

Лекция 2

Извлечение признаков и особые точки

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

predicted: Powell
true: Powell



predicted: Bush
true: Bush



predicted: Chavez
true: Chavez



predicted: Bush
true: Bush



predicted: Rumsfeld
true: Rumsfeld



predicted: Blair
true: Schroeder



predicted: Sharon
true: Sharon



predicted: Schroeder
true: Schroeder



Пример. Поиск похожих изображений (CBIR)

Query Image



Retrieved Results



РСА - анализ главных компонент (eigenface)

РСА - анализ главных компонент

- изображение можно представить в виде вектора длины $H \times W$
- большая размерность данных (число пикселей) затрудняет их обработку
- для сокращения размерности применяется метод РСА

РСА - анализ главных компонент

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

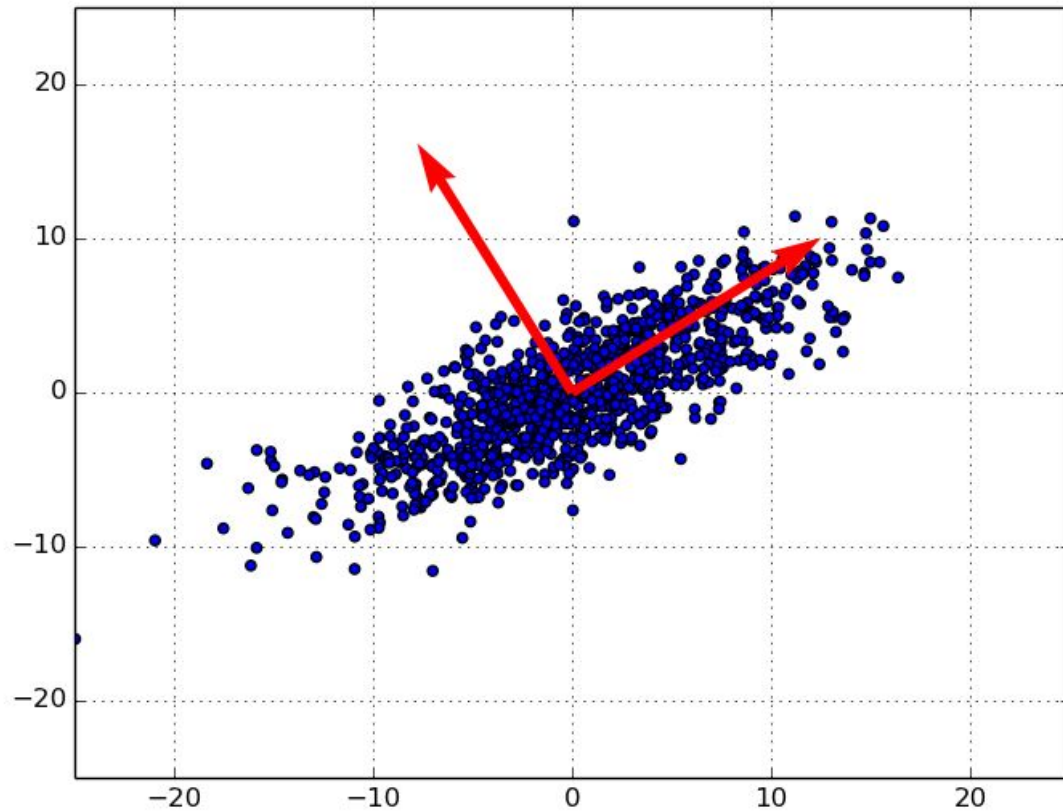
РСА - анализ главных компонент



РСА - анализ главных компонент



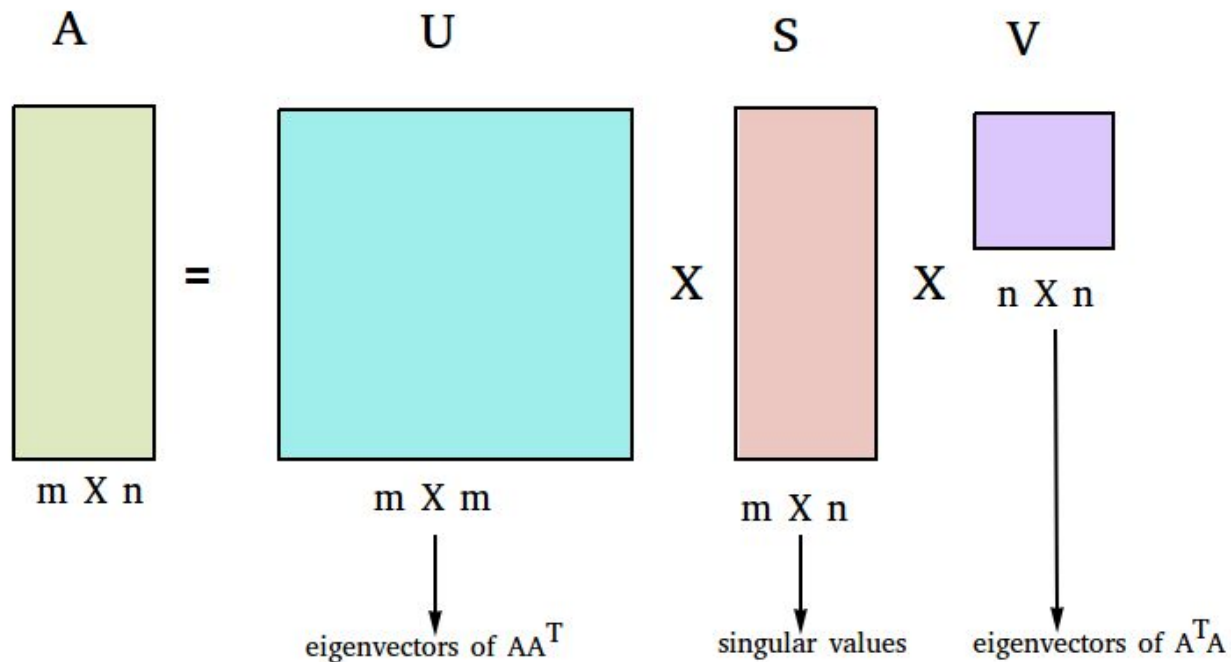
РСА - анализ главных компонент



РСА - матрица ковариации

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

PCA - разложение SVD



РСА - последовательность вычислений

- подготавливаем данные, представляем изображения в виде векторов длиной $H \times W$
- 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

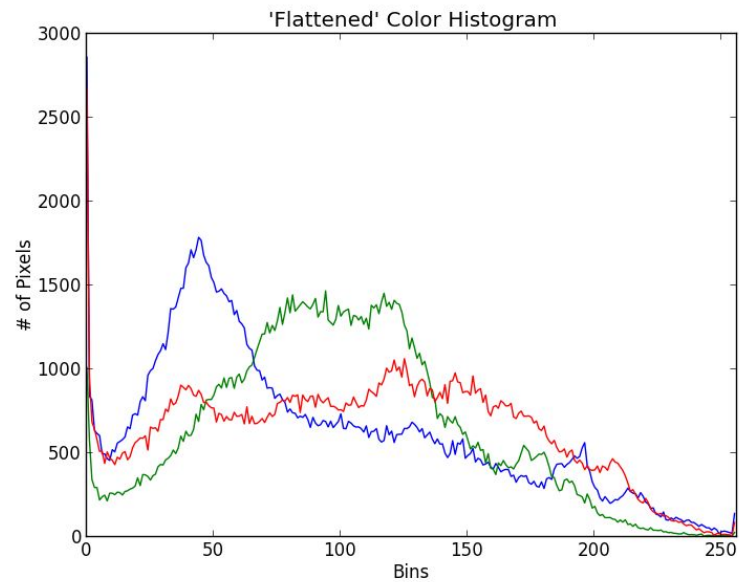
Выделение признаков изображения

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)

