

извлечение признаков и поиск

# ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ



# ВЯЧЕСЛАВ МУРАШКИН

Руководитель исследовательской группы  
Яндекс



[mvjacheslav@gmail.com](mailto:mvjacheslav@gmail.com)



[@a4tunado](https://t.me/a4tunado)

# Сегодня мы научимся

**1** решать задачу распознавания изображений методом понижения размерности

**2** выделять признаки для поиска и анализа изображений

**3** определять характерные точки на изображении

**4** строить систему поиска изображений



# ПЛАН ЗАНЯТИЯ

# 1

## Анализ главных компонент в задачах CV Представление свойств изображения с помощью гистограмм

- Гистограммы цветов
- Гистограммы градиентов

# 2

## Характерные точки

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

# 3

## Поиск изображений по контенту CBIR

- Выделение признаков и индексация
- Обзор архитектуры
- Поиск по индексу



# ПРИМЕРЫ ЗАДАЧ КОМПЬЮТЕРНОГО ЗРЕНИЯ



## 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



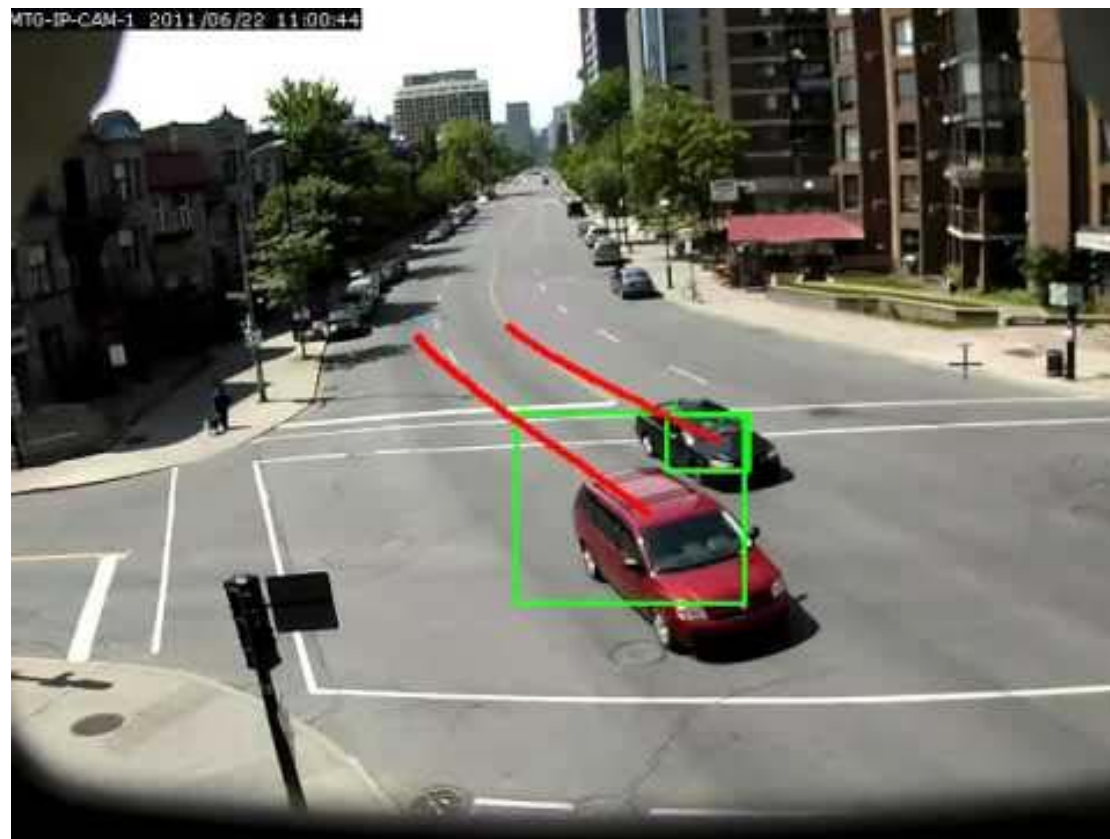
predicted: Sharon  
true: Sharon



predicted: Schroeder  
true: Schroeder



# Optical Flow. *Пример*



# Поиск похожих изображений (CBIR). *Пример*

Query Image



Retrieved Results



---

# РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ (EIGENFACE)

## РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

---

1

Изображение можно представить в виде вектора длины  $H \times W$

2

Большая размерность данных (число пикселей) затрудняет их обработку

3

Для сокращения размерности применяется метод РСА

## РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

---

1

В результате преобразование РСА получаем представления изображений в базисе меньшей размерности

