности, как они работают.

Глава 1. Методы нахождения особых точек

1.1. SURF

Как работает метод SURF:

- Выполняется поиск особых точек с помощью Гессиана (матрицы Гессе)

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

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

где H — матрица Гессе, $f(x, y)$ — функция изменения градиента яркости.

- Для найденных точек вычисляется направления наибольшего изменения яркости
- Перебираются масштабы матрицы Гессе по октавам
- 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 использует фильтры различных масштабов для вычисления Гессианов. Затем для каждой особой точки вычисляется градиент и масштаб. Градиент в точке вычисляется при помощи фильтров Хаара. Размерность фильтра берется равным $4s$, где s — 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. 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.

1.2. BRISK

Основное отличие текущего метода 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.

1.3. Harris

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

Для изображения I рассмотрим некоторое окно W (как правило, размер окна 5×5 , но может зависеть от размера изображения) с центром в точке (x, y) , а также его сдвиг на (u, v) (рис. 1).

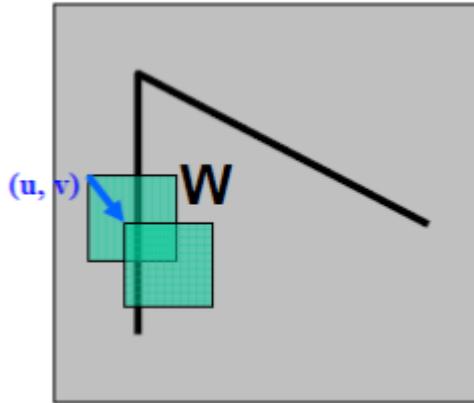


Рис.1. Окно с центром в (x,y) и его сдвиг на (u,v)

Сумма квадратов разностей между исходным и сдвинутым окном будет равна:

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

где $w(x,y)$ – весовая функция.

$$w(x,y) = \begin{array}{c} \text{[Red step function]} \\ \text{1 – в окне, 0 – вне окна} \end{array} \quad \text{или} \quad \begin{array}{c} \text{[Red Gaussian curve]} \\ \text{Функция Гаусса} \end{array}$$

M – автокорреляционная матрица:

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

При больших изменениях функции $E(x,y)$ по всевозможным направлениям (x,y) определяется угол, следовательно получаются большие по модулю собственные значения матрицы M . Ниже приведено расположение собственных значений (рис. 2).

