

Цифровая обработка изображений

2. Извлечение признаков и поиск

План занятия

- Задача поиска по шаблону
- Представление свойств изображения с помощью гистограмм
 - Гистограммы цветов
 - Гистограммы градиентов

План занятия

- Характерные точки
 - Поиск характерных областей на изображении
 - Выделение признаков (дескрипторов) характерных областей на изображении
 - Матчинг характерных точек на изображениях
 - Пример

План занятия

- Поиск изображений по контенту CBIR
 - выделение признаков и индексация
 - обзор архитектуры
 - поиск по индексу

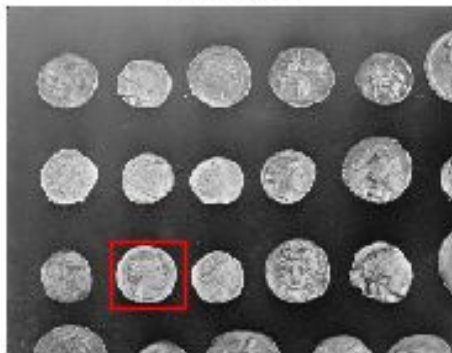
Примеры задач компьютерного зрения

Пример. Поиск по шаблону

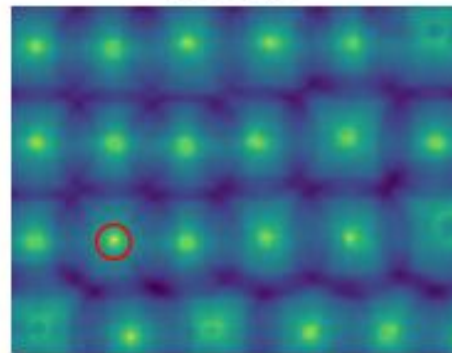
template



image



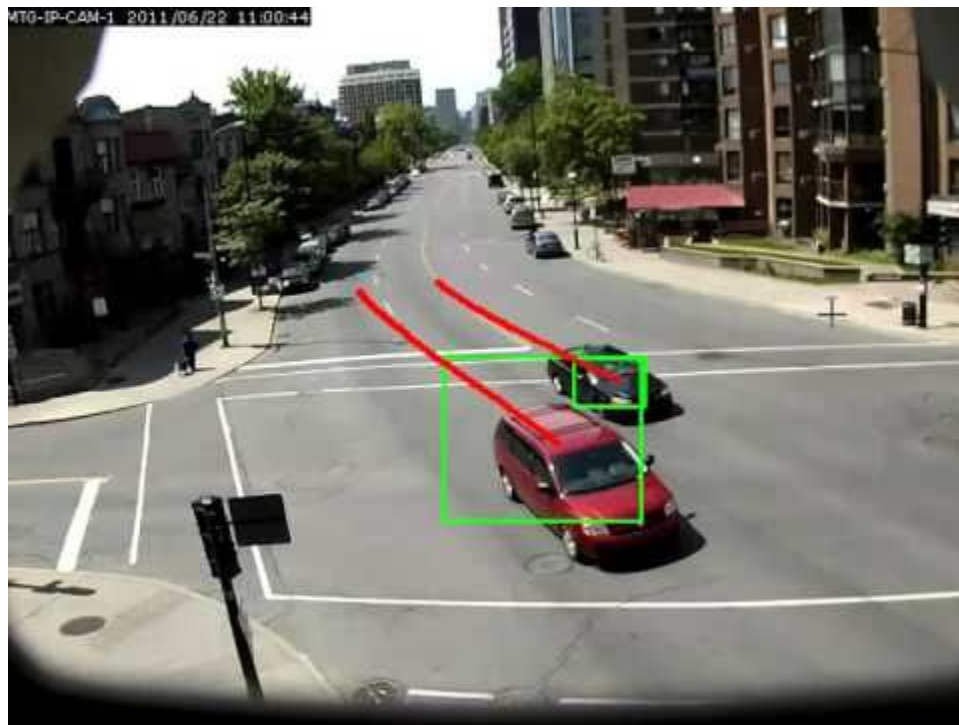
`match_template`
result



Пример. Склейка (stitching)



Пример. Optical Flow



Пример. Поиск похожих изображений (CBIR)

Query Image



Retrieved Results

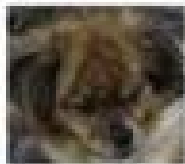


Поиск по шаблону

Поиск по шаблону



+



=

x2



Поиск по шаблону

- итеративно сканируем изображение
- на каждой итерации сравниваем шаблон с изображением
- определяем то место на изображении, которое наиболее похоже на шаблон

Поиск по шаблону

- квадратичное отклонение

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

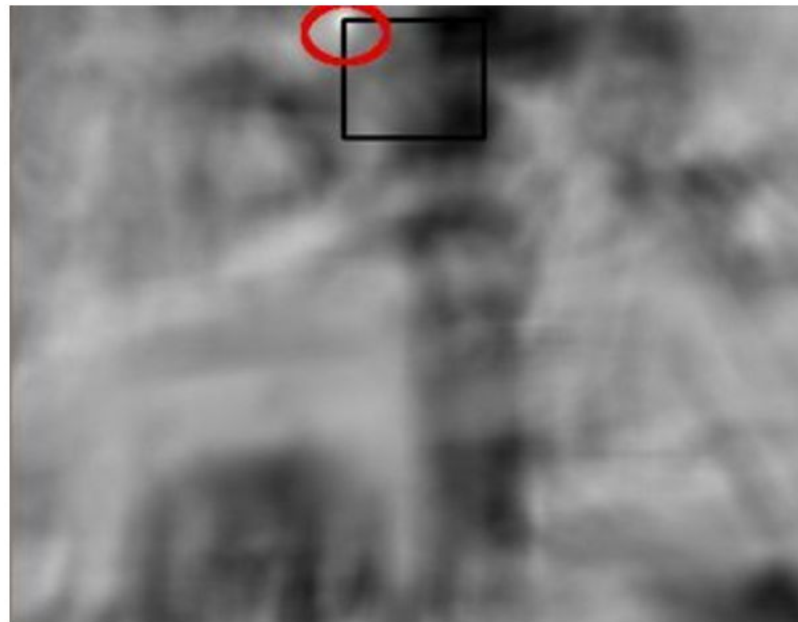
- коэффициент корреляции

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$$

T - шаблон

I - изображение

Оцениваем корреляцию



Поиск по шаблону

Python: `cv2.matchTemplate`(image, templ, method[, result]) → result

image - изображение на котором необходимо найти шаблон

templ - шаблон для поиска

method - метод сравнения (квадратичное отклонение, корреляция)