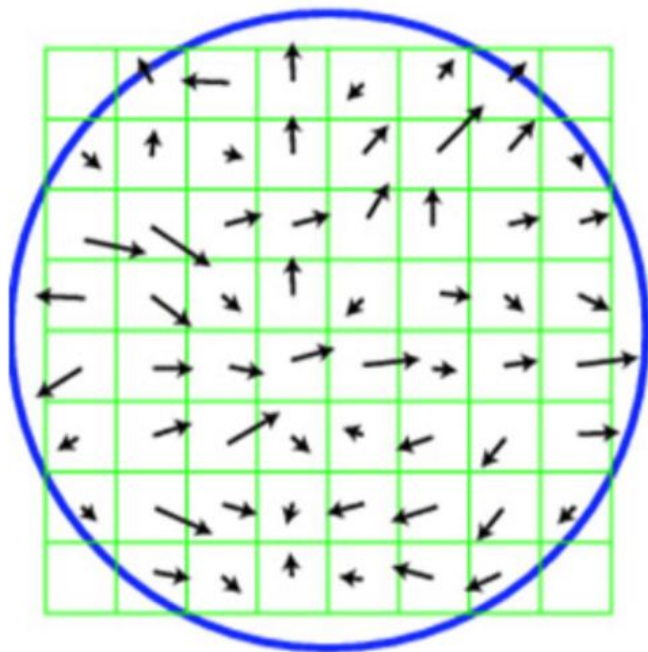
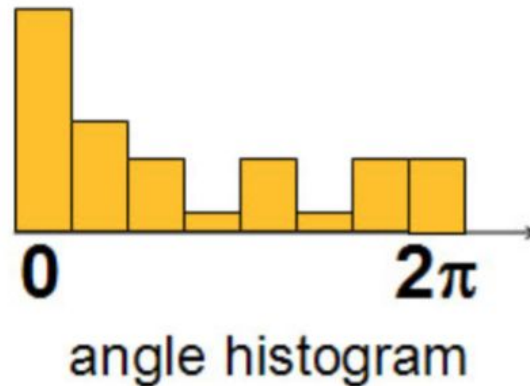


Image gradients



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

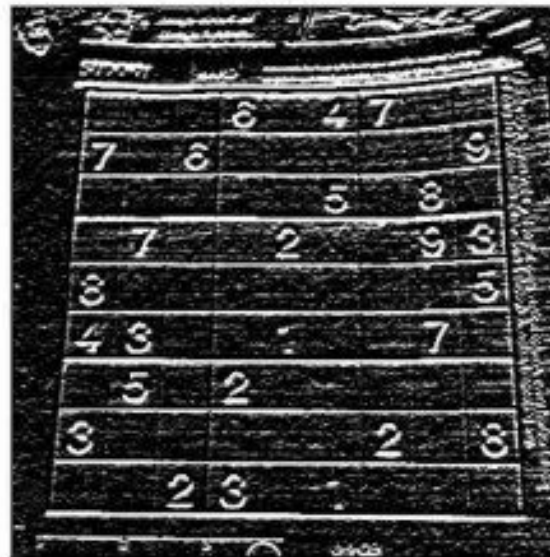
Original



Sobel X



Sobel Y



Гистограммы градиентов (HOG)

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

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

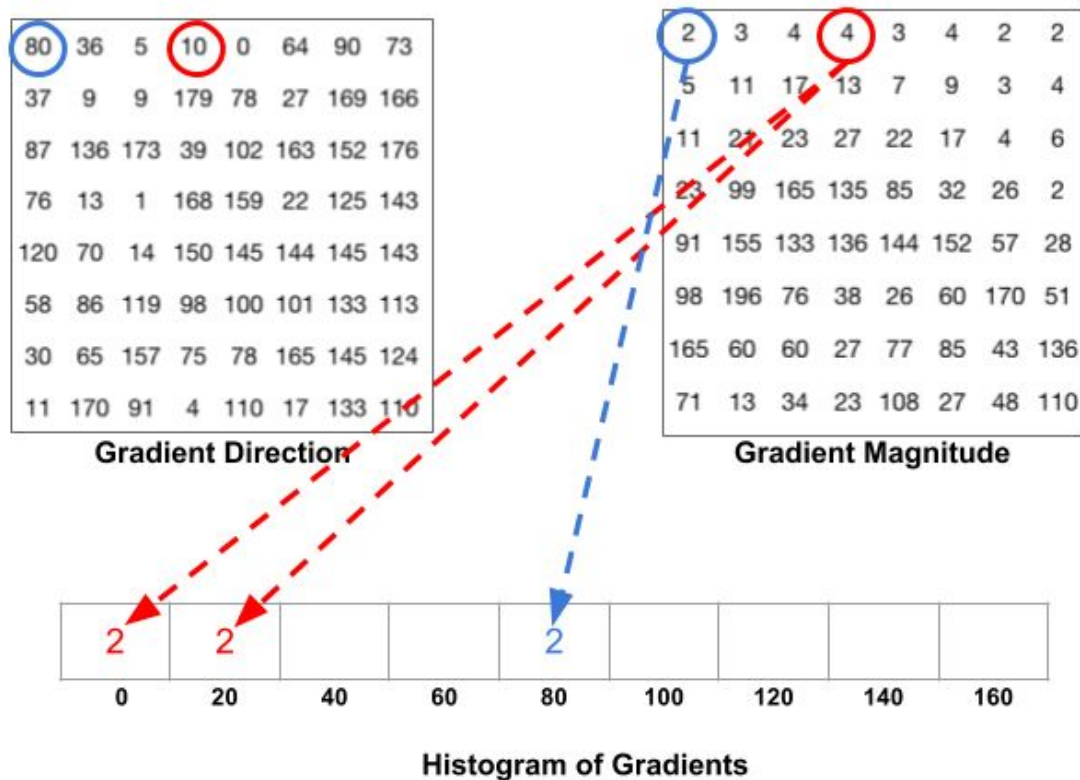
g , g_x , g_y - длина вектора градиента и его составляющих

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)

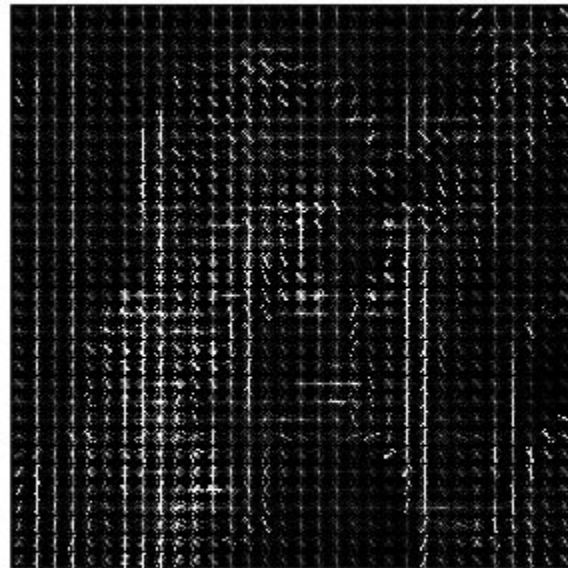


Гистограммы градиентов (HOG)

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
- устойчивы к изменению масштаба

Характерные точки

Вводный пример

Где медведь?



