Цифровая обработка изображений

Извлечение признаков и поиск

Сегодня мы научимся

- решать задачу распознавания изображений методом понижения размерности
- выделять признаки для поиска и анализа изображений
- определять характерные точки на изображении
- строить систему поиска изображений

План занятия

- Анализ главных компонент в задачах CV
- Представление свойств изображения с помощью гистограмм
  - Гистограммы цветов
  - Гистограммы градиентов

План занятия

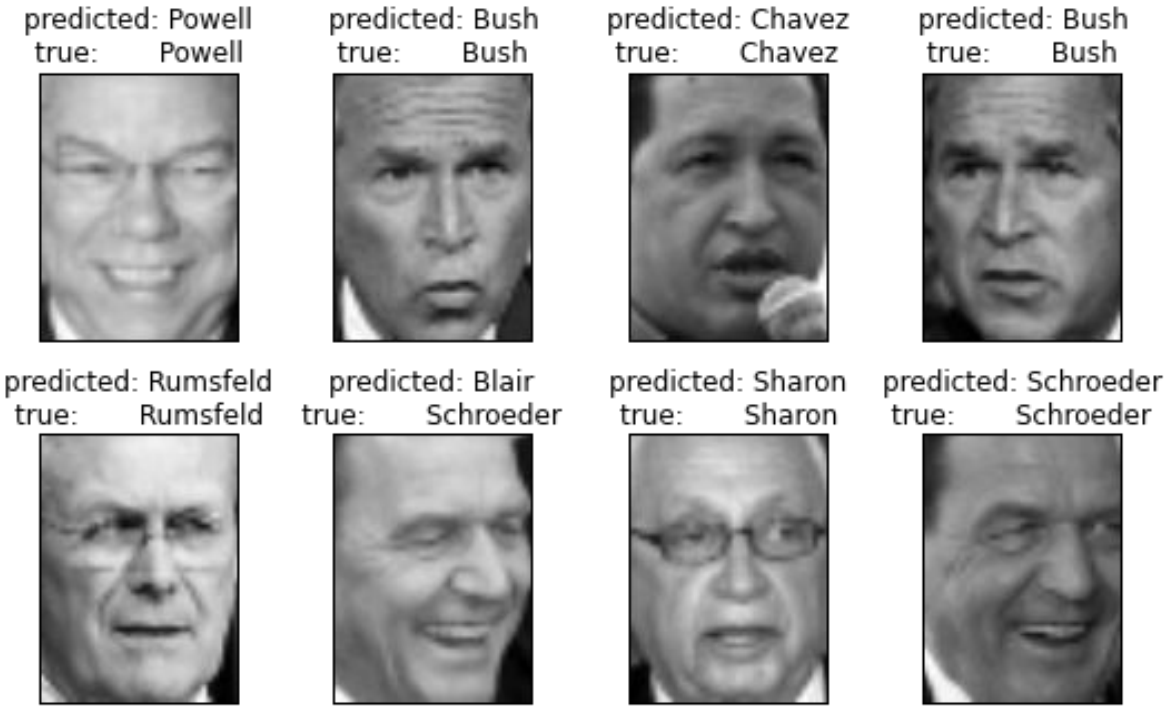
- Характерные точки
  - Поиск характерных областей на изображении
  - Выделение признаков (дескрипторов) характерных областей на изображении
  - Матчинг характерных точек на изображениях
  - Пример

План занятия

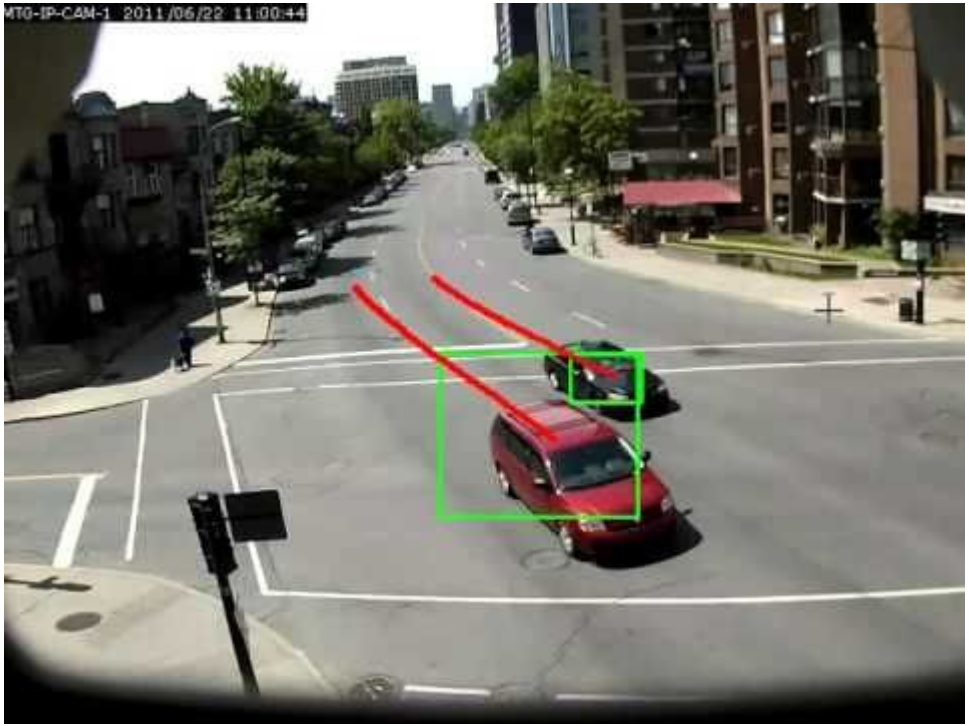
- Поиск изображений по контенту CBIR
  - выделение признаков и индексация
  - обзор архитектуры
  - поиск по индексу

Примеры задач компьютерного зрения

Распознавание лиц



Пример. Optical Flow



Пример. Поиск похожих изображений (CBIR)

Query Image

Retrieved Results

PCA - анализ главных компонент (eigenface)

PCA - анализ главных компонент

- изображение можно представить в виде вектора длины HxW
- большая размерность данных (число пикселей) затрудняет их обработку
- для сокращения размерности применяется метод PCA