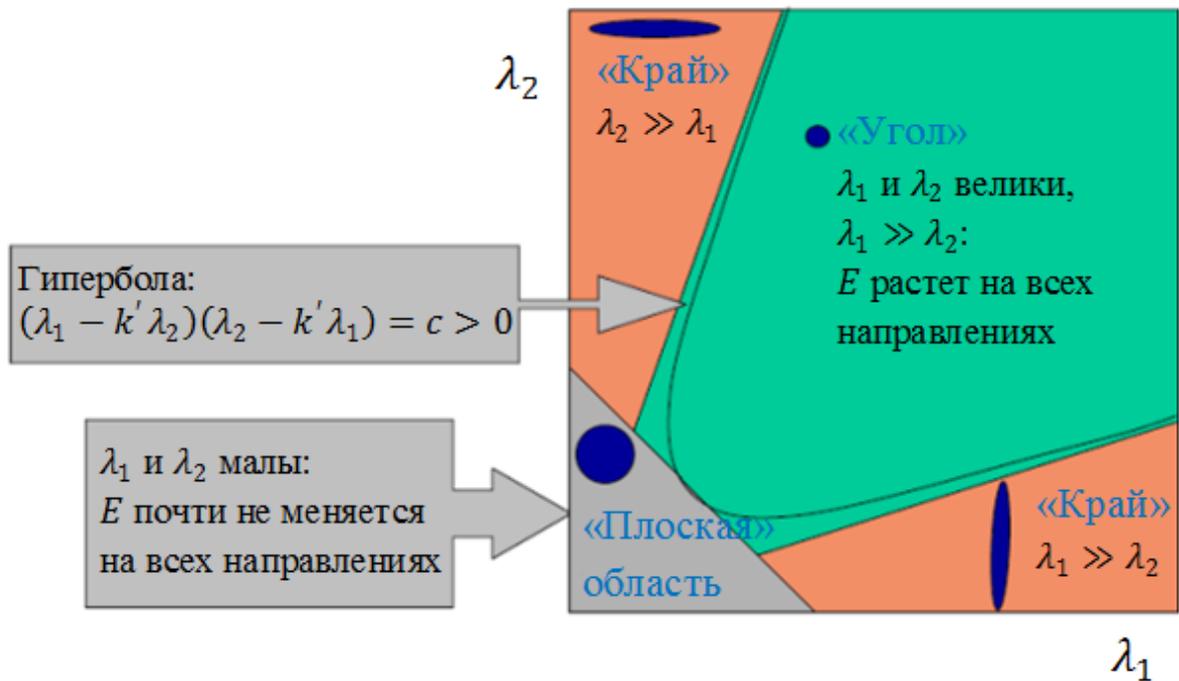


Рис.2. Расположение собственных значений

В связи с трудоемкостью вычисления собственных чисел матрицы Харрис и Стефан предложили ввести меру отклика: $R = \det M - k(\text{tr} M)^2 > k$, где k – эмпирическая константа, $k \in [0,04; 0,06]$. В таком случае, значение R будет положительно для угловых особых точек. Затем убираются точки, которые меньше некоторого порога R (те, у которых значение R меньше какого-либо порога). Далее вычисляются локальные максимумы функции R 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 ином случае используют многомасштабный метод Харриса (multi-scale Harris detector))

1.4. 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 решений для классификации пикселей.

Для каждого пикселя p рассматривается окружность радиусом 4, вписанная в квадратную область со стороной 7 пикселей (рис.3).

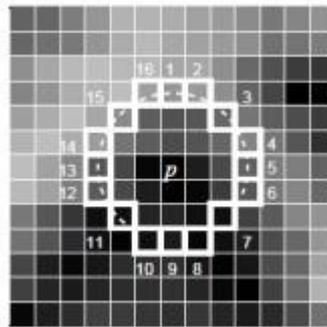


Рис.3. Рабочая окрестность пикселя при использовании FAST детектора

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

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

Для каждого x и найденного $S_{p \rightarrow x}$ для каждого $p \in \mathbb{P}$ (\mathbb{P} - множество всех пикселей тренировочного набора изображений) разделим множество \mathbb{P} на 3 подмножества $\mathbb{P}_d, \mathbb{P}_s, \mathbb{P}_b$ точек, которые темнее, схожи либо светлее точки x соответственно. Затем строится дерево решений. По

результатам построенного дерева определяются углы на тестовых изображениях.

1.5. MSER

Метод MSER решает проблему инвариантности особых точек при разных масштабах изображения. Выделяются несколько различных регионов с экстремальными свойствами функции интенсивности как внутри региона, так и на её границах.

Алгоритм рассматривается для черно-белого изображения. Представляются всевозможные отсечения изображения. В итоге получается набор бинарных изображений при разных значениях некоторого порога $\tau \in \{0, 1, \dots, 255\}$ (пиксель, значение которого меньше этого порога считается черным, иначе – белым). Далее строится пирамида, у которой при минимальном пороговом значении получается белое изображение, при максимальном – черное. В случае, если замечено какое-либо движение – на белом изображении появляются черные пятна, являющиеся локальными минимумами интенсивности. При увеличении порога черные пятна начинают увеличиваться и сливаться, в конечном итоге образуя полностью черное изображение. Такая пирамида позволяет построить множество связанных компонент, соответствующих белым областям – регионам с максимальным значением интенсивности. Соответственно, при инвертировании изображения получим черные пятна – регионы с минимальным значением интенсивности.

Алгоритм состоит из нескольких этапов:

- Сортируется множество всех пикселей изображения в порядке возрастания/убывания интенсивности. Такая сортировка возможна за время, пропорциональное количеству пикселей.