Задача 1



Задача 2

Вводный пример

Где медведь?



Задача 1



Задача 2

- 1) Почему во втором случае было проще найти?
- 2) Были видны “характерные” фрагменты медведя

Особенности (features)



- Хорошо различимые “фрагменты” объекта
 - особенности (features)
 - характеристические точки (characteristic points)
 - ключевые точки (keypoints)
 - Локальные особые точки (local feature points)
- Характерные фрагменты позволяют справиться с изменениями ракурса масштаба и перекрытиями

Требования



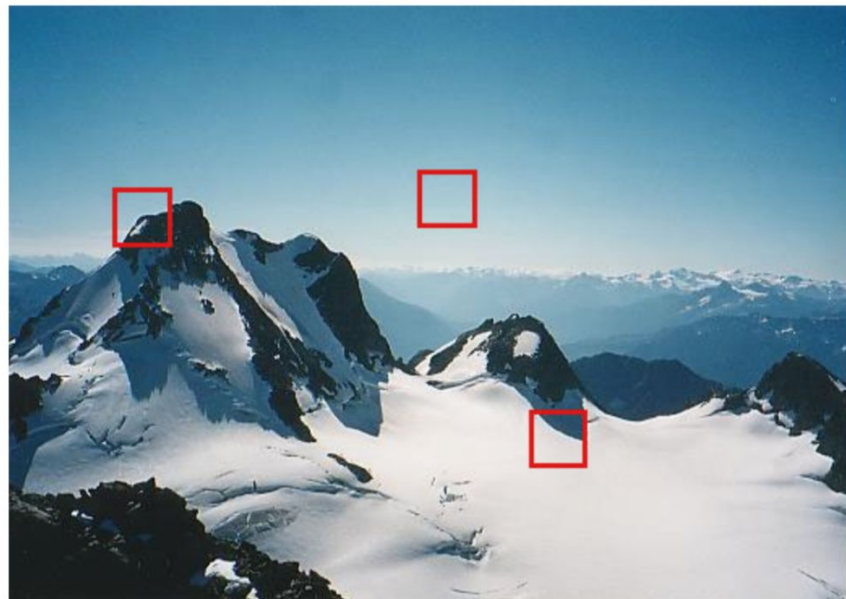
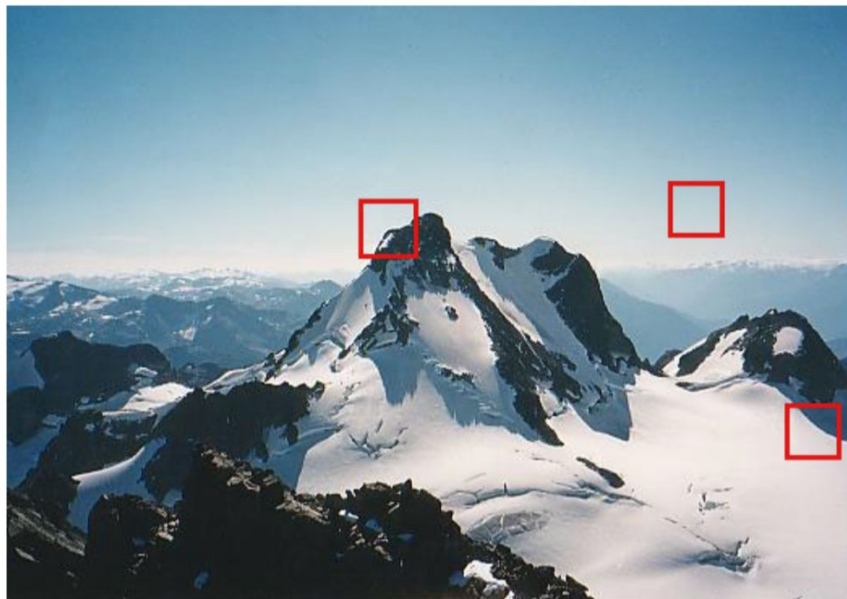
- Как можно сформулировать требования к хорошо различимым фрагментам объекта?

Требования



- Как можно сформулировать требования к хорошо различимым фрагментам объекта?
- Отличаются от большинства других фрагментов объекта
- Инвариантны к изменению освещения
- Инвариантны к изменению ракурса
 - Можно находить одну и ту же точку на измененных изображениях
 - Можем “идентифицировать эту точку”

Характерные точки



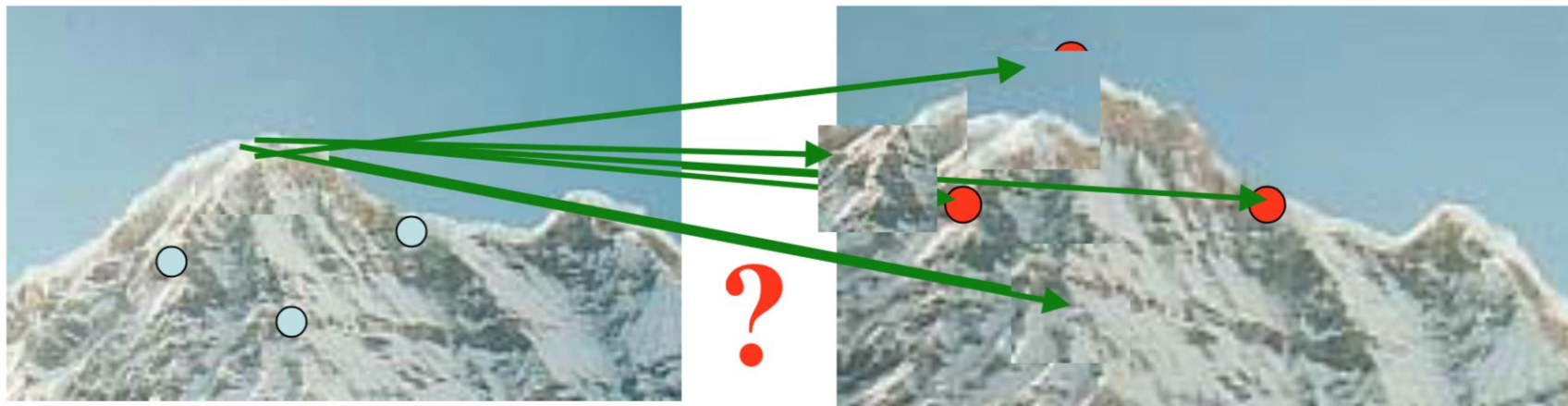
Требования

- Локальность (locality)
 - Особенность занимает маленькую область изображения, поэтому работа с ней не чувствительна к перекрытиям
- Повторимость (repeatability)
 - Особенность находится в том же месте объекта не смотря на изменения масштаба, положения, ракурса и освещения
- Значимость (saliency)
 - Каждая особенность имеет уникальное (distinctive) описание
- Компактность и эффективность
 - Количество особенностей существенно меньше числа пикселей изображения

Характерные точки

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

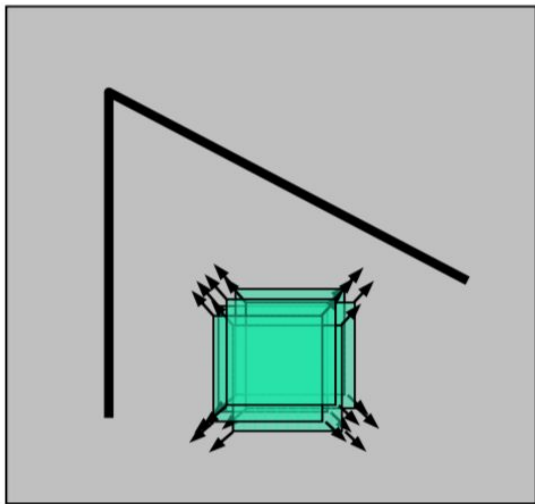
Повторимость



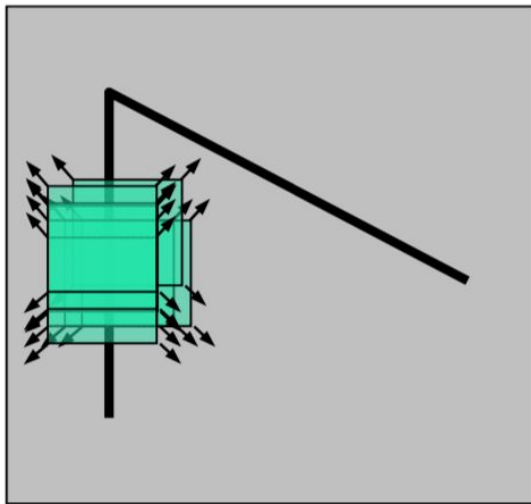
Поиск характерной точки на изображении

- 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

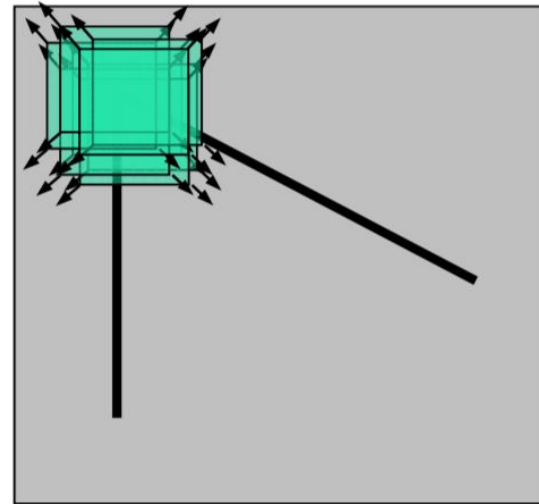
Локальные особенности



Монотонный регион
В любом направлении изменений нет



Край
Вдоль края изменения нет



Уголок
Изменения при перемещении в любую сторону

Детектор Харриса

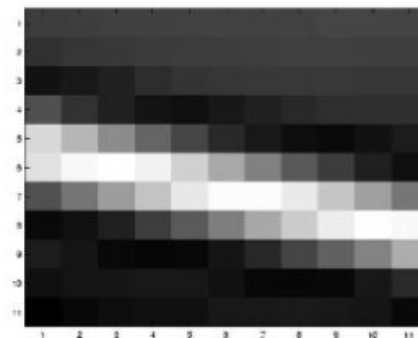
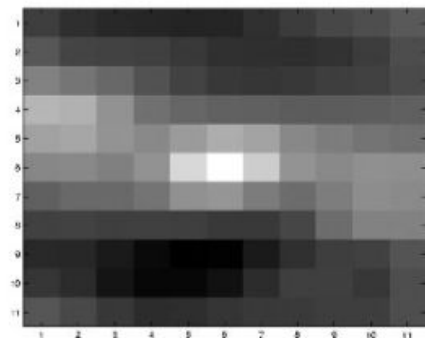
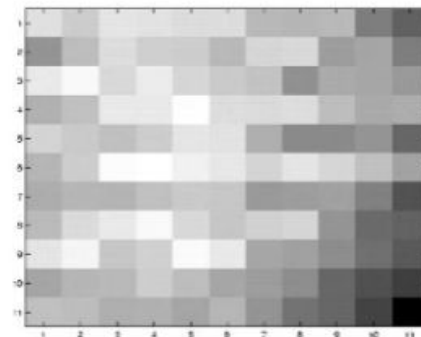
- Наиболее популярный детектор локальных особенностей точек - детектор Харриса (Harris)
- Ищет такие точки (x, y) , окрестность которых меняется при любом сдвиге $(x+u, y+v)$
- Такие точки часто оказываются углами, поэтому метод еще называют “детектор углов”

Автокорреляция

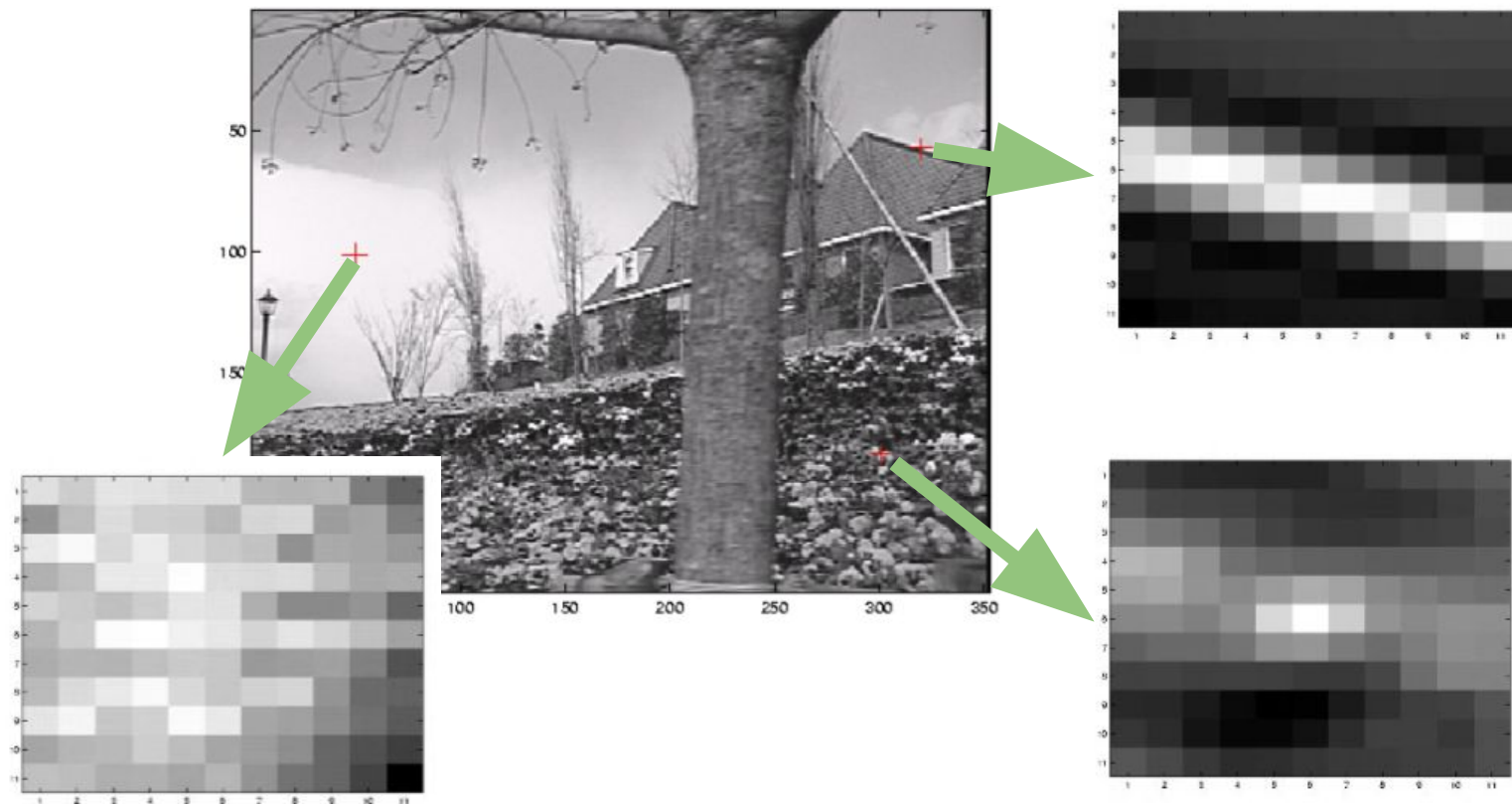
$$E_{AC}(\Delta \mathbf{u}) = \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

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

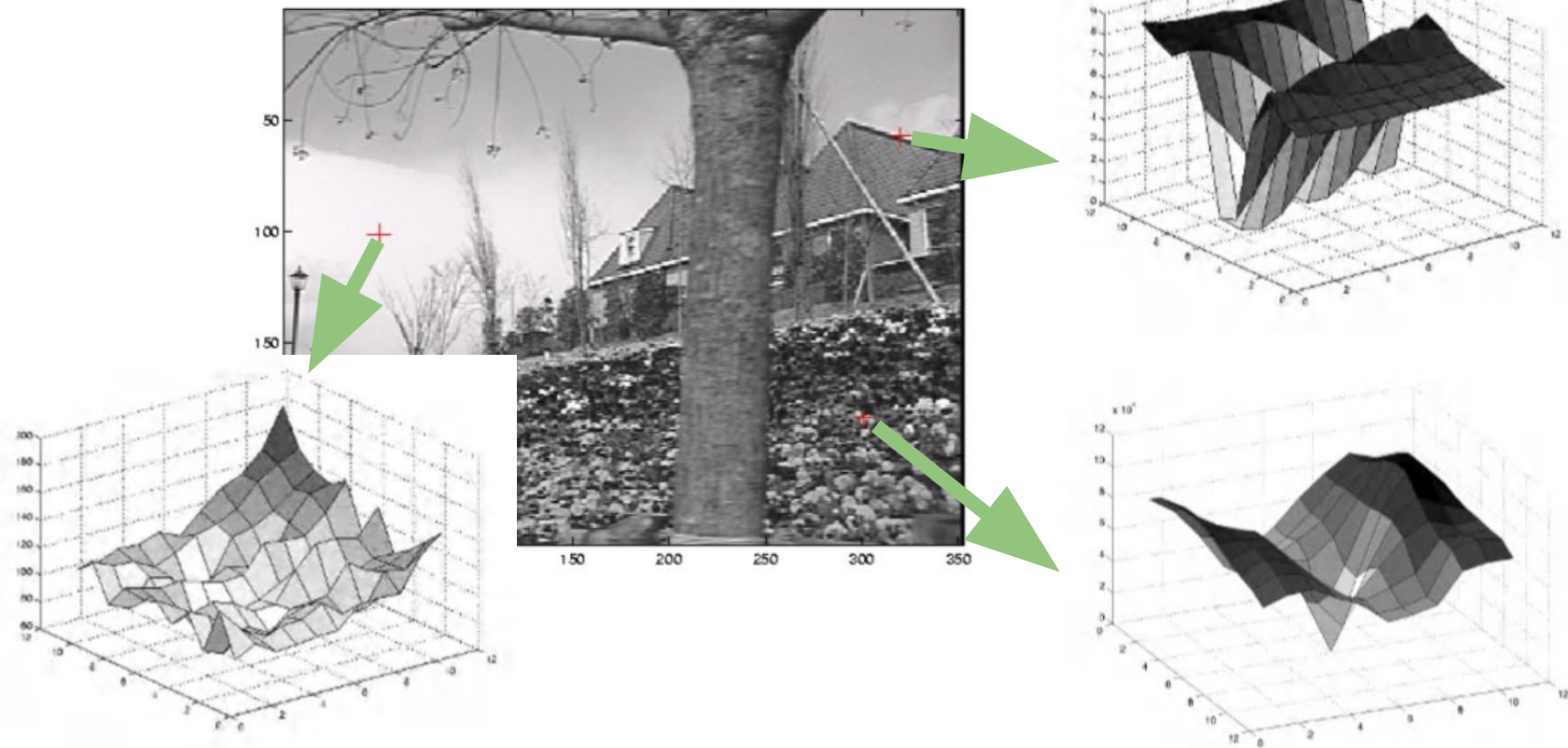
Матрица автокорреляции



Матрица автокорреляции



Матрица автокорреляции



Разложение в ряд Тейлора

Функцию $f(x)$ в точке a можно разложить в ряд:

$$f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n + \dots$$

Математика детектора Харриса

$$E(u, v) = \sum_{x,y} \underbrace{w(x, y)}_{\text{window function}} \underbrace{[I(x+u, y+v) - I(x, y)]}_{\text{shifted intensity} - \text{intensity}}^2$$

$$E(u, v) = \sum_{x,y} \underbrace{w(x, y)}_{\text{window function}} \underbrace{[I(x, y) + uI_x + vI_y - I(x, y)]}_{\text{shifted intensity} - \text{intensity}}^2 \quad \text{Taylor Series}$$

$$E(u, v) = \sum_{x,y} w(x, y) [uI_x + vI_y]^2$$

$$E(u, v) = \sum_{x,y} w(x, y) \left[\begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} I_x \\ I_y \end{pmatrix} \right]^2$$

$$E(u, v) = \sum_{x,y} w(x, y) \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} I_x \\ I_y \end{pmatrix} \begin{pmatrix} I_x & I_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

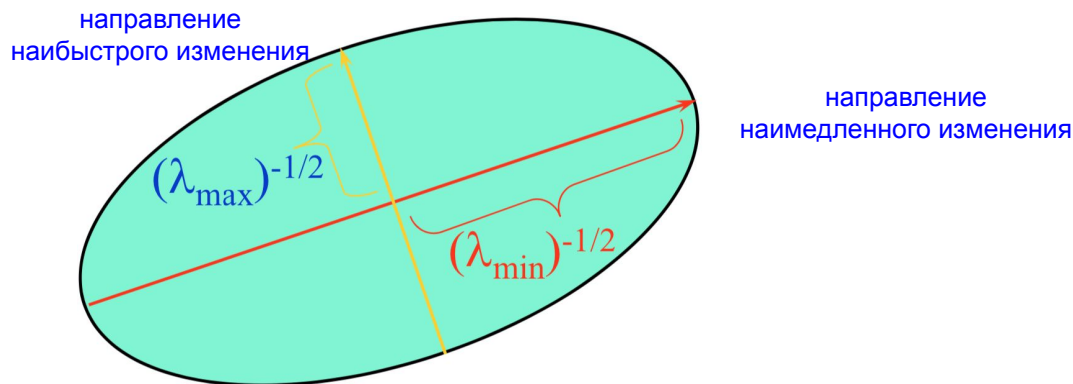
$$E(u, v) = \begin{pmatrix} u & v \end{pmatrix} \left[\sum_{x,y} w(x, y) \begin{pmatrix} I_x \\ I_y \end{pmatrix} \begin{pmatrix} I_x & I_y \end{pmatrix} \right] \begin{pmatrix} u \\ v \end{pmatrix} \quad E(u, v) = \begin{pmatrix} u & v \end{pmatrix} M \begin{pmatrix} u \\ v \end{pmatrix}$$

Математика детектора Харриса

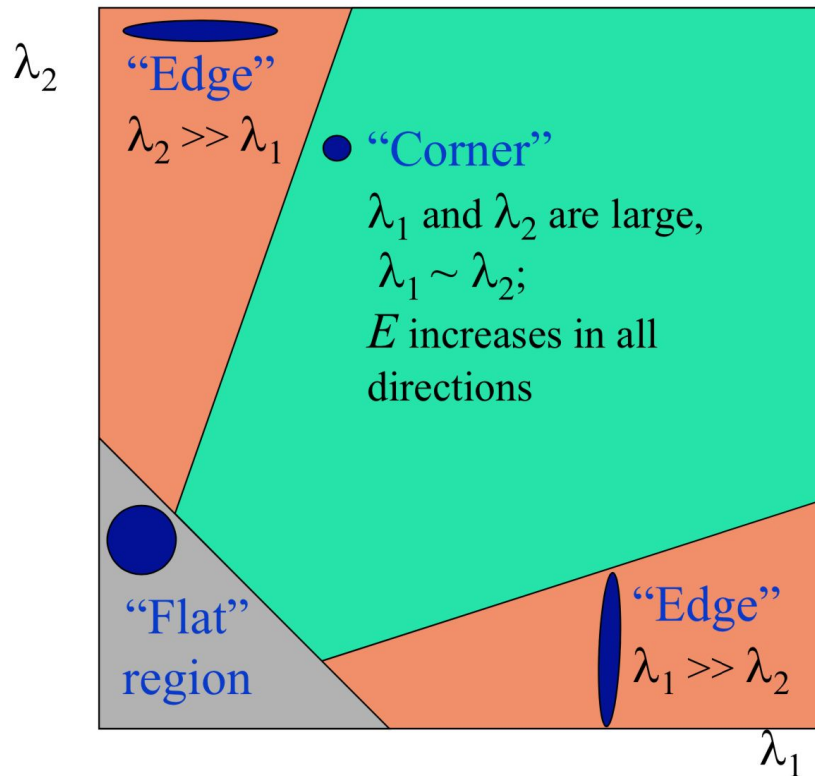
$$E(u, v) = \begin{pmatrix} u & v \end{pmatrix} M \begin{pmatrix} u \\ v \end{pmatrix}$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Если посмотрим на собственные числа матрицы M

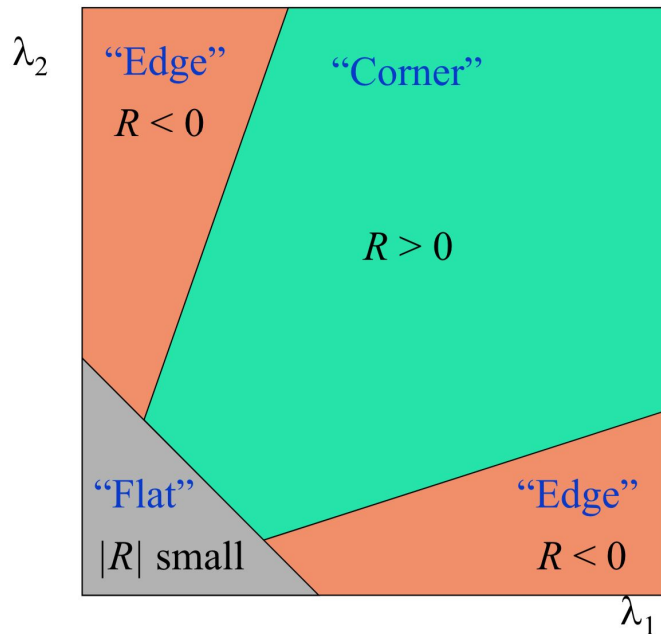


Зависимость E от собственных значений



Функция отклика углов по Харрису

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad \text{где } k = 0.04 - 0.06$$

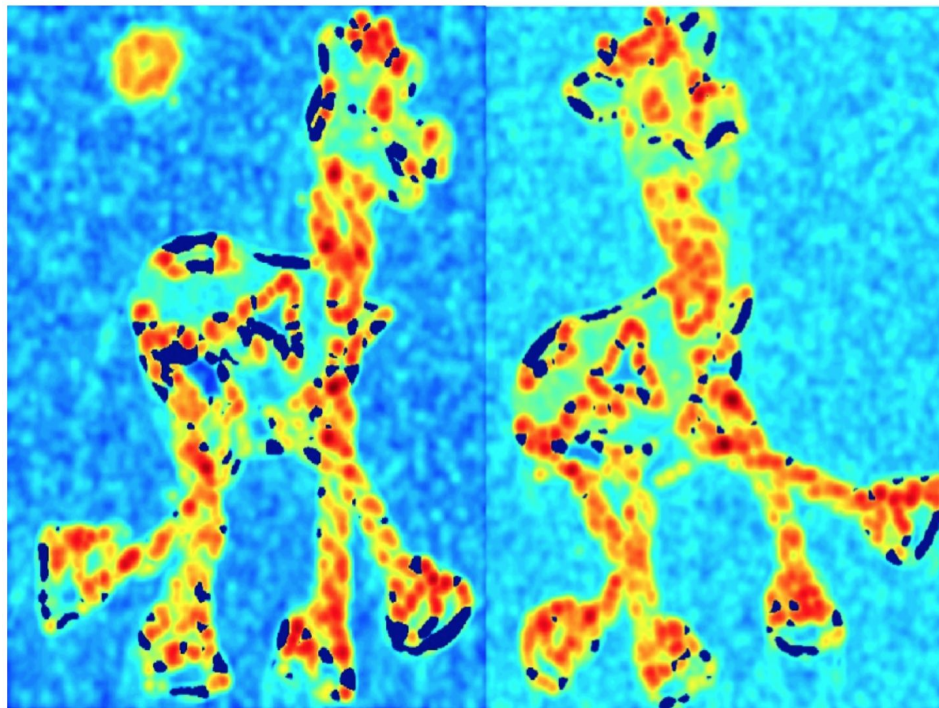


Пример



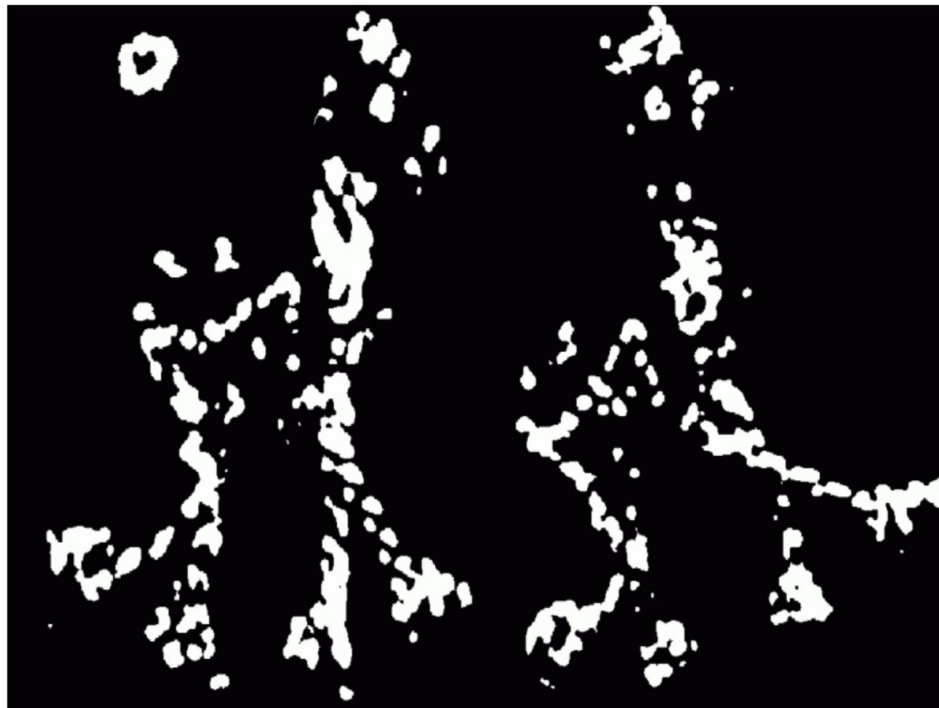
Демонстрация по шагам

Считаем функцию отклика угла R



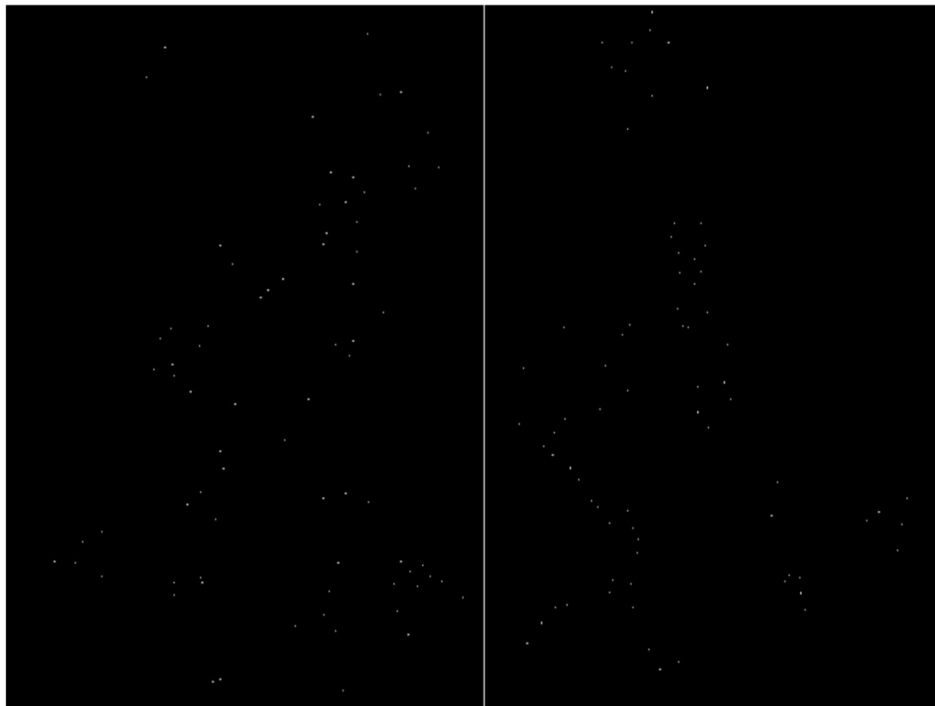
Демонстрация по шагам

Найдем точки с большим откликом $R > \text{порог}$



Демонстрация по шагам

Оставляем точки только локальных максимумов R



Алгоритм детектора Харриса

1. Вычислить градиент изображения в каждом пикселе
 - а. С использованием гауссова сглаживания
2. Вычислить матрицу вторых моментов M по окну каждого пикселя
3. Вычислить отклик угла R
4. Отсечение R по порогу
5. Найти локальные максимумы функции отклика (non-maximum suppression) по окрестности заданного радиуса
6. Выбор N самых сильных локальных максимумов

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

Вычисление дескрипторов характерных точек

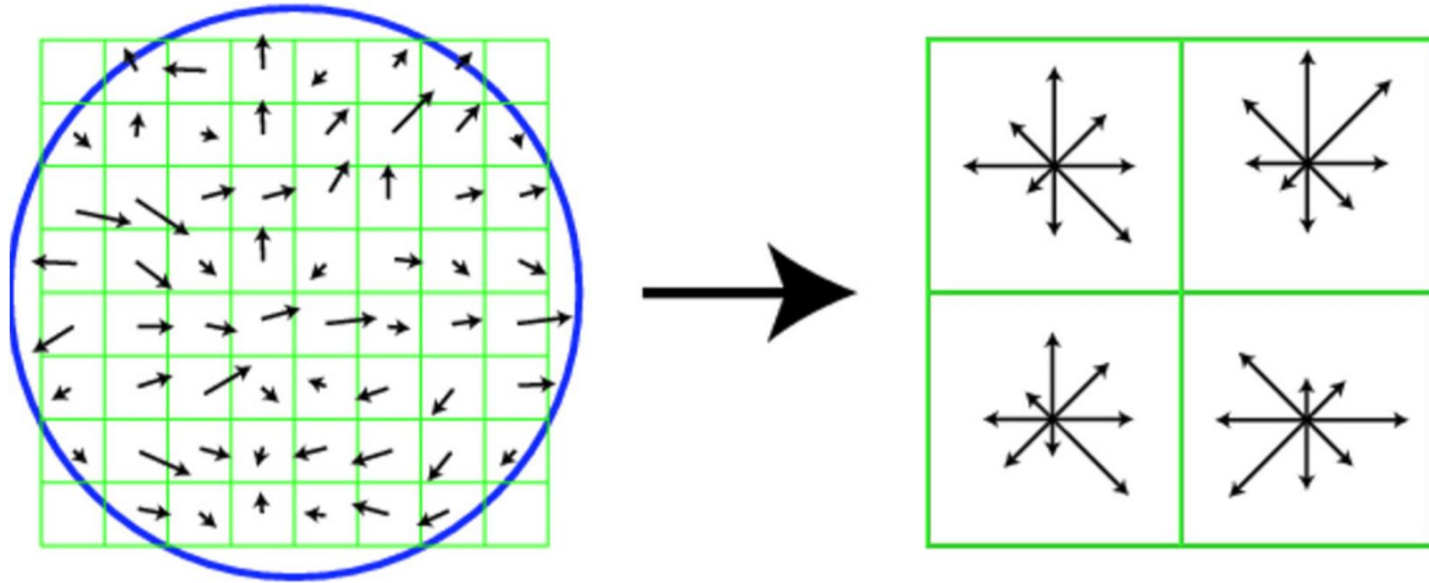
SIFT - Scale Invariant Feature Transform

- 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

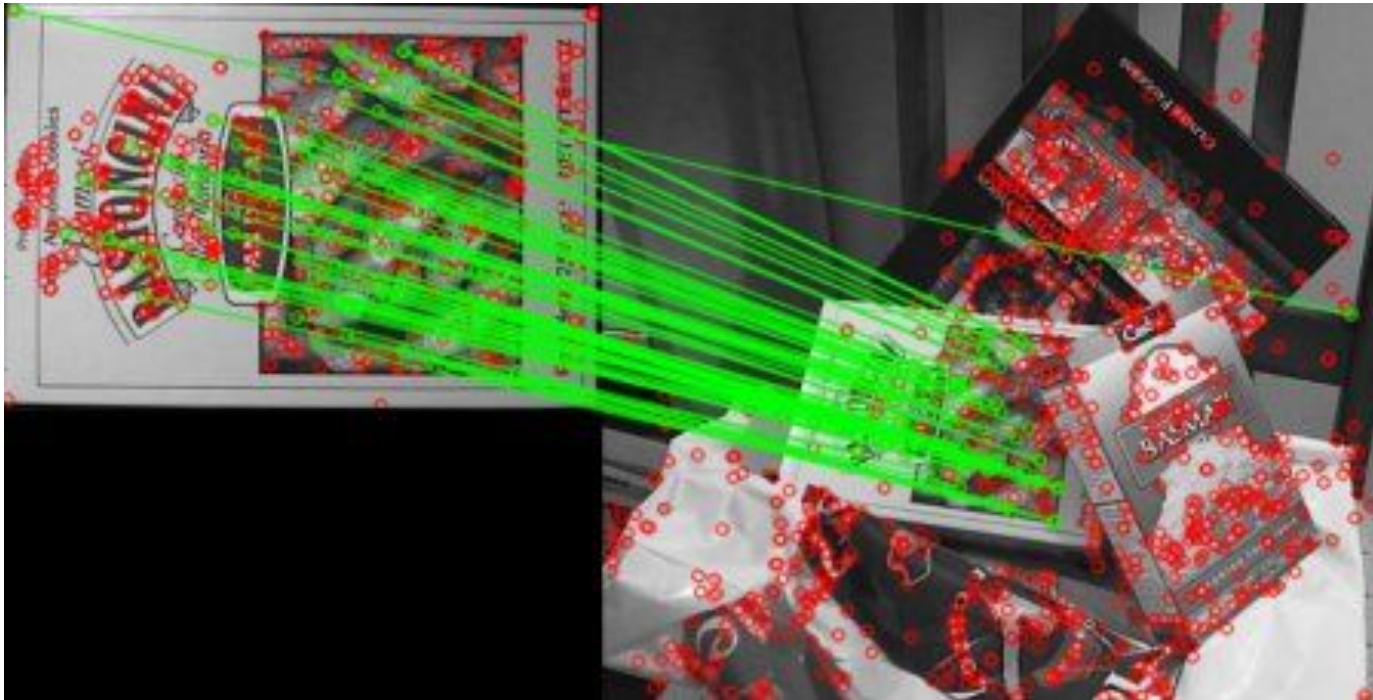


Матчинг характерных точек

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

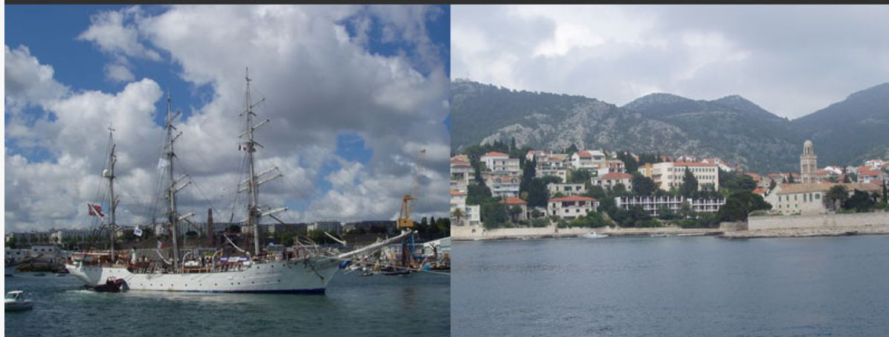
- картинки могут быть похожи по-разному
 - мета теги
 - время создания
 - цвет
 - изображения с похожими предметами

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

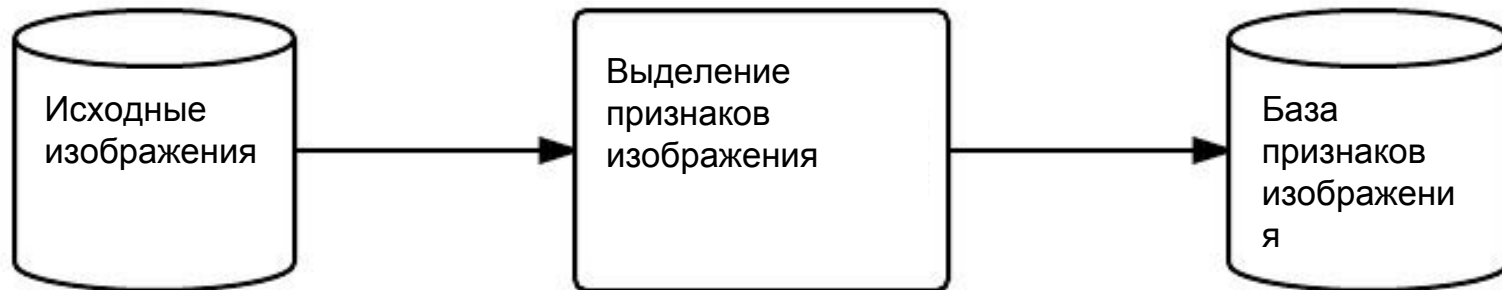
Выделение признаков изображения

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

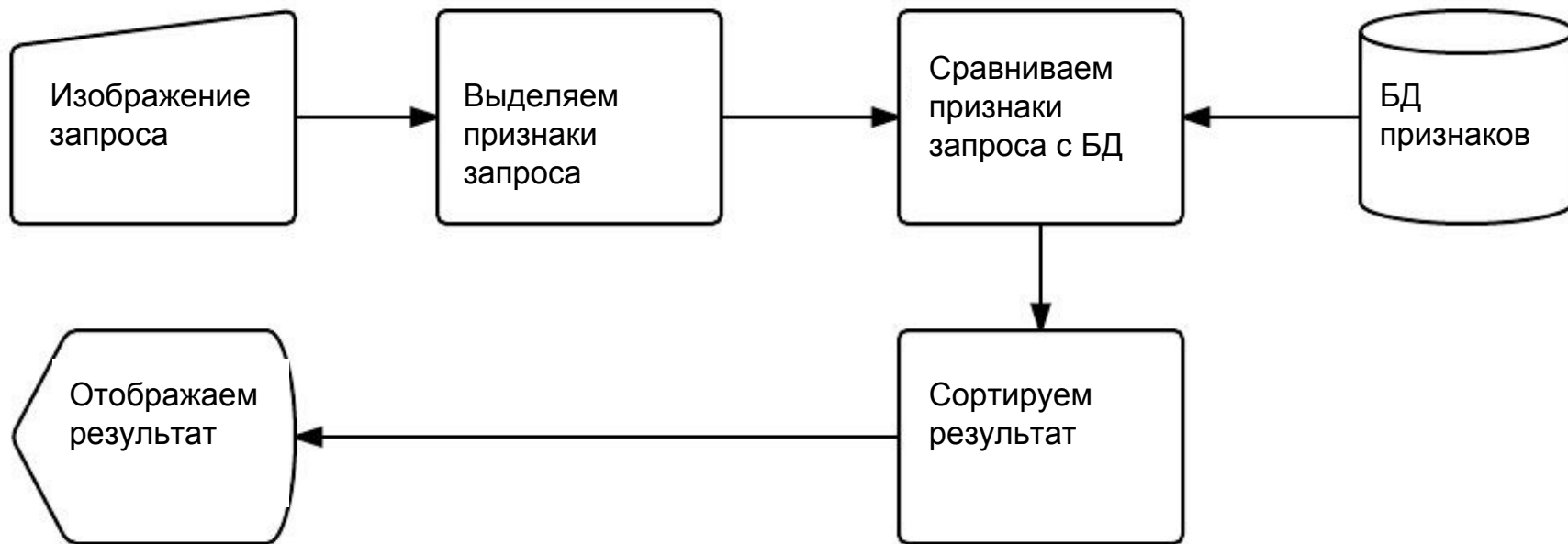
Выделение признаков изображения



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

- 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](#)