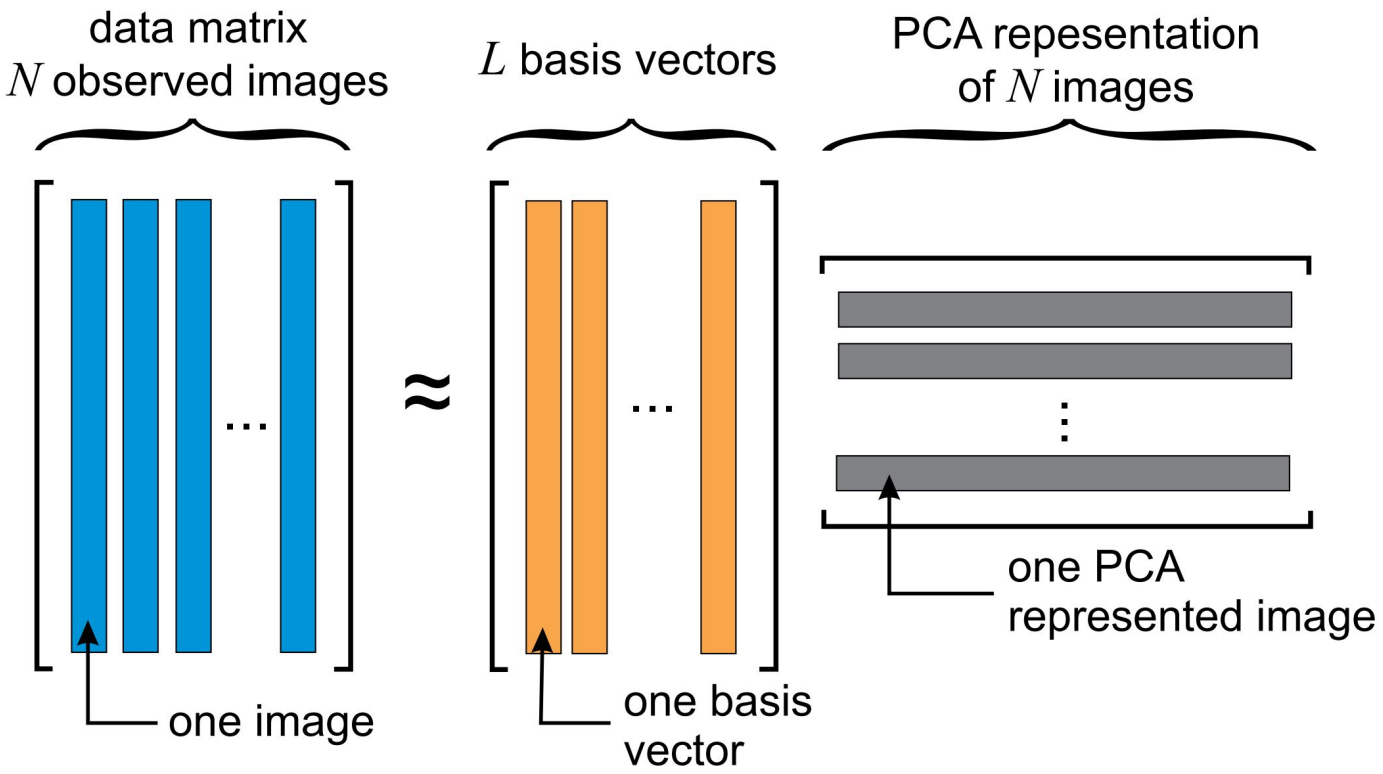
PCA - анализ главных компонент

- в результате преобразование PCA получаем представления изображений в базисе меньшей размерности
- полученное сжатое представление можно использовать для распознавания изображений

