

Old School Vision

Жукова Алина

Задачи компьютерного зрения: Как решать?



What We See

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

What Computers See

Сопоставление шаблонов

- Выбираем объект (напр. глаз)
- Выбираем шаблон (pattern)
- Хотим найти объект на изображении

[pattern]

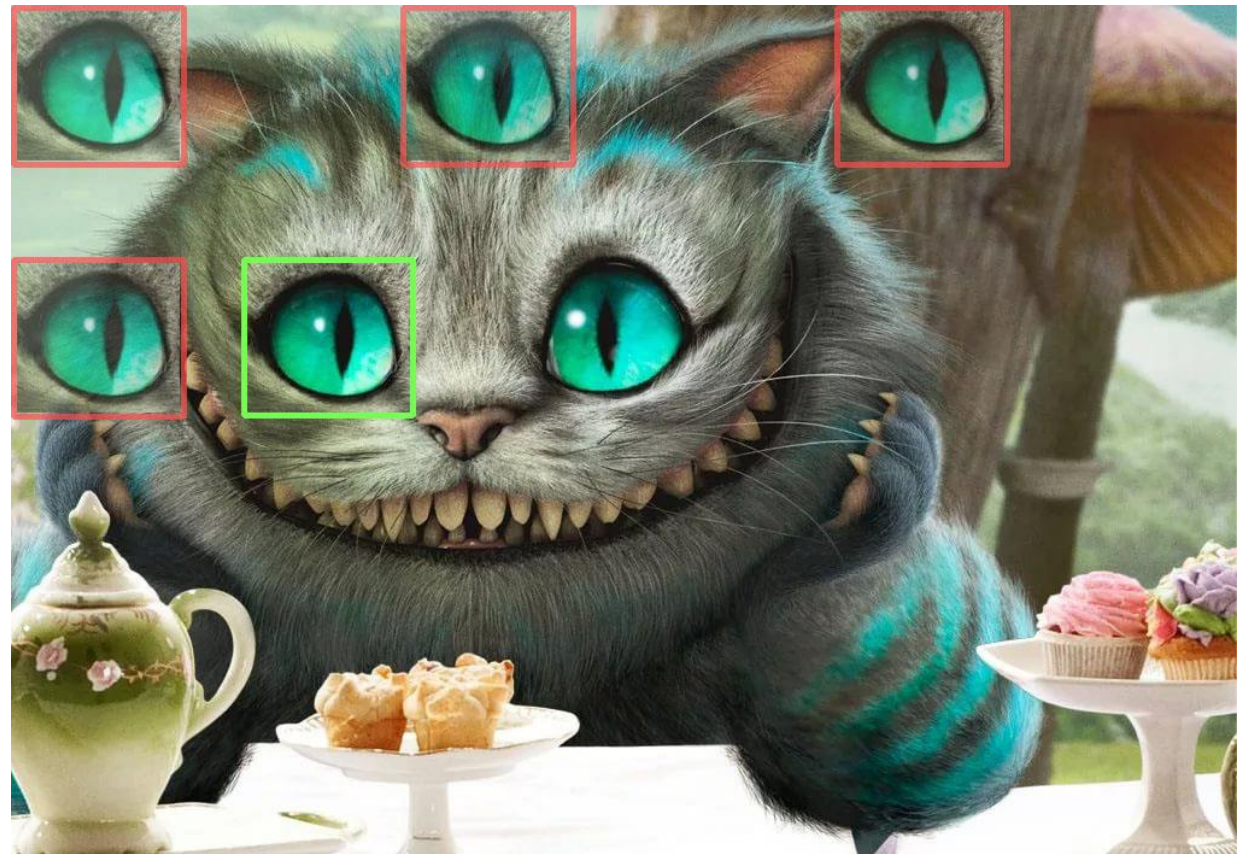
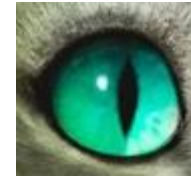


[Image]

Сопоставление шаблонов

- Выбираем объект (напр. глаз)
 - Выбираем шаблон (pattern)
 - Хотим найти объект на изображении
 - Попиксельно сравниваем шаблон со всеми фрагментами изображения
- «Pattern matching»

[pattern]



[Image]

Метрики

$I(X, Y)$ – значение интенсивности пикселя (X, Y)

$$\sum_X \sum_Y |I_1(X, Y) - I_2(X, Y)| \quad \text{L1 метрика (SAD – Sum of absolute differences)}$$

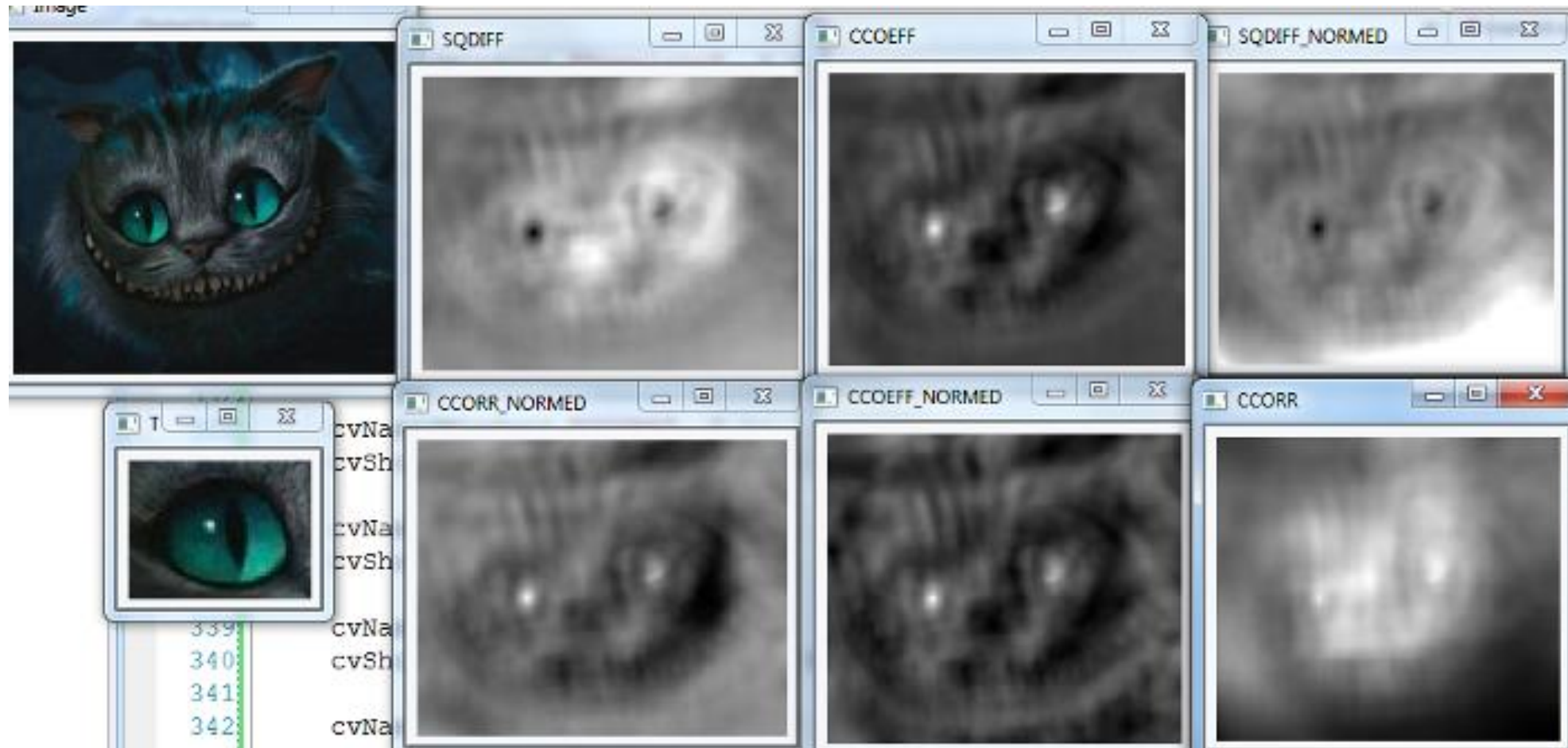
$$\sum_X \sum_Y (I_1(X, Y) - I_2(X, Y))^2 \quad \text{L2 метрика (SSD – Sum of squared differences)}$$

$$\sum_X \sum_Y I_1(X, Y) I_2(X, Y) \quad \text{Кросс-корреляция (CC – Cross-correlation)}$$

- SAD, SSD – минимизируются (0 – точное совпадение)
- CC – максимизируется (1 – точное совпадение)

Сопоставление шаблонов

- Визуализация результата вычисления метрик



Ограничения и проблемы

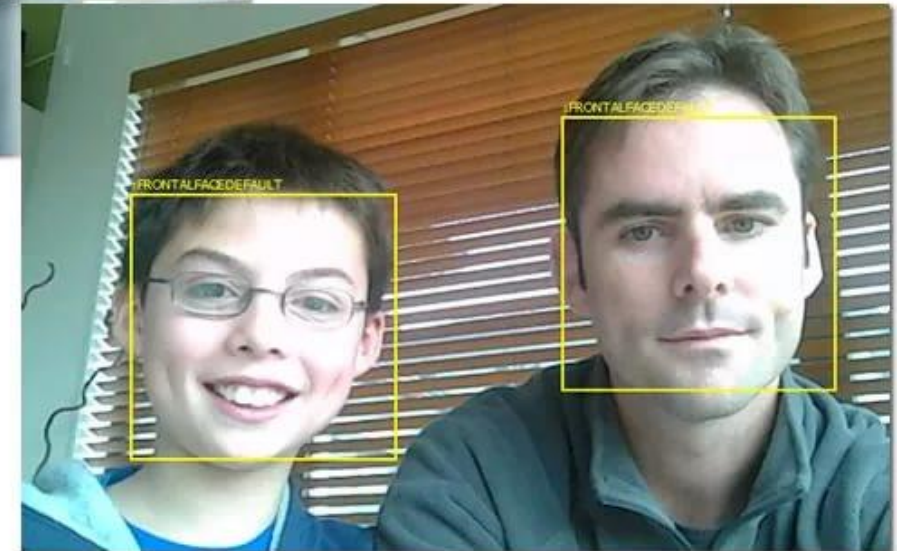
- Ищем конкретный объект, а не класс / категорию объектов
 - ~ не «символ», а конкретную букву в конкретном шрифте
- Трудоёмкость
 - ~ полный перебор параметров
- Модель преобразования
 - ~ в простом варианте неизвестно только положение, размер и ориентация фиксированы
 - ~ чтобы учесть поворот и ориентацию, придётся перебрать все возможные параметры
- Шаблонов может быть много
 - ~ Оптическое распознавание символов (OCR)
 - ~ по шаблону на каждый символ

Где используется?