result - результат сравнения

Python: `cv2.minMaxLoc`(src[, mask]) → minVal, maxVal, minLoc, maxLoc

src – массив для поиска минимального и максимального элемента

mask - маска для ограничения региона поиска

minVal/maxVal - минимальное/максимальное значение

minLoc/maxLoc - точка, соответствующая минимальному/максимальному значению

Поиск по шаблону

- простой в реализации метод
- не устойчив к изменениям изображения
 - масштабирование
 - поворот
 - изменение освещенности
 - перспективные искажения

Выделение признаков изображения

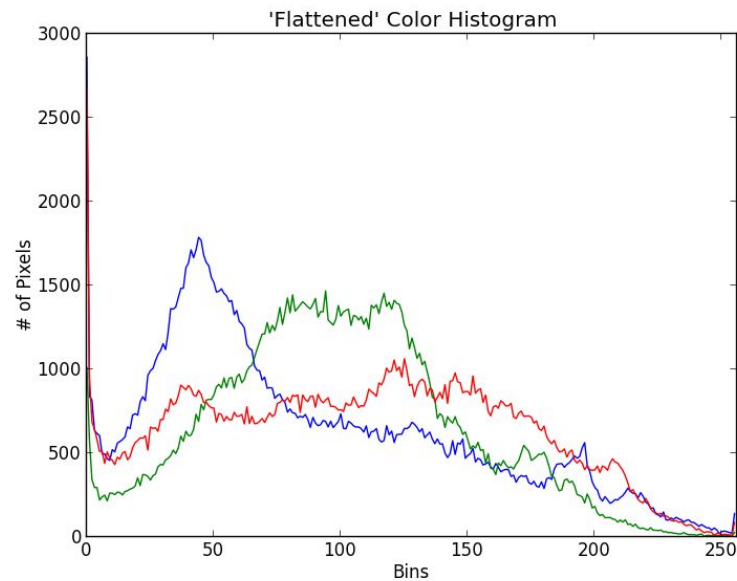
Гистограммы признаков изображения

- представляют собой обобщенное описание изображения
- позволяют сравнивать и находить похожие изображения

Гистограммы цветов

- разбиваем диапазон значений цвета (0..255) на фиксированное число ячеек
- для каждой ячейки считаем число соответствующих пикселей на изображении
- можно вычислять как в пространстве RGB, так и в других цветовых пространствах, например, HSV

Гистограммы цветов



Гистограммы цветов

`cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])` → `hist`

images – набор входных изображений для оценки гистограммы

channels – каналы по которым оцениваются гистограммы

mask – маска ограничивает область оценки гистограммы

histSize – массив размеров гистограмм по каждому измерению

ranges – пороги разбиения диапазонов значений каждого измерения

hist – результат оценки гистограммы (опционально)

accumulate – если значение True, то результат суммируется с `hist`

Гистограммы цветов

- не зависят от изменения масштаба изображения
- устойчивы к повороту и перспективным искажениям
- в цветовых пространствах HSV и HSL менее чувствительны к изменению яркости

Гистограммы градиентов (HOG)

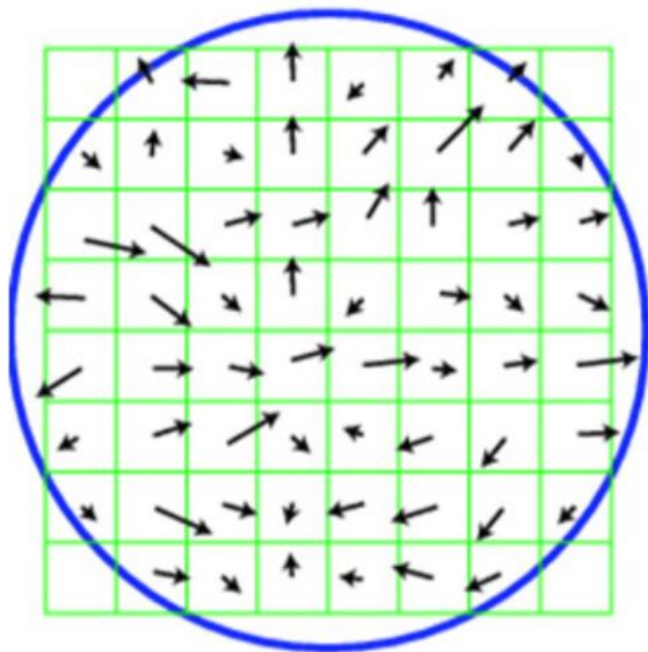
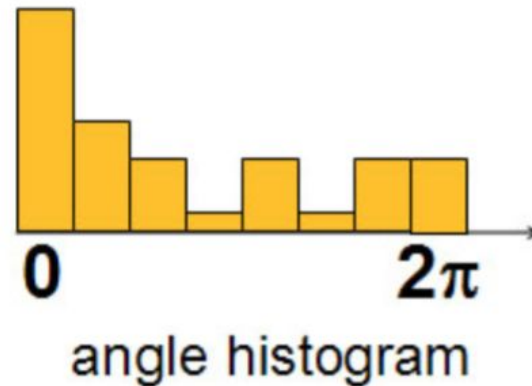


Image gradients



angle histogram



Гистограммы градиентов (HOG)

- в каждой точке оцениваем составляющие градиента по осям x и y
- определяем направление и длину вектора градиента
- оцениваем гистограмму градиентов
- циклически сдвигаем гистограмму таким образом, чтобы максимальное значение было в нулевой ячейки гистограммы
- полученные гистограммы нормализуют, таким образом, чтобы вектор признаков был единичной длины

Оператор Собеля

-1	0	+1
-2	0	+2
-1	0	+1

x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

Оператор Собеля

Original



Sobel X



Sobel Y



Гистограммы градиентов (HOG)

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

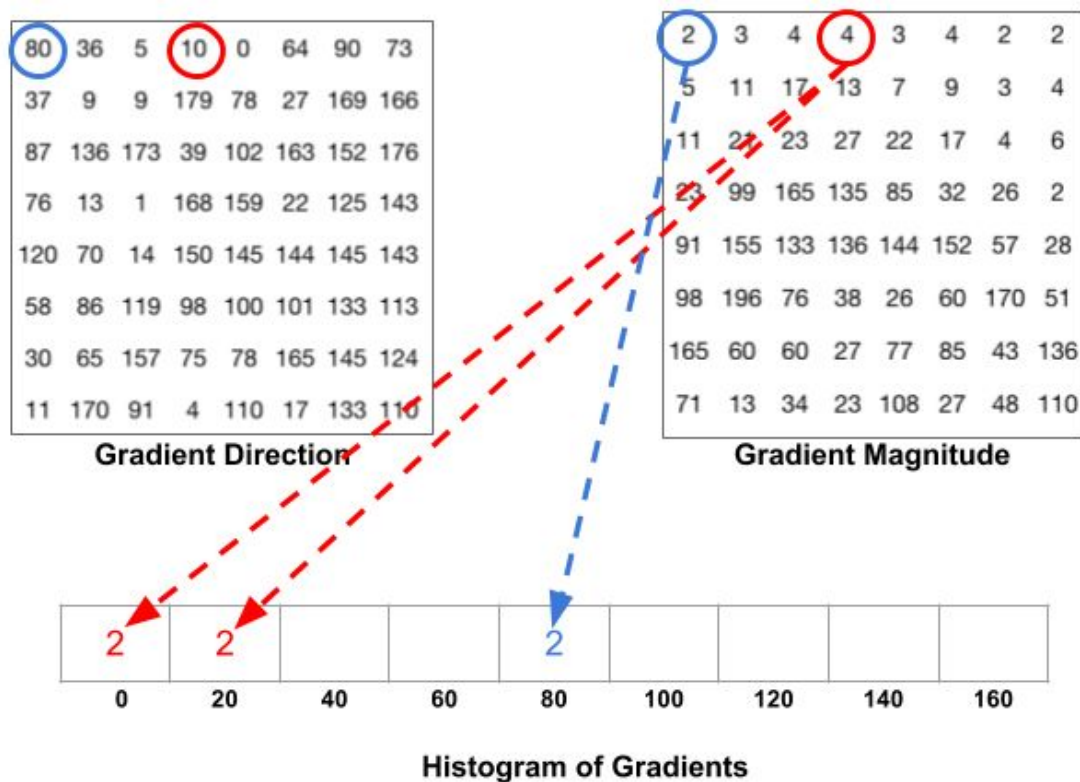
g , g_x , g_y - длина вектора градиента и его составляющих

θ - угол наклона градиента в полярной системе координат

Гистограммы градиентов (HOG)

- как правило гистограмму градиентов строят для диапазона углов $0..180$
- при оценке гистограммы градиентов учитывается как угол, так и длина вектора
- чем больше длина вектора, тем больший вклад вносится в соответствующую ячейку гистограммы

Гистограммы градиентов (HOG)



Гистограммы градиентов (HOG)

Input image



Histogram of Oriented Gradients



Гистограммы градиентов (HOG)

`cv2.Sobel(src, ddepth, dx, dy[, dst[, ksize]])` → `dst`

src – входное изображение

ddepth - тип данных для вычисления производной, например, `cv2.CV_64F`

dx/dy - порядок производной по осям, как правило 0 или 1

dst - выходное изображение

ksize – размер ядра фильтра 1, 3, 5, или 7

`cv2.cartToPolar(x, y)` → `magnitude, angle`

x,y – вектора с координатами x и y

magnitude - длины векторов

angle - соответствующие углы

Гистограммы градиентов (HOG)

- не чувствительны к изменению цвета
- устойчивы к изменению яркости
- не зависят от поворота

Характерные точки

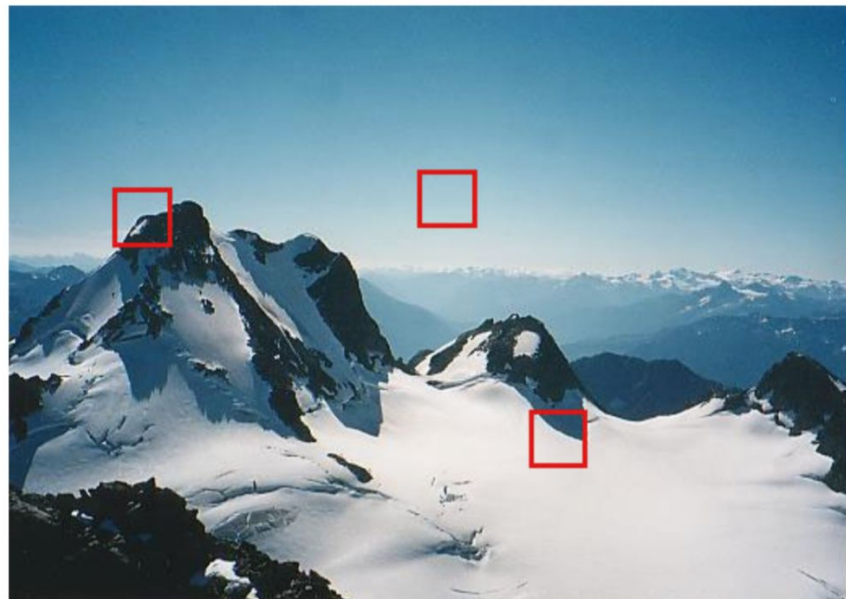
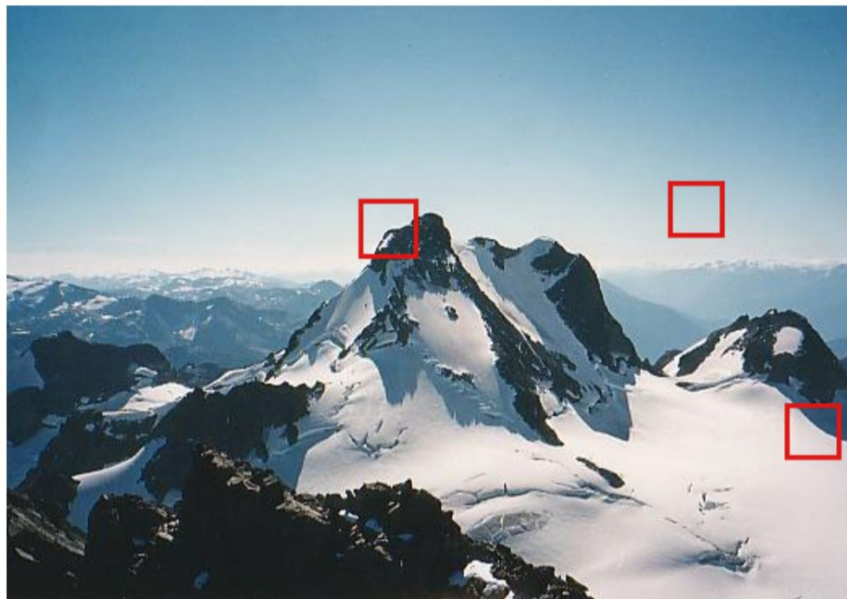
Характерные точки

- позволяют находить одинаковые области на изображениях
- используются для склейки панорам и составления карт по спутниковым снимкам

Характерные точки

- точка, обладающая уникальными свойствами
- положение точки на изображении однозначно определяется по ее свойствам (дескриптору)
- дескриптор точки вычисляется на основе ее окружения
- дескриптор характерной точки инвариантен к изменениям изображения (освещенность, поворот, масштабирование)

Характерные точки



Этапы поиска и матчинга характерных точек

1. Определяем области на изображении, которые наиболее вероятно содержат характерную точку
2. Вычисляем дескрипторы точки по каждой из областей
3. Находим точки с одинаковыми дескрипторами для матчинга изображений

Поиск характерной точки на изображении

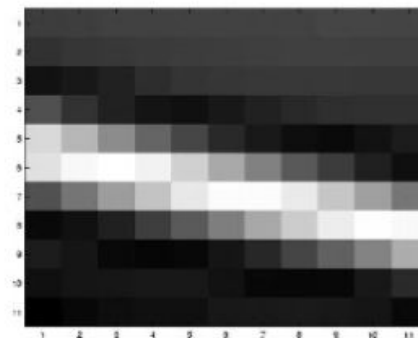
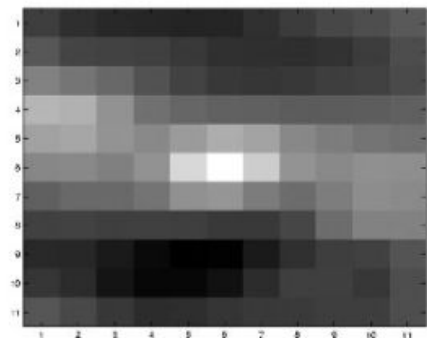
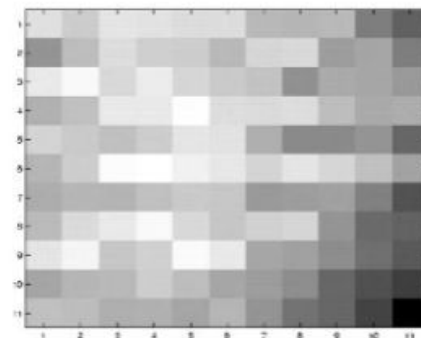
- Как понять что выбранная область содержит характерную точку?
- Область вокруг характерной точки должна сильно варьироваться
- В области характерной небольшой сдвиг изображения должен приводить к существенному различию по сравнению с исходным изображением

Автокорреляция

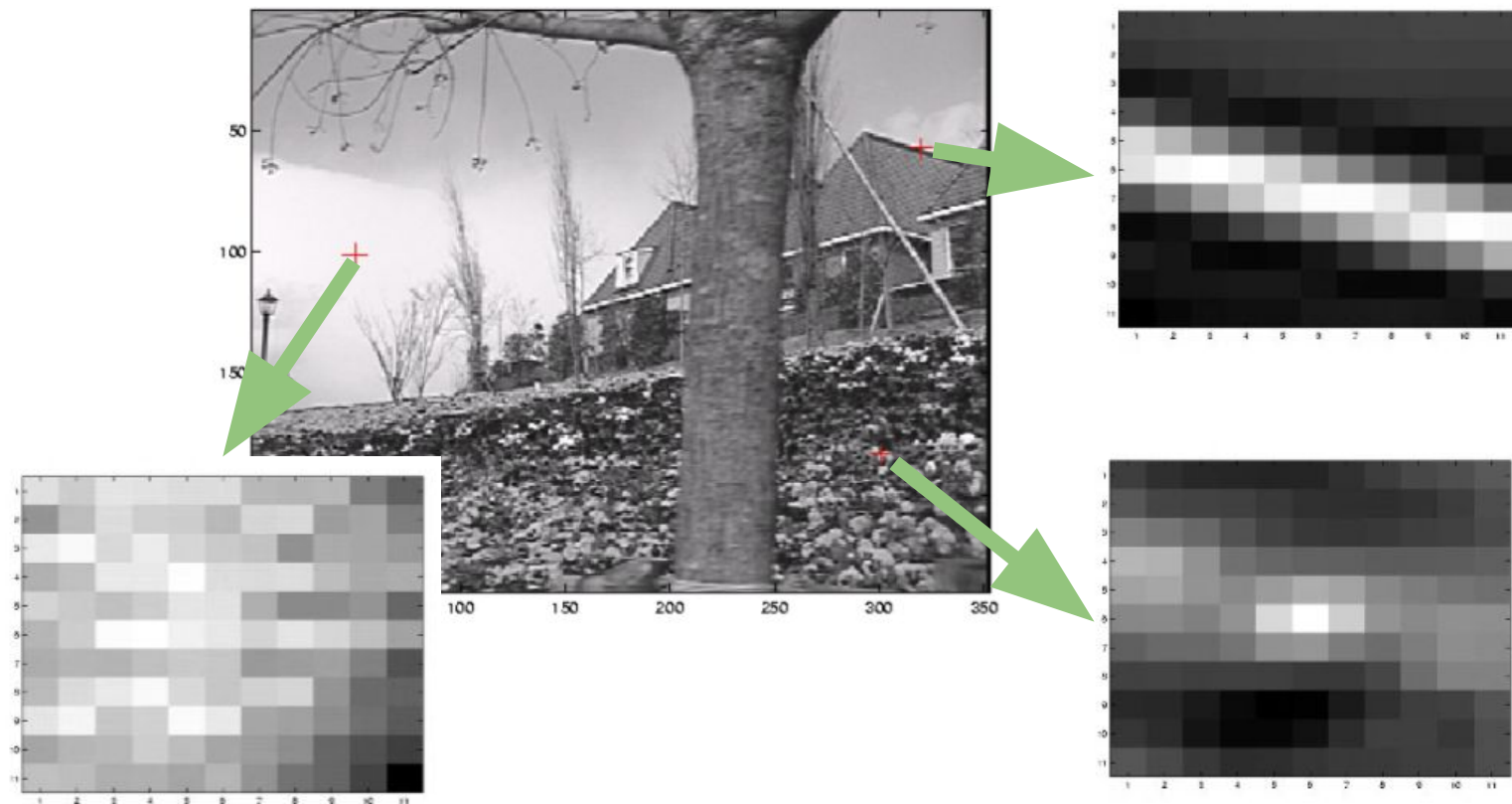
$$E_{AC}(\Delta \mathbf{u}) = \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

$\Delta \mathbf{u}$ - вектор смещения по осям x и y
 \mathbf{x}_i - вектор координат пикселя изображения
 w - окно или фильтр
 I_0 - исходное изображение

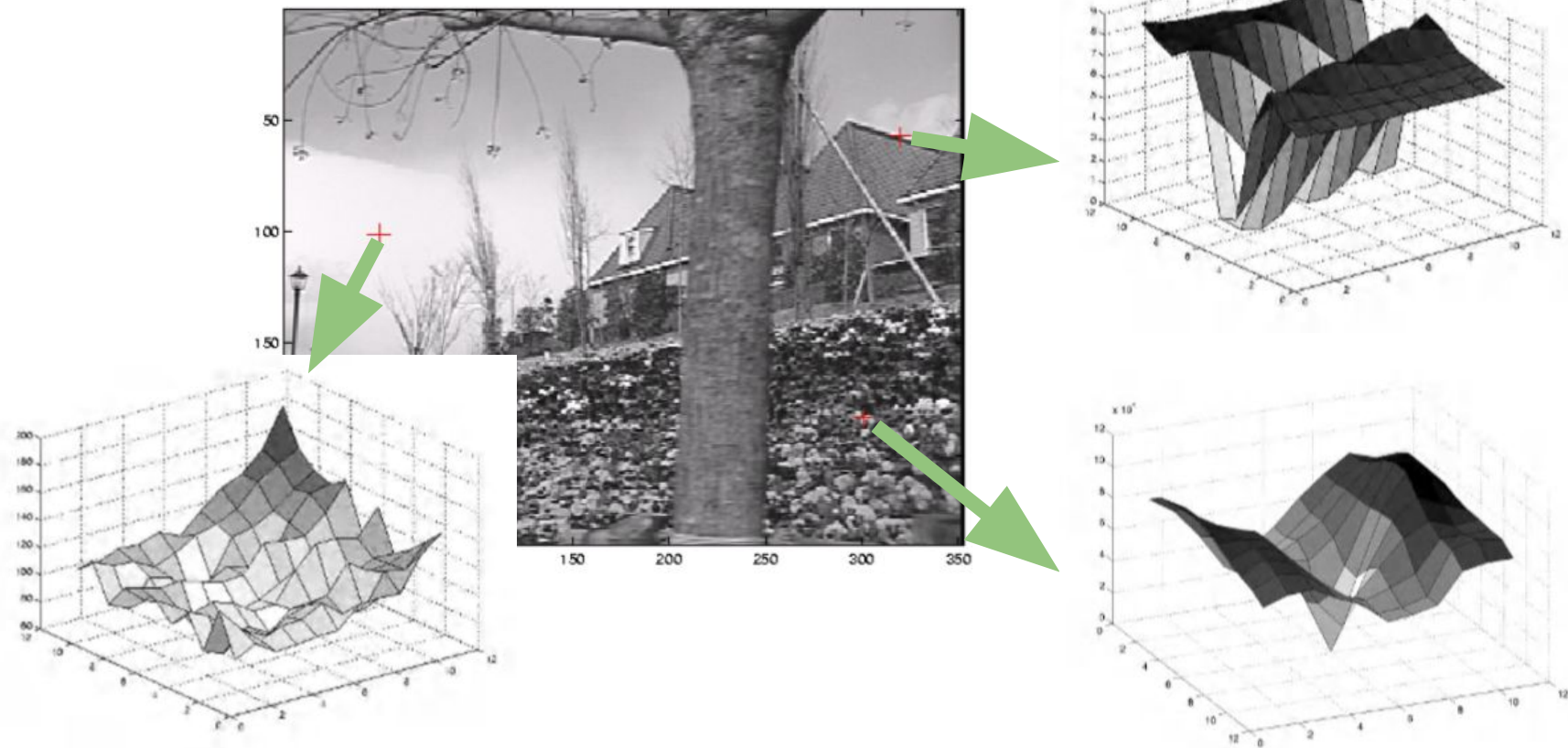
Матрица автокорреляции



Матрица автокорреляции



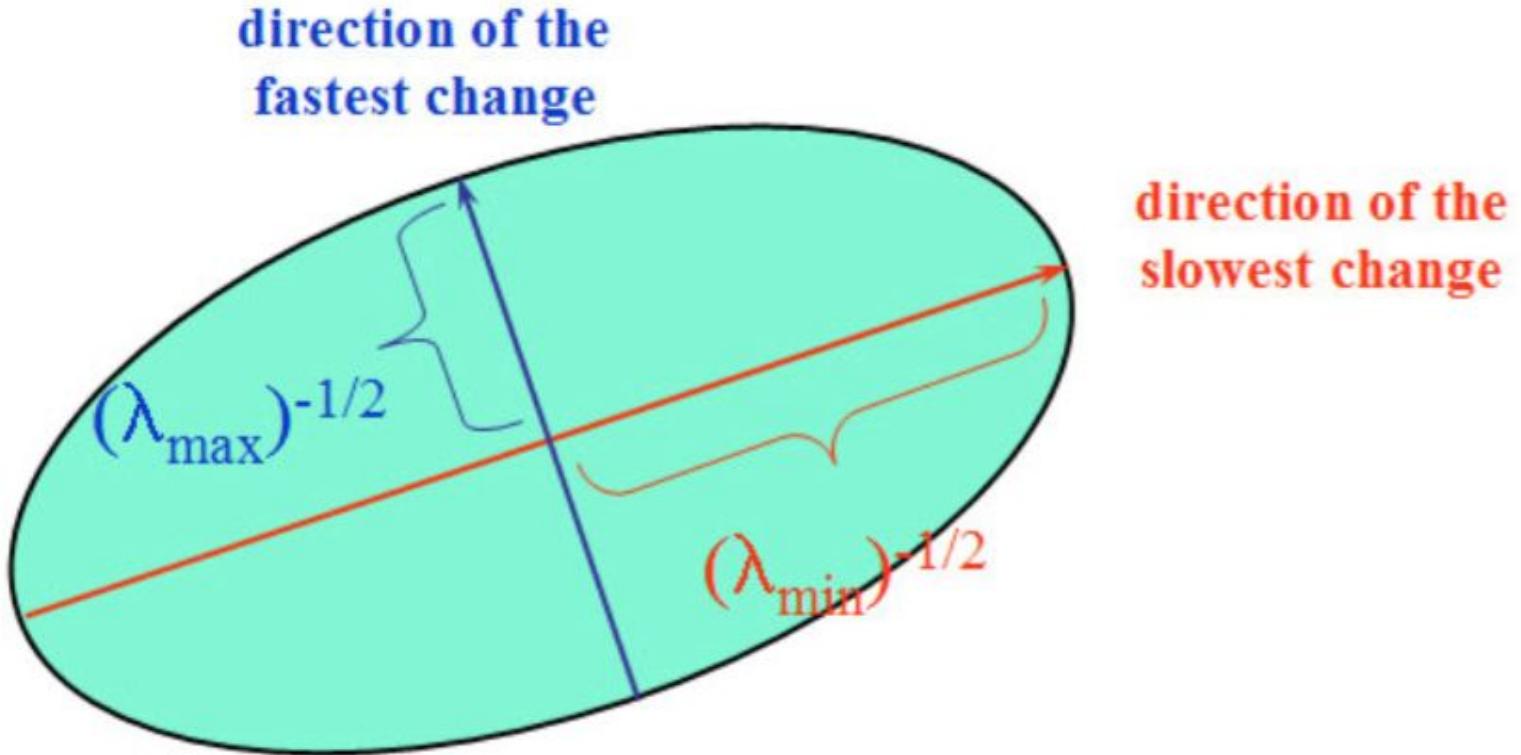
Матрица автокорреляции



Поиск характерной области на изображении

- необходима количественная мера для поиска области с характерной точкой
- большая часть мер основана на собственных чисел матрицы автокорреляции в анализируемой области
- тк область вокруг точки должна сильно варьироваться, то нас интересуют области с большими значениями собственных чисел матрицы автокорреляции

Собственные числа матрицы



Собственные числа матрицы

`cv2.eigen(src[, eigenvalues[, eigenvectors]])` → `eigenvalues`, `eigenvectors`

src – исходное изображение

eigenvalues - выходной массив собственных значений

eigenvectors – выходной массив собственных векторов

Поиск характерной точки на изображении

1. Строим автокорреляционную матрицу изображения
2. Определяем количественную меру качества области на основе собственных чисел матрицы автокорреляции
3. Оставляем области с локальным максимумом меры (Non-Maximum Suppression)
4. Полученные области содержат характерные точки

Вычисление дескрипторов характерных точек

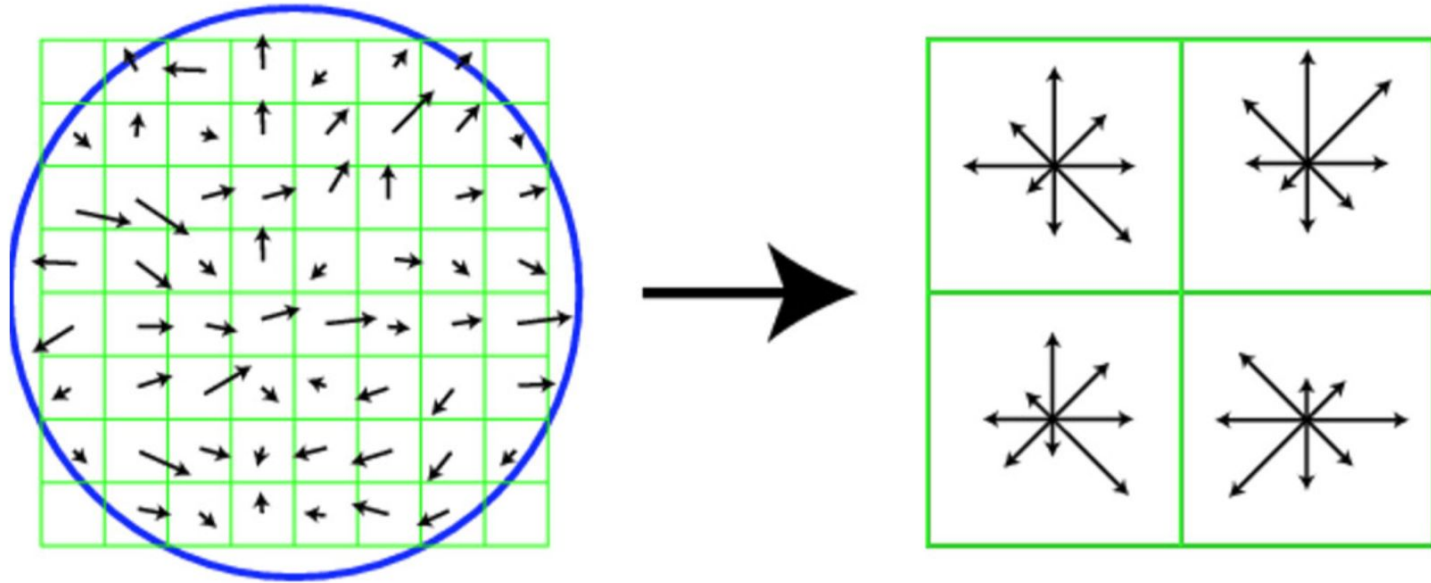
SIFT - Scale Invariant Feature Transform