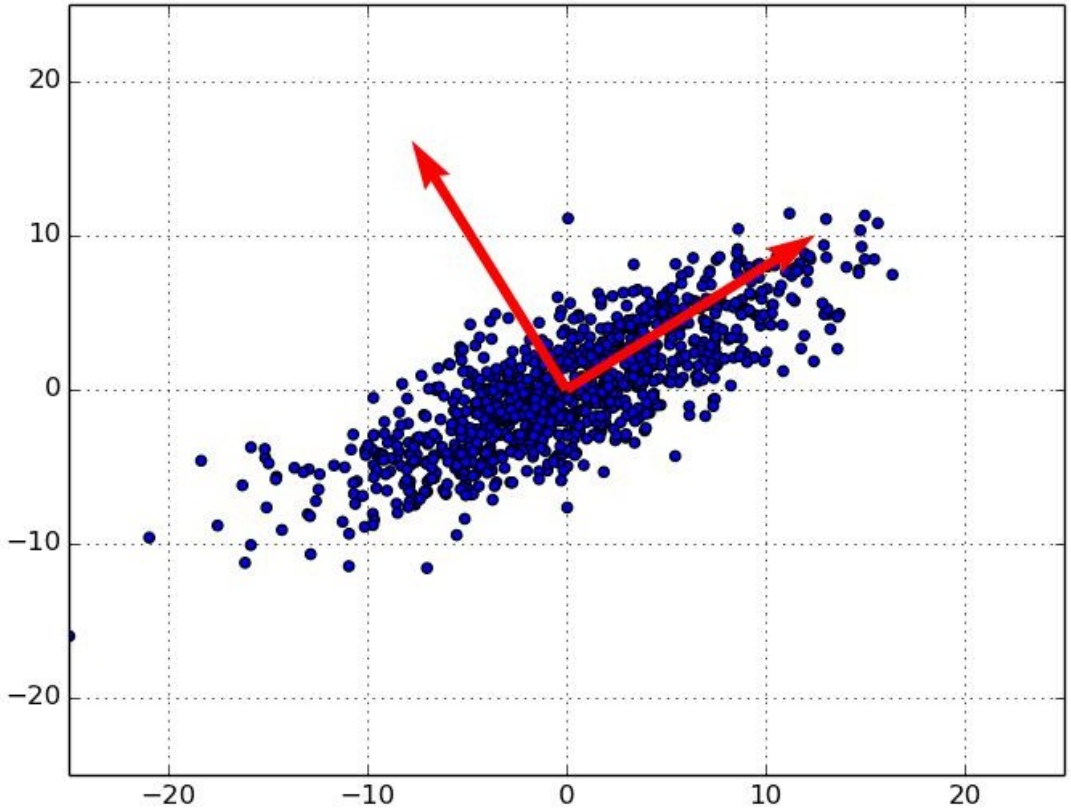
PCA - анализ главных компонент

PCA - анализ главных компонент

PCA - анализ главных компонент



PCA - анализ главных компонент

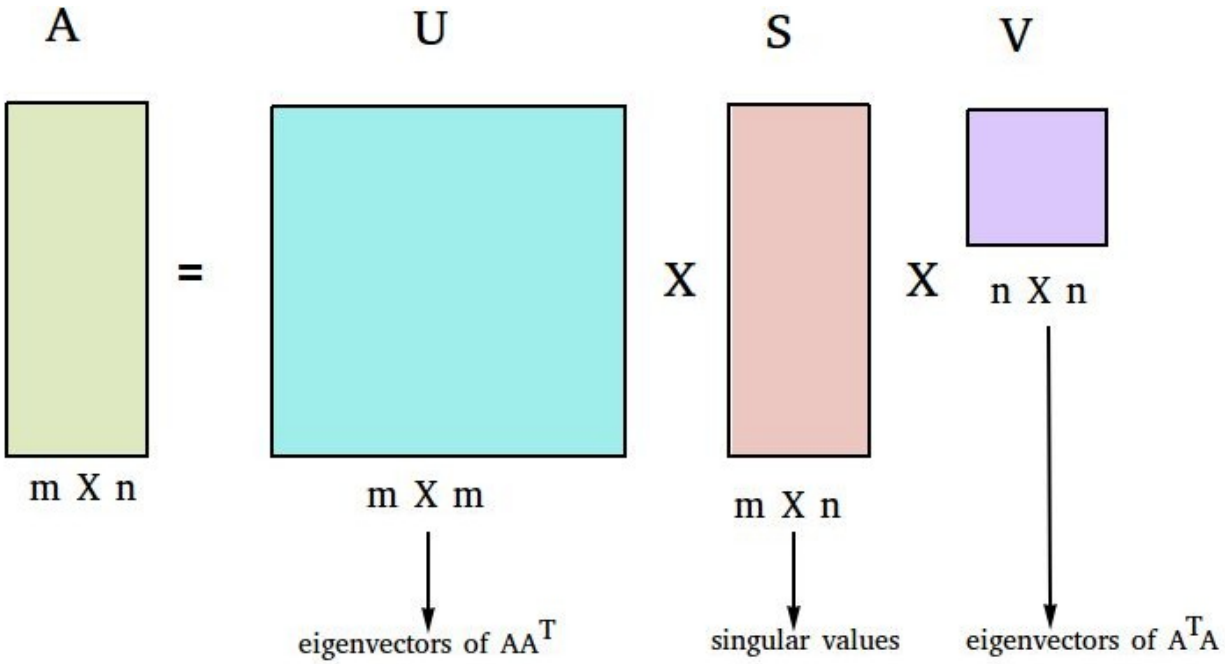


PCA - матрица ковариации

$$\mathrm{Cov}(A) = \begin{bmatrix} \frac{\sum (x_i - \bar{X})(x_i - \bar{X})}{N} & \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{N} \\ \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{N} & \frac{\sum (y_i - \bar{Y})(y_i - \bar{Y})}{N} \end{bmatrix}$$

$$= \begin{bmatrix} \mathrm{Cov}(X,X) & \mathrm{Cov}(Y,X) \\ \mathrm{Cov}(X,Y) & \mathrm{Cov}(X,Y) \end{bmatrix}$$

PCA - разложение SVD



PCA - последовательность вычислений

- подготавливаем данные, представляем изображения в виде векторов длиной HxW
- вычитаем среднее значение из каждой компоненты вектора
- получаем собственные вектора в результате SVD разложения ковариационной матрицы изображений
- выбираем размерность (число собственных векторов) на основе собственных значений

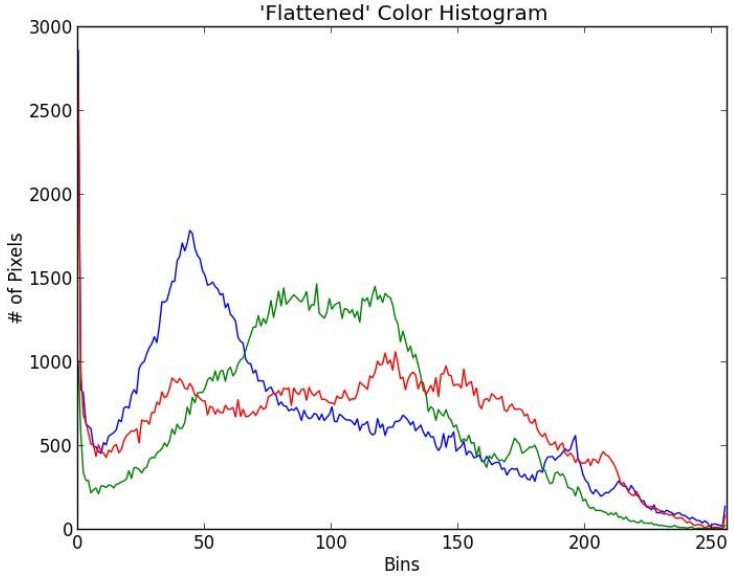
Выделение признаков изображения

Гистограммы признаков изображения

- представляют собой обобщенное описание изображения
- как правило гистограммы инвариантны к масштабу и повороту изображений
- позволяют сравнивать изображения и находить похожие

Гистограммы цветов

Гистограммы цветов



Гистограммы цветов

- разбиваем диапазон значений цвета (0..255) на фиксированное число ячеек (bins), например с шагом 1
- задаем в каких срезах (каналах) необходимо построить гистограмму
- 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