- Отслеживание объектов в видео
- ~ находим объект – получаем шаблон
- Вычислительно эффективно



Tracking Hand

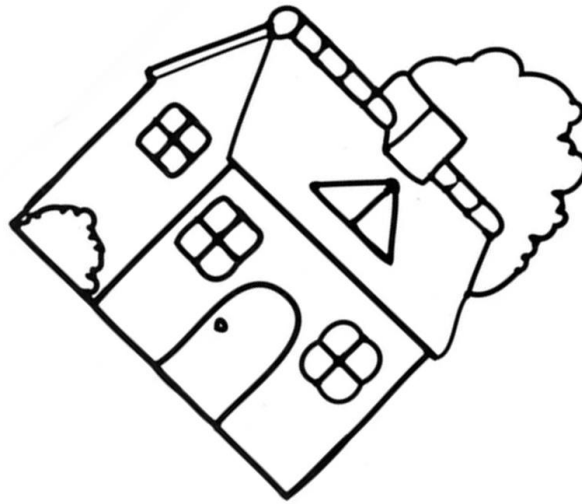


Tracking Faces

Выравнивание изображений

Есть два изображения одного и того же объекта.

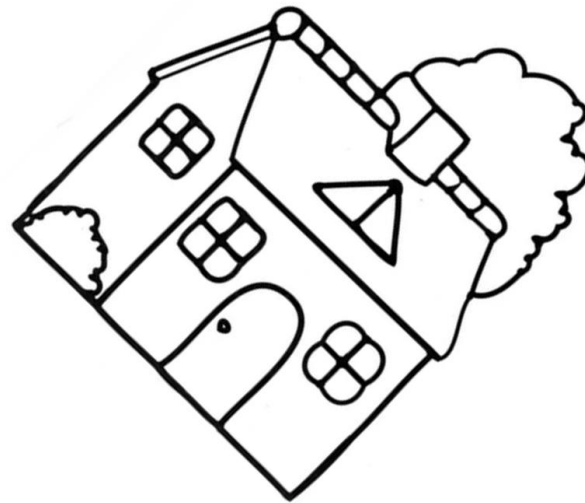
- Как нам совместить изображения?



Выравнивание изображений

1. Brute force («грубая сила»)

- Выбираем модель преобразования (сдвиг, поворот) и набор параметров
- Выбираем метрику для сопоставления
- Перебираем всевозможные значения параметров (в разумных пределах)
- Сопоставляем шаблоны

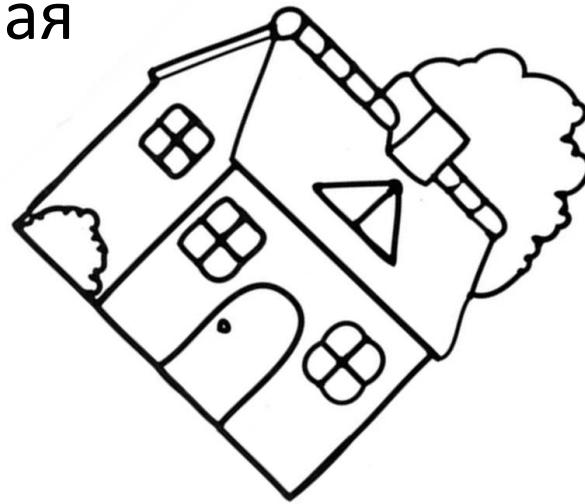


Выравнивание изображений

2. Градиентный спуск

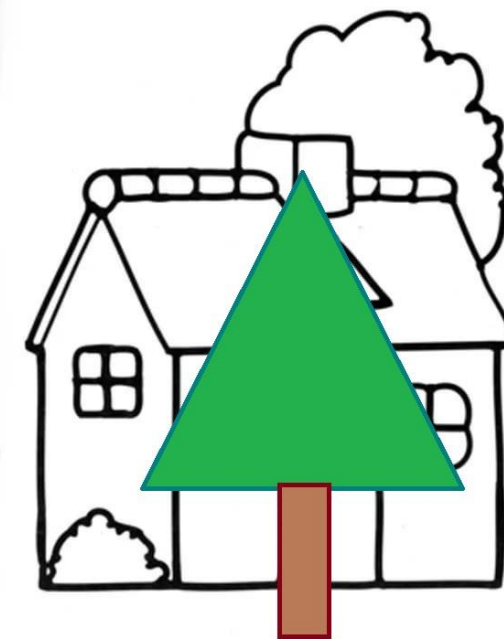
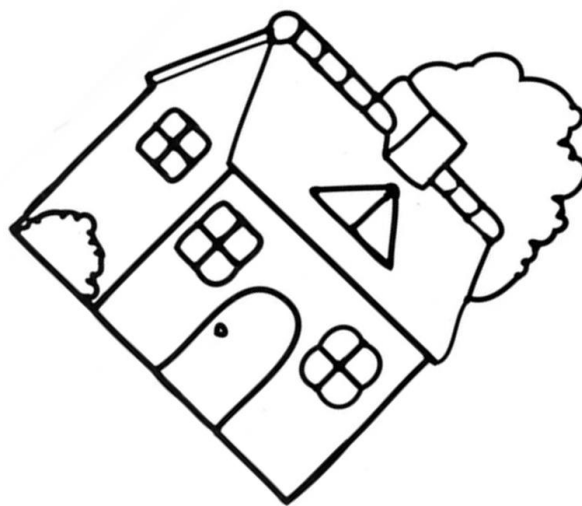
- Используем для оптимизации параметров преобразования
- Целевая функция – выбранная метрика

(!) Но, требуется точное начальное приближение (иначе попадём в loc min)



Выравнивание изображений

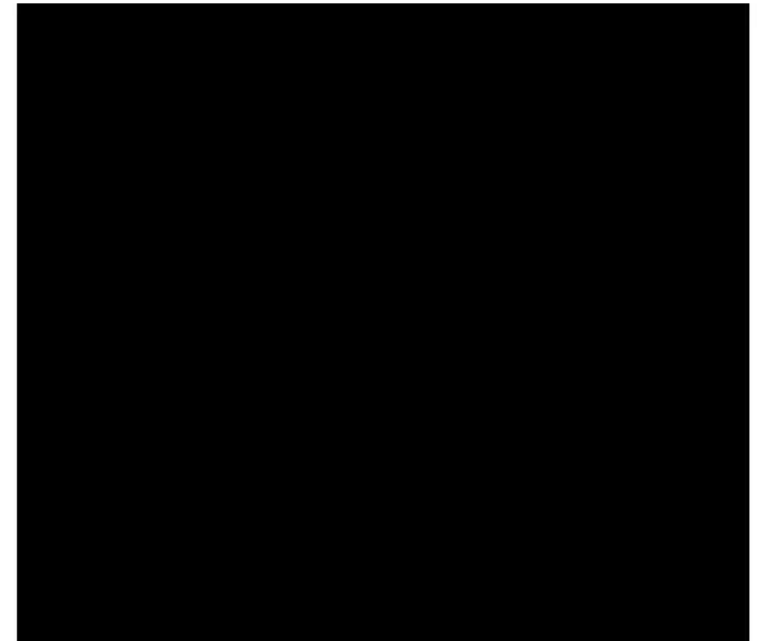
- Что делать, если часть объекта загорожена другим?



Особые точки

- «хорошо различимые» фрагменты изображения
- Нужны для анализа изображения

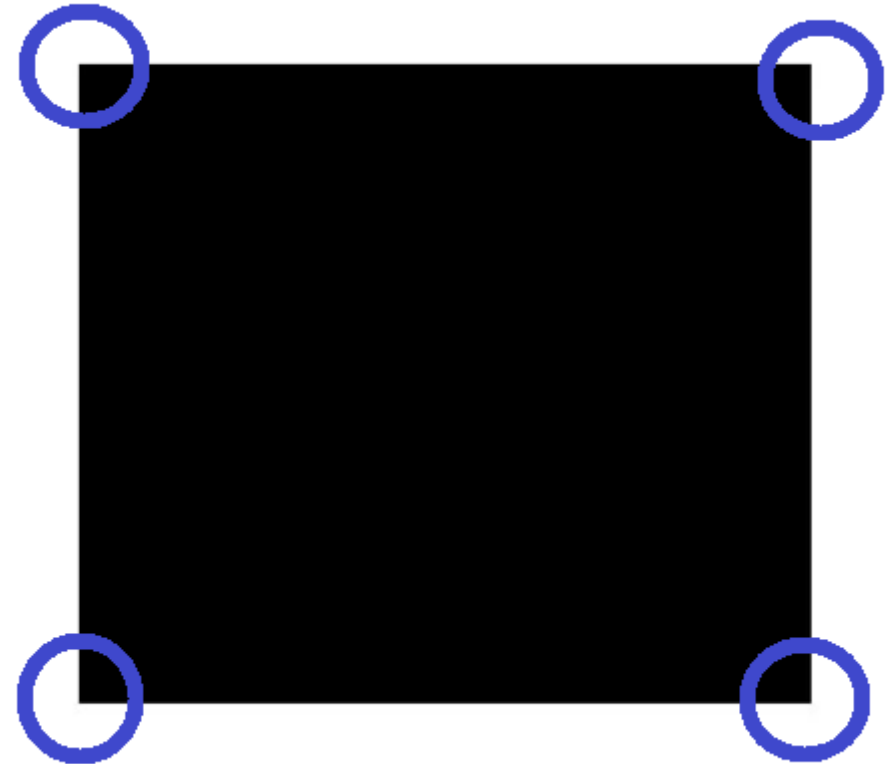
Какие точки на изображении – особые?



Особые точки

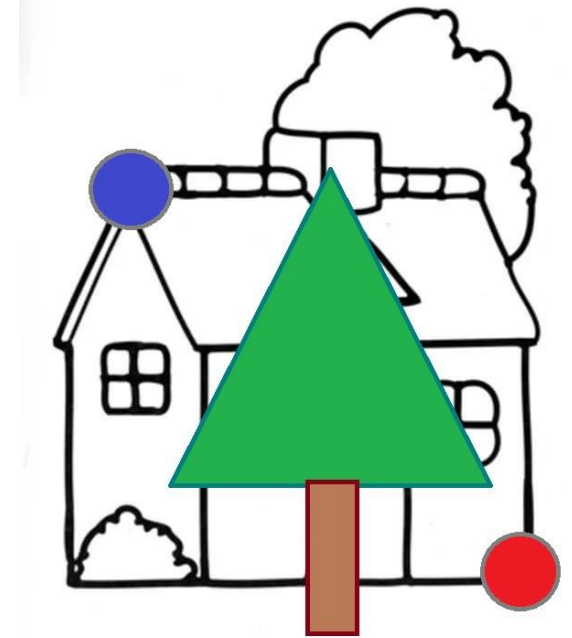
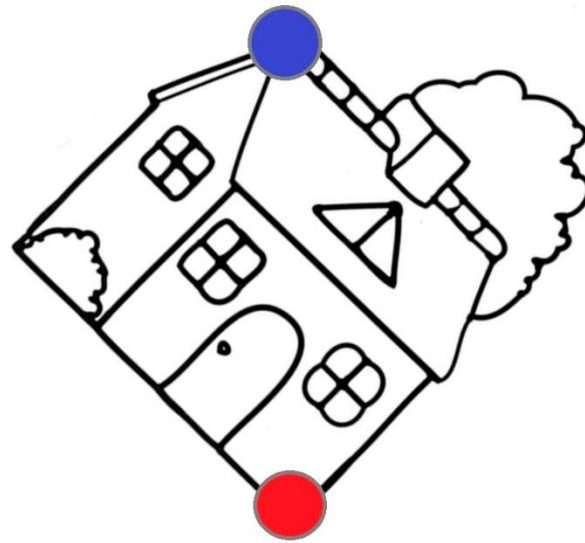
- «хорошо различимые» фрагменты изображения
- Нужны для анализа изображения

Какие точки на изображении – особые?