- Выполняется построение пирамиды связных компонент. Для каждого пикселя отсортированного множества выполняется последовательность действий:
 - обновление списка точек, входящих в состав компоненты
 - обновление областей следующих компонент, в результате чего пиксели предыдущего уровня будут подмножеством пикселей следующего уровня
- Выполняется поиск локальных минимумов для всех компонент, т.е. поиск пикселей, присутствующих в данной компоненте, но не входящих в состав предыдущих). Набор локальных минимумов уровня соответствует экстремальному региону на изображении.

1.6. Minimal Eigenvalue

Метод Ши-Томаси (Shi-Tomasi или Kanade-Tomasi) очень схож с методом Харриса, но отличается при вычислении меры отклика: алгоритм напрямую вычисляет значение $\min(\lambda_1, \lambda_2)$. Делается предположения, что при использовании этого алгоритма поиск углов будет более стабильным.

Глава 2. Дескрипторы

2.1 SURF

Алгоритм построения дескриптора следующий:

- Вокруг точки строится квадратная окрестность размером $20s$, где s – масштаб, на котором получено максимальное значение детерминанта матрицы Гессе
- Полученная квадратная область разбивается на блоки, в результате область будет разбита на 4×4 региона (рис. 4)

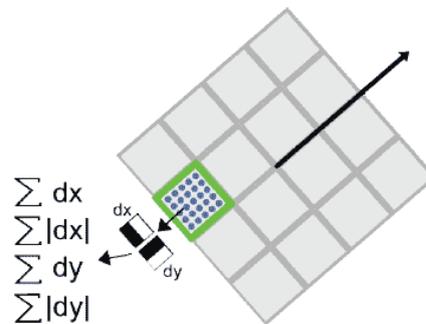


Рис.4. Область, разбитая на 4×4 региона

- 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 (рис.5)

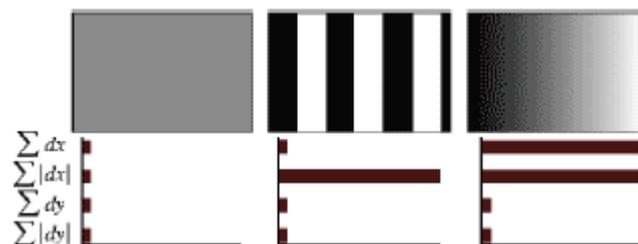


Рис.5. Поведение величин сумм для разных участков изображений

- 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 удаленных точках.

- Дополнительно к дескриптору заносится след матрицы Гессе. Эти компоненты необходимы, чтобы различать темные и светлые пятна. Для светлых точек на темном фоне след отрицателен, для темных 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.

2.2 BRISK

Область вокруг особой точки разбивается на 60 участков p (рис. 6):

$$\mathcal{A} = \{(p_i, p_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\}$$

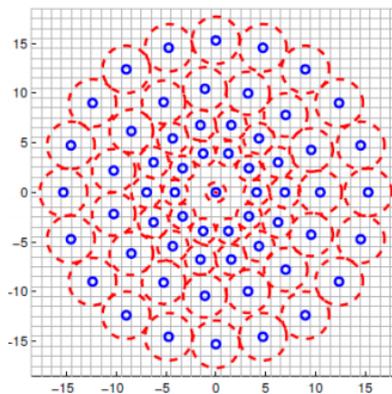


Рис.6. Область вычисления дескриптора

Множество \mathcal{A} разбивается на 2а подмножества:

$$\mathcal{S} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| < \delta_{max}\} \subseteq \mathcal{A}$$

$$\mathcal{L} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| > \delta_{min}\} \subseteq \mathcal{A}$$

где $\delta_{min} = 13.67t$, $\delta_{max} = 9.75t$, t – размер особой точки.