cv2.**calcHist**(images, channels, mask, histSize, ranges) → hist

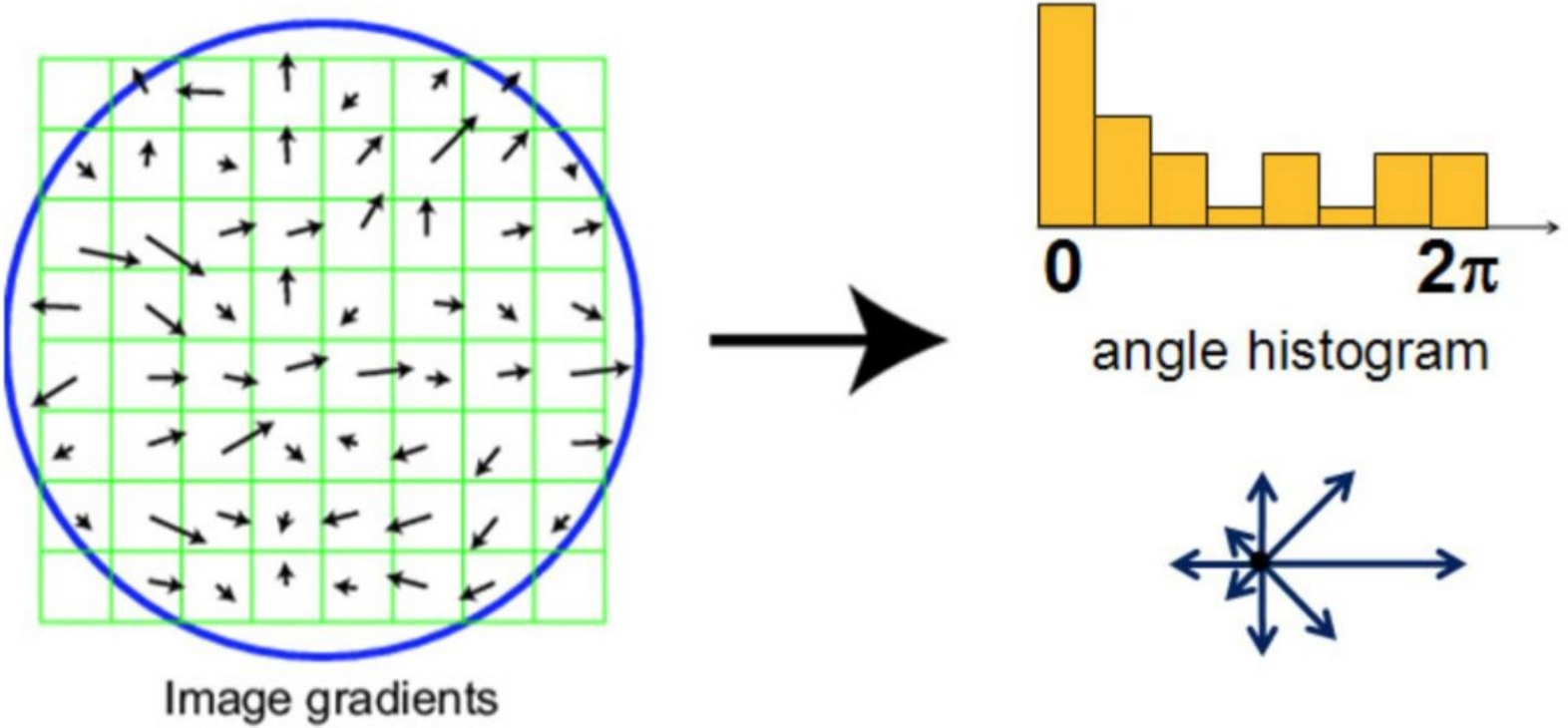
images – набор входных изображений для оценки гистограммы
channels – каналы по которым оцениваются гистограммы **mask** – маска ограничивает область оценки гистограммы **histSize** – массив размеров гистограмм по каждому измерению **ranges** – диапазоны значений каждого измерения

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)

Гистограммы градиентов (HOG)



Гистограммы градиентов (HOG)

- в каждой точке оцениваем составляющие градиента по осям x и y
- определяем направление и длину вектора градиента
- оцениваем гистограмму градиентов
- 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)

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

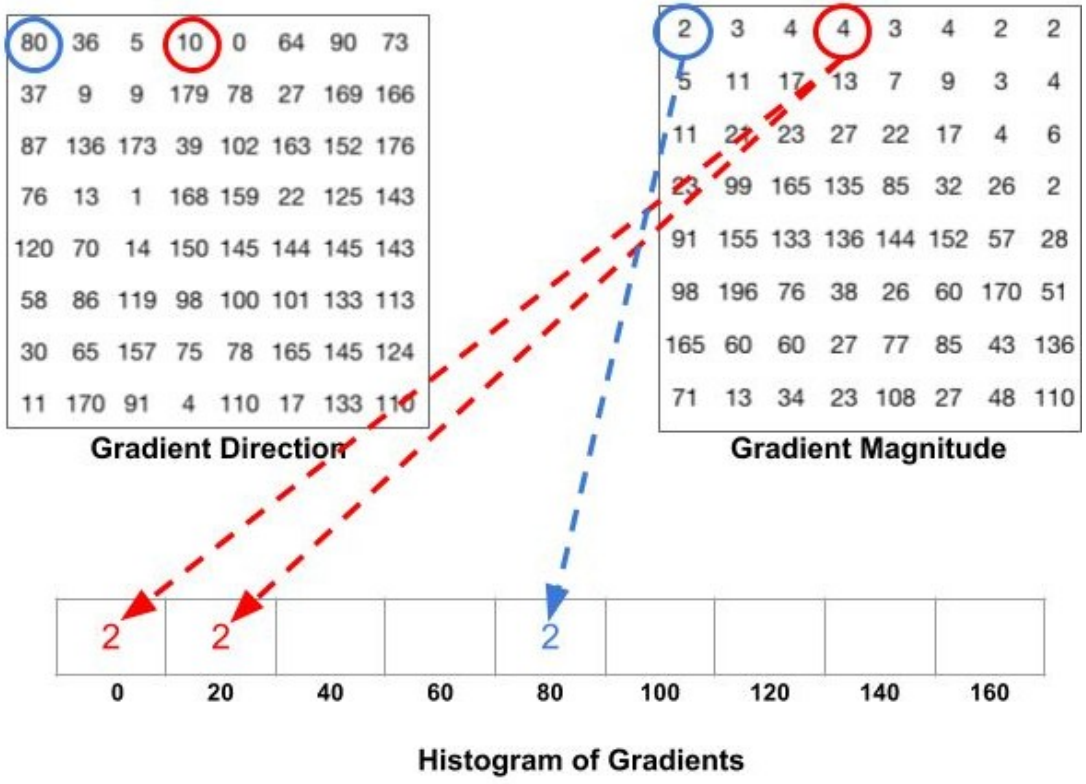
$$\theta = arctan\frac{g_y}{g_x}$$

g, g_x, g_y - длина вектора градиента и его составляющих
theta - 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
- устойчивы к изменению масштаба

Характерные точки

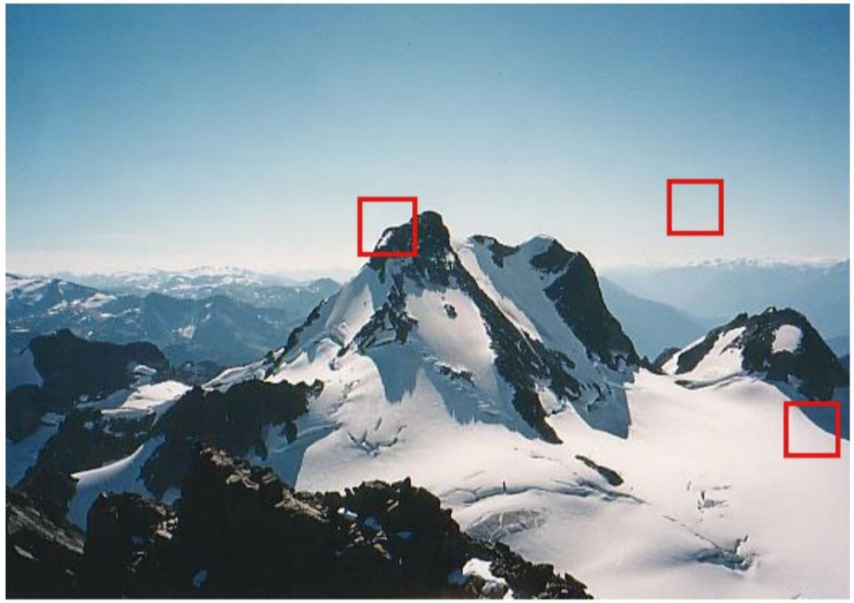
Характерные точки

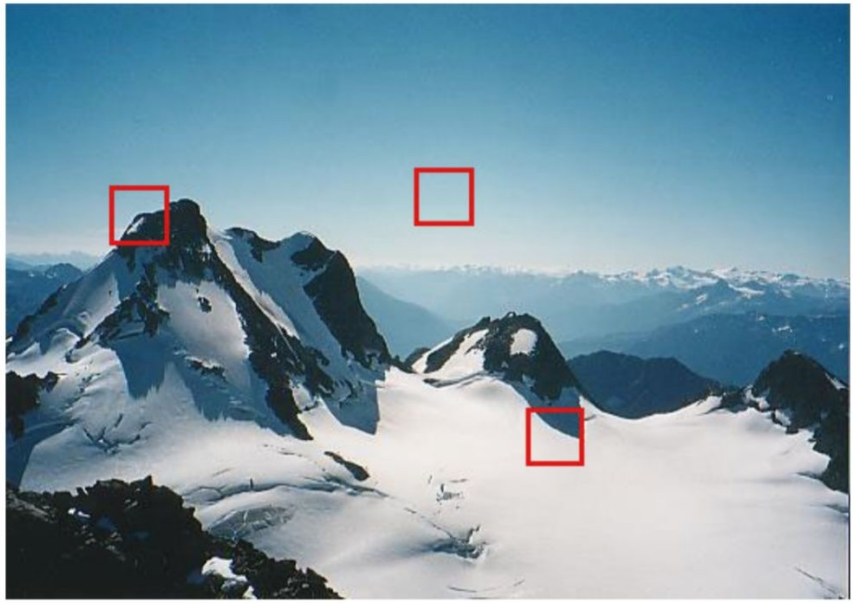
- позволяют находить одинаковые области (предметы) на разных изображениях
- используются для склейки панорам и составления карт по спутниковым снимкам

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

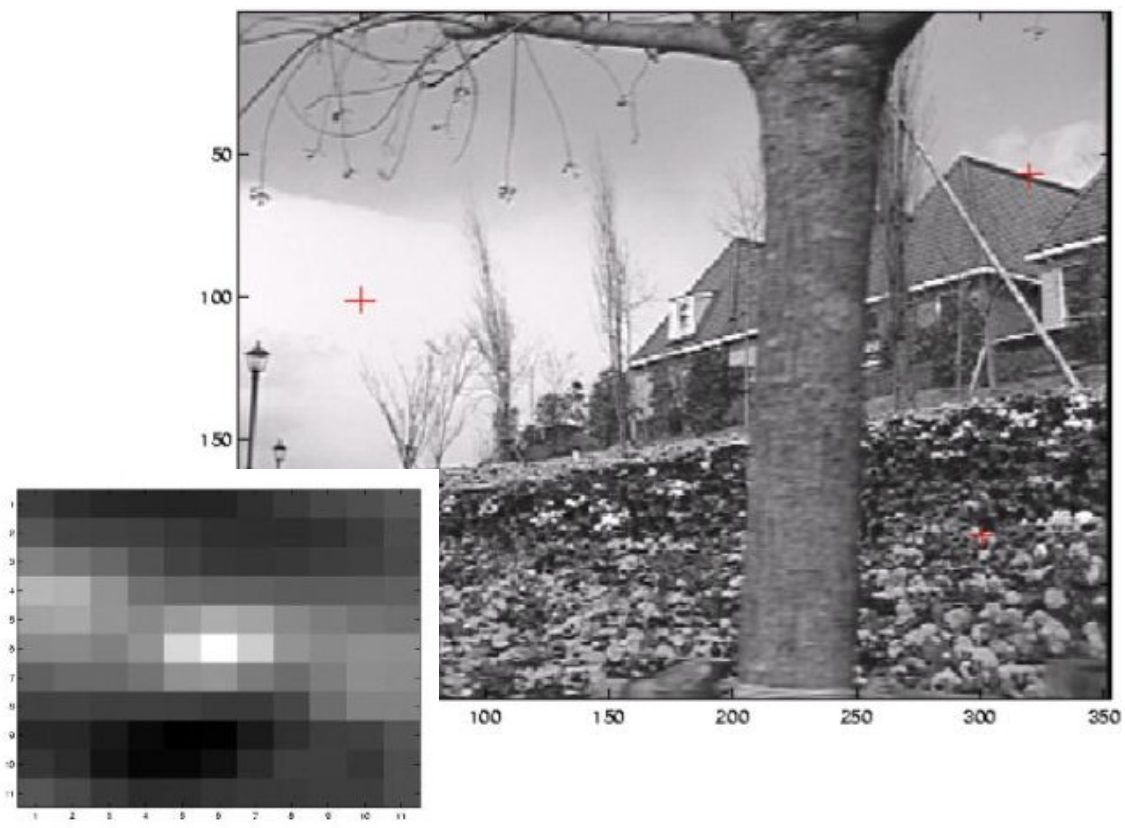
- Как понять что выбранная область содержит характерную точку?
- Область вокруг характерной точки должна сильно варьироваться
- В области характерной точки небольшой сдвиг изображения должен приводить к существенному различию по сравнению с исходным изображением

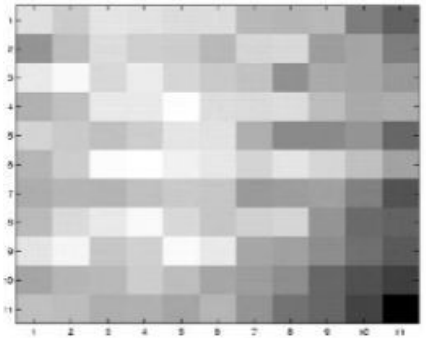
Автокорреляция

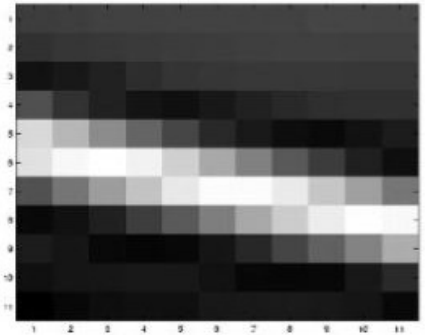
$$E_{\mathrm{AC}}(\Delta \boldsymbol{u}) = \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2$$

du - вектор смещения по осям *x* и *y*
x_i - вектор координат пикселя изображения
w - окно или фильтр (гауссовский)
I_o - исходное изображение

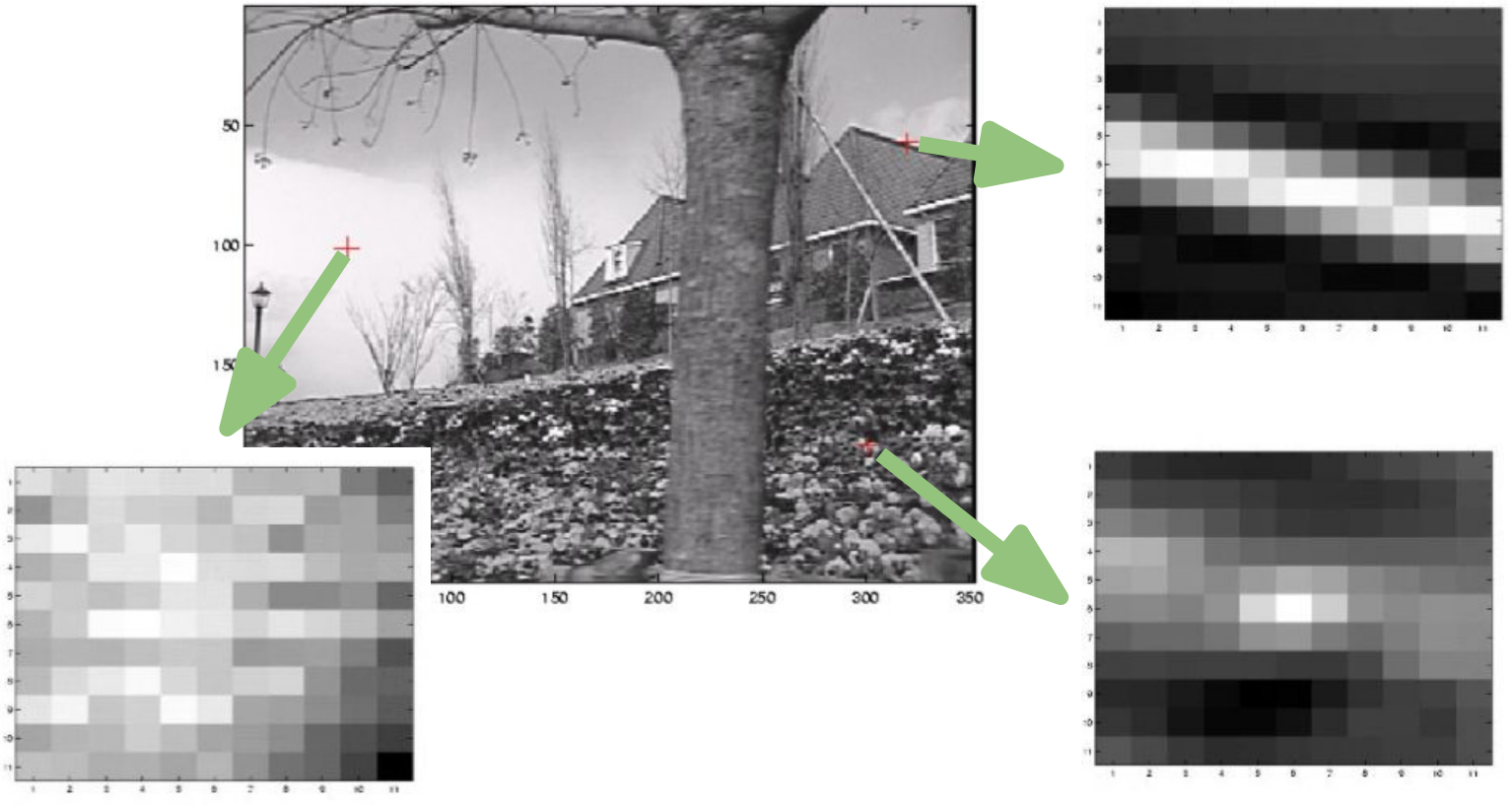
Матрица автокорреляции



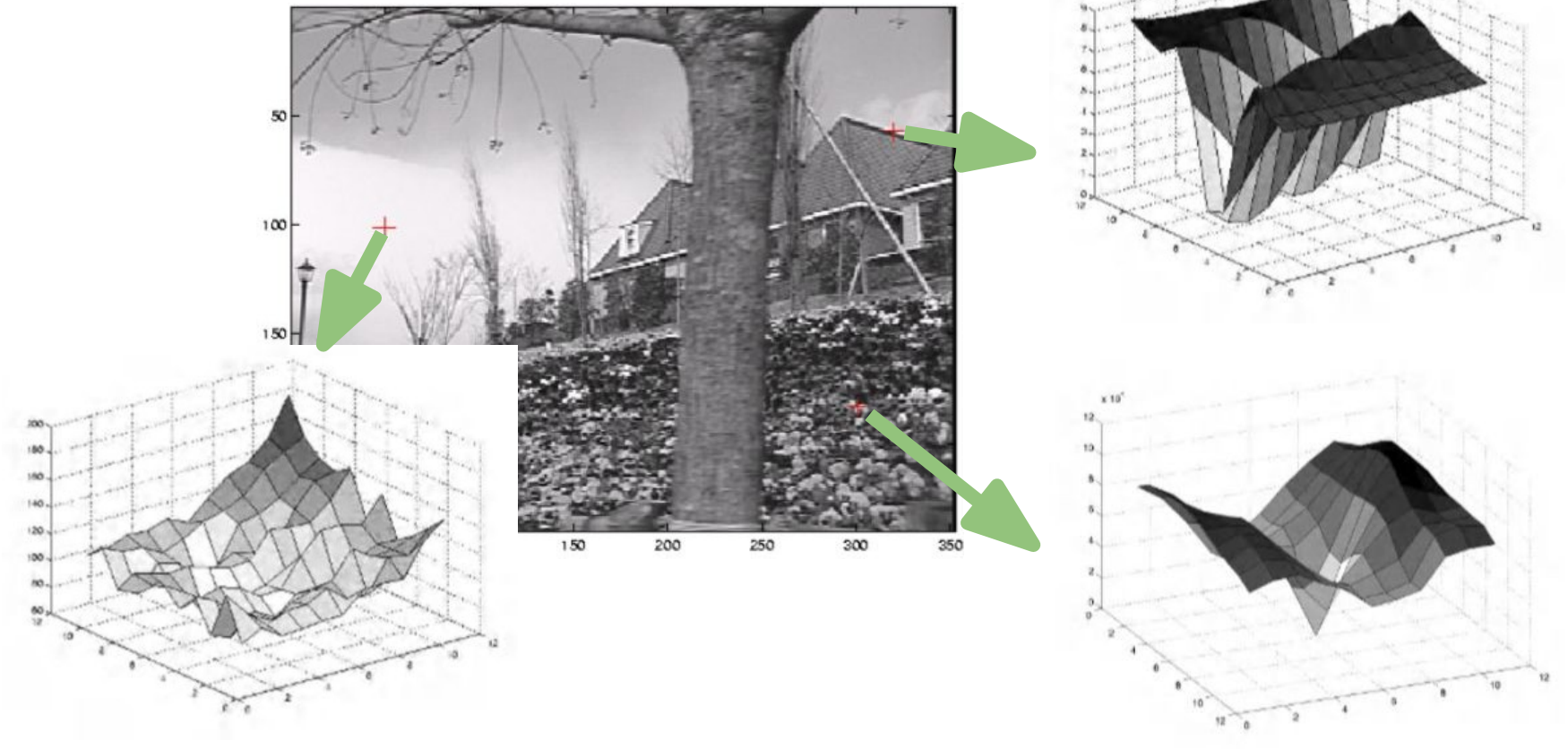




Матрица автокорреляции



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

Поиск характерной точки на изображении

1. строим автокорреляционную матрицу изображения
2. для каждой точки изображения вычисляем собственные числа в соответствующей окрестности матрицы автокорреляции
3. оставляем точки с локальным максимумом меры (Non-Maximum Suppression)
4. полученные области содержат характерные точки

Вычисление дескрипторов характерных точек

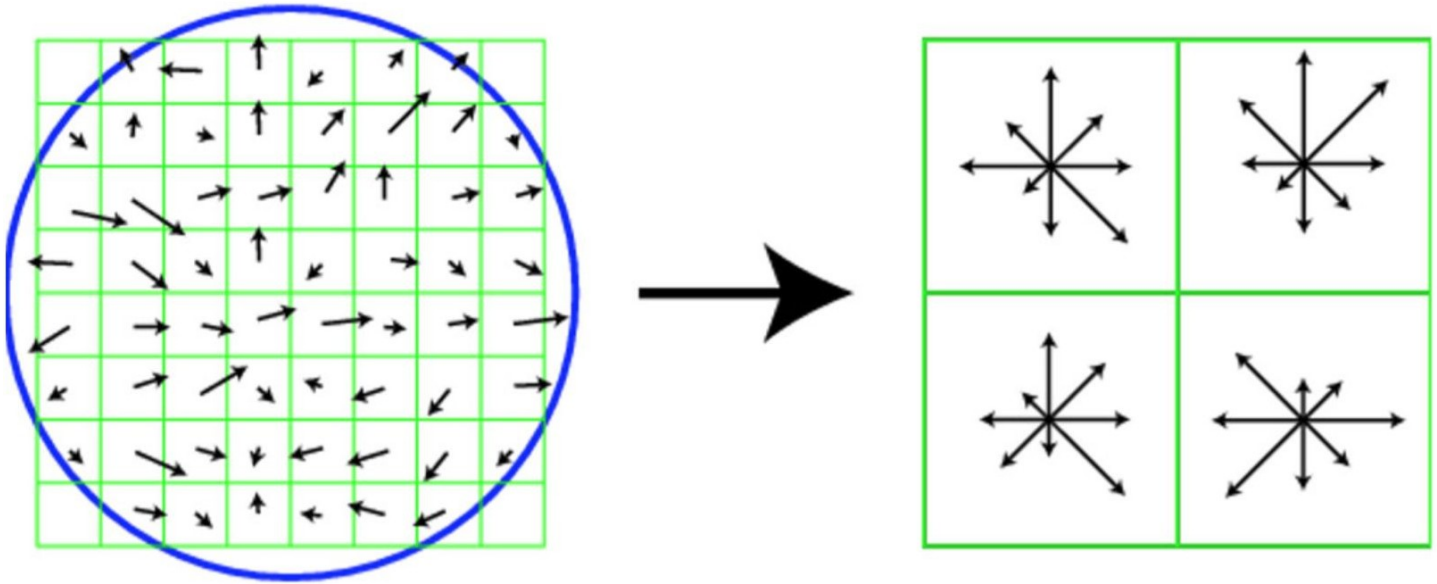
SIFT - Scale Invariant Feature Transform

- 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 гуассовским фильтром, таким образом, чтобы пиксели удаленные от характерной точки имели меньший вес

SIFT - Scale Invariant Feature Transform

- исходная область 16x16 разбивается на части размера 4x4
- 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