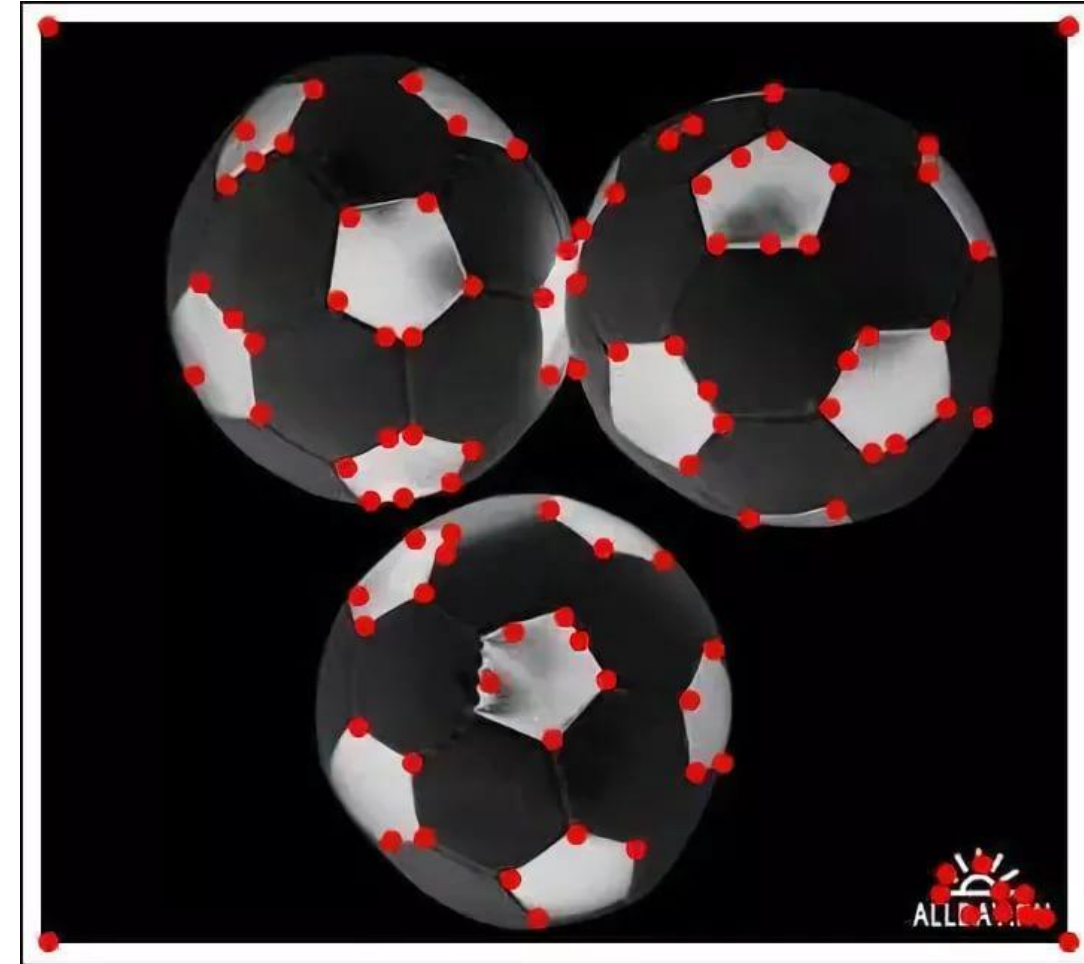
Особые точки

- Найдём преобразование, совмещающее особые точки
- Проверим «совпадение» с помощью попиксельного сравнения



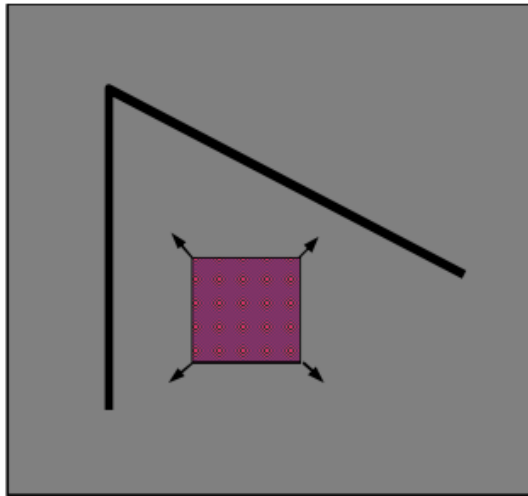
Требования к особенностям

- Повторяемость (Repeatability)
 - i. Особенность (feature) находится в том же месте сцены несмотря на изменения точки обзора и освещения
- Значимость (Saliency)
 - i. Каждая особенность имеет уникальное (distinctive) описание
- Компактность
 - i. Количество особенностей существенно меньше числа пикселей изображения
- Локальность (Locality)
 - i. Особенность занимает маленькую область изображения, поэтому работа с ней нечувствительна к перекрытиям

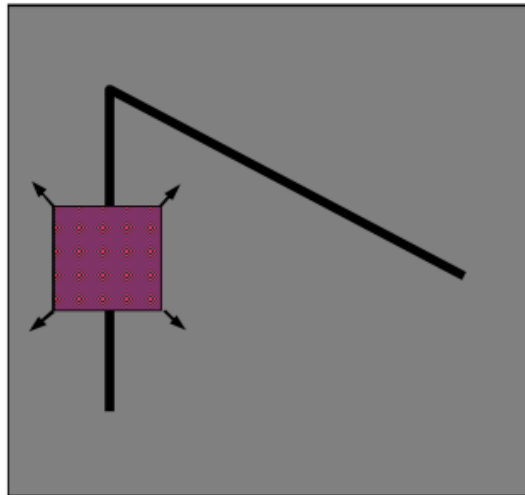


Особые точки

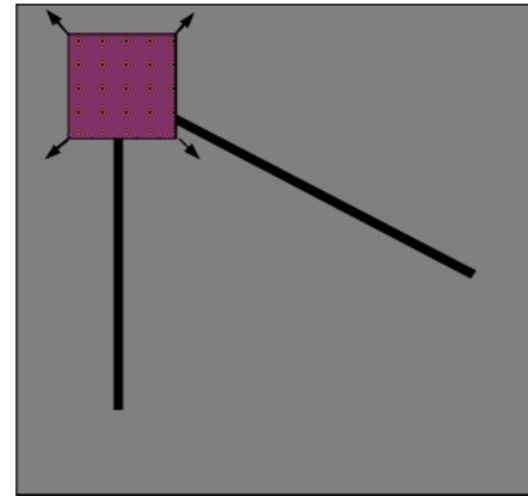
- Самые «хорошие» особенности – углы



монотонный регион:
в любом направлении
изменений нет



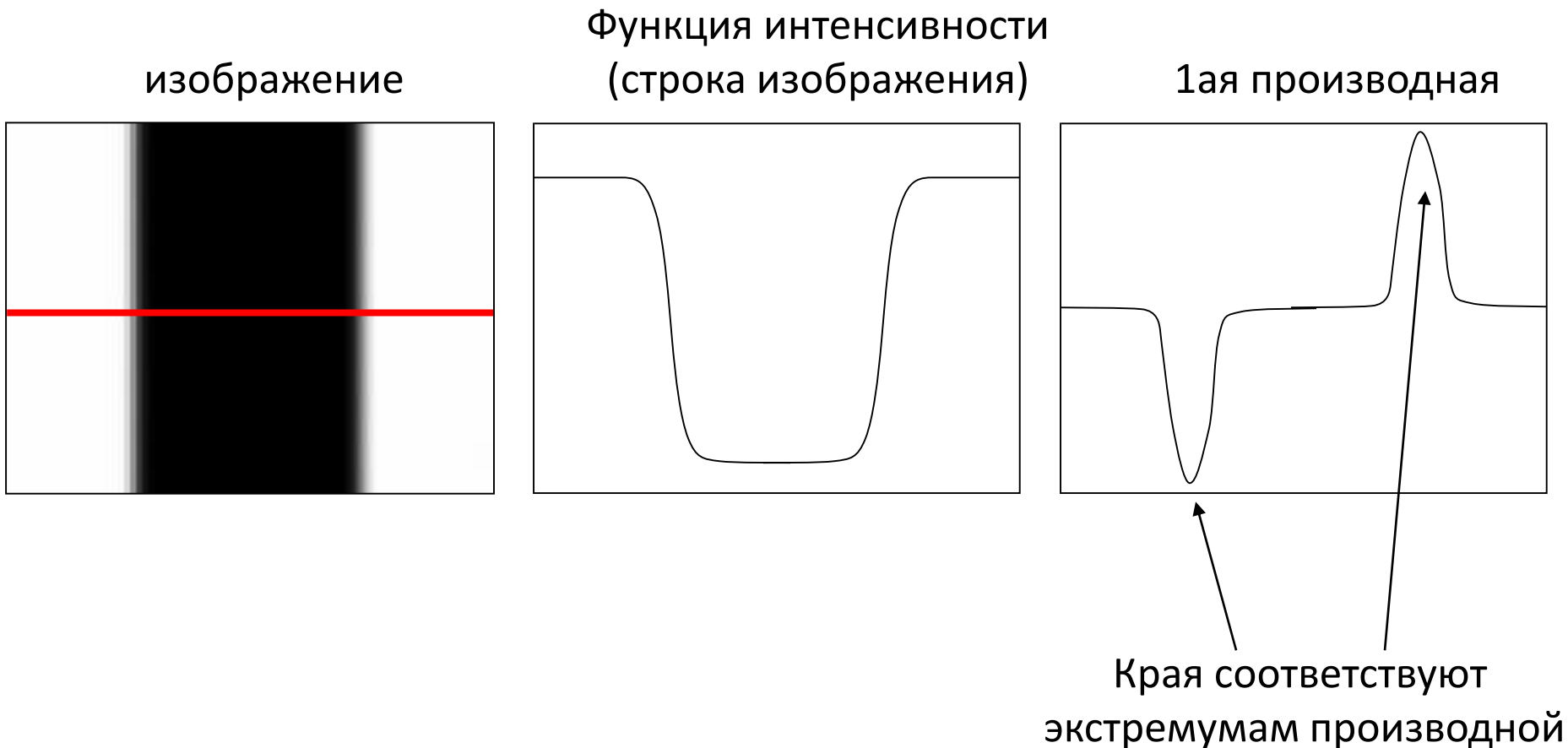
«край»:
вдоль края
изменений нет



«уголок»:
изменения при
перемещении
в любую сторону

Описание «края»

- Край – это точка резкого изменения значений функции интенсивности изображения

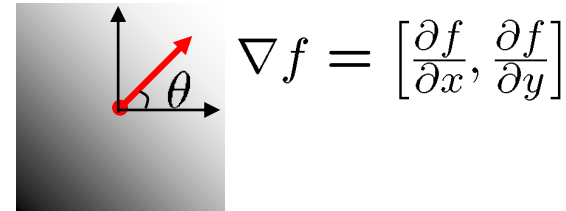
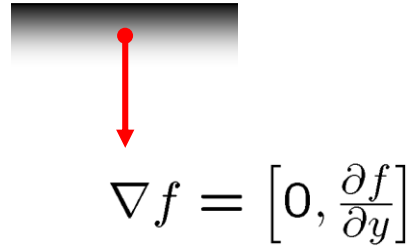
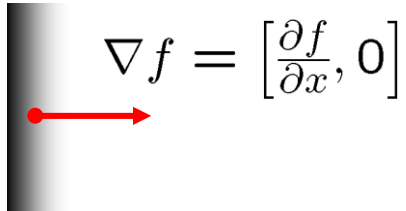


Градиент изображения

Image is a function!

$$f(x, y) = \text{Image of a yellow fish}$$

- Градиент изображения: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$



- Градиент направлен в сторону наибольшего изменения интенсивности
- Направления градиента задается как: $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$
- Величина (норма) градиента: $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2}$

Вычисление градиента

- Приближенное вычисление градиента

- Свертка по функциям:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Роберта Собеля

Матрица



Входное изображение

12	14	41
43	84	24
2	1	43



Матрица

0,5	0,75	0,5
0,75	1,0	0,75
0,5	0,75	0,5

—
—
—

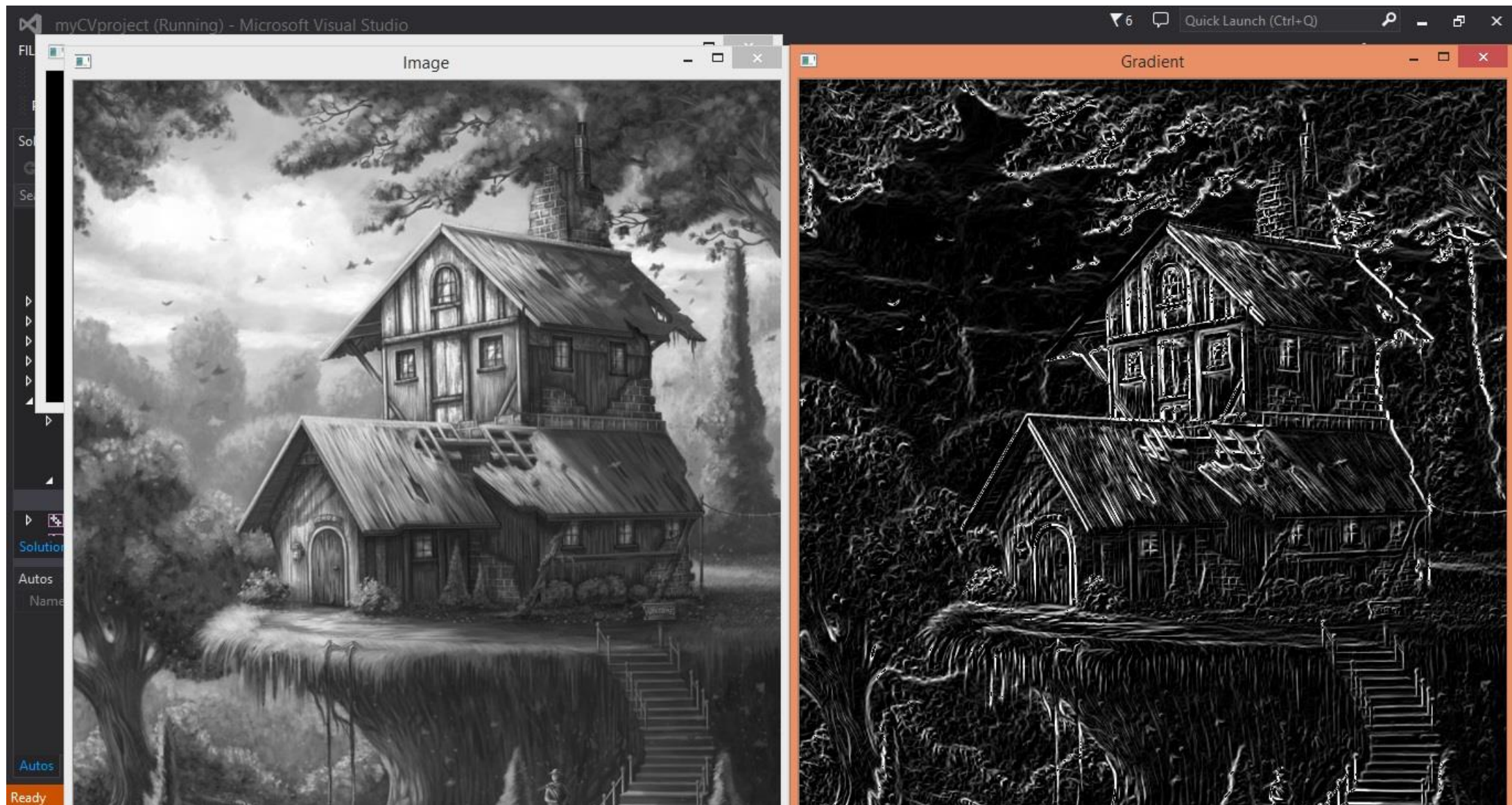
$$\begin{pmatrix} 12 * 0,5 + 14 * 0,75 + 41 * 0,5 + \\ 43 * 0,75 + 84 * 1,0 + 24 * 0,75 + \\ 2 * 0,5 + 1 * 0,75 + 43 * 0,5 \end{pmatrix} \times \frac{1}{\text{SUM}}$$