2

Полученное сжатое представление можно использовать для распознавания изображений



## РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

---

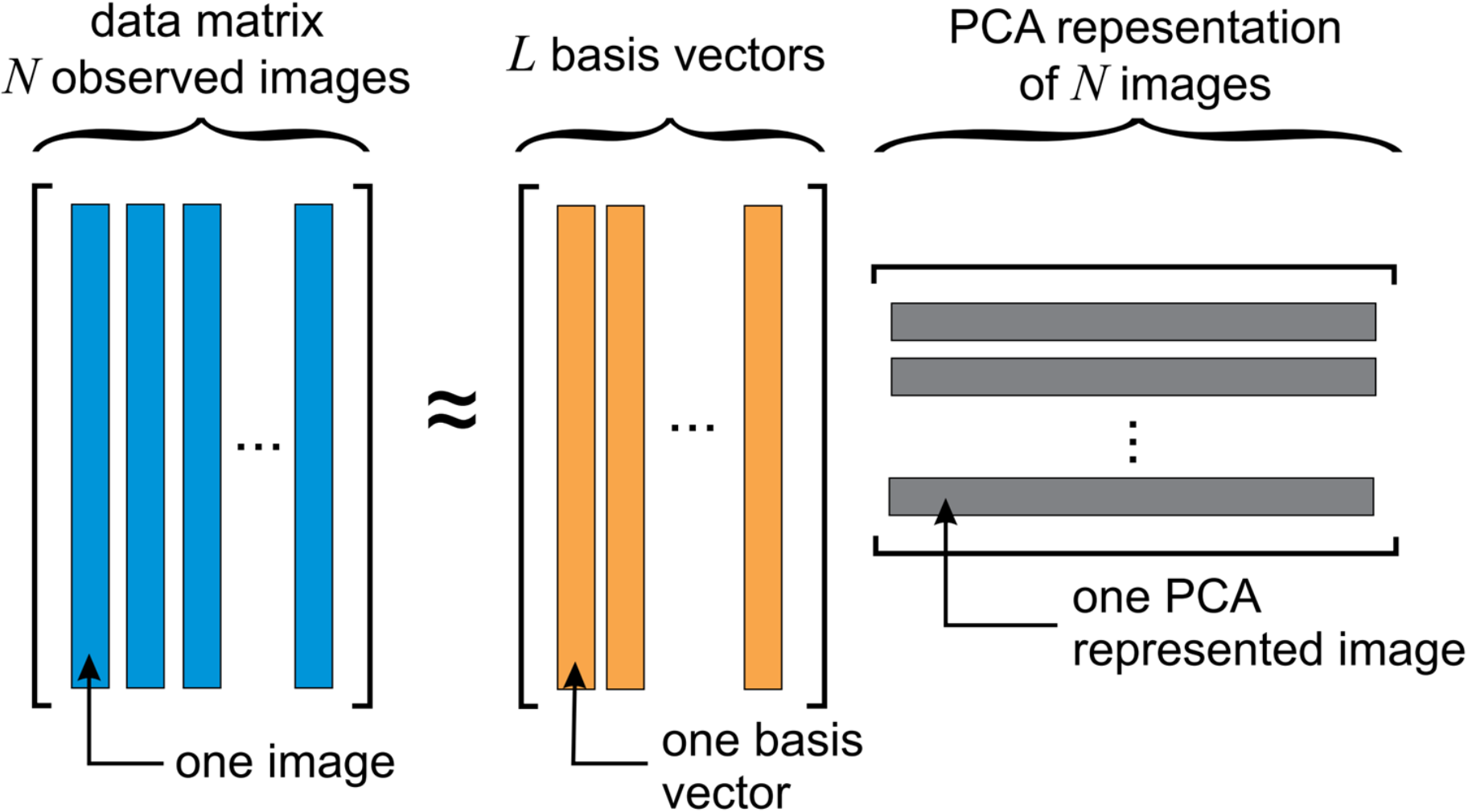


## РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

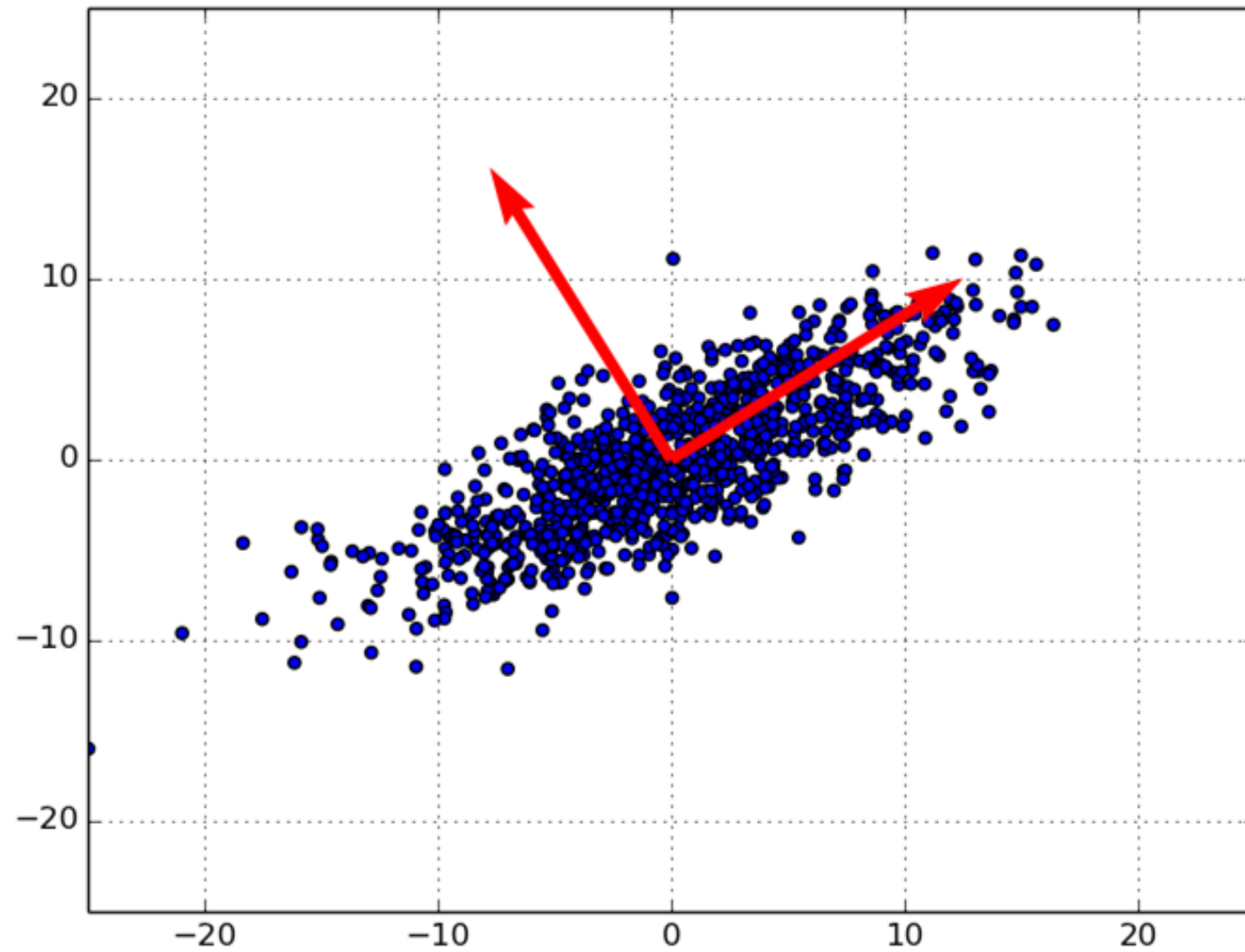




РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ



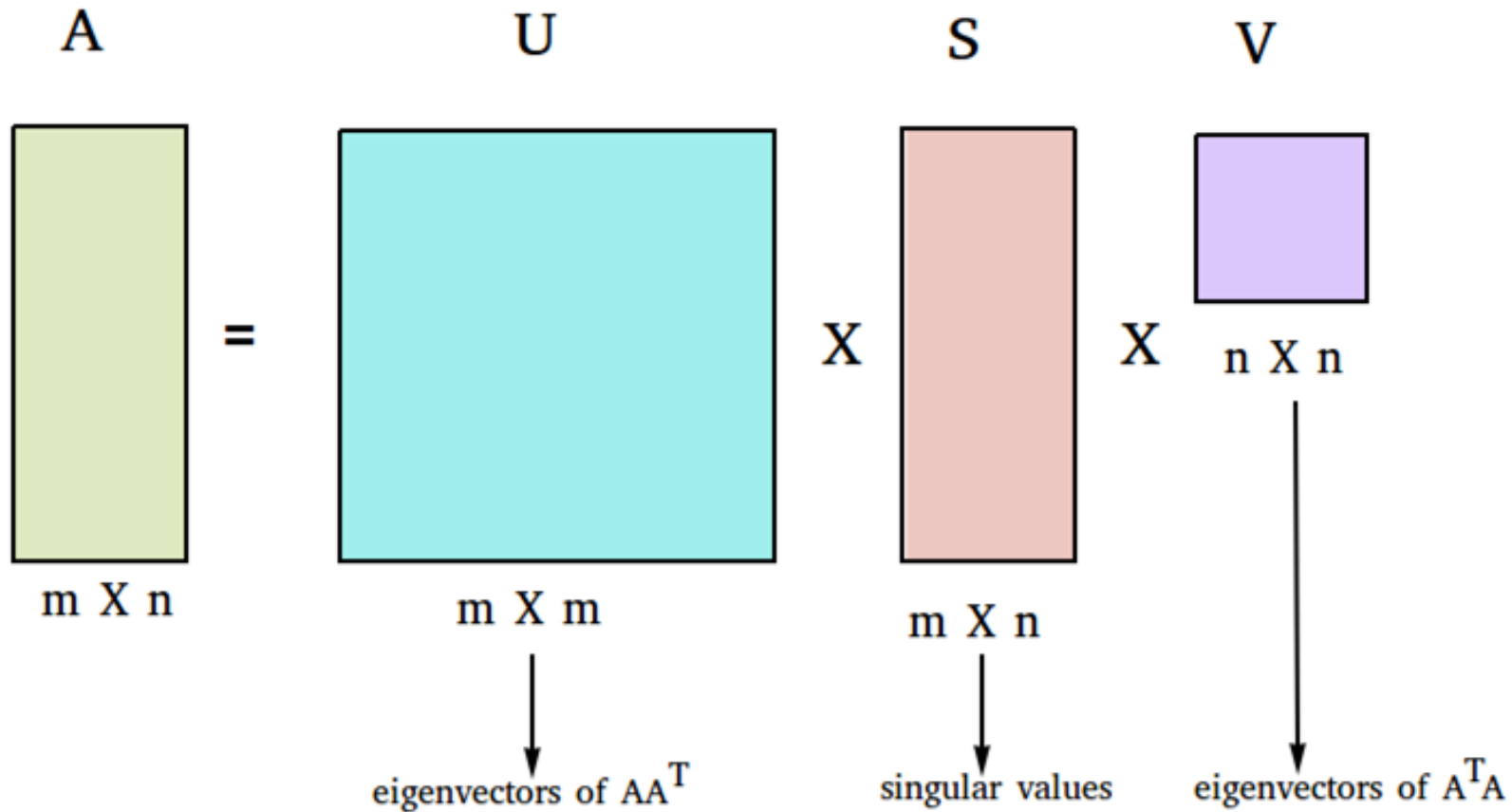
## РСА — АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ



## Матрица ковариации

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

# Разложение SVD



# 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

1

представляют  
собой обобщенное  
описание  
изображения

2

как правило  
гистограммы  
инвариантны  
к масштабу  
и повороту  
изображений

3

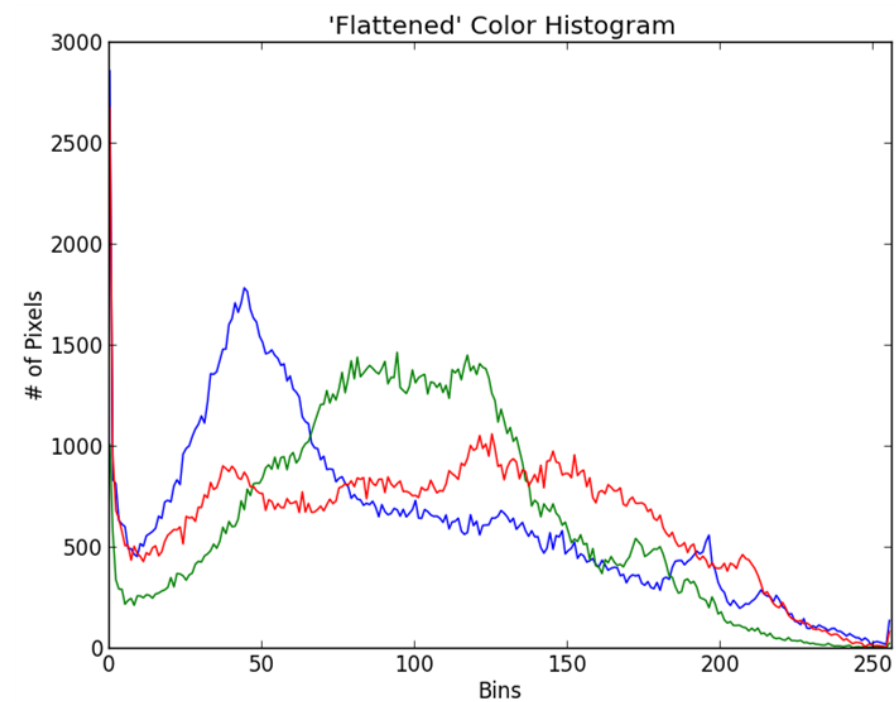
позволяют  
сравнивать  
изображения  
и находить  
похожие



# ГИСТОГРАММЫ ЦВЕТОВ



# ГИСТОГРАММЫ ЦВЕТОВ





## ГИСТОГРАММЫ ЦВЕТОВ

---

- 1** Разбиваем диапазон значений цвета (0..255) на фиксированное число ячеек (bins), например с шагом 1
- 2** Задаем в каких срезах (каналах) необходимо построить гистограмму
- 3** Для каждой ячейки считаем число соответствующих пикселей на изображении
- 4** Можно вычислять как в пространстве RGB, так и в других цветовых пространствах, например, HSV

## ГИСТОГРАММЫ ЦВЕТОВ

---

cv2.calcHist → 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

---

**1** Не зависят  
от изменения  
масштаба  
изображения

**2** Устойчивы  
к повороту  
и перспективным  
искажениям

**3** В цветовых  
пространствах  
HSV и HSL менее  
чувствительны  
к изменению  
яркости

---

# ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

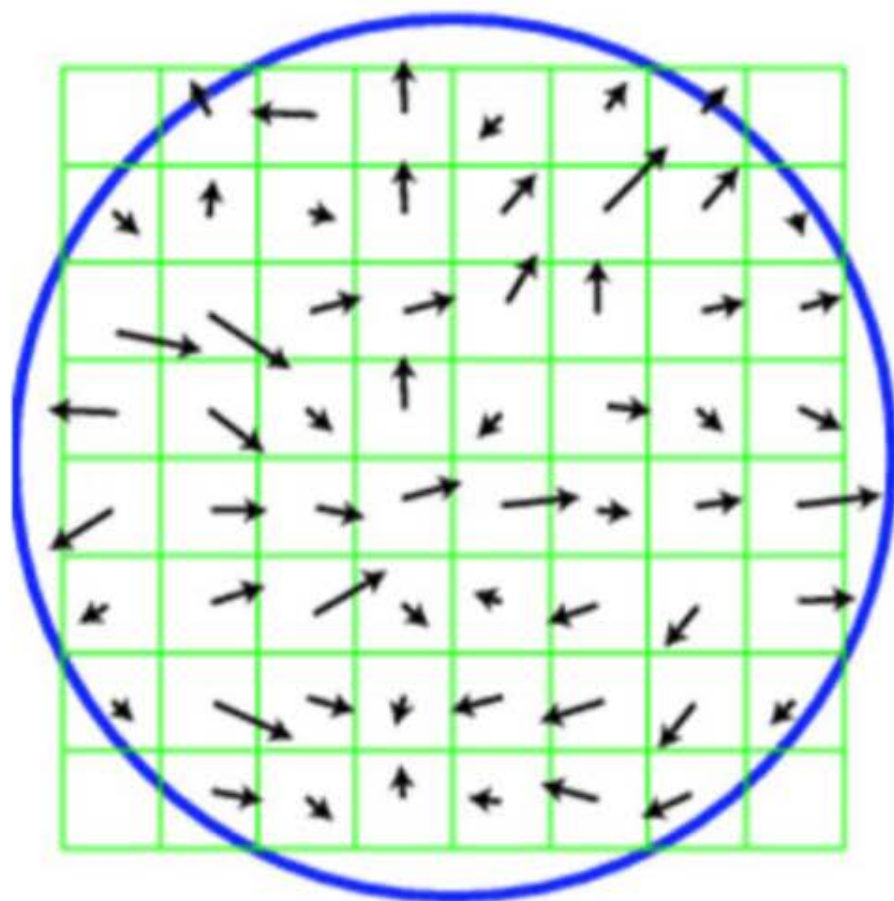
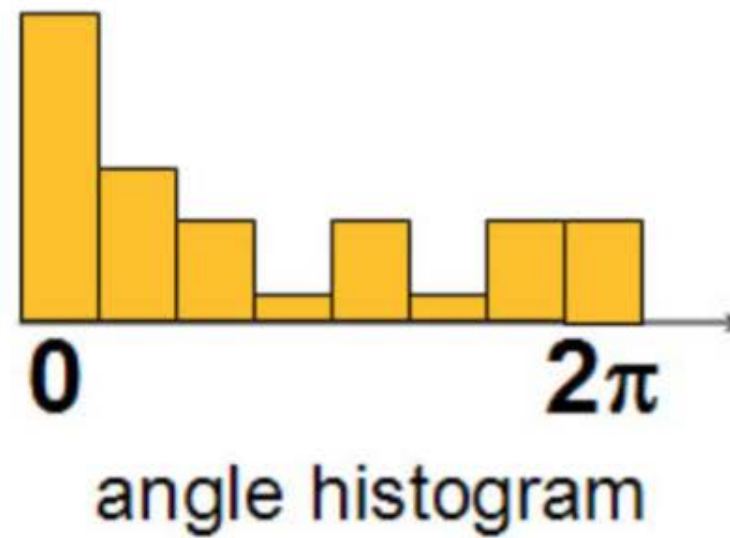


Image gradients



## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

---

1

В каждой точке оцениваем составляющие градиента по осям  $x$  и  $y$

2

Определяем направление и длину вектора градиента

3

Оцениваем гистограмму градиентов

4

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)

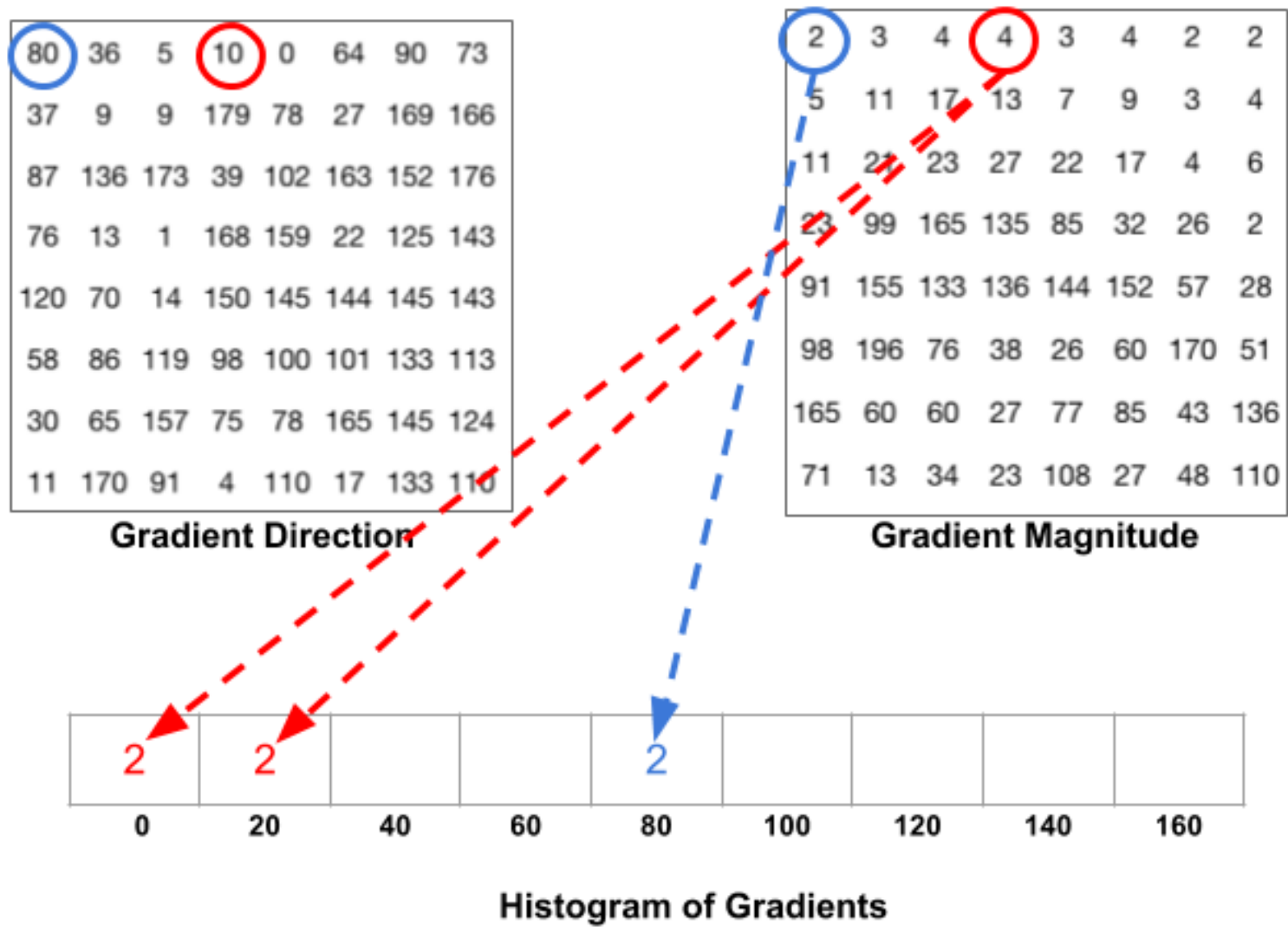
---

**1** Как правило гистограмму градиентов строят для диапазона углов 0..180

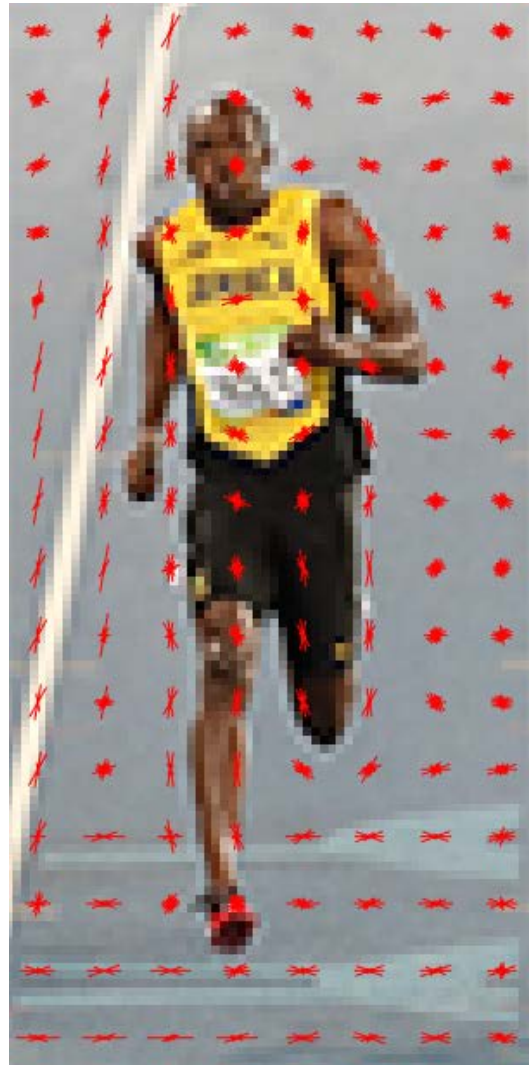
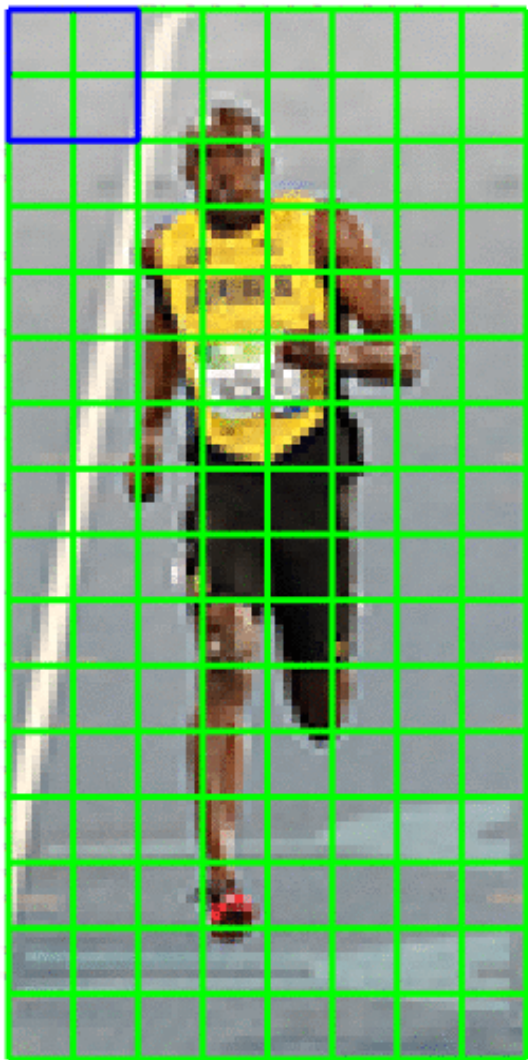
**2** При оценке гистограммы градиентов учитывается как угол, так и длина вектора

**3** Чем больше длина вектора, тем больший вклад вносится в соответствующую ячейку гистограммы

# ГИСТОГРАММЫ ГРАДИЕНТОВ (НОГ)



## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)



[www.learnopencv.com/histogram-of-oriented-gradients/](http://www.learnopencv.com/histogram-of-oriented-gradients/)

## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

Input image



Histogram of Oriented Gradients





## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

cv2.Sobel → **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 → **magnitude, angle**

- **x,y** — вектора с координатами x и y
- **magnitude** — длины векторов
- **angle** — соответствующие углы



## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

---

```
def hog_features(image):  
    hog_computer = cv2.HOGDescriptor()  
    return hog_computer.compute(image)
```





## ГИСТОГРАММЫ ГРАДИЕНТОВ (HOG)

---

1

Не чувствительны  
к изменению  
цвета

2

Устойчивы  
к изменению  
яркости

3

Устойчивы  
к изменению  
масштаба



# ХАРАКТЕРНЫЕ ТОЧКИ

## ХАРАКТЕРНЫЕ ТОЧКИ

---

Позволяют находить  
**одинаковые области**  
(предметы) на разных  
изображениях

### Используются для:

- склейки панорам
- составления карт по спутниковым снимкам



## ХАРАКТЕРНЫЕ ТОЧКИ

---

1

Точка,  
обладающая  
уникальными  
свойствами

2

Положение точки  
на изображении  
однозначно определяется  
по ее свойствам

*дескриптору*

3

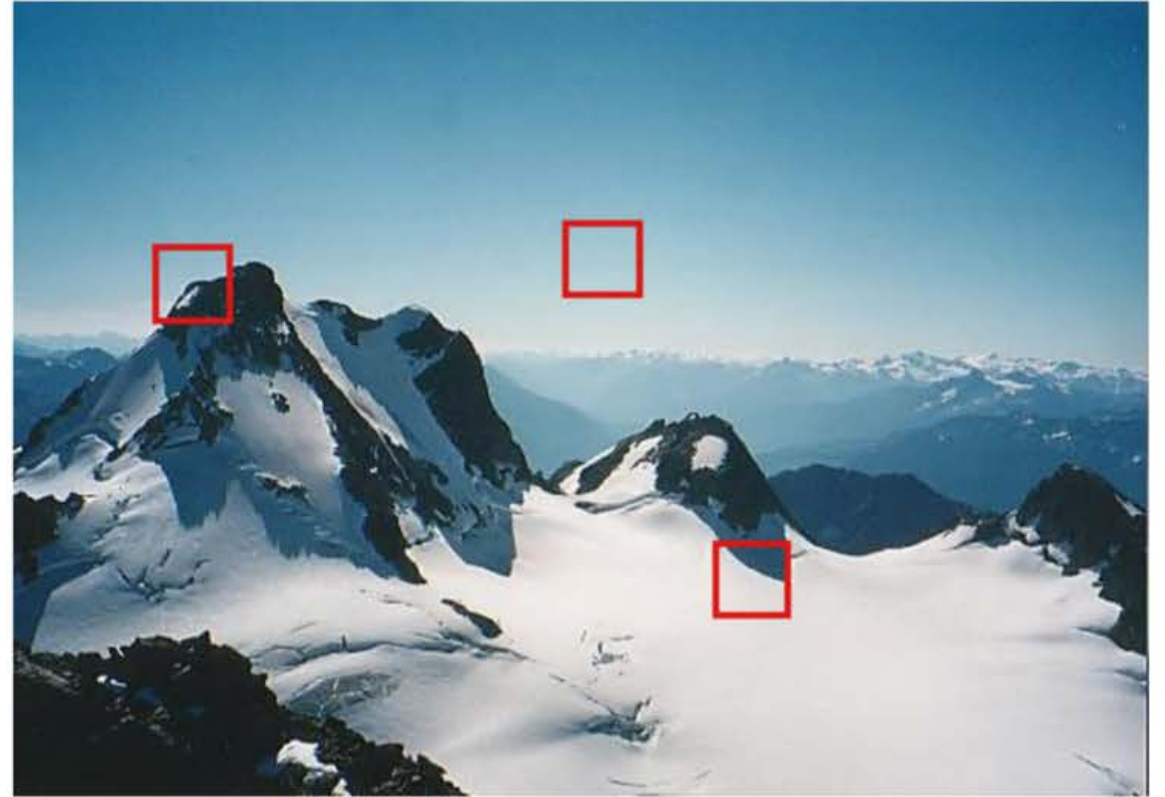
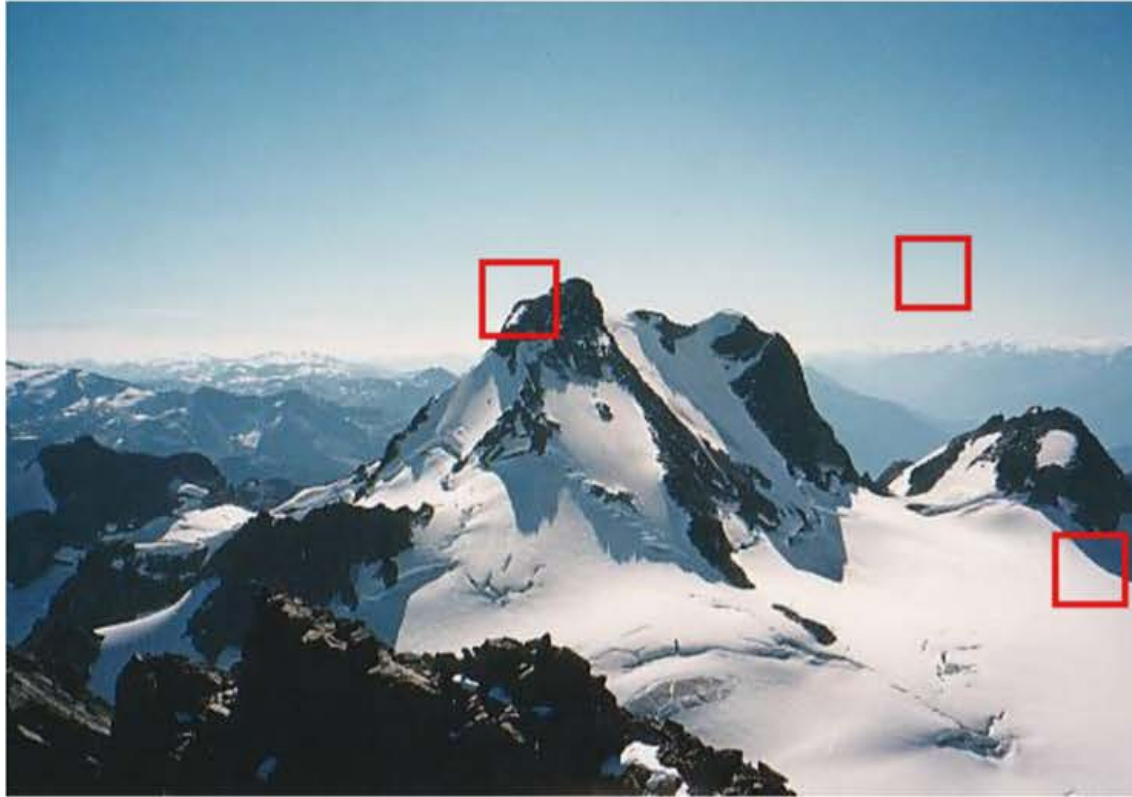
Дескриптор точки  
вычисляется  
на основе  
ее окружения

4

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?*

1. Область вокруг характерной точки **должна сильно варьироваться**
2. В области характерной точки **небольшой сдвиг изображения** должен приводить к существенному различию по сравнению с исходным изображением

# Автокорреляция

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

---

**du** вектор смещения по осям x и y

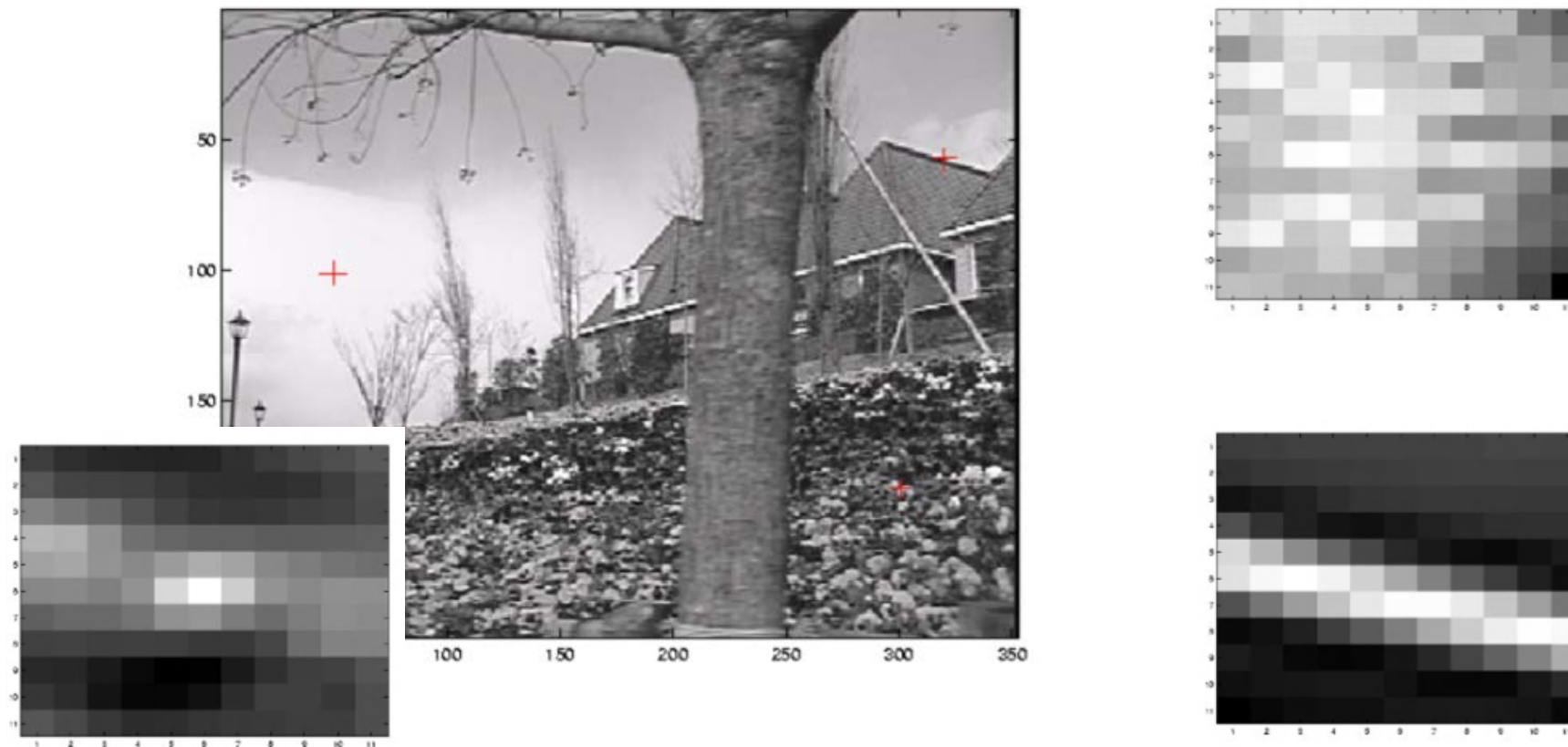
**w** окно или фильтр (гауссовский)

**x<sub>i</sub>** вектор координат пикселя  
изображения

**I<sub>0</sub>** 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 числах матрицы автокорреляции в анализируемой области
- Так как 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

1. Дескриптор основан на построении гистограммы градиентов  
*HOG*
2. В окрестности характерной точки выделяется область размером 16x16 пикселей
3. Для каждого пикселя оценивается вектор градиента
4. Длина вектора градиента взвешивается гауссовским фильтром, таким образом, чтобы пиксели удаленные от характерной точки имели меньший вес



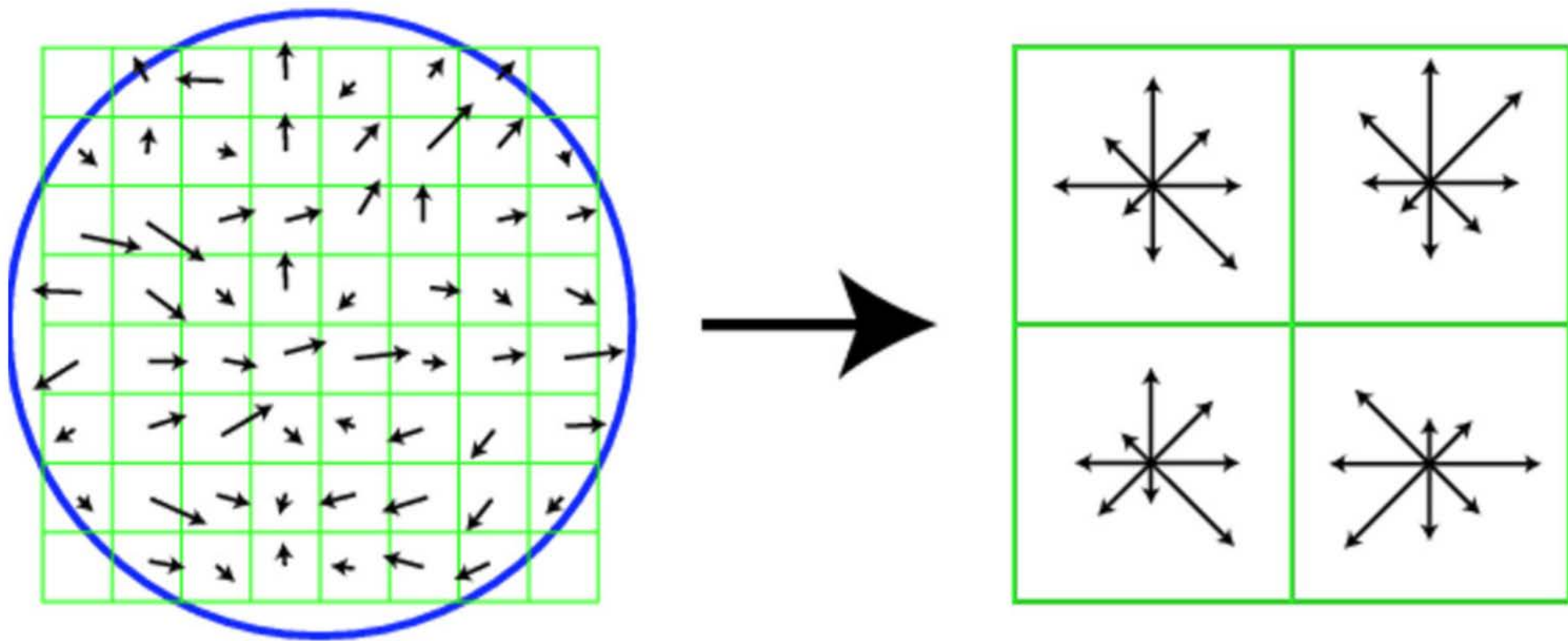
# SIFT — Scale Invariant Feature Transform

1. Исходная область  $16 \times 16$  разбивается на части размера  $4 \times 4$
2. Для каждой части строится гистограмма градиентов с 8 ячейками
3. В результате получается вектор из 128 признаков
4. 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](#)

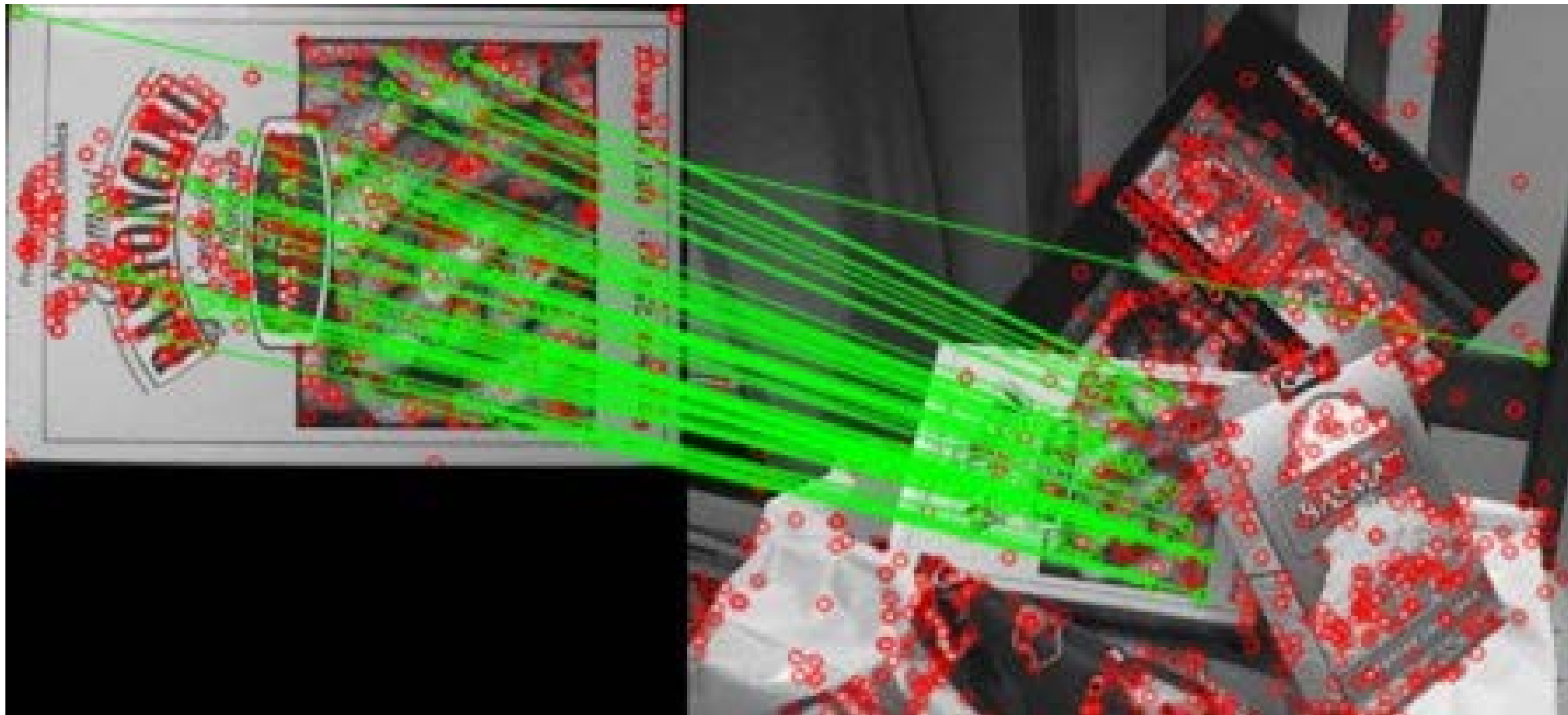
Попарное сравнение всех точек — полный перебор, долго

Индексация перед поиском и поиск по индексу

- Поиск точек в окрестности — [kd-tree](#)
- 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



ПОИСК ПОХОЖИХ ИЗОБРАЖЕНИЙ

# 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

1

Признаки можно считать  
как по всему изображению

2

**Альтернативный способ:**

разбить изображение на области  
и посчитать признаки для каждой  
области отдельно, и затем  
объединить результат

## ПОИСК ПОХОЖИХ ИЗОБРАЖЕНИЙ

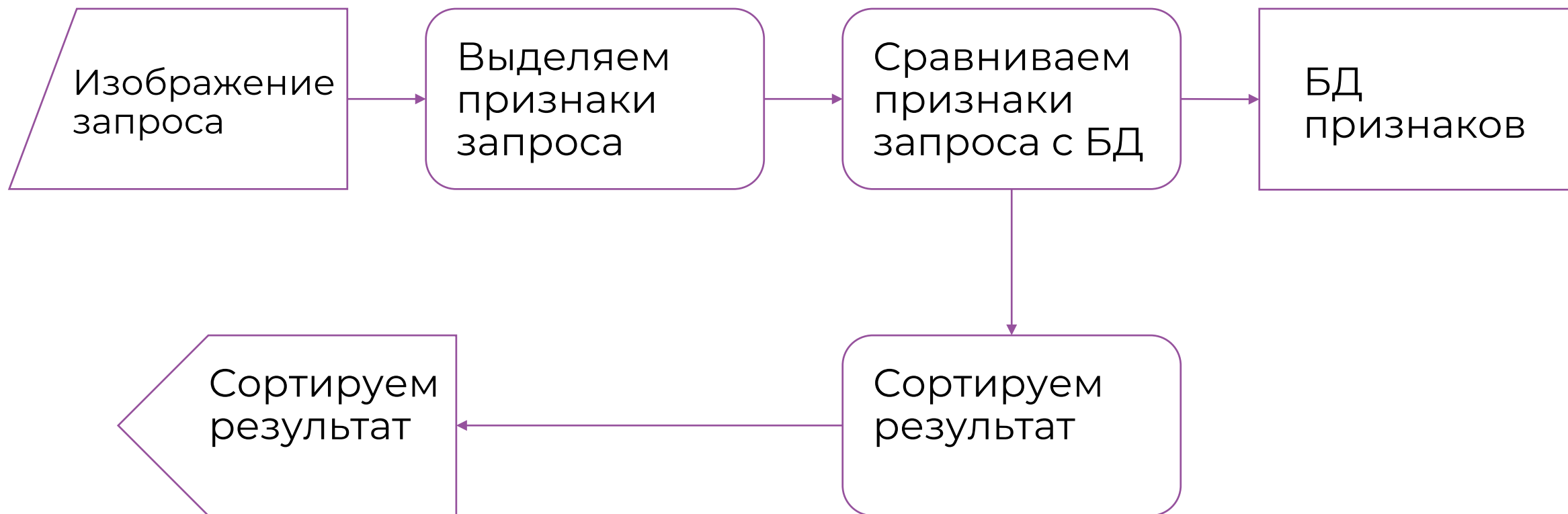


Выделение  
признаков  
изображения

# Выделение признаков изображения



## Архитектура





# Индексация и поиск

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



**РЕЗЮМЕ**

## 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

---

1

Для матчинга изображений используют характерные точки

2

Характерные точки выделяются большими значениями автокорреляционной матрицы

3

В качестве дескриптора характерной точки можно использовать гистограмму градиентов

4

Для ускорения поиска одинаковых точек на изображениях используется [K-d tree](#)

## РЕЗЮМЕ

---

- 1 Дескрипторы для поиска изображения зависят от задачи
- 2 Одним из вариантов дескрипторов могут быть гистограммы цвета или градиента
- 3 Для ускорения поиска необходима индексация базы
- 4 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](#)



**СПАСИБО**