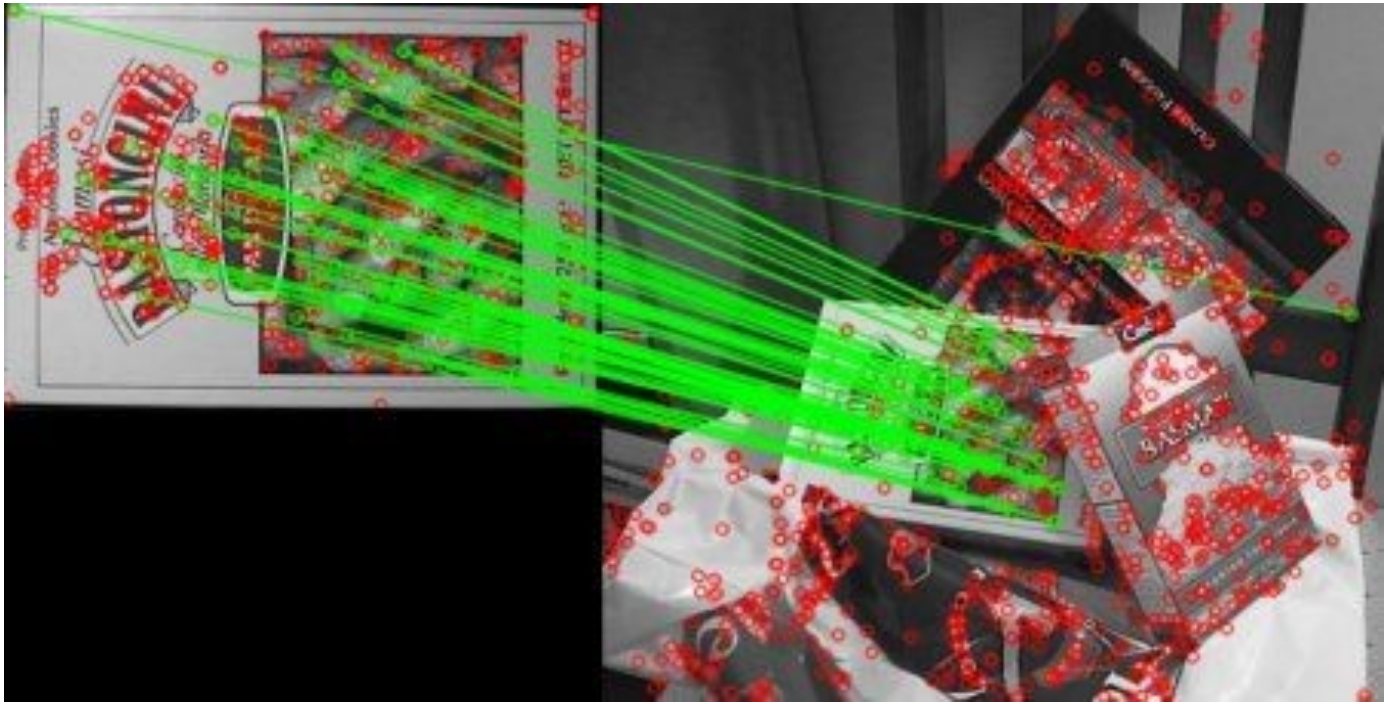


Матчинг характерных точек

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



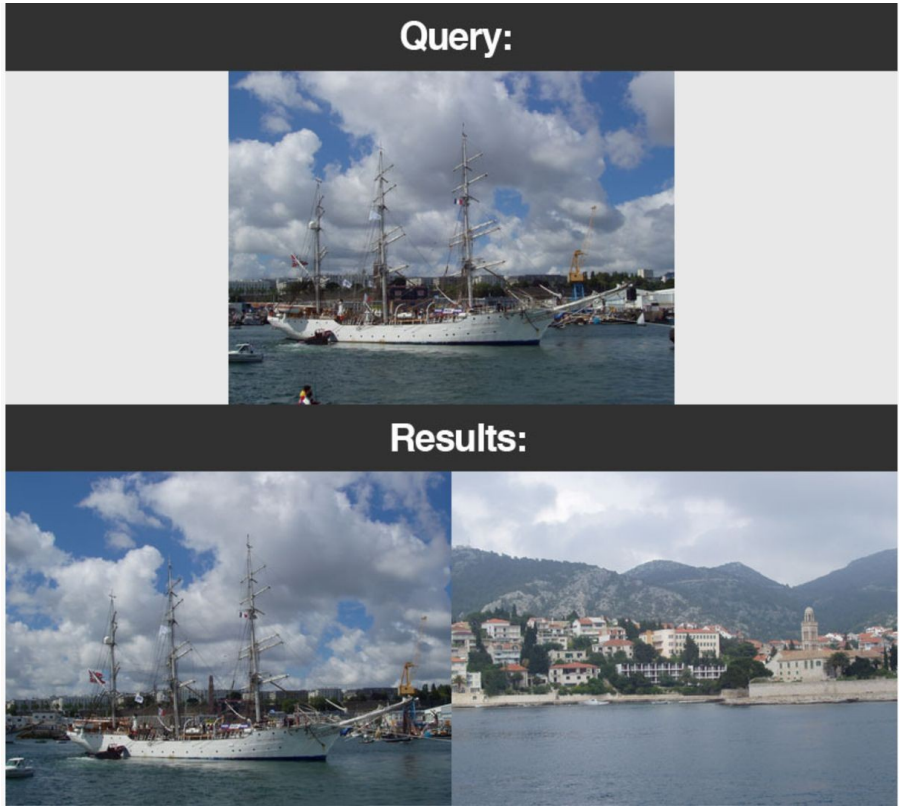
Поиск похожих изображений

Content Based Image Retrieval (CBIR)

Постановка задачи

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

Постановка задачи



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

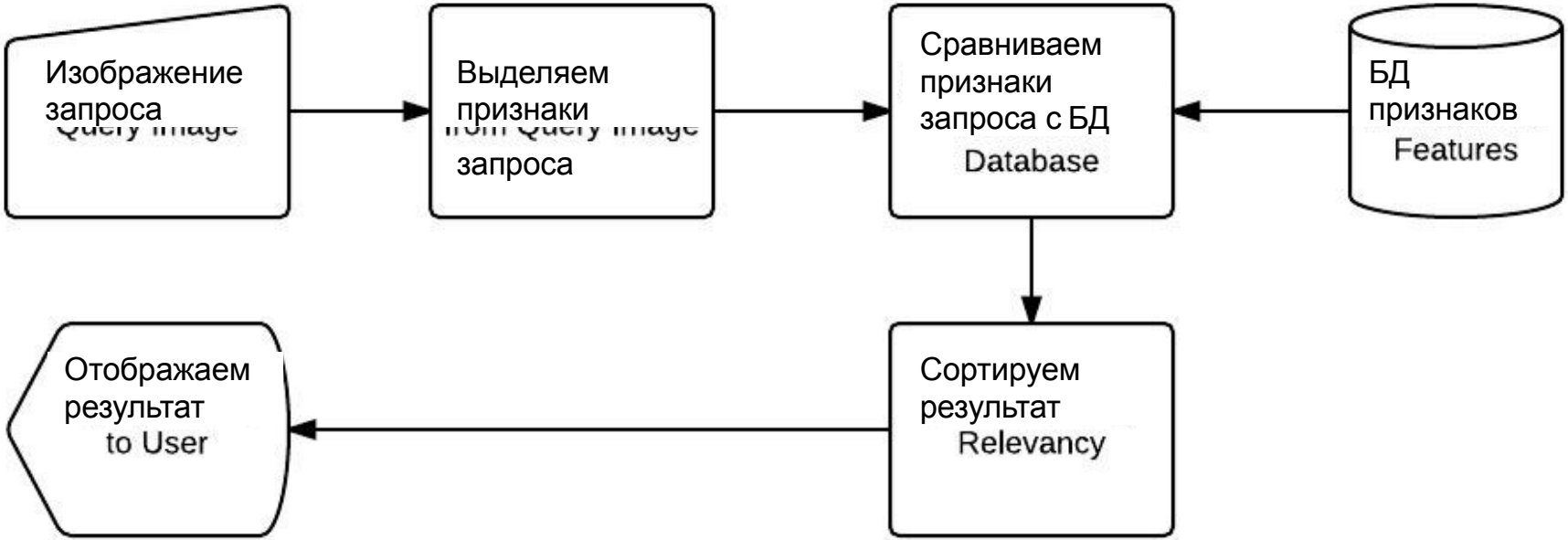
Выделение признаков изображения

- 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

- в результате PCA преобразования можно получить сжатое представление изображения, это представление удобно использовать для распознавания
- 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

- Eigenface
- 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