SUM = 6 -- сумма всех элементов матрицы свёртки,
т.е. коэффициент нормировки

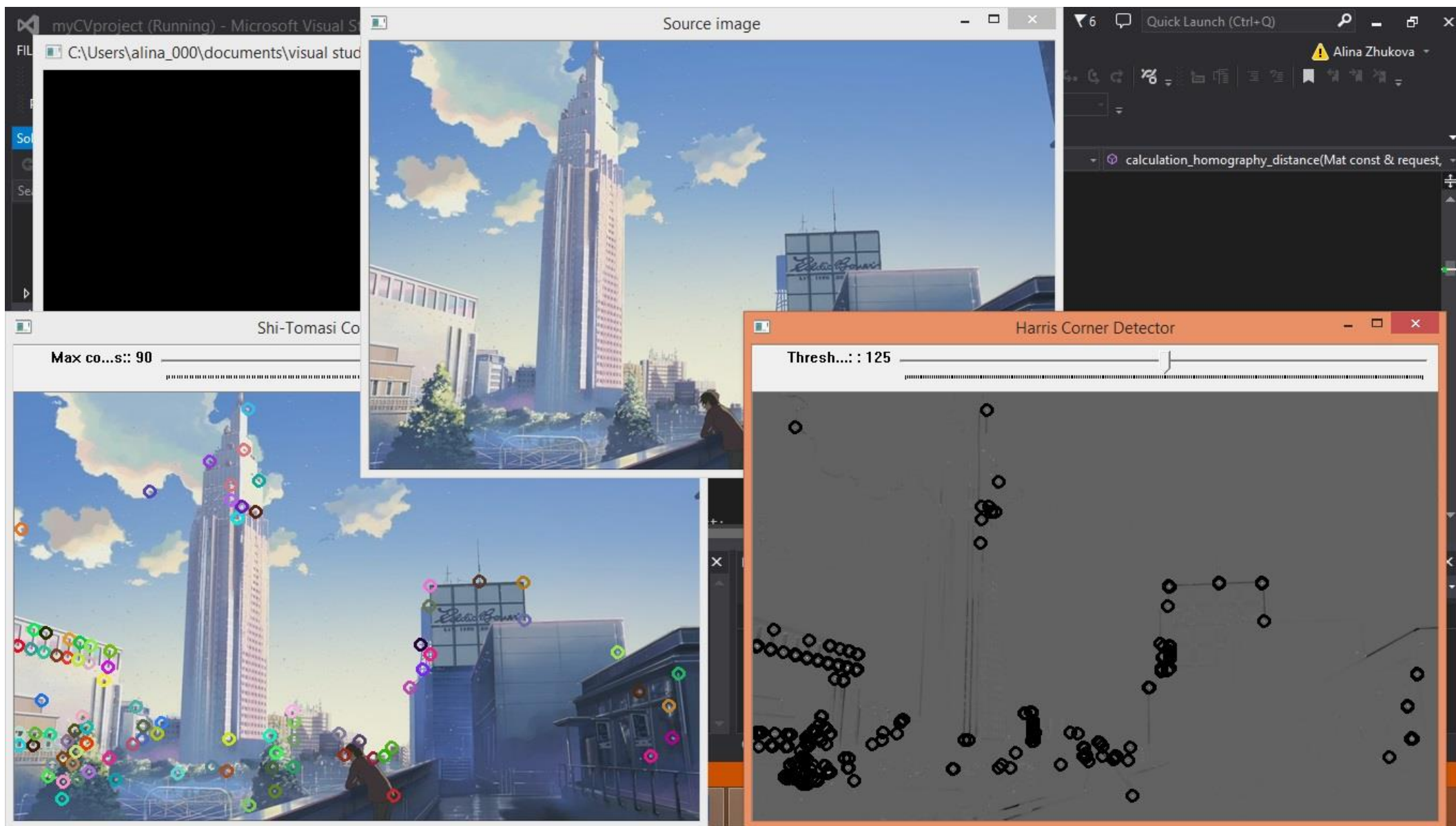
Результат

32,41667

Вычисление градиента

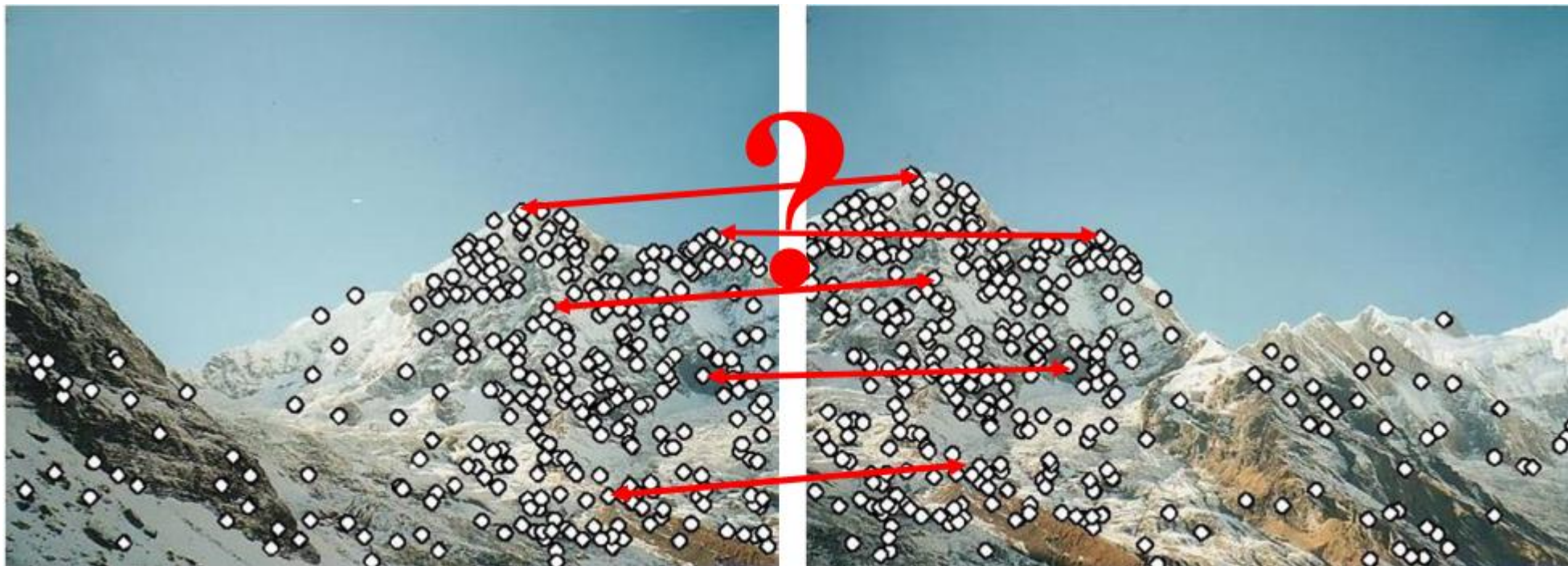


Нахождение особых точек



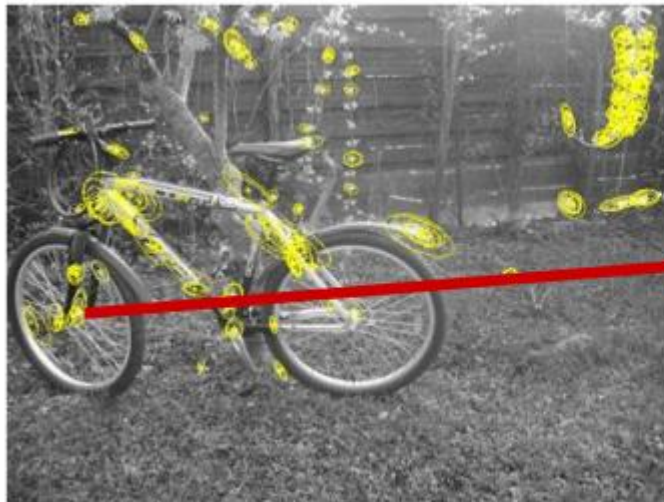
Дескрипторы

- Как сопоставить друг другу точки?
- Нужно придумать описание



Дескрипторы

- Необходимо каждую интересную точку или область описать набором параметров

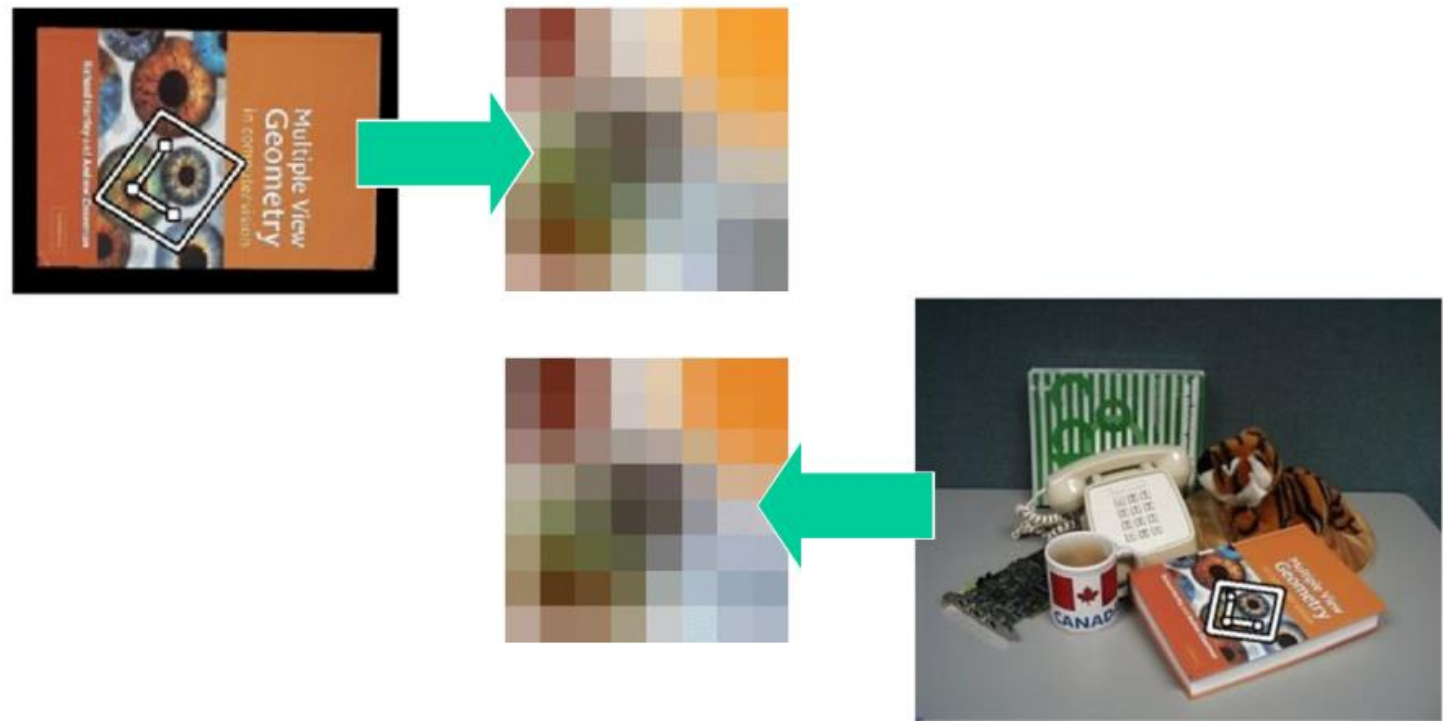


$$\mathbf{f}_n = (f_{n,1}, \dots, f_{n,j})^T$$

- Дескрипторы должны быть:
 - Простыми – представление должно быть быстро вычислимым.
 - Уникальными – разные точки должны иметь разное представление.
 - Локальными – как и сама особая точка ее представление должно зависеть лишь от небольшой окрестности.
 - Инвариантными к максимально большому числу преобразований.

Дескрипторы

- Простейший случай – вектор интенсивностей пикселей окрестности
- Сравниваем, как обычные изображения (L1 – SAD, L2 – SSD)
- Инвариантность – только сдвиг



Дескрипторы

Недостатки простой окрестности:

- Небольшие сдвиги приводит к существенному изменению
- Детектор инвариантен к повороту, а дескриптор - нет

Нужны:

- Инвариантность к поворотам
- Инвариантность к небольшим сдвигам
- Желательно: устойчивость к другим геометрическим искажениям

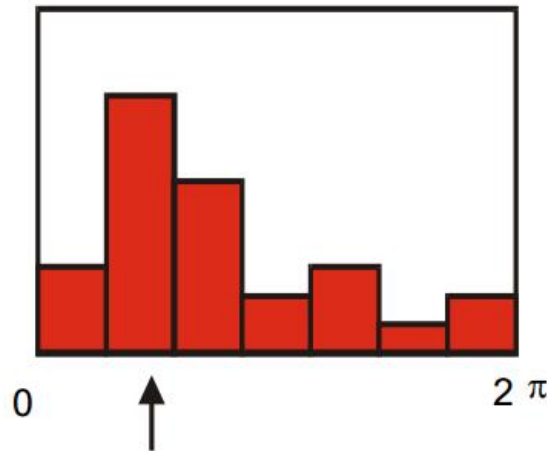
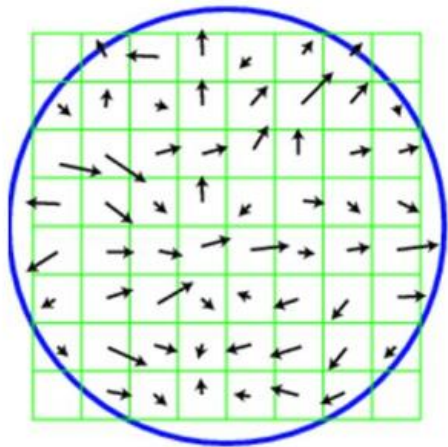


SIFT

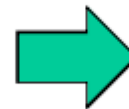
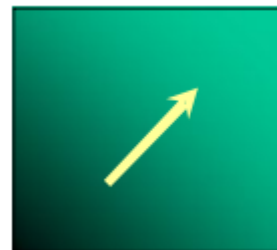
- Scale-Invariant Feature Transform:
 - Детектор DoG
 - Определение положения и масштаба особенности
 - Ориентация
 - Определение доминантной ориентации по градиентам
 - Deskриптор
 - Использование статистик по направлению градиентам
- Устойчив к изменениям освещенности и небольшим сдвигам

Детекция и ориентация

- Идея: найти основное (доминантное) направление градиентов пикселей окрестности



- Повернуть фрагмент так, чтобы доминантное направление градиента было направлено вверх



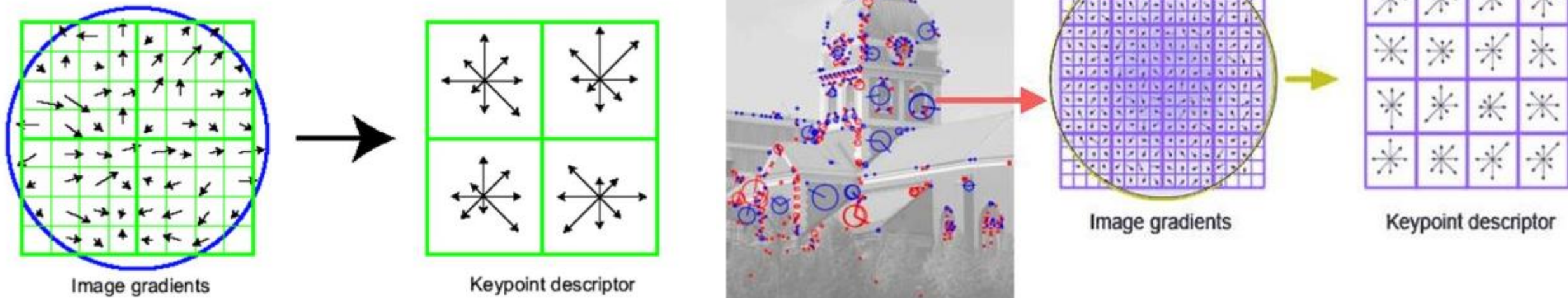
- Если локальных максимумов несколько – считаем, что несколько точек с разной ориентацией

Окрестность особой точки

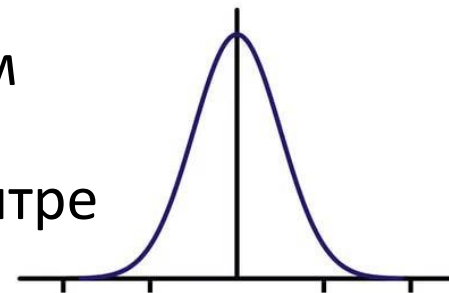


- Для каждой найденной особенности теперь знаем масштаб и ориентацию
- Выберем соответствующую прямоугольную окрестность (Rotation Invariant Frame)

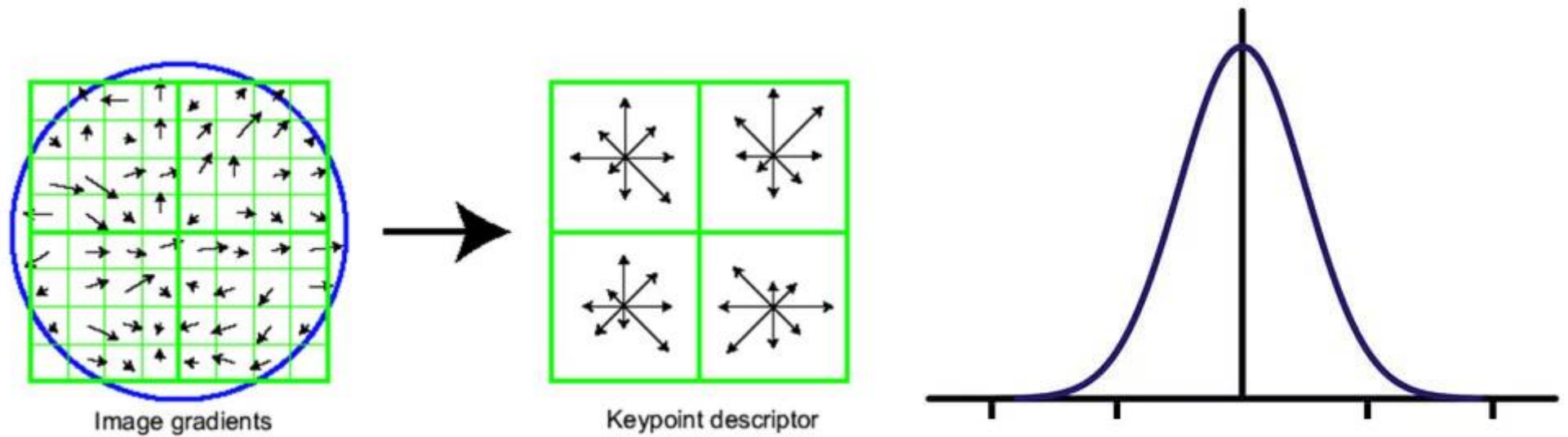
Гистограммы градиентов



- Вычисляем градиент в каждом пикселе
- Делим изображение на блоки
- Строим гистограммы направлений градиентов по прямоугольным областям
- Вклад каждого пикселя взвешиваем по гауссиане с центром в центре окрестности
- Обычно – сетка 4x4, в каждой клетке гистограмма с 8ю ячейками
- Стандартная длина вектора-дескриптора – 128 ($4 \cdot 4 \cdot 8$)
- Сравниваем как вектор (разные метрики)



SIFT



- Детектор SIFT устойчив к изменениям освещения, небольшим сдвигам
- Вся схема SIFT (детектор, выбор окрестностей, дескриптор) оказалась очень эффективным инструментом для анализа изображений

Визуальные слова

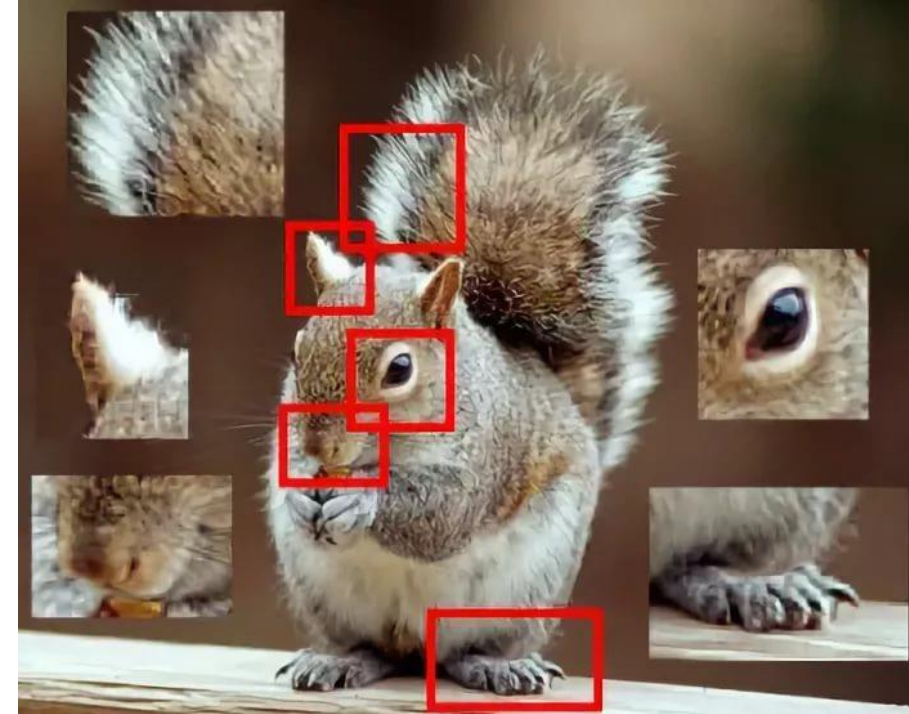


Что в нашем случае «слово» и «словарь»?

- «Визуальное слово» – часто повторяющийся фрагмент изображения («visual word»)
- В изображении визуальное слово может встречаться только один раз, может ни разу, может много раз

Визуальный словарь

- Словарь – набор фрагментов, часто повторяющихся в коллекции изображений
- Как составить словарь?
- Составить большой список всех фрагментов по всей коллекции
- Разделить весь список на похожие группы (кластеризовать)
- Будем считать все фрагменты в одной группе – «экземплярами» одного и того же слова



Мешок визуальных слов

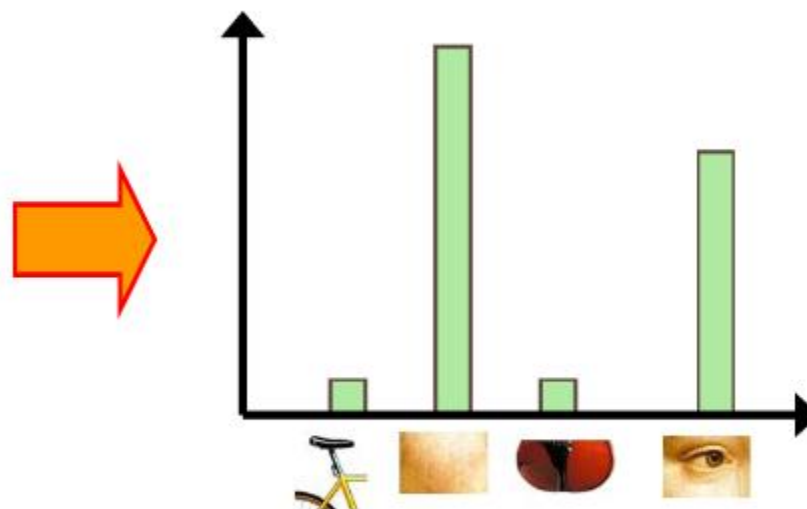
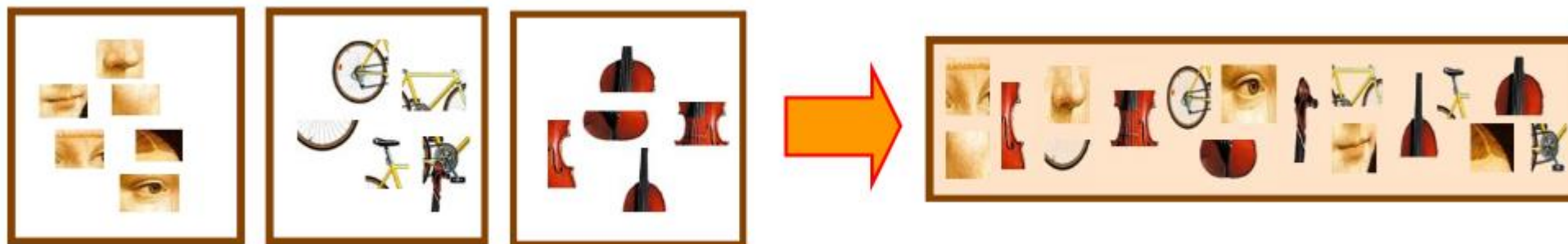
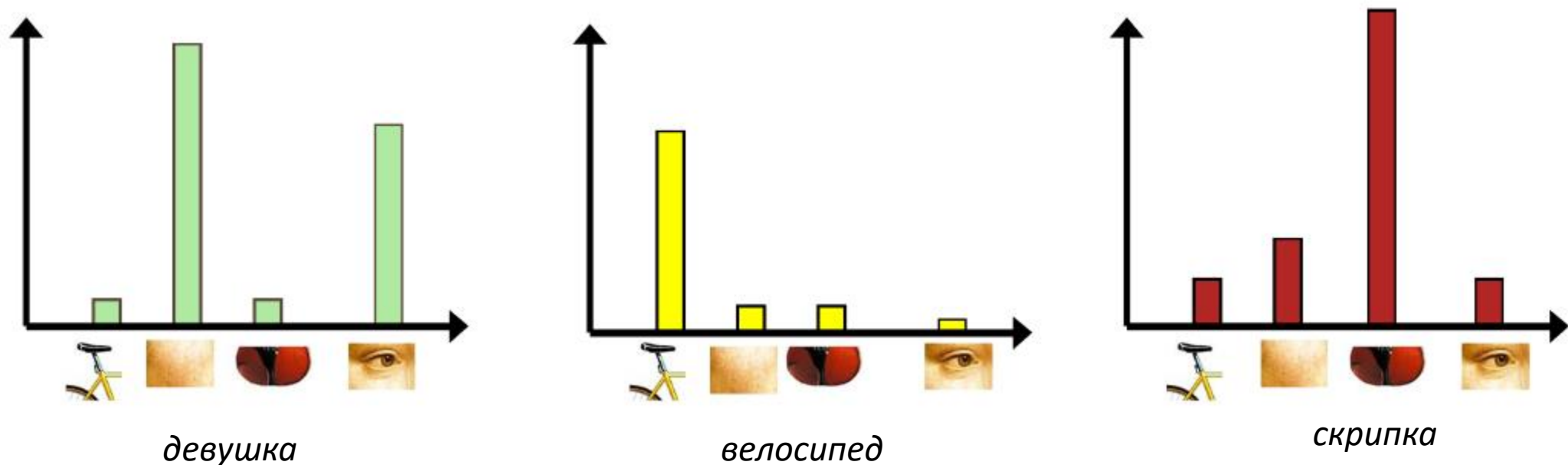


Схема метода «мешок слов»

1. Извлечение особенностей (детекторы, дескрипторы)
2. Составление «визуального словаря»
3. Сопоставление особенностей изображения по словарю
4. Описание изображения частотами «визуальных слов»



Свойства «мешка слов»



- Визуальные слова – «характерные» фрагменты изображения
- Детектор и дескриптор SIFT дают инвариантность к локальным поворотам и масштабированию
- Переход к частотам отбрасывает пространственную информацию и обеспечивает инвариантность к относительному расположению фрагментов



лицо



НЕ лицо



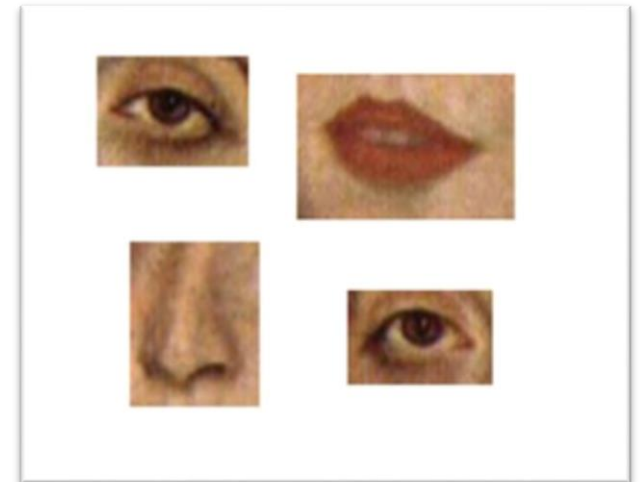
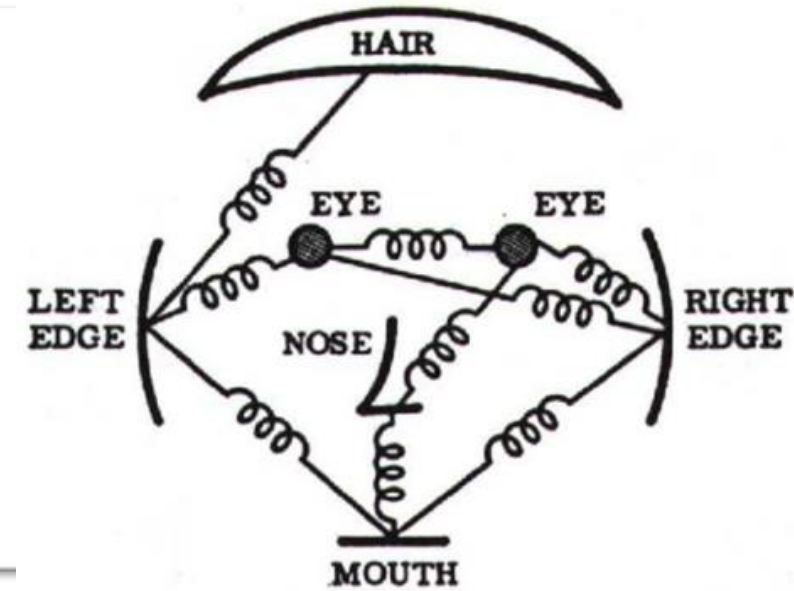
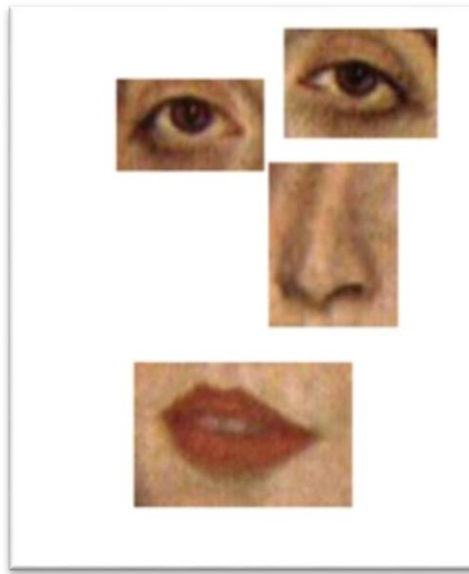
Оценка «мешка слов»

Плюсы

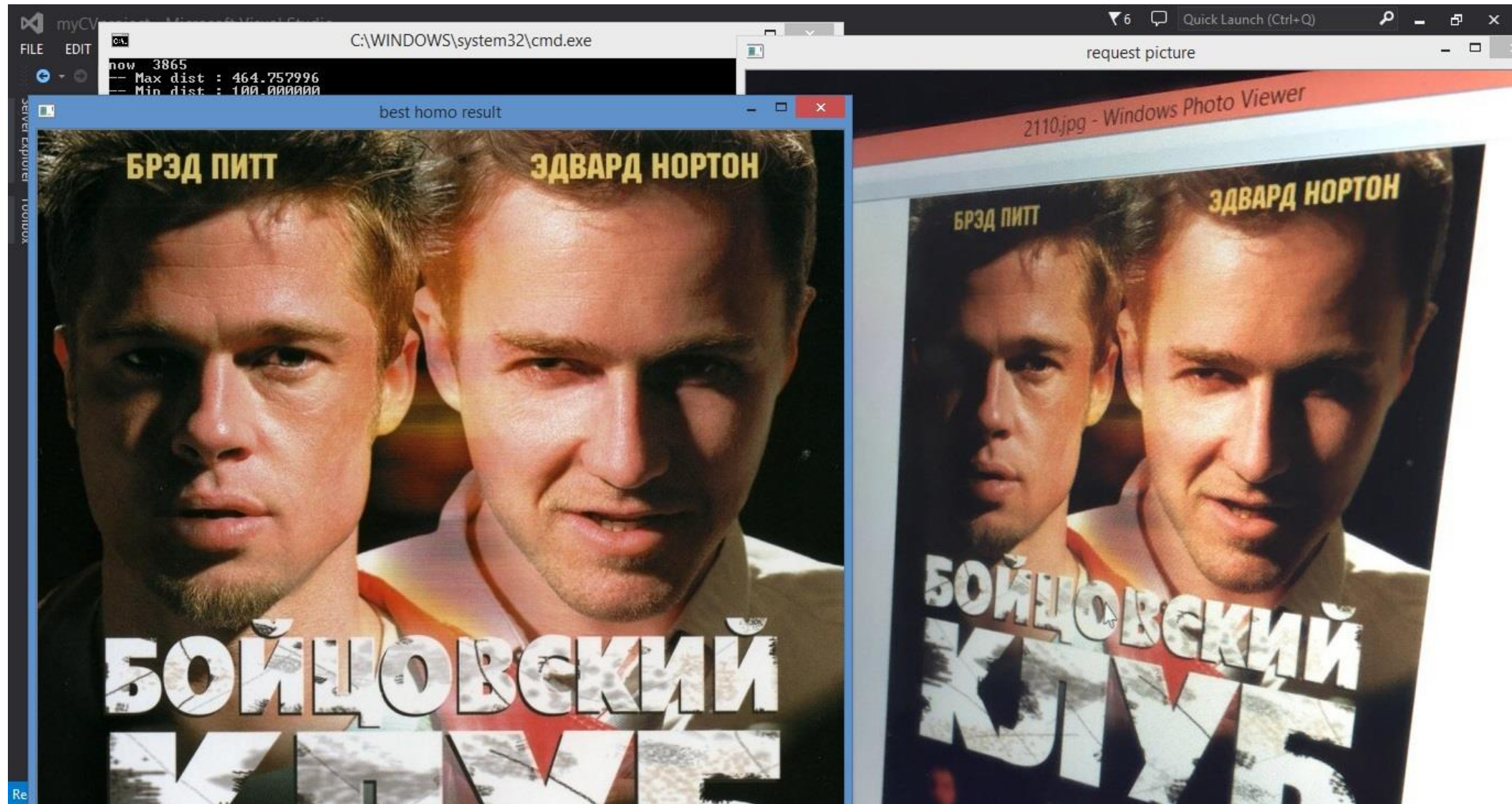
- Не учитывается (явно) пространственное расположение
⇒ инвариантность к положению и ориентации в изображении
- Вектор фиксированной длины
⇒ применение стандартных методов машинного обучения

Минусы

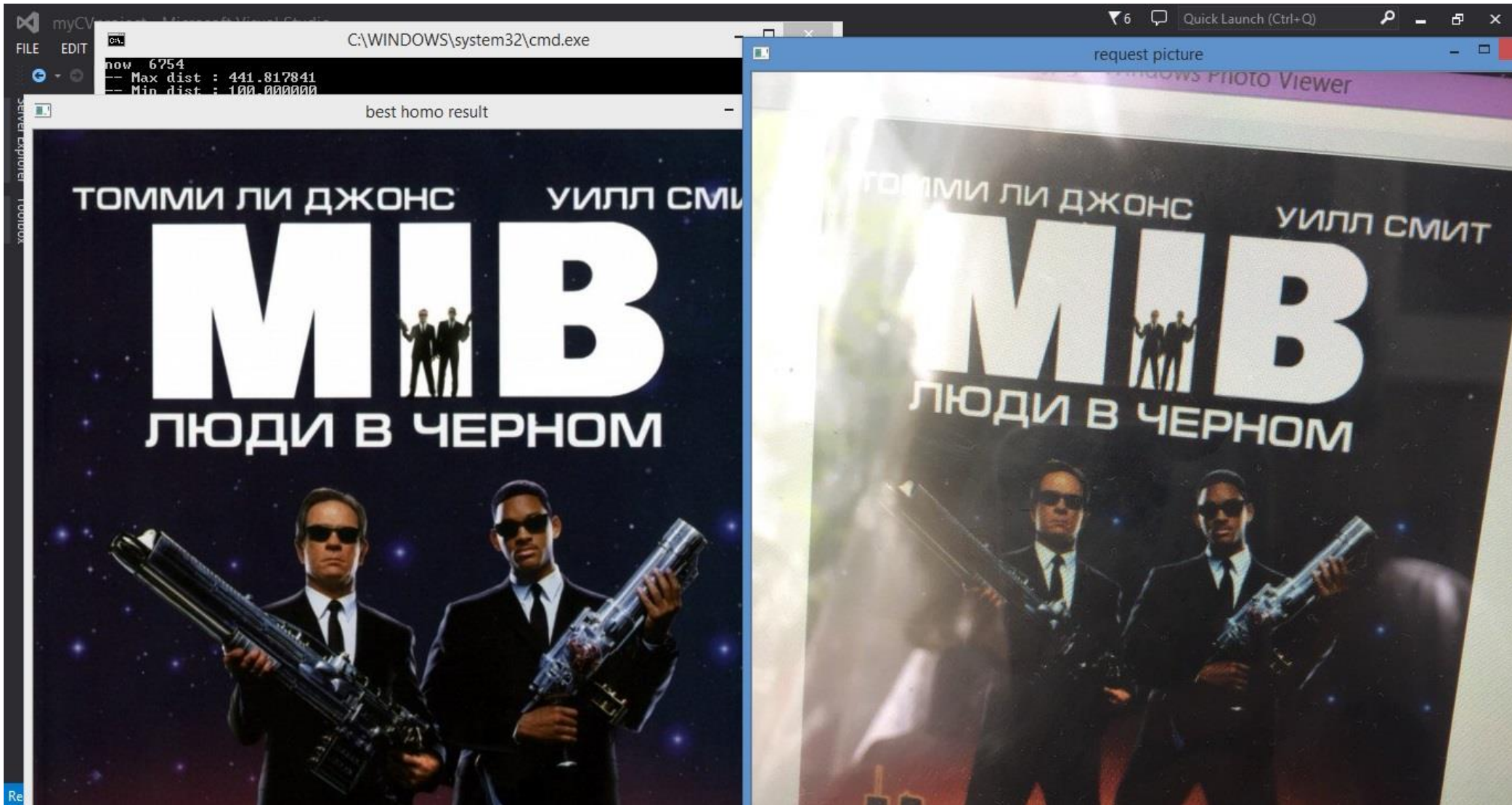
- Не учитывается пространственное расположение
⇒ хуже различающая способность (discriminative power)



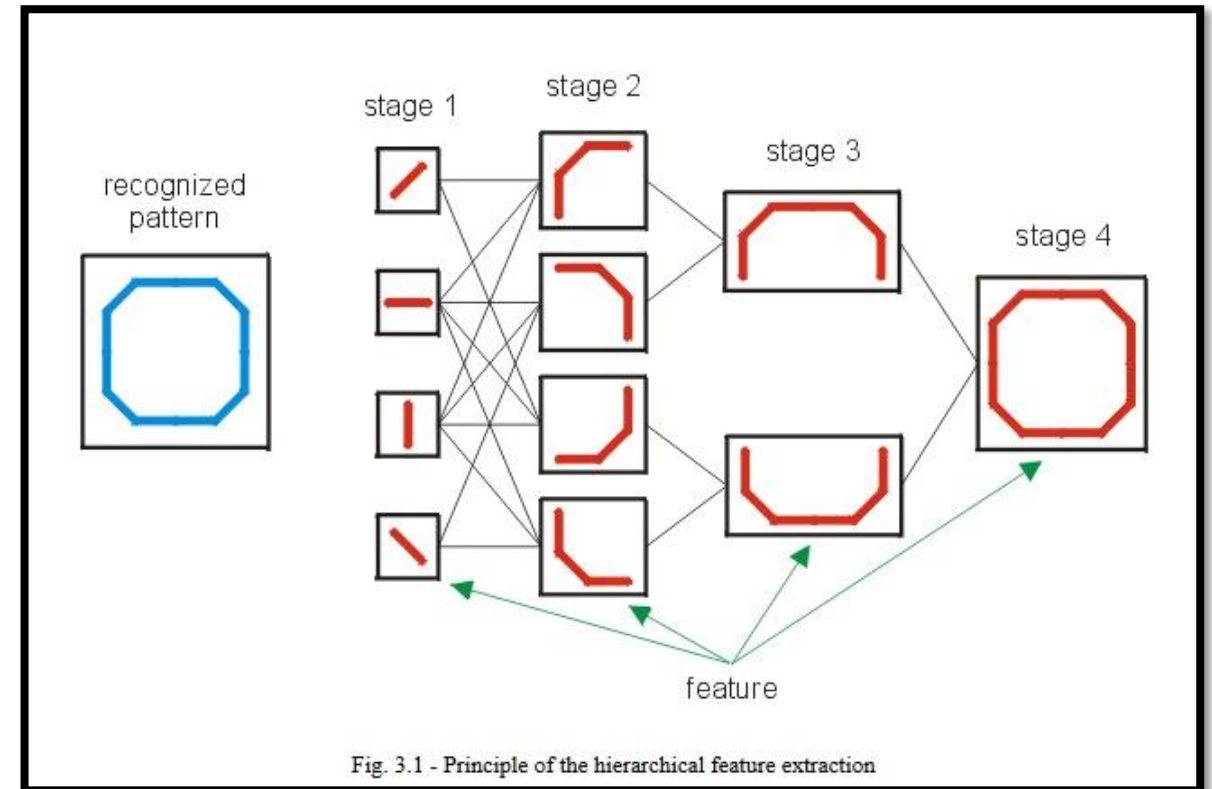
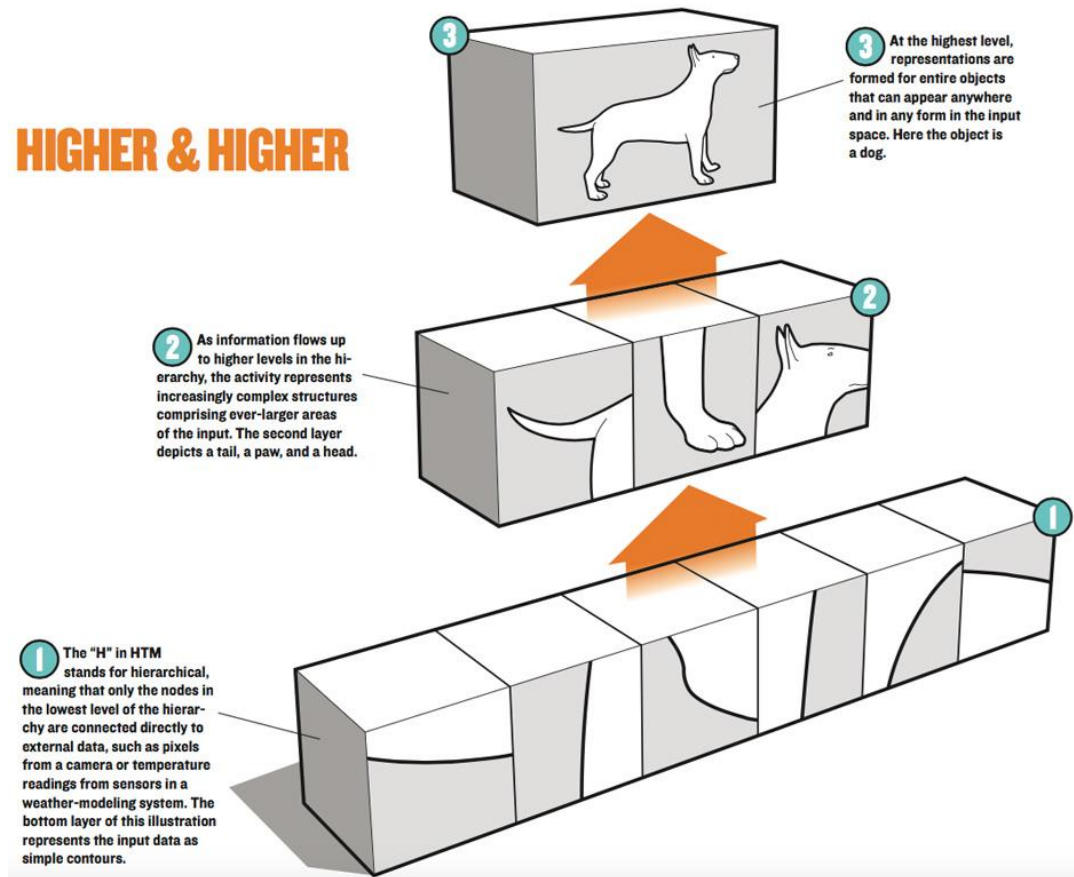
Пример решение задачи поиска похожих



Пример решение задачи поиска похожих



Современные методы – свёрточные нейронные сети



Сравнение SIFT и CNN (биология)

SIFT

- Процесс извлечения особенностей сильно отличается от процессов, протекающих в мозге

CNN

- Процесс извлечения и обобщения особенностей от простых к сложным очень похож на процесс распознавания в зрительной коре

Сравнение SIFT и CNN (сложность и потребление ресурсов)

SIFT

- Более простая архитектура, меньшее количество параметров
- Требуется меньше мощности для обработки
- Память используется для хранения особенностей изображения
- Небольшая обучающая выборка
- Быстрый

CNN

- Нужен опыт, чтобы принимать решения
- Требуется много мощности для обработки во время обучения
- Память используется для хранения весов сети
- Чем больше обучающая выборка, тем лучше
- Медленнее, чем SIFT

	Good	Bad
SIFT	<ul style="list-style-type: none">• Решает задачи распознавания• Прост в реализации• Быстрый	<ul style="list-style-type: none">• Плохая обобщающая способность• Не устойчив к нелинейным преобразованиям
CNN	<ul style="list-style-type: none">• Решает задачи классификации• Высокая биологическая правдоподобность• Хорошо обобщает данные	<ul style="list-style-type: none">• Высокая вычислительная мощность• Большие обучающие выборки• Необходимость устанавливать параметры

Список источников

1. МГУ введение в компьютерное зрение
[<https://www.lektorium.tv/course/22847>]
2. Особые точки [<http://my-it-notes.com/2013/05/03-features-characteristic-points/>]
3. Свёртка [<https://habrahabr.ru/post/142818/>]
4. SIFT vs CNN [[https://tams.informatik.uni-hamburg.de/lehre/2015ws/seminar/ir/pdf/slides/JosipJosifovski-Object Recognition SIFT vs Convolutional Neural Networks.pdf](https://tams.informatik.uni-hamburg.de/lehre/2015ws/seminar/ir/pdf/slides/JosipJosifovski-Object%20Recognition%20SIFT%20vs%20Convolutional%20Neural%20Networks.pdf)]

Практическая реализация

- Нахождение градиента, использование детекторов и дескрипторов, сопоставление особых точек
- Библиотека opencv
- https://github.com/linazh/cv_practice