- дескриптор основан на построении гистограммы градиентов (HOG)
- в окрестности характерной точки выделяется область размером 16x16 пикселей
- для каждого пикселя оценивается вектор градиента
- длина вектора градиента взвешивается гауссовским фильтром, таким образом, чтобы пиксели удаленные от характерной точки имели меньший вес

SIFT - Scale Invariant Feature Transform

- исходная область 16×16 разбивается на части размера 4×4
- для каждой части строится гистограмма градиентов с 8 ячейками
- в результате получается вектор из 128 признаков
- полученный вектор нормируется до единичной длины

SIFT - Scale Invariant Feature Transform

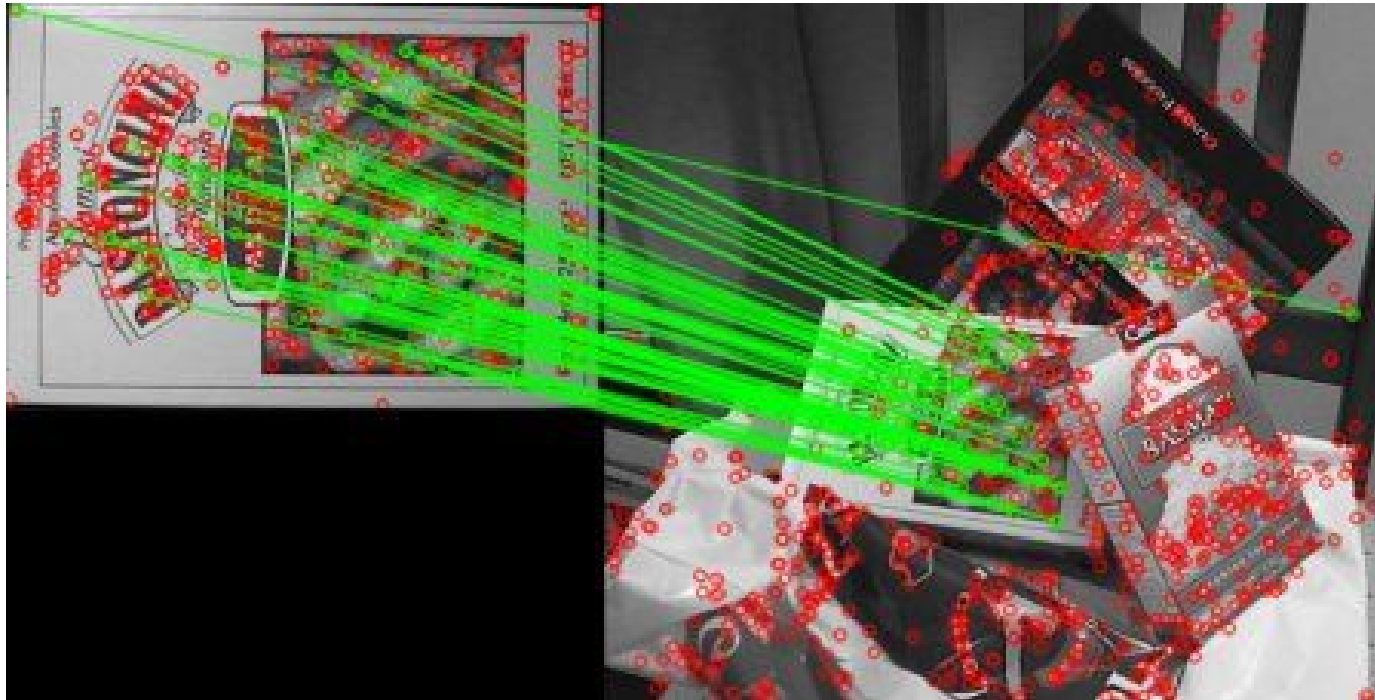


Матчинг характерных точек

Матчинг характерных точек

- выбрать меру расстояния для дескрипторов - [евклидова мера \(L2\)](#), [L1](#), [Hamming](#)
- попарное сравнение всех точек - полный перебор, долго
- индексация перед поиском и поиск по индексу
 - поиск точек в окрестности - [kdtree](#)
 - хеширование точек таким образом, чтобы точки с похожими дескрипторами оказывались рядом - [locality sensitive hashing](#)

Матчинг характерных точек



Поиск похожих изображений

Content Based Image Retrieval (CBIR)

Постановка задачи

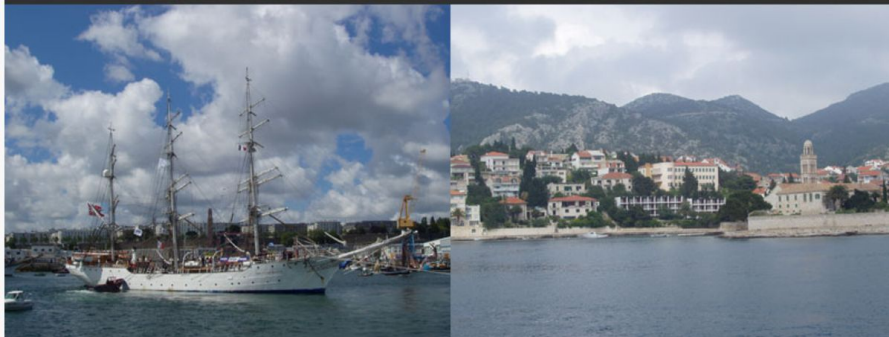
- картинки могут быть похожи по-разному
 - мета теги
 - время создания
 - цвет
 - изображения с похожими предметами

Постановка задачи

Query:



Results:



Постановка задачи

- необходимо определиться с критерием поиска
- от критерия поиска зависит способ выделения признаков изображения