Вычисляется среднее значение градиента множества \mathcal{L} :

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{|\mathcal{L}|} * \sum_{(p_i, p_j) \in \mathcal{L}} \left[(p_j - p_i) \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \right]$$

Дескриптор состоит из бинарной строки длиной 512, заполненной результатами проведенных тестов в множестве \mathcal{S} :

$$b = \begin{cases} 1, I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i); \forall (p_i^\alpha, p_j^\alpha) \in \mathcal{S}, \\ 0, \text{ иначе} \end{cases}$$

где $I(p_i^\alpha, \sigma_i)$ интенсивность окрестности радиуса σ_i точки p_i ,

$\alpha = \arctan 2(g_y, g_x)$ – угол направления градиента g .

2.3 Block

Алгоритм очень прост: вокруг особых точек берется окрестность с заданным радиусом (обычно берется радиус 11 пикселей), затем формируется дескриптор, состоящий из множества чисел, являющихся параметрами яркости пикселей.

Глава 3. Реализация

3.1. Структура программы

Для нахождения особых точек изображения используются следующие функции:

```
detectSURFFeatures (I) ;  
detectBRISKFeatures (I) ;  
detectHarrisFeatures (I) ;  
detectFASTFeatures (I) ;  
detectMSERFeatures (I) ;  
detectMinEigenFeatures (I) ;
```

Для начала программа считывает изображения и преобразует их в черно-белый формат с помощью функций `rgb2gray` (`imread('1.JPG')`). Затем выполняется поиск особых точек с помощью вышеуказанных функций.

Далее описываются дескрипторы точек функцией `extractFeatures`, в которой в качестве одного из параметров указываем необходимый нам дескриптор. Затем по этим дескрипторам создаются пары соответствий между двумя изображениями функцией `matchFeatures`. После этого линии между схожими точками проецируются на изображения, которые размещены рядом друг с другом с помощью функции `showMatchedFeatures`.

Проанализировав все рассматриваемые методы и дескрипторы, было проведено исследование, какие же из них наиболее подходят для тех или иных типов изображений. На основании принципов их работы можно объяснить результаты проведенных наблюдений. Ниже предоставлена таблица, на которой можно ознакомиться со средним количеством особых точек и процентным содержанием правильных сопоставлений, а также анализ проведенных исследований с пояснениями причин высокой эффективности работы методов и дескрипторов на разных типах изображений.

3.2. Результат

Дескриптор Метод	SURF	BRISK	FREAK	Block
SURF	Пейзаж 266/267 99% Портрет 30/35 86% Документ 1000/1200 83%	Пейзаж 182/182 100% Портрет 15/29 50% Документ 1450/1456 99%	Пейзаж 157/157 100% Портрет 3/4 75% Документ 228/234 97%	Пейзаж 140/172 81% Портрет 12/36 33% Документ 357/562 64%
BRISK	Пейзаж 24/25 96% Портрет 2/9 22% Документ 140/186 75%	Пейзаж 21/21 100% Портрет 1/16 6% Документ 97/100 97%	Пейзаж 18/18 100% Портрет 0/6 0% Документ 57/66 83%	Пейзаж 35/43 81% Портрет 0/4 0% Документ 350/543 64%
Harris	Пейзаж 343/346 99% Портрет 5/16 31% Документ 600/782 76%	Пейзаж 330/330 100% Портрет 5/5 100% Документ 1290/1298 99%	Пейзаж 211/211 100% Портрет 0/3 0% Документ 200/227 88%	Пейзаж 383/386 98% Портрет 7/7 100% Документ 800/1043 80%
FAST	Пейзаж 152/152 100% Портрет 0/3 0% Документ 800/1284 62%	Пейзаж 127/127 100% Портрет 0/0 0% Документ 1600/1665 96%	Пейзаж 79/79 100% Портрет 1/7 14% Документ 500/556 90%	Пейзаж 200/203 98% Портрет 2/4 50% Документ 1200/1500 80%
MSER	Пейзаж 146/146 100% Портрет 38/46 84% Документ 100/126 79%	Пейзаж 65/65 100% Портрет 0/0 0% Документ 68/72 94%	Пейзаж 54/54 100% Портрет 3/4 75% Документ 23/27 85%	Пейзаж 72/80 90% Портрет 1/2 50% Документ 200/272 74%
MinEigen	Пейзаж 894/905 98% Портрет 111/172 65% Документ 500/939 53%	Пейзаж 835/835 100% Портрет 36/40 90% Документ 270/297 90%	Пейзаж 392/394 99% Портрет 3/3 100% Документ 240/297 80%	Пейзаж 700/767 91% Портрет 14/15 93% Документ 600/1151 52%