Выделение признаков изображения

- нас интересуют признаки, которые описывают изображение в целом
 - гистограммы цветов в пространстве HSV
 - гистограммы градиентов

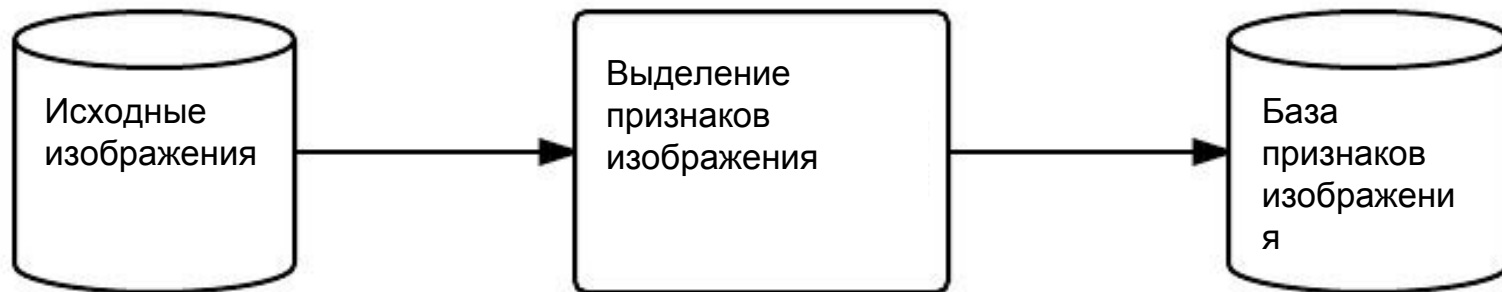
Выделение признаков изображения

- Признаки можно считать как по всему изображению
- Альтернативный способ: разбить изображение на области и посчитать признаки для каждой области отдельно, и затем объединить результат

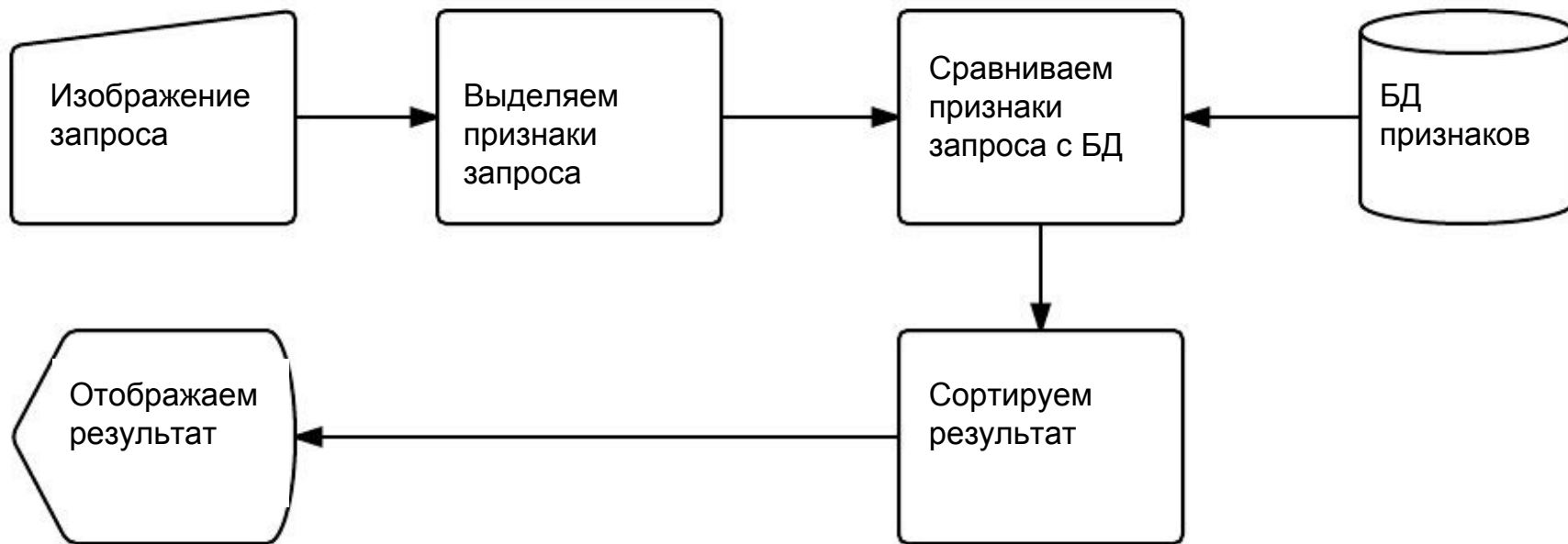
Выделение признаков изображения



Выделение признаков изображения



Архитектура



Индексация и поиск

- FLANN - Fast Library for Approximate Nearest Neighbors
<http://www.cs.ubc.ca/research/flann/>
- Faiss: A library for efficient similarity search
<https://github.com/facebookresearch/faiss/wiki/Getting-started-tutorial>
- FAst Lookups of Cosine and Other Nearest Neighbors
<https://pypi.python.org/pypi/FALCONN>

Резюме

- попарное сравнение пикселей не устойчиво к изменению цвета / освещенности и масштабу
- гистограммы цветов и градиентов более устойчивы к изменению цвета и поворотам и хорошо подходят для поиска визуально похожих изображений

Резюме

- для матчинга изображений используют характерные точки
- характерные точки выделяются большими значениями автокорреляционной матрицы
- в качестве дескриптора характерной точки можно использовать гистограмму градиентов
- для ускорения поиска одинаковых точек на изображениях используется [K-d tree](#)

Резюме

- дескрипторы для поиска изображения зависят от задачи
- одним из вариантов дескрипторов могут быть гистограммы цвета или градиента
- для ускорения поиска необходима индексация базы
- наиболее распространенный способ индексации - [Locality Sensitive Hashing](#)

Полезные материалы

- [Computer Vision: Algorithms and Applications \(Chapter 4\)](#)
- [OpenCV: Feature Detection and Description](#)
- [OpenCV-Python Tutorials](#)
- [Repository for OpenCV's extra modules](#)
- [Histogram of oriented gradients](#)
- [CBIR: Content-based image retrieval](#)
- [List of CBIR engines](#)