Выводы

Для пейзажей лучше всего подходит метод Minimal Eigenvalue, т.к. он схож с методом Harris, который инвариантен к аффинным преобразованиям, но чувствителен к шумам, более стабильный поиск углов, которые встречаются на фотографиях зданий с высоты гор, деревья, и т.д. Deskriptor же лучше всего использовать BRISK.

Для портретов наиболее подходит метод MSER: инвариантность к вращениям, хорошо подходит для нахождения сходств по шаблону, метод ищет точки внутри регионов и на внешних границах. Наиболее подходящий дескриптор – SURF, где опять же встречается инвариантность к вращениям и изменениям масштаба; большинство точек находятся на границах и в ярко-выраженных местах, что свойственно данному дескриптору.

Для текстовых документов отлично подходит метод Harris, который может и чувствителен к шумам, но в данном случае этим можно пренебречь. Метод FAST нашел больше точек, но соответствий при каждом дескрипторе значительно меньше. Больше всего соответствий при использовании дескриптора BRISK.

Заключение

На сегодняшний день существует множество программ по построению панорам. Известно, что при работе большинства из них в процессе склейки изображений на протяжении всей панорамы использовался один метод и дескриптор. Но что если на панораме будет несколько типов изображений? Рассмотрим ситуацию: пользователь фотографирует пейзаж, на кадре появляется портрет, после портрета идет макросъемка, а далее на кадре присутствует какой-либо документ. Большинство программ будет склеивать все одним способом. Проведенные исследования могут послужить для написания более универсальной программы, где при фотосъемке панорамы изначально будет определяться тип фотографии методом распознавания изображения (с использованием сегментаций, к примеру), затем будет подбираться метод нахождения точек и их описания, после чего произойдет сопоставление кадров с минимальными ошибками.

Список литературы

1. Детекторы углов. <https://habrahabr.ru/post/244541/>
2. Ипатов Ю.А., Кревецкий А.В. Методы обнаружения и пространственной локализации групп точечных объектов // Журнал "Кибернетика и программирование". Содержание № 06, 2014
3. Золотых Н.Ю., Кустикова В.Д., Мееров И.Б. Обзор методов поиска и сопровождения транспортных средств на потоке видеоданных // Вестник Нижегородского университета им Н.И. Лобачевского. Выпуск № 5-2 / 2012
4. Применение метода SURF в системах контроля и управления доступом <https://habrahabr.ru/post/152679/>
5. М.О. Гончаренко. Сравнительный анализ методов формирования дескрипторов изображений в контексте задачи сегментации видеопотока. // Бионика интеллекта. 2015.

Приложение

Приложение 1. Пример программы для нахождения особых точек методом Harris и дескрипторов Block

```
I1c=imread('1.JPG');
I2c=imread('2.JPG');
I1 = rgb2gray(I1c);
I2 = rgb2gray(I2c);

points1 = detectHarrisFeatures(I1);
points2 = detectHarrisFeatures(I2);
[f1,vpts1] = extractFeatures(I1,points1, 'Method', 'Block');
[f2,vpts2] = extractFeatures(I2,points2, 'Method', 'Block');
indexPairs = matchFeatures(f1,f2);
matchedPoints1 = vpts1(indexPairs(:,1));
matchedPoints2 = vpts2(indexPairs(:,2));

figure; showMatchedFeatures(I1c, I2c, matchedPoints1, matchedPoints2,
'montage');
```

Приложение 2. Примеры изображений

Пейзаж:



Портрет:



