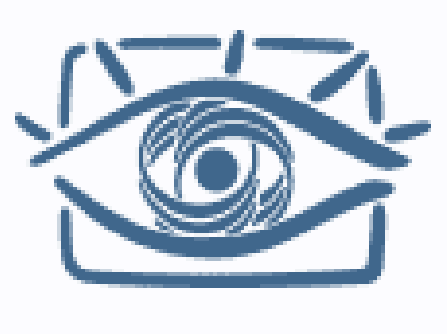


Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №1

«Введение в предмет. Цифровое изображение. Свет и цвет»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

9 сентября 2019 года

План лекции

- Что такое «компьютерное зрение»?
- Цифровое изображение
- Цветные изображения, свет и цвет, модели цвета.
- Коррекция яркости и цветопередачи в изображении

Что такое «компьютерное зрение»?

Что такое компьютерное зрение?

Задача зрения: понять, что находится на изображении

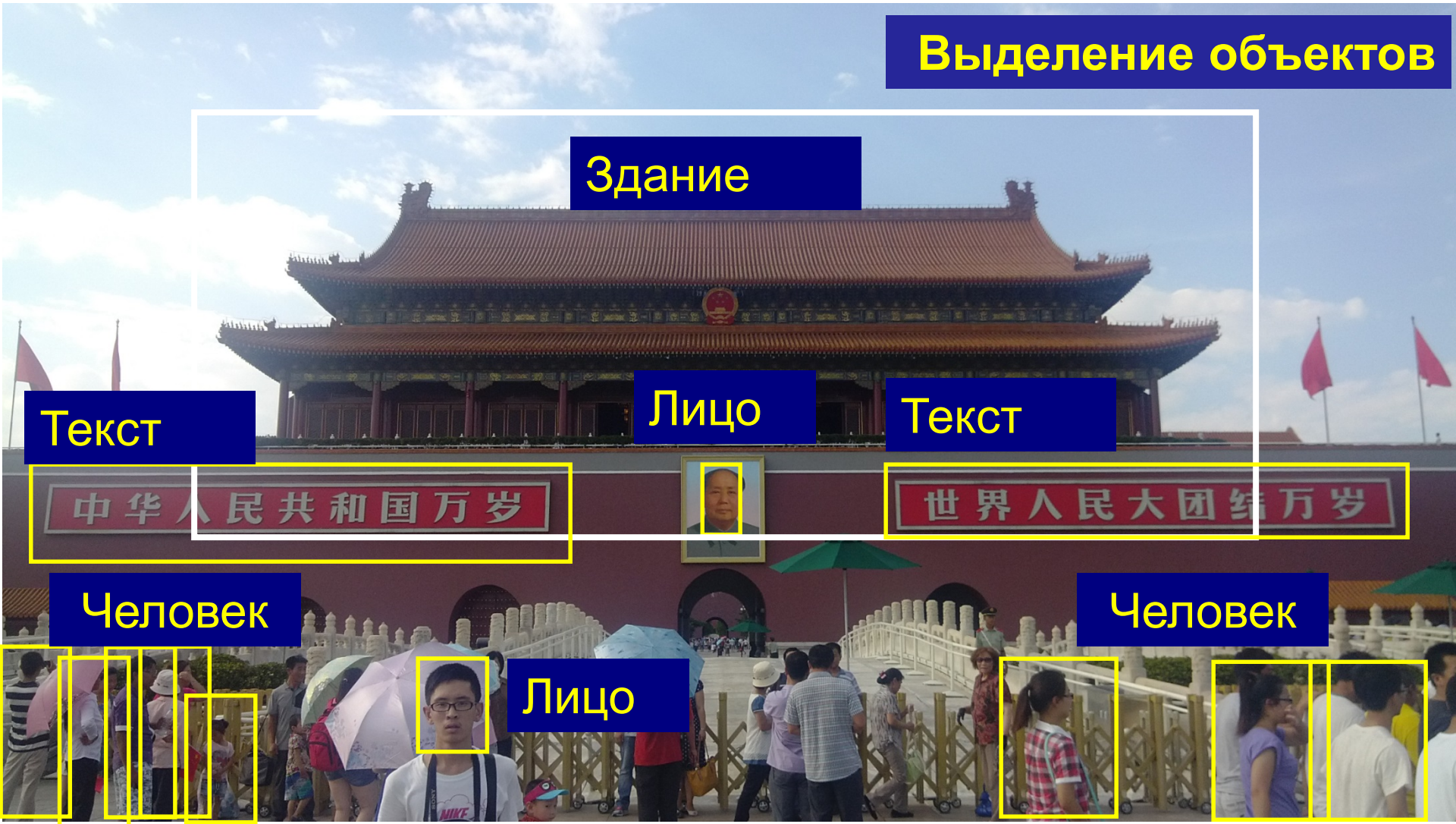
Компьютерное зрение: построение компьютерной модели системы зрения

Компьютерное зрение – часть области искусственного интеллекта (AI)

Тест Тьюринга для компьютерного зрения:
Ответить на любой вопрос про изображении, на который может ответить человек.

Source: S. Narasimhan

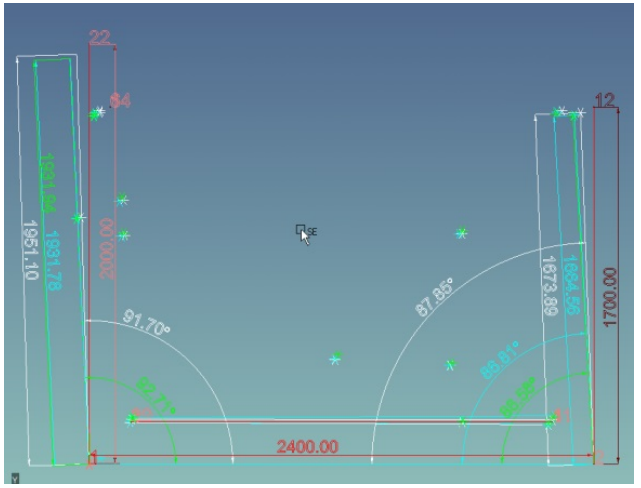
Что и где находится на изображении?



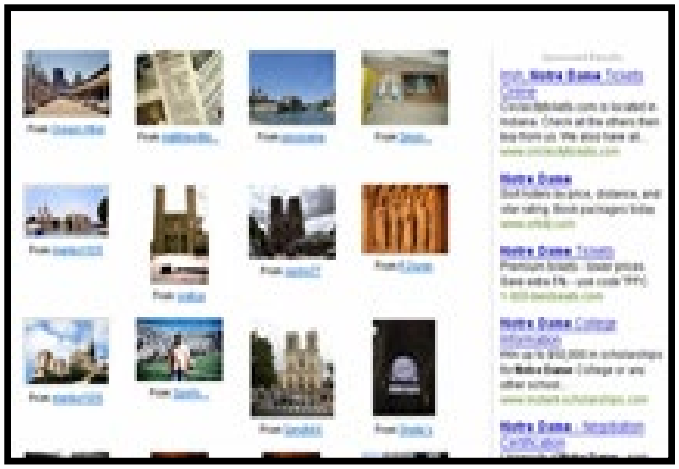
Необходимо определить, есть ли на изображении объекты заданного типа и если да, то определить их положение

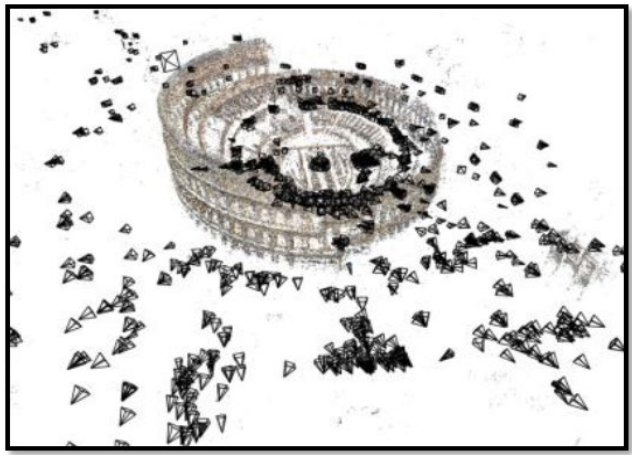
Свойства объектов и атрибуты

Какой формы? (Метрическое зрение)



Фотограмметрия





Source: Pollefeys et al.

Image-based 3D modeling

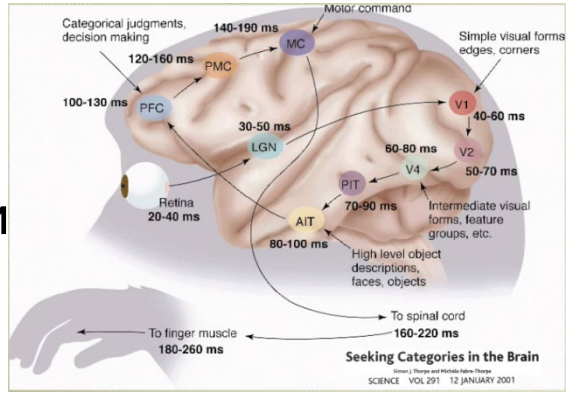
Зрение человека

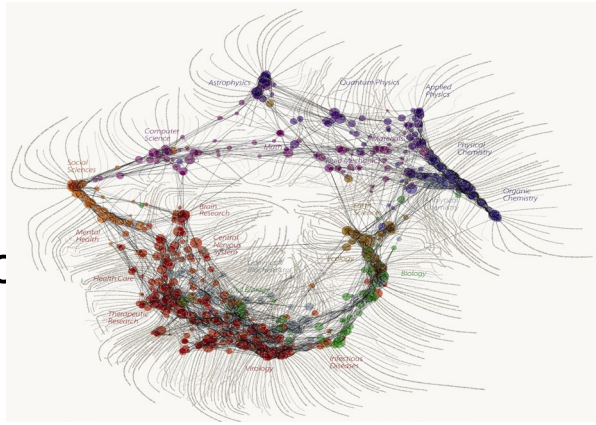
Image source: J. Koenderink

- Мы сопоставляем наблюдения (подсказки) и априорные знания для интерпретации изображения
- По оценкам, 25% мозга занято решением задачи зрения
- Какие подсказки и априорные знания вы можете придумать/вспомнить?

Давид Марр (1970е)

- «Primal sketch»
  - Низкоуровневые («low-level») свойства изображения: направленные края, отрезки и т.д.
- «2.5D sketch»
  - Упорядочивание по глубине (бинокулярное стерео), учёт текстуры и т.д.
- «3D model»
  - Распознавание объектов и представление о 3х мерном мире





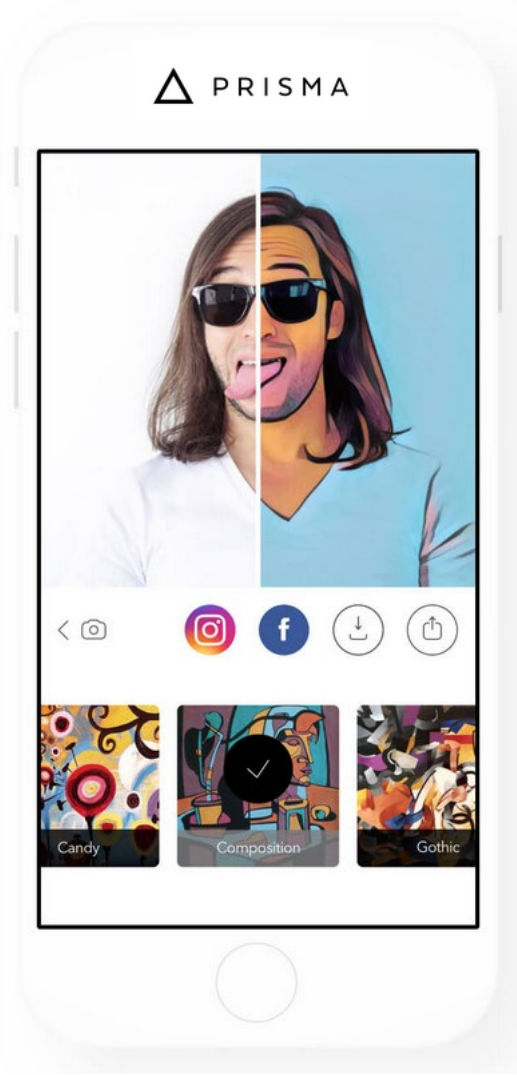
“The Marr Prize” – главная премия в области компьютерного зрения

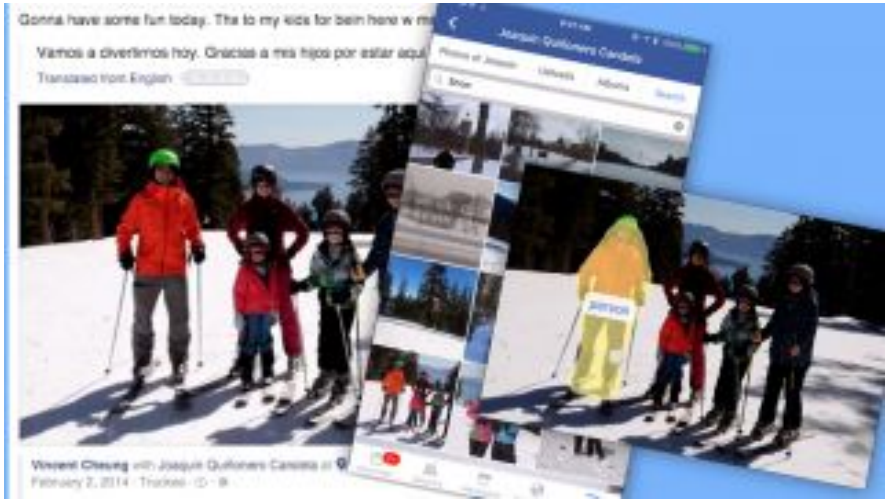
Детектор лиц Viola-Jones (2001)

Алгоритм Viola-Jones – первый быстрый и надежный алгоритм поиска лиц. Демонстрация силы машинного обучения.

Source: S. Seitz

Глубинное обучение (Deep learning)



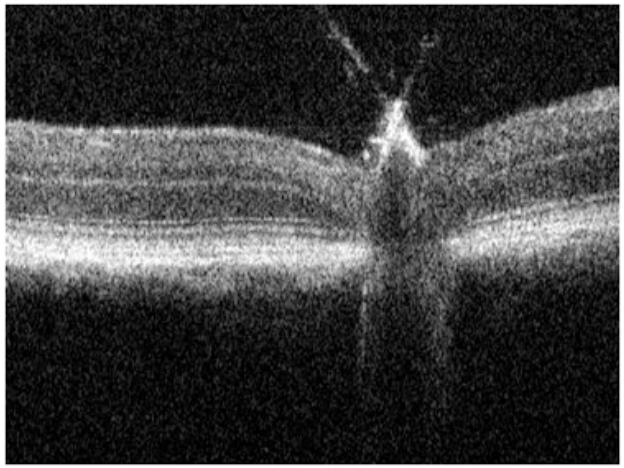


Цифровое изображение

Изображение

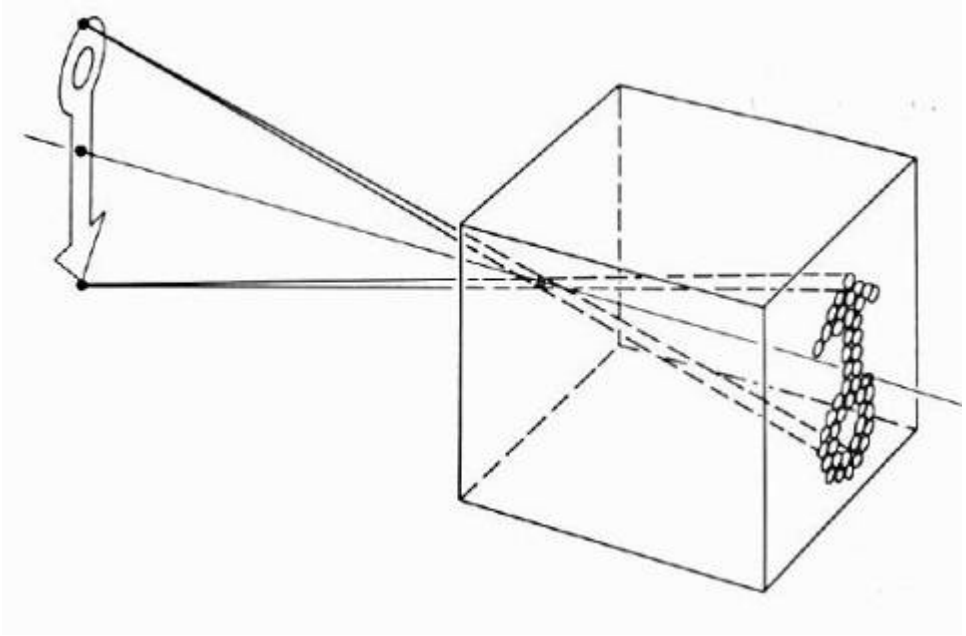
Изображение оптическое – картина, получаемая в результате прохождения через оптическую систему лучей, распространяющихся от объекта, и воспроизводящая его контуры и детали.

(Физический энциклопедический словарь.)



Оптические системы могут быть очень разными!

Камера-обскура



- Простейшее устройство для получения изображений
- Основная модель оптической системы получения изображений

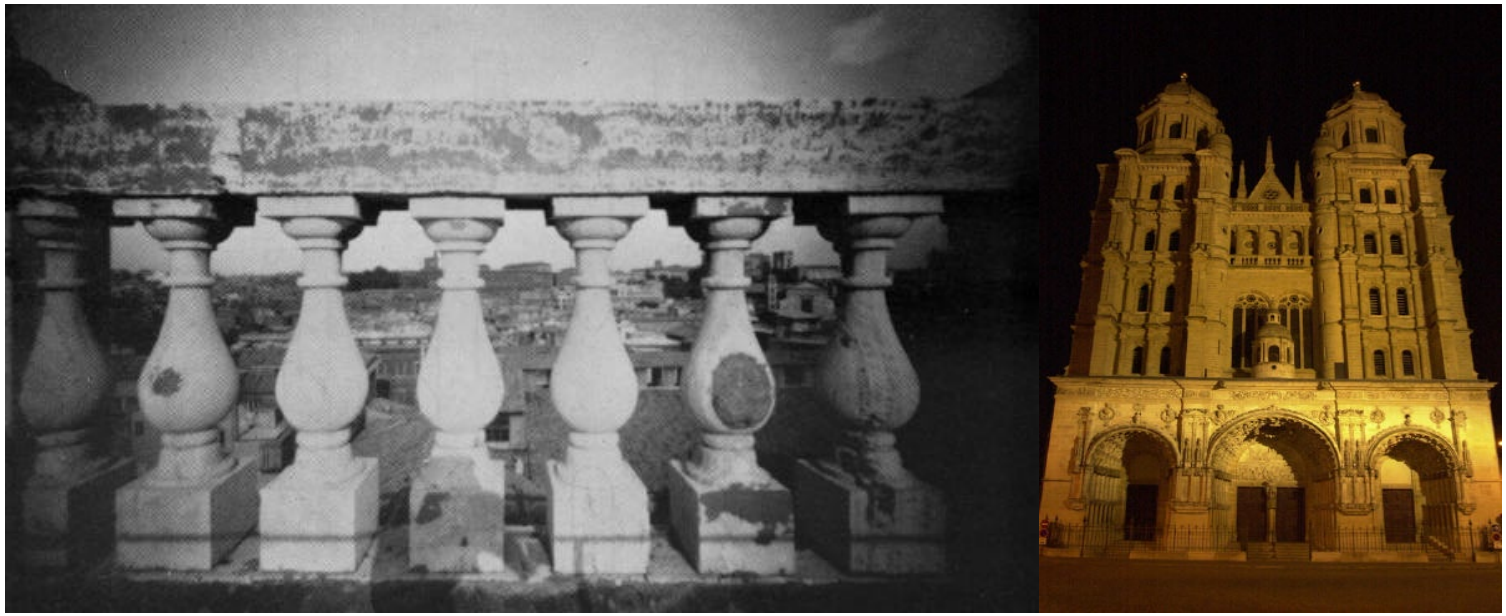
Математическая модель – перспективная проекция:

- Пучок лучей проходит через одну точку (точечное отверстие)
  - Эта точка называется «центр проекции» (фокальная точка / **focal point**)
- Изображение формируется на картинной плоскости (**image plane**)

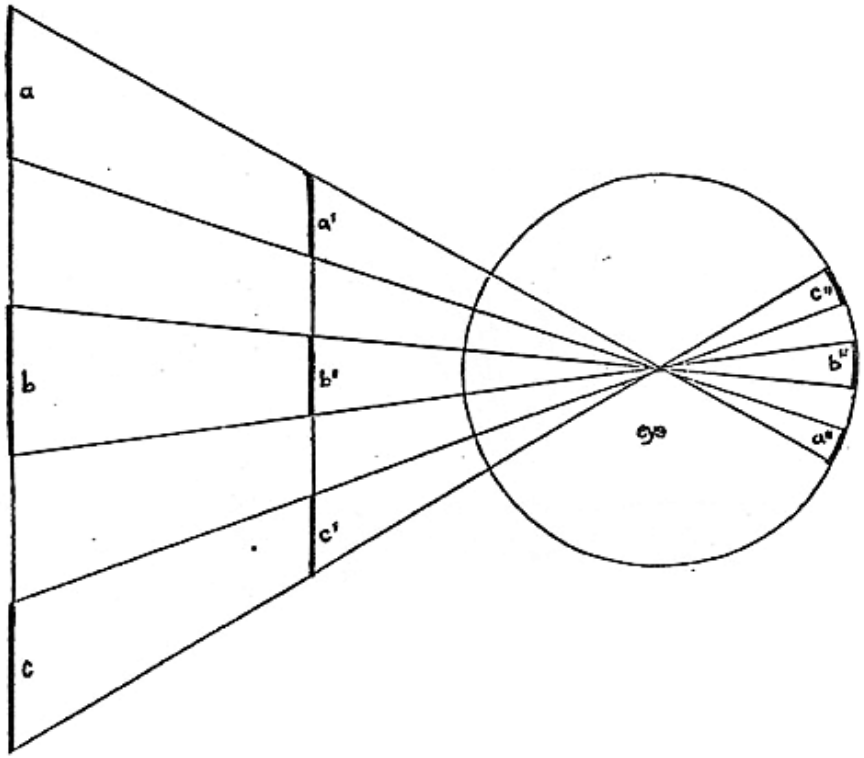
Slide by Steve Seitz

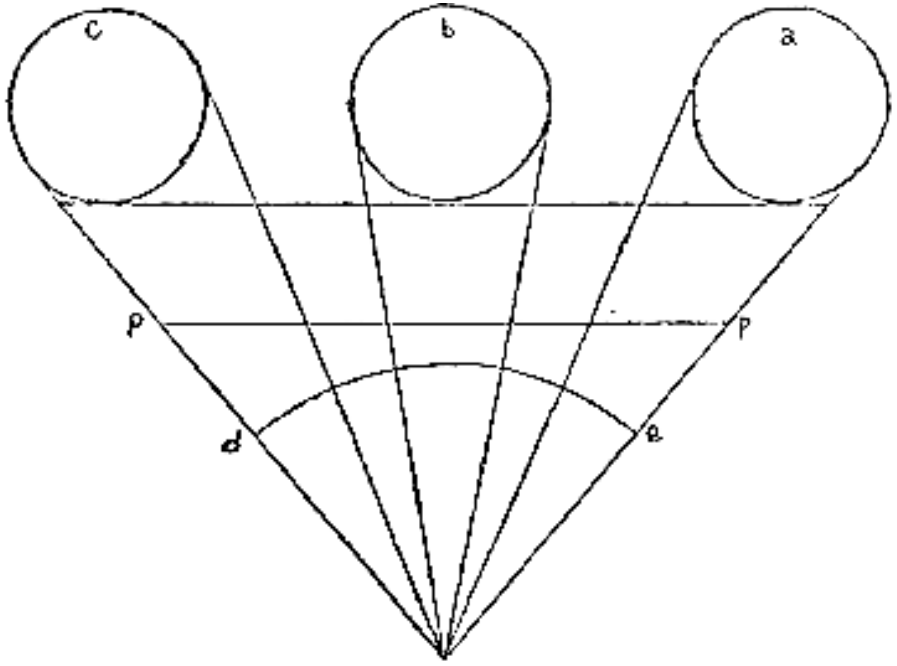
Перспективные искажения

- Перспективная проекция приводит к искажениям свойств объектов
  - Сходящиеся вертикальные линии
  - Крайние колонны кажутся толще
- Эти искажения вызваны не погрешностью оптической системы!



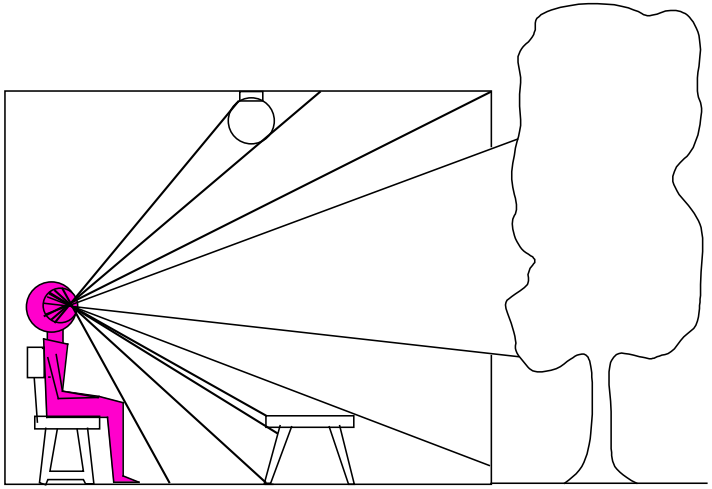
Перспективные искажения



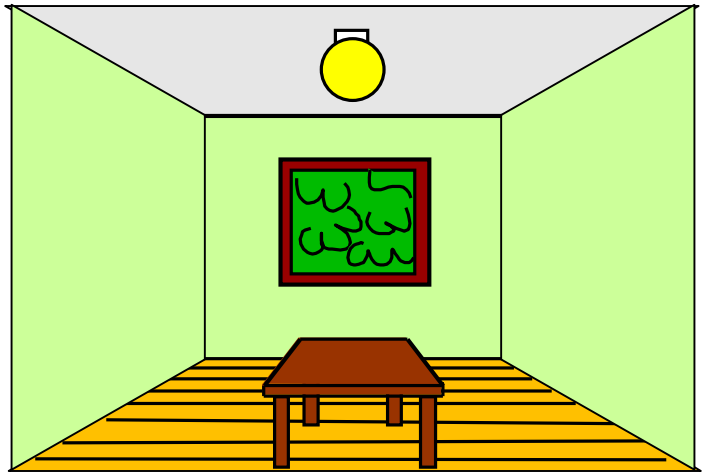


Машина Понижения Размерности

3D мир



2D картина



Что мы теряем?

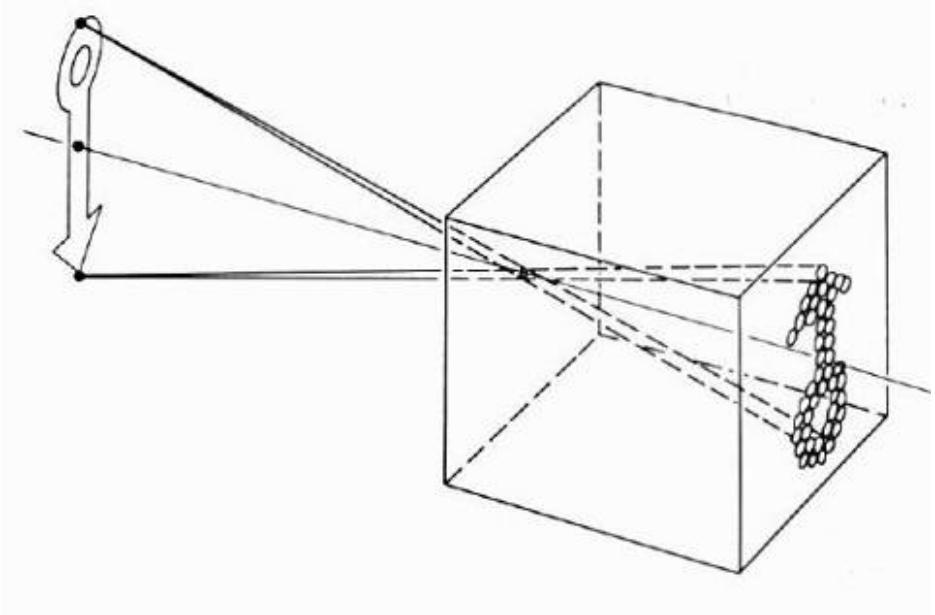
- Углы
- Расстояния и длины

Подробнее геометрическая модель рассматривается позже

Slide by A. Efros

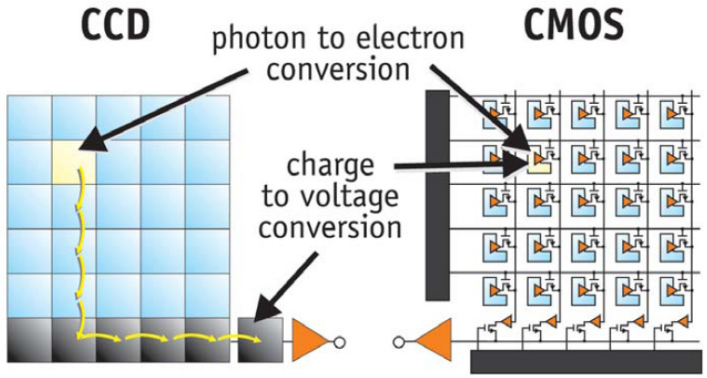
Figures © Stephen E. Palmer, 2002

Современная камера

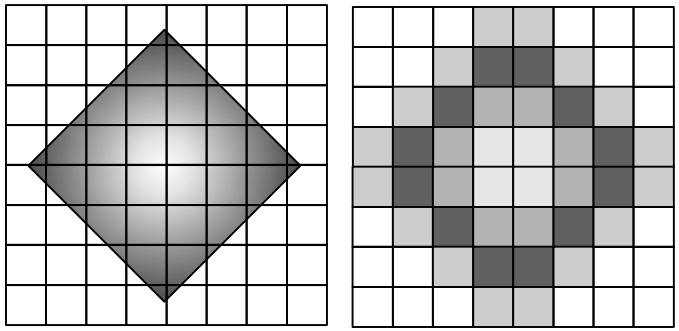


Тот же самый принцип, но с объективом и цифровой матрицей

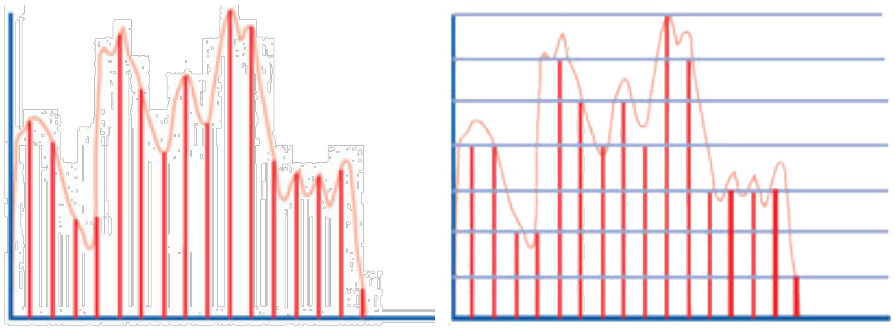
Цифровая камера - дискретизация



CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.



По пространству



По яркости

Первое цифровое фото (1957)

Разрешение
176*176 пикселов

http://listverse.com/history/top-10-incredible-early-firsts-in-photography/

Slide by S. Lazebnik

Цифровое изображение

- Цифровое полутоновое изображение – матрица $\mathrm{I} \in (b_{ij})^{n,m}$, элементами которой b_{ij} являются значения интенсивности света, измеренного на 2х мерной прямоугольной сетке
  - b (интенсивность) можно описать вещественным числом. Обычно ограничиваются интервалом $b_{ij} \in [0,1]$, где 0 (нет света), 1 (максимальная яркость)
  - Обычно b кодируем 1 байтом, т.е. $b \in [0,255]$
  - Можем использовать большую точность (10-16бит)

Свет и цвет

Как получить цветные фотографии?

Что такое цвет?

Цвет – это психологическое свойство нашего зрения, возникающее при наблюдении объектов и света, а не физические свойства объектов и света (S. Palmer, *Vision Science: Photons to Phenomenology*)

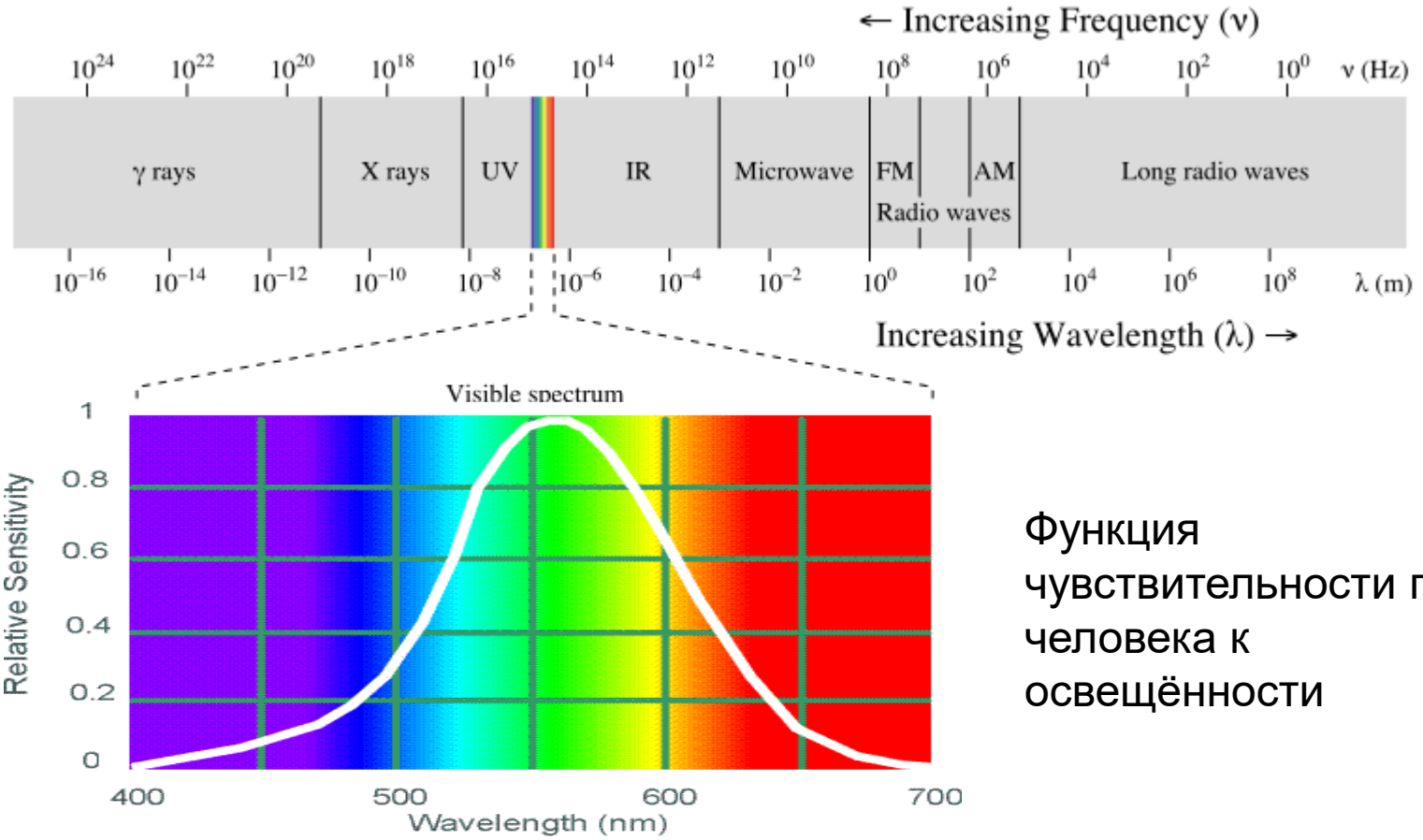
Цвет – это результат взаимодействия света, сцены и нашей зрительной системы

Восприятие света человеком изучают науки фотометрия и колориметрия

Wassily Kandinsky (1866-1944), Murnau Street with Women, 1908

Slide by S. Lazebnik

Видимый свет



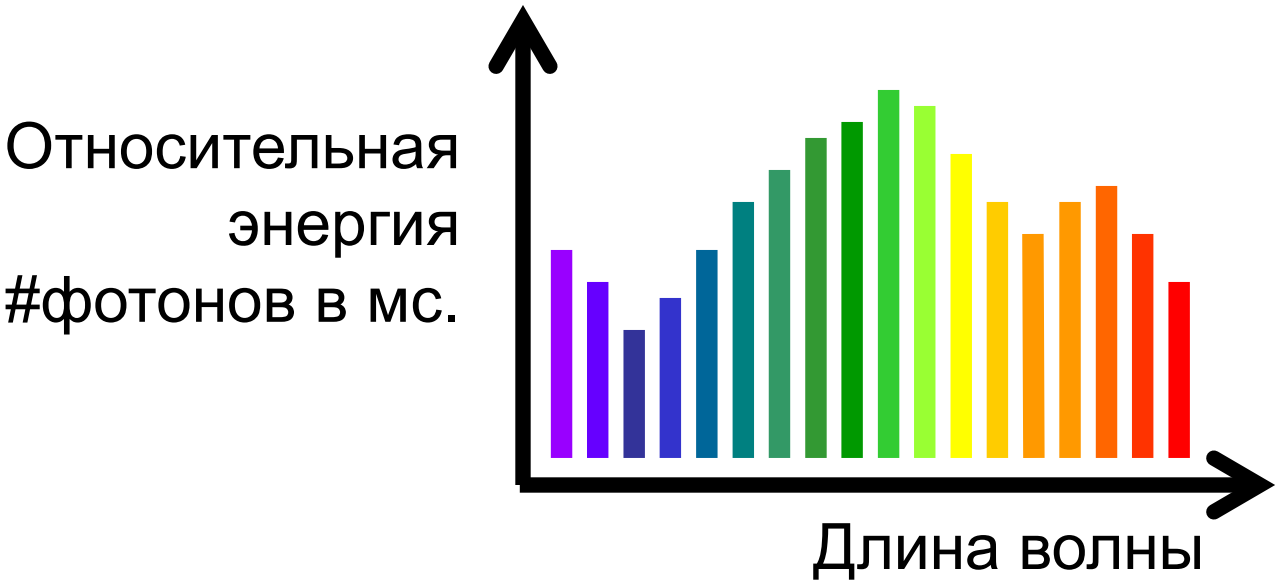
Функция чувствительности глаза человека к освещённости

- Электромагнитное излучение в диапазоне [380nm,780nm]
- Видимый свет попадает в основное «оптическое окно» земной атмосферы (~46% энергии)

Slide by S. Lazebnik

Физика света

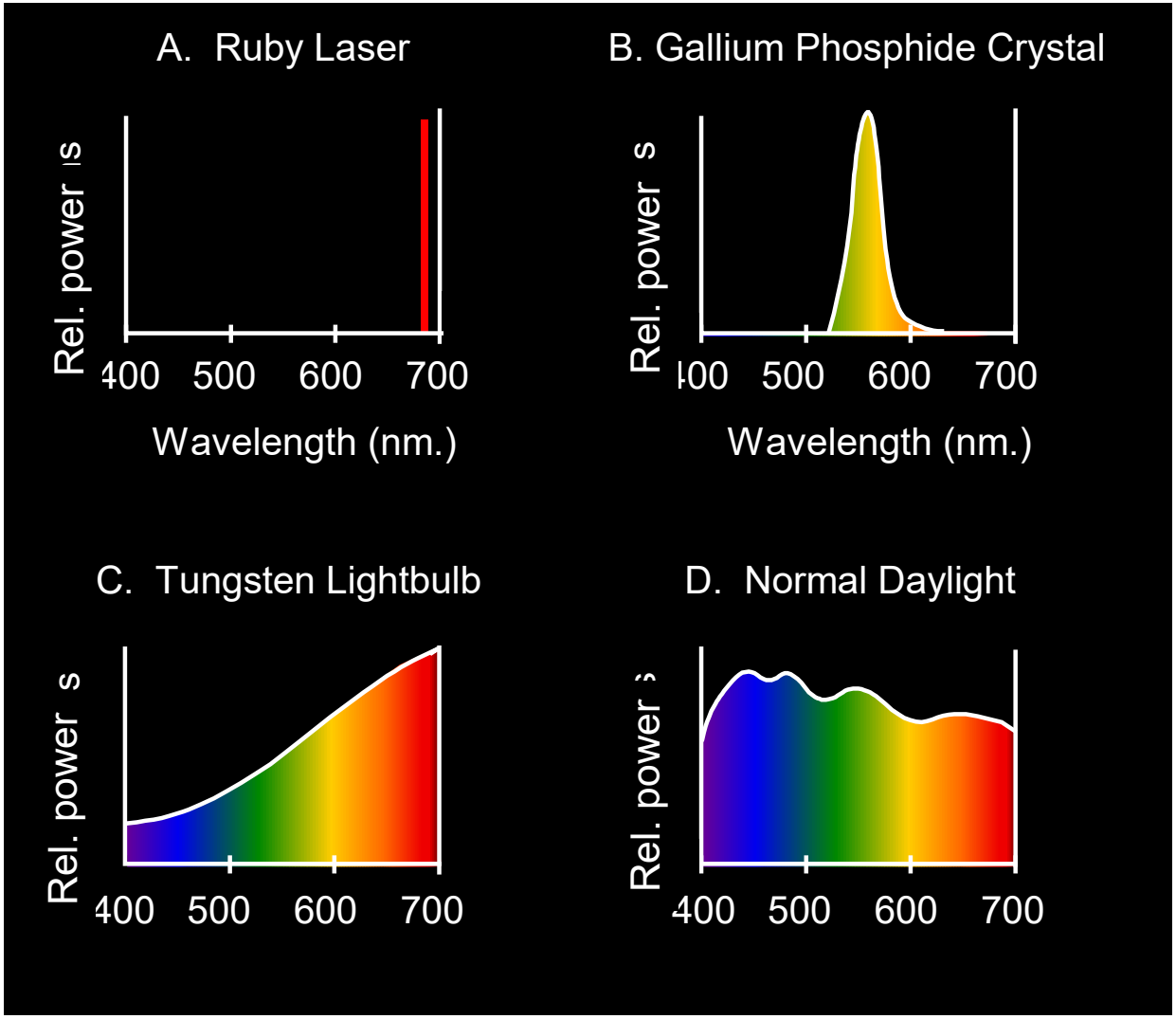
Любой источник света можно полностью описать спектром: количество излученной энергии в единицу времени для каждой длины волны в интервале 380 - 780 nm.



© Stephen E. Palmer, 2002

Физика света

Примеры спектров разных источников света

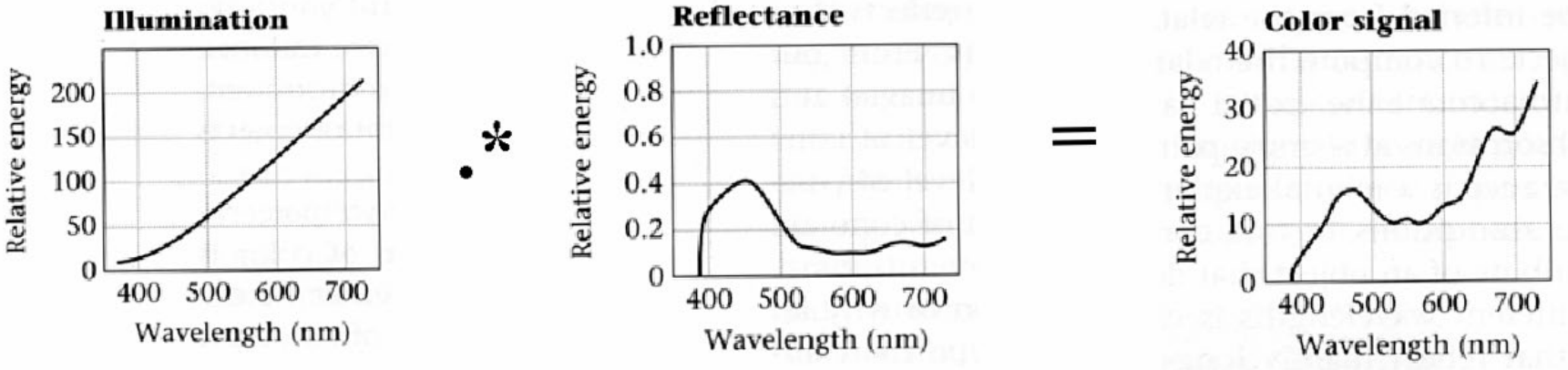


© Stephen E. Palmer, 2002

Slide by S. Lazebnik

Взаимодействие света и объектов

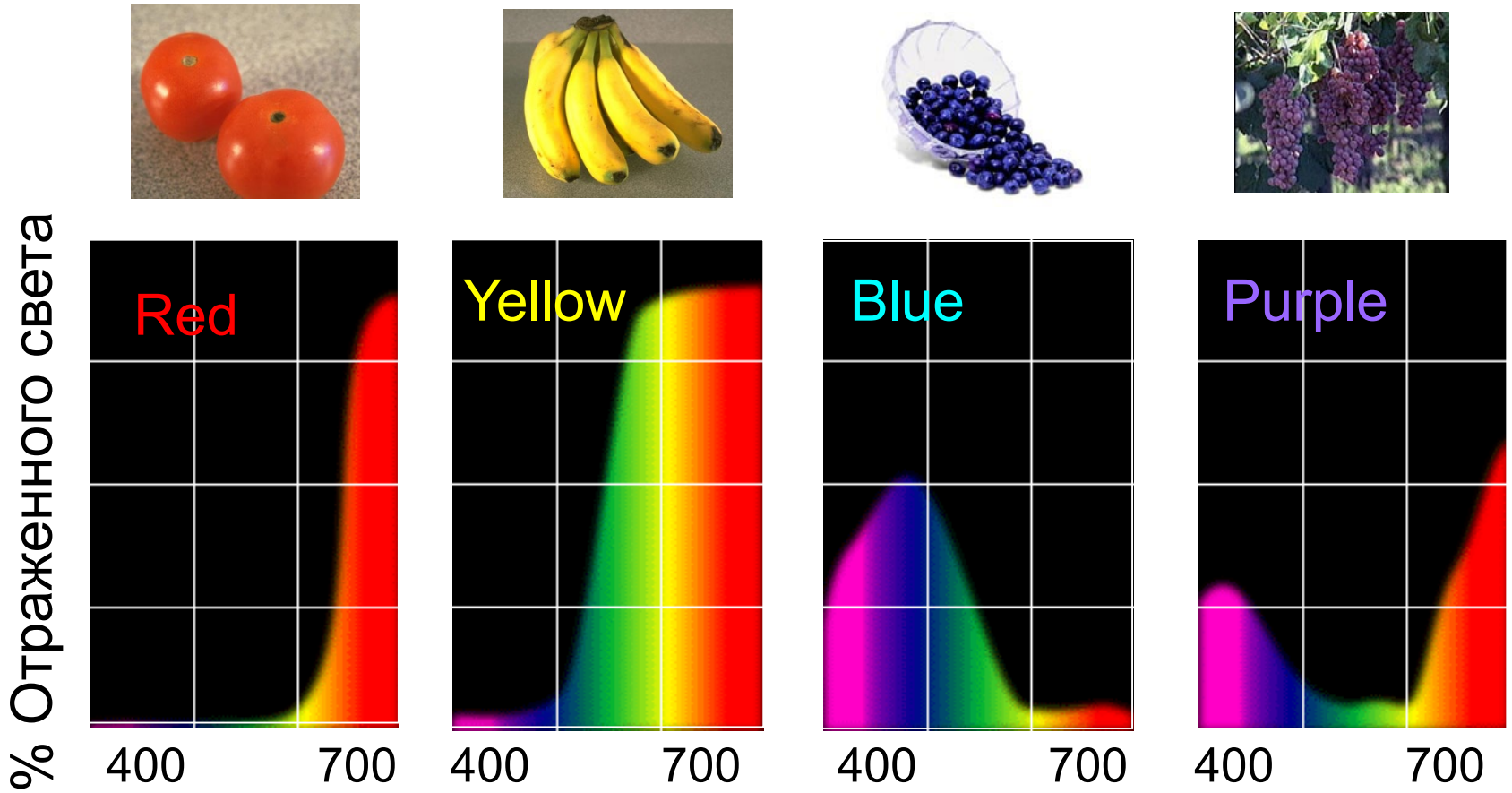
Отраженный свет это результат взаимодействия излучаемого света и поверхности



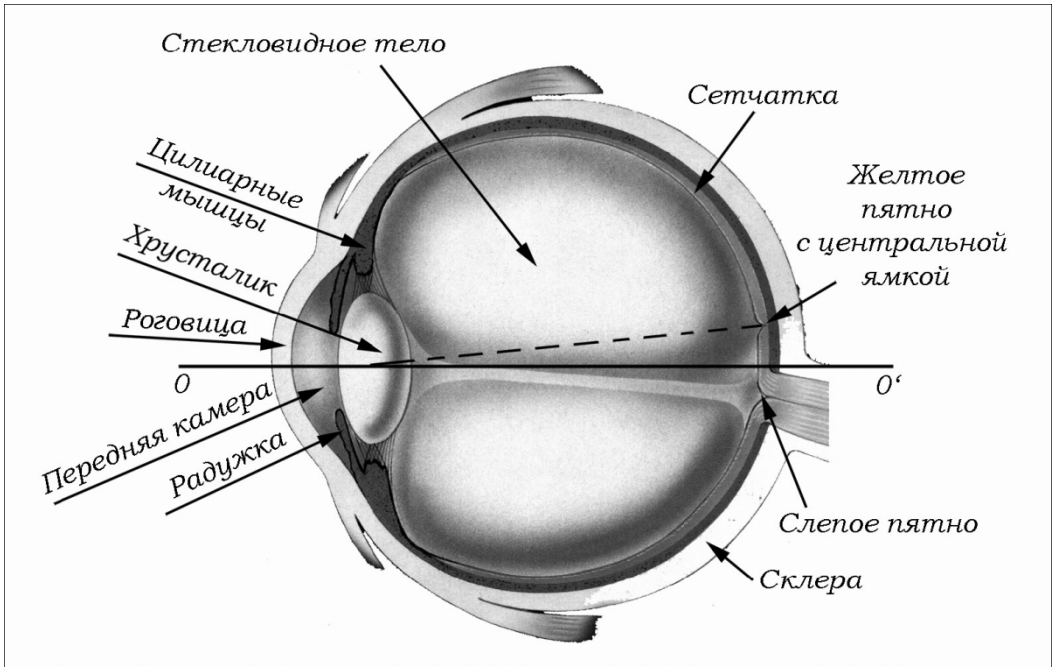
Slide by S. Lazebnik

Физика света

Примеры спектров отраженного света от предметов



Человеческий глаз



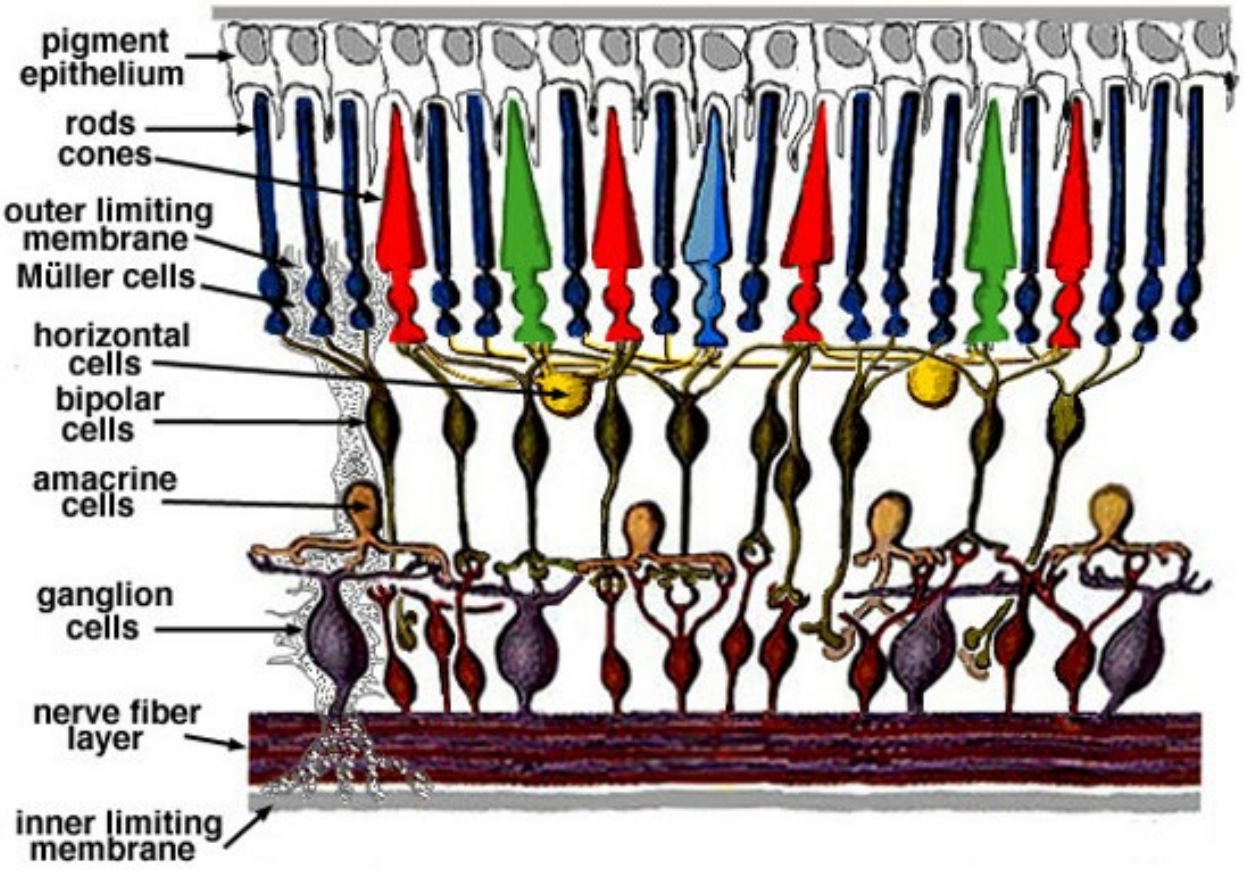
Глаз как камера!

- **Радужка** – цветная пленка с радиальными мышцами
- **Зрачок -** отверстие (апертура), диаметр управляется радужкой
- **Хрусталик** – «линза», меняющая форму под действием мышц
- Где матрица?
  - Клетки-фоторецепторы на сетчатке

Slide by Steve Seitz

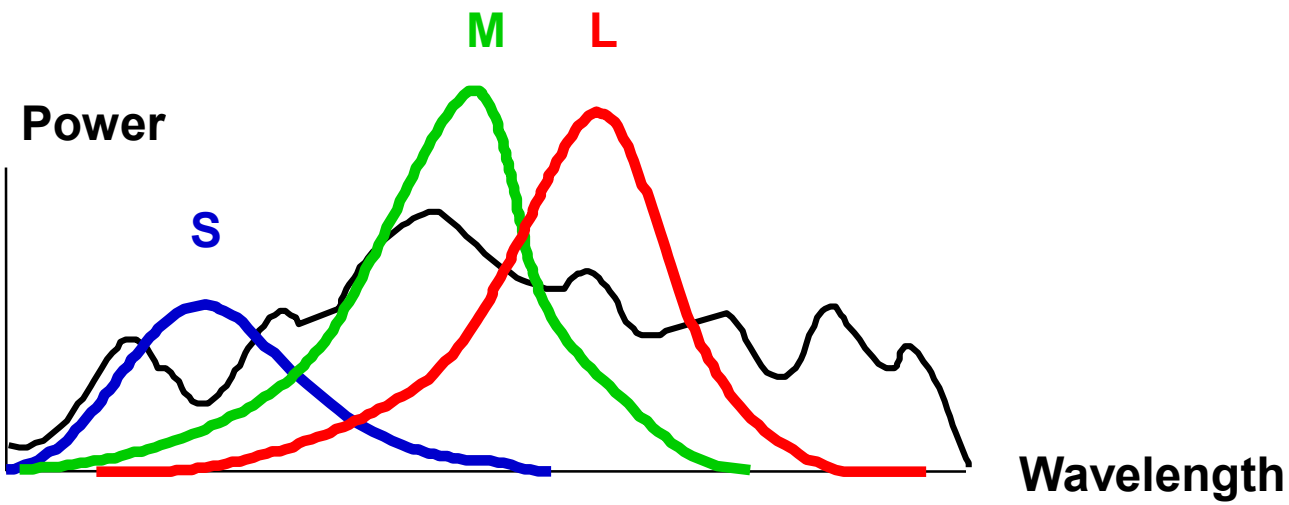
Сетчатка глаза

Свет



- Палочки (Rods) измеряют яркость
- Колбочки (Cones) измеряют цвет

Восприятие цвета



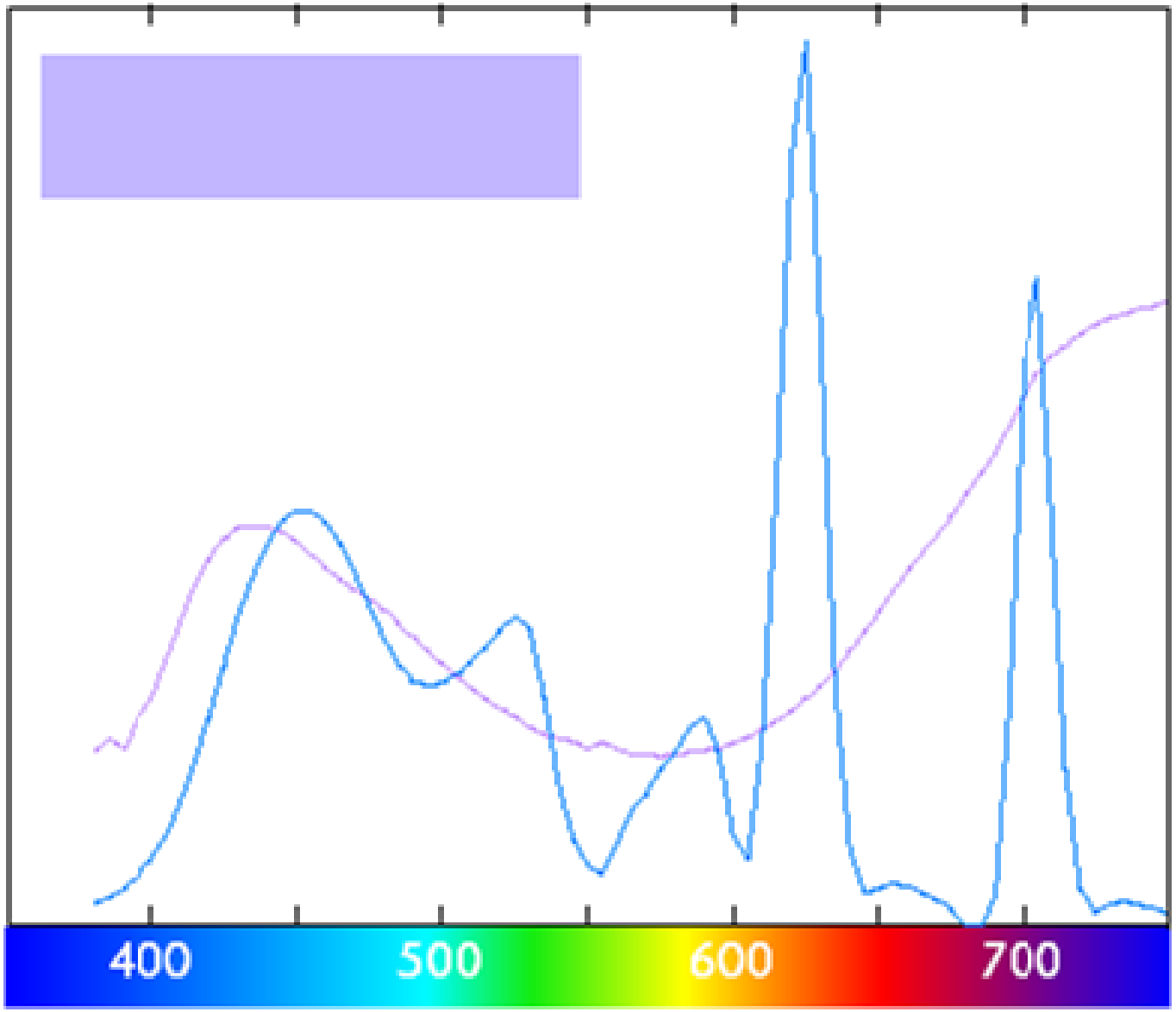
Палочки и колбочки – фильтры спектра

- Спектр умножается на кривую отклика, производится интегрирование по всем длинам волн
  - Каждый тип колбочек даёт 1 число

- В: Как же мы можем описать весь спектр 3мя числами?
- О: Мы и не можем! Большая часть информации теряется.
  - Два разных спектра могут быть неотличимы
    - » Такие спектры называются **метамеры**

Slide by Steve Seitz

Метамеры фиолетового

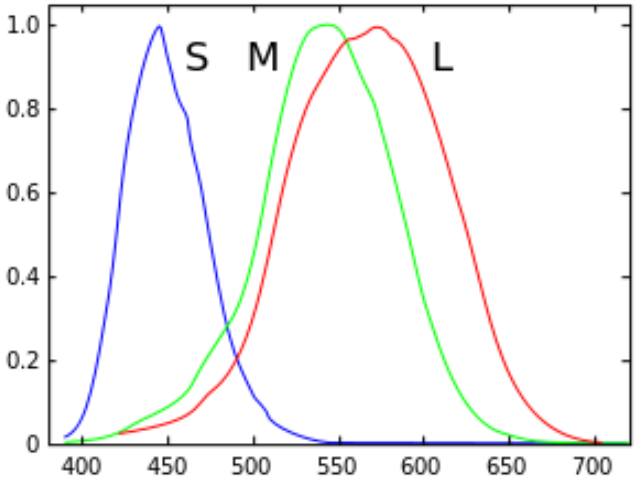


Slide by S. Lazebnik

Модели цвета

Модель LMS

- Три значения, описывающие вклад (возбуждение), каждого из трёх типов фоторецепторов
  - L(long)
  - M(medium)
  - S(short)

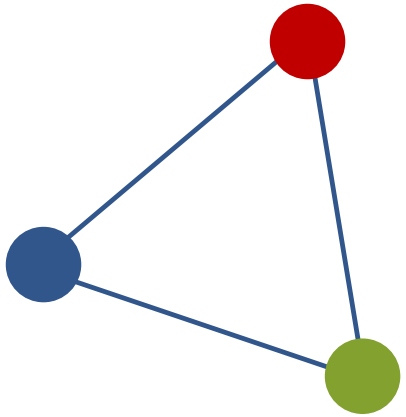


- Некоторые комбинации значений LMS невозможны для реального света
  - M > 0, S = L = 0

- Невозможно на практике построить дисплей, работающий в LMS

Трихроматическая теория и линейные модели

- Выбираем базовые цвета P_1, P_2, P_3
- Остальные цвета задаются линейной комбинацией базовых цветов (primaries). Веса – «координаты» цвета



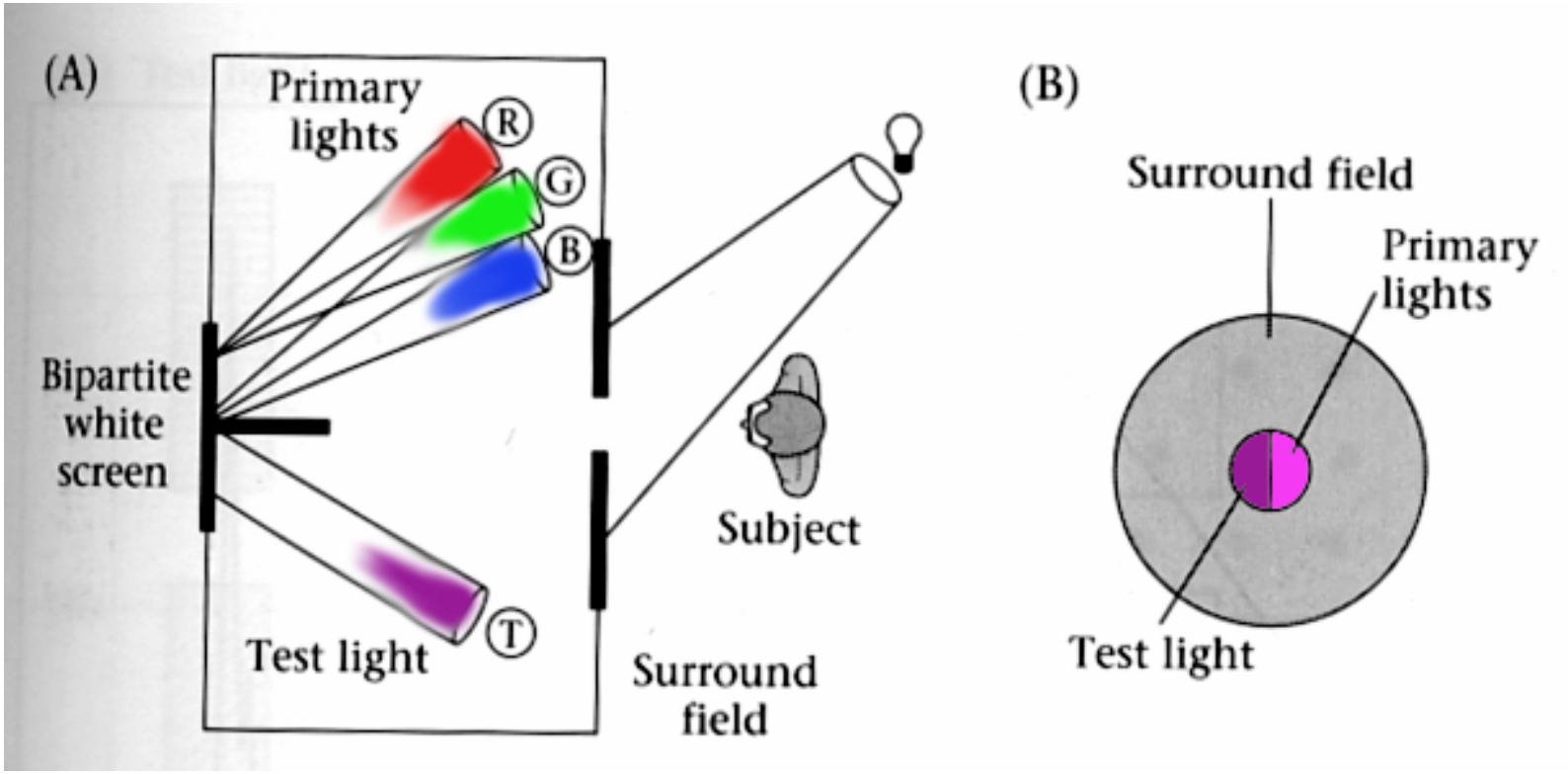
$$A = u_1 P_1 + u_2 P_2 + u_3 P_3$$

Смешение трех цветов

- Трихроматическая теория гласит, что все видимые цвета можем получить комбинацией трёх базовых цветов
- Можем делать эксперименты по сопоставлению цветов

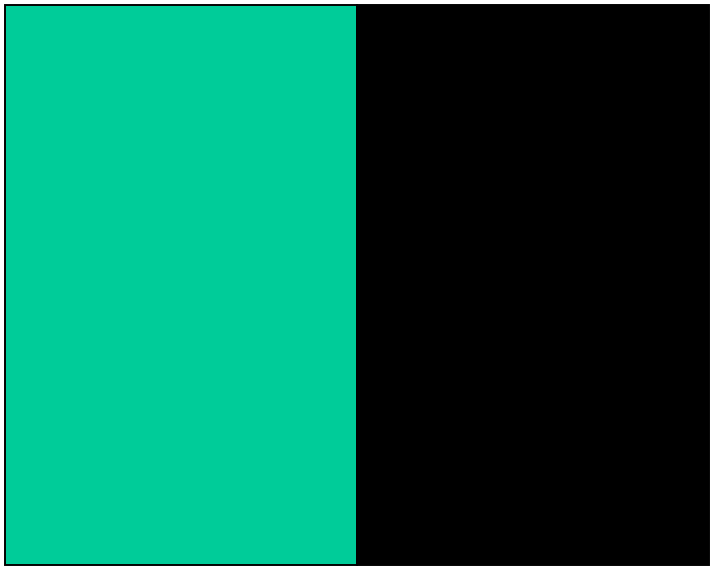
Сопоставление цветов

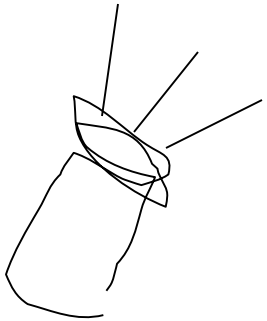
- Берём «тестовый» источник и пробуем сопоставить его комбинацией базовых цветов (источников света)



Foundations of Vision, by Brian Wandell, Sinauer Assoc., 1995

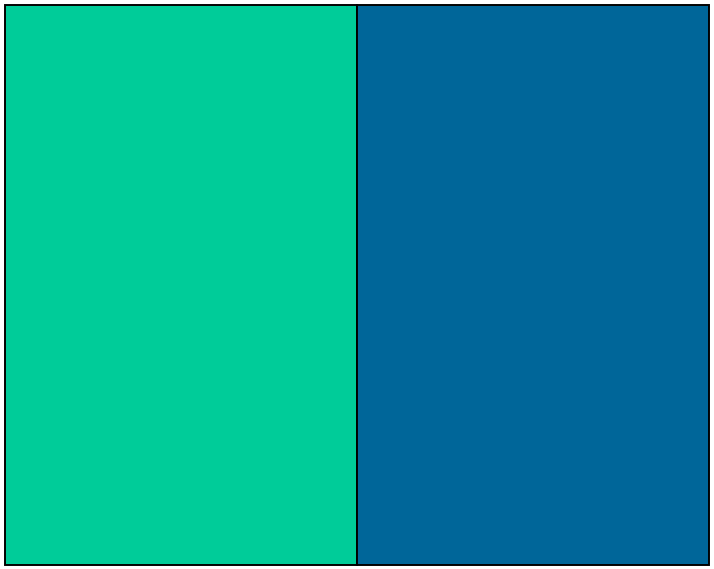
Эксперимент №1

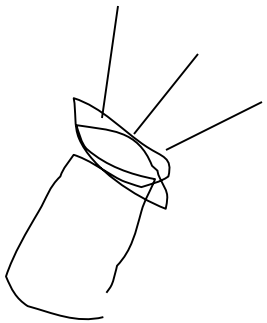


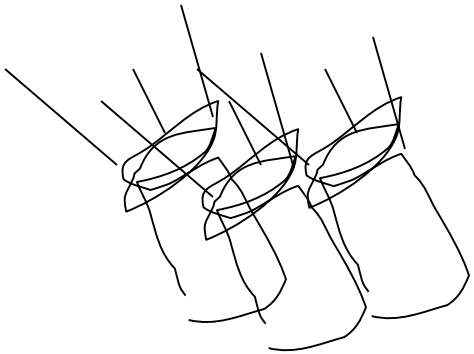


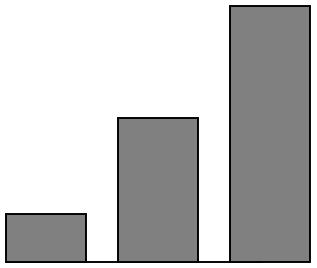
Source: W. Freeman

Эксперимент №1



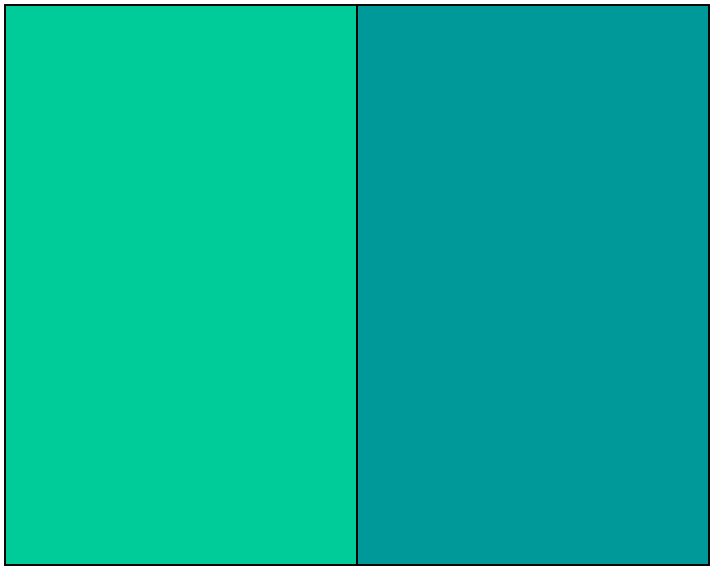


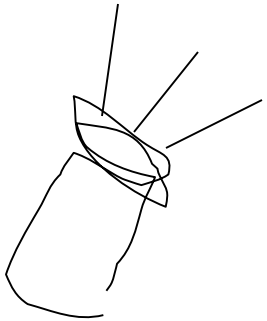


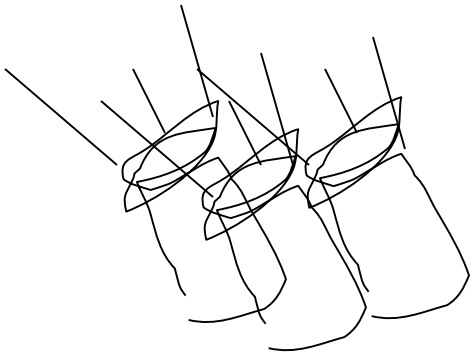


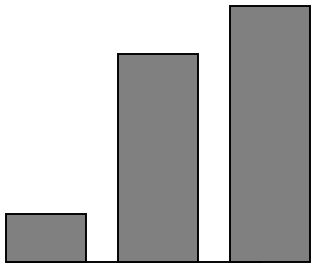
Source: W. Freeman

Эксперимент №1



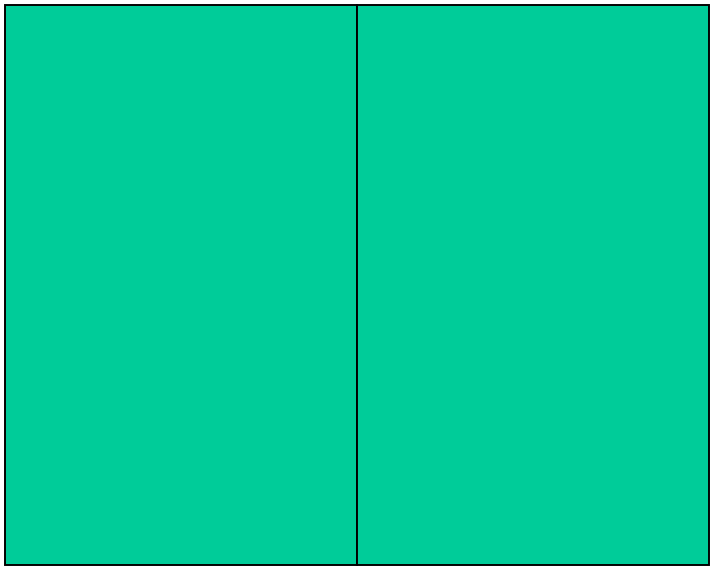




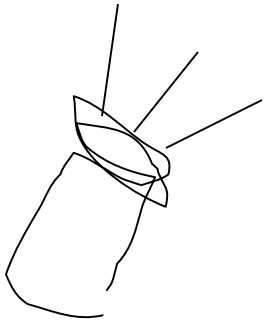


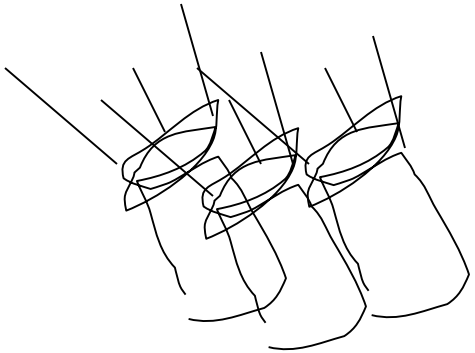
Source: W. Freeman

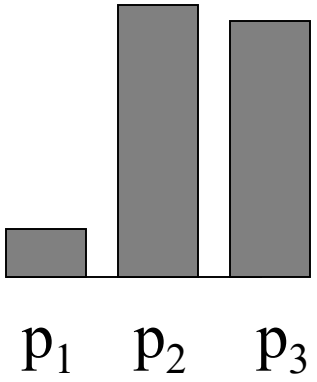
Эксперимент №1



«Координаты» света в нашей модели







Source: W. Freeman

Закон аддитивности Грассмана (1853)

- Сопоставление цветов линейное
- Если два источника света сопоставляются одинаковыми весами базовых источников, то они воспринимаются одинаково
  - Пусть $A = u_1P_1 + u_2P_2 + u_3P_3$ и $B = u_1P_1 + u_2P_2 + u_3P_3$
  - Тогда $A = B$
- Если мы смешиваем два источника, тогда смешение соответствующих им базовым источников будет восприниматься также
  - Пусть $A = u_1P_1 + u_2P_2 + u_3P_3$ и $B = v_1P_1 + v_2P_2 + v_3P_3$
  - Тогда $A + B = (u_1 + v_1)P_1 + (u_2 + v_2)P_2 + (u_3 + v_3)P_3$
- Если мы увеличиваем яркость источника света, тогда яркость соответствующих базовых светов должна быть увеличена на такое же значение
  - Пусть $A = u_1P_1 + u_2P_2 + u_3P_3$
  - Тогда $kA = k(u_1)P_1 + (ku_2)P_2 + (ku_3)P_3$

Модель CIE RGB 1931

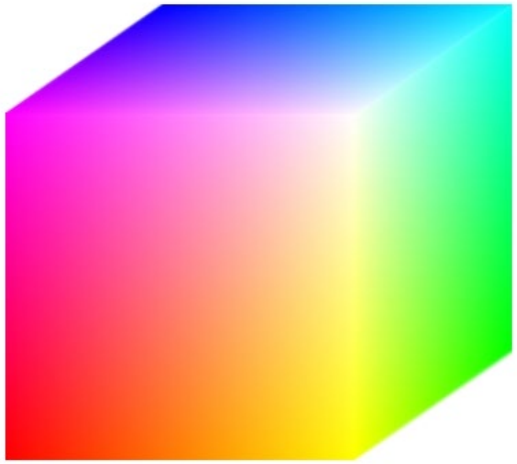
- Выберем три монохроматических цвета в качестве основных на основании информации о чувствительности колбочек
- CIE RGB 1931 от International Commission on Illumination (CIE)

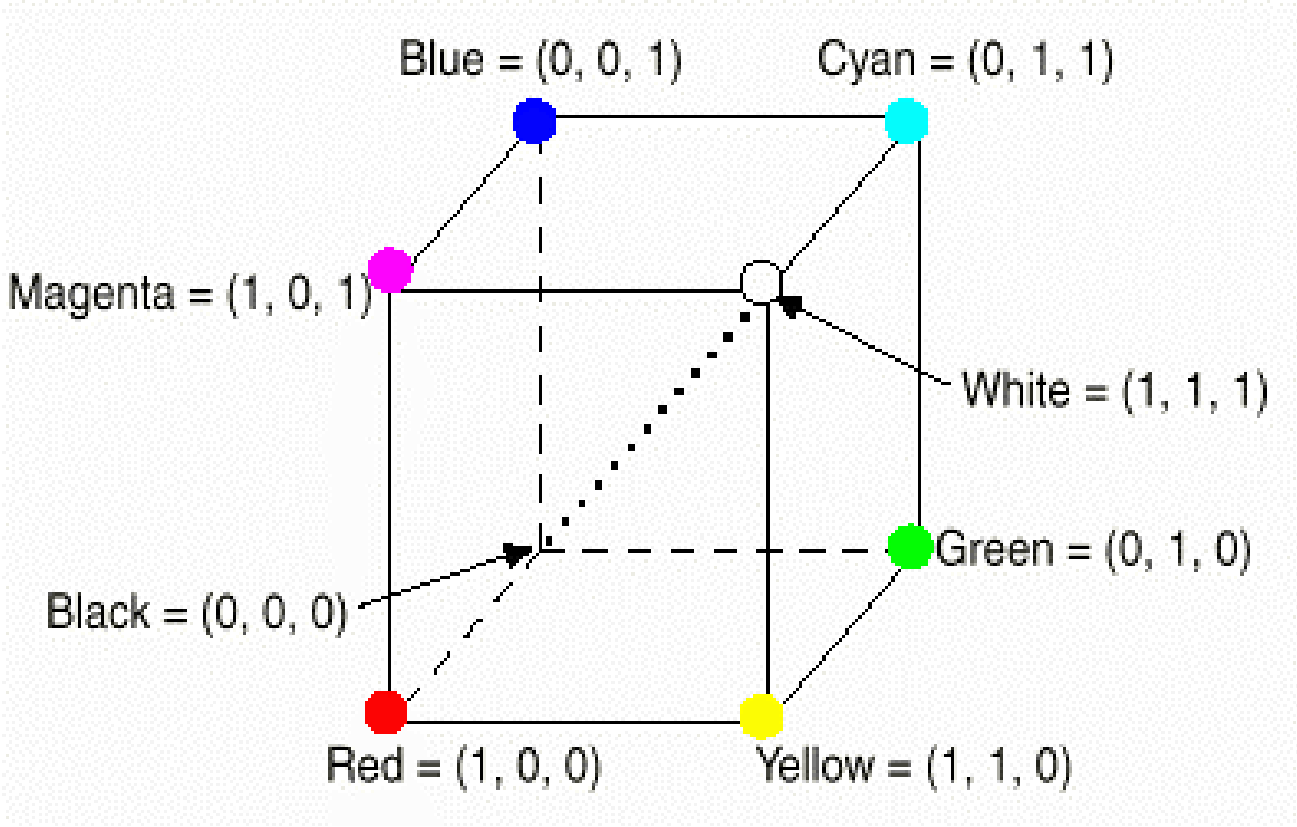
$p_1 = 645.2\ nm$

$p_2 = 525.3\ nm$

$p_3 = 444.4\ nm$

Цветовой куб RGB





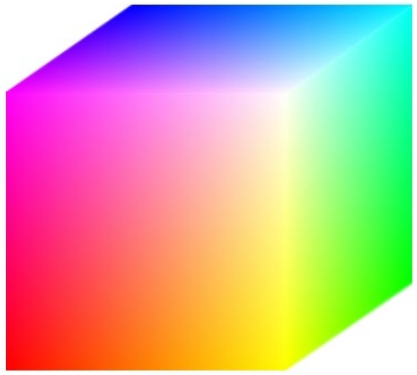
Множество цветов, задаваемых цветовой моделью RGB

Модель RGB

$p_1 = 645.2$ nm

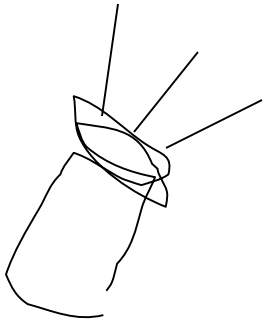
$p_2 = 525.3$ nm

$p_3 = 444.4$ nm



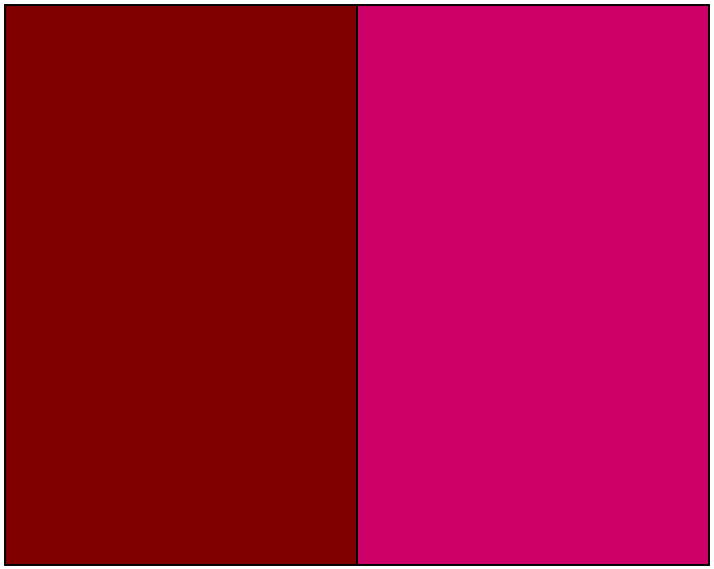
- Описывает ли модель RGB всё множество видимых человеком цветов?
- Для проверки проведём эксперимент и попробуем сопоставить каждой длине волны видимого света (когерентному источнику света) цвет в модели RGB
- *Функции сопоставления* - веса, необходимые для сопоставления с когерентными источниками света
- Будем усреднять результаты участников эксперимента

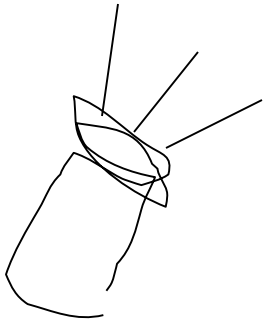
Эксперимент №2

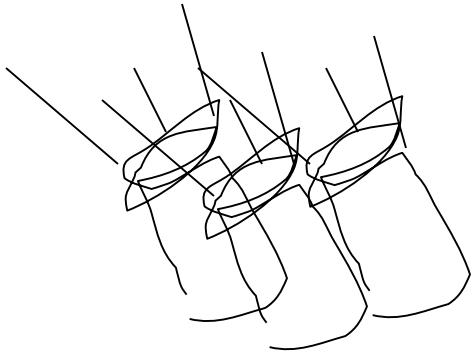


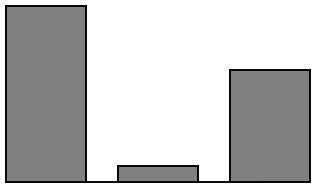
Source: W. Freeman

Эксперимент №2





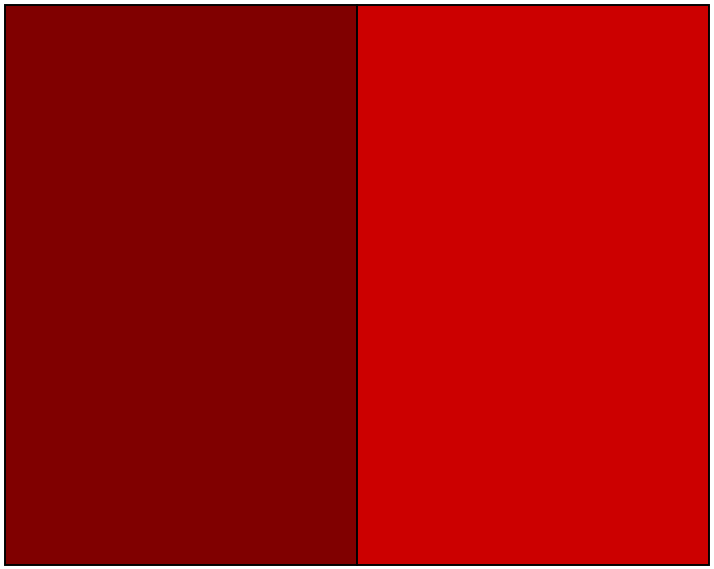


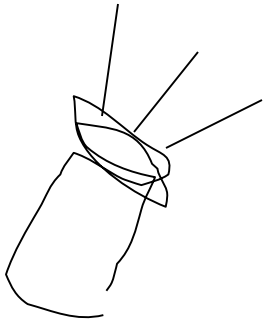


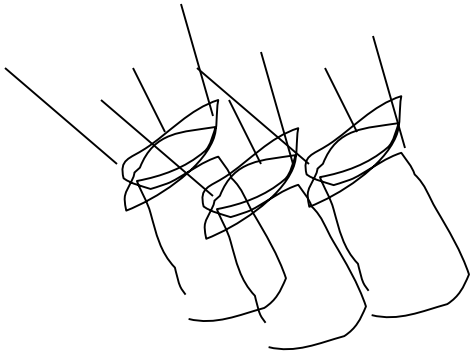
p_1 p_2 p_3

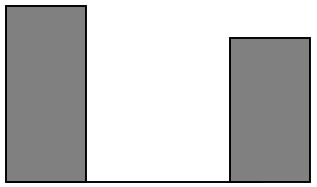
Source: W. Freeman

Эксперимент №2







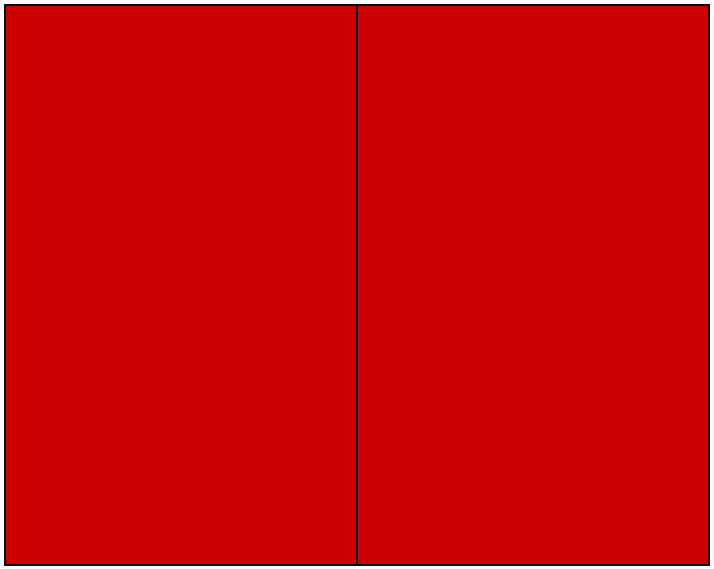


p_1 p_2 p_3

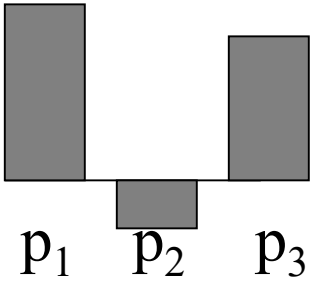
Source: W. Freeman

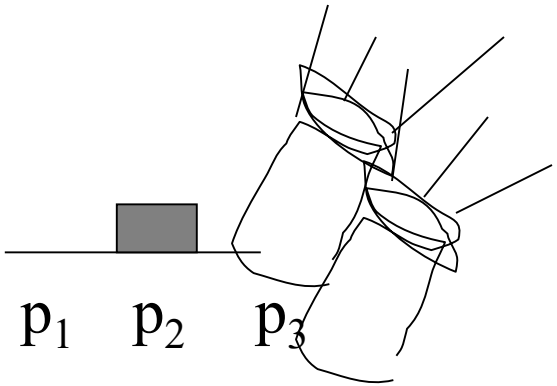
Эксперимент №2

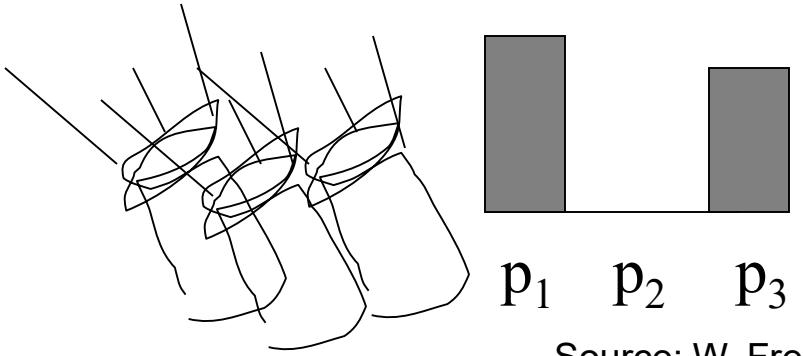
Мы называем м «отрицательным» весом основного цвета, если цвет нужно добавлять к сопоставляемому свету.



Веса основных цветов, необходимых для сопоставления:







Source: W. Freeman

Линейная цветовая модель RGB

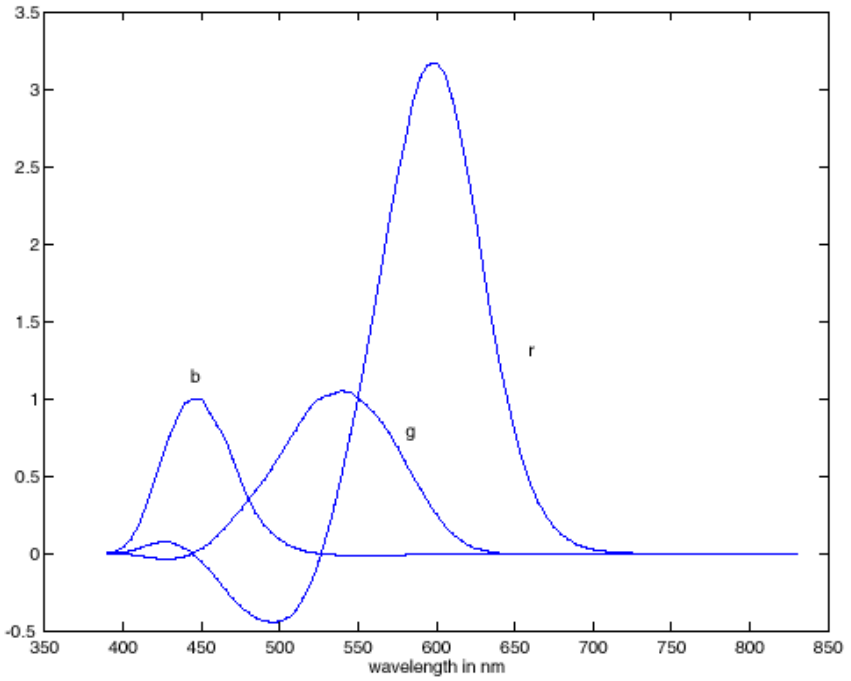
$p_1 = 645.2$ nm

$p_2 = 525.3$ nm

$p_3 = 444.4$ nm

Модель RGB не полностью описывает всё множество видимых цветов

«Вычитание» необходимо для соответствия некоторым длинам волны

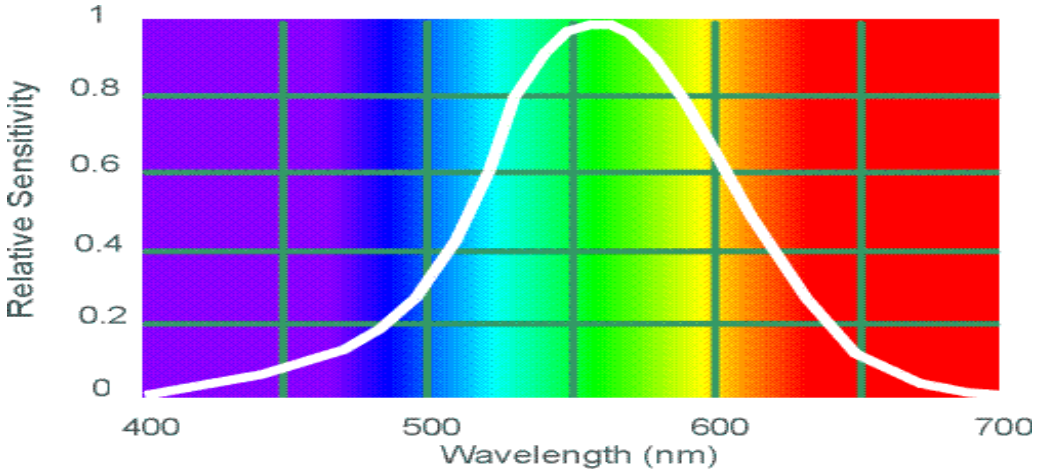


Функции сопоставления для RGB

Яркость и цветность

Интуитивно можно выделить характеристики света:

- Яркость (brightness)
- Цветность (chromaticity)



Функция чувствительности глаза человека к свету (воспринимаемая яркость)

В цветности можно выделить тон (hue) и насыщенность (saturation)

Модель YIQ

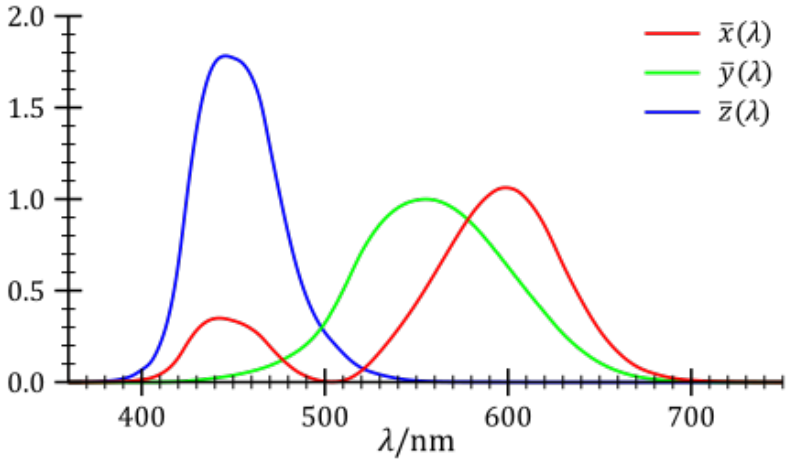
Пример модели с отдельной яркостью

- Y = .299R+.587G+.114B
- I = .596R-.275G-.321B
- Q = .212R-.528G+.311B
- R = 1.000 Y + 0.956 I + 0.621 Q
- G = 1.000 Y – 0.272 I – 0.647 Q
- B = 1.000 Y – 1.106 I + 1.703 Q

- Цветовая модель YIQ используется в коммерческом цветном телевидении США
- Модель YIQ совместима с черно-белым телевидением
- Модель YIQ используется в стандарте JPEG
- I = R – C ; Q = M - G

CIE 1931 (XYZ)

- Хотим получить линейную аддитивную модель XYZ:
  - покрывает всё множество видимых цветов
  - Y должен соответствовать видимой яркости света
  - Функции сопоставления должны быть неотрицательными



Функции сопоставления для XYZ

- X,Z тогда будут описывать «хроматическую» (цветовую) компоненту
- Точки (1,0,0), (0,1,0), (0,0,1) будут мнимыми базовыми цветами
- X,Y,Z изменяются от 0 до ∞

$$X = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\,\bar{x}(\lambda)\,d\lambda,$$

$$Y = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\,\bar{y}(\lambda)\,d\lambda,$$

$$Z = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\,\bar{z}(\lambda)\,d\lambda.$$

Перевод из XYZ в RGB

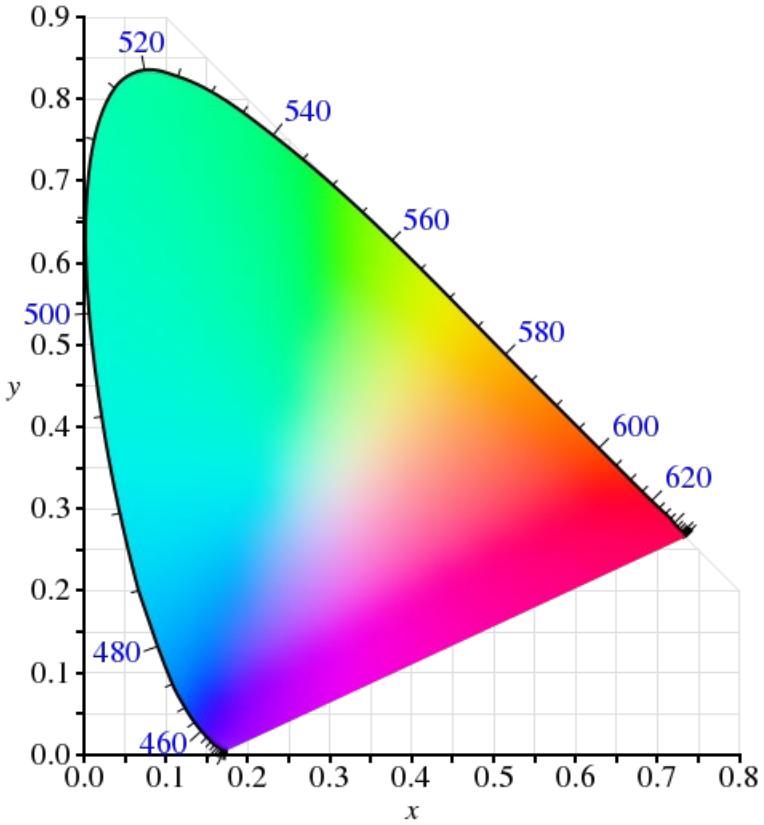
- XYZ, RGB – линейные трихроматические модели
- Из одной модели в другую можем перейти с помощью линейного преобразования

$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

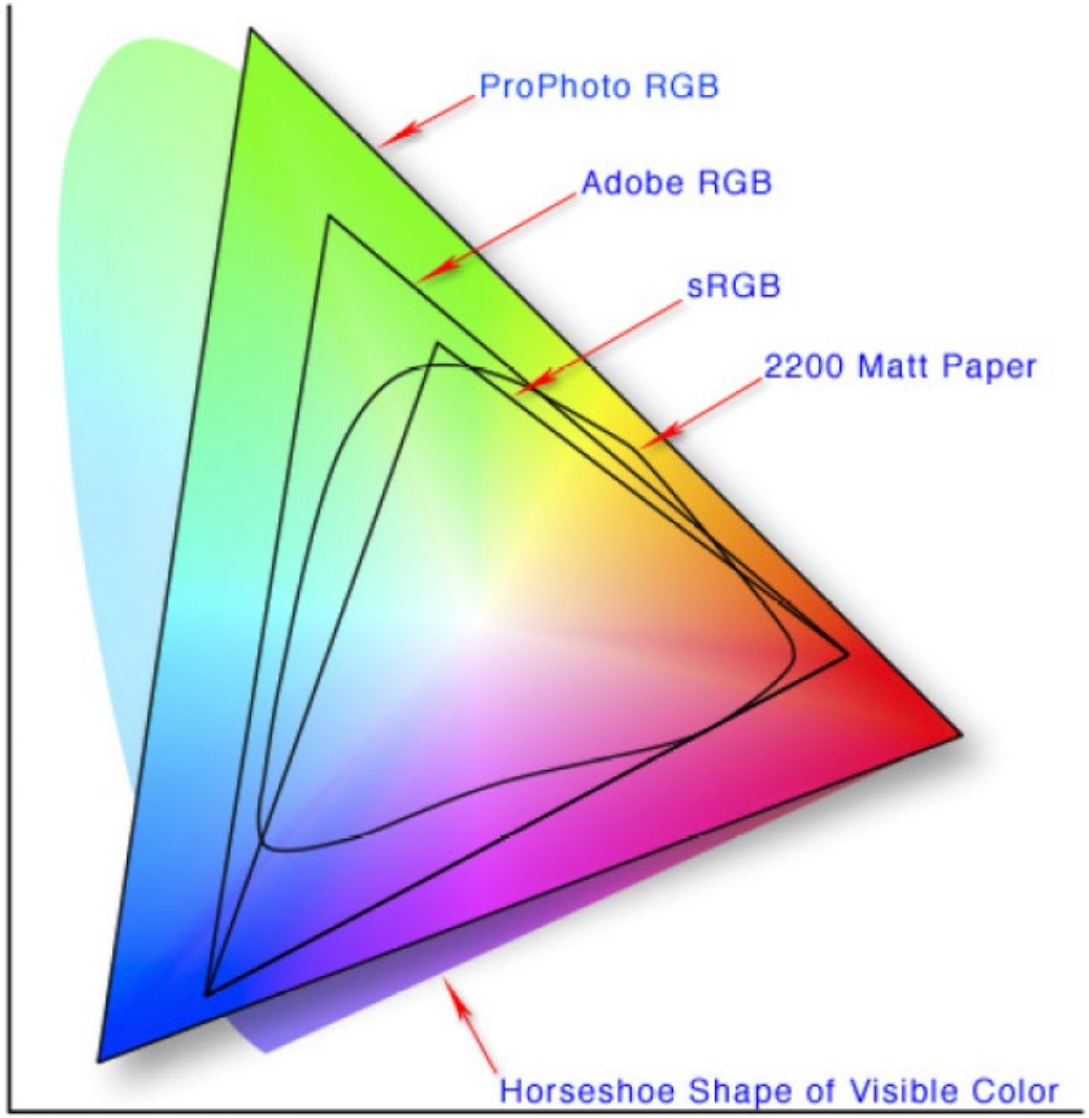
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix}$$

CIE xyY и диаграмма цветов (Gamut)

- Опишем цветность двумя нормированными параметрами x и y
  - x=X/(X+Y+Z)
  - y = Y/(X+Y+Z)
- Можем построить диаграмму цветов для $x, y \in [0,1]$
- Наблюдения:
  - Когерентные источники света располагаются по дуге
  - Нижняя прямая «фиолетового» соответствует цветам, которые невозможно получить когерентным источником света
  - Никакими тремя реальными базовыми цветами невозможно покрыть видимый диапазон цветов



Различные модели RGB



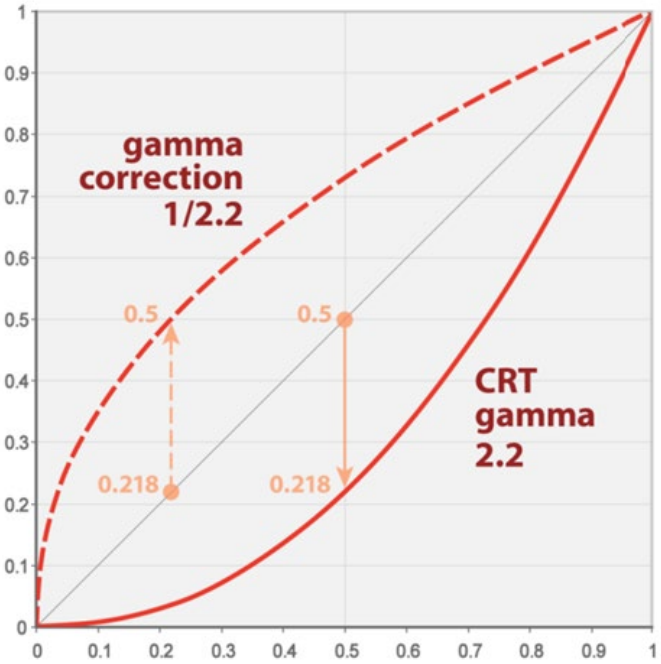
Нелинейность яркости

Физически равномерные яркости

Субъективно равномерные яркости

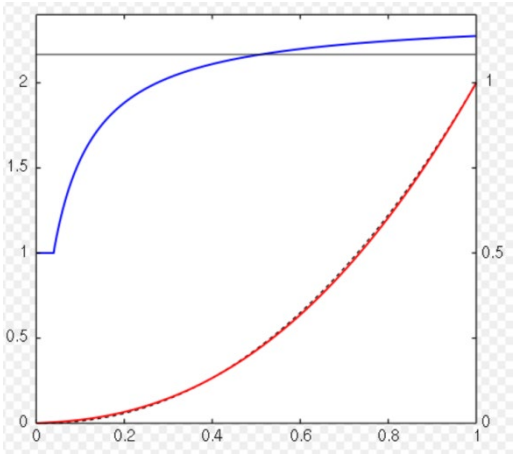
- Глаз лучше различает изменения яркости в тёмных областях, чем в светлых
- Поэтому хранить линейную яркость неэффективно
- Помогает гамма-преобразование

$$y = c \cdot x^{\gamma}$$



Модель sRGB

- Стандартная RGB модель для HDTV, мониторов, цифровых камер и т.д.
- Из XYZ в sRGB



$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$C_{\text{srgb}} = \begin{cases} 12.92 C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1+a) C_{\text{linear}}^{1/2.4} - a, & C_{\text{linear}} > 0.0031308 \end{cases}$$

a=0.055

- Из sRGB и XYZ

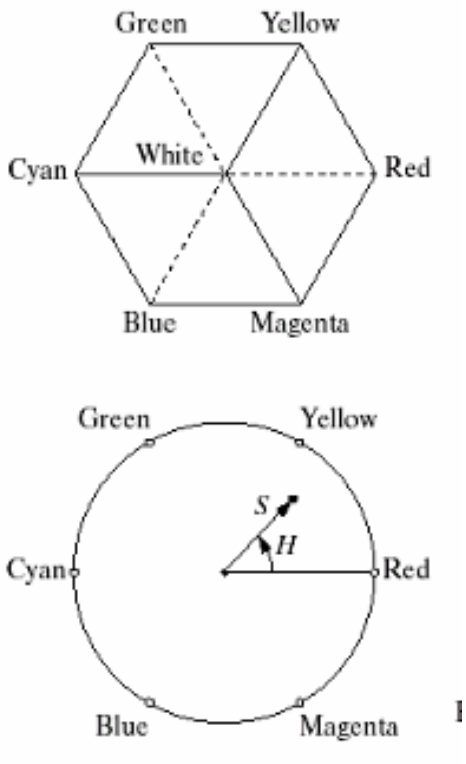
$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{srgb}}}{12.92}, & C_{\text{srgb}} \leq 0.04045 \\ \left(\frac{C_{\text{srgb}}+a}{1+a}\right)^{2.4}, & C_{\text{srgb}} > 0.04045 \end{cases}$$

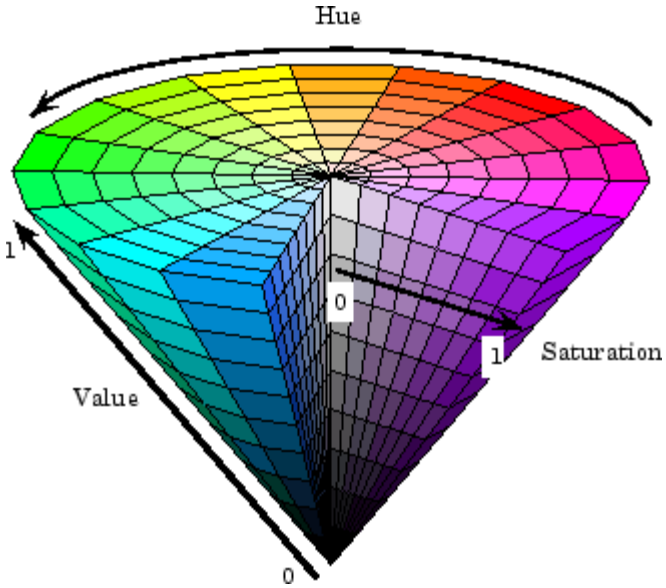
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix}$$

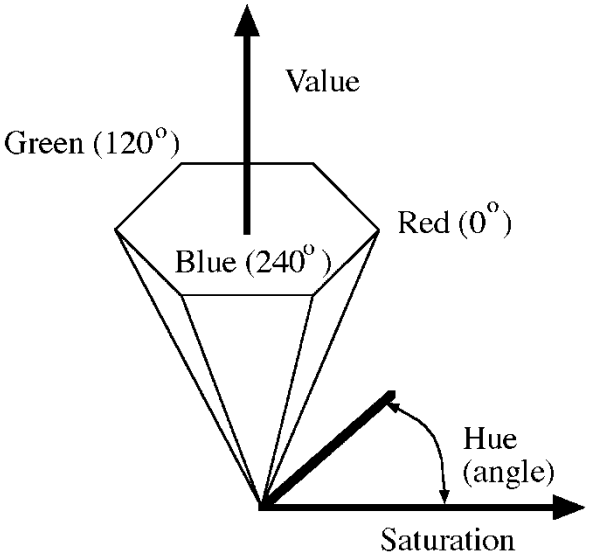
- вычисление яркости

Source: http://en.wikipedia.org/wiki/gamma_correction

Модель HSV (HIS)





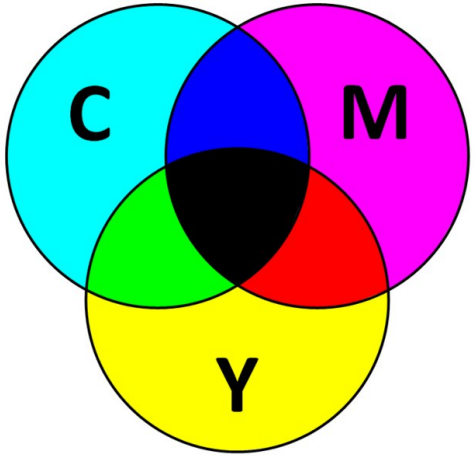


В отличие от YIQ, координаты выбраны с учетом человеческого восприятия: Hue (Тон), Saturation(Насыщенность), Value (Intensity) (Интенсивность)

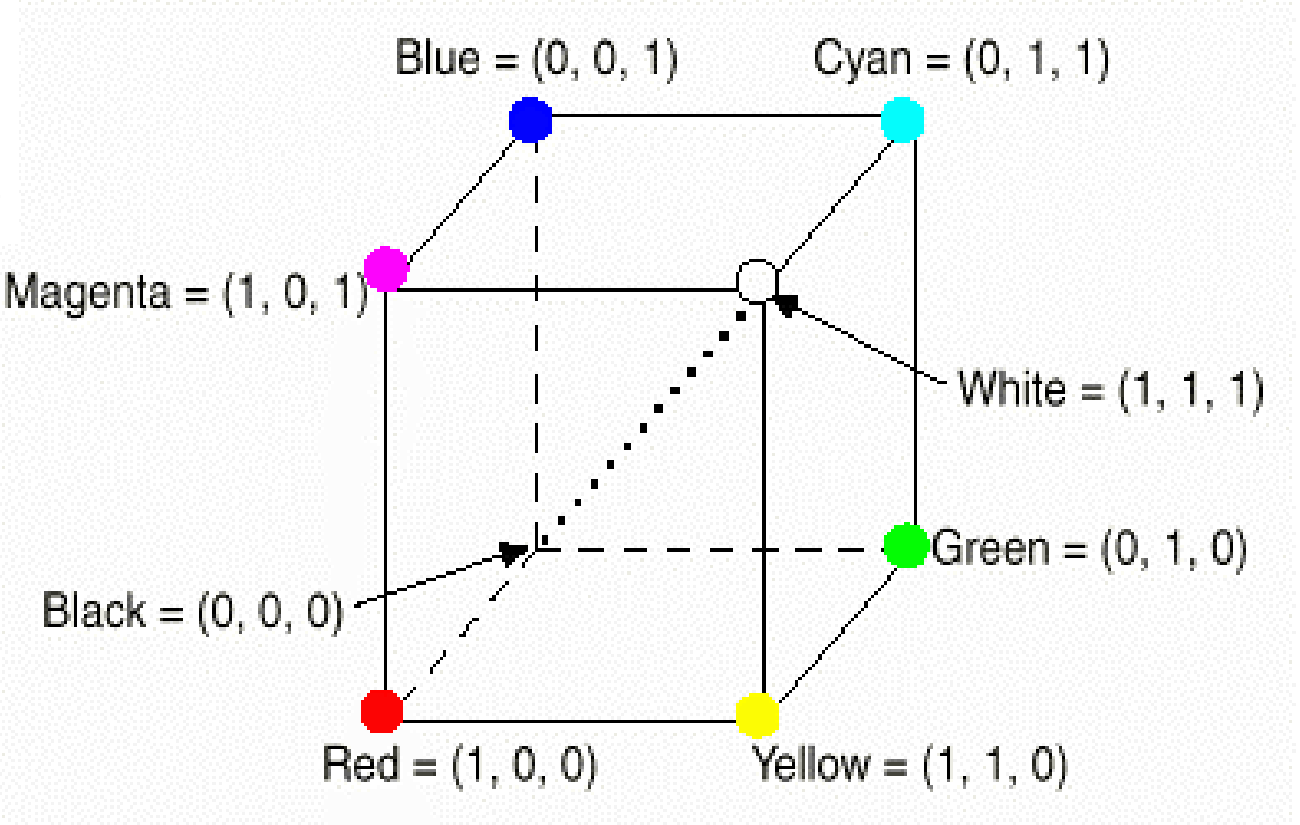
Является ли эта модель линейной?

Slide by S. Lazebnik

Субтрактивная модель CMYK



CMY



$C=G+B=W-R$

$M=R+B=W-G$

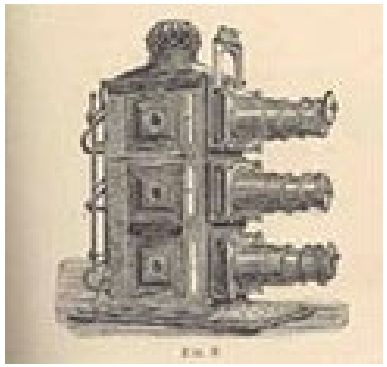
$Y=R+G=W-B$

- Аддитивная система – RGB
- Субтрактивная система – CMY
- CMY**K** для повышения качества

Первые цветные фотографии

Сергей Прокудин-Горский (1863-1944)

Фотографии Российской империи(1909-1916)



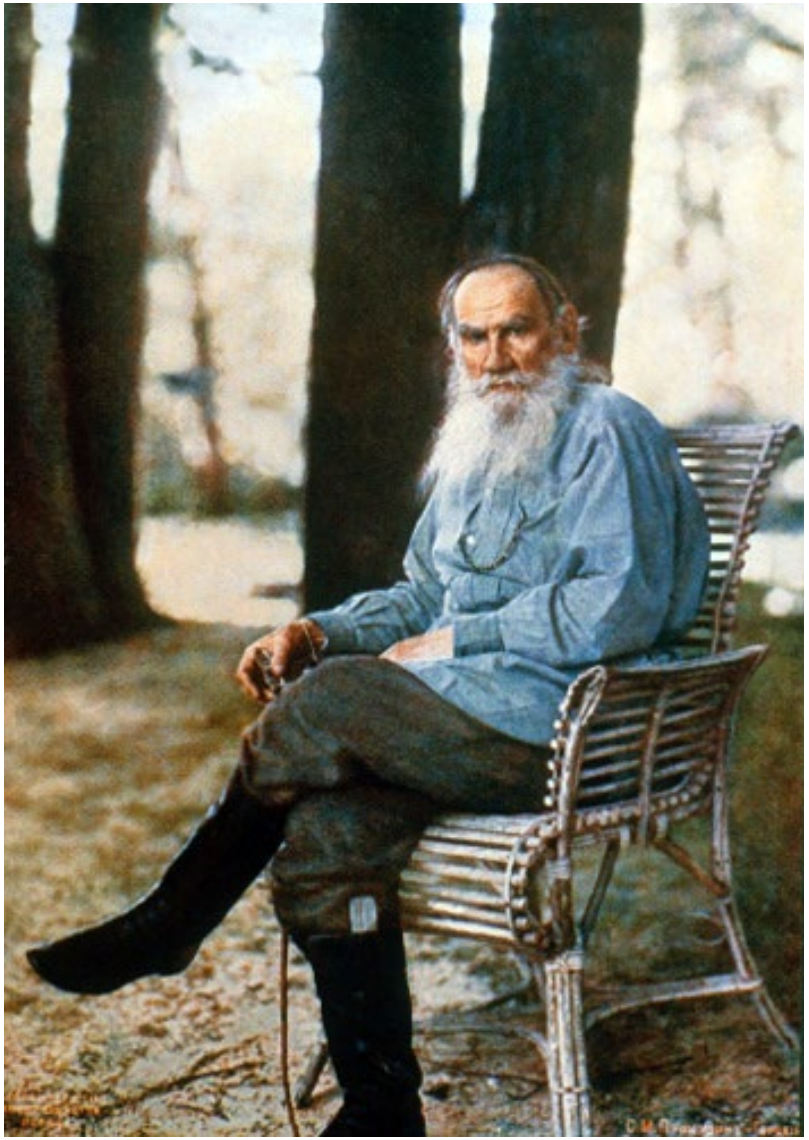
Ламповый проектор

http://en.wikipedia.org/wiki/Sergei_Mikhailovich_Prokudin-Gorskii

http://www.loc.gov/exhibits/empire/

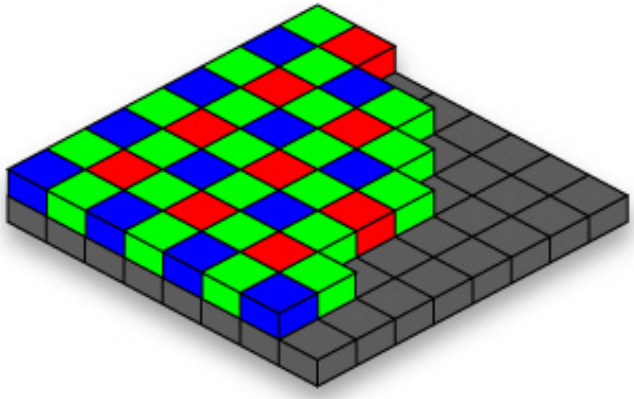
Slide by A.Efros

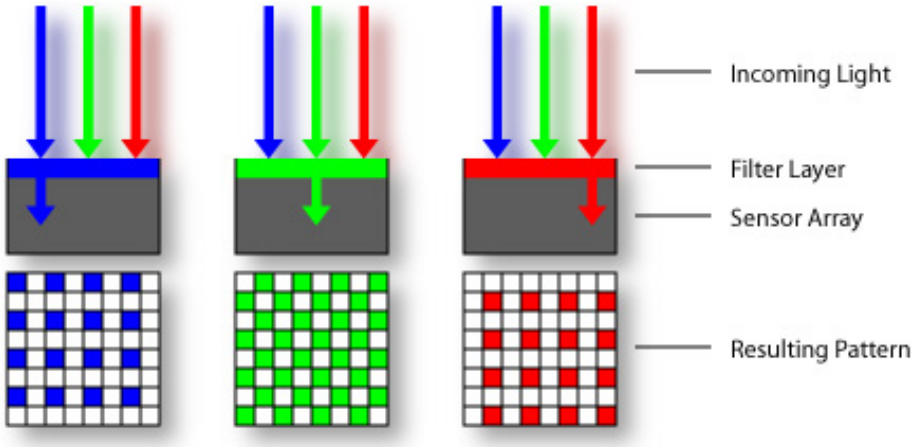
Лев Толстой



Цветное цифровое изображение

Байеровский шаблон

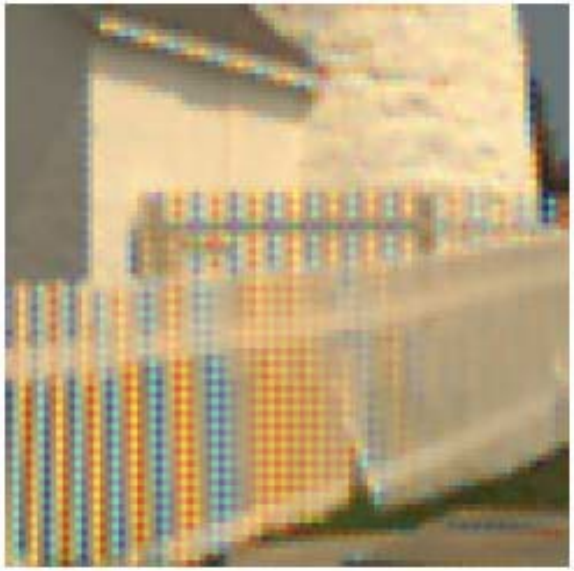




Демозаикинг (оценка пропущенных значений цвета)

Устранение мозаичности и ошибки

Original image



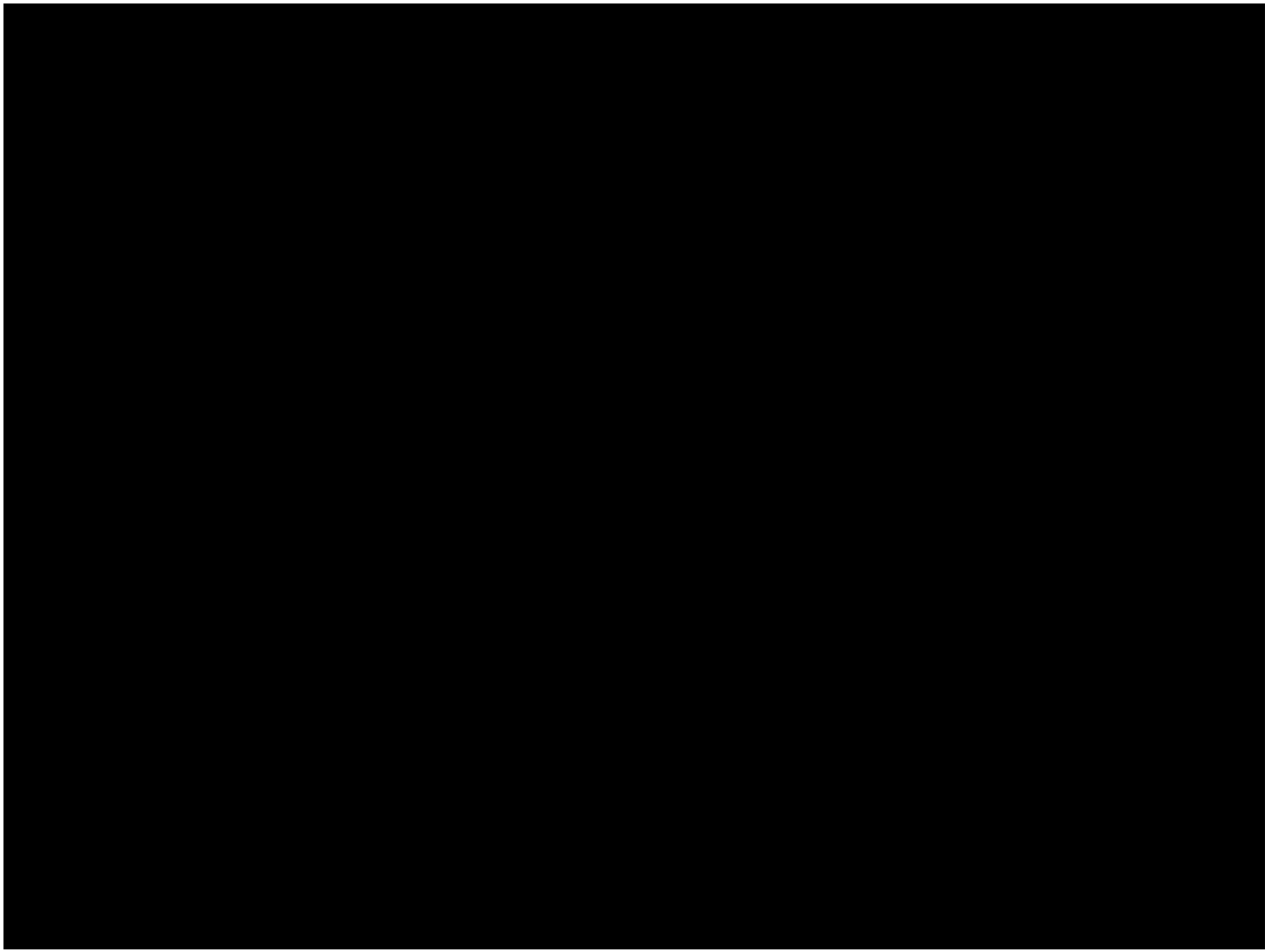
Bilinear interpolation

Proposed method

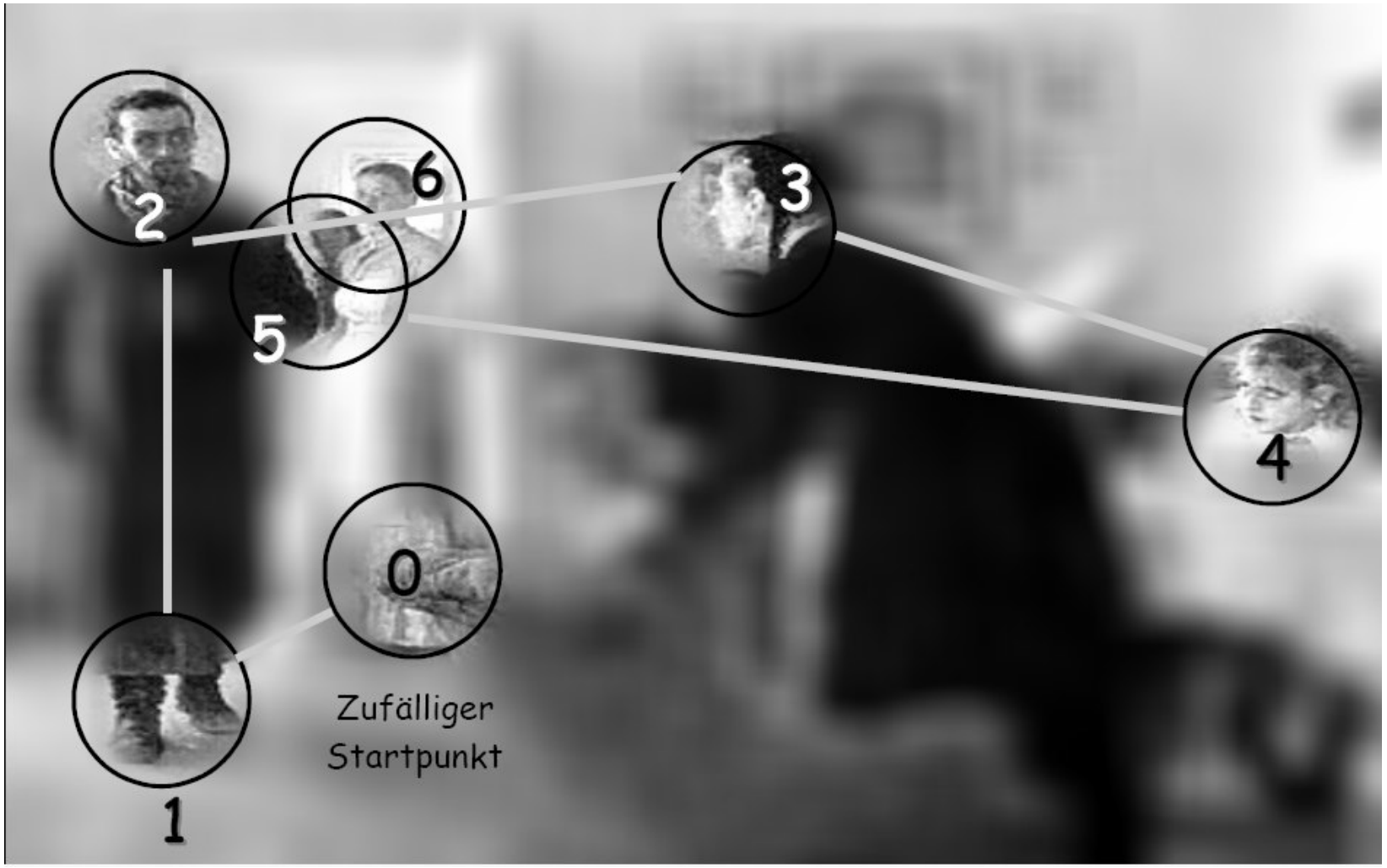
Тонкие черные и белые детали
интерпретируются как изменения цвета

Особенности человеческого зрения

Тест на внимательность

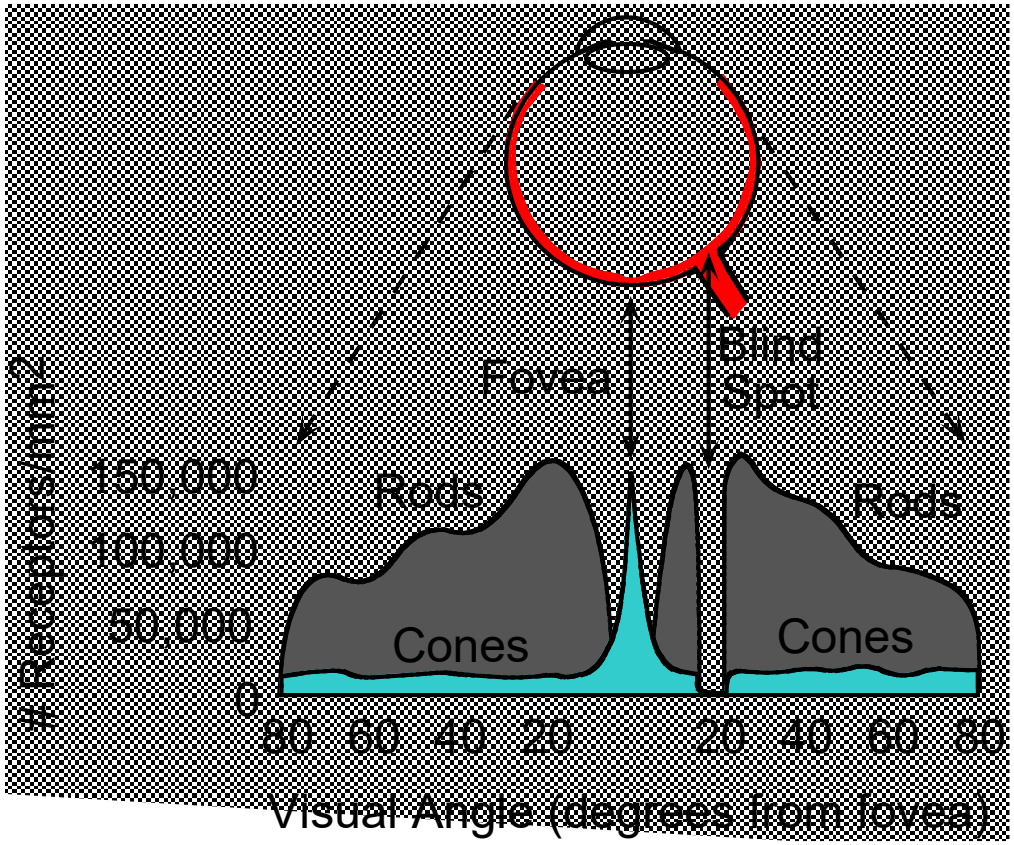


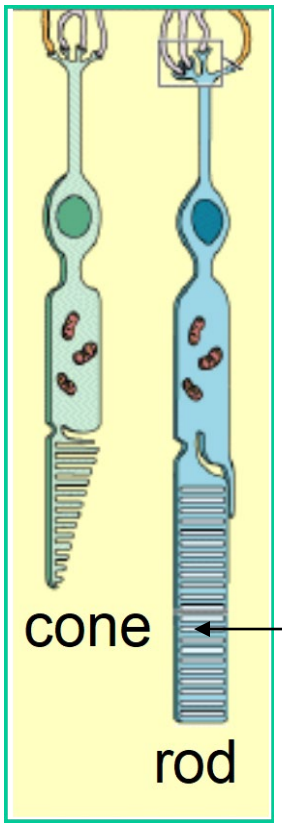
Что мы на самом деле видим



Yarbus, A. L. (1967), *Eye Movements and Vision*, New York: Plenum.

Неравномерность пространственного разрешения





Палочки и колбочки распределены неравномерно

- Палочки измеряют яркость, колбочки цвет
- **Fovea** («желтое пятно»)– маленькая область(1 or 2°) в центре визуального поля с наибольшей плотностью колбочек и без палочек
- На периферии все больше палочек подсоединены к одному нейрону

Slide by Steve Seitz

Движения глаз

1

Free examination.

2

Estimate material circumstances of the family

3

Give the ages of the people.

4

Surmise what the family had been doing before the arrival of the unexpected visitor.

5

Remember the clothes worn by the people.

6

Remember positions of people and objects in the room.

7

Estimate how long the visitor had been away from the family.

3 min. recordings of the same subject

Постоянство цвета и освещенности

J. S. Sargent, The Daughters of Edward D. Boit, 1882

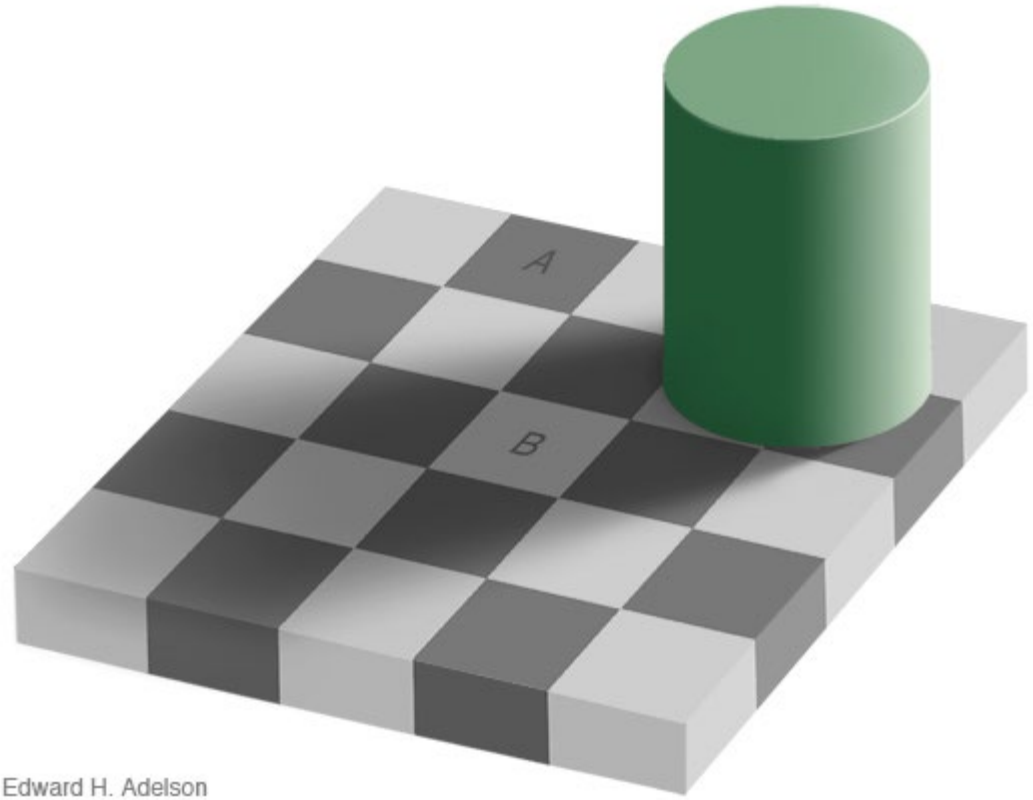
Способность зрительной системы человека оценивать собственные отражательные свойства поверхностей в не зависимости от условий освещенности

Освещённость:

- Полнолуние – 0.27 люкс
- Пасмурный день – 100-1000 люкс
- Яркий день (в тени) – 20000 люкс
- На солнце - 100000

Slide by F. Durand

Постоянство яркости

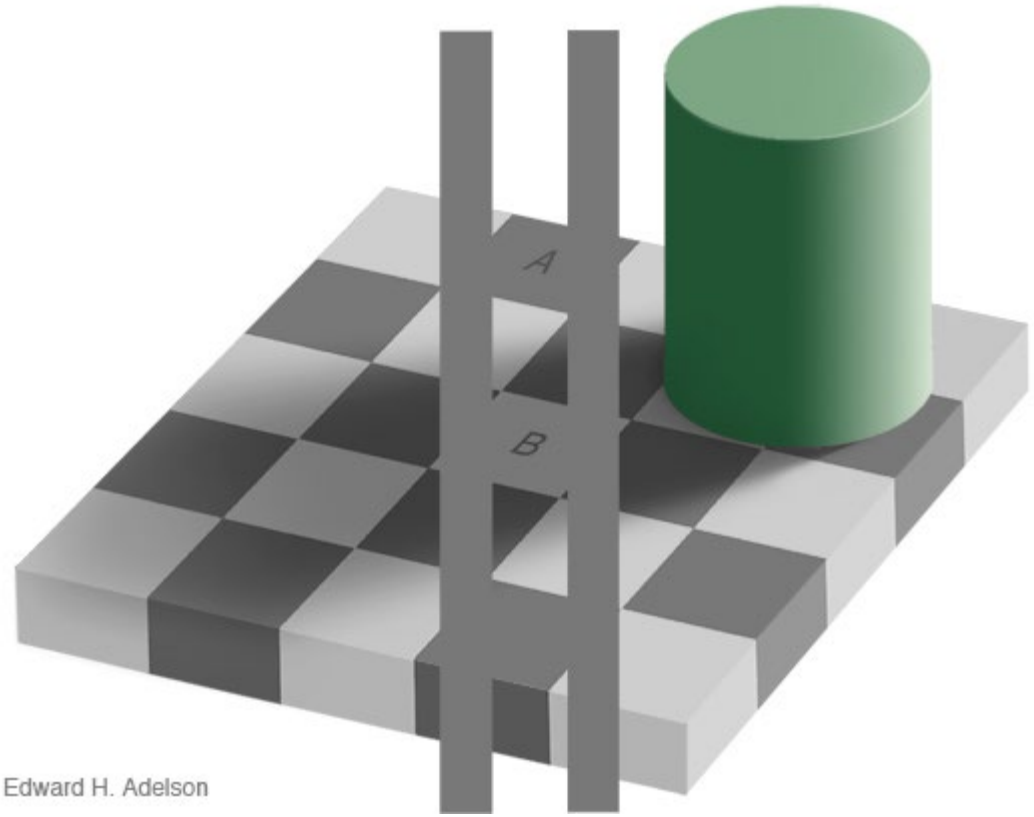


Edward H. Adelson

http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

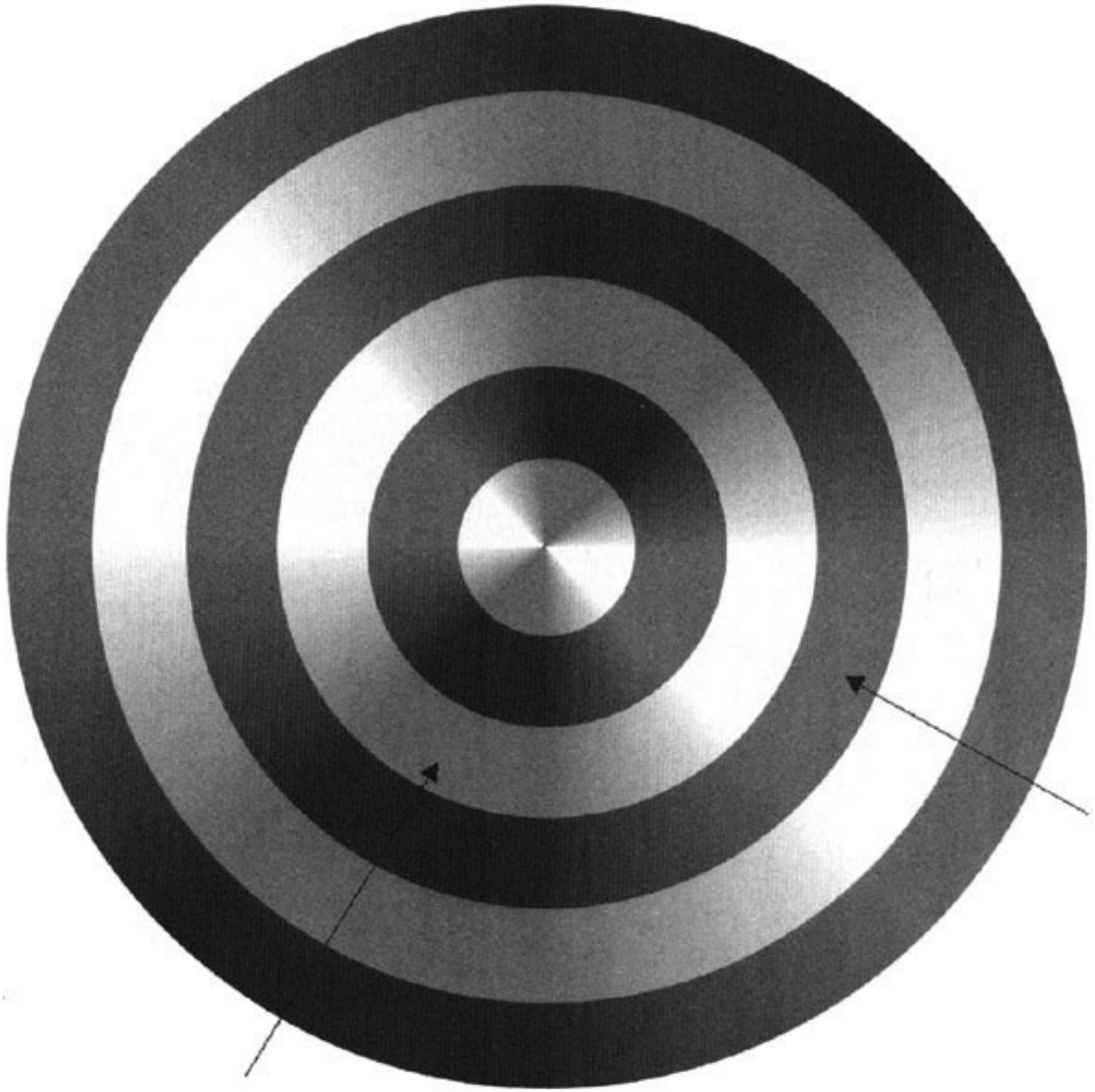
Slide by S. Lazebnik

Постоянство яркости



http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

Slide by S. Lazebnik



Недостаточный контраст изображений

Цветовой баланс («баланс белого»)

Когда мы смотрим на фотографию или монитор, глаза адаптируются к освещению в комнате, а не к освещению сцены на фотографии.

Если «баланс белого» неточен, цвета фотографии кажутся неестественными.

Неправильный баланс

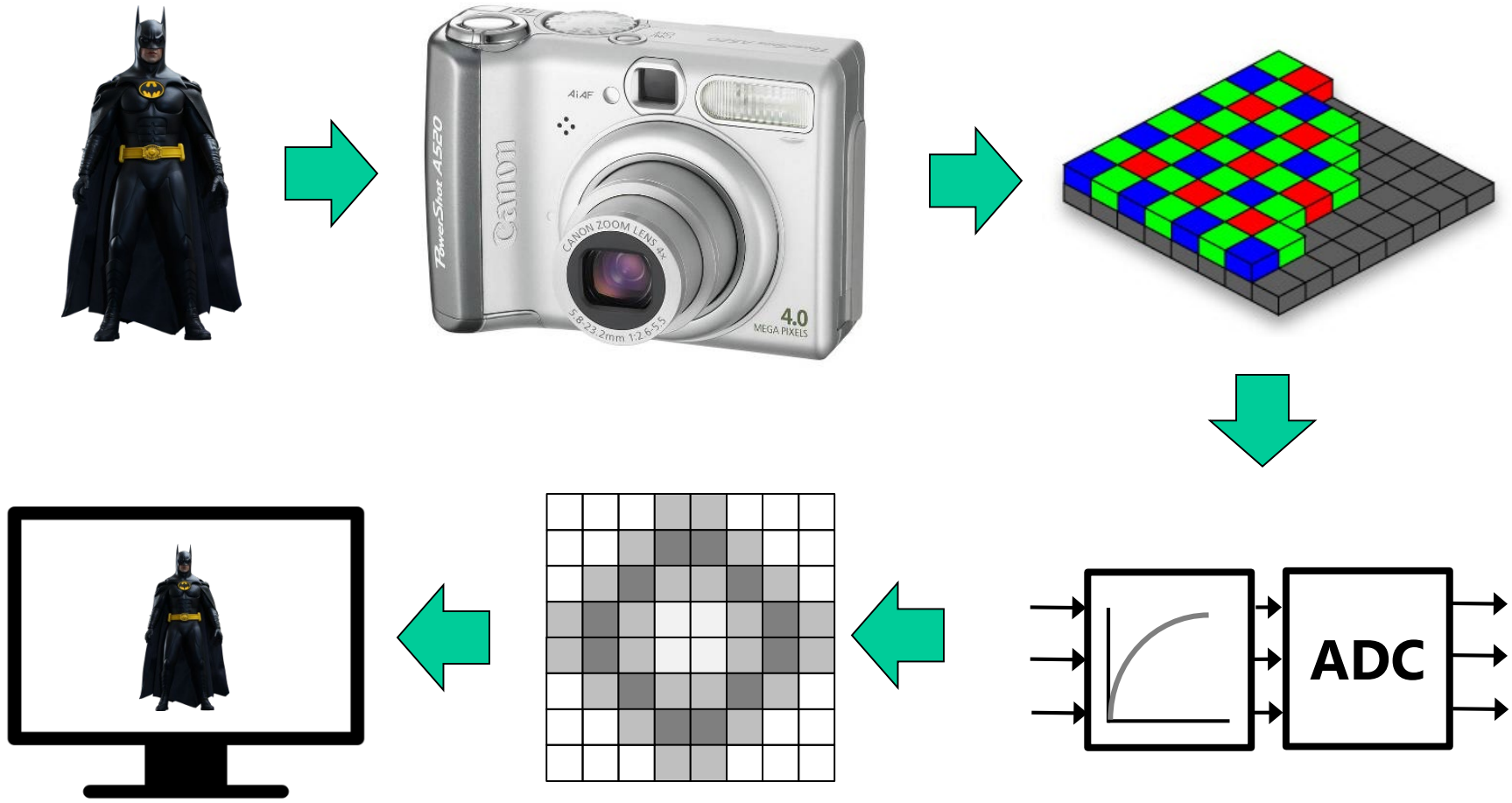
Правильный баланс

http://www.cambridgeincolour.com/tutorials/white-balance.htm

Slide by S. Lazebnik

Тональная и цветовая коррекция изображений

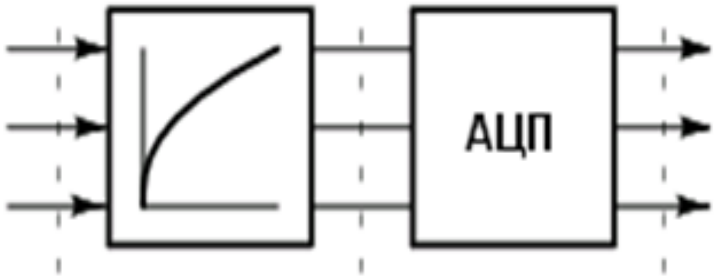
Тракт передачи изображений



Где кроются причины неправильной передачи яркости и цвета?

Причины плохой передачи яркости

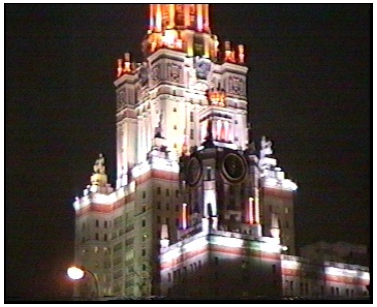
- Ограниченный диапазон чувствительности датчика
- “Плохая” функция передачи датчика



Нужно оценить качество передачи тонов в изображении

Гистограмма и её оценка

Гистограмма – это график распределения яркостей на изображении. На горизонтальной оси - шкала яркостей тонов от белого до черного, на вертикальной оси - число пикселей заданной яркости.



- Не полностью используется диапазон яркостей
- Концентрация яркостей вокруг определенных значений - неравномерное заполнение диапазона яркостей)

Точечные операторы

Оператор, который определяет значение выходного пиксела по значению только одного входного пиксела. Все пикселы обрабатываются независимо друг от друга.

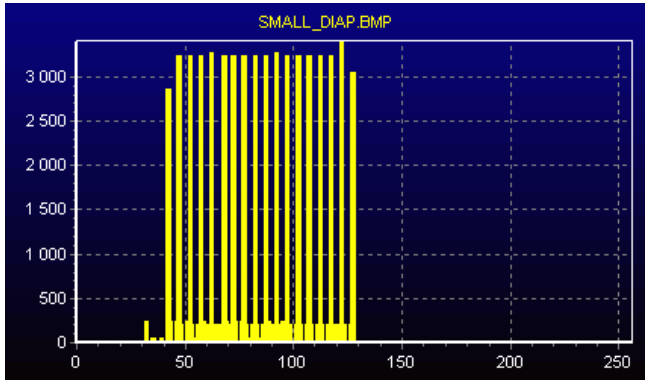
$$f^{-1}(y) = x$$

y – яркость пикселя на исходном изображении,
x – яркость пикселя после коррекции.

Пишем f^{-1}, потому что «восстанавливаем» истинное значение яркости по неправильному

Линейная коррекция

Компенсация узкого диапазона яркостей – ***линейное растяжение гистограммы (*Histogram equalization)**:



$$f^{-1}(y) = (y - y_{\min}) * \frac{(255 - 0)}{(y_{\max} - y_{\min})}$$

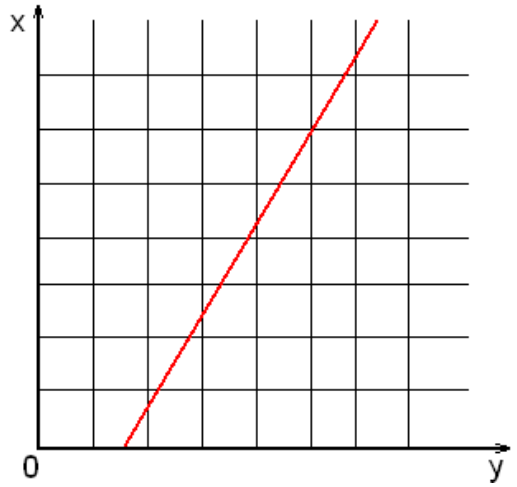
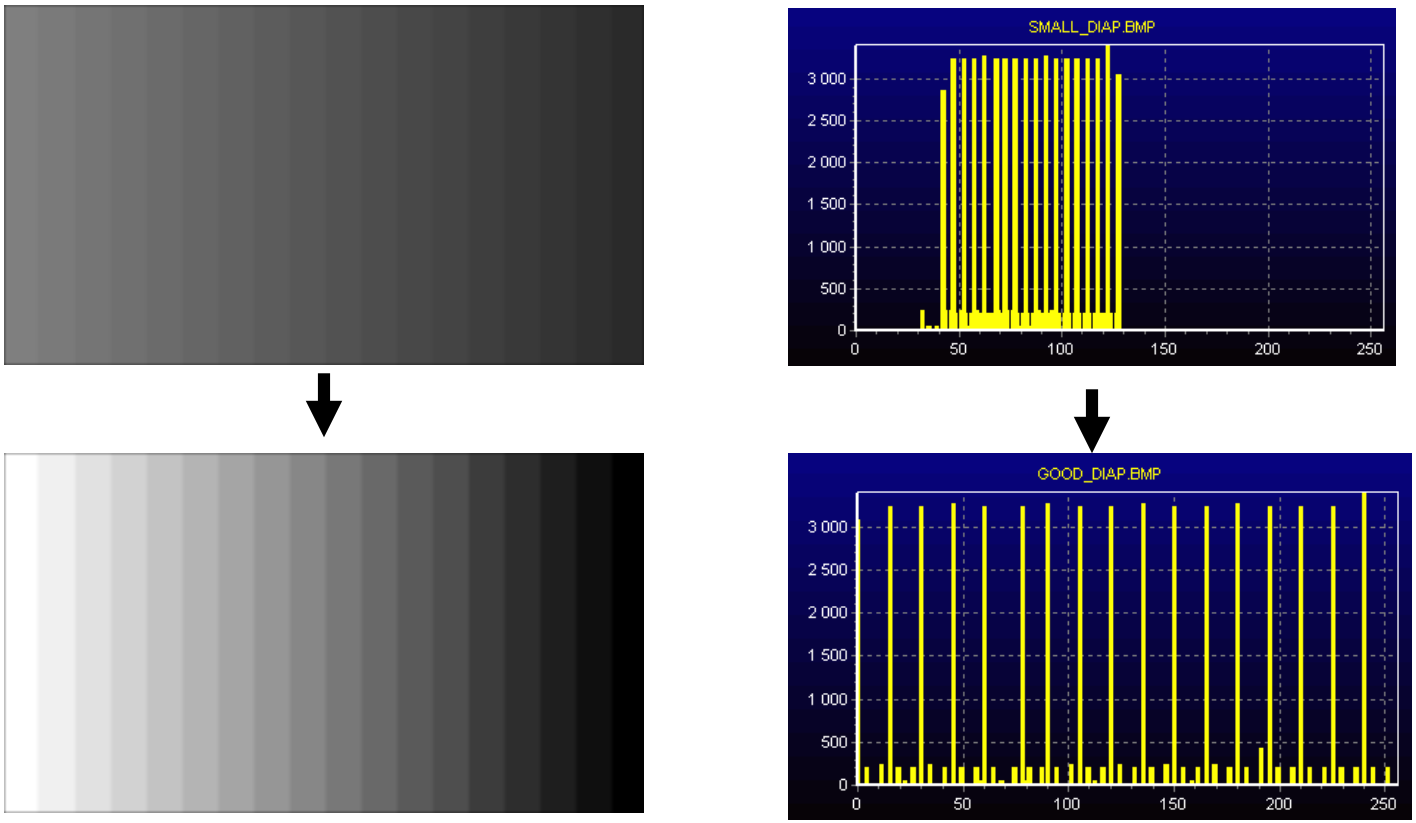


График функции $f^{-1}(y)$

Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение гистограммы:



Робастная линейная коррекция

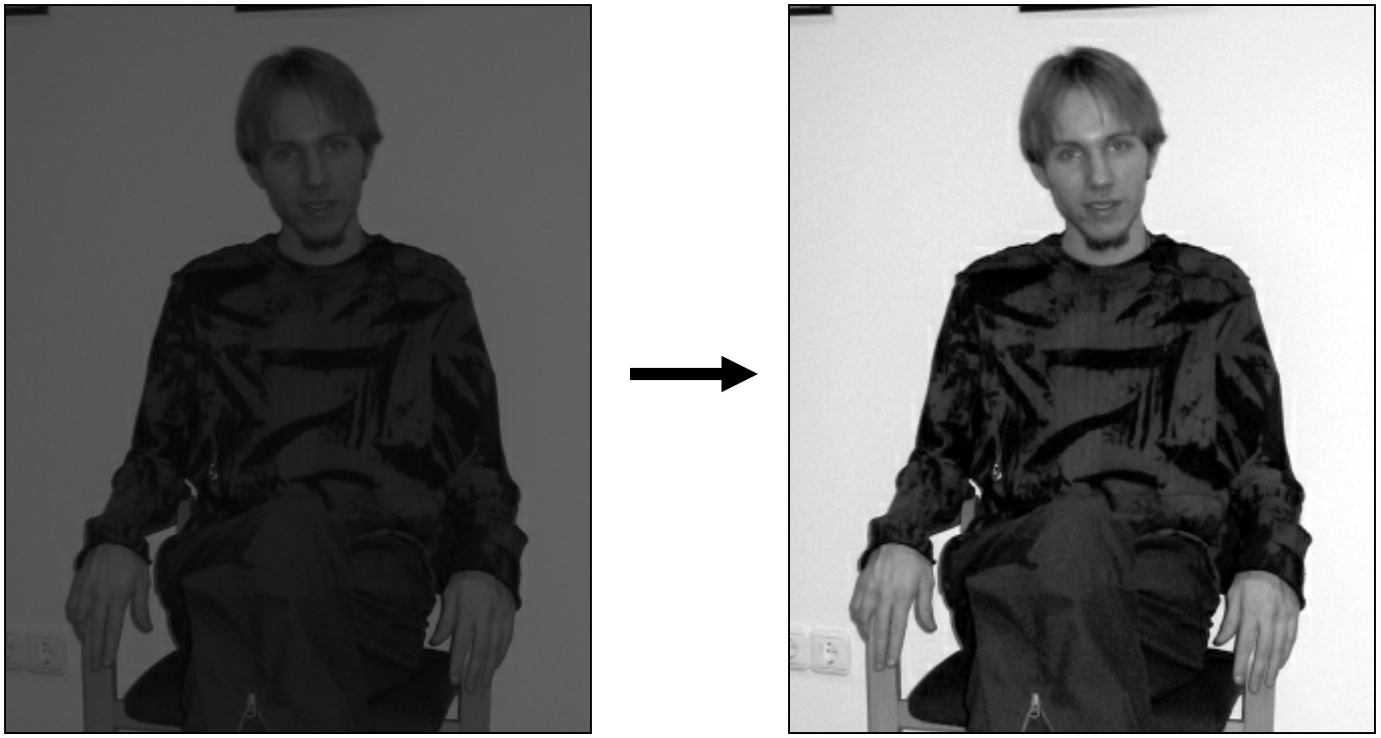
Что будет при применении линейной коррекции к такой картинке?

Робастная (устойчивая) версия метода:

- Вычислим такую линейную коррекцию, чтобы 5% самых темных пикселов стали черными и 5% самых светлых стали белыми

Линейная коррекция

Линейное растяжение – «как AutoContrast в Photoshop»



Линейная коррекция

Линейная коррекция помогает не всегда!

К слову, в чём может быть причина дефекта такого изображения?

Нелинейная коррекция

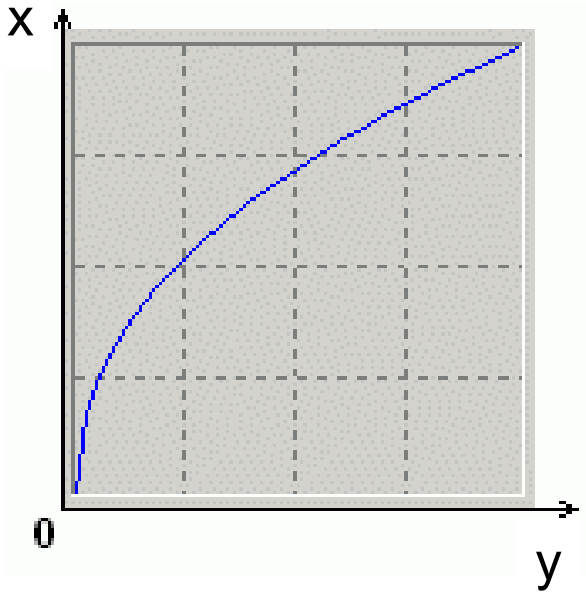
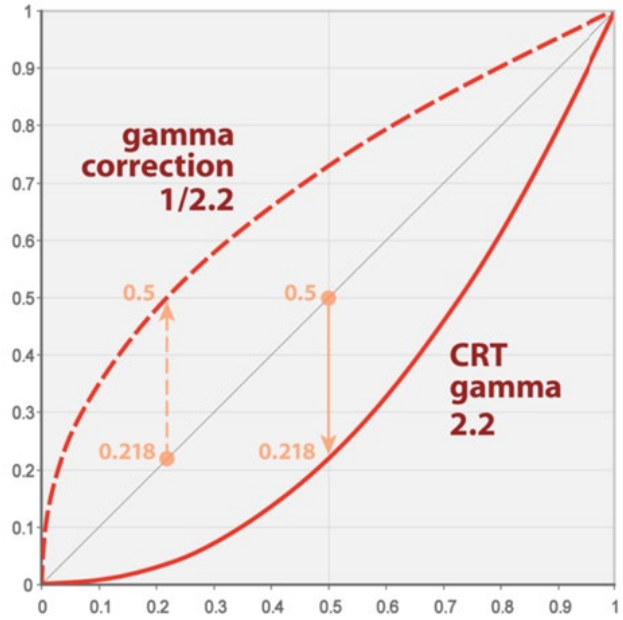


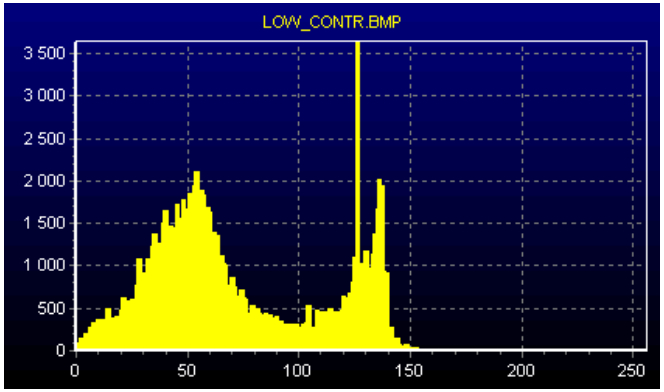
График функции $f^{-1}(y)$

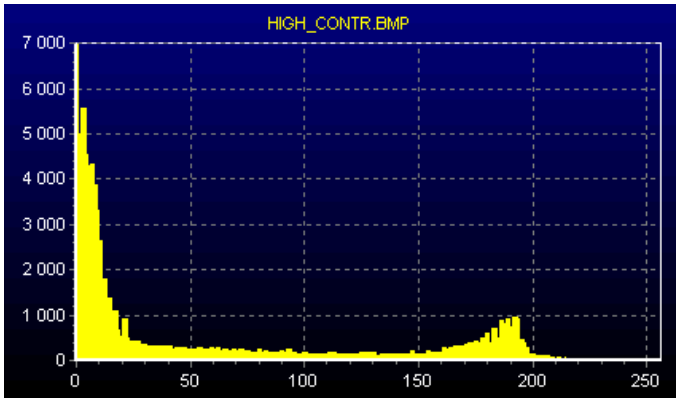
Гамма-коррекция



$$y = c \cdot x^{\gamma}$$

Произвольная нелинейная коррекция





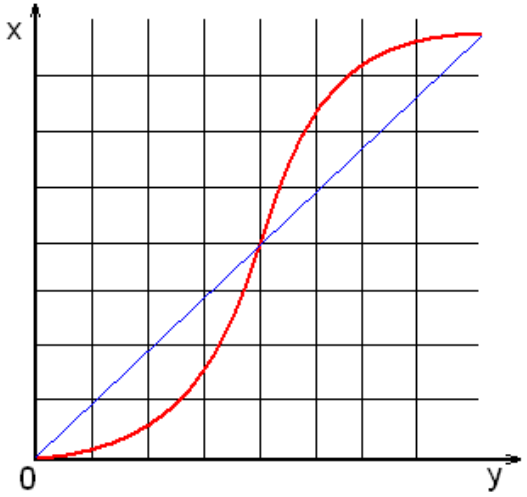
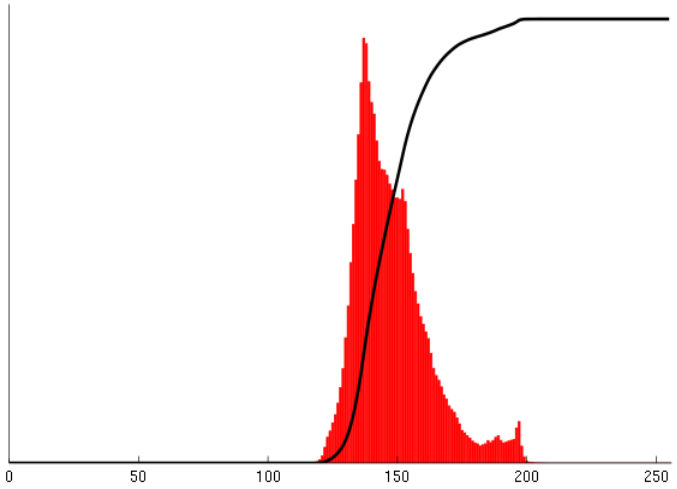
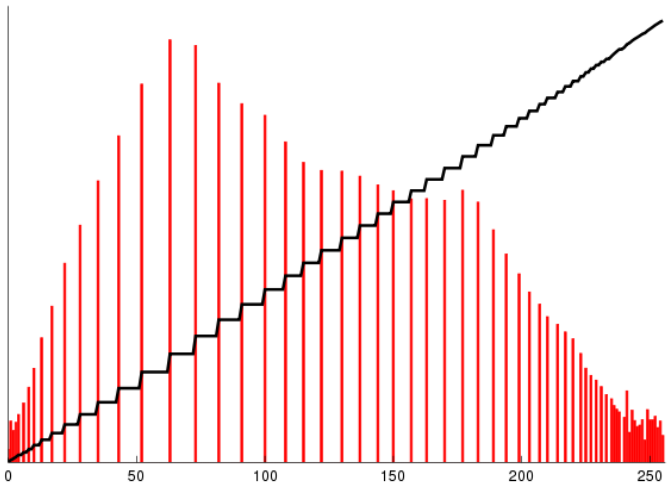


График функции $f^{-1}(y)$

Выравнивание гистограммы





https://en.wikipedia.org/wiki/Histogram_equalization

Цветокоррекция

Цветокоррекция

Неправильный баланс

Правильный баланс

http://www.cambridgeincolour.com/tutorials/white-balance.htm

- Как определить, что цветопередача неправильная?
- Как скорректировать изображение?

Коррекция по шаблону

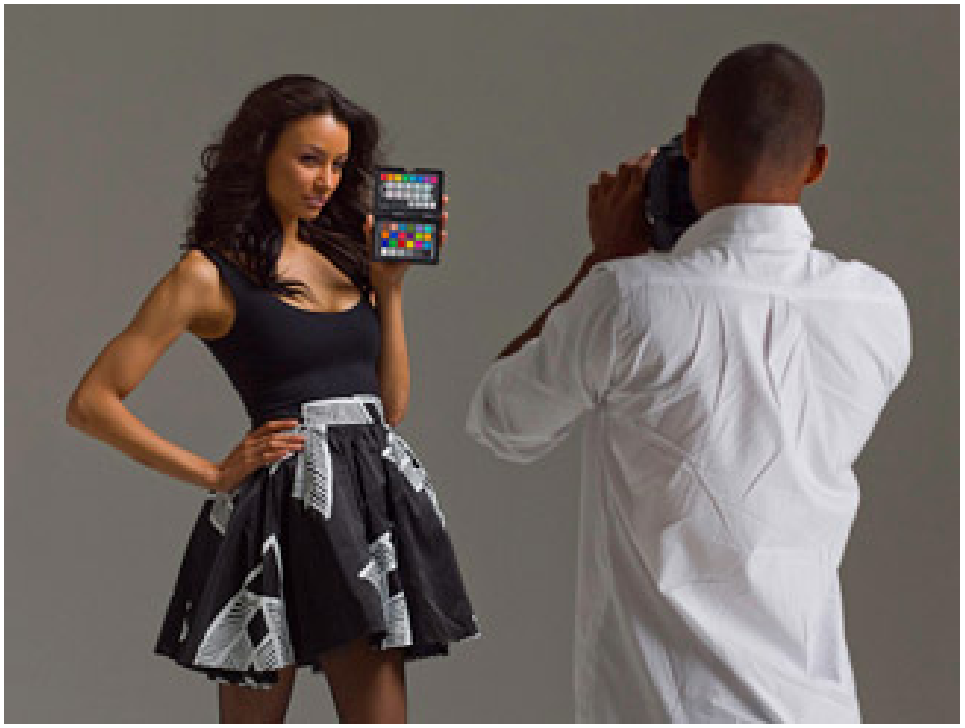
- Разумный подход:
  - Сфотографировать объект с известным цветом (шаблон)
  - Вычислить цветовое преобразование, чтобы цвет объекта на фотографии совпал с нужным
- Простейшая реализация:
  - Возьмём однотонные карточки (белые)
  - Будем домножать каждый канал отдельно, чтобы цвет карточек стал белым
  - Вычисление коэффициентов - Если цвет объект записывается как r_w, g_w, b_w , тогда веса $1/r_w$, $1/g_w$, $1/b_w$

Насколько такое преобразование корректно, какие могут быть недостатки?

Slide by S. Lazebnik

Профессиональная цветокоррекция

Source: The dark knight



Source: http://x-rite.com

Используем цветной шаблон с многими цветами

Какое преобразование в камере?

Сложные модели

http://vision.middlebury.edu/color/

- Авторы собрали большую коллекцию разных изображений для оценки различных моделей преобразования в камере
- Полиномиальная модель (24 параметра)

$$y_i = g_i([M_D k]_i)$$

A. Chakrabarti, D. Scharstein, and T. Zickler. An empirical camera model for Internet color vision. BMVC 2009

Оценка параметров цветокоррекции

Если нет цветовых шаблонов, тогда нам нужно угадать (или оценить) коэффициенты усиления

Модель «Серого мира» (Grayworld)

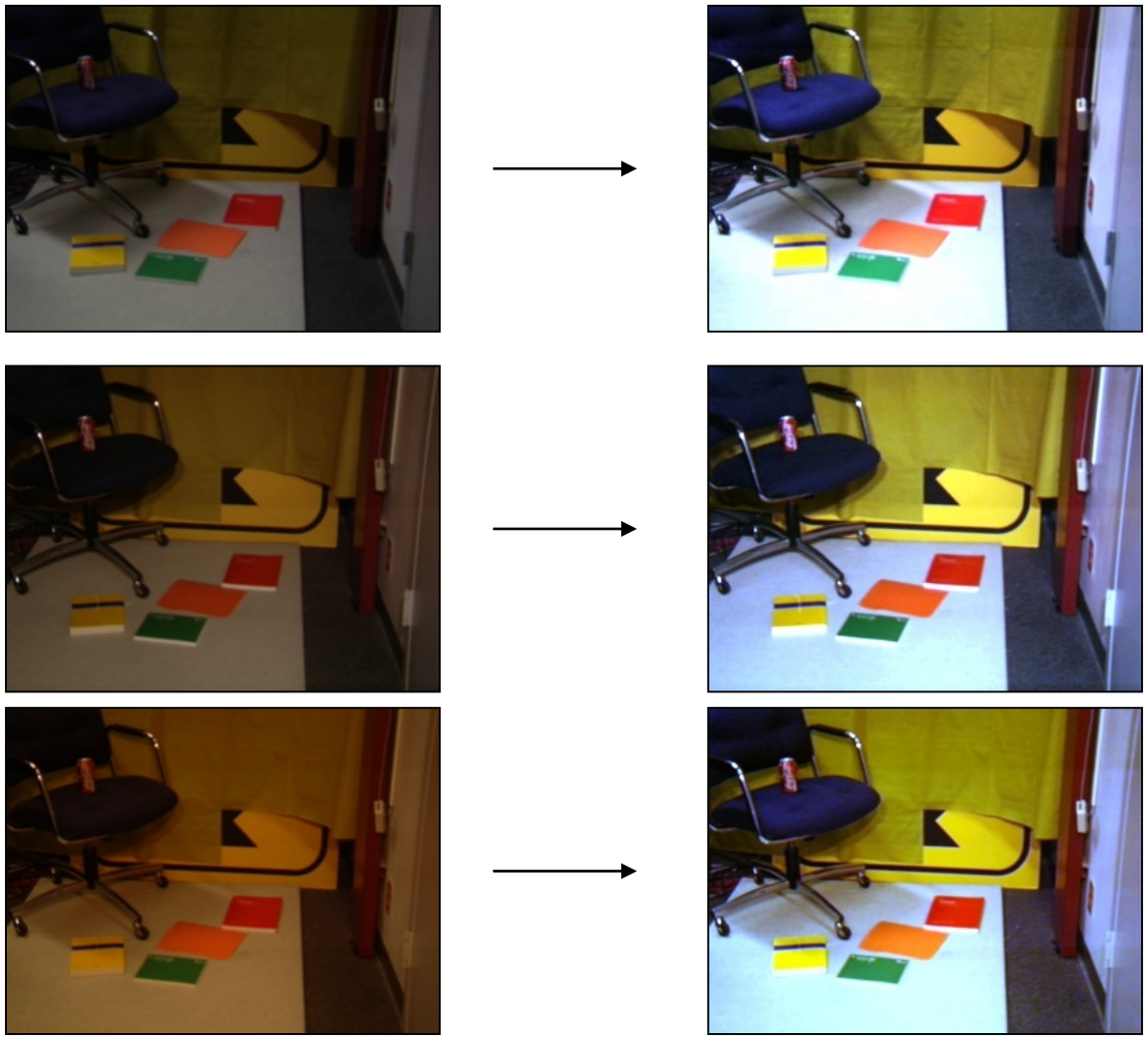
- Средний уровень («серый») по каждому каналу должен быть одинаков для всех каналов
- Если цветовой баланс нарушен, тогда «серый» в этом канале больше «серого» других каналов
- Вычислим коэффициенты усиления так, чтобы среднее в каждом канале стало одинаковым:

$$\bar{R} = \frac{1}{N}\sum R(x,y); \quad \bar{G} = \frac{1}{N}\sum G(x,y); \quad \bar{B} = \frac{1}{N}\sum B(x,y); \quad Avg = \frac{\bar{R}+\bar{G}+\bar{B}}{3};$$

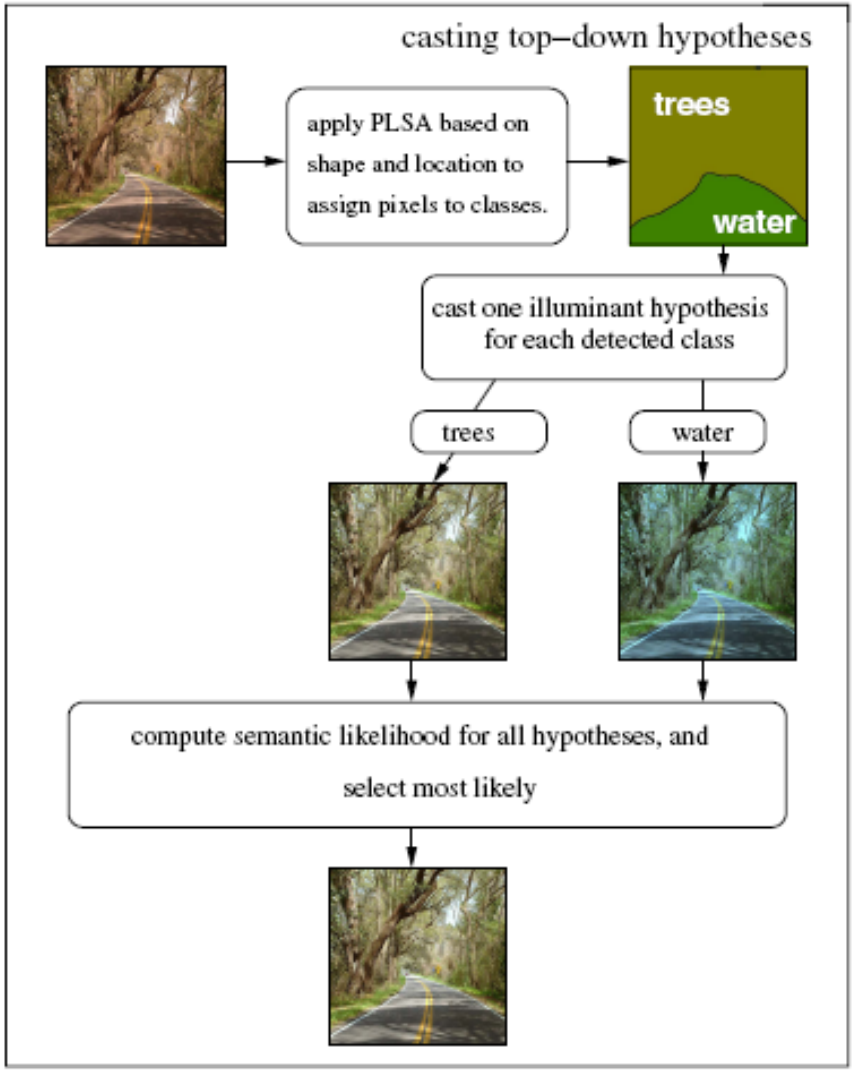
$$R' = R \cdot \frac{Avg}{\bar{R}}; \quad G' = G \cdot \frac{Avg}{\bar{G}}; \quad B' = B \cdot \frac{Avg}{\bar{B}}$$

Slide by S. Lazebnik

«Серый мир» - примеры



Распознавание баланса белого



Методы цветовой коррекции до сих пор развиваются.

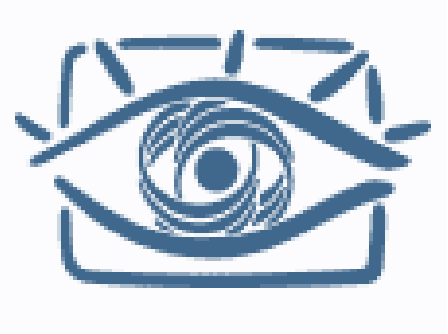
Пример: Для каждого класса объектов, присутствующих в сцене, вычисляем преобразование таким образом, чтобы диапазон цветов объекта совпадал со средним диапазоном объектов этого класса на «типичных» изображениях

J. Van de Weijer, C. Schmid and J. Verbeek, Using High-Level Visual Information for Color Constancy, ICCV 2007.

Slide by S. Lazebnik

Резюме лекции

- Понятие о компьютерном зрении
- Устройство глаза и фотокамеры
- Трихроматическая теория цвета
- Цветовые модели RGB, CMYK, YIQ, HSV
- Тональная коррекция
- Цветокоррекция



Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №2
«Основы обработки изображений»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

16 сентября 2019 года

Обработка изображений

- Семейство методов и задач, где входной и выходной информацией являются изображения
- Формально $Y = f(X)$, где $X, Y \in R^{n \times m \times k}$ - изображения с k-каналами, разрешением n на m пикселей

Цели:

1. Улучшение изображения для восприятия человеком
2. Улучшение изображения для восприятия компьютером
3. Извлечение признаков изображений для последующего анализа
4. Преобразование для технических нужд
5. Развлечение (спецэффекты)

Свёртка изображения

Пример приложения - шумоподавление

Шум фотоаппарата

- Изображения обычно содержат «шум», т.е. значение пикселя отличается от истинного
- Нужно получить «чистое» изображение без шума («подавить шум»)

Виды шума

Original

Gaussian noise

Salt and pepper noise

Impulse noise

- **Гауссов** (*Gauss noise*)
  - Аддитивный шум
  - Image(i,j) = true(i,j) + noise(i,j)
  - Колебания яркости, распределенные по нормальному закону
  - $Noise(i,j) \sim N(\mu, \sigma)$

- **Потеря информации (***data drop-out noise)*
  - **Соль и перец**: случайные черные и белые пиксели
  - **Импульсный:** случайные белые пиксели

Source: S. Seitz

Усреднение нескольких кадров

«Временная фильтрация» - усреднение пикселей между несколькими кадрами

Зашумленные изображения

Усреднение по 10 изображениям

$$I(i,j) = g_r(i,j) + Err(i,j);$$

$$\bar{I}(i,j) = \frac{1}{N}\sum_{k=1}^{N} I_k(i,j);$$

$$E(\bar{I}(i,j)) = g_r(i,j);$$

Какой вид шума можем подавить?

Шумоподававление в одном кадре

Как быть, если изображение только одно?

Пространственная фильтрация

- Вычислим новое значение y_{ij} для пиксела x_{ij} как функцию от его локальной окрестности:

$$y_{ij} = f([x_{kl}]), x_{kl} \in neighbour(x_{ij})$$

- В простейшем случае мы возьмём просто взвешенное среднее по всем пикселам из окрестности
- Веса обозначаются как *ядро фильтра*
- Веса для усреднения задаются так:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

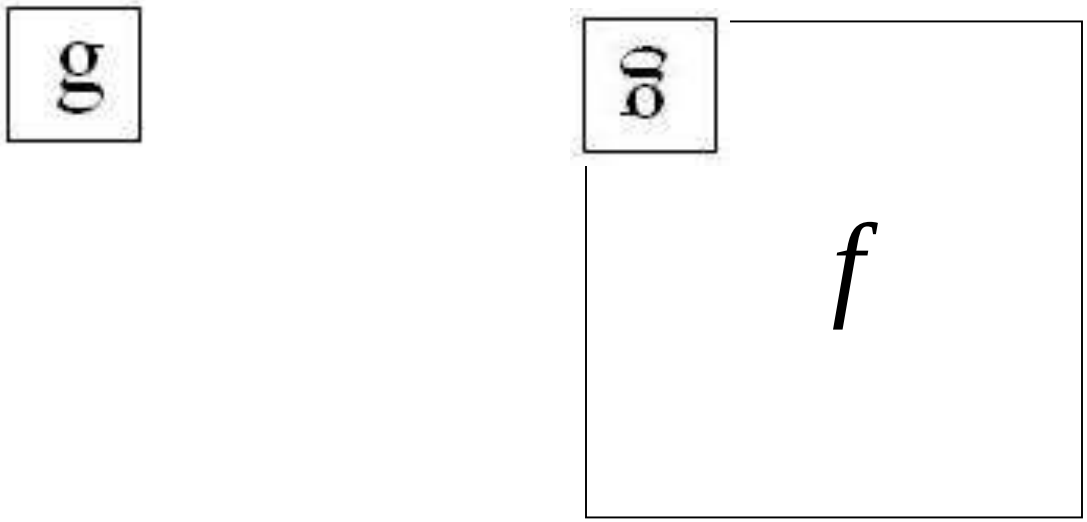
“box filter”

Source: D. Lowe

Свертка

- Пусть *f – изображение, g -ядро*. Свертка изображения *f* с помощью *g* обозначается как *f * g* и называется:

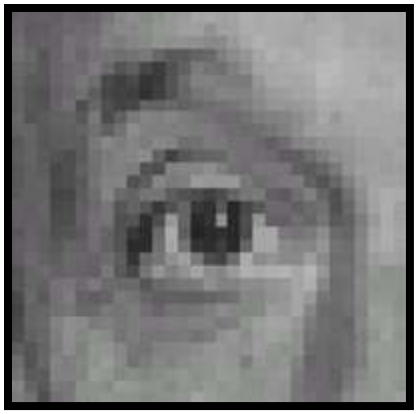
$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$



Свёртка – ядро фильтра «перевёрнуто»!

Source: F. Durand

Примеры фильтров



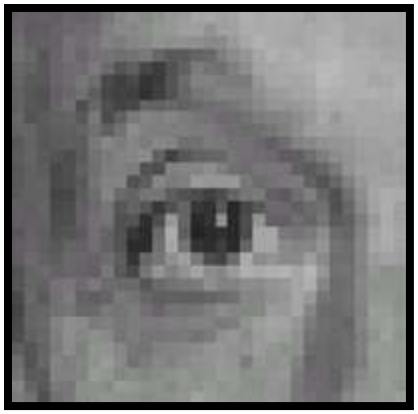
Original

0	0	0
0	1	0
0	0	0

?

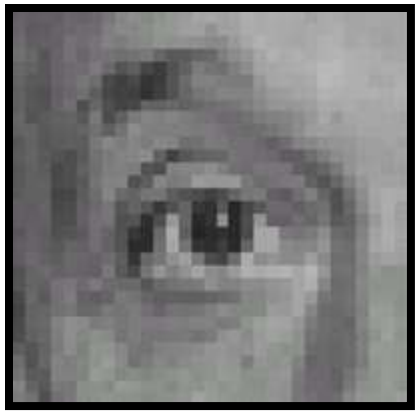
Source: D. Lowe

Примеры фильтров



Original

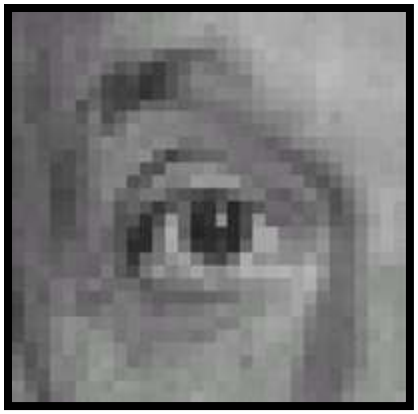
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Примеры фильтров



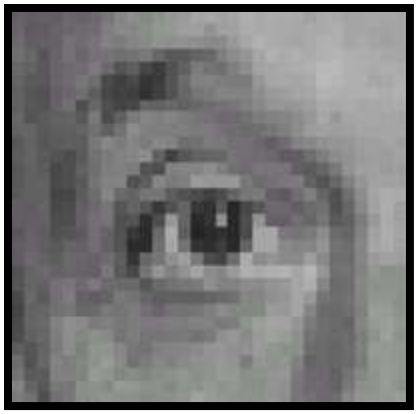
Original

0	0	0
0	0	1
0	0	0

?

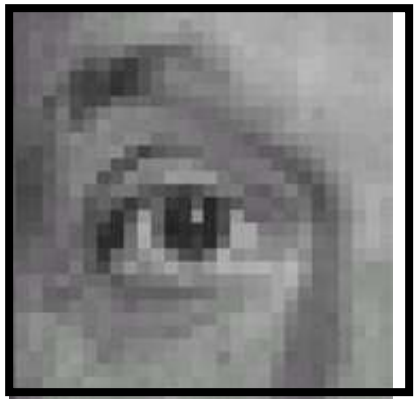
Source: D. Lowe

Примеры фильтров



Original

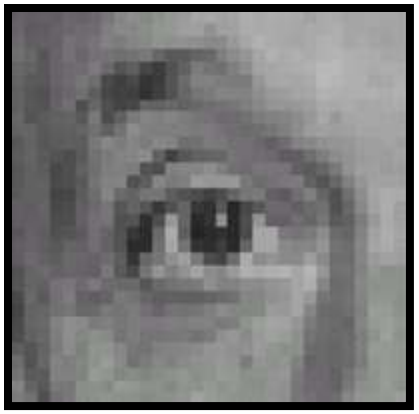
0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Примеры фильтров



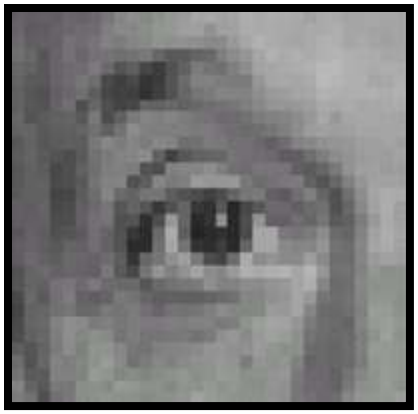
Original

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

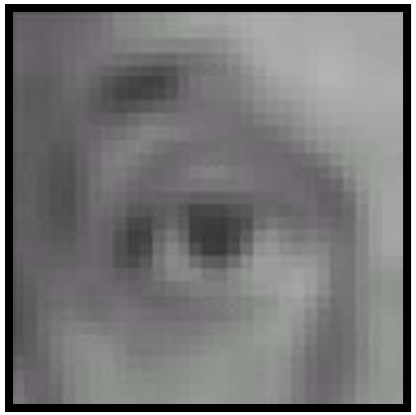
Source: D. Lowe

Примеры фильтров



Исходное

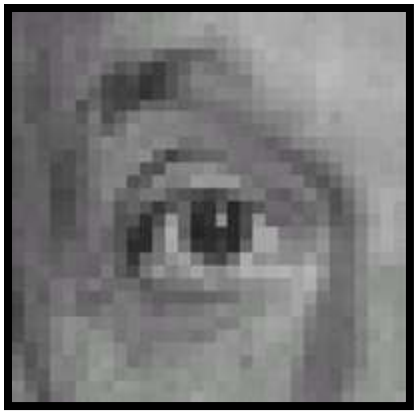
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



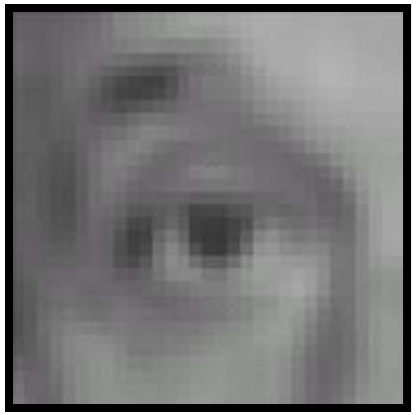
Результат

Source: D. Lowe

Примеры фильтров



$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Исходное

Результат

Любой фильтр со всеми положительными весами и суммой весов = 1 будет сглаживать изображение

Source: D. Lowe

Примеры фильтров

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

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

Примеры фильтров

$$\frac{1}{10} \cdot \begin{Vmatrix} -1 & -2 & -1 \\ -2 & 22 & -2 \\ -1 & -2 & -1 \end{Vmatrix}$$

Визуализация и варианты реализации

https://ezyang.github.io/convolution-visualizer/index.html

Линейные фильтры и свёртка

- Пространственный фильтр называется **линейным**, если выполняются свойства **линейности**:
  - filter($f_1 + f_2$) = filter(f_1) + filter(f_2)
  - filter(af_1) = a filter(f_1)
- Фильтр называются инвариантным к сдвигу, если выполняется следующее условие:

  filter(shift(f)) = shift(filter(f))
- Теорема: любой линейный оператор, инвариантный к сдвигу, может быть записан в виде свертки
- Чтобы доказать нелинейность фильтра, можно воспользоваться основными свойствами, и показать их не выполнение на примере

Slide by S. Lazebnik

Свойства

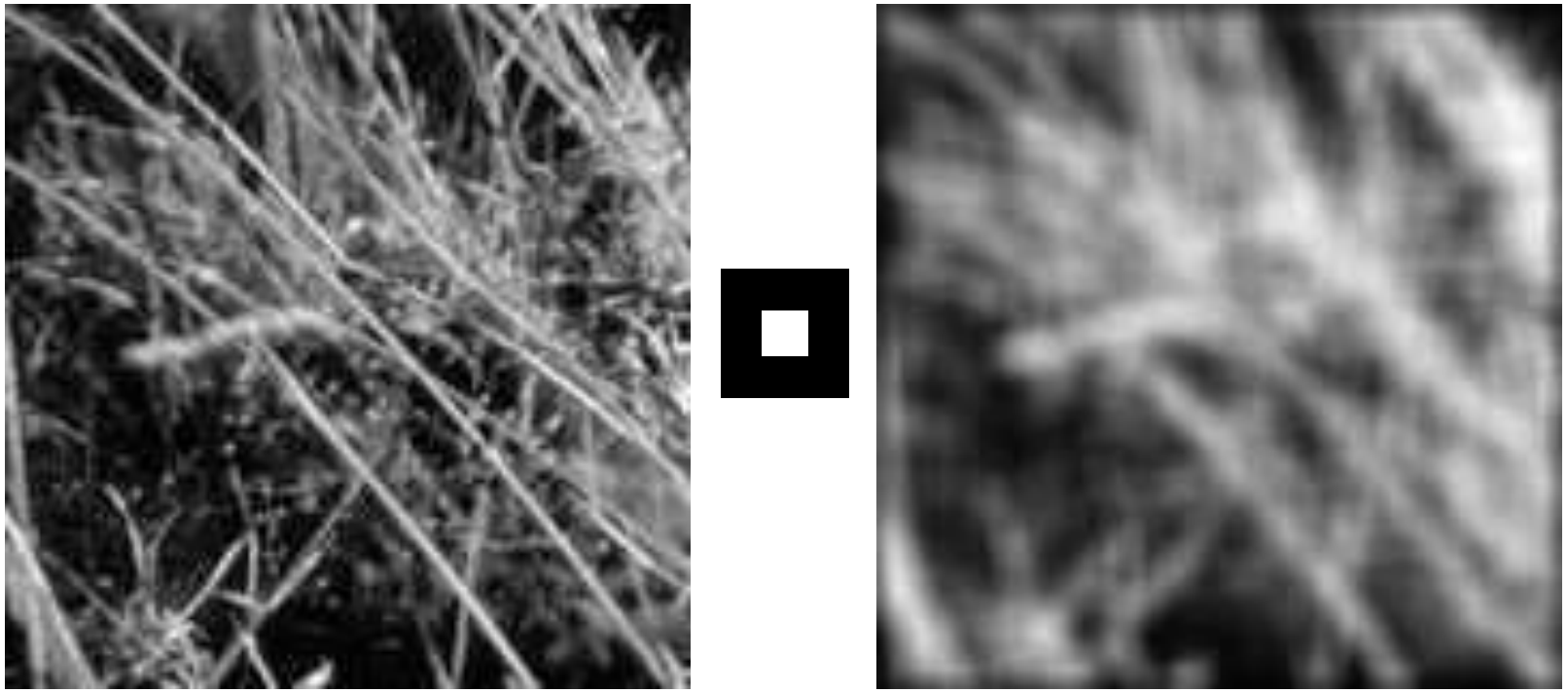
- Коммутативность: $a * b = b * a$
  - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $a * (b * c) = (a * b) * c$
  - Последовательное применение фильтров: $(((a * b_1) * b_2) * b_3)$
  - Эквивалентно применению такого фильтра: $a * (b_1 * b_2 * b_3)$
- Дистрибутивность по сложению:

  $a * (b + c) = (a * b) + (a * c)$
- Домножение на скаляр можно вынести за скобки: $ka * b = a * kb = k (a * b)$
- Единица: $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$,
  $a * e = a$

Slide by S. Lazebnik

Вернёмся к шумоподавлению

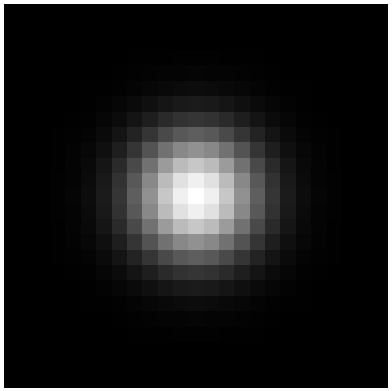
При сглаживании с box-фильтром на изображении могут образовываться паразитные линии



Source: D. Forsyth

Сглаживание

- Точка света, наблюдаемая с расфокусированного объектива, выглядит как кружок света, а усреднение (box) дает квадратик
- Другой способ: взвешиваем вклад пикселей по окрестности с учетом близости к центру

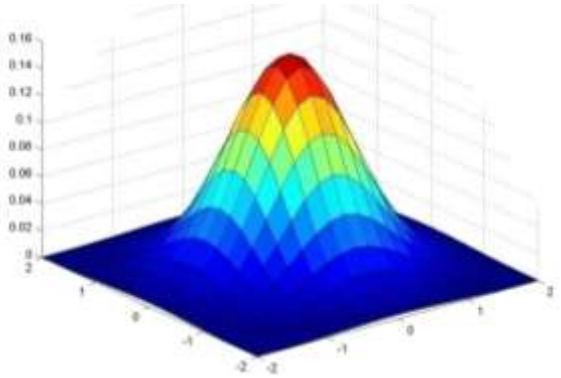


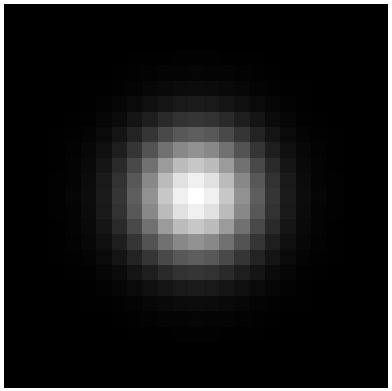
“размытый кружок”

Slide by S. Lazebnik

Ядро фильтра гаусса

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$





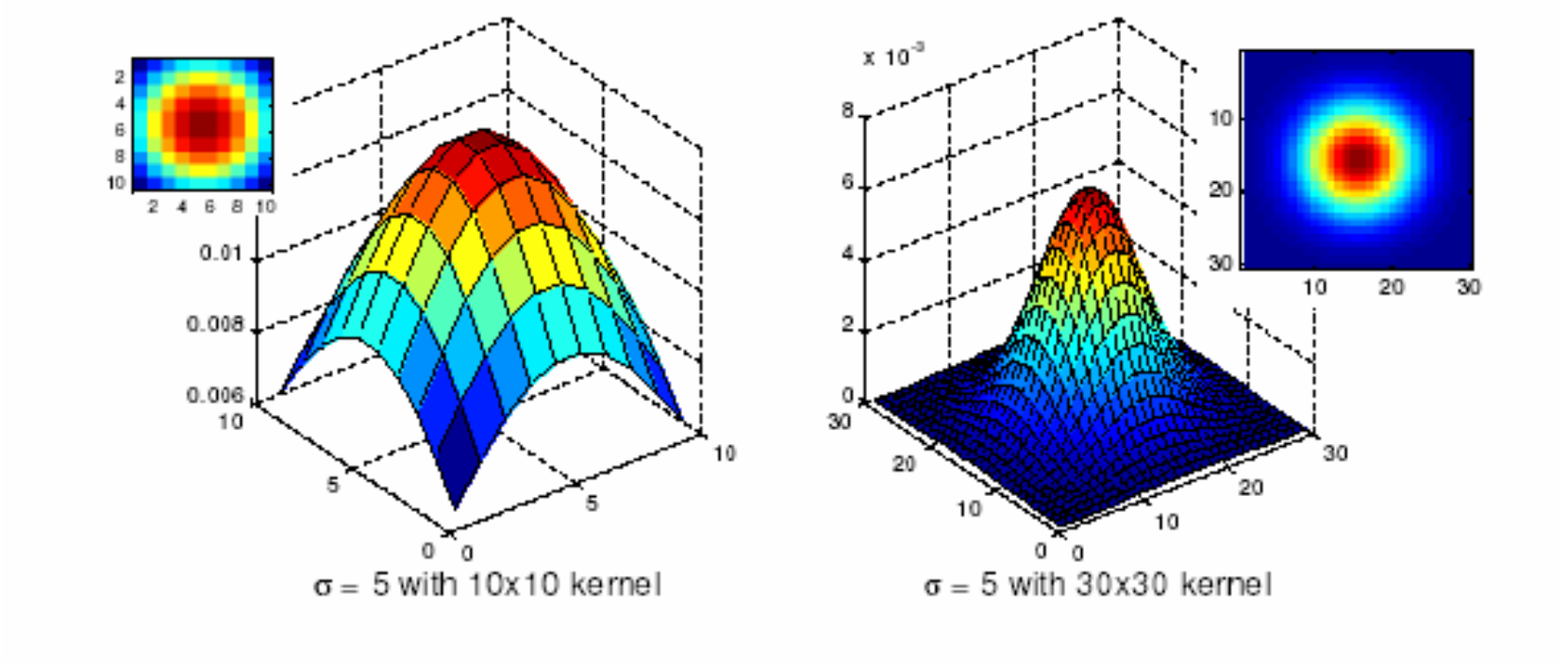
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

Source: C. Rasmussen

Выбор размера ядра

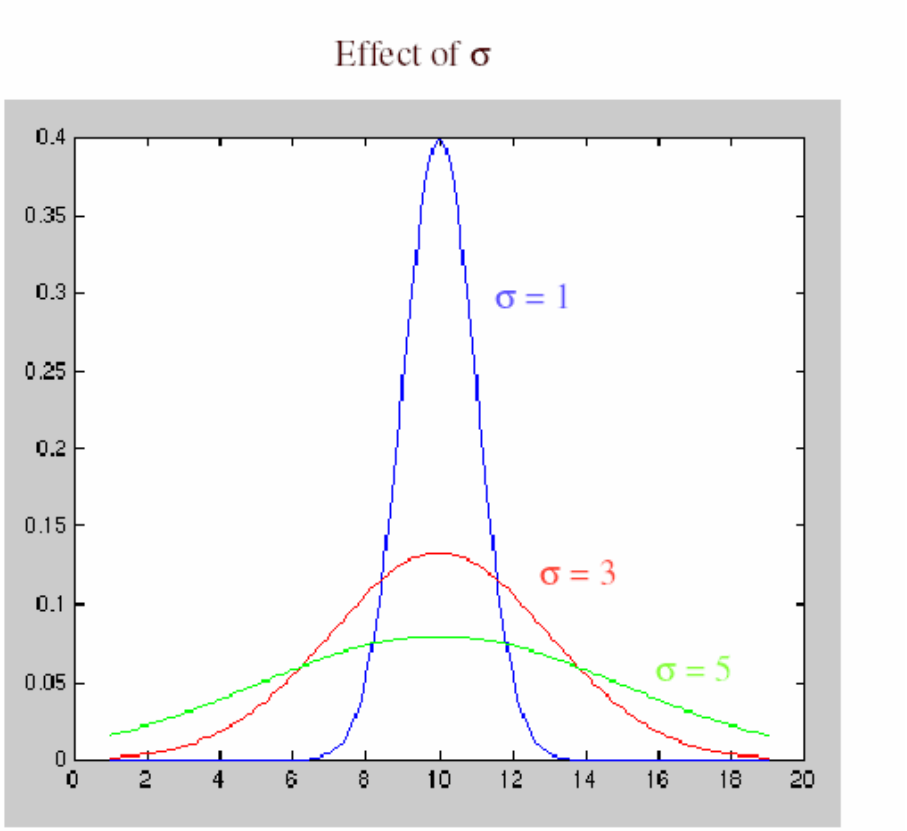
Размер ядра дискретного фильтра ограничен



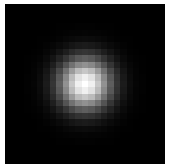
Source: K. Grauman

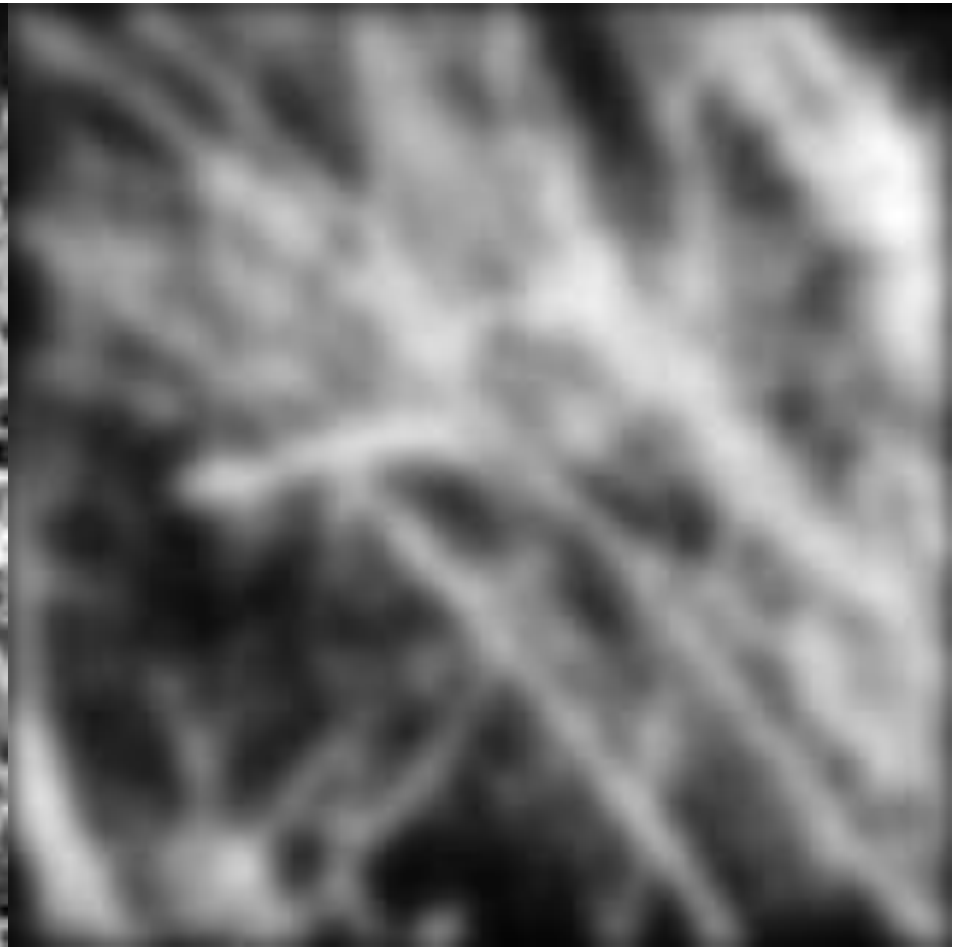
Выбор размера ядра

Эмпирика: полуразмер фильтра равен 3σ

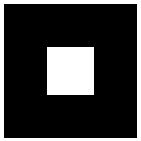


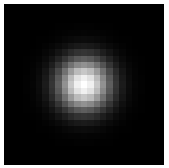
Сглаживание фильтром гаусса

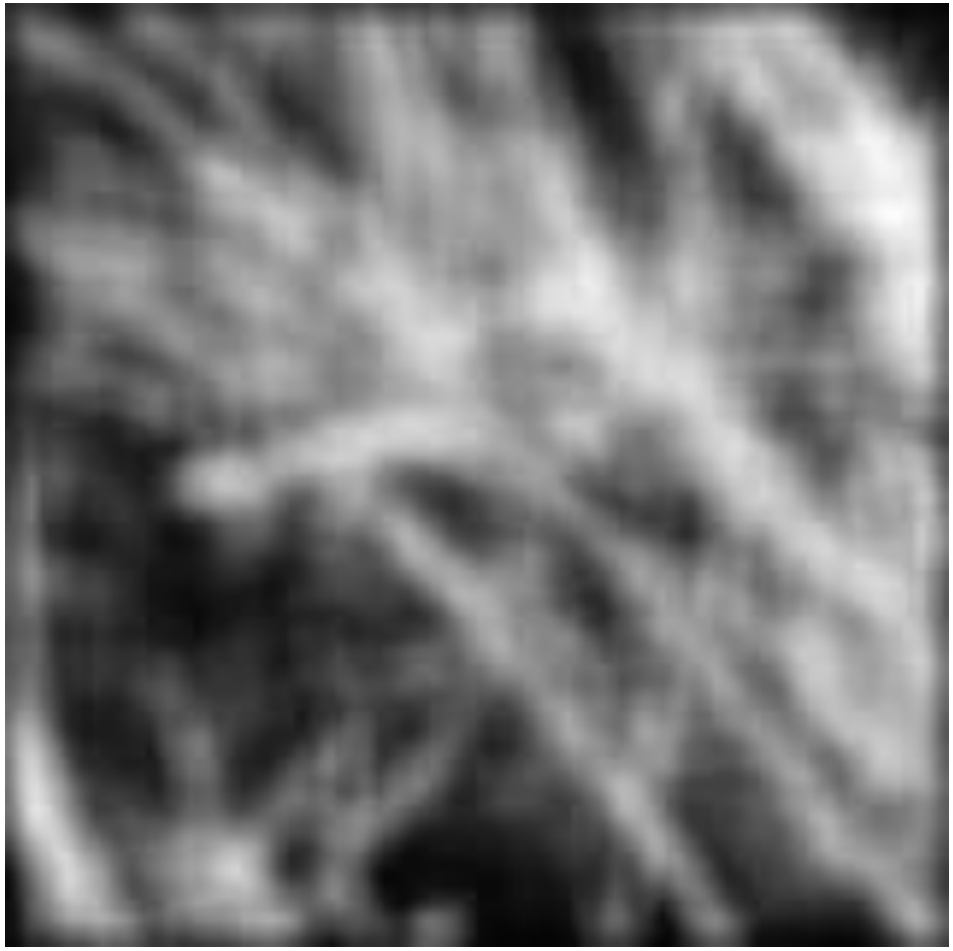


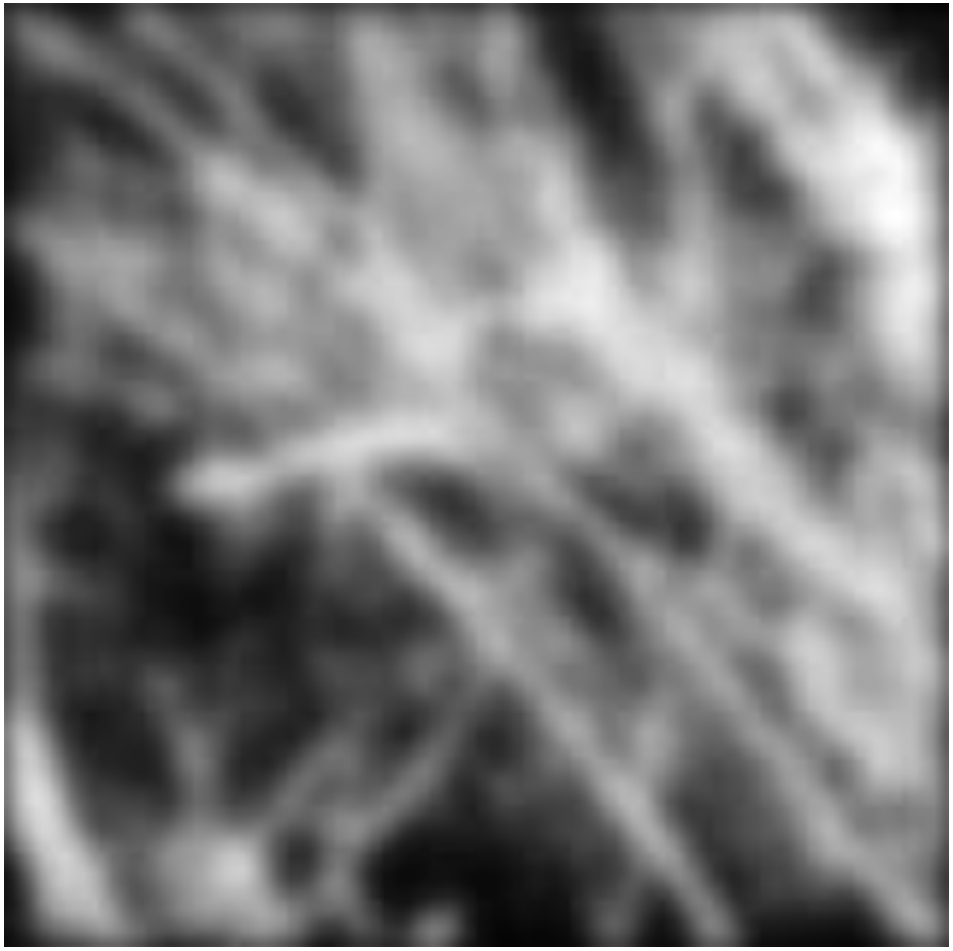


Сравнение бокс и гаусс-фильтра

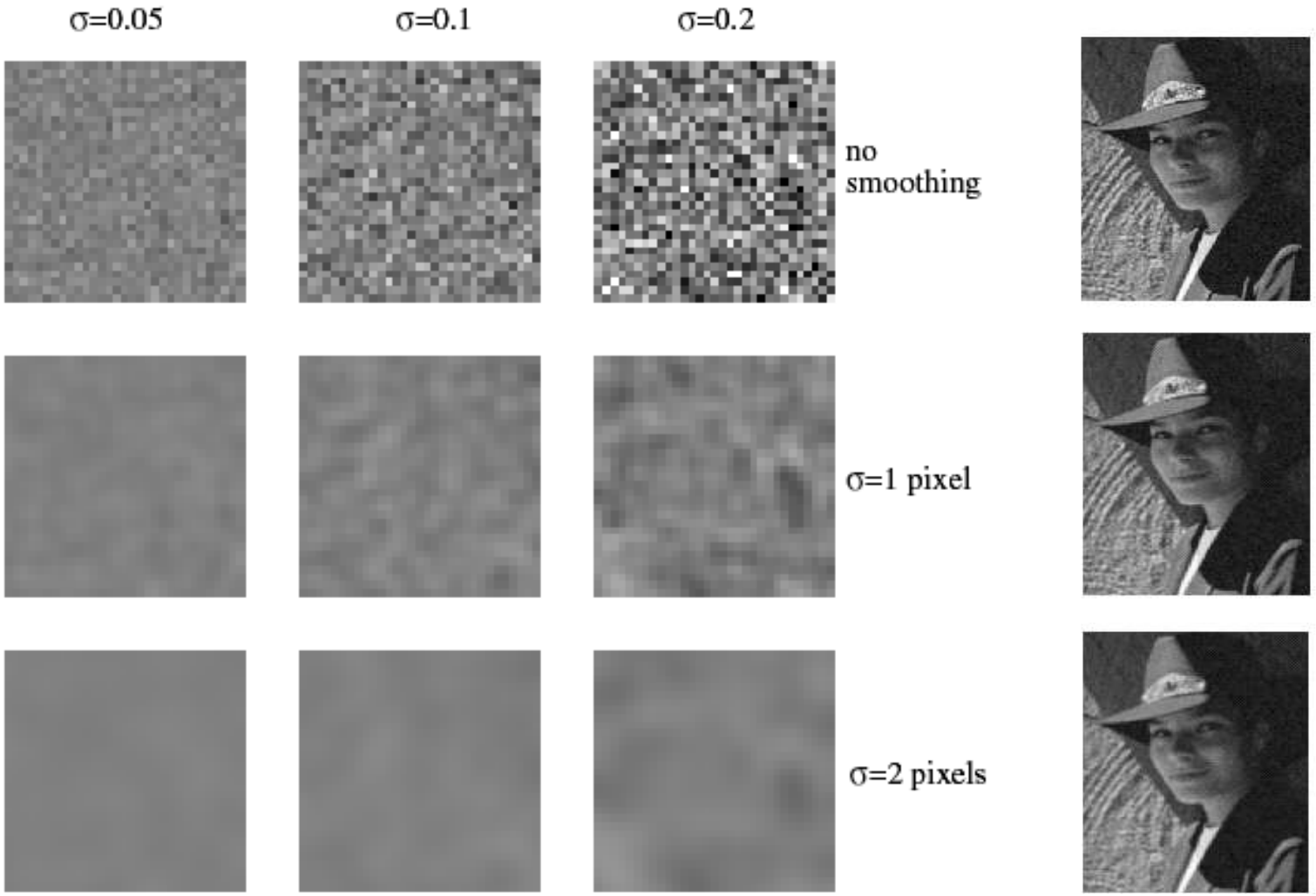








Подавление гауссова шума



Сглаживание фильтрами большого радиуса подавляет шум, но размывает изображение

Slide by S. Lazebnik

Свойства фильтра Гаусса

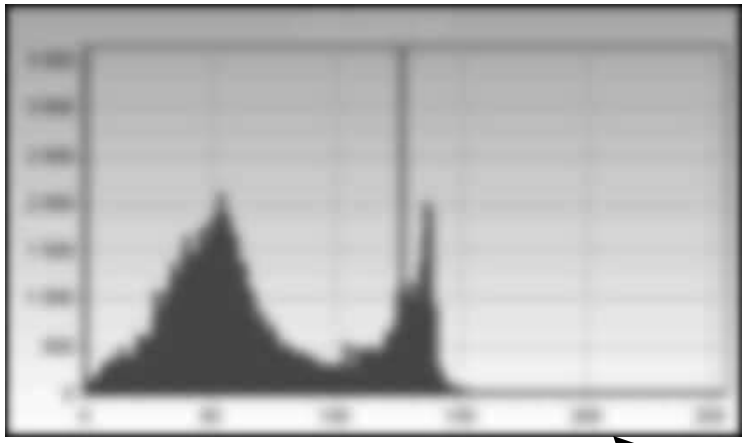
- Свертка с сами собой дает тоже фильтр гаусса
  - Сглаживание несколько раз фильтром с маленьким ядром дает результат, аналогичный свертке с большим ядром
  - Свертка 2 раза с фильтром радиуса σ дает тот же результат, что с фильтром радиуса $\sigma\sqrt{2}$
- Сепарабельное ядро
  - Раскладывается в произведение двух одномерных фильтром гаусса

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

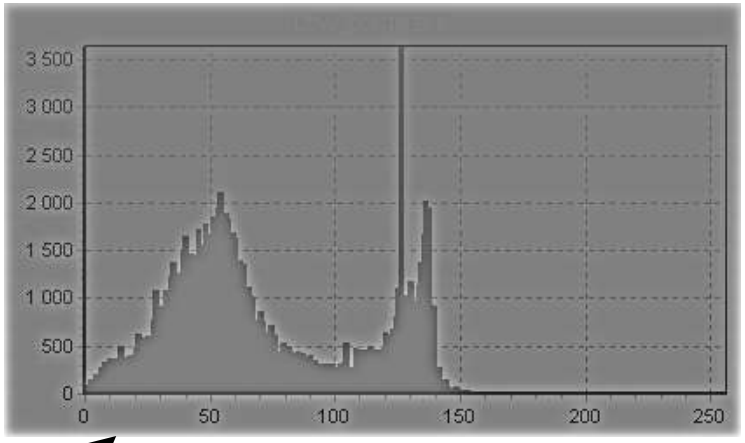
$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{y^2}{2\sigma^2}}\right)$$

Source: K. Grauman

Маленькая экскурсия к Фурье

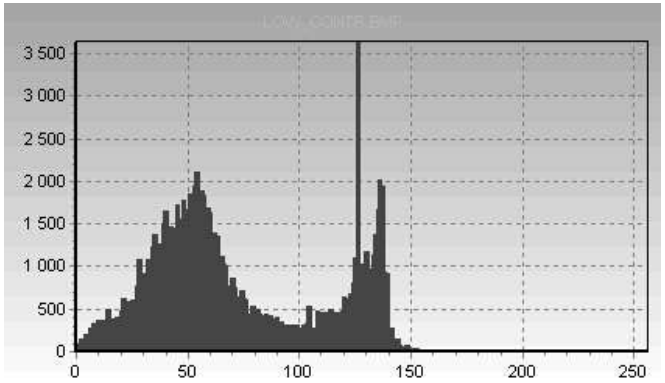


+



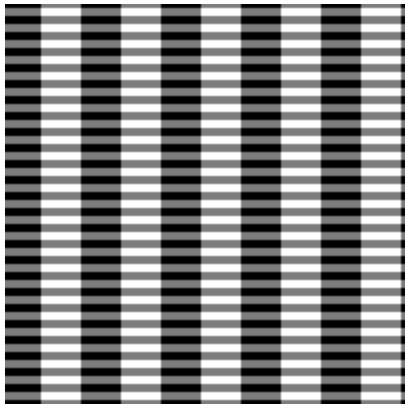
Низкие частоты

Высокие частоты

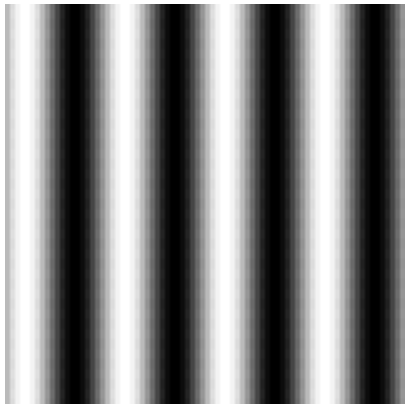


Фильтр Гаусса

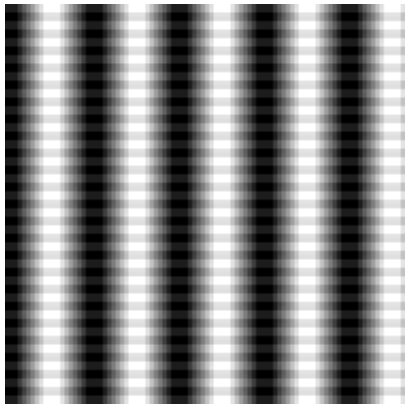
Результат свертки фильтром гаусса и усреднения



Исходное изображение



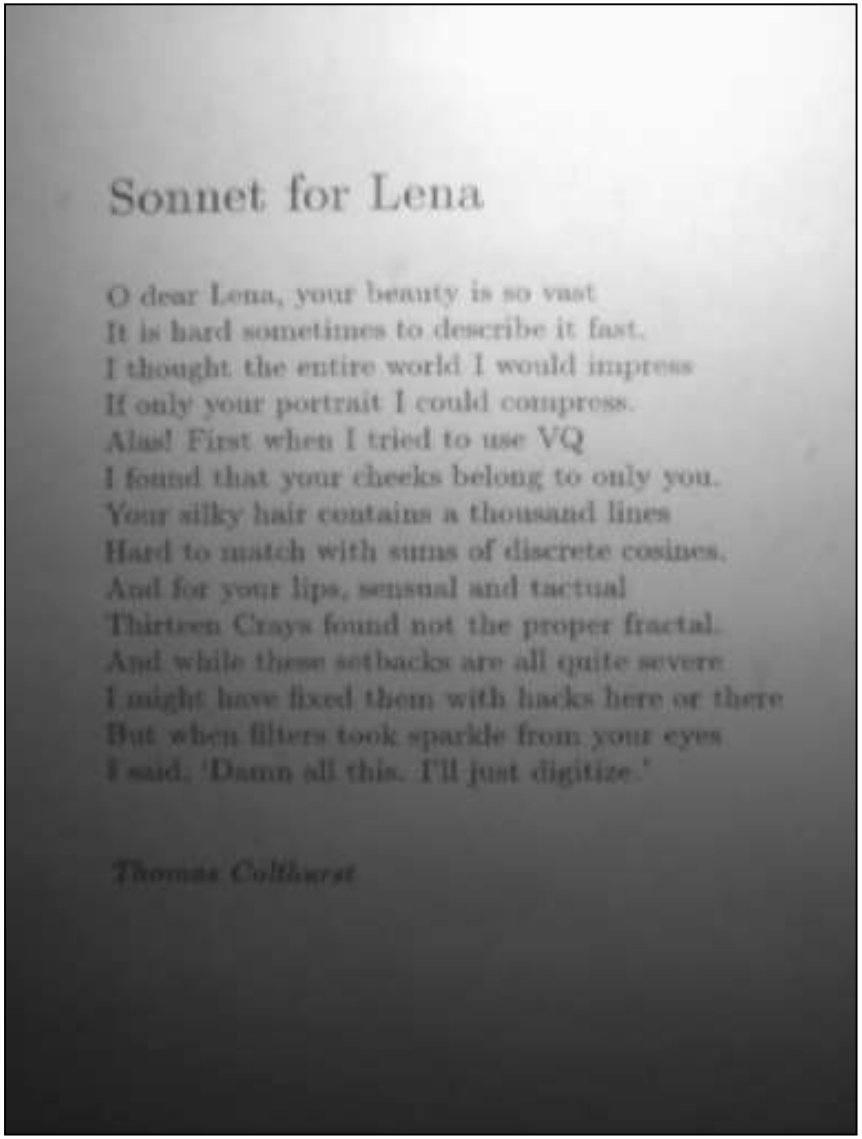
Фильтр Гаусса с Sigma = 4



Усреднение по 49 пикселям (7x7)

Важное свойство фильтра Гаусса – он по сути является фильтром низких частот.

Пример применения фильтра Гаусса



Что можно сказать
про это изображение?

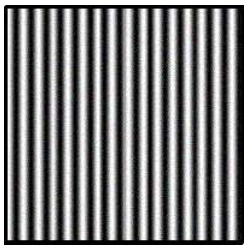
Компенсация разности освещения

Идея:

Формирование изображения:

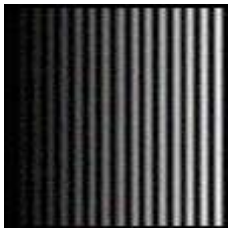
$$I(i, j) = l(i, j) \cdot r(i, j)$$

Плавные изменения яркости относятся к освещению, резкие - к объектам.



объект $r(i, j)$

освещение $l(i, j)$



Изображение освещенного объекта $I(i, j)$

Алгоритм Single scale retinex (SSR)

- Получить приближенное изображение освещения путем низочастотной фильтрации

$$\hat{l}(i, j) = G * I(i, j)$$

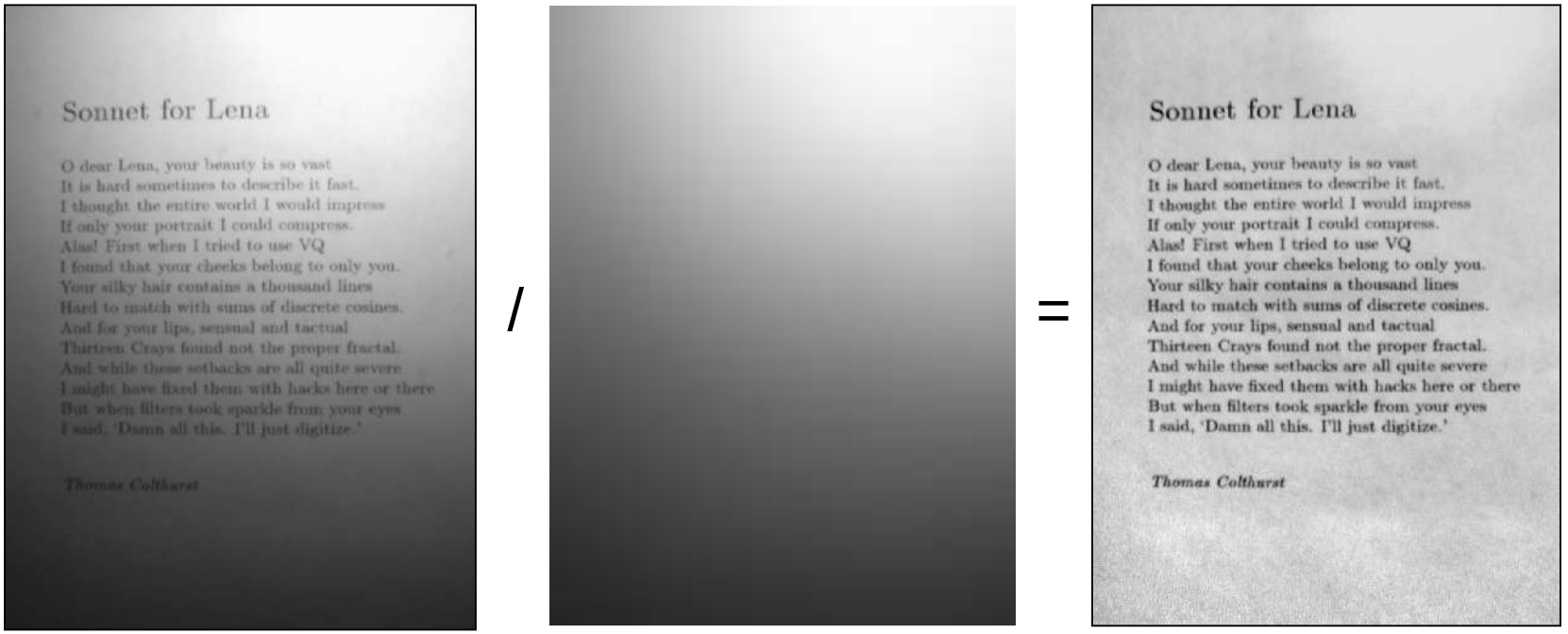
- Восстановить изображение по формуле

$$\hat{r}(i, j) = \frac{I(i, j)}{\hat{l}(i, j)}$$

- После преобразования потребуется применить тональную коррекцию и определить значения, которые будут соответствовать черному и белому

Компенсация разности освещения

Пример

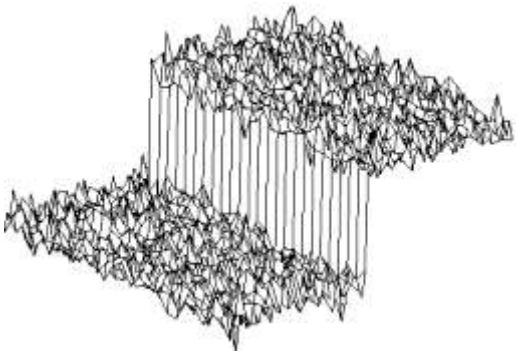


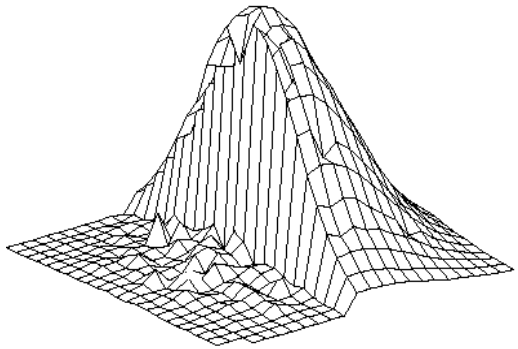
Gauss 14.7 пикселей

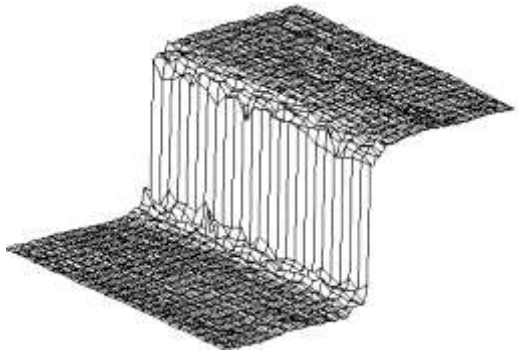
Билатеральный фильтр

- В операции свёртки каждый пиксель обрабатывается одинаково
- Но мы можем сделать параметры фильтра зависящими от изображения
- Билатеральный фильтр:

$$y_{ij} = \sum x_{i-k,j-l} \underbrace{w(x_{i-k,j-l} - x_{i,j})}_{\text{Вес, пропорциональный близости пикселов по цвет}} \underbrace{d(x_{i-k,j-l}, x_{i,j})}_{\text{Параметры фильтра Гаусса}}$$







Вернёмся к видам шума

Original

Gaussian noise

- **Гауссов**: колебания яркости, распределенные по нормальному закону
  - Аддитивный шум
  - Image(i,j) = true(i,j) + noise(i,j)
  - $Noise(i,j) \sim N(\mu, \sigma)$

Salt and pepper noise

Impulse noise

- **Потеря информации (***data drop-out noise)*
  - **Соль и перец**: случайные черные и белые пиксели
  - **Импульсный:** случайные белые пиксели

Source: S. Seitz

Подавление шума «соль и перец»

3x3

5x5

7x7

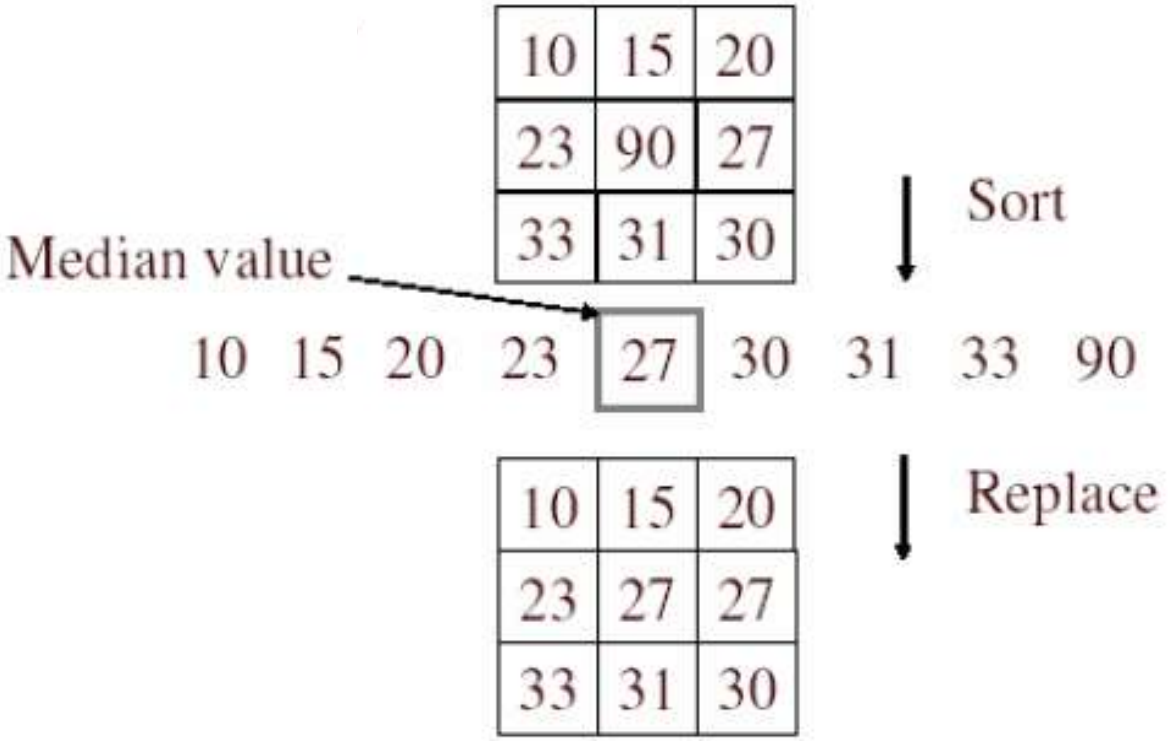
Применим фильтр Гаусса

Чем результат плох?

Slide by S. Lazebnik

Медианный фильтр

- Выбор медианы из выборки пикселей по окрестности данного

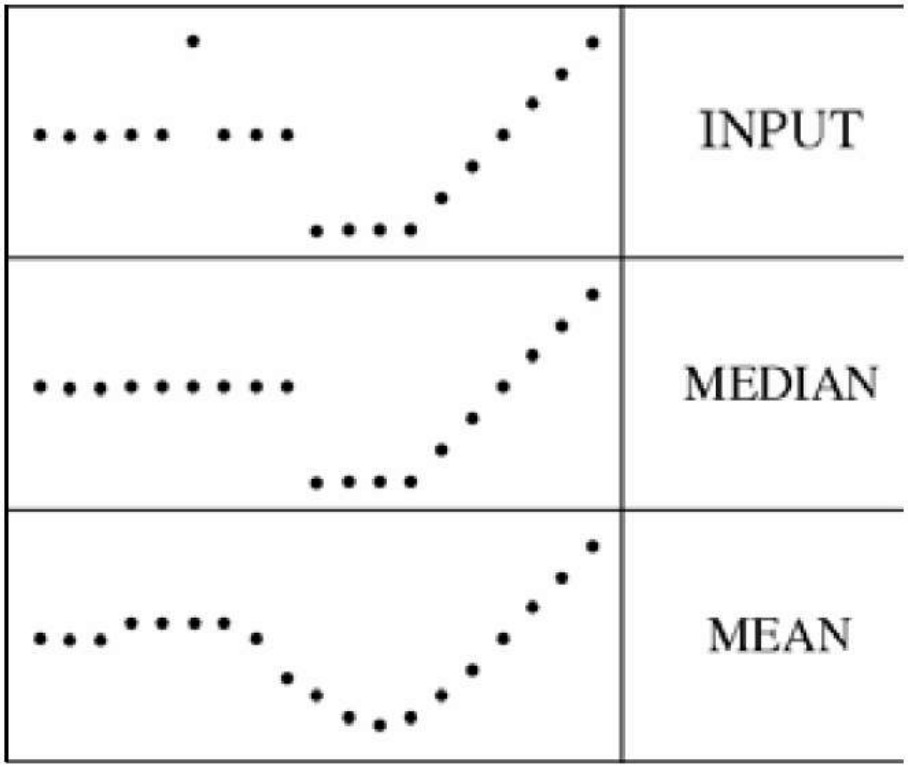


- Является ли фильтр линейным?

Source: K. Grauman

Медианный фильтр

- В чем главное отличие медианного фильтра перед фильтром гаусса?
  - Устойчивость к выбросам (outliers)

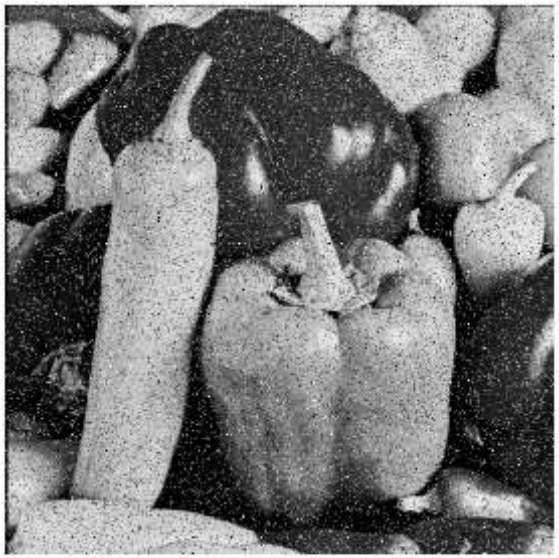


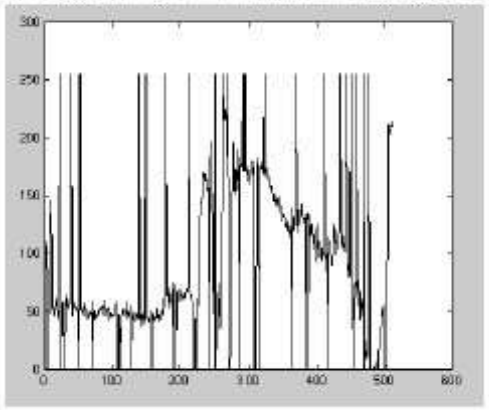
Фильтры размером в 5 пикселей

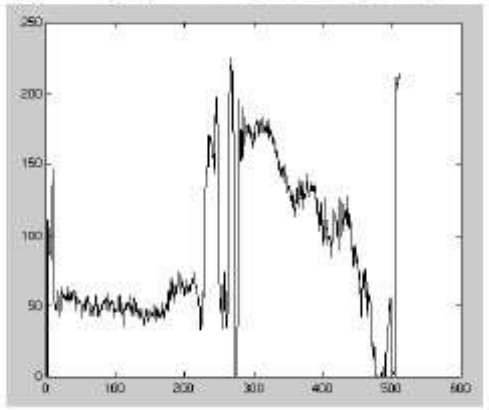
Source: K. Grauman

Медианный фильтр

Шум «соль и перец» Медианная фильтрация

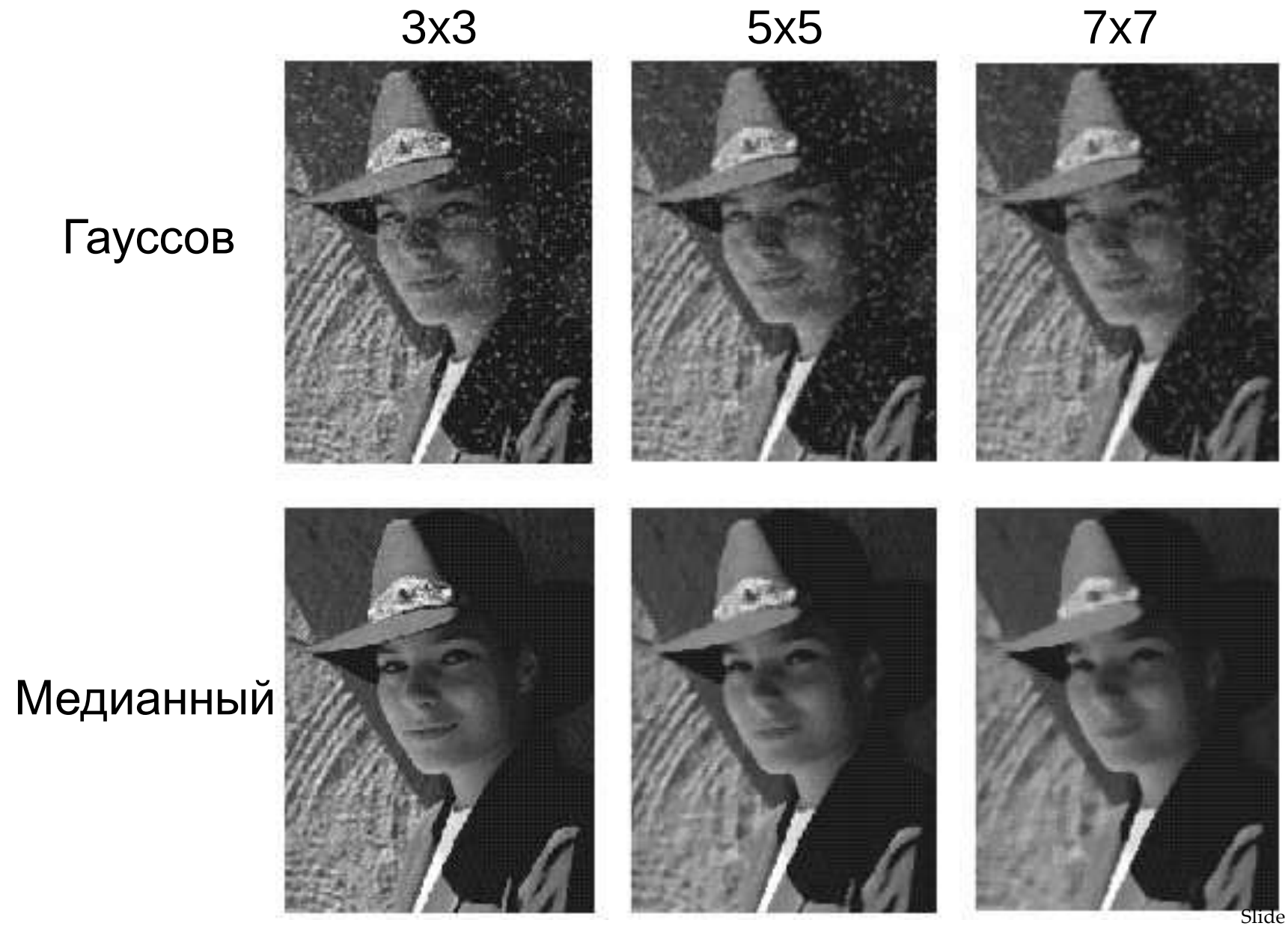






Source: M. Hebert

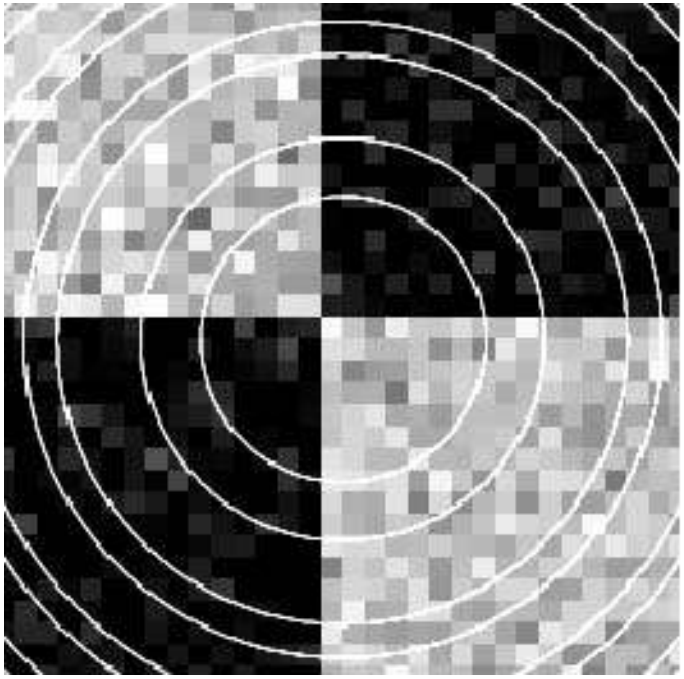
Сравнение фильтров

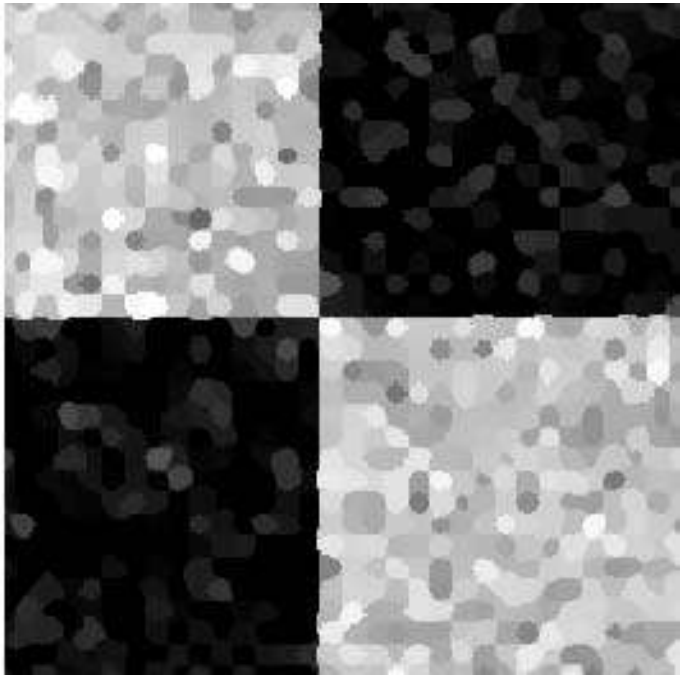


Slide by S. Lazebnik

Медианный фильтр

Результат применения медианного фильтра с радиусом в 7 пикселей к изображению с шумом и артефактами в виде тонких светлых окружностей.





Как оценить качество шумоподавления?

- Оценить результат можно только сравнив исходное изображение без шума с результатом шумоподавления по зашумлённому изображению

исходное изображение без шума

Результат шумоподавления

- Т.е. нужна тестовая выборка пар изображений и метрики для сравнения изображений

Метрика PSNR

- Пусть X – изображение без шума, Y – изображение с шумом после шумоподавления
- Среднеквадратичная ошибка (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

N – число пикселей

- Пиковое отношение сигнал/шум (PSNR)

$$PSNR_{dB} = 10\lg\frac{M^2}{MSE}$$

M – максимальное значение пикселя

Ограничения метрики на примере Lena

PSNR и MSE не учитывают особенности человеческого восприятия, пусть и часто используются на практике

Оригинальное изображение

Lena (Lenna)

- Обрезанная (512x512) часть изображения с разворота Playboy, Nov 1972
- Самый популярный, но не первый случай использования Playboy в обработке изображений (первый в 1961)
- Пригласили на 50ую конференцию Society for Imaging Science and Technology (IS&T) in 1997

Далее будут использованы рисунки из статьи
Wang, Bovik, Lu “WHY IS IMAGE QUALITY ASSESMENT SO DIFFICULT?”

Метрики качества

У этих изображений одинаковые PSNR с оригиналом (примерно 25 dB)

Повышена контрастность

Добавлен белый гауссов шум

Метрики качества

И у этих – тоже примерно 25 dB!

Добавлен импульсный шум

Размытие

Метрики качества

- И у этого – тоже!

Артефакт блочности после JPEG

Метрики качества

- Вывод: PSNR не всегда отражает реальный видимый уровень искажений.
- Как улучшить?

HVS models
(human visual system)

- Использовать функцию чувствительности глаза к различным частотам (CSF)
- Использовать свойство маскировки
- Использовать равномерные к восприятию цветовые пространства (CIE Lab, CIEDE2000)

Извлечение информации из изображения

Пример

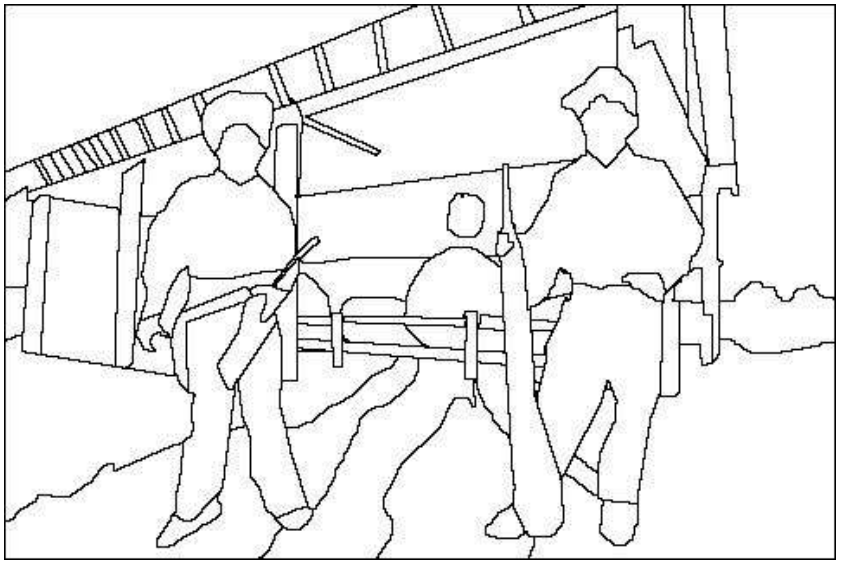
Что можно сказать про изображение?

Что такое край (edge)?

Разметка человеком

изображение

разметка вручную



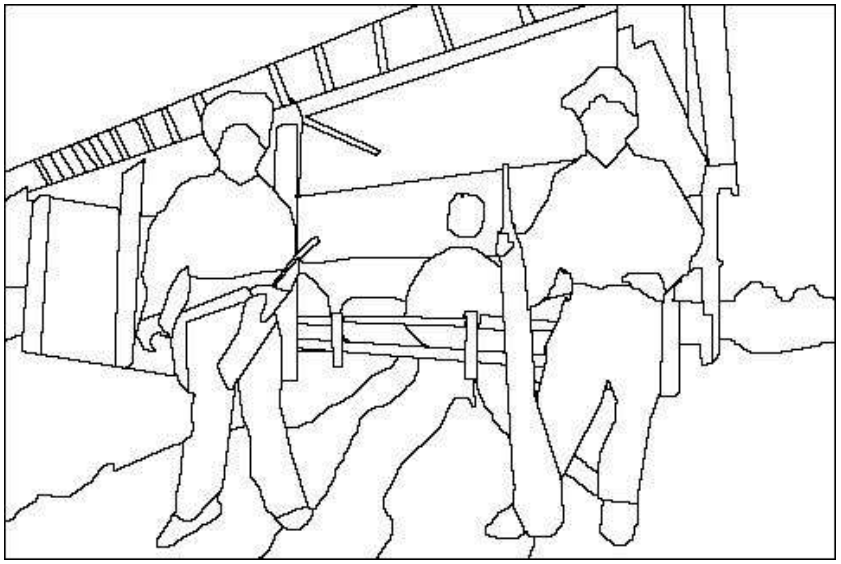
Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Края обычно соответствует границам визуально отличающихся друг от друга областей изображения

Края и области

изображение

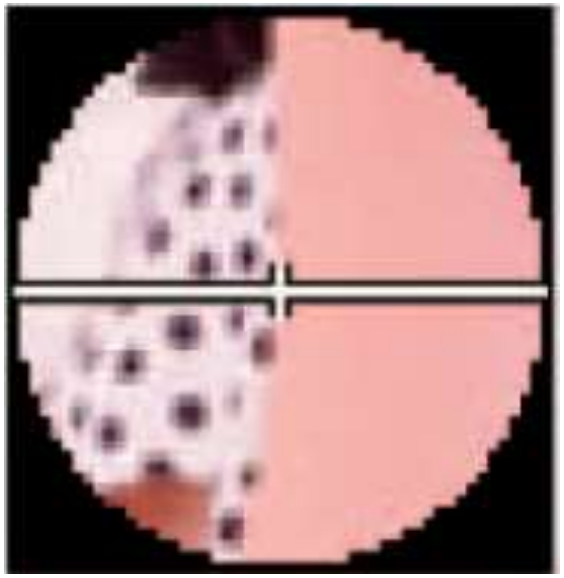
разметка вручную

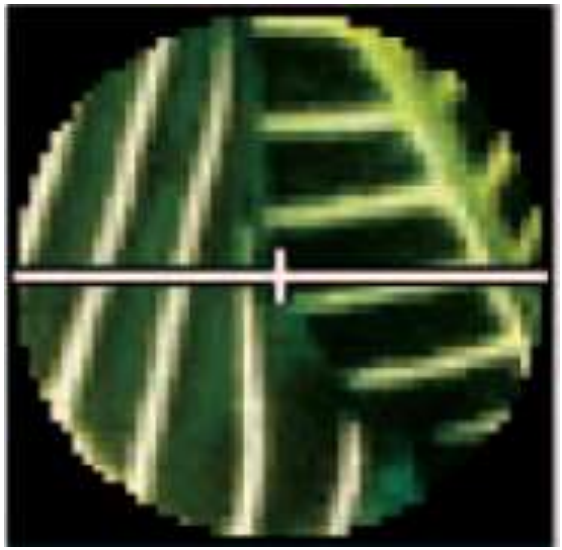


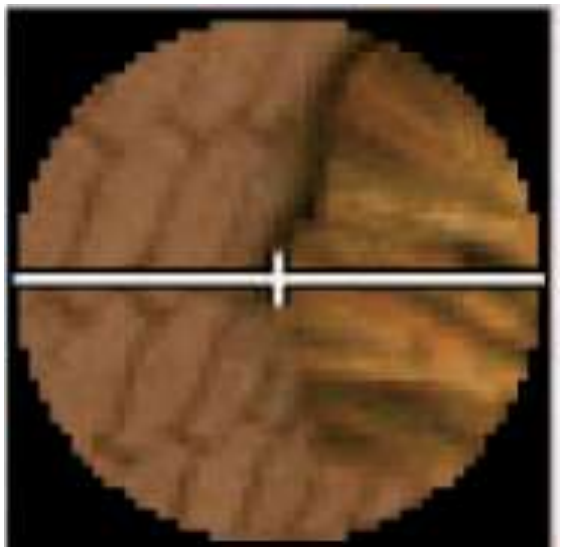
Смежные задачи:

- Нахождение краёв в изображении
- Разделение изображения на визуально однородные области (визуальная сегментация)

Примеры краёв



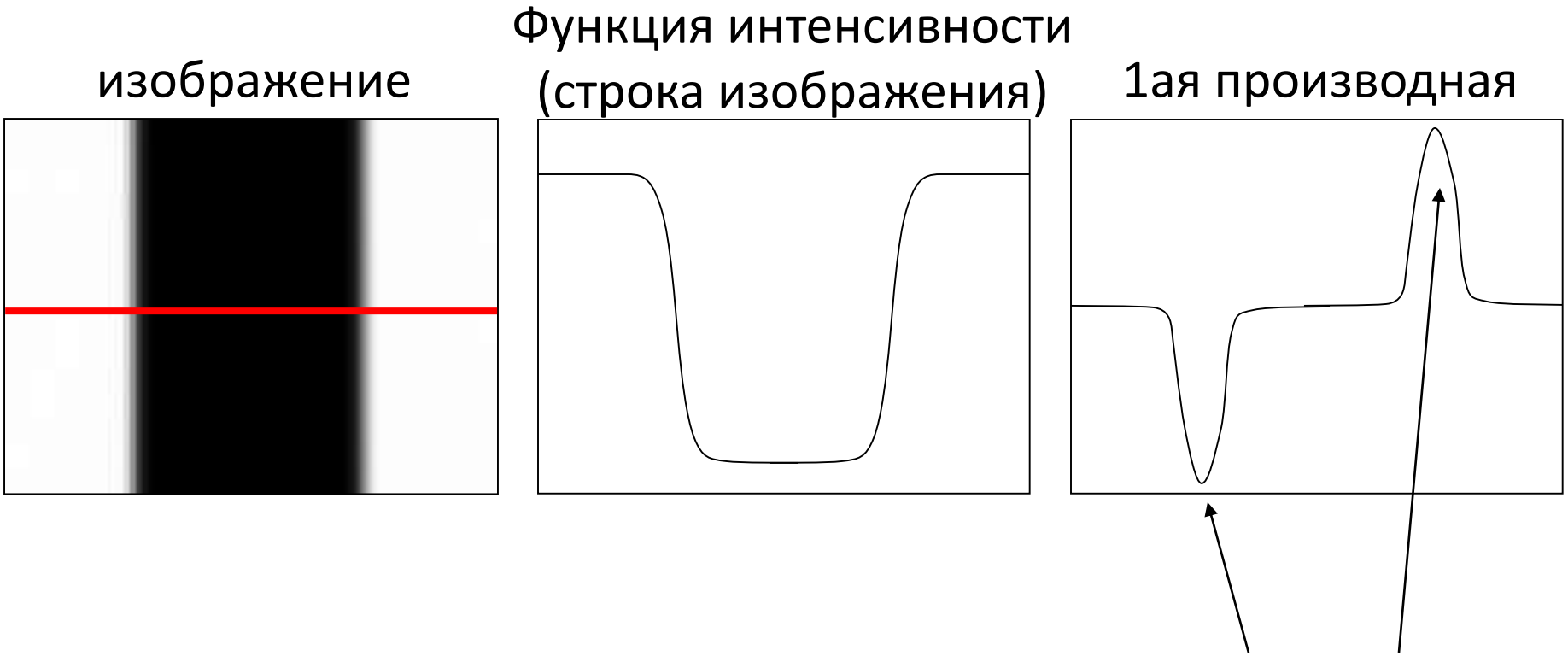




Какие признаки используются?

Базовое определение края

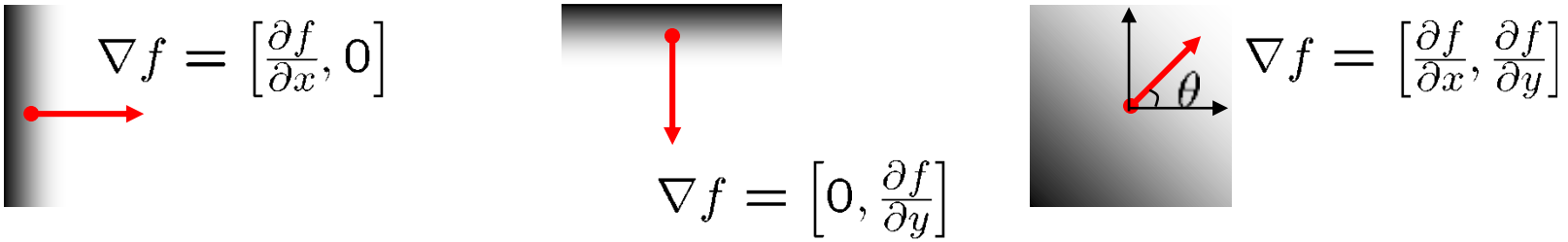
- Край – это точка резкого изменения значений функции интенсивности изображения



Slide by S. Lazebnik

Градиент изображения

- 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: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

- Как направление градиента соответствует направлению края?
- *Сила края* задается величиной (нормой) градиента:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Дифференцирование и свёртка

• Для функции 2х переменных, f(x,y):

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \left(\frac{f(x+\varepsilon, y)}{\varepsilon} - \frac{f(x,y)}{\varepsilon} \right)$$

• Разностная производная:

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

- Разностная производная - линейная и инвариантная к переносу
- Можно записать как свёртку

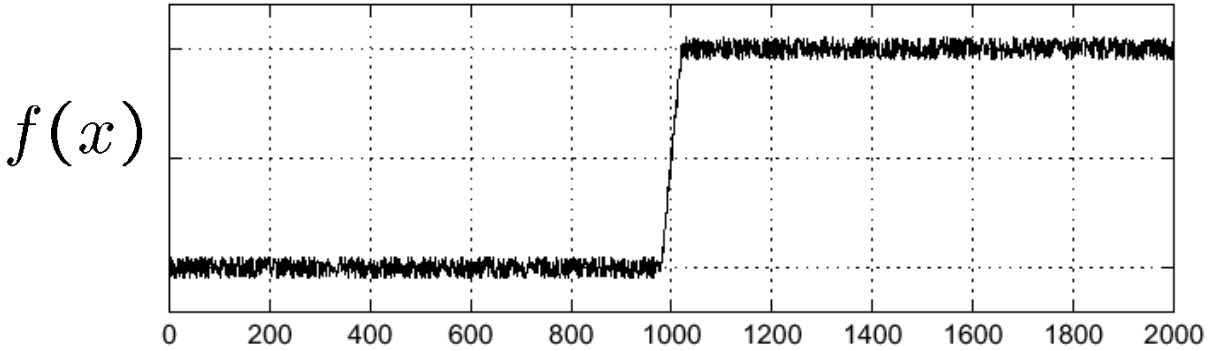
-1	1

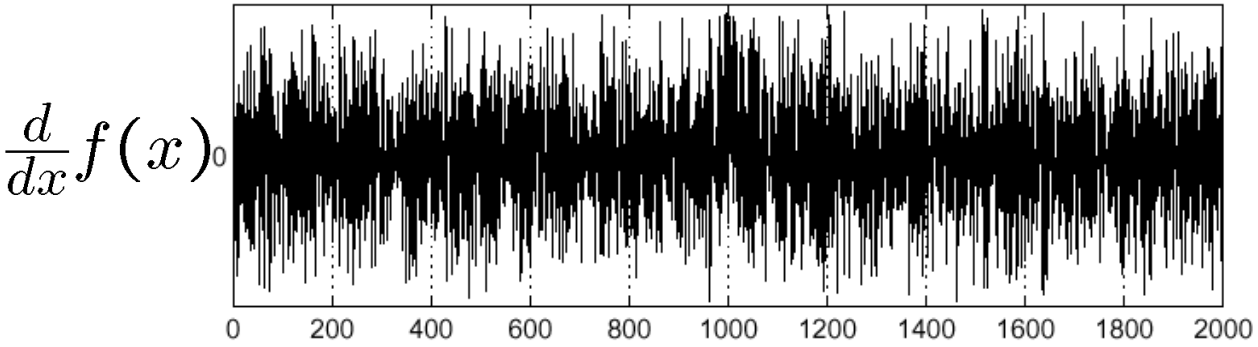
Простейший фильтр

Source: D. Forsyth, D. Lowe

Влияние шума

- Рассмотрим строку или столбец изображения
  - Интенсивность от положения можно рассматривать как сигнал





Край исчез

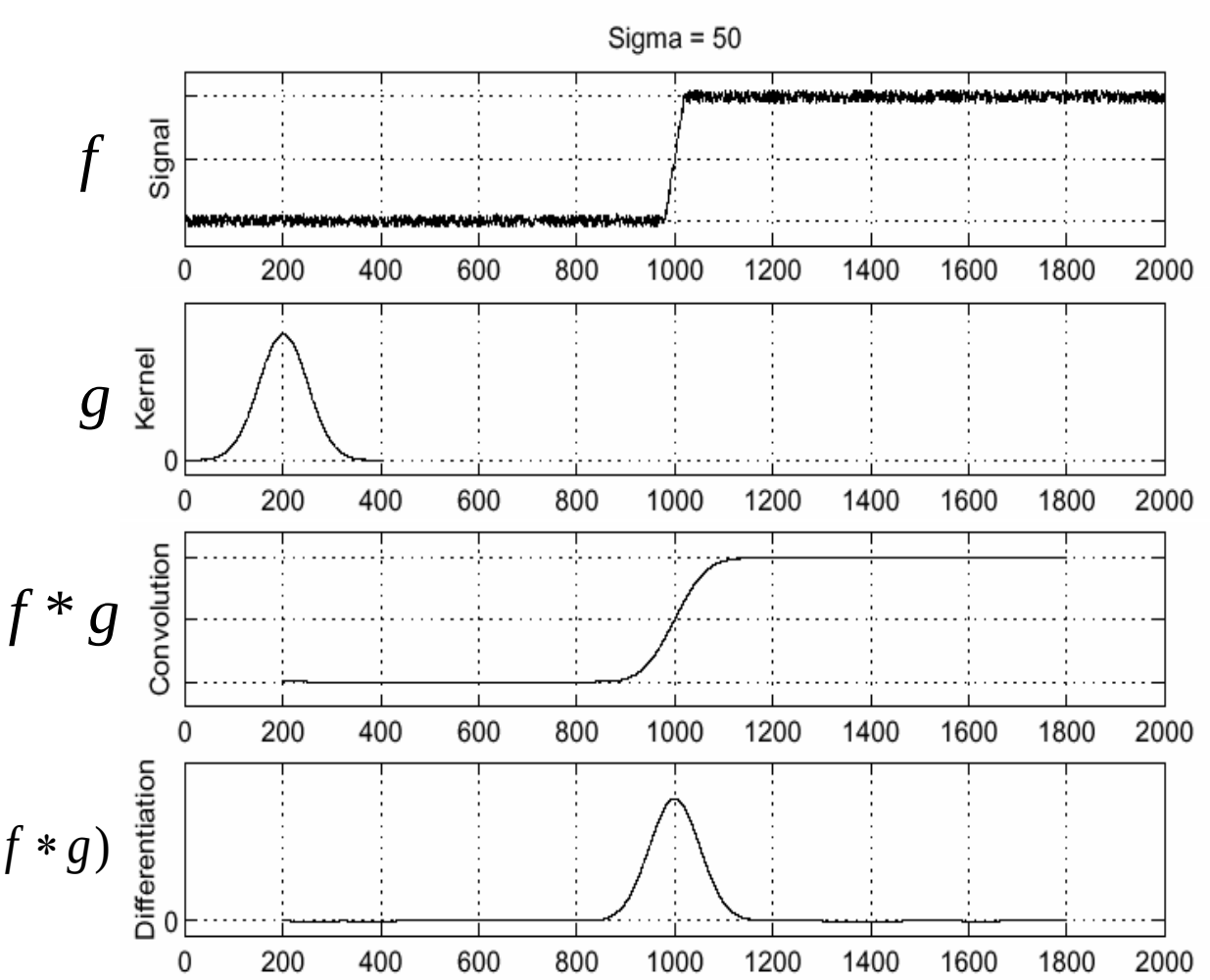
Source: S. Seitz

Влияние шума

- Разностные производные очень чувствительны к шуму
  - Зашумленные пиксели отличаются от соседей
  - Чем сильнее шум, тем выше отклик
- Сглаживание
  - Сглаживание делает все пиксели (зашумленные?) чуть более похожими на соседей

Source: D. Forsyth

Предобработка (сглаживание)



- Для поиска краев ищем пики в: $\frac{d}{dx}(f * g)$

Source: S. Seitz

Свойства свертки

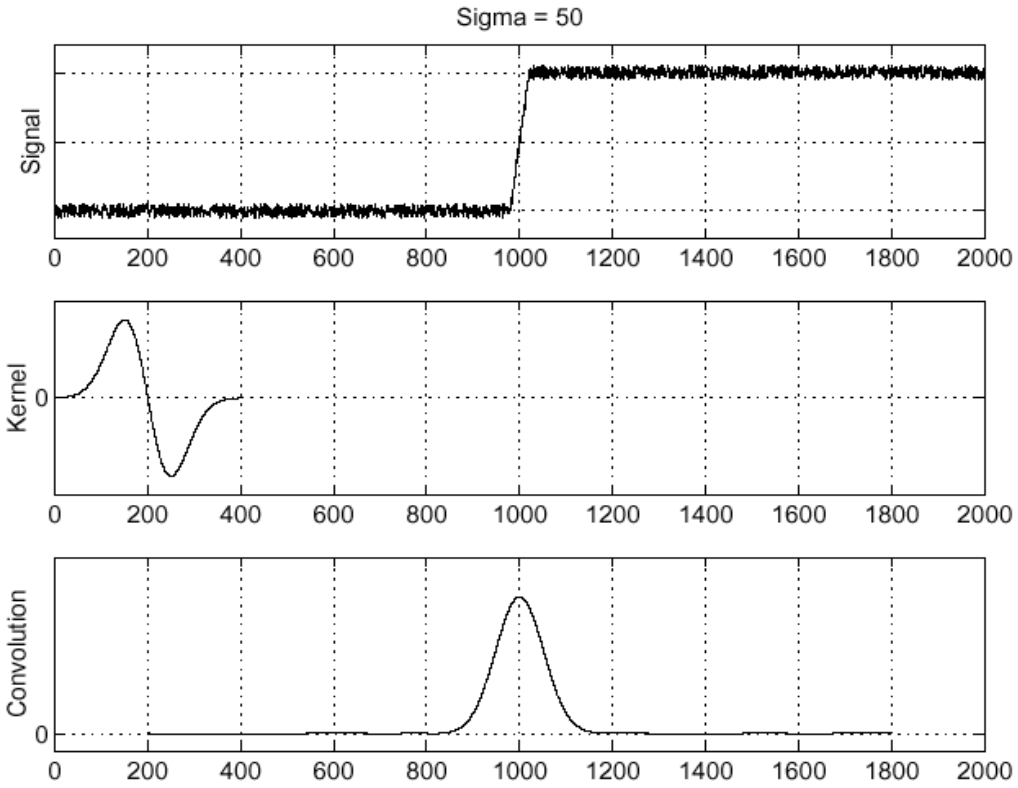
- Операции свертки и дифференцирования ассоциативны:
- Это экономит 1 операцию:

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

f

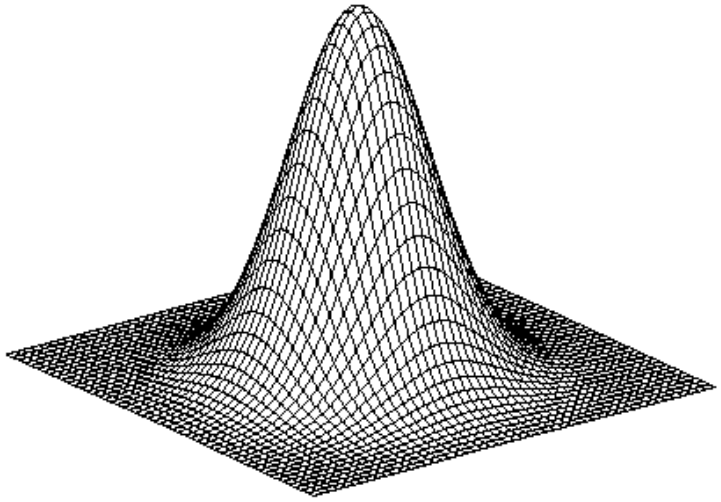
$\frac{d}{dx} g$

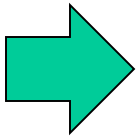
$f * \frac{d}{dx} g$

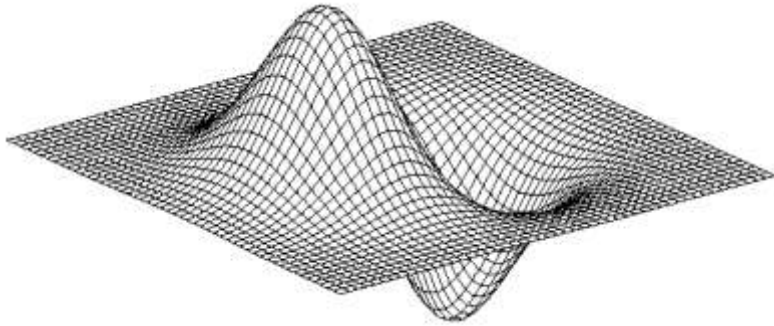


Source: S. Seitz

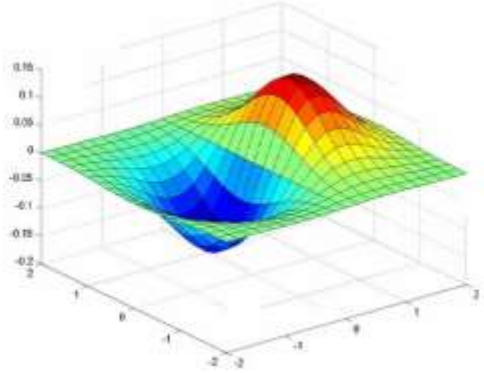
Производная фильтра Гаусса

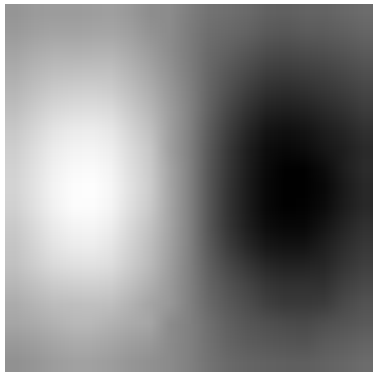




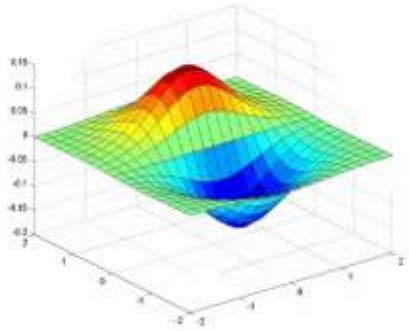


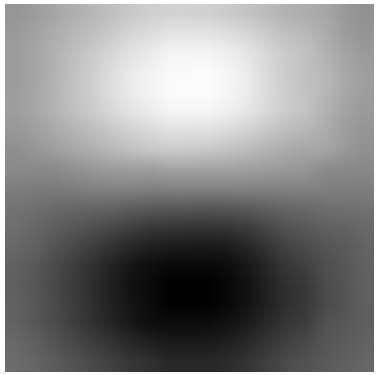
По y





По y:





Slide by S. Lazebnik

Известные фильтры

Несколько фильтров, по разному оценивающие производные по направлению и интегрирующие шумоподавление:

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

Робертса

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

Превитт

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

Собель 3x3

$$\begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}$$

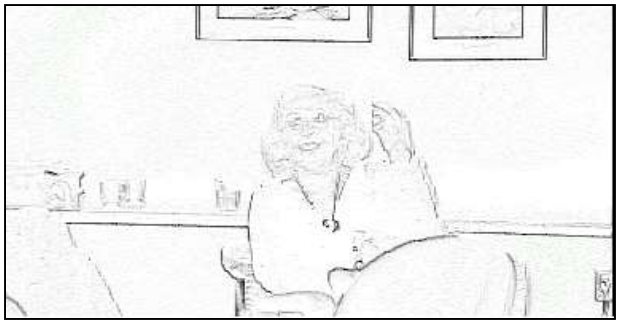
Scharr фильтр

$$S_x = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} S_y = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Собель 5x5

Карта силы краев

Примеры:



Робертса

Превитт

Собеля

Выделение краев

- Вычисление градиента – не идеальный метод для поиска краёв.

Исходное изображение

Карта силы краев

- Чего не хватает?
  - Точности – края «толстые» и размытые
  - Информации о связности

Детектор Canny

1. Свертка изображения с ядром – производной от фильтра гаусса
2. Поиск силы и направления градиента
3. Выделение локальных максимумов (Non-maximum suppression)
   - Утоньшение полос в несколько пикселей до одного пикселя
4. Связывание краев с использованием гистерезиса (двойного порога)
   - Выше верхнего порога – «сильные края»
   - Ниже нижнего порога – шум (отсекаем)
   - Посередине – потенциальные края

J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Source: D. Lowe, L. Fei-Fei

Пример

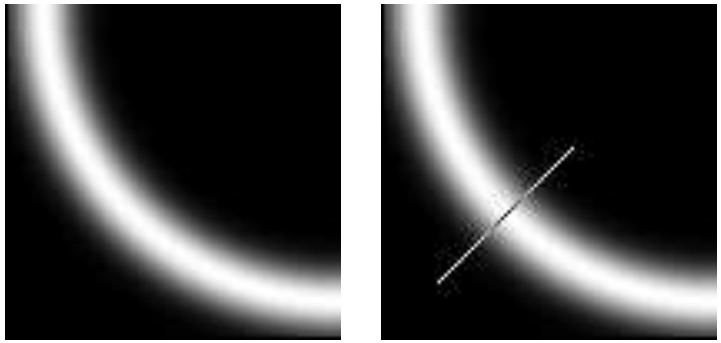
- Исходное изображение (Lena)

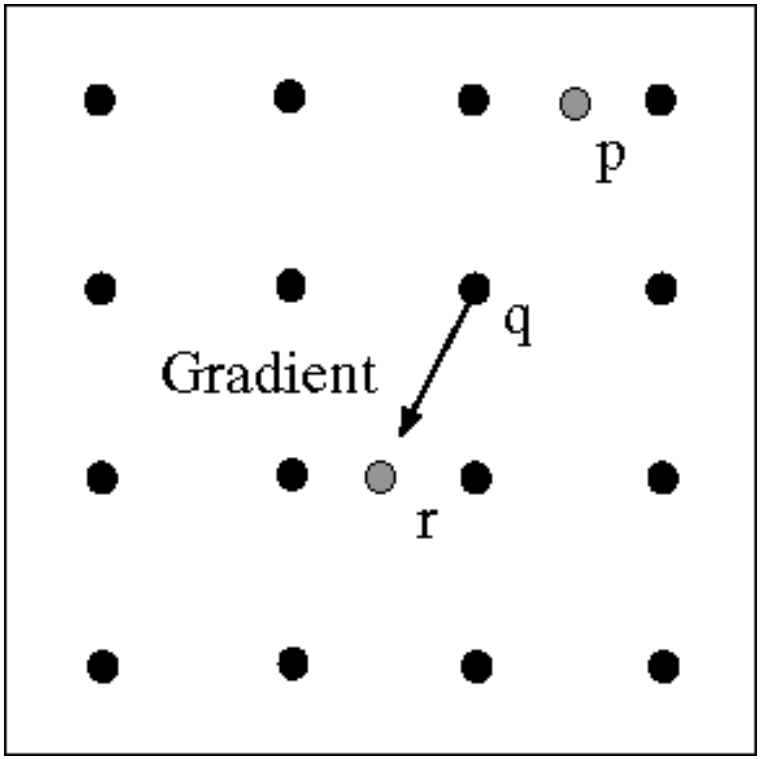
Пример

Норма градиента

Slide by S. Lazebnik

Поиск локальных максимумов





Несколько вариантов фильтров:

- Фильтр 3x3. Пиксель оставляем, если градиент в нём больше всех остальных в окрестности.
- Фильтр по направлению градиента. Оставляем градиент q, если значение больше p и r. Значения в p и r интерполируем.

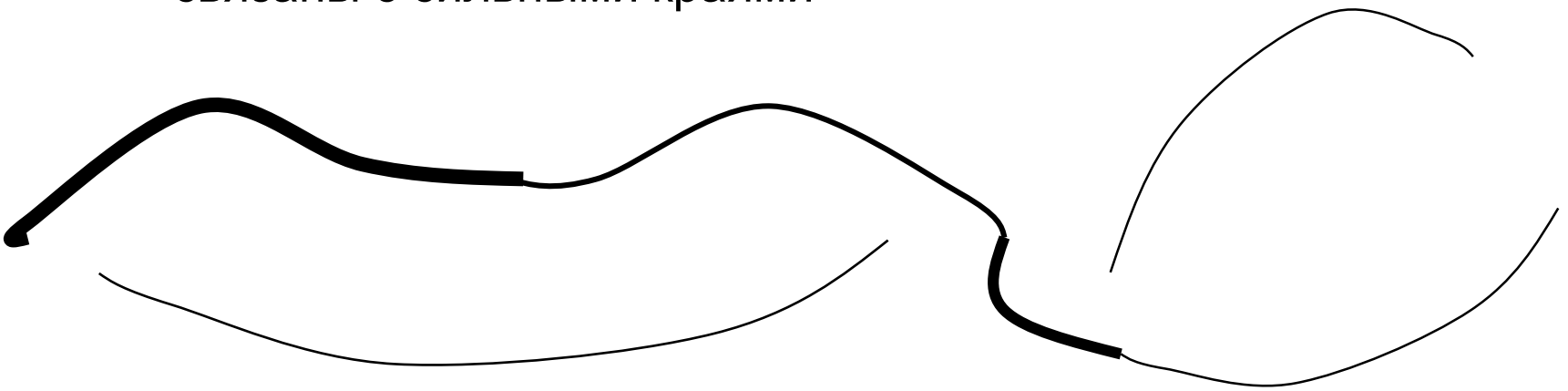
Source: D. Forsyth

Результат работы

Slide by S. Lazebnik

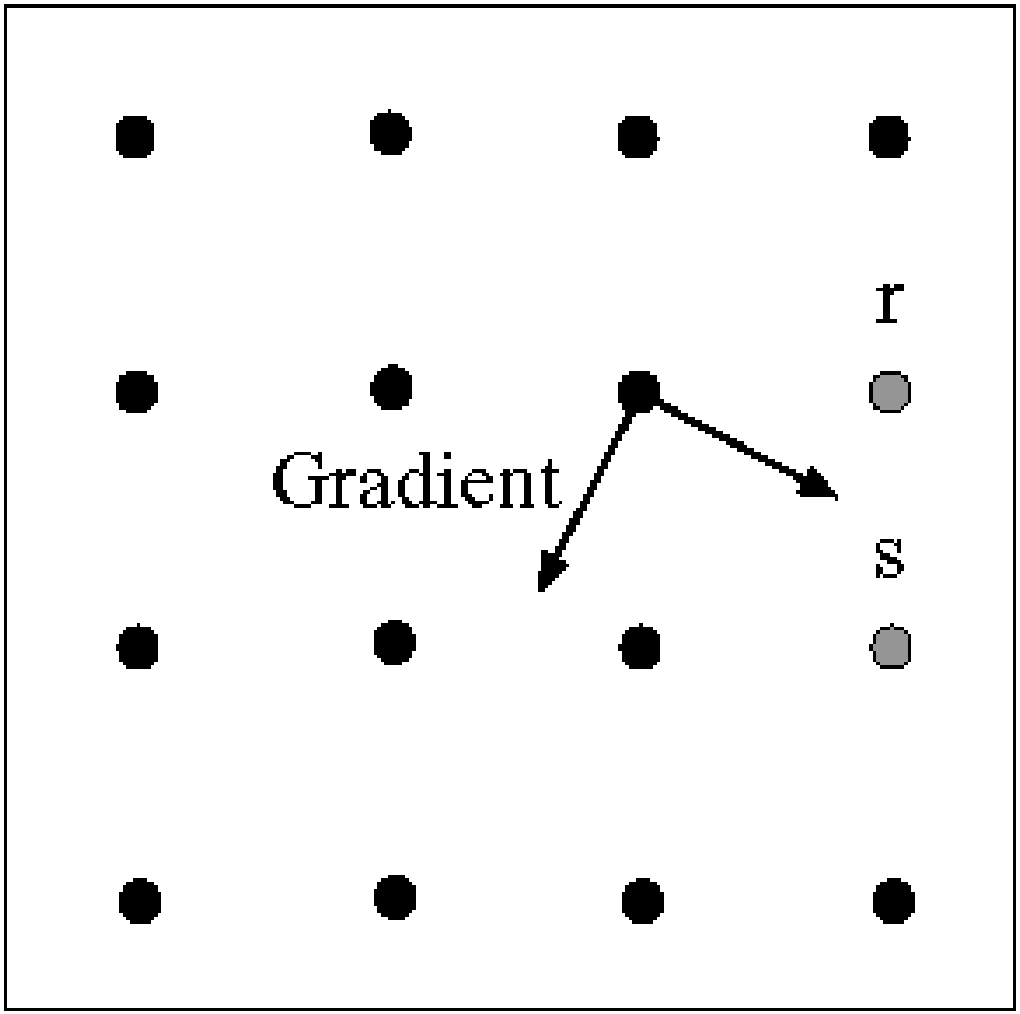
Прослеживание с гистерезисом

- Выбираем два порога «верхний» и «нижний»
- Размечаем все пиксели изображений (оставшиеся)
  - Выше верхнего – «сильные края»
  - Выше нижнего – «слабые края»
  - Ниже нижнего - шум
- Идея прослеживания:
  - Оставляем «слабые» края только в том случае, если они связаны с сильными краями

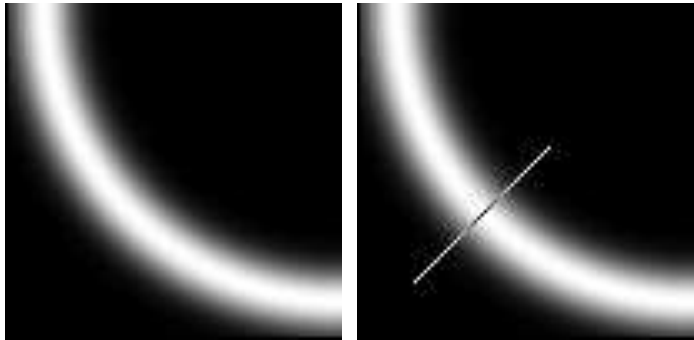


Source: S. Seitz

Прослеживание границ



- Пусть отмеченная точка – «слабый край», который ещё не проверен.
- Будем искать «сильного» соседа, переходя по соседним неподавленным краям вдоль границы



Source: D. Forsyth

Эффект гистерезиса

Исходное изображение

Высокий порог
(сильные края)

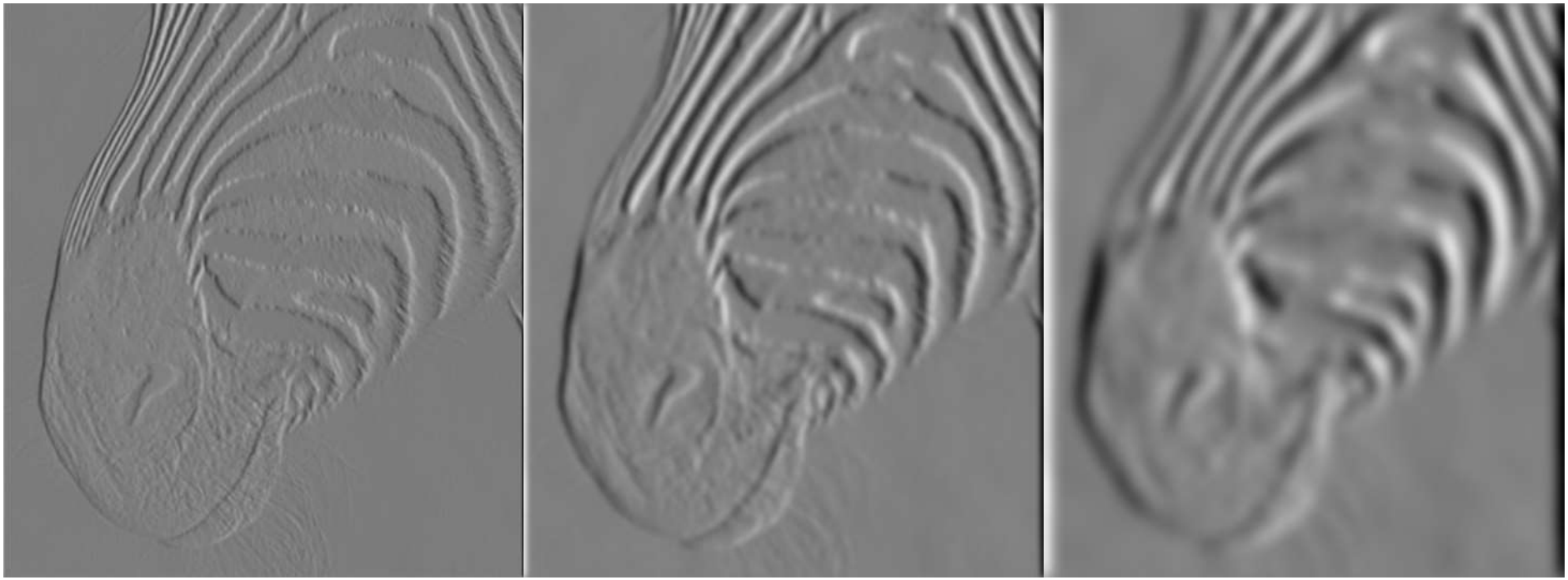
Низкий порог
(слабые + сильные края)

Порог по гистерезису

Source: L. Fei-Fei

Сглаживание и локализация

Применим сглаженные производные разного размера:



1 pixel 3 pixels 7 pixels

Сглаженные производные подавляют шум, но размывают края. Плюс края находится на разных «масштабах»

Source: D. Forsyth

Влияние σ в Cany (Размер ядра размытия)

original

Canny with $\sigma = 1$

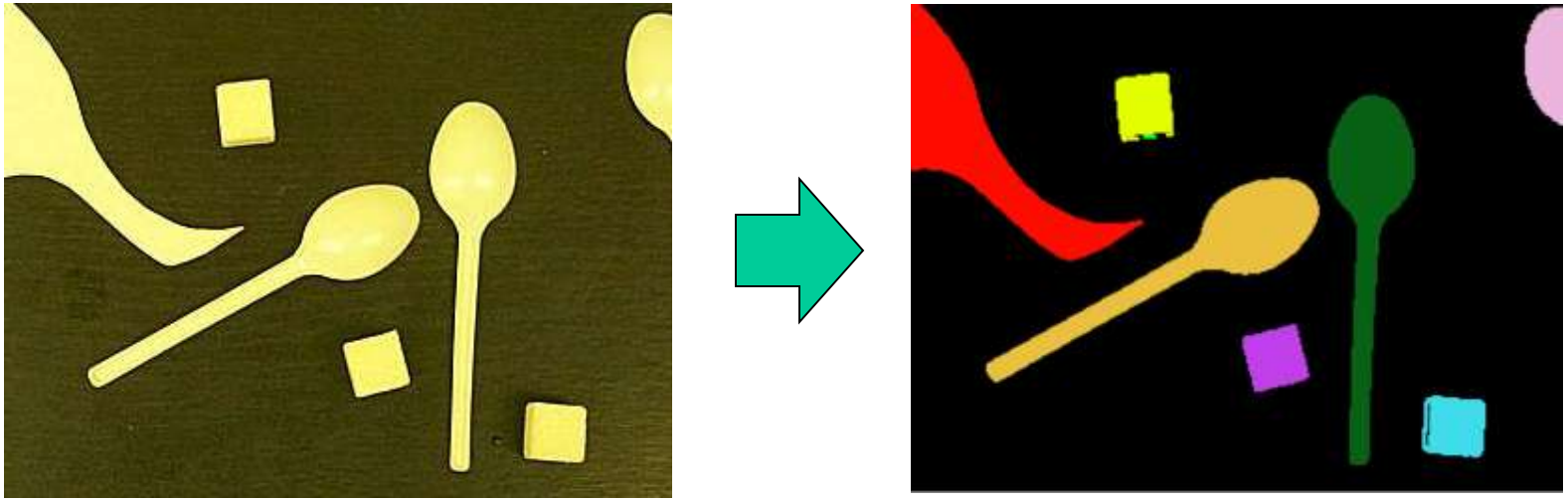
Canny with $\sigma = 2$

Выбор σ зависит от задачи

- большое σ - поиск крупных границ
- маленькое σ - выделение мелких деталей

Source: S. Seitz

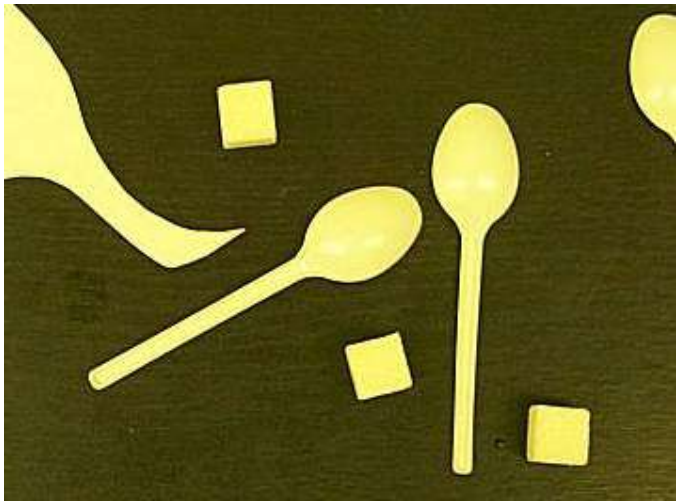
Вернёмся к сегментации изображений



- Рассмотрим основы смежной задачи – сегментации изображений
- Как мы будем записывать результат сегментации?
- Сделаем карту разметки – изображение, в каждом пикселе которого номер сегмента, которому принадлежит этот пиксель
- Визуализировать удобно каждый сегмент своим цветом

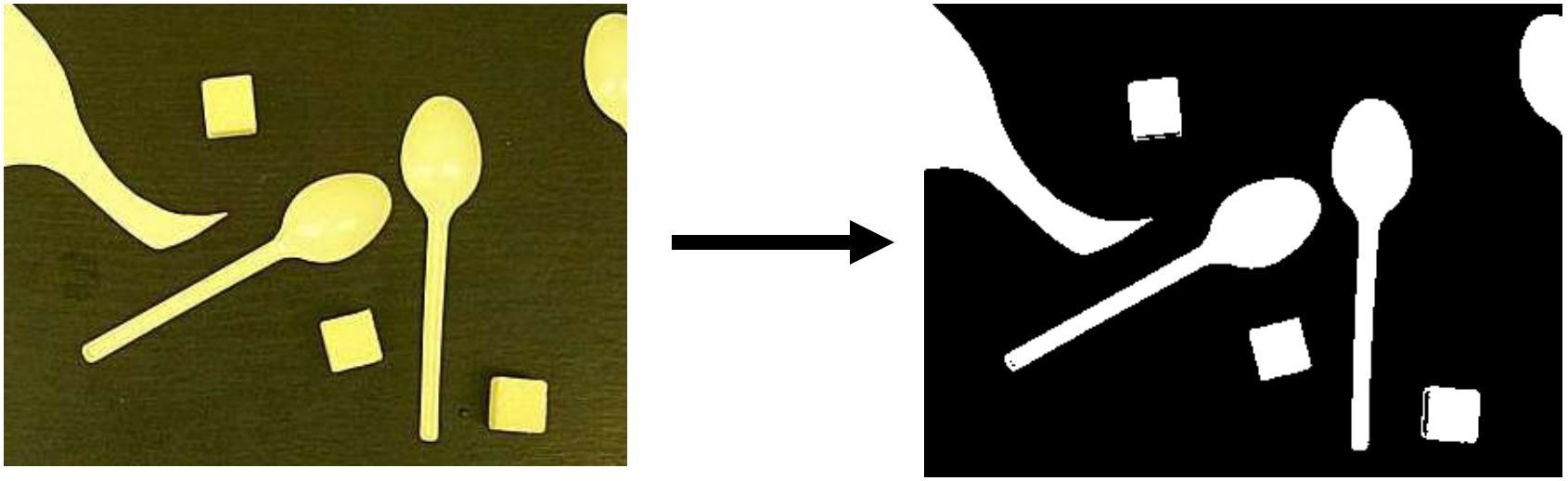
Простейшая сегментация

Чем отличаются объекты на этом изображении?



- Все объекты яркие, фон тёмный
- Для сегментации такого изображения нам достаточно:
  - пороговая бинаризация
  - обработки шума
  - выделения связанных компонент

Пороговая бинаризация

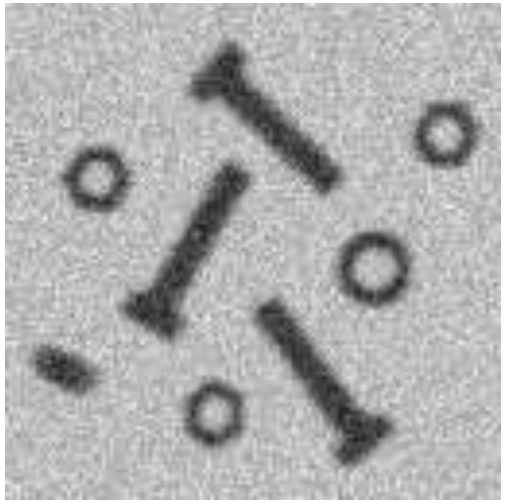


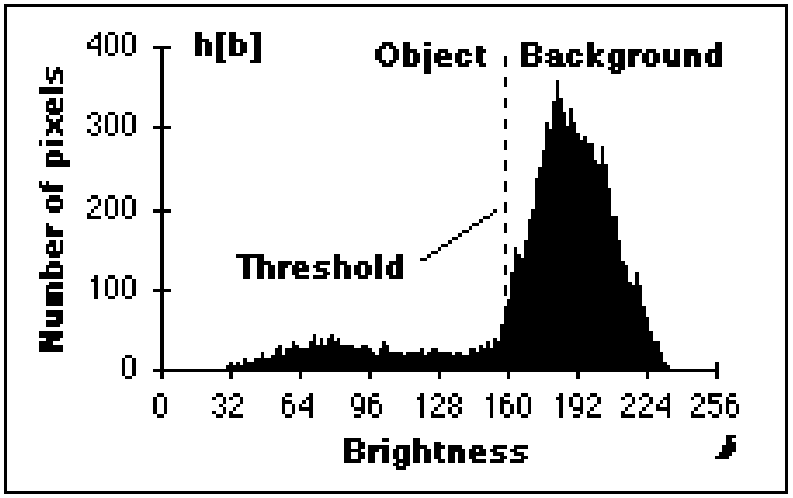
- Пороговая фильтрация (thresholding)
  - Пиксели, которых выше/ниже некоторого порога, заданного «извне», помечаются 1
  - Ниже порога помечаются 0
- Бинарное изображение – пиксели которого могут принимать только значения 0 и 1
- Бинаризация - построение бинарного изображения по полутоновому / цветному

Пороговая фильтрация

Более интересный способ – определение порога автоматически, по характеристикам изображения

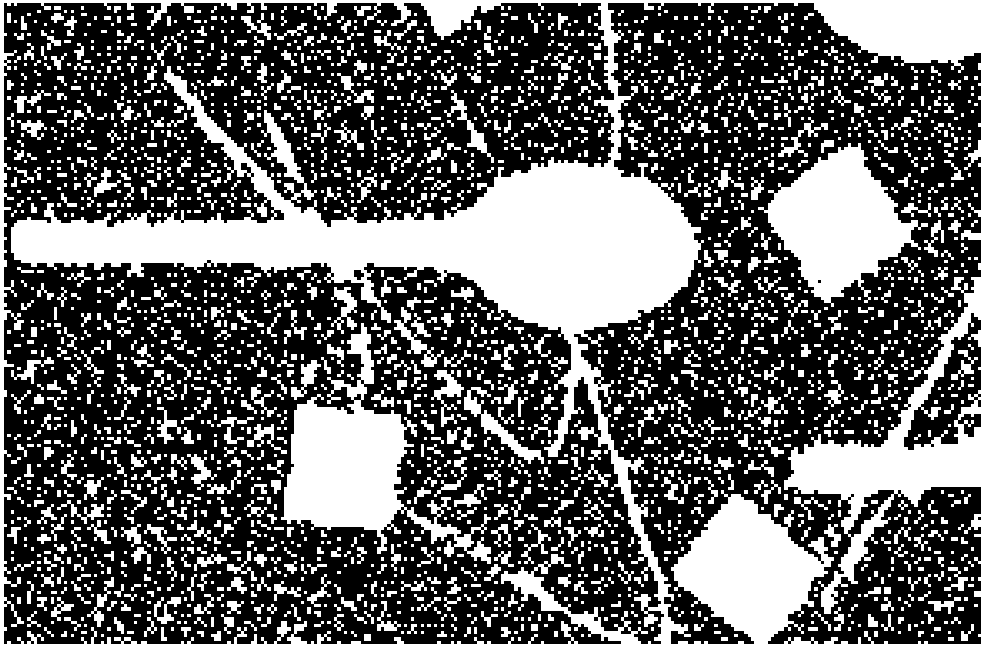
- Анализ гистограммы





Шум в бинарных изображениях

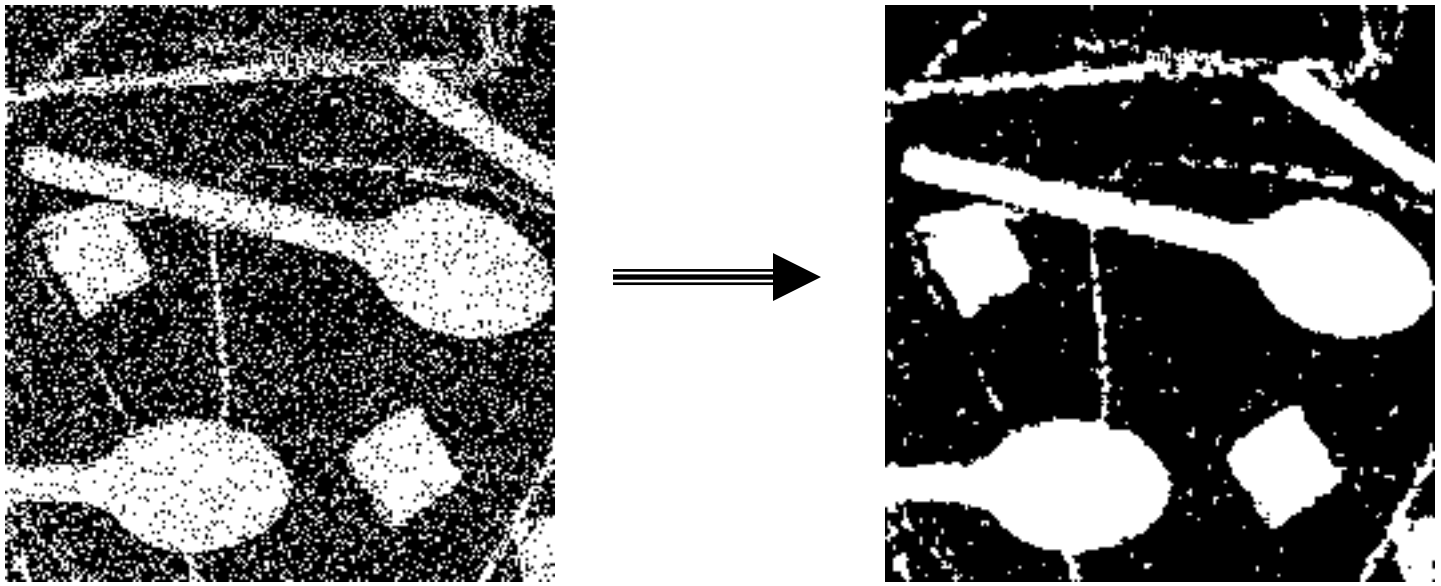
Пример бинарного изображению с сильным шумом



Часто возникает из-за невозможности полностью подавить шум в изображениях, недостаточной контрастности объектов и т.д.

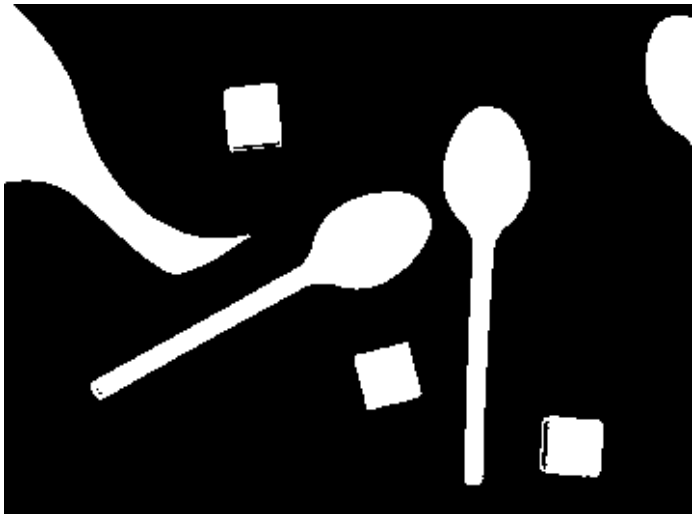
Подавление шума

Применим медианный фильтр с окрестностью 3x3

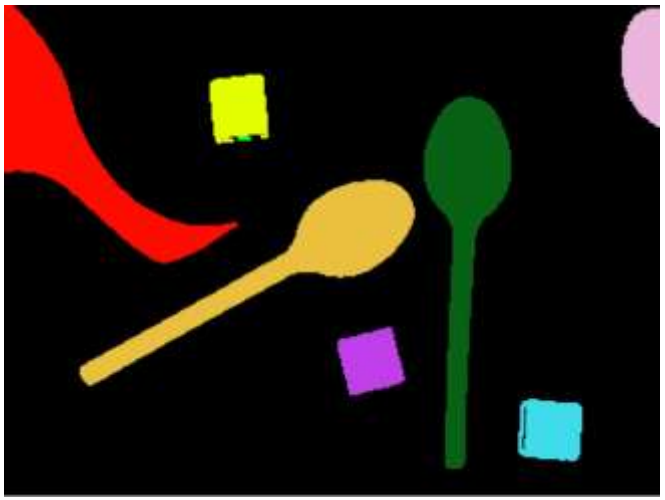


Математическая морфология - теория и техника анализа и обработки геометрических структур, основанная на теории множеств, топологии и случайных функциях. В основном применяется в обработке цифровых изображений, но также может быть применима на графах, полигональной сетке, стереометрии и многих других пространственных структурах.

Что дальше?



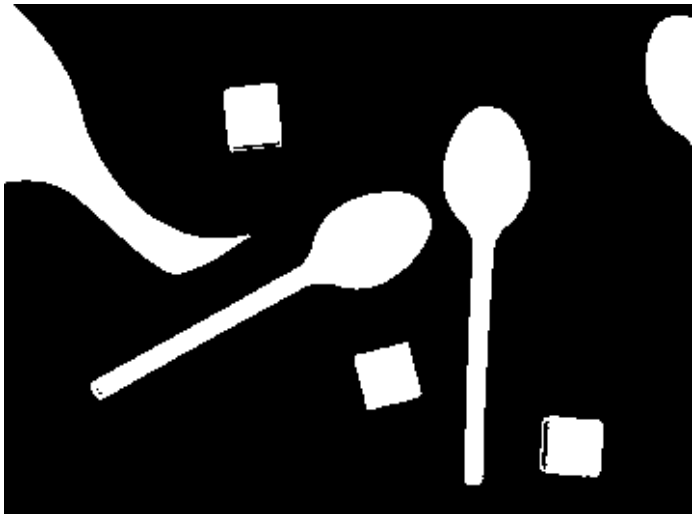
Получили бинарное изображение



Нужна карта разметки

Выделение связных областей

- Определение связной области:
  - Множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству.



Соседи пикселей:

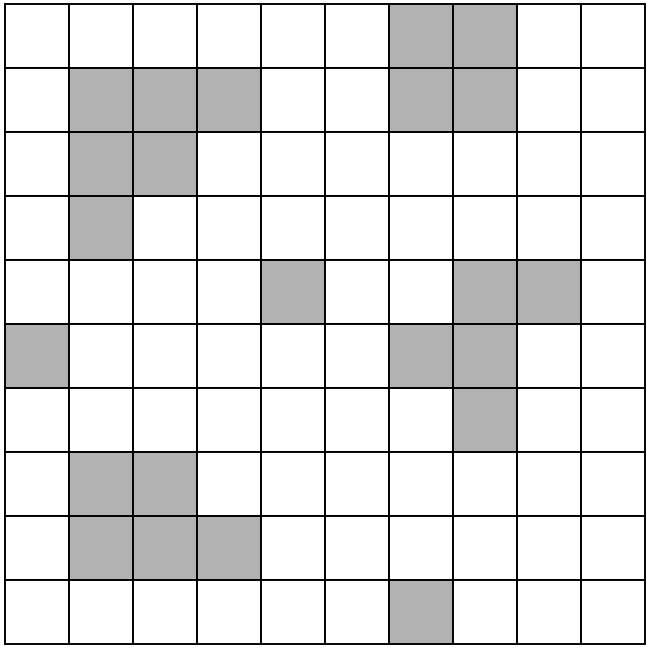
	1	
2	*	3
	4	

4-связность

1	2	3
4	*	5
6	7	8

8-связность

Разметка связных областей



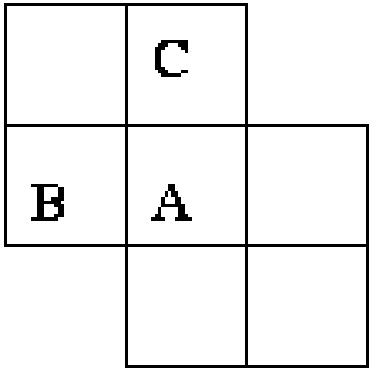
Бинарное изображение

						1	1		
	2	2	2			1	1		
	2	2							
	2								
				3			4	4	
5						4	4		
							4		
	6	6							
	6	6	6						
						7			

Размеченное изображение

Метод последовательного сканирования

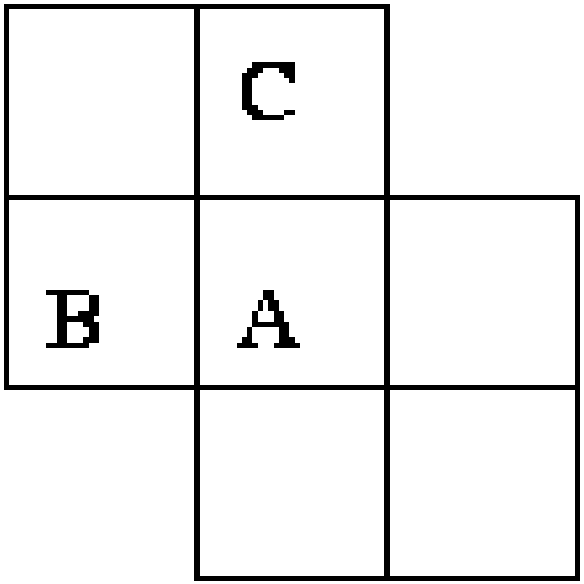
Последовательно, сканируем бинарное изображение сверху вниз, слева направо:




```
if A = O
     do nothing

else if (not B labeled) and (not C labeled)
     increment label numbering and label A

else if B xor C labeled
     copy label to A

else if B and C labeled
     if B label = C label
          copy label to A
     else
          copy either B label or C label to A
          record equivalence of labels
```

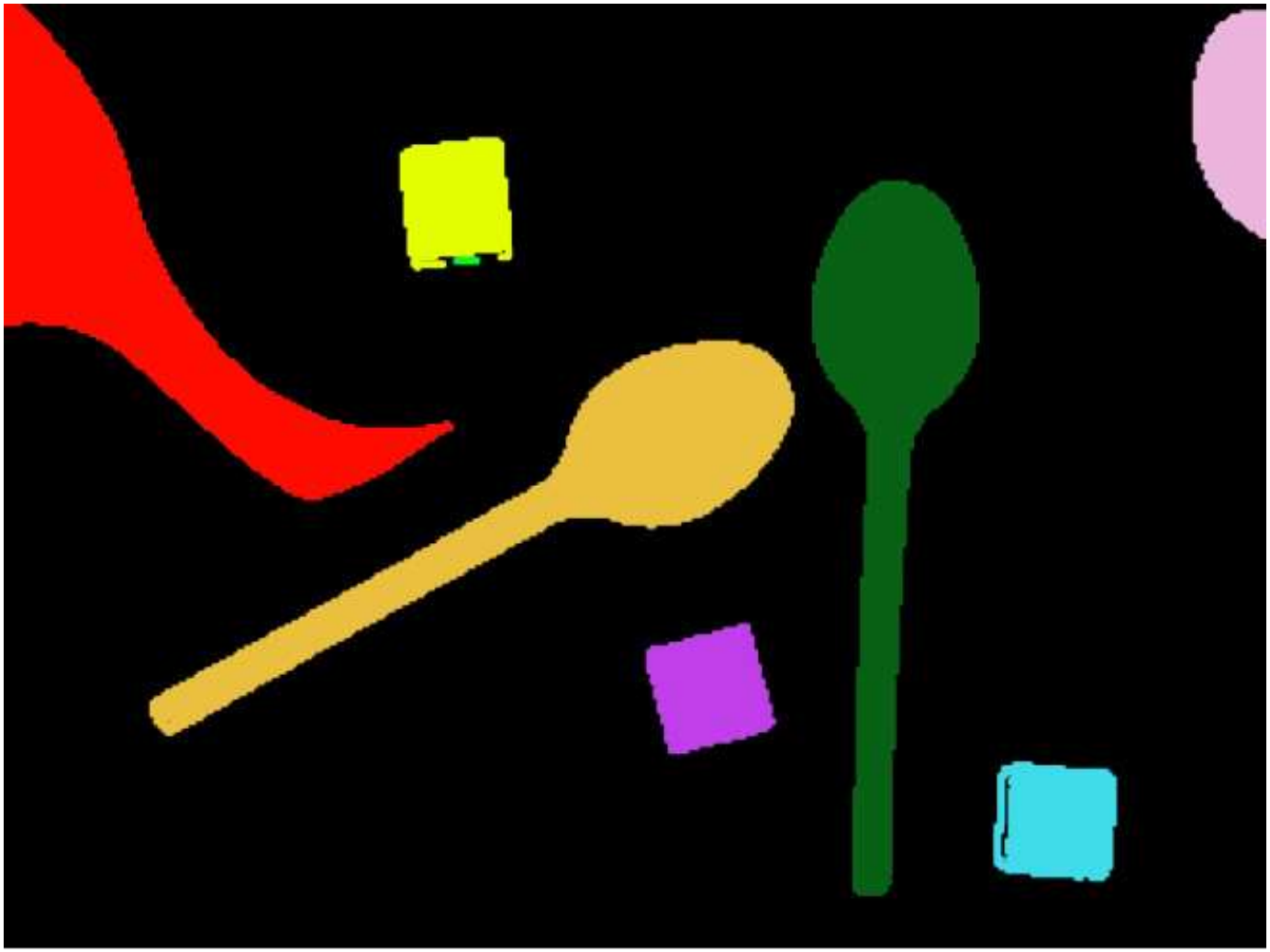
Последовательное сканирование

Случай конфликта:

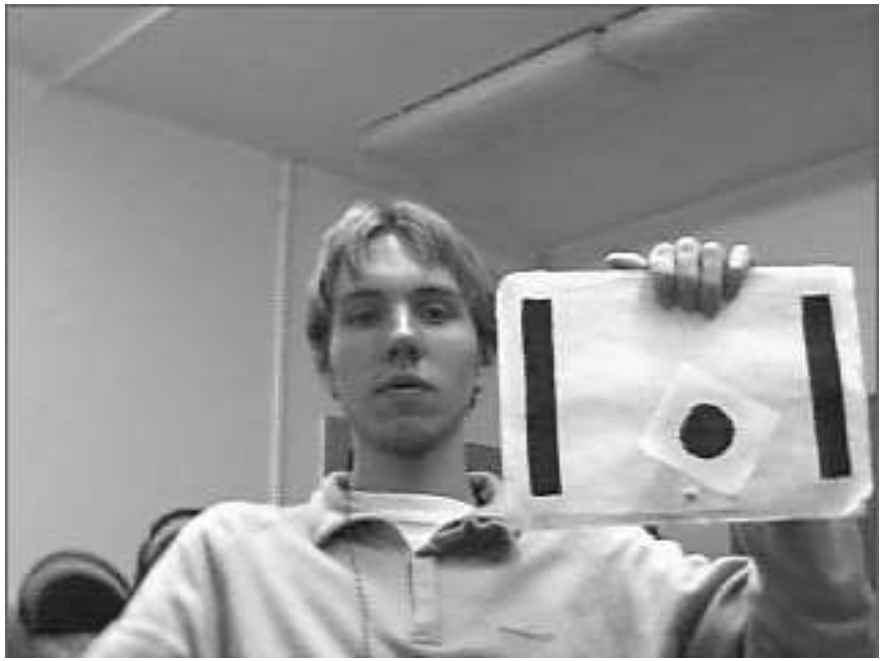


Постобработка - переразметка с учетом эквивалентностей областей (второй проход в алгоритме)

Выделенные связанные компоненты



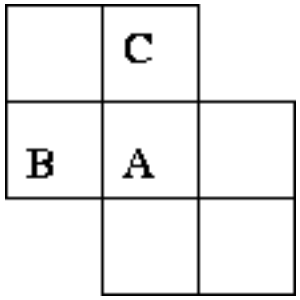
Сегментация



Тот же метод можно использовать для сегментации и обычных изображений на однородные области

Выделение однородных областей методом последовательного сканирования

Сканируем изображение сверху вниз, слева направо:



1. if $I(A) - I_{avg}(Cl(B)) > \delta$ and $I(A) - I_{avg}(Cl(C)) > \delta$ -
   создаем новую область, присоединяем к ней пиксел A

2. if $I(A) - I_{avg}(Cl(B)) < \delta$ xor $I(A) - I_{avg}(Cl(C)) < \delta$ –
   добавить A к одной из областей

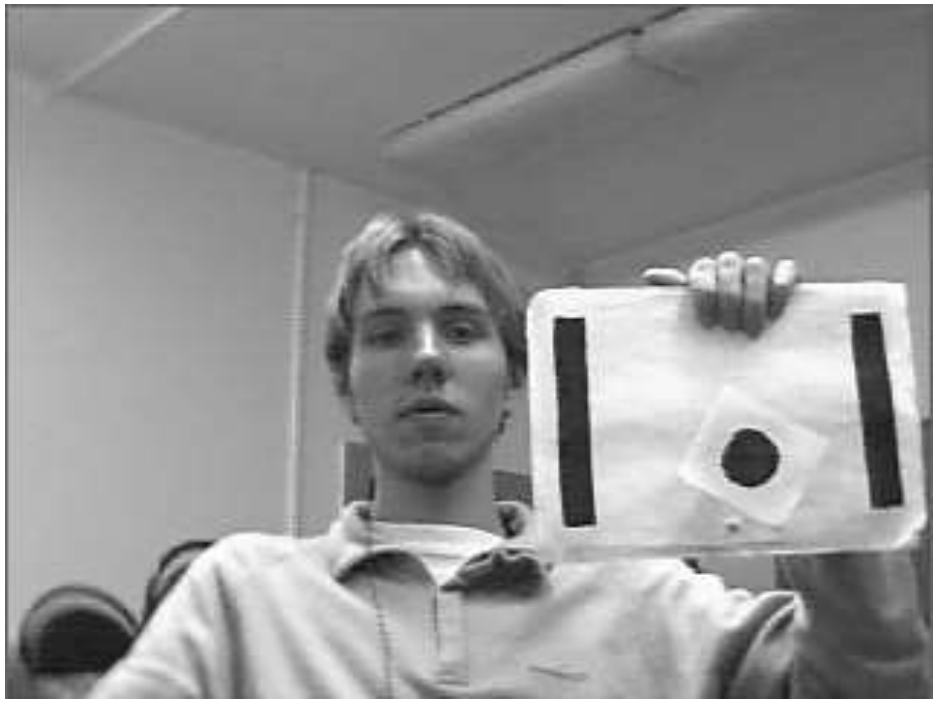
3. if $I(A) - I_{avg}(Cl(B)) < \delta$ and $I(A) - I_{avg}(Cl(C)) < \delta$:

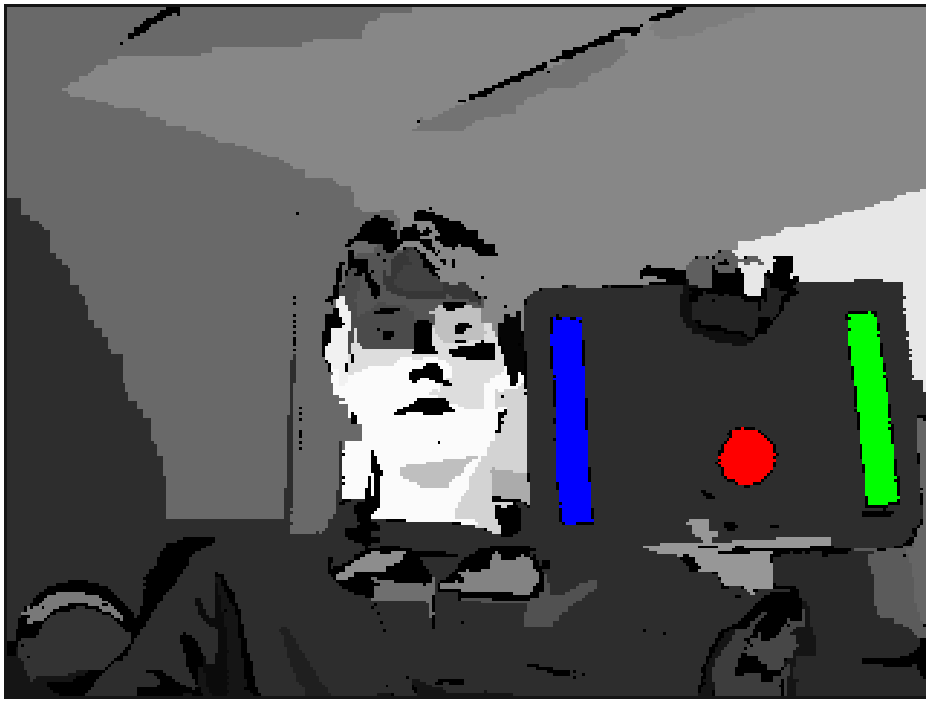
   1. $I_{avg}(Cl(B)) - I_{avg}(Cl(C)) < \delta$ –
      сливаем области B и C.

   2. $I_{avg}(Cl(B)) - I_{avg}(Cl(C)) > \delta$–
      добавляем пиксел A к тому классу, отклонение от которого минимально.

I(A) – яркость пиксела A
Cl(B) – область к которой принадлежит пиксел B
$I_{avg}(Cl(B))$ – средняя яркость области к которой принадлежит B

Пример работы



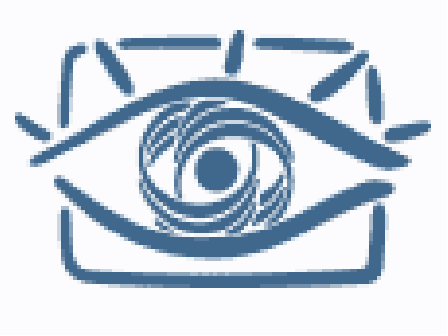


Современные алгоритмы сегментации

- Задача сегментации изображений продолжает активно исследоваться, есть много постановок
- Им посвящена ещё одна лекция

На лекции рассмотрели:

- Линейную фильтрацию (свёртку) изображения, которая позволяет решать целый ряд задач – шумоподавление, оценка градиента, оценки карты освещённости
- Смежные задачи выделения краёв и сегментации изображений
- Выделение краёв изображения в простом случае можно достичь поиском локальных максимумов градиента яркости
- Простейшая сегментация контрастных объектов – пороговая бинаризация и выделение связанных компонент



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

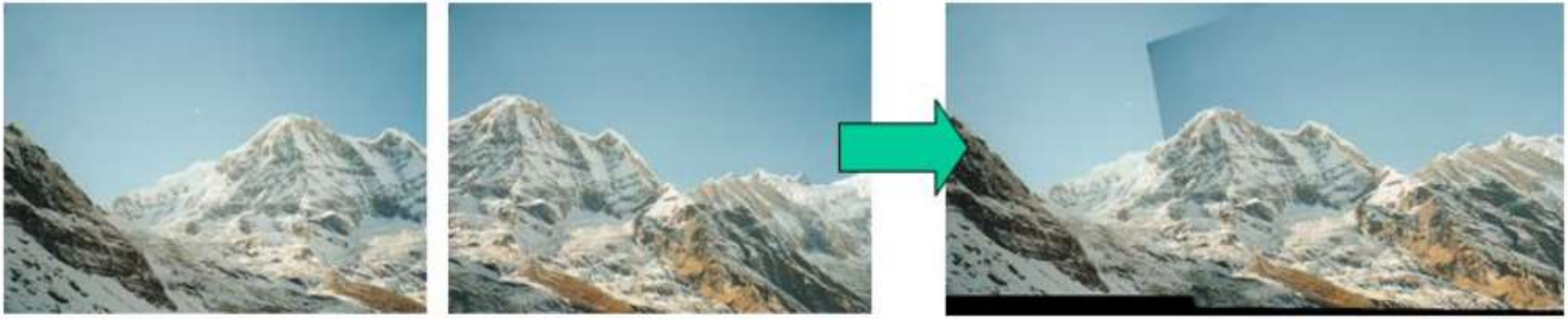
Лекция №3 «Сопоставление изображений»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

23 сентября 2019 года

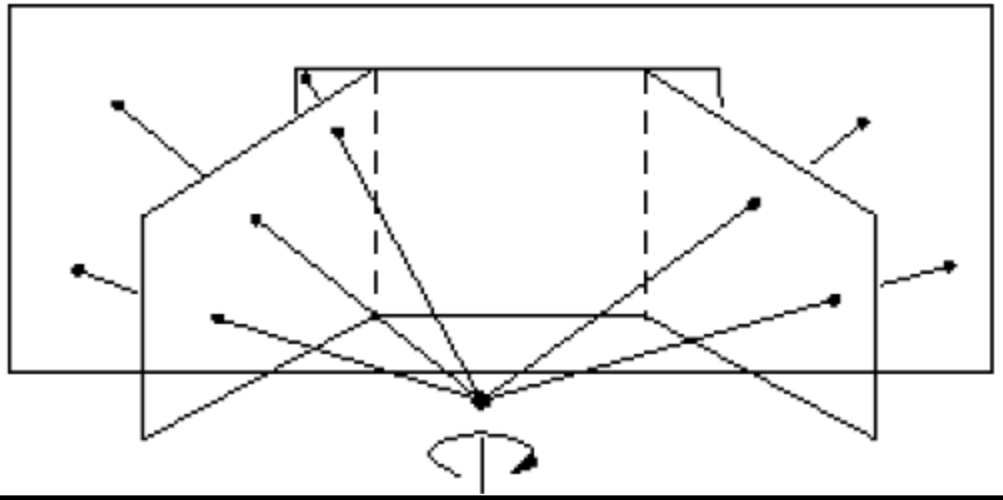
Совмещение изображений

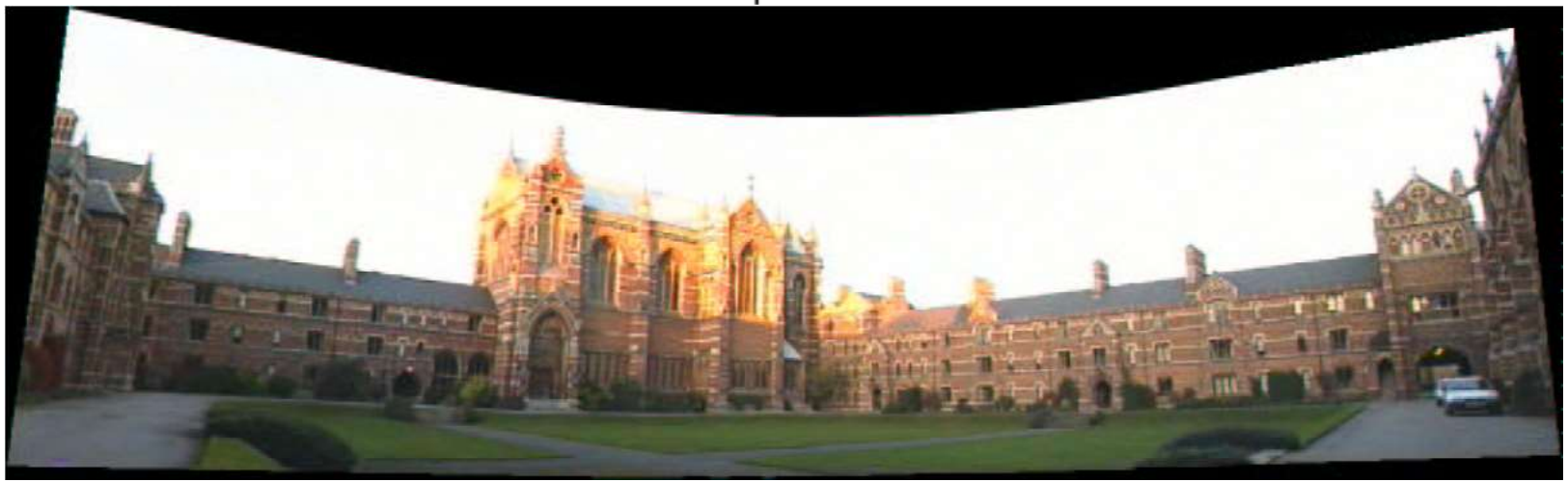


Совмещение изображений (image alignment) – определение совместно наблюдаемой области и совмещение изображений до совпадения общей наблюдаемой области

Сопоставление изображений (image matching) – определение соответствия точек и фрагментов между изображениями (пр. определение пар точек, которые соответствуют одной и той же точке сцены)

Построение панорамы (мозаики)





Геоинформатика

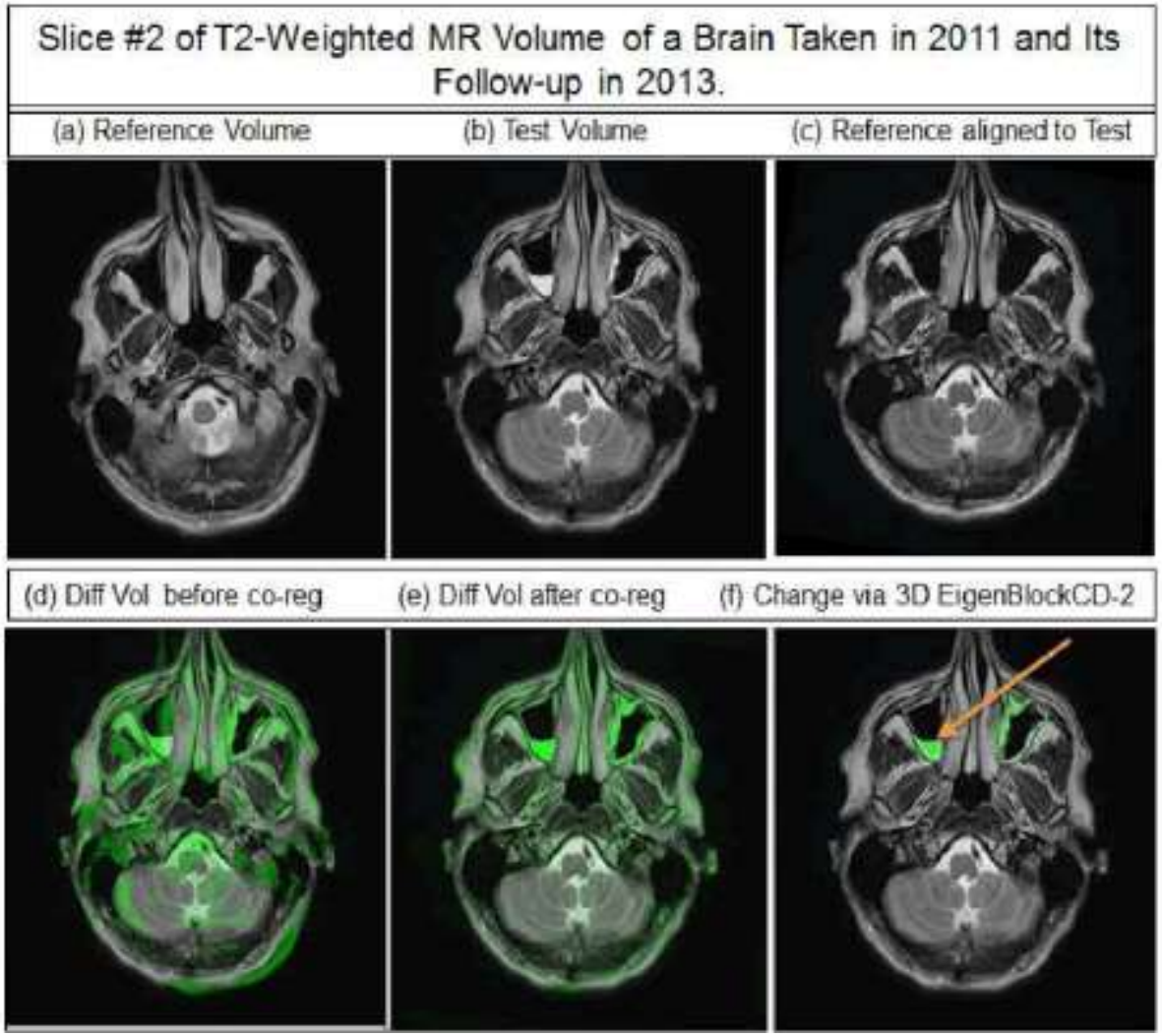
Сопоставление изображений, построение карт, карт высот и т.д.

Трехмерная реконструкция по изображениям

Коллекция изображений
из интернета

Трёхмерная модель

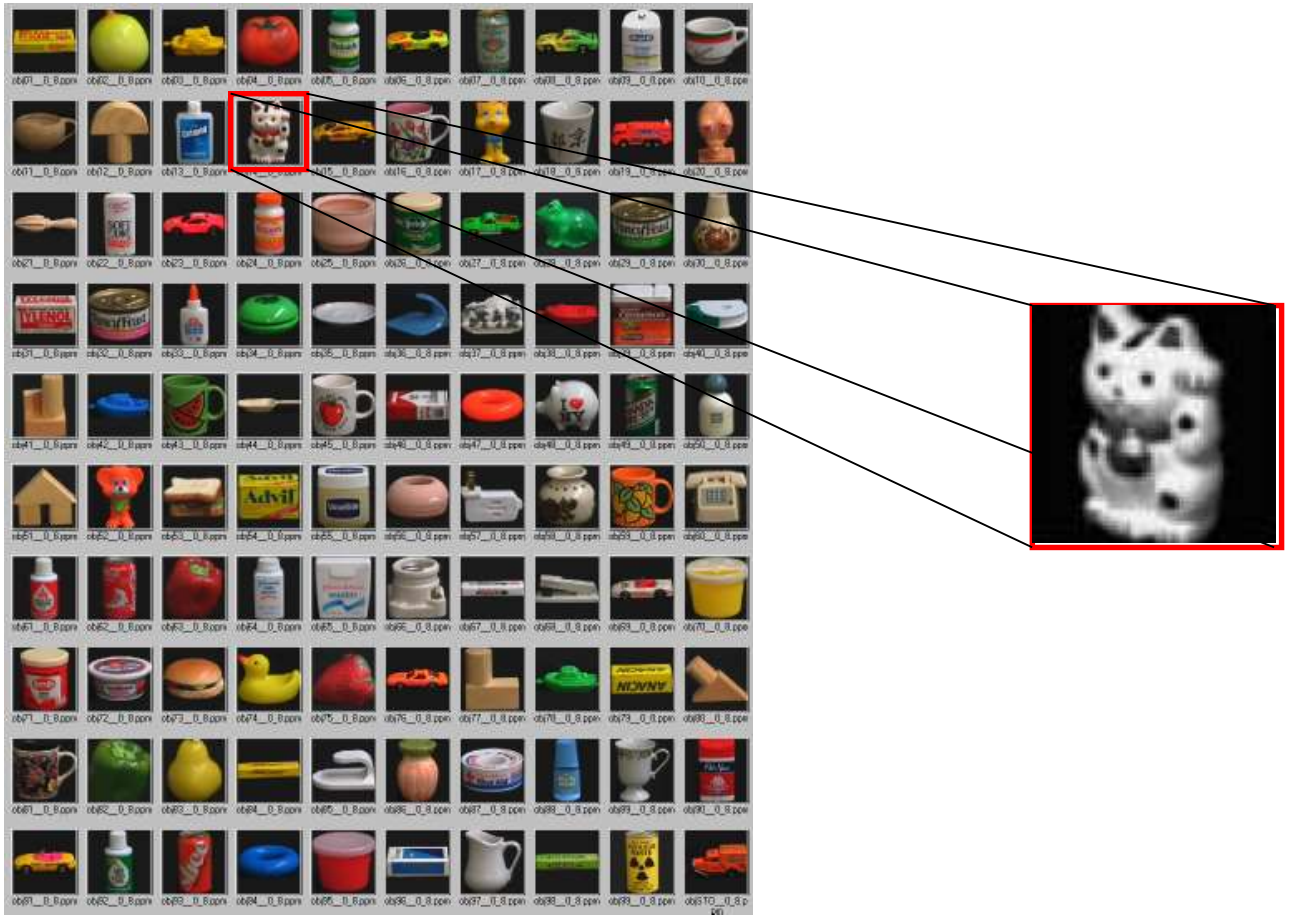
Медицинские изображения



Source: https://www.omicsonline.org/open-access/change-detection-of-medical-images-for-three-dimensional-volumetricdata-jtco-1000135.php?aid=65108

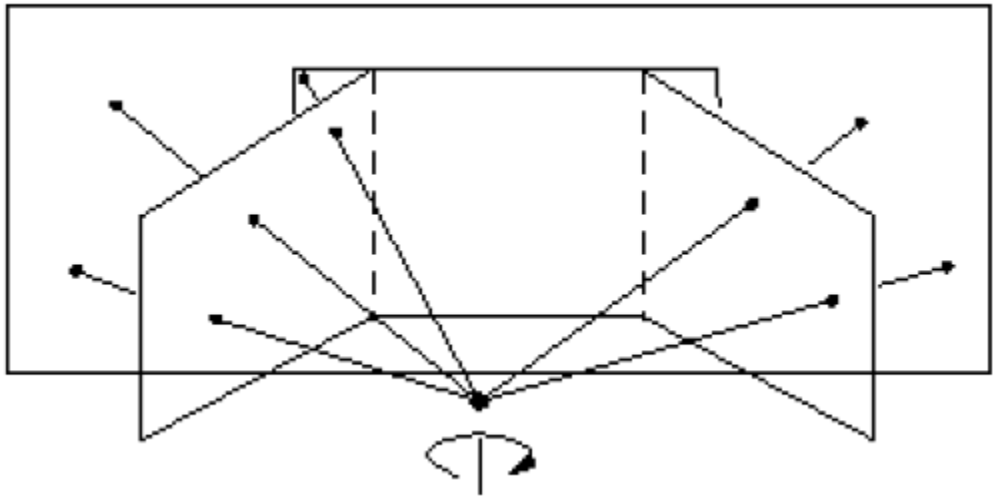
Selected slices from a real T2-weighted MR of a brain volume taken in 2011 and its follow-up in 2013. (a) and (b) Reference and Test volume, (c) Reference aligned to test after six affine transformations, (d) and (e) change volumes via differencing before and after the co-registration step respectively done by our algorithm, and (f) change volume via the 3D EigenBlockCD-2 algorithm. Note the new maxillary sinus mucosal thickening shown in green in Figure.6 (f).

Поиск и распознавание

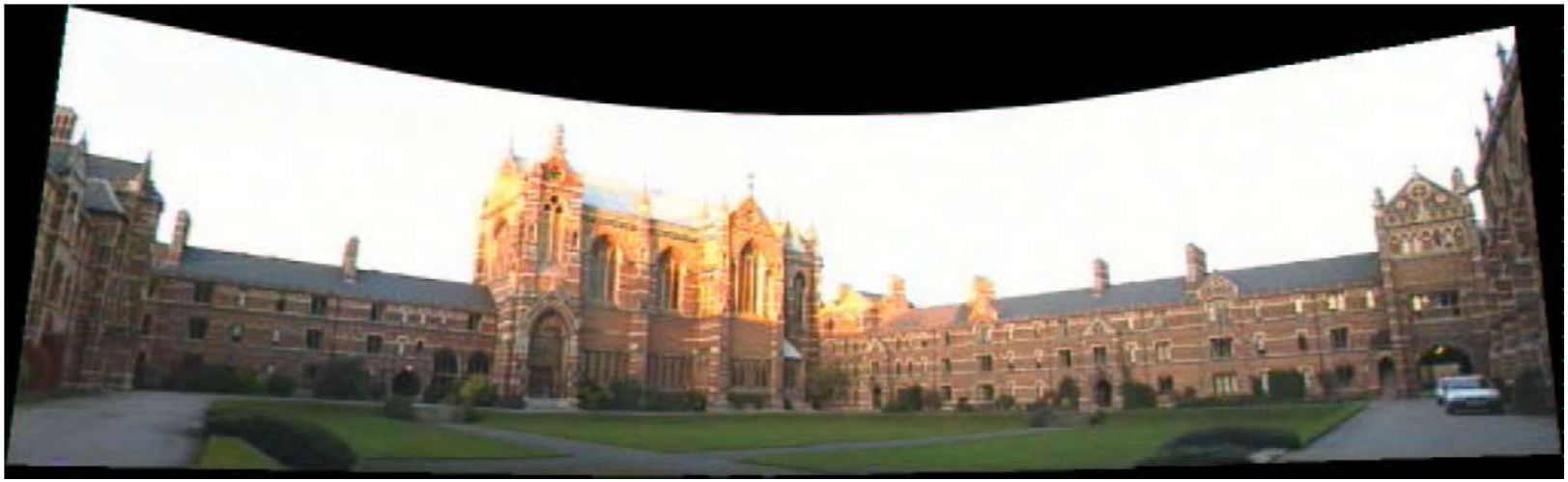


Поиск изображений по содержанию в базе изображений, выделение объектов, классификация изображений

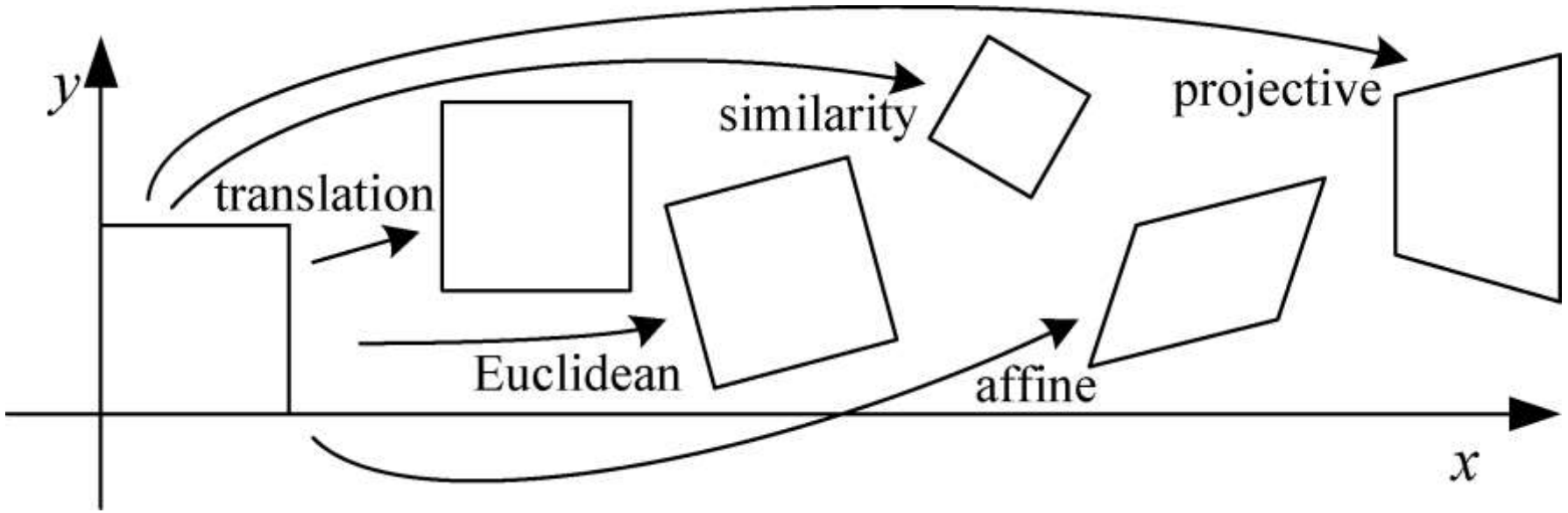
Пример – построение панорамы



Как по шагам это можно сделать?



2D преобразования изображений



Однородные координаты

Для удобства записи преобразований в матричной форме вводятся «однородные» (homogenous) координаты. Они активно используются в проективной геометрии, но нам они только ограниченно понадобятся

Перевод из обычных («неоднородных») в однородные:

$$(x, y) \Rightarrow \begin{bmatrix} wx \\ wy \\ w \end{bmatrix}$$

Однородные координаты
2D точки изображения

$$(x, y, z) \Rightarrow \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

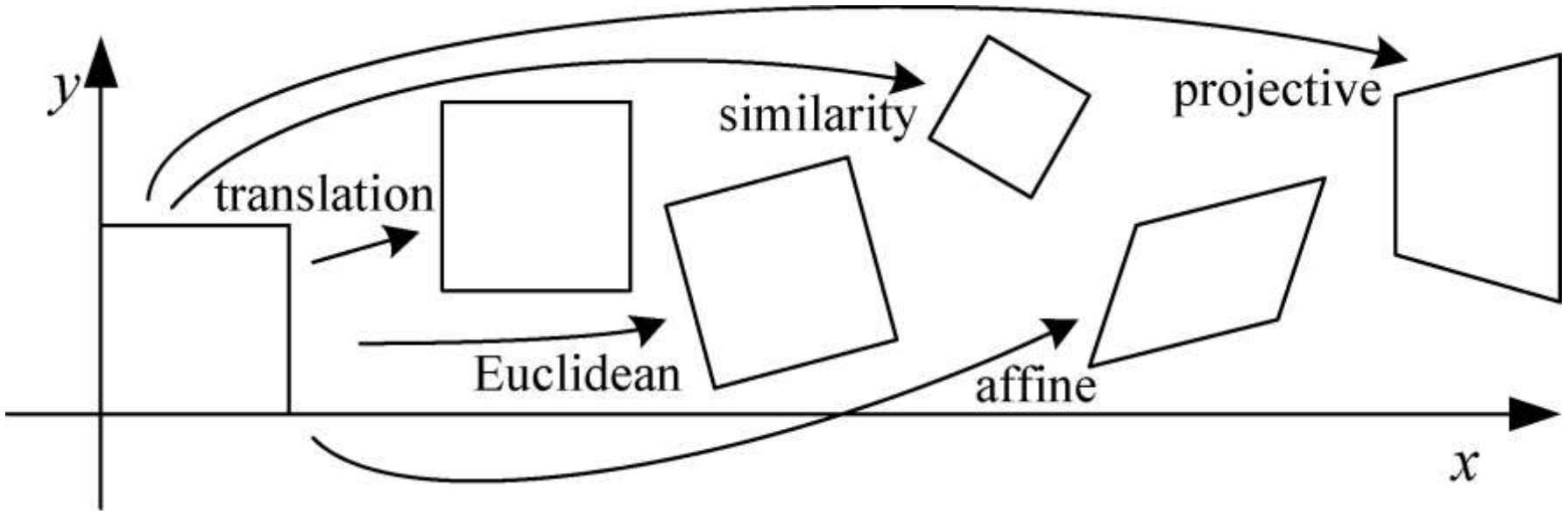
Однородные координаты
3D точки сцены

Перевод из однородных в обычные:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

2D преобразования изображений



Перенос (translation)

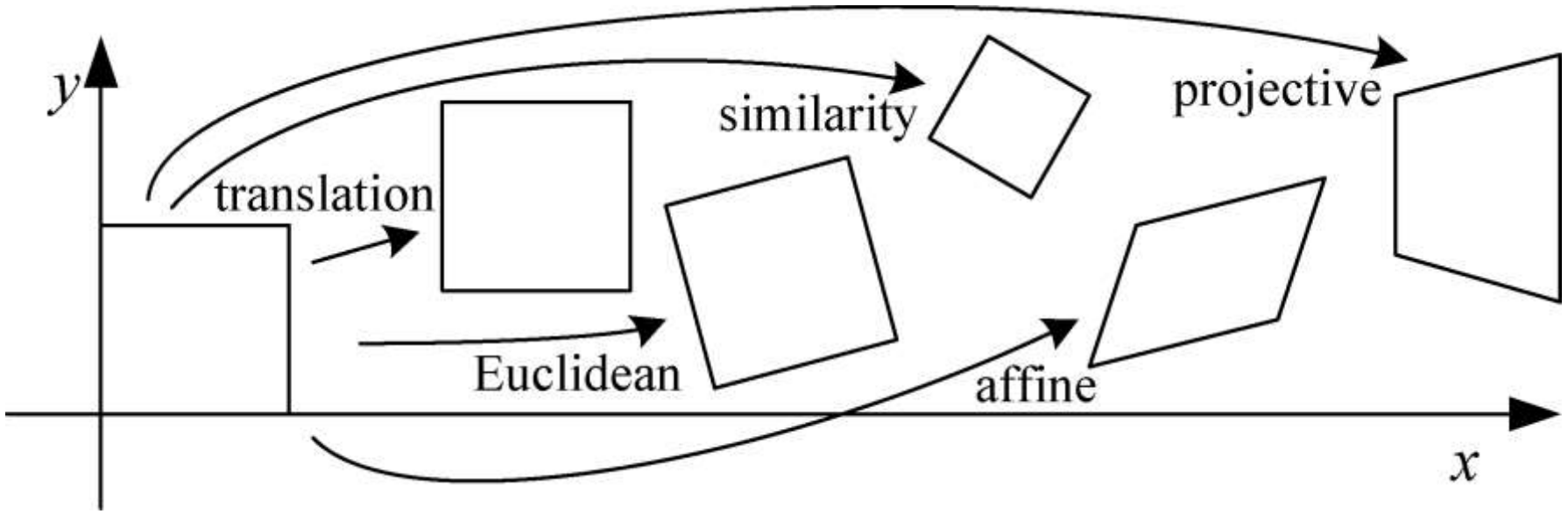
$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{x}}$$

Поворот (rotation)

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{x}}$$

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2D преобразования изображений



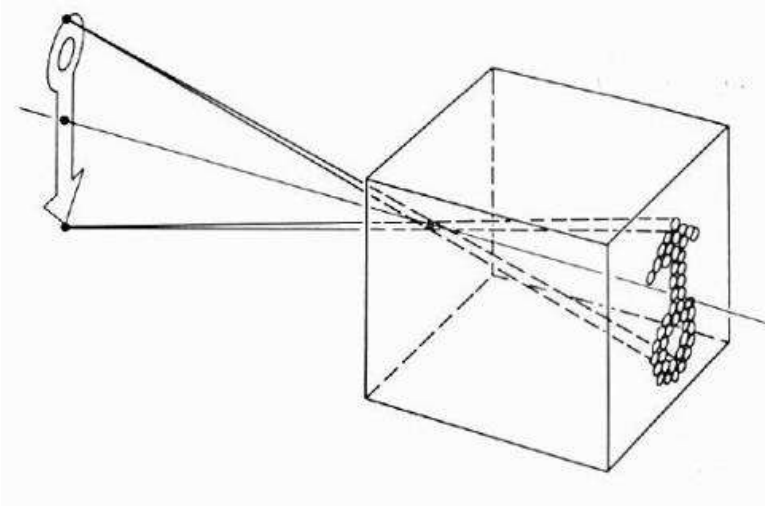
Подобие (similarity)

$$\boldsymbol{x}' = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \quad \tilde{\boldsymbol{x}} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \tilde{\boldsymbol{x}}$$

Аффинное (affine)

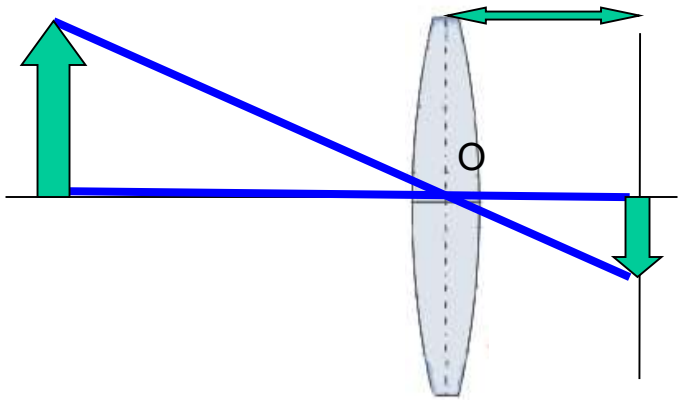
$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \tilde{\boldsymbol{x}}$$

Модель камеры



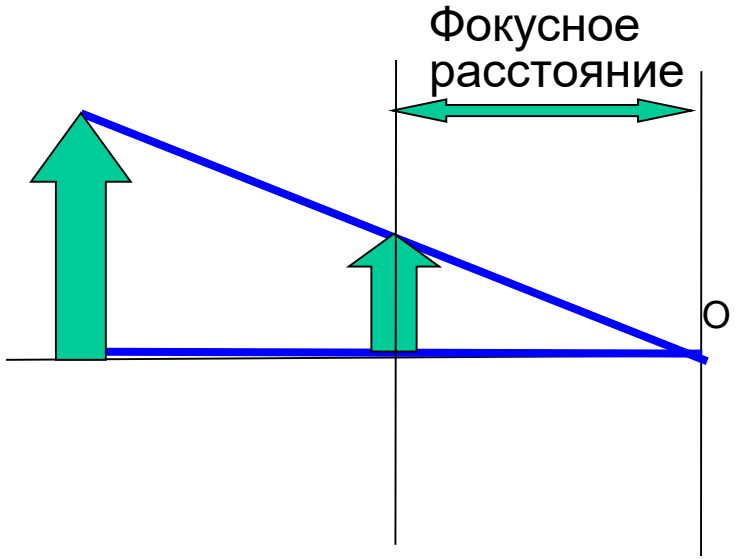
Камера-обскура

Фокусное расстояние

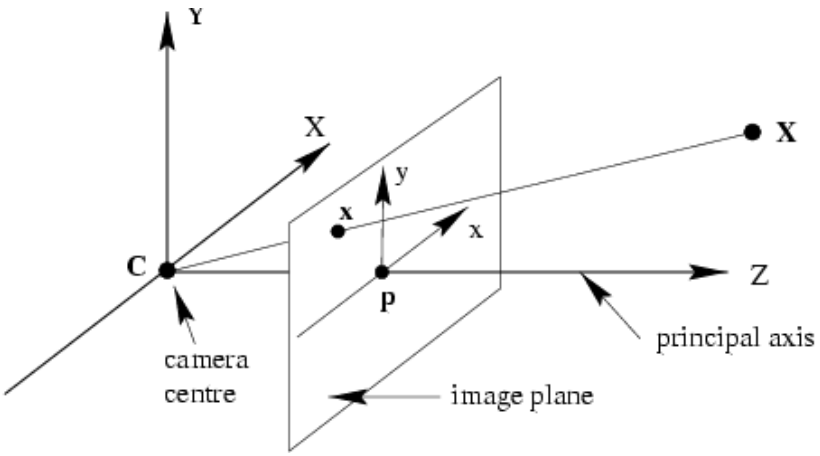


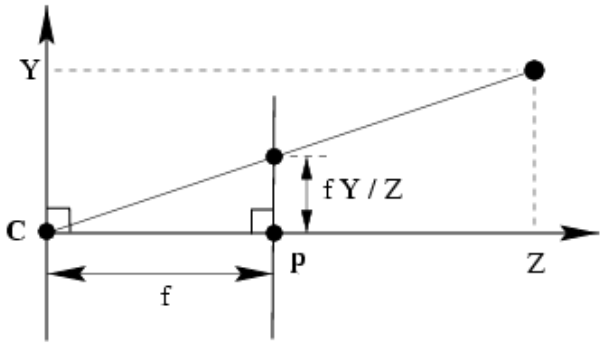
Изображение перевернутое

- Модель камеры-обскуры - перспективная проекция
- Перенесем объект на противоположенную сторону
- То же самое фокусное расстояние!
- Изображение нормальное, не перевернутое



Перспективная проекция





- **C** = (0,0,0) – центр камеры (проекции)
- **X**=**(**X,Y,Z) – точка в 3х мерном пространстве
- **x**=(x,y,f) – проекция **X** на картинную плоскость

C, **x**, **X** лежат на одной прямой

Тогда простейшее уравнение перспективной проекции:

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z}$$

Перспективная проекция – нелинейное преобразование!

Уравнение перспективной проекции

В матричном виде x=PX , где **X**- точка в 3D, x **–** точка на цифровом изображении (в пикселях), **P –** матрица проекции

$$P = K \cdot [I \mid 0] \cdot C^{-1}$$

Матрица проекции камеры

Внутренняя калибровка

Центральная проекция

Внешняя калибровка

$$K = \begin{bmatrix} a_x & s & c_x \\ 0 & a_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f/pix & s & c_x \\ 0 & f/pix & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Внутренняя калибровка отображает точки с идеальной картинной плоскости в пиксели

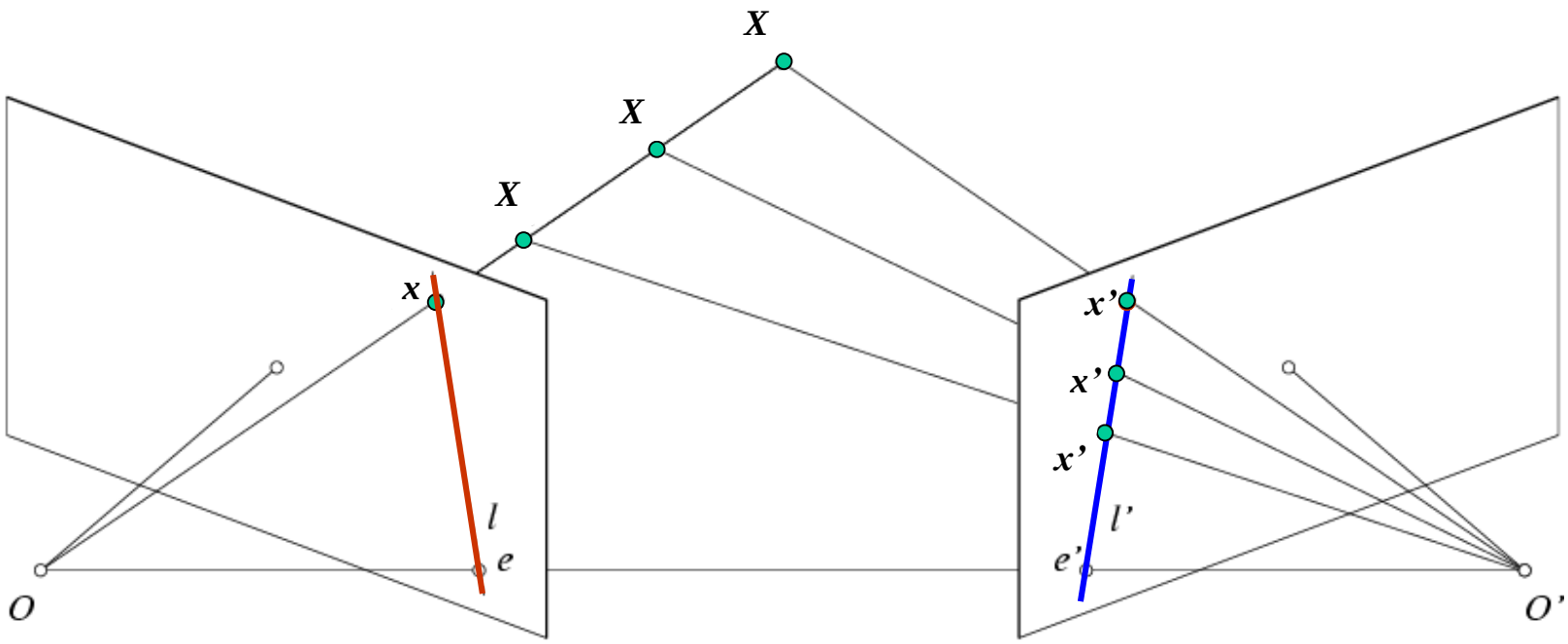
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Проекция на идеальную картинную плоскость

$$C^{-1} = \left(\begin{bmatrix} R & T \\ [0,0,0] & 1 \end{bmatrix} \right)^{-1} = \begin{bmatrix} R^T & -R^T T \\ [0,0,0] & 1 \end{bmatrix}$$

Матрица внешней калибровки определяется положением и ориентацией камеры в мировой системе координат

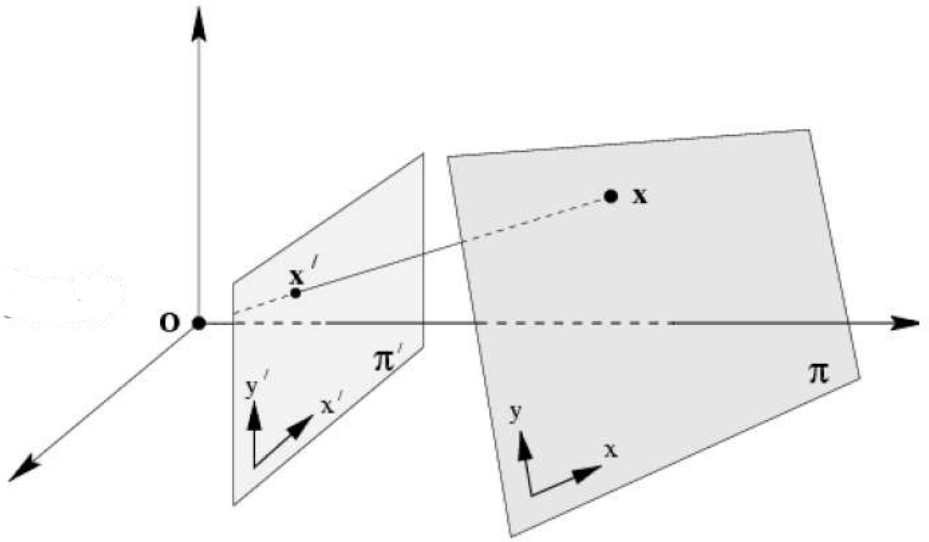
3D преобразование изображений



В общем случае преобразование между 2-мя изображениями с разных ракурсов требует для каждой точки изображения знать «глубину»

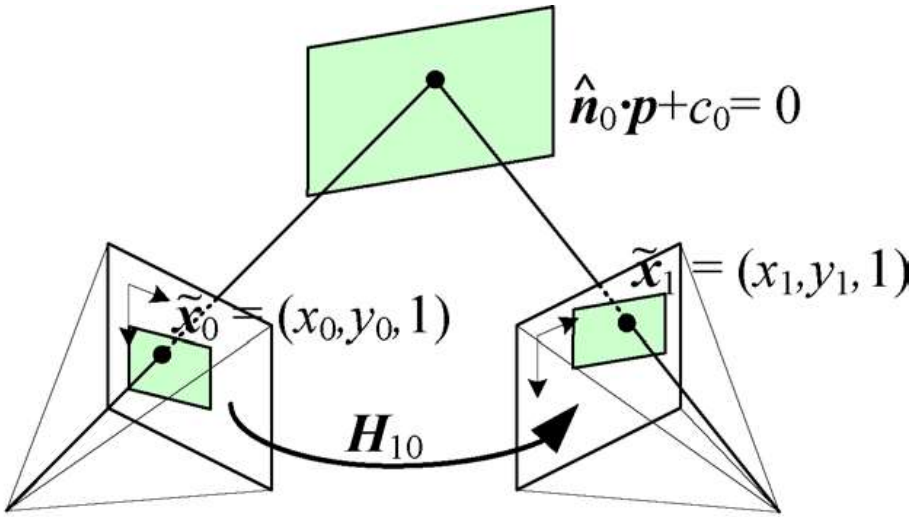
Гомография

- Особый случай - перспективное преобразование плоскости (гомография)
- Глубину знать не обязательно!
- В однородных координатах можем записать как x’=Hx
- 8 степеней свободы (с точностью до масштаба)

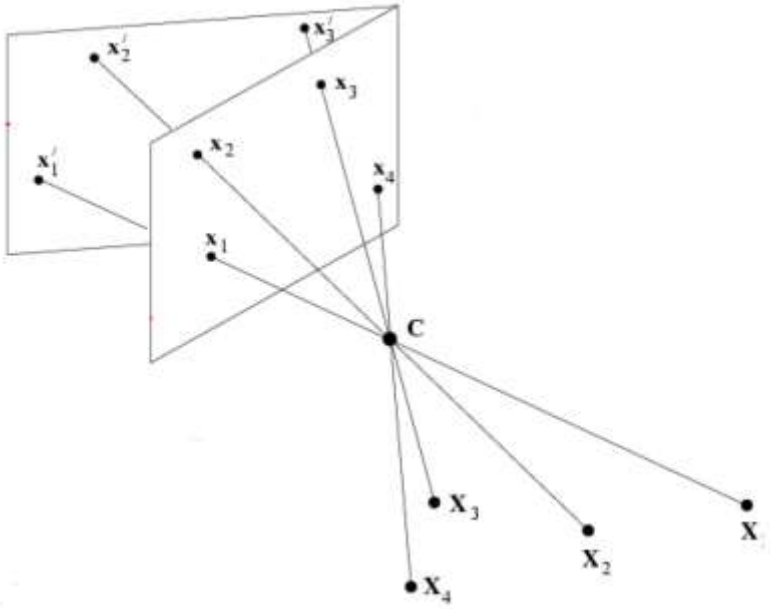


$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{12} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Применение гомографии



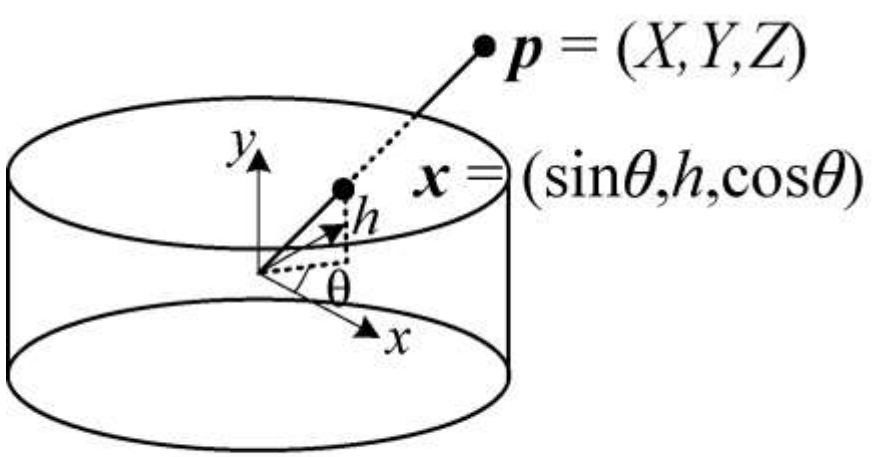
Плоская сцена

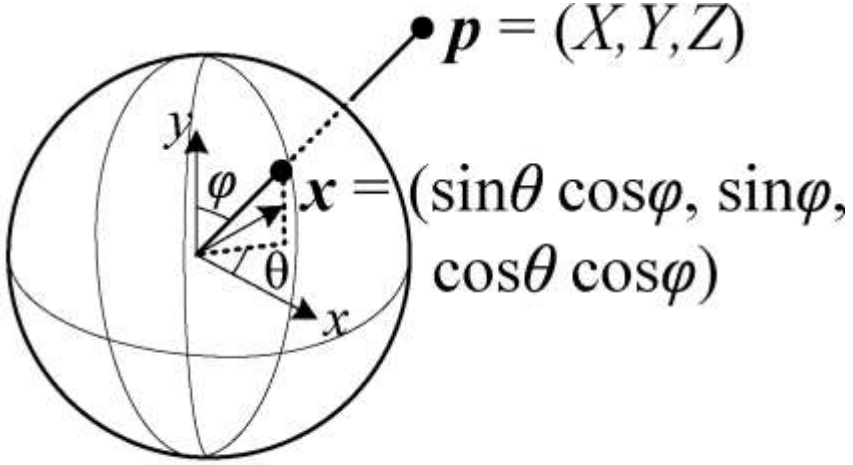


Камера на штативе

Вывод: если строим панораму или карту местности (считая её «плоской»), то одно изображение преобразуется в другое с помощью гомографии

Цилиндрические и сферические координаты





Координаты на цилиндре (θ, h)

$$(\sin\theta, h, \cos\theta) \propto (x, y, f)$$

Преобразование:

$$x' = s\theta = s\tan^{-1}\frac{x}{f},$$

$$y' = sh = s\frac{y}{\sqrt{x^2+f^2}}$$

Координаты на сфере (θ, ϕ)

$$(\sin\theta\cos\phi, \sin\phi, \cos\theta\cos\phi) \propto (x, y, f)$$

Преобразование:

$$x' = s\theta = s\tan^{-1}\frac{x}{f},$$

$$y' = s\phi = s\tan^{-1}\frac{y}{\sqrt{x^2+f^2}}$$

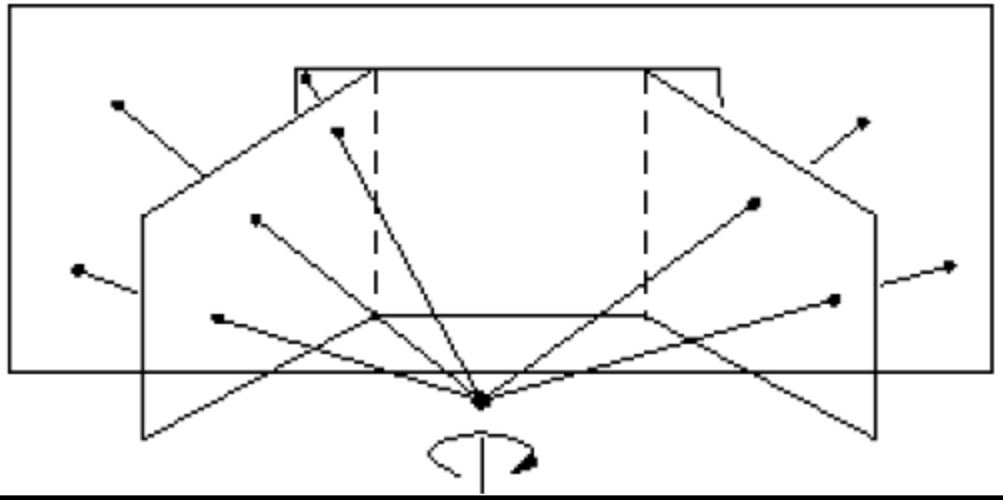
Пример цилиндрической панорамы

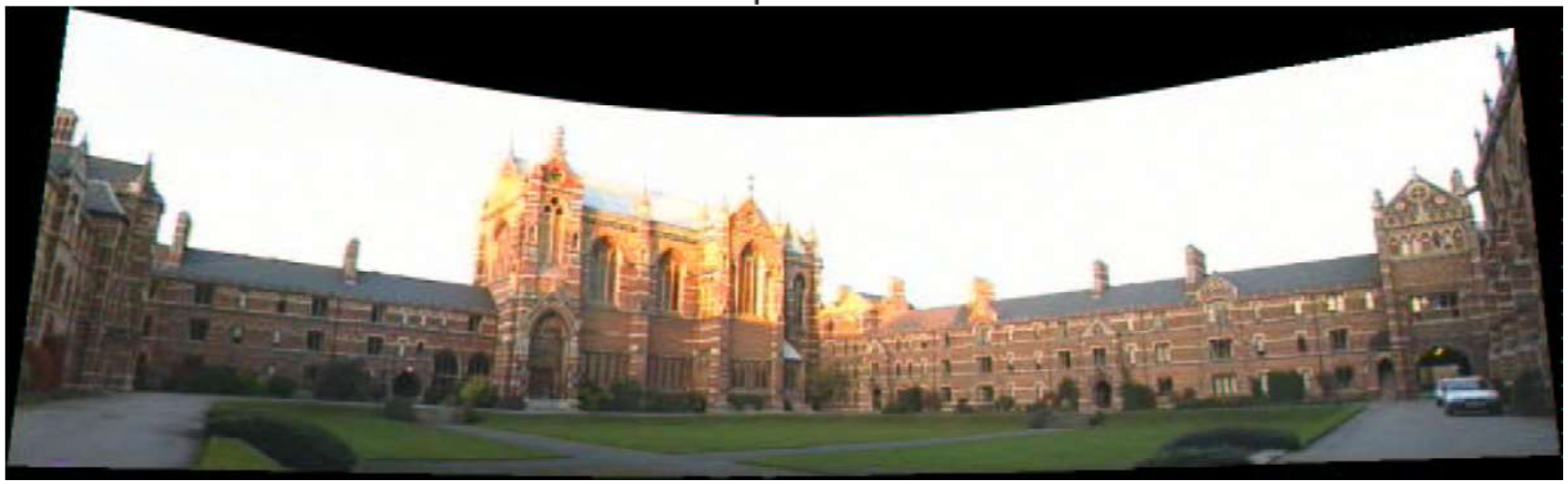
(a)

(b)

Пример сферической панорамы

Как посчитать преобразование?





Прямое выравнивание (Direct Alignment)

- Выберем модель преобразования
  - Пр.: перенос T при построении цилиндрической панорамы
- Зададим какие-то параметры преобразования T
- Можем оценить «качество» сопоставления через попиксельное сопоставление изображений в области пересечения

Метрики

Sum of Squared Distances (SSD)

$$E_{\mathrm{SSD}}(\boldsymbol{u}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 = \sum_i e_i^2,$$

$$e_i = I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)$$

Робастные метрики

- Sum of Robust Distances (SRD)

$$E_{\mathrm{SRD}}(\boldsymbol{u}) = \sum_i \rho(I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)) = \sum_i \rho(e_i).$$

$\rho(e)$ - робастная функция, которая при больших ошибках растёт медленнее, чем квадрат

- Sum of Absolute Distances (SAD)

$$E_{\mathrm{SAD}}(\boldsymbol{u}) = \sum_i |I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)| = \sum_i |e_i|$$

- Дифференцируемый вариант - функция Geman-McClure:

$$\rho_{\mathrm{GM}}(x) = \frac{x^2}{1 + x^2/a^2}$$

Корреляционные метрики

- Cross-Correlation (CC)

$$E_{\mathrm{CC}}(\boldsymbol{u}) = \sum_i I_0(\boldsymbol{x}_i) I_1(\boldsymbol{x}_i + \boldsymbol{u})$$

- Normalized Cross-Correlation (NCC)

$$E_{\mathrm{NCC}}(\boldsymbol{u}) = \frac{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]\,[I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]}{\sqrt{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]^2 [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]^2}}$$

$$\overline{I_0} = \frac{1}{N} \sum_i I_0(\boldsymbol{x}_i) \quad \text{and}$$

$$\overline{I_1} = \frac{1}{N} \sum_i I_1(\boldsymbol{x}_i + \boldsymbol{u})$$

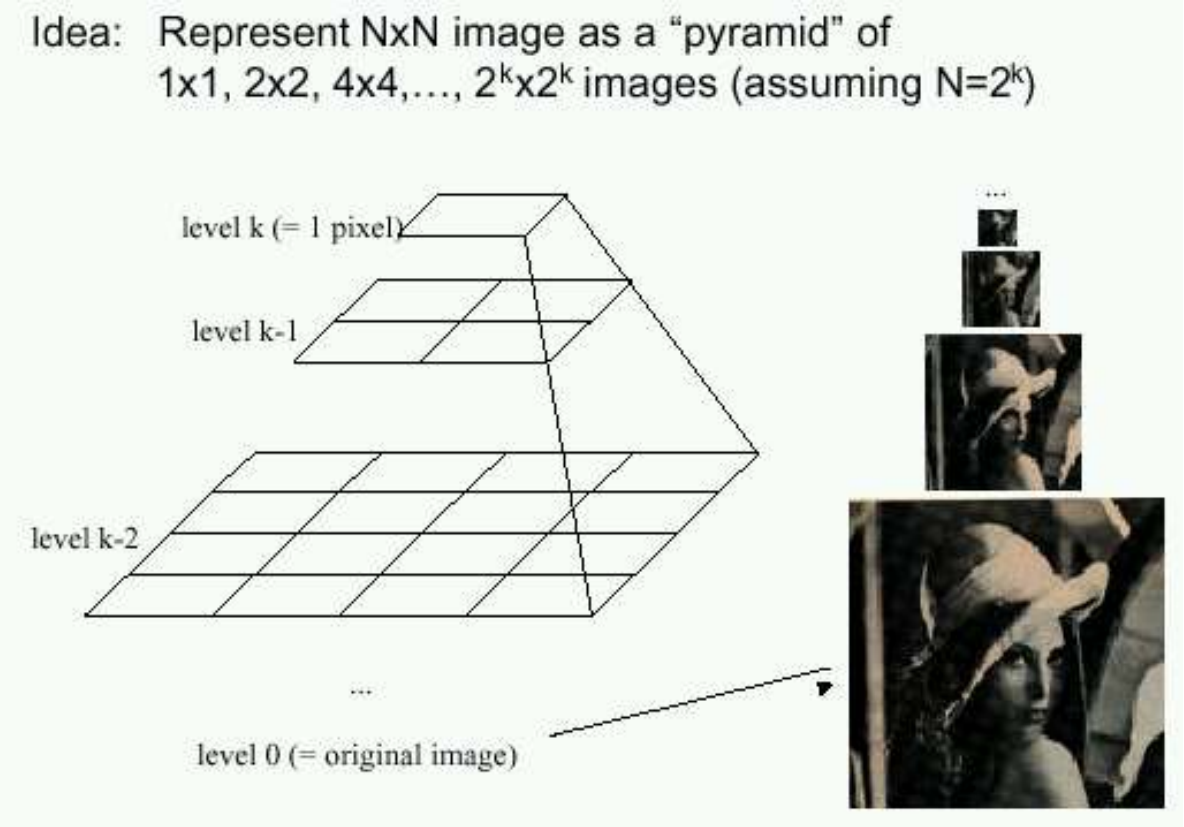
Поиск преобразования

- Имеем функцию ошибки, которую можем оценить для текущих параметров преобразования
- Как найти оптимальные параметры?

- Оптимизация функции ошибки по параметрам!
  - Дискретизация параметров
    - Перебор всех вариантов (Grid Search)
    - Какой-то метод дискретной оптимизации
  - Непрерывные параметры
    - Градиентный спуск

- Как быть, если преобразование слишком сильное (большой диапазон возможных параметров)?

Пирамиды изображений

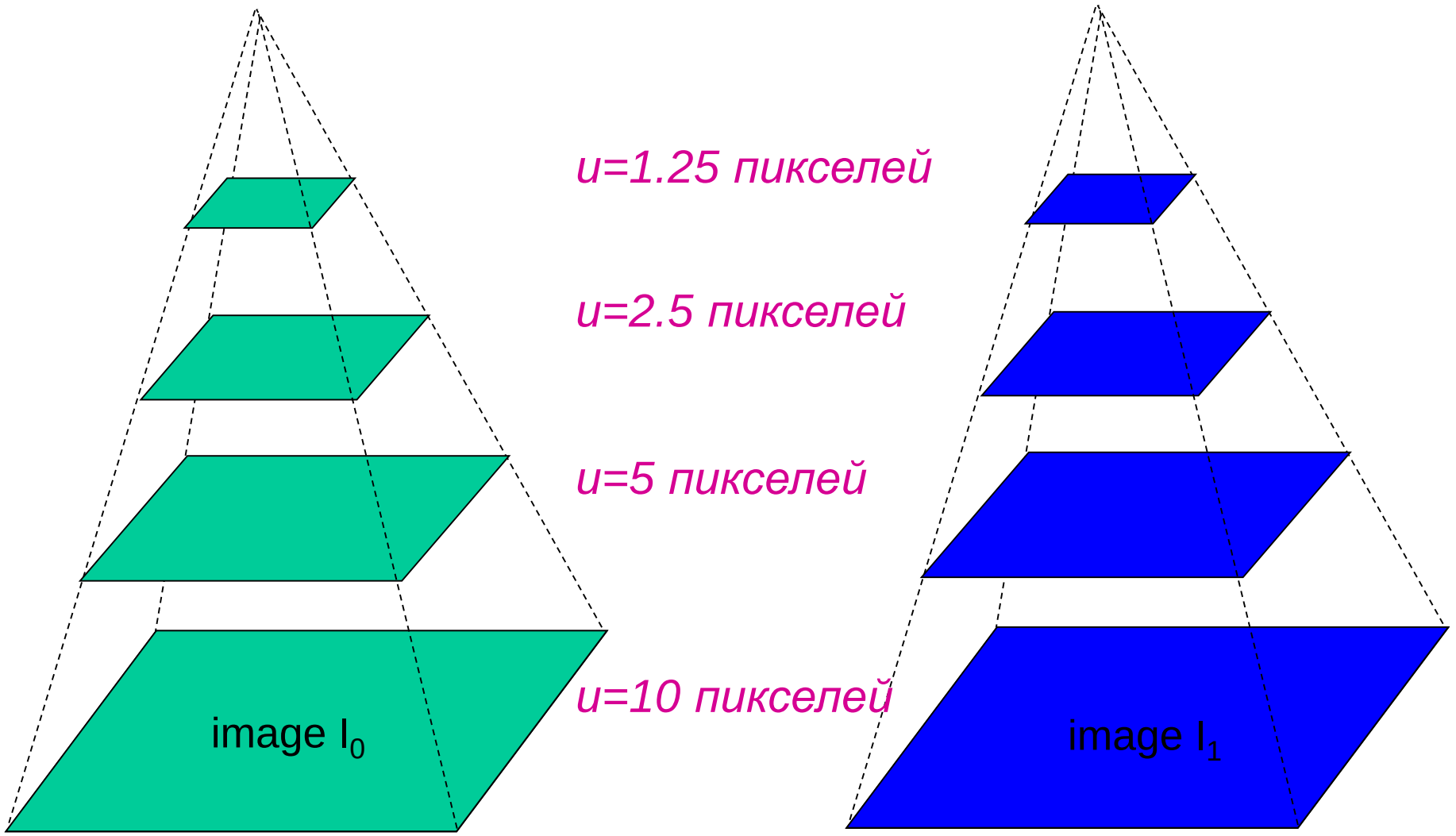


- Известна как Пирамида Гауссиан
- Сглаживаем фильтром Гаусса и берём каждый k-ый пиксель
- В компьютерной графике – “mip map” [Williams, 1983]

P. Burt and E. Adelson, “The Laplacian pyramid as a compact image code,” *IEEE Trans. Commun., vol. 31, no. 4, pp. 532–540, 1983.*

Slide by Steve Seitz

Иерархический метод



Гауссова пирамида для I_0

Гауссова пирамида для I_1

Резюме прямого сопоставления

- Выбираем модель преобразования
- Выбираем функцию ошибки (согласования изображений)
- Применяем метод оптимизации для поиска оптимальных параметров

Вводный пример

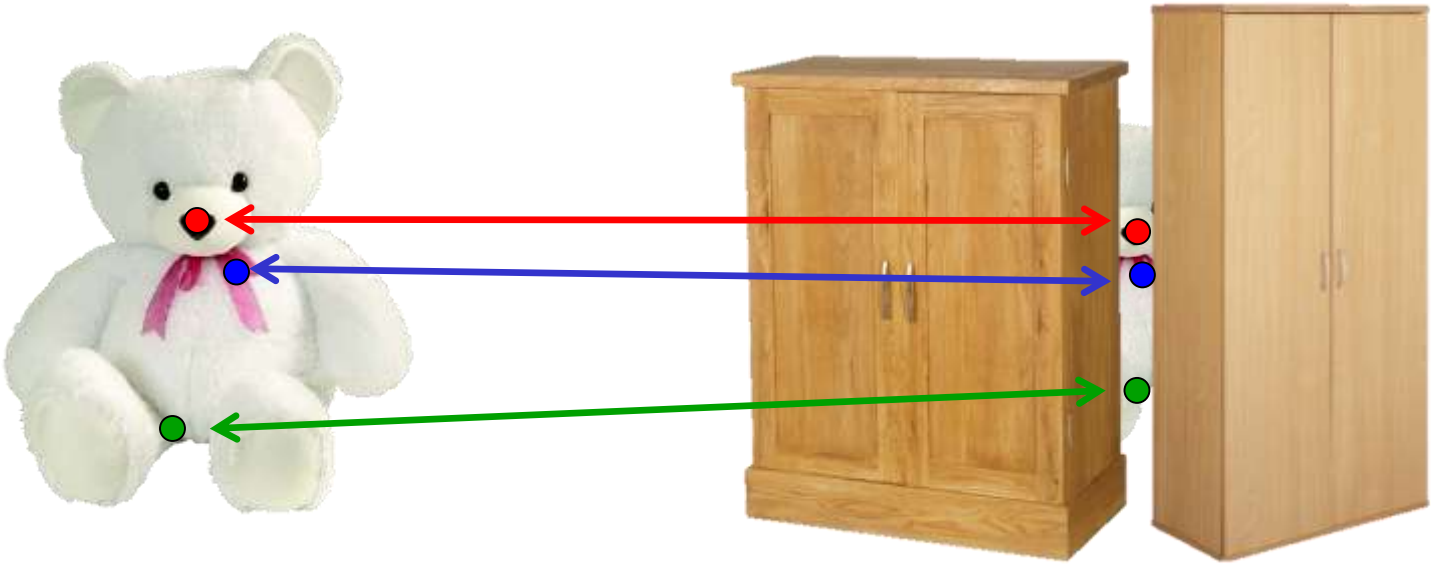
Где медведь?

Задача №1

Задача №2

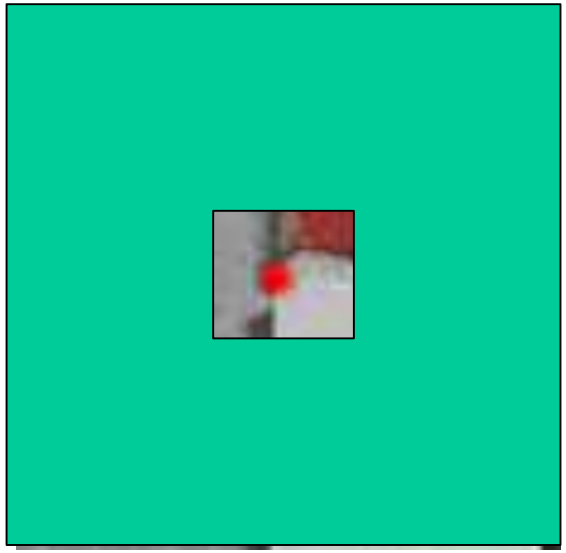
- Почему во втором случае было легче его найти?
- Были видны «характерные» фрагменты медведя

Особенности (features)

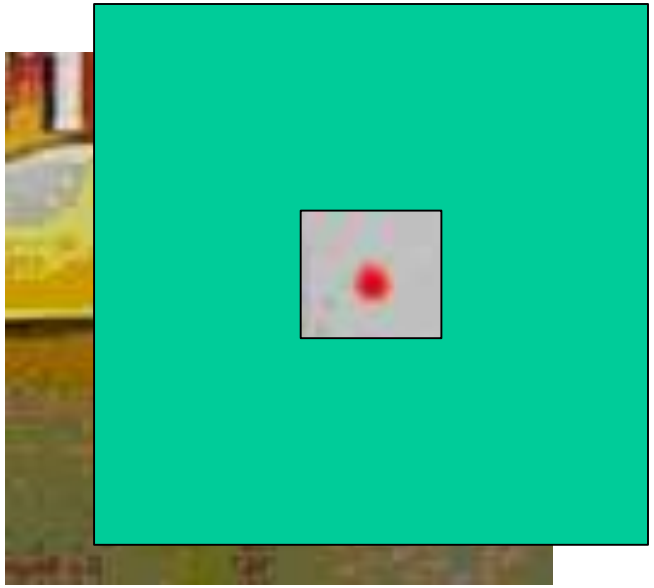


- «Хорошо различимые фрагменты» объекта
  - «особенности» (features)
  - «характеристические точки» (characteristic points)
  - «ключевые точки» (keypoints)
  - «локальные особые точки» (local feature points)
- Характерные фрагменты позволяют справится с изменениями ракурса, масштаба и перекрытиями

Локальные особенности



Пример особой точки



Пример точки, не являющейся особой

- Какая из двух точек является характерной («особой»)?
- Локальная (особая) точка *p* изображения *I* должна обладать «характерной окрестностью» D, т.е. отличаться от всех точек в некоторой окрестности *p*
- Для определения, является ли точка «характерной», нам достаточно только её окрестности

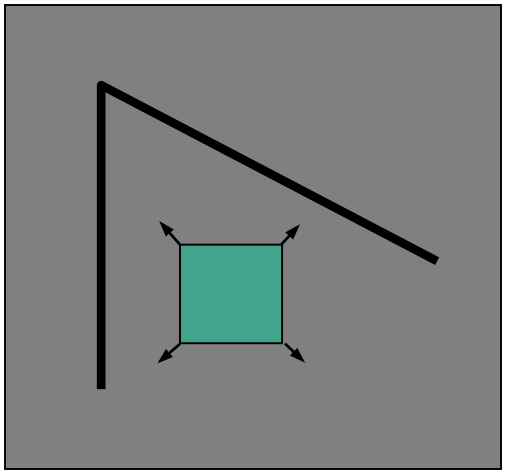
Два основных класса особенностей

уголки (corners)

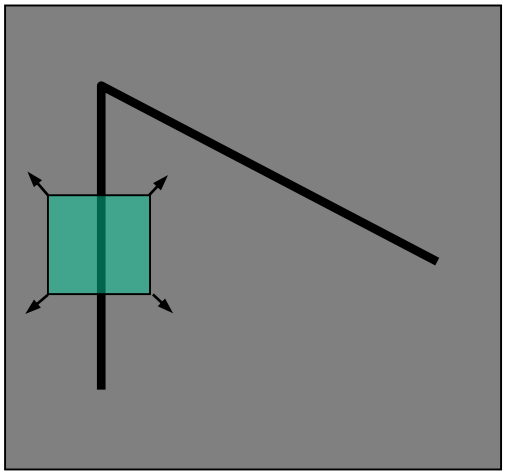
пятна (blobs)

Локальные особенности

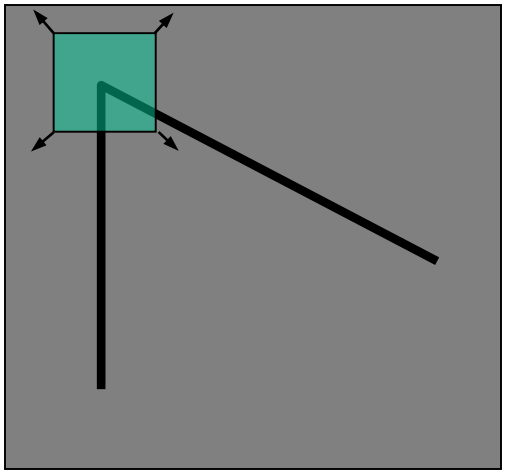
Проведём эксперимент, будем рассматривать разные точки на изображении и проверять, являются ли они локальными особенностями



монотонный регион:
в любом направлении изменений нет

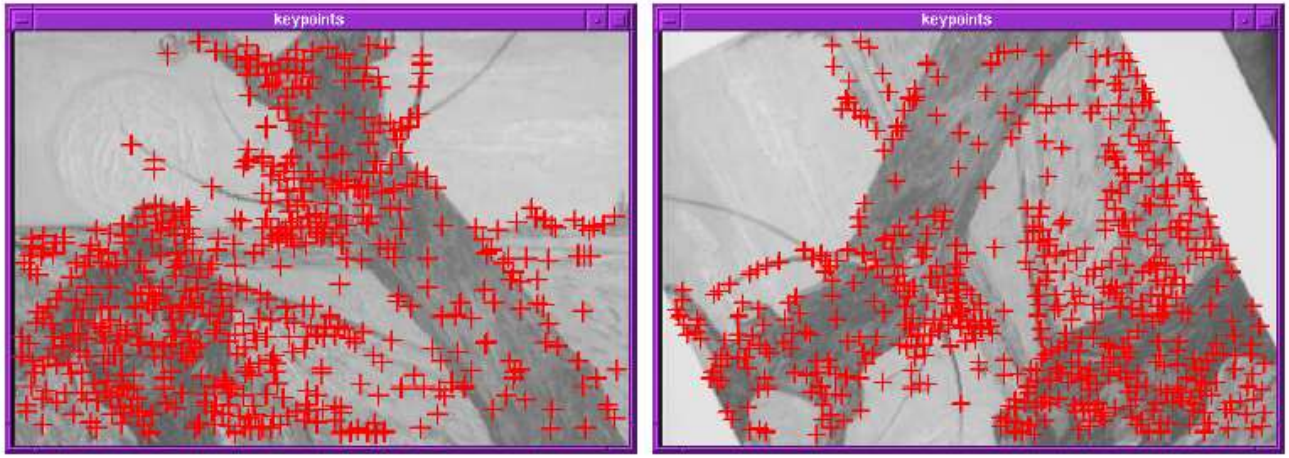


«край»:
вдоль края изменений нет



«уголок»:
изменения при перемещении в любую сторону

Детектор Харриса



- Наиболее популярный детектор локальных особенность точек – детектор Харриса (Harris)
- Ищет такие точки (x,y), окрестность которых меняется при любом сдвиге $(x+u, y+v)$
- Такие точки обычно оказываются углами, поэтому метод ещё называют «детектор углов»

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988

Устройство метода

Изменение окрестности точки (x,y) при сдвиге [*u,v*]:

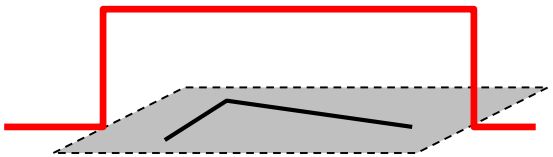
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Функция окна

Сдвиг яркости

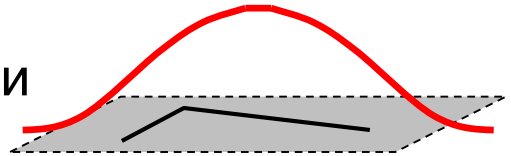
Яркость

Функция окна $w(x,y)$ =



1 в окне, 0 вне

или



Гауссиана

Source: R. Szeliski

Устройство метода

Изменение окрестности точки при сдвиге [*u,v*]:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Разложение в ряд Тейлора 2го порядка *I*(*x*,*y*) вокруг (x,y)
(билинейная интерполяция при маленьких сдвигах)

$$[I(x+u, y+v) - I(x,y)]^2 \cong [I(x,y) + I_x u + I_y v - I(x,y)]^2$$

$$= [I_x u + I_y v]^2 = I_x^2 u^2 + 2 I_x I_y uv + I_y^2 v^2$$

$$= (u,v)\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}(u,v)^T$$

Slide by S. Lazebnik

Устройство метода

Итого изменение окрестности можно свести к:

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

где *M* – матрица 2×2 вычисленная по частным производным:

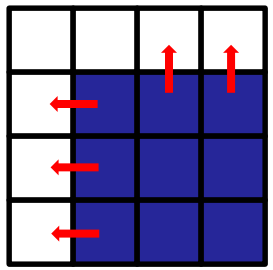
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Slide by S. Lazebnik

Интерпретация матрицы моментов

Рассмотрим случай, когда градиенты выровнены по осям (вертикальные или горизонтальные)



$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Если одно из λ близко к 0, тогда это не угол, и нужно искать другие точки

Slide by S. Lazebnik

Общий случай

M – симметричная, поэтому её можно привести к диагональному виду:

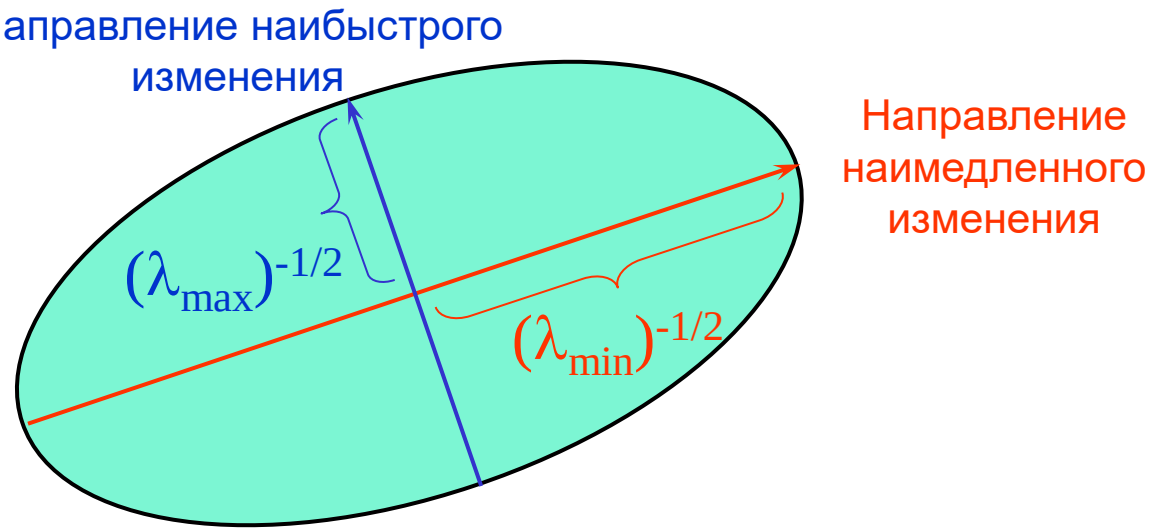
$$M = R^{-1} D R = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

R – ортогональная матрица из собственных векторов M, D – диагональная из собственных значений M

Матрицу *M* можно визуализировать в виде эллипса, у которого длины осей определены собственными значениями, а ориентация определена матрицей R

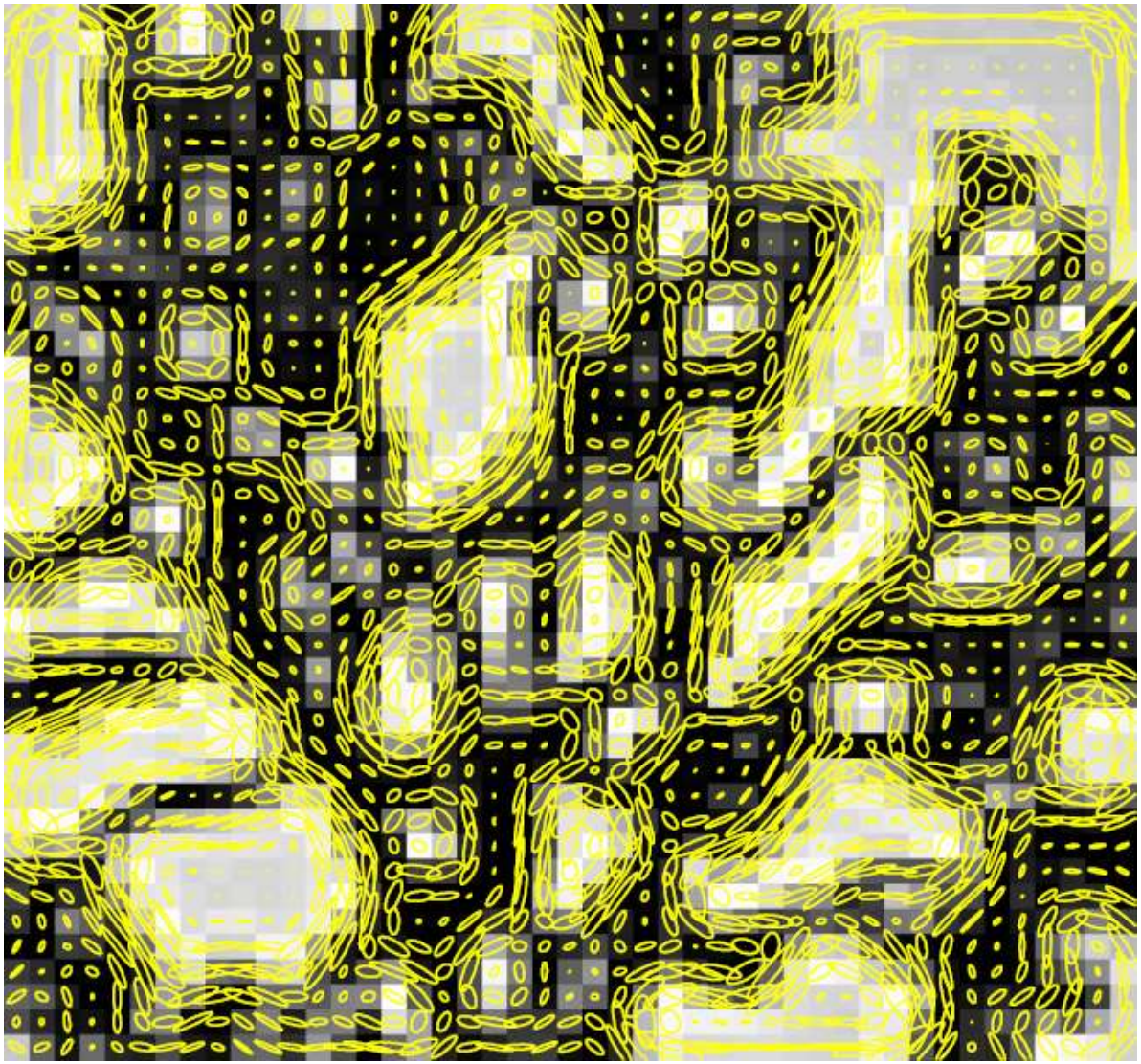
Уравнение эллипса:

$$[u \;\; v] \;\; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



Slide by S. Lazebnik

Пример



Визуализация матриц вторых моментов (Гессианов)

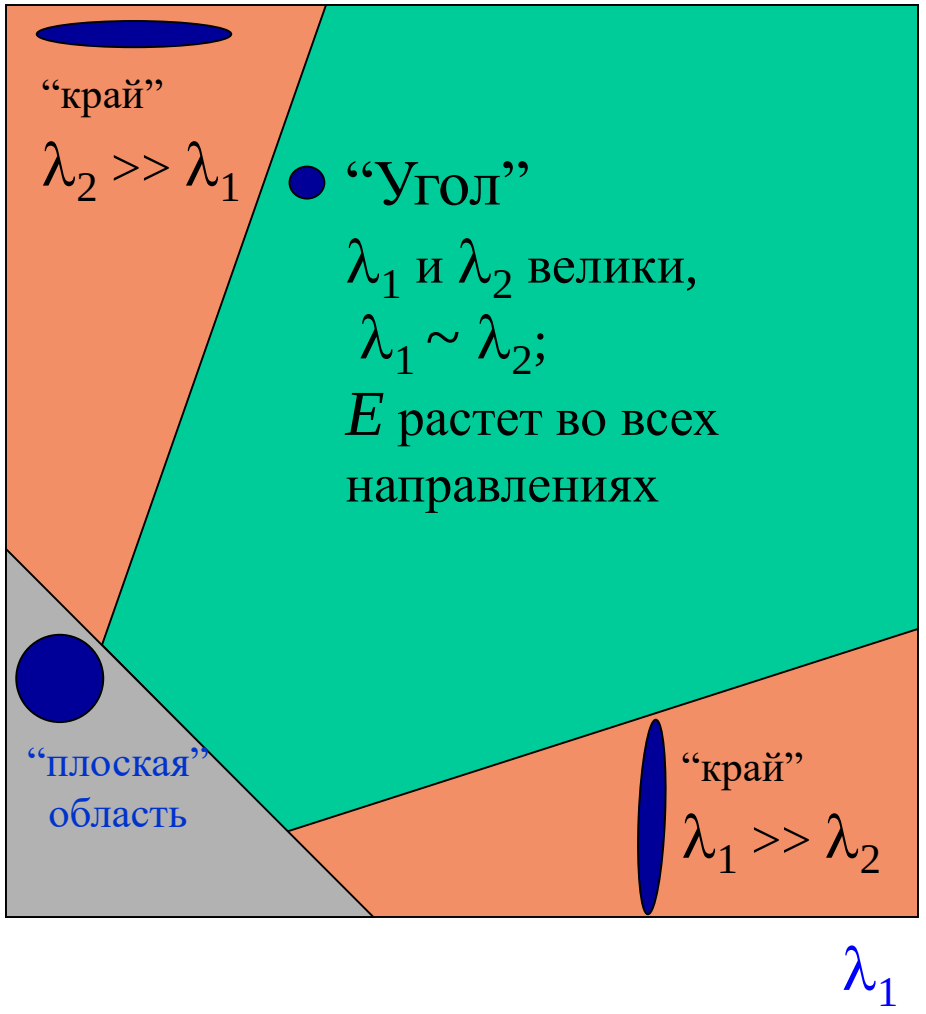
Зависимость E от λ

Классификация точек изображения по собственным значениям матрицы производных *M*

$$E(u,v) = (u,v)M(u,v)^T$$

$$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

λ_1 и λ_2 малы;
E не меняется по всем направлениям



Функции отклика углов

- Функция отклика угла по Харрису:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

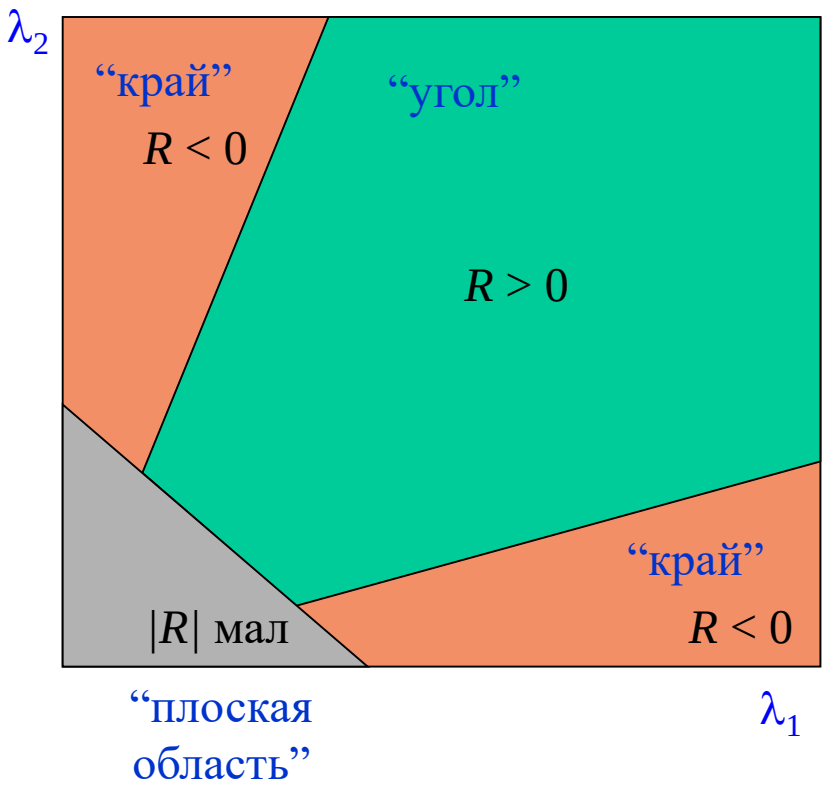
$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

$$(k = 0.04\text{-}0.06)$$

- Функция по Фёрстнеру (Forstner):

$$R = \det M \,/\, \operatorname{trace} M$$

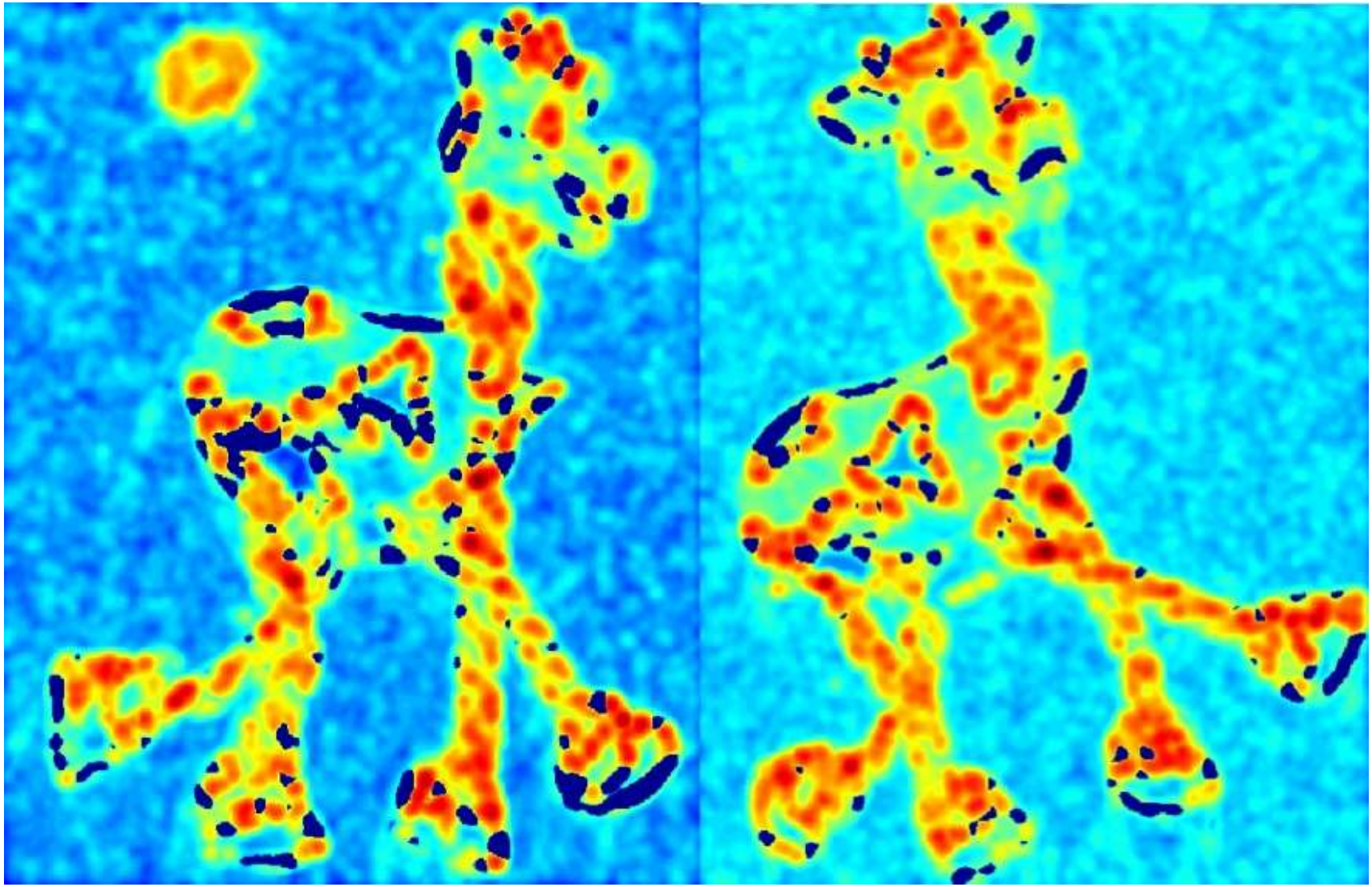


Foerstner, W; Gulch. "A Fast Operator for Detection and Precose Location of Distinct Points, Corners and Centres of Circular Features". *ISPRS 1997*

Демонстрация по шагам

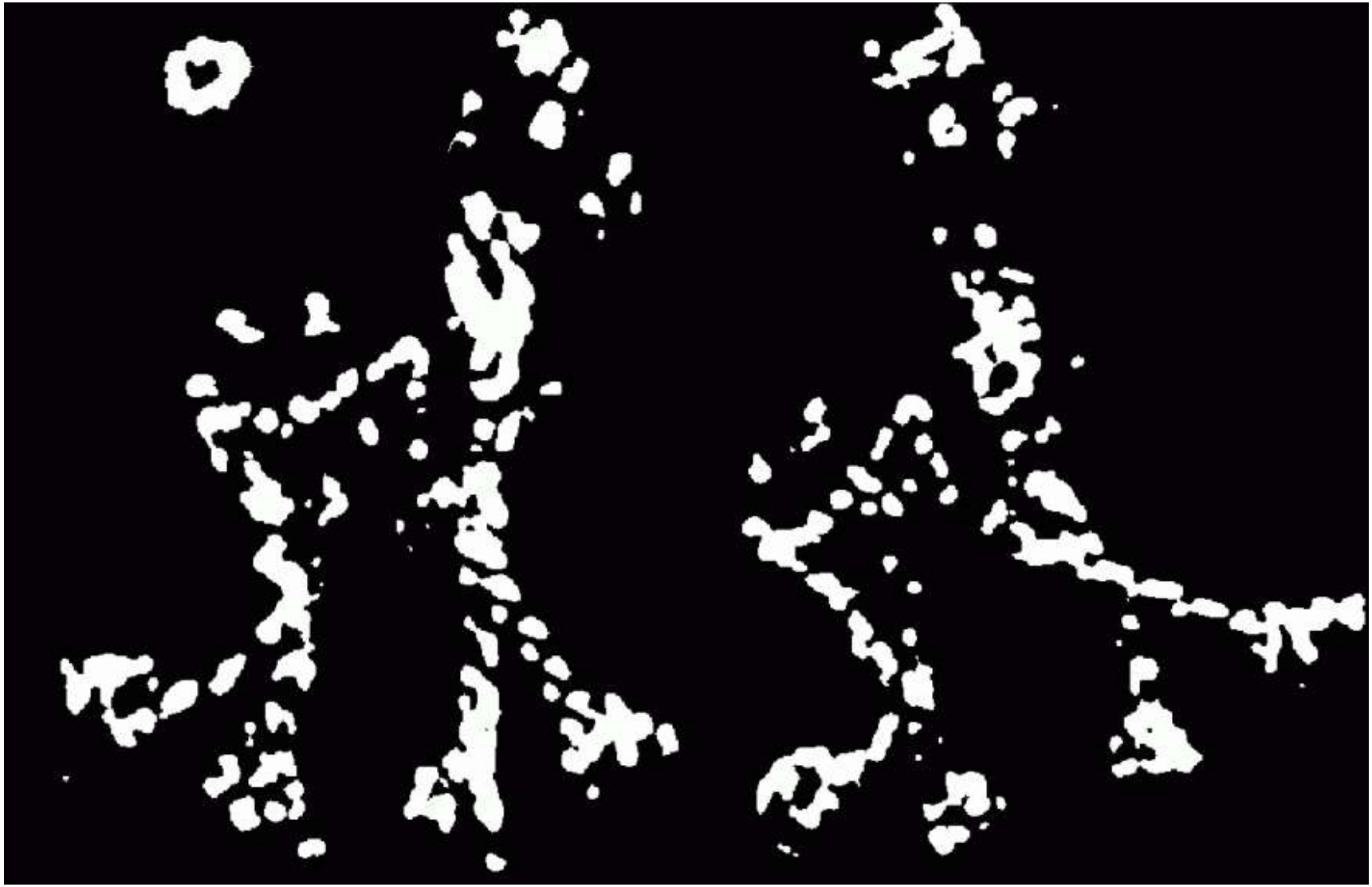
Демонстрация по шагам

Вычисление отклика угла *R*



Демонстрация по шагам

Найдём точки с большим откликом R>порог

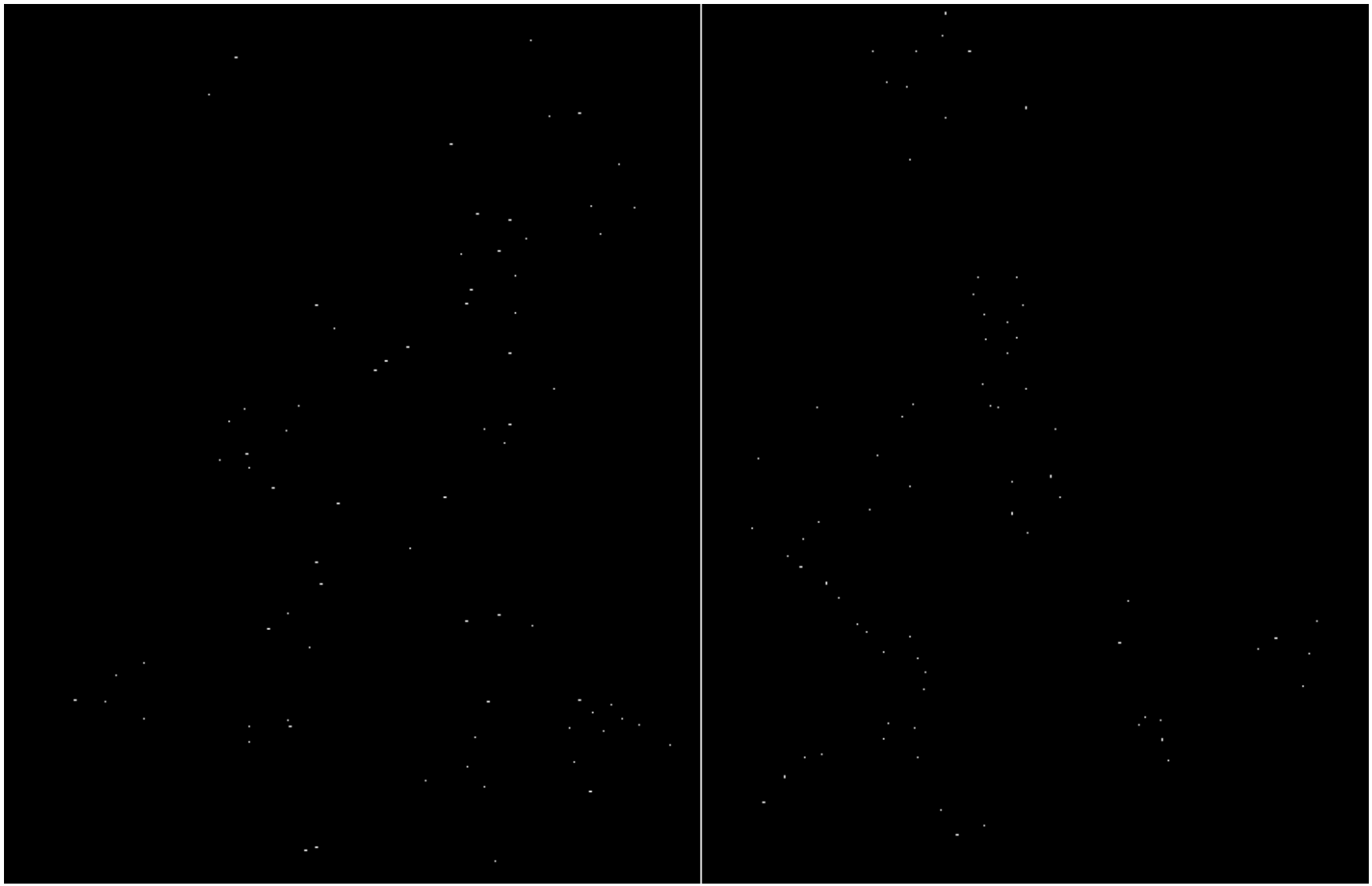


Демонстрация по шагам

- Как быть с тем, что функция отклика угла больше порога в некоторых областях?
- Как нам выбрать конкретные точки в областях?

Демонстрация по шагам

Оставим только точки локальных максимумов R



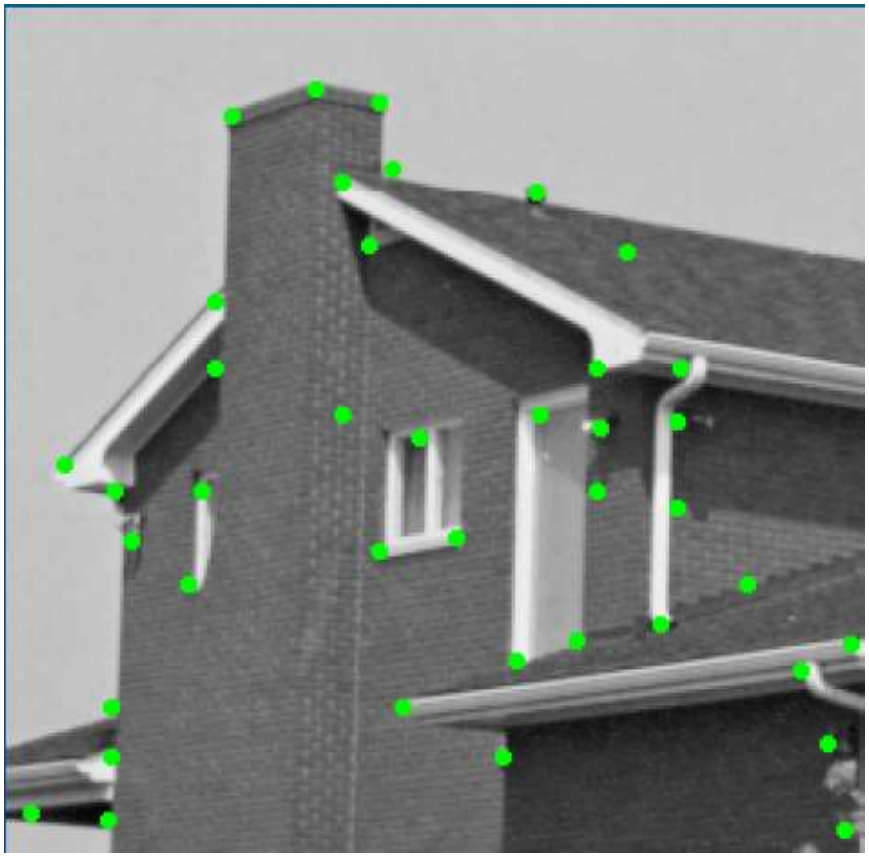
Демонстрация по шагам

Алгоритм детектора Харриса

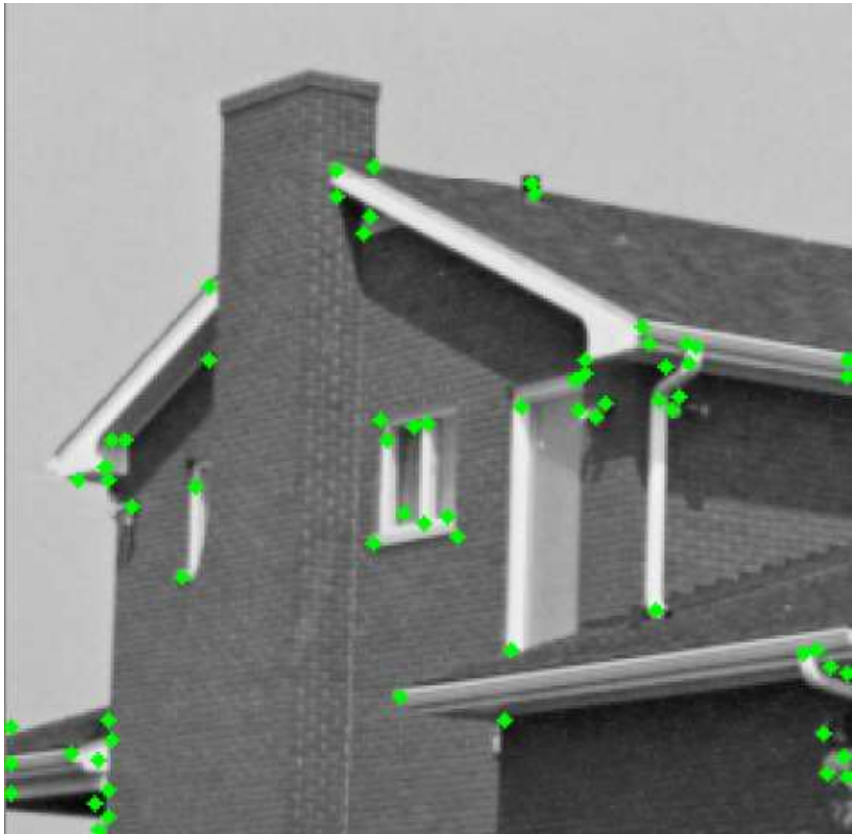
1. Вычислить градиент изображения в каждом пикселе
   - С использованием гауссова сглаживания
2. Вычислить матрицу вторых моментов M по окну вокруг каждого пикселя
3. Вычислить отклик угла *R*
4. Отсечение по порогу *R*
5. Найти локальные максимумы функции отклика (non-maximum suppression) по окрестности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

Slide by S. Lazebnik

Результат работы детектора

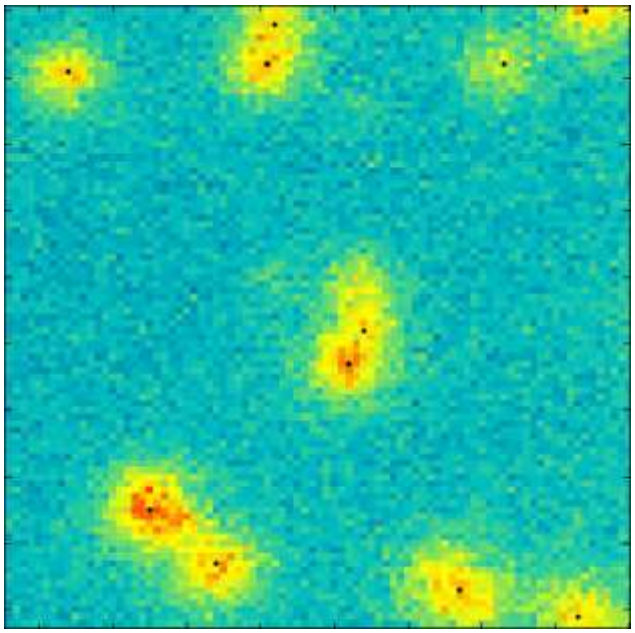


детектор Фёрстнера



детектор Харриса

Важный вывод

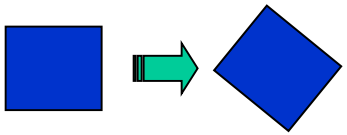


Детектор – суть некоторая функция, локальные максимумы которой мы используем в качестве особенностей изображений

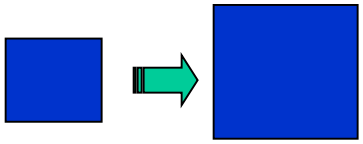
Инвариантность

Что у детектора Харриса с инвариантностью к геометрическим преобразованиям?

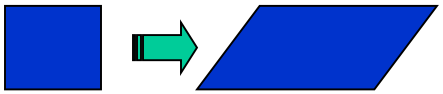
- Поворот



- Масштаб

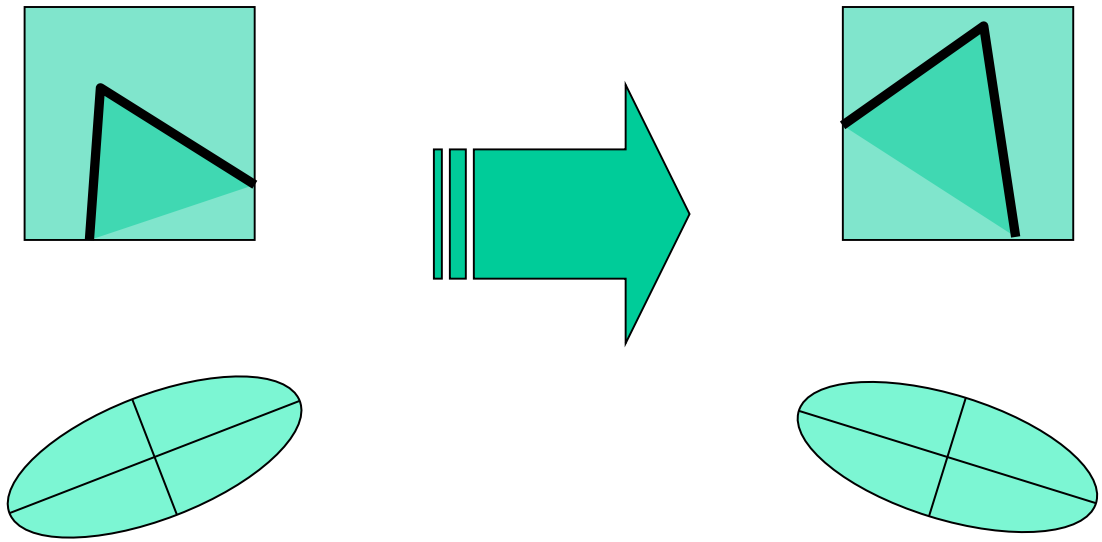


- Аффинное



Детектор Харриса

Инвариантность к вращению изображения:



$$M = R^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}R$$

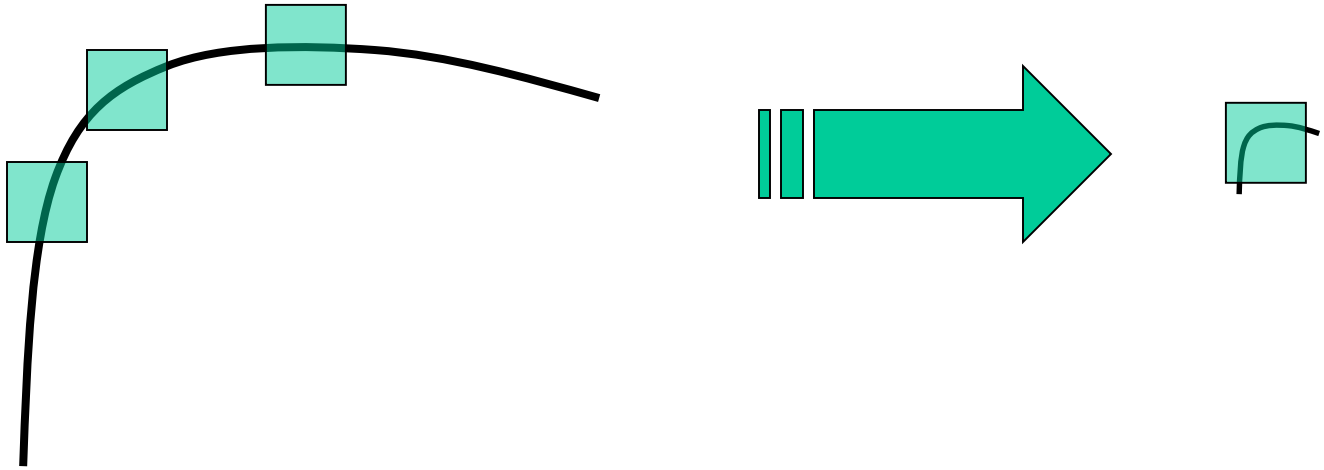
Эллипс вращается, но его форма (собственные значения) остаются неизменными

Отклик угла R инвариантен относительно вращению изображения

Slide by S. Lazebnik

Масштабирование?

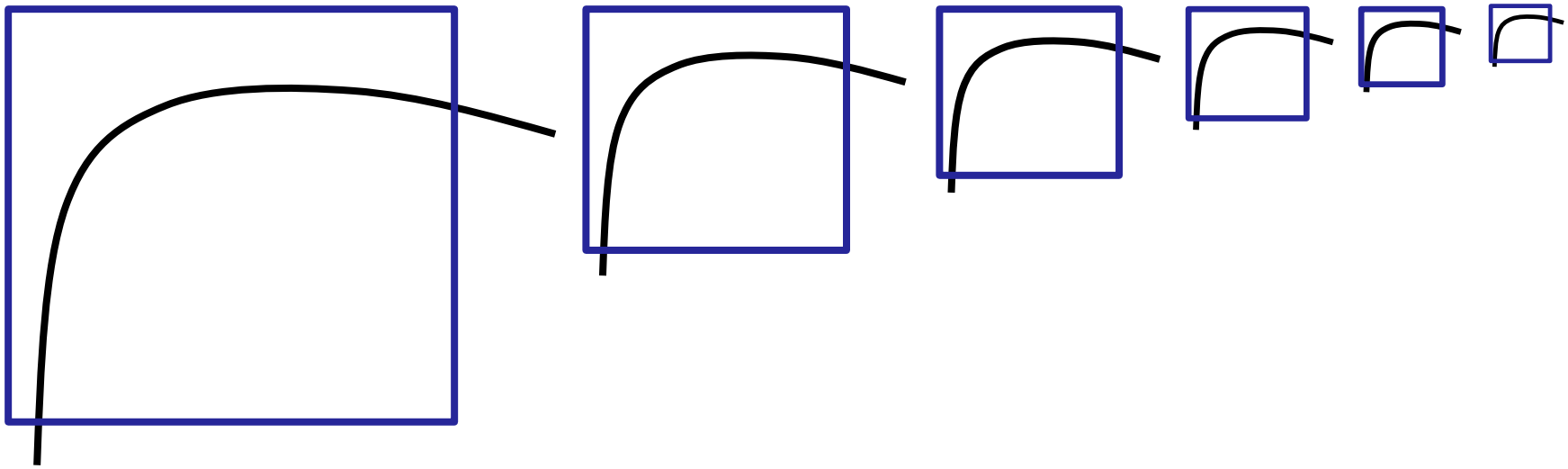
- Угол или нет? - Зависит от масштаба изображения!



Все эти точки будут помечены как ***края***

Угол !

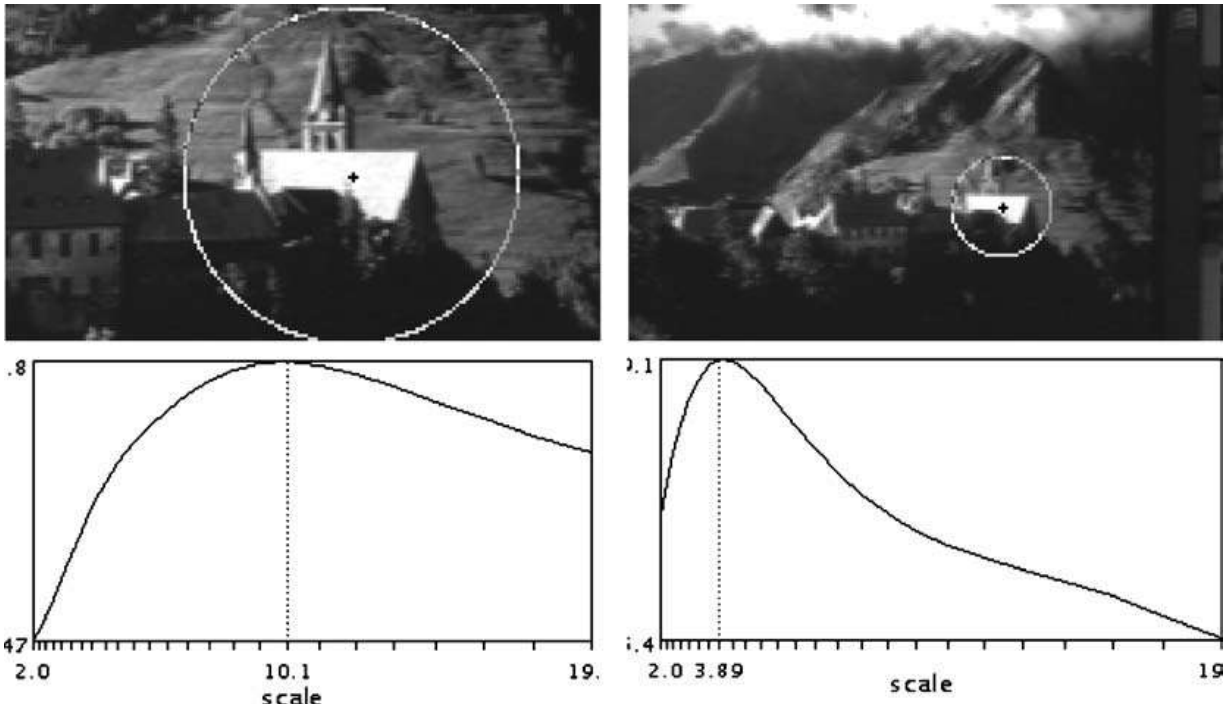
Характерный масштаб



- С какого момента фрагмент считается «углом»?
- Нам хочется находить точечные особенности одновременно с характерным масштабом (размером окрестности, при которой точку можно считать особенностью)

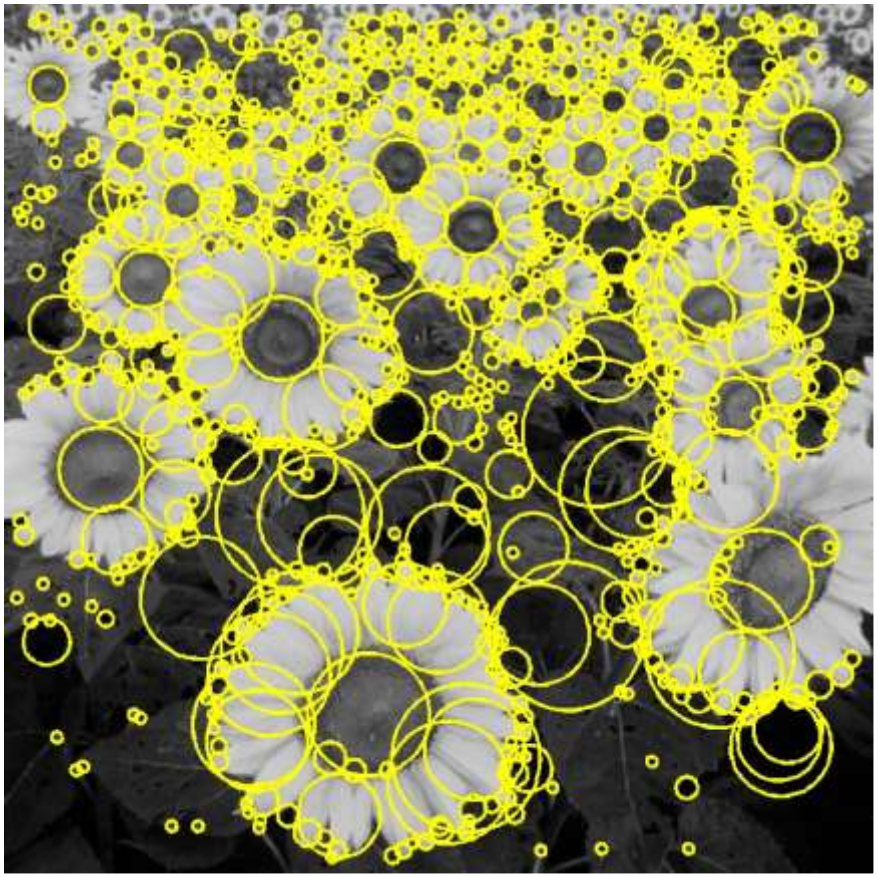
Инвариантность к масштабированию

- Цель: определять размер окрестности особой точки в масштабированных версиях одного и того же изображения
- Требуется метод выбора размера характеристической окрестности



Slide by S. Lazebnik

Блобы

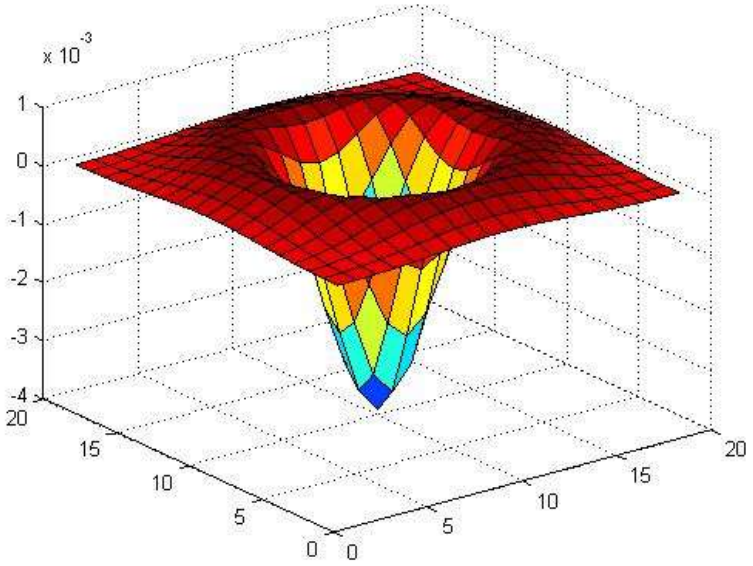


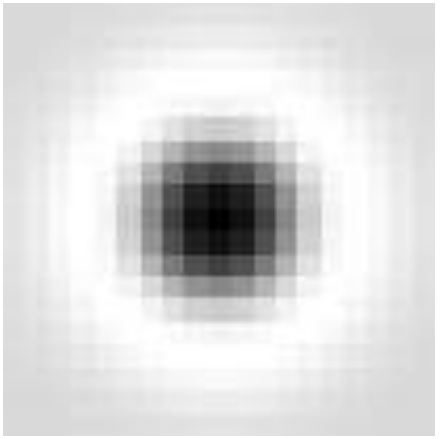
«Пятна», «Капля», «Blob» - вначале для особенностей такого типа была разработана теория выбора характерного размера

Slide by S. Lazebnik

Поиск блобов в 2D

Лапласиан Гауссианы: Центрально-симметричный оператор поиска блобов в 2D



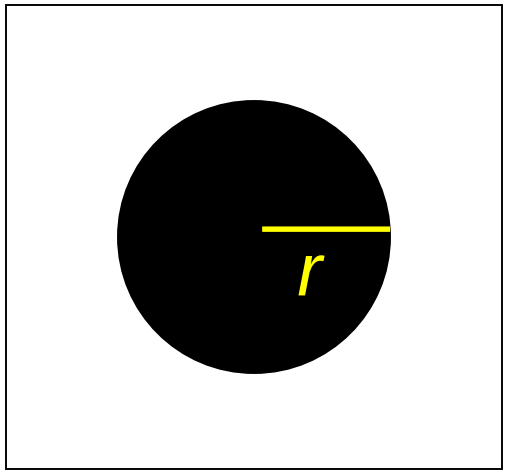


Нормализация: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

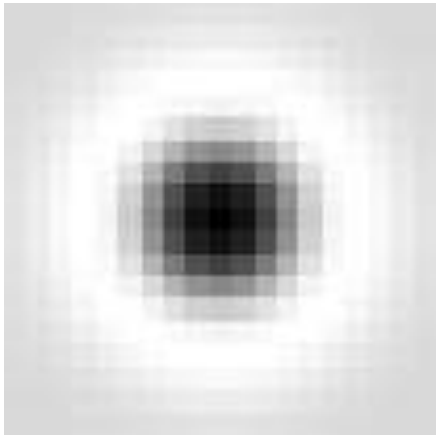
Slide by S. Lazebnik

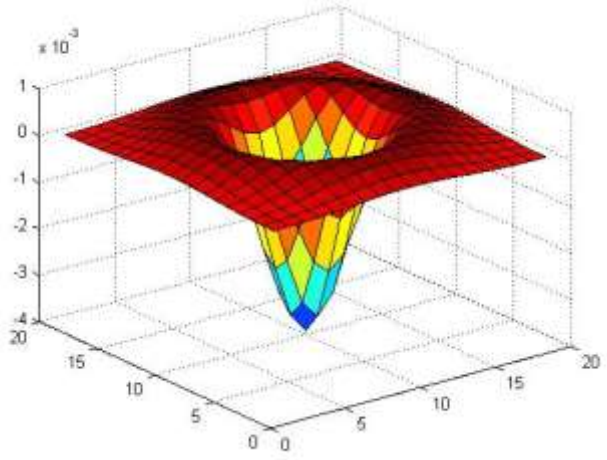
Выбор масштаба

- На каком масштабе Лапласиан достигает максимума отклика на бинарный круг радиуса r?



изображение





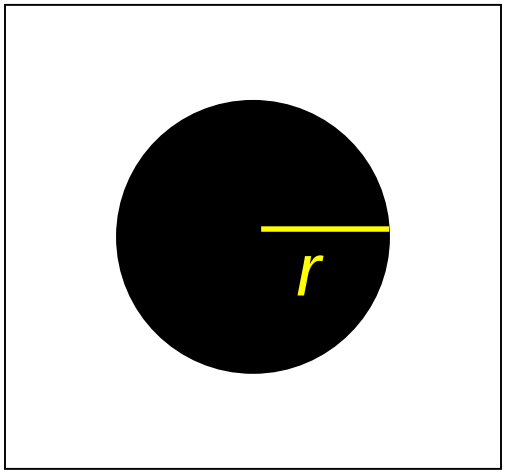
Лапласиан

Выбор масштаба

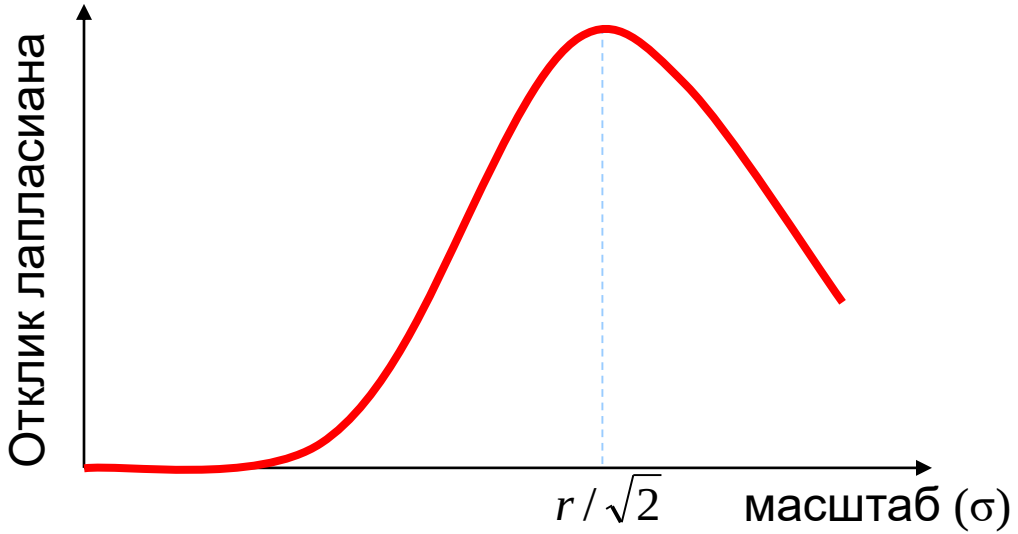
- 2D Лапласиан задается формулой:

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$ (с точностью до масштаба)

- Для бинарного круга радиуса r, Лапласиан достигает максимума в $\sigma = r/\sqrt{2}$



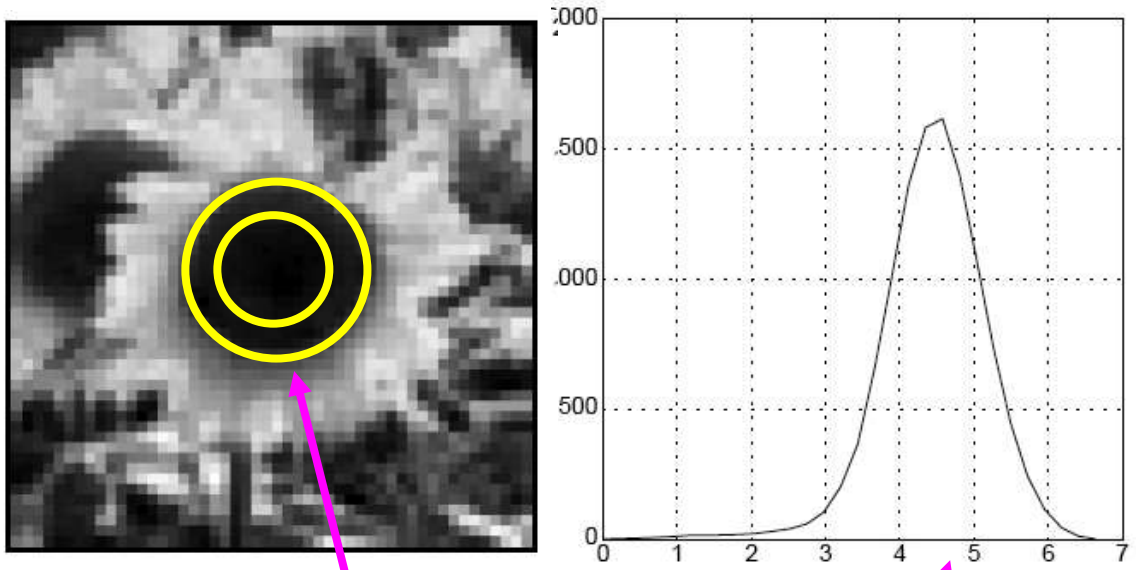
изображение



Slide by S. Lazebnik

Характеристический размер

- Характеристический размер определяется как масштаб, на котором достигается максимум отклика Лапласиана

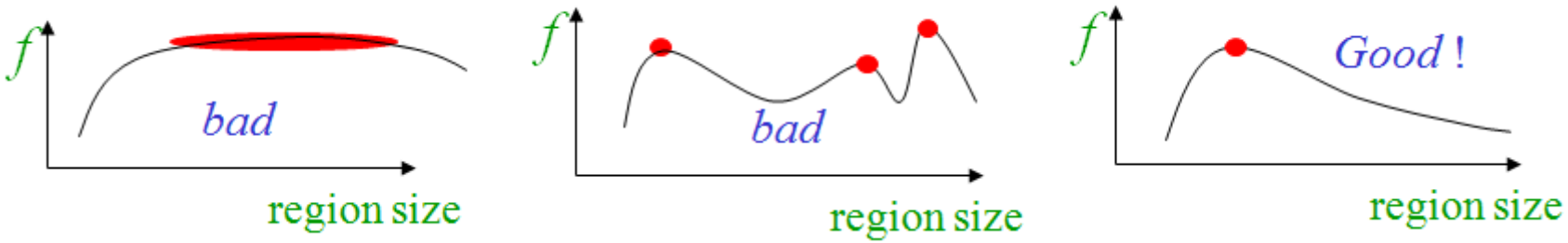


Характеристический масштаб

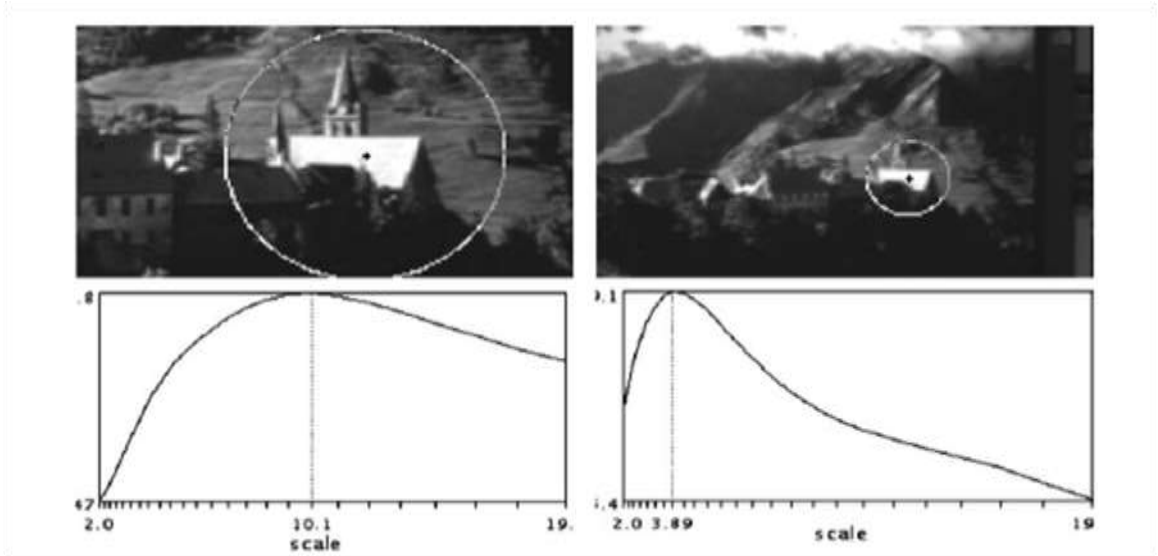
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

Slide by S. Lazebnik

Характеристический размер



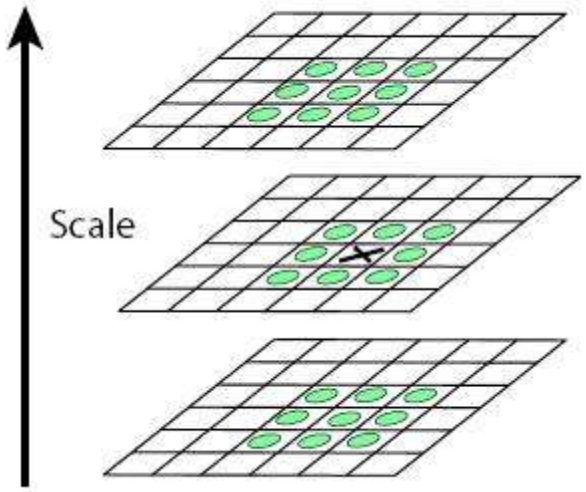
У «хорошего блоба»– один ярко выраженный пик функции



Slide by S. Lazebnik

Многомасштабный детектор блобов

1. Свертываем изображение нормализованным фильтром Лапласианом на разных масштабах
2. Ищем максимум отклика Лапласиана в 3D



Slide by S. Lazebnik

Пример

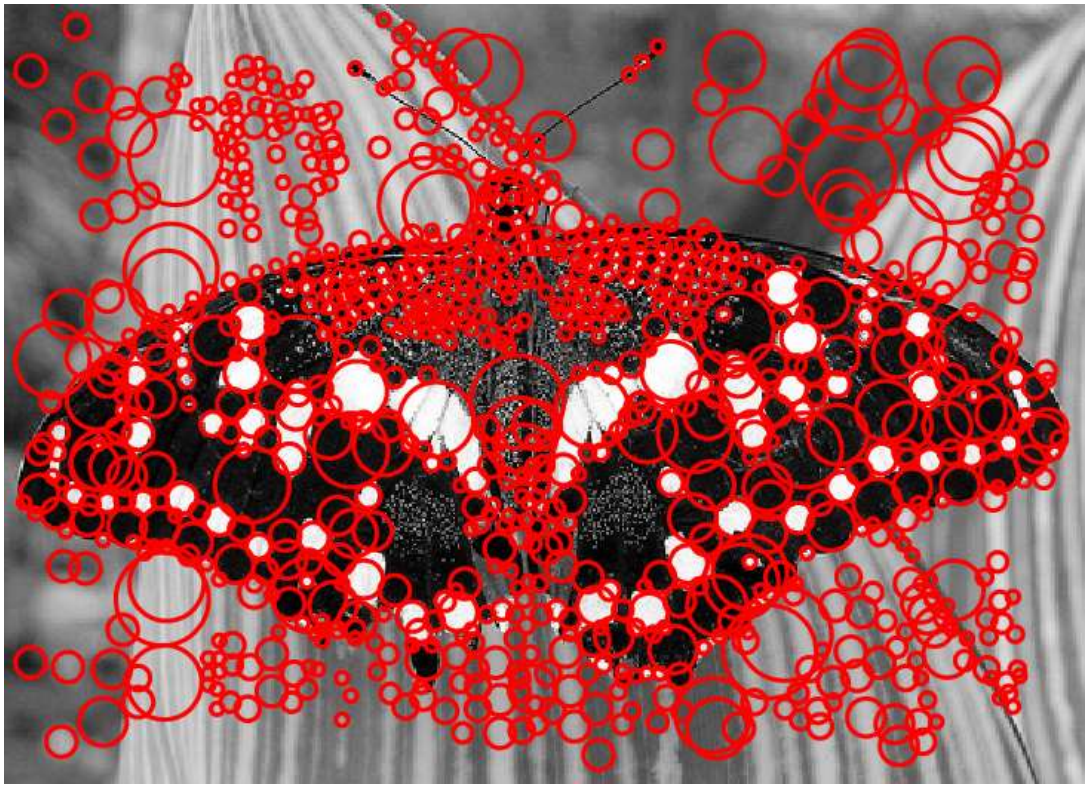
Slide by S. Lazebnik

Пример

sigma = 11.9912

Slide by S. Lazebnik

Пример



Slide by S. Lazebnik

Эффективная реализация (DoG)

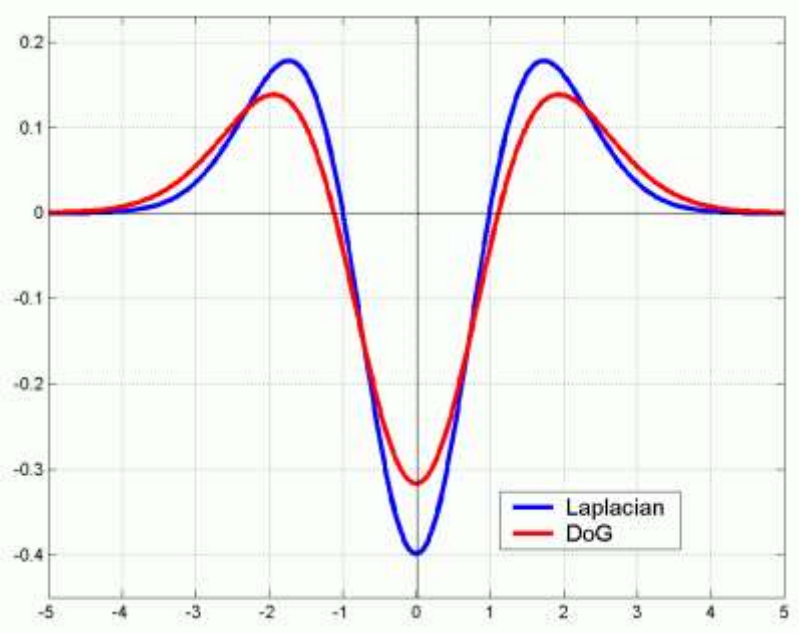
Приближение Лапласиана с помощью разницы гауссиан:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Лапласиан)

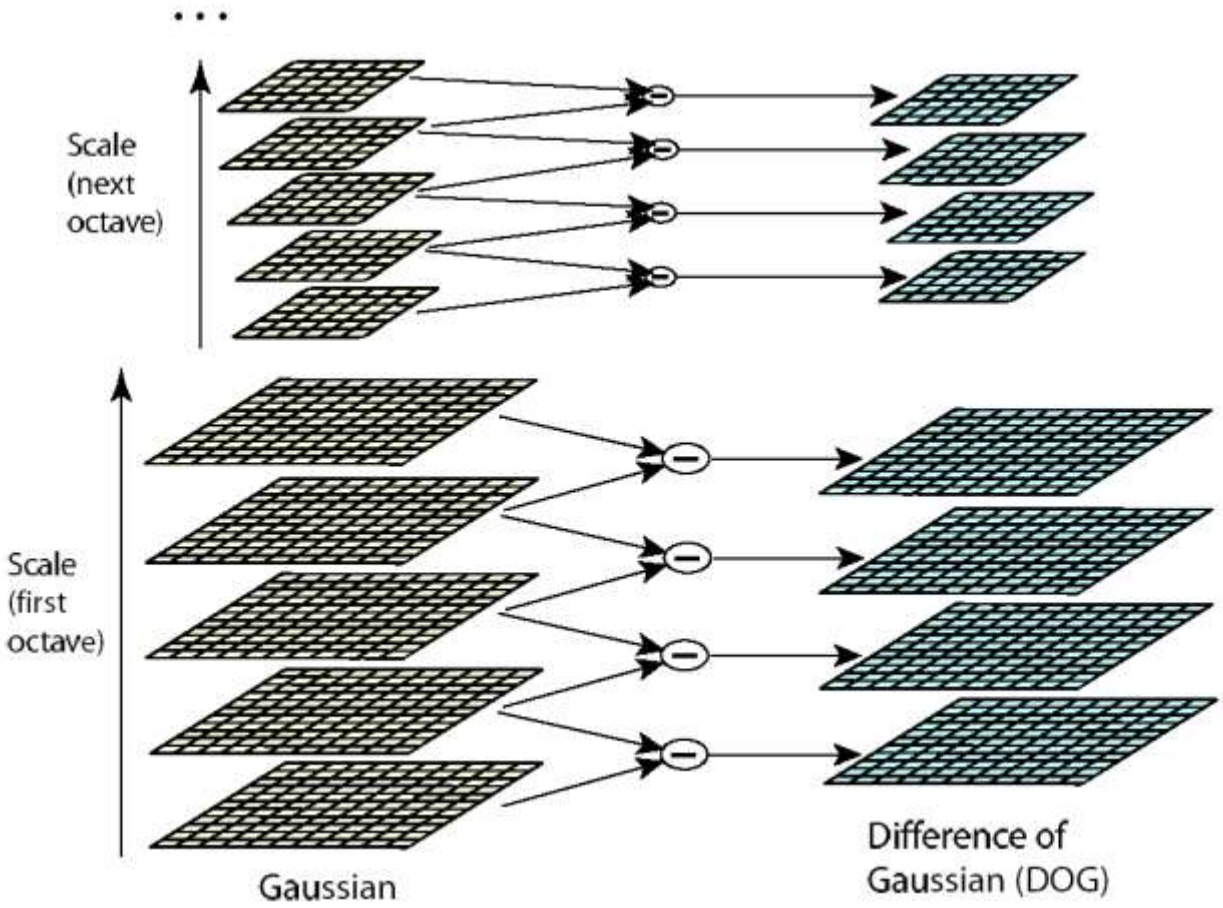
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Разница Гауссиан)



Difference of Gaussian = DoG

Эффективная реализация (DoG)



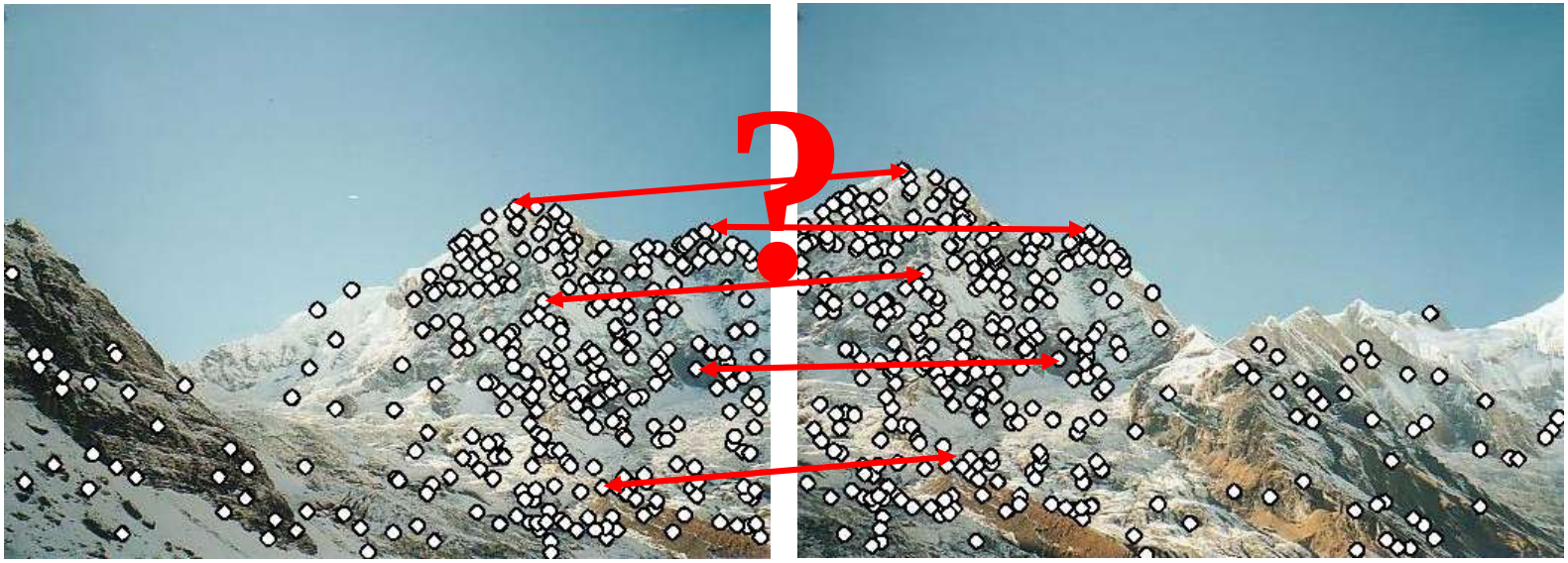
Детектор DoG также выделяет «блобы» на изображении

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Slide by S. Lazebnik

Дескрипторы

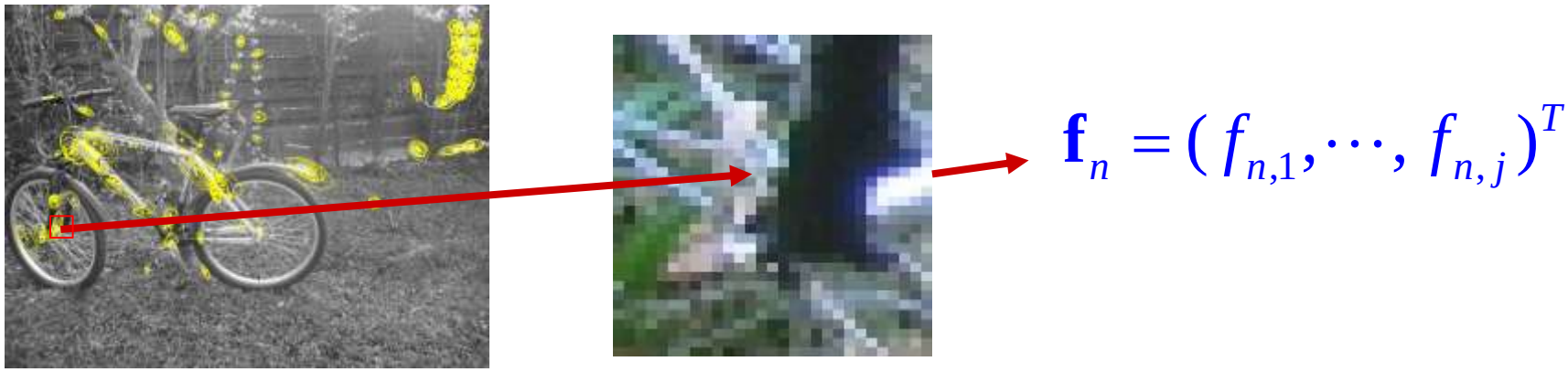
Точки найдены – как их сопоставить?



- Нужно как-то описать каждую точку, чтобы можно было отличать одну от другой!
- Дескриптор (Descriptor) - вектор признак окрестности точки

Дескрипторы

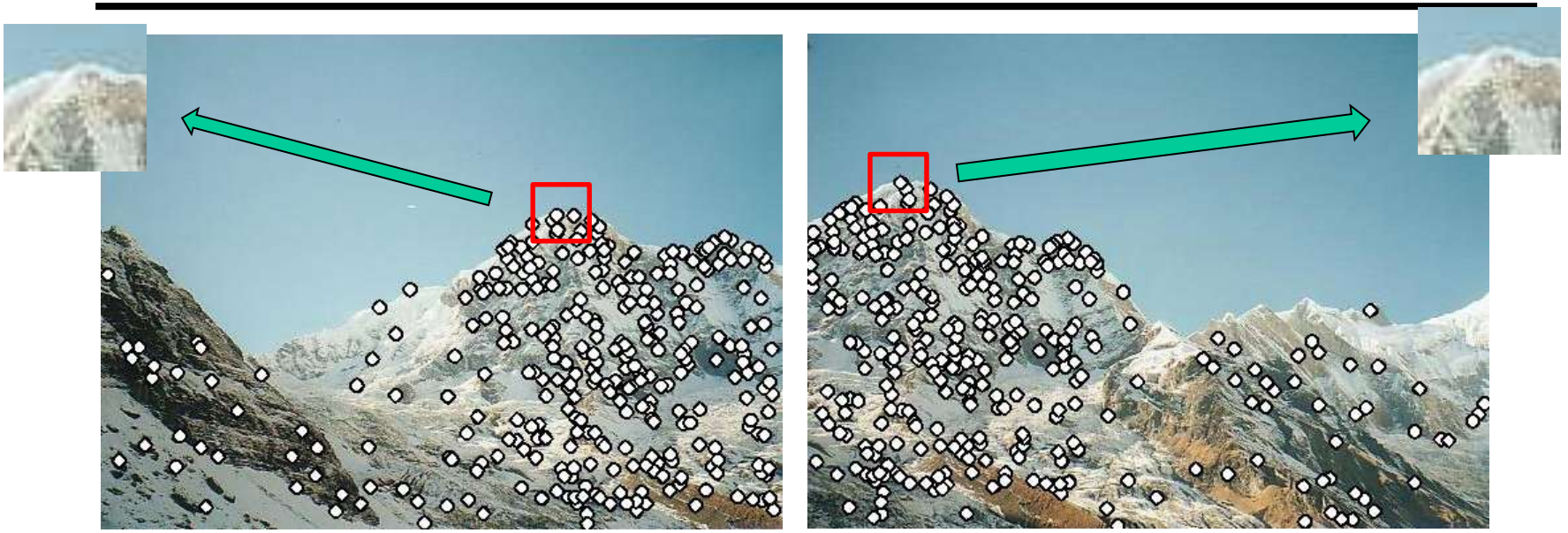
Необходимо каждую интересную точку описать набором параметров:



Как будем поступать:

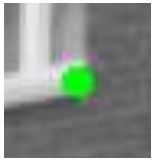
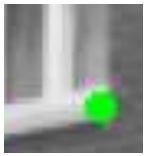
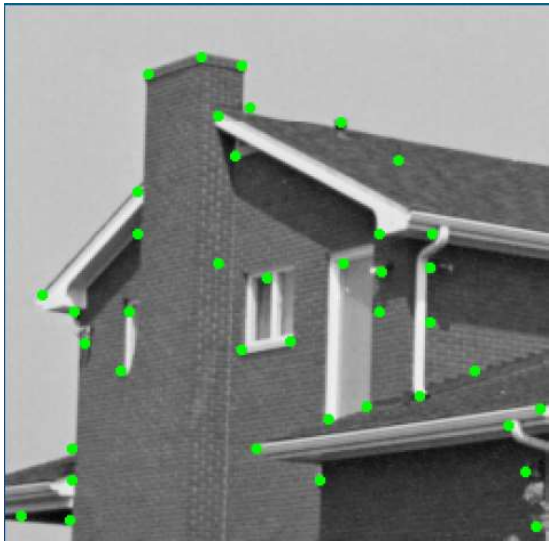
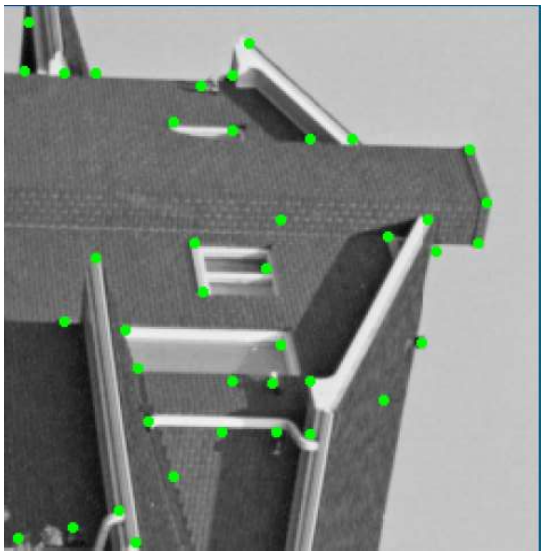
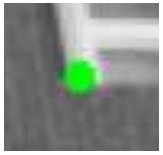
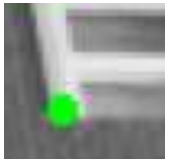
- Возьмём окрестность точки
  - Какой формы?
  - Какого размера?
- Вычислим по окрестности набор признаков
  - Какие?

Простейший подход



- Возьмём квадратные окрестности, со сторонами, параллельными строкам и столбцами изображения
- Яркости пикселов будут признаками
- Сравнивать будем как два изображения - попиксельно (SAD, SSD)
- Такая окрестность инвариантна только к сдвигу изображения

Недостаток простой окрестности

- Детектор точек инвариантен к повороту, а окрестность нет
- Небольшие сдвиги, т.е. ошибки в нахождении точки делают невозможным попиксельное сравнение

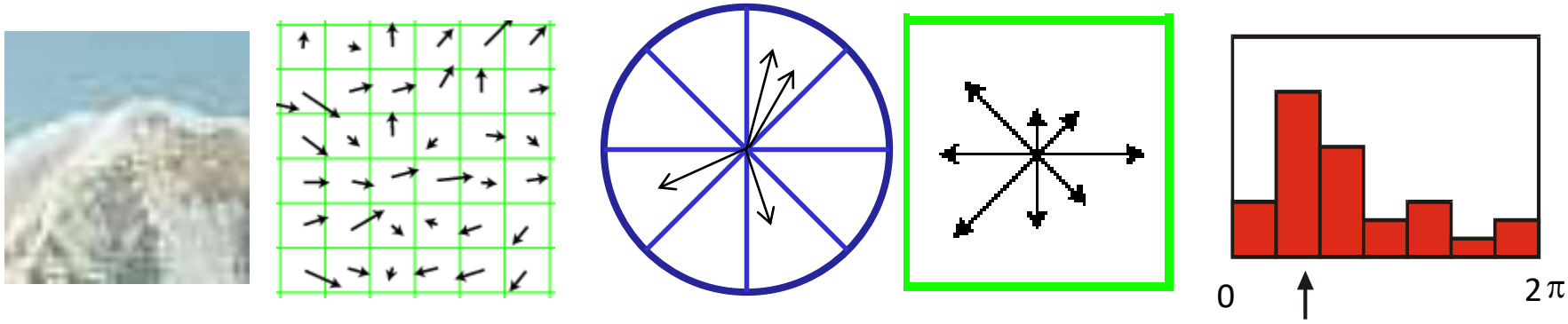
Метод SIFT

- Scale-Invariant Feature Transform:
  - Детектор DoG
    - Определение положения и характерного масштаба особенности
  - Ориентация
    - Определение доминантной ориентации особенности по градиентам
  - Дескриптор
    - Использование статистик по направлению градиентам

- Устойчив к изменениям освещенности и небольшим сдвигам

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

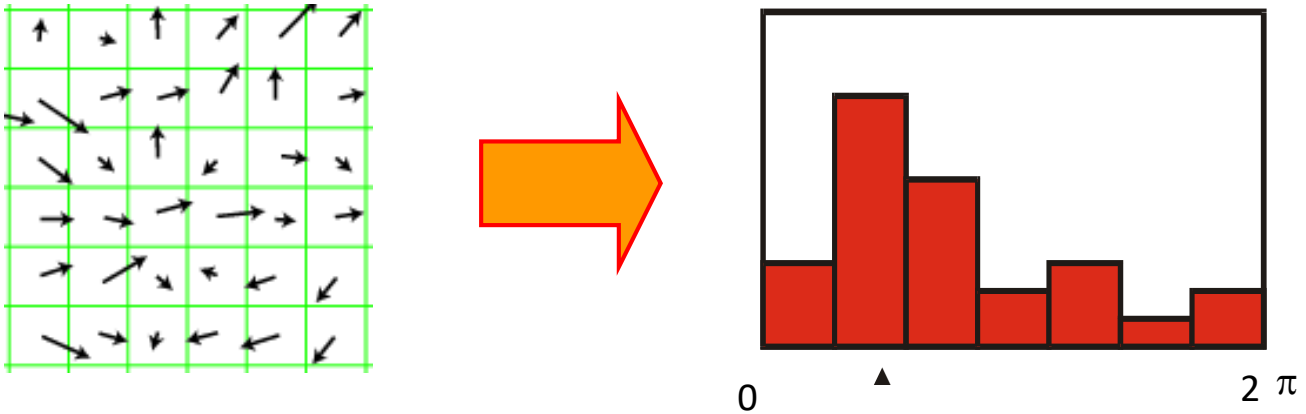
Гистограмма ориентаций градиентов



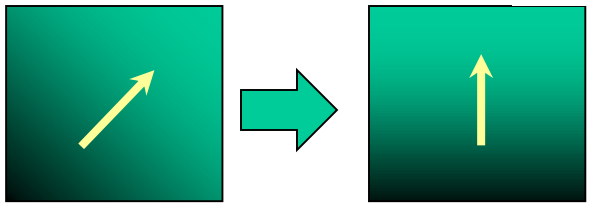
- Основа дескриптора SIFT – подсчёт гистограммы ориентаций градиентов
  - Вычислим направление градиента в каждом пикселе
  - Квантуем ориентации градиентов на 8 ячеек (направлений)
    - Пометим каждый пиксель номером ячейки
  - Посчитаем гистограмму направлений градиентов
    - Для каждой ячейки посчитаем количество пикселов с номером этой ячейки

Ориентация фрагмента

- Идея: найти основное (доминантное) направление градиентов пикселей в окрестности точки
- Выберем в гистограмме ячейку с максимальным значением, возьмём это направление как доминирующее



- Повернем фрагмент так, чтобы дом е направление градиента было направлен вверх

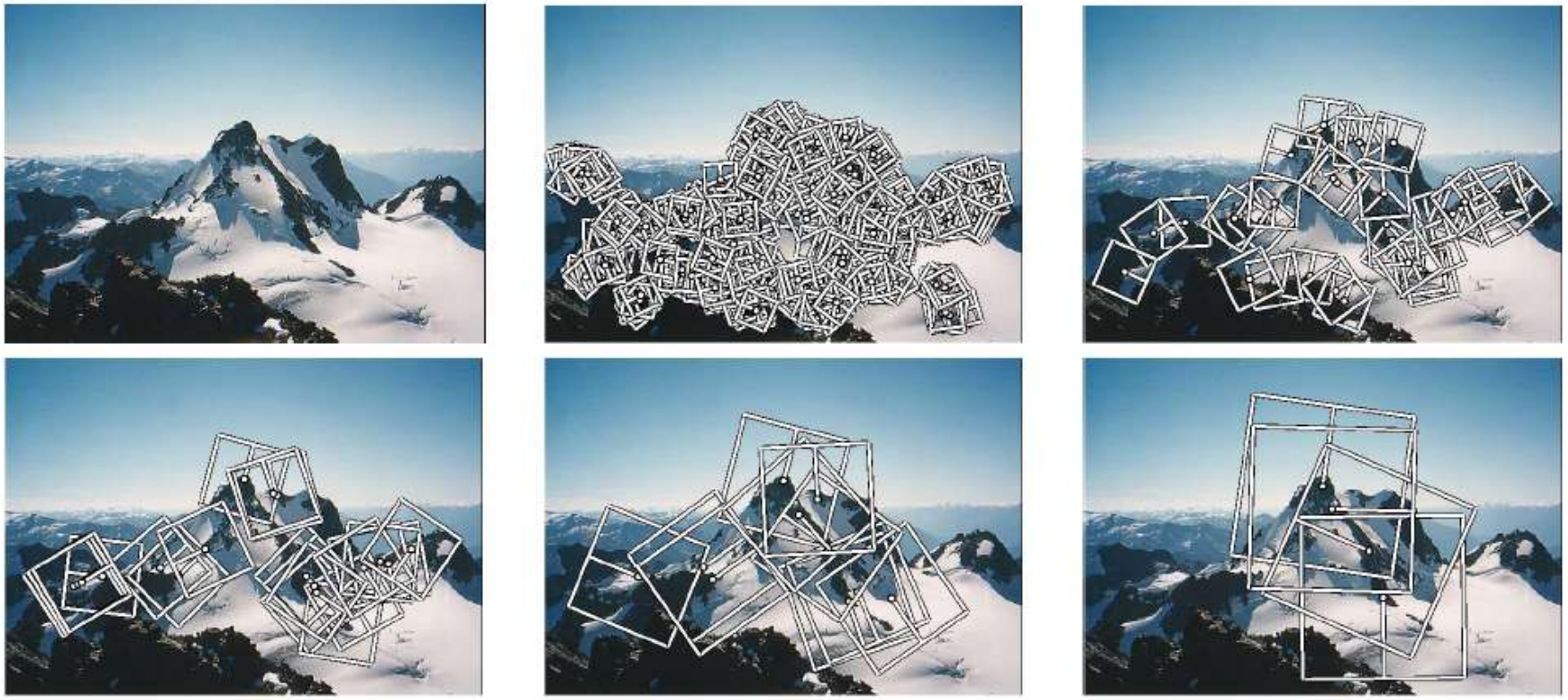


- Если локальных максимумов несколько – считаем, что несколько точек с разной ориентацией

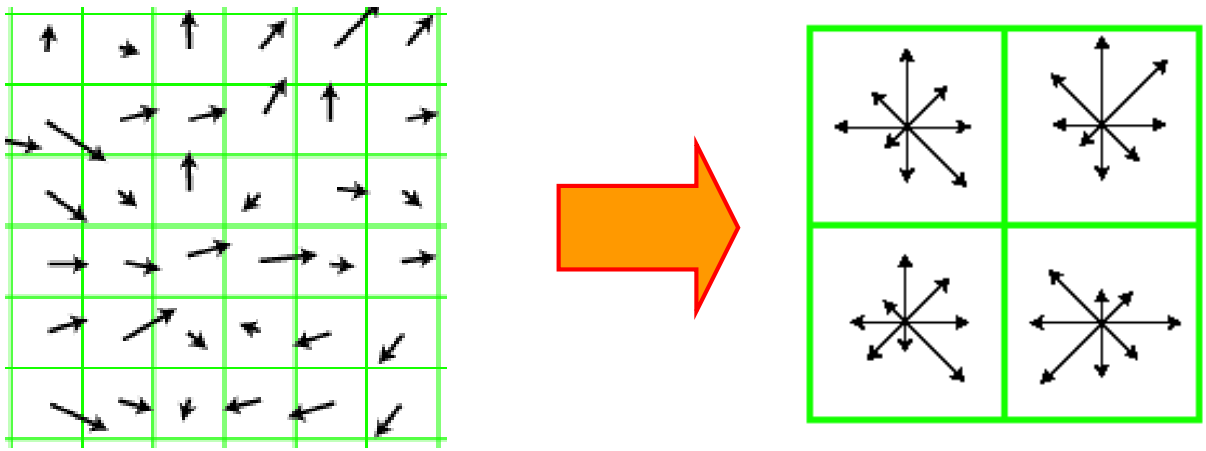
Окрестность особенности

- Для каждой найденной особенности теперь знаем характеристические масштаб и ориентацию
- Выберем соответствующую прямоугольную окрестность
  - (Rotation Invariant Frame)
- Приведем окрестность к стандартному размеру (масштабируем)

Пример локальных особенностей

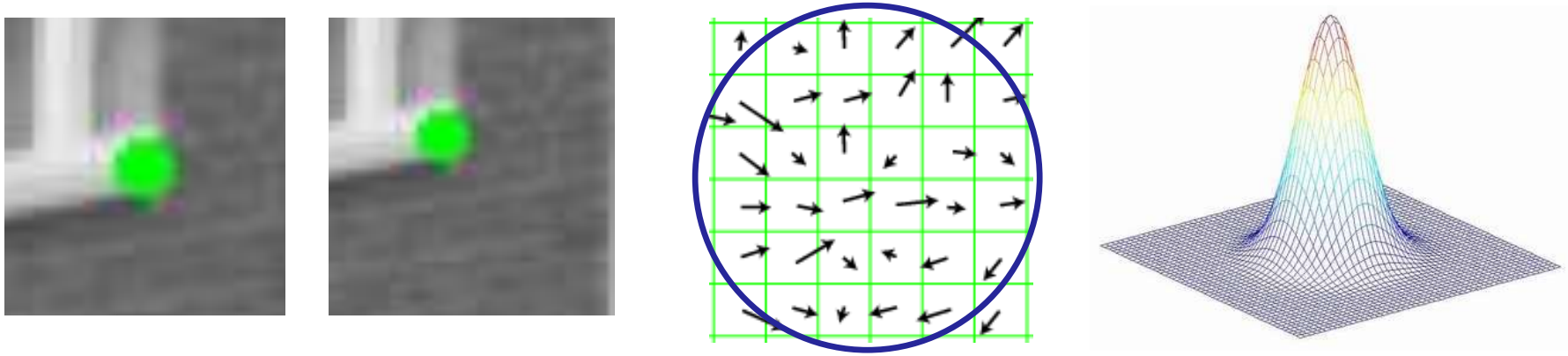


Построение дескриптора



- Для учета пространственного распределения свойств разделим окрестность на блоки сеткой, в каждом блоке посчитаем свою гистограмму градиентов
- Обычно – сетка 4x4, в каждой гистограмма с 8ю ячейками
- Стандартная длина вектора-дескриптора – 128 (4*4*8)
- Можем использовать обычную меру SSD для сравнения дескрипторов
- Можем использовать другие метрики, учитывающие, что дескриптор SIFT – это гистограмма

Устойчивость к сдвигам



- За счёт чего можно дескриптор сделать устойчивым к небольшим сдвигам?
- Использовать ядро при расчёте гистограммы
  - Взвешиваем вклад пикселей по ядрау
  - Веса рассчитываем в зависимости от близости к центру, по Гауссине
  - Небольшие изменения в локализации (положении, масштабе и ориентации) будут приводить к небольшим изменения дескриптора

Резюме SIFT

- Дескриптор SIFT весьма специфичен, устойчив к изменениям освещения, небольшим сдвигам
- Вся схема SIFT (детектор, выбор окрестностей, дескриптор) оказалась очень эффективным инструментов для анализа изображений
- Очень широко используется
- Запатентован
- Модель для многих других дескрипторов

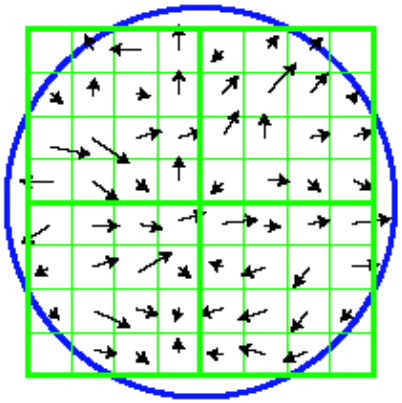
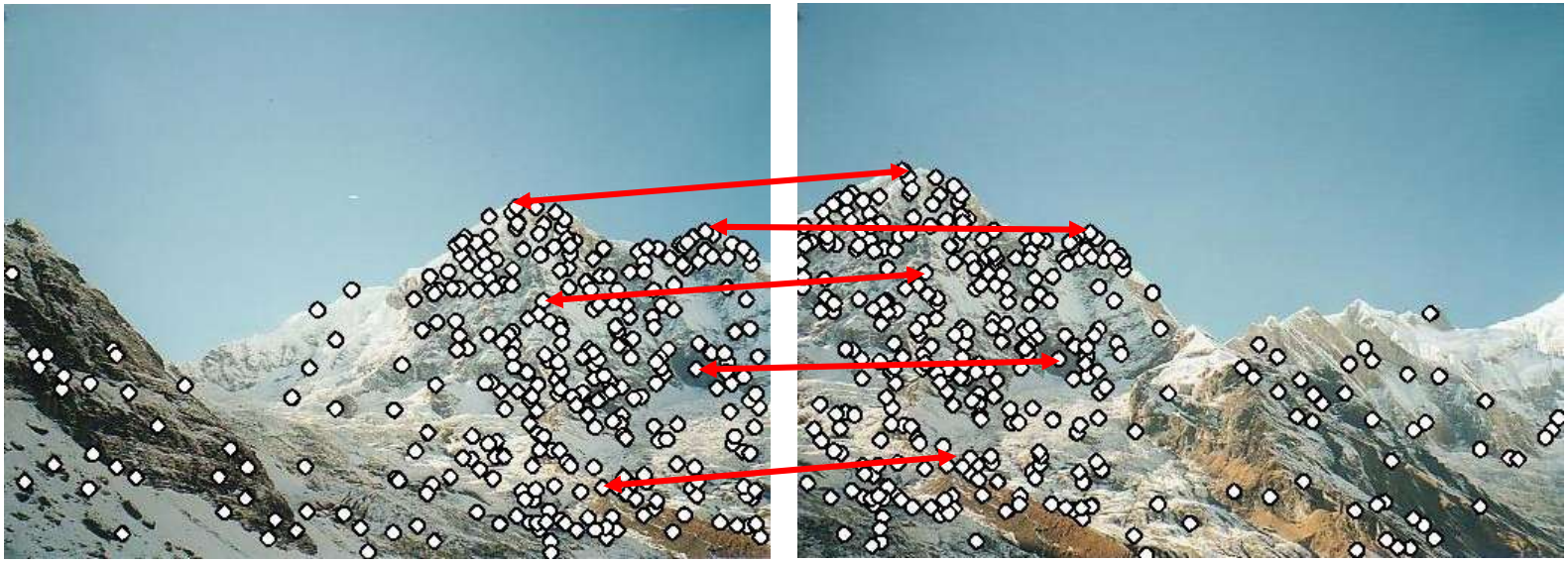


Image gradients

Расчёт преобразования

Есть набор соответствующих точек

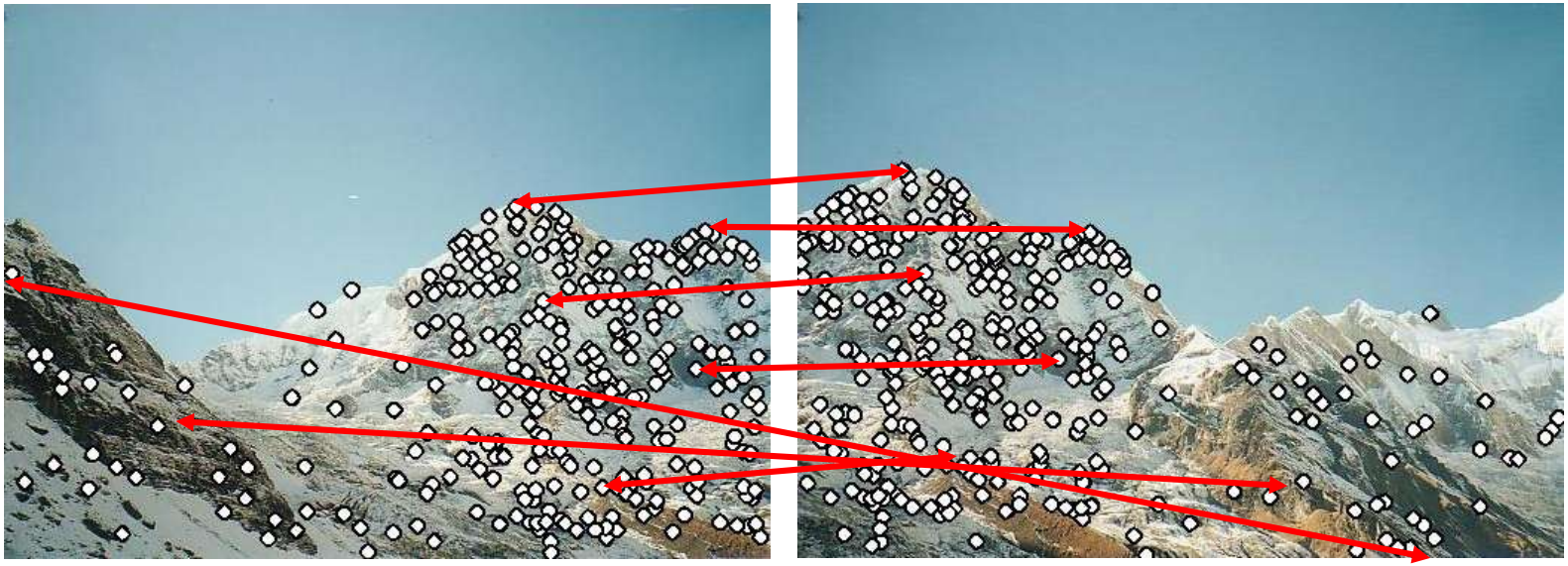


Как посчитать преобразование?

Пример - вычисление гомографии

- Пусть (x, x’) – известные соответствия на 2х изображениях
- Предположим, что среди них нет ложных соответствий
- Тогда подойдет линейный метод
- Составим систему линейных уравнений $\lambda x' = Hx$
  - Каждое соответствие даст 2 уравнения на параметры H
  - 4 соответствия дадут 8 уравнений на 8 неизвестных
  - Никакие 3 не должны лежать на одной прямой!
  - Можем взять и больше соответствий…
- Сведем к однородной задаче наименьших квадратов
- Из 4х соответствий (x,x’) никакие 3 не должны лежать на одной прямой

Робастная оценка



- Что делать, если есть ложные соответствия?
- Нужно применить «робастные» методы, устойчивые к ошибкам

RANSAC

- *Идея* – проведение оценки не по всем данным, а выборке, не содержащей выбросов
- Поскольку какие элементы данных – выбросы – заранее неизвестно, то…
  - …строим много выборок случайным образом!
- По каждой выборке строим гипотезу
- Среди всех гипотез выберем ту, которая лучше всего согласуется со всеми данными
- Random sample consensus (RANSAC)

M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

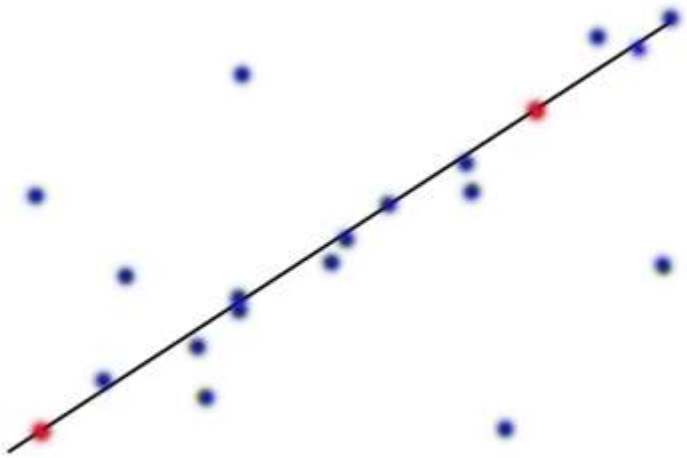
Выборка как основа схемы

• Основой RANSAC является оценка модели по небольшой выборке $S \subset x$

• Проблема - количество таких выборок огромно

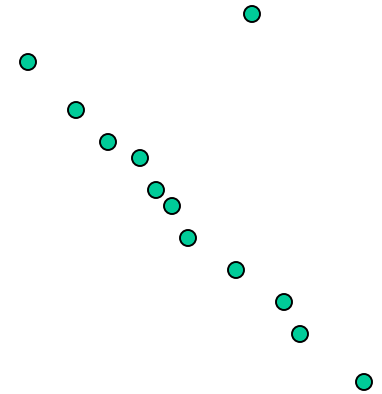
• Поэтому будем строить гипотезы по выборке минимального размера

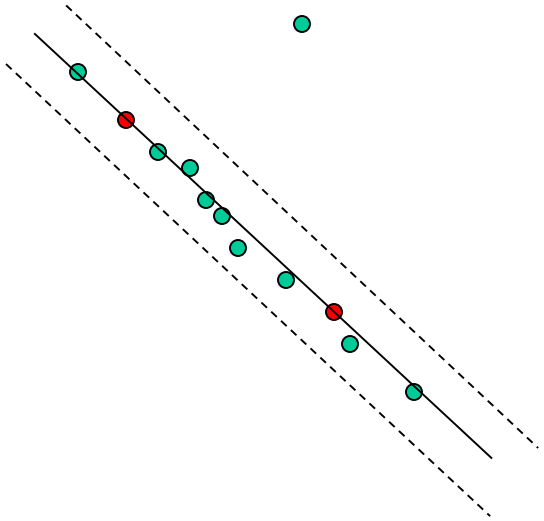
• Для прямой:



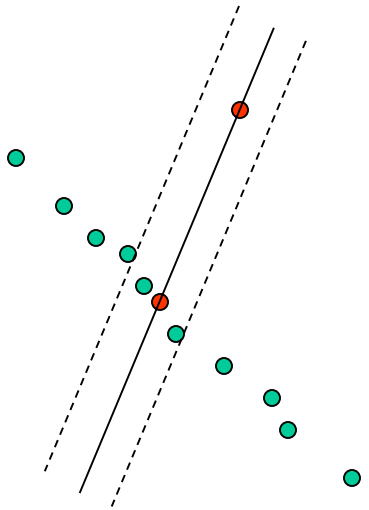
А для геометрических преобразований сколько?

Функция качества гипотезы в RANSAC

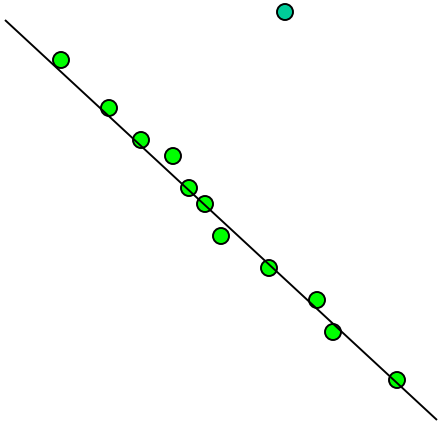




11 «голосов»



4 «голосов»



Базовая схема RANSAC

Повторяем N раз:

(i) Построение случайной выборки $S \subset X$

(ii) Построение гипотезы Θ по выборке S

(iii) Оценка качества гипотезы Θ по набору исходных данных X

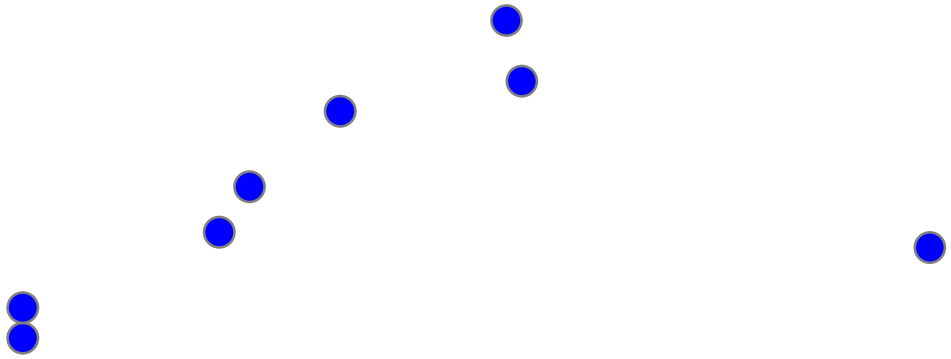
(iv) Если гипотеза Θ лучше предыдущих, запоминаем её

После завершения итераций:

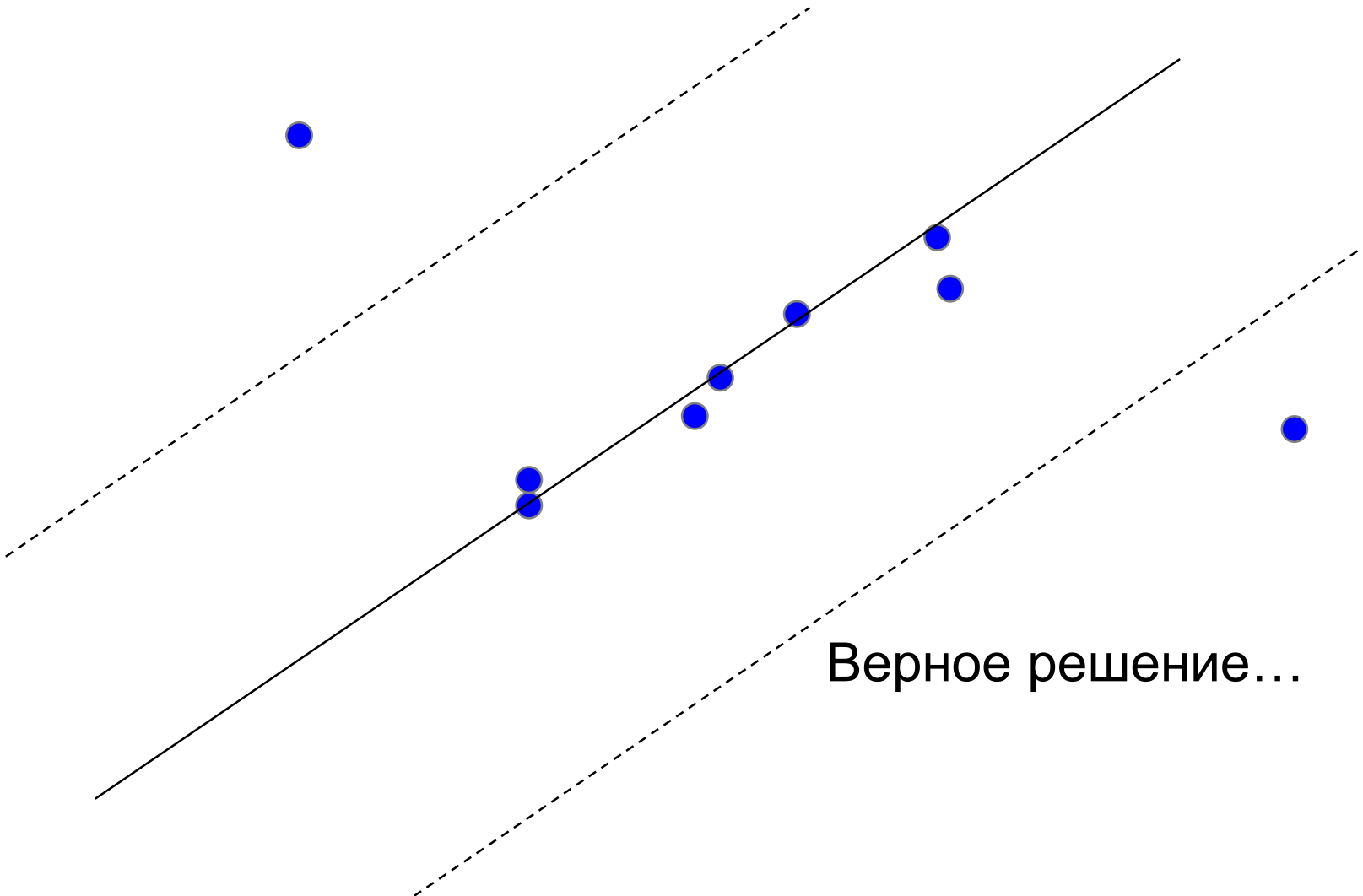
(i) Построение чистой выборки X’ путём фильтрации выбросов, не удовлетворяющих Θ_{best}

(ii) Уточнение гипотезы Θ_{best} по X’

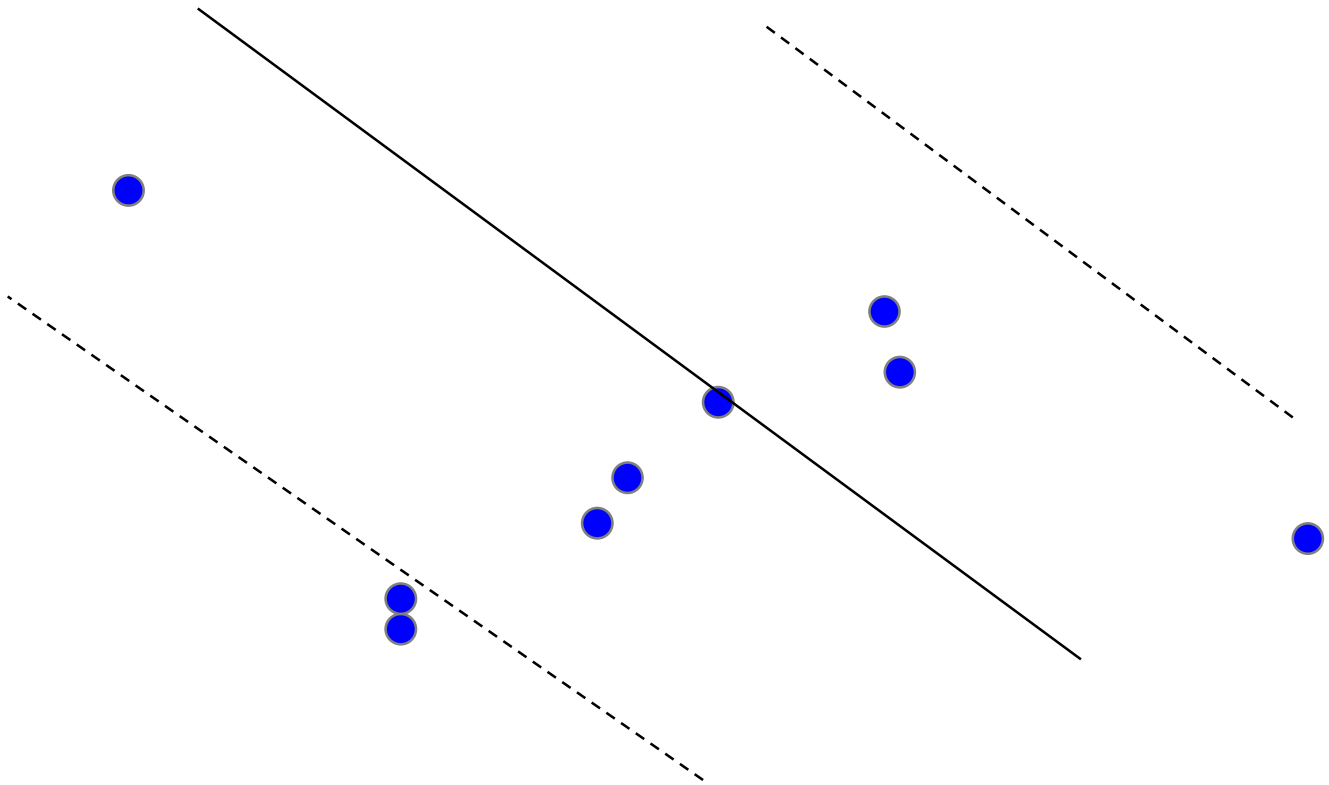
Большой порог – проблема RANSAC!



Большой порог – проблема!

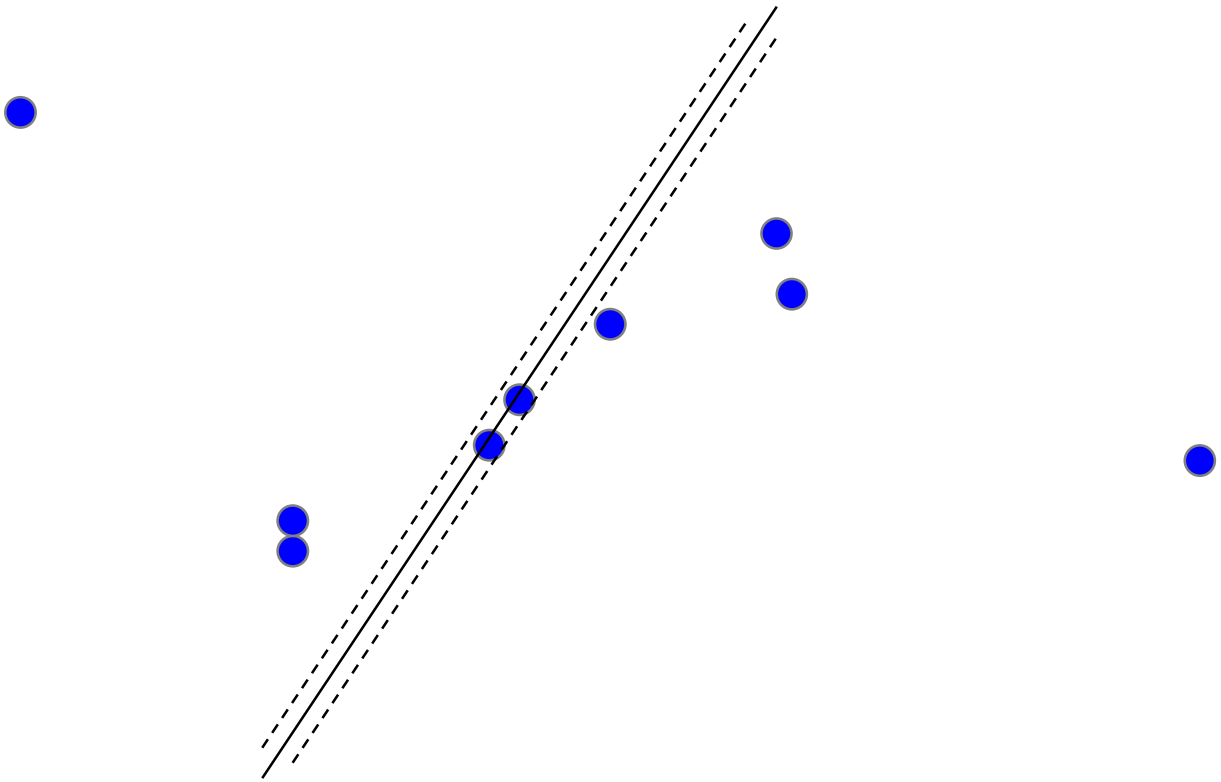


Большой порог – проблема!



…эквивалентно
неверному

Маленький порог – проблема!



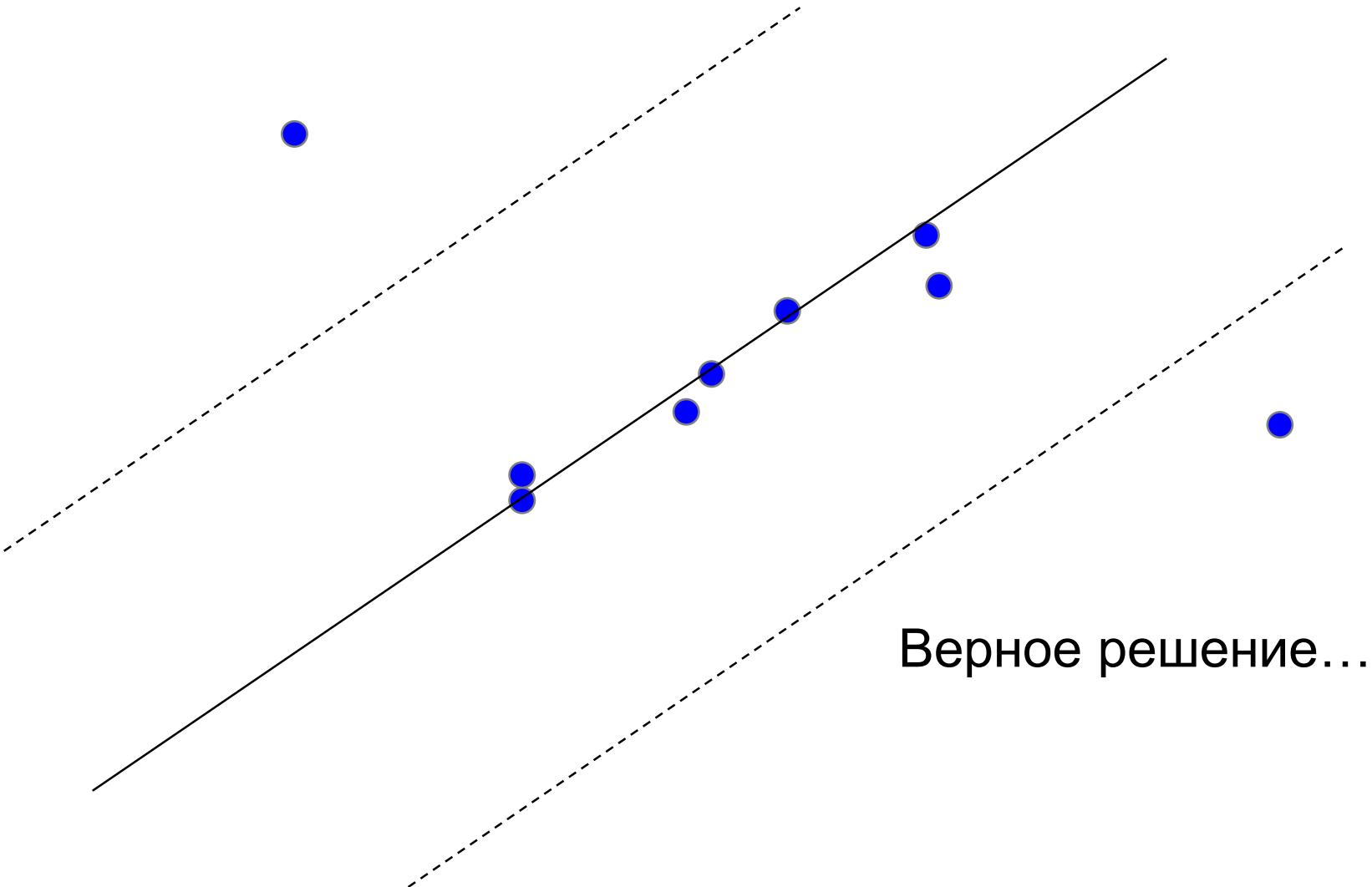
Функции качества

- M-SAC
  - Возьмём робастную функцию, называемую М-оценкой, в качестве целевой функции:

$$R(\theta) = \sum_i p(\varepsilon_i(\theta)^2), \;\; p(\varepsilon_i^2) = \begin{cases} \varepsilon_i^2 & \varepsilon_i^2 \le T^2 \\ T^2 & \varepsilon_i^2 > T^2 \end{cases}, i = \overline{1,n}$$

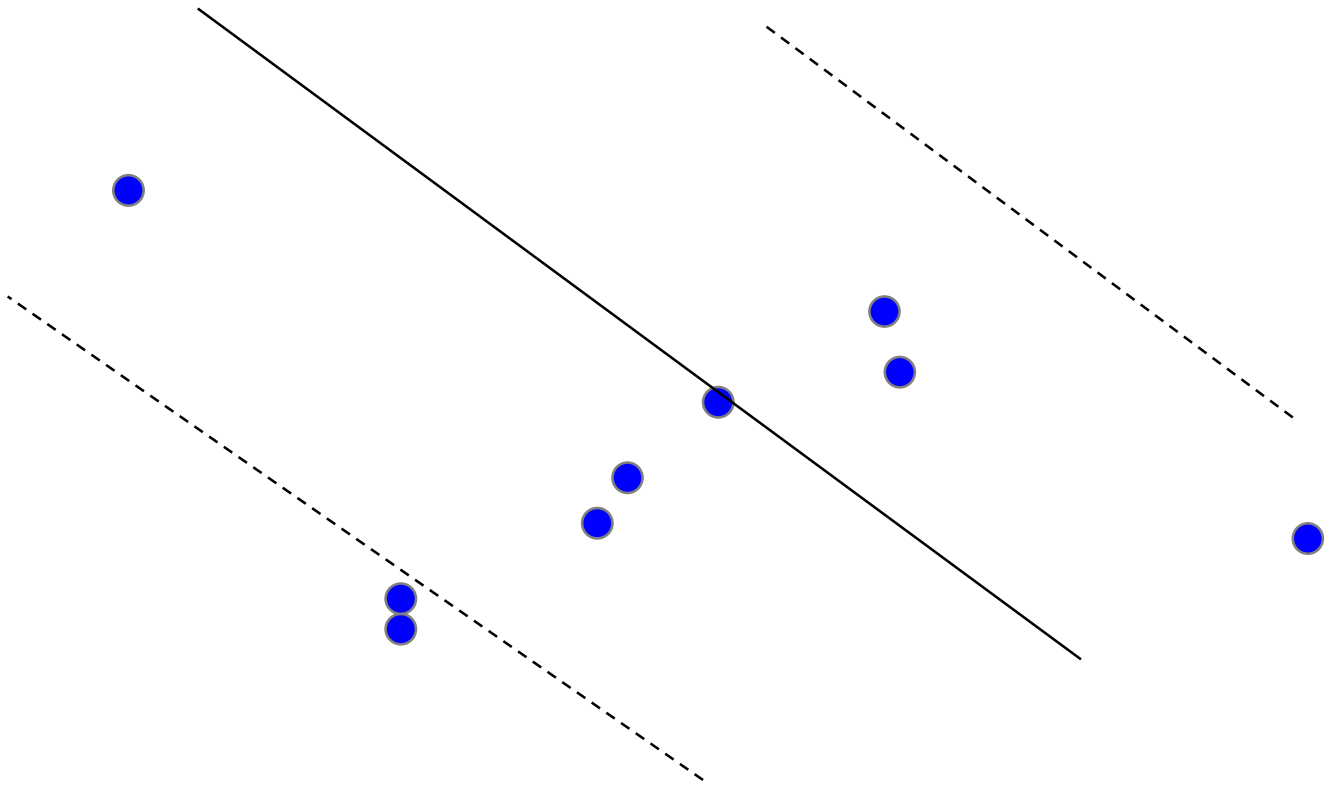
- M-SAC дает более точную оценку без увеличения вычислительной сложности

M-SAC



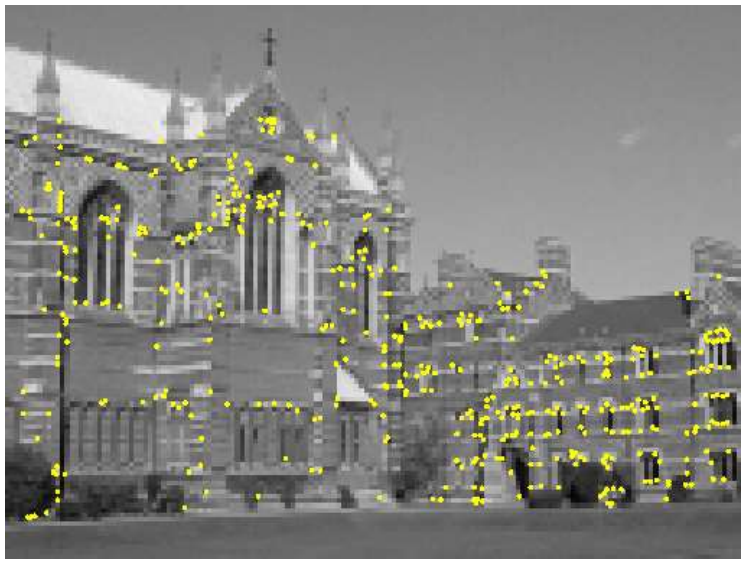
Верное решение…

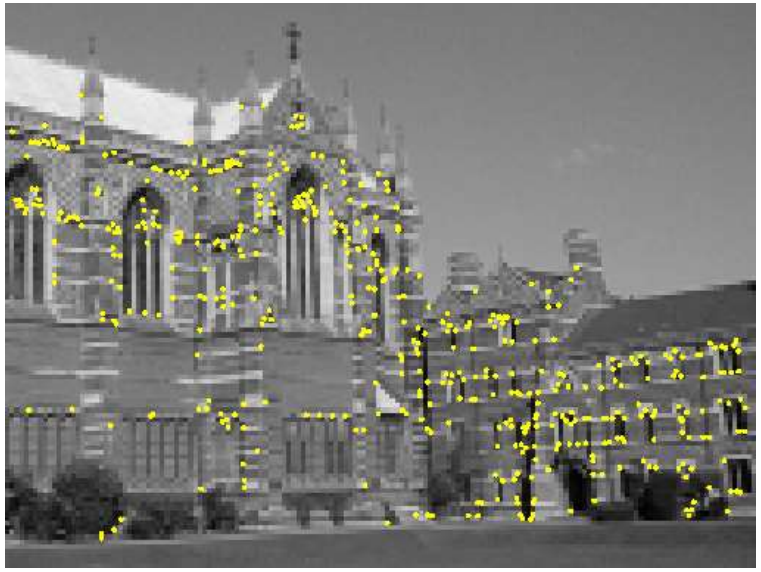
M-SAC



Лучше неверного

Применим для оценки гомографии

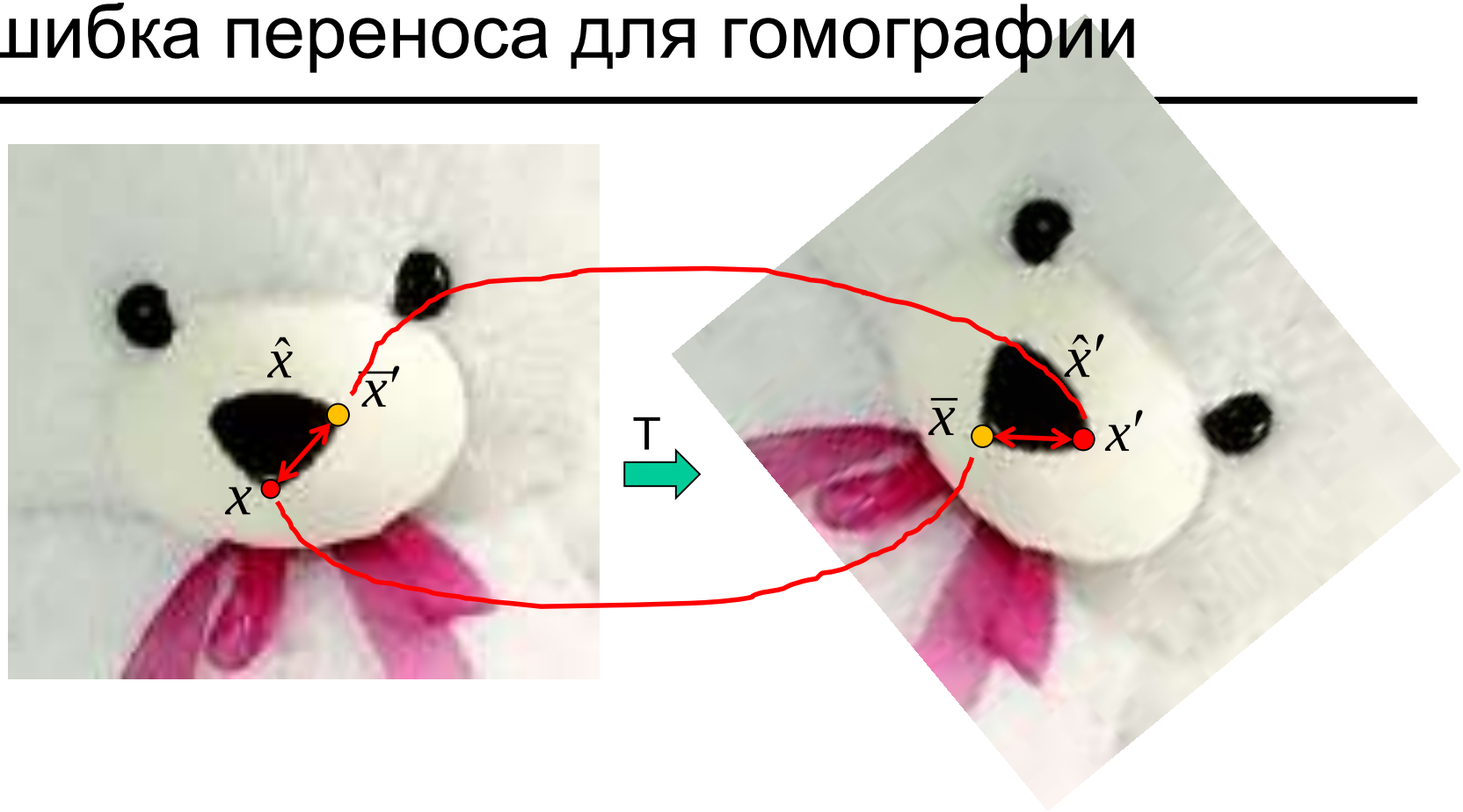




Умеем оценивать гомографию линейным методом

Нужно ввести функцию ошибки для оценки качества

Ошибка переноса для гомографии



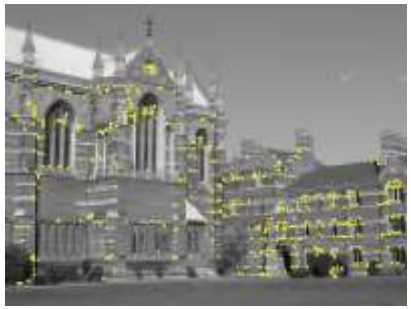
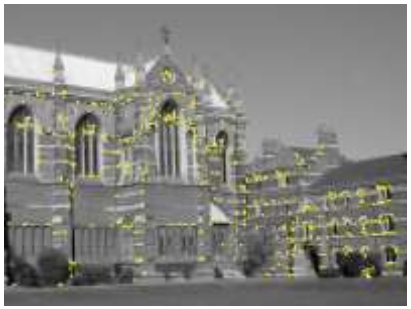
- Пусть (x,x') – пара соответствующих точек, H – гомография
- Тогда $\bar{x} = Hx$ и $\bar{x}' = H^{-1}x'$ - образы точек, полученных с помощью гомографии
- Ошибка переноса (transfer error):

$$\mathrm{r}(\mathrm{x},\mathrm{x}') = d^2(x'_i, \bar{x}_i) + d^2(x_i, \bar{x}'_i)$$

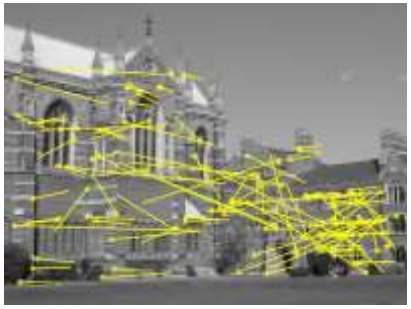
Алгоритм сопоставления

- Дано {(x,x')} – набор пар соответствующих точек на изображениях I и I'
- Вычислим модель H гомографии между изображениями I и I' по парам соответствующих точек с помощью RANSAC
- Отфильтруем выбросы в {(x,x')} по модели H
  - Если ошибка r(x,x') > T, тогда (x,x') – выброс (ложное соответствие)
- Уточним модель H
  - Метод наименьших квадратов по всем оставшимся точкам (inliers)
  - Итеративный пересчёт H и истинных/ложных соответствий
  - Нелинейная оптимизация по всем данным с M-функцией в качестве целевой

Примеры работы

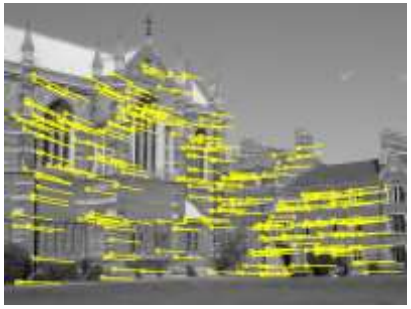



500 особенностей



Соответствия (268)

Выбросы (117)

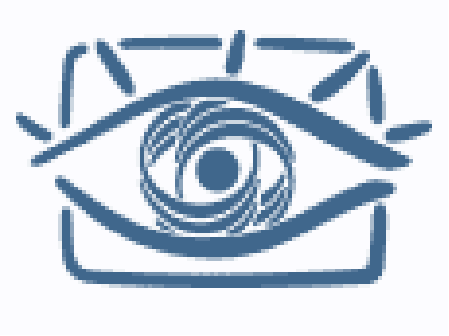


Хороших соответствий (151)

Пример работы для видео

Резюме сопоставления изображений

- Важная задача компьютерного зрения
- Использует модель преобразования изображений
- Два основных подхода поиска параметров преобразования - прямое сопоставление и на основе локальных особенностей
- Для оценки параметров преобразования по контрольным точкам можем пользоваться линейными и робастными методами (такими как RANSAC)



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №4 «Классификация изображений и поиск похожих»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

30 сентября 2019 года

Классификация изображений

Бинарная классификация

- Есть ли на этом изображении пешеход?
- Бинарный ответ

$$y \in [0,1], 1 - \text{да}, 0 - \text{нет}$$

Car

Многоклассовая классификация

- Какой из заданных s классов присутствует на данном изображении?
- Эталонный ответ - метка класса

  $y \in [1, S]$
- Обычно ответ

  $y_i \in [0,1], i = [1, S] \quad , \sum y_i = 1$

Атрибуты объектов

Мужчина

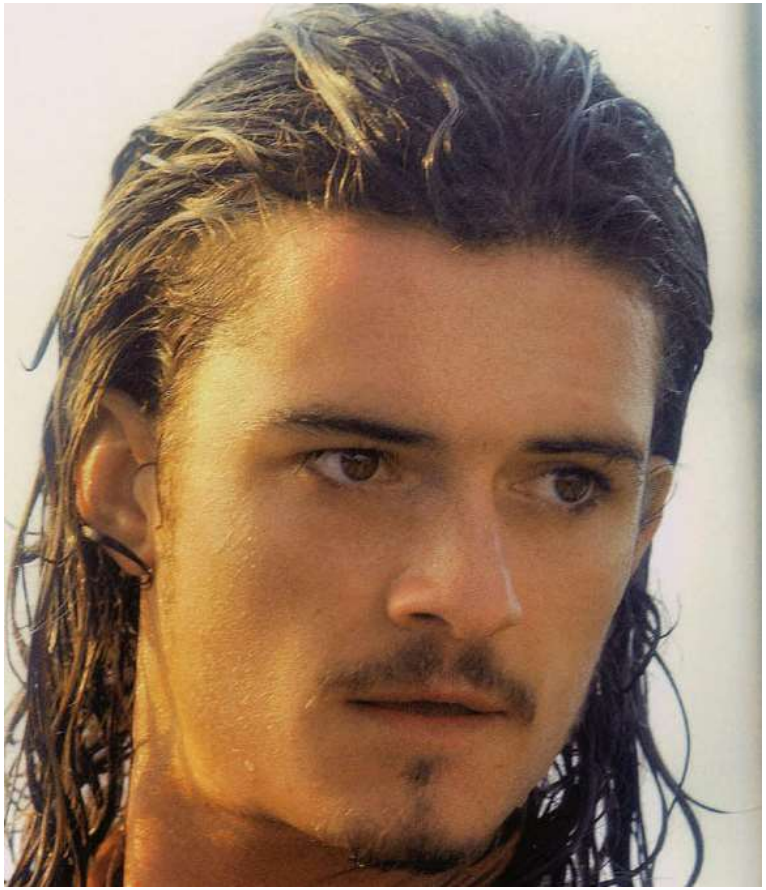
Азиат

Бородат

Улыбается

- Атрибуты – «типичные» характеристики объекта
- Для человека - пол, возраст, раса, борода, усы, улыбка, очки и т.д.
- Можем свести к задачам классификации
  - Определение пола – бинарная классификация
  - Определение расы – многоклассовая классификация
  - Определение возраста – либо классификация (возрастные группы), либо регрессия (определение числового параметра)

Распознавание по лицу - верификация



На обоих изображениях один и тот же человек, или нет?

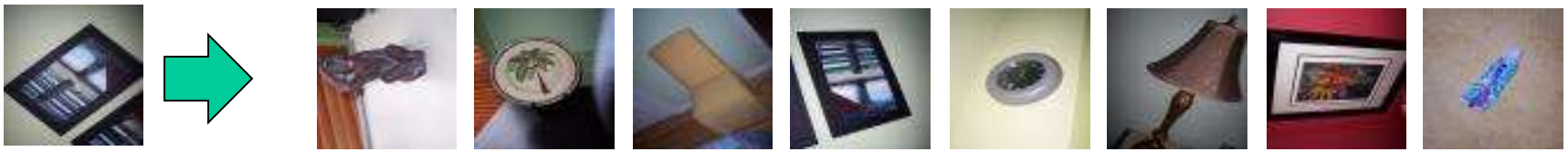
Базовая задача распознавания лиц, которую и человеку проще всего решать

Распознавание человека по лицу

- «Watch list»
- Есть список людей с фотографиями
- Необходимо определить, входит ли человек в этот список по его фотографии
- «One-shot classification» - на каждый класс у нас одно изображение-пример

Поиск похожих изображений

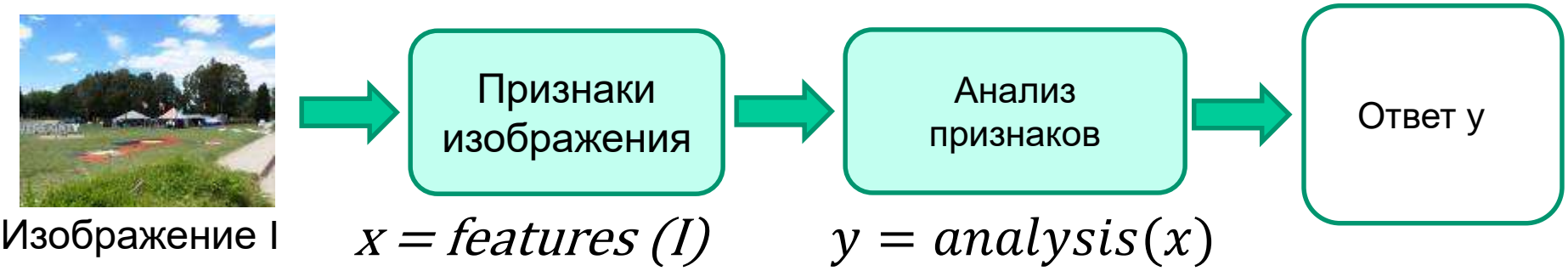
- Или «поиск изображений по содержанию» (Content-based image retrieval)
- Есть коллекция изображений. Мы формулируем запрос в виде изображения-примера («найди то же самое»)



- Поиск по «визуальному сходству» изображения в целом или объектов в изображениях
- Чаще всего – нужно найти тот же самый объект, и другие похожие на него
- Классификация с открытым и заранее неизвестным набором классов

Классификация изображений

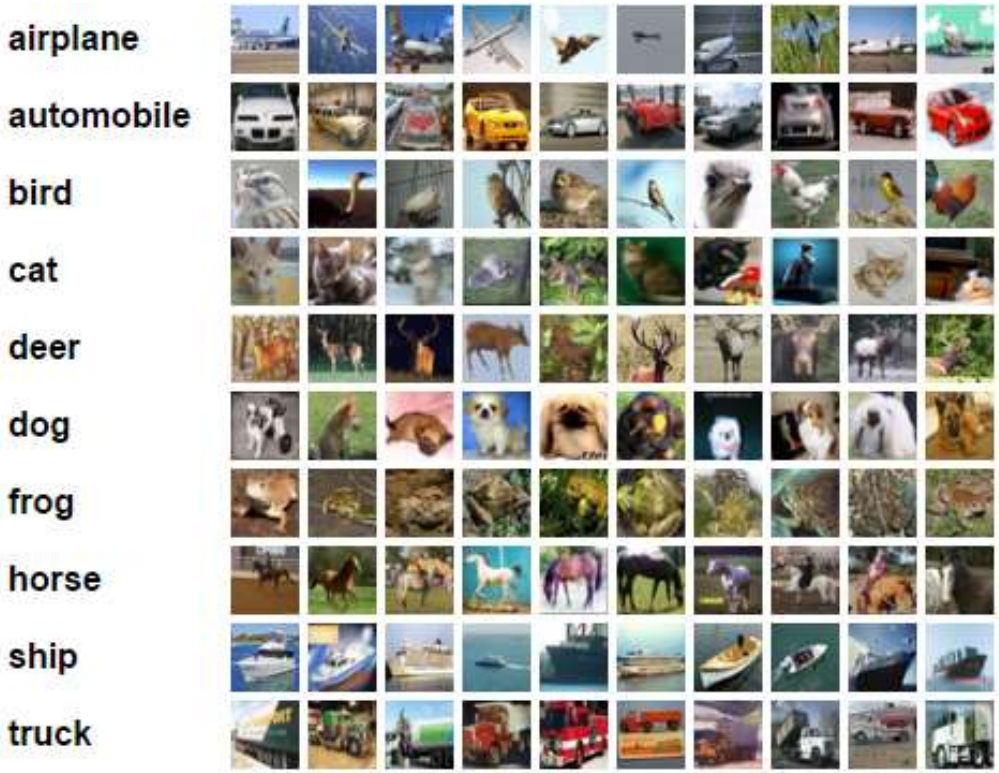
Декомпозиция задачи:



- Мы можем использовать любой классификатор из машинного обучения
- Нам потребуется:
  - Тестовая и обучающая выборки
  - Критерий качества работы
  - Метод построения вектор-признака изображения
- После этого применим выбранный метод машинного обучения для построения классификатора

CIFAR-10 и CIFAR-100

Выборки из TinyPictures



- CIFAR-10
  - 10 классов
  - 60000 изображений
  - 5000 обучающих и 1000 тестовых на класс
- CIFAR-100
  - 100 классов
  - 60000 изображений
  - 500 обучающих и 100 тестовых на класс

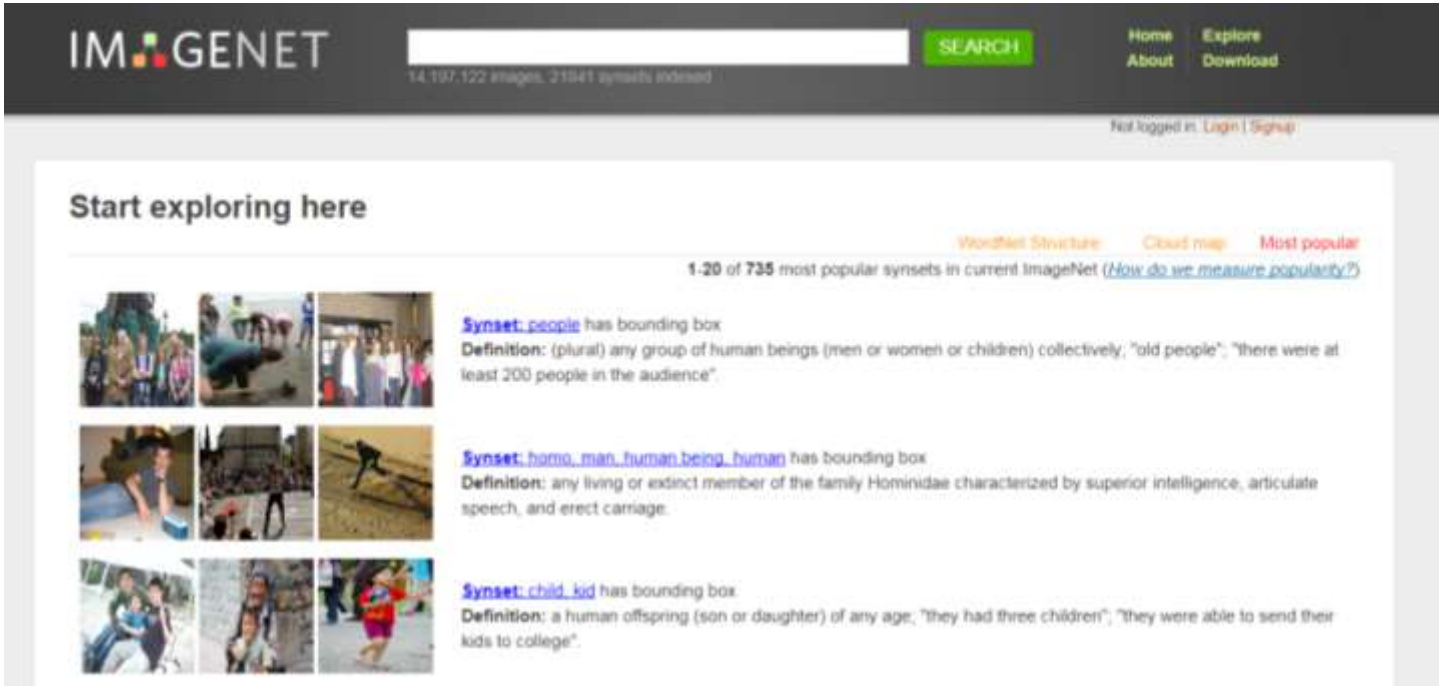
http://www.cs.toronto.edu/~kriz/cifar.htm

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

ImageNet

Цель: собрать коллекцию с 1000 изображений на каждый из 117 тыс. synsets.

Текущая статистика:
Total number of non-empty synsets: 21841
Total number of images: 14,197,122
Number of images with bounding box annotations: 1,034,908



Source: http://image-net.org

Large-scale visual recognition (LSVR)

Task 1: Classification

Car

- Predict a class label
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 50% of training.

Task 2: Detection
(Classification + Localization)

classification Car

- Predict a class label and a bounding box
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 40% of training.

Task 3: Fine-grained classification

classification Walker hound

- Predict a class label given a bounding box in test
- 1 prediction / image
- 120 dog classes (subset)
- ~200 images per class for training (subset)
- Bounding boxes for 100% of training

Fine-graded classification

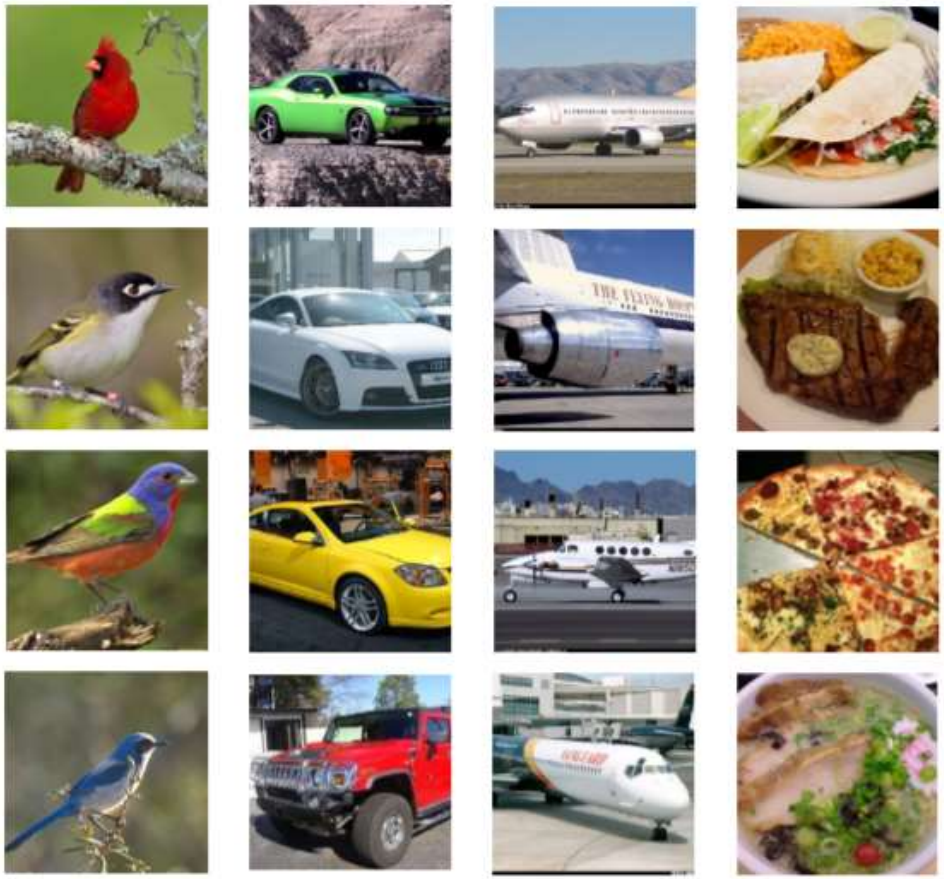
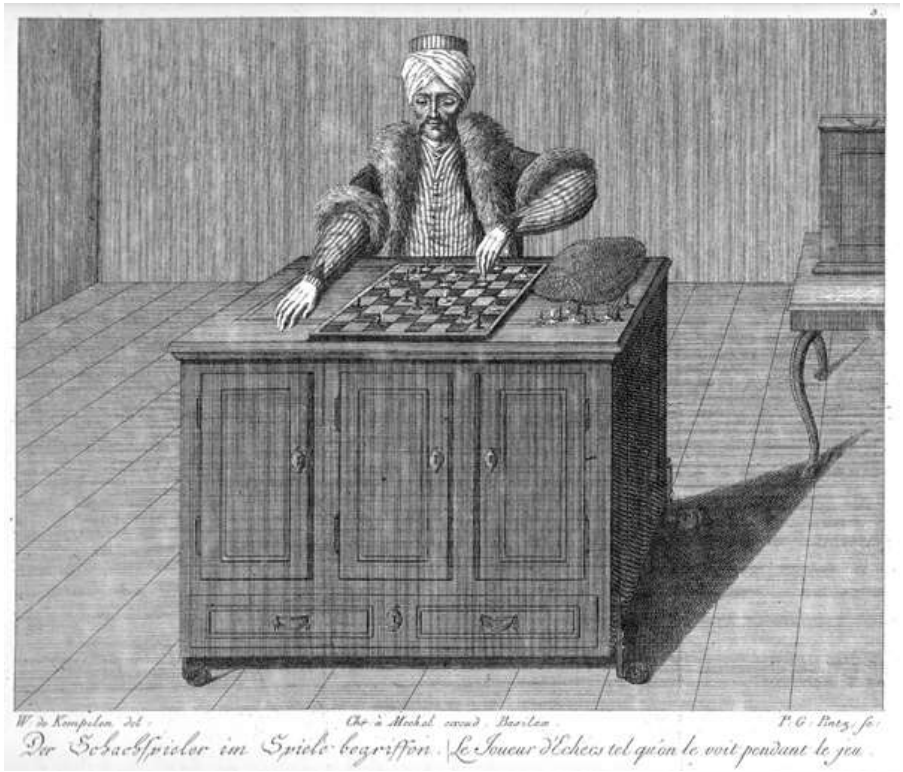


Figure 7. Images we used for visual recognition. From left to right, each column contains examples from CUB Bird [43], Stanford Car [19], Aircraft [27] and Food-101 [4].

Как готовить коллекции?



- Mechanical Turk - Automaton Chess Player – робот, игравший в шахматы
  - Автоматон двигает фигуры, говорит «Чек» и обыгрывает всех!
- С 1770 по 1854 развлекал публику, только в 1820 году раскрыли обман

Galaxy Zoo

GALAXY ZOO http://www.galaxyzoo.org/

- Классификация изображений галактик
- Первый масштабный проект такого рода
- Более 150000 волонтёров за первый год бесплатно сделали более 60 млн. меток

Система LabelMe

- Просим пользователей выделять любые объекты замкнутыми ломаными и давать им название

B. Russell, A. Torralba, K. Murphy, W. T. Freeman LabelMe: a database and web-based tool for image annotation, IJCV, 2008

Метрики

1) % правильно классифицированных изображений

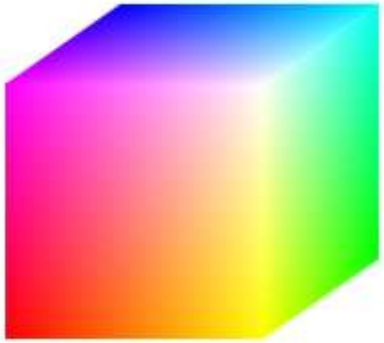
Dataset	CNN	Original	BP[23]	CBP[11]	KP	Others	
CUB [43]	VGG-16 [38]	73.1*	84.1	84.3	**86.2**	82.0	84.1
	ResNet-50 [15]	78.4	N/A	81.6	84.7	[18]	[16]
Stanford Car [19]	VGG-16	79.8*	91.3	91.2	**92.4**	**92.6**	82.7
	ResNet-50	84.7	N/A	88.6	91.1	[18]	[14]
Aircraft [27]	VGG-16	74.1*	84.1	84.1	**86.9**	80.7	
	ResNet-50	79.2	N/A	81.6	85.7	[14]	
Food-101 [4]	VGG-16	81.2	82.4	82.4	84.2	50.76	
	ResNet-50	82.1	N/A	83.2	**85.5**	[4]	

Table 2. Performance comparisons among all baselines, where KP is the proposed kernel pooling method with learned coefficients. Following the standard experimental setup, we use the input size of 448×448 for CUB, Stanford Car and Aircraft datasets except the original VGG-16 (marked by an asterisk *), which requires a fixed input size of 224×224. For Food-101, we use the input size of 224×224 for all the baselines.

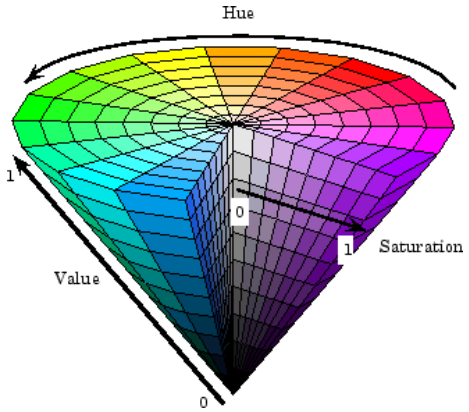
2) Rank X

- Если классификация многоклассовая, мы ранжируем ранжируем все выходы по качеству
- Если истинный ответ попадает в первые X выходов, тогда результат считается верным

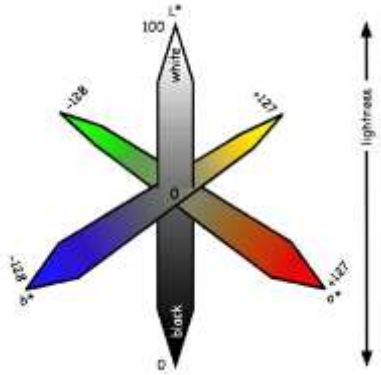
Какие мы знаем признаки пикселей?



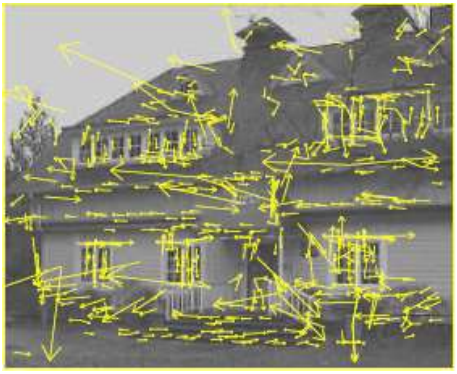
Пространство цветов RGB

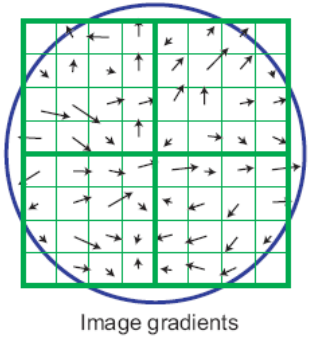


Пространство цветов HSV

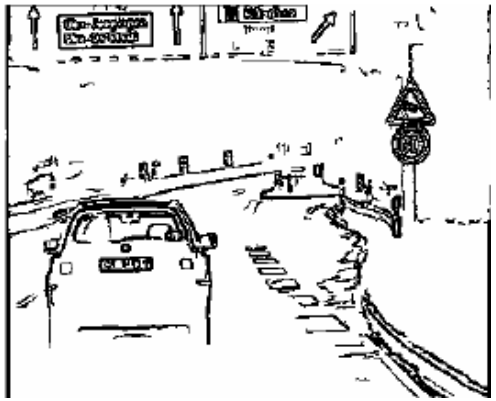


Пространство цветов L*a*b*





Градиенты в каждом пикселе



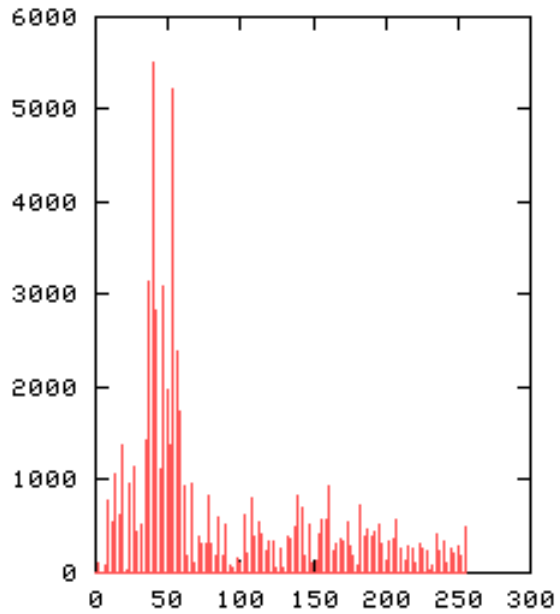
Наличие и ориентация края в каждом пикселе

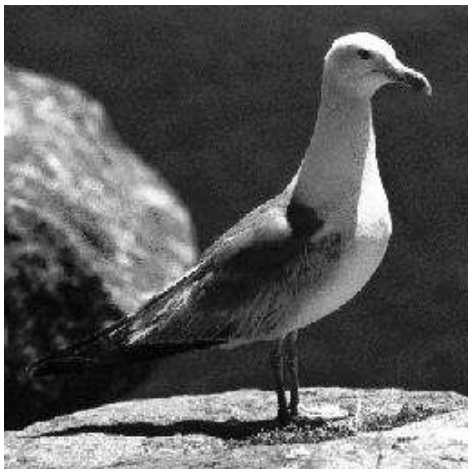
Использование напрямую пикселей

- Можно ли использовать признаки всех пикселей напрямую?
- Можно, если все изображения для классификации будут одинакового размера
- Нормализуем изображения, т.е. приведем их к одному размеру
- Вытянем изображение в вектор, признаки пикселей будут элементами вектора

Для распознавания, изображение должно описываться вектор-признаком фиксированной длины!

Гистограммы





- Вместо использования пикселов напрямую, можем считать разные статистики распределения характеристик
  - Цвет, края, градиенты, и т.д.
- Гистограммы – стандартный способ непараметрического описания распределения признаков

Images from Dave Kauchak

Квантование признаков

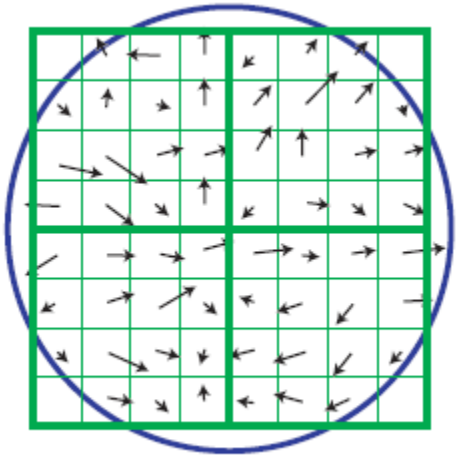
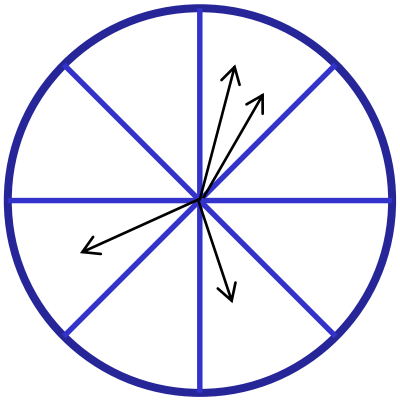
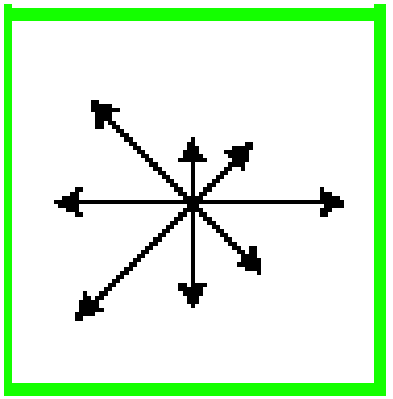


Image gradients

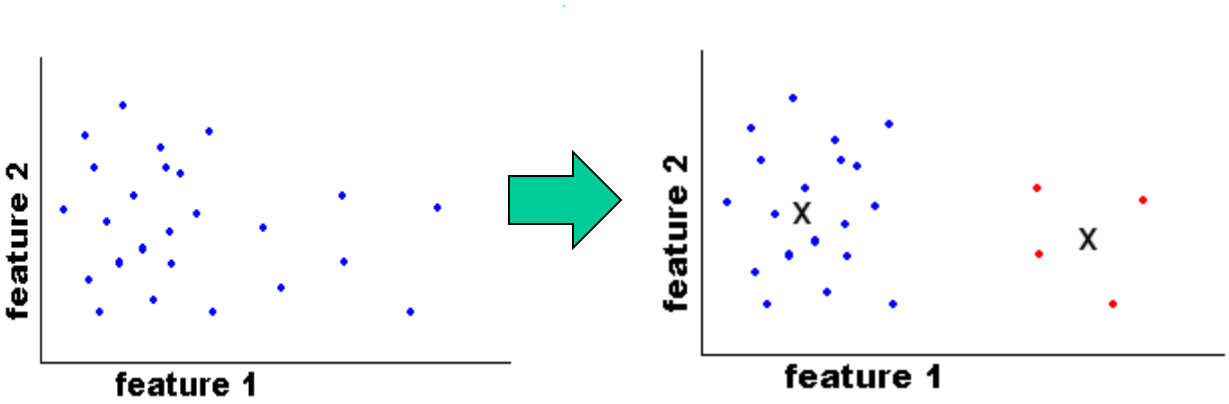




- Для некоторых признаков потребуется дискретизация (квантование)
  - Если признаки вещественные
  - Если слишком большое разрешение
- Пример для направления градиента
  - Разбиваем 360 градусов на 8 секторов
  - Каждый сектор нумеруем от 1 до 8
  - Считаем число пикселов, градиенты которых попадают в соответствующие сектора

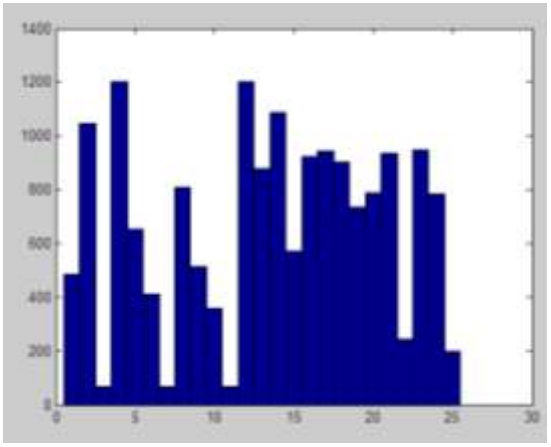
Квантование через кластеризацию

Не все цвета встречаются!



- Во многих случаях равномерное квантование по сетке не подходит
  - Высокая размерность
  - Неравномерные распределения
- Кластеризация данных
  - K-средних и производные чаще всего используются
- Для каждого нового изображения мы квантуем признаки и строим гистограмму частот

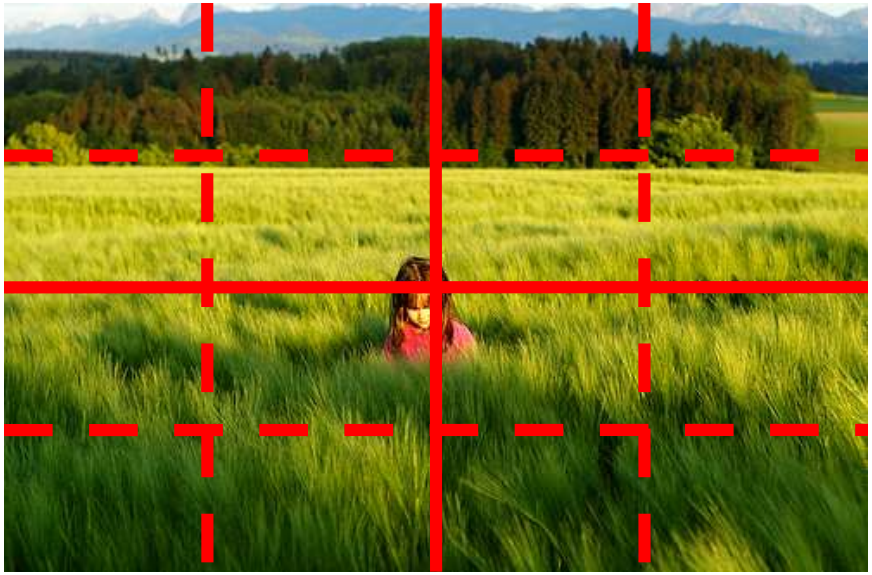
Недостаток гистограммы как признака

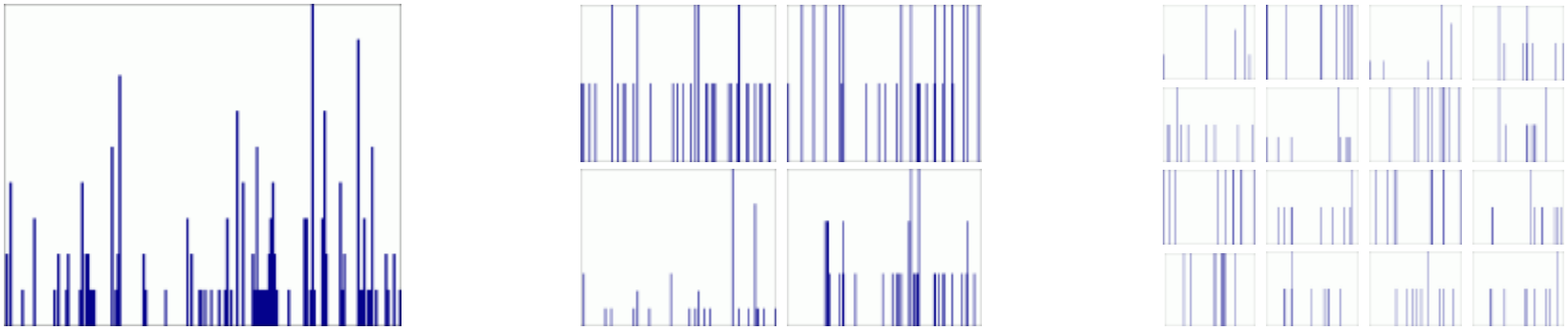


У всех этих трех изображений похожие гистограммы цветов

Пространственная пирамида

Вычислим гистограму в каждом блоке и объединим все гистограммы в один вектор-признак

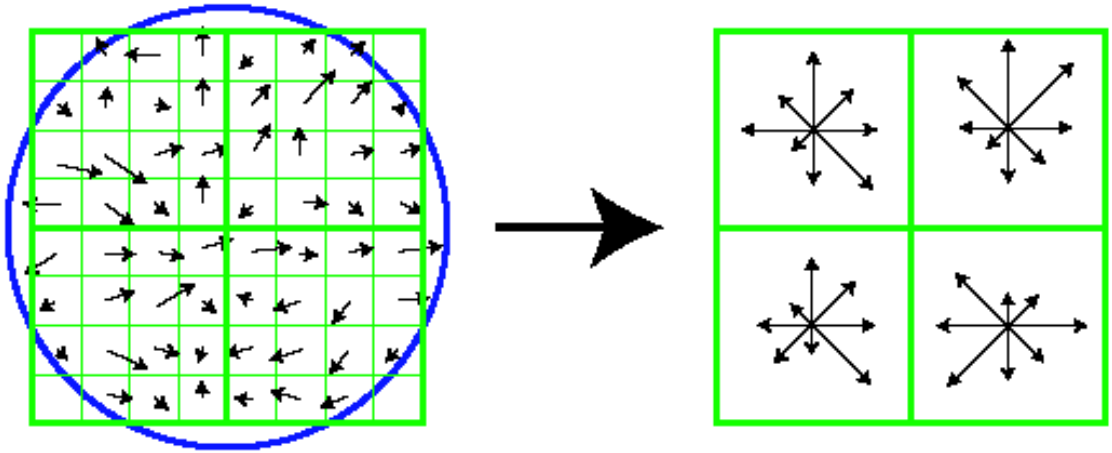




Уровень 0 Уровень 1 Уровень 2

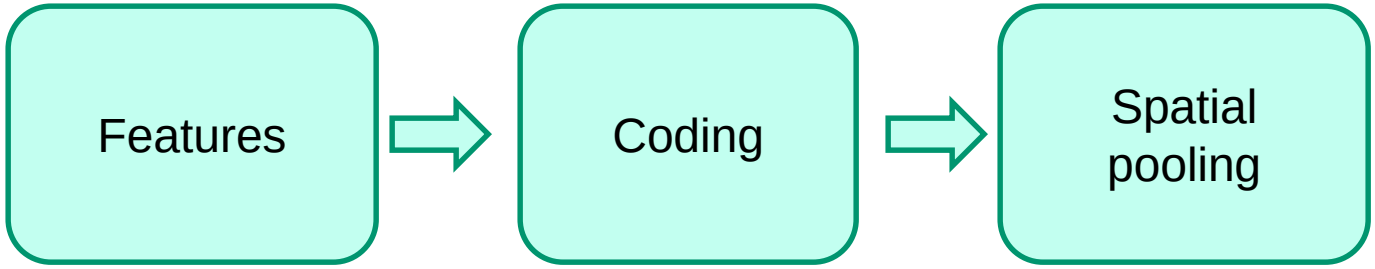
S. Lazebnik et. Al. **Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories** *CVPR 2006*

Гистограммы градиентов

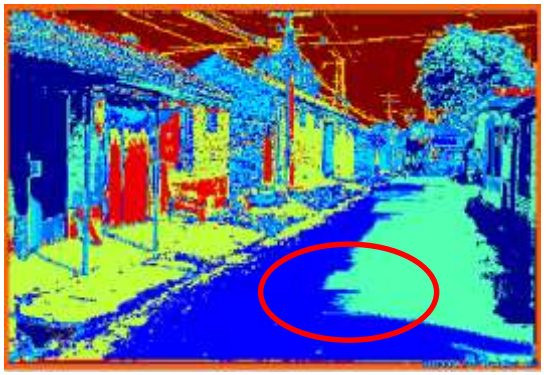


- Много названий – HOG (Histogram of Oriented Gradients), дескриптор SIFT
- Построение
  - Разбиваем изображение на области сеткой
  - В каждой сетке считаем гистограмму распределения пикселов по ориентации градиентов
  - Обычно 6-8 корзин в каждой гистограмме
  - Вычисляем градиент в каждой точке
- Какие достоинства у таких признаков?

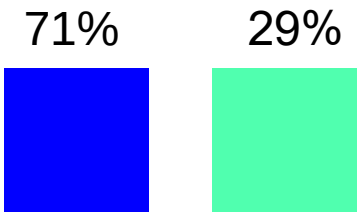
Общая схема для признаков



RGB



Квантование на 10 уровней



Посчитанная гистограмма

Spatial pooling (пространственная агрегация) = выбор областей для агрегации и метод агрегации

Мешок слов

Классификация текстов

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

Классификация текстур

Текстýра — преимущественная ориентация элементов, составляющих материал

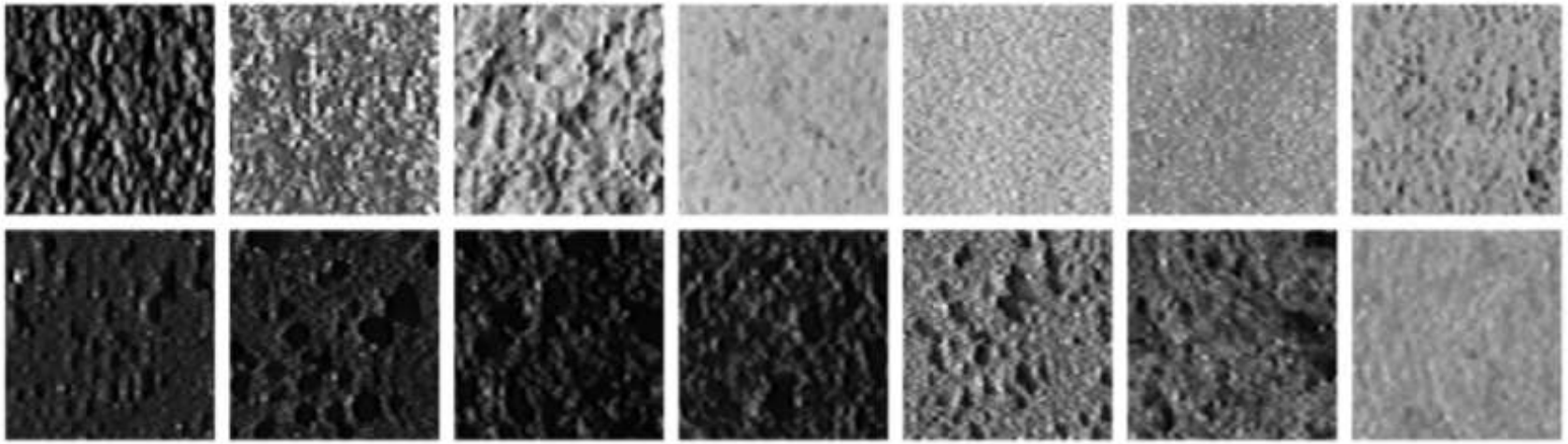
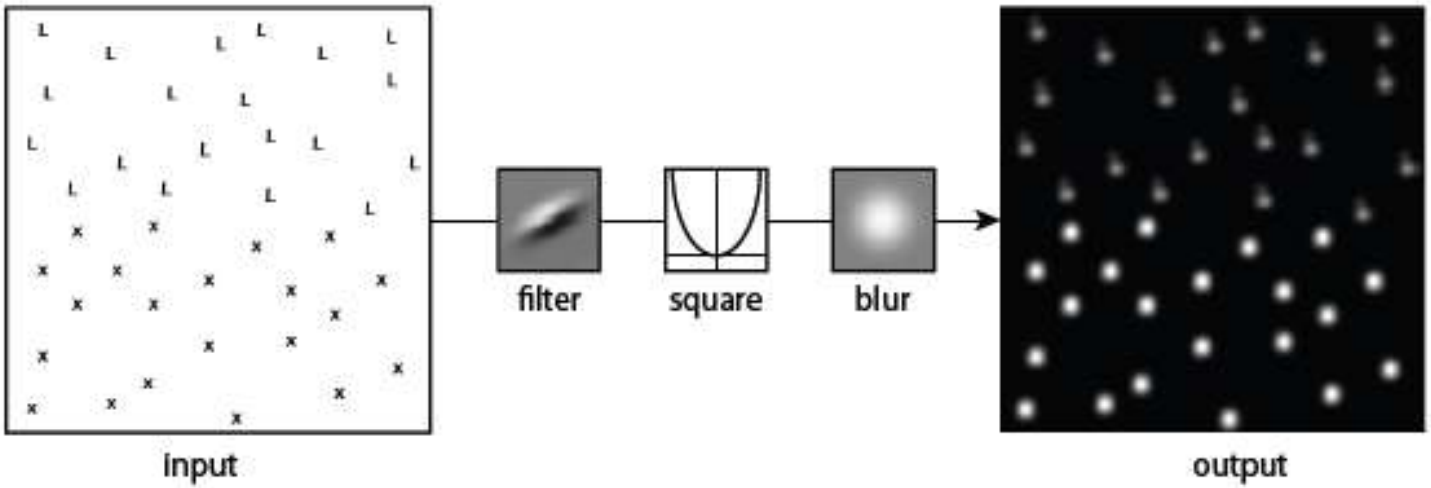
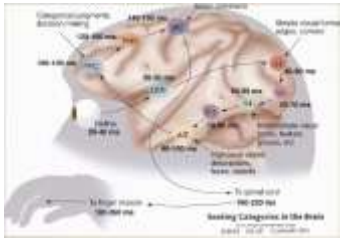
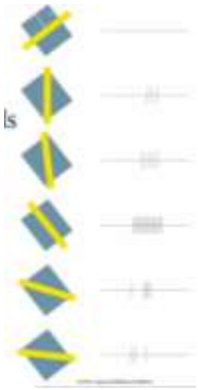


Figure 2. Small inter class variations between textures can make the problem harder still. In the top row, the first and the fourth image are of the same texture while all the other images, even though they look similar, belong to different classes. Similarly, in the bottom row, the images appear similar and yet there are three different texture classes present.

Анализ текстуры







- Выберем фильтр, чувствительный к краю определенной ориентации
- Результат фильтрации сгладим
- Будут «подсвечены» области, содержащие текстуру с краями заданной ориентации

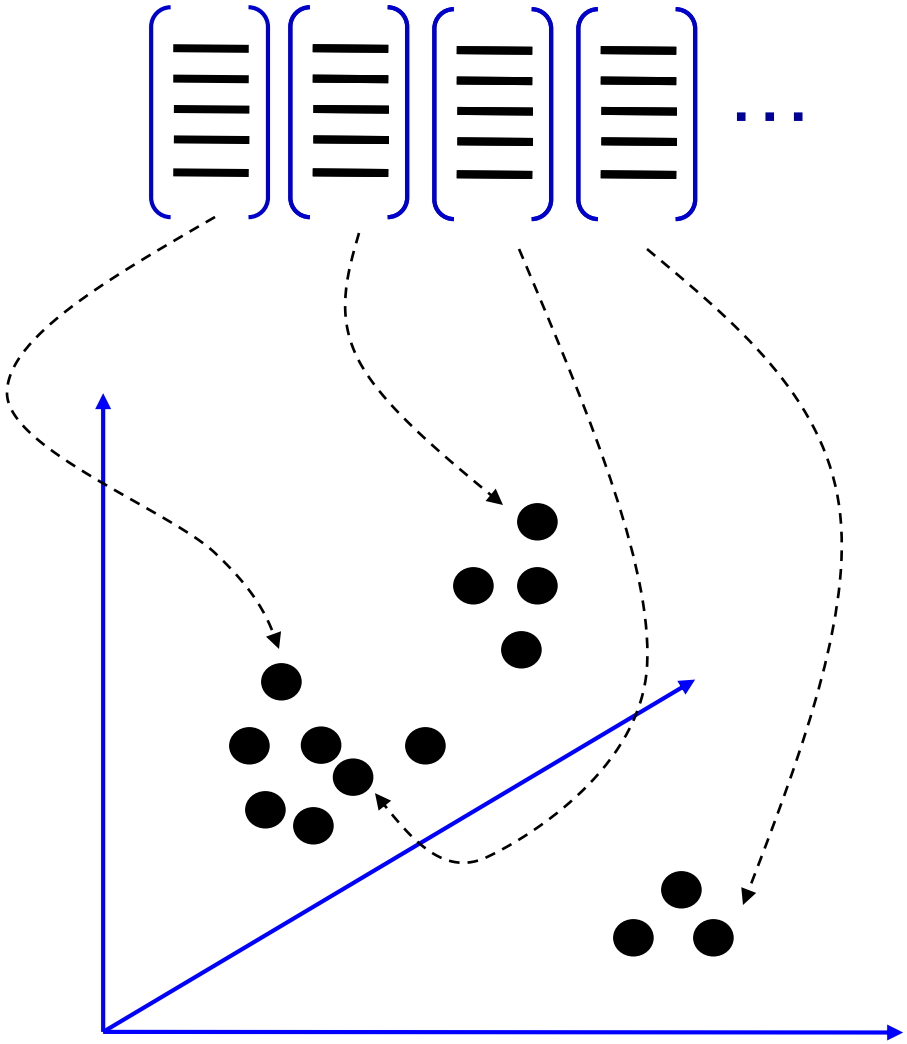
Pietro Perona and Jitendra Malik «Detecting and Localizing edges composed of steps, peaks and roofs», ICCV 1990

Банки фильтров и признаки текстуры

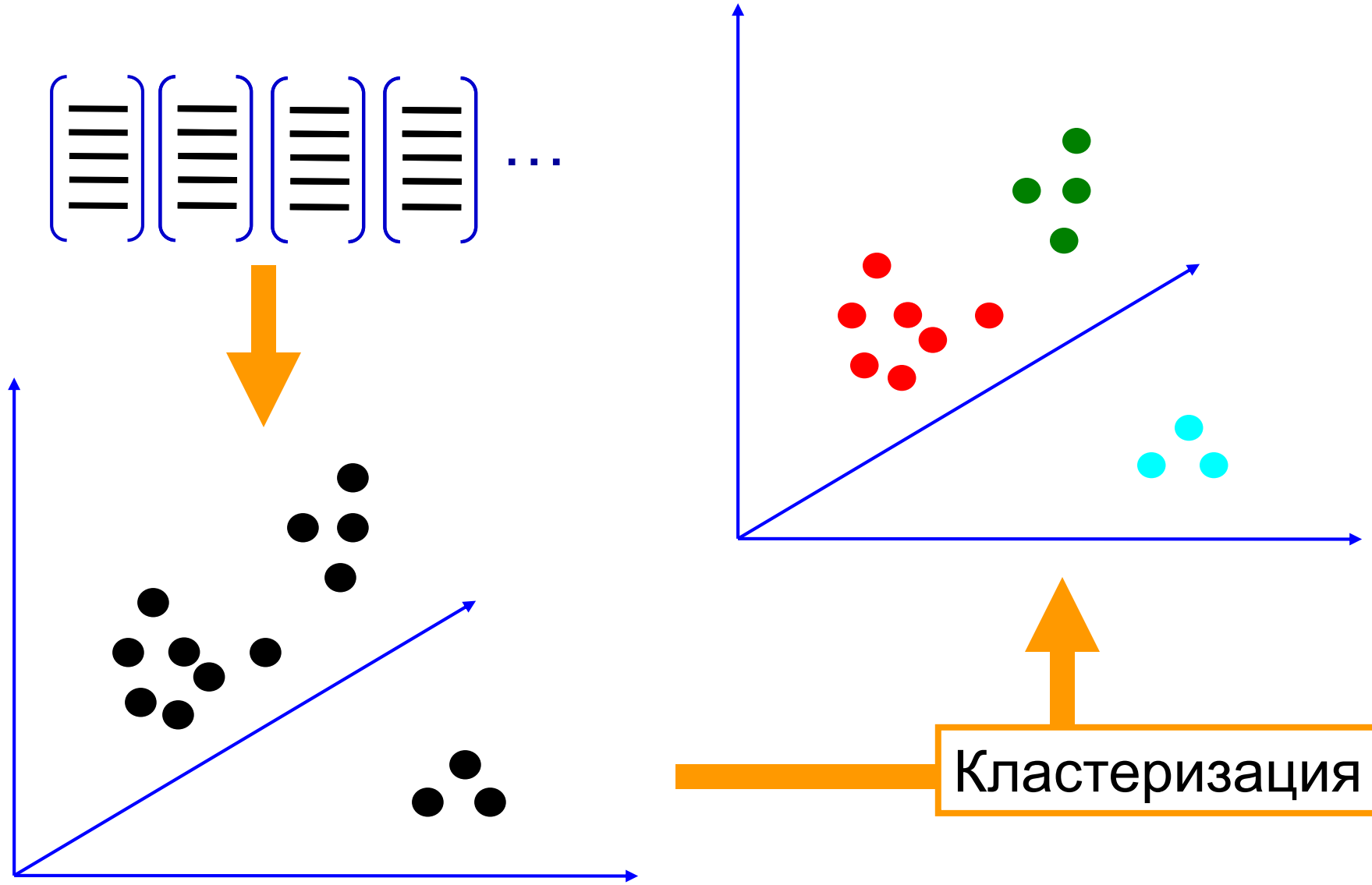
- Возьмём теперь несколько фильтров разного масштаба и ориентации
- Такой набор называют «банк фильтров»
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окресности пикселя

- Как нам построить «словарь» элементов текстуры?

Как построить словарь?

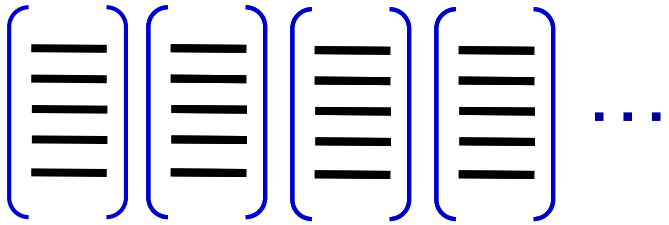


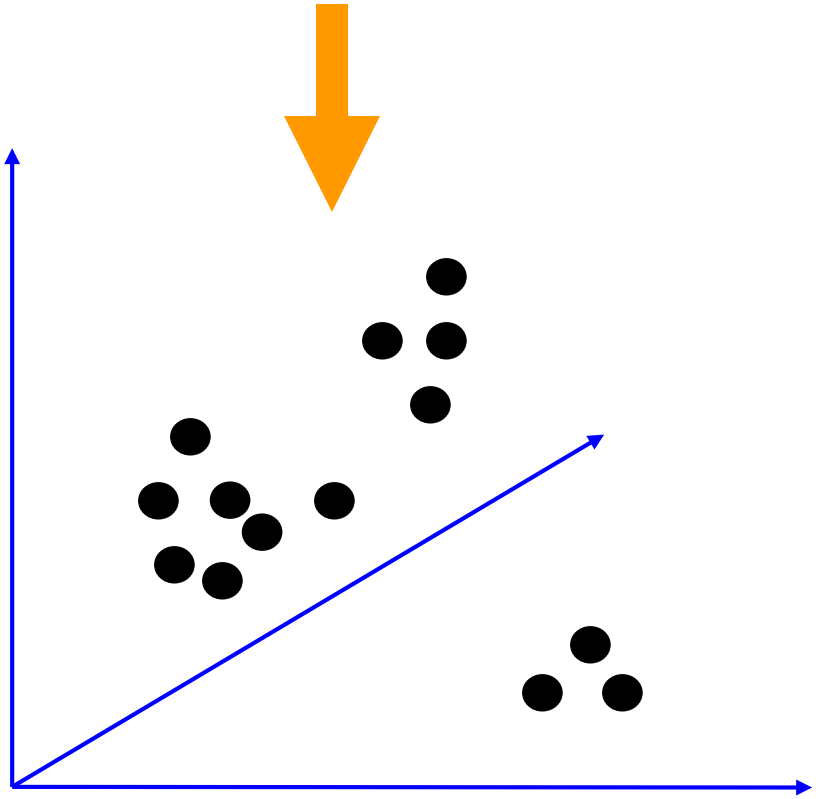
Обучение словаря

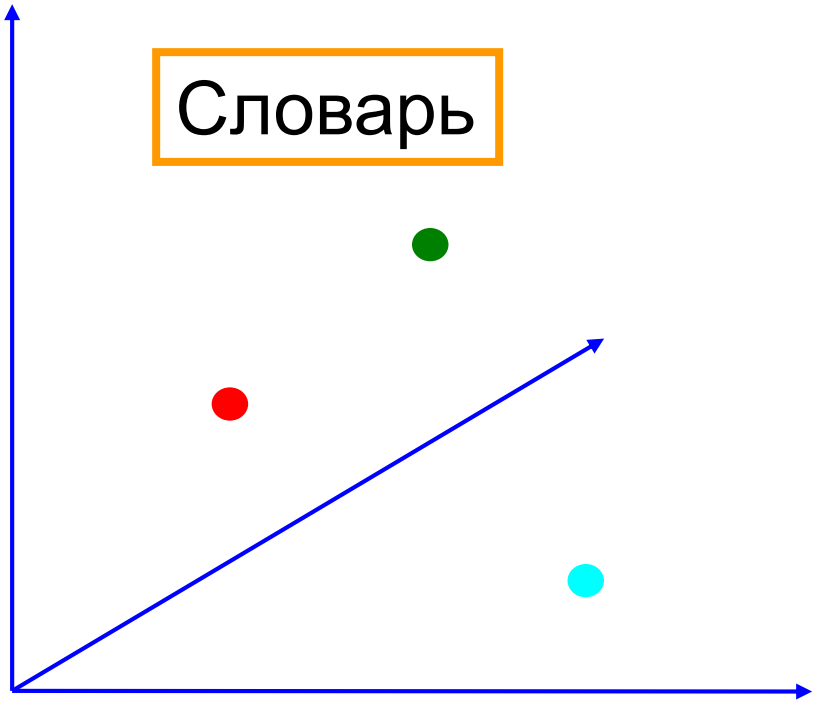


Slide credit: Josef Sivic

Обучение словаря





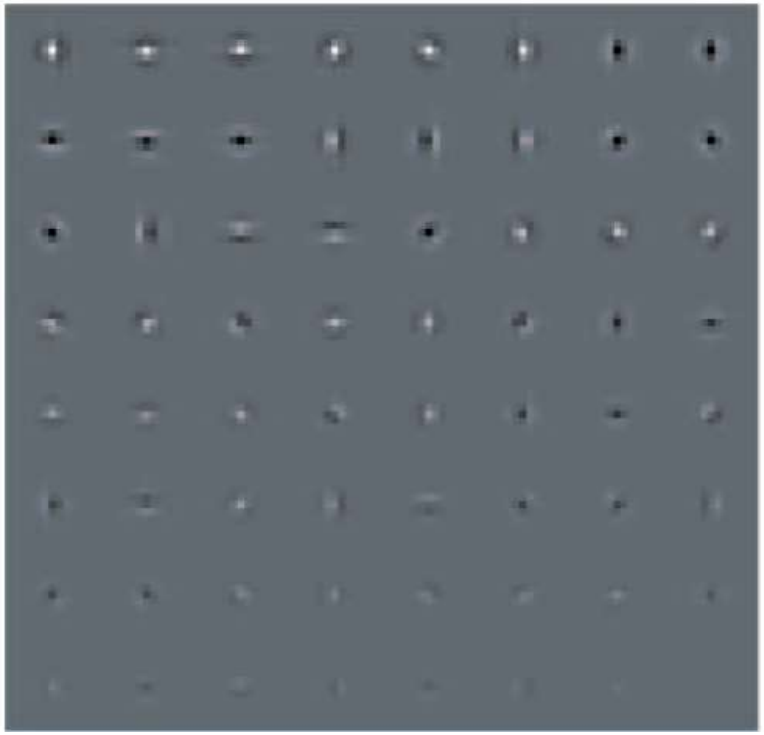


Кластеризация

Slide credit: Josef Sivic

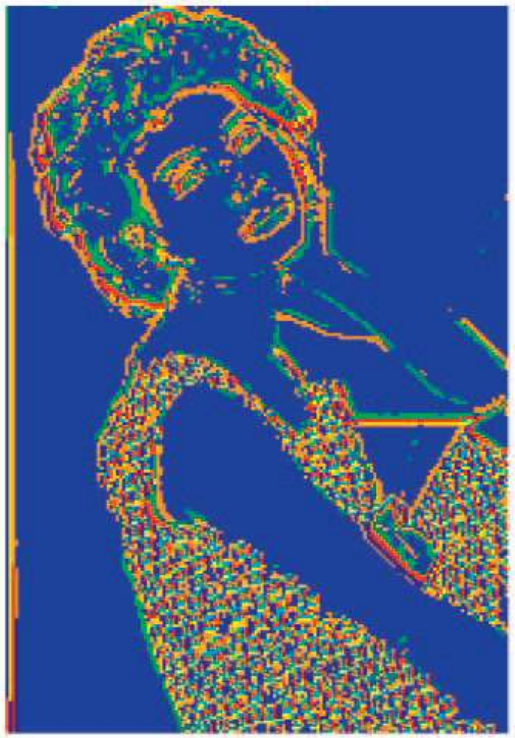
Текстоны

«Текстон» = характерный фрагмент текстуры изображения



Часть текстонов из словаря

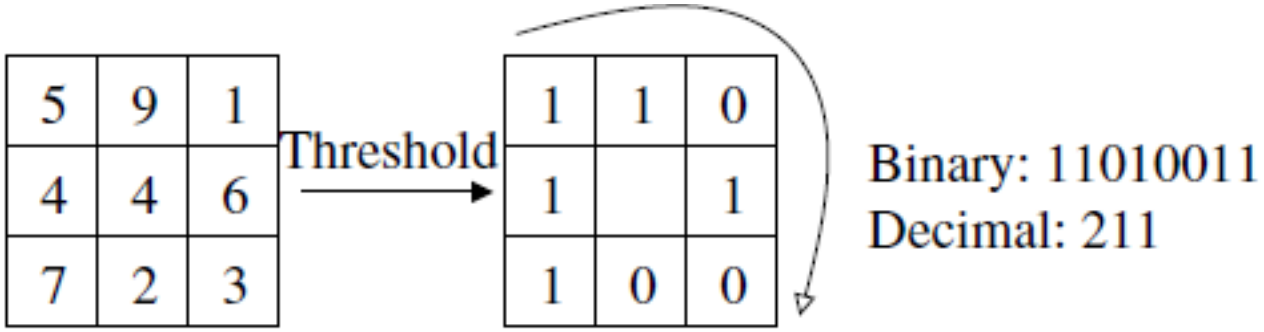
Изображение



Карта текстонов

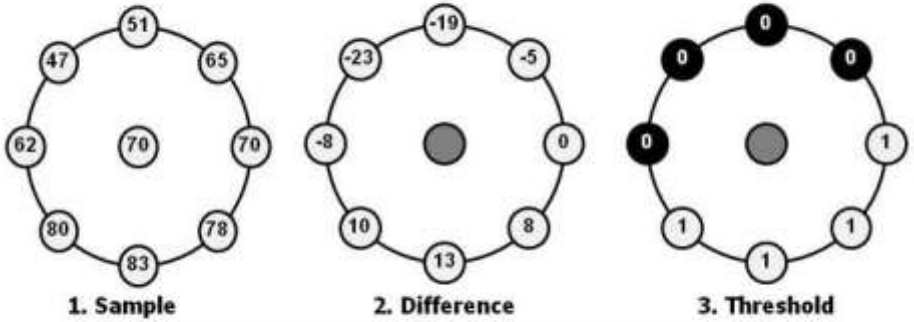
Текстуру фрагмента изображения можно описать гистограмой частот текстонов

Local Binary Patterns



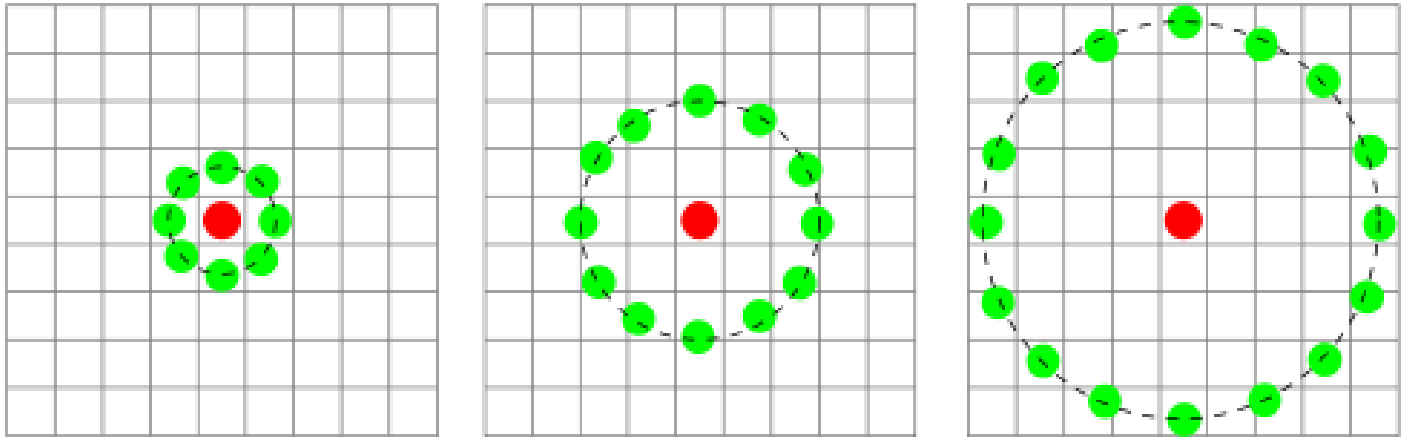
The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$



T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", ICPR 1994

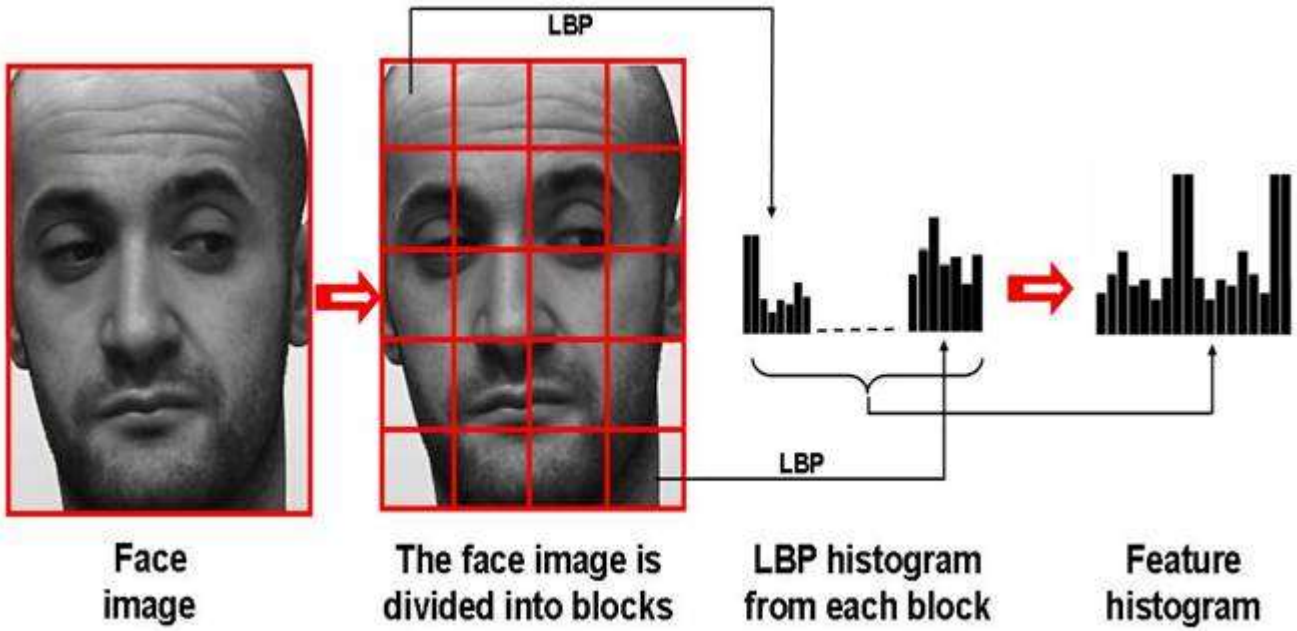
Расчёт LBP



Может потребоваться интерполировать значения в точках, необходимых для расчёта LBP

Мы можем использовать LBP код как номер визуального слова в фиксированном словаре

Применение LBP



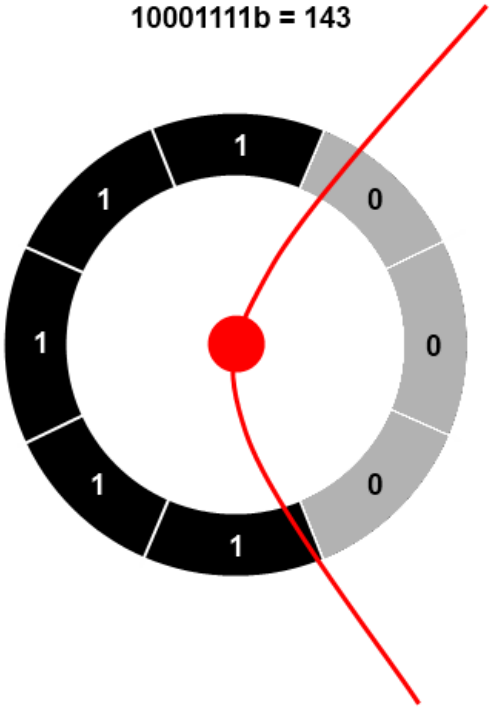
- Изображение разбивается на области. В каждой области применяются LBP операторы к каждому пикселю. Строится гистограмма.
- Объединение гистограмм – LBP дескриптор для изображения.
- Для пары изображений считается разность дескрипторов по какой-нибудь метрике (например, Хи-квадрат)

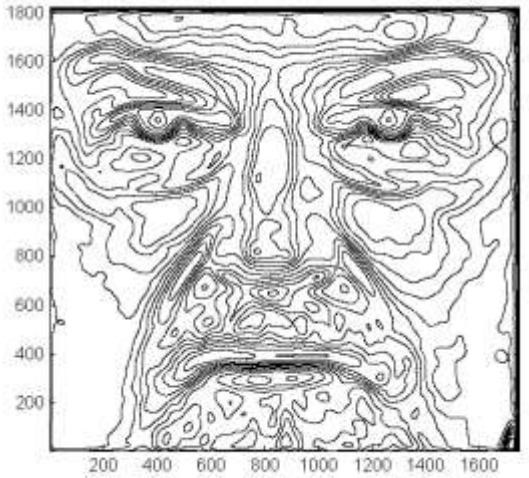
Ahonen, T., Hadid, A. and Pietikäinen, M. (2006), Face Description with Local Binary Patterns: Application to Face Recognition. IEEE PAMI 28(12):2037-2041.

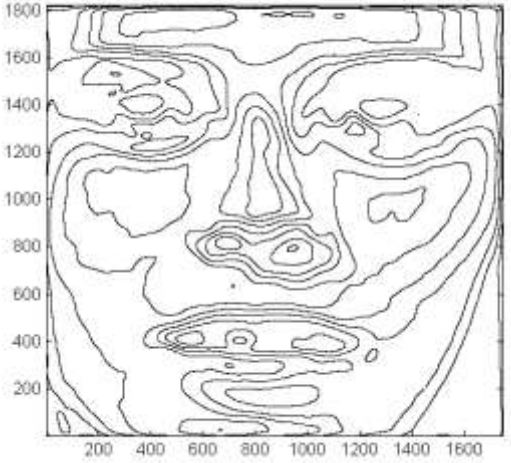
Uniform LBP

- Рассмотрим количество переходов:
  - 00000000 (0 transitions)
  - 01110000 (2 transitions)
  - 11001111 (2 transitions)
  - 11001001 (4 transitions)
  - 01010010 (6 transitions)
- Однородными LBP называются такие коды, в которых переходов <= 2
- Однородные составляют от 80 до 90% всех кодов
- Неоднородные коды иногда объединяют в один
- Для кода из 8 бит
  - 256 кодов всего
  - 58 однородных (59 разных меток)

Интерпретация однородных LBP

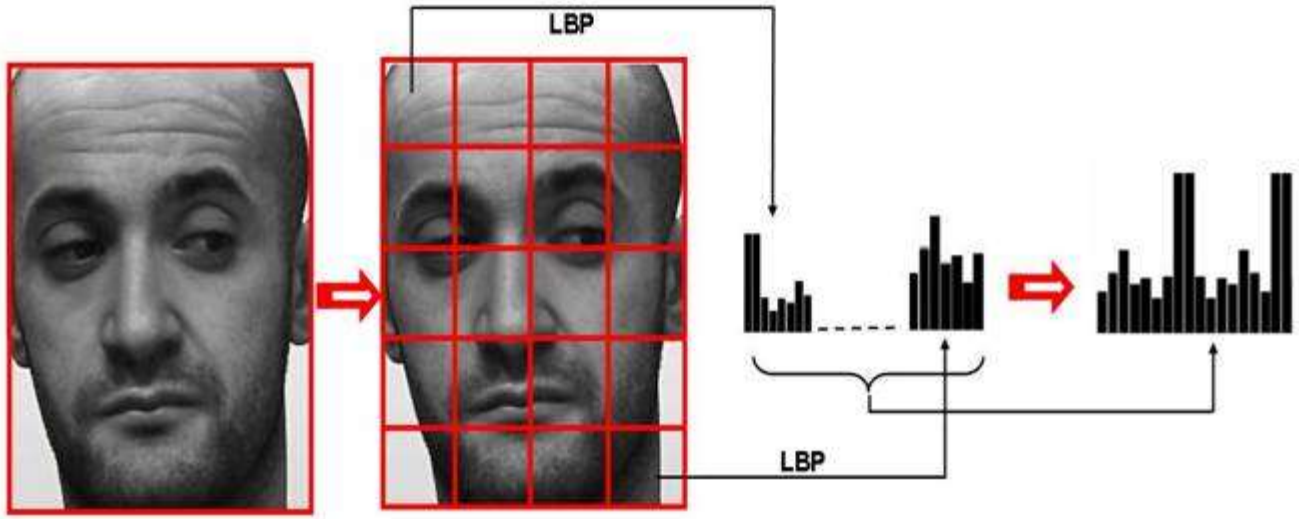






- Коды с 2мя переходами - кодирование изолиний

Текстоны и LBP-коды



- Два варианта описания пикселов по их локальной окрестности
- Мы делим пиксели на классы и считаем частоту классов пикселом по окрестностям

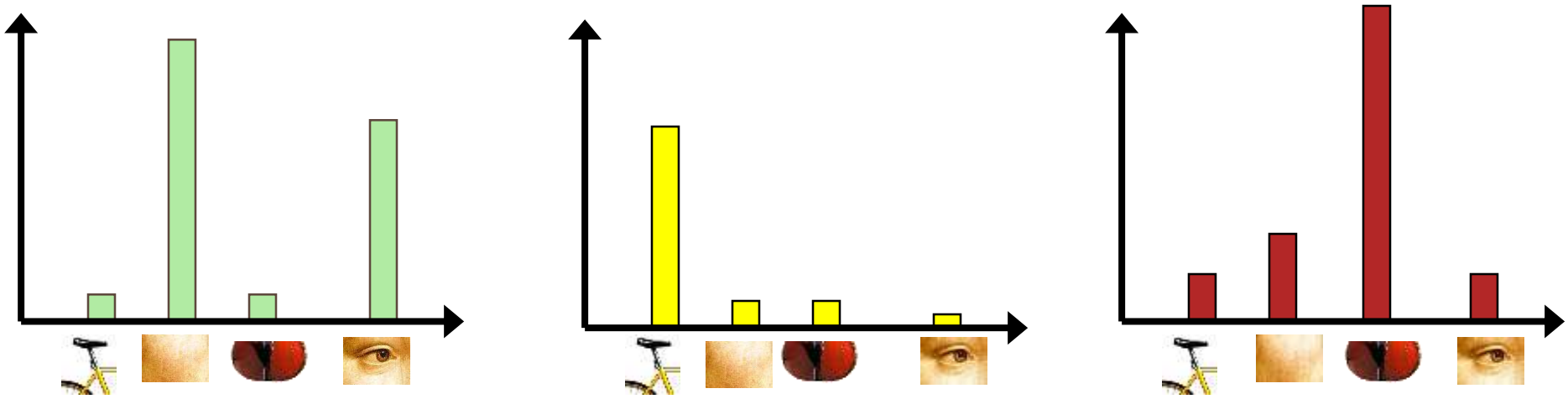
«Визуальные слова»

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

Sivic, J. and Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV 2003

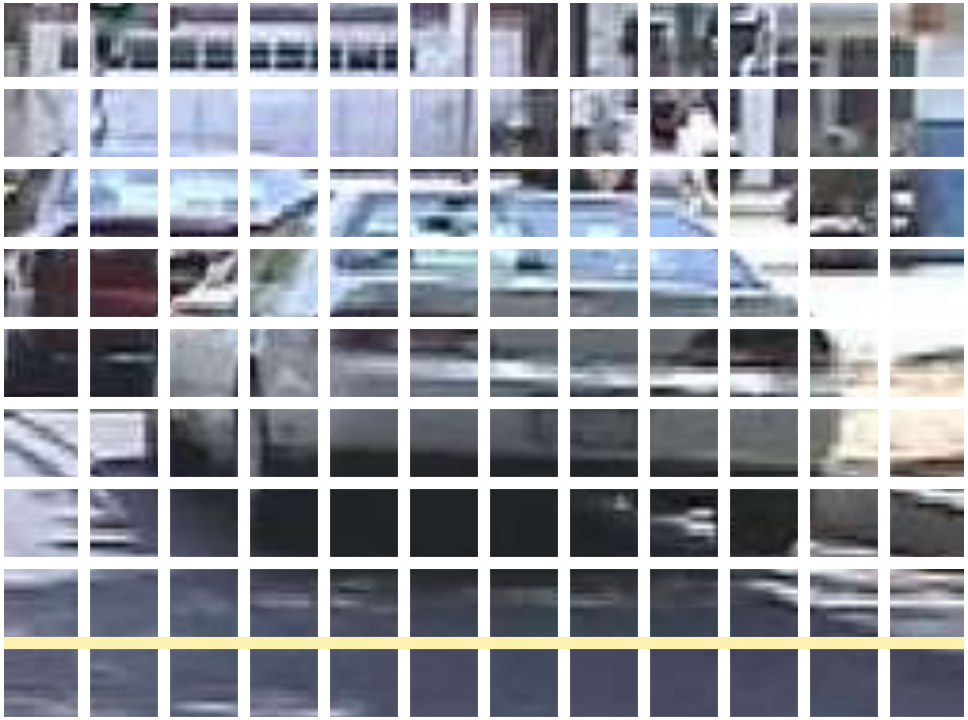
Схема метода «мешок слов»

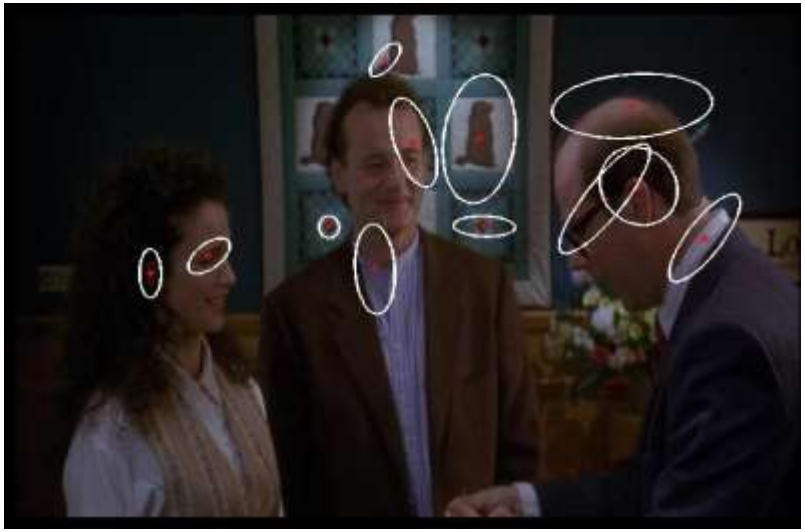
1. Построение словаря
   1. Извлечение фрагментов из обучающей выборки
   2. Кластеризация и построение «визуального» словаря
2. Построение дескрипторов
   1. Извлечение фрагментов из изображения
   2. Квантование фрагментов по словарю
   3. Построение гистограммы частот визуальных слов



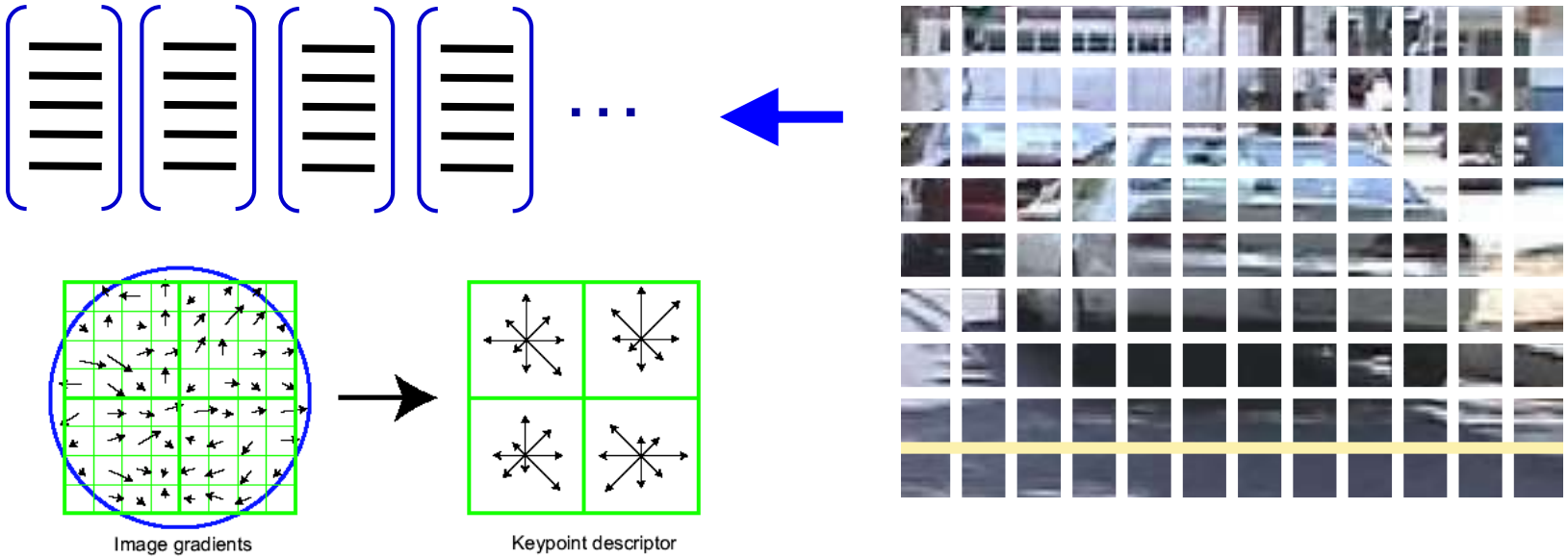
1. Извлечение фрагментов

- По регулярной сетке
  - Можем без перекрытия, можем с перекрытием
- Вокруг характерных точек
  - Точки, которые можно устойчиво идентифицировать на изображении после деформации



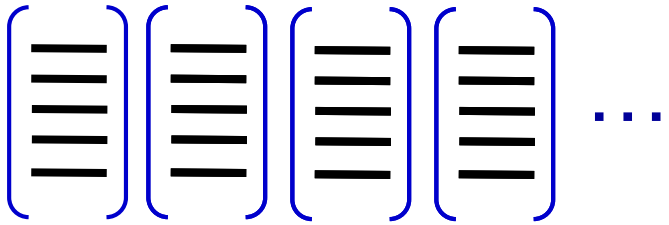


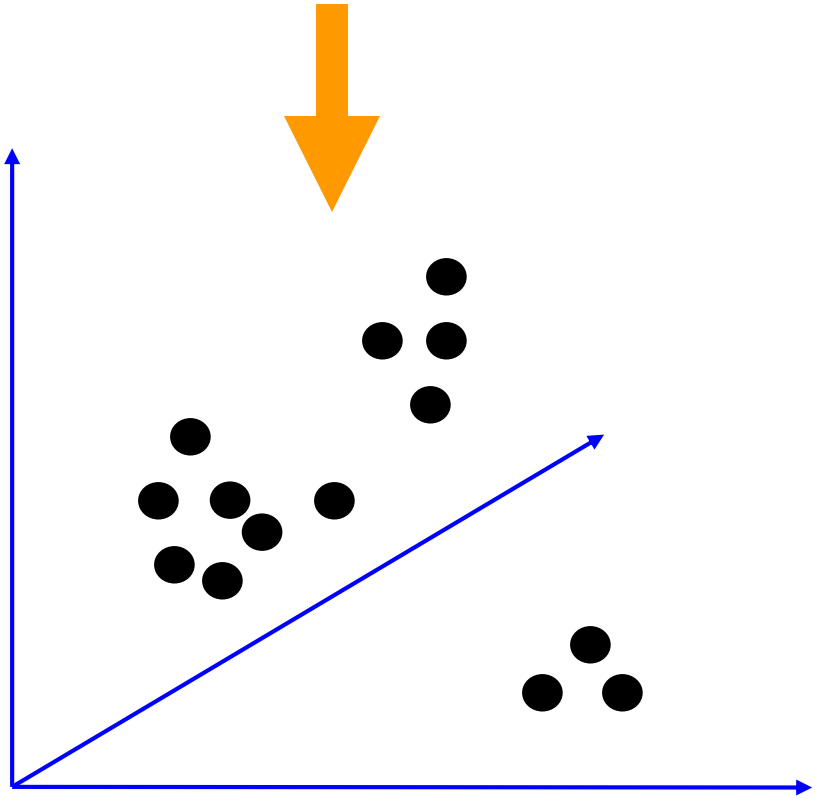
1. Вычисление признаков фрагментов



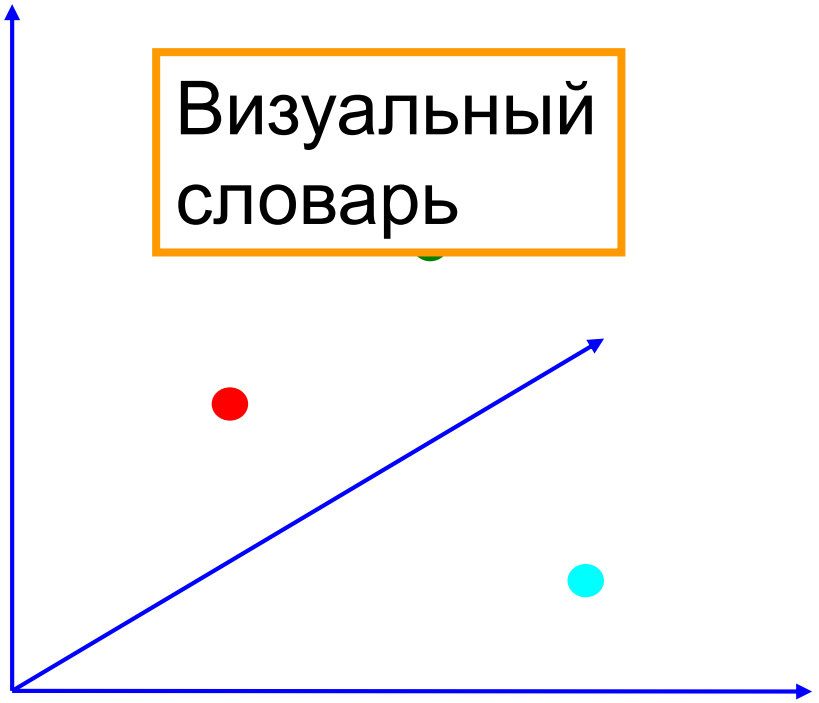
- Неупорядоченный список фрагментов из множества изображений
- Нужно описать каждый фрагмент вектор-признаком
- Например, с помощью SIFT

2. Обучение словаря через кластеризацию





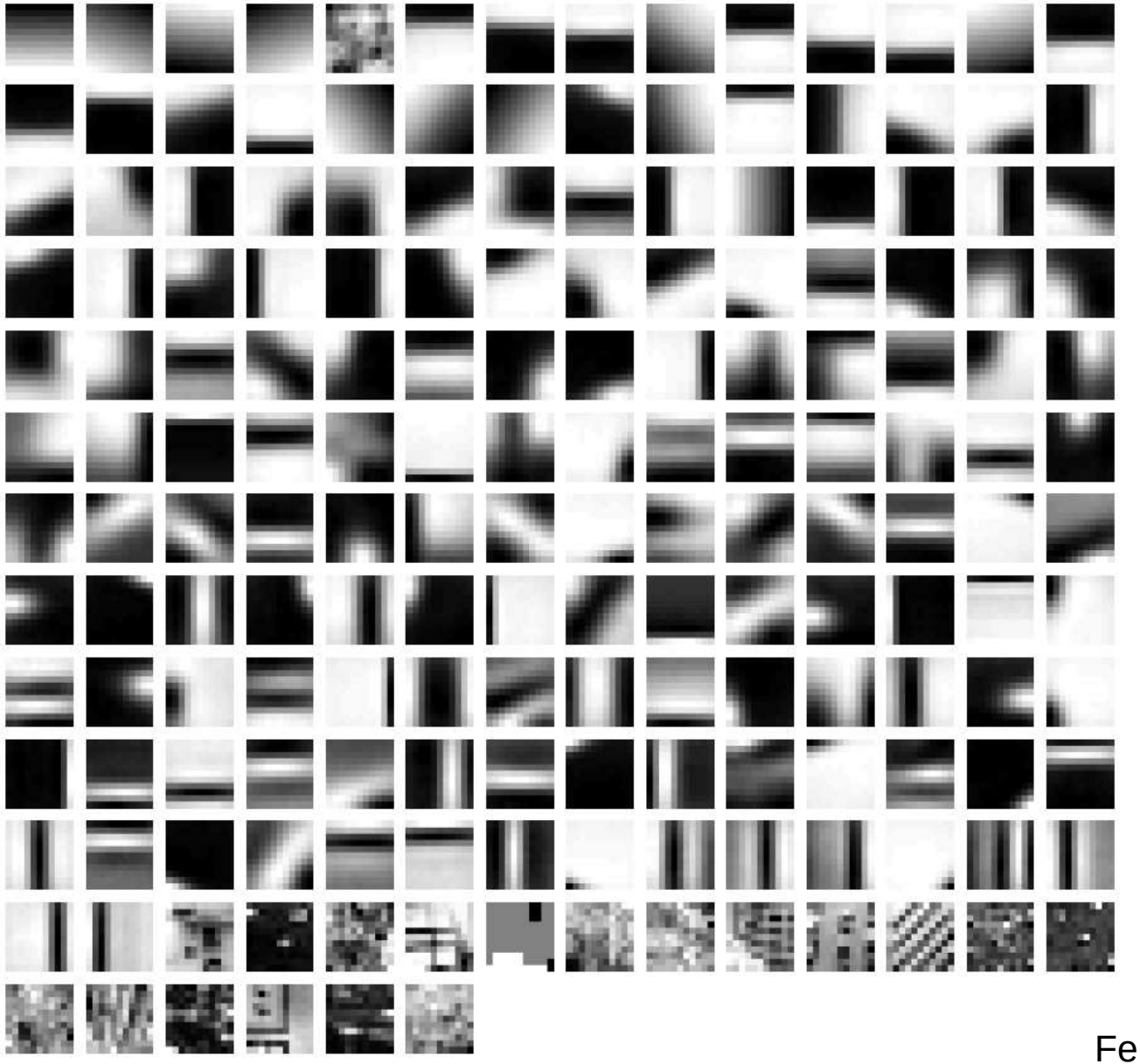
Визуальный словарь



Кластеризация

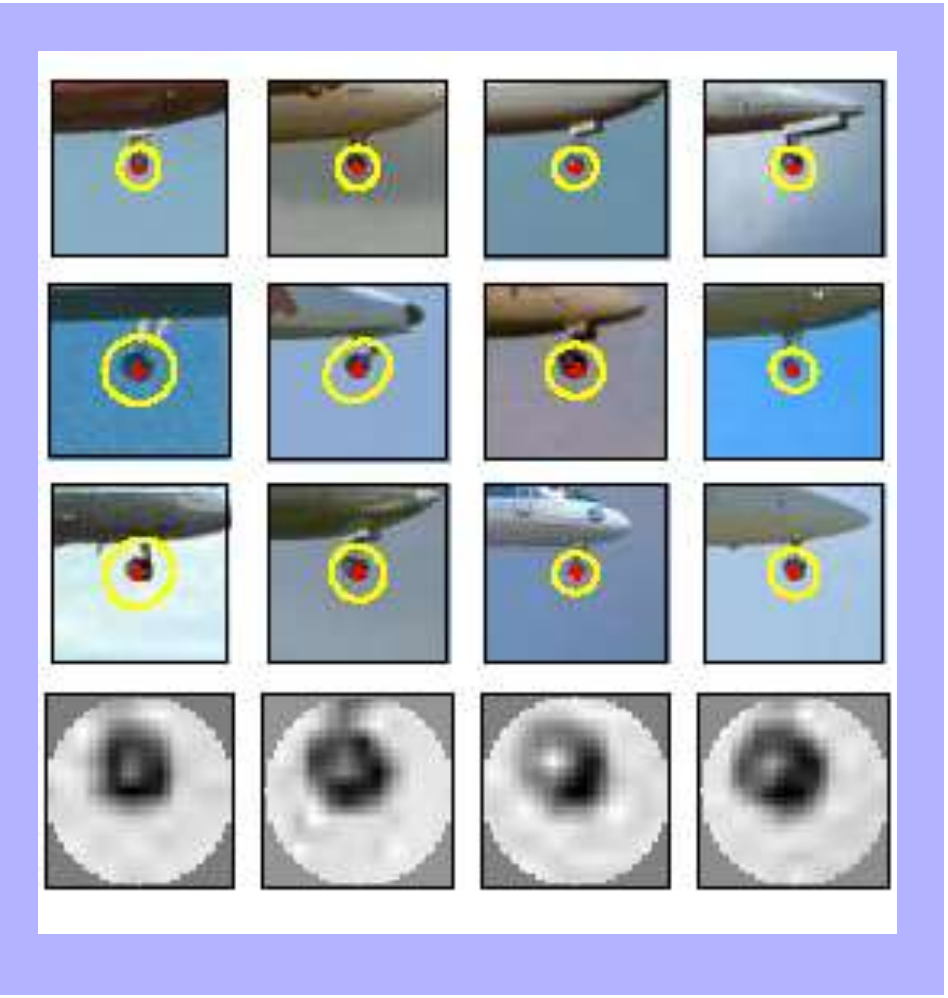
Slide credit: Josef Sivic

Пример словаря



Fei-Fei et al. 2005

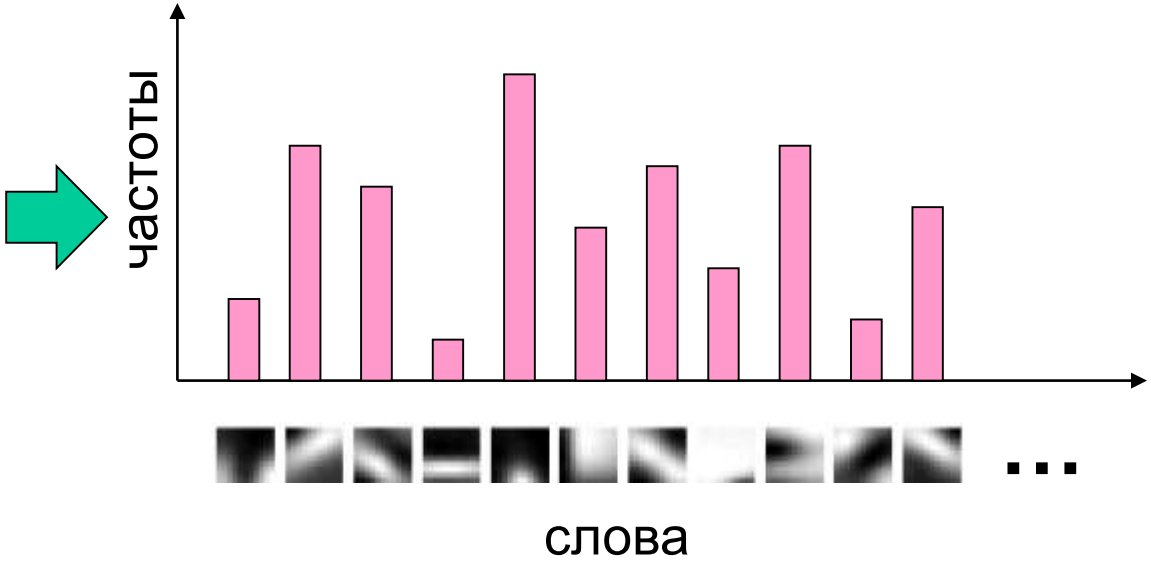
Примеры визуальных слов



Построены по характерным точкам

Sivic et al. 2005

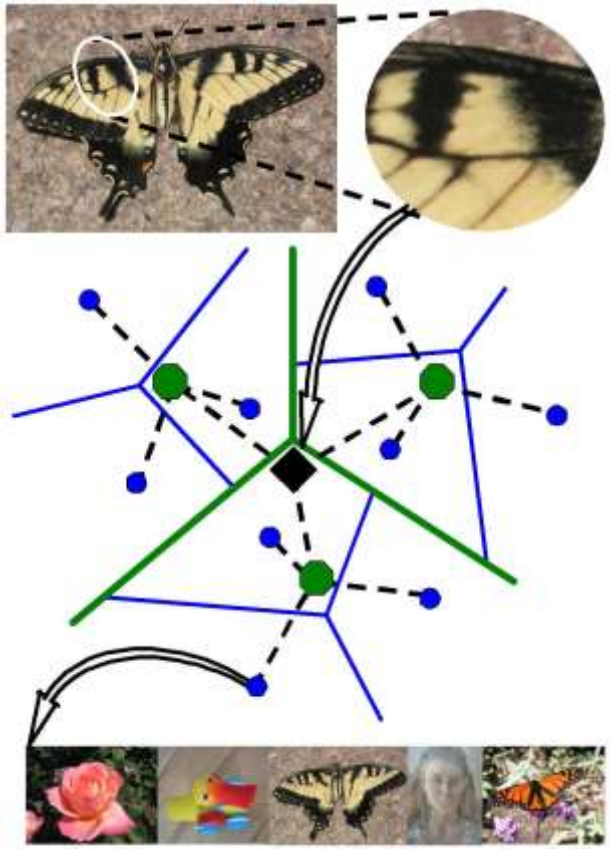
3. Построение «мешка слов»



- Соберём тем же способом фрагменты для выбранного изображения
- «Квантуем» каждый фрагмент
  - Определим, какому слову из словаря он соответствует (фактически, сопоставим номер от 1 до N, где N – размер словаря)
- Построим гистограмму частот фрагментов для изображения

Визуальные словари

- Как выбрать размер словаря?
  - Маленький: слова не могут описать все особенности
  - Большой: переобучение
- Вычислительная сложность
  - Деревья словарей (Nister & Stewenius, 2006)
  - Приближенные методы
  - Хеширование



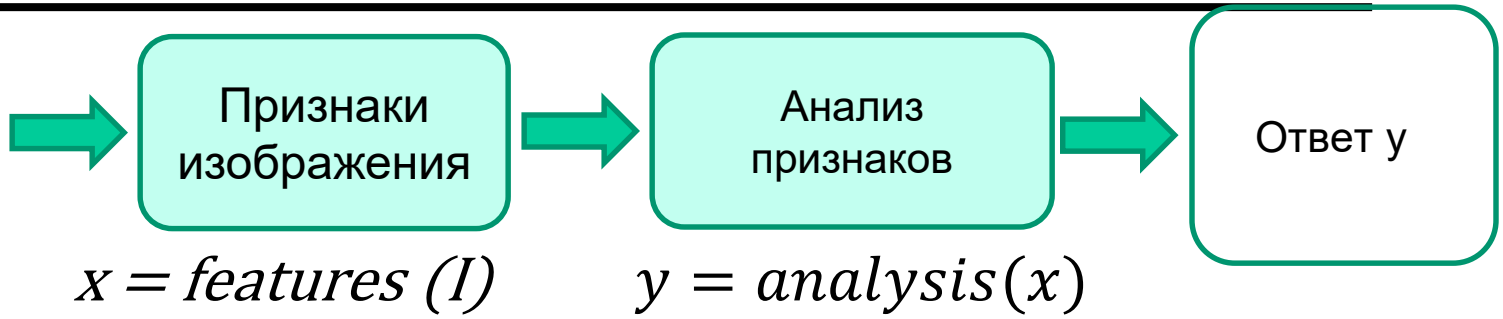
Резюме мешка слов

- Идея «мешка слов» оказалась очень интересной и эффективной в приложении к картинкам
- Началось с описания текстуры, и затем перешли к «визуальным» словам
- Мешок слов используется как сам по себе, так и совместно с другими признаками изображения
- Часто используются совместно и разреженная и плотная версии:
  - Мешок слов по характеристическим точкам
    - Bags of visual words
  - Плотные слова
    - Dense words (PhowGray, PhowColor)

- Реализация – библиотека VLFeat http://www.vlfeat.org/

Резюме классификации

Изображение I



$x = features\ (I)$ $y = analysis(x)$

- Задача классификации начинается с построения тестовой выборки, и затем обучающей
- «Классический» подход к классификации – какой-то из эвристических методов вычисления признаков, и затем построение классификатора с помощью машинного обучения по выборке
- Основные признаки - гистограммы цветов, HOG, BoW, LBP признаки, с использованием решеток и пирамид для учета пространственной информации

Поиск похожих изображений

- Запрос в виде изображения-примера («найди то же самое»)
- Классификация с открытым и заранее неизвестным набором классов

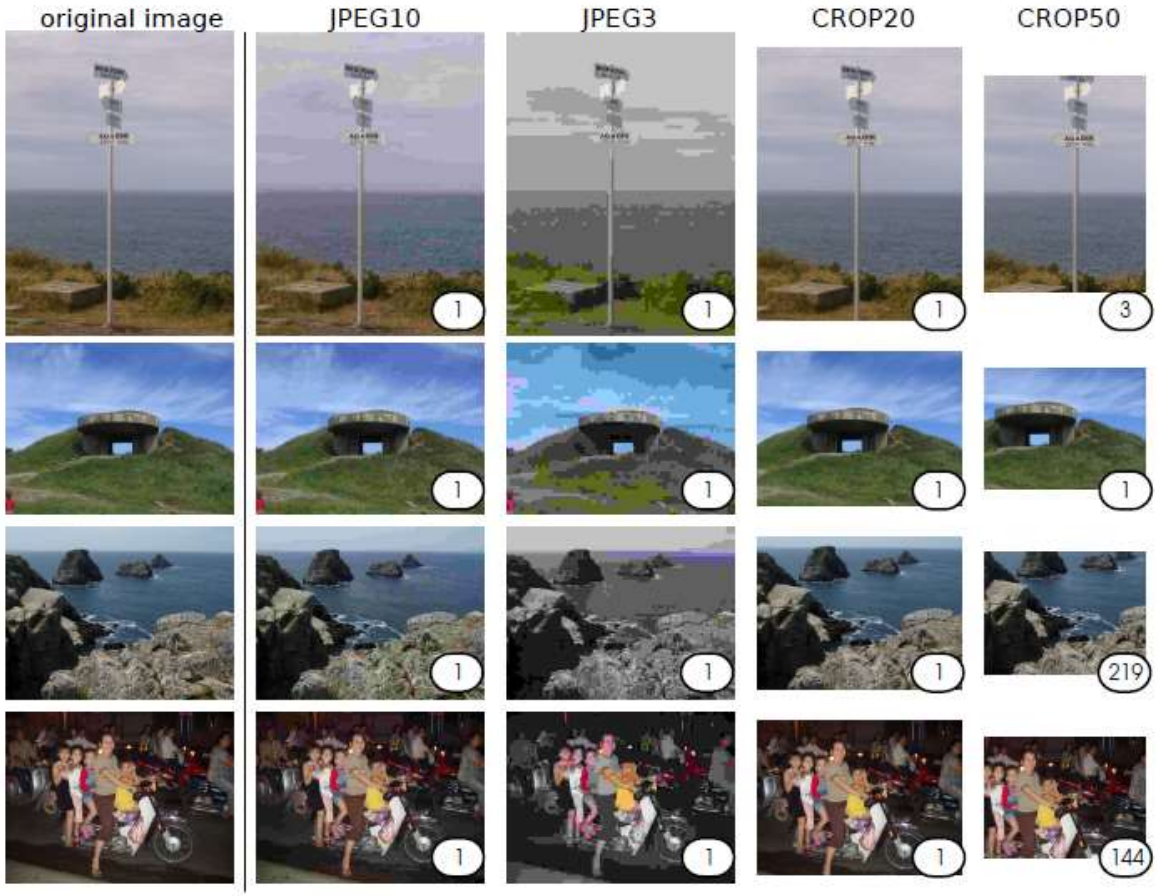
- Разные формулировки «сходства»

Визуальное подобие

Семантическое подобие

Пример – поиск полудубликатов

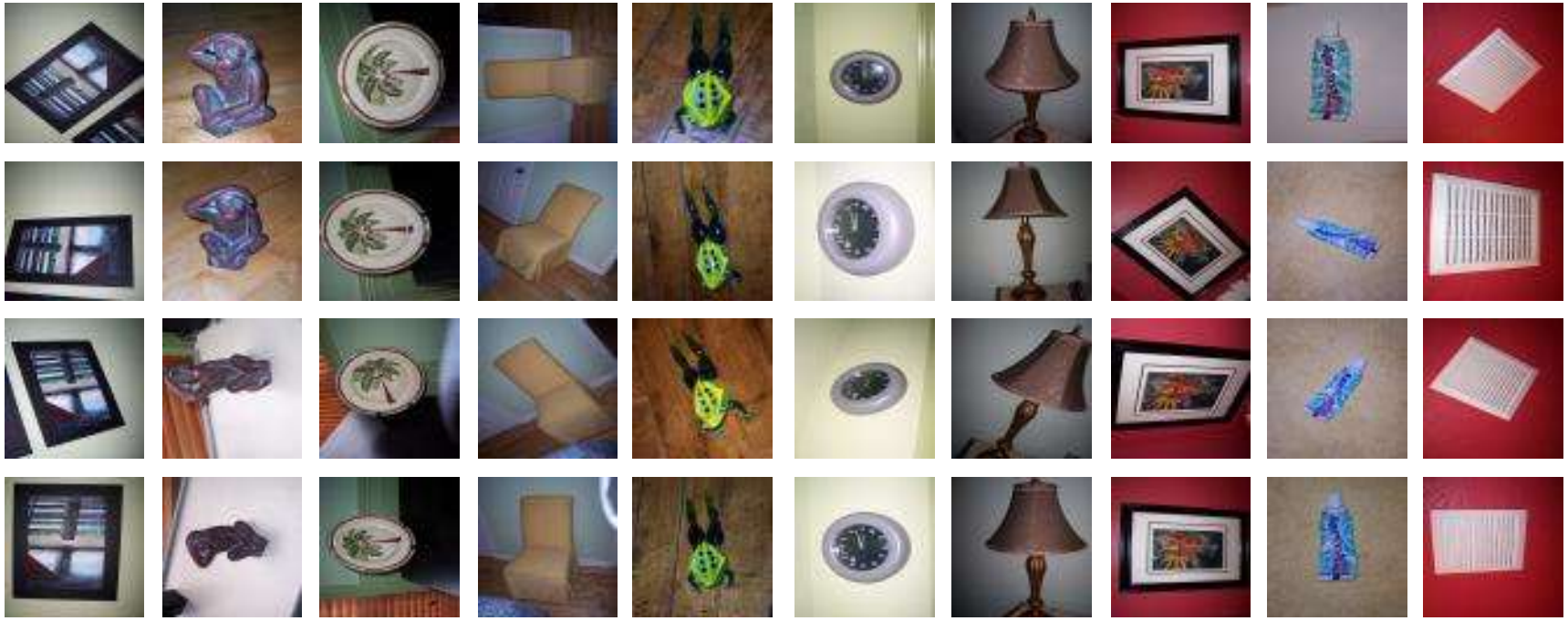
Полудубликаты (Near-duplicates) – слегка измененная версия изображения (ракурс, цвета)



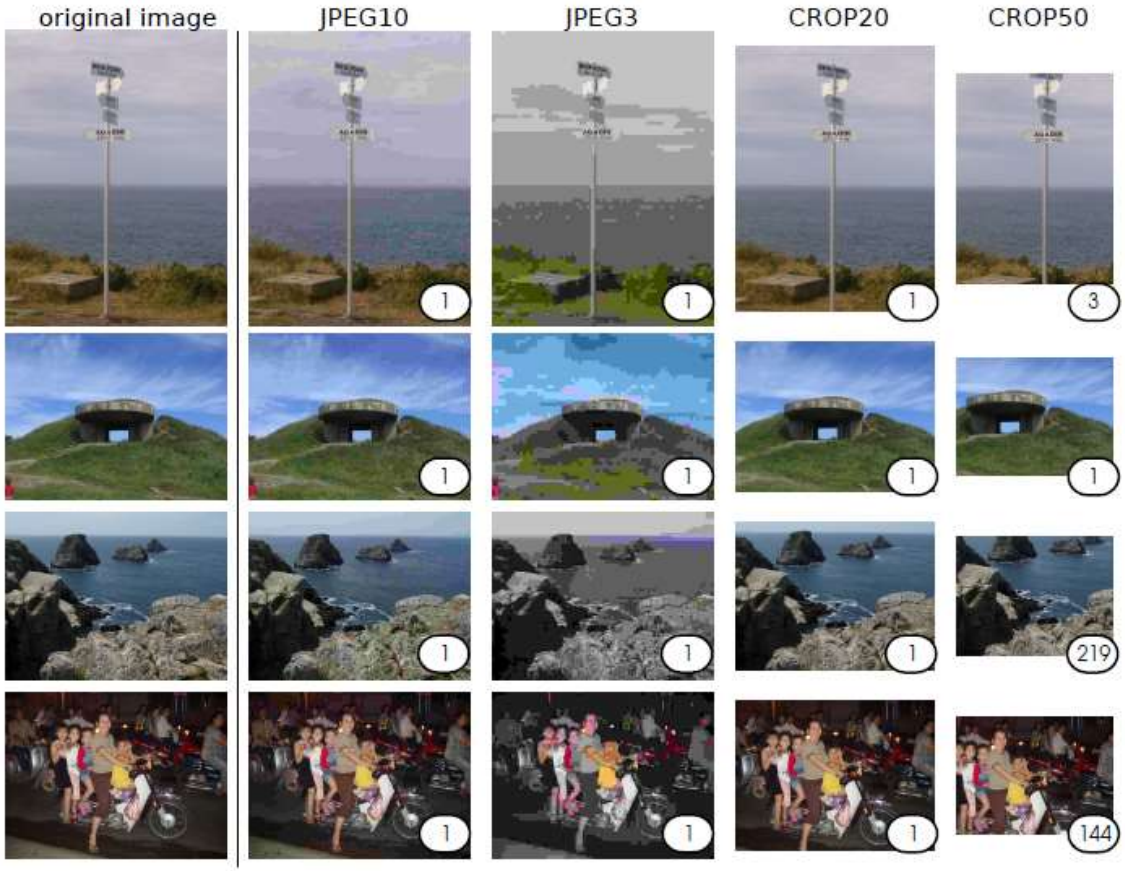
Object retrieval

Тот же самый объект или сцена

- Большие вариации ракурсов, фона, и т.д., чем при поиске полудубликатов



Copydays



- 157 исходных изображений
- Resize, crop, «сильные изменения» (print & scan, blur, perspective, etc)

Датасет Oxford Buildings dataset

- Oxford5K
  - 5062 изображения 1024x768 достопримечательностей (landmark) Окфорда
- 100K и 1M
  - коллекции изображений-дистракторов по самым популярным во flickr запросам

- 55 изображений запросов – по 5 изображений на 11 достопримечательностей
- По каждому запросу база размечена (Good – хорошо видно, OK - > 25%, Junk <25%, Absent – не видно)

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR, 2007.

Оценка точности

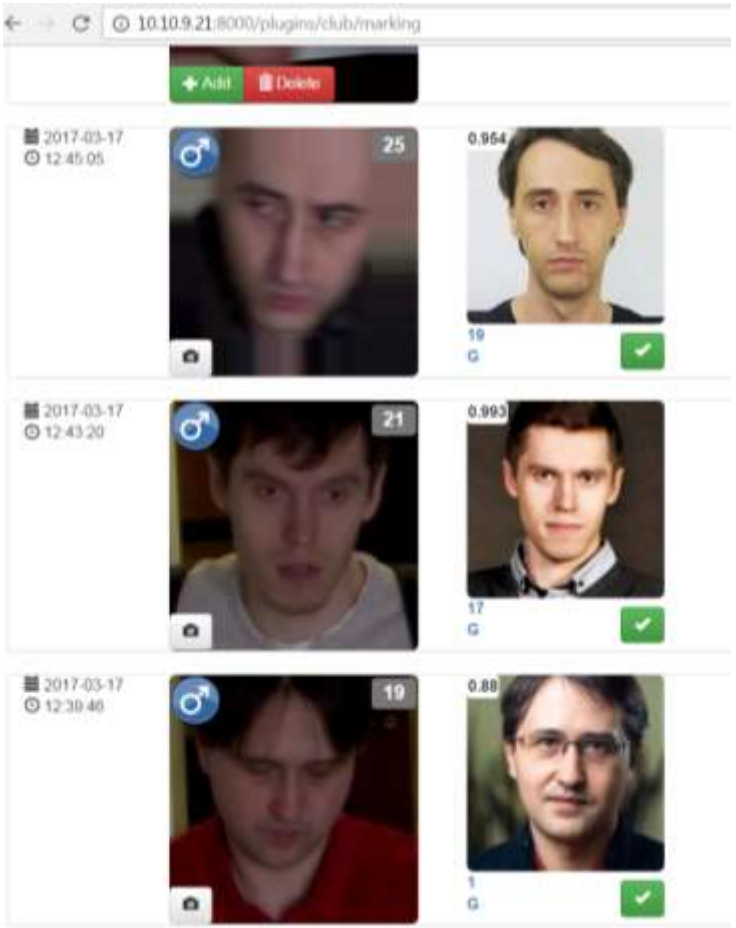
- AP – average precision для каждого запроса
  - Площадь под кривой precision-recall
  - Good и OK трактуются как положительные примеры
  - Absent - отрицательные
  - Junk – null, игнорируются при расчётах
- mAP – mean average precision
  - Усреднение по 5 запросам одной достопримечательности

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR, 2007.

Современные коллекции

- Сбор коллекций в полуавтоматическом режиме путем запросов к поисковой системе
- Миллионные размеры

Коллекции для распознавания человека



- Много вариантов:
  - Лицо
  - Радужка
  - Отпечаток пальцев
  - Фигура (и т.д.
- Тоже многомилионных размеров размеров и автоматическое построение
- У Гугла – 250 млн. изображений лиц

Общая схема поиска изображений

Коллекция изображений

Построение индекса коллекции

Вектор-признак изображения → Индекс коллекции

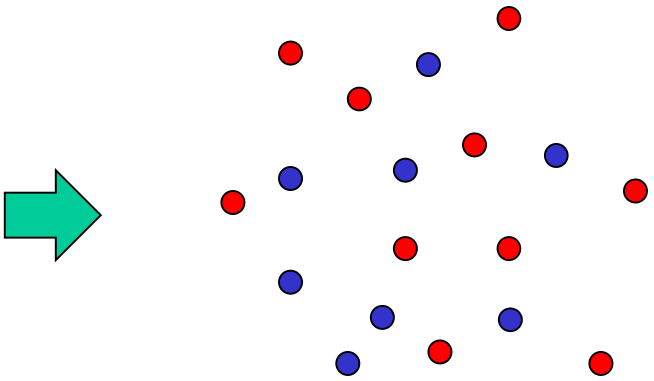
Изображение - запрос

Поиск изображения в коллекции

Вектор-признак → Поиск по индексу → Ранжируем список похожих

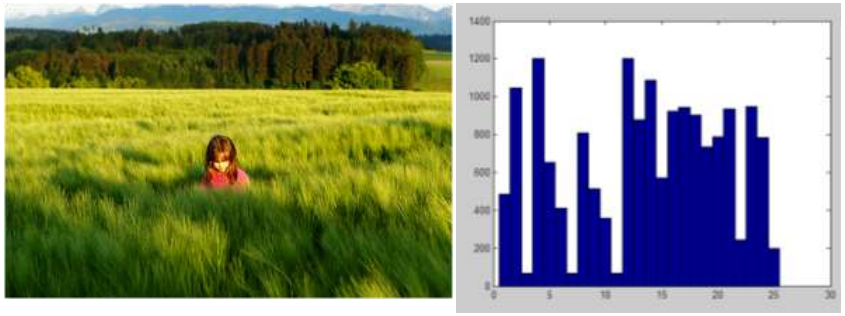
Ищем ближайших соседей по выбранной метрике

Память и производительность

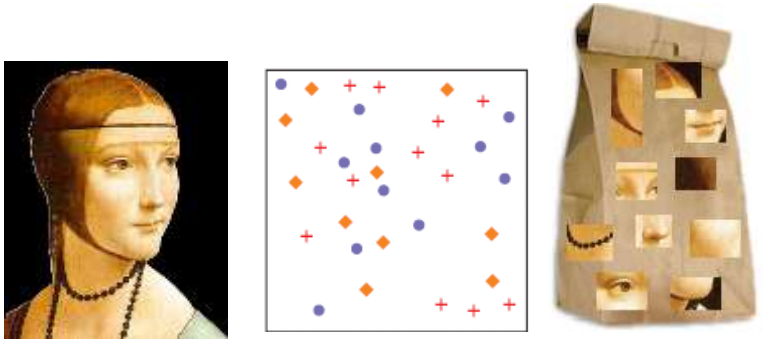


- Задача CBIR сводится к поиску ближайших соседей по дескриптору во всей коллекции (Nearest Neighbor)
- Простейшее и точное решение – линейный поиск
- Размер коллекции – миллионы и миллиарды векторов
- Нужны быстрые методы, пусть и с потерей точности
- Хранить все вектор-признаки в RAM накладно, поэтому нужно и вектор-признаки поменьше

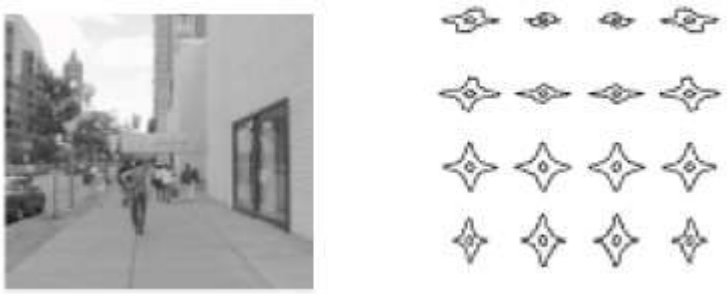
Дескрипторы изображений



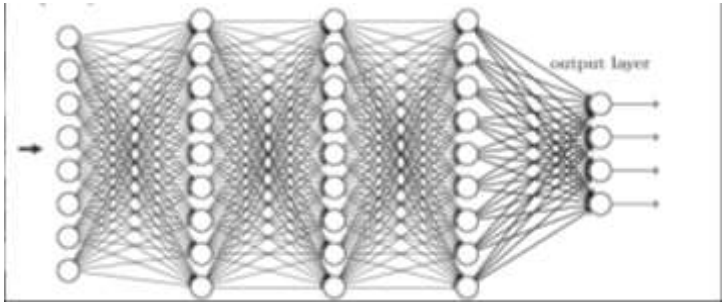
Гистограммы цветов (1995)



Локальные особенности
и мешок слов (2003)



Гистограммы градиентов
(2003)



Нейросетевые признаки (2013)

Гистограммы градиентов

- Насколько SIFT и HOG подходят для сравнения визуального подобия произвольных изображений?

- Как в этих методах учитывается масштаб/размер объекта?

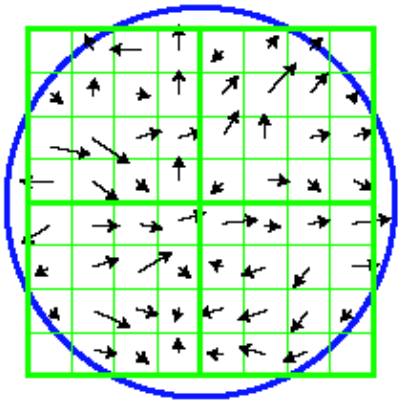
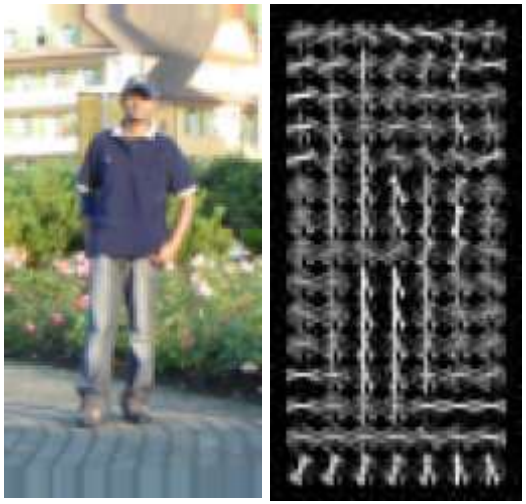


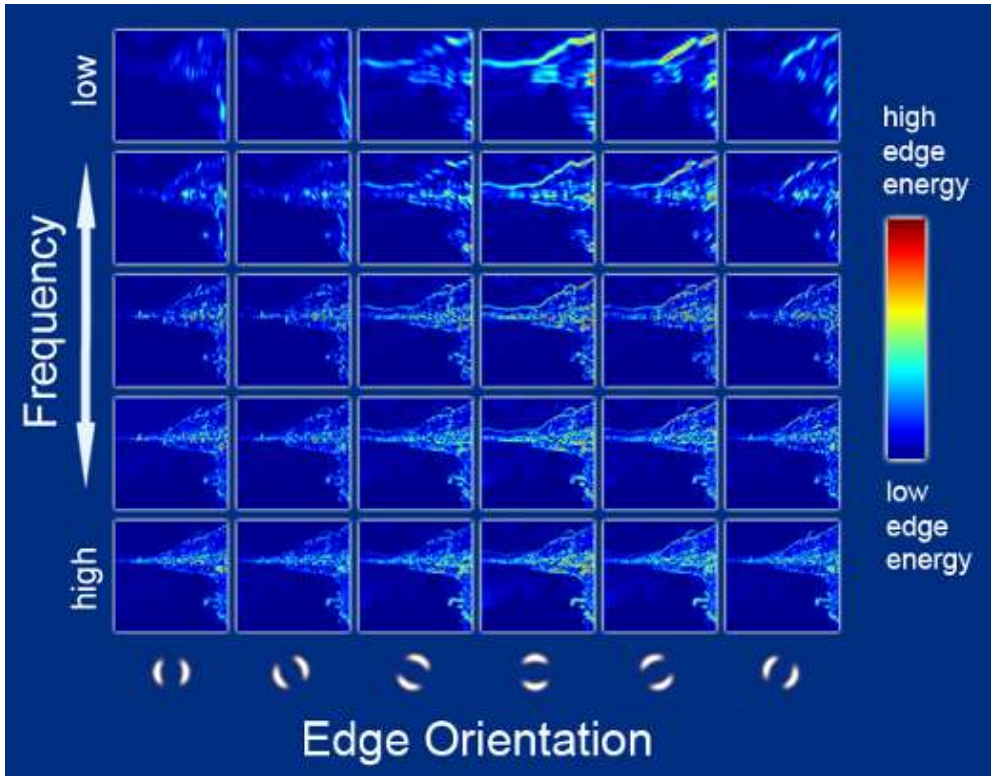
Image gradients

SIFT (2003)



HOG (2003)

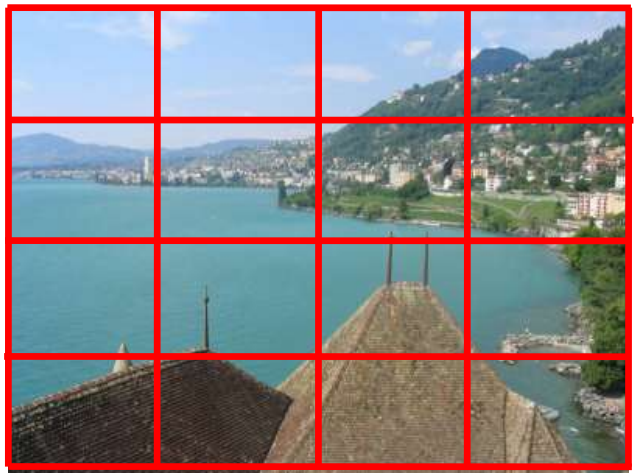
Дескриптор изображения GIST

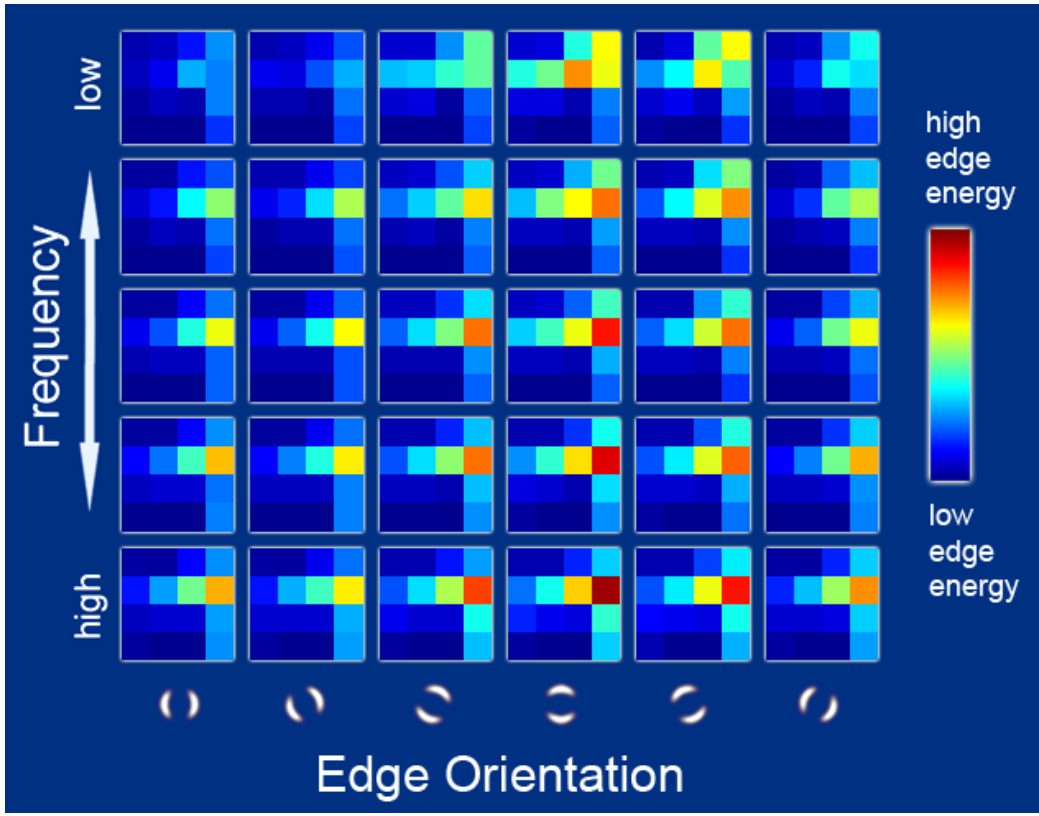


- Воспользуемся методом, похожим на SIFT.
- Посчитаем отклики детекторов краёв на 5 разных масштабах и 6 ориентациях края
- Получим 33 «канала» - RGB-цвет и 30 откликов фильтров края

TORRALBA, A., MURPHY, K. P., FREEMAN, W. T., AND RUBIN. Context-based vision system for place and object recognition. In ICCV 2003

Дескриптор изображения GIST

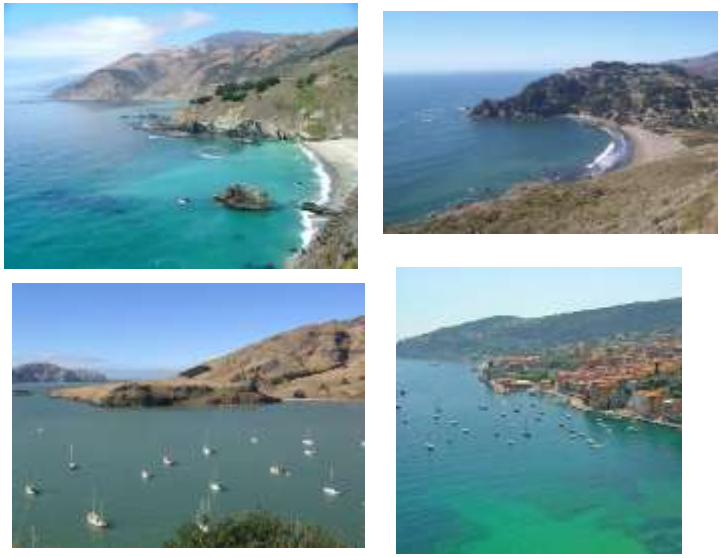




- Разобьём изображение сеткой 4x4 на 16 ячеек
- В каждой ячейке усредним значения всех каналов
- Получим дескриптор GIST

Применение GIST

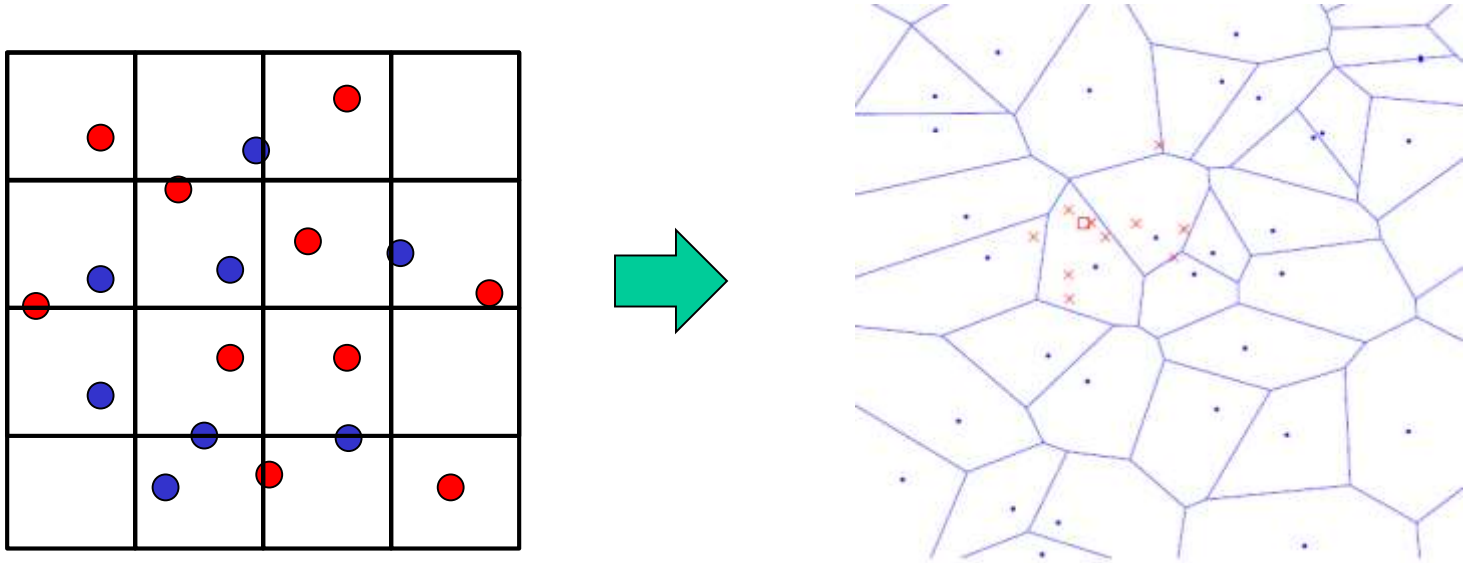
Запрос



Похожие по GIST + цвету

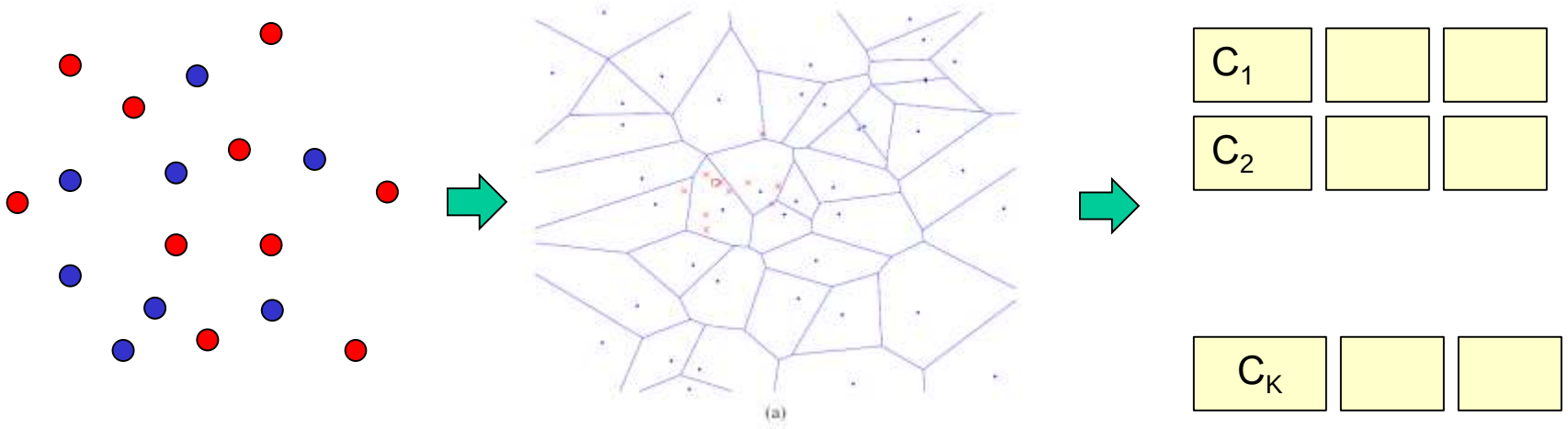
- Типичный размер – 8 ориентаций x 4 масштаба x 16 ячеек = 512 параметров (2 кбайта при float)
- Можно считать по маленьким изображениям от 32 до 128 пикселов
- GIST хорошо себя показал для поиска полудубликатов
- Но искать линейным поиском по всей коллекции всё равно слишком долго

Квантование векторов



- «Vector quantization» – способ ускорения поиска за счёт уменьшения точности
  - Отображение x на q(x), где q(x) in $\{c_i\}$, c_i – центроиды, i от 1 до K
- Индекс q(x) можно записать в виде битового кода длиной $\log_2(K)$
- Фактически сжимаем x в код q(x)
- Можно воспользоваться K-средними для разбиения на группы похожих векторов

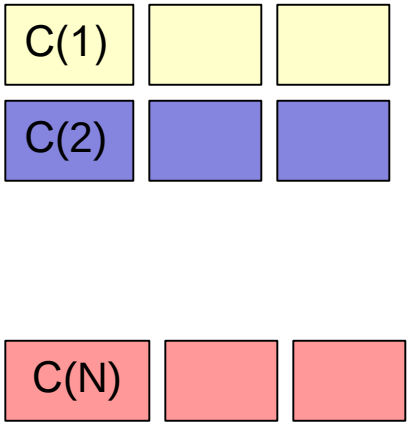
Инвертированный индекс

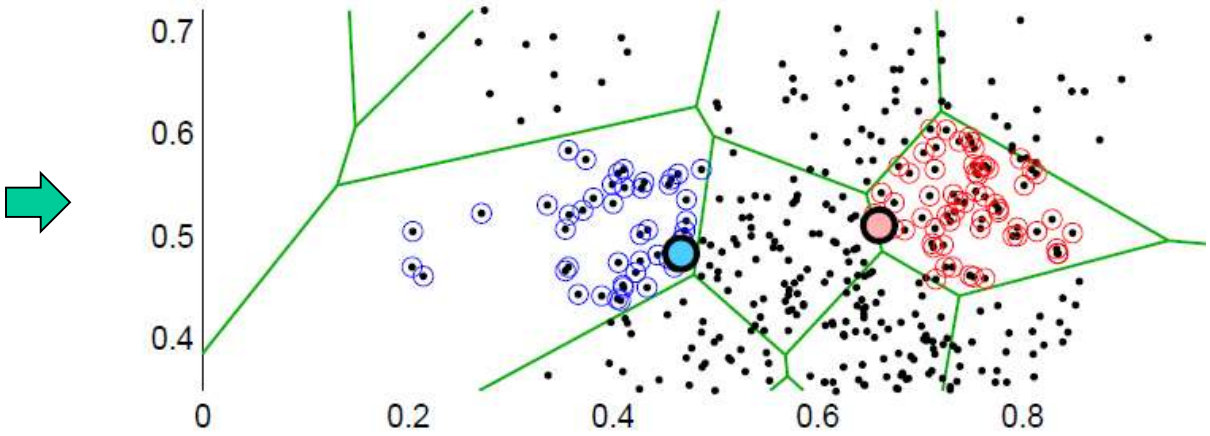


Квантованные вектора удобно записывать в форме «инвертированного индекса»

- K списков по числу центроидов (кодовых слов)
- В каждом списке храним индексы всех векторов, квантованных до C_i

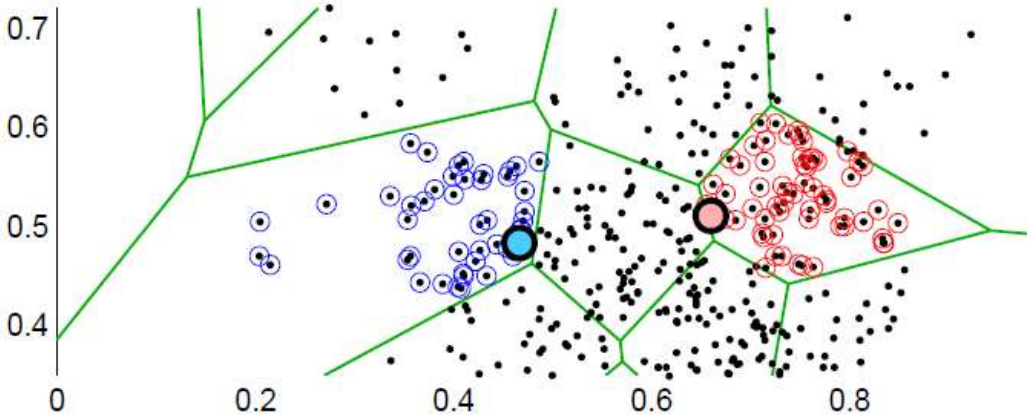
Инвертированный индекс





- Поиск по индексу:
  - Просмотрим инвертированный индекс, найдём ближайший кластер
    - Или группу ближайших кластеров
  - Все элементы в списке – ближайшие вектора (приближенно)

Ранжирование результатов



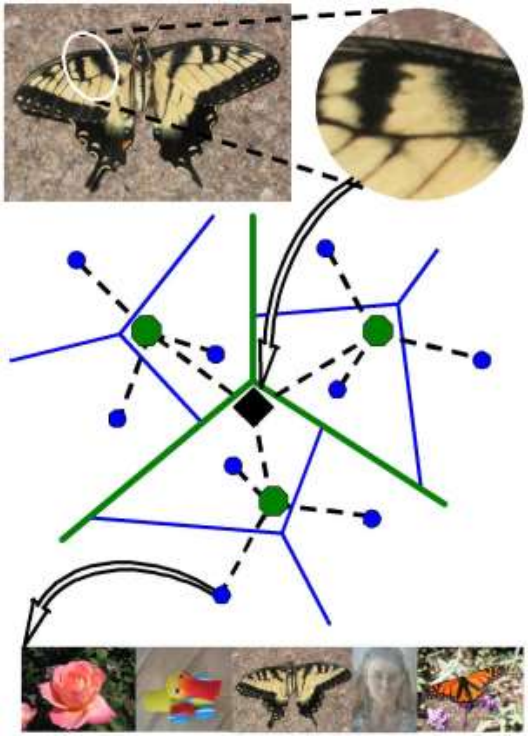
- Зачем выдавать все элементы в списке из одного кластера неупорядоченно?
- Упорядочим результаты (Re-ranking)
  - Рассчитаем расстояния от вектора-запроса до каждого элемента списка по полному дескриптору
  - Упорядочим результаты по близости (первые – ближайшие)
- Есть и другие способы ранжирования!

Проблемы квантования K-средними

- Чем крупнее кластеры, тем меньше точность аппроксимации
- Для повышения точности требуется существенно увеличивать число кластеров K
- Квантование до кодов 64 бита требует подсчёта и хранения 2^{64} центроидов, что невозможно
- Сложность одного этапа O(NK)
  - Медленно на больших выборках (большие N)
  - Медленно при больших K
- Поэтому нужны другие, более быстрые приближенные методы.

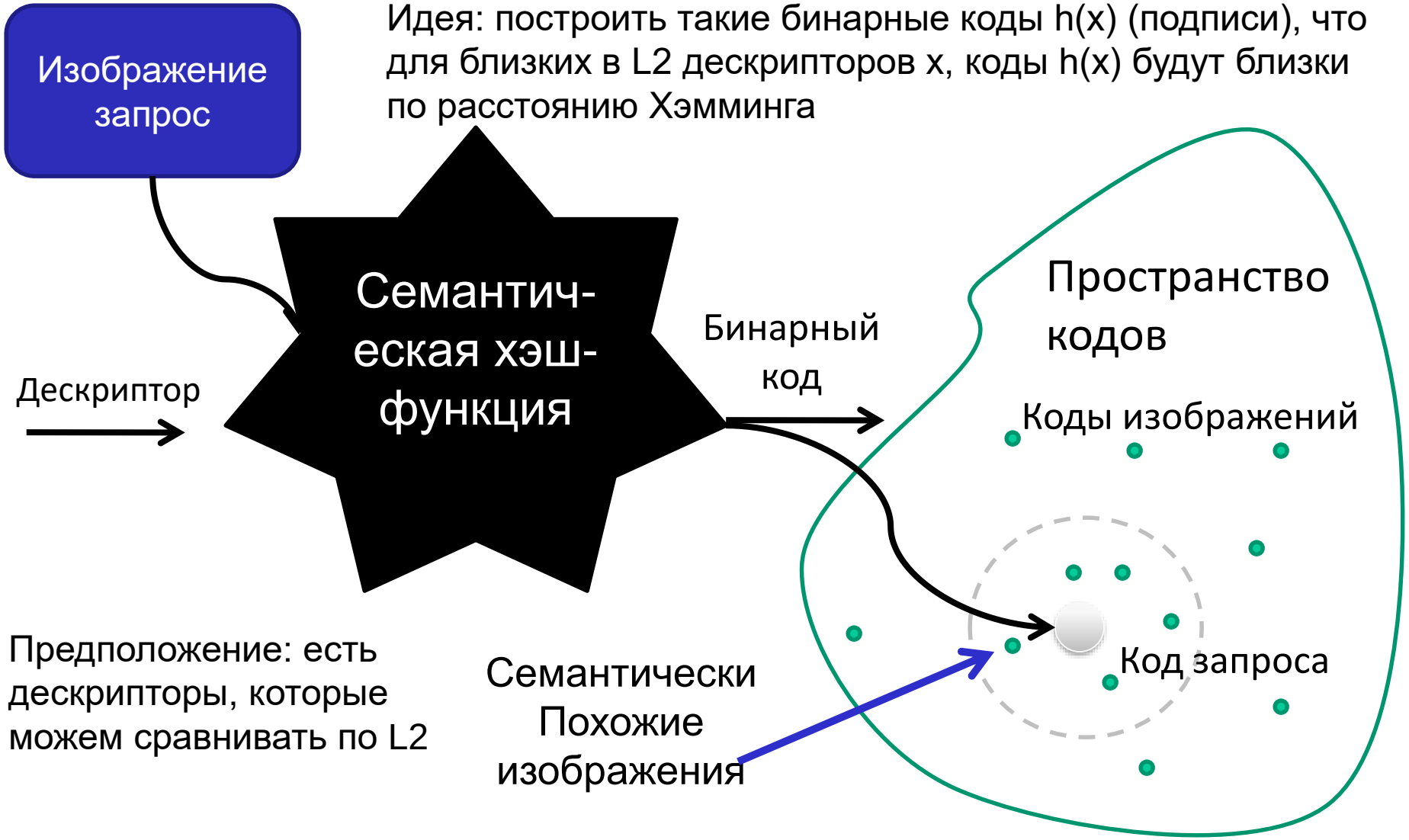
Hierarchical k-means (HKM)

- Иерархическое разбиение
  - Кластеризуем всё на K кластеров (K=10) с помощью K-средних
  - Затем данные в каждом кластере снова разбиваем на K кластеров
  - Повторяем до достижения нужной глубины
- Пример:
  - Глубина 6 даёт 1M листьев
- По точности проигрывает существенно K-средним



D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In Proc. CVPR, 2006.

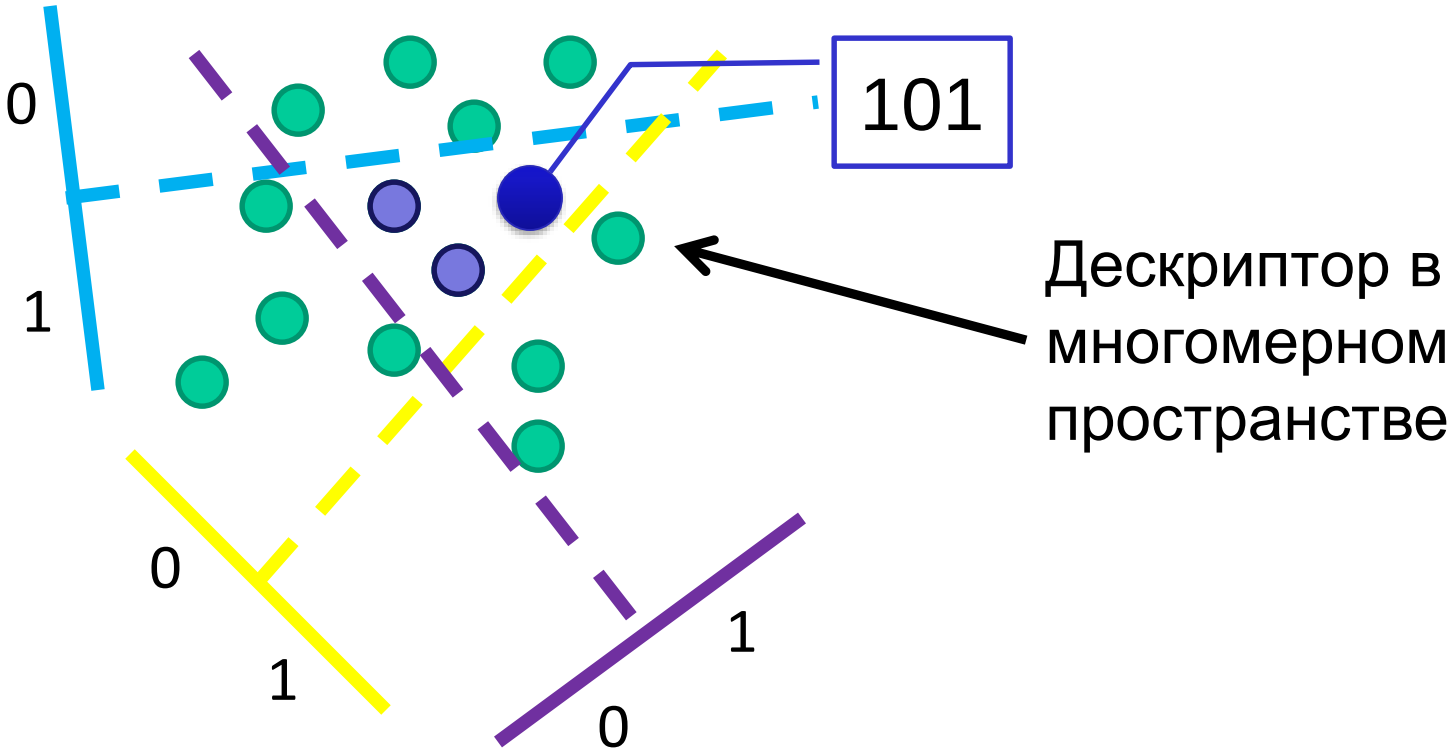
Семантическое хеширование



R. R. Salakhutdinov and G. E. Hinton. Semantic hashing. In SIGIR workshop on Information Retrieval and applications of Graphical Models, 2007.

Locality Sensitive Hashing (LSH)

- Возьмем случайную проекцию данных на прямую
- Выберем порог близко к медиане проекций на прямую
- Пометим проекции 0 или 1 (1 бит подписи)
- С увеличением числа бит подпись приближает L2-метрику в исходных дескрипторах



A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In FOCS, 2006

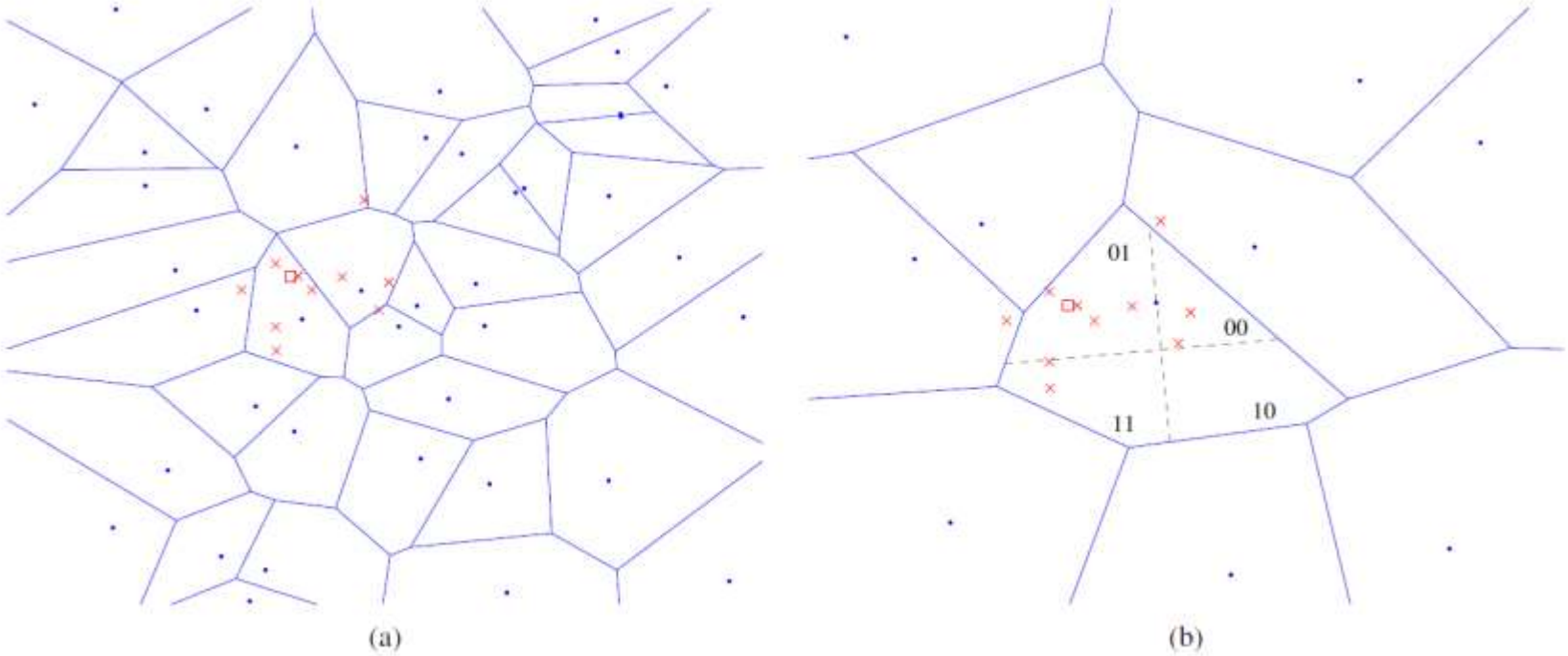
Locality Sensitive Hashing (LSH)

- Плюсы
  - Приближенный способ вычисления ближайшего соседа
  - Быстрое квантование (быстро вычисляем бинарный код)

- Недостатки:
  - Приближение L2 лишь асимптотическое
  - На практике может потребоваться слишком много бит для подписи

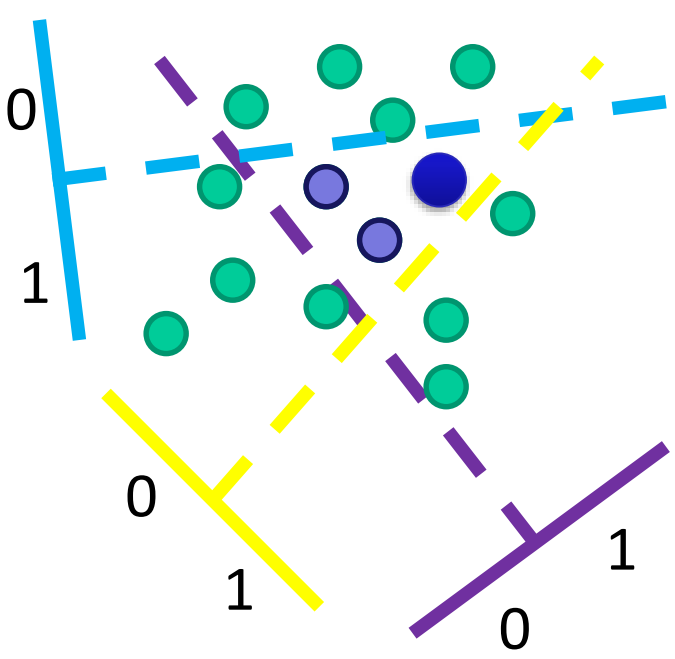
- Вывод:
  - Использовать как замену K-средних нельзя
  - Можно использовать в дополнение

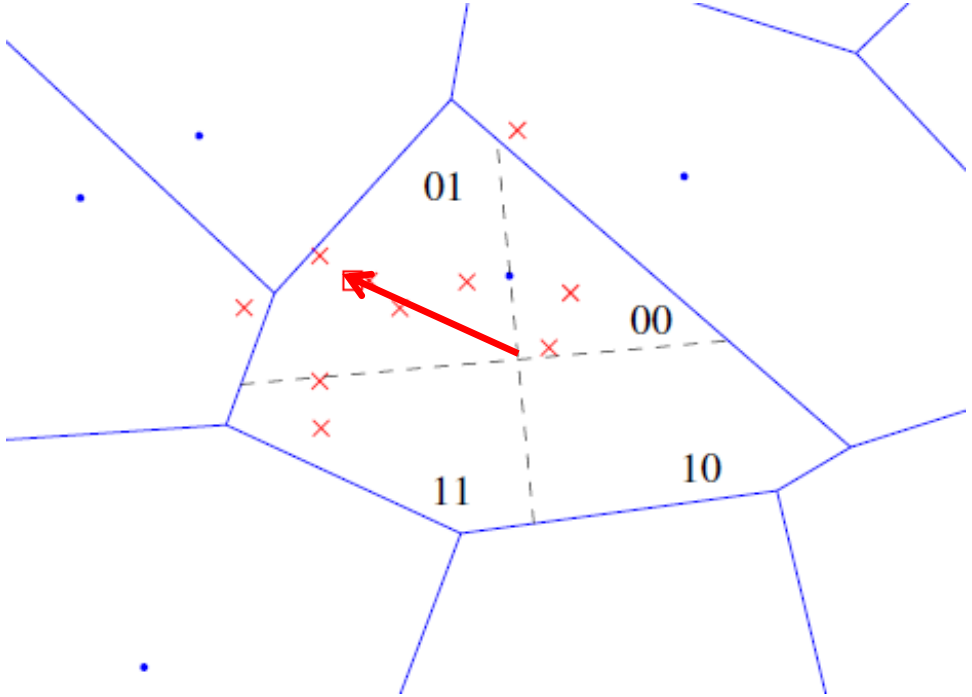
Проблемы K-средних



- Маленькое K – большие ячейки
  - Слишком грубый порог на сравнение!
- Большое K – маленькие ячейки
  - Медленнее кластеризация и квантование
  - Ячейка может оказаться слишком маленькой, не все соседи попадут

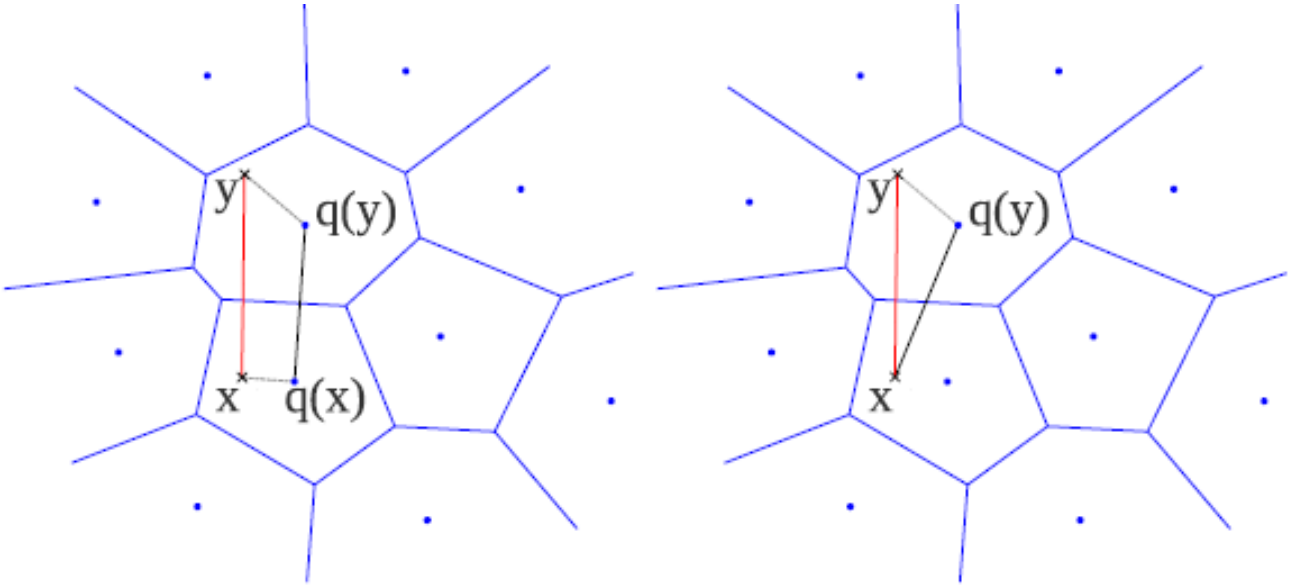
Hamming Embedding





- Идея – закодируем положение нашей точки (дескриптора) внутри ячейки диаграммы Вороного (т.е. вектор x – q(x)) с помощью семантического хеширования, построив код b(x)
- Получим аппроксимацию x как vec(q(x)) + vec(b(x))
- Для каждой ячейки можем построить с помощью LSH своё отображение в коды

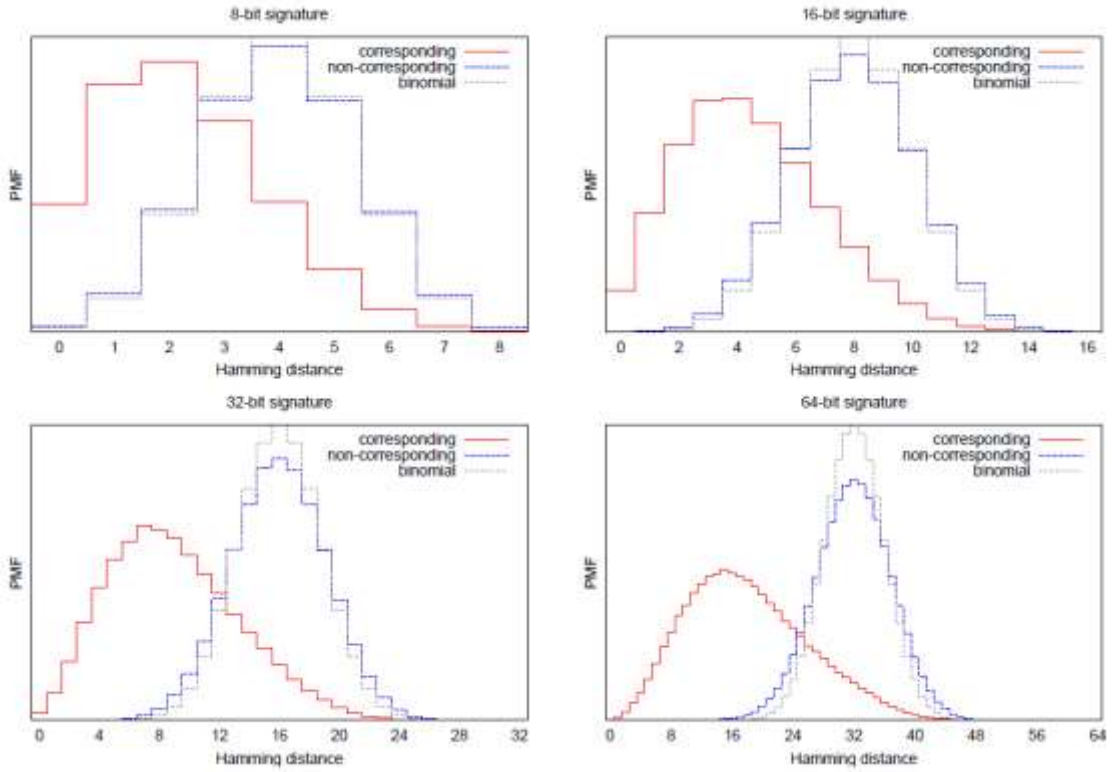
Варианты сравнения векторов



symmetric case asymmetric case

- Симметричное сравнение:
  - (q(x) + b(x)) vs (q(y) + b(y)) – квантуем и запрос и записи в базе
- Ассиметричное сравнение
  - (x) vs (q(y) + b(y)) – т.е. исходный вектор-запрос с реконструированной версией закодированного вектора из базы
  - Ассиметричный вариант оказывается точнее

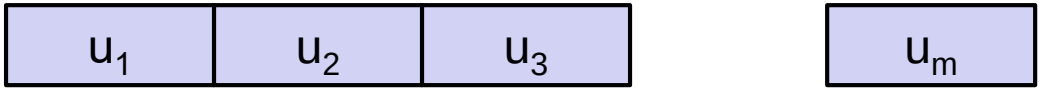
Влияние кодов



- Посмотрим распределение расстояний Хэмминга между правильными и неправильными парами
- Сравнение по кодам показывает, что при длине кода > 32 бит наблюдается заметная разница между правильными и неправильными сопоставлениями

Product quantization

- Вспомним:
  - Квантование до кодов 64 бита (0.5 бита на параметр) требует подсчёта и хранения 2^{64} центроидов, что невозможно
- Простая идея:
  - Разобьём вектор x длины D на m частей
  - Квантуем каждый подвектор u_j независимо от остальных



  - Пусть каждый подвектор квантуем на k* центроидов, тогда всего центроидов $(k^*)^m$
  - Длина кода $l = m \log_2 k^*$
- Сравнение с K-средними по памяти
  - Память: kD (K-средние) и $mk^*(D/m)=k^{1/m}D$

H.Jégou et. al. Product quantization for nearest neighbor search, PAMI 2011

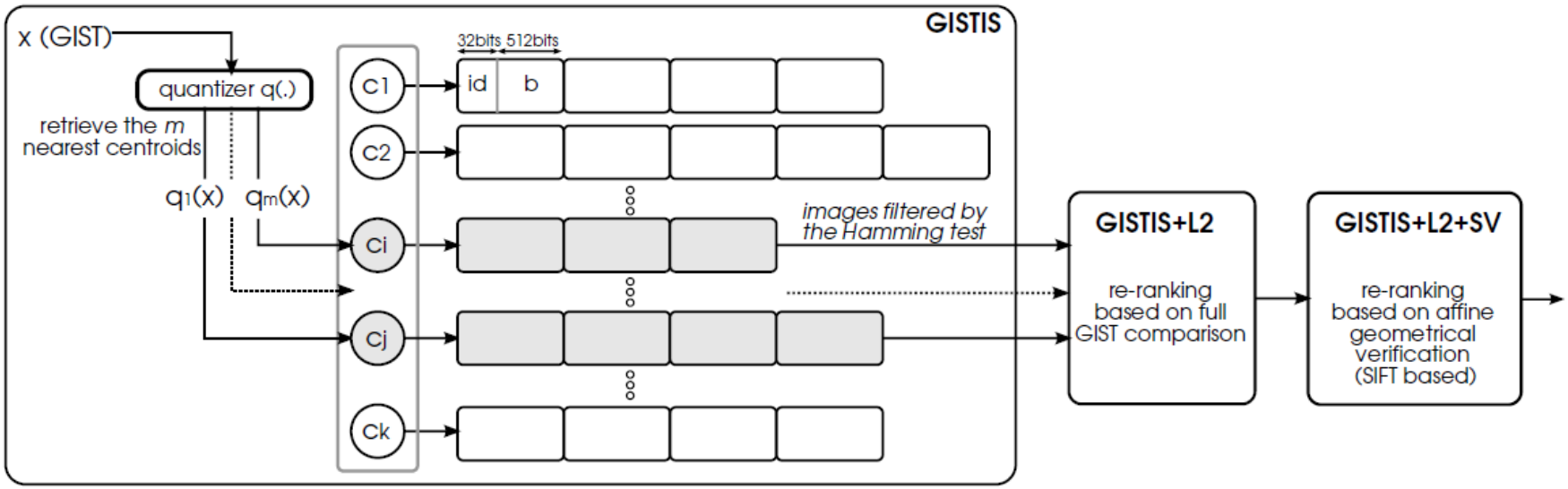
Алгоритм GISTIS

- GIST Indexing Structure
  - Добавим в наш простой алгоритм для поиска полудубликатов hamming embedding

- Схема метода:
  - Строим GIST для каждого изображения
  - Кластеризуем все дескрипторы с помощью k-means на k=20000 кластеров
  - Применяем LSH / PQ к каждому кластеру и для всех векторов в кластере считаем бинарную подпись длиной 512 бит
  - Идентификатор картинки (64 бит) и бинарная подпись (512 бит) хранится в индексе в RAM
  - Остальная информация (GIST и т.д.) хранится на диске и считывается по необходимости

M.Douze et.al., Evaluation of gist descriptors for web-scale image search. In International Conference on Image and Video Retrieval. ACM 2009.

Схема метода



- Поиск (фильтрацию) проводим в несколько стадий:
  - Выбираем 200 ближайших кластеров (при идеальном распределении 1% картинок, на практике 2.3% из-за неравномерности кластеров), получаем q1(x),…q200(x)
  - Считаем бинарные коды b1(x), …, b200(x)
  - Фильтрация по бинарным кодам с порогом h=220 (отбрасываем 94%)
  - Итого возвращаем 0.13% картинок, отсортированных по расстоянию Хэмминга

Результаты

	Байт на изображение в RAM	Время на построение дескриптора	Время поиска в базе из 110M изображений
GIST	3840	35мс	1.26с
GISTIS	68	36мс	2мс
GISTIS + L2	68	36мс	6/192мс

- Лучшие изображения мы можем дополнительно отсортировать по всему GIST (L2- метрика)
- Поскольку GIST приходится читать с диска, то сортируем только 200 топовых ответов

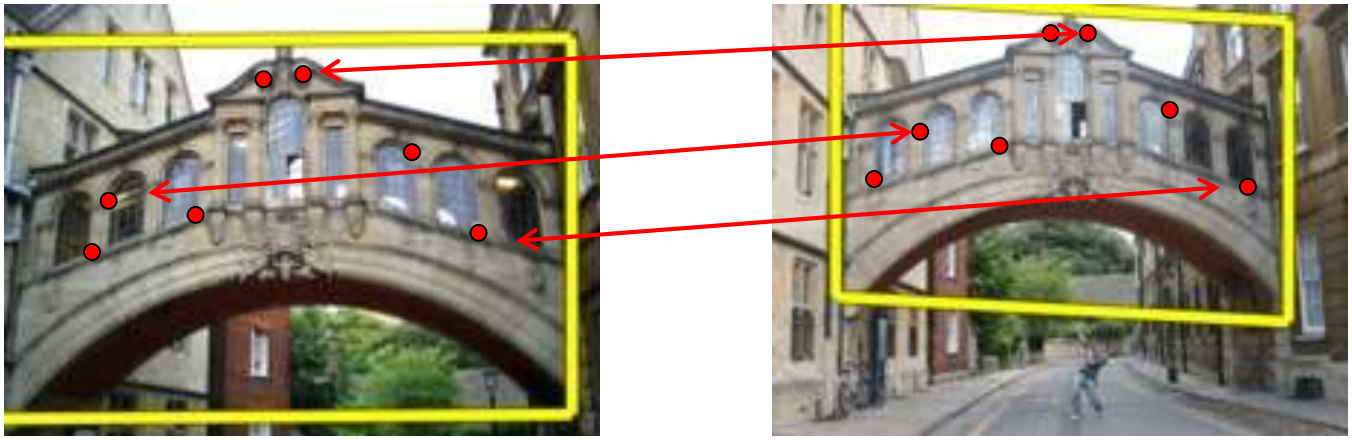
M.Douze et.al., Evaluation of gist descriptors for web-scale image search. In International Conference on Image and Video Retrieval. ACM,2009.

Object Retrieval

- Ищем изображения одного и того же объекта

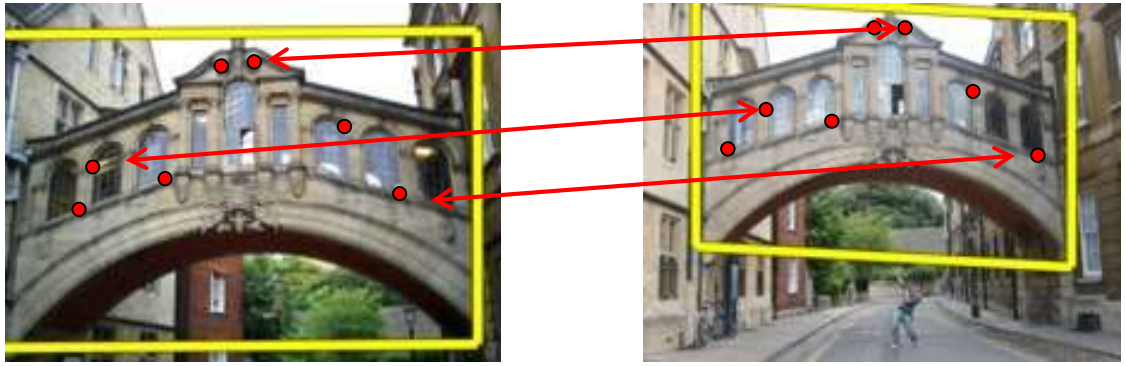
- Как с этой задачей справится GIST?
- Не очень хорошо, т.к. размеры и ориентация объектов могут значительно меняться
- Может помочь сопоставление изображений по ключевым точкам

Геометрическое сопоставление



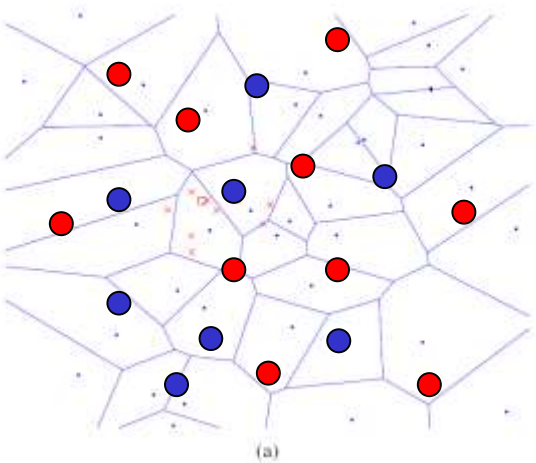
- Построим «дескриптор изображения» как набор найденных точек SIFT с дескрипторами
- Визуальное подобие будем оценивать через сопоставление двух изображений по набору SIFT
  - Найдём соответствующие пары точек по дескрипторам
  - Чем больше соответствий, тем более похожи два изображения
  - Можем оценить геометрическое преобразование с помощью RANSAC и отфильтровать ложные

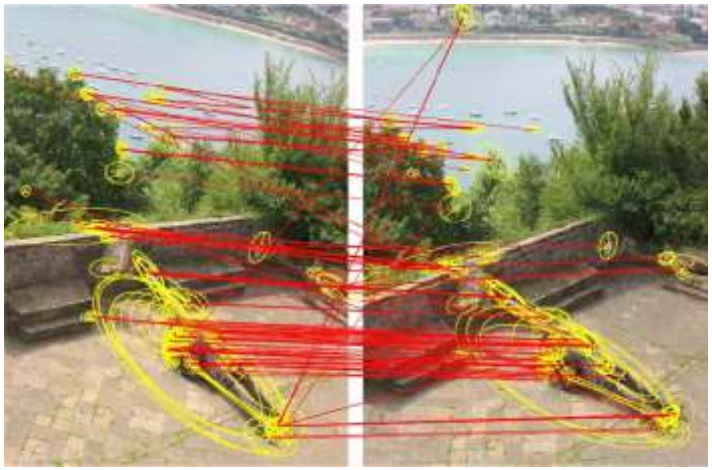
Геометрическое сопоставление



- Самая тяжелая операция – поиск ближайшего (NN) для сравнения ключевых точек
- Можем решать её приближенно, за счёт квантования дескрипторов

Мешок слов

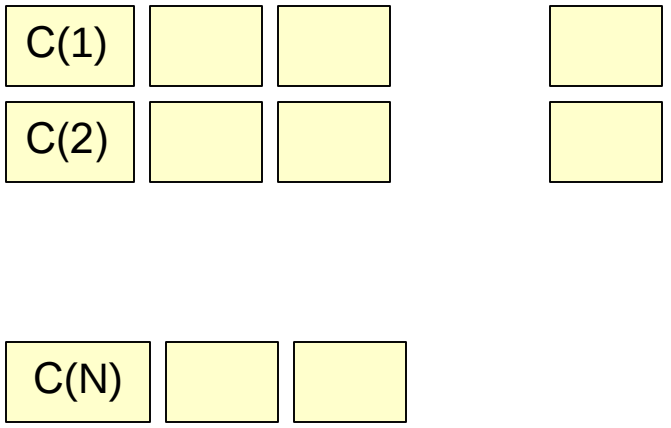




- Построим «словарь» ключевых точек (K-средние для выборки дескрипторов)
- Теперь для поиска пар соответствующих точек нужно сравнить их индексы по словарю
- Количество пар соответствующих точек («putative correspondences») можно вычислить как пересечение гистограмм (histogram intersection) для 2х «мешков слов»!
- Получаем аппроксимацию числа соответствующих точек
- Вектор-признак изображения = мешок слов (BoW)

Инвертированный индекс

- Построим инвертированный индекс для мешка слов / квантованных особенностей
  - Таблица (слова)x(изображения)
  - Список слов в словаре (терминов)
  - Для каждого слова храним список изображений, в котором слово встречается



- Эффективность
  - Если словарь большой, то большая часть элементов мешка слова нулевая
  - Храним мы только ненулевые значения
  - Самые «употребительные слова» ставим в начало списка для ускорения поиска

Пример объёмов данных

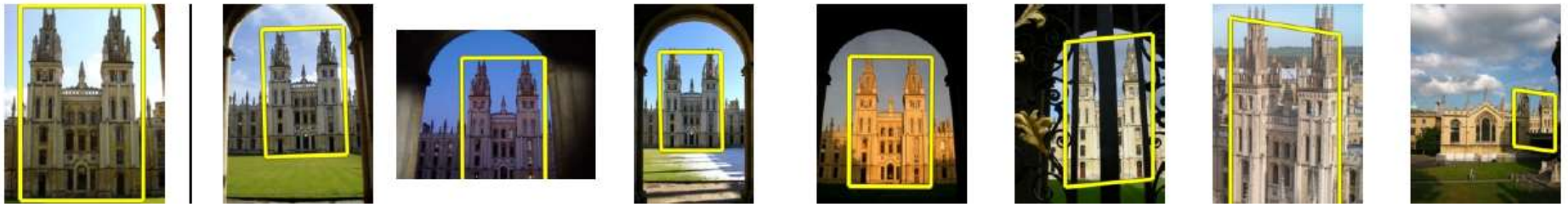
- Посчитаем SIFT для ключевых точек из большой коллекции изображений

Dataset	# images	# features	Size of descriptors
5K	5,062	16,334,970	1.9 GB
100K	99,782	277,770,833	33.1 GB
1M	1,040,801	1,186,469,709	141.4 GB
Total	1,145,645	1,480,575,512	176.4 GB

- Понятно, что на таких объёмах данных обычные k-средних не сработают, нужно применять приближённые методы поиска ближайших соседей внутри k-средних
- Approximate K-Means

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR, 2007.

Поиск по «мешку слов»



Базовый алгоритм:

- Дескриптор «мешок слов» большой размерности (1M)
- Приближенный метод кластеризации для построения словаря по части коллекции (5к)
- Инвертированный индекс для хранения

- Тестирование:
  - 5к+100к изображений, 1M слов, 1GB индекс, поиск в нём 0.1с
  - 5к+100к+1M изображений, 1M слов, 4GB+, хранение файла на диске, поиск 10-35с

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR, 2007.

Резюме поиска похожих изображений

- Задача классификации с открытым набором классов сводится к построению индекса, в котором хранятся вектор-признаки изображений, и поиска ближайшего соседа
- Для построения вектор-признаков можем использовать те же методы построения признаков, что и для классификации
- Для ускорения поиска по большим коллекциям применяем «умное» квантование

Школа Анализа
Данных Яндекса

Факультет
компьютерных наук
НИУ ВШЭ

Лаборатория компьютерной графики
и мультимедиа ВМК МГУ

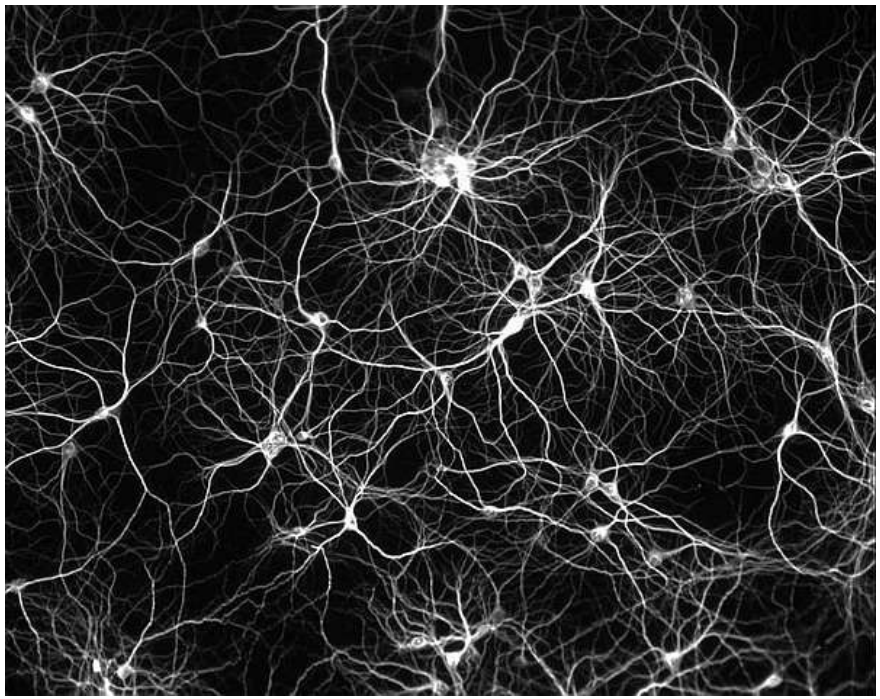
Курс «Компьютерное зрение»

Лекция №5
«Введение в свёрточные нейросети»

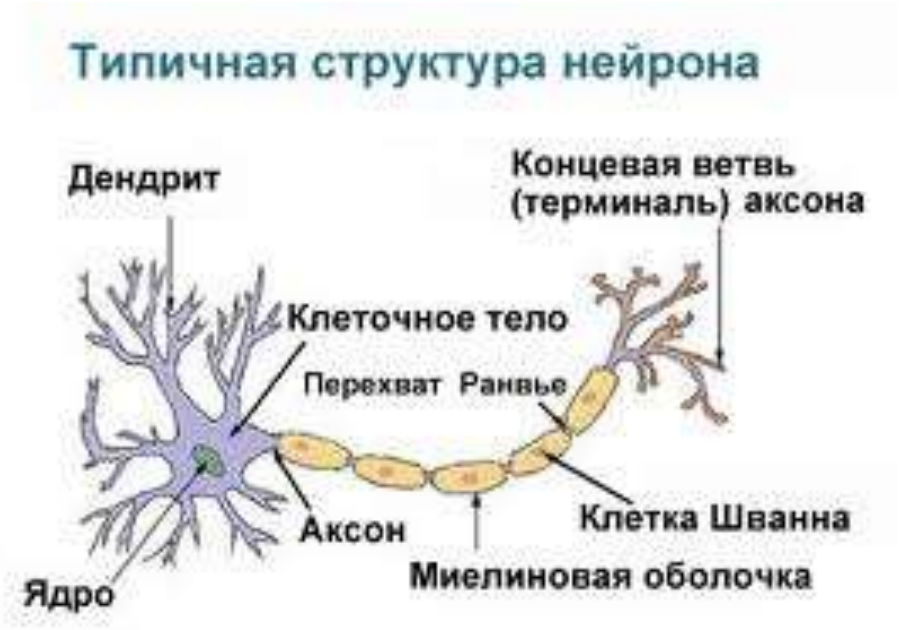
Антон Конушин

7 октября 2019 года

Структура мозга человека

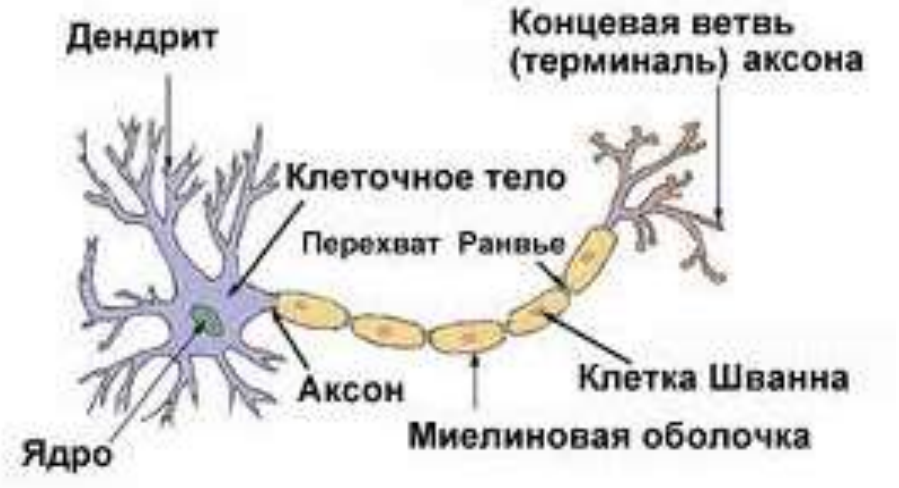


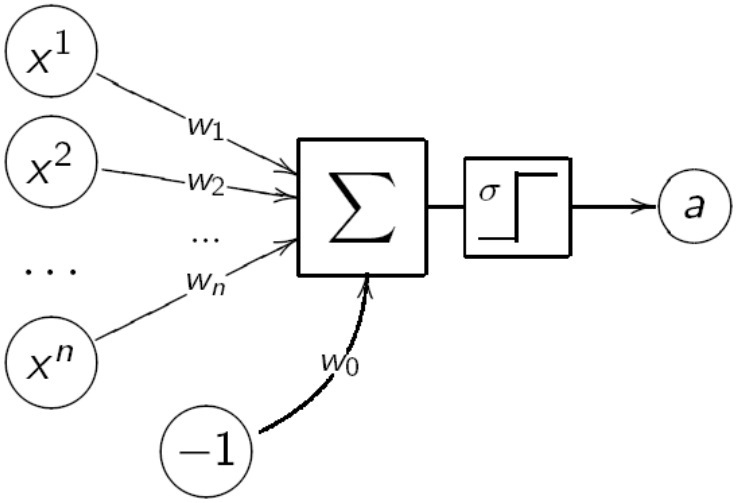
Нейросеть



Отдельный нейрон

Линейная модель МакКаллока-Питтса





$$a(x, w) = \sigma\big(\langle w, x\rangle\big) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right),$$

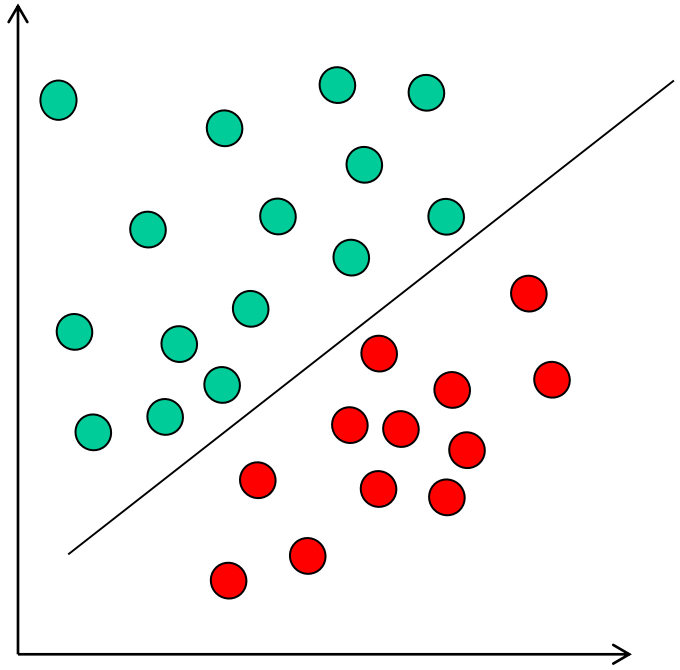
$\sigma(z)$ — функция активации (например, sign),

w_j — весовые коэффициенты синаптических связей,

w_0 — порог активации,

$w, x \in \mathbb{R}^{n+1}$, если ввести константный признак $f_0(x) \equiv -1$

Нейрон как линейный классификатор



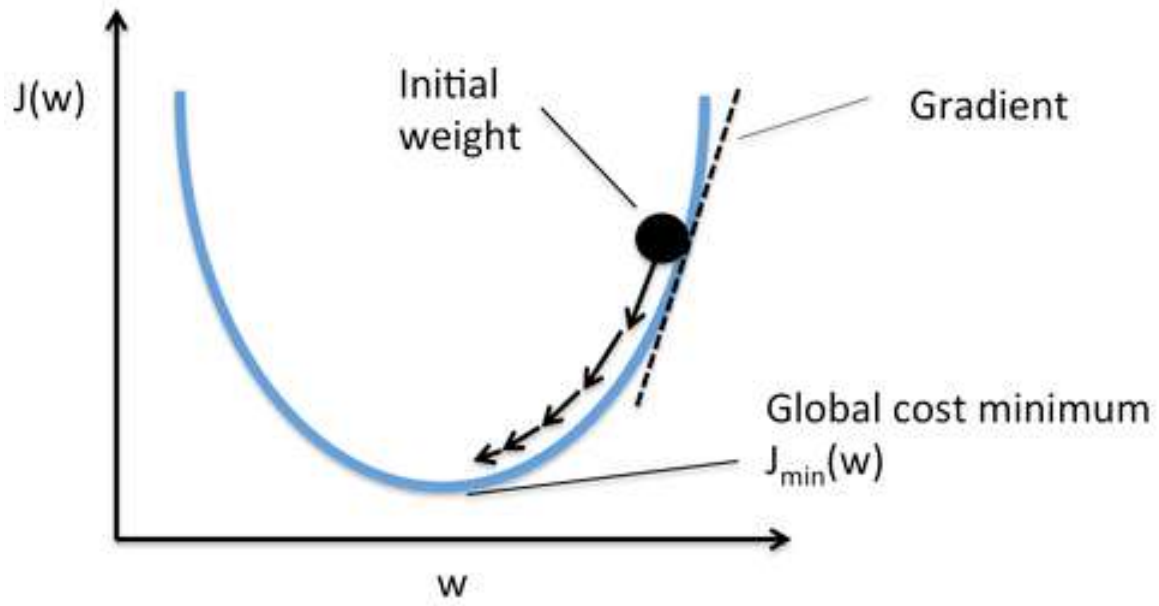
$$a(x, w) = \text{sign}\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right)$$

$\mathbf{x}_i$ *положительные*: $\mathbf{x}_i \cdot \mathbf{w} - w_0 \geq 0$

$\mathbf{x}_i$ *отрицательные*: $\mathbf{x}_i \cdot \mathbf{w} - w_0 < 0$

- Нейрон с функцией активации sign задаёт бинарный линейный классификатор (гиперплоскость в x)
- «Обучение» нейрона = настройка весов w_j и w_0
- Настраивать веса линейного классификатора можем с помощью градиентного спуска или SVM

Напоминание про градиентный спуск



- Есть функция стоимости от параметров w, нужной найти параметры, при которых она достигает минимума
- Считаем градиент функции с точке начального приближения и сдвигаем w в сторону уменьшения стоимости
- Повторяем до сходимости
- Попадаем в локальный минимум (который может быть глобальным, или нет)

Градиентный метод обучения

Минимизация аппроксимированного эмпирического риска:

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(\langle w, x_i\rangle y_i\big) \to \min_w .$$

Численная минимизация методом *градиентного спуска*:

$$w^{(0)} := \text{начальное приближение;}$$

$$w^{(t+1)} := w^{(t)} - \eta \cdot \nabla Q(w^{(t)}), \qquad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=0}^{n},$$

где η — *градиентный шаг*, называемый также *темпом обучения*.

$$w^{(t+1)} := w^{(t)} - \eta \sum_{i=1}^{\ell} \mathscr{L}'\big(\langle w^{(t)}, x_i\rangle y_i\big)\, x_i\, y_i.$$

Идея ускорения сходимости:
брать (x_i, y_i) по одному и сразу обновлять вектор весов.

Стохастический градиентный спуск

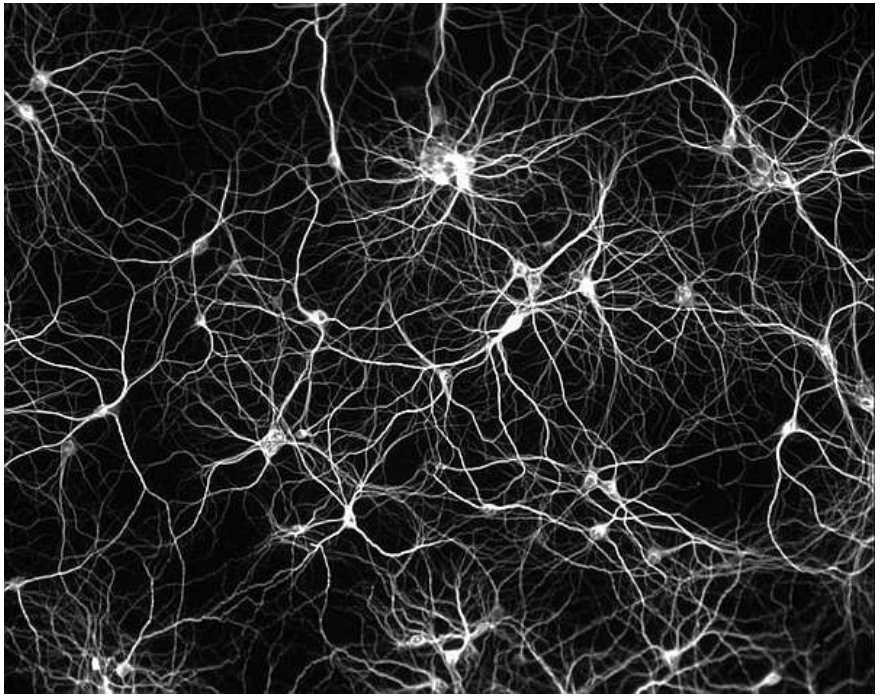
Stochastic Gradient Descend (SGD)

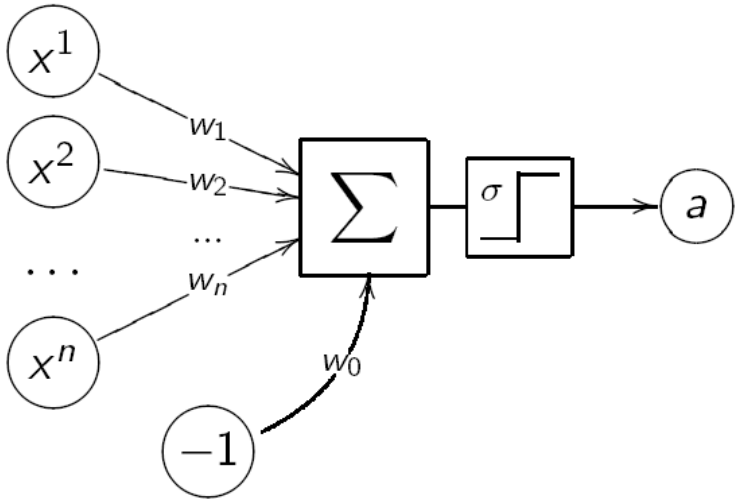
Вход: выборка X^ℓ, темп обучения h, темп забывания λ;
Выход: вектор весов w;

инициализировать веса w_j, $j = 0, \ldots, n$;
инициализировать оценку функционала: $\bar{Q} := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}_i(w)$;
повторять
    выбрать объект x_i из X^ℓ случайным образом;
    вычислить потерю: $\varepsilon_i := \mathscr{L}_i(w)$;
    сделать градиентный шаг: $w := w - h\nabla \mathscr{L}_i(w)$;
    оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda\varepsilon_i$;
пока значение $\bar{Q}$ и/или веса w не сойдутся;

Robbins, H., Monro S. A stochastic approximation method // Annals of Mathematical Statistics, 1951, 22 (3), p. 400–407.

Представимость функций нейросетями



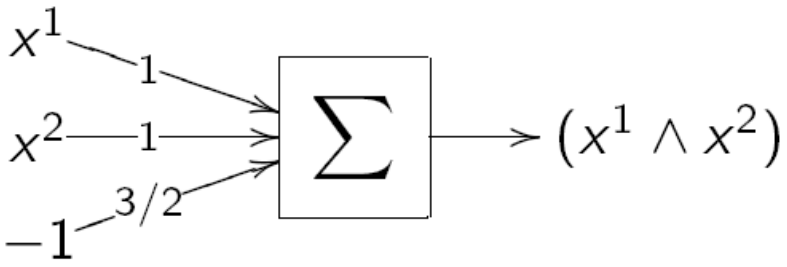


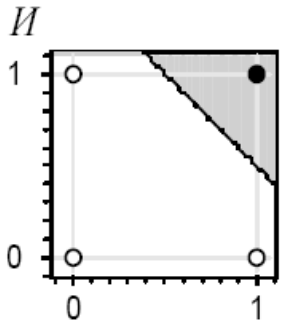
- Какие функции можно реализовать одним нейроном?
- А суперпозицией (сетью) нейронов?

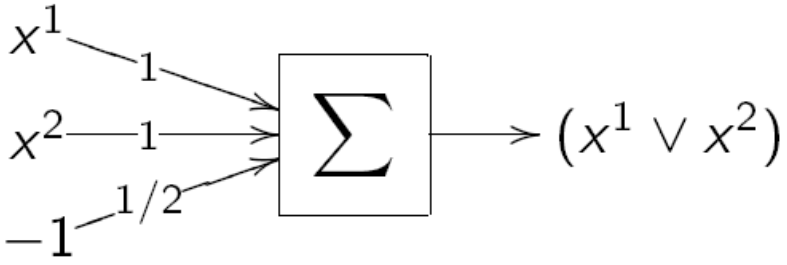
Логические элементы

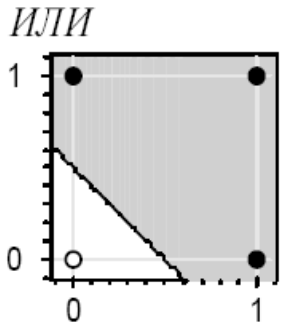
Функции И, ИЛИ, НЕ от бинарных переменных x^1 и x^2:

$$x^1 \wedge x^2 = \left[x^1 + x^2 - \tfrac{3}{2} > 0\right];$$
$$x^1 \vee x^2 = \left[x^1 + x^2 - \tfrac{1}{2} > 0\right];$$
$$\neg x^1 = \left[-x^1 + \tfrac{1}{2} > 0\right];$$







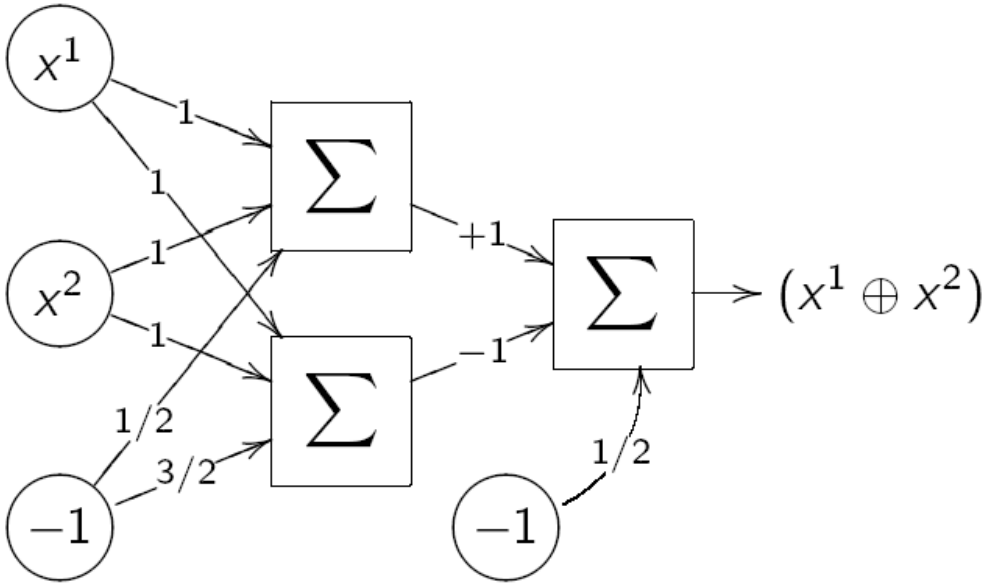


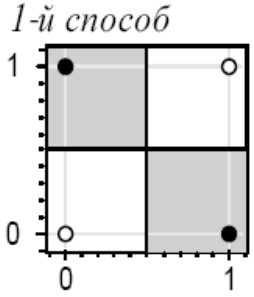
Исключающее ИЛИ (XOR)

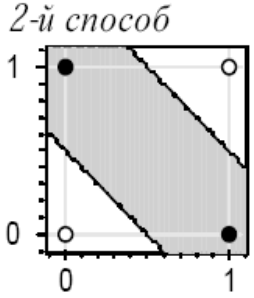
Функция $x^1 \oplus x^2 = [x^1 \neq x^2]$ не реализуема одним нейроном.

Два способа реализации:

- Добавлением нелинейного признака:
  $x^1 \oplus x^2 = \left[x^1 + x^2 - 2x^1x^2 - \frac{1}{2} > 0\right]$;
- Сетью (двухслойной суперпозицией) функций И, ИЛИ, НЕ:
  $x^1 \oplus x^2 = \left[(x^1 \vee x^2) - (x^1 \wedge x^2) - \frac{1}{2} > 0\right]$.







Приближение функций нейросетью

Утверждение

Любая булева функция представима в виде ДНФ, следовательно, и в виде двухслойной сети.

Решение тринадцатой проблемы Гильберта:

Теорема (Колмогоров, 1957)

Любая непрерывная функция n аргументов на единичном кубе $[0,1]^n$ представима в виде суперпозиции непрерывных функций одного аргумента и операции сложения:

$$f(x^1, x^2, \ldots, x^n) = \sum_{k=1}^{2n+1} h_k \left(\sum_{i=1}^{n} \varphi_{ik}(x^i) \right),$$

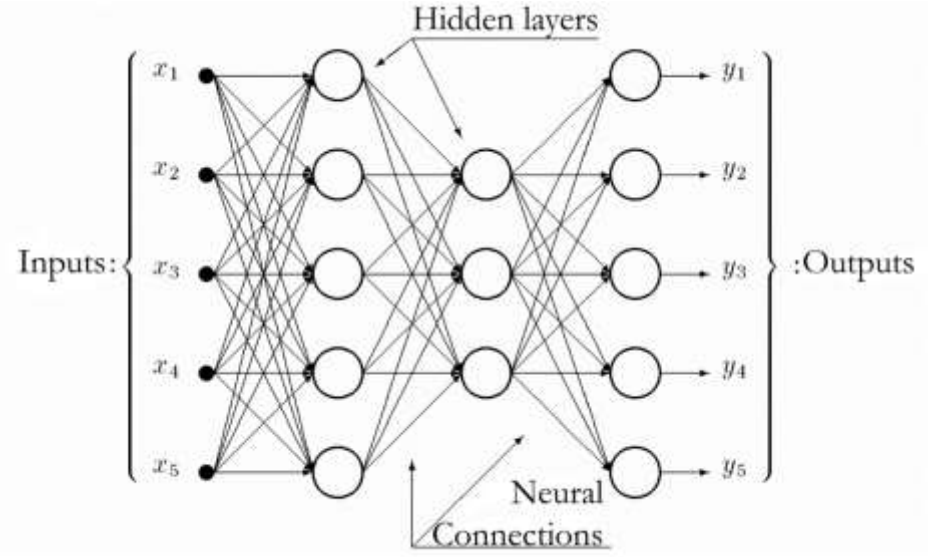
где h_k, φ_{ik} — непрерывные функции, и φ_{ik} не зависят от f.

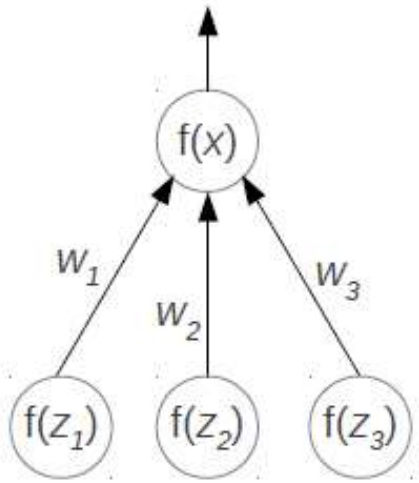
Представимость функций

- Итого, теоретически доказано, что с помощью линейных операций и одной нелинейной функции активации можно вычислить любую непрерывную функцию с любой желаемой точностью

- Однако из доказательств не следует, как должна быть устроена сеть, сколько в ней должно быть нейронов, какие у них должны быть веса

Задание нейросети





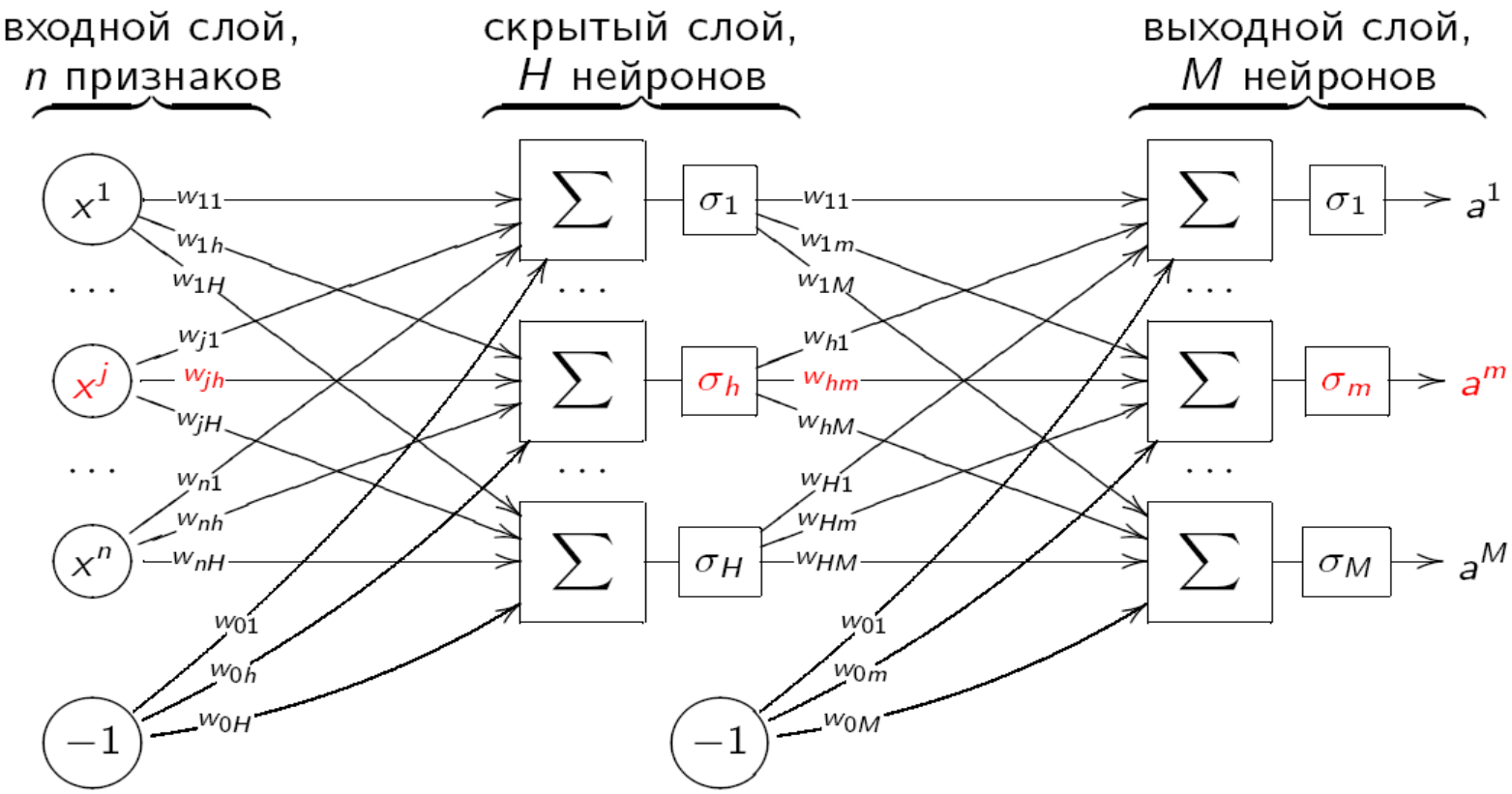
$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$

x is called the total input to the neuron, and $f(x)$ is its output

- **Архитектура нейросети** – взвешенный ориентированный граф, в котором вершины – нейроны, ребра – связи
  - Архитектуру обычно задаёт разработчик на основании опыта и «лучших примеров»
  - Архитектуру только недавно начали «учить» (Neural Architecture Search)
- **Веса нейросети** – совокупность весов всех рёбер (веса каждого нейрона)
  - Настройка весов – «обучение» нейросети, для этого предложено несколько подходов

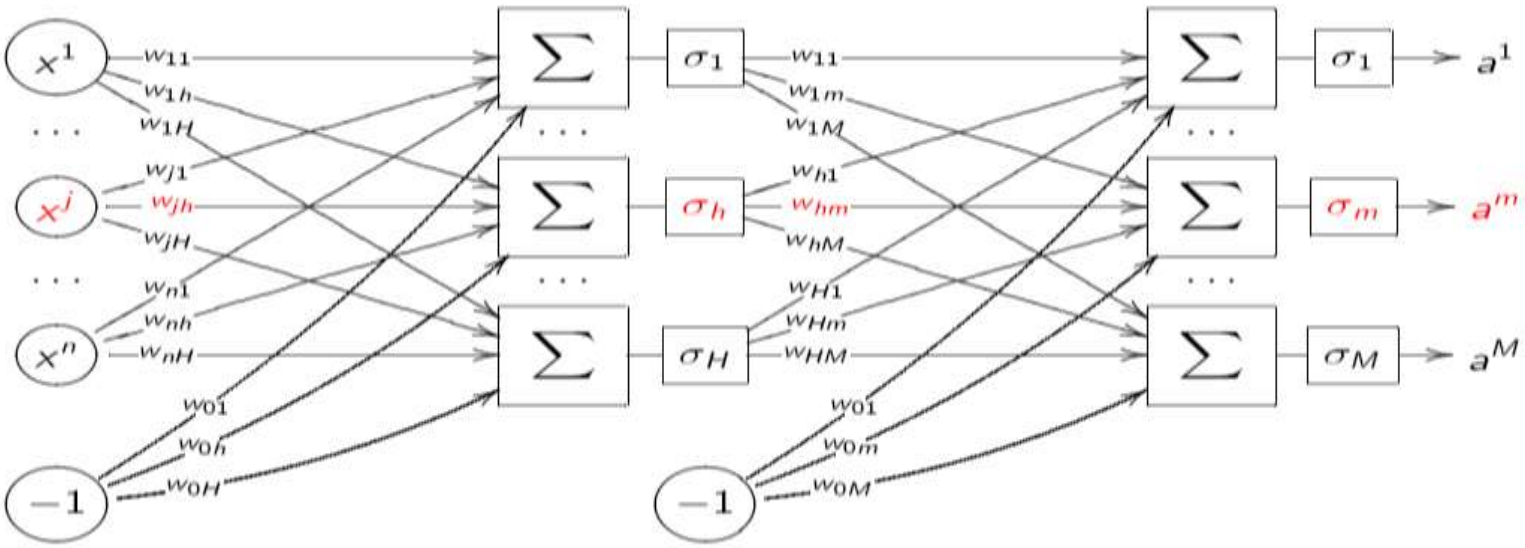
Многослойная feed-forward нейросеть

Пусть для общности $Y = \mathbb{R}^M$, для простоты слоёв только два.



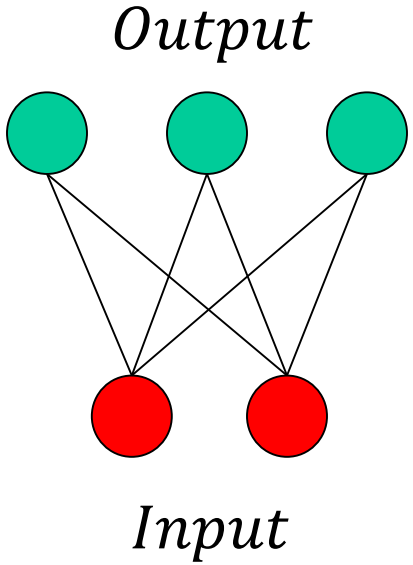
- Передача сигналов идёт в одном направлении (feed-forward)
- Сеть можно разделить на «слои нейронов», по числу предшествующих нейронов на пути сигнала

Выходы нейросети?



- Бинарная классификация
  - Один нейрон (+1, -1)
  - Два нейрона – один для +1 класса, второй для -1 класса
- Многоклассовая классификация с K-классами
  - K нейронов
- Регрессия
  - Одно число – один нейрон на выходе
  - K чисел – k выходных нейронов (пример - (x1, y1, x2, y2) для bbox объекта при решении задачи детекции)

Линейный перспептрон



- Простейшая нейронная сеть с входом x и 1 слоем, выход которого является выходом всей неросети.
- Такая сеть реализует линейную функцию:

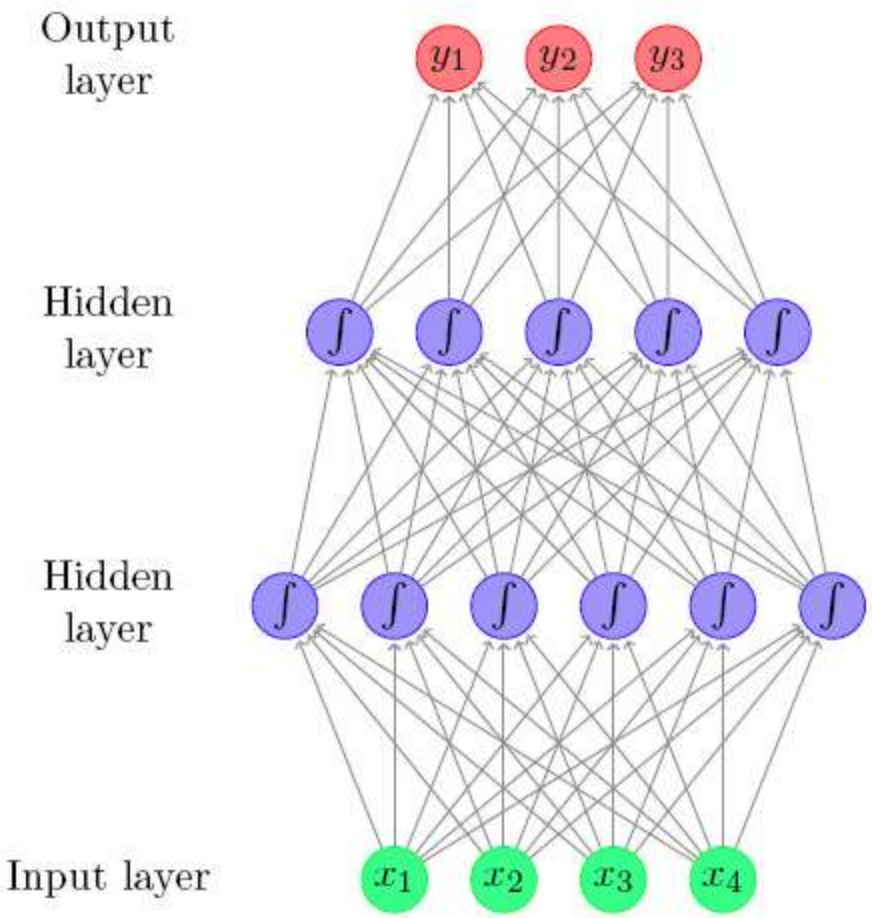
$$NN_{perceptron}(x) = xW + b$$

$$x \in \mathbb{R}^{d_{in}} \quad W \in \mathbb{R}^{d_{in} x d_{out}} \quad b \in \mathbb{R}^{d_{out}}$$

Вход и выход мы видим, поэтому это видимые слои

Вопрос – сколько у нас параметров (весов) в такой сети?

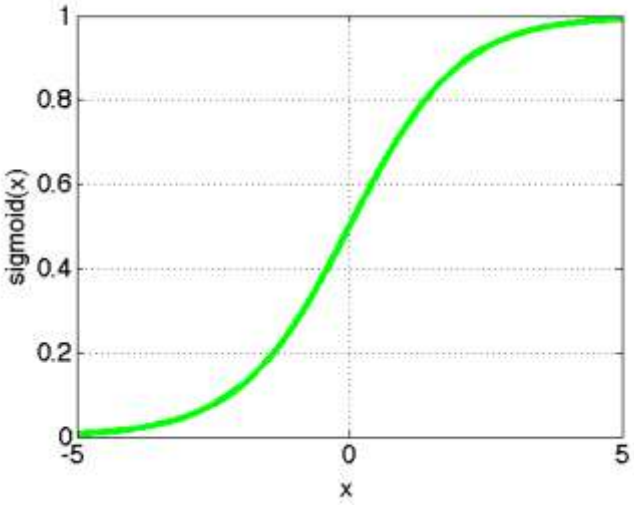
Многослойный персептрон

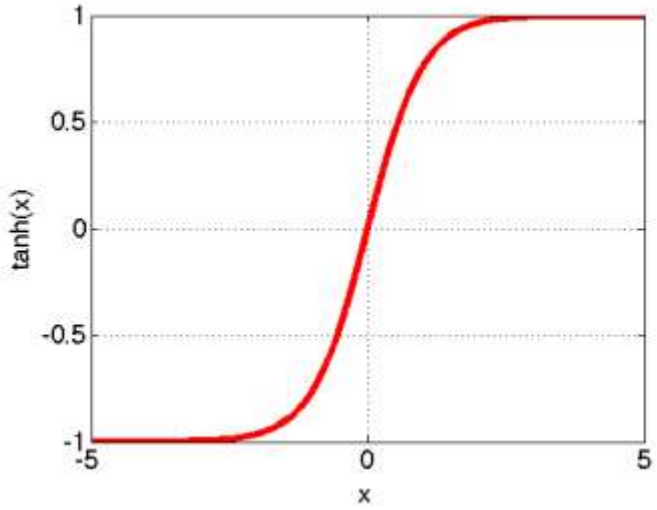


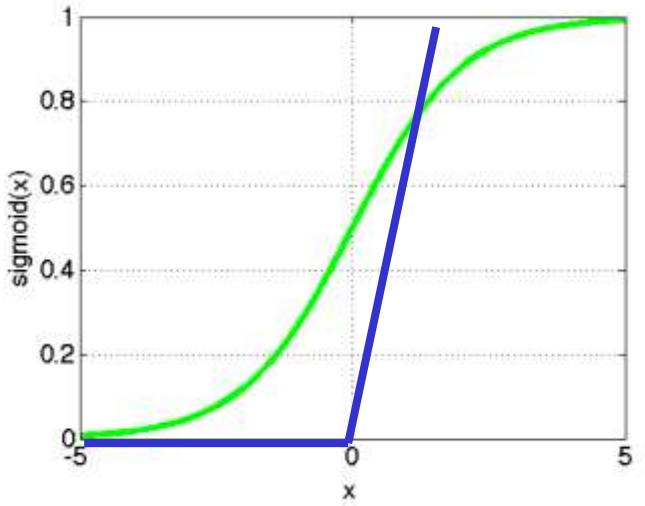
- Промежуточные слои – «скрытые»
- Если каждый нейрон слоя связан с каждым нейроном предыдущего слоя, то такой слой – полносвязанный (fully connected)
- Нейроны скрытых слоёв обычно имеют функцию активации (нелинейную)

Функции активации

- Tanh
  - $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$
- Sigmoid:
  - $\mathrm{sigm}(x) = \frac{1}{1+e^{-x}}$ - преобразует все значения в интервал [0,1]
- Rectified linear
  - ReLU(x) = max(0,x)







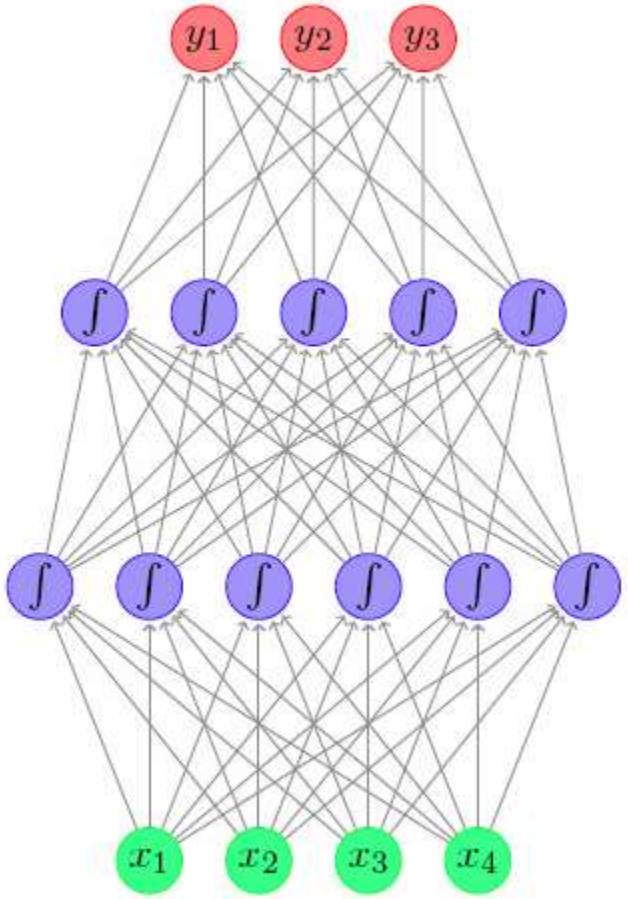
Функция перспетрона

- Запишем послойные преобразования:

$$
\begin{aligned}
NN_{MLP2}(\mathbf{x}) &= \mathbf{y} \\
\mathbf{h^1} &= g^1(\mathbf{x}\mathbf{W^1} + \mathbf{b^1}) \\
\mathbf{h^2} &= g^2(\mathbf{h^1}\mathbf{W^2} + \mathbf{b^2}) \\
\mathbf{y} &= \mathbf{h^2}\mathbf{W^3}
\end{aligned}
$$

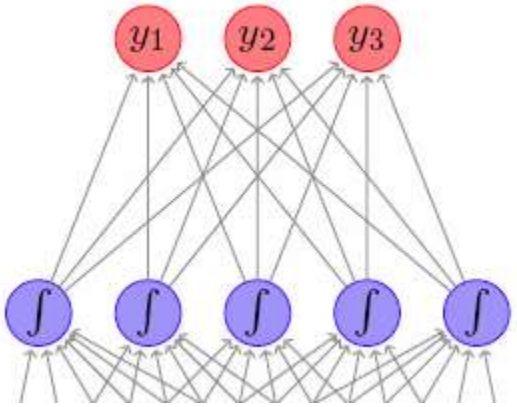
- Одна функция:

$$NN_{MLP2}(\mathbf{x}) = (g^2(g^1(\mathbf{x}\mathbf{W^1} + \mathbf{b^1})\mathbf{W^2} + \mathbf{b^2}))\mathbf{W^3}$$



Softmax-преобразование

- Если мы решаем задачу многоклассовой классификации, то мы хотим интерпретировать выходы y_i как вероятности меток классов
- На надо, чтобы $y_i \in [0,1], \sum_{i=1}^{N} y_i = 1$
- Это можно сделать с помощью softmax преобразования

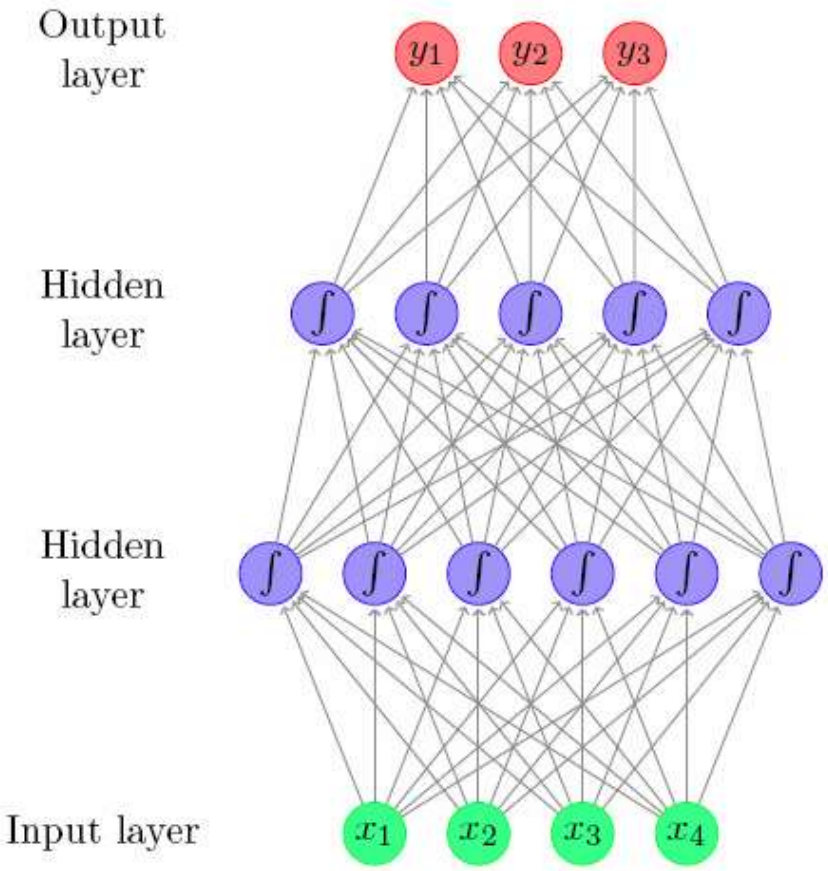


$$x = x_1, \ldots, x_k$$

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$$

- «Сжимаем» произвольный вектор длины N в вектор со значениями (0,1) и суммой 1
- Функция взята из многомерной логистической регрессии

Как обучать персептрон?



- С помощью персептрона (даже только с 1 скрытым слоем) мы можем задавать произвольные функции
- Нам потребуется:
  - **(А) Функция потерь**
  - **(Б) Метод её минимизации**

Функции потерь для классификации

- Categorical cross-entropy loss

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)}$$

- Измеряет близость истинного и оцененного распределения меток (обычно после softmax)

Подробнее можно тут: https://sisyphus.gitbook.io/project/deep-learning-basics/basics/multi-class-and-cross-entropy-loss

Обучение с помощью SGD

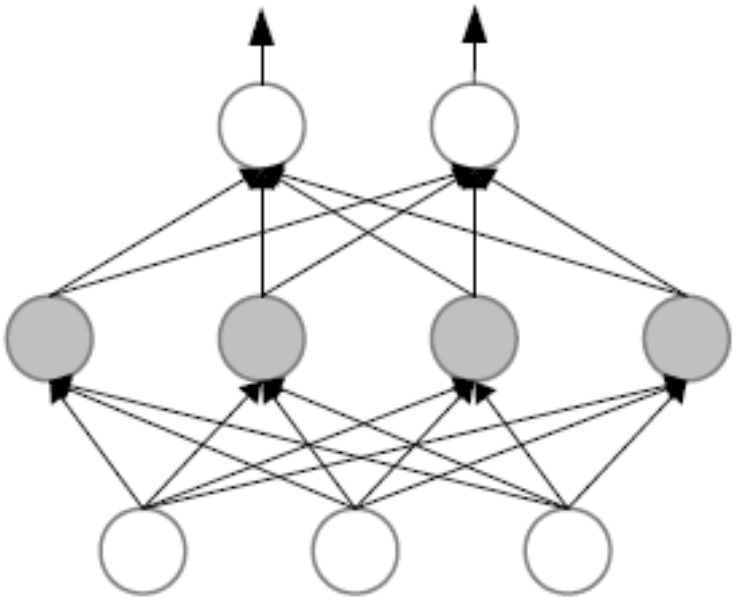
Точно также, как при обучении одного нейрона (линейного классификатора), можем обучать все параметры нейросети

Algorithm 1 Online Stochastic Gradient Descent Training

Input: Function $f(\mathbf{x}; \theta)$ parameterized with parameters θ.
Input: Training set of inputs $\mathbf{x_1}, \ldots, \mathbf{x_n}$ and outputs $\mathbf{y_1}, \ldots, \mathbf{y_n}$.
Input: Loss function L.
while stopping criteria not met **do**
    Sample a training example $\mathbf{x_i}, \mathbf{y_i}$
    Compute the loss $L(f(\mathbf{x_i}; \theta), \mathbf{y_i})$
    $\hat{\mathbf{g}} \leftarrow$ gradients of $L(f(\mathbf{x_i}; \theta), \mathbf{y_i})$ w.r.t θ
    $\theta \leftarrow \theta + \eta_k \hat{\mathbf{g}}$
return θ

Как посчитать градиент для всех параметров нейросети?

Расчёт градиента



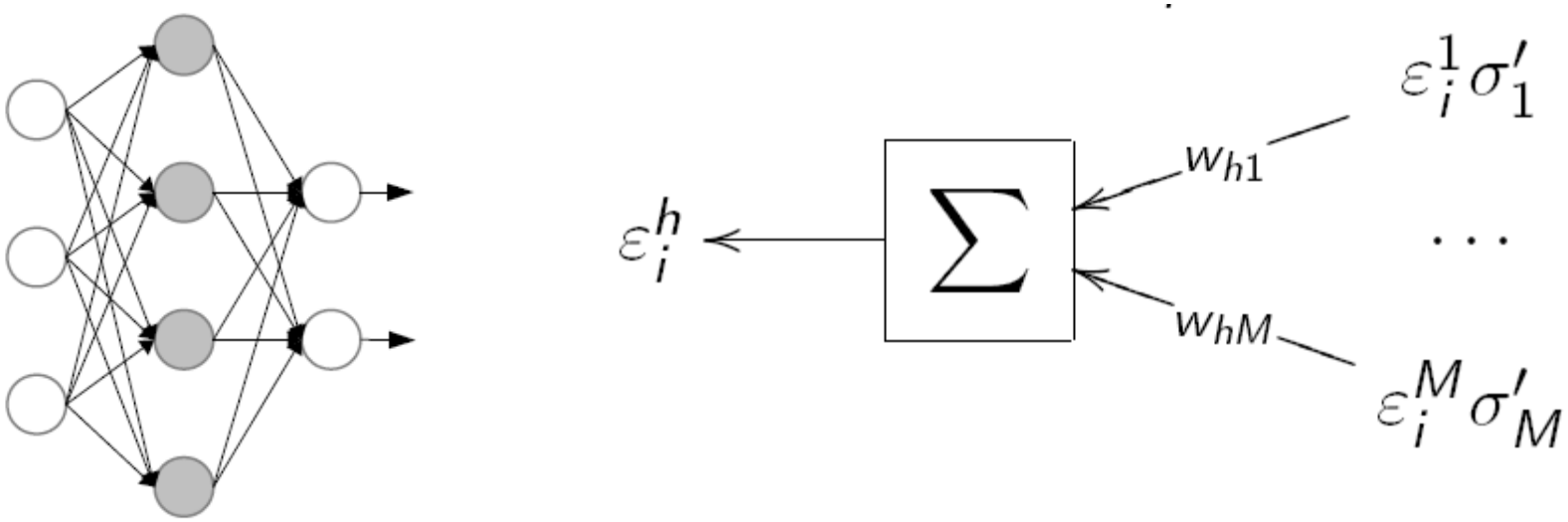
- Нейросеть вычисляет дифференцируемую функцию от своих входов
- Можем последовательно применять правило дифференцирования сложных функций для вычисления производных по каждому параметру нейросети
- Метод получил название «обратное распространение ошибки» и имеет длинную историю

•↑ *Галушкин А. И.* Синтез многослойных систем распознавания образов. — М.: «Энергия», 1974.

•↑ *Werbos P. J.*, Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.

•↑ 1 2 *Rumelhart D.E., Hinton G.E., Williams R.J.*, Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318—362. Cambridge, MA, MIT Press. 1986.

Процедура обратного распространения ошибки



- Градиент для каждого параметра нейрона можем рассчитать как взвешенное произведение ошибок всех связанных с ним последующий нейронов на производную функции активации нейрона
- Вначале посчитаем градиент нейронов выходного слоя, затем предыдущего и т.д.
- Для всей сети мы можем это сделать за один обратный проход, как-бы запустив сеть «задом наперёд»
- Тема следующего семинара!

Minibatch SGD

Algorithm 2 Minibatch Stochastic Gradient Descent Training

Input: Function $f(\mathbf{x}; \theta)$ parameterized with parameters θ.
Input: Training set of inputs $\mathbf{x_1}, \ldots, \mathbf{x_n}$ and outputs $\mathbf{y_1}, \ldots, \mathbf{y_n}$.
Input: Loss function L.
while stopping criteria not met **do**
    Sample a minibatch of m examples $\{(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_m}, \mathbf{y_m})\}$
    $\hat{\mathbf{g}} \leftarrow 0$
    for $i = 1$ to m **do**
        Compute the loss $L(f(\mathbf{x_i}; \theta), \mathbf{y_i})$
        $\hat{\mathbf{g}} \leftarrow \hat{\mathbf{g}} +$ gradients of $\frac{1}{m} L(f(\mathbf{x_i}; \theta), \mathbf{y_i})$ w.r.t θ
    $\theta \leftarrow \theta + \eta_k \hat{\mathbf{g}}$
return θ

- Формируем случайную группу примеров (minibatch) и усредняем градиенты по всем примерам
- Много плюсов:
  - Менее шумный градиент (по сравнению с одним примеров)
  - Быстрее сходимся (меняем параметры после расчёта части примеров, а не всех)
  - Вычислительно эффективнее (в RAM держим minibatch)

Инициализация сети

- Пусть веса слоя $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$

- Инициализация Xavier (2010):

$$\mathbf{W} \sim U\left[-\frac{\sqrt{6}}{\sqrt{d_{in}+d_{out}}}, +\frac{\sqrt{6}}{\sqrt{d_{in}+d_{out}}}\right]$$

где U[a,b] – равномерное распределение в интервале

- Инициализация He (2015):

Нормальное распределение с матожиданием 0 и стандартным отклонением $\sqrt{\frac{2}{d_{in}}}$

Заметки по процедуре обучения

- Темп обучения выбирают таким образом, чтобы веса не слишком колебались и сходились
- С течением времени темп обучения уменьшают, чтобы обеспечить сходимость весов
- «Эпоха» – этап обучения на всех примерах из обучающей выборки
- Обычно обучение состоит из нескольких эпох, между которыми меняют параметры обучения
  - Обычно экспоненциально снижает скорость обучения

Расписание

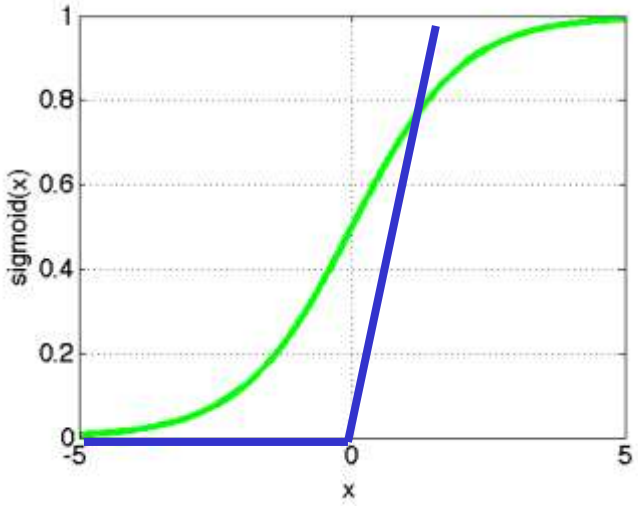
- Очень важен процесс формирования mini-batch
- Он должен хорошо представлять обучающую выборку
- В нём должны присутствовать и положительные, и отрицательные примеры (часто в равных долях)
- Можно управлять составлением minibatch, составив расписание

Развитие SGD

- Сохранение и использование градиентов с предыдущих минибатчей
  - SGD + Momentum (Поляк, 1964)
  - Nesterov momentum
- Адаптивный выбор темпа обучения для каждого минибатча
  - AdaGrad
  - AdaDelta
  - Adam
  - Ranger (https://medium.com/@lessw/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d)
- Обычно все реализованы в библиотеках и можно экспериментировать с ними

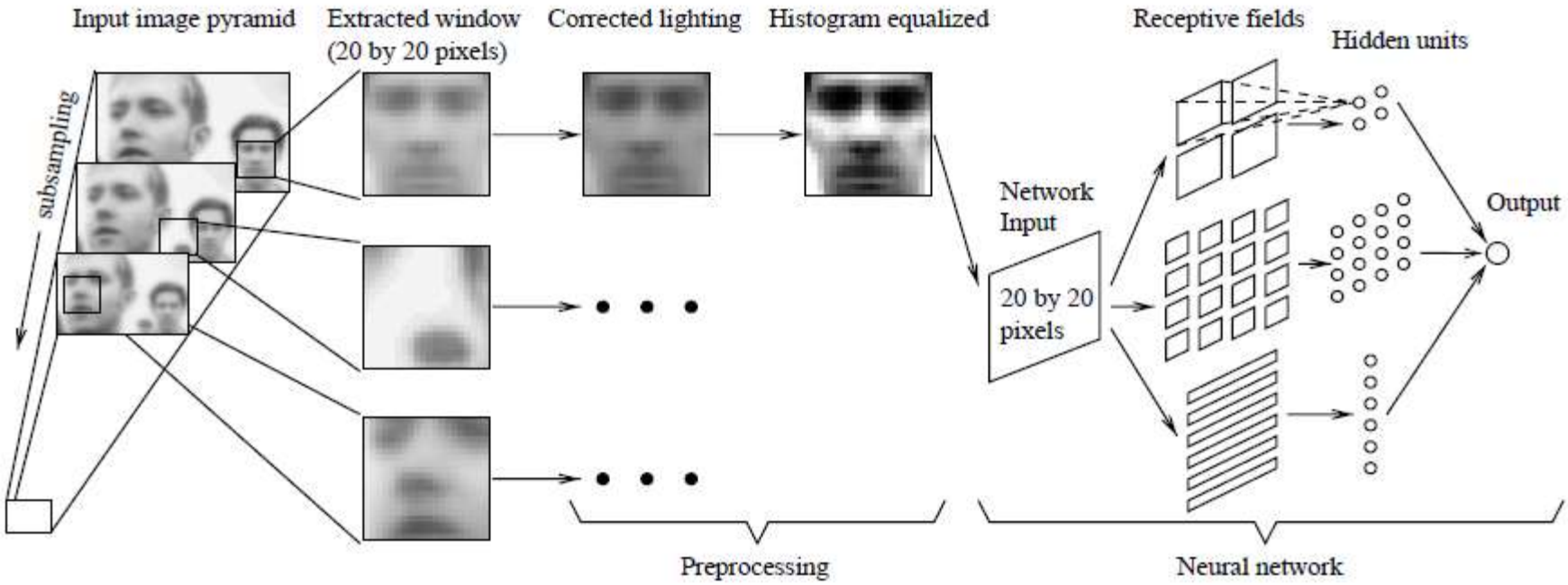
Проблемы обучения

- «Затухание» (saturation) градиентов, если попадаем на область низкого градиента функции активации
  - Sigmoid очень страдает от этого
  - ReLU не страдает
- «Мёртвые» (dead) нейроны, которые, так получилось, получают только отрицательные или нулевые входы
- Исчезающие или взрывные градиенты (vanishing or exploding gradients) в глубоких сетях



Rowley face detector (1998)

- Метод обратного распространения ошибки оказался очень эффективным
- Пример – детектор лица, лучший до Viola-Jones

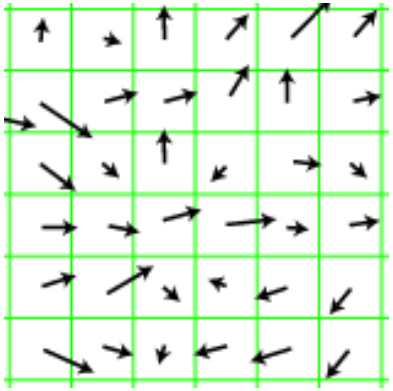


B. Rowley, T. Kanade. Neural Network-Based Face Detection. PAMI, 1998.

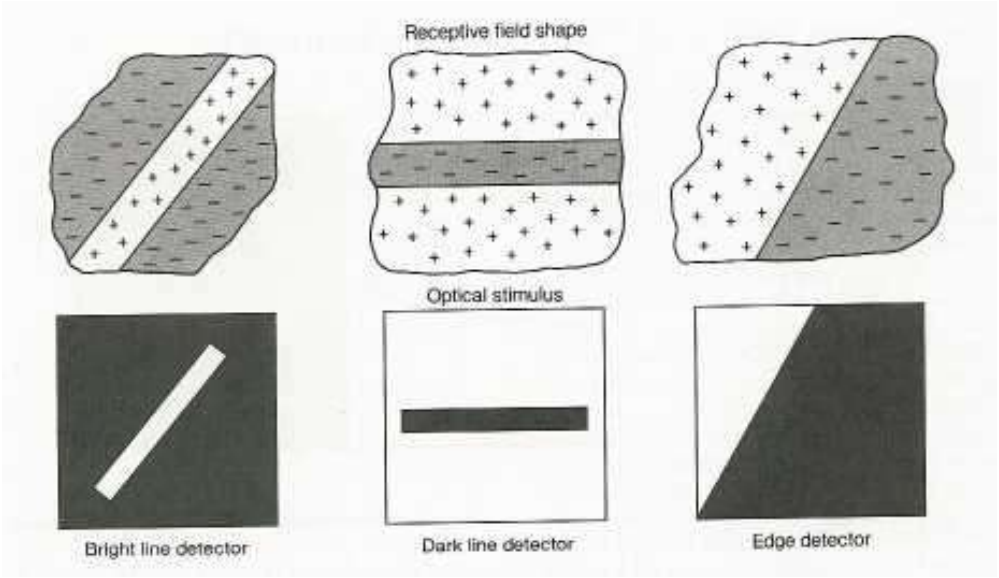
Нейросети для обработки картинок

- Персептрон работает с векторами
- Мы хотим обрабатывать изображение, т.е. на вход подавать 3D матрицу $X \in \mathbb{R}^{m \times n \times k}$
- Как должна быть устроена нейросеть, чтобы обрабатывать изображения?

- Мы также хотим учесть знания об обработке изображений в мозге человека и эвристическими методами распознавания
- Края, градиенты и т.д. хорошо помогают!

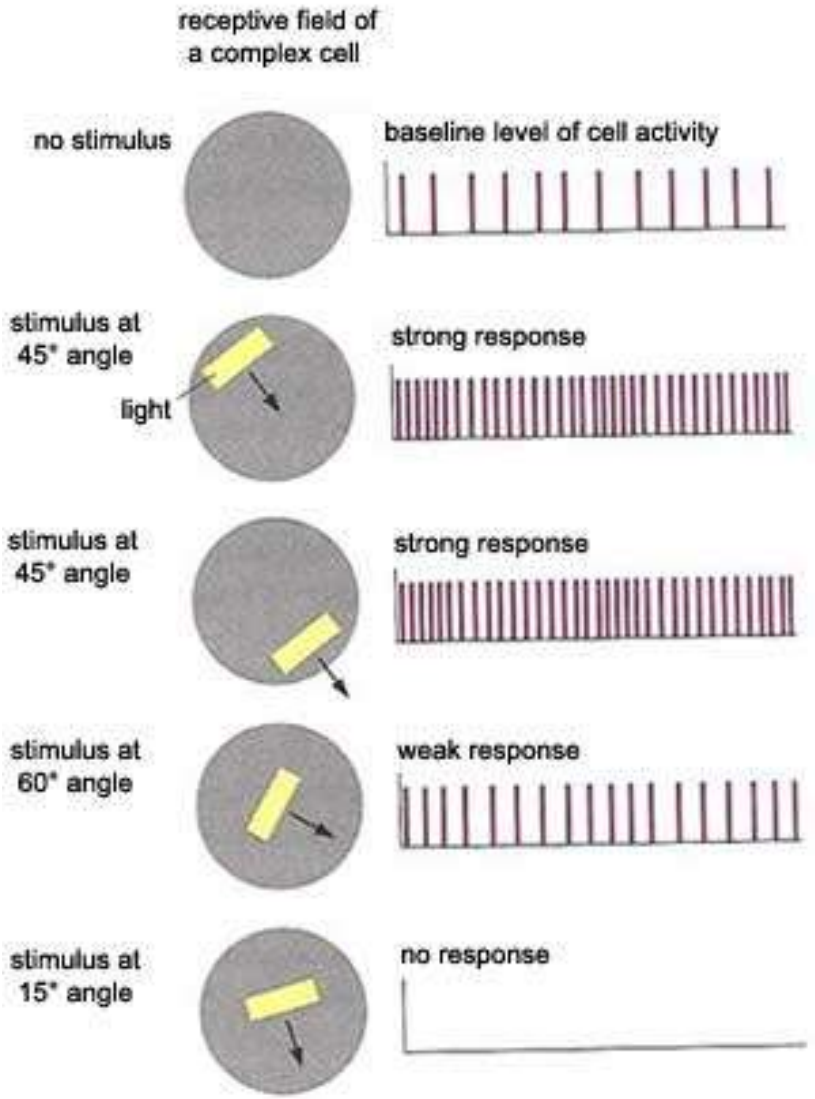


Простые и сложные клетки визуальной коры



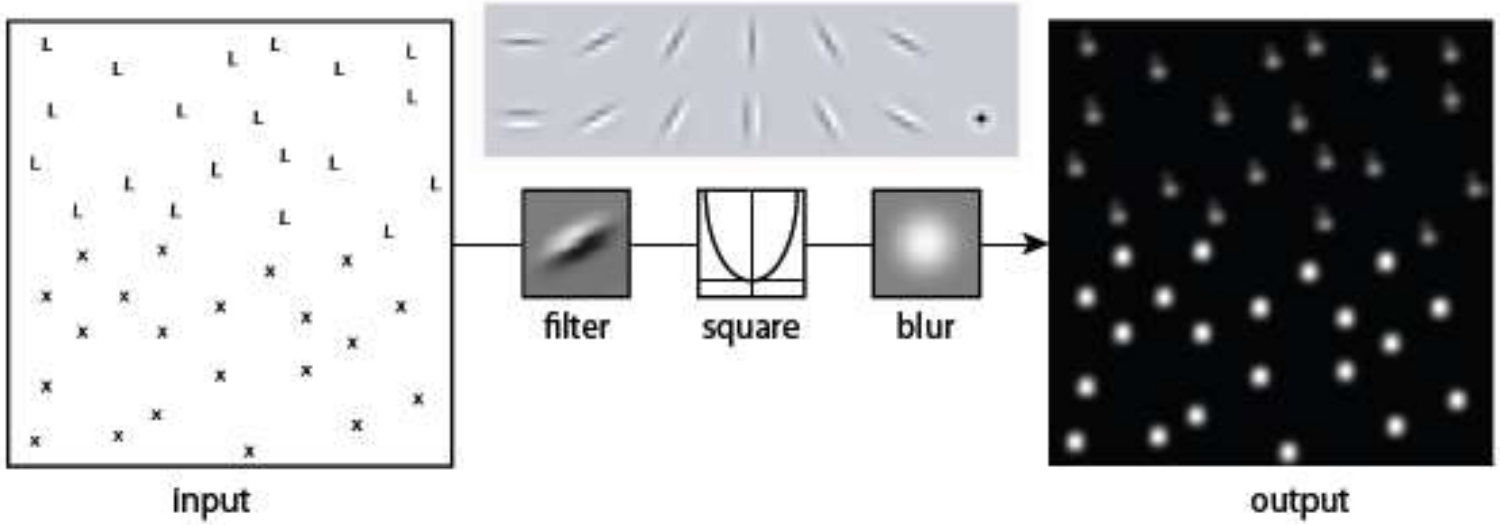
Простые клетки (S)

- Простые клетки чувствительны к контрастным объектам определённого размера, ориентации и положения
- Сложные клетки ***инвариантны*** к сдвигам в небольшой окрестности
- Как S и C смоделировать?



Сложные клетки (C)

Банки текстурных фильтров



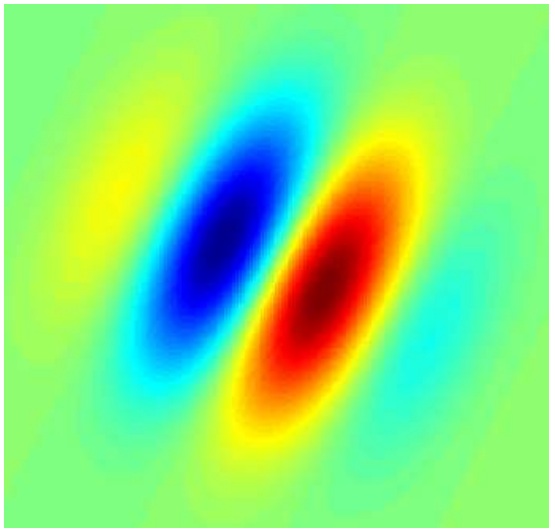
- Выберем набор (банк) фильтров, каждый из которых чувствителен к краю определенной ориентации и размера
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя

Pietro Perona and Jitendra Malik «Detecting and Localizing edges composed of steps, peaks and roofs», ICCV 1990

Фильтры Габора как модель простых клеток

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

$$x' = x\cos(\theta) + y\sin(\theta)$$
$$y' = -x\sin(\theta) + y\cos(\theta)$$



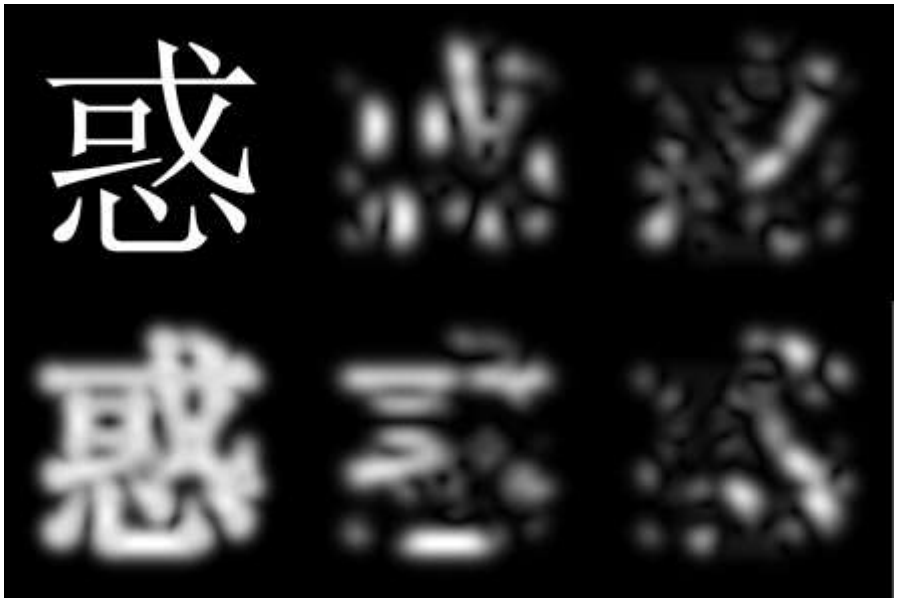
θ - ориентация

λ - длина волны

σ - сигма гауссиана

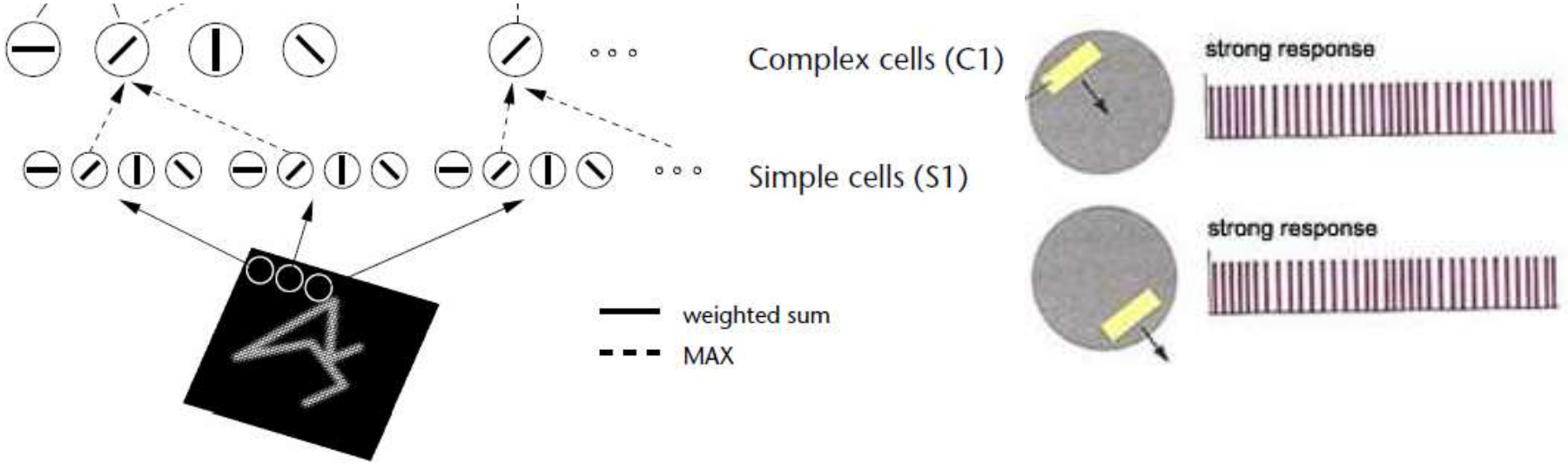
γ - соотношение размеров (aspect ratio), «эллиптичность фильтра»

ψ - сдвиг фазы



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году
- Позволяет сделать банк фильтров, для выделения краёв разной ориентации, масштаба и положения в окрестности

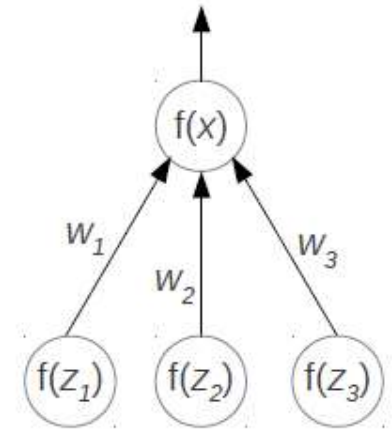
MAX-pooling как модель сложных клеток



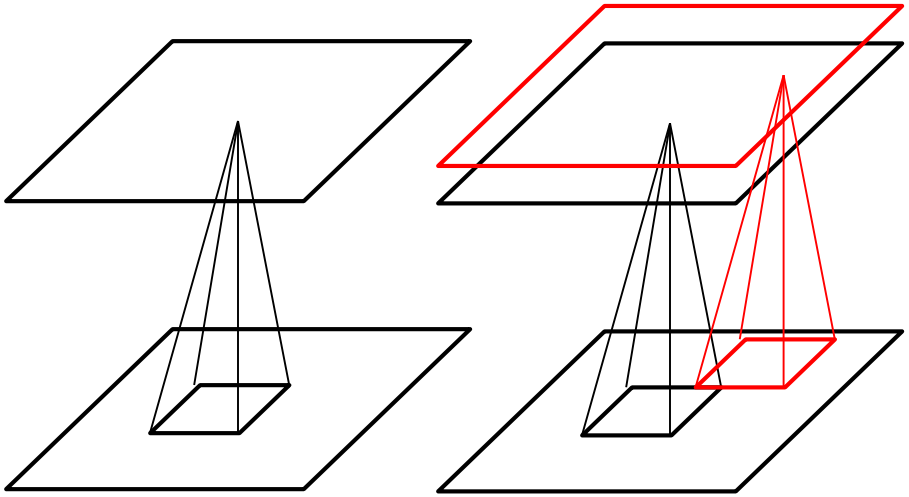
Инвариантность можно обеспечить за счёт применения оператора MAX на выходах набора «простых» клеток, чувствительных к краю одной ориентации, но в разных точках одной области

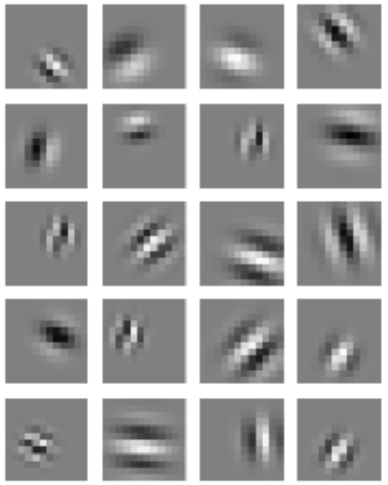
Riesenhuber, M. & Poggio, T. (1999). Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience* **2**: 1019-1025.

Свёрточный слой



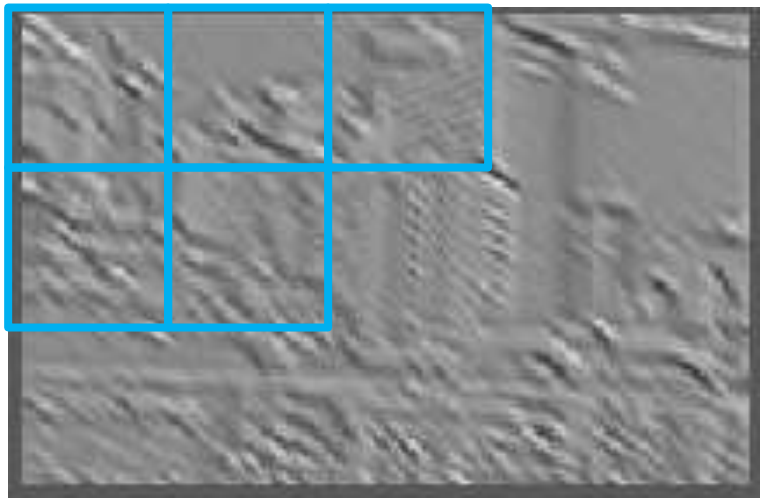
$$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$$





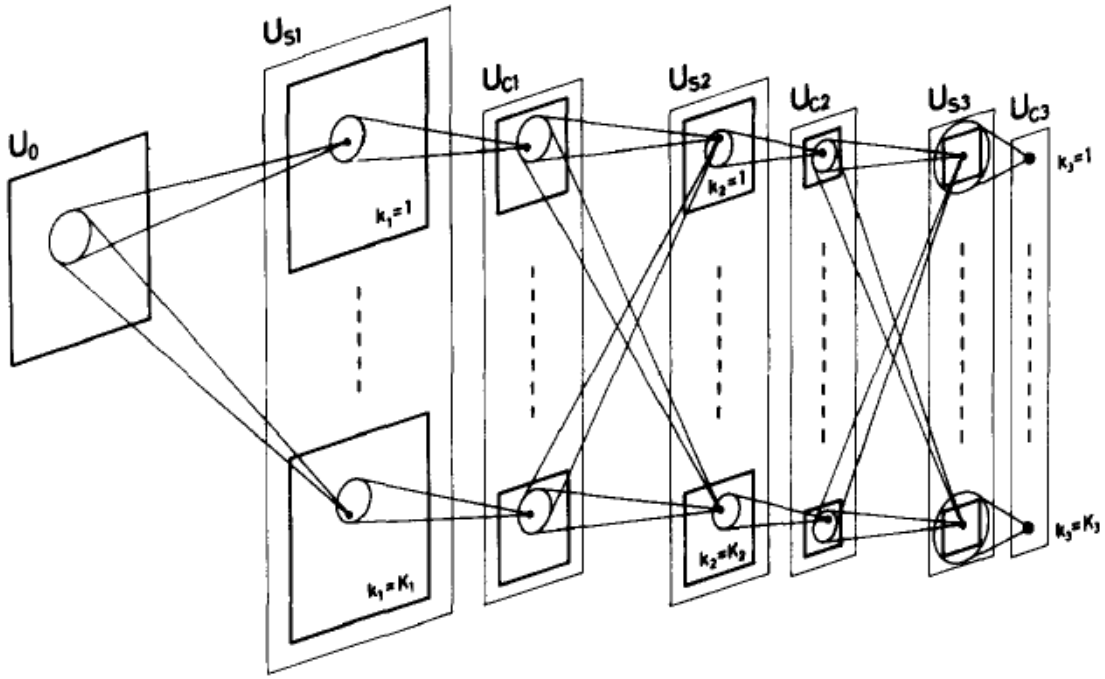
- Операцию линейной фильтрации (свёртки) для одного пикселя можно реализовать одним нейроном
- Свёртку изображения целиком можно реализовать как «слой» нейронов, веса которых одинаковы
- Обычно под «свёрточным слоем» понимают набор свёрток, применяемых к одному входу («банк фильтров»)
- Результаты свёрток объединяют в один выход $X \in \mathbb{R}^{m \times n \times k}$, где k – число свёрток в слое

Pooling слой



- Можно сделать специальный слой, который реализует операцию max-pooling
- Мы просто применяем операцию max по выбранным областям (по сетке)
- На вход получаем 3D матрицу, и выдаем 3D матрицу меньшего (обычно) разрешения
- При расчёте градиентов ошибки пробрасываем в тот нейрон предыдущего слоя, который дал max
- Можем делать не max, а sum-pooling

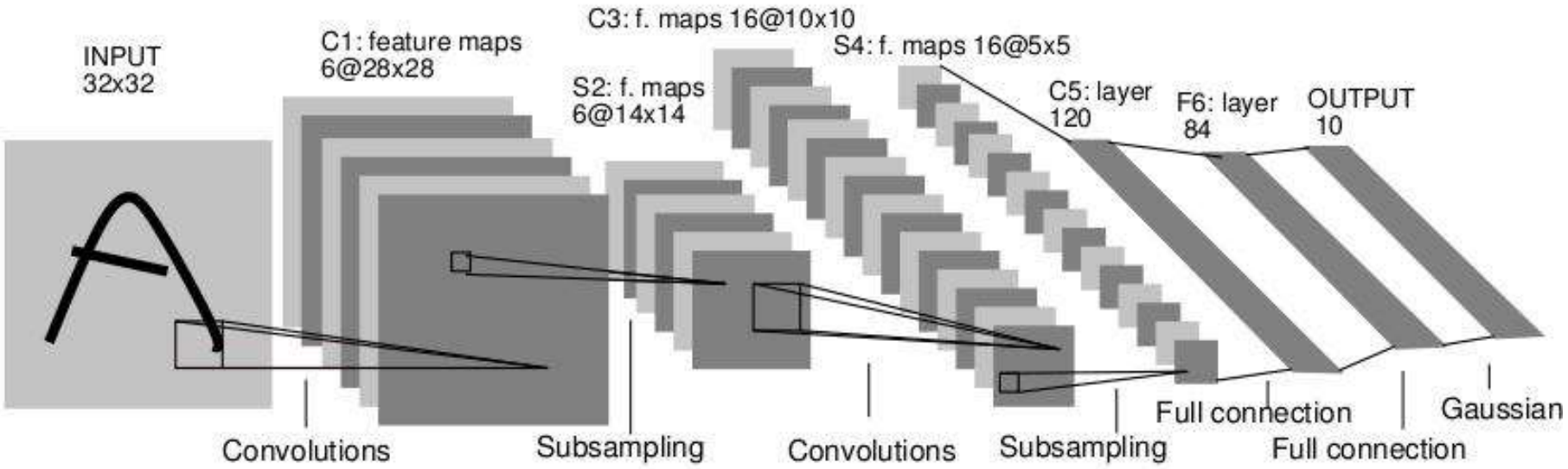
Neocognitron (1980)



- Многослойная нейросеть с чередующимимся S и C слоями
  - S-слои – линейные фильтры изображения («свёрточный слой»)
  - C-слои – MAX операторы, дающие инвариантность
- На верхнем уровне обеспечивается инвариантность по положению по всему изображению

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4): 93-202, 1980.

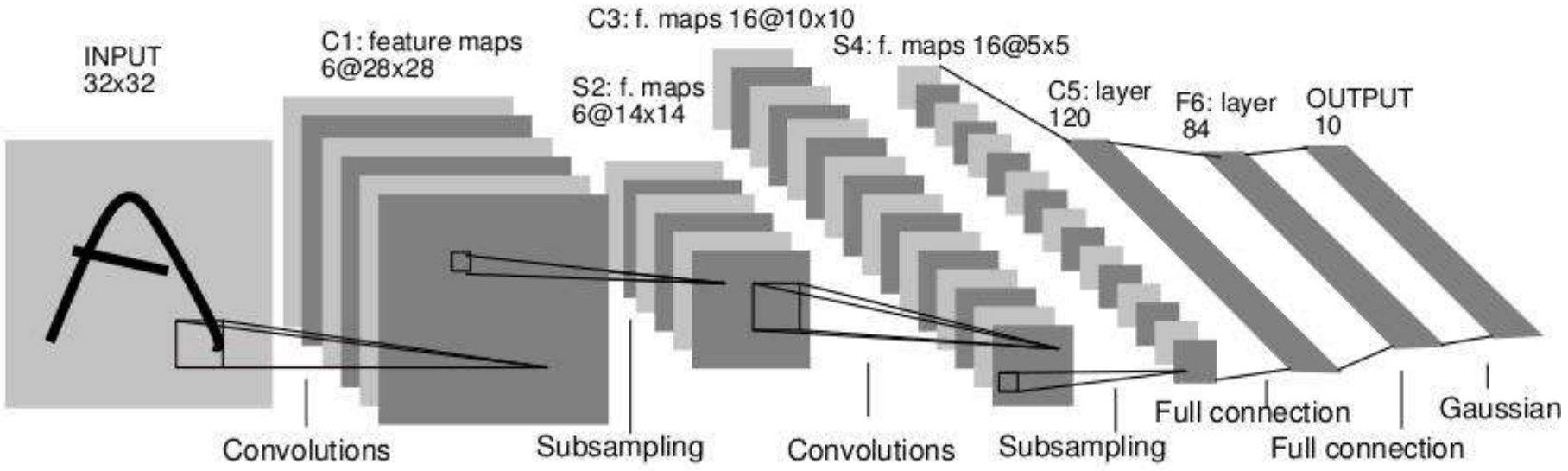
Свёрточные сети



- Неокогнитрон + обратное распространение ошибки = свёрточная сеть (Convolutional Neural Network, CNN)
- Поскольку для сверточного слоя нужно задать параметры только всех свёрток, что число параметров заметно меньше общего числа весов слоя
- Очень эффективная архитектура для распознавания изображений

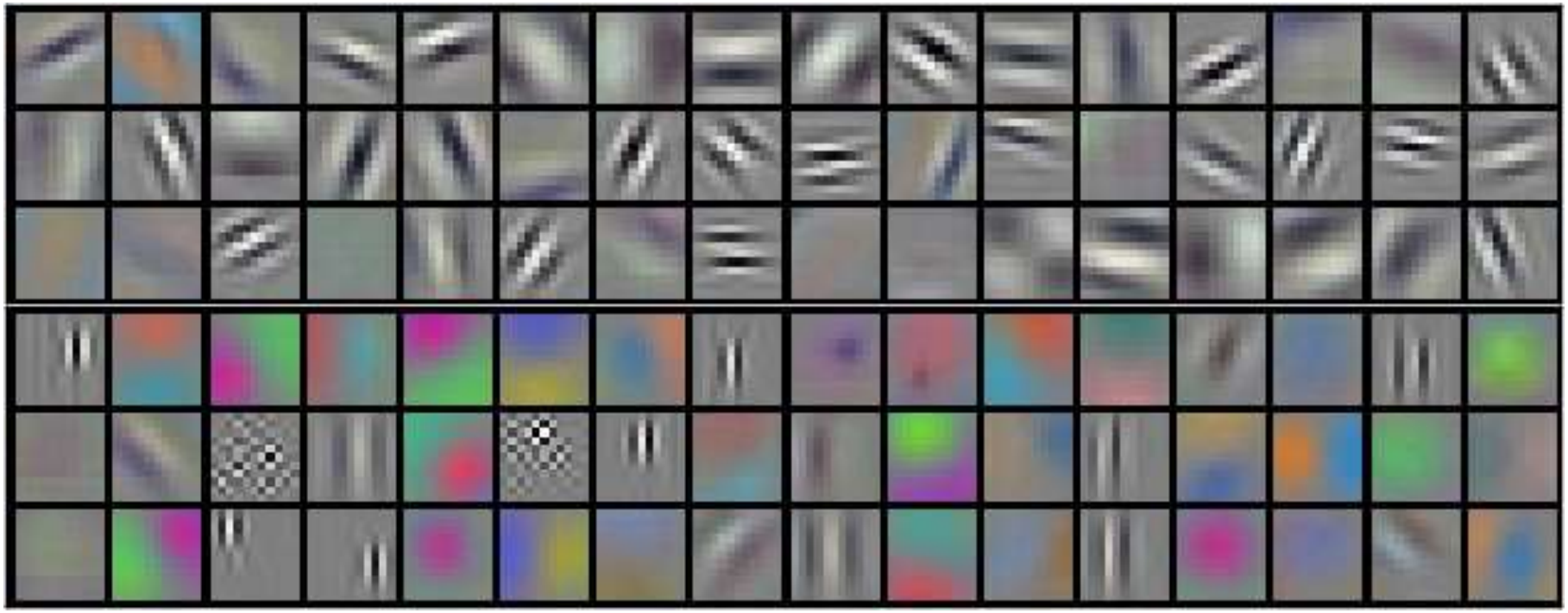
LeCun, Y., Bottou, L., Bengio, Y., and Haner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE

Подсчёт параметров



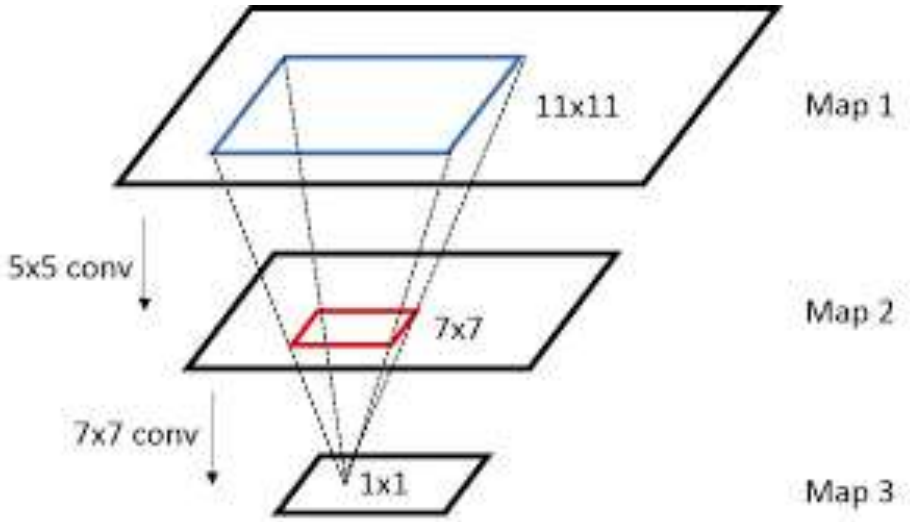
- Каковы размеры фильтров свёрток на разных слоях?
  - 5*5*1 на первом слое
  - 5*5*6 на втором свёрточном слое
  - «Глубина» тензора на выходе свёрточного слоя равна числу свёрток в свёрточном слое
  - 3е измерение свёртки равно «толщине» входного тензора
- Число весов и параметров второго слоя:
  - Параметров = 16 свёрток 5x5x6 = 150*16 +16 = 2416
  - Весов = (примерно) Параметров x 10 x 10 = 240000

Фильтры первого уровня



- Визуализируем веса фильтров
- Поскольку сворачиваем RGB изображение, то визуализация весов в RGB
- Обратите внимание на вычисление градиентов цветов

Рецептивное поле (Receptive field)

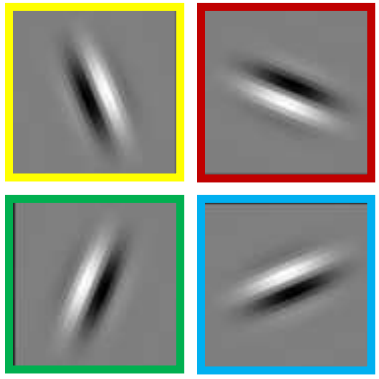


- Receptive field нейрона – область изображения, от которой зависит выход этого нейрона
- Размер и положения поля определяется глубиной нейрона, размерами свёрток, размерами областей пулинга

Шаг свёртки

- Свёртка для каждого пикселя может быть слишком долго или избыточно
- Можем повторять фильтры со сдвигом на n пикселей
- Тензор на выхода имеет меньшую размерность по ширине и высоте при шаге >=2, по сравнению с обычной свёрткой

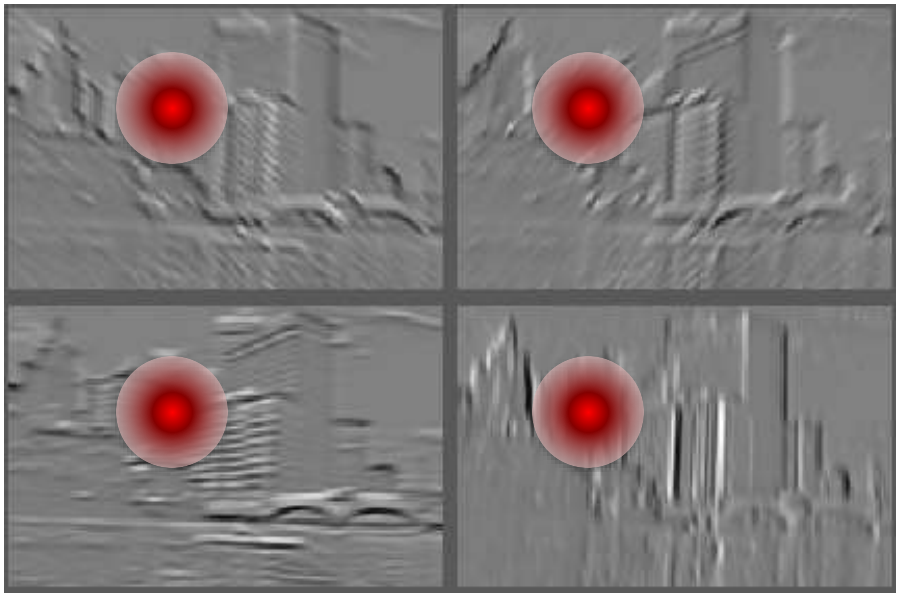
- Можем и не делать отдельный subsampling слой



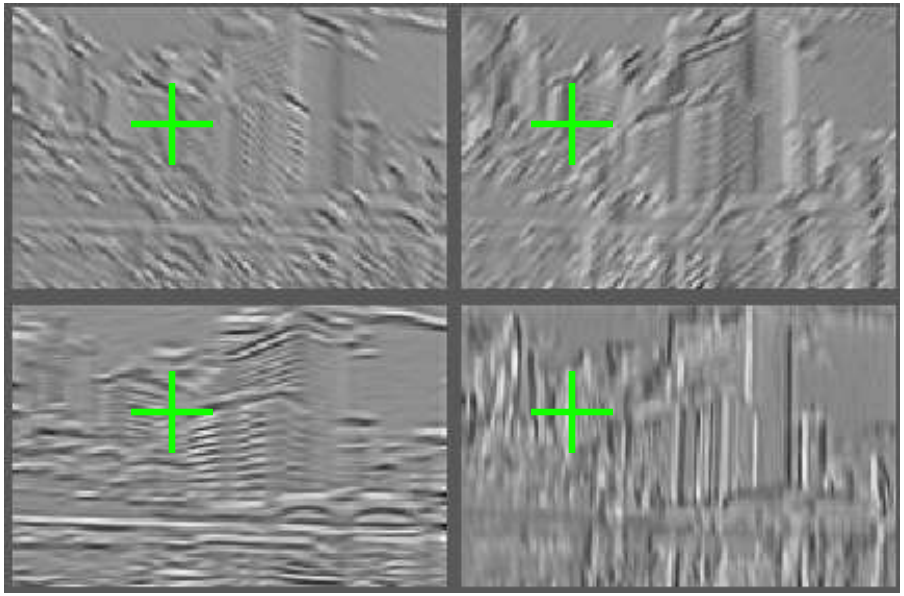
Нормализующие слои

Локальная нормализация:

- Local mean = 0, local std. = 1, “Local” → 7x7 Gaussian



Карты признаков



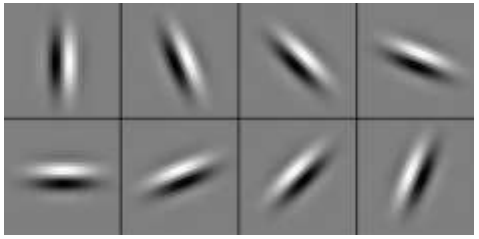
После нормализации контраста

Вспомним построение признаков

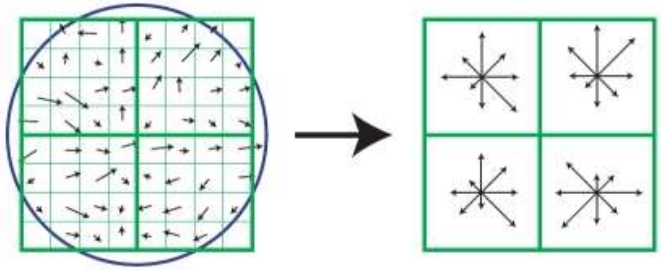
Дескриптор HOG/SIFT

Пикселы

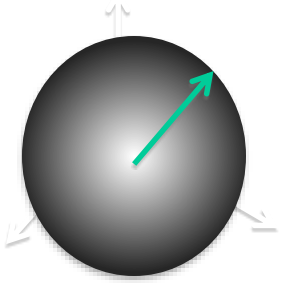
Фильтры Габора



Суммирование по ячейкам



Нормализация вектора до единичной длины

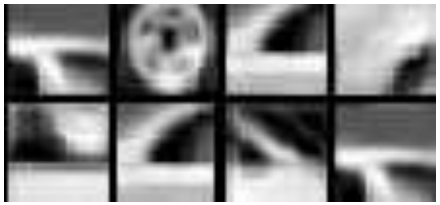


Вектор-признак

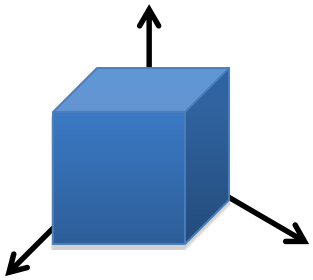
Мешок визуальных слов

HOG

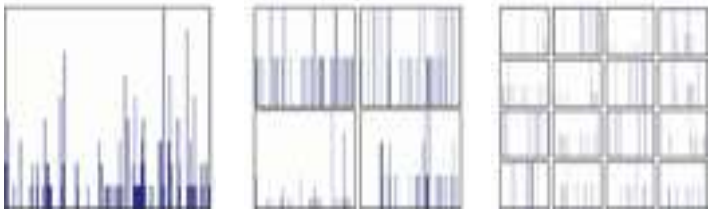
Свёртка с словарём



Max
(Находим самое близкое слово из словаря – квантуем)

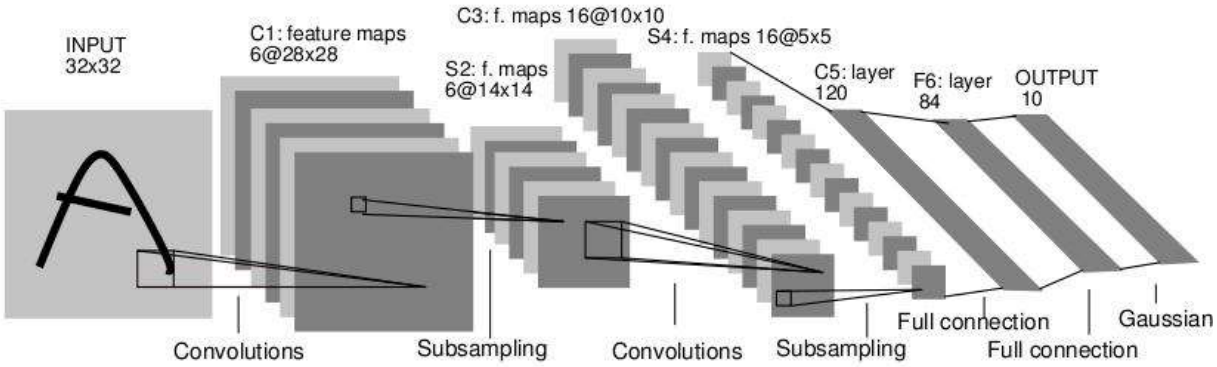


Суммирование по области
(Sum)



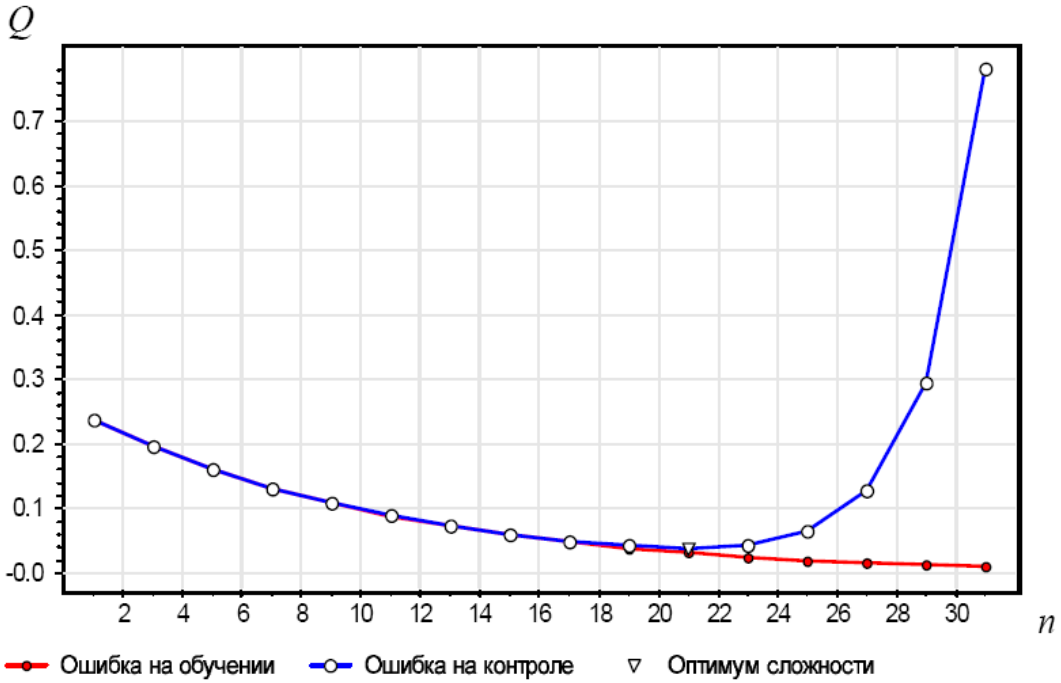
На классификатор

Важный вывод



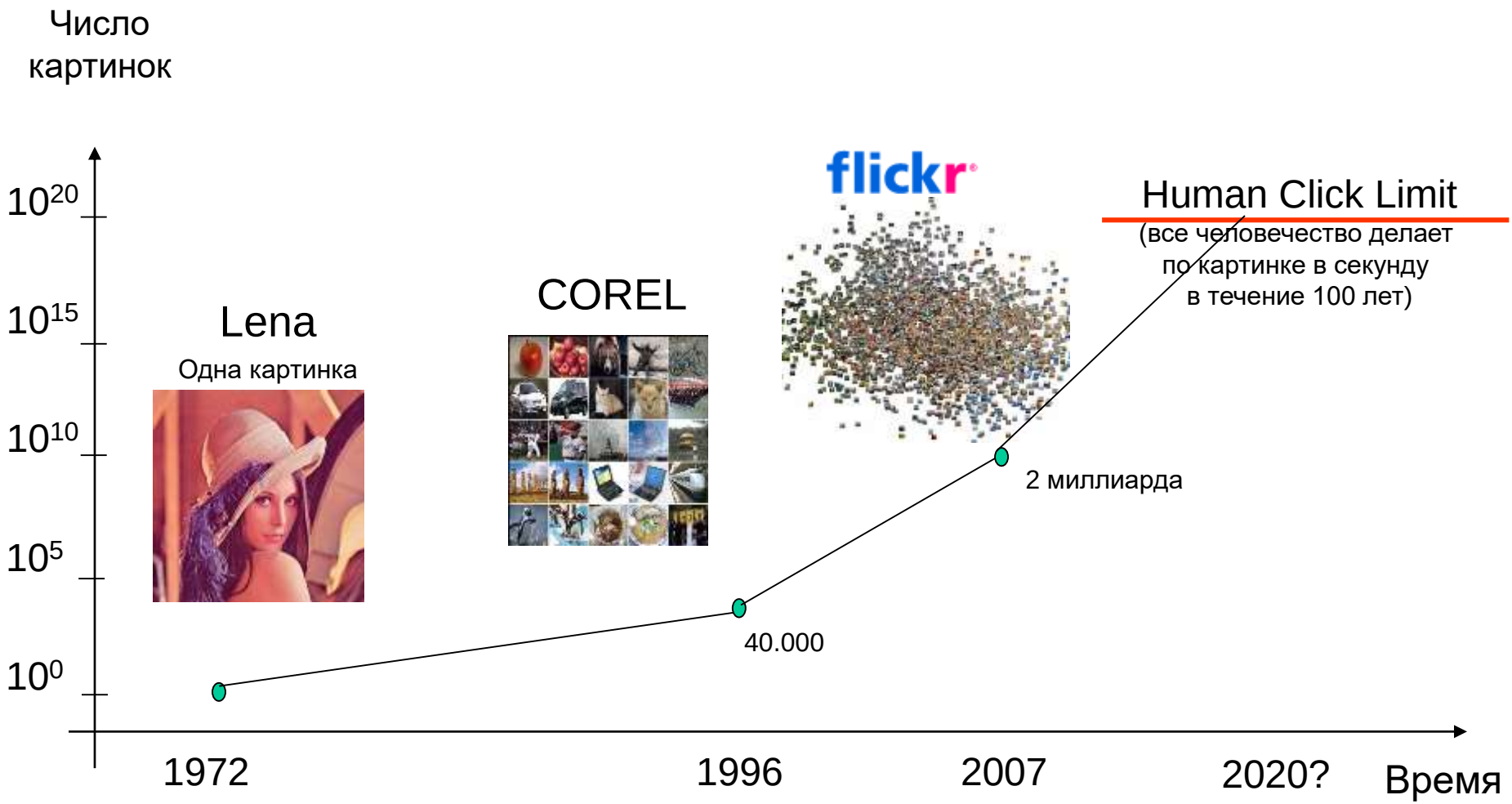
- С помощью свёрточной нейросети из 2х свёрточных слоёв можно реализовать большинство эвристических методов вычисления признаков изображения (гистограммы цветов, HOG, мешки визуальных слов)
- Последующие слои реализуют какие-то признаки «более высокого уровня»
  - Какие именно – хороший вопрос, активно исследуется
- При обучении свёрточной сети эти признаки *обучаются* под решение поставленной задачи, а не задаются пользователем

Переобучение



- Чем сложнее задача – тем более сложная нейросеть нужна
- Но параметрой нейросети очень много, и нейросеть быстро «переобучалась»
- Происходило «запоминание» всей обучающей выборки без её «обобщения»

«Интернет-бум» + «Закон Мура»



Slide by Antonio Torralba

Этапный результат

IMAGENET

14M изображений

Будет 1000 на каждую категорию

http://www.image-net.org

Large-scale visual recognition (LSVR)

Task 1: Classification

Car

- Predict a class label
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 50% of training.

Task 2: Detection (Classification + Localization)

classification Car

- Predict a class label and a bounding box
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 40% of training.

Task 3: Fine-grained classification

classification Walker hound

- Predict a class label given a bounding box in test
- 1 prediction / image
- 120 dog classes (subset)
- ~200 images per class for training (subset)
- Bounding boxes for 100% of training

1.2M изображений для обучения

Нейросети – победители 2012 года

Car

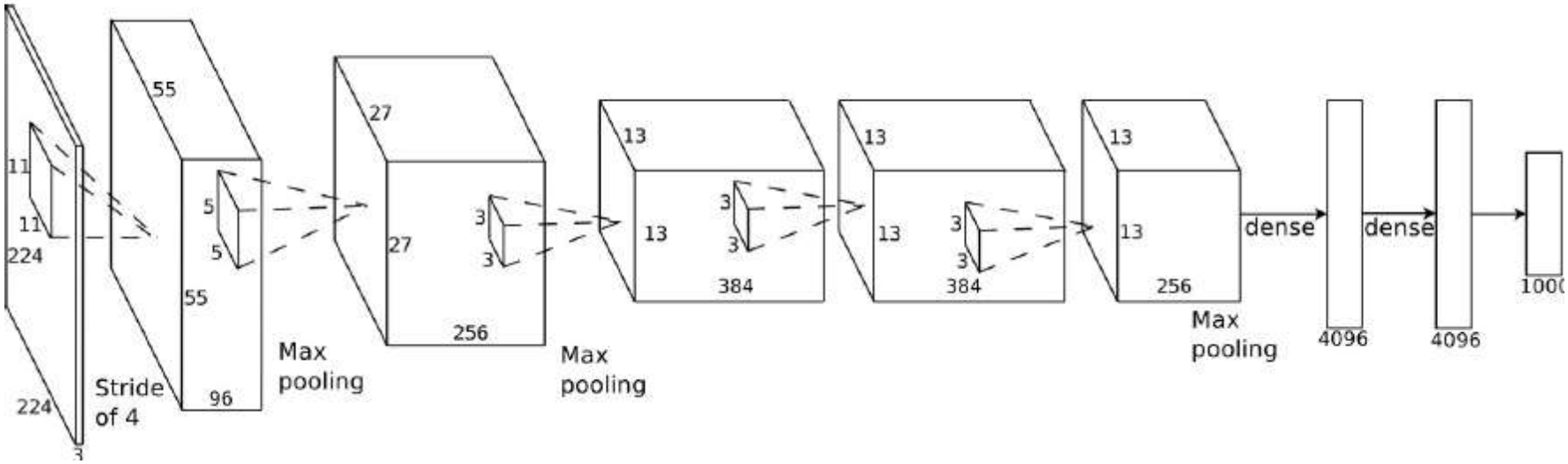
Winner

SuperVision

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton

University of Toronto

SuperVision

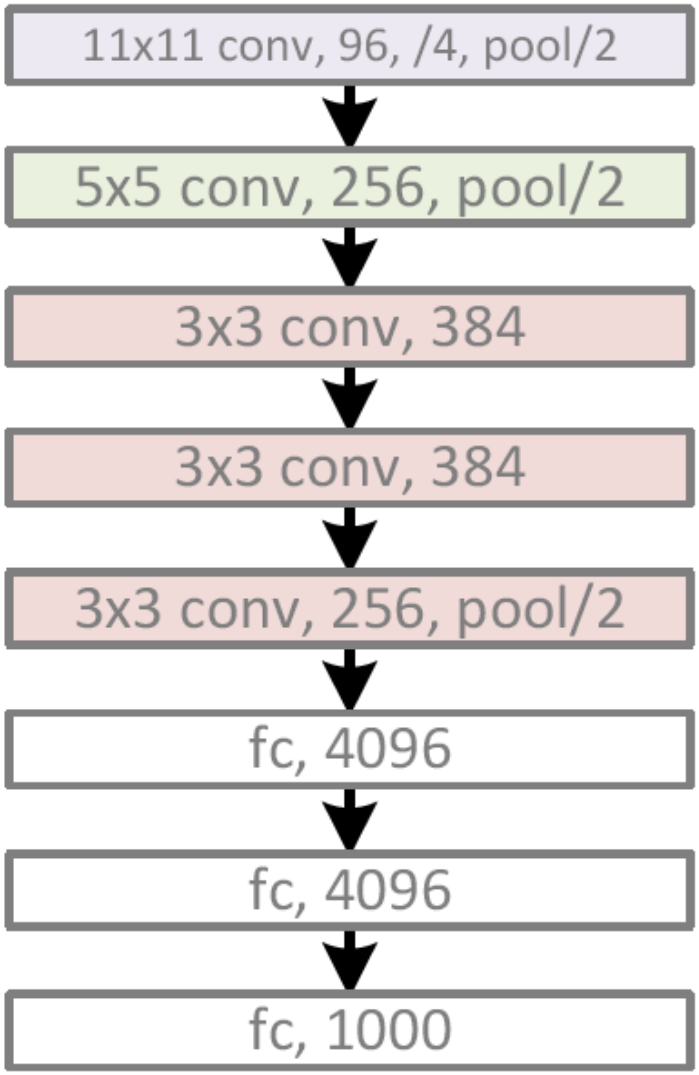


- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- 1 машина, 2 GPU по 2Gb, 5GB Ram, 27Gb HDD, 1 неделя на обучение

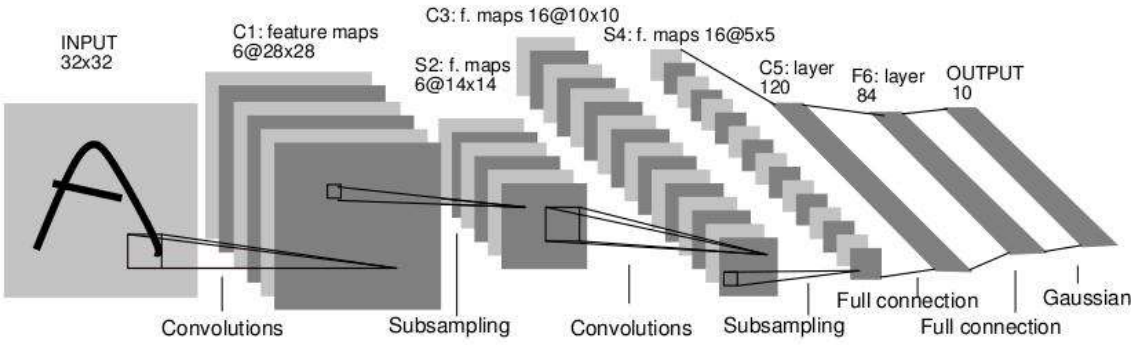
Krizhevsky A., Sutskever I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012

AlexNet

AlexNet, 8 layers
(ILSVRC 2012)

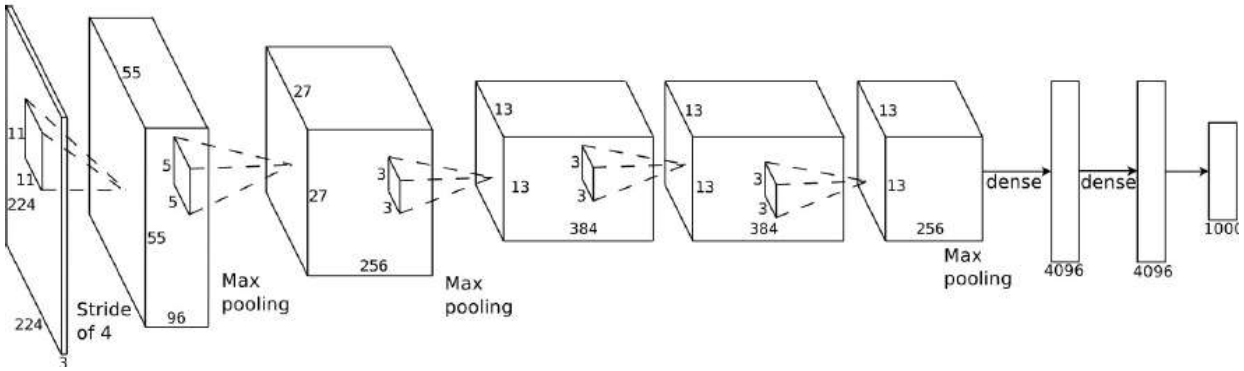


CNN раньше и сейчас



1998 год

- 2 свёрточных слоя (6 и 6 фильтров)
- 2 полносвязанных (120 и 84 нейрона)



2012 год

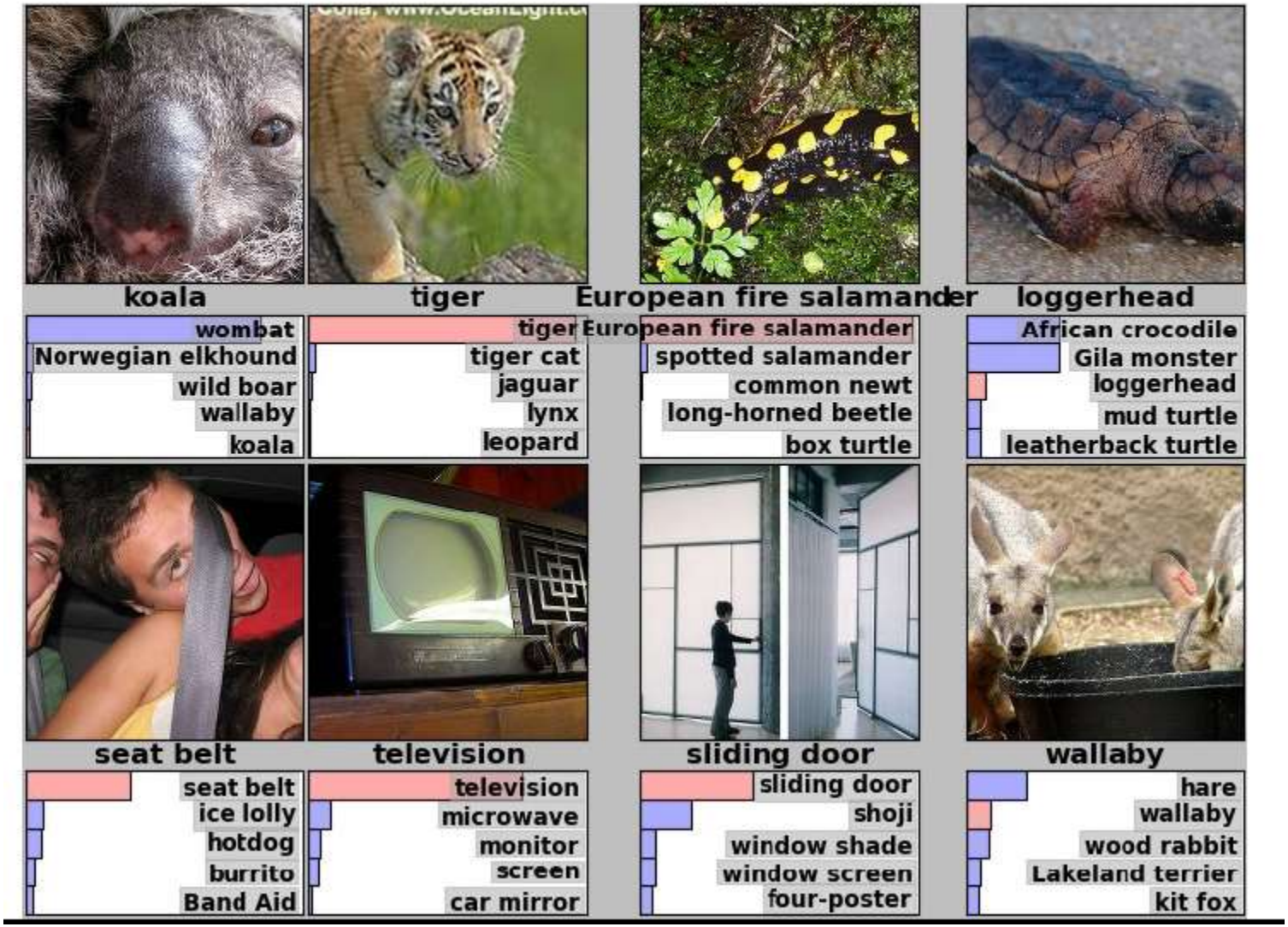
- 5 свёрточных слоёв (96, 256, 384, 384, 256 фильтров)
- 2 полносвязанных (4096 и 4096 нейрона)

- Больше слоёв, фильтров, нейронов
- За счёт большого объёма данных, вычислительной мощности и некоторых улучшений (ReLU и т.д.) смогли обучить такую большую сеть

Krizhevsky A., Sutskever I., Hinton, G. E. (2012) ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012: Neural Information Processing Systems. Lake Tahoe, Nevada.

Примеры работы

Примеры работы



Важно - размножение данных

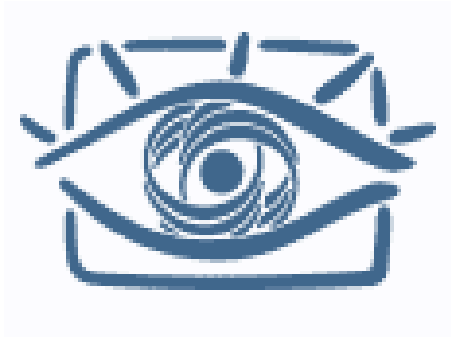
- «Data augmentation»
- Борьба с переобучением
- Из 256x256 случайно выбираем фрагменты 224x224 и их отражения
- Добавляем цветовые искажения

Варианты размножения данных

Небольшие сдвиги, отображения, повороты, изменения масштаба

Резюме

- Концептуально нейросети остались такими же, как в 1990х, но было предложено множество относительно небольших изменений, которые в совокупности с ростом доступных данных позволили сети эффективно обучать
- Есть целый ряд библиотек и уже обученных моделей
- Возможность взять обученную модель и настроить её на другие данные позволяет расширять круг решаемых задач на те, где данных не так много
- Нейросетевые модели стали применять очень широко



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

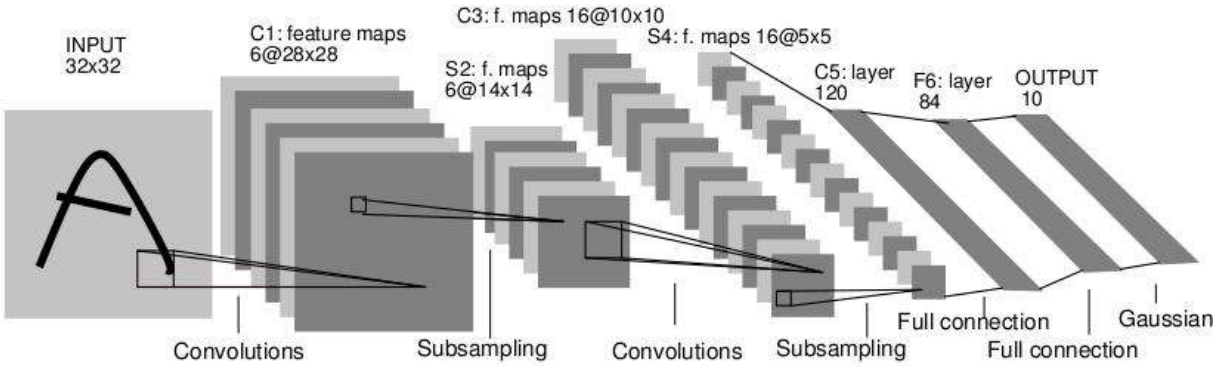
Лекция №6
«Введение в свёрточные нейросети, часть 2»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

21 октября 2019 года

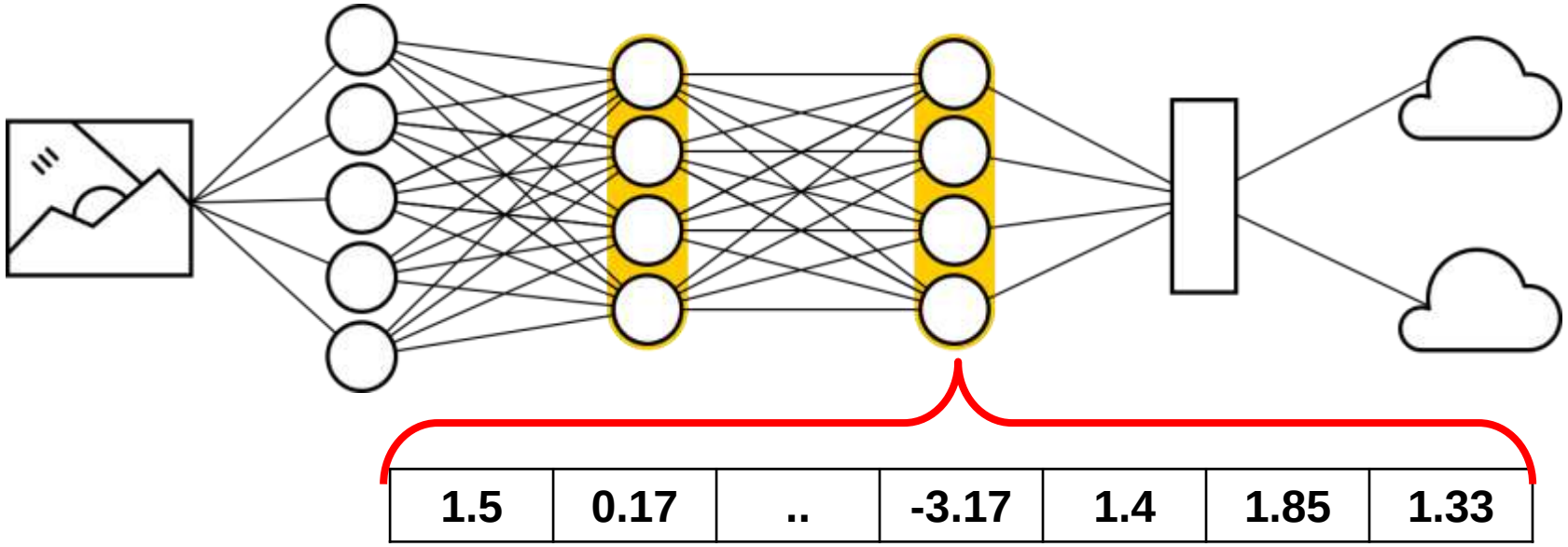
Из прошлой лекции



- С помощью свёрточной нейросети из 2х свёрточных слоёв можно реализовать большинство эвристических методов вычисления признаков изображения (гистограммы цветов, HOG, мешки визуальных слов)
- Последующие слои реализуют какие-то признаки «более высокого уровня»
- При обучении свёрточной сети эти признаки *обучаются* под решение поставленной задачи, а не задаются пользователем
- Визуализация работы нейросети позволяет немного понять, что именно выучивается на глубоких свёрточных слоях

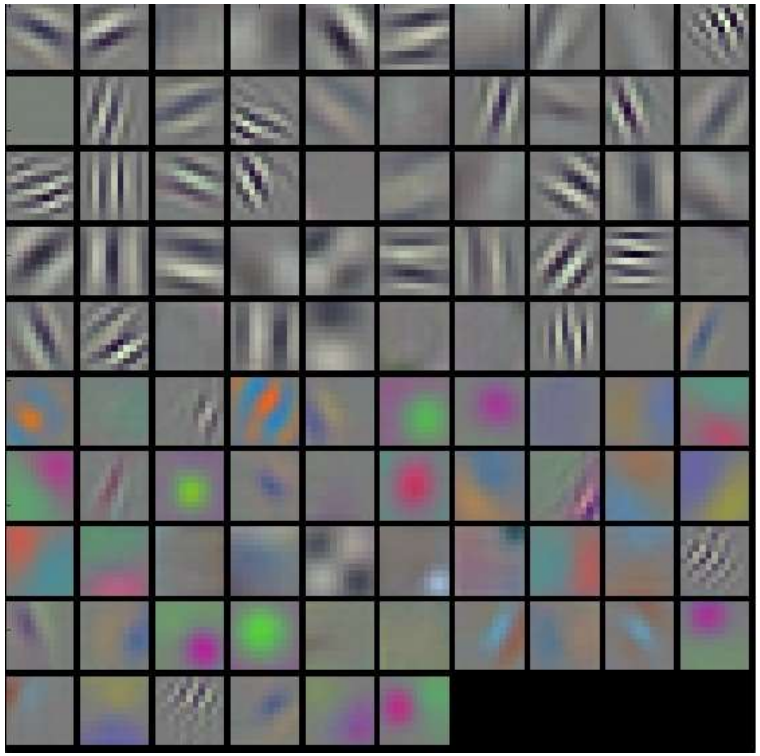
Что происходит внутри сети?

Что будем визуализировать?

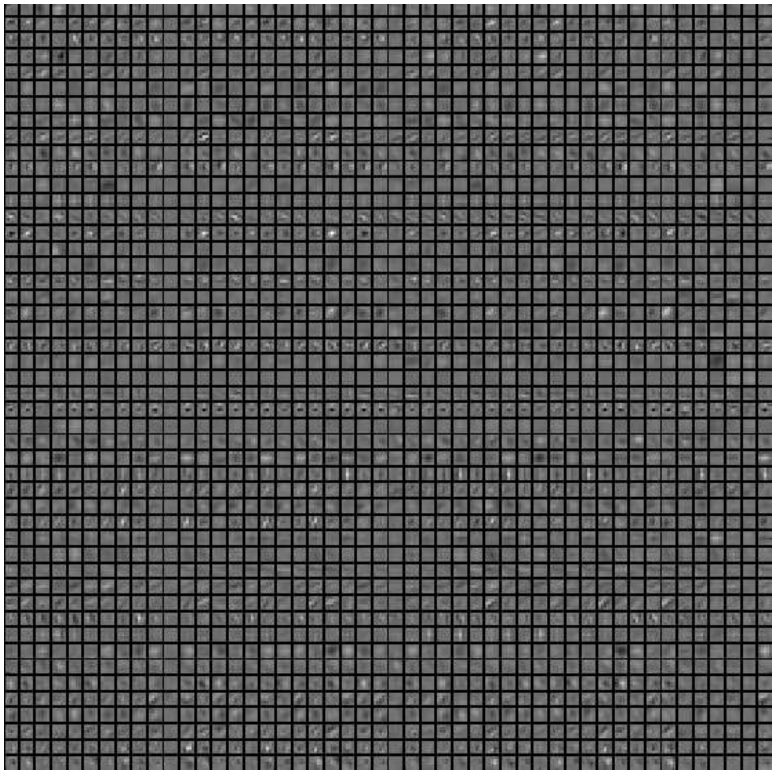


- Веса связей
- Выходы нейронов
- Образы, на которые реагируют нейроны

Визуализация фильтров



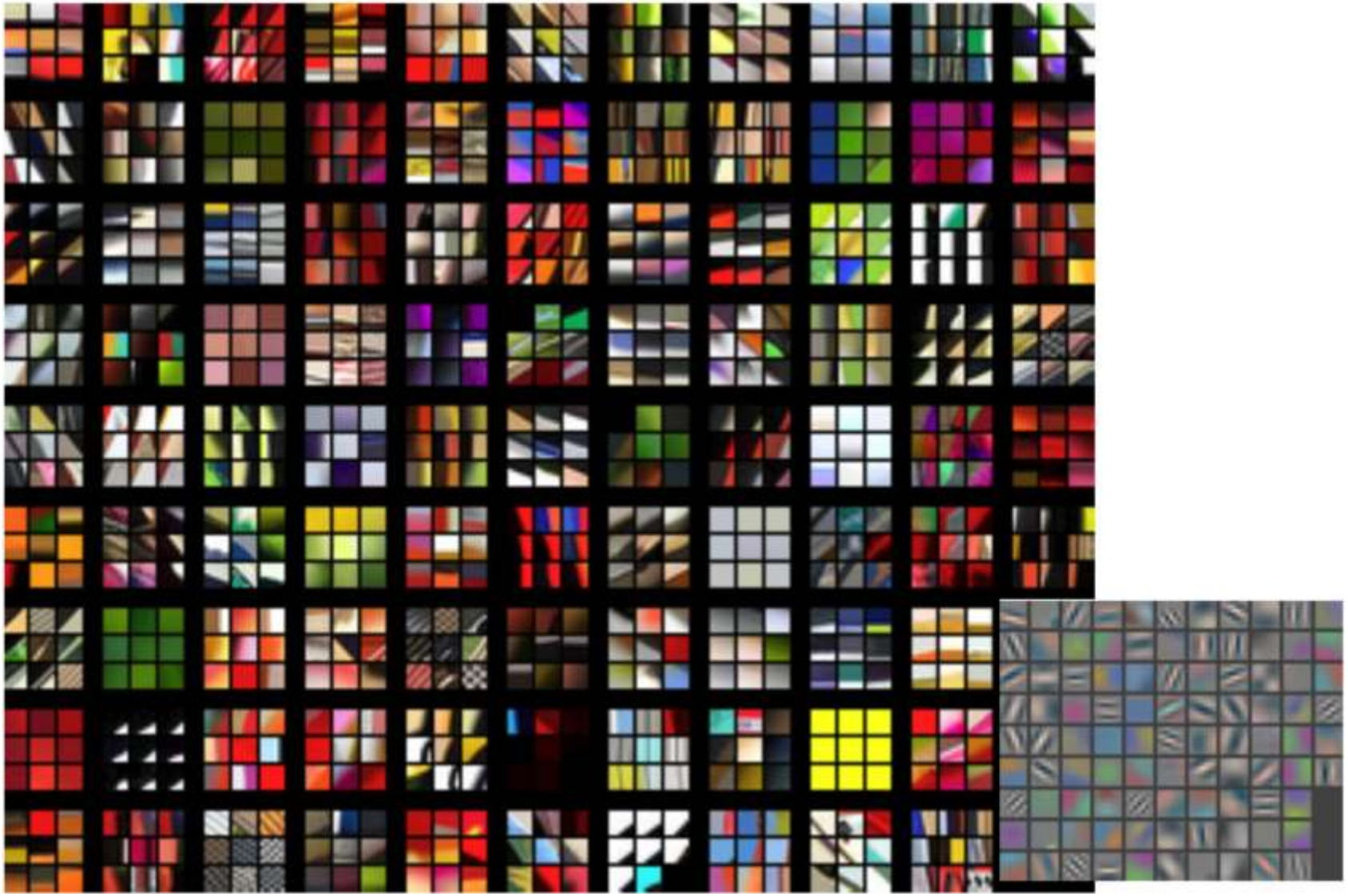
Слой conv1



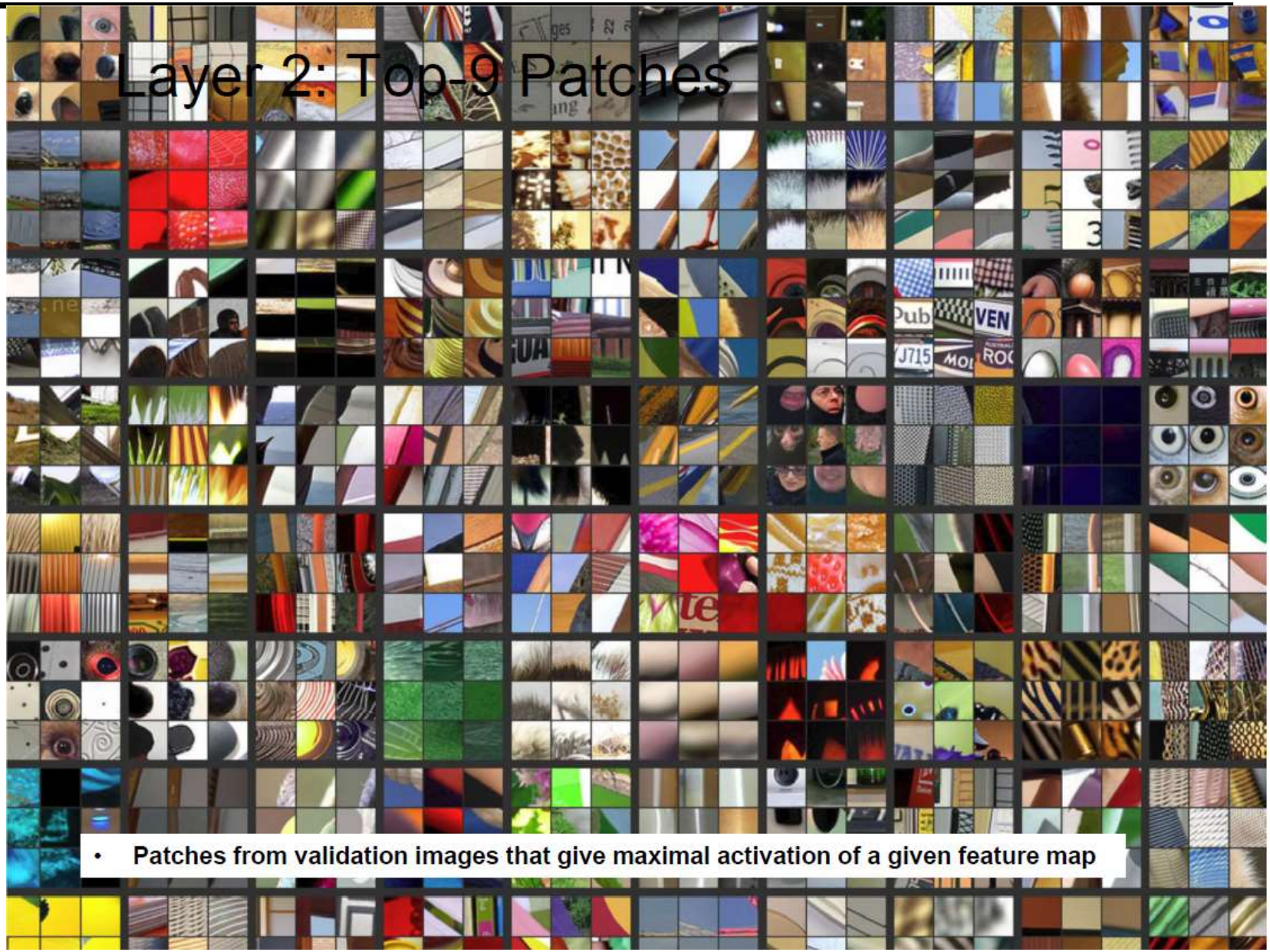
Слой conv2

http://cs231n.github.io/understanding-cnn/

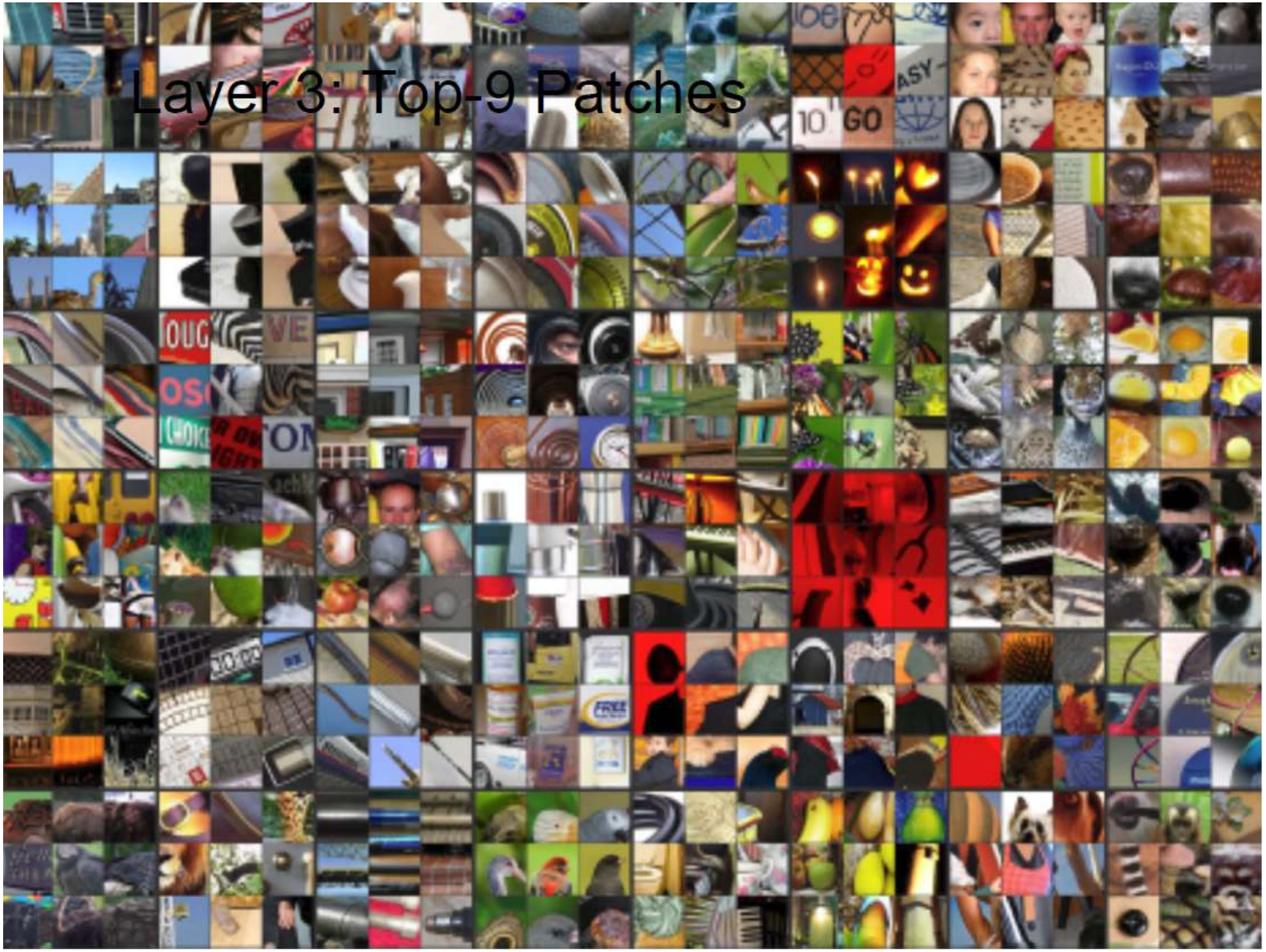
Слой 1: Top-9 фрагментов



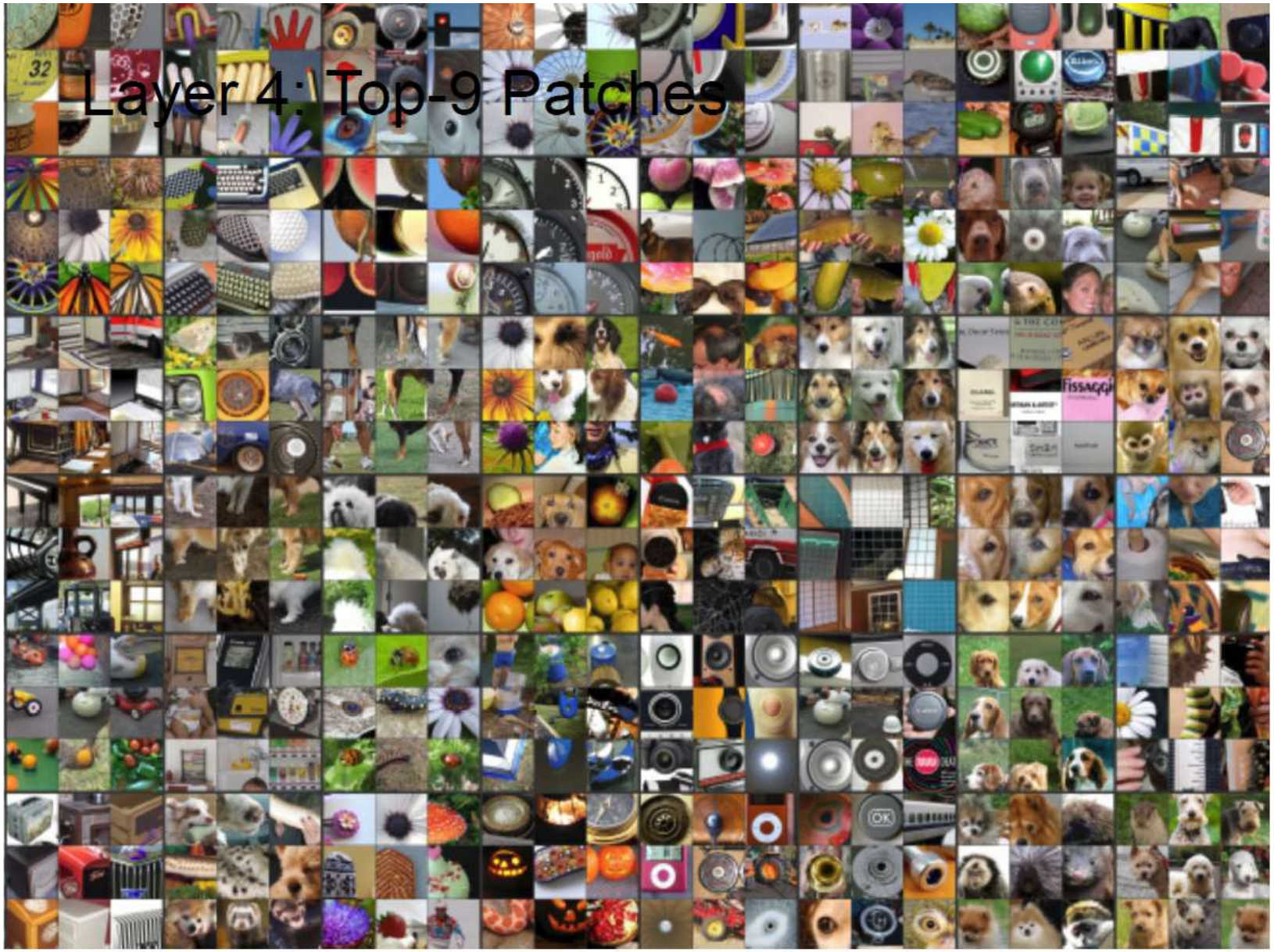
Слой 2: Топ-9 фрагментов



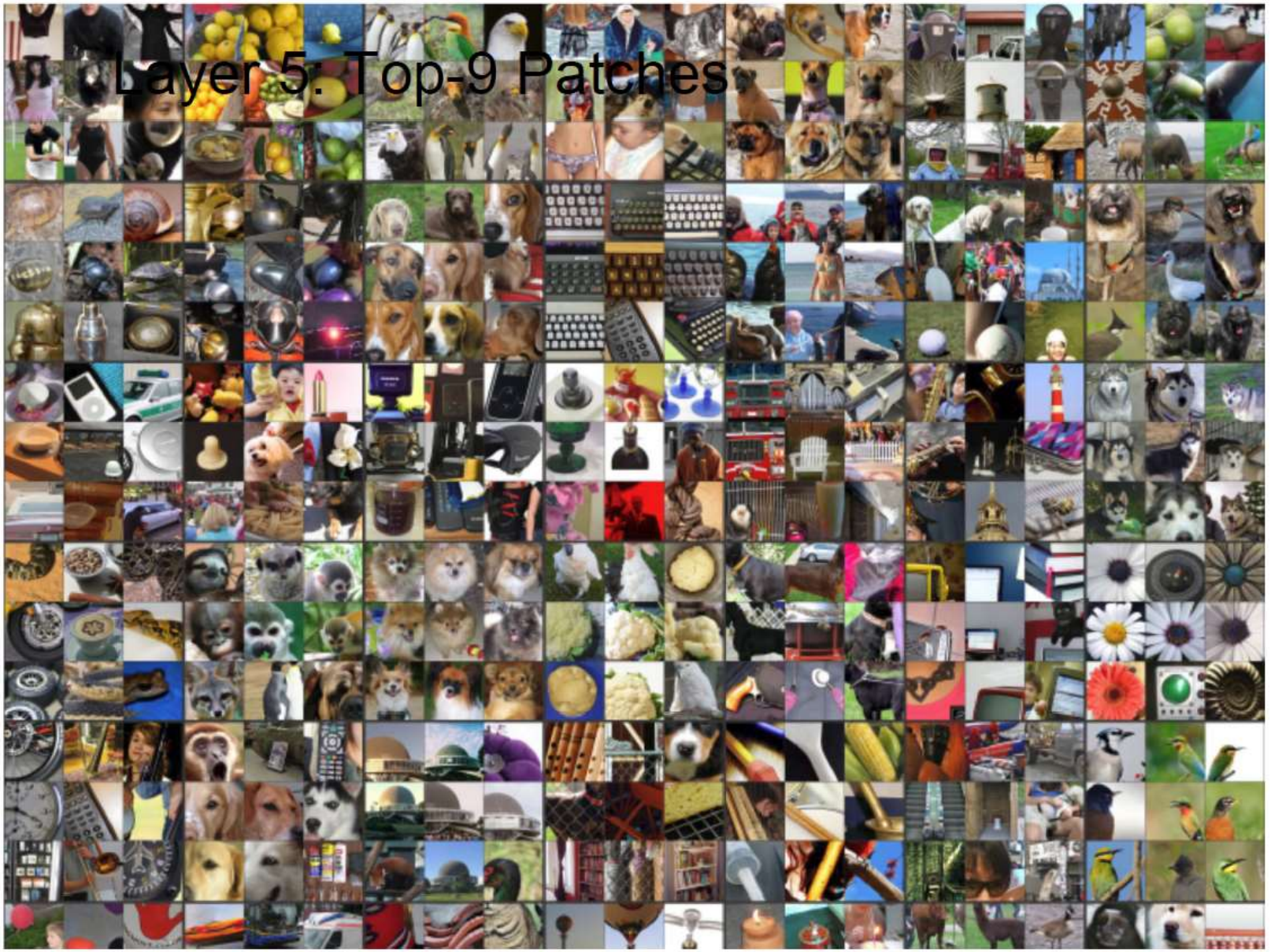
Слой 3: Топ-9 фрагментов



Слой 4: Топ-9 фрагментов



Слой 5: Топ-9 фрагментов



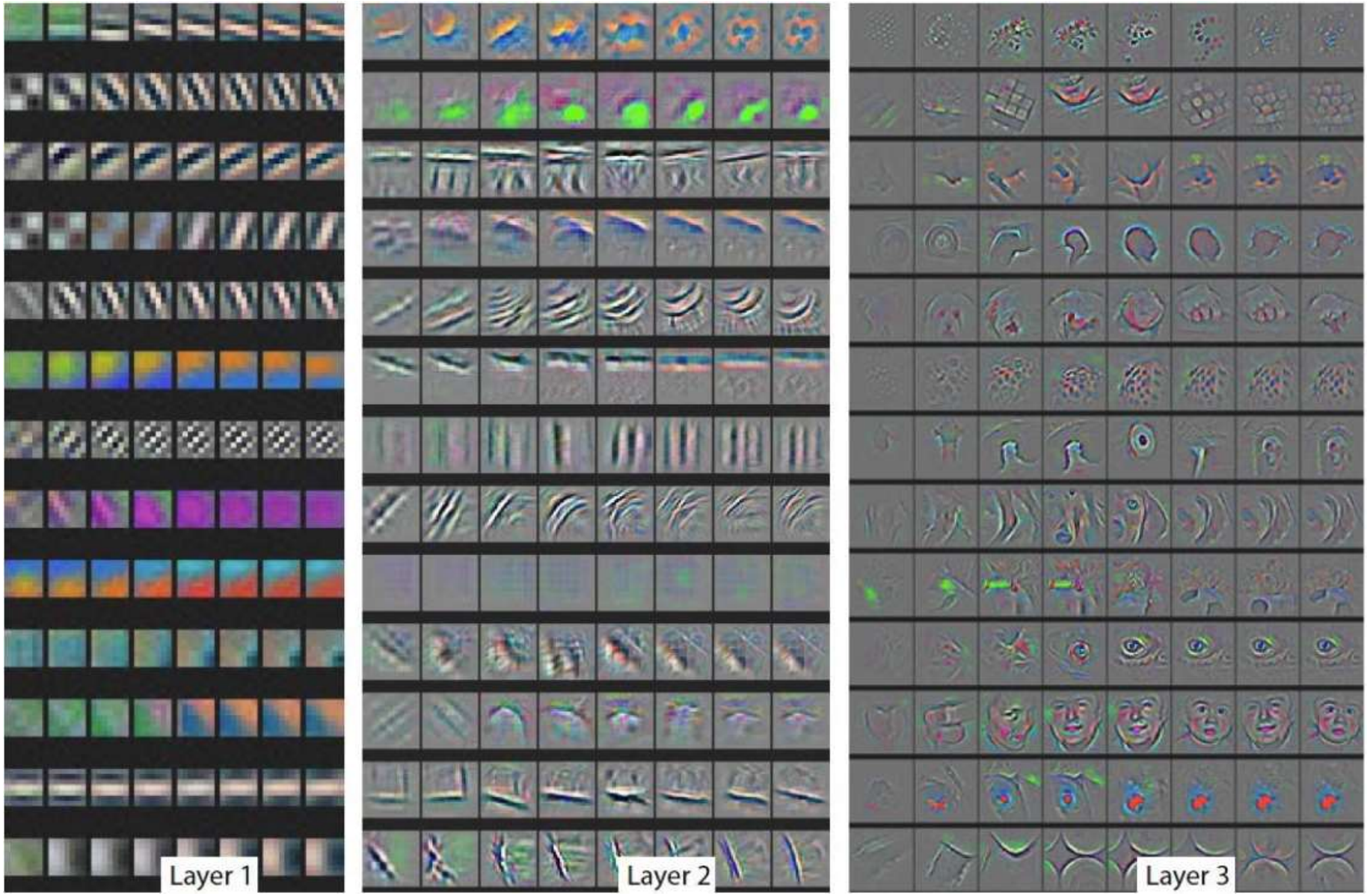
Визуализация работы нейросети

Изображения, на которых достигается максимальный отклик фильтра

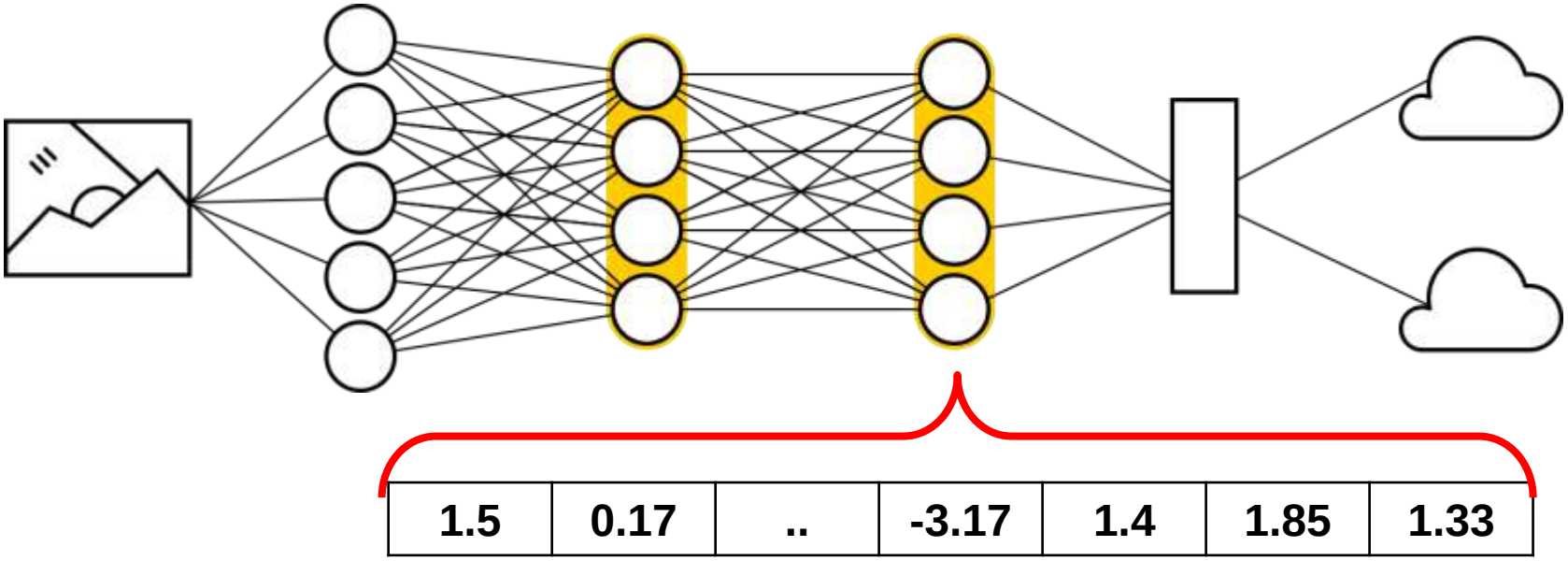
Фильтры слоя pool5

http://cs231n.github.io/understanding-cnn/

Эволюция признаков



Выход слоя как вектор-признак



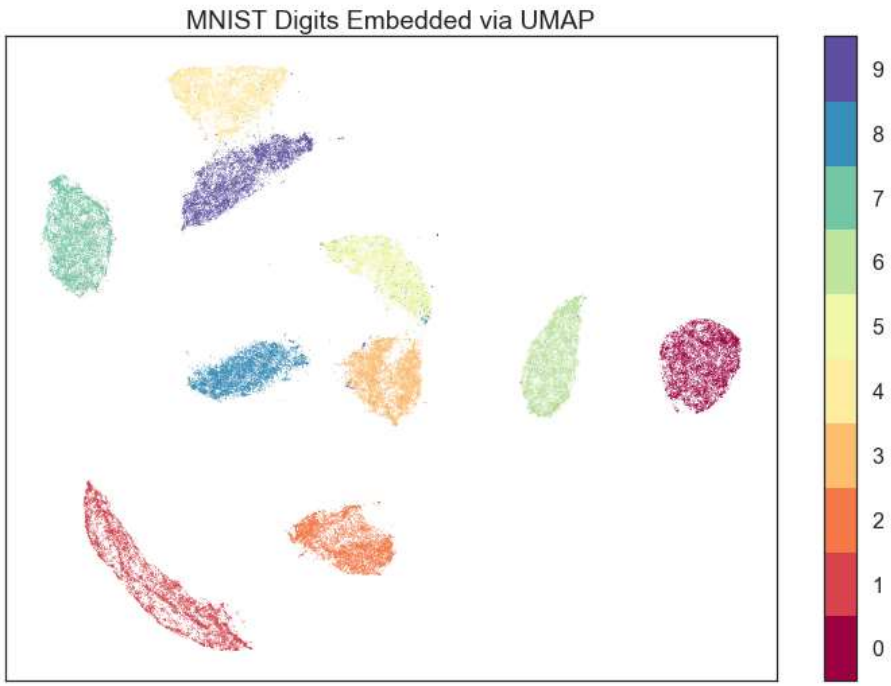
- Мы увидели, что отдельные нейроны каждого слоя несут важное семантическое значение
- Совокупность выходов слоя можно рассматривать как вектор-признак изображения
- Попробуем проанализировать его

t-SNE

- Можем вычислить L2 расстояние между выходами full6 или full7 слоёв
- Воспользуемся отображением точек из 4096-мерного пространства на 2х мерное, сохраняющее L2 расстояния (приближенно)
- Визуализируем изображения
- Видим, что близкие по смыслу изображения оказываются близки друг к другу

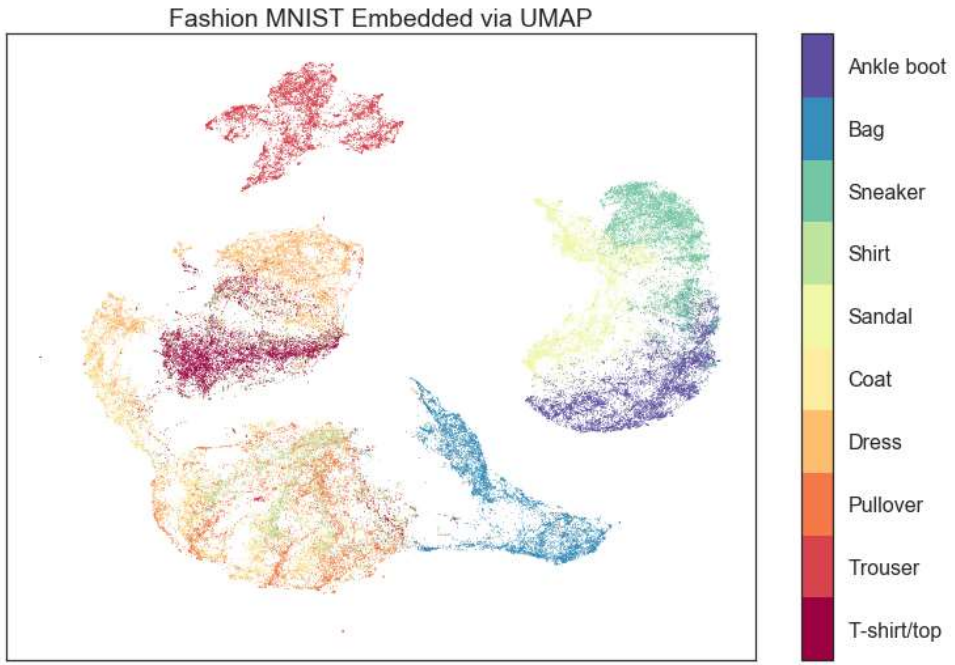
http://cs.stanford.edu/people/karpathy/cnnembed/

UMAP визуализация



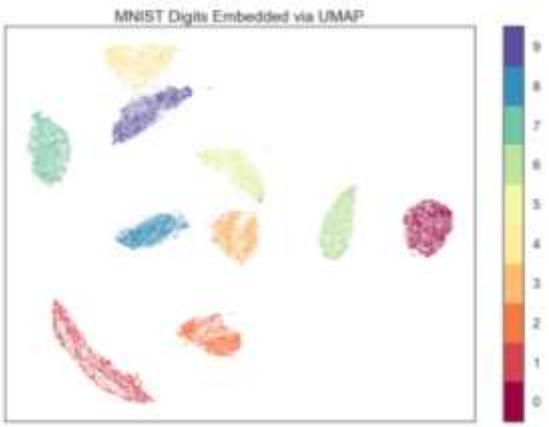
https://github.com/lmcinnes/umap

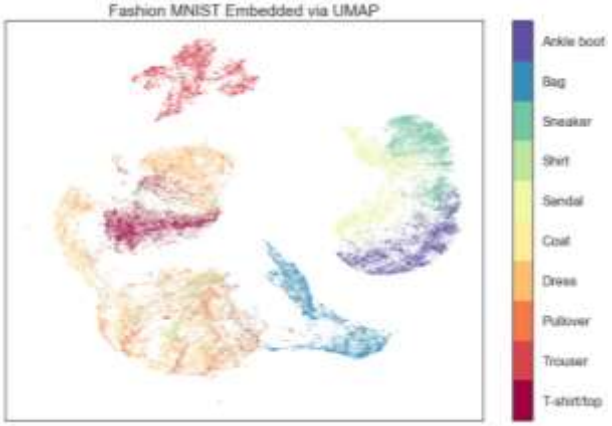
UMAP визуализация



https://github.com/lmcinnes/umap

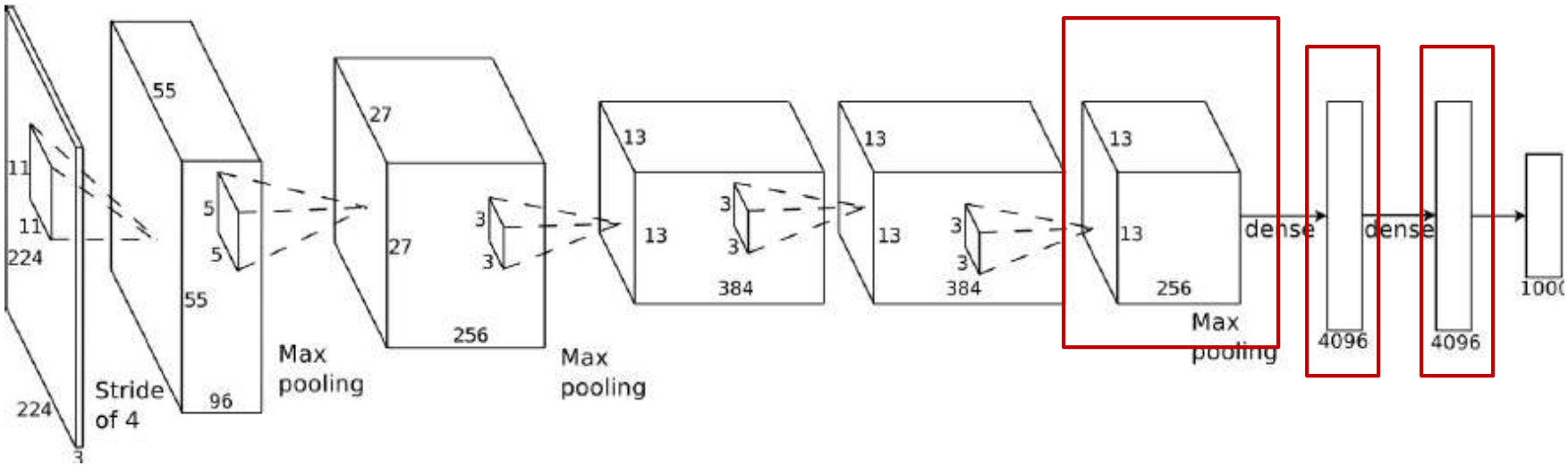
Выводы





- Нейросеть обучается отображать изображения в вектор-признаки, кодирующие семантическую «похожесть» изображений.
- Близкие и визуально похожие друг на друга объекты оказываются близки по нейросетевым вектор-признакам.
- Вектор-признаки объектов разных классов далеки друг от друга

Нейросетевые признаки



- Можно использовать выходы выбранного слоя нейросети как вектор-признак
- Можно обучить сеть на одних данных (ImageNet) и применять её на других для вычисления признаков, обучая поверх классификатор

Donahue et. al.DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, 2013

Распознавание на других базах

	$DeCAF_5$	$DeCAF_6$	$DeCAF_7$
LogReg	63.29 ± 6.6	84.30 ± 1.6	84.87 ± 0.6
LogReg with Dropout	-	86.08 ± 0.8	85.68 ± 0.6
SVM	77.12 ± 1.1	84.77 ± 1.2	83.24 ± 1.2
SVM with Dropout	-	$\mathbf{86.91 \pm 0.7}$	85.51 ± 0.9
Yang et al. (2009)		84.3	
Jarrett et al. (2009)		65.5	

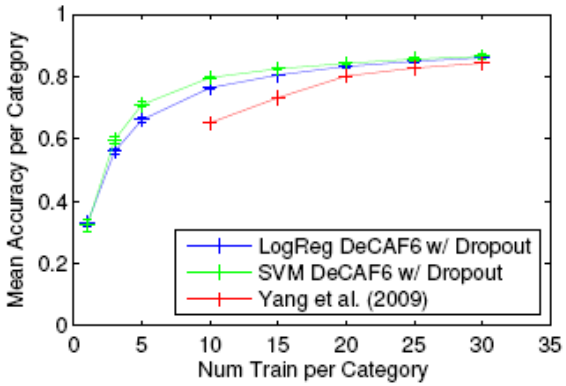


Figure 4. Left: average accuracy per class on Caltech-101 with 30 training samples per class across three hidden layers of the network and two classifiers. Our result from the training protocol/classifier combination with the best validation accuracy – SVM with Layer 6 (+ dropout) features – is shown in **bold**. Right: average accuracy per class on Caltech-101 at varying training set sizes.

Обучили нейросеть для извлечения признаков на ImageNet и применили для Caltech 101

Donahue et. al.DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, 2013

Классификация близких объектов

Method	Part info	mean Accuracy
Sift+Color+SVM[45]	✗	17.3
Pose pooling kernel[49]	✓	28.2
RF[47]	✓	19.2
DPD[50]	✓	51.0
Poof[5]	✓	56.8
CNN-SVM	✗	53.3
CNNaug-SVM	✗	**61.8**
DPD+CNN(DeCaf)+LogReg[10]	✓	**65.0**

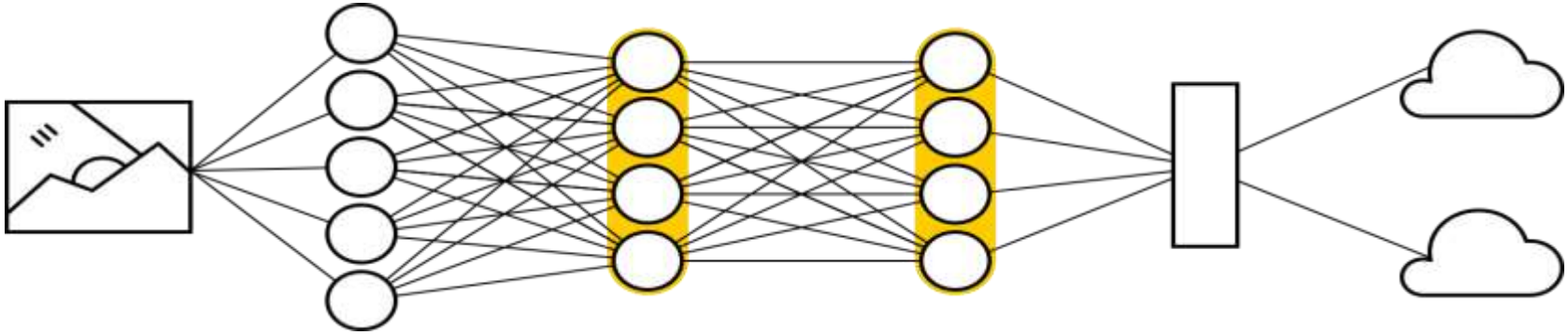
Table 3: **Results on CUB 200-2011 Bird dataset.** The table distinguishes between methods which use part annotations for training and sometimes for evaluation as well and those that do not. [10] generates a pose-normalized CNN representation using DPD [50] detectors which significantly boosts the results to **64.96**.

- Fine-graded classification – например, определение видов птиц
- Возьмём нейросеть, обученную для классификации ImageNet
- Применим её для получения вектор-признаков изображений
  Обучаем классификатор поверх этих признаков
- Profit!

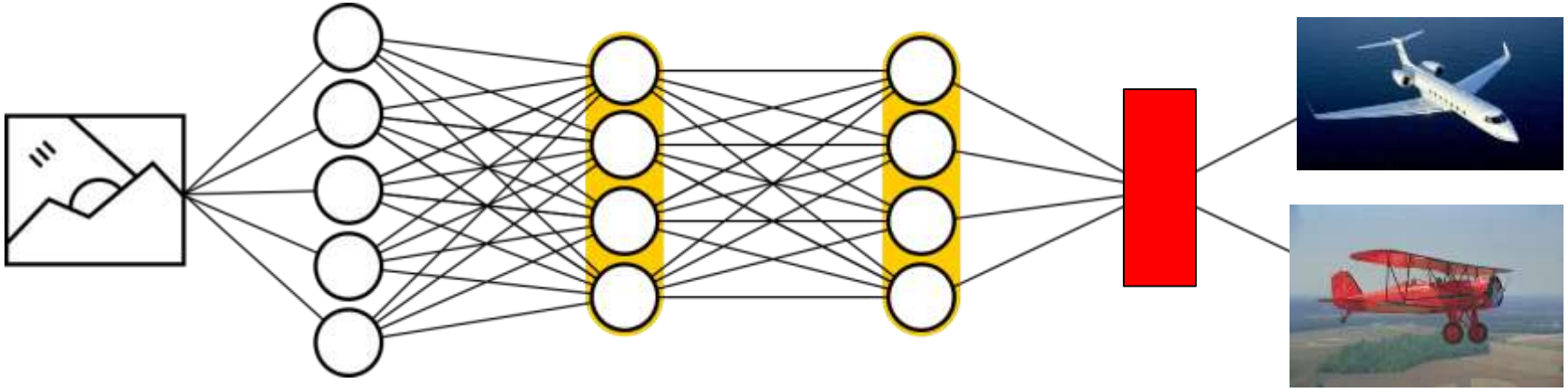
Ali Sharif Razavian Hossein Azizpour Josephine Sullivan Stefan Carlsson
CNN Features off-the-shelf: an Astounding Baseline for Recognition. 2014

Дообучение (fine-tuning)

Возьмём сеть A, обученную на одной коллекции (например, ImageNet)

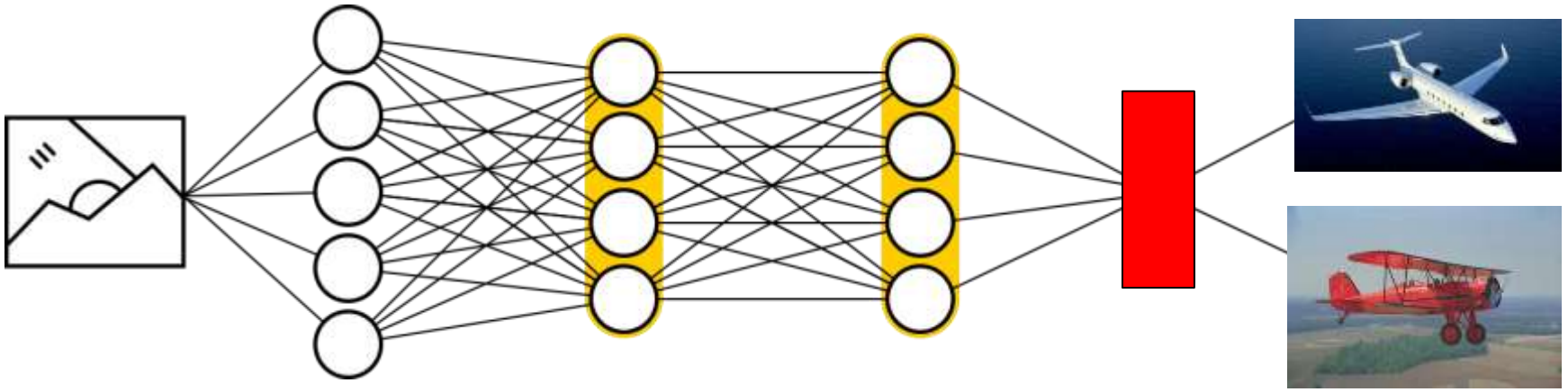


Заменим в ней последний слой (классификатор)



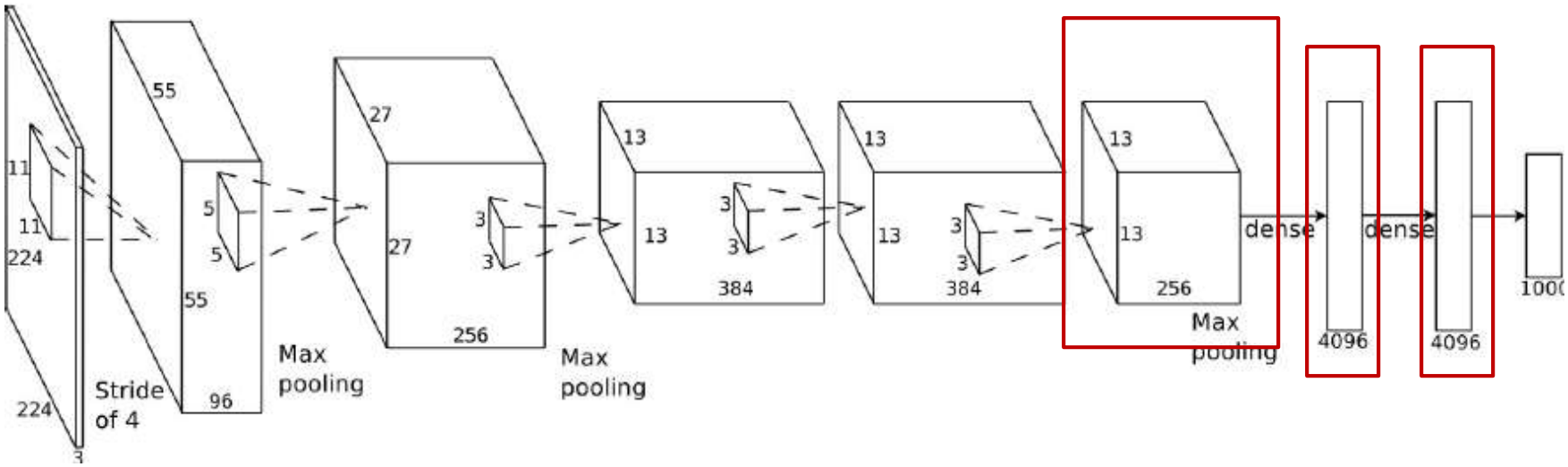
Обучим («дообучим») эту сеть на новой коллекции для решения новой задачи

Плюсы дообучения



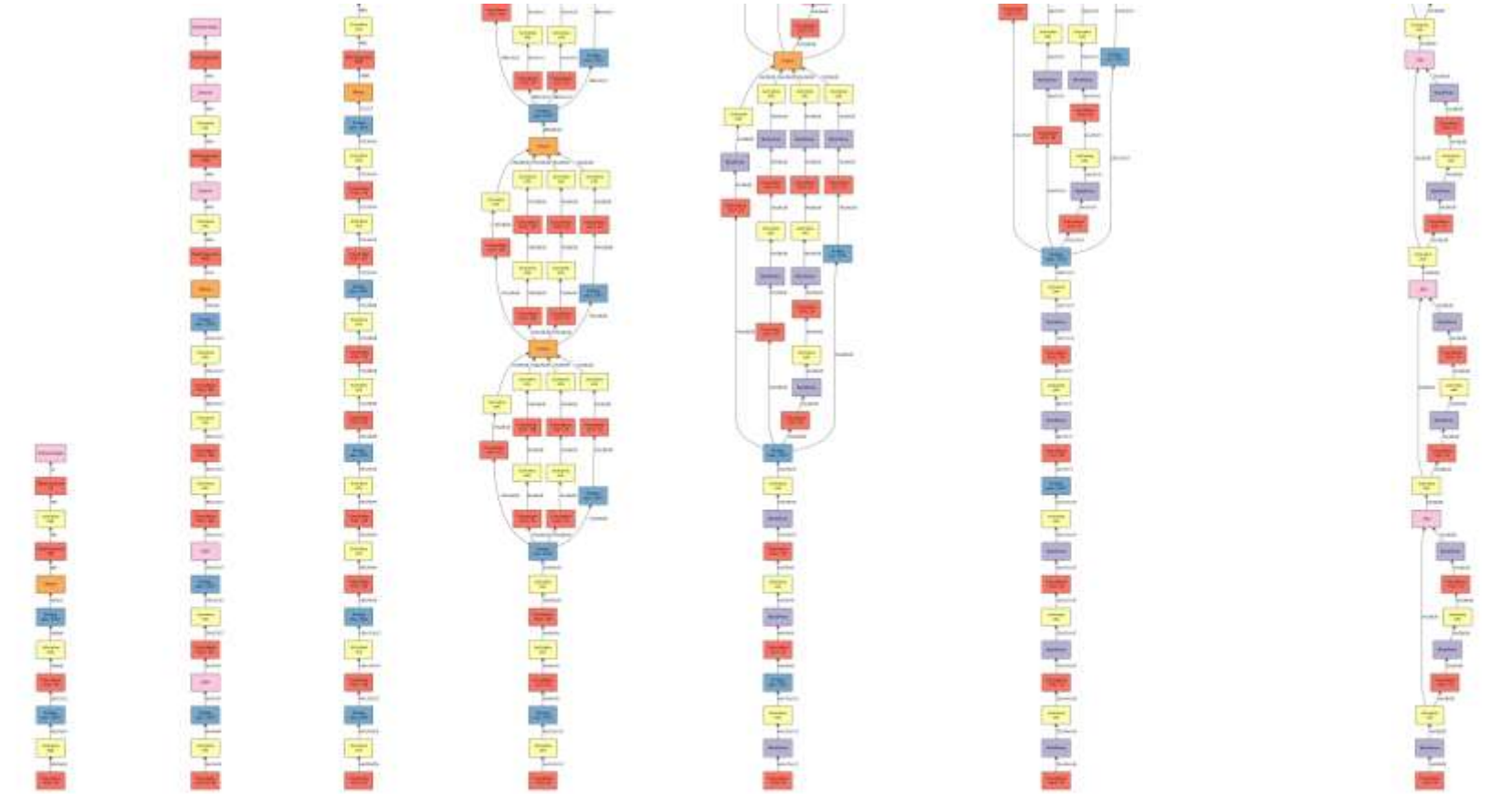
- Вторая коллекция может быть недостаточного размера для обучения нейросети «с нуля» (from scratch)
- Предобучение на большой и разнообразной коллекции может «хорошо» инициализировать новую сеть, и дообучится она быстрее и лучше

Идея «базовой» архитектуры

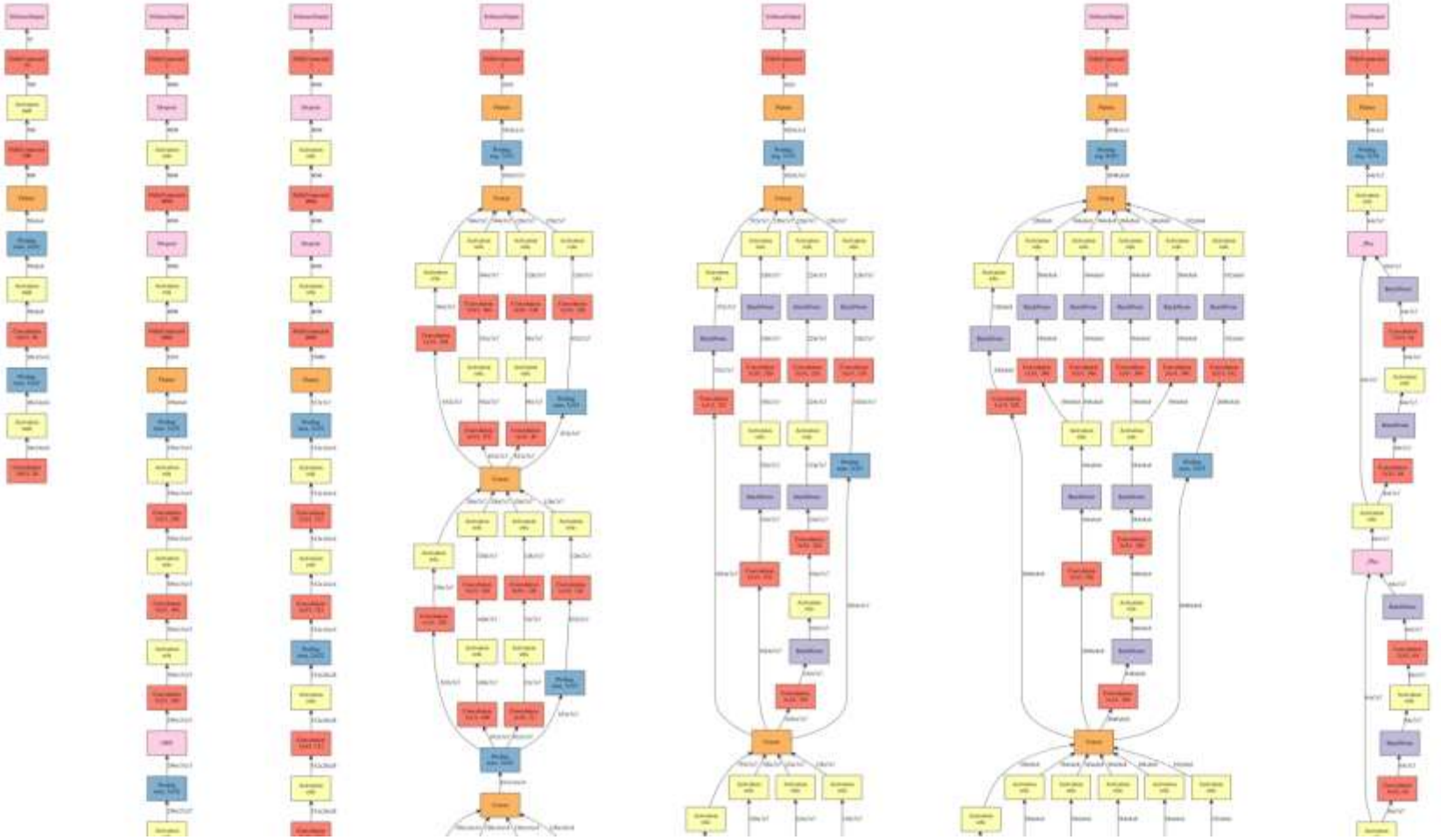


- Можем обучать нейросети-классификаторы на больших коллекциях («базовые сети»), ииспользовать их как основу для решения новых задач (дообучением)
- Создали коллекции моделей (zoo – зоопарки)
- Пока в основном обучают на ImageNet

Развитие базовых архитектур

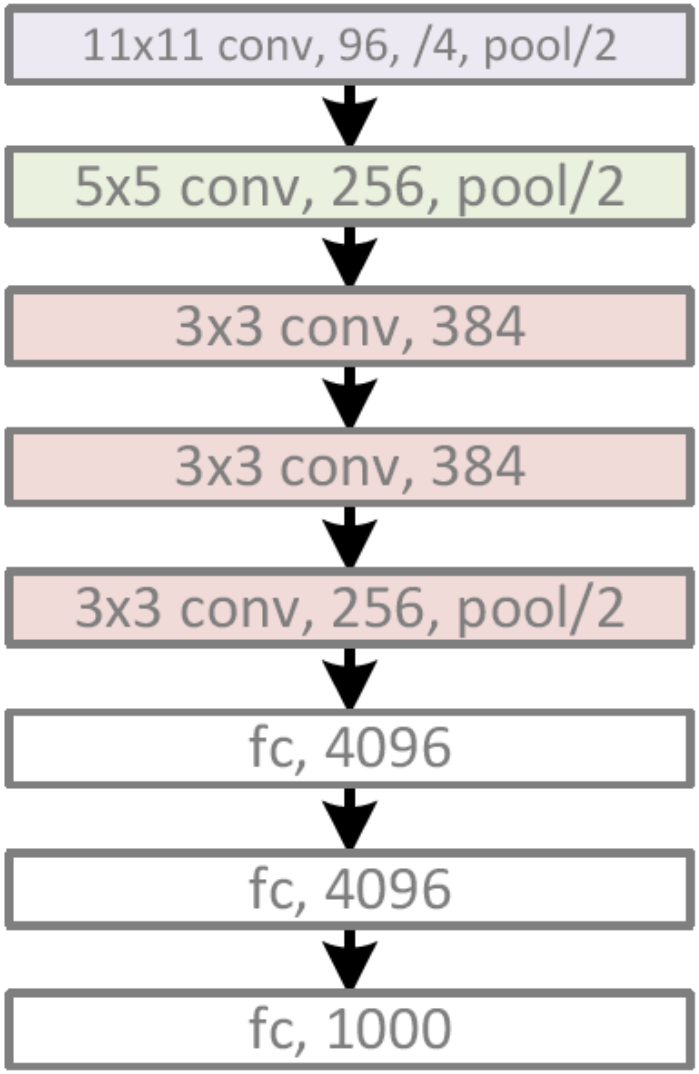


Развитие базовых архитектур



AlexNet

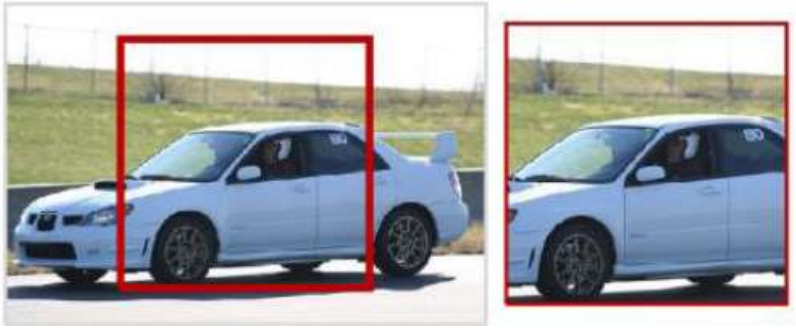
AlexNet, 8 layers
(ILSVRC 2012)



Krizhevsky A., Sutskever I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012

Применение AlexNet

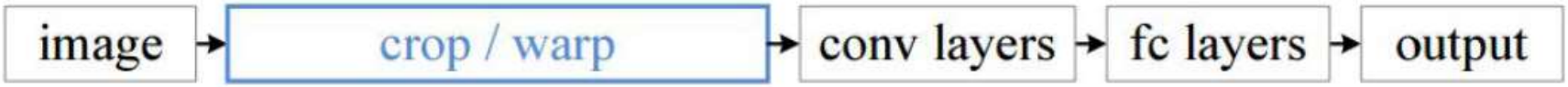
- На вход нейросеть принимает изображение фиксированного размера 224x224 пиксела. Можем ли мы подать другое?
- Как быть с изображениями других размеров?
- 2 варианта действий:



Масштабировать к разным разрешениям, пр. [256;512]xN случайно и вырезать фиксированные (image crop)

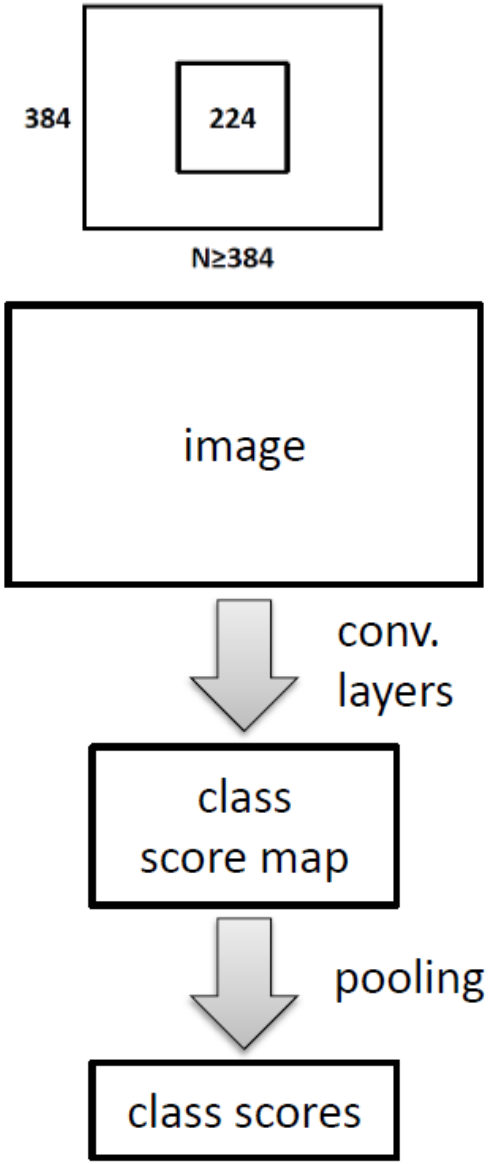
Приводить к фиксированному разрешению (image warp)

- Схема применения



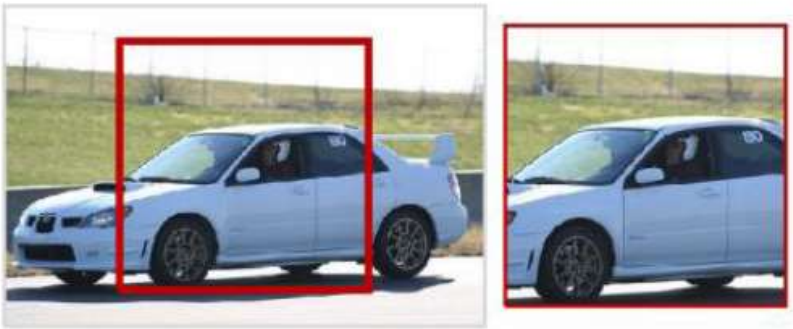
Применение моделей

- Вариант 1
  - Применить к фиксированному изображению, как сказали раньше
- Вариант 2
  - Применить несколько раз к разным версиям изображения
- Вариант 3
  - Применить ко всем фрагментам заданного разрешения, т.е. «просканировать» изображение



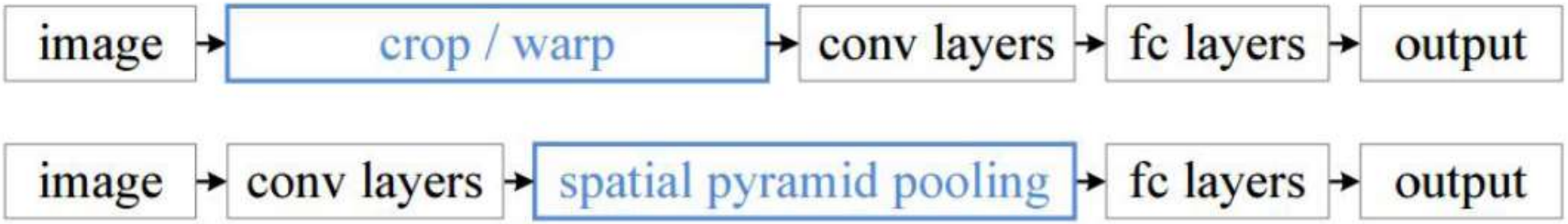
Spatial Pyramid Pooling (SPP)

Идея: преобразовать выходы свёрточного слоя (признаки) произвольного разрешения к вектор-признаку фиксированной длины

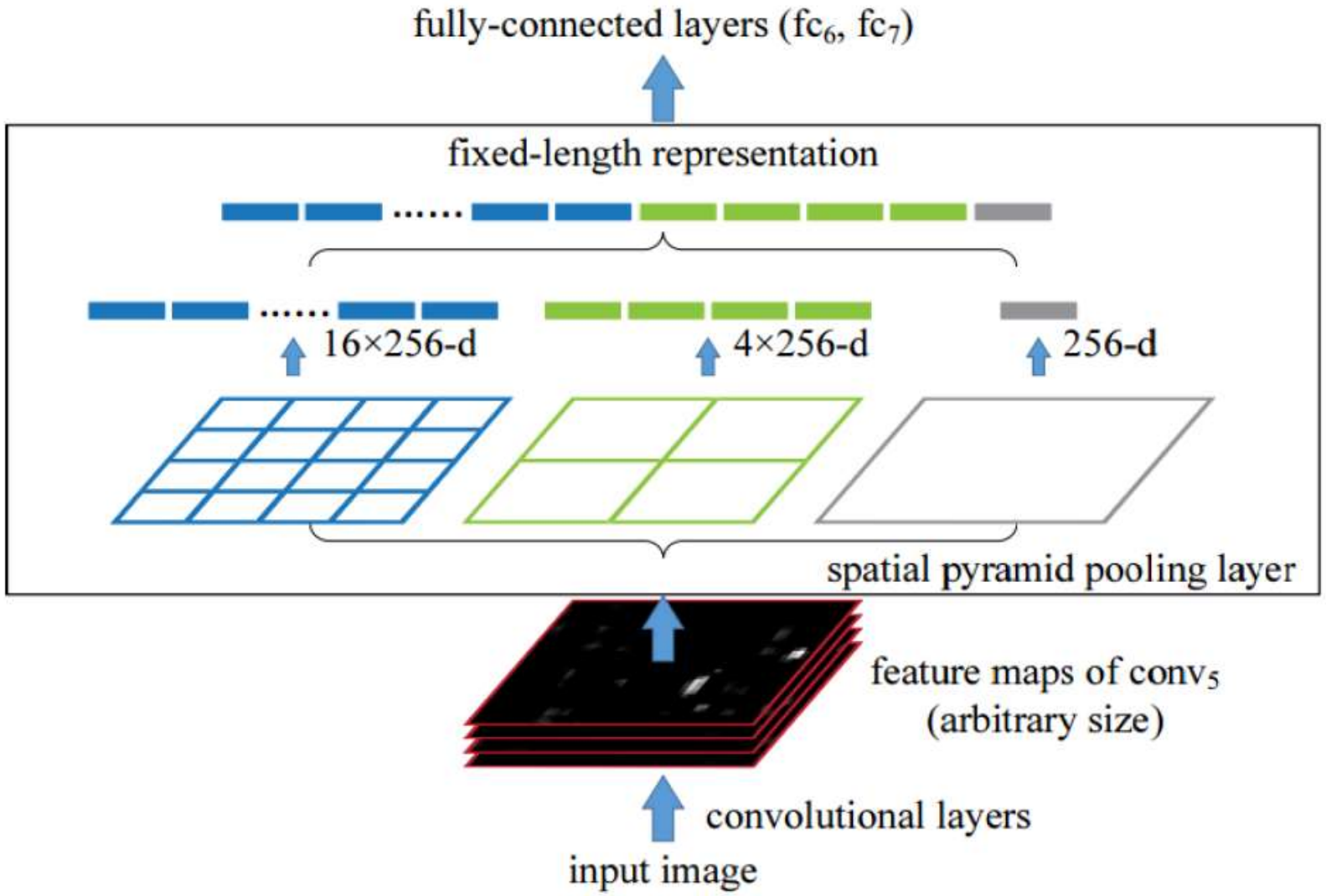


crop

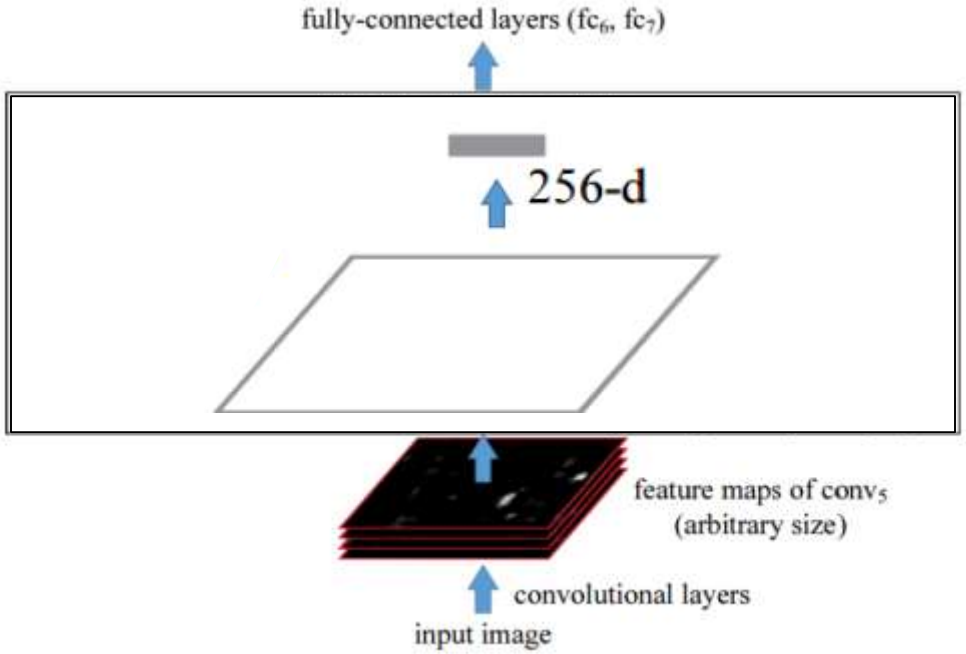
warp



Spatial Pyramid Pooling (SPP)



Average Pooling



- Можно обойтись без пирамиды, ограничившись только Average Pooling по всему слою
- Тогда у нас получится вектор признак длиной равной числу свёрток на последнем уровне
- Для ряда задач этого оказывается достаточно

Очень глубокие (VGG)

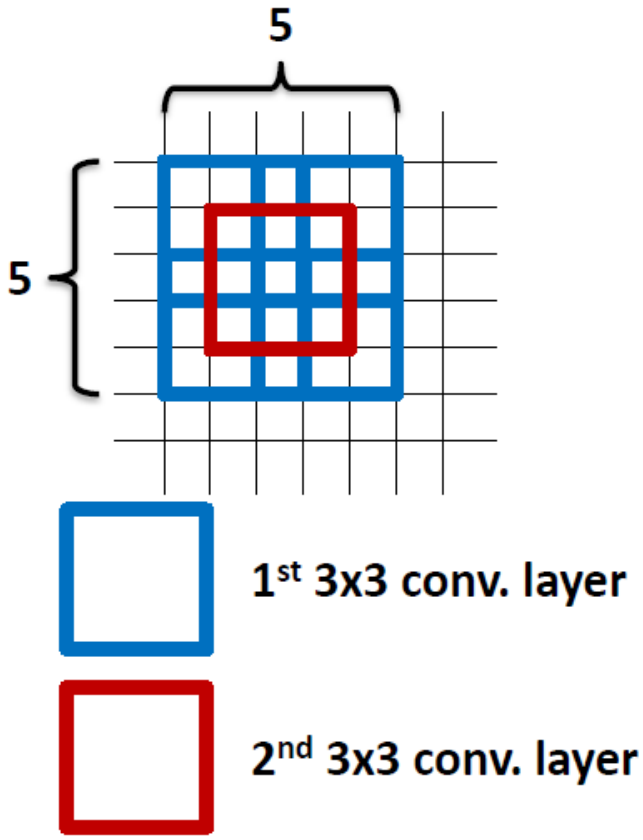
Идеи:

- Исследовать рост качества за счёт увеличения глубины нейросети
- Использовать только маленькие 3x3 свёртки
- Stride 1 в свёртках чтобы не терять информацию
- ReLU активация
- Нет нормализации
- Уменьшение разрешения через maxpooling
- Число фильтров x2 при уменьшении разрешения в 2 раза

K. Simonyan, A. Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition . ICLR 2015

Свёртки 3x3

- Стек свёрток позволяет обеспечить бОльшее рецептивное поле (reception field)
  - 5x5 для 2-х свёрток
  - 7x7 для 3-х свёрток
- БОльшая нелинейность за счёт ReLU активаций
- Меньше параметров
  - 18x (2 3x3) vs 25x (5x5)
  - 27x (3 3x3) vs 49x (7x7)



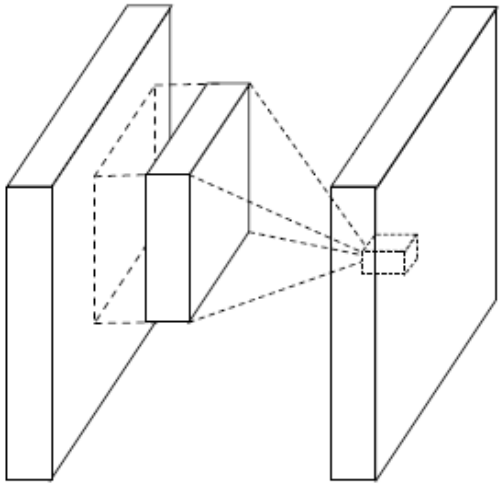
Исследование вариантов

A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

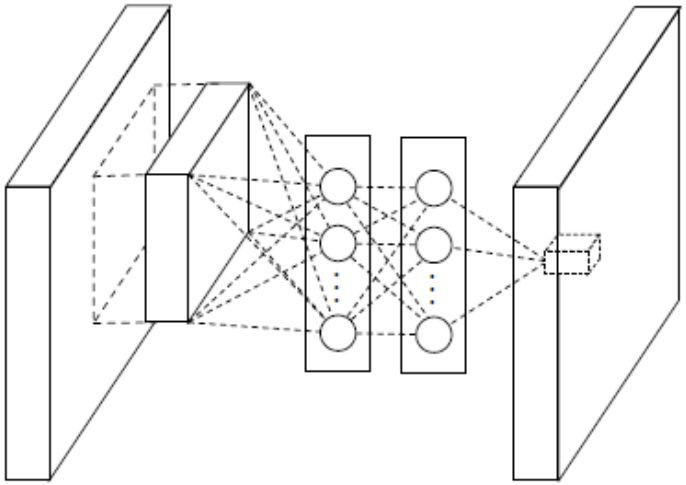
D - VGG-16

E - VGG-19

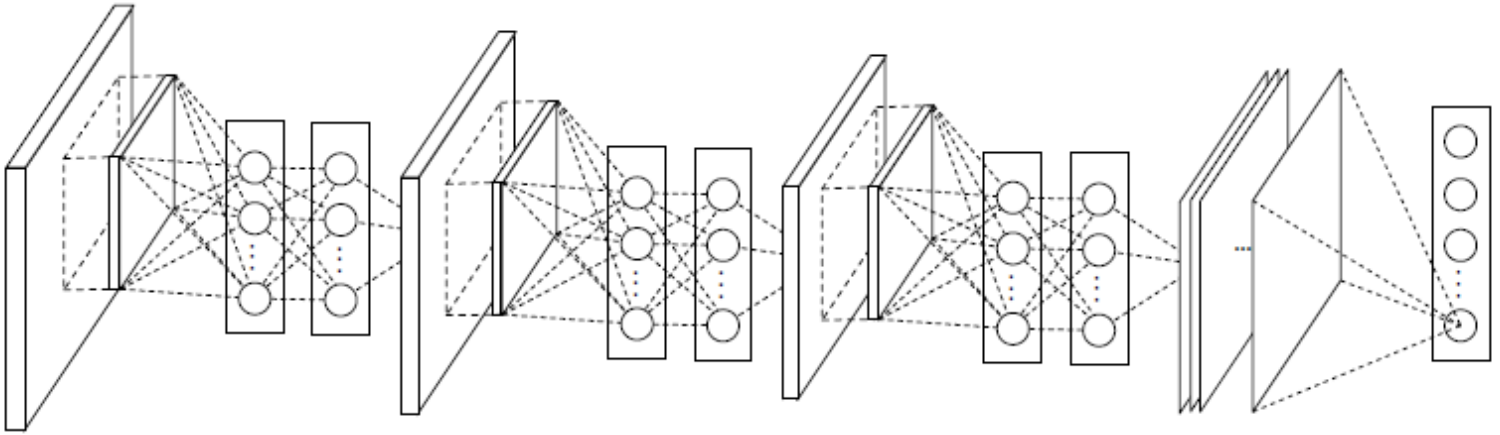
Network in Network



(a) Linear convolution layer

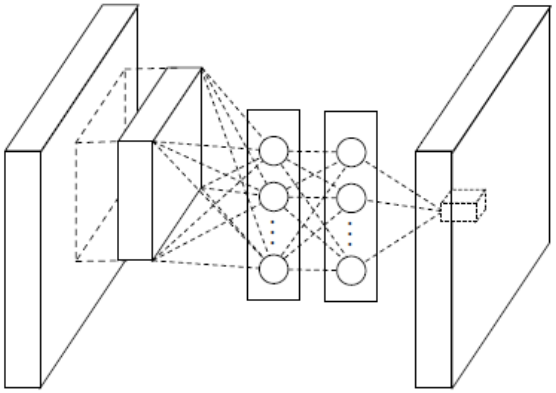


(b) Mlpconv layer

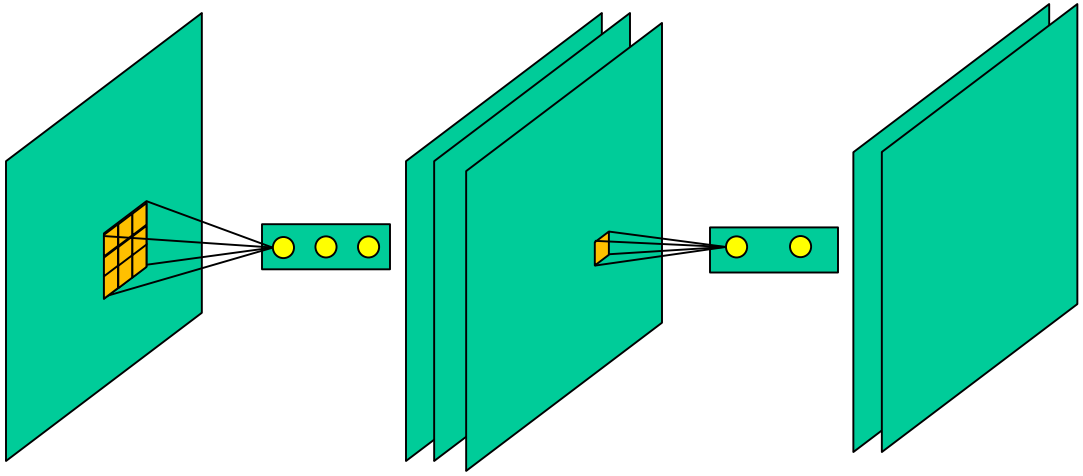


M. Lin, Q. Chen, S. Yan Network In Network. ICLR, 2014

Свёртка 1x1

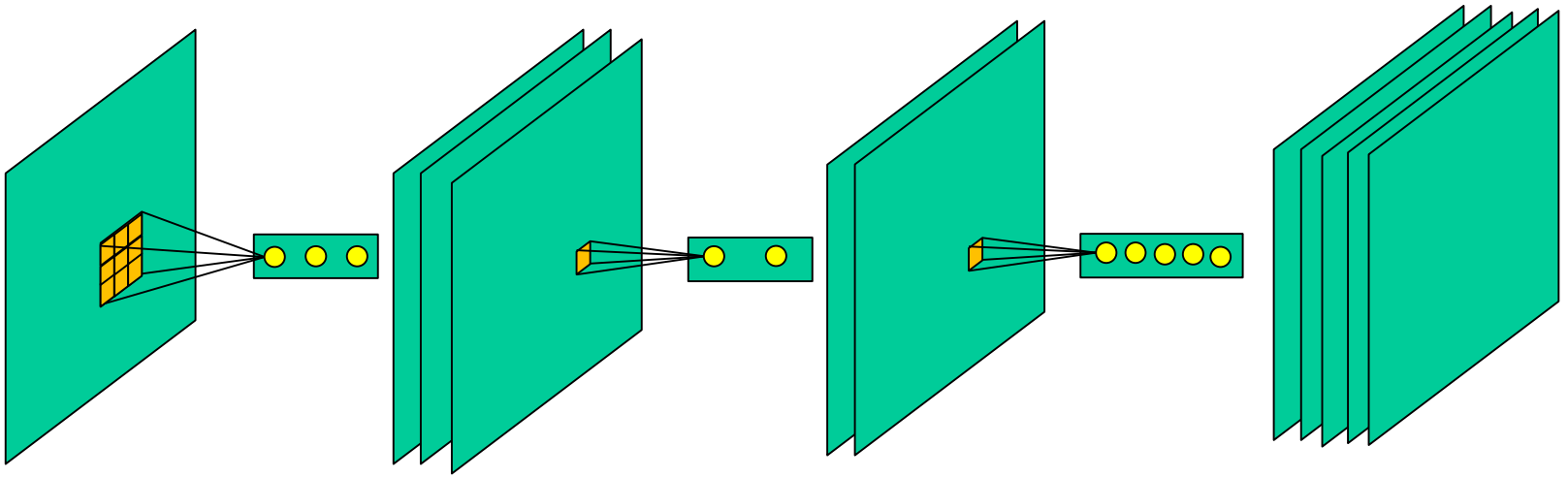


(b) Mlpconv layer



Мы можем реализовать второй и далее слои «вложенного» перспептрона как 1x1 свёртку с предыдущим слоем

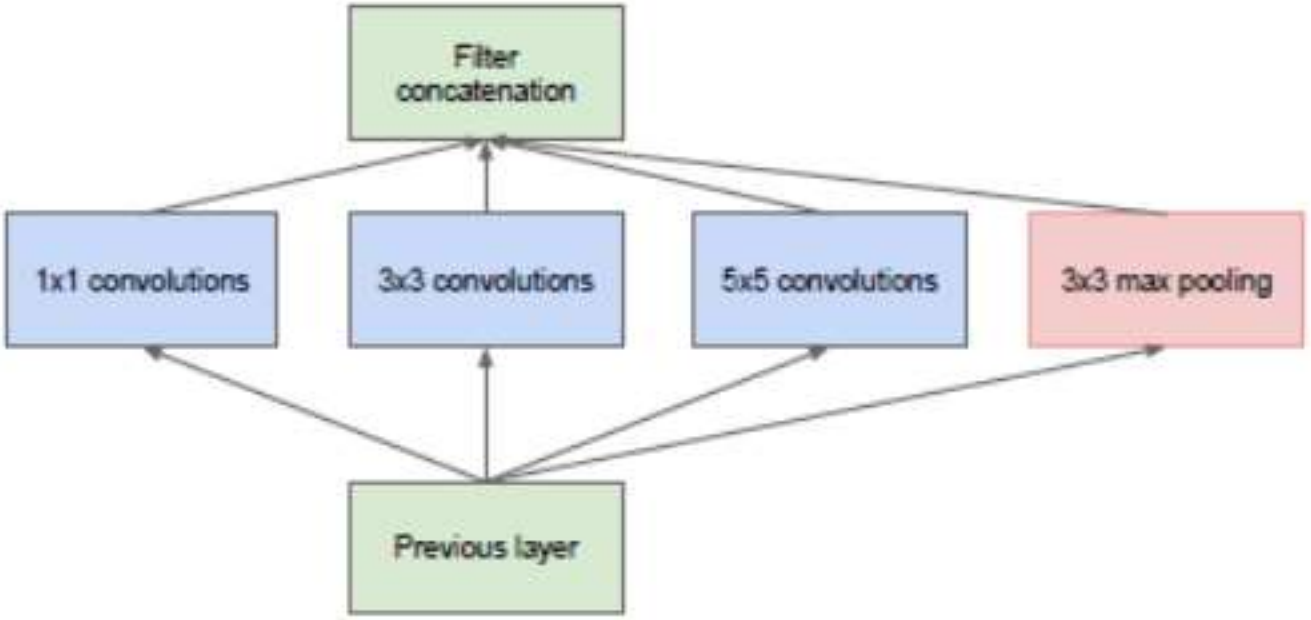
Свёртка 1x1



- Как трактовать свёртку 1x1?
  - Отображение вектора длины n (толщина входного слоя = число свёрток предыдущего слоя) на вектор длины k (число свёрток 1x1)
  - Или как набор «локальных классификаторов»
- Можем управлять «глубиной» тензора, регулируя k - число свёрток 1x1, по сравнению с n – глубиной предыдущего тензора
  - K < N, значит мы уменьшили до k глубину тензора (сжали)
  - K > N, значим мы увиличили глубину тензора

Архитектура Inception

Построим нейросеть из модулей более сложной структуры, чем просто набор свёрточных слоёв VGG

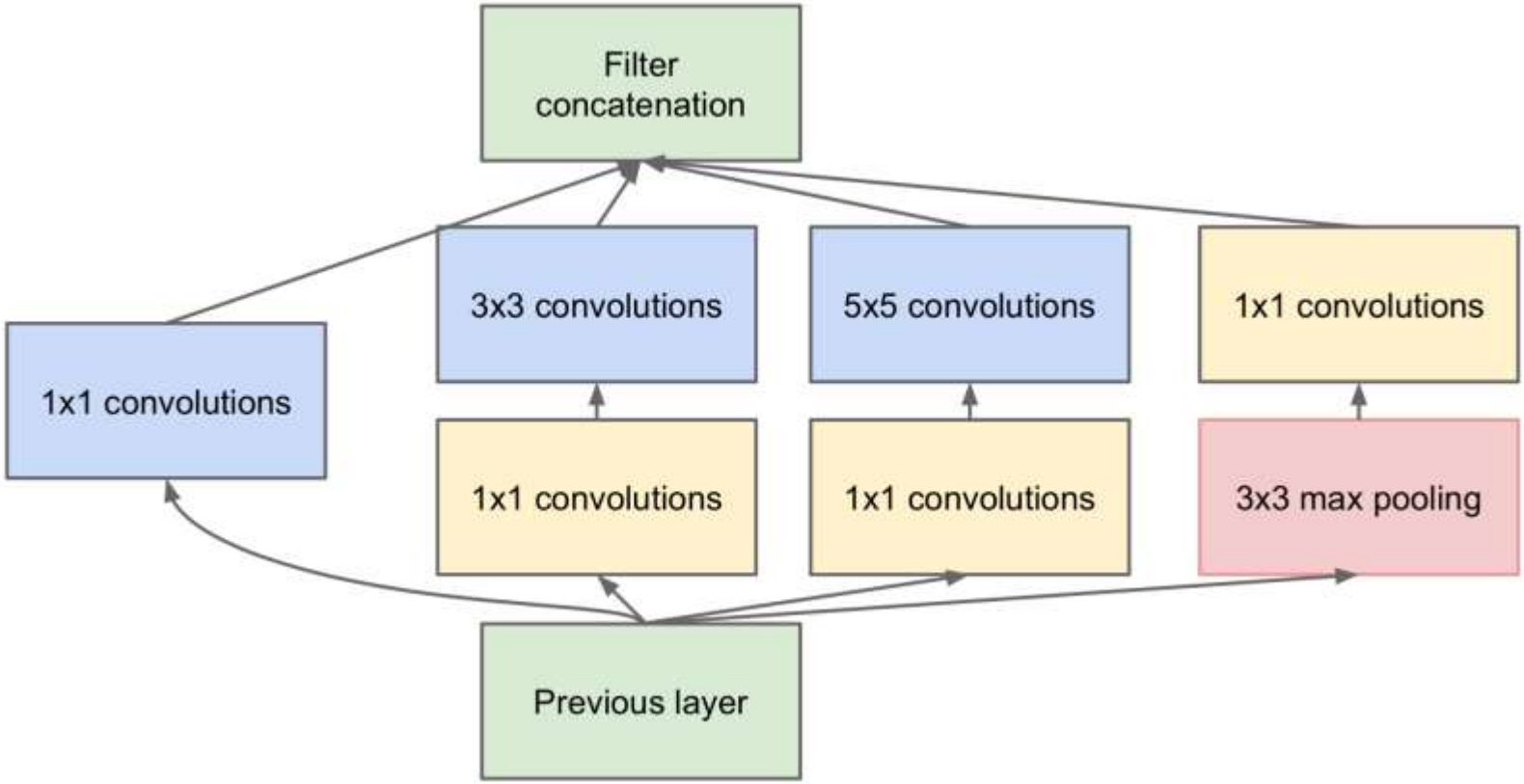


В чём смысл применять одновременно свёртки разных размеров?

Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

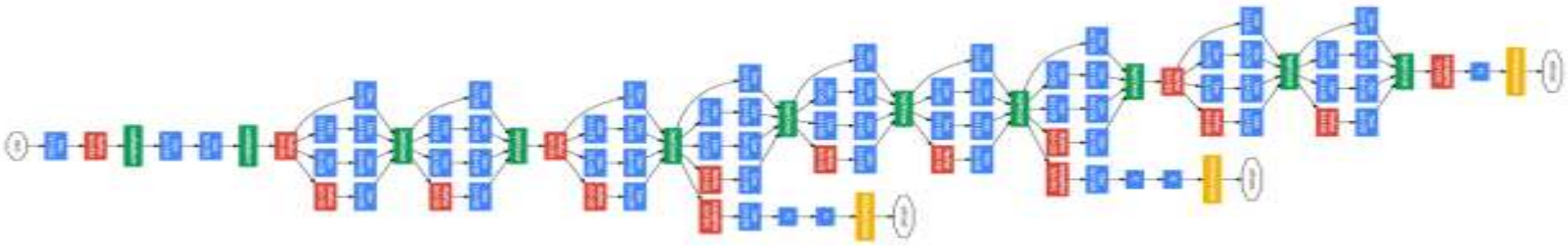
Модуль Inception

Добавим 1x1 свёртки для повышения производительности



Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

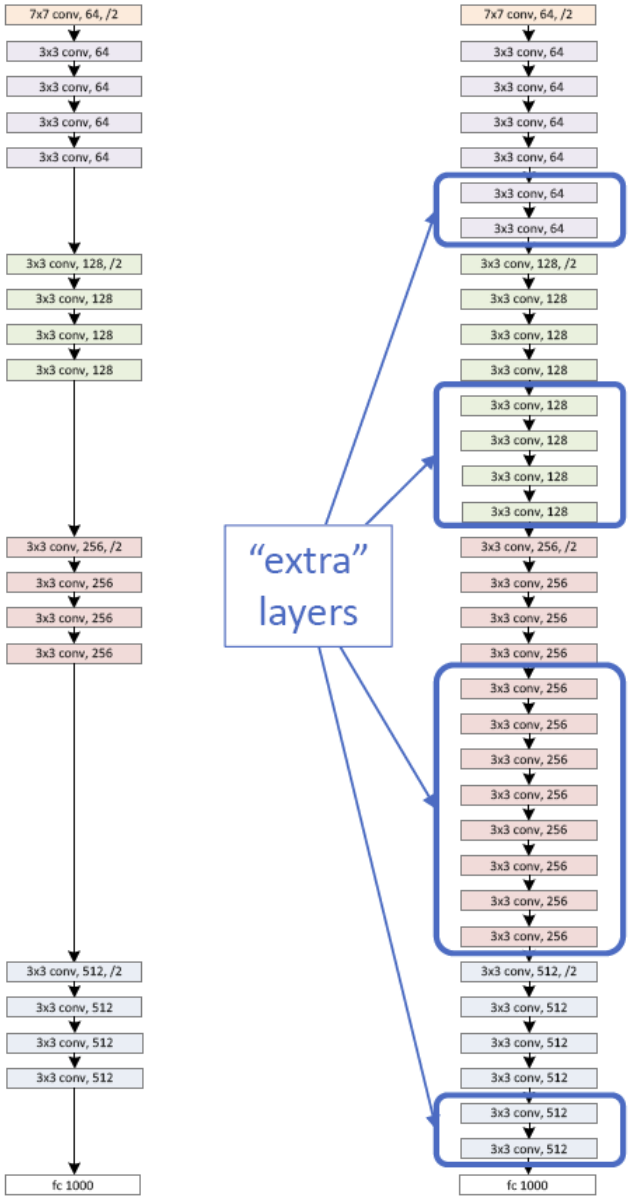
Архитектура Inception



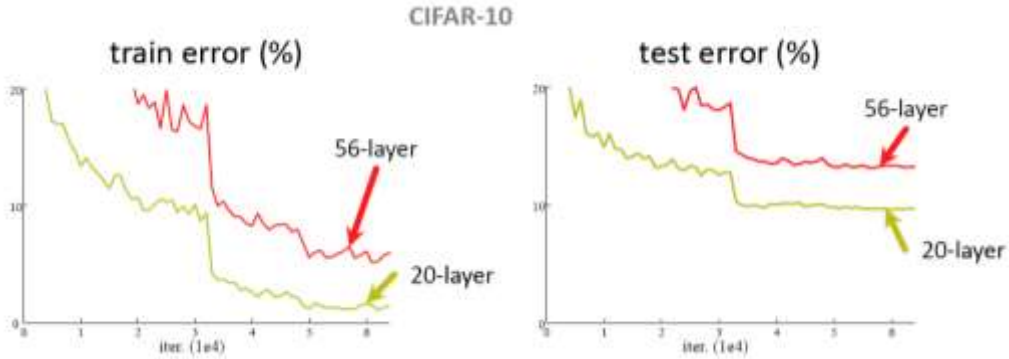
- Глубокая сеть
- Inception-модули
- Несколько уровней supervision

Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

Проблема глубины



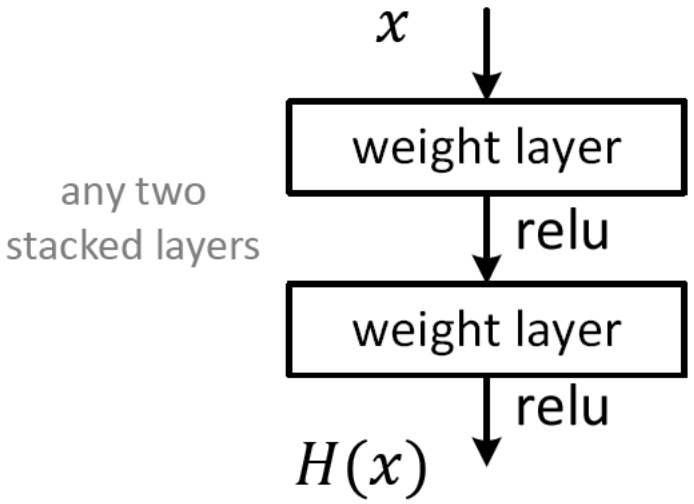
- Дальнейшее увеличение числа слоёв приводило к падению качества итоговой модели



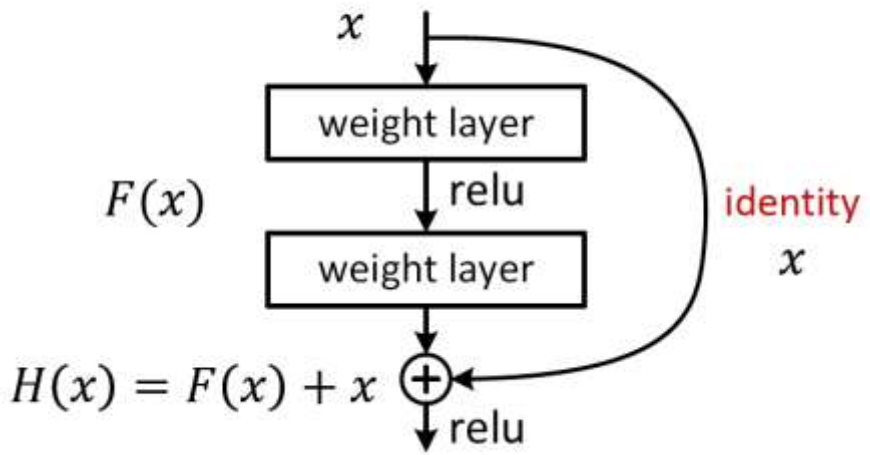
- При этом можно добавить «единичные» слои к сети, оставив ту же самую функцию и качество
- Значит, проблема в обучении сети

Residual net (или skip-connections)

- Plaint net

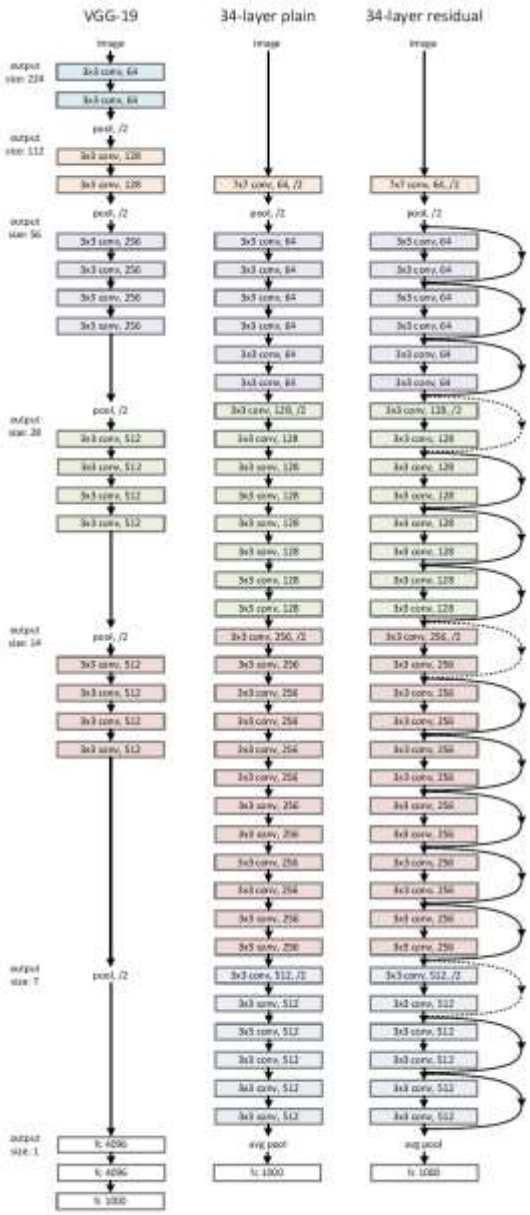


- Residual net



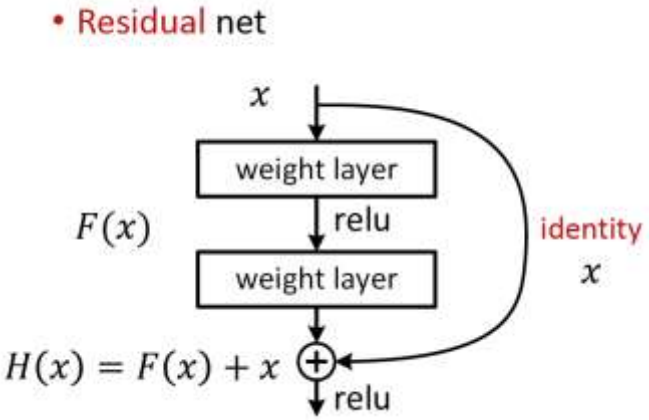
- Будем учить не преобразование, а добавку (пертурбацию) к тождественному преобразования
  - Если единичное преобразование оптимально, тогда мы его сохраняем
  - Небольшие флуктуации оказывается обучать проще

ResNet



Базовая модель

- Свёртки 3x3
- Subsampling через свёртку с шагом 2

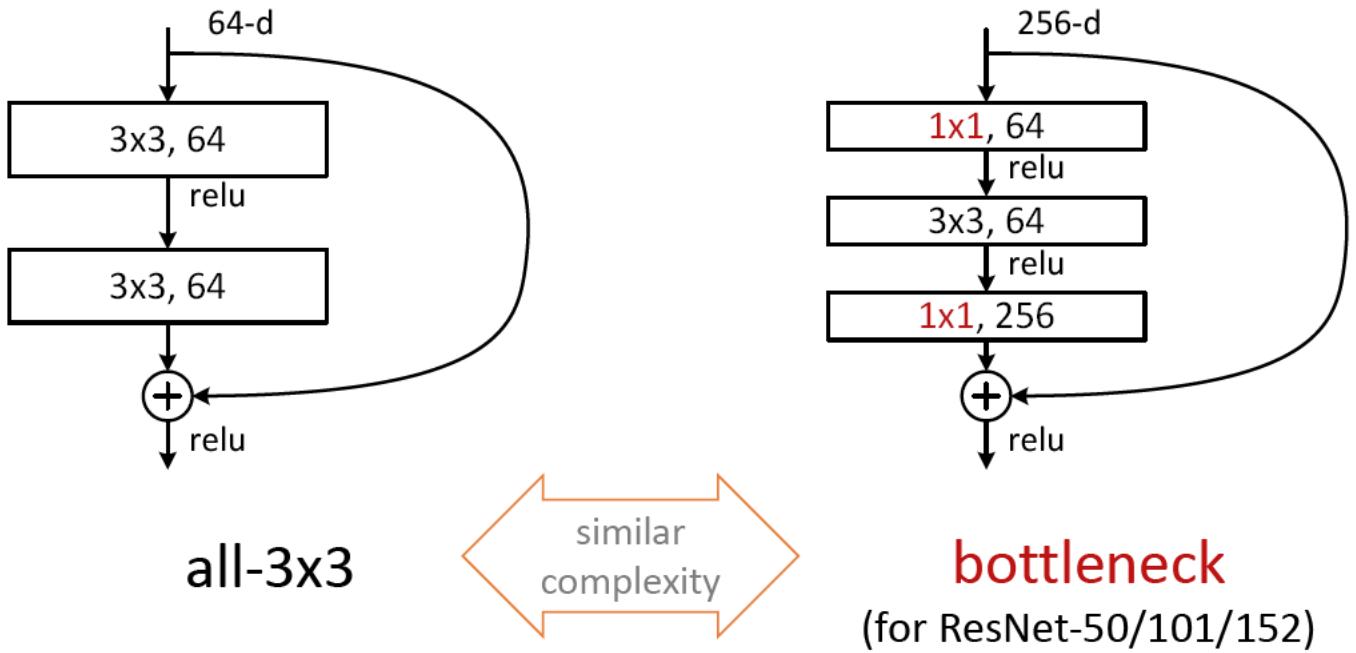


При изменении размеров тензоров пробуют варианты:

- Добавление нулями
- Линейная проекция
- Шаг 2

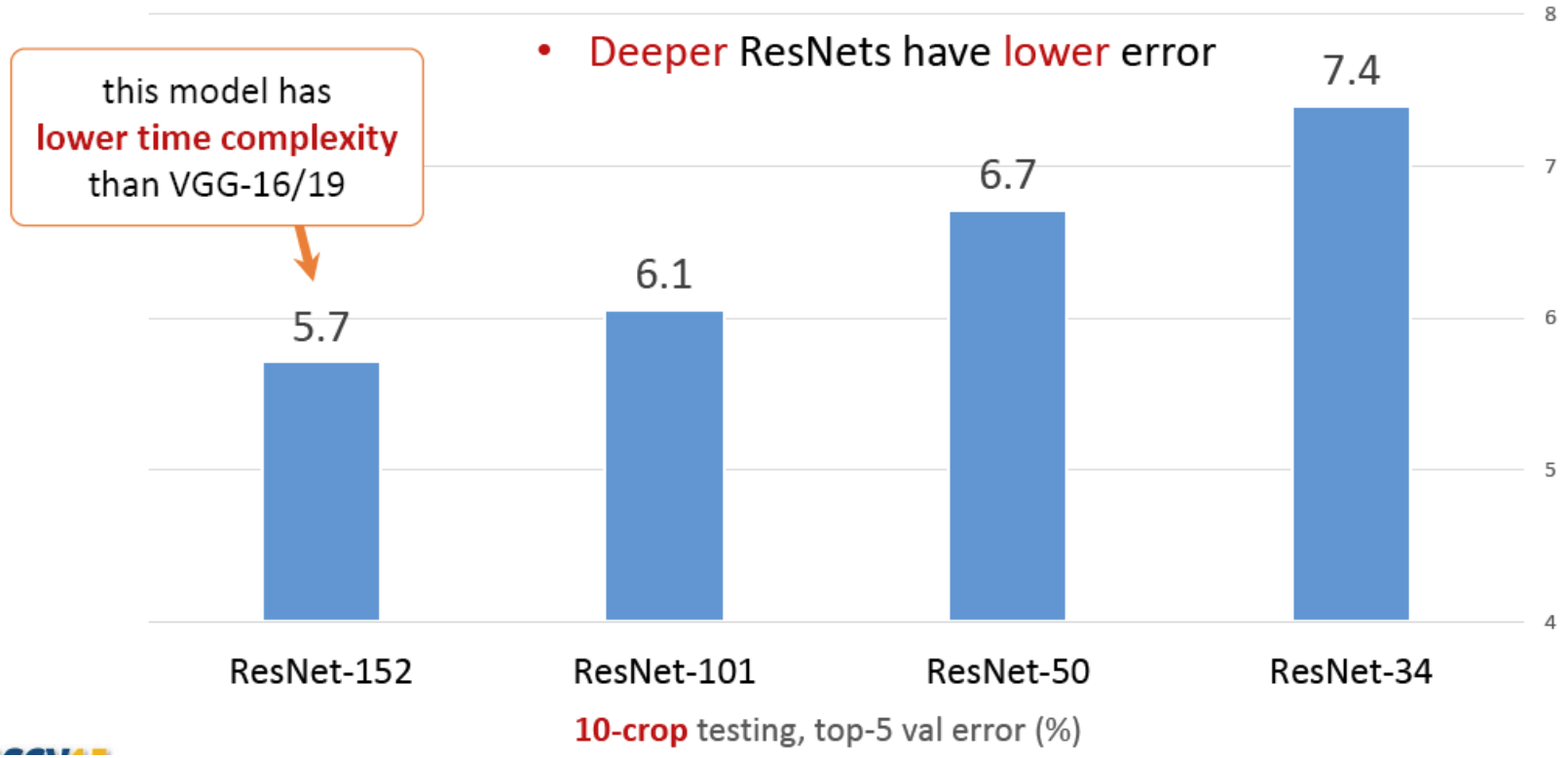
Блок для очень глубоких сетей

- A practical design of going deeper

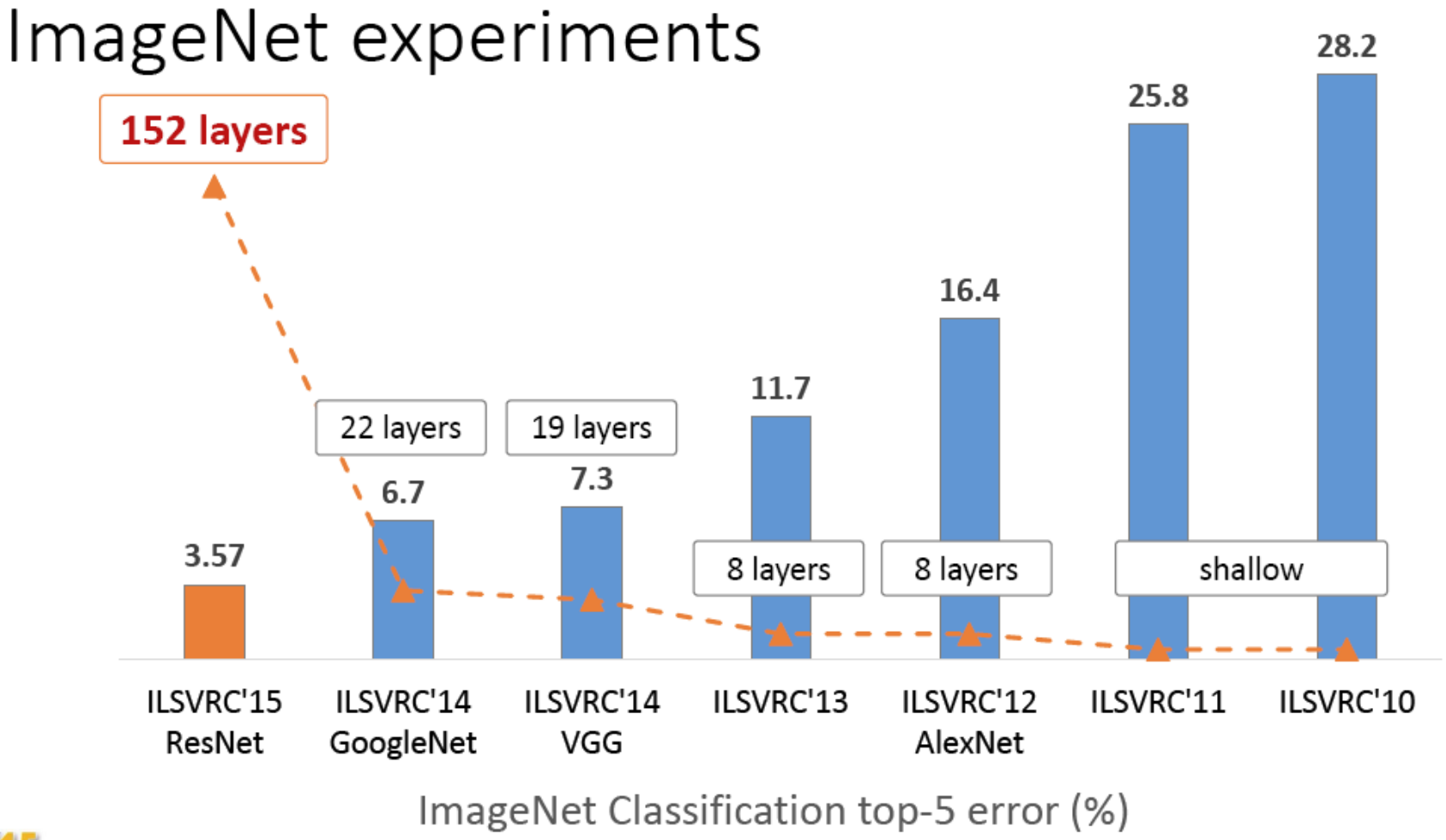


- Понижение размерности (256->64)
- Свёртка 3x3 на тензоре меньшей глубины
- Повышение размерности

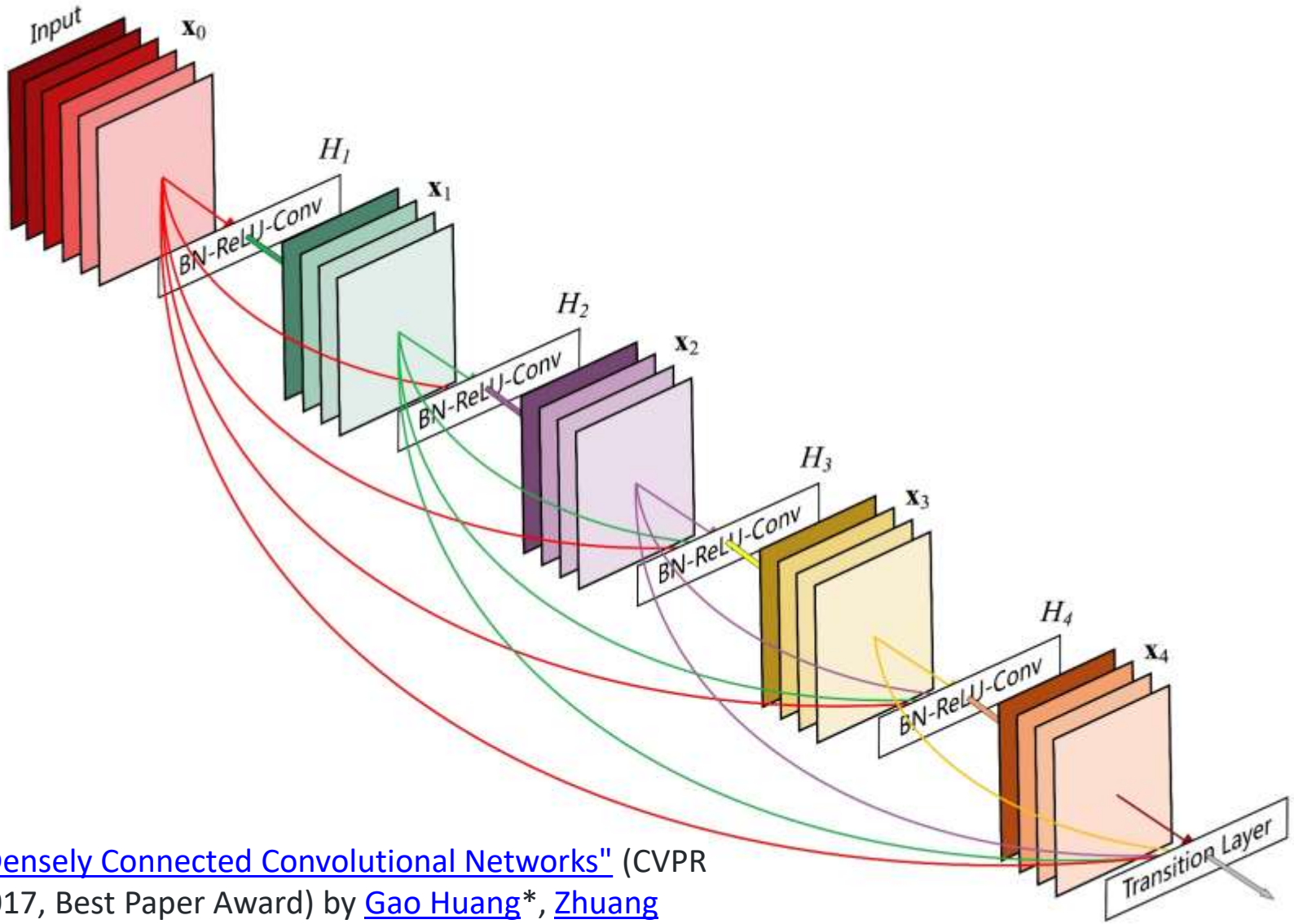
Результаты на ImageNet



Результаты на ImageNet

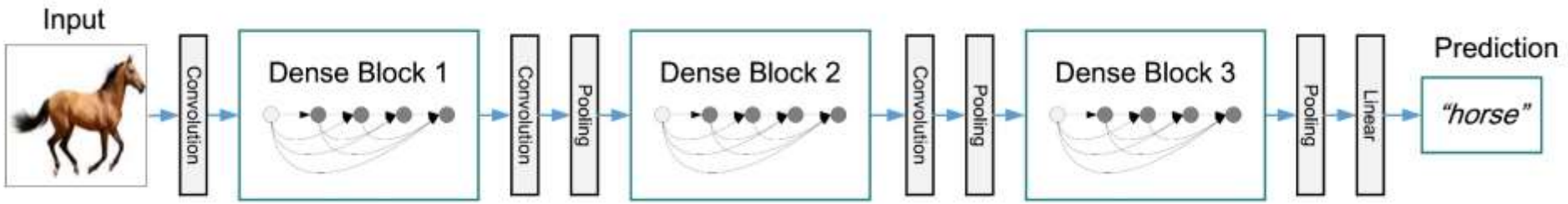


DenseNet



"Densely Connected Convolutional Networks" (CVPR 2017, Best Paper Award) by Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger

DenseNet

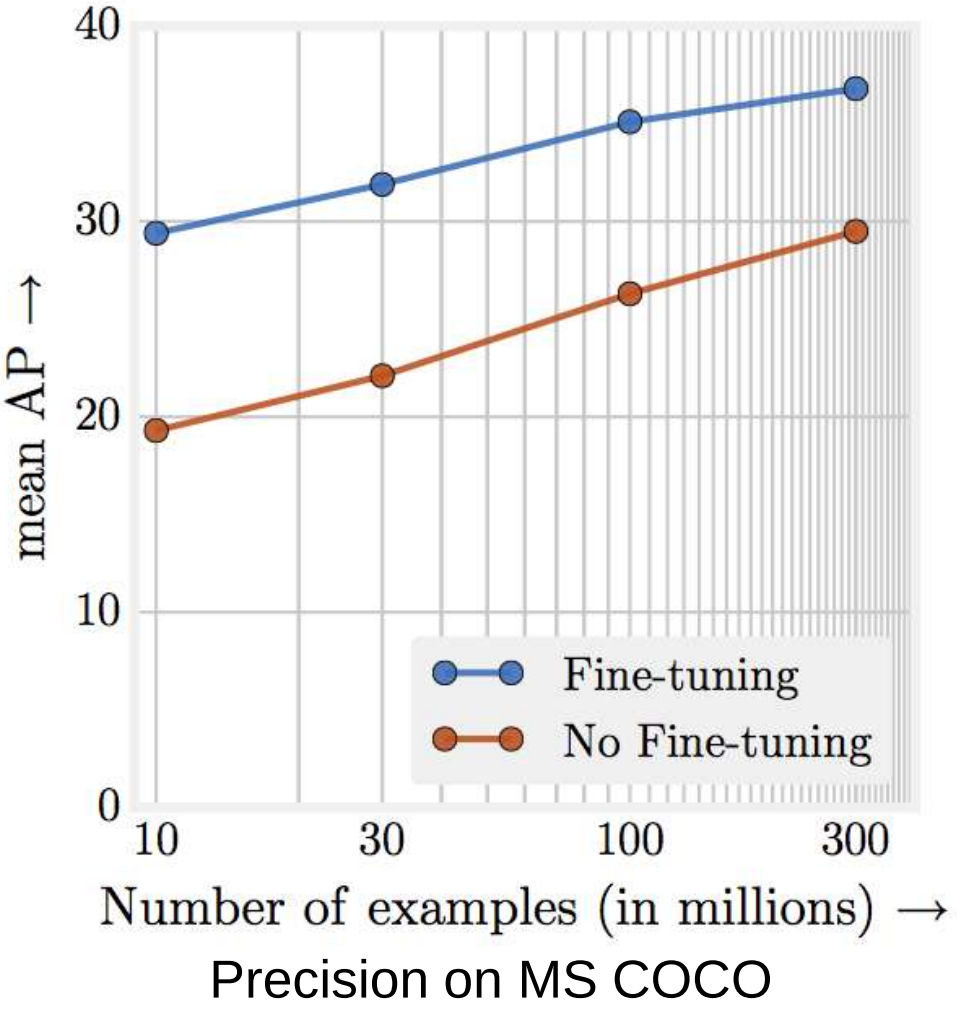


Network	Top-1 error	Torch Model
DenseNet-121 (k=32)	25.0	Download (64.5MB)
DenseNet-169 (k=32)	23.6	Download (114.4MB)
DenseNet-201 (k=32)	22.5	Download (161.8MB)
DenseNet-161 (k=48)	22.2	Download (230.8MB)

Model	Parameters	CIFAR-10	CIFAR-10+	CIFAR-100	CIFAR-100+
DenseNet (L=40, k=12)	1.0M	7.00	5.24	27.55	24.42
DenseNet (L=100, k=12)	7.0M	5.77	4.10	23.79	20.20
DenseNet (L=100, k=24)	27.2M	5.83	3.74	23.42	19.25
DenseNet-BC (L=100, k=12)	0.8M	5.92	4.51	24.15	22.27
DenseNet-BC (L=250, k=24)	15.3M	**5.19**	3.62	**19.64**	17.60
DenseNet-BC (L=190, k=40)	25.6M	-	**3.46**	-	**17.18**

https://github.com/liuzhuang13/DenseNet

Предобучение (JFT-300M by Google)

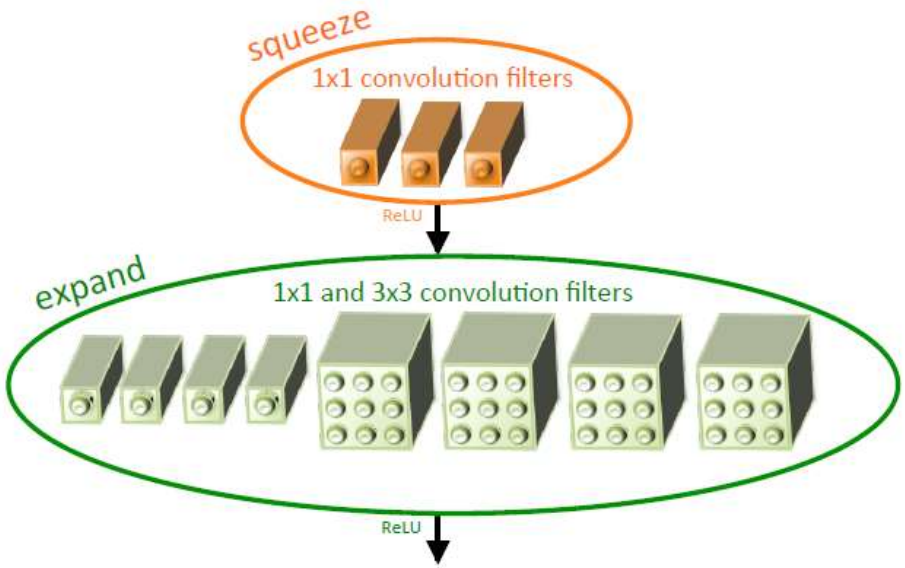


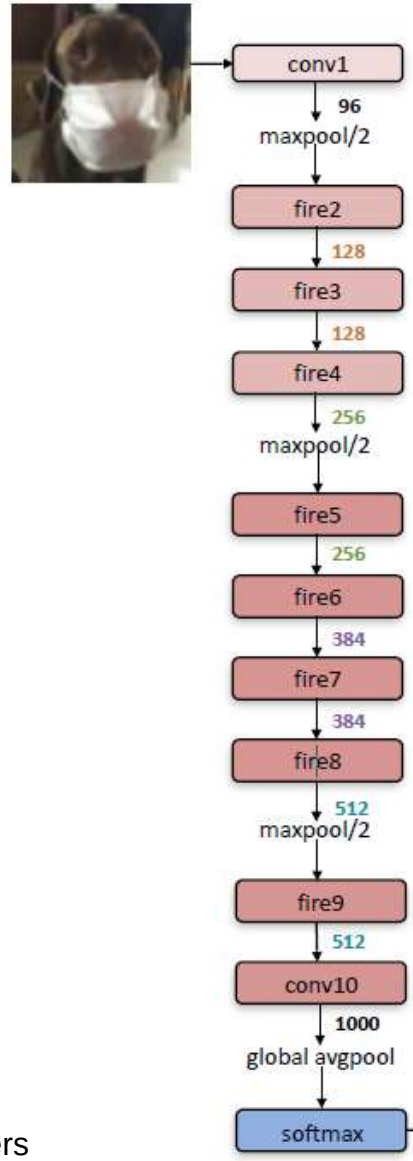
Precision on MS COCO

- 300M annotated images, 1B labels of 18291 categories
- 375M labels selected to maximize label precision, still ~20% errors
- ResNet-101 training on 50 x K80 for month
- Uses as pre-training for other tasks (object detection, semantic segmentation, pose estimation)

Chen Sun et. al. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. https://arxiv.org/abs/1707.02968

SqueezeNet

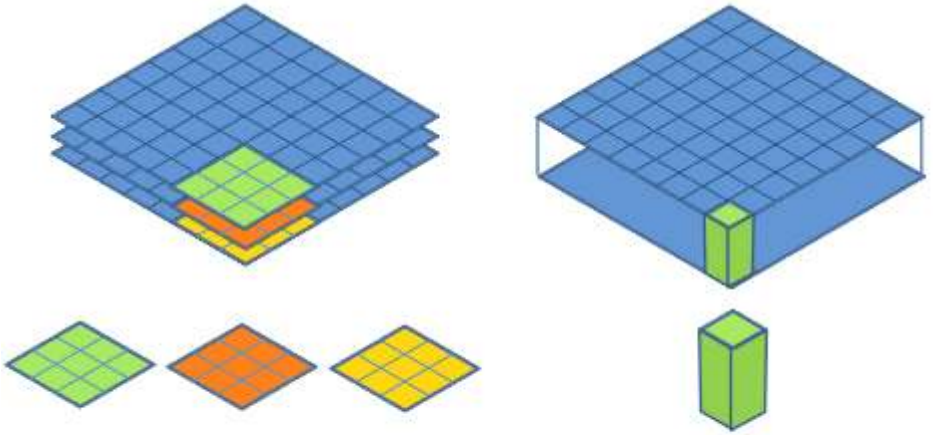




- Активно использовать 1x1 свёртки для уменьшения числа параметров
- «Сжимать» мы будем для того, чтобы на вход 3x3 фильтрам подавать меньше данных

Forrest N. Iandola et.al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. ArXiv 2016

Факторизация свёрток



Depthwise Convolutional Filters Pointwise Convolutional Filters

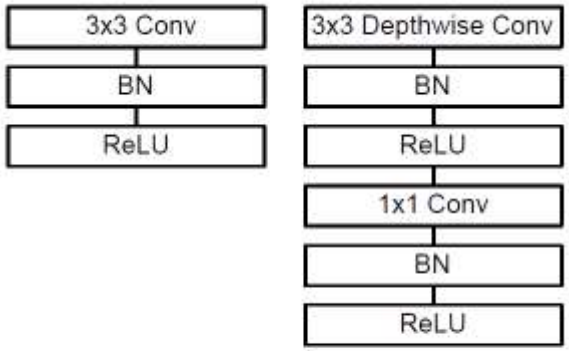


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

- Сложность обычного свёрточного слоя:

$$D_K * D_K * M * N * D_F * D_F$$

- Сложность комбинации depthwise convolution + pointwise convolution::

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

MobileNet

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 2. Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Оценка вариантов MobileNet

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

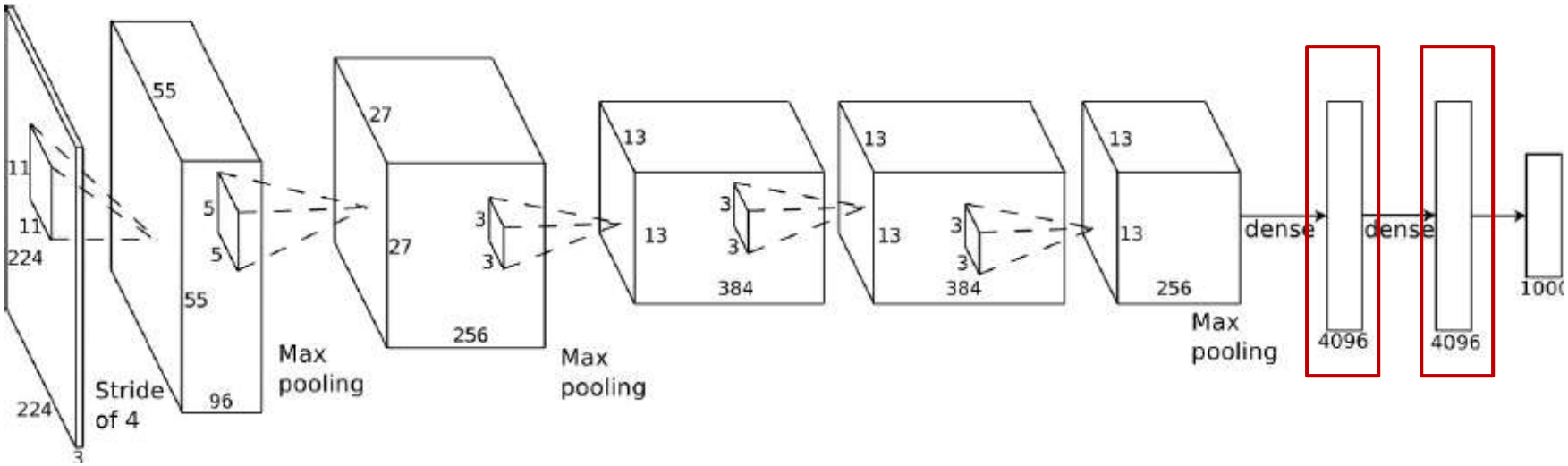
Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Резюме базовых архитектур

- Основные принципы построения архитектур
  - Строить глубокие модели из похожих блоков
  - Большие свёртки можно приближать последовательностью свёрток 3x3
  - Свёртки 1x1 позволяют управлять «толщиной» слоя и уменьшать вычислительную сложность
  - Можно обработать данные параллельно в нескольких ветвях и затем объединить
  - Residual-связи (или skip-connections) позволяют снизить проблему затухания градиента и обучать очень глубокие сети
- Предобучать лучше на как можно большей и разносторонней коллекции
- Придумывают «быстрые» архитектуры

Нейросетевые признаки для поиска похожих



- Выходы верхних слоёв можно использовать как вектор-признак изображения
- Попробуем искать похожие изображения по этим вектор-признакам
- Можно брать «готовую» сеть, но лучше дообучать её на целевой коллекции

Ali Sharif Razavian Hossein Azizpour Josephine Sullivan Stefan Carlsson
CNN Features off-the-shelf: an Astounding Baseline for Recognition. 2014

Адаптация к конкретной задаче

Leeds Casle

Kiev Pechersk Lavra

- Соберём коллекцию в полуавтоматическом режиме путем запросов к поисковой системе
- Нейросеть дообучалась на коллекции изображений достопримечательностей, чтобы формируемые дескрипторы «настроились» на семантику конкретной поисковой задачи

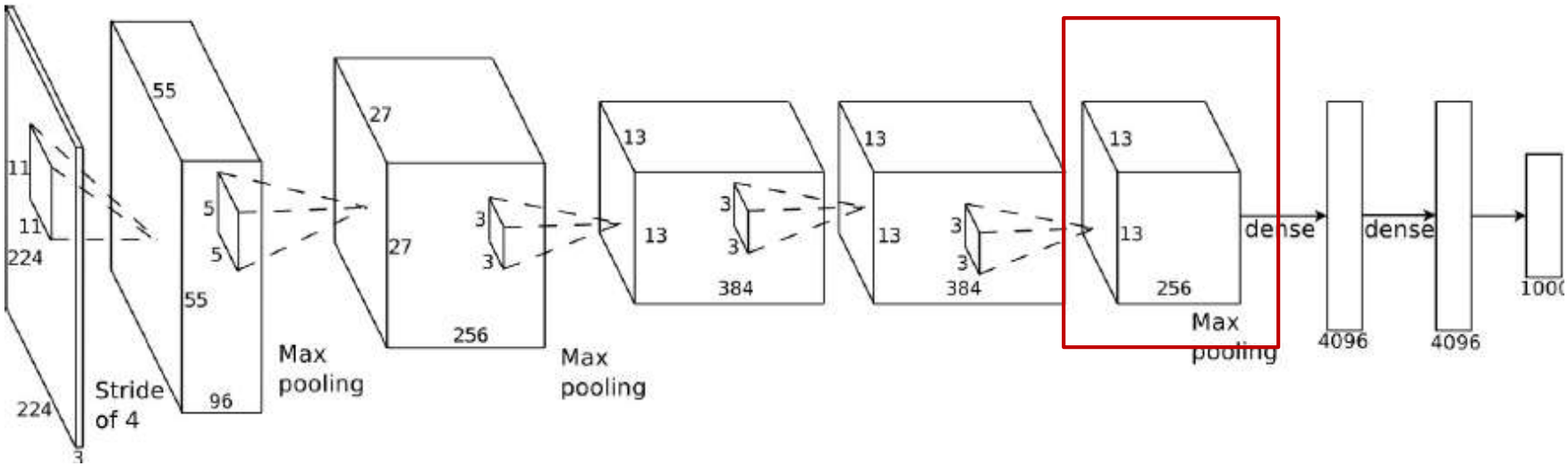
До и после адаптации

Вклад адаптации и сжатие

Дескриптор	Oxford	Oxford105K	Holidays
Fisher+color	-	-	0.723
VLAD+adapt+innorm	0.448	0.374	0.625
Sparse-coded features	-	-	0.727
Triangulation embedding	0.433	0.353	0.617
Нейродескрипторы			
Слой 6	0.433	0.386	0.727
Нейродескрипторы с адаптацией			
Слой 6	**0.557**	**0.523**	**0.769**

- Дескрипторы получаются довольно большими (например, 4096 x float), поэтому их нужно сжимать
- Например, можно применить PCA до 128 параметров

Использование свёрточных признаков



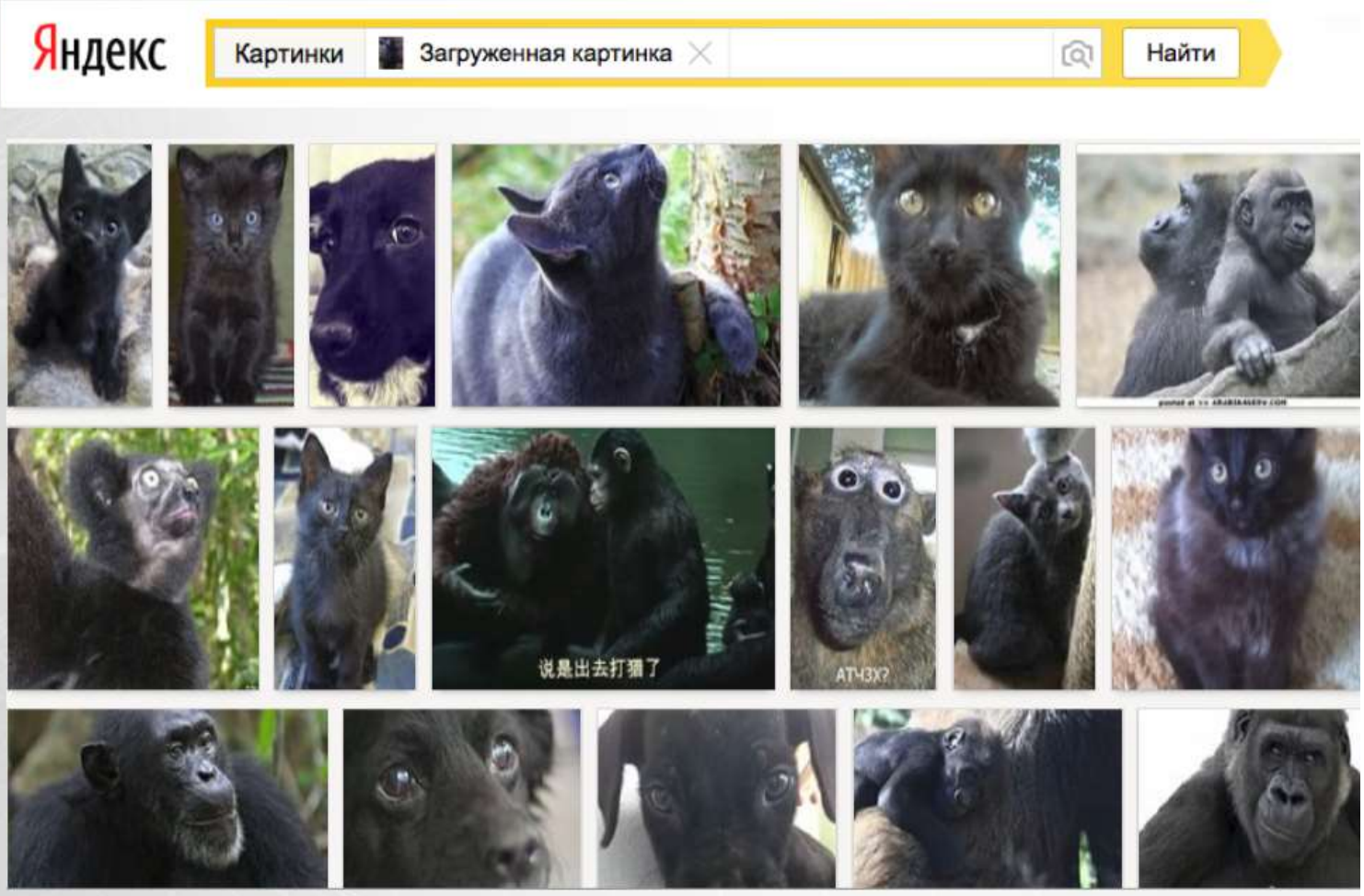
- Дескрипторы могут быть вычислены только для квадратных изображений небольшого фиксированного размера
- При уменьшении изображений большого разрешения теряются маленькие детали
- Решение: использовать выходы сверточных слоев, они могут быть вычислены для любых изображений
- Делать average по выходу каждого свёрточного слоя

A Babenko, V Lempitsky Aggregating local deep features for image retrieval CVPR 2015

Полносвязанные vs свёрточные

Результаты с полносвязным дескриптором:

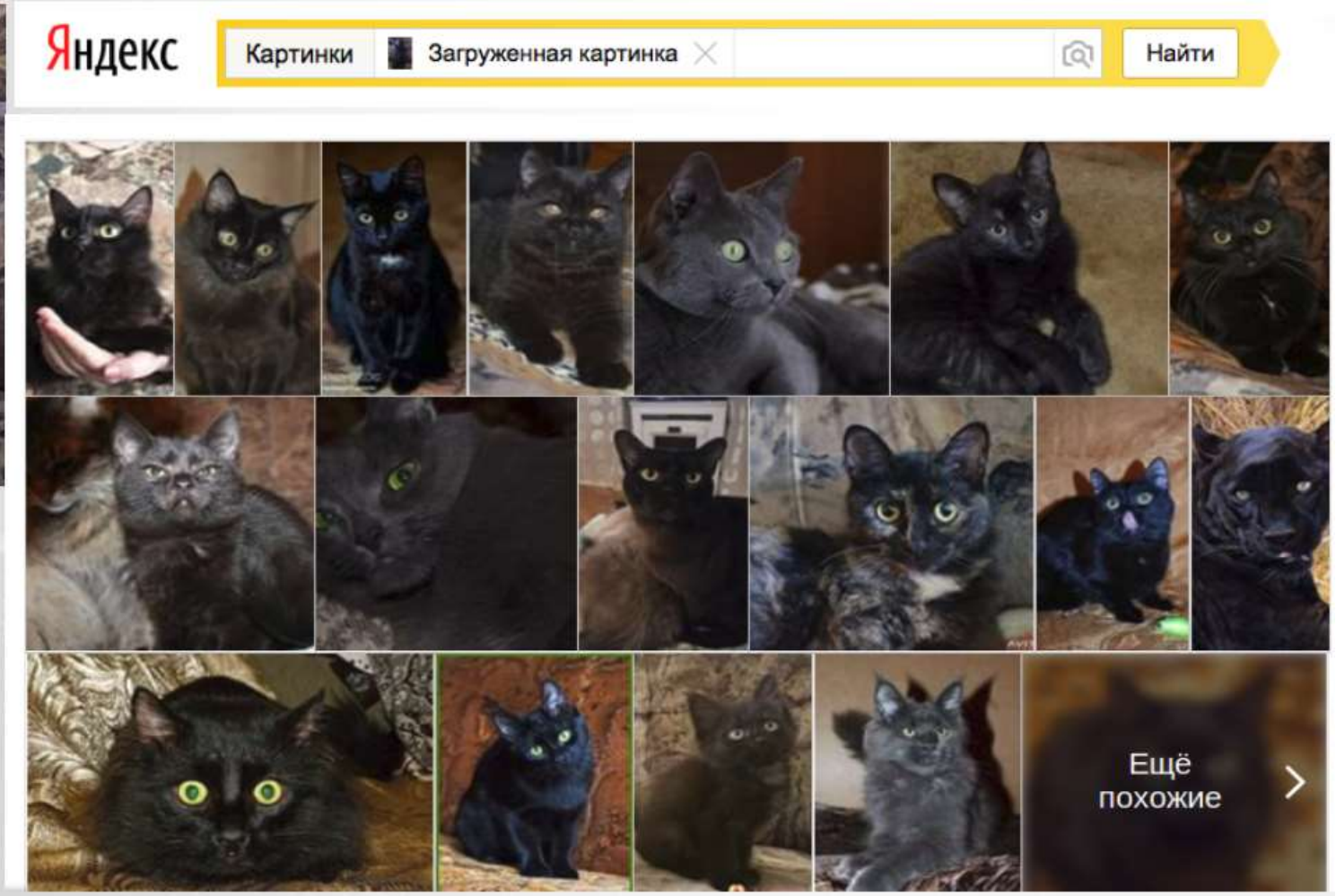
Запрос



Полносвязанные vs свёрточные

Результаты со сверточным дескриптором:

Запрос

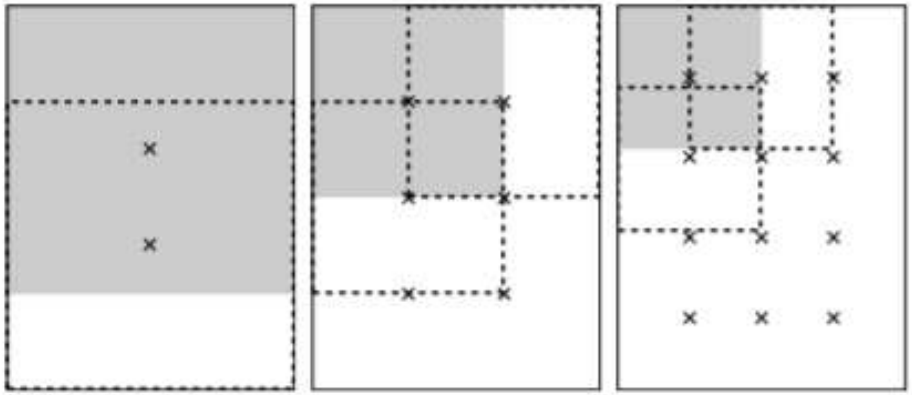


Свёрточные признаки

Дескриптор	Oxford	Oxford105K	Holidays
Fisher+color	-	-	0.723
VLAD+adapt+innorm	0.448	0.374	0.625
Sparse-coded features	-	-	0.727
Triangulation embedding	0.433	0.353	0.617
Нейродескрипторы			
Слой 6	0.433	0.386	0.727
Нейродескрипторы с адаптацией			
Слой 6	0.557	0.523	0.769
Сверточный слой	**0.657**	**0.642**	**0.802**

Тоже применяется PCA до 128 параметров

Пример другого признака

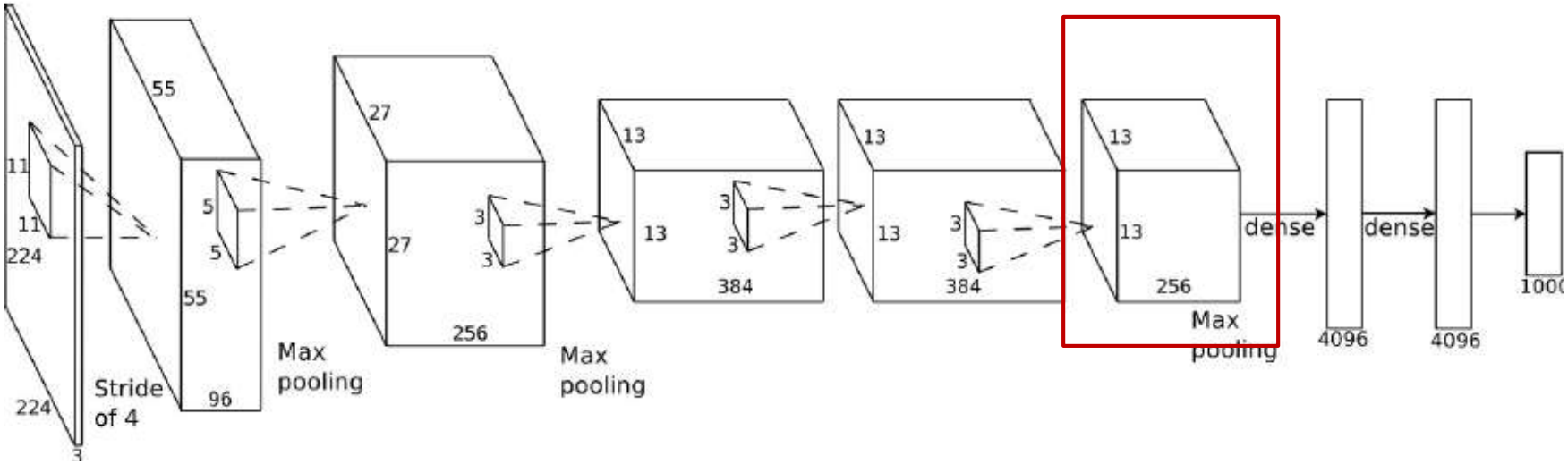


Regional maximum activation of convolutions

- Задаём сетку регионов на изображении
- В каждом считаем «Maximum activations of convolutions» (MAC) по свёрточному слою
- Преобразуем каждый признак – нормализация, PCA-whitening, нормализация
- Суммируем вектора по всем регионам
- Сравниваем вектора косинусной метрикой

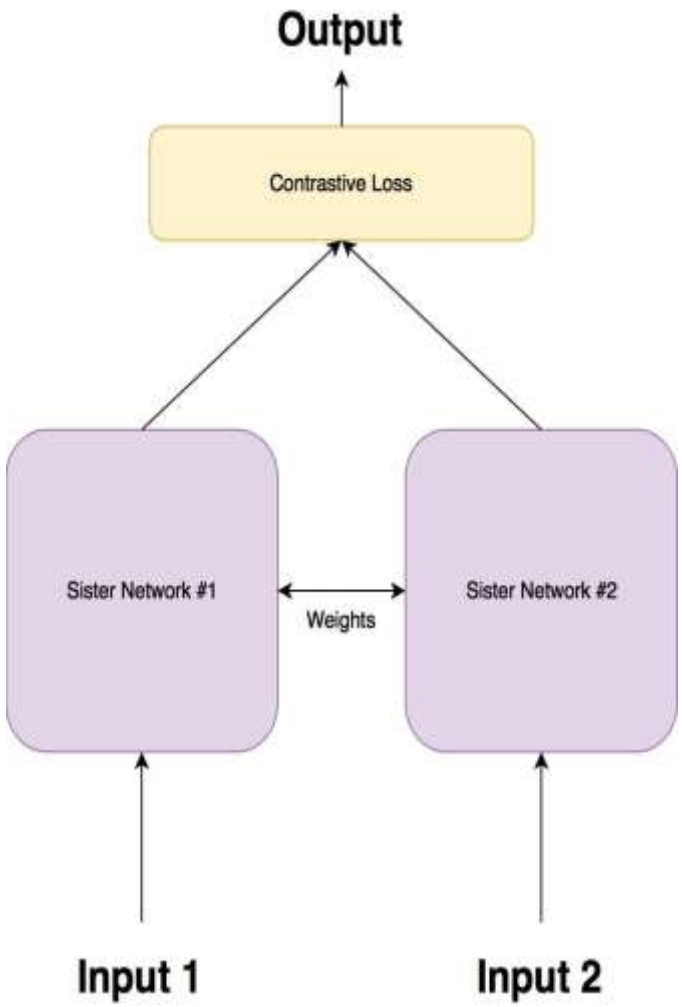
Tolias, G., Sicre, R., & Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*.

Что плохо в самом подходе?



- Мы строим отображение в вектор-признак косвенно, решая другую задачу – классификации
- Мы можем переформулировать задачу, строя специально хорошие признаки

Обучение представлений



Классический contrastive loss с использованием евклидовой метрики:

$$L(\theta, (X_1, X_2, Y)) = (1 - Y)L_G(||X_1 - X_2||_2) + YL_I(||X_1 - X_2||_2)$$

L_G - функция потерь для положительной пары, L_I - для отрицательной.
Как правило, $L_G(d) = d^2$, $L_I(d) = \max(0, m - d)^2_+$

[Chopra, et.al., 2005]

В качестве меры близости можно использовать и косинусную меру.

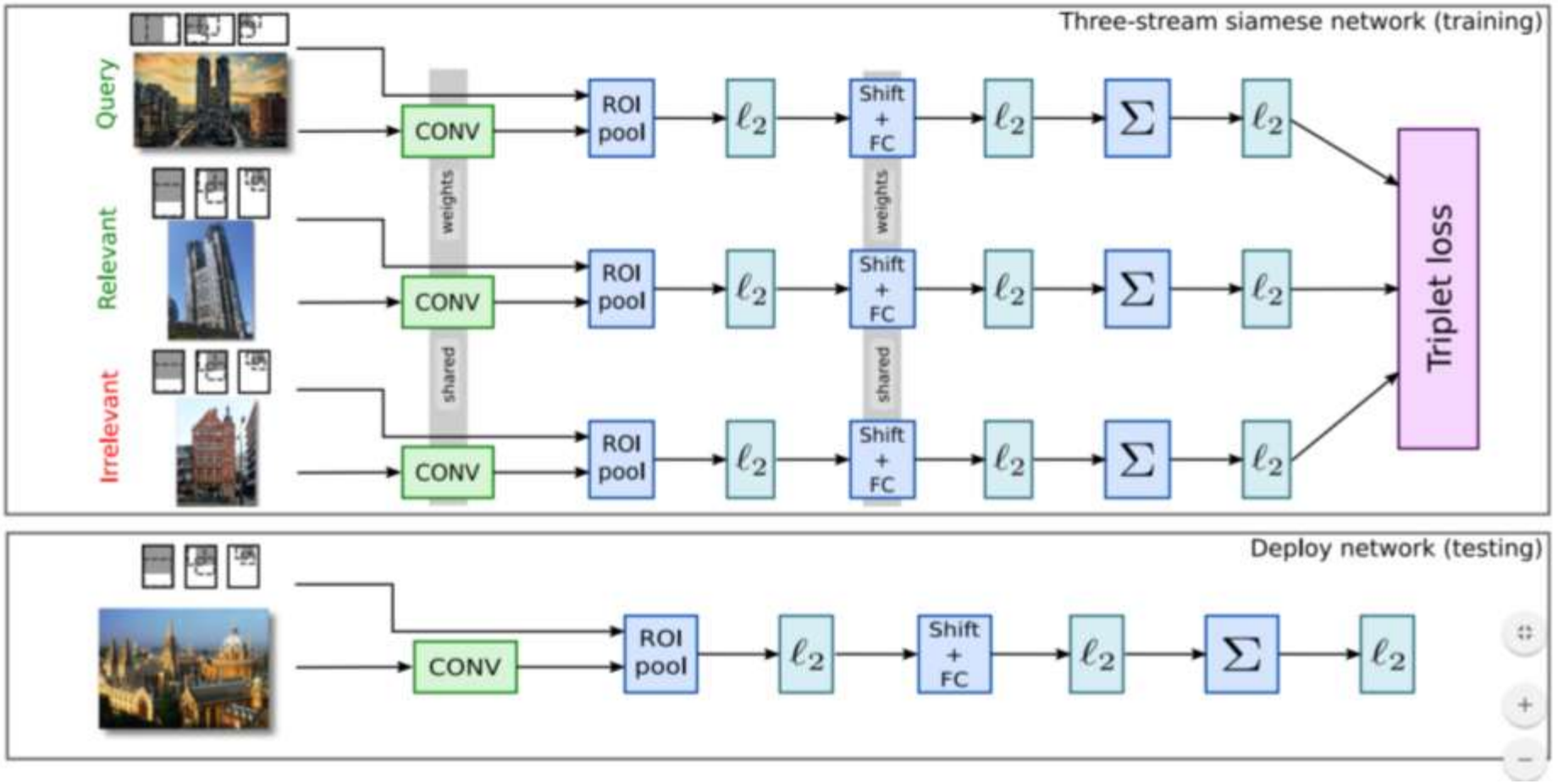
$$L(\theta, X_1, X_2, Y) = \frac{1}{2}(Y - \sigma(wd + b)),$$

где близость $d = \frac{X_1 X_2}{||X_1||_2 ||X_2||_2}$

[Sun, et.al., 2014]

Radenović F., Tolias G., Chum O. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. ECCV 2016

Triplet loss



Триплет состоит из запроса q (query), релевантного изображения и нерелевантного изображения

Triplet-loss

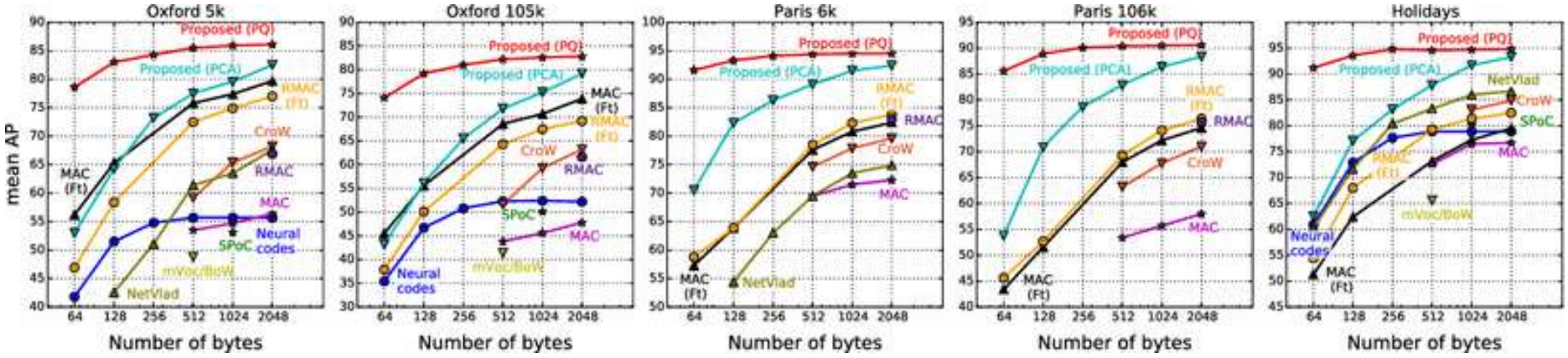
We use the following ranking loss. Let I_q be a query image with R-MAC descriptor q, I^+ be a relevant image with descriptor d^+, and I^- be an irrelevant image with descriptor d^-. We define the ranking triplet loss as

$$L(I_q, I^+, I^-) = \frac{1}{2}\max(0, m + \|q - d^+\|^2 - \|q - d^-\|^2), \quad (1)$$

where m is a scalar that controls the margin. Given a triplet that produces a non-zero loss, the sub-gradients are given by:

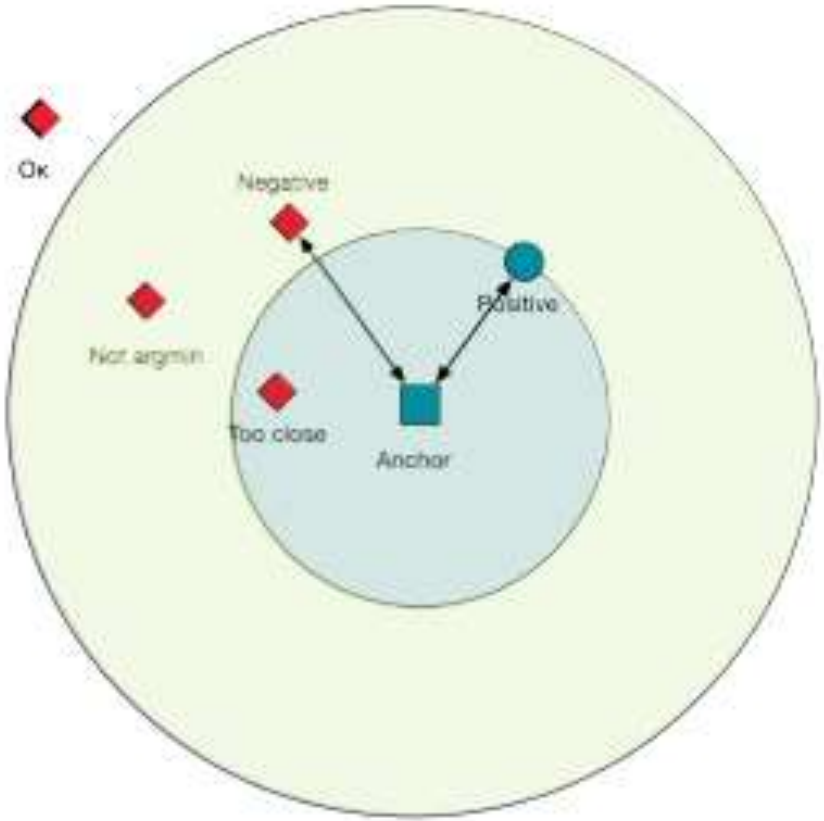
$$\frac{\partial L}{\partial q} = d^- - d^+, \quad \frac{\partial L}{\partial d^+} = d^+ - q, \quad \frac{\partial L}{\partial d^-} = q - d^-. \quad (2)$$

Сжатие



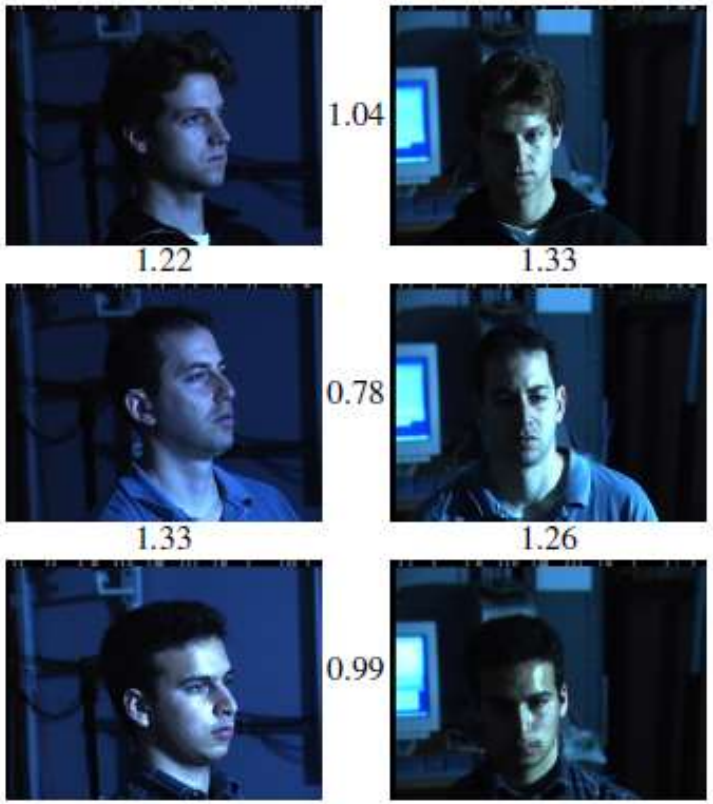
Сжатие обученных признаков с помощью PCA и PQ(product quantization)

Выбор триплетов

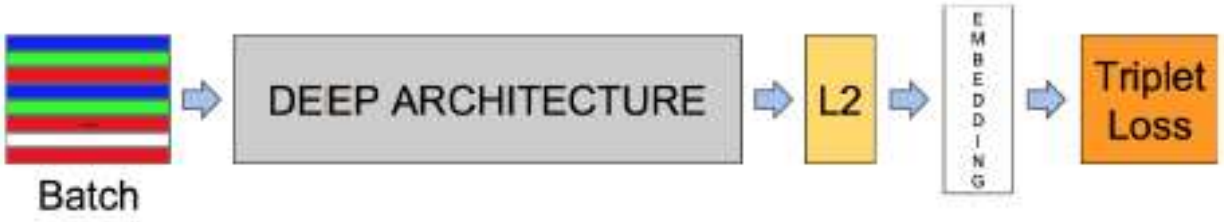


Пространство, полученное отображением $f(.)$.

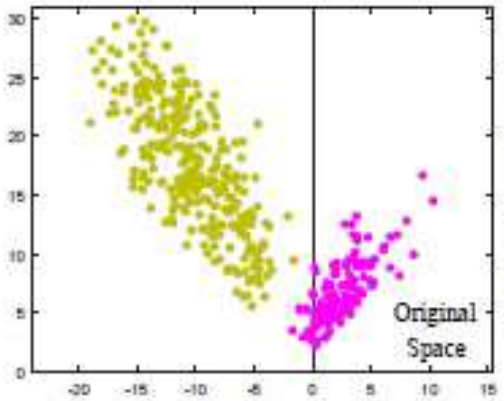
FaceNet – пример подобного подхода



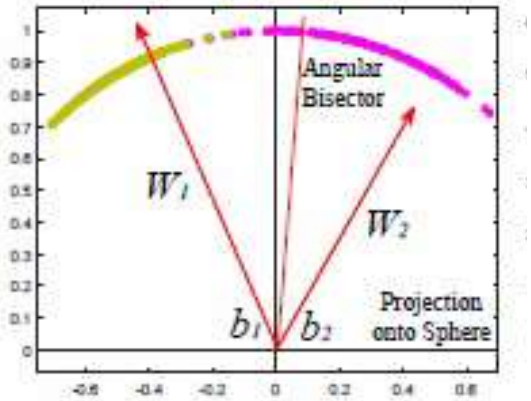
- Обучение представлений
- Функция y=f(I) отображает изображение I в 128-мерное пространство, в котором L2-метрика соответствует близости людей
- f(x) – глубокая нейросеть
- Обучается triplet loss



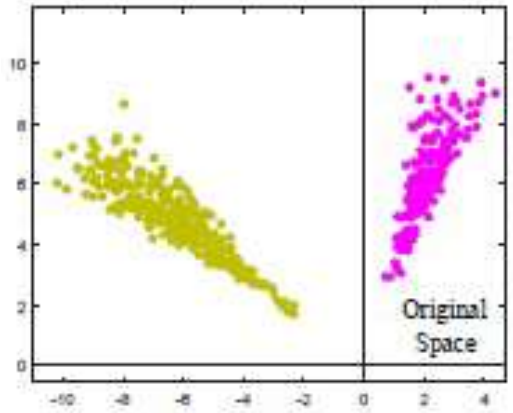
A-softmax (SphereFace)



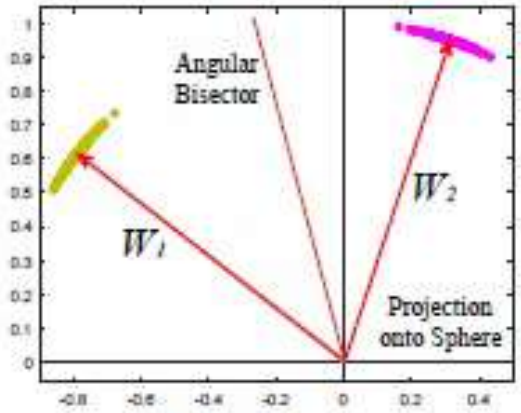
(a) Original Softmax Loss



(b) Original Softmax Loss



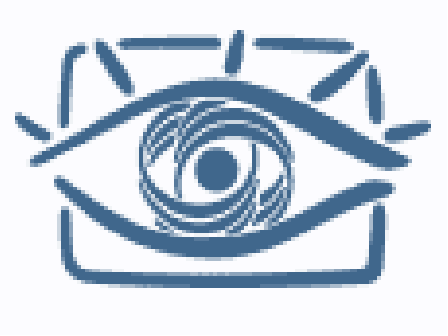
(e) A-Softmax Loss



(f) A-Softmax Loss

Резюме поиска

- Для задачи CBIR сейчас используются нейросетевые признаки
- Для поиска похожих нужно дообучение на близкой по смыслу коллекции
- Стоит явно учить «представление», чтобы у похожих объектов были похожие вектор-признаки
  - Contrastive loss
  - Triplet loss и производные варианты



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №7
«Детекторы объектов»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

28 октября 2019 года

Выделение объектов

Необходимо определить, есть ли на изображении объекты заданного типа и если да, то определить их положение

Формализуем задачу

- Пусть задан набор из s интересующих нас классов объектов, пронумерованных от 1 до S
- Вход:
  - Цветное изображение $I \in R^{N \times M \times C}$
- Выход:
  - Набор найденных объектов $\mathrm{Obj} = \{Obj_i | i = 1, p\}$, где Obj_i - структура, описывающая локализацию объекта и его класс $c_i \in [1, S]$
  - Какие варианты для Obj_i?
    - $Obj_i = (x_i, y_i, w_i, h_i, c_i)$. x_i, y_i, w_i, h_i - координаты левого верхнего угла, ширина и высота *i*-го ограничивающего прямоугольника, содержащего объекта класса
    - $Obj_i = (mask_i, c_i)$. $mask_i$ - попиксельная маска объекта

Алгоритмы выделения объектов на изображениях – ***детекторы*** объектов (object detection)

Объекты и материалы

Объекты (Things)

- Имеют определенные размер и формы

Найти «человека»

Материалы (Stuff)

- Однородный или повторяющийся шаблон мелких деталей
- Нет определенного размера и формы

Найти «дорогу»

Датасеты - один важный класс

- Face
- Head
- Pedestrian
- People
- Traffic signs
- Cars

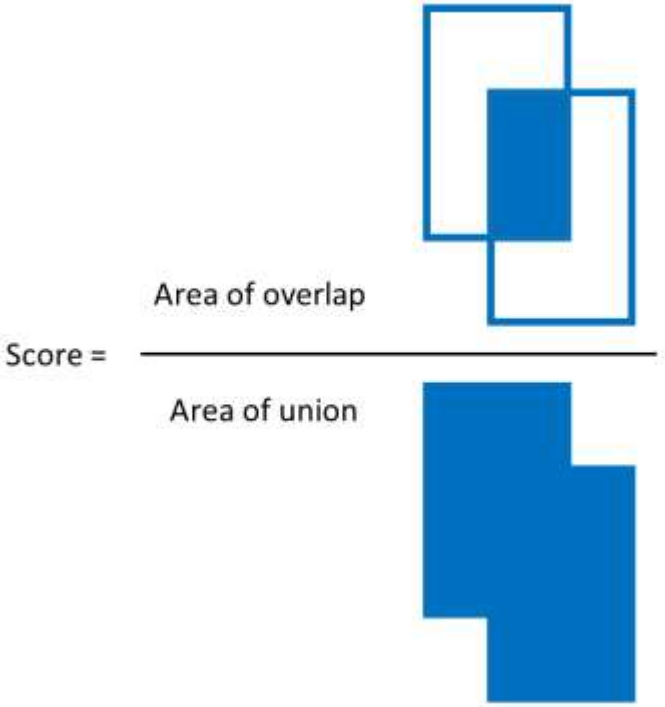
Множество классов - Microsoft COCO

Common Objects in Context

- 200К изображений (train, val, test)
- 80 классов
- 500К выделенных объектов (с масками)

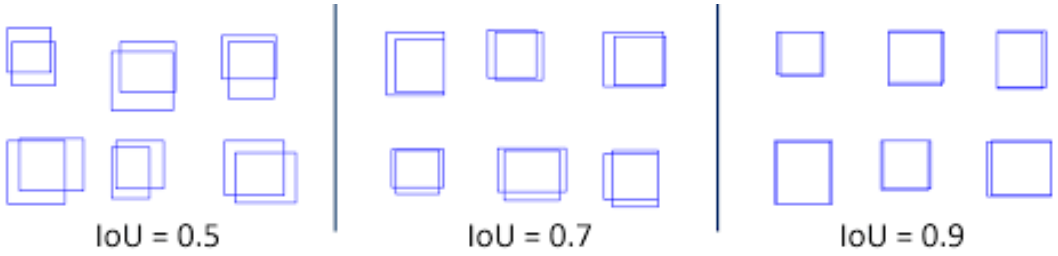
Сейчас в ряде конкурсов отказались от «детекции объектов» через bbox, и рассматривают только маски («instance segmentation»)

Критерий обнаружения (IoU)



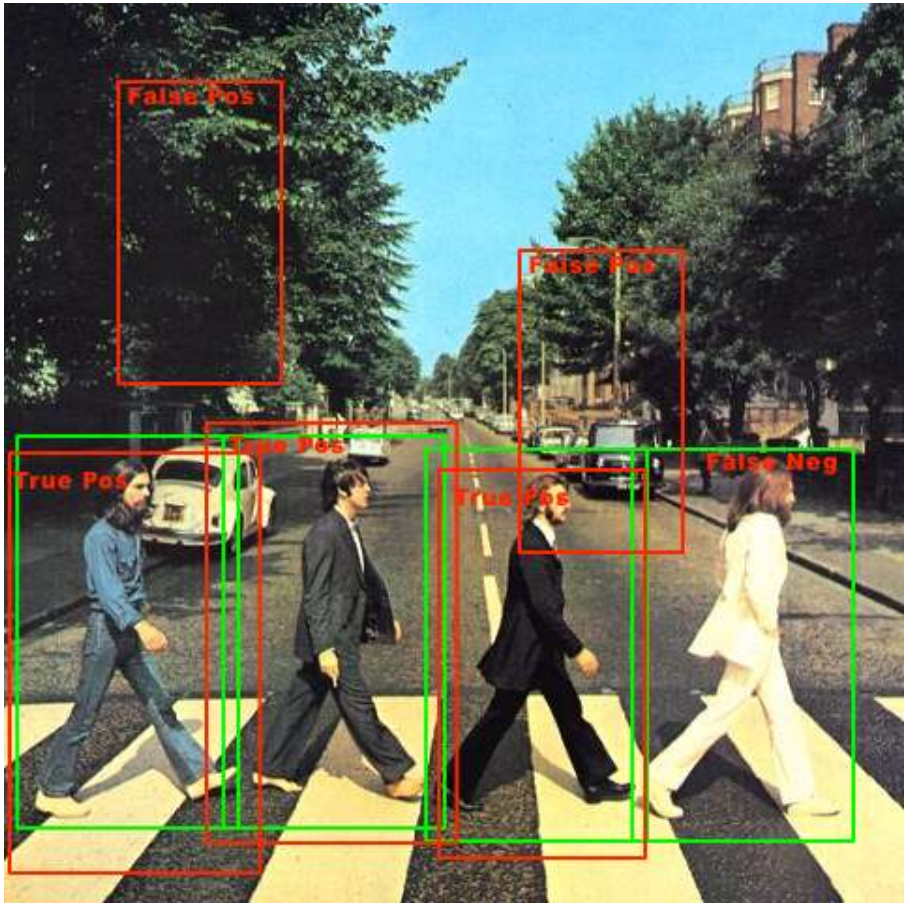
IoU = Intersection over Union

Обнаружение, если
IoU > *p* (пр.: 0.5)



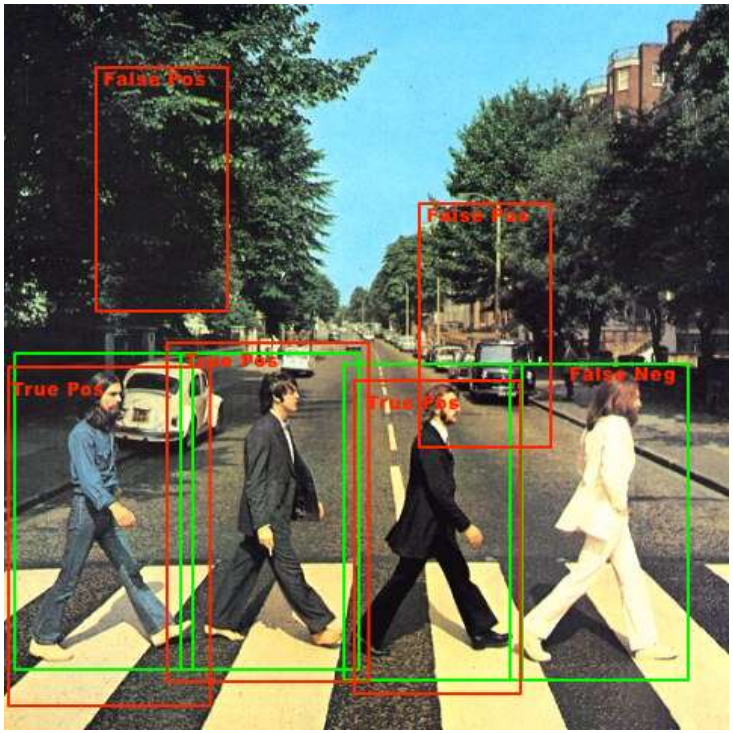
Оценка качества детектора

- Выход алгоритма: отсортированные по качеству обнаружения
- Все обнаружения оцениваются:
  - IoU > p => true positive
  - IoU < p => false positive
- Пропущенный пример => false negatives



Precision & recall

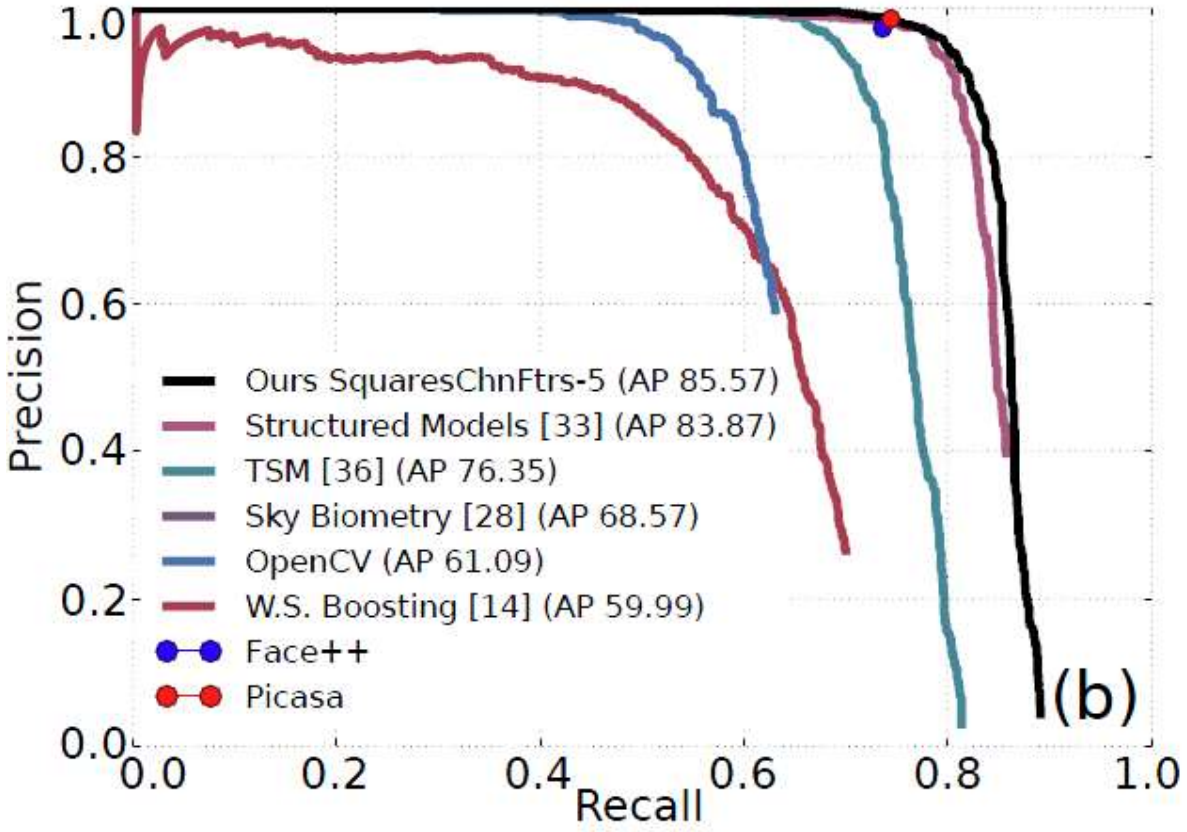
- Точность (Precision)
  - Доля истинных объектов основного класса среди всех классифицированных, как основной класс
- Полнота (Recall)
  - Доля правильно распознанных объектов основного класса среди всех объектов основного класса из тестовой выборки



$$Precision = \frac{Number\ of\ true\ detections}{Number\ of\ detections}$$

$$Recall = \frac{Number\ of\ true\ detections}{Number\ of\ gt\ objects}$$

Кривая точности-полноты



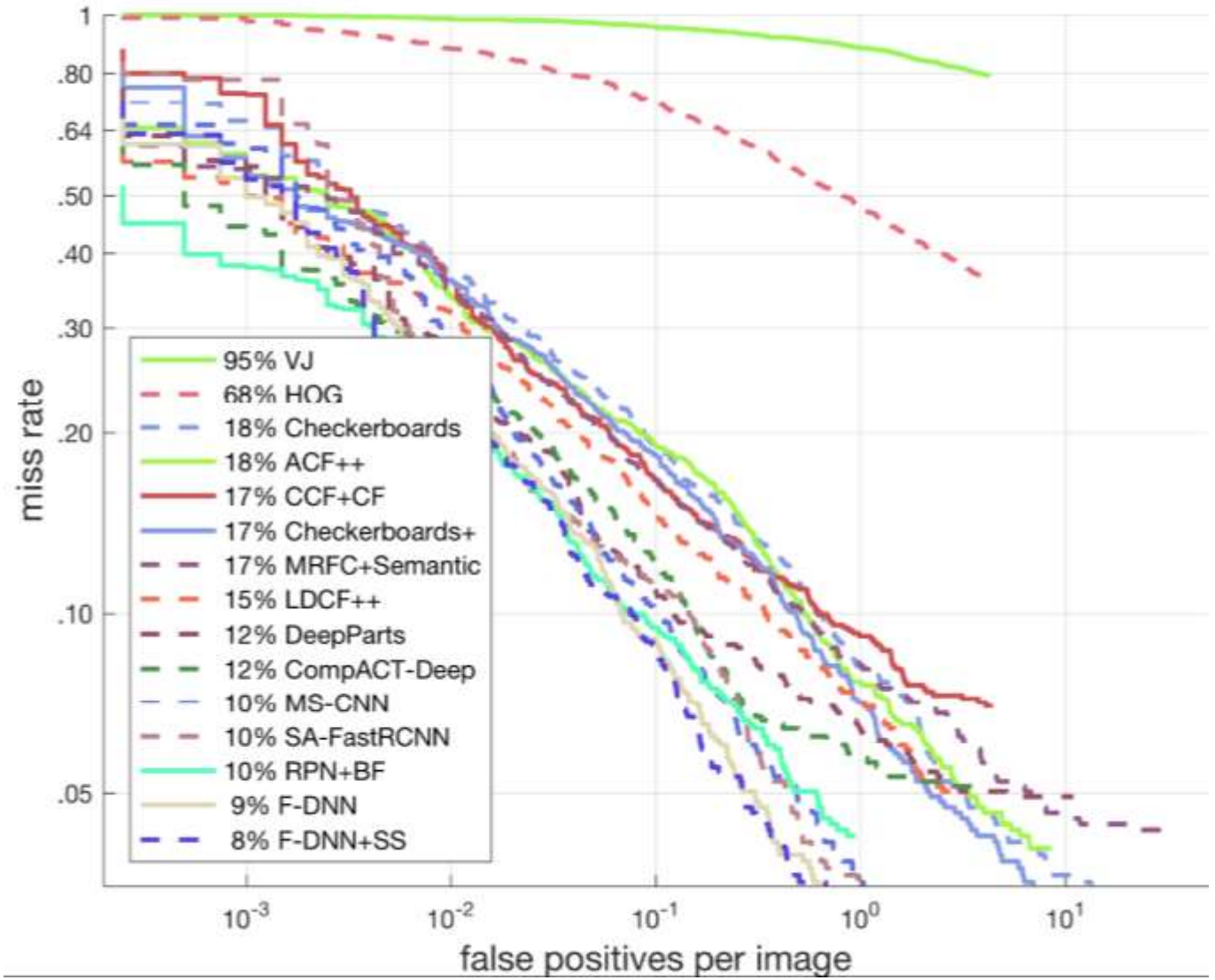
- Зависимость между точностью и полнотой в зависимости от параметров
- Average Precision (AP):

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{interp}(r)$$

- Mean Average Precision (mAP): average across classes
- По кривой точности-полноты мы можем выбрать параметр с оптимальным для наших целей соотношением ошибок

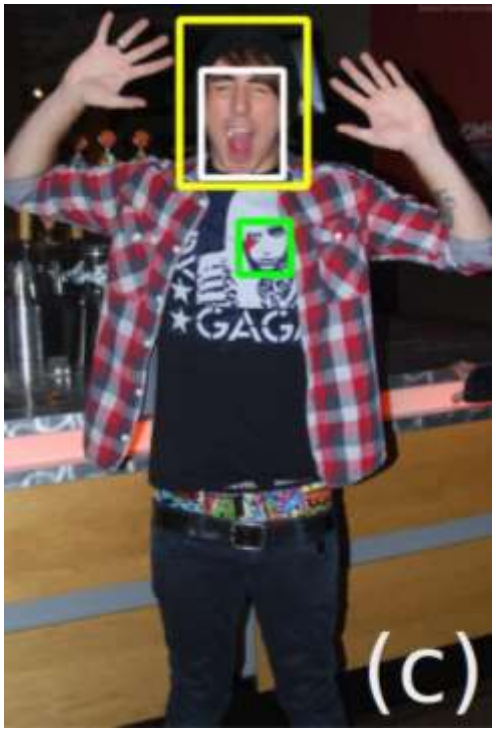
M. Mathias. et. al. Face detection without bells and whistles. ECCV2014

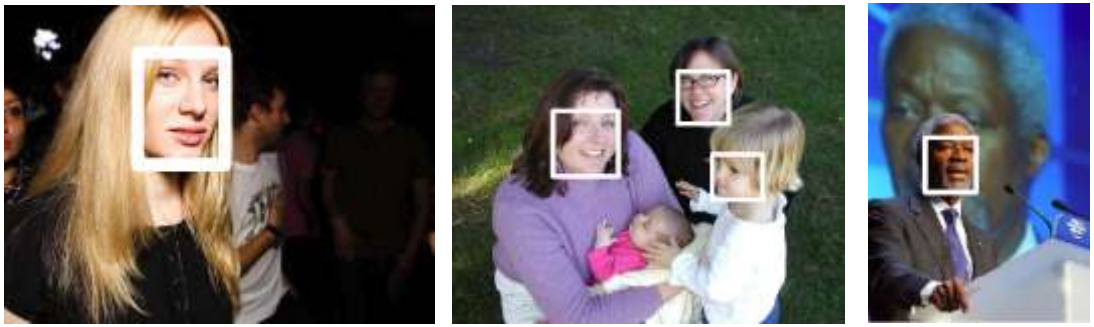
Miss rate vs FPPI



Source: http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

Проблемы с разметкой





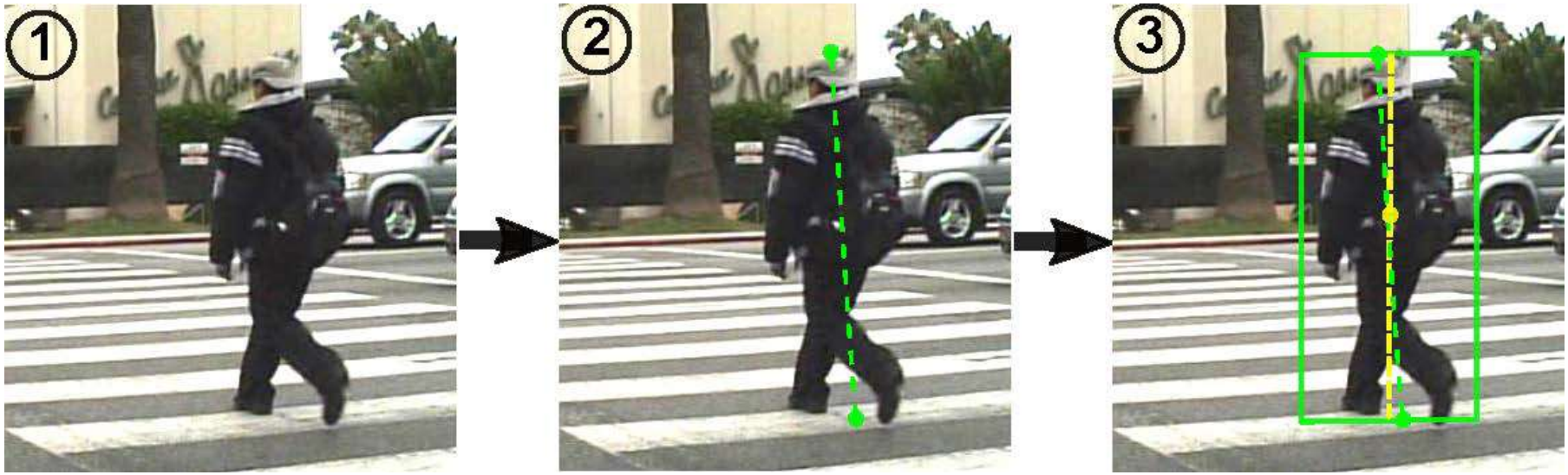
Исходная аннотация

Скорректированная разметка

- Разметка в разных датасетах и от разных асессоров существенно различается
- Есть объекты, которые лучше игнорировать
- Почти всегда есть проблемы с маленькими и большими объектами

M. Mathias. et. al. Face detection without bells and whistles. ECCV2014

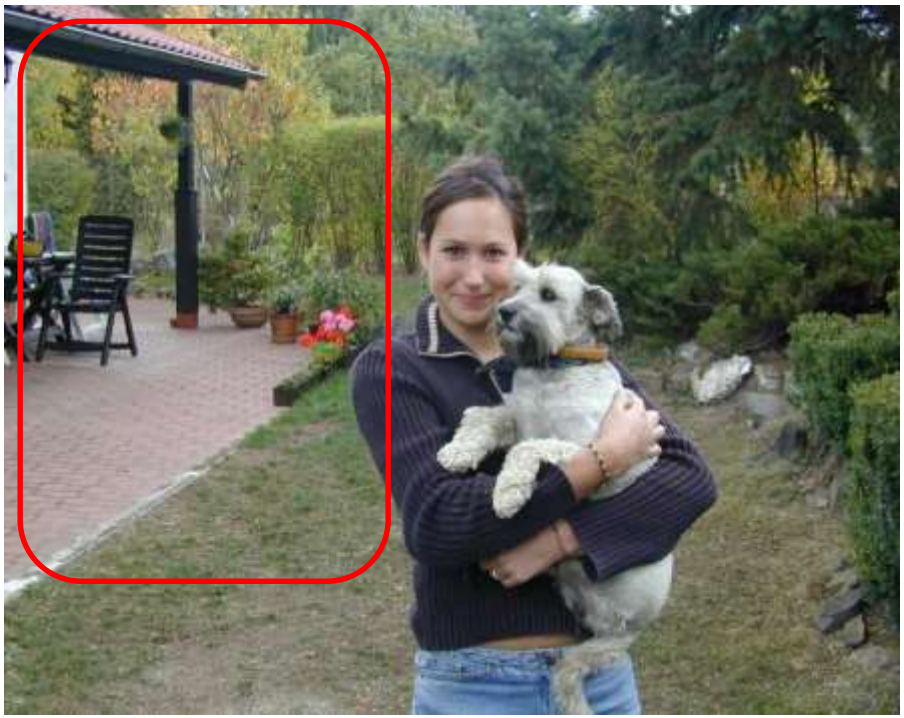
Протокол разметки



- Строгий протокол разметки очень важен!
- Лучше согласовывать разметку от нескольких ассессоров
- Пример – разметка пешеходов в CityPersons

S. Zhang et. al. How Far are We from Solving Pedestrian Detection? CVPR 2016

Сведение выделения к классификации

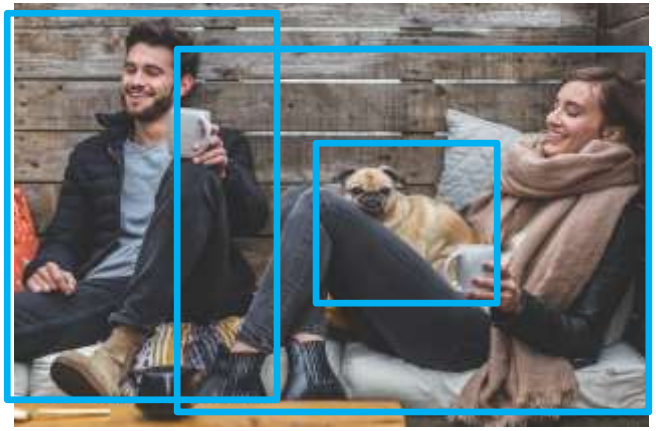


- Мы умеем обучать бинарные классификаторы «объект»/ «не объект». Как нам решить задачу локализации?
- Скользящее окно – просмотр всех фрагментов изображения и применения к ним классификатора

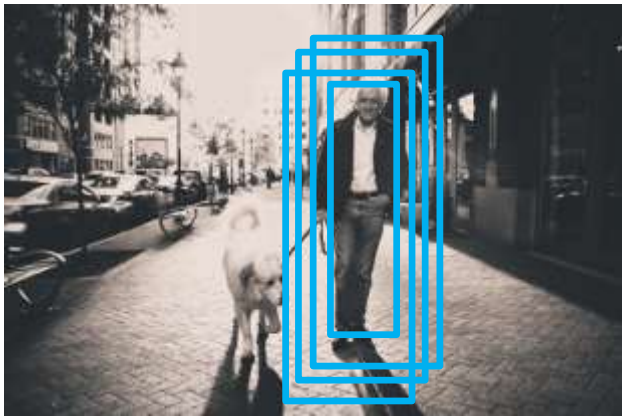
Проблемы скользящего окна

Разные размеры

Разные пропорции



Перекрытия и сложная форма

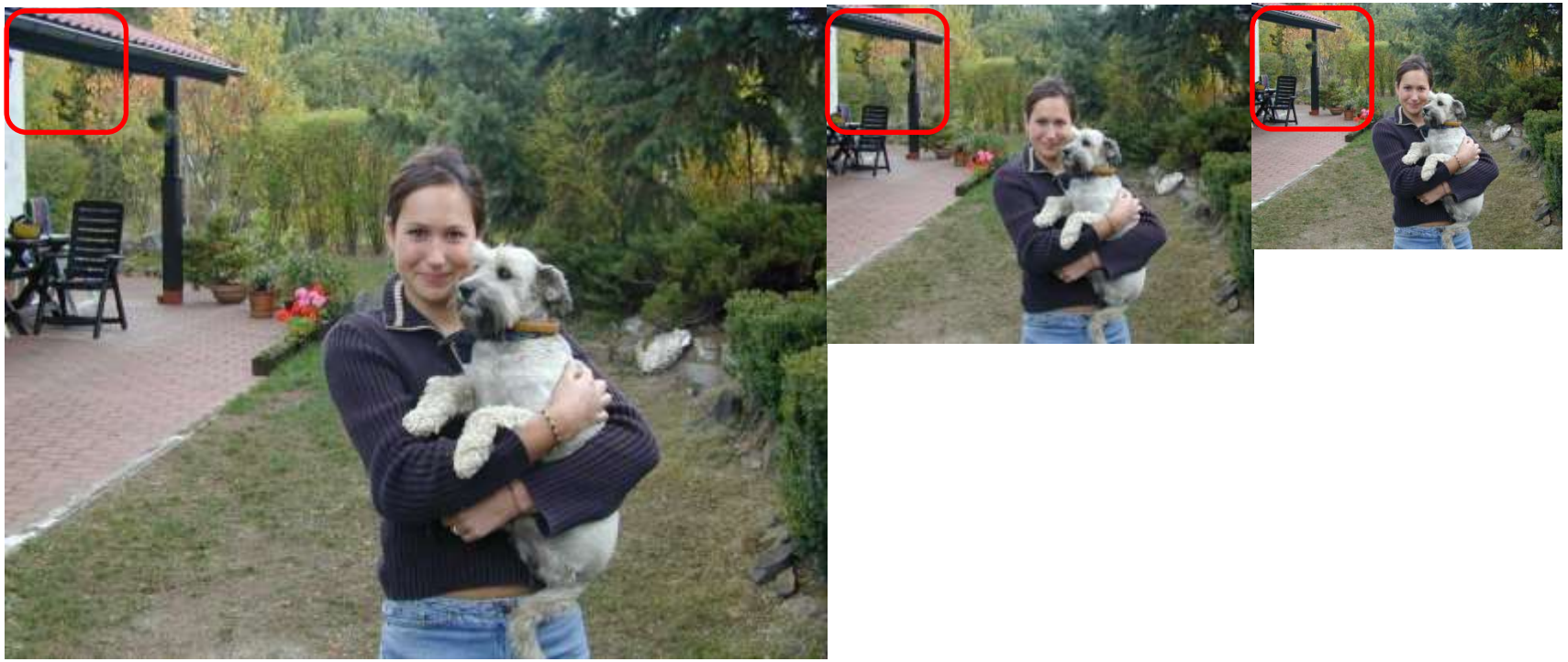


Множественные отклики

Разные размеры окна

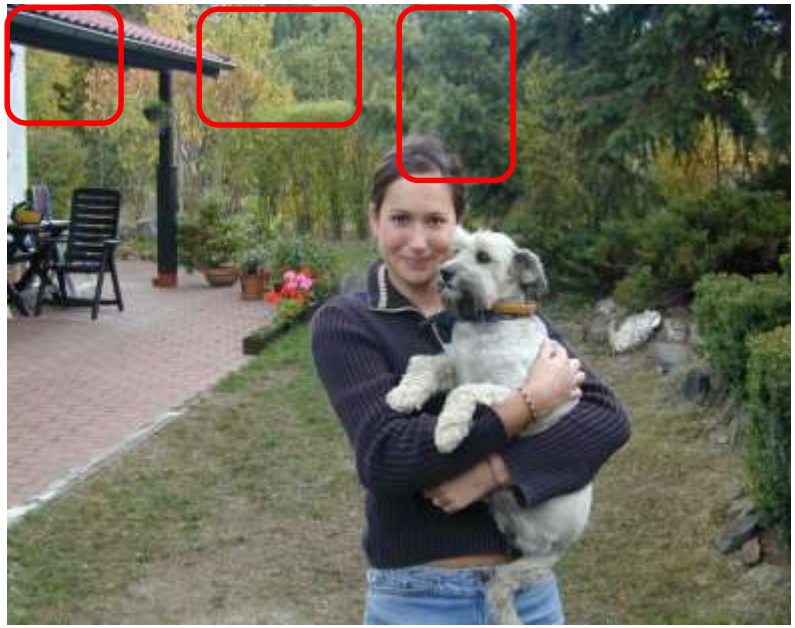
Сканируем изображение окнами разного размера

Разные размеры объектов



- Строим пирамиду масштабов и сканируем окном одного размера
- В зависимости от качества признаков и классификатора может потребоваться разное число масштабов

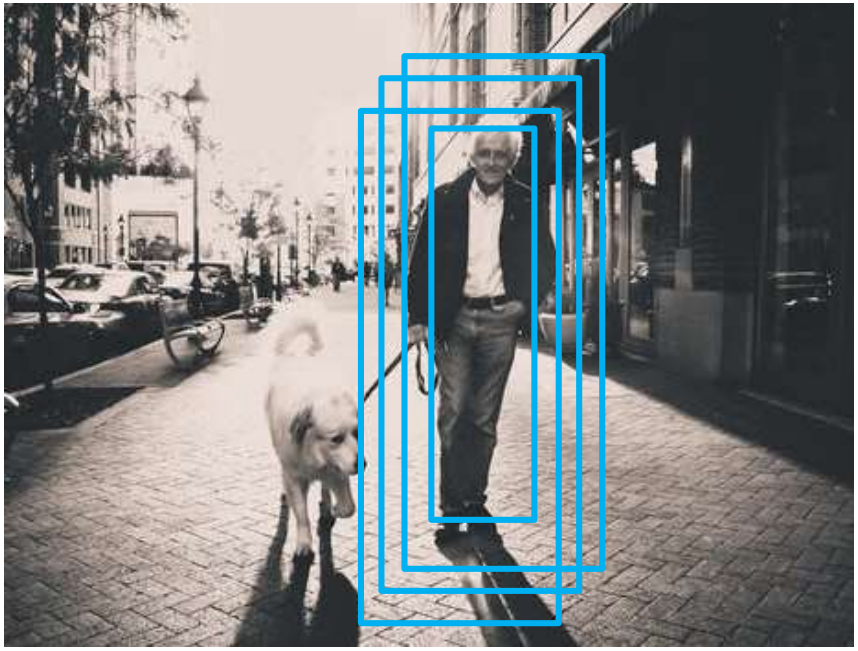
Разные пропорции объектов



- Используем набор окон с разными пропорциями
- “Number of images scanned” =

  “Number of scales” x “Number of aspect ratios”

Множественные отклики



Множественные отклики вокруг объекта

Как выбрать наилучшие?

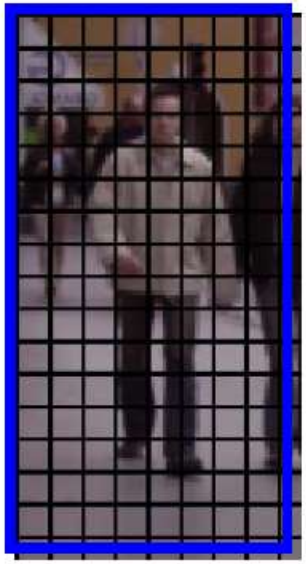
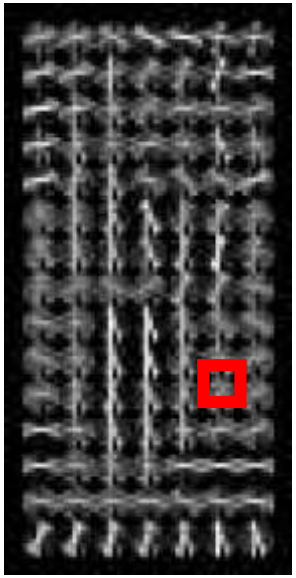
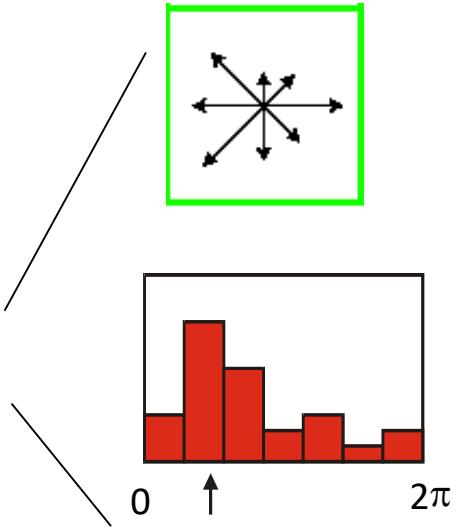
- Алгоритм жадного подавления
  - На вход список окон, отсортированных по убыванию score (выхода классификатора)
  - Выбираем окно с наибольшим score
  - Находим и убираем все окна, для которых IoU > 0.5
  - Повторяем до сходимости

Сопоставление шаблонов

- Рассмотрим простейший алгоритм
- Пусть изображение объекта «фиксировано», т.е. мало меняется
- Назовём изображение объекта шаблоном (template)
- В качестве вектор-признака изображения I в области r векторизацию фрагмента изображения r(I)
  - Длина вектора признака – число пикселей фрагмента
  - Можем брать цветной фрагмент
- В качестве классификатора будем сравнивать шаблон с окном, например, по L2-метрики
  - $d = Distance\ (x, template)$
  - $y = \begin{cases} 0, d \geq t \\ 1, d < t \end{cases}$
  - где t – порог на сходство изображений

template

HOG+SVM для поиска пешеходов

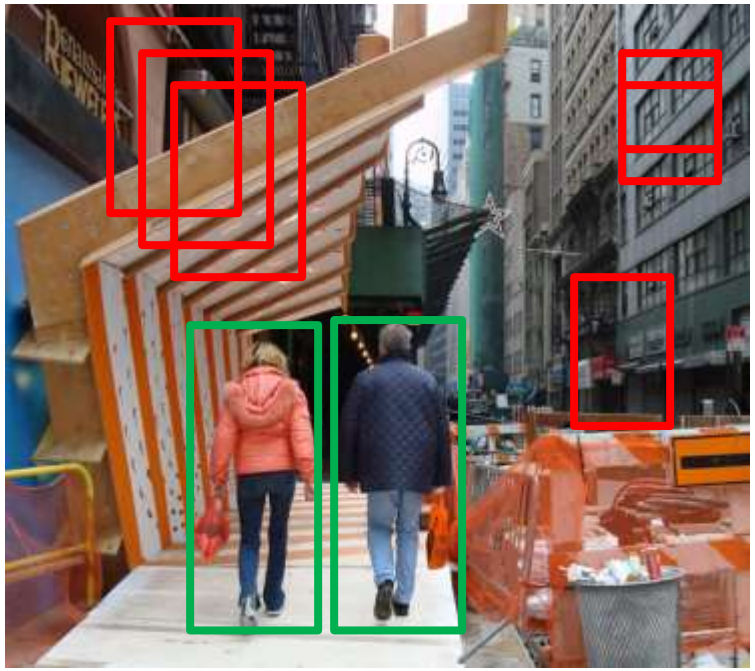




- Скользящее окно
- Признаки – HOG/SIFT
  - Окно 64 x 128 пикселей
  - Разобъём его на блоки 8 x 8 пикселей
  - Всего будет 8 * 16 = 128 блоков
  - В каждом блоке посчитаем гистограмму ориентаций градиентов с 8 ячейками (8 параметров)
  - Всего у нас получится 128 x 8 = 1024 признака
- Обучим линейный SVM признак

Navneet Dalal, Bill Triggs,Histograms of Oriented Gradients for Human Detection, CVPR 2005

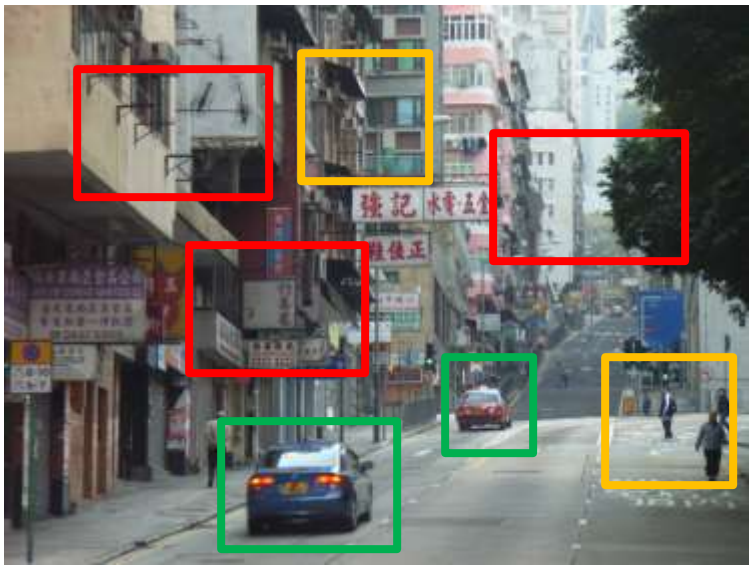
Обучение детектора

- Сколько на изображении объектов «пешеход» и сколько фрагментов фона?



- Выделение объектов ассиметричная задача: объектов гораздо меньше, чем «не-объектов»
- Вдобавок, класс «не объект» очень сложный – нужно много разных данных для обучения
- Для SVM желательно одинаковое количество и фона, и объекта

Hard-negative mining



- Выбираем отрицательные примеры случайным образом
- Обучаем классификатор
- Применяем к данным
- Добавляем ложные обнаружение к выборке
- Повторяем

- Смысл:
  - Ложные обнаружения для первого детектора – сложные (**hard negative**)
  - Пусть наша выборка фона будет маленькой, но сложной и представительной

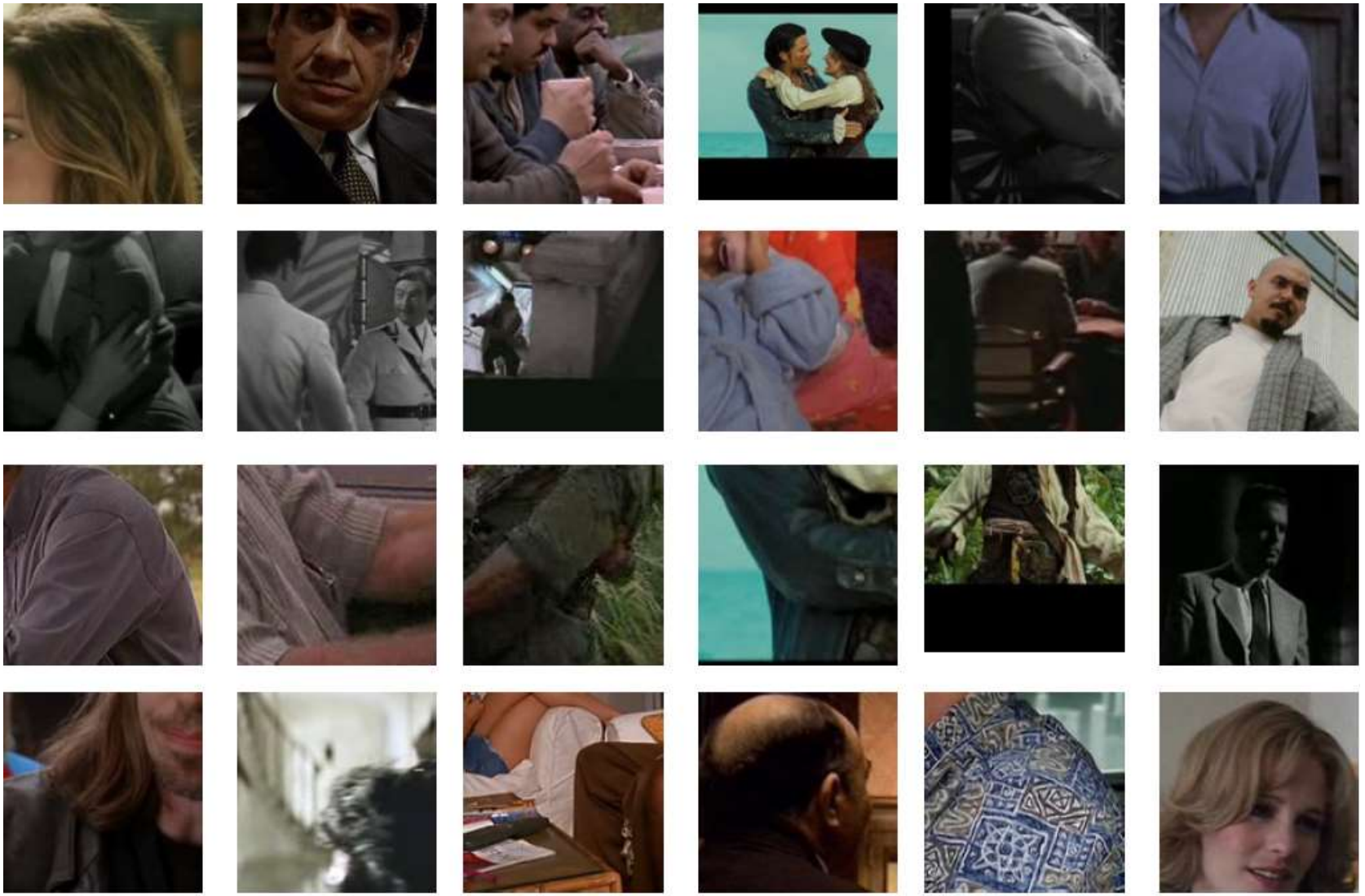
I. Laptev "Improvements of Object Detection Using Boosted Histograms« BMVC 2006

Пример – поиск «торса»

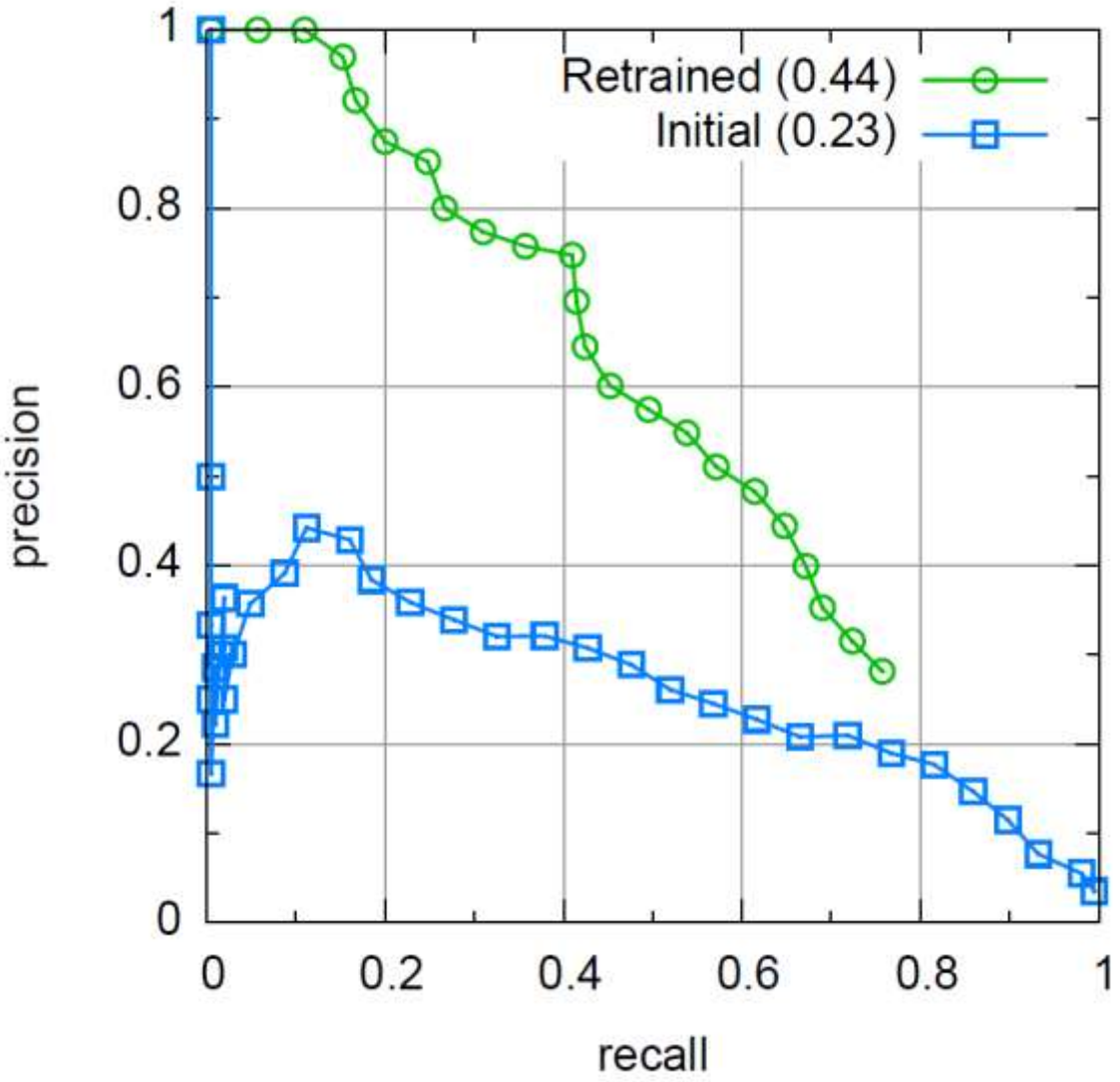
- Хотим построить детектор «верхней части тела и головы»
- Воспользуемся схемой HOG + линейный SVM с бустраппингом
- Данные
  - 33 фрагмента фильмов из базы Hollywood2
  - 1122 кадров с размеченным объектами
- На каждом кадре отмечены 1-3 человека, всего 1607 людей

Исходные случайные фрагменты фона

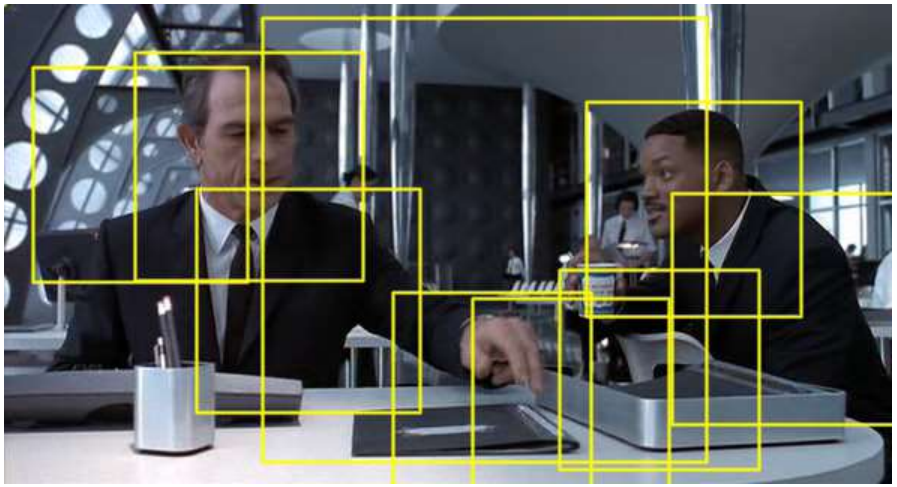
Трудные примеры (ложные обнаружения)

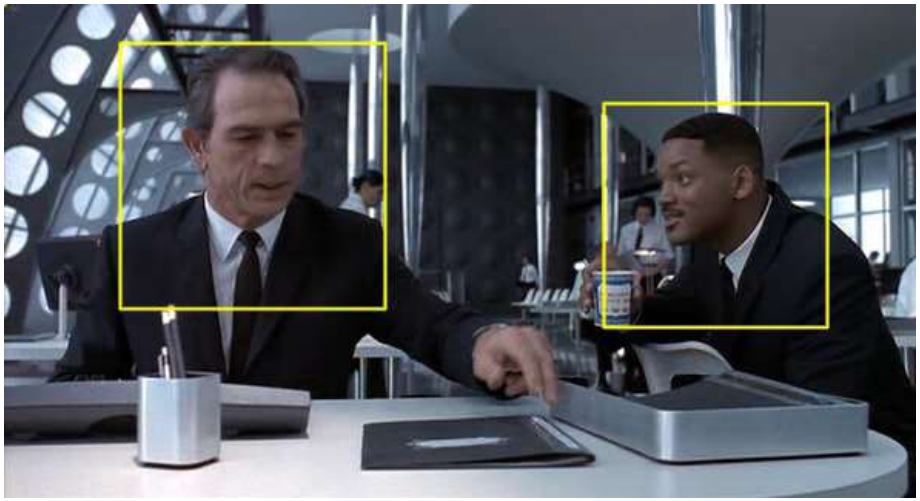


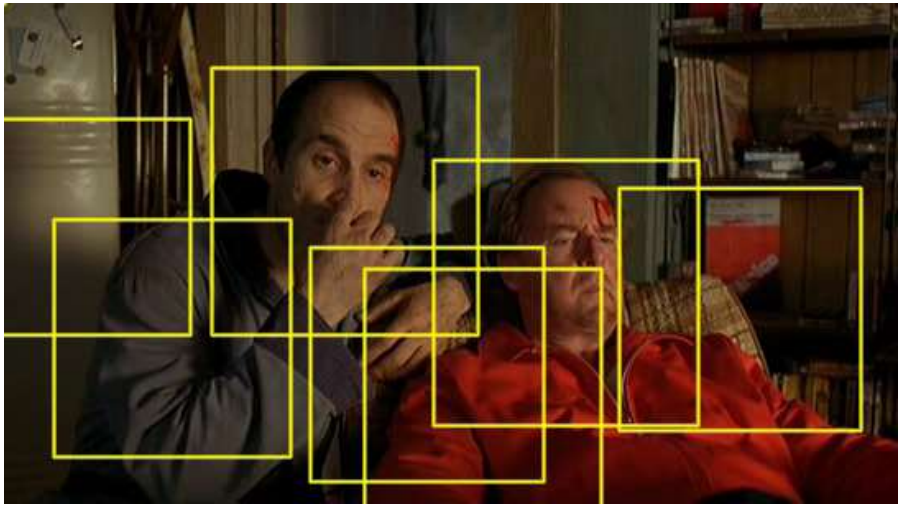
До и после hard-negative mining

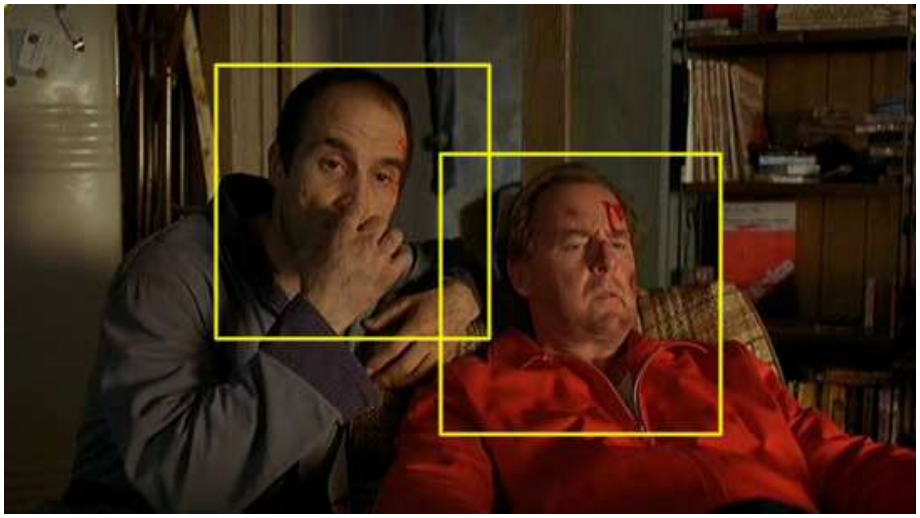


Сравнение



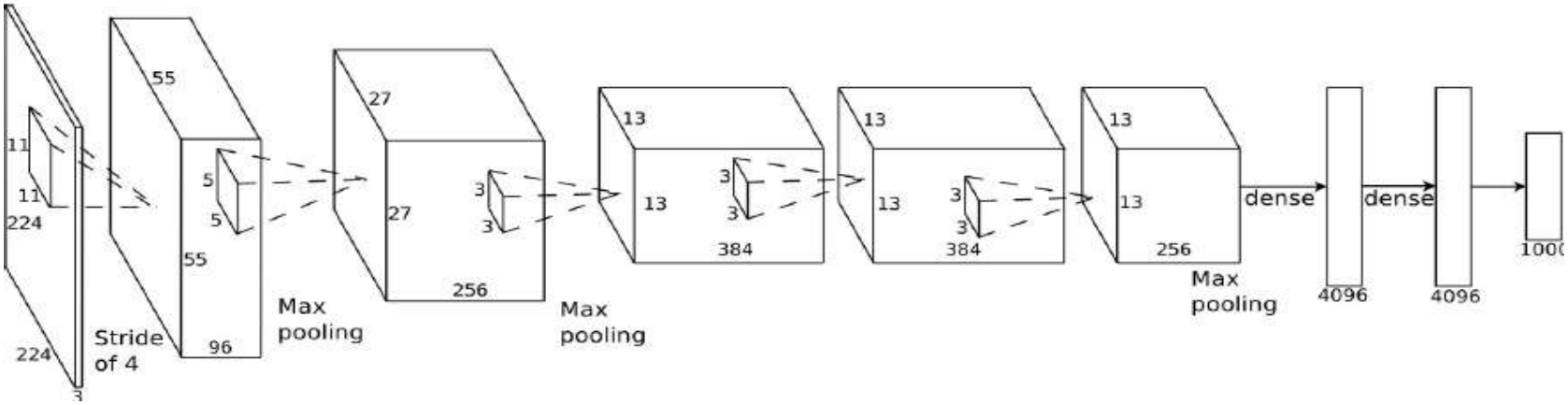






Нейросетевые методы в детекторах

Мы умеем обучать мощные нейросети для классификации изображений



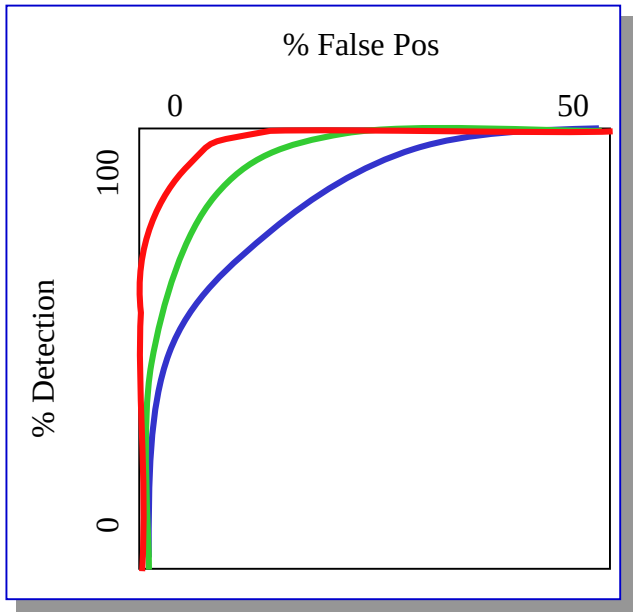
Как будем применять?

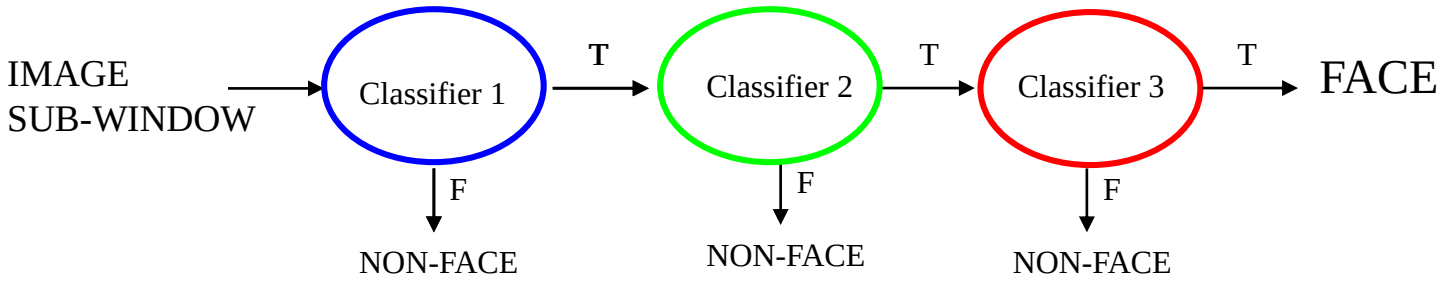
Что нам может помешать?

Каскад (Attentional cascade)

- Цепочка классификаторов
- Положительный отклик первого классификатора запускает вычисление второго, более сложного, классификатора, и т.д.
- Если они срабатывает отрицательно, дальнейшая обработка прерывается
- С каждым уровнем детектор становится более сложным, ошибка второго рода постоянно снижается

ROC-кривая

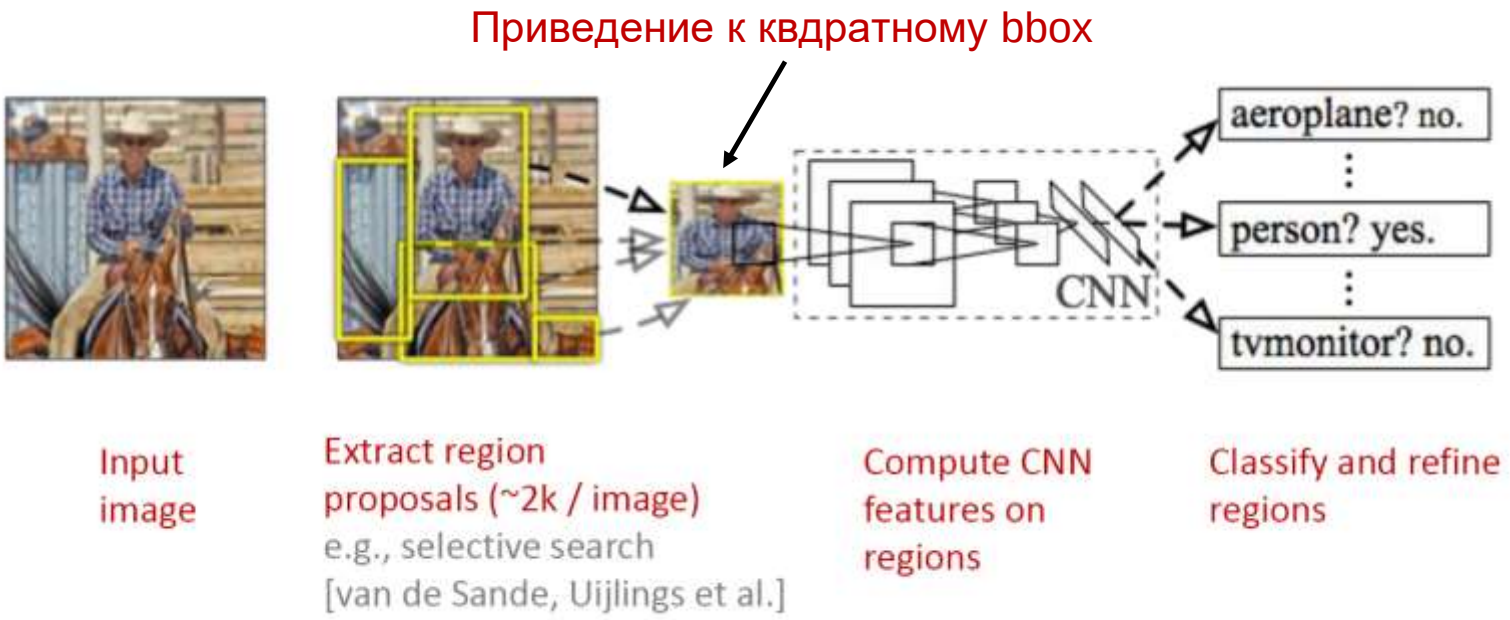




Region-based Convolution Networks (R-CNN)

Двухэтапный каскад:

- Находим гипотезы объектов
- Применяем мощный CNN классификатор к гипотезам



R. Girshick, J. Donahue, T. Darrell, J. Malik Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014

Пример генератора гипотез

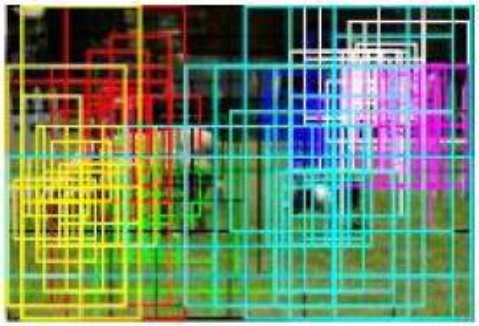
В оригинальном R-CNN алгоритме использовался метод Selective Search

Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

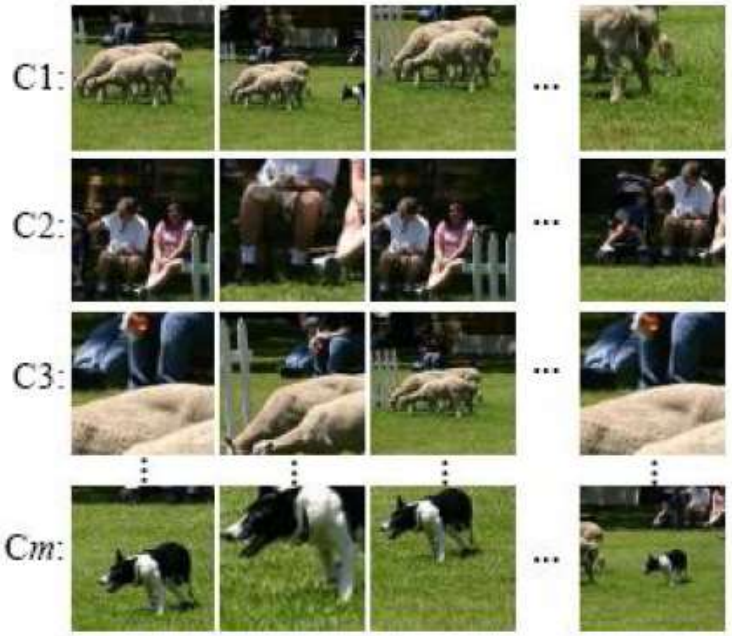
J. R. R. Uijlings et.al. Selective Search for Object Recognition. IJCV 2013

Приведение гипотез к квадратным bbox

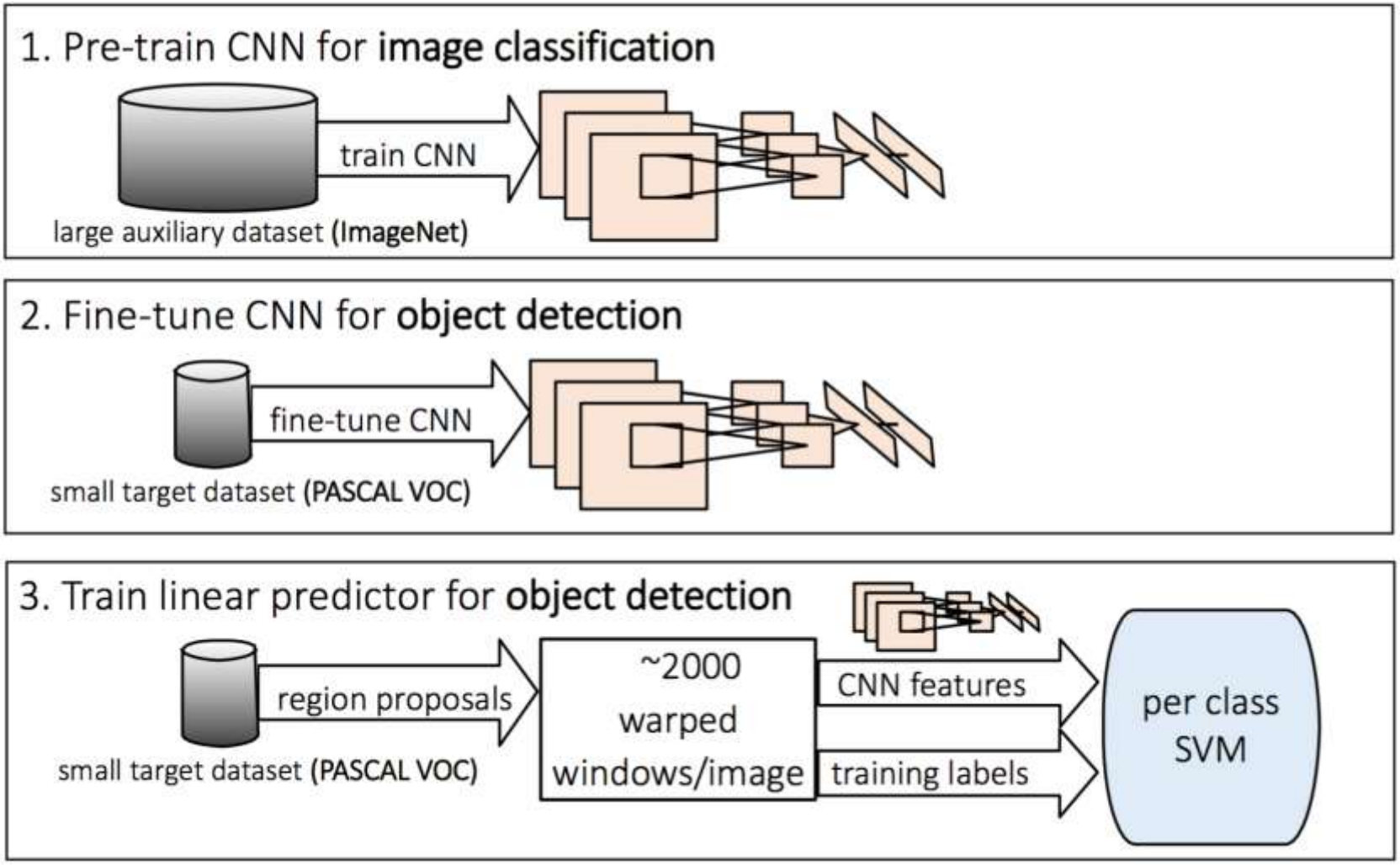
(a)



(b)



Обучение R-CNN



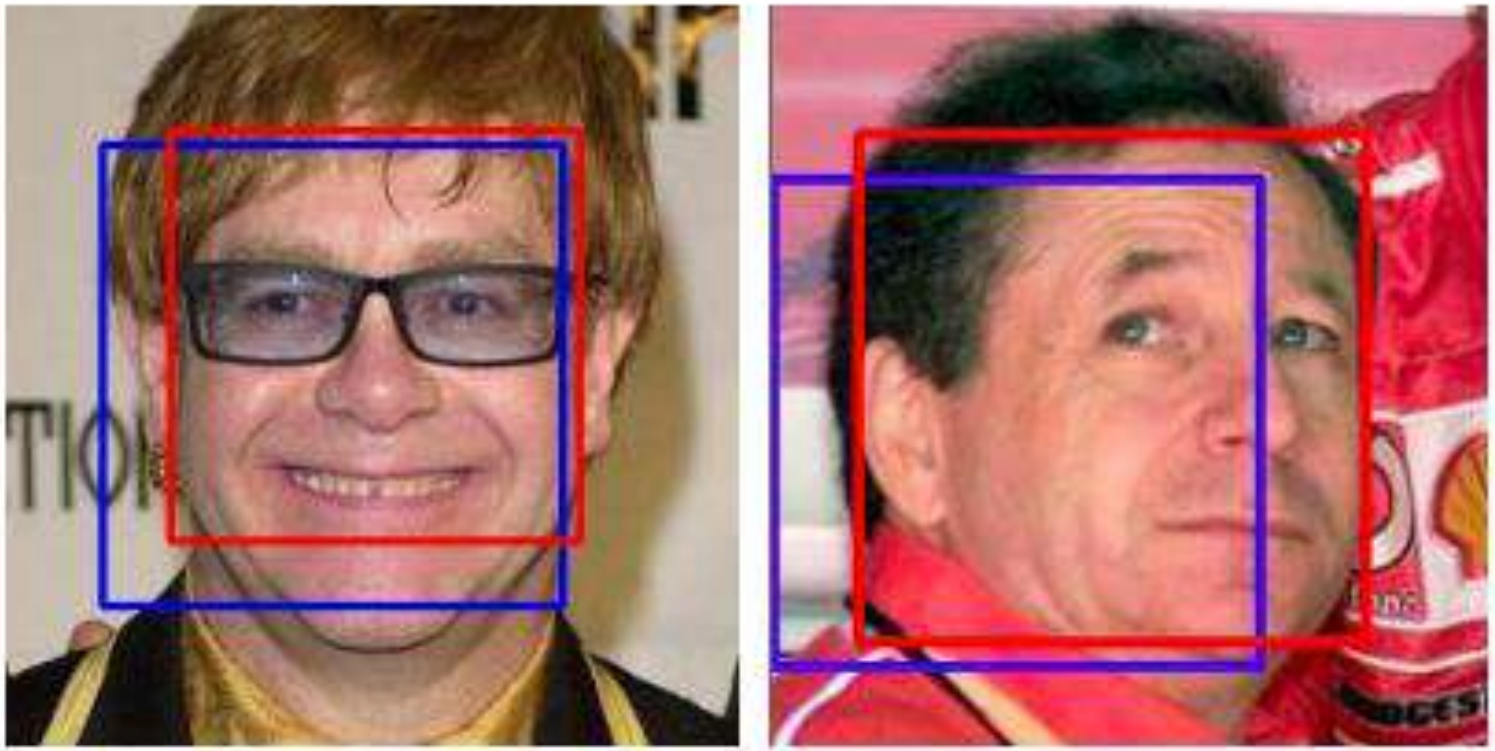
Оценка R-CNN по mAP

		VOC 2007	VOC 2010
reference	DPM v5 (Girshick et al. 2011)	33.7%	29.6%
	UVA sel. search (Uijlings et al. 2012)		35.1%
	Regionlets (Wang et al. 2013)	41.7%	39.7%
pre-trained only	R-CNN $pool_5$	44.2%	
	R-CNN fc_6	46.2%	
	R-CNN fc_7	44.7%	
fine-tuned	R-CNN $pool_5$	47.3%	
	R-CNN fc_6	53.1%	
	R-CNN fc_7	54.2%	50.2%%
	R-CNN fc_7 (Bounding Box regression)	**58.5%**	53.7%

Обратите внимание на последнюю строку

Уточнение bbox

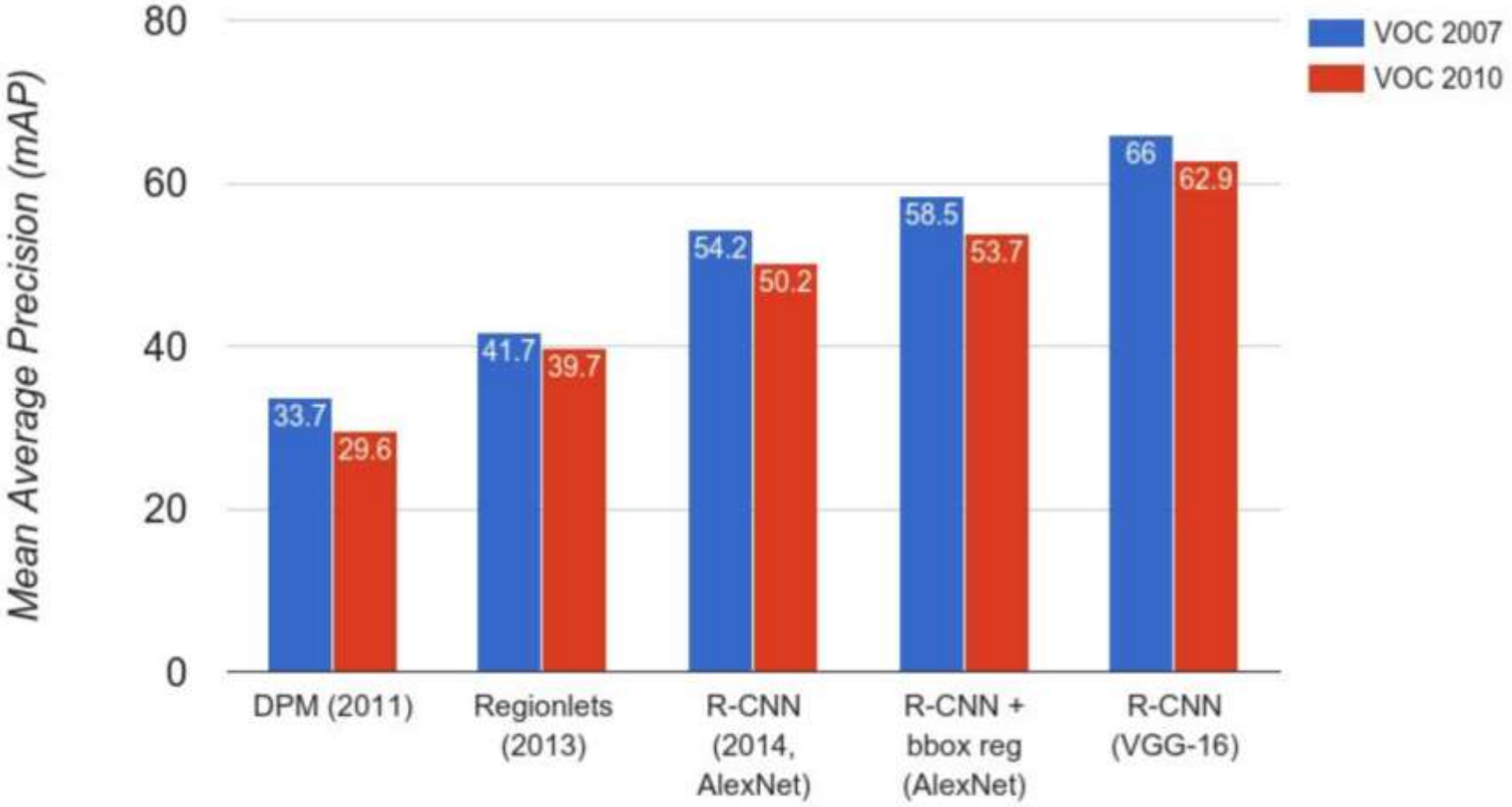
- Исходные гипотезы bbox могут быть очень неточными
- Мы можем уточнить (регрессировать) координаты bbox, используя те же признаки, что и для классификации
- Обычно используется линейный регрессор



Синие прямоугольники – до уточнения, красные -после

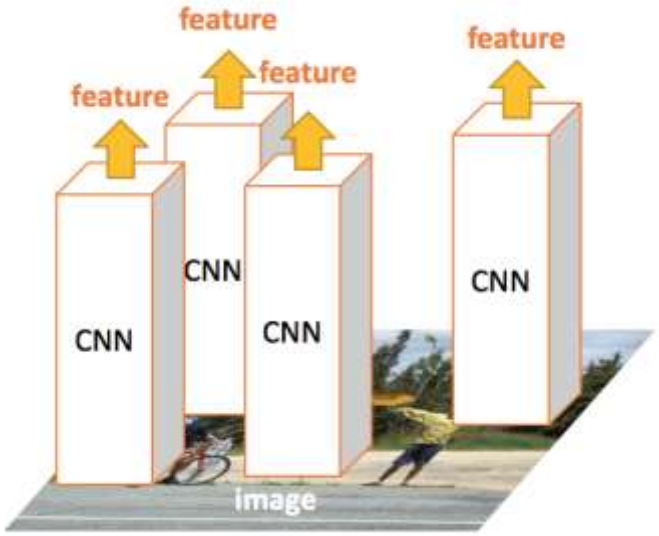
Развитие R-CNN

Улучшение классификатора (AlexNet –> VGG16) повышает качество детектора



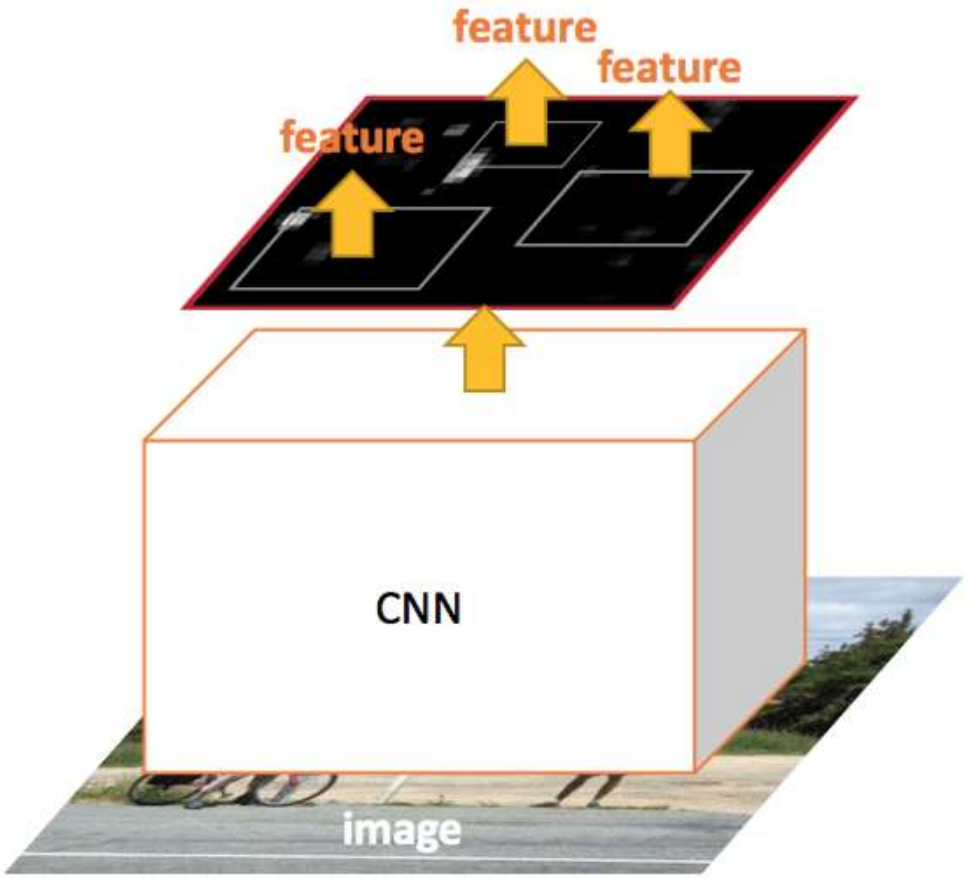
Оценка точности на Pascal VOC

Что плохо в R-CNN?



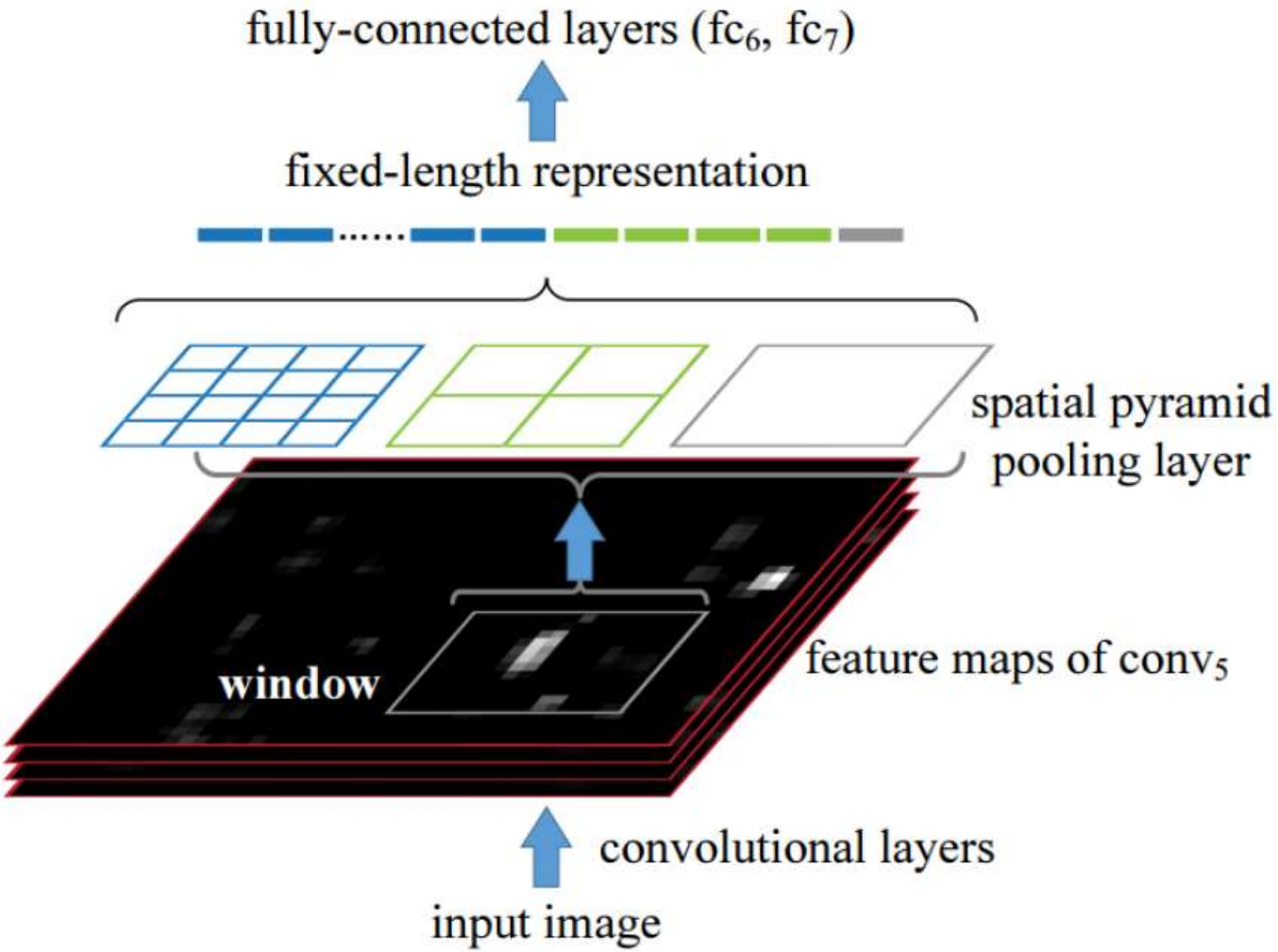
- Вычисляем признаки для каждого окна-гипотезы. Поскольку они пересекаются, это ведёт к избыточным вычислениям
- Нужно масштабировать фрагменты-гипотез до нужного разрешения
- Сложная процедура обучения
- Зависимость от внешнего алгоритма генерации гипотез

ROI-pooling



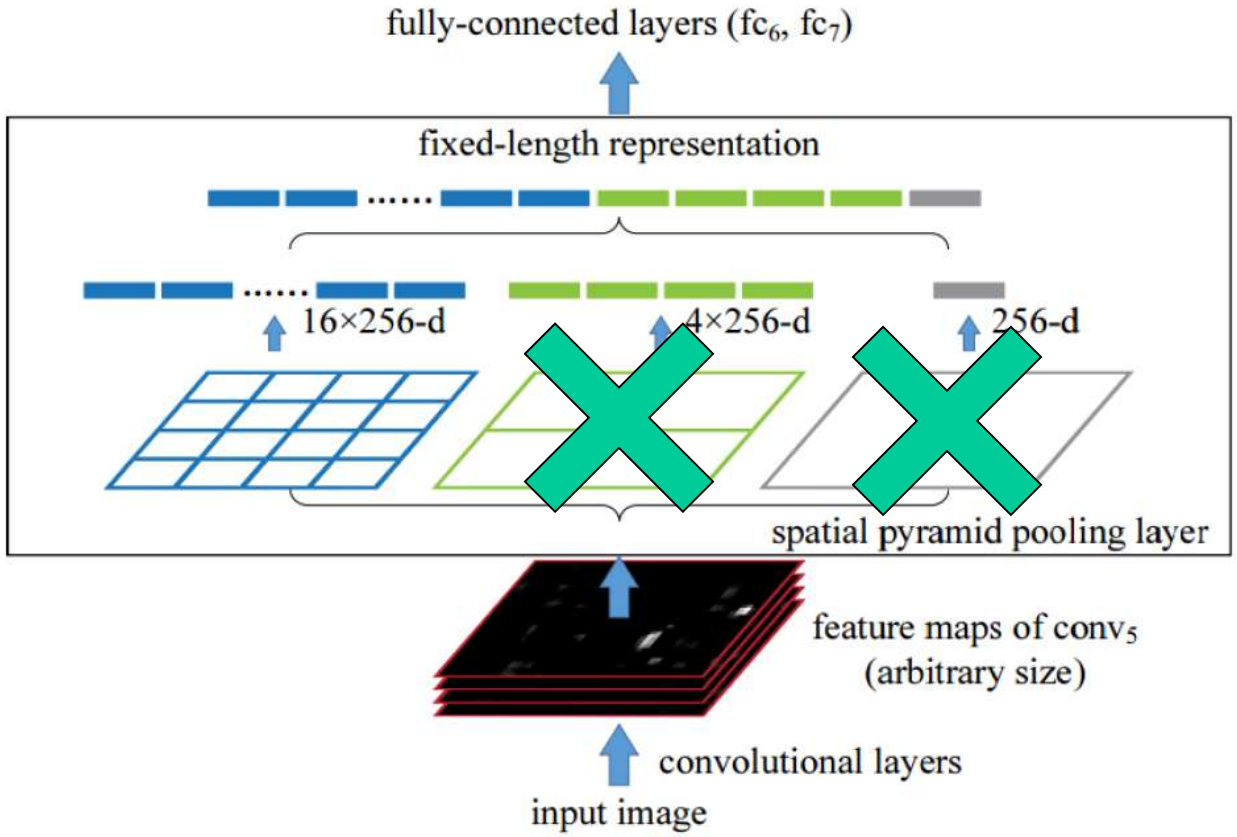
- ROI = Region of Interest
- Хотим вычислить нейросетевые признаки один раз для всего изображения
- Вычислять вектор-признак для заданных гипотез по этим признакам

Spatial pyramid pooling (SPP)



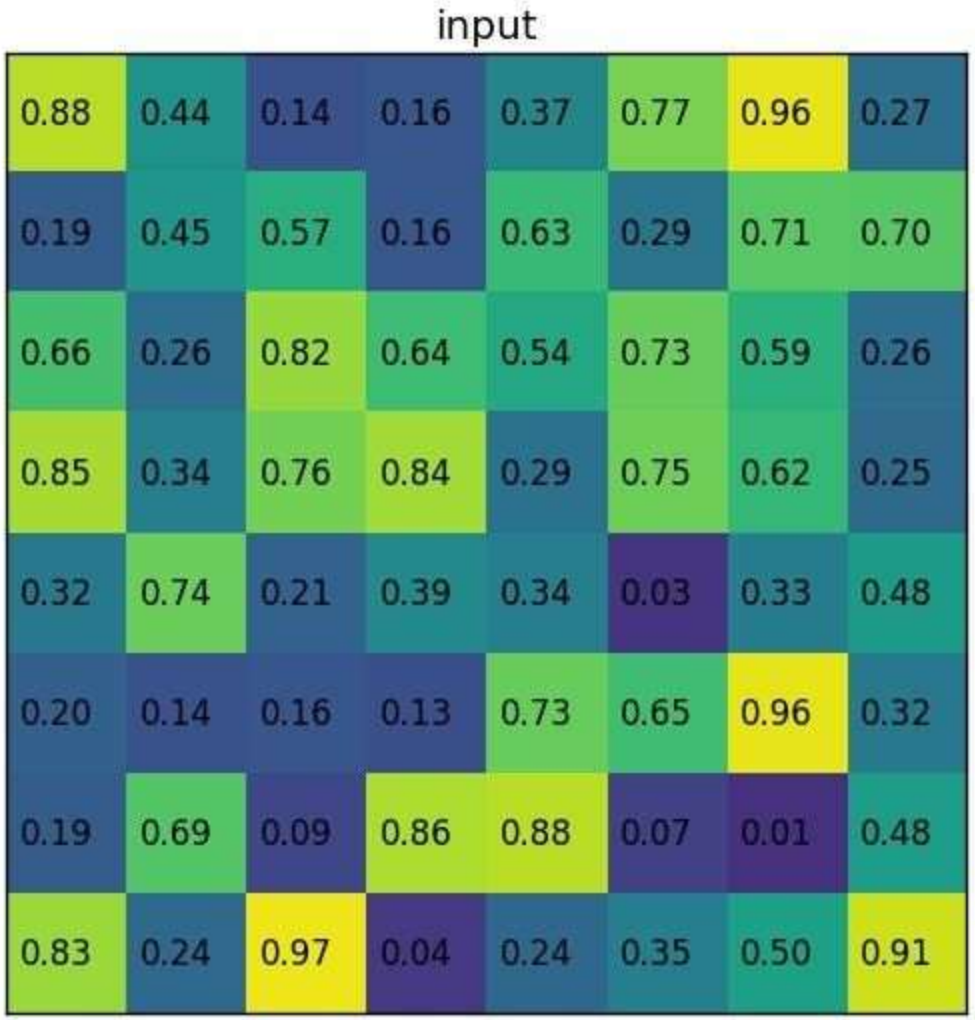
ROI-pooling

Вариант SPP с одним уровнем пирамиды



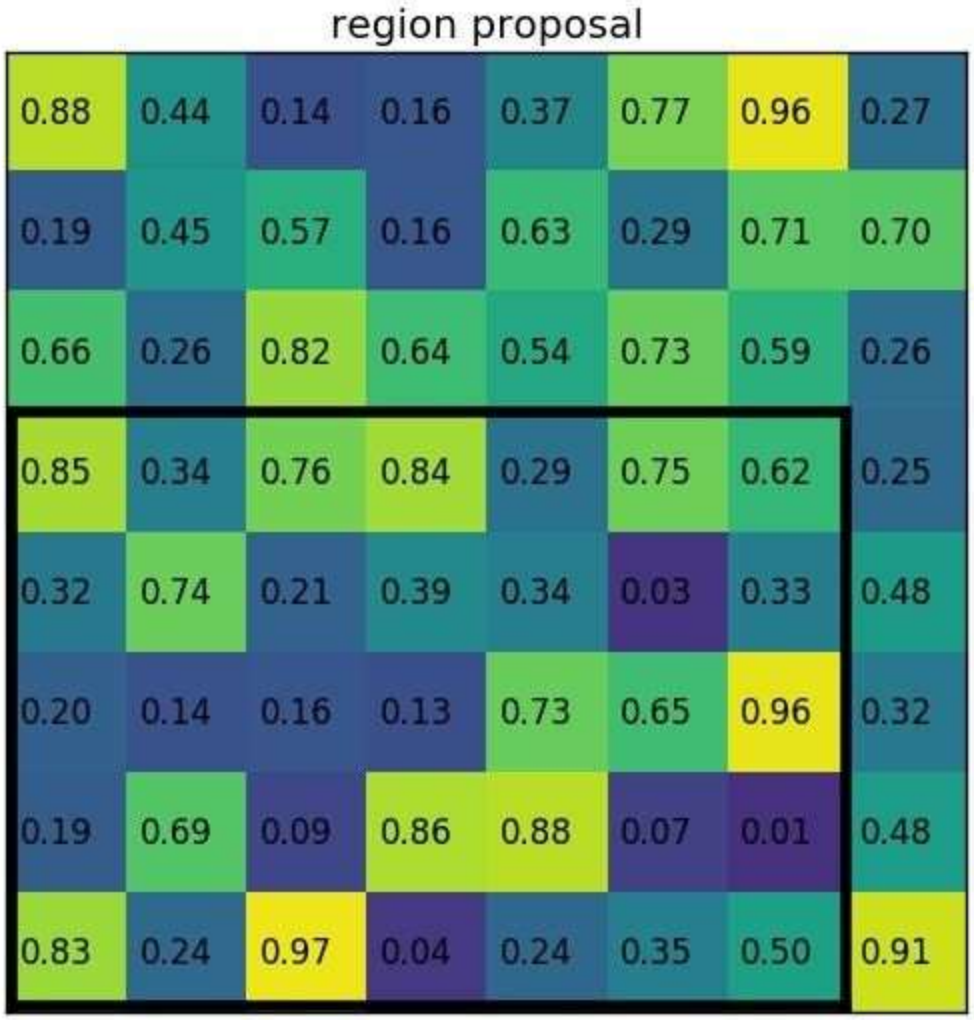
R. Girshick. Fast R-CNN. ICCV 2015

Пример: ROI-pooling в 2x2



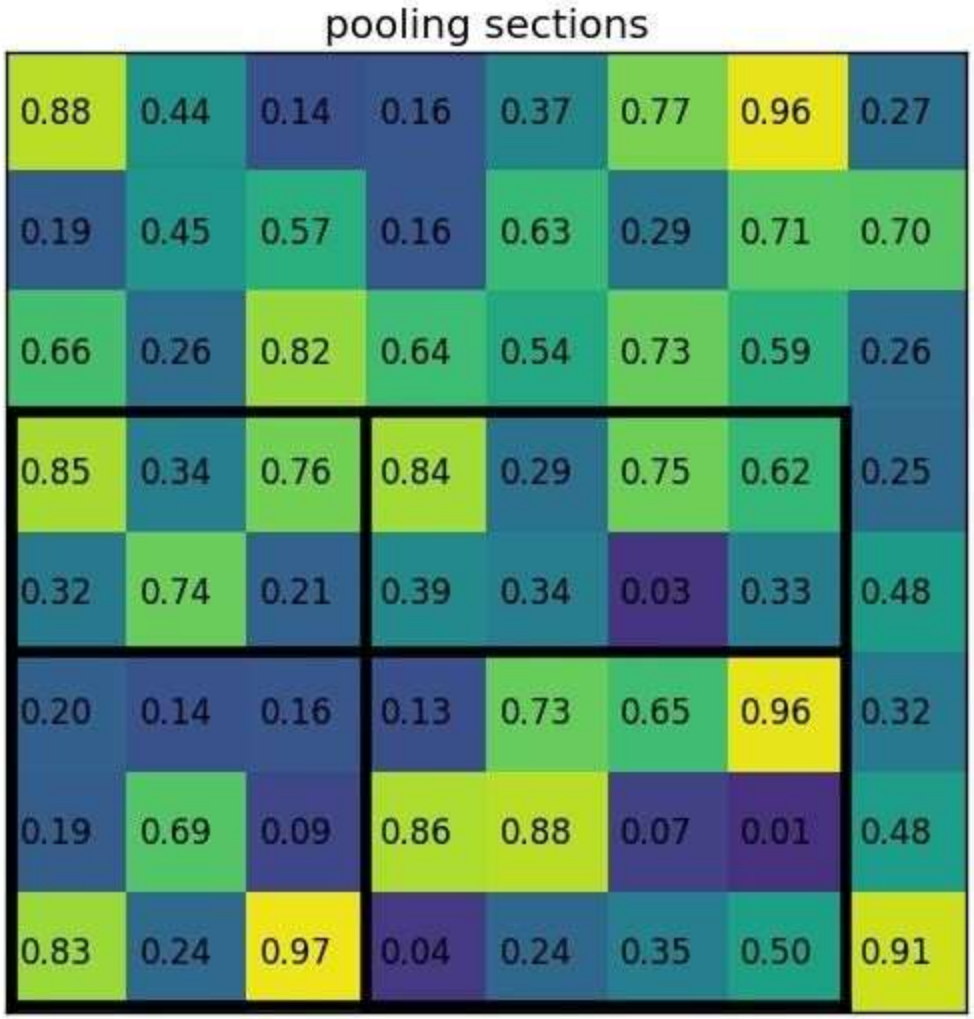
Source: https://deepsense.ai/region-of-interest-pooling-explained/

Пример: ROI-pooling в 2x2



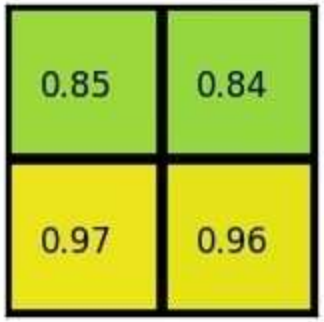
Source: https://deepsense.ai/region-of-interest-pooling-explained/

Пример: ROI-pooling в 2x2



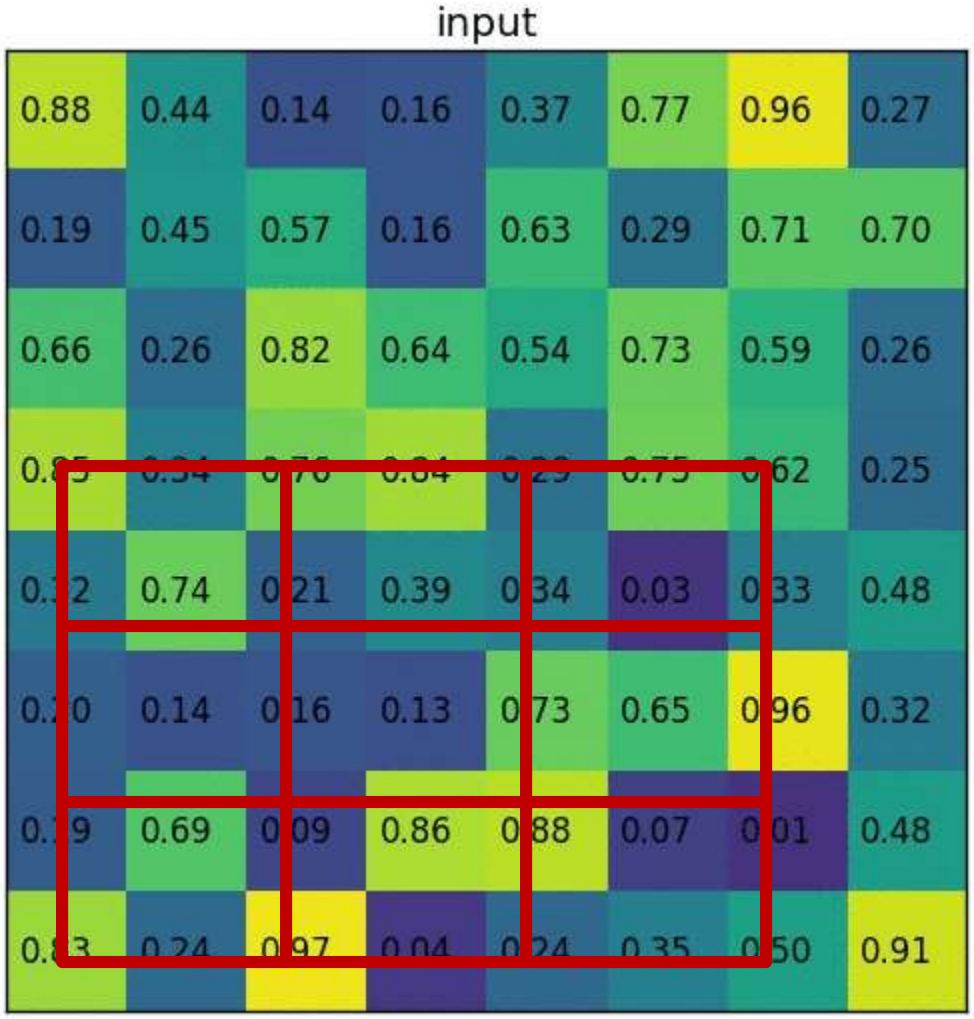
Source: https://deepsense.ai/region-of-interest-pooling-explained/

Пример: ROI-pooling в 2x2



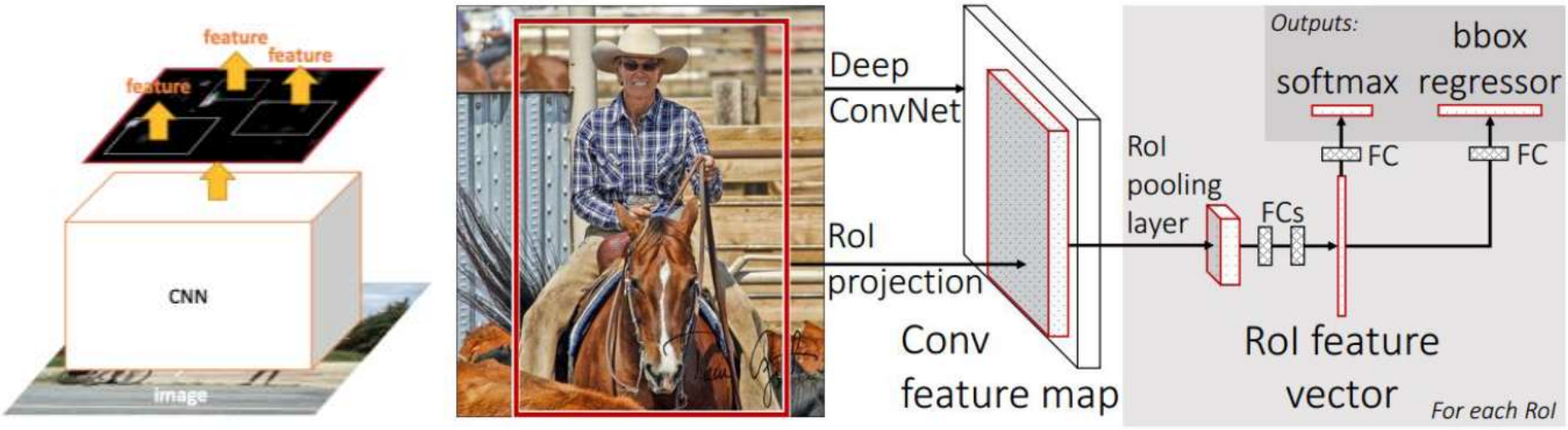
Source: https://deepsense.ai/region-of-interest-pooling-explained/

ROI-Align



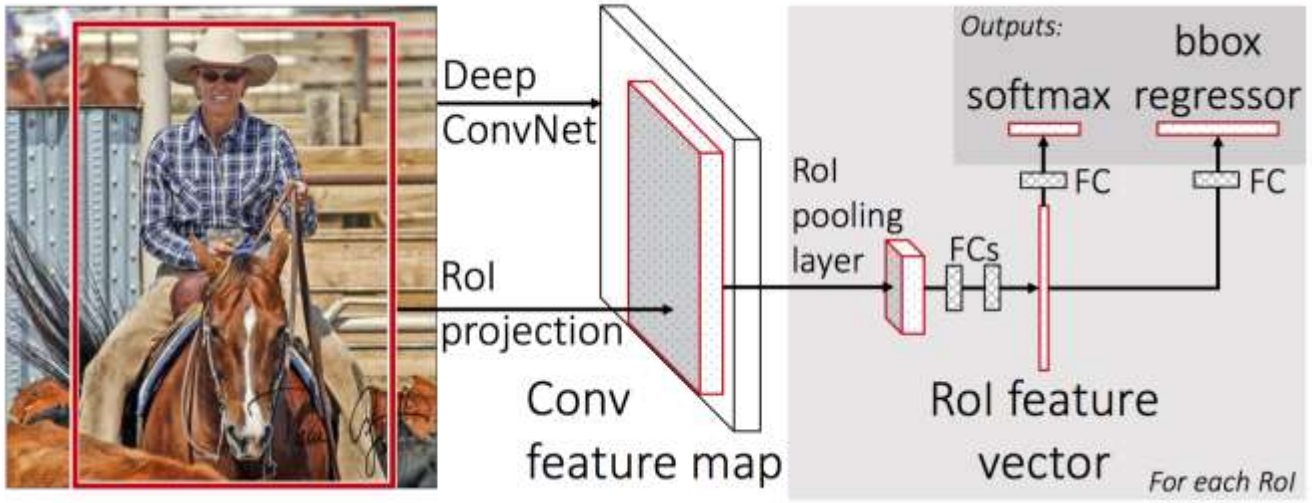
- Вместо операции MAX-pooling в окнах с дискретизированными координатами
- Rescale выбранную область в карту нужно разрешения (4x4 в примере)
- Значения получаем с помощью билинейной интерполяции
- В ряде случаев повышает mAP

Fast R-CNN



- Вычисляем нейросетевые признаки по всему изображению
- Вычисляем признаки гипотез с помощью ROI-pooling (гипотезы поступают извне)
- Вместо обучения SVM обучаем на этих признаках нейросеть с 2-мя типами выходов – классификатором типа объекта и регрессором bbox относительно гипотезы
- Можем обучать всё вместе

Fast R-CNN: multi-task loss



- Multi-task loss $L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v)$
- Classification loss $L_{\text{cls}}(p, u) = -\log p_u$
- Bounding box regression loss

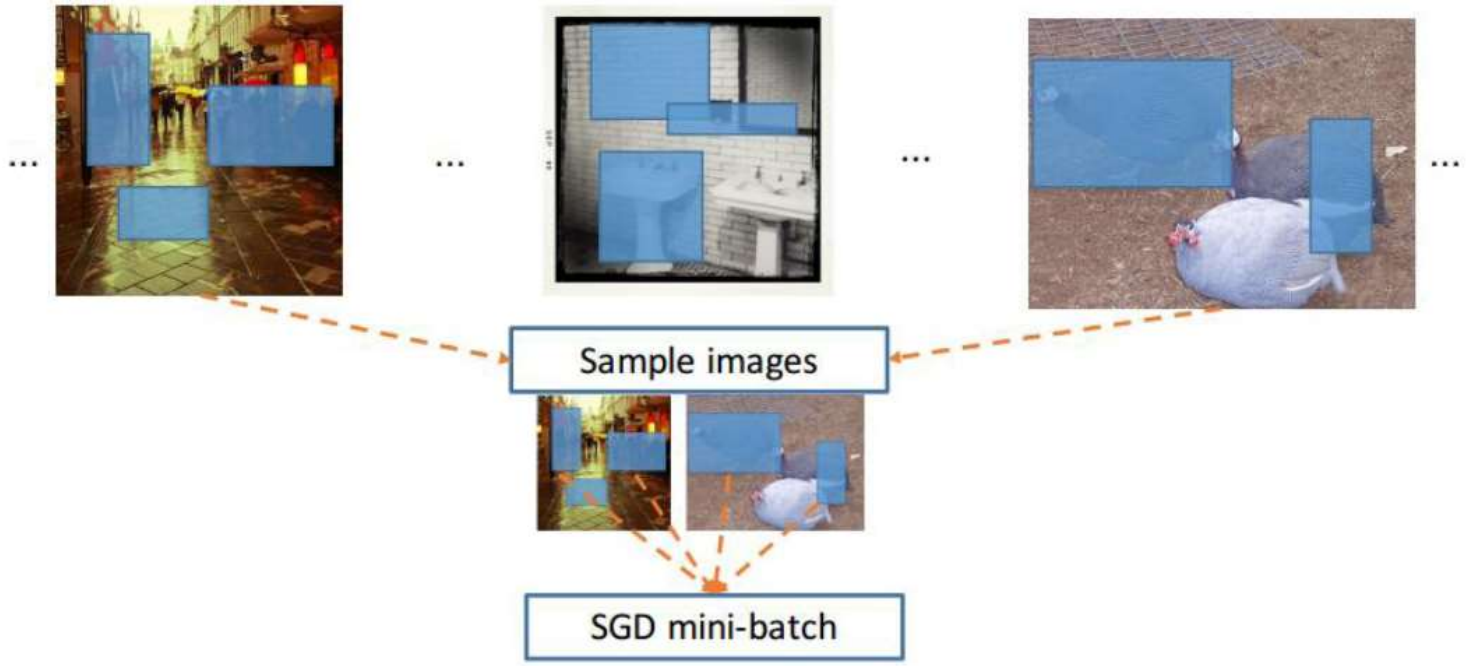
$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

$$t^u = (t_{\text{x}}^u, t_{\text{y}}^u, t_{\text{w}}^u, t_{\text{h}}^u)$$

Обучение Fast R-CNN

Чтобы использовать общие признаки:

- В минибатч включаем мало изображений (2)
- Сэмплируем много гипотез для каждого изображения (64)



Оценка Fast R-CNN

	R-CNN	SPP-net	Fast R-CNN
Training time	84h	25.5h	8.75h
Testing time per image	47s	2.3s	0.32s
Testing time with selective search	49s	4.3s	2.32s
VOC07 test mAP	66.0 %	63.1 %	68.1%

Осталась только зависимость от внешнего генератора гипотез!

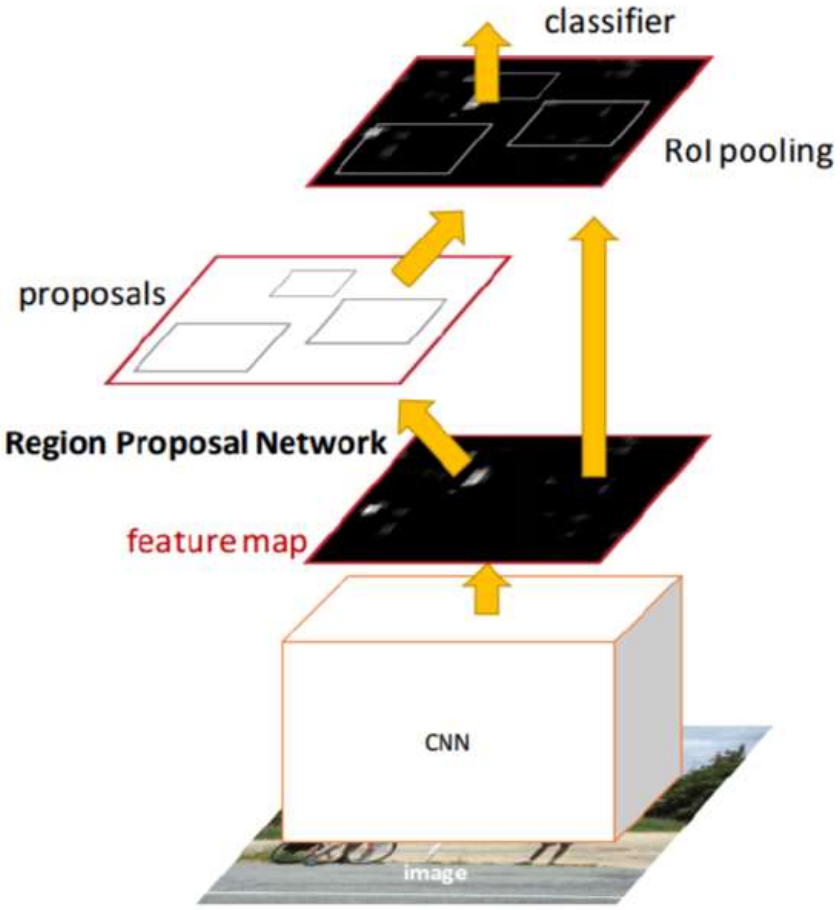
Faster R-CNN

Region Proposal Network

Faster R-CNN =

Fast R-CNN + RPN

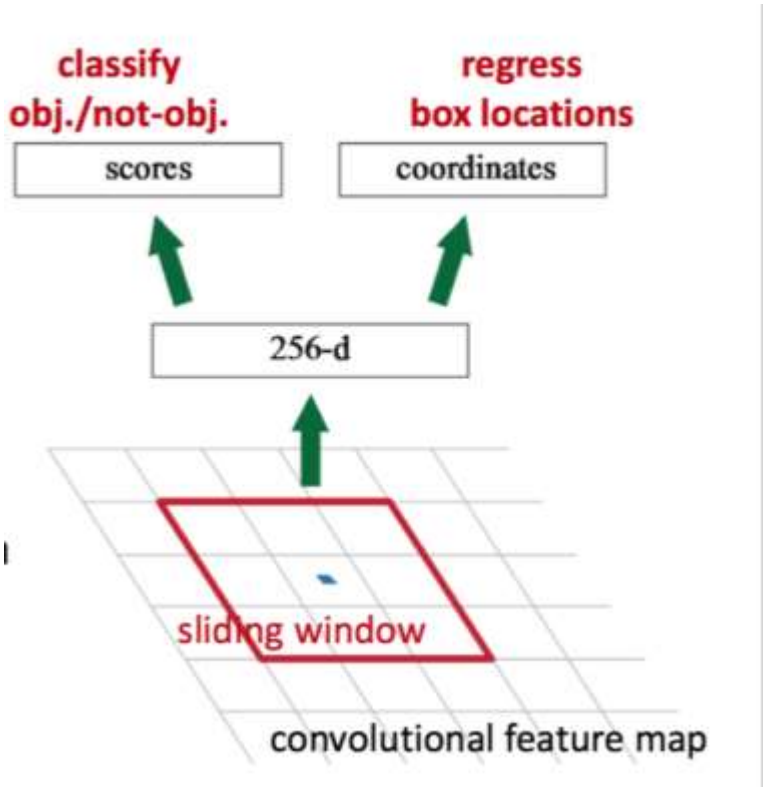
- Устраняется зависимость от внешнего алгоритма генерации гипотез
- Детекция объектов осуществляется в один проход единственной сетью



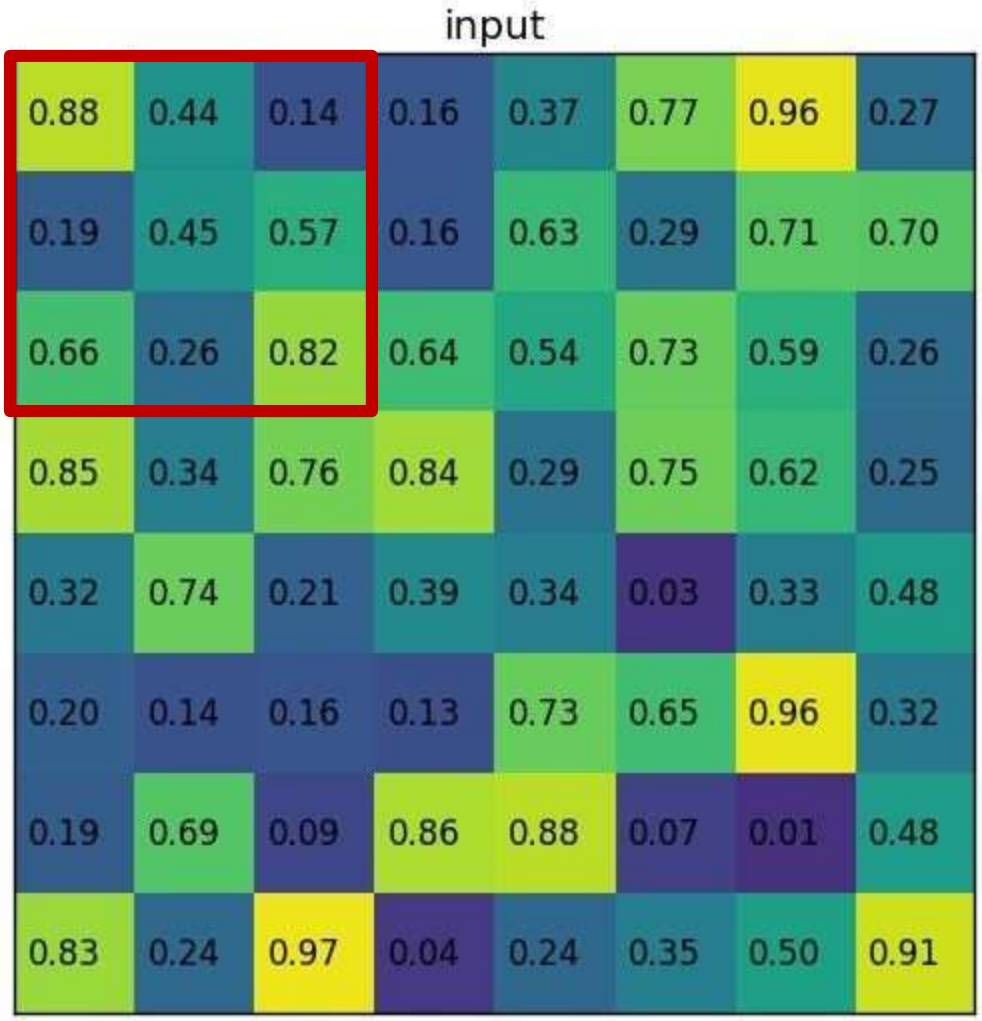
S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

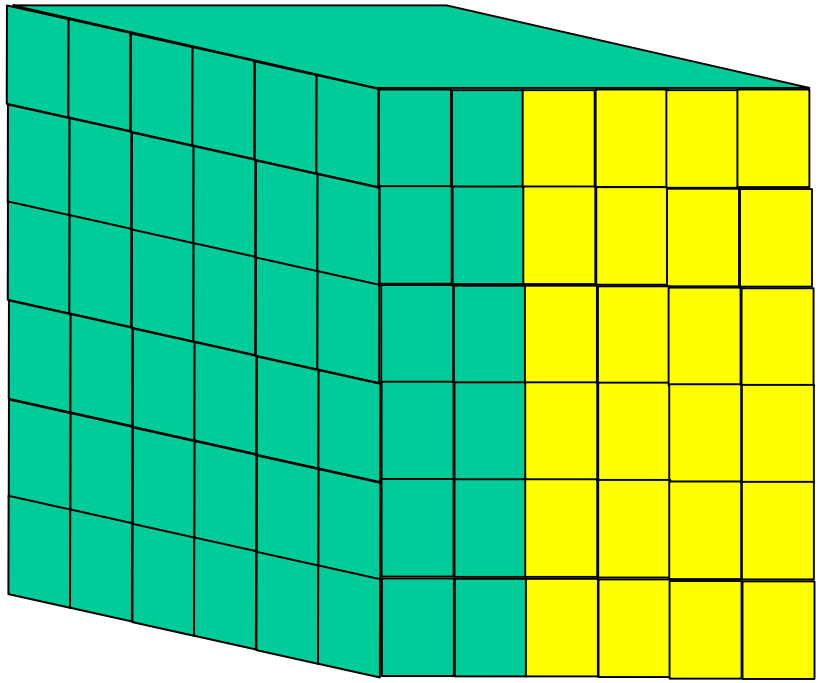
Region Proposal Network (RPN)

- Маленькое скользящее окно по карте признаков (feature map)
- Маленькая нейросеть для:
  - Классификации объект/не объект
  - Регрессии bbox
- Позиция окна показывает локализацию объекта относительно изображения
- Регрессия bbox показывает положение bbox относительно положения скользящего окна



Визуализация RPN для детектора лиц



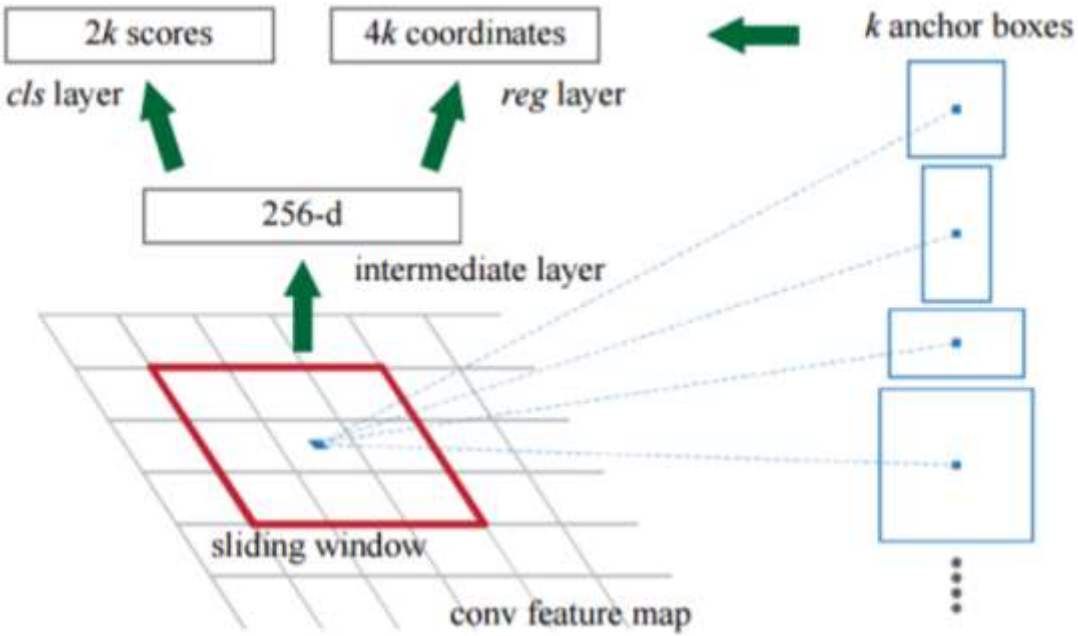


Каков размер выхода?

Якоря в RPN

В каждом месте:

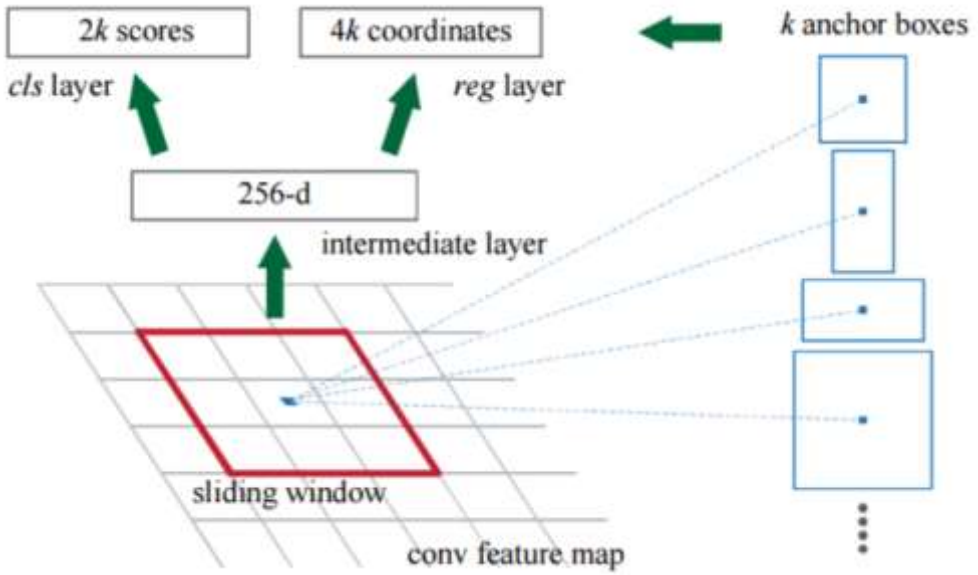
- Определяем набор из k гипотез
- У каждой гипотезы свой размер и пропорция



- Такие гипотезы называем “якоря”
- Якоря позволяют улучшить обнаружение объектов разного размера и пропорций (как обычно в схеме скользящего окна)

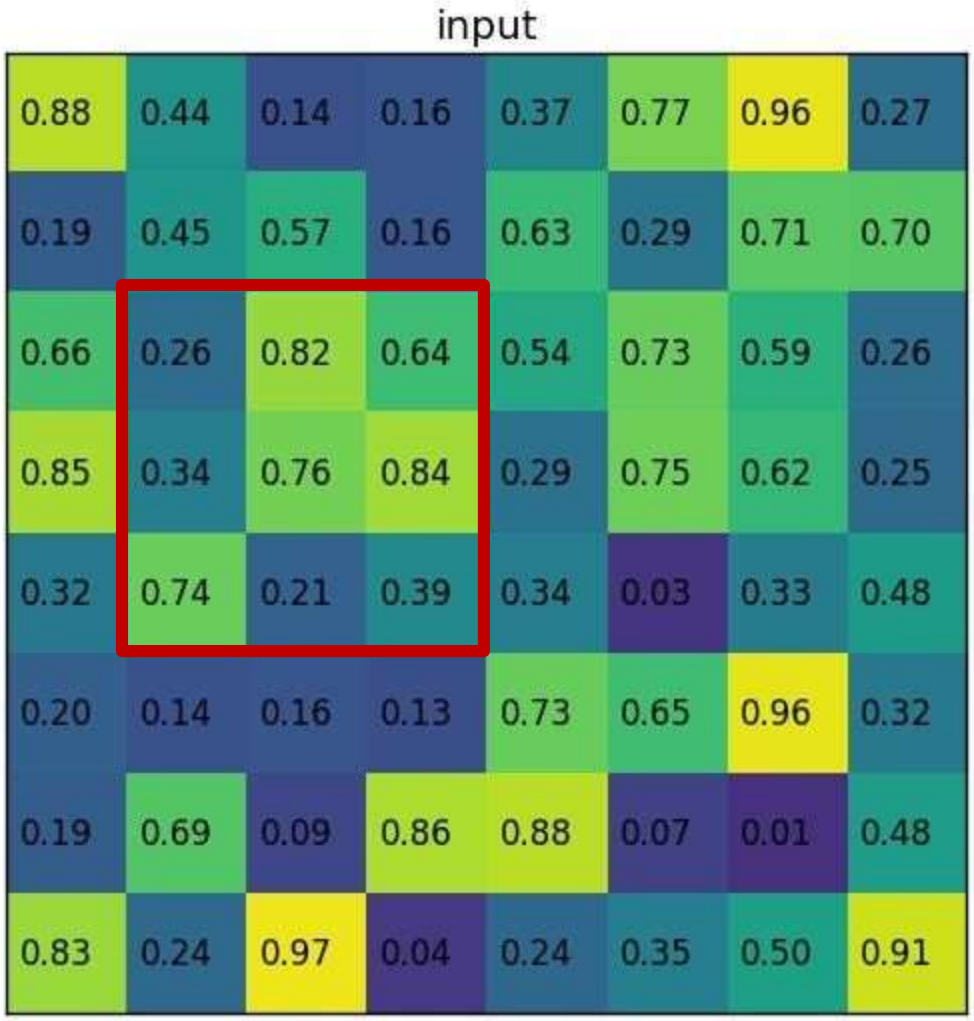
Region Proposal Network (RPN)

- Регрессия bbox относительно якоря
  - regressing an anchor box to a ground truth box



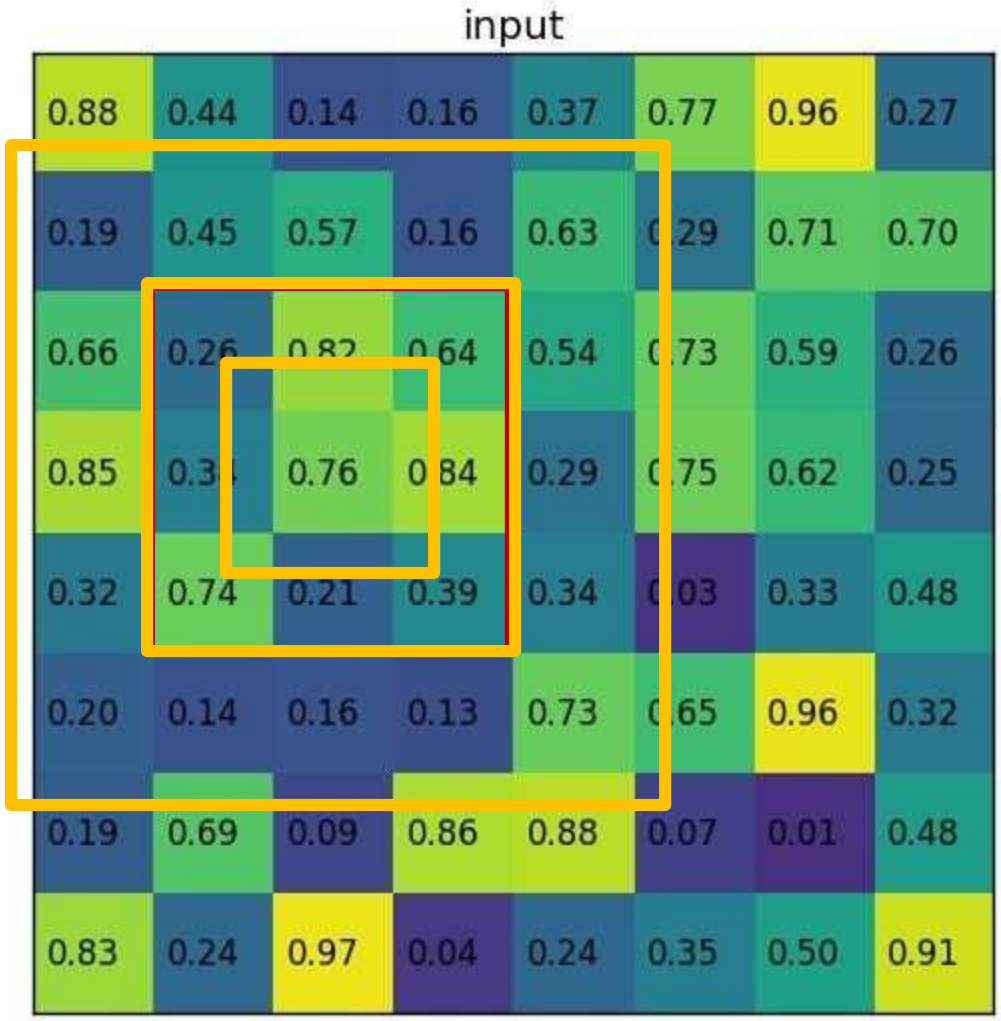
- Вероятность объекта определяется относительно якоря:
  - Якоря – положительные примеры, если IoU > 0.7
  - Якорь – отрицательный пример, если IoU < 0.3
- Для картинки 1000x600 и сети VGG-16 у нас будет 60x40 положений скользящего окна x 9 якорей = ~21500 гипотез

Визуализация якорей



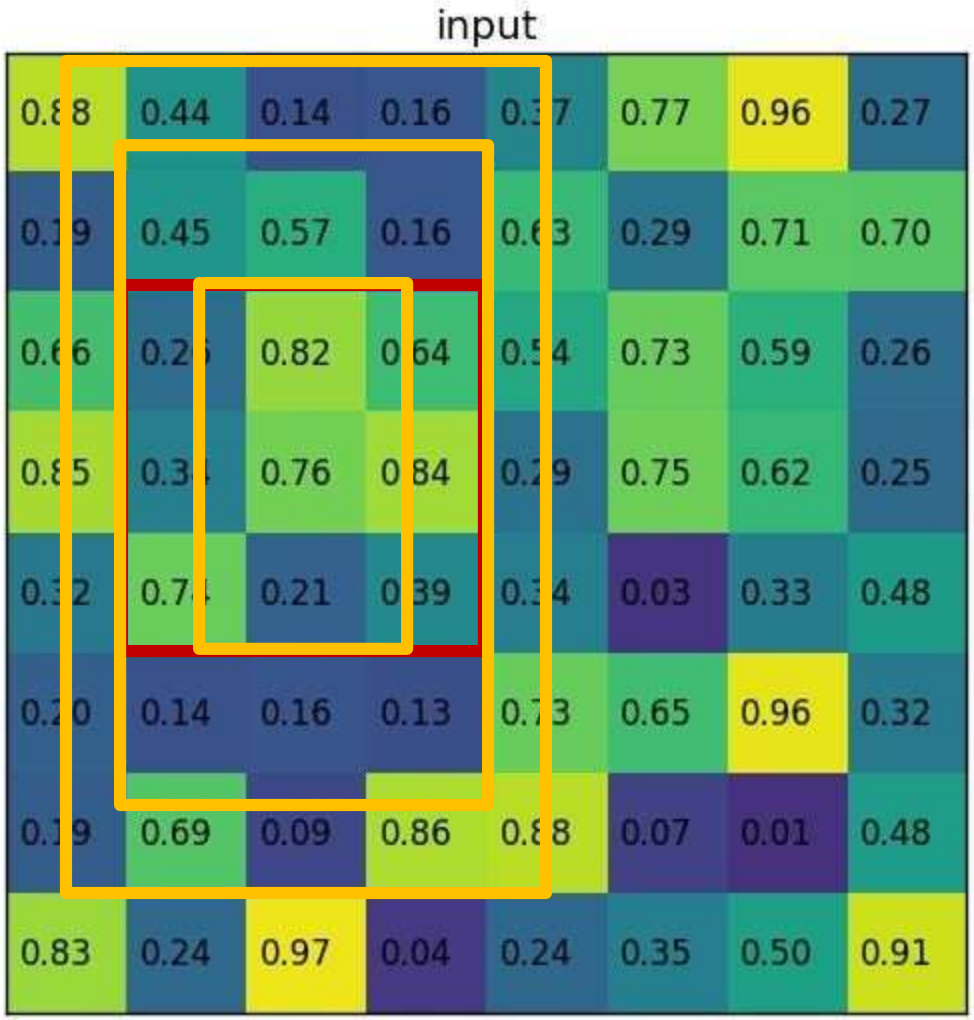
Применяем к этом области RPN

Визуализация якорей



Первые 3 якоря (гипотезы bbox)

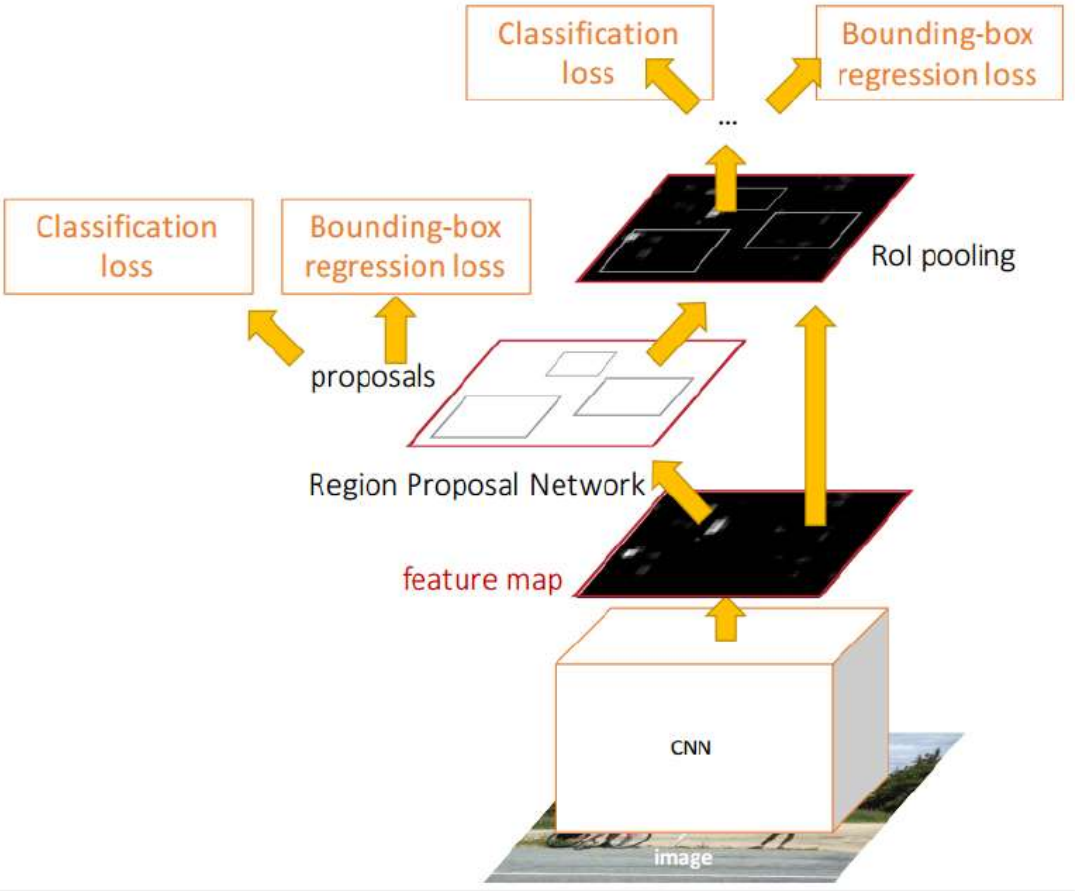
Визуализация якорей (+3 штуки)



Вторые 3 якоря (гипотезы bbox)

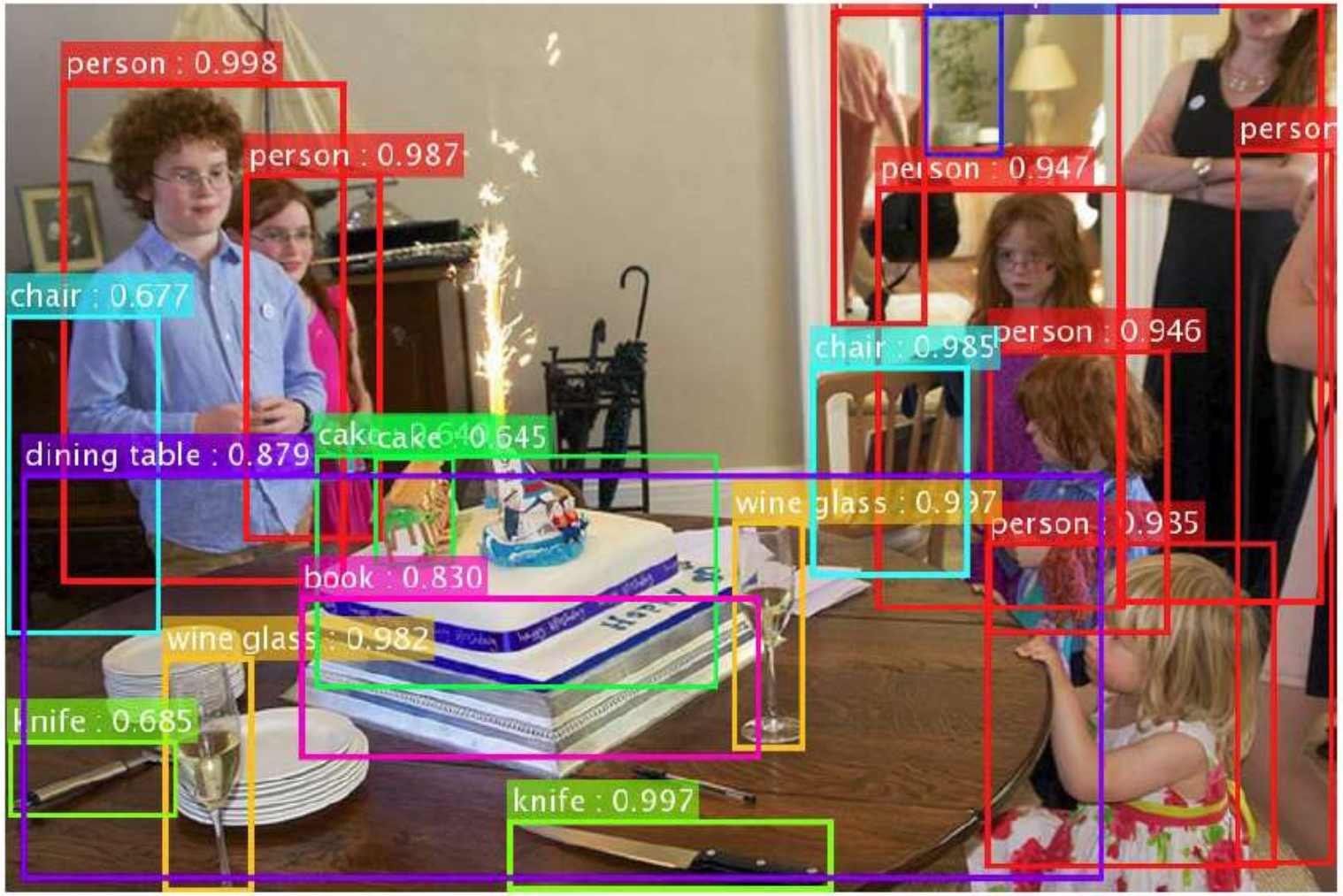
Faster R-CNN

- Одна сеть, четыре выхода



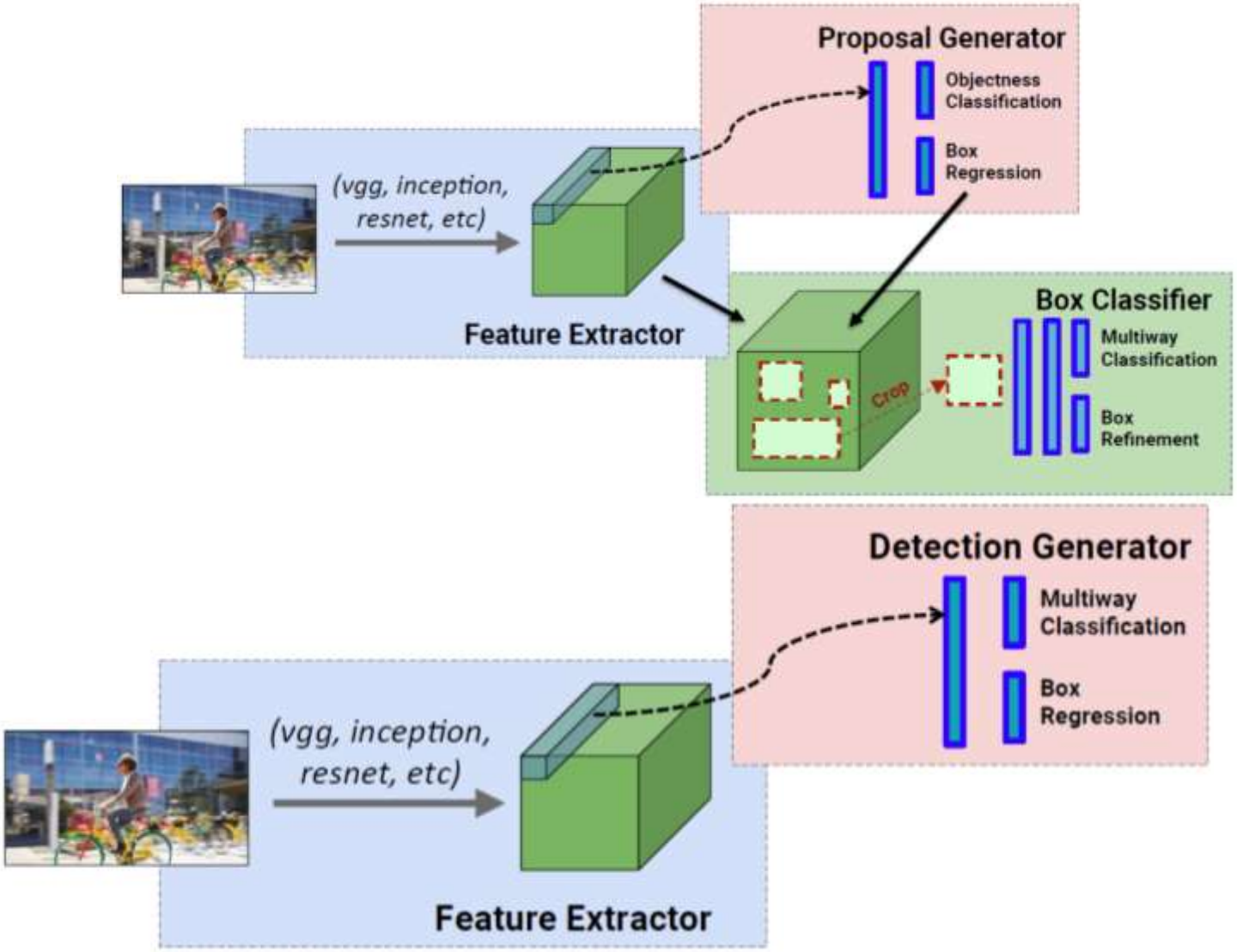
Обучаем всё совместно в multi-task режиме

Пример результатов



Our results on COCO – too many objects, let's check carefully!

R-CNN -> Single Shot Detectors

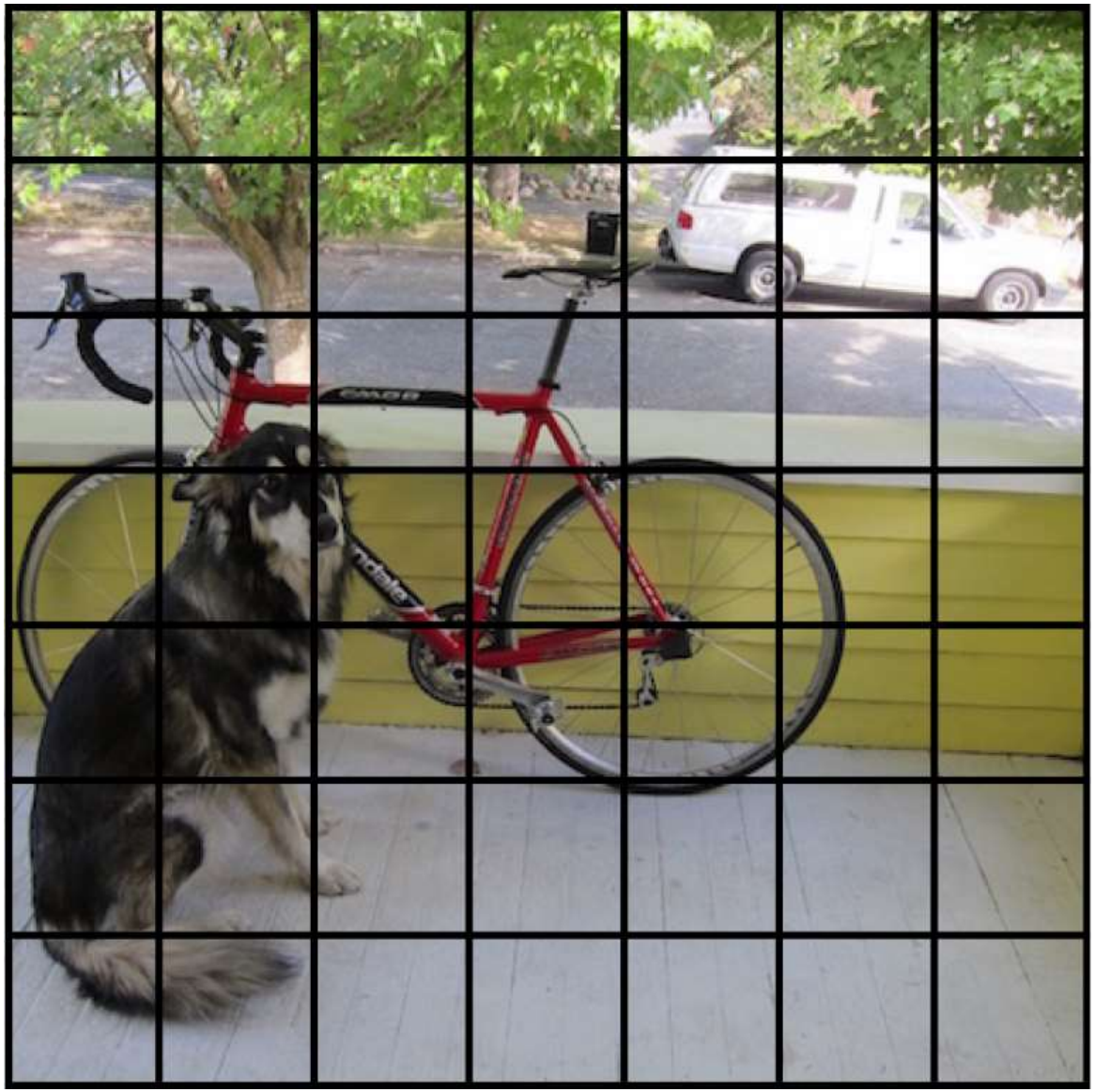


Source: J. Huang et. al. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv:1611.10012

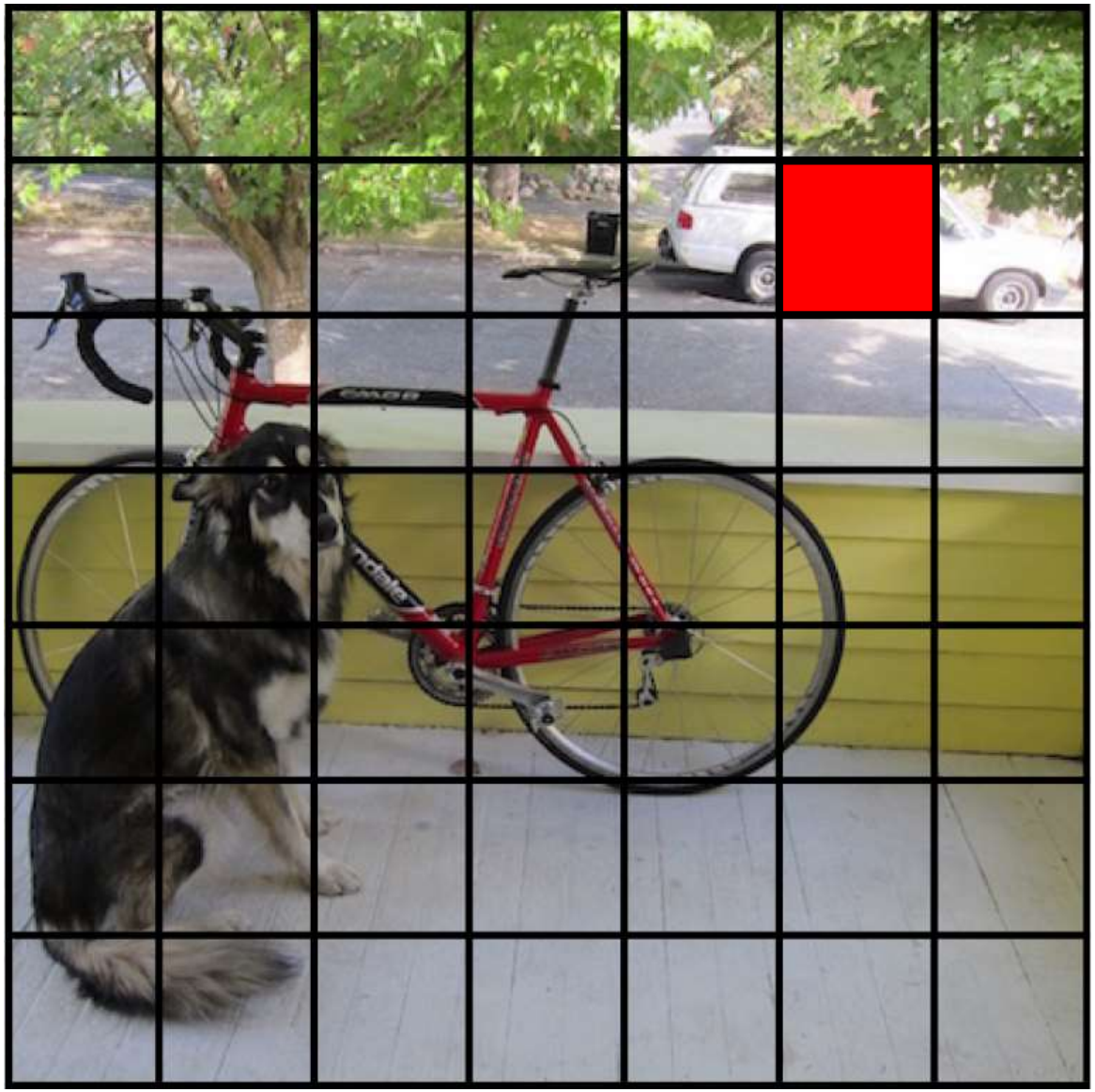
Детектор YOLO

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.

Разбиваем изображение сеткой

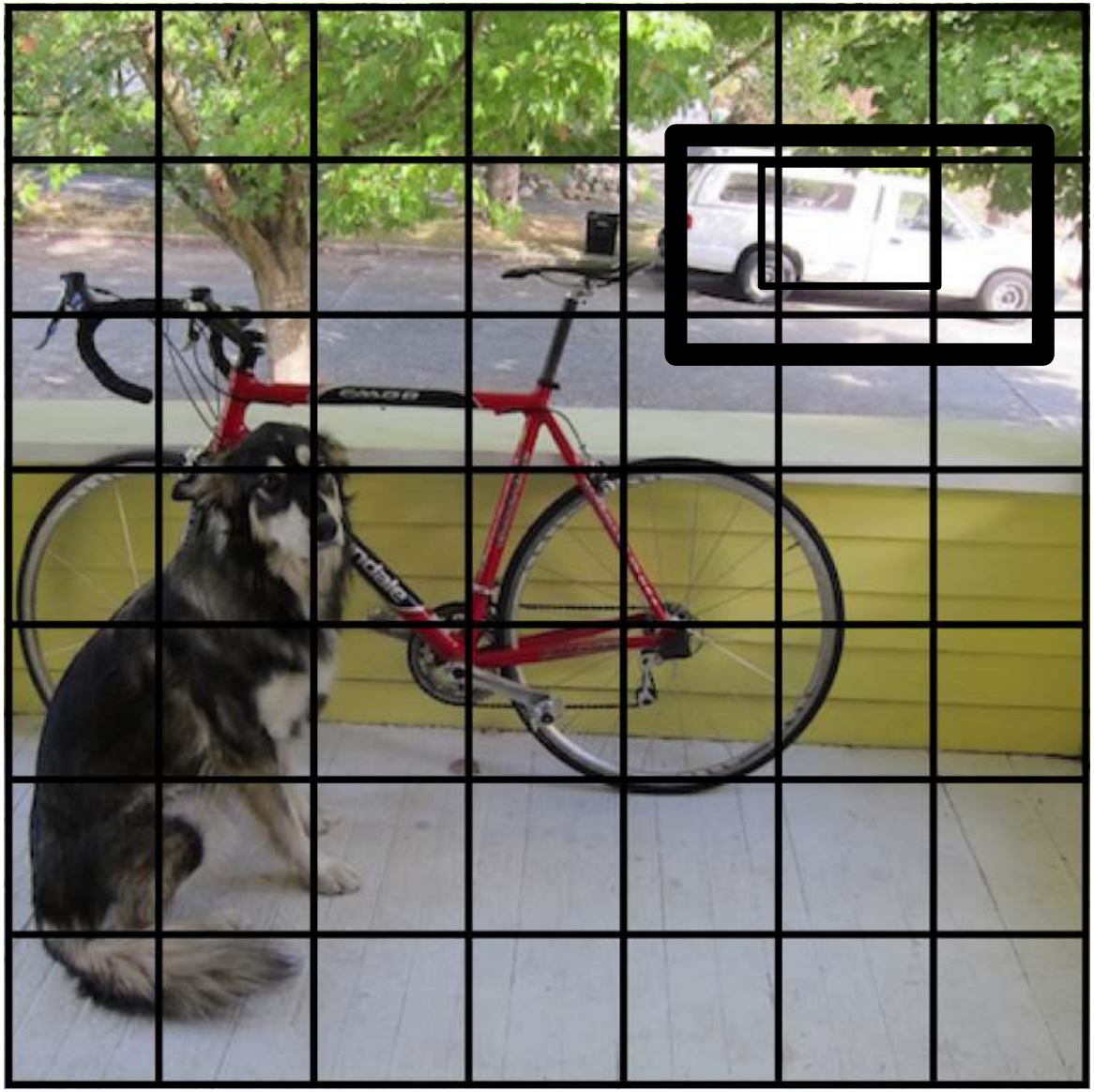


Каждая секта предсказывает bbox и P(Object)

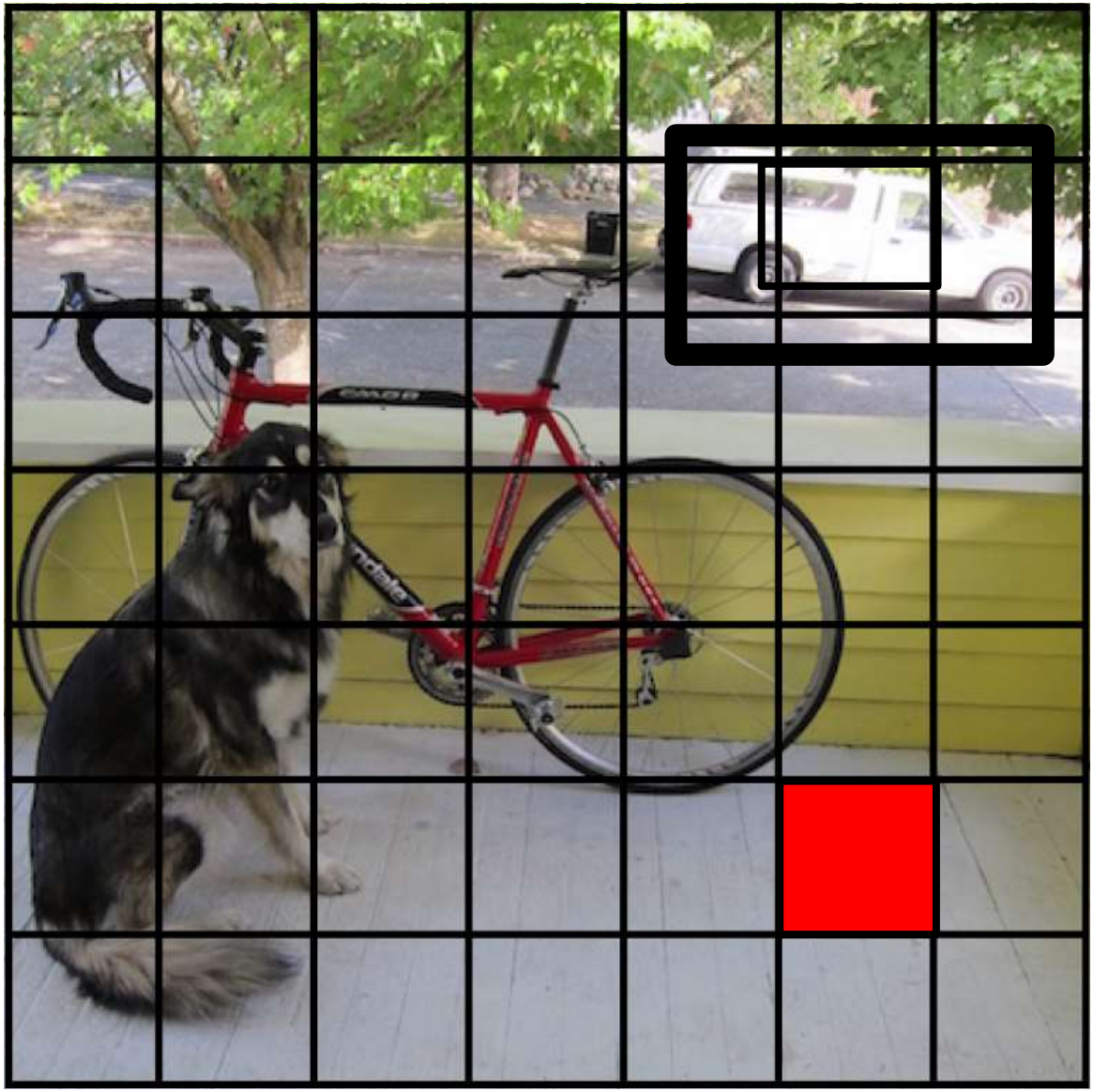


Каждая секта предсказывает bbox и P(Object)

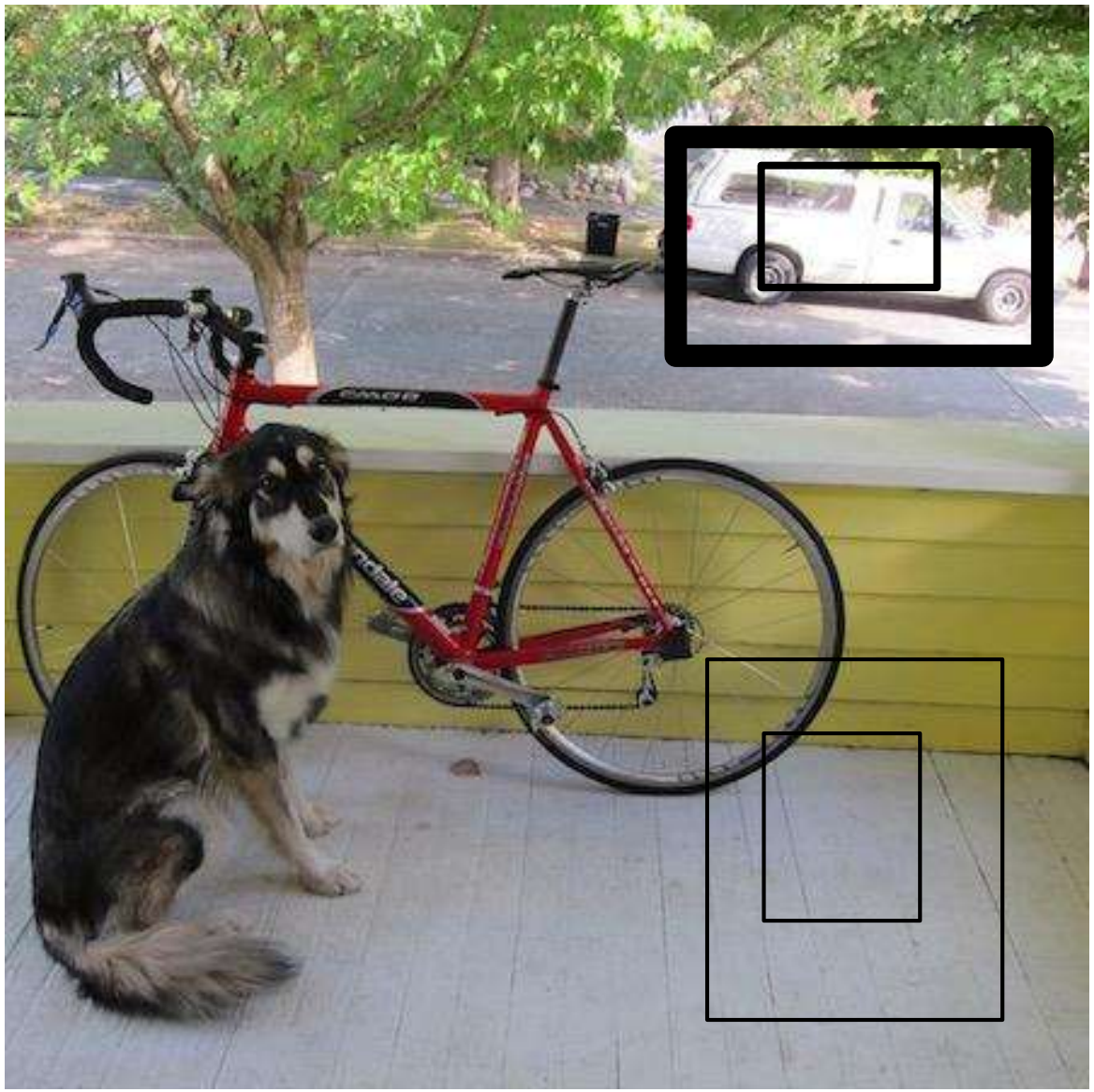
Каждая секта предсказывает bbox и P(Object)



Каждая секта предсказывает bbox и P(Object)

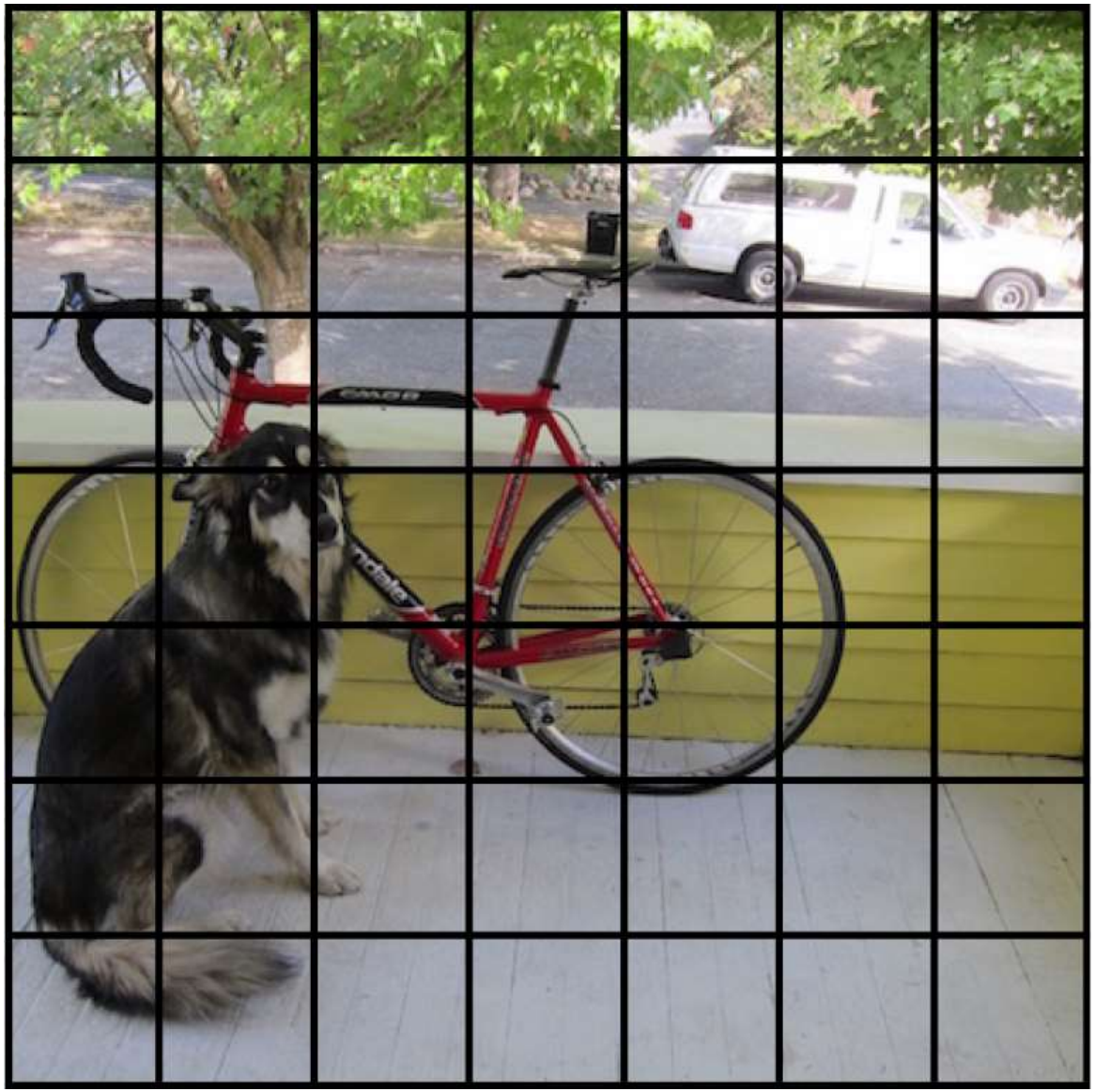


Каждая секта предсказывает bbox и P(Object)

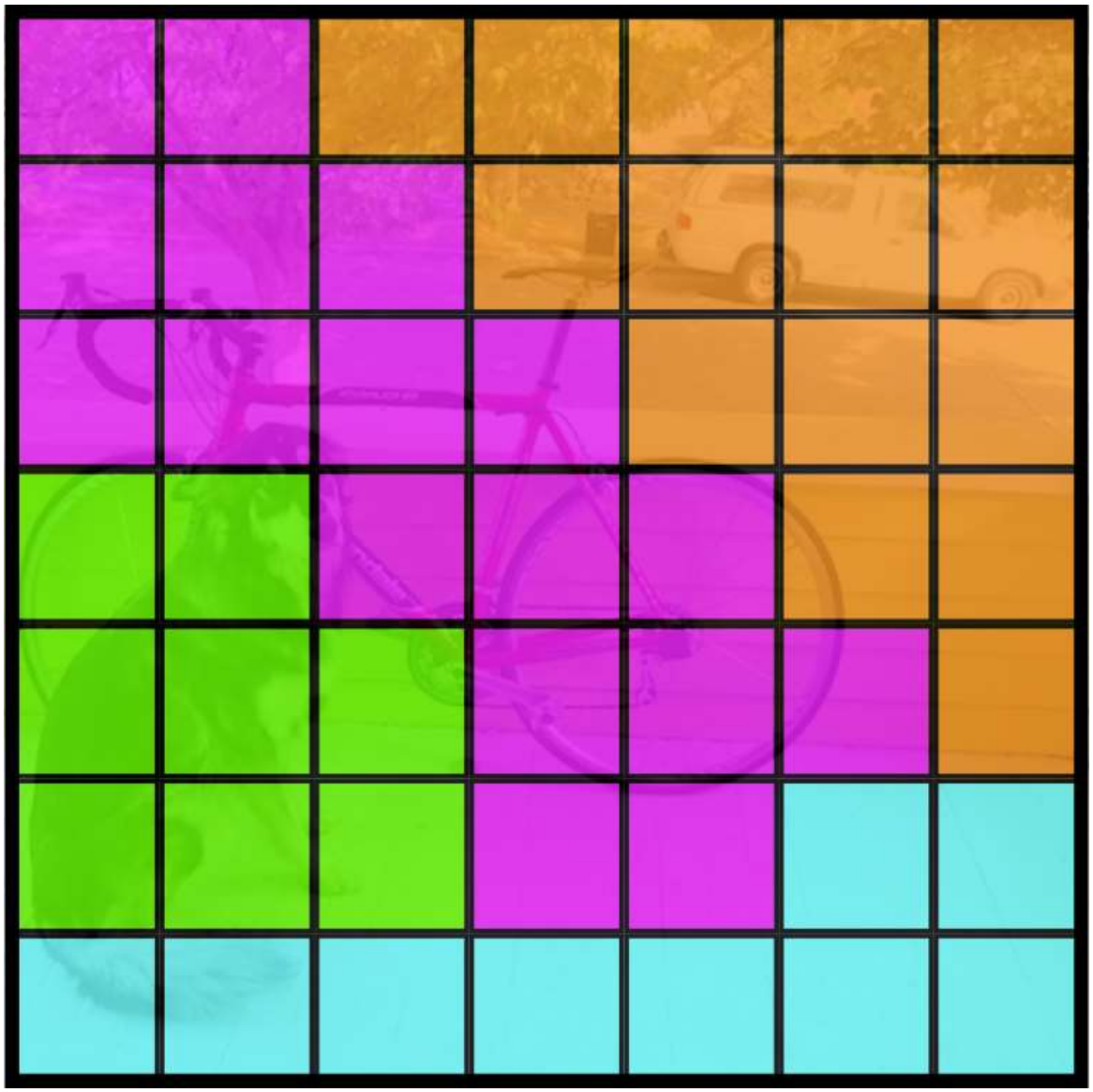


Каждая секта предсказывает bbox и P(Object)

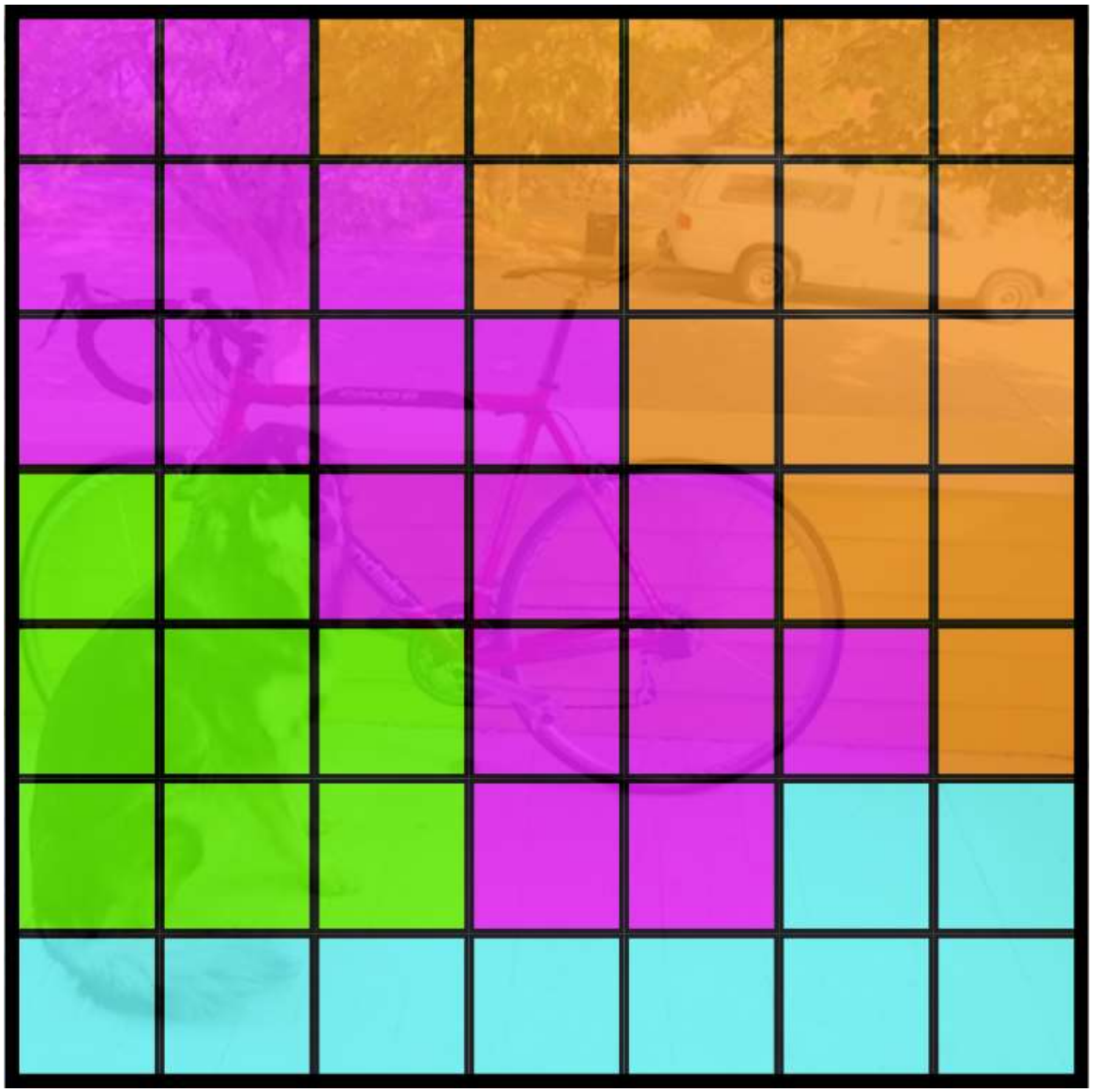
Каждая клетка также предсказывает класс



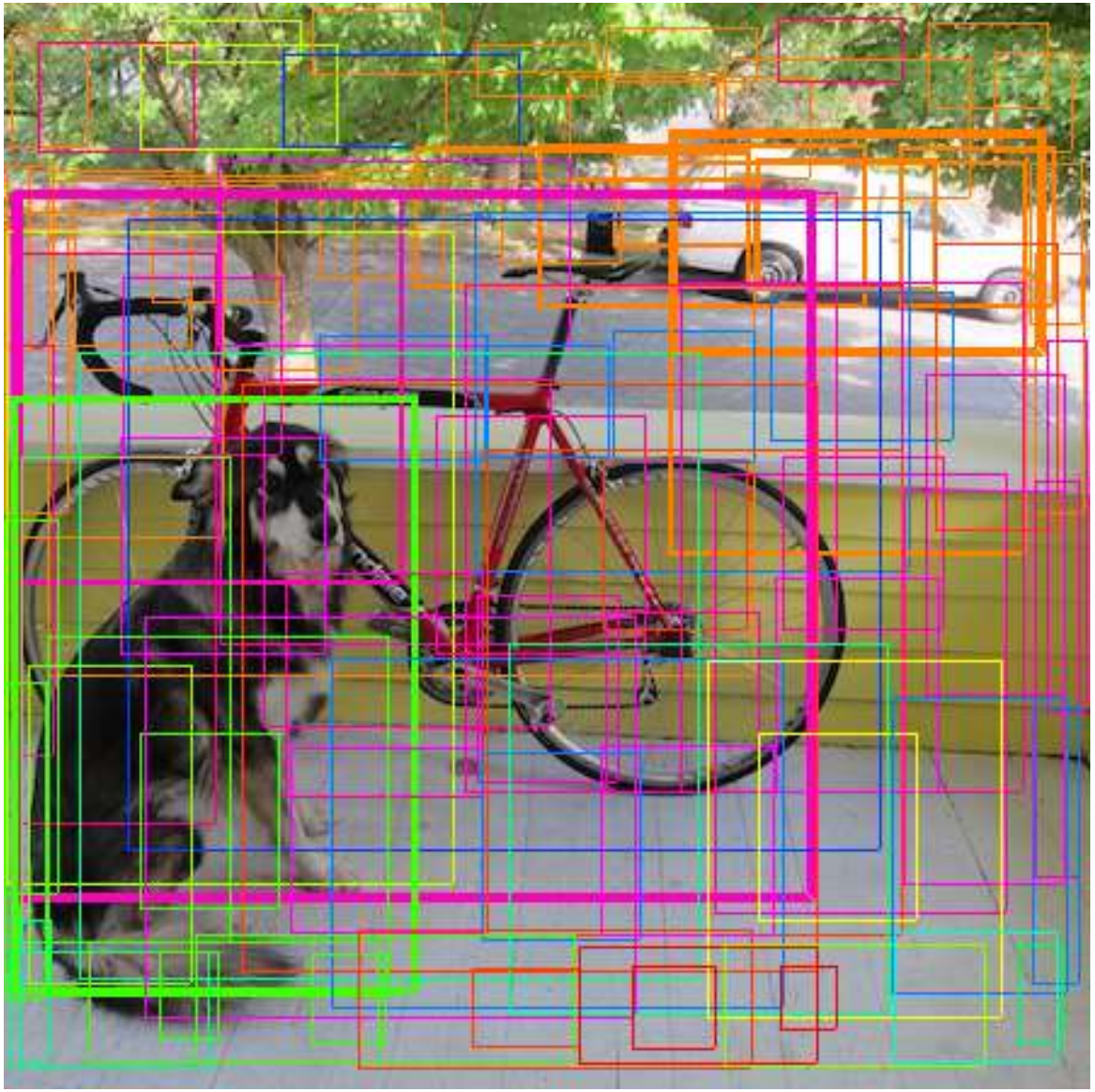
Каждая клетка также предсказывает класс



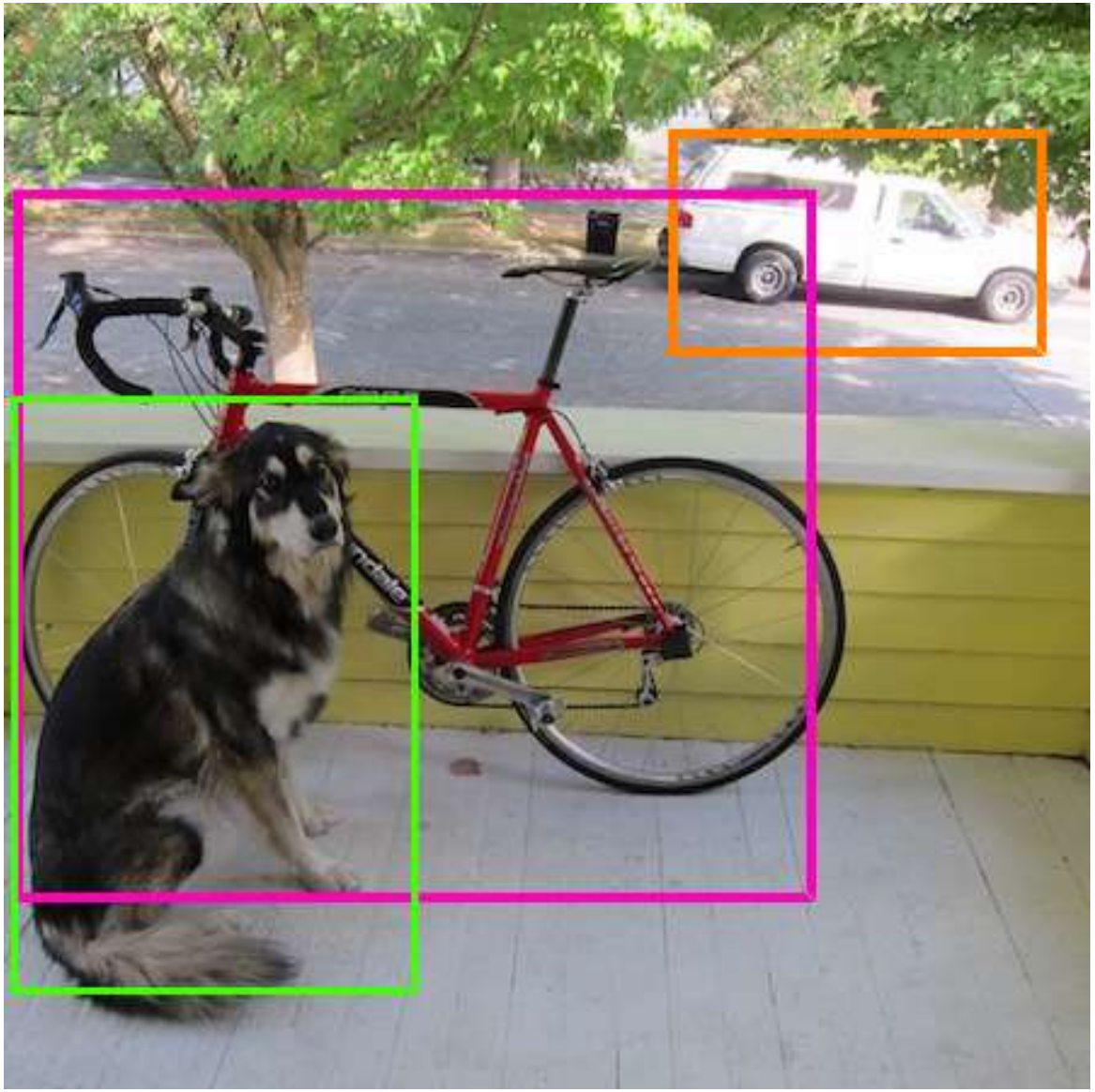
При условии объекта P(Car | Object)



Объединяем bbox и классы



Применяем MNS и порог на P(Object)



Параметризация определяет выход

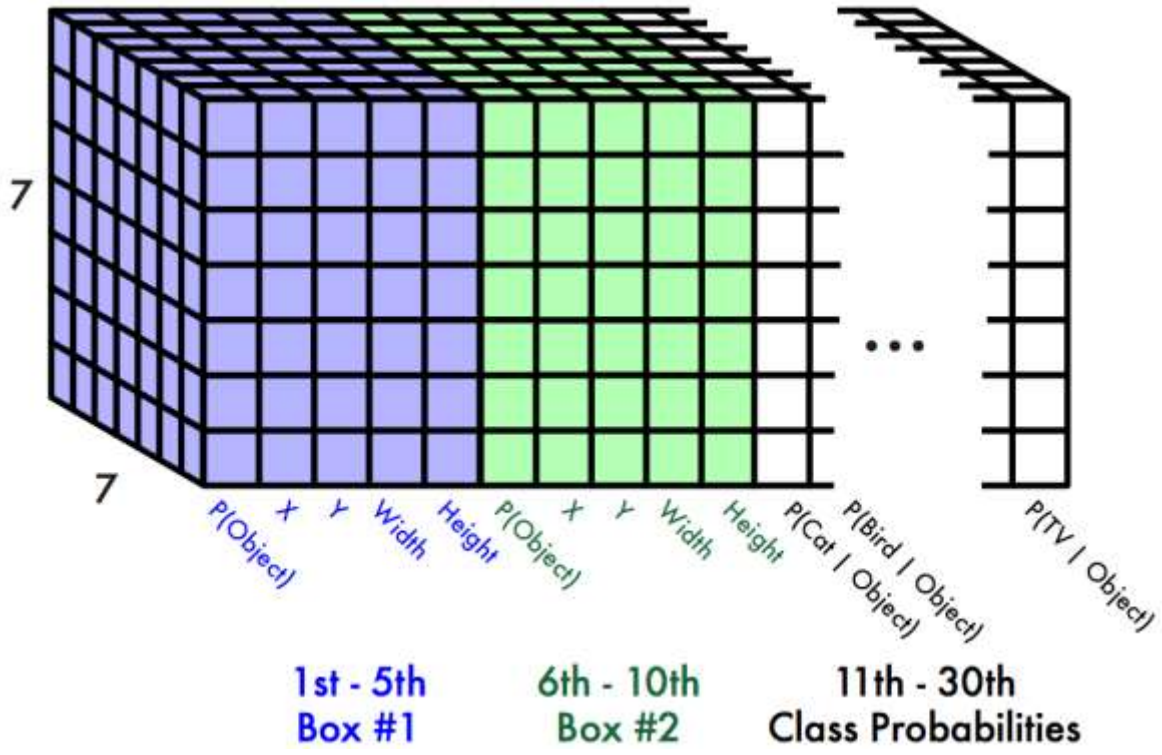
Each cell predicts:

- For each bounding box:
  - 4 coordinates (x, y, w, h
  - 1 confidence value
- Some number of class probabilities

For Pascal VOC:

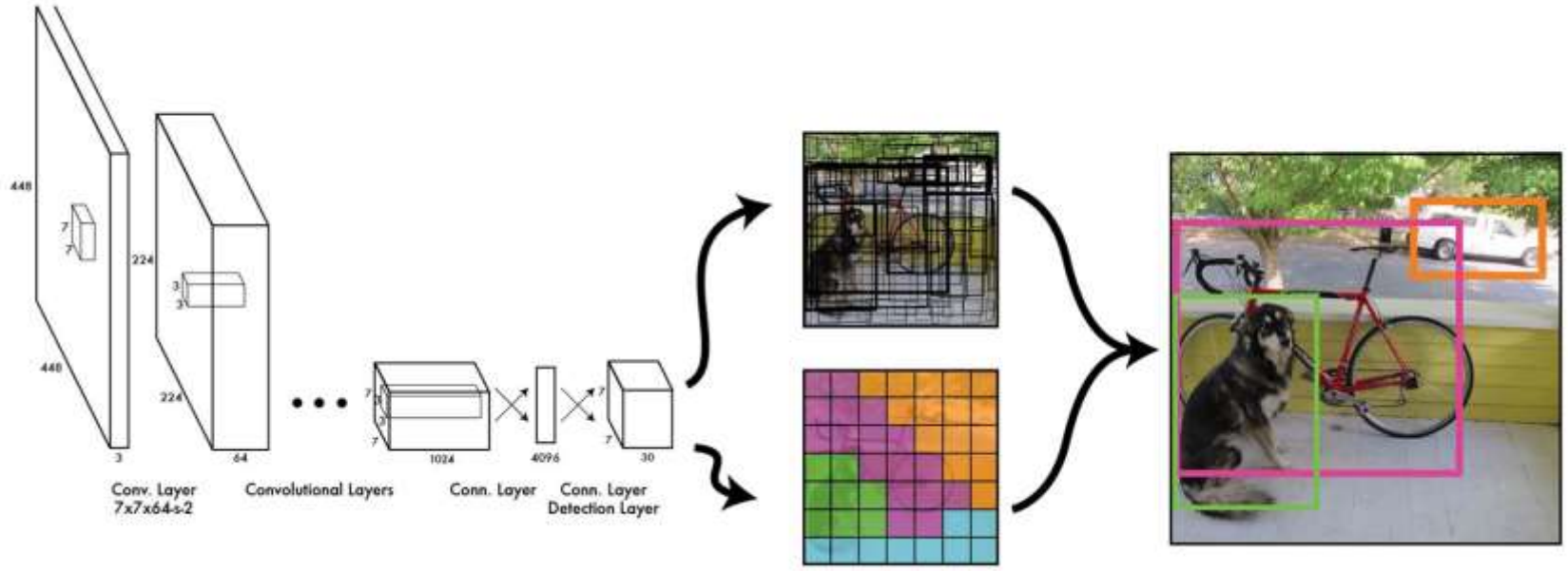
- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

7 x 7 x (2 x 5 + 20) = 7 x 7 x 30

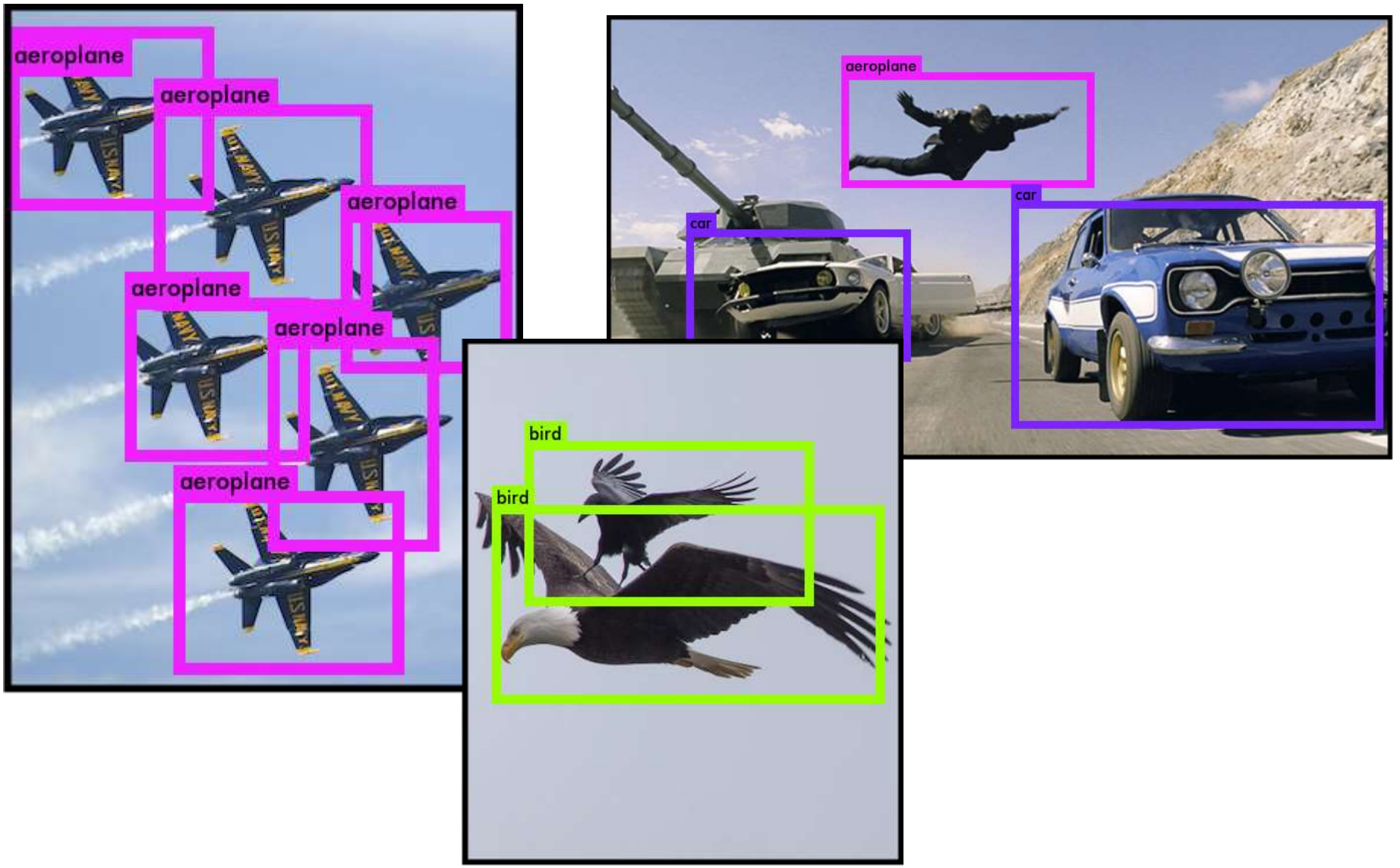


Общая схема

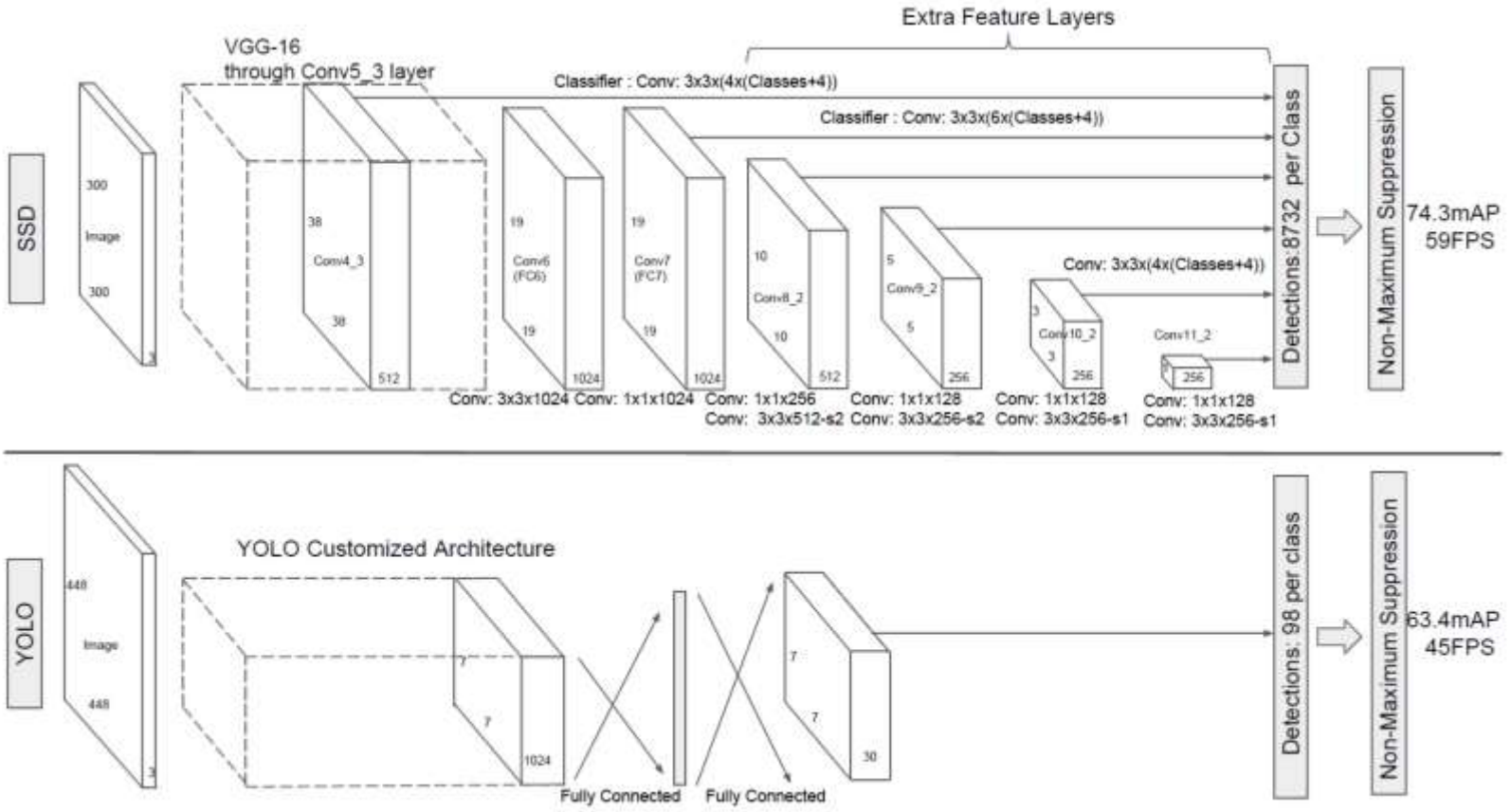
- Одностадийный детектор
- Однай нейросеть задаёт весь детектор
- Очень разреженное скользящее окно + уточнения bbox за счет регрессии



Примеры работы YOLO от авторов



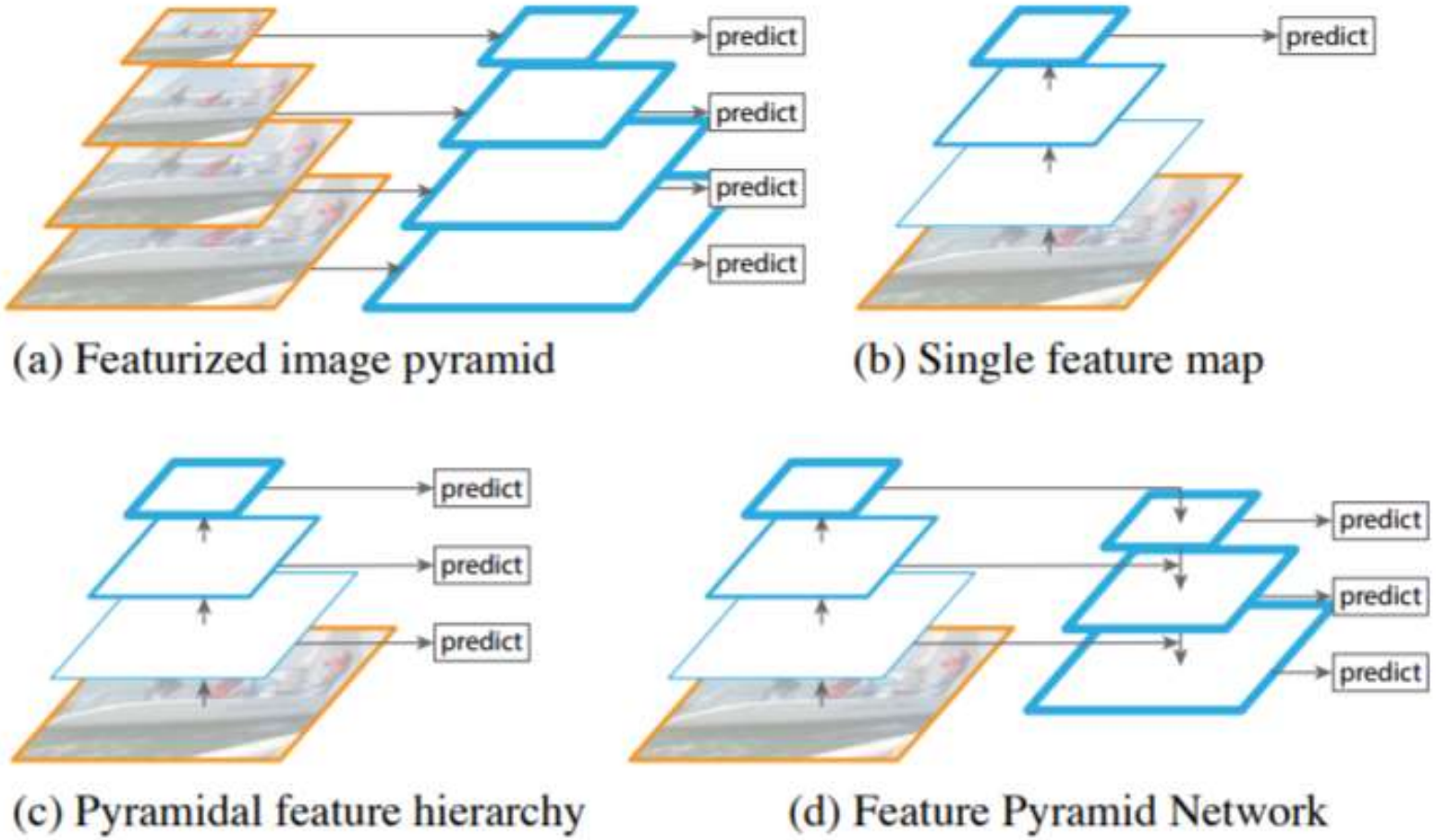
Single Shot Detector (SSD)



- Любая базовая архитектура для признаков
- Fully-convolutional + Multi-Scale Predictions

Source: W. Liu et.al.. Ssd: Single shot multibox detector. arXiv preprint arXiv:1512.02325, 2015

Feature pyramid network (FPN)



(a) Featurized image pyramid

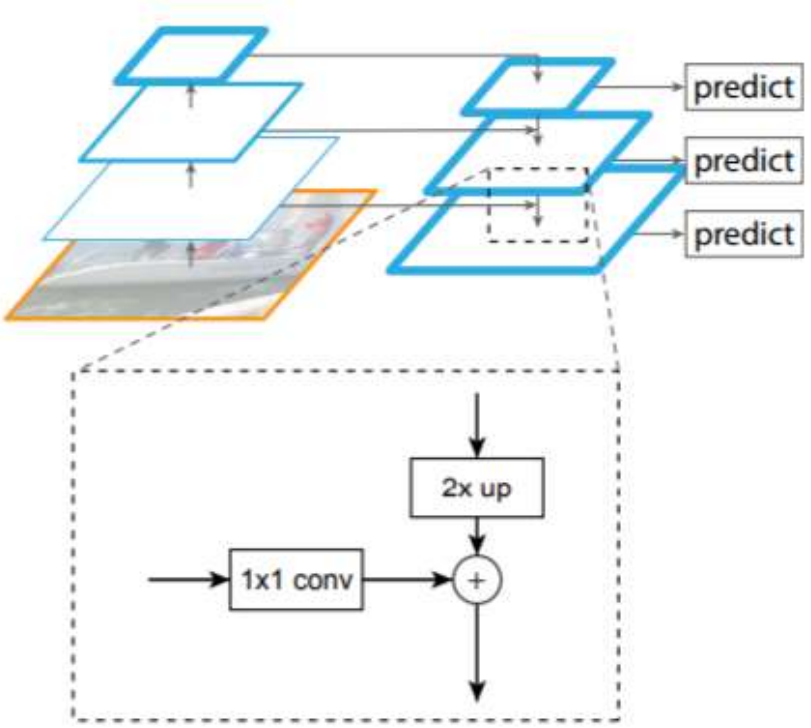
(b) Single feature map

(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network

Мощный способ для многомасштабного обнаружения

Построение FPN



- Пусть глубина каждой карты признаков 256
- На каждом уровне мы складываем увеличенную в 2 раза карту меньшего разрешения с преобразованными признаками из feature extractor
- После объединения можем пройтись фильтрами 3x3
- Никаких нелинейностей при построении FPN

RetinaNet = SSD + FPN + FocalLoss

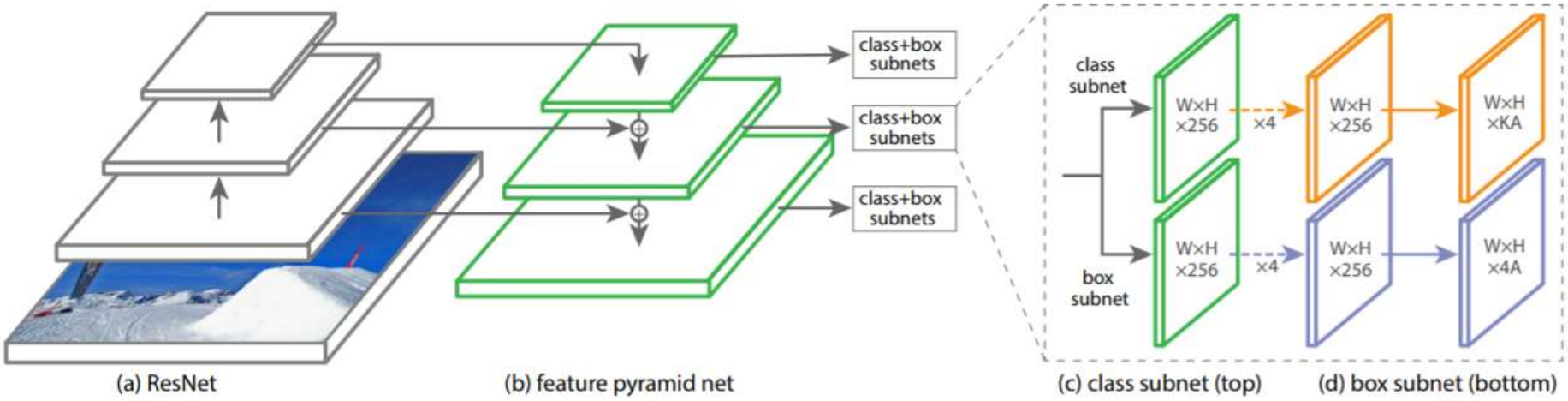


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [19] backbone on top of a feedforward ResNet architecture [15] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [19] while running at faster speeds.

Классификатор объектов строим как полносвёрточную сеть, из нескольких (4) слоёв свёрток

Focal Loss

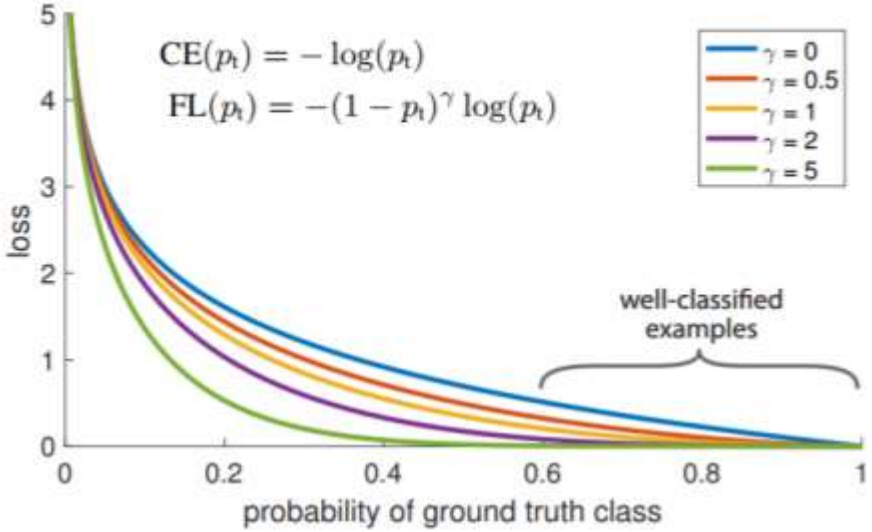
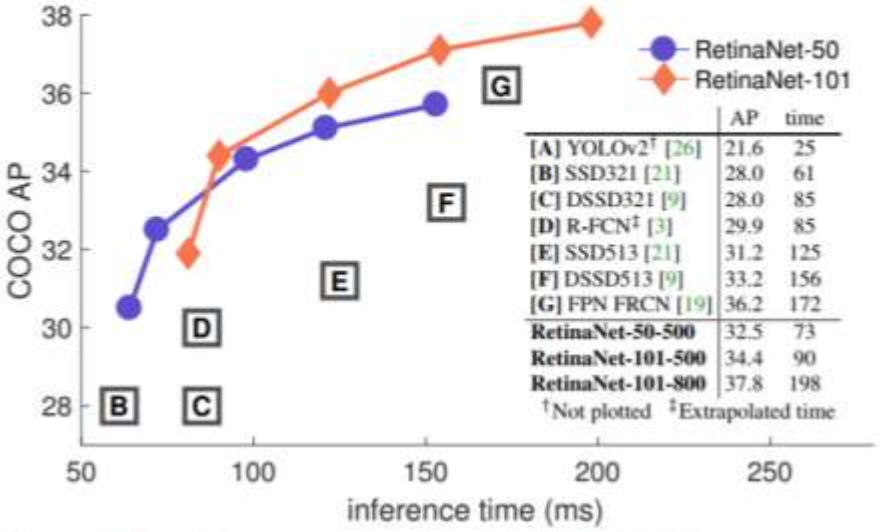


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.



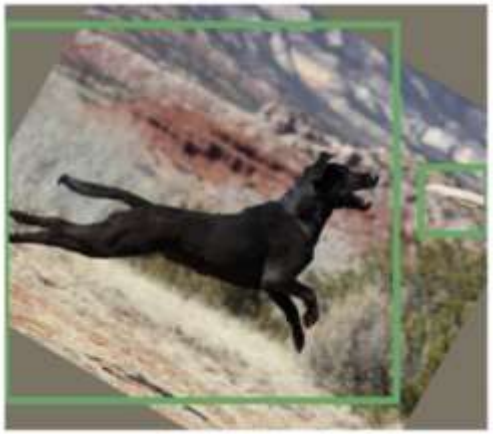
	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [27] system from [19]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and a variant trained for longer (not shown) achieves 39.1 AP. Details are given in §5.

Альтернатива hard-negative mining

Что развивают сейчас?



https://arxiv.org/abs/1906.11172v1

Backbone	Baseline	Our result	Difference
ResNet-50	36.7	39.0	+2.3
ResNet-101	38.8	40.4	+1.6
ResNet-200	39.9	42.1	+2.2

Table 1: **Improvements with learned augmentation policy across different ResNet backbones.** All results employ RetinaNet detector [24] on the COCO dataset [25].

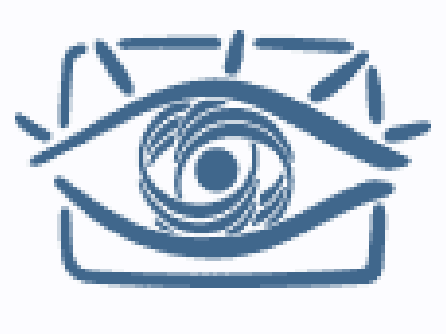
Method	mAP
baseline	36.7
baseline + DropBlock [13]	38.4
Augmentation policy with color operations	37.5
+ geometric operations	38.6
+ bbox-only operations	**39.0**

Table 2: **Improvements in object detection with learned augmentation policy.** All results employ RetinaNet detector with ResNet-50 backbone [24] on COCO dataset [25]. DropBlock shows gain in performance employing a state-of-the-art regularization method [13].

- Хорошие и обучаемые аугментации
- Общие улучшения архитектур и процедур обучения
- Новые функции потерь
- Более общая задача – panoptic segmentation

Резюме детекции объектов

- Скользящее окно позволяет свести задачу детекции к классификации
- Для работы с разными масштабами / пропорциями рассматриваем гипотезы (окна) разных размеров и пропорций
- На один объект получаем множество откликов, поэтому приходится подавлять «слабые» (NMS)
- ROI-pooling и варианты позволяют считать «скользящие окна» по картам нейросетевых признаков
- Для работы с разными размерами объектов лучше собирать признаки с разных слоёв нейросети
- Ключевые методы – SSD, Faster R-CNN



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №8
«Сегментация изображений»

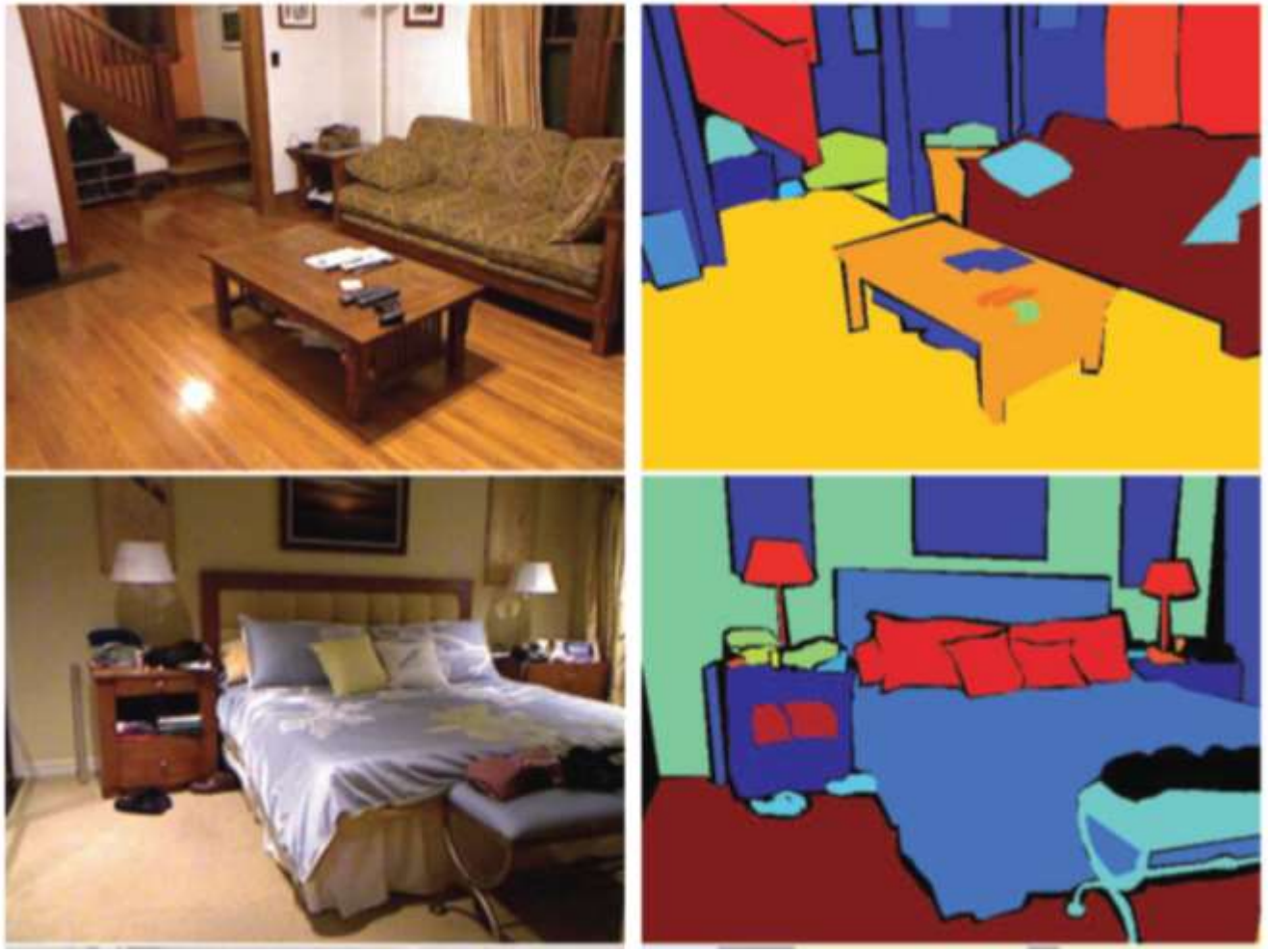
Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

11 ноября 2019 года

Задача сегментации

Разделение изображения на фрагменты(группы пикселов) по определённому критерию

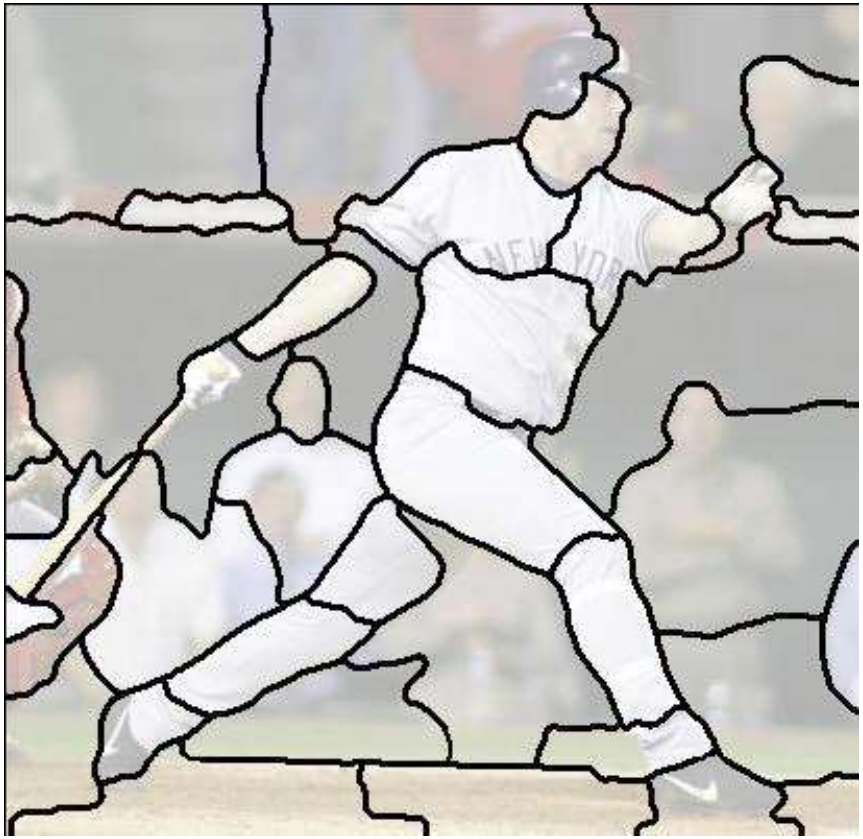


Извлечение объекта

Выделение конкретного произвольного объекта, указанного пользователем или по другому заданного

Сегментация без учителя

Разделение изображения на регионы, однородные по своим визуальным характеристикам, и отличающихся от соседних регионов



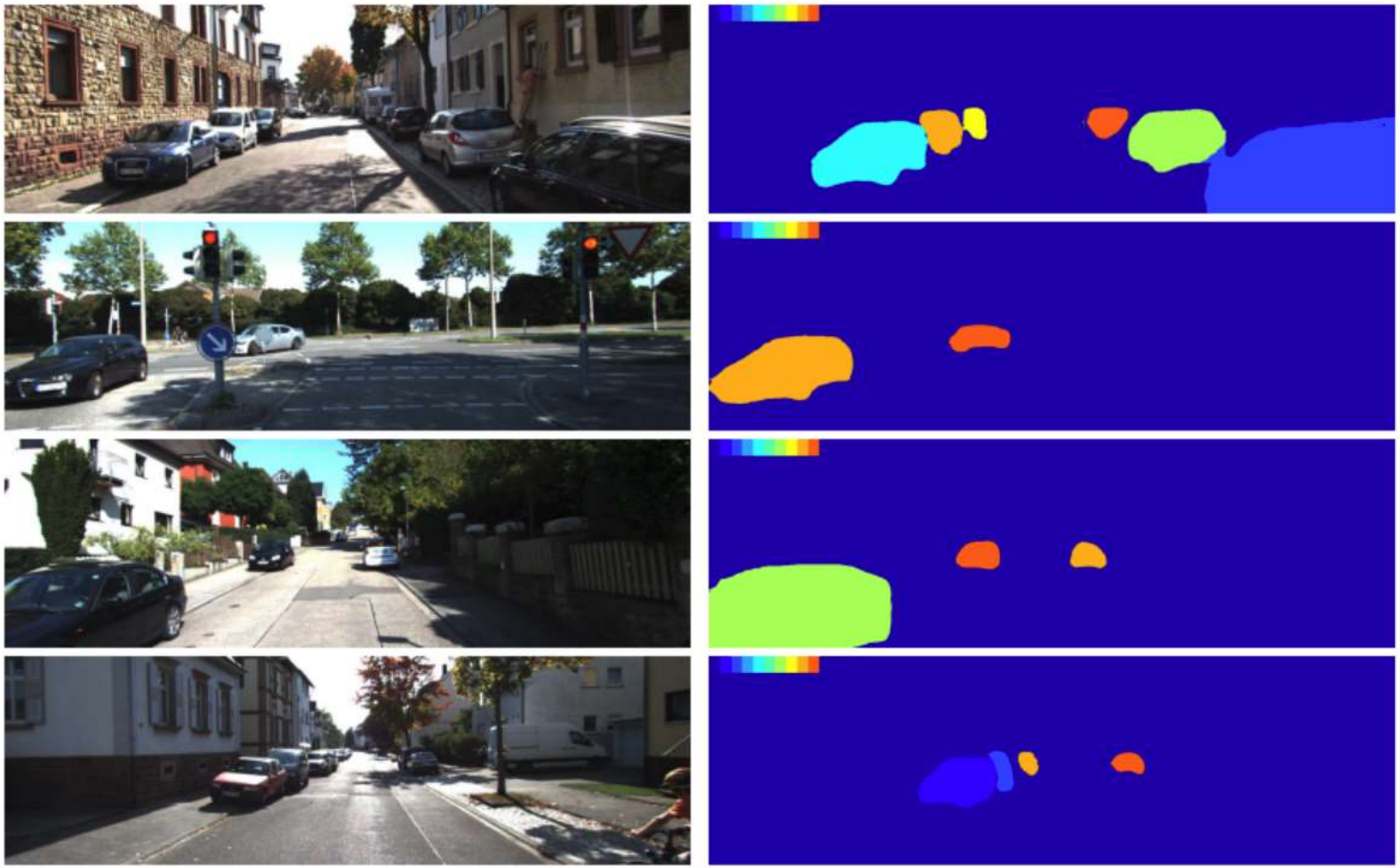
Семантическая сегментация

Попиксельная разметка изображения, где каждая метка соответствует определенному объекту

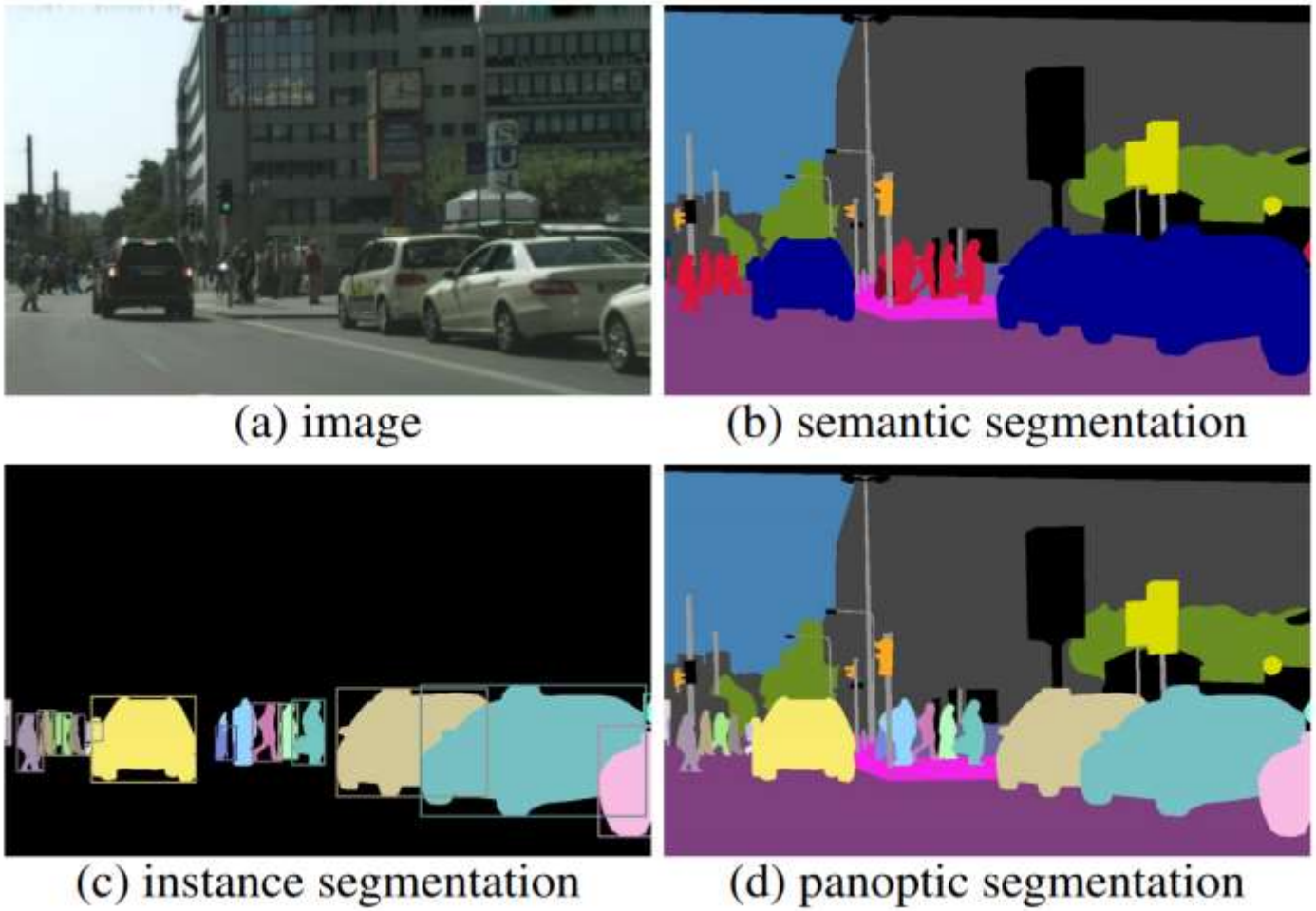
- Разные пиксели одного объекта существенно отличаются друг от друга по признакам (яркости, цвету, текстуре окрестности)
- Единственное, что у них общее – это «семантика»
- Поэтому задача сегментации объекта тесно связана с задачей распознавания

Instance segmentation

Выделение всех отдельных экземпляров объектов определённого класса. Каждый объект помечен своей меткой.



Panoptic segmentation

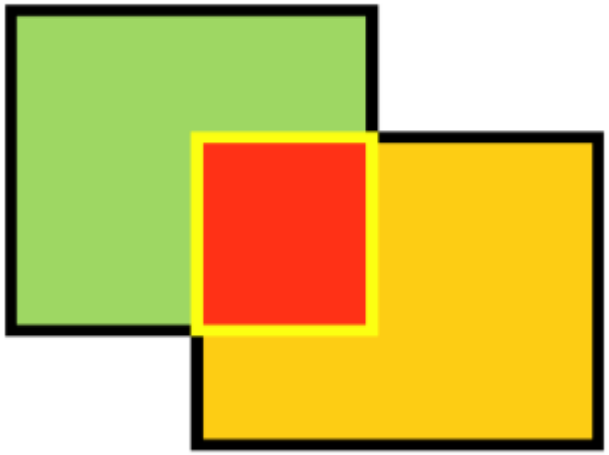


Объединение задача semantic и instance segmentation в единую модель

Source: https://arxiv.org/pdf/1801.00868.pdf

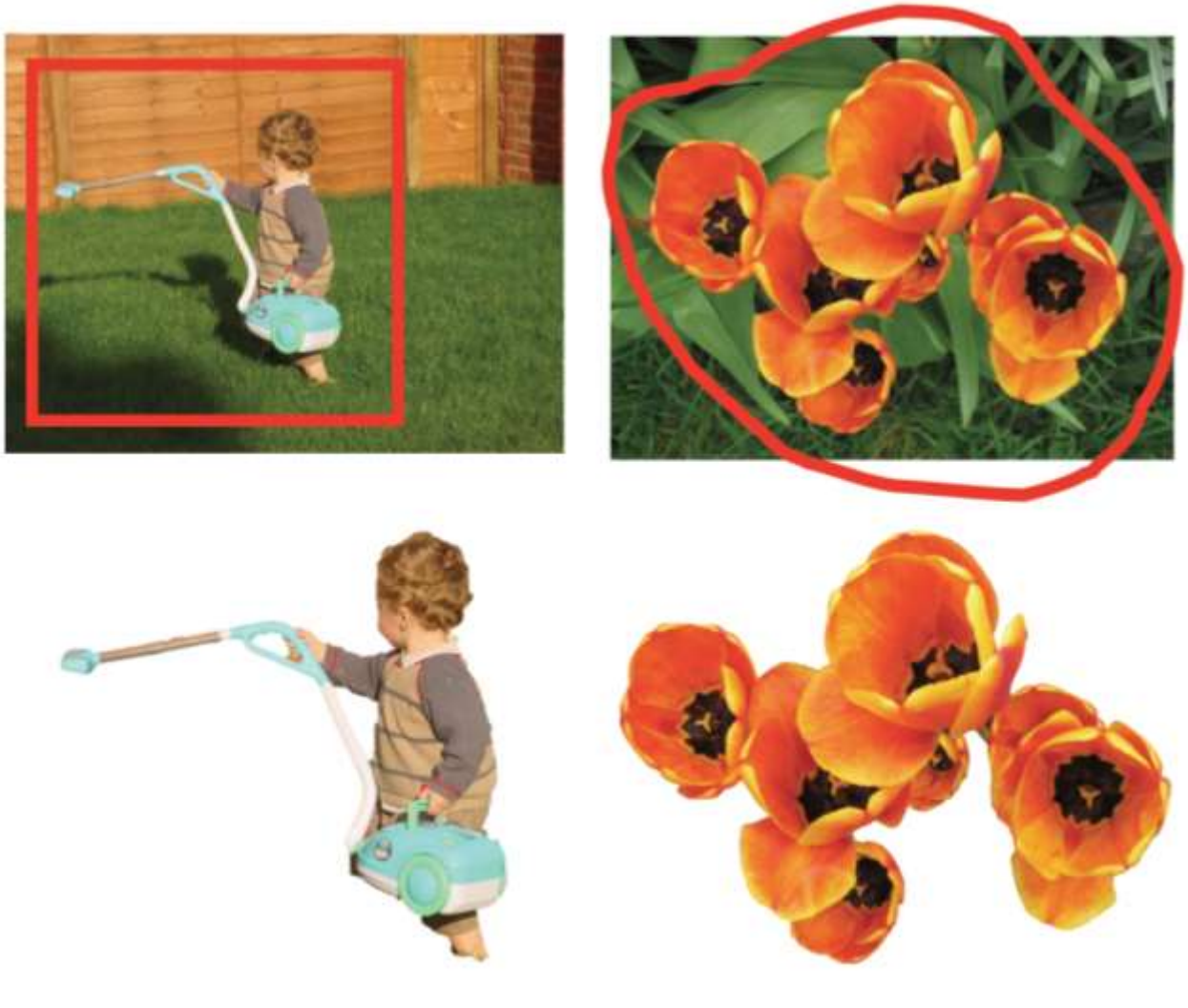
Оценка точности сегментации

- Intersection over Union



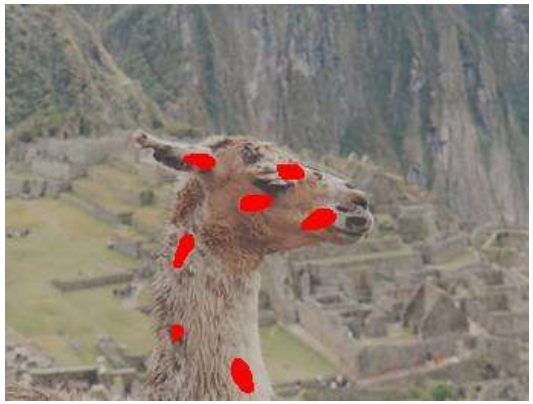
$$\text{segmentation accuracy} = \frac{\text{true positives}}{\text{true positives} + \text{false positives} + \text{false negatives}}$$

Интерактивная сегментация



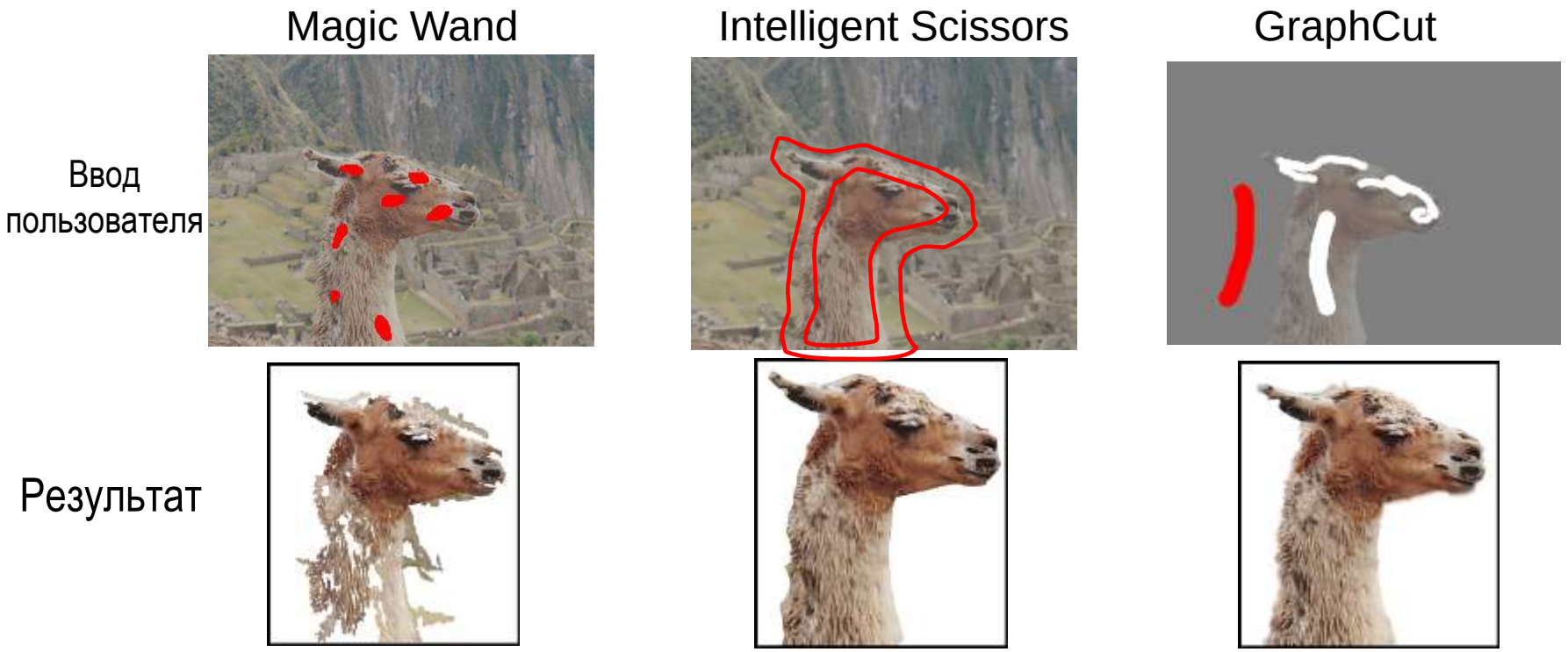
Самые первые подходы к решению

- MagicWand – цветовая сегментация по меткам пользователя
- Intelligent Scissors – проведение пути по области с высоким градиентом на основе динамического программирования через отмечаемые пользователем точки на границе

	Magic Wand	Intelligent Scissors
Ввод пользователя		
Результат		

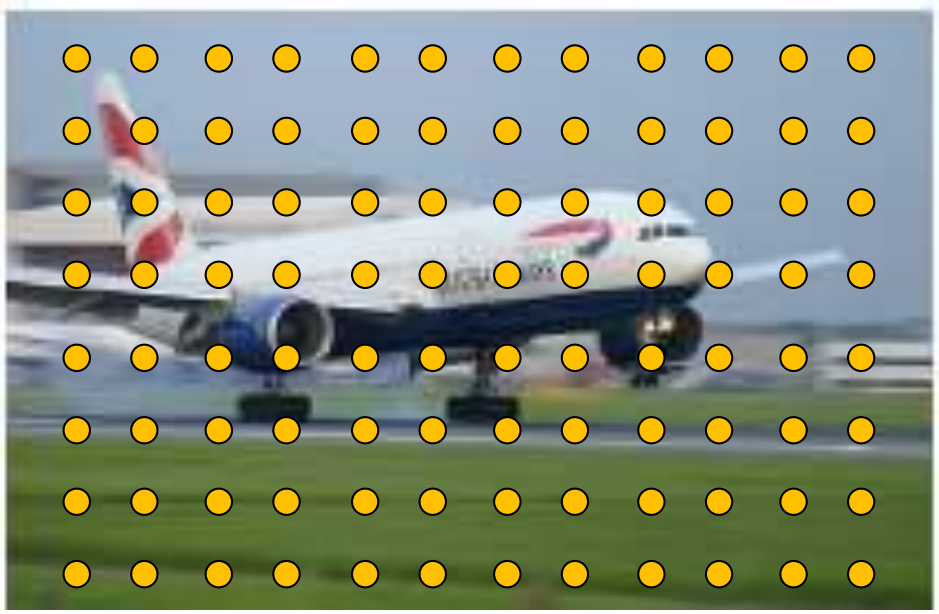
Алгоритм Interactive GraphCuts

- Идея: учитывать **совместно** информацию о свойствах областей (как в MagicWand), информацию о границах/градиентах (как в Intelligent Scissors) и указания пользователя в едином целевом функционале
- Интерфейс: жесткие ограничения в виде мазков (strokes) для объекта (O) и фона (B)



Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images. ICCV 2001.

Формализация задачи



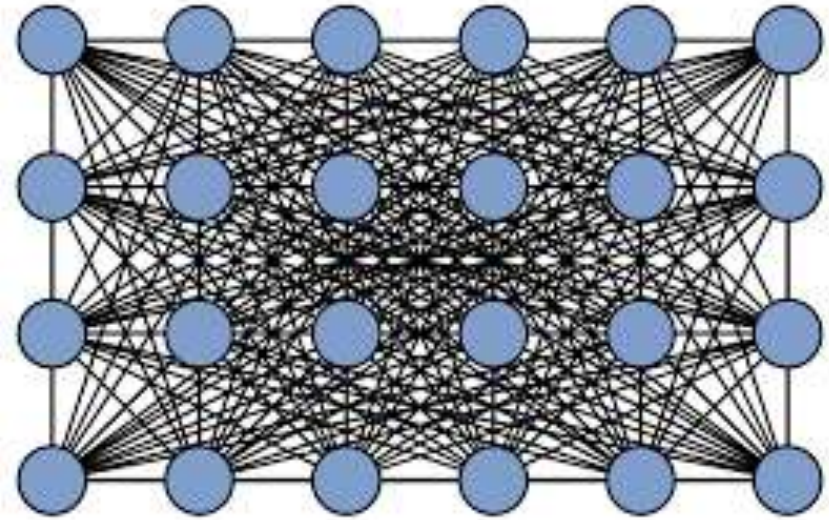
Обозначим:

- x_i – множество всех пикселей (наблюдаемые переменные)
- y_i – метки всех пикселей x_i , $y_i \epsilon [0,1], i \in [1, N]$

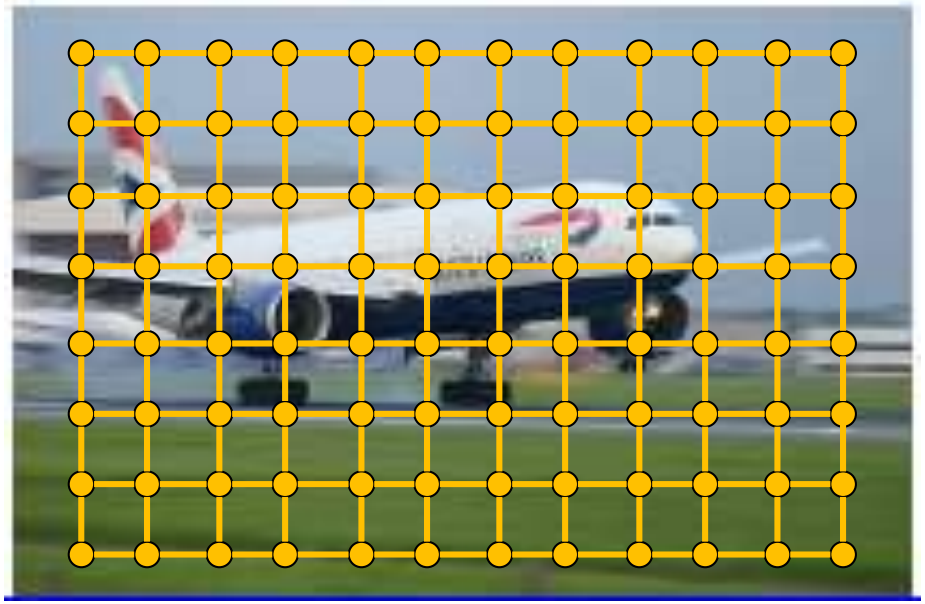
- Сформулируем задачу так: $Y = \arg\max_{Y} P(Y \mid X)$
- В такой формулировке подразумевается, что между переменными есть зависимости, и их нужно учитывать, но при этом оставив задачу вычислимой на практике

Зависимости между переменными

- Между какими переменными есть зависимости?
- Визуализировать зависимость удобно в виде ребра графа, связывающего переменные – узлы графа
- «Структура» зависимостей может быть разная!



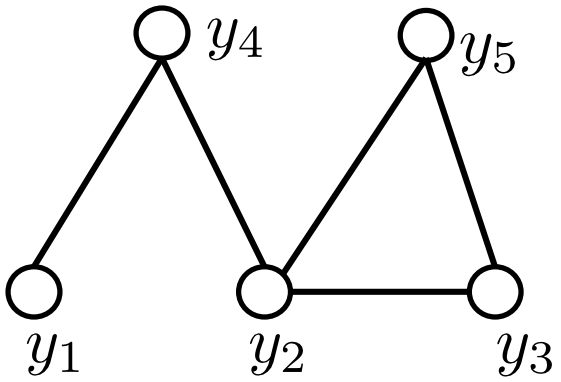
Между всеми



Между соседними

Графическая модель

- Построим граф G=(V,E), V – вершины, E – рёбра, такой что:
  - Все переменные из Y проиндексированы вершинами V
  - Ребро $E=(V_iV_j)$ присутствует, если между y_i и y_j есть непосредственная (а не опосредственная) вероятностная зависимость



Отсутствие ребра между y_i и y_j означает, что они условно независимы, т.е.:

$$P(y_i, y_j \mid Y_{\{i,j\}}) = P(y_i \mid Y_{\{i,j\}})P(y_i \mid Y_{\{i,j\}})$$

(Это ещё называется марковское свойство)

- Пару (Y,G) называют графической моделью. Поскольку для неё выполняется свойство Марковости, то такую модель называют **«Марковским случайным полем» (МСП)**

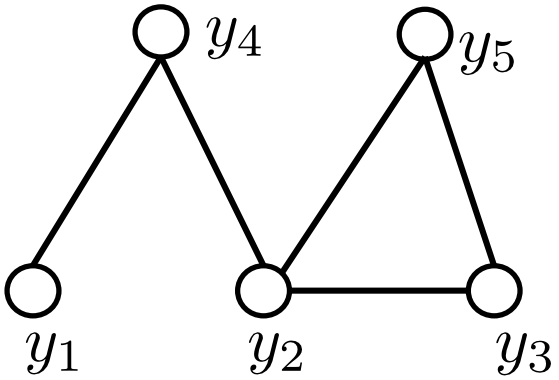
Факторизация совместного распределения

Теорема Хаммерсли-Клиффорда:

- Совместное распределение P(Y) для Марковского случайного поля (МСП) можно факторизовать в произведение неотрицательных функций Φ_i (называемых «факторами») определённых на максимальных кликах D_i графа G
- Функции Φ_i называют «факторами»

$$P(\mathbf{y}) = \frac{1}{Z}\prod_i \Phi_i(\mathsf{D}_i) \quad Z = \sum_{\{Y\}}\prod_i \Phi_i(D_i) \qquad \Phi_i(D_i) > 0$$

нормализующая константа
(partition function)



$$P(Y) = \frac{1}{Z}\Phi(y_1, y_4)\Phi(y_2, y_4)\Phi(y_2, y_3, y_5)$$

Функция энергии

- Запишем фактор Ф(D) как $\Phi(D) = \exp(-\varepsilon(D))$
- $\varepsilon(D) = -\ln \Phi(D)$ называется функцией энергии (energy function) или потенциалом (potential)
- Название пришло из статистической физики, в которой обычно вероятность состояния системы обратно пропорционально энергии
- Теперь можем записать P(Y) в логарифмическом виде:

$$P(Y) \propto \exp\left(-\sum_{i=1}^{m} \varepsilon_i(D_i) \right)$$

- Максимизация P(Y) эквивалентна минимизации суммарной энергии системы!

$$\arg\max P(Y) = \arg\max \exp\left(-\sum_{i=1}^{m} \varepsilon_i(D_i) \right) = \arg\min \sum_{i=1}^{m} \varepsilon_i(D_i)$$

Conditional Random Field (CRF)

- Пусть Y – набор целевых переменных, X – набор наблюдаемых переменных
- Формализм графических моделей можно применить для записи P(Y|X)
- ***Условным случайным полем*** назовём (Y,X,G), в котором вершины G соответствуют Y, граф аннотирован факторами $\Phi_i(D_i)$, обусловленных наблюдаемыми переменными X

$$P(Y \mid X) == \frac{1}{Z} \prod_i \Phi_i(Y_i \mid X)$$

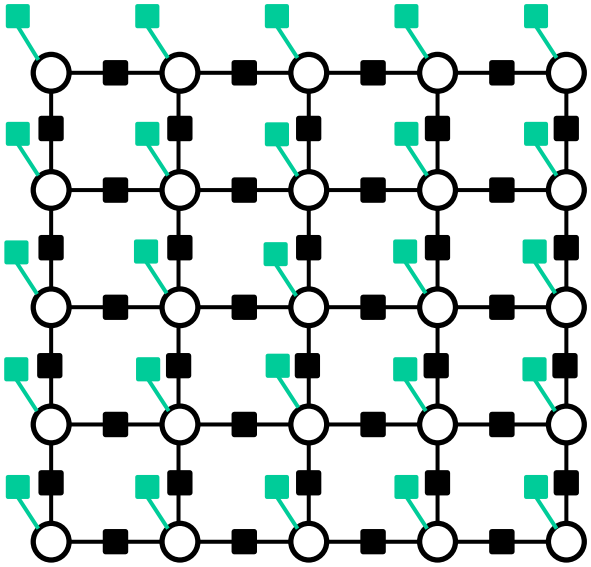
Conditional Random Field

- Плюсом модели условного случайного поля является то, что нам нужно моделировать зависимости только между скрытыми переменными
- Мы можем использовать большой набор наблюдаемых переменных, зависимости между которыми сложны или вообще непонятны

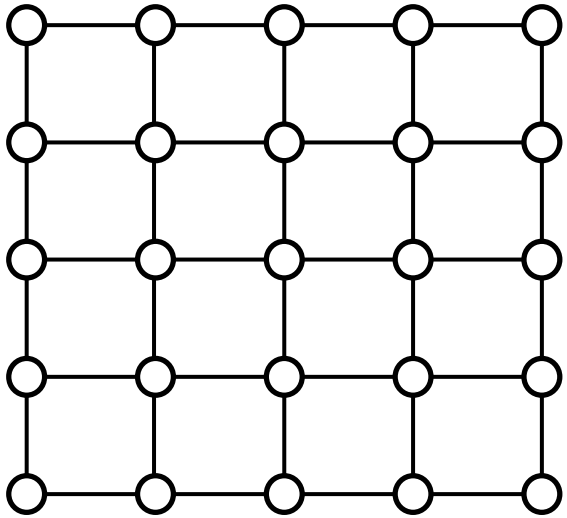
- Поэтому в большинстве случаев используется или подразумевается именно CRF

Наш пример

Каждому пикселю изображения задаём метку y, и зависит метка пиксела только от своих соседей



MRF



CRF

$$P(Y,X) =$$

$$= \frac{1}{Z} \prod_{i} \Phi_i(y_i, x_i) \prod_{(i,j)\in N} \Psi_{i,j}(y_i, y_j)$$

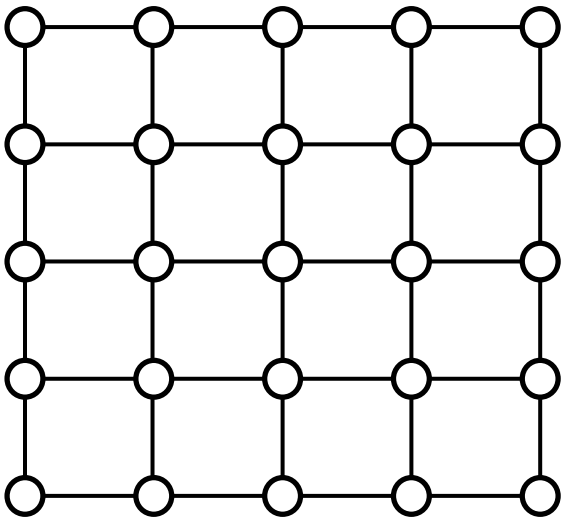
$$P(Y \mid X) = \prod_{(i,j)\in N} A_{i,j}(y_i, y_j \mid X)$$

$$= \frac{1}{Z} \prod_{i} \Phi_i(y_i \mid X) \prod_{(i,j)\in N} \Psi_{i,j}(y_i, y_j \mid X)$$

Факторы в модели изображения

$$P(Y \mid X) = \frac{1}{Z} \prod_{i} \Phi_i(y_i \mid X) \prod_{(i,j) \in N} \Psi_{i,j}(y_i, y_j \mid X)$$

$$E(Y \mid X) = -\sum_{i} \log(\Phi_i(y_i \mid X)) - \sum_{(i,j) \in N} \log\left(\Psi_{i,j}(y_i, y_j \mid X)\right)$$



- Унарные (размер 1) – зависимость метки от изображения (классификатор)
- Парные (размер 2) – штраф за сходство или различие соседних переменных в зависимости от свойств изображения
- Более высоких порядков – кривизну контура, площадь фигуры и т.д.

Обычно используют унарные и парные потенциалы, т.к. для потенциалов более высоких порядков оптимизация сложнее

Бинарная сегментация

$$Energy_{MRF} = E(Y) = \sum_{p}(\phi(I \mid y_p) + \sum_{\{p,q\}\in N}(\phi(I \mid y_p, y_q) * \psi(y_p, y_q))) + const$$

Унарное правдоподобие
(Unary likelihood)

Член контраста
(Contrast Term)

Модель априорной вероятности Поттса
(Potts Model Prior)

$$MAP\ \ solution \quad Y^* = \arg\min_{Y} E(Y)$$

Данные (I)

Унарное правдоподобие
(Unary likelihood)

Бинарные члены
(Pair-wise Terms)

Решение

Потенциалы

- **Унарные**
  - похожесть по цветовой модели

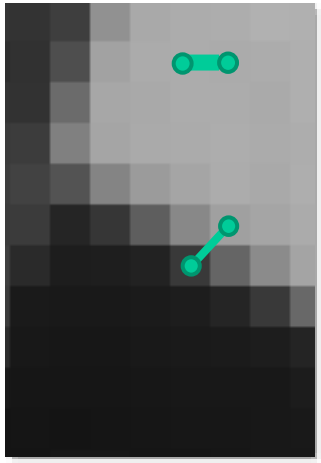
$$\varphi(I|"obj") = -\ln P(I_p|O)$$

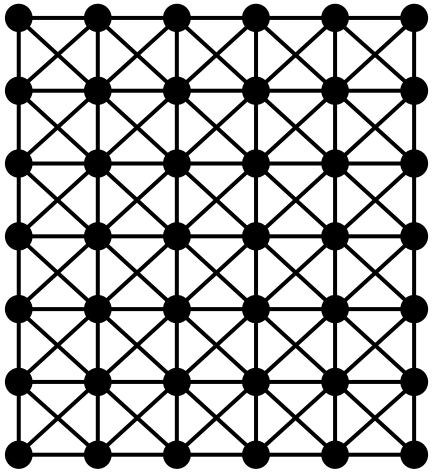
$$\varphi(I|"bkg") = -\ln P(I_p|B)$$

  - Модели цвета считаются по отмеченным областям

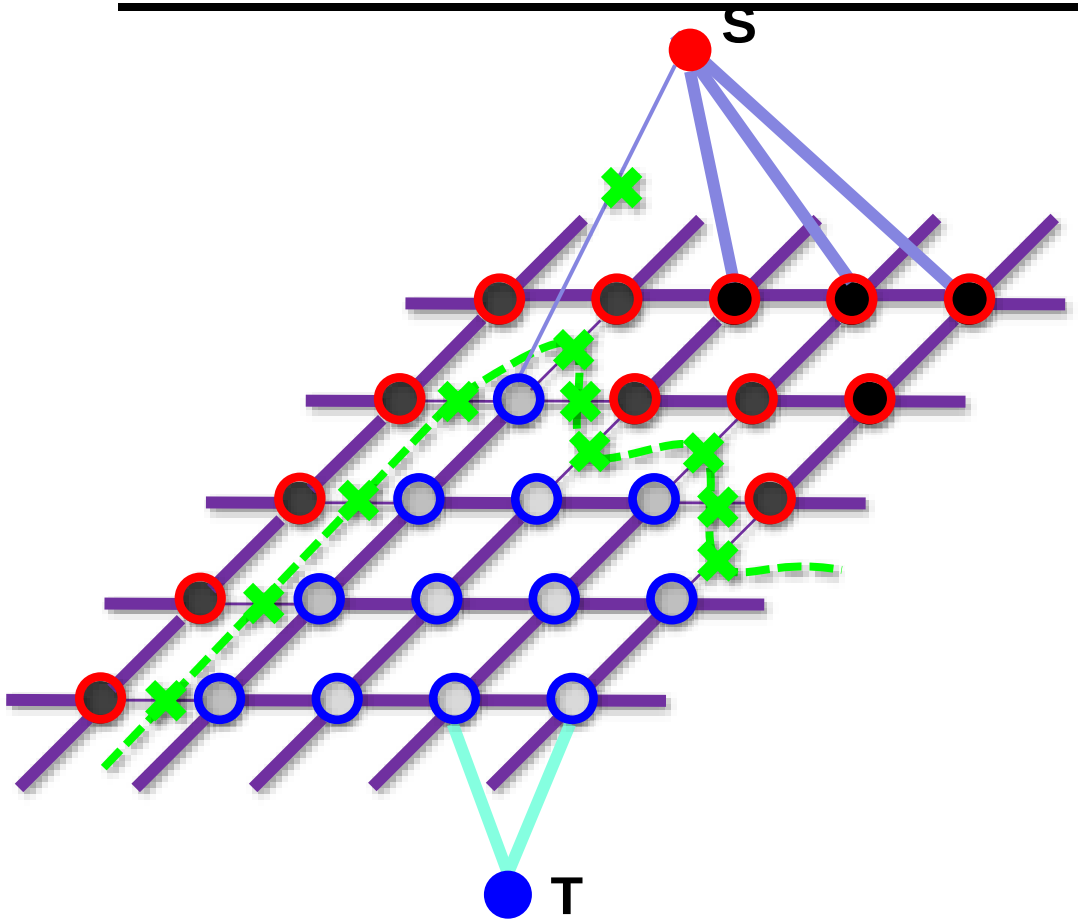
- **Парные**
  - Пропорционально контрасту и удалённости

$$P^{pq} \propto \exp\left(-\frac{||I_p - I_q||^2}{2\sigma^2}\right) \cdot \frac{1}{||p-q||^2}$$

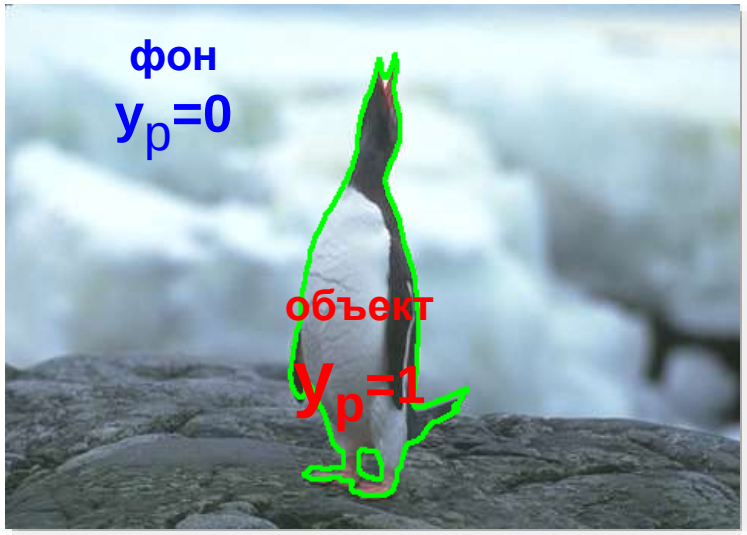




Минимальный разрез в сети = минимуму энергии

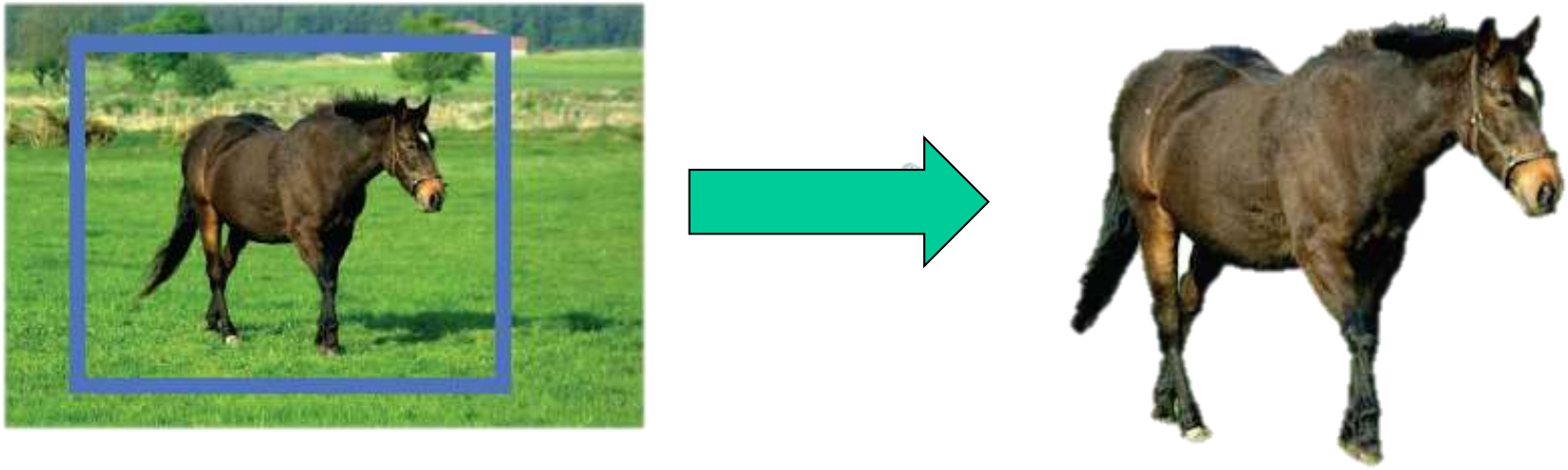


[Boykov & Jolly ‘01]



$$E(\mathbf{y}) = \sum_{p \in \mathcal{V}} F^p y_p + \sum_{p \in \mathcal{V}} B^p (1 - y_p) + \sum_{p,q \in \mathcal{N}} P^{pq} |y_p - y_q|$$

GrabCut



Идеи:

- Заменить интерфейс с мазков на выделение объекта рамкой
- Собирать информацию о цвете со всего изображения, а не только из мазков пользователей
- Итеративно обновлять цветовую модель

C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *Proc. ACM Siggraph, 2004.*

Итеративные разрезы графов

Инициализация пользователя

K-средних для обучения статистик цветов

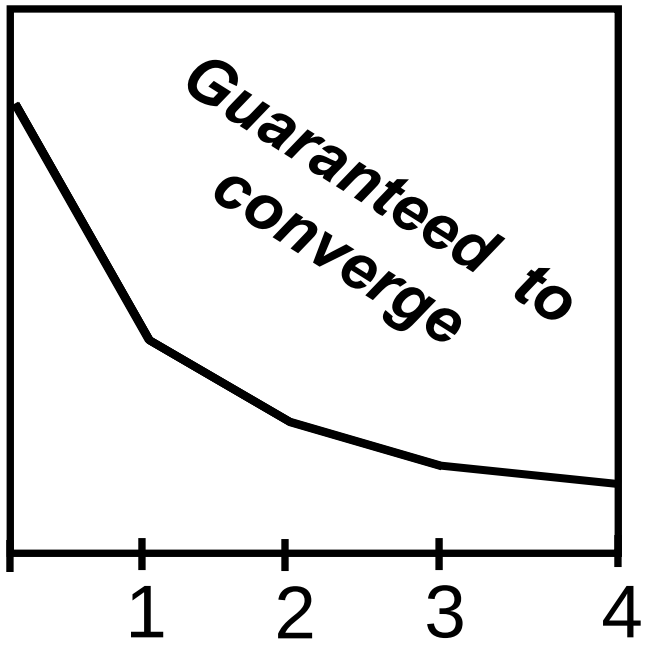
Graph cuts для сегментации

- Цвет фона и объекта моделируется смесью гауссиан (обычно 5-8 компонент)
- Каждый пиксель объекта/фона сопоставляется компоненте смеси
- Затем уточняются параметры смеси

Slide by C. Rother

Итеративные разрезы графов

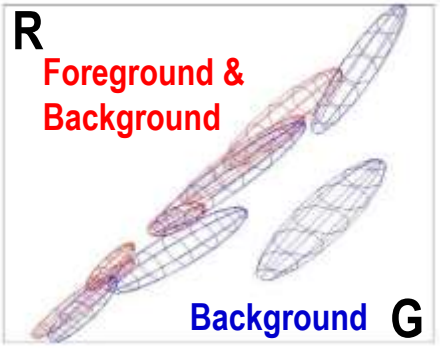
Результат



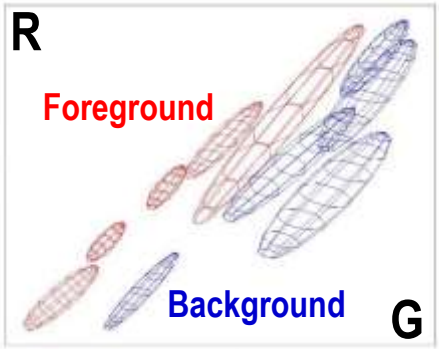
Энергия после итераций

Slide by C. Rother

Цветовая модель



Итеративное уточнение



Цвет фона и объекта моделируется смесью гауссиан (обычно 5-8 компонент)

Slide by C. Rother

Несколько примеров

Вычисляются автоматически по прямоугольной рамке

Slide by C. Rother

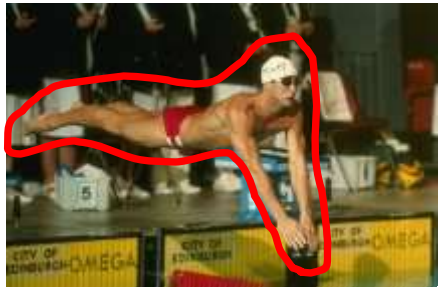
Сложные примеры

Камуфляж и низкий контраст

Тонкие структуры

Нет телепатии

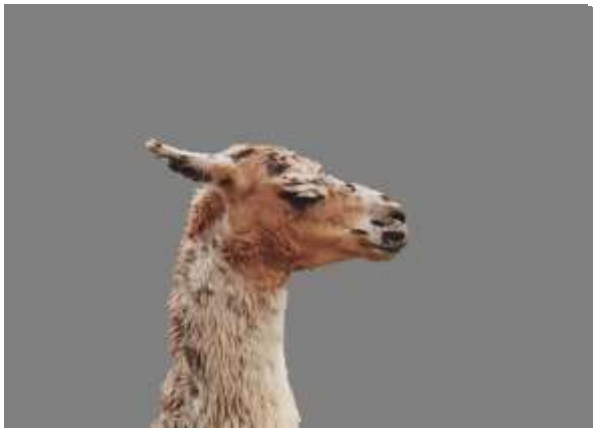
Исходный прямоугольник



Исходный результат

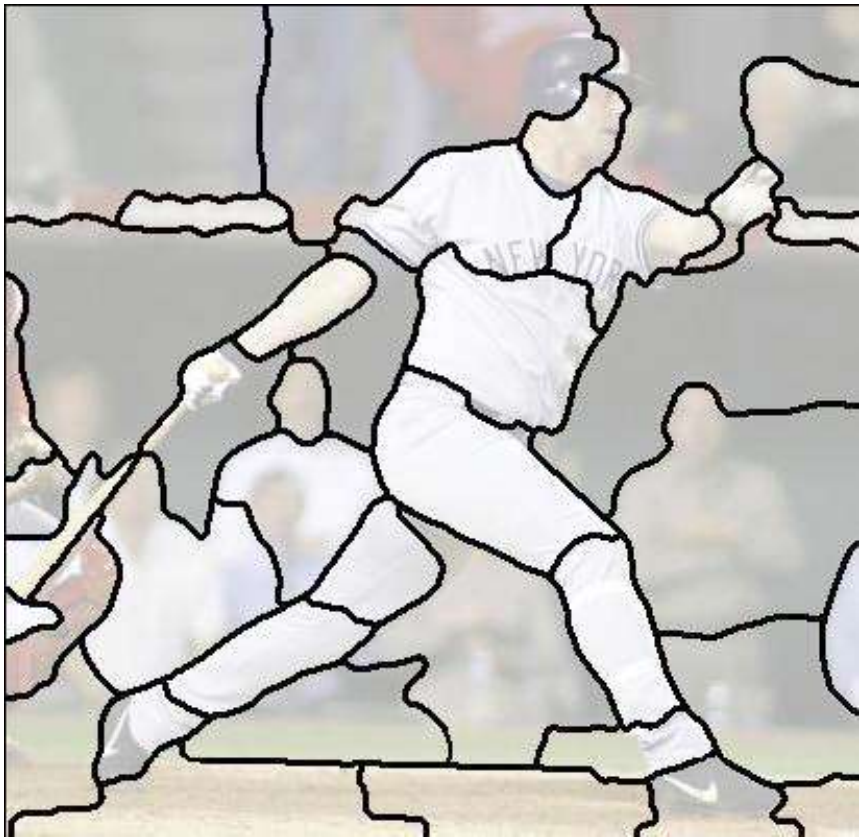
Slide by C. Rother

Сравнение

	Boykov and Jolly (2001)	GrabCut
Пользовательский ввод		
Результат		
	Error Rate: 0.72%	**Error Rate: 0.72%**

Slide by C. Rother

Визуальная сегментация

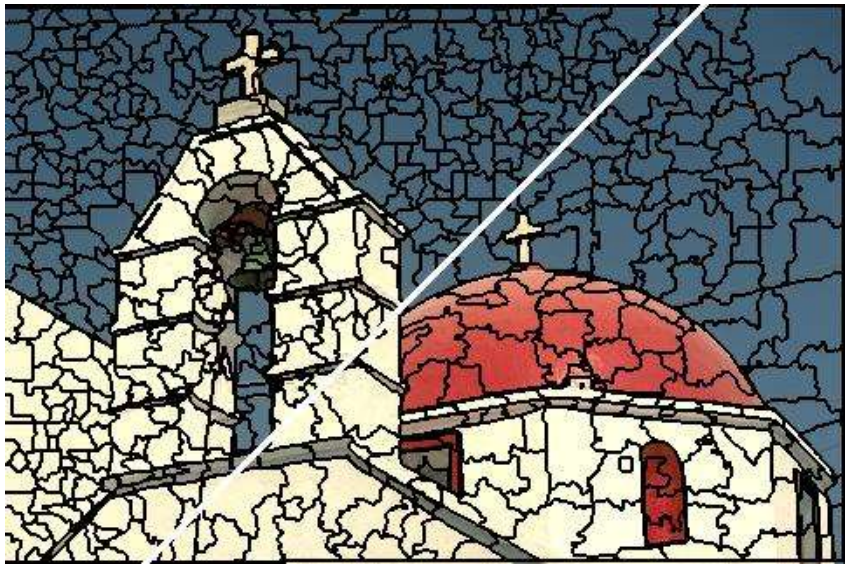


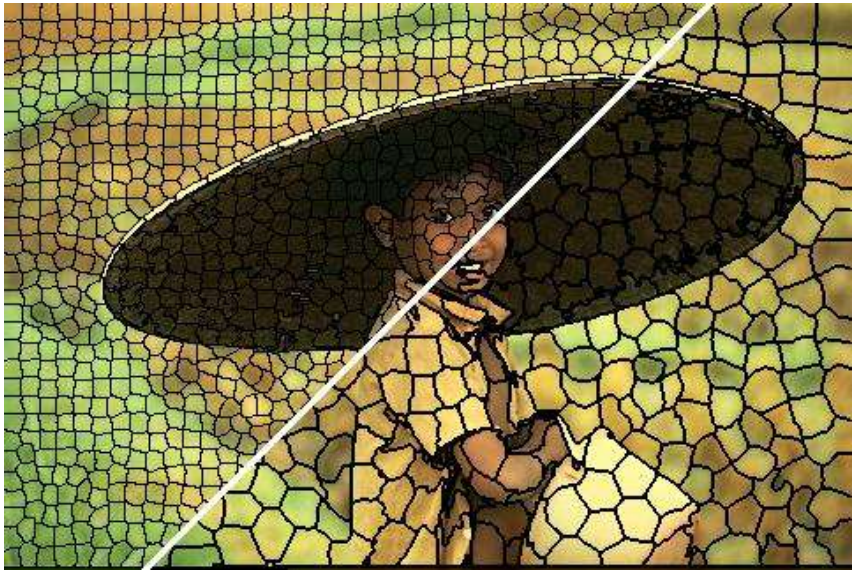
Требования к сегментации?

Каким условиям должны удовлетворять сегменты?

- Границы сегментов должны соответствовать границам объектов
- Сегмент должен целиком содержаться внутри объекта
- Небольшие объекты не должны быть частью сегмента, а описываться своим сегментом
- Сегмент должен быть однородным по визуальным характеристикам
- Сегменты должны быть достаточно большими, чтобы быть «информативными»
- Компактные, примерно одного размера
- Равномерно распределенные по изображению
- Алгоритм должен работать быстро

Суперпиксели



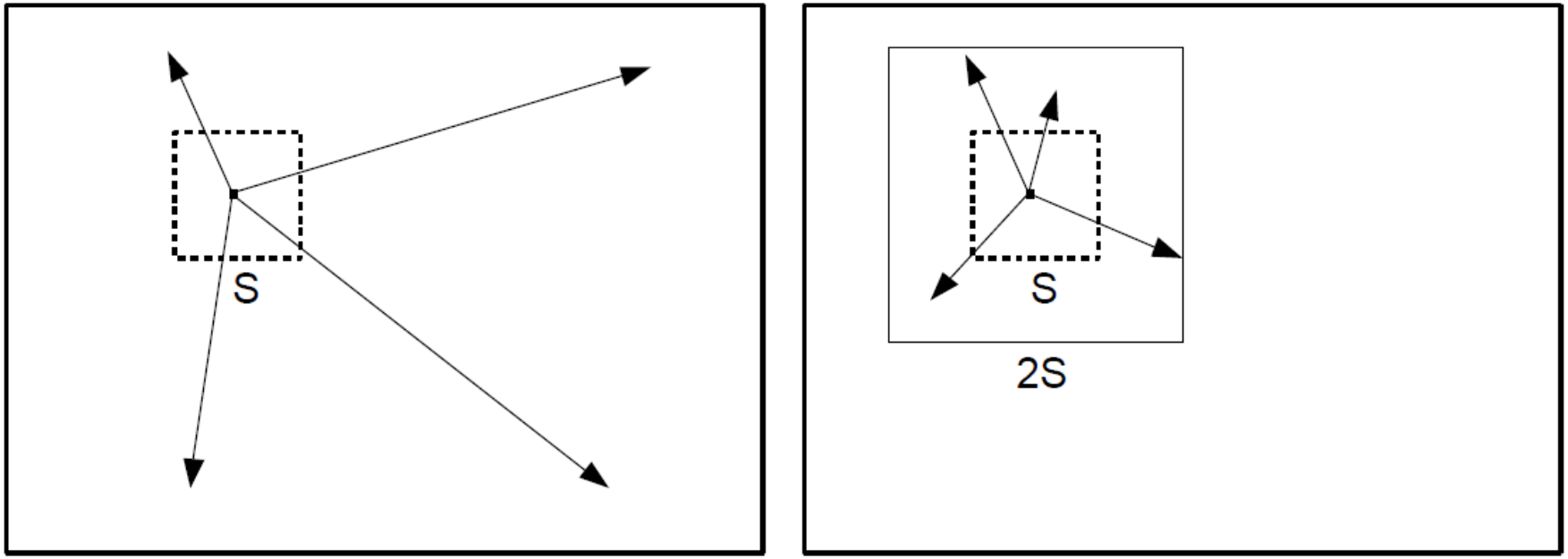


- Сегменты, «удовлетворяющие» вышеуказанным требованиям называют «суперпиксели»
- «Суперпиксельная сегментация»
- Ещё называют «пересегментацией» (oversegmentation)

Пересегментация через кластеризацию

- Представим множество пикселов как выборку (множество вектор-признаков)
- Применим какой-нибудь метод кластеризации к данным в пространстве признаков
- Каждый кластер будет соответствовать одному сегменту
- Как обеспечить «локальность» кластеров в изображении?
  - (x,y) можно включить в вектор-признак
  - Жесткие ограничения на расстояния между пикселами

Simple linear iterative clustering (SLIC)



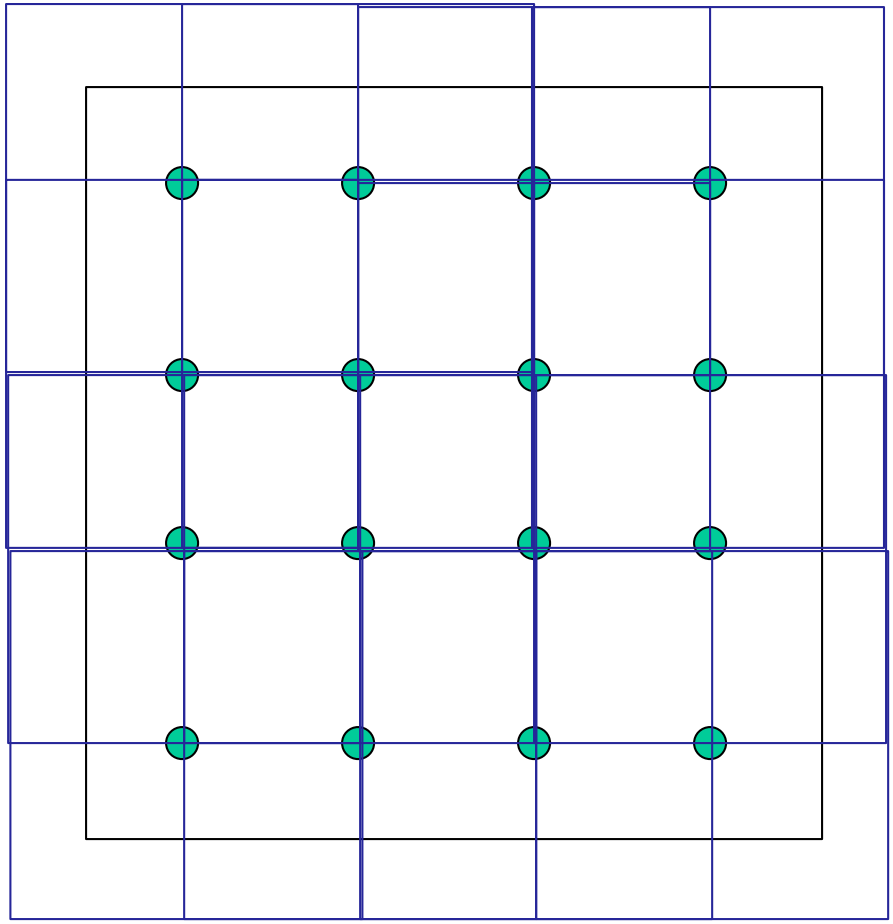
(a) standard k-means searches the entire image

(b) SLIC searches a limited region

- Зачем сопоставлять пиксель со *всеми* остальными?
- Будем сравнивать с «центром кластера» только пиксели, которые *могут принадлежать* этому сегменту (на расстоянии < s)
- Инициализируем кластеры по сетке на расстоянии s

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE PAMI 2012.

Алгоритм SLIC



- Инициализируем центры кластеров C_k по сетке с шагом S
- Инициализируем в каждом пикселе метку L(i)=-1 и расстояние D(i) до ближайшего кластера -∞
- Проходим по кластерам C_k
  - Для каждого пикселя в области 2Sx2S считаем расстояние до C_k
  - Если расстояние меньше D(i), тогда ставим метку L(i)= k, и записываем в D(i) новое значение
- Повторяем до тех пор, пока суммарное изменение кластеров не будет меньше порога

Функция сравнения

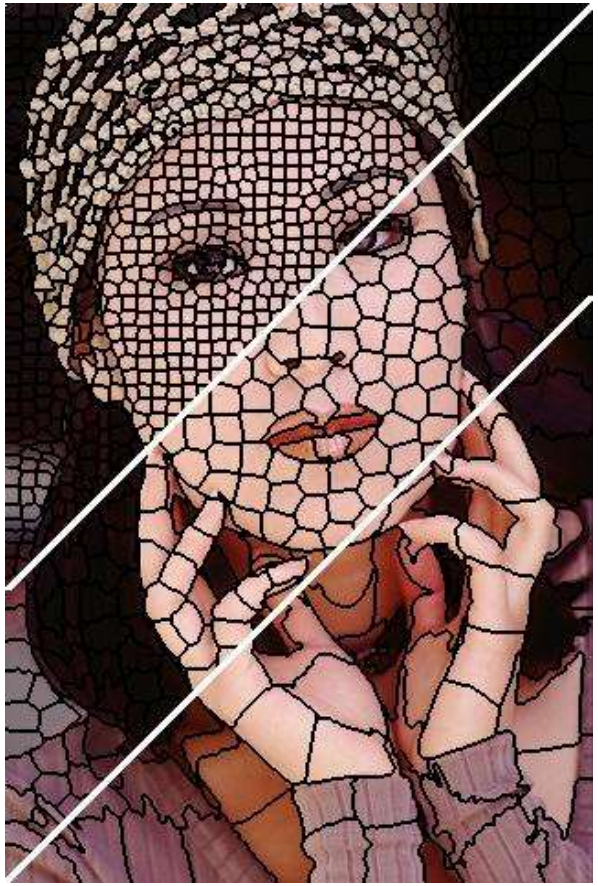
- Работаем в LABCIE + (x,y) пространстве
- Суммируем расстояния по цвету и положению и нормализуем на максимальное расстояние внутри кластера

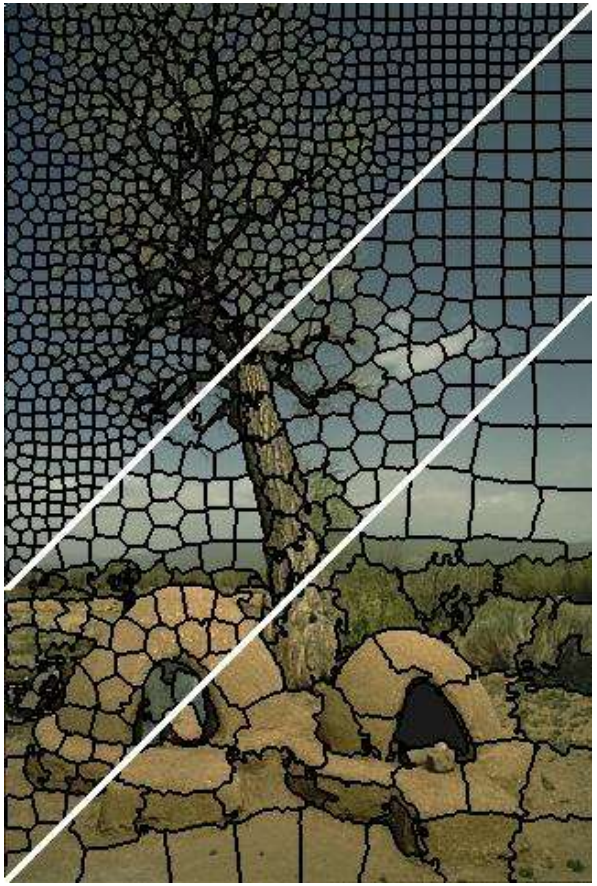
$$
\begin{aligned}
d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\
d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}.
\end{aligned}
$$

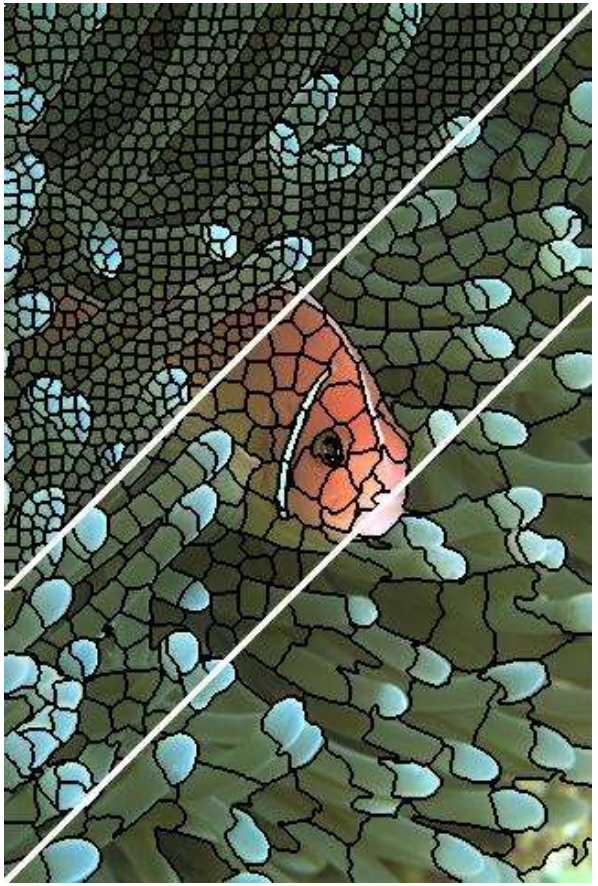
- Поскольку расстояние по цвету неизвестно, можем зафиксировать его как m

$$
D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \quad \Rightarrow \quad D = \sqrt{{d_c}^2 + \left(\frac{d_s}{S}\right)^2 m^2}
$$

Примеры работы



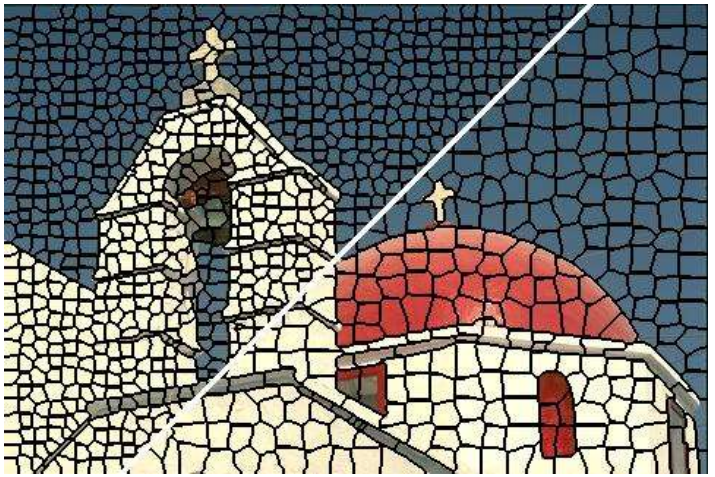




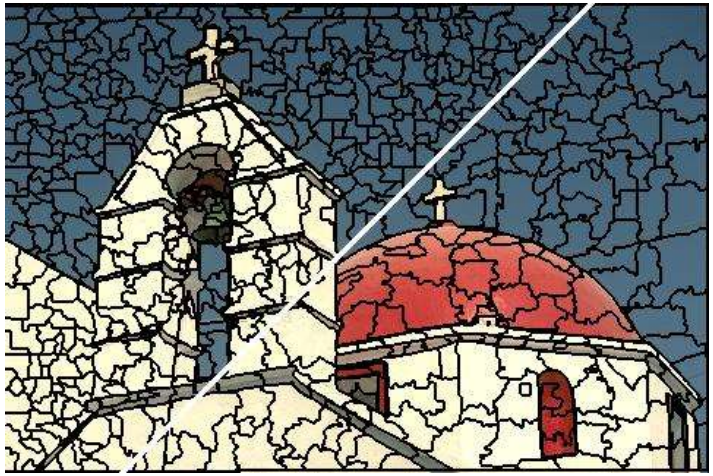
Сложность - $\left(\frac{n}{s}\right)^2 S^2 t = n^2 t = Nt$

Визуальное сравнение

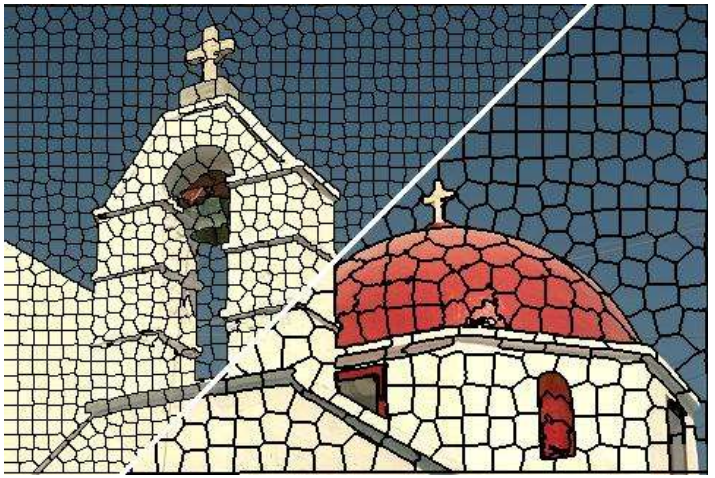
Efficient Graph-Based



TurboPixel



QuickShift



SLIC

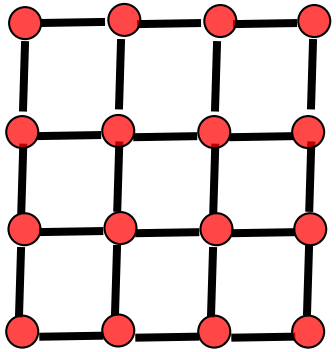
Семантическая сегментация

Использование CRF

Объединяем в целевую функцию локальные классификаторы пикселов и учёт связей с помощью CRF

Изображение

CRF



Парные члены

$$Energy_{MRF} = E(F) = \sum_{p}(\phi(I \mid F_p) + \sum_{\{p,q\}\in N}(\phi(I \mid F_p, F_q) * \psi(F_p, F_q))) + const$$

Унарное правдоподобие (Unary likelihood)

Член контраста (Contrast Term)

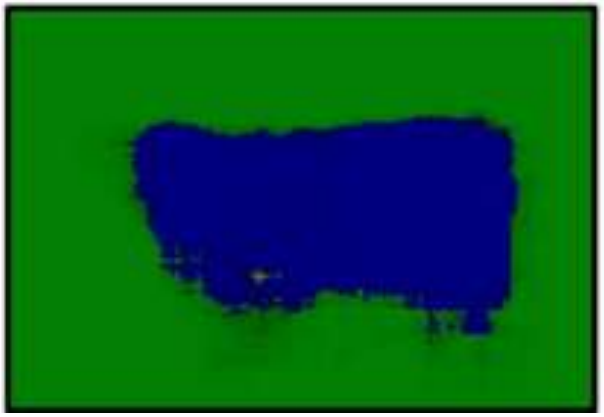
Модель априорной вероятности Поттса (Potts Model Prior)

$$MAP\ \ solution \quad F^* = \arg\min_F E(F)$$

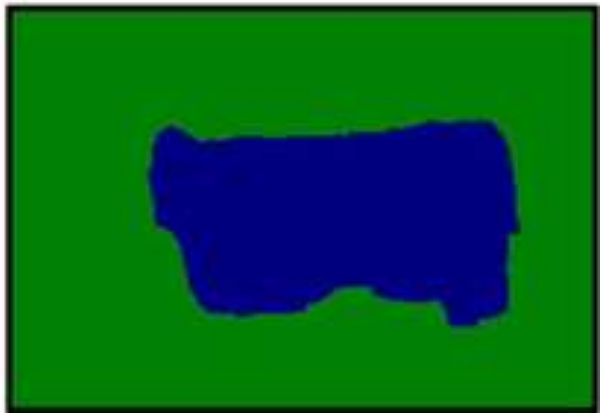
Вклад CRF

Пример из работы TextonBoost

Исходное изображение



Независимая попиксельная классификация

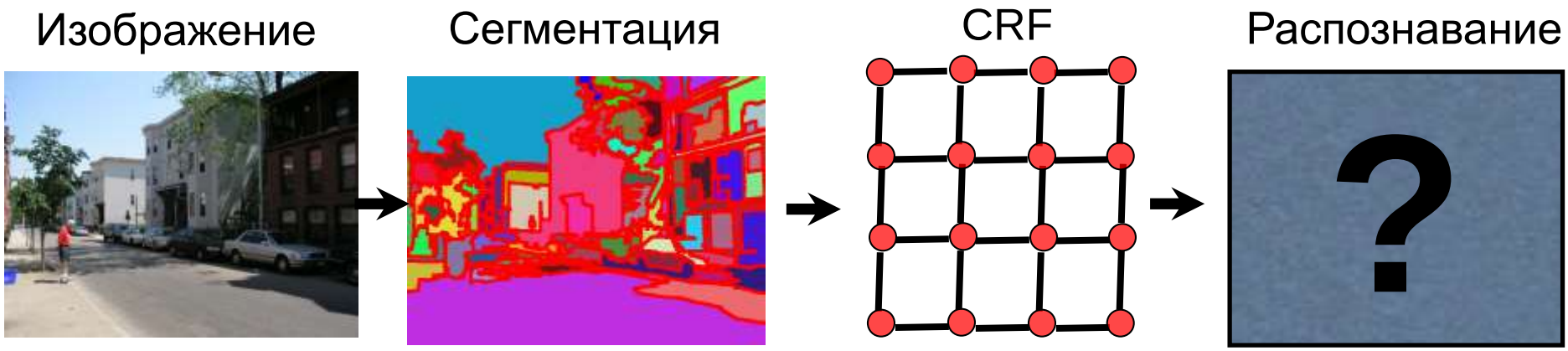


Сглаживание CRF

- При использовании только визуального сходства в парных потенциалах
- Вклад CRF – сглаживание
- Наблюдается «shrinking bias» (укорачиваем границу, «срезая» мелкие детали)

J. Shotton, J. Winn, C. Rother, A. Criminisi, *TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation*, ECCV 2006

CRF на суперпикселях

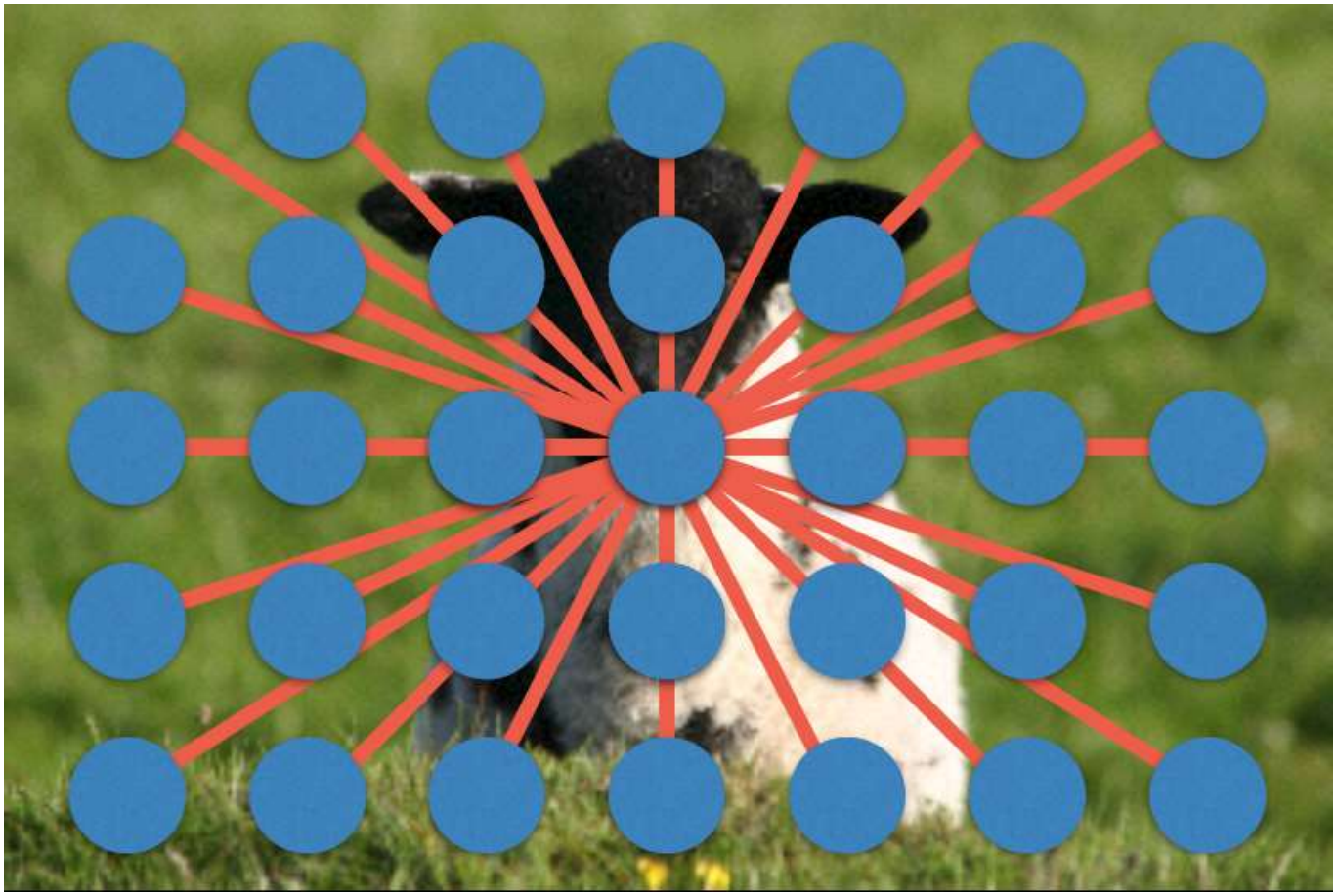


- Сегментация изображения на суперпиксели
- Задание CRF (условного случайного поля) на графе смежности на суперпикселях
- Разметка - вывод в CRF с помощью альфа-расширения
  - Метод оптимизации дискретного функционала с k-метками, основанный на последовательном решении бинарных задач (замена текущих меток на a)

Brian Fulkerson, Andrea Vedaldi, Stefano Soatto. Class Segmentation and Object Localization with Superpixel Neighborhoods, CVPR 2009

Плотносвязанные CRF

$$E(\mathbf{x}) = \sum_i \phi_i(\mathbf{x}_i) + \sum_{i,j \in N} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$$



Плотносвязанные CRF

- Bilateral Pairwise Term

$$\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mu(\mathbf{x}_i, \mathbf{x}_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right)$$

Label compatibility

Location smoothness

Appearance smoothness

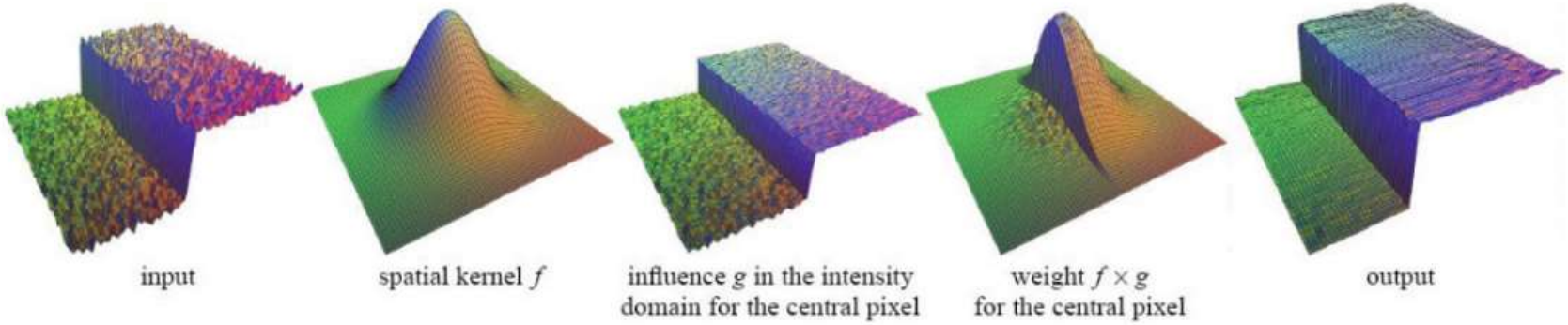


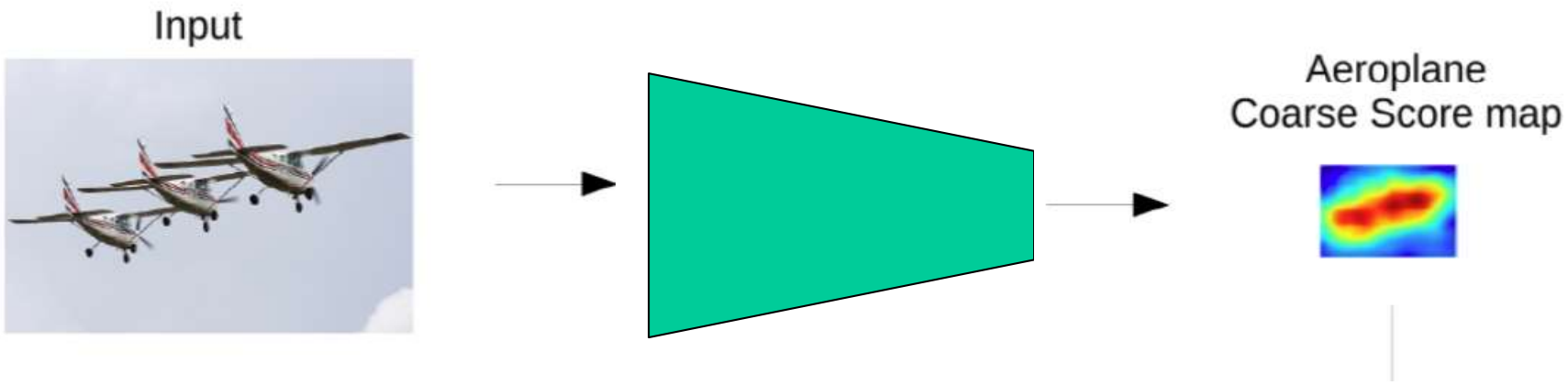
Image credit: Krähenbühl et al.

Плотносвязанные CRF

- Дальние связи позволяют преодолеть «shrinking bias»
  - Граница получается «точной», а не «укороченной»
- Огромное количество связей, вывод можно сделать приближённо

Image credit: Krähenbühl et al.

Нейросетевые модели для сегментации



- Применим свёрточную архитектуру для классификации к изображению, пр. VGG16
- Получим матрицу признаков с разрешением в 32 раз меньше
- Для каждого пикселя признаков применим классификатор для оценки классов
  - Примерно как RPN в Faster R-CNN
- Получим карту разметки низкого разрешения
- Нужно повысить разрешение до исходного!

DeepLab

Воспользуемся билинейной интерполяцией и Dense CRF для повышения разрешения

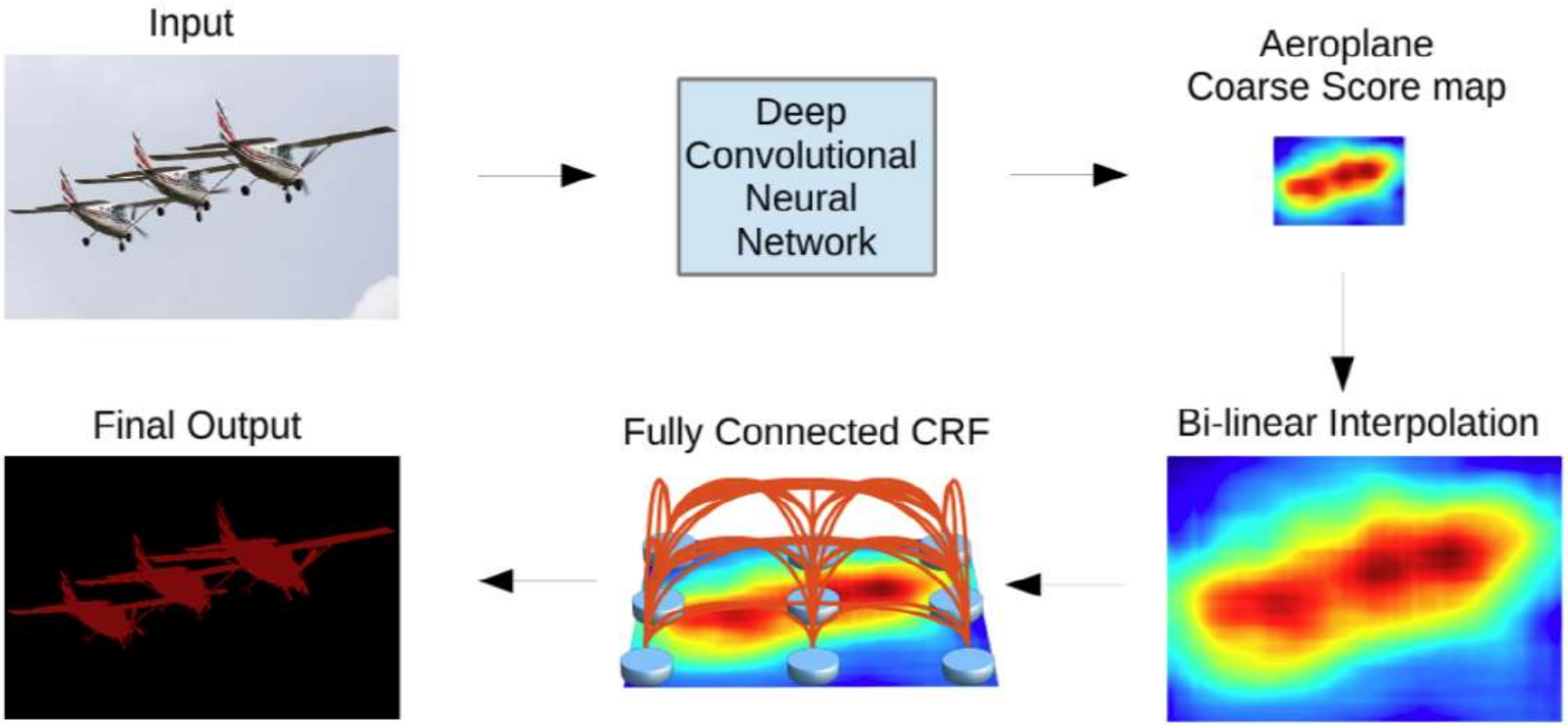
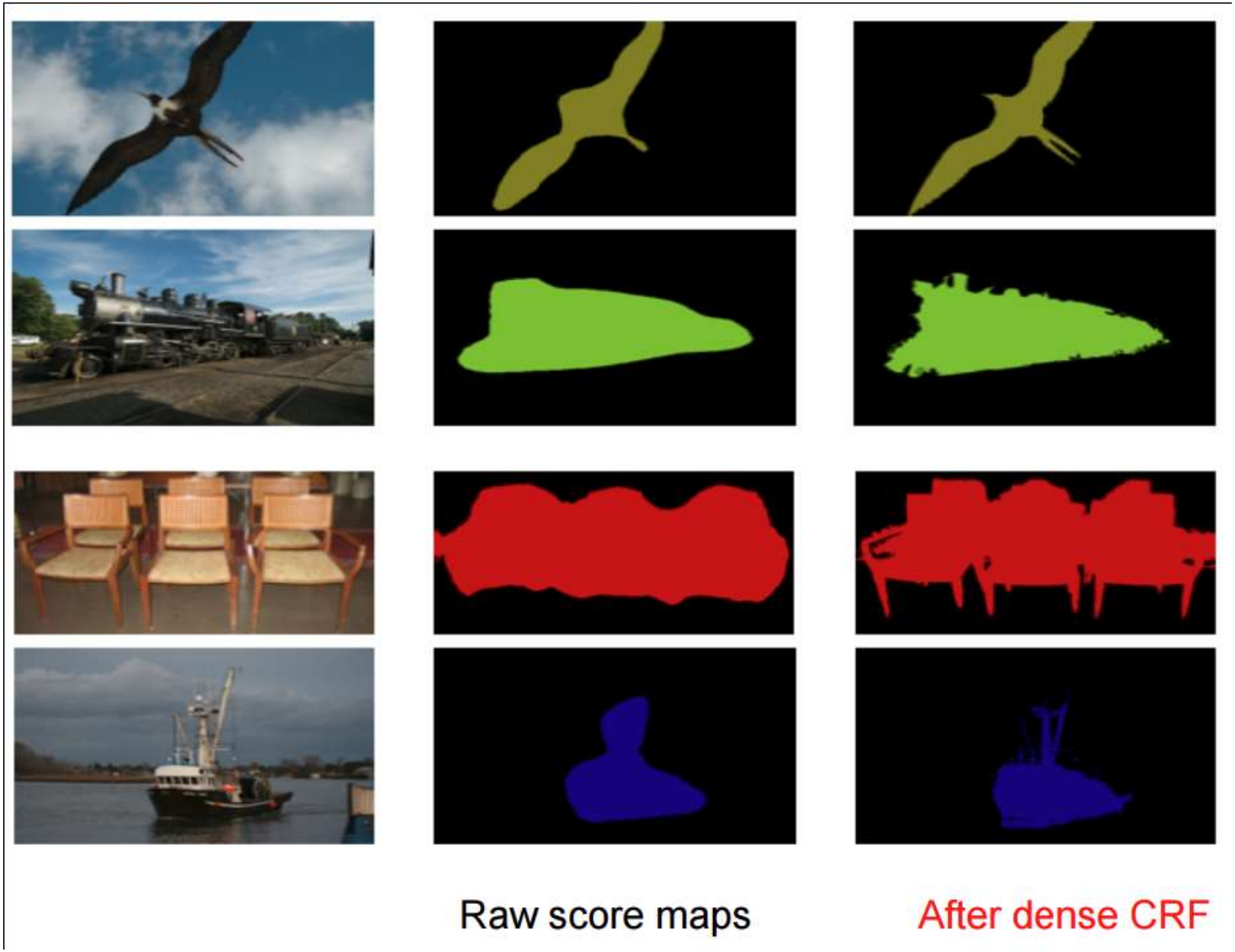
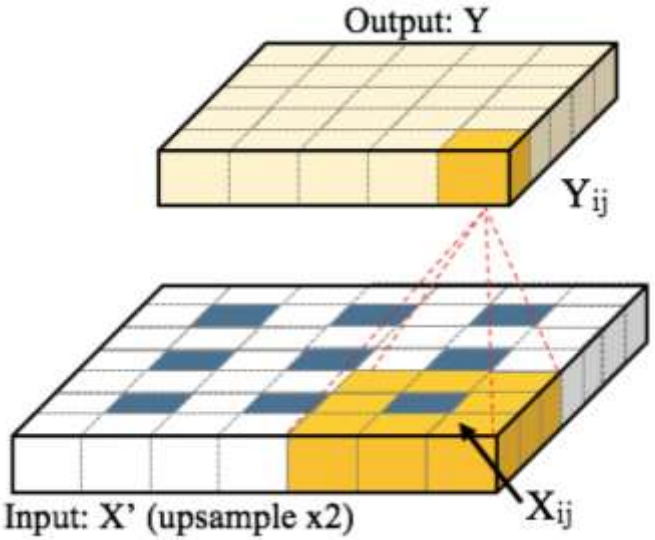


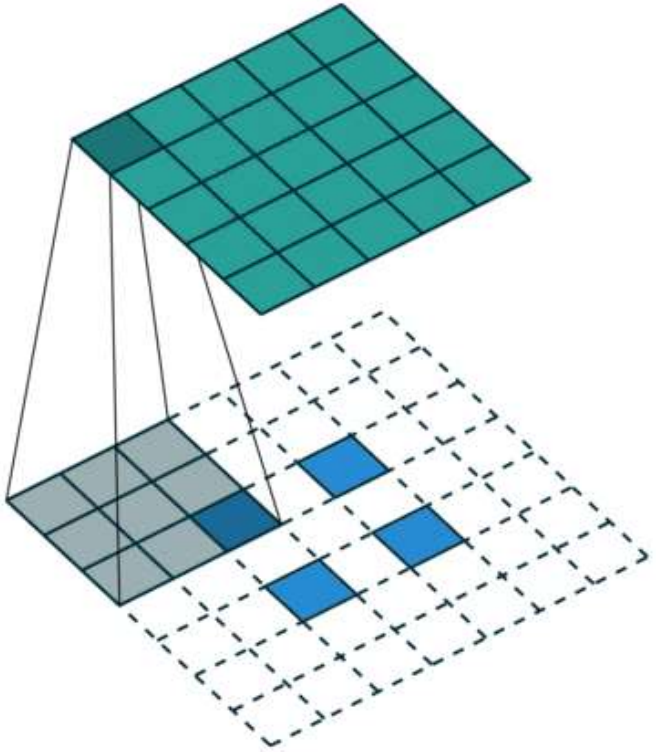
Image credit: Chen et al.

Результаты



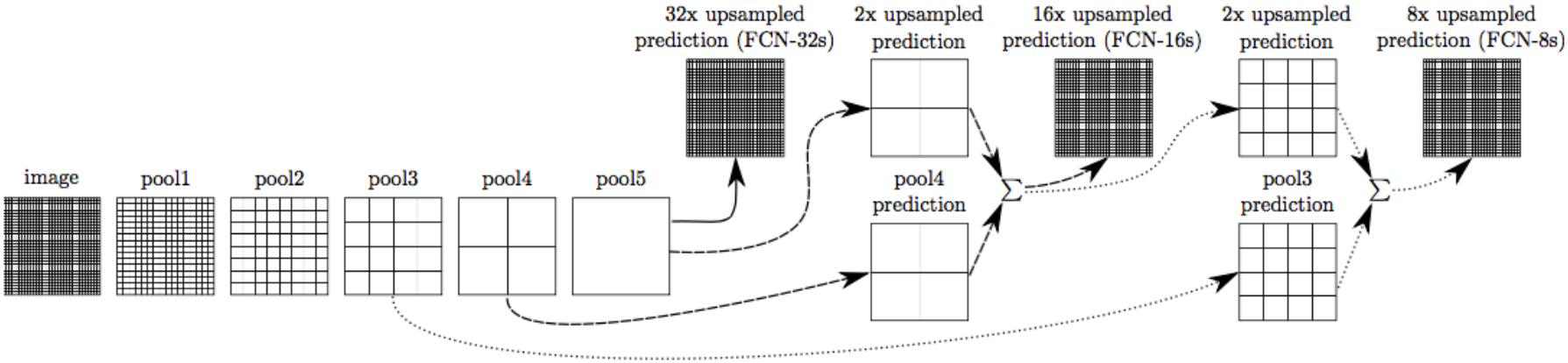
Transposed convolution

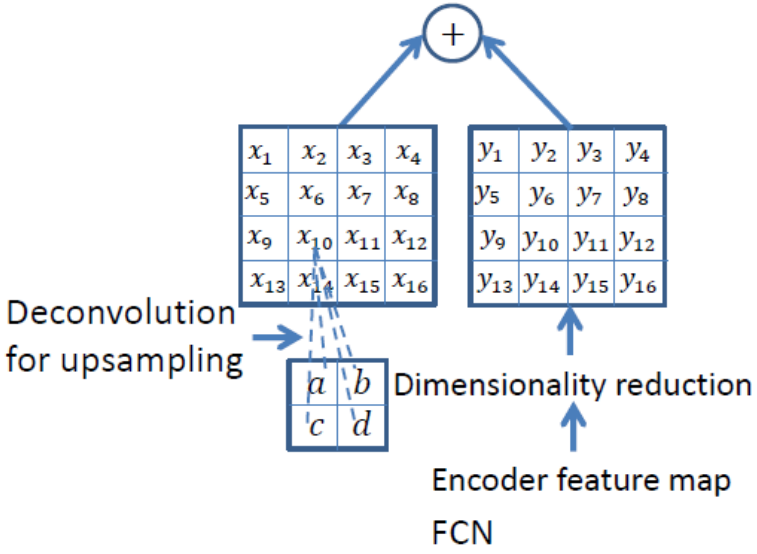




Можем увеличить разрешение картинки, расставив пиксели редко, и затем интерполируя значения в промежутках с помощью свёртки

Fully convolutional networks

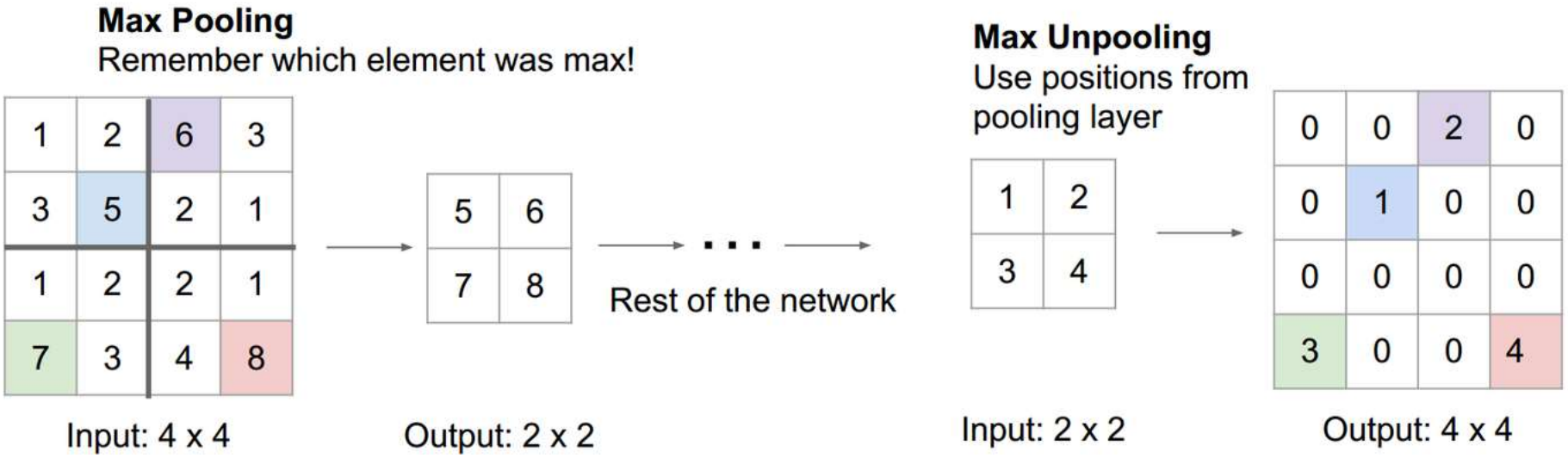




- CNN признаки низкого разрешения на последнем уровне (32x уменьшение)
- Несколько этапов повышения разрешения через “deconvolutional filters”
- Выход уровня повышения разрешения складывается с признаками соответствующего уровня кодирования

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.

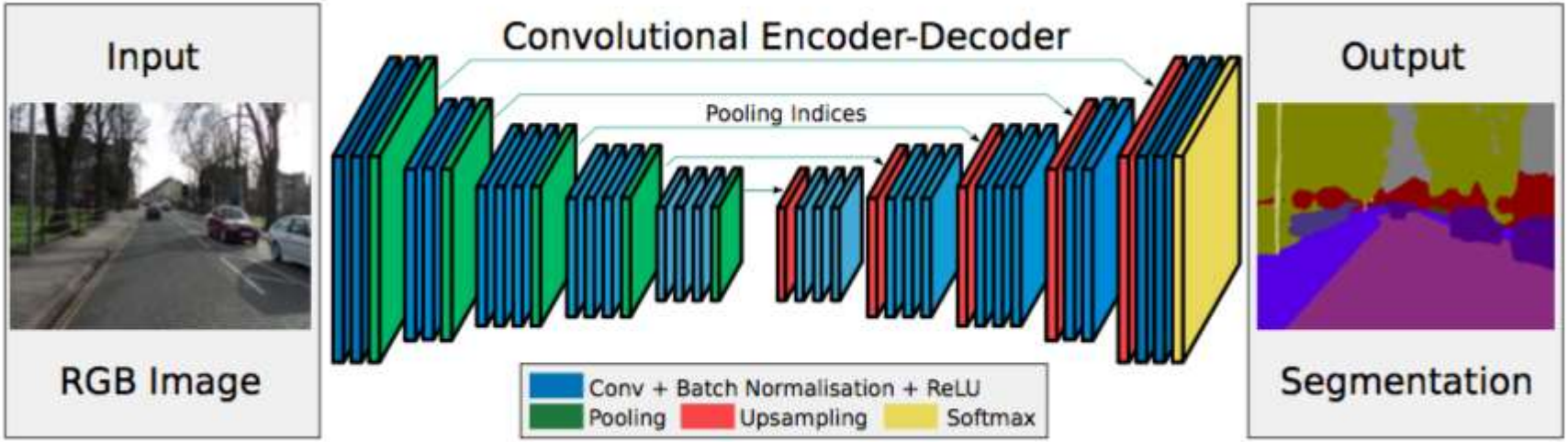
Max Unpooling



- Сохраняем индексы каждого max-pooling слоя
- Про повышении разрешения делаем так:
  - Копируем значения из выхода max-pooling слоя с учётом запомненный индексов
  - Применяем обученные свёртки для сглаживания

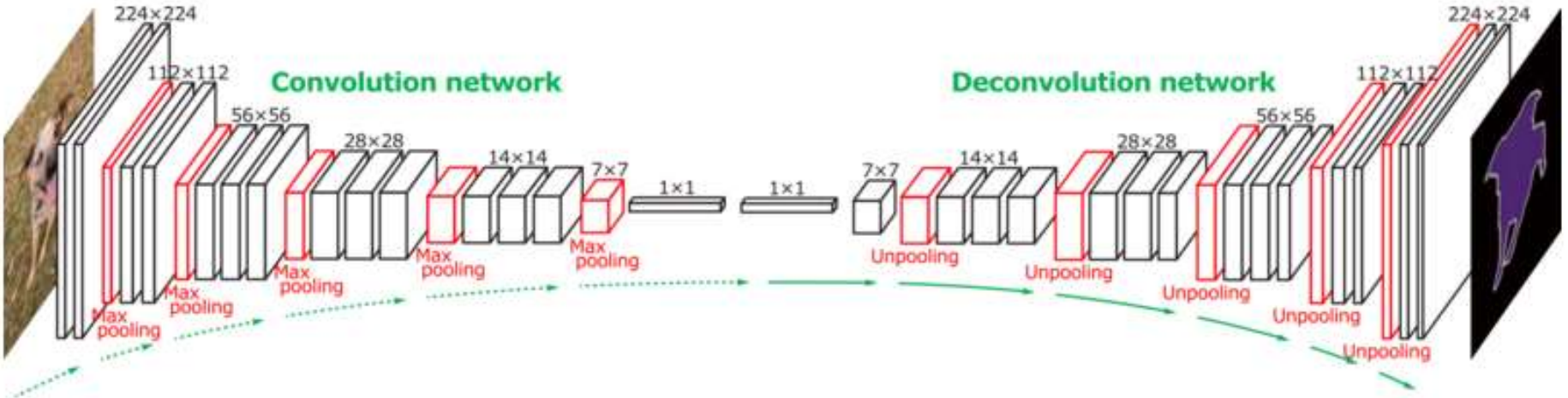
Source: https://tariq-hasan.github.io/concepts/computer-vision-semantic-segmentation/

Encoder-Decoder with Max Unpooling



Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR, abs/1511.00561, 2015.

Deconvolutional network

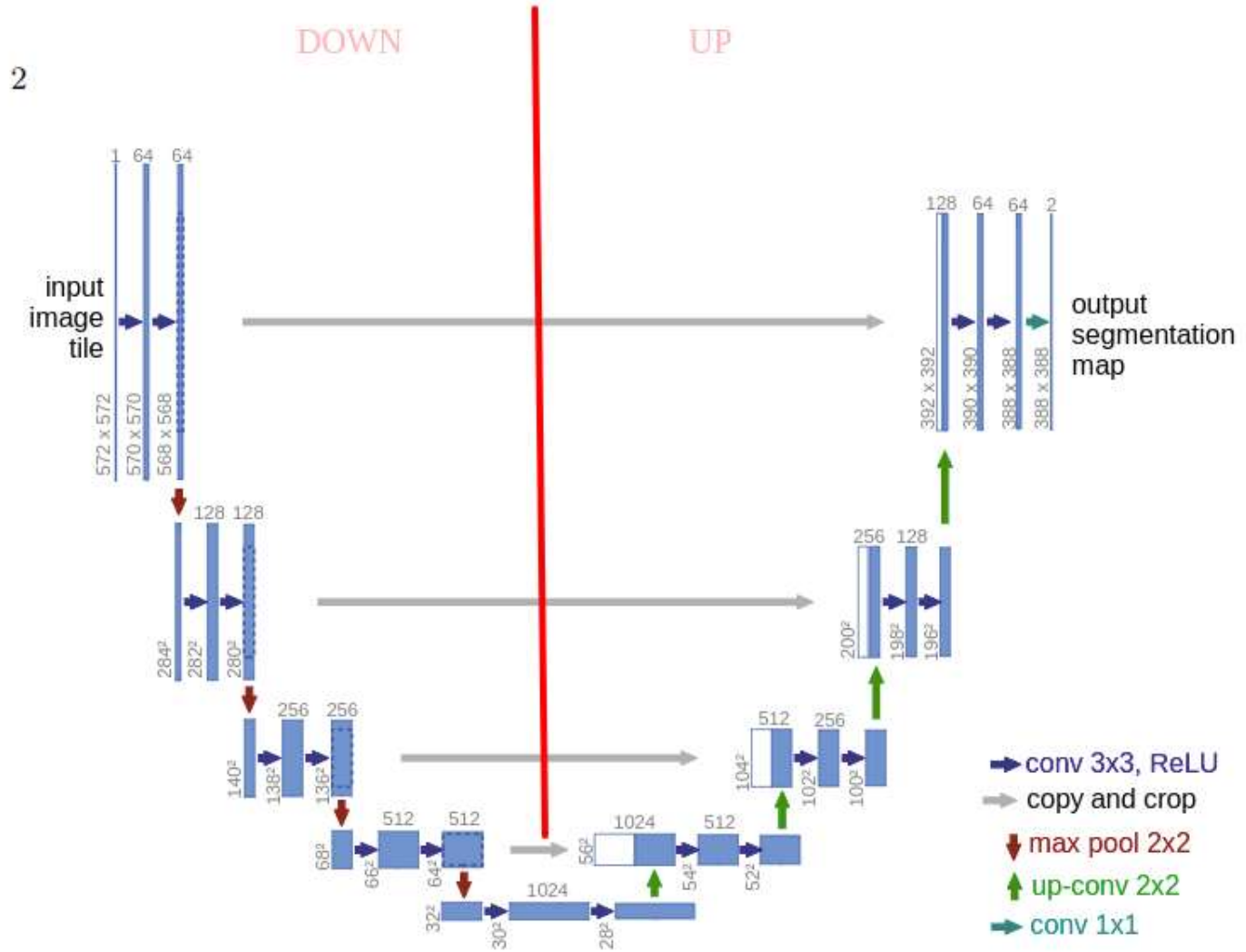


- Доведение идеи encoder-decoder до конца
  - По всему изображению строим вектор признаков 1x1
  - Используя сохранённые max-pooling индексы повышаем разрешение до исходного, предсказывая карту разметки

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. ICCV 2015

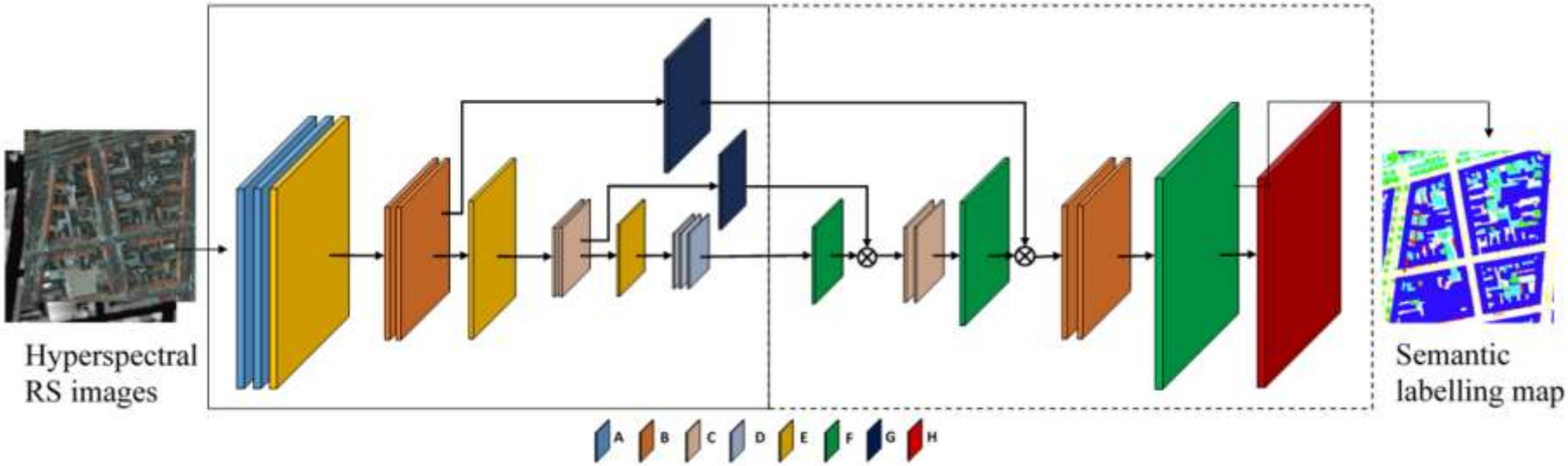
U-Net

- Добавим обходные пути (skip connections)



Hourglass

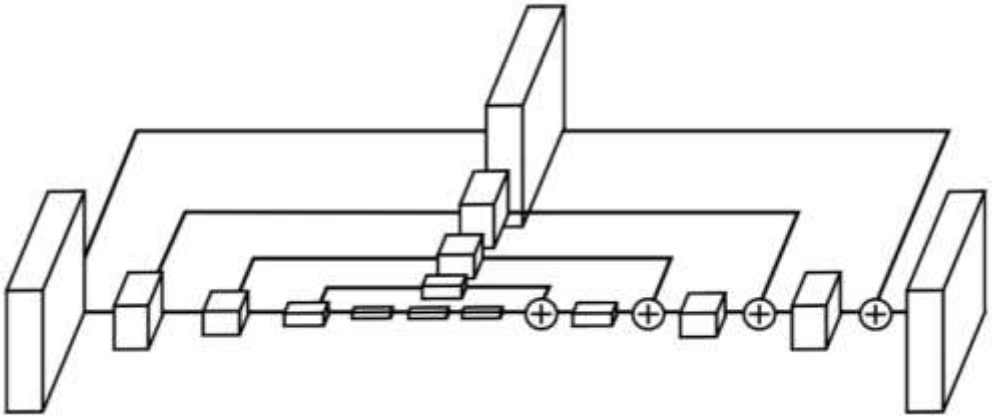
“Residual” модули на skip-connections

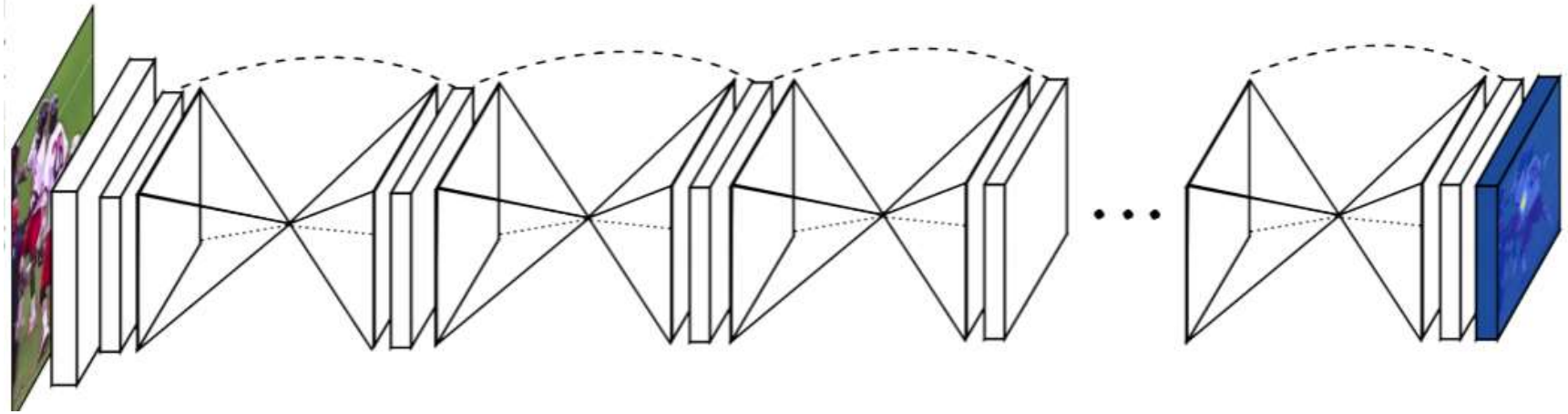


- A, B - convolutional layers
- C, D - inception modules;
- E - max pooling layer
- F - transposed convolutional layer
- G - residuals modules
- H - loss layer.

Stacked hourglass

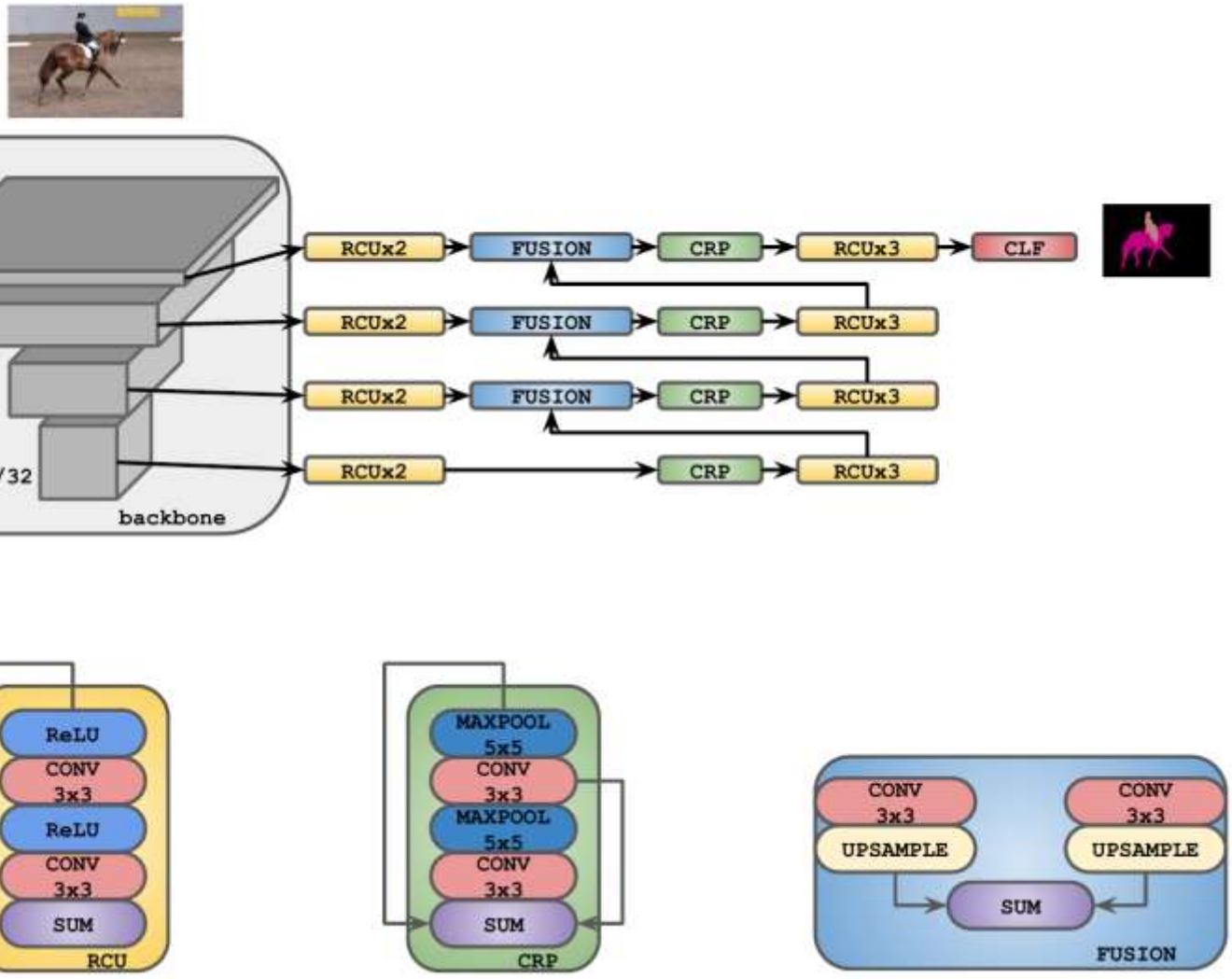
Объединим модули в цепочку (каскад)



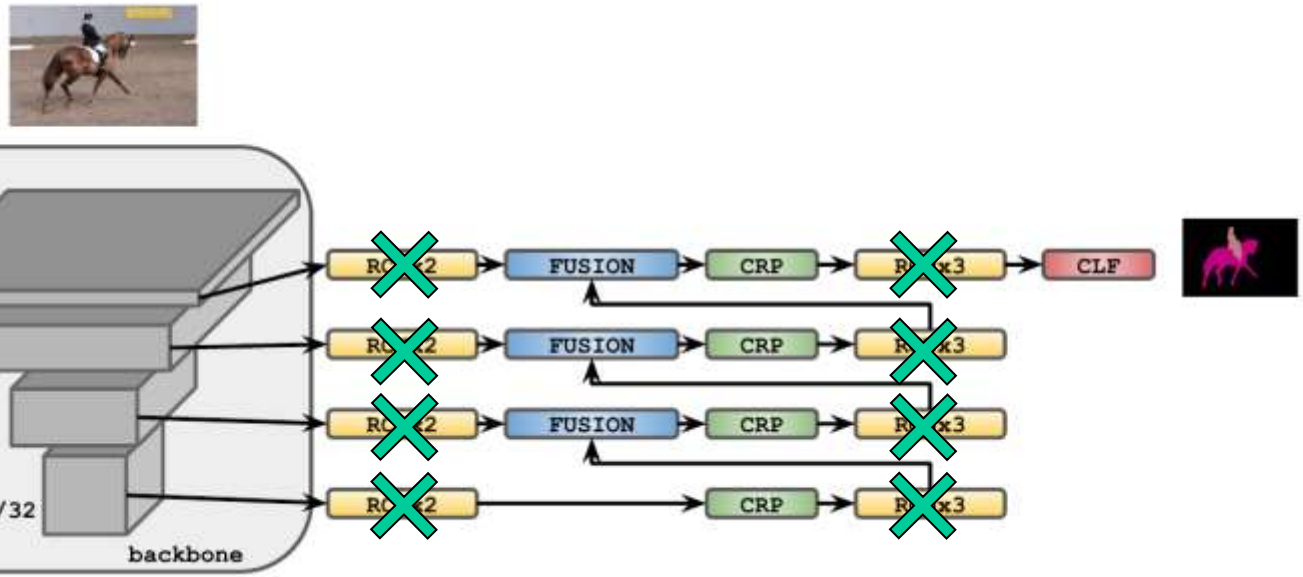


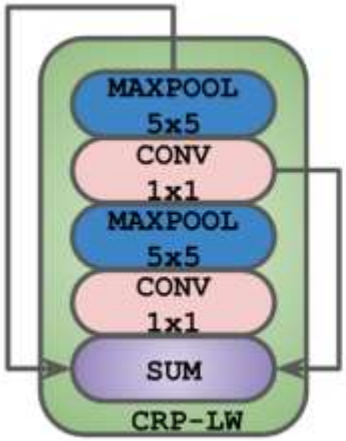
https://arxiv.org/pdf/1603.06937.pdf

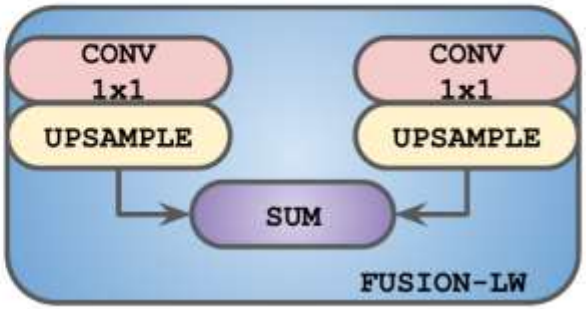
RefineNet


1/4
1/8
1/16
1/32
backbone
RCUx2
FUSION
CRP
RCUx3
CLF
RCUx2
FUSION
CRP
RCUx3
RCUx2
FUSION
CRP
RCUx3
RCUx2
CRP
RCUx3
ReLU
CONV 3x3
ReLU
CONV 3x3
SUM
RCU
MAXPOOL 5x5
CONV 3x3
MAXPOOL 5x5
CONV 3x3
SUM
CRP
CONV 3x3
CONV 3x3
UPSAMPLE
UPSAMPLE
SUM
FUSION

Light RefineNet







Кое-что ещё про сегментацию

Deep GrabCut

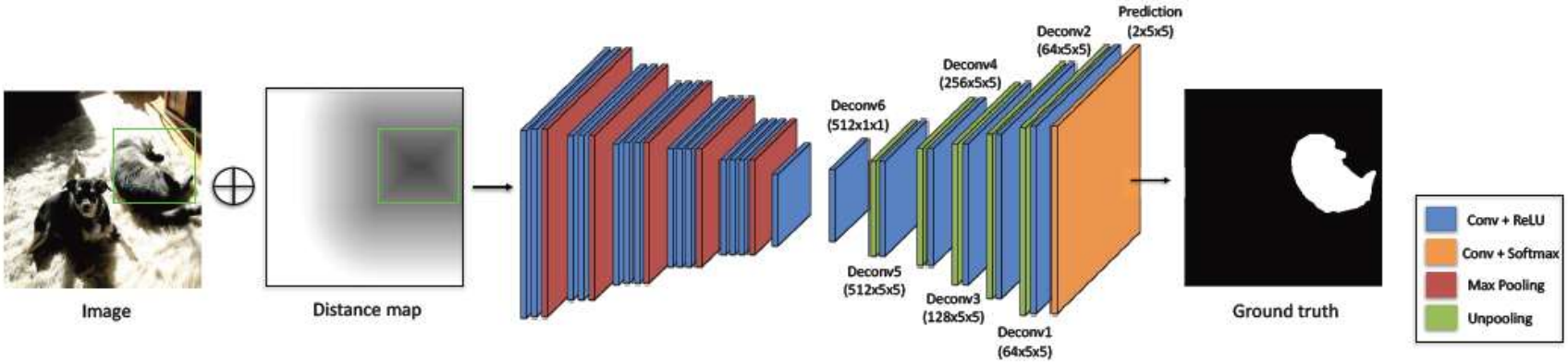


Figure 2: The framework of our segmentation model. The rectangle is indicated in green in the “Image" and “Distance map". The symbol $\oplus$ denotes the concatenation operation.

Mask R-CNN для Instance Segmentation

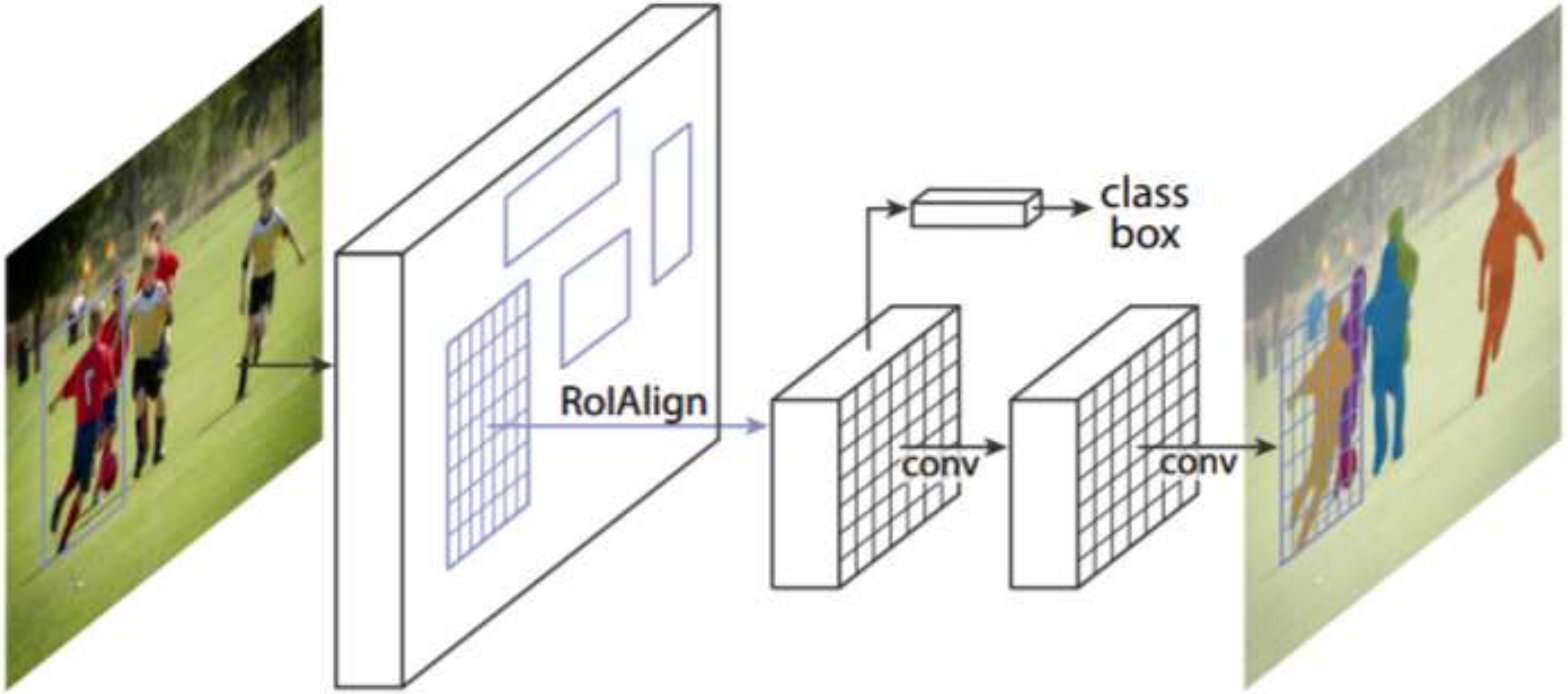


Figure 1. The **Mask R-CNN** framework for instance segmentation.

Mask R-CNN

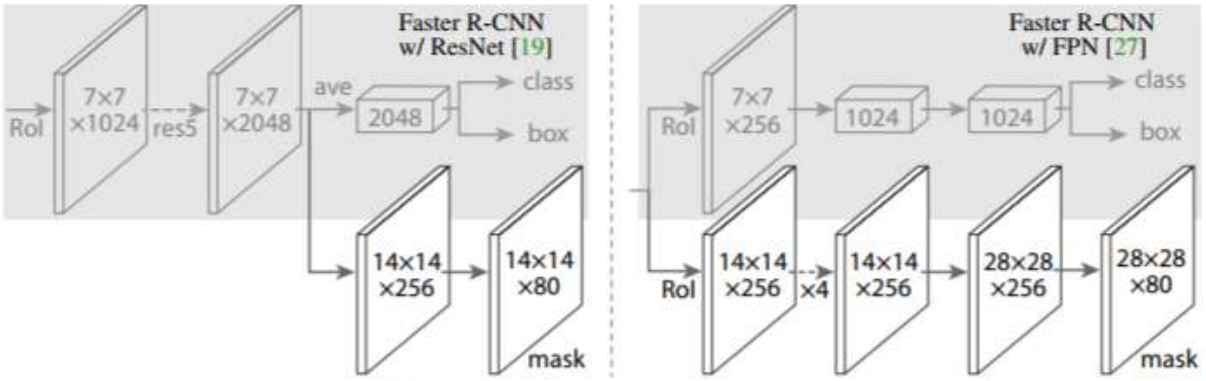


Figure 3. **Head Architecture**: We extend two existing Faster R-CNN heads [19, 27]. Left/Right panels show the heads for the ResNet C4 and FPN backbones, from [19] and [27], respectively, to which a mask branch is added. Numbers denote spatial resolution and channels. Arrows denote either conv, deconv, or *fc* layers as can be inferred from context (conv preserves spatial dimension while deconv increases it). All convs are 3×3, except the output conv which is 1×1, deconvs are 2×2 with stride 2, and we use ReLU [30] in hidden layers. *Left*: 'res5' denotes ResNet's fifth stage, which for simplicity we altered so that the first conv operates on a 7×7 RoI with stride 1 (instead of 14×14 / stride 2 as in [19]). *Right*: '×4' denotes a stack of four consecutive convs.

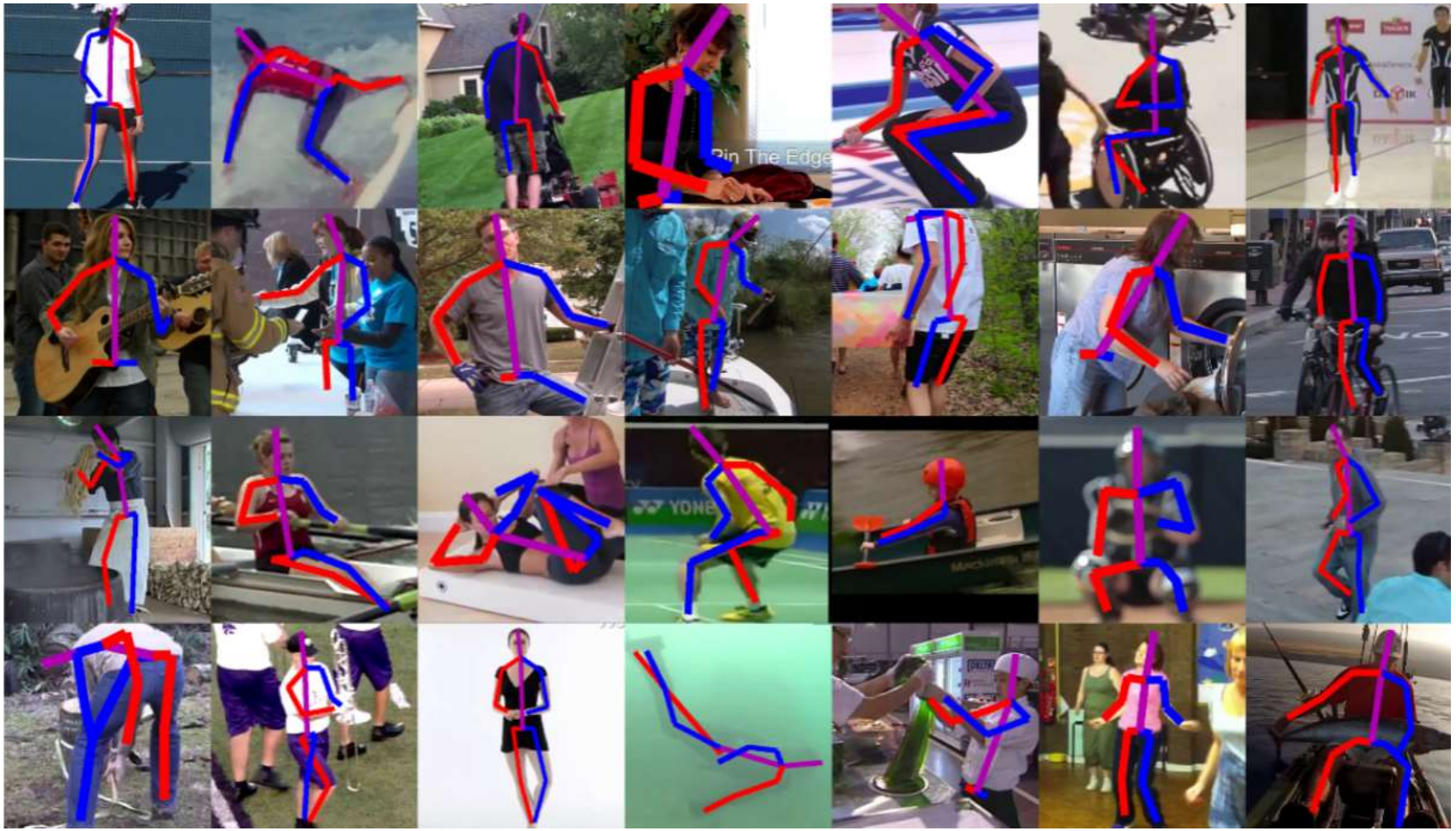
Результаты

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

Поза человека через сегментацию

Fig. 2. Example output produced by our network. On the left we see the final pose estimate provided by the max activations across each heatmap. On the right we show sample heatmaps. (From left to right: neck, left elbow, left wrist, right knee, right ankle)

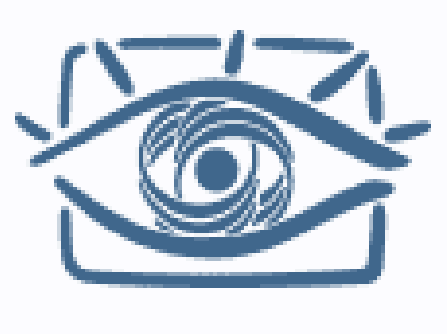
Примеры работы



Резюме лекции

- Интерактивная сегментация
- Визуальная сегментация
- Семантическая сегментация
- Instance segmentation
- Поза человека через сегментацию

- Текущее развитие
  - Multi-task learning (сегментация + что-то ещё)
  - Новые loss + небольшие изменения архитектуры



Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №9

«Перенос стиля и синтез изображений»

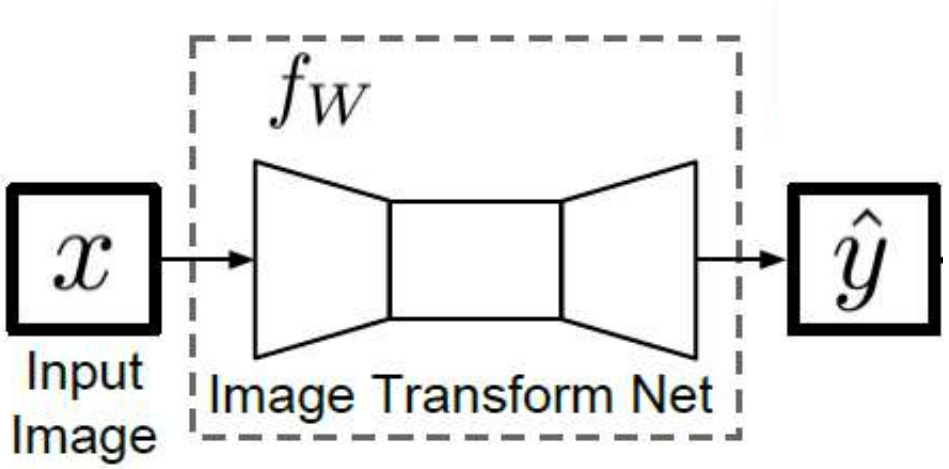
Антон Конушин

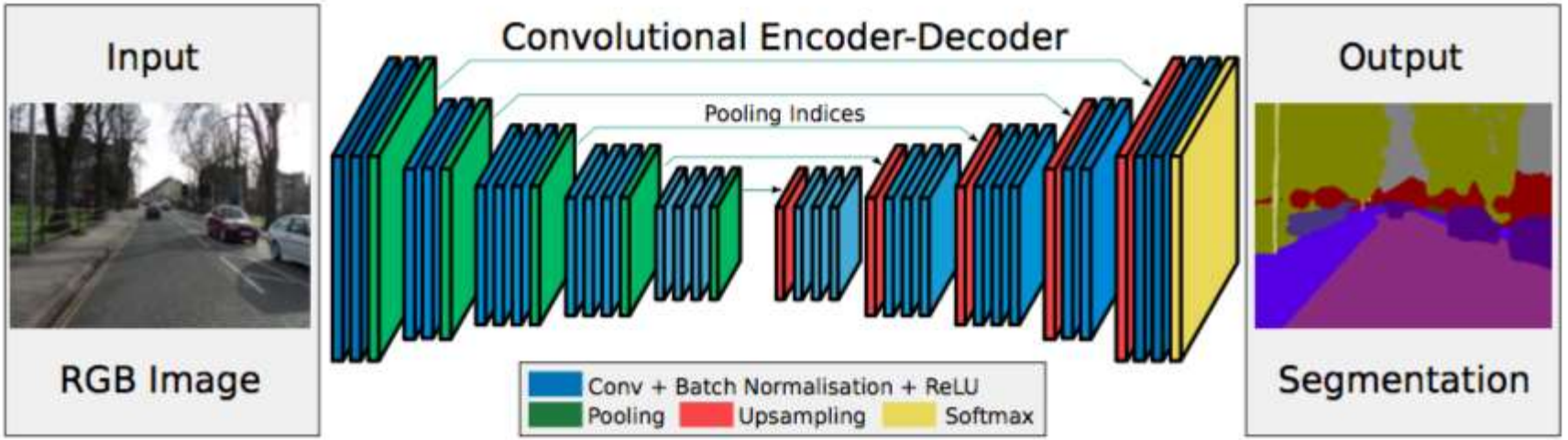
Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

18 ноября 2019 года

Модификация изображений

Пример решения - преобразователи



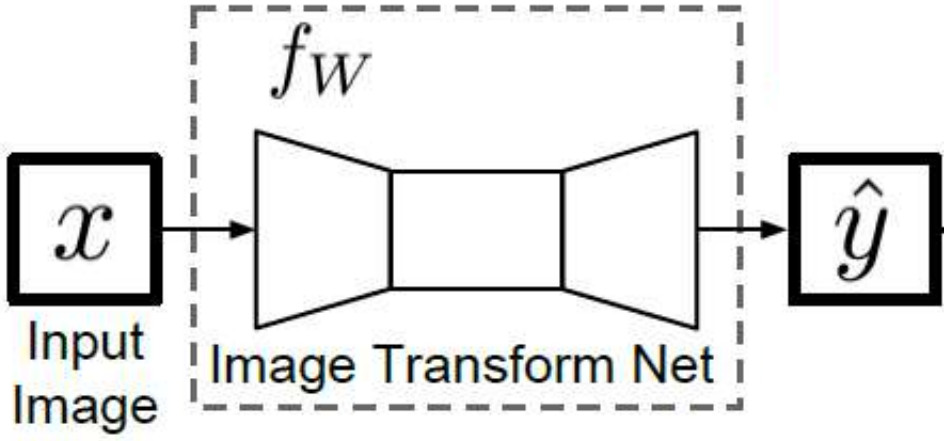


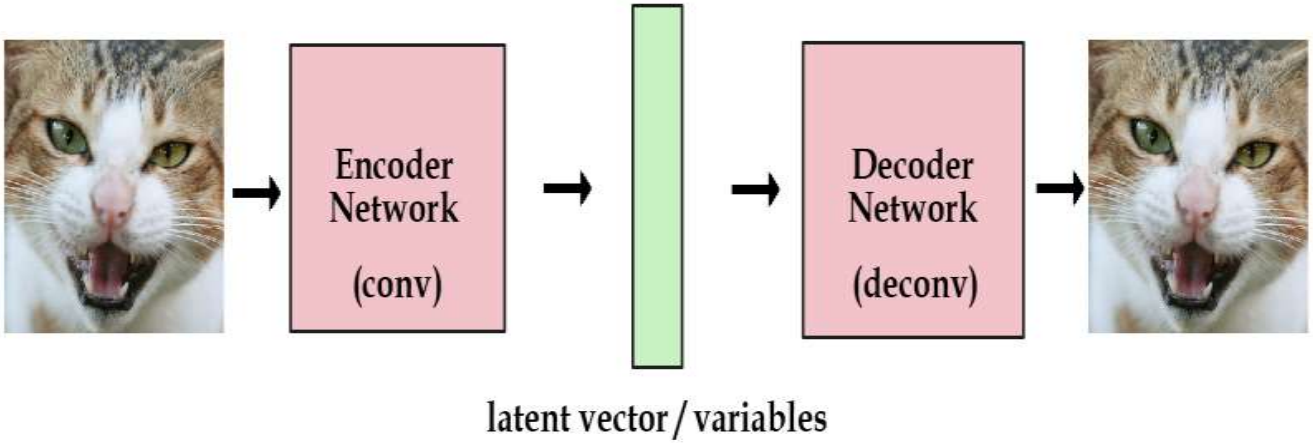
Перенос стиля изображения

+

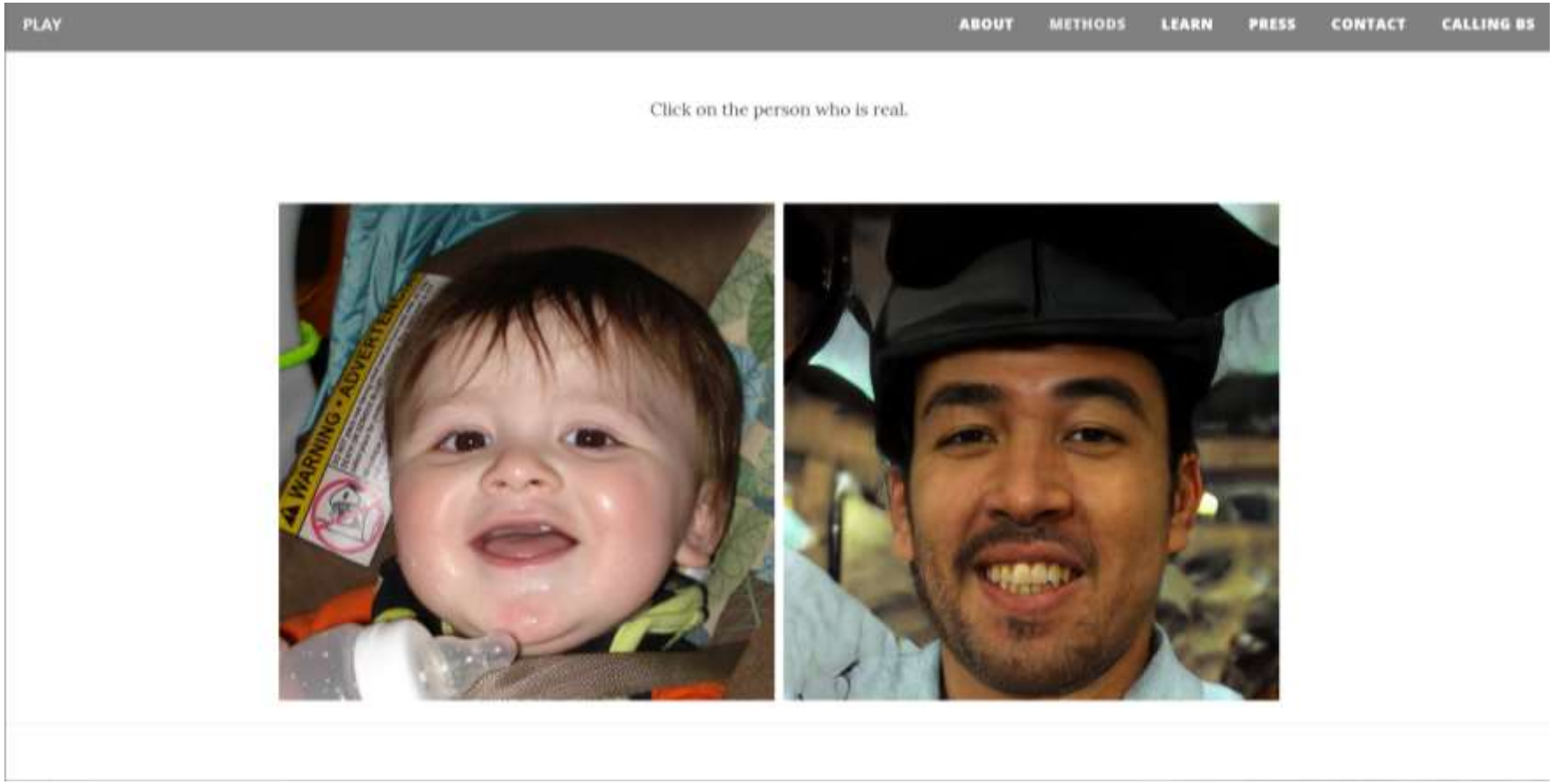
=

Подходы к решению





Синтез изображений

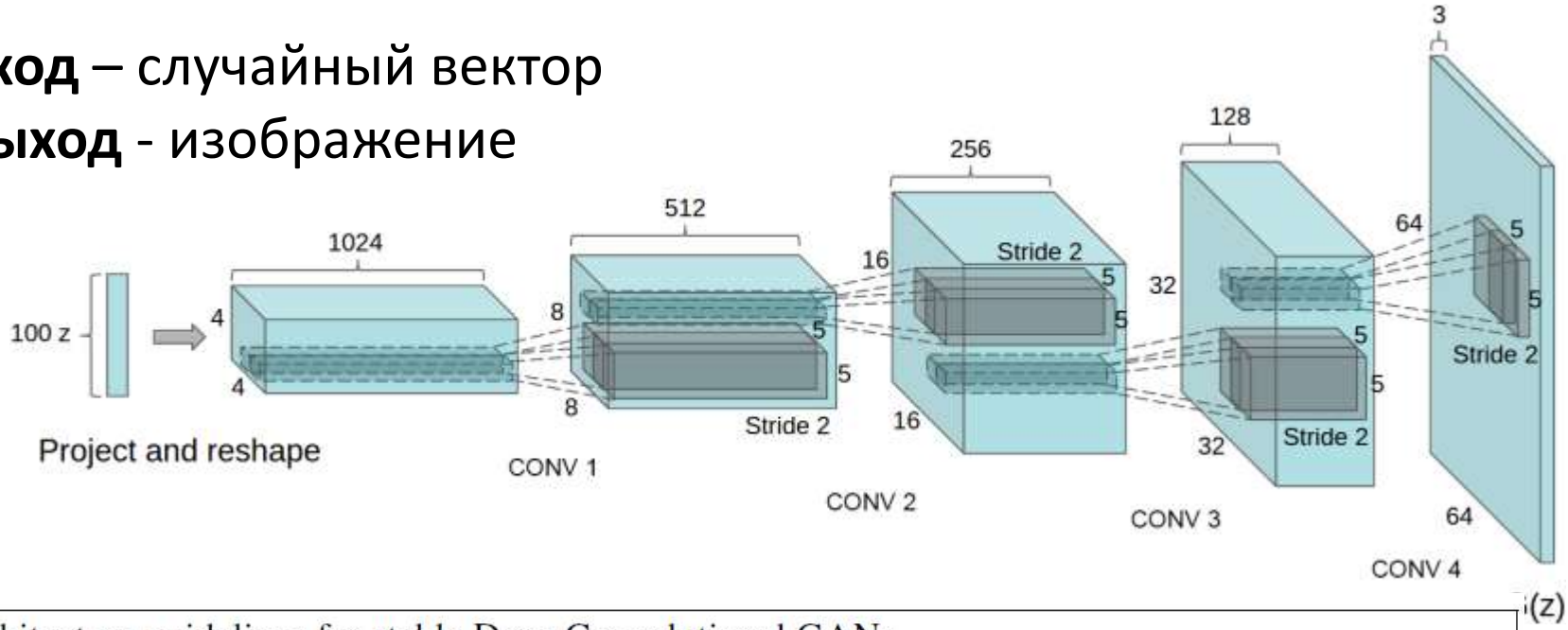


http://www.whichfaceisreal.com/

Пример решения - DCGAN

Вход – случайный вектор

Выход - изображение



Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

A.Radford, L. Metz, S. Chintala UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS. ICLR 2016

Визуализация вектор-признака

Найдём такое изображение x, которое даёт такой же вектор-признак $\Phi_0 = \Phi(x_0)$ от изображения x_0

Mahendran and Vedaldi. Understanding Deep Image Representations by Inverting Them, 2014

Визуализация изображения по выходам

Процедура поиска:

- Инициализируем x белым шумом
- Будем оптимизировать по x следующий функционал l :

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

$R(x)$– регуляризатор, например, Total Variation (TV):

$$R(x) = \sum_{i,j} \left(\left(x_{i,j+1} - x_{i,j} \right)^2 + \left(x_{i+1,j} - x_{i,j} \right)^2 \right)$$

- Оптимизируем функционал градиентным спуском

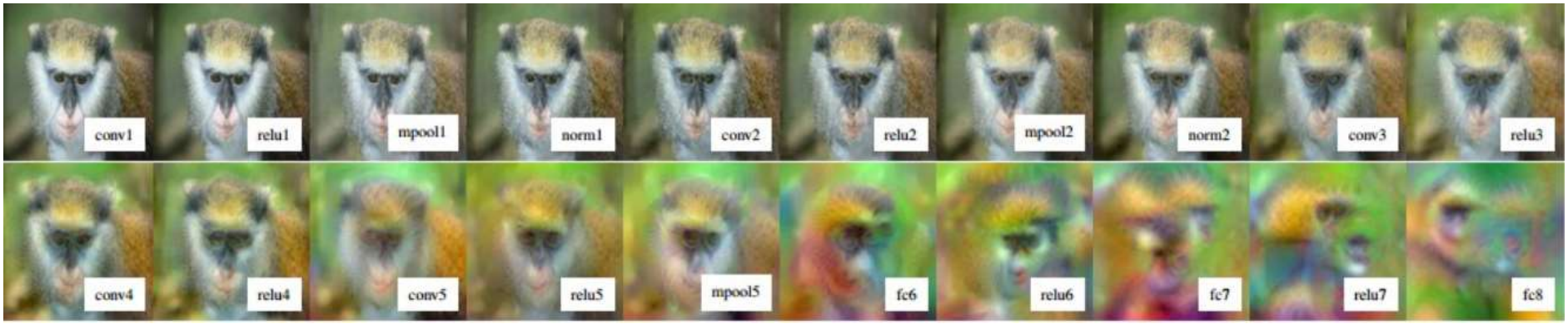
Реконструкция

С последнего свёрточного слоя

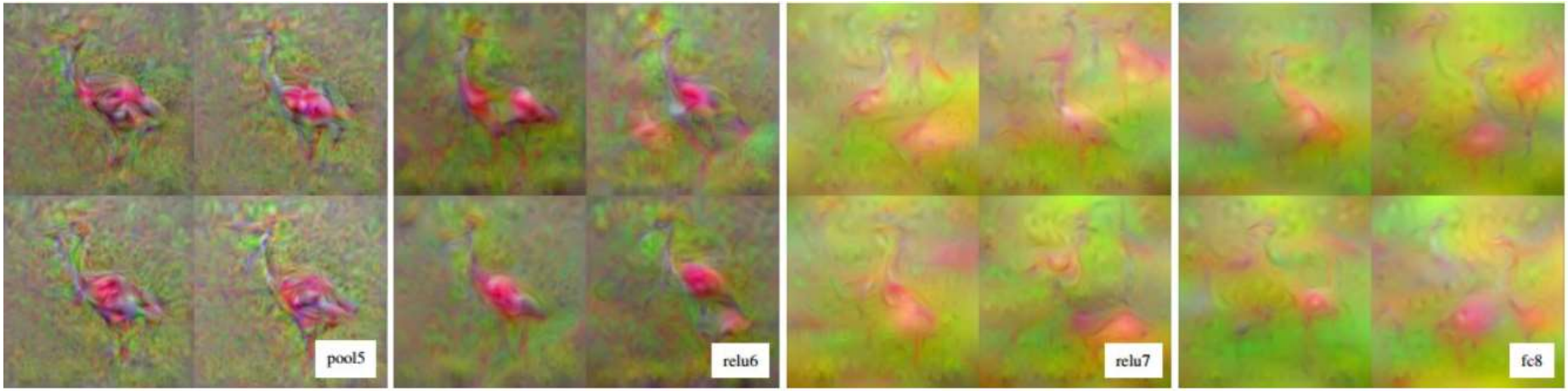
Пространственная информация во многом сохраняется

Реконструкция

С разных уровней



Множественные реконструкции (разные изображения с одинаковыми признаками)



Применим для переноса стиля

+

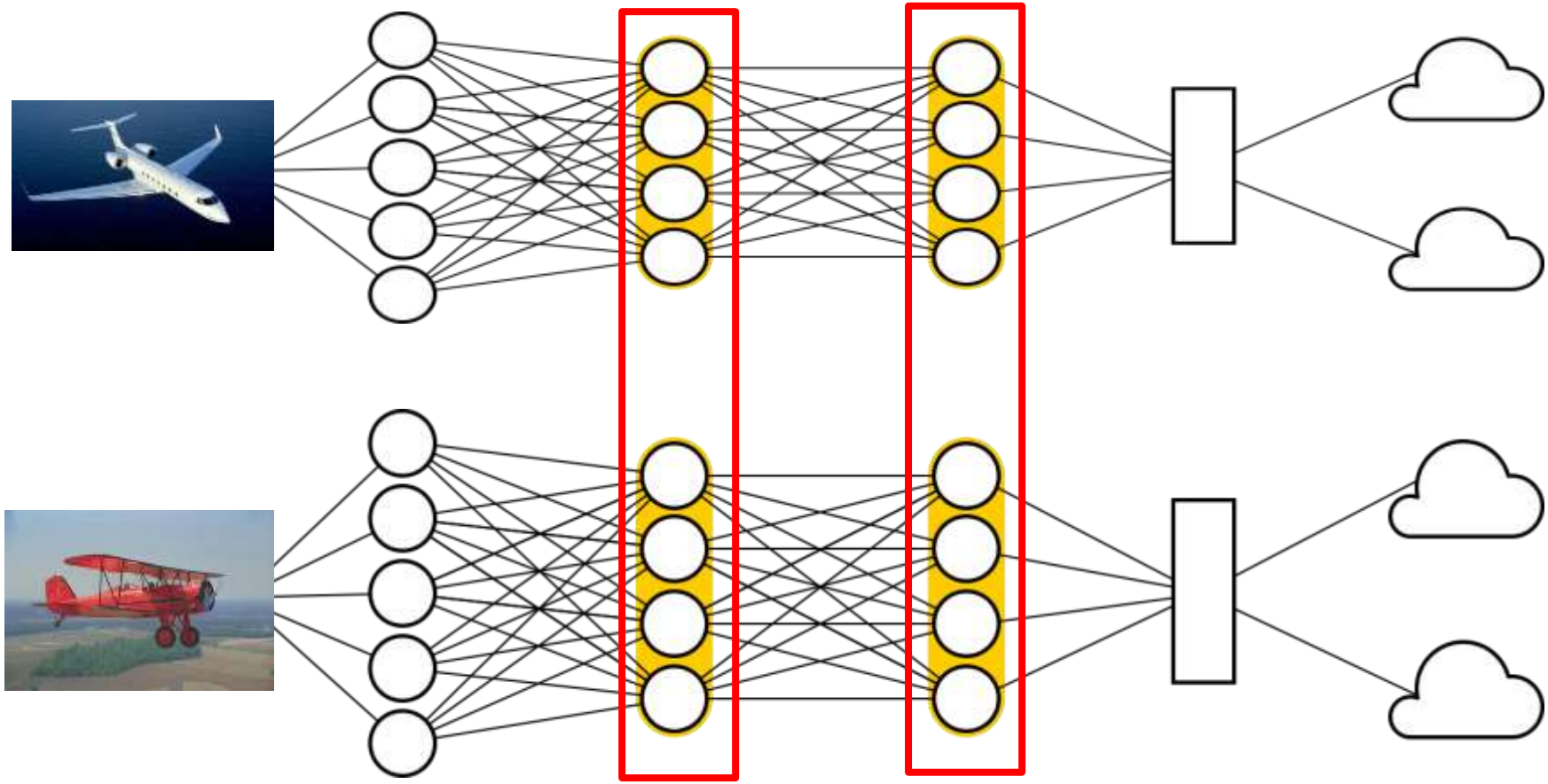
=

Простейшие методы переноса стиля

Базовые методы переноса цвета работают со свойствами всей картинки целиком. Пример такого метода – приравнивание среднего и дисперсии по каждому каналу в Lab

Основная идея – модификация изображения таким образом, чтобы по части признаков оно было похоже на исходное (содержание), а часть – на целевое (стиль)

Как описать похожесть по содержанию?



Perception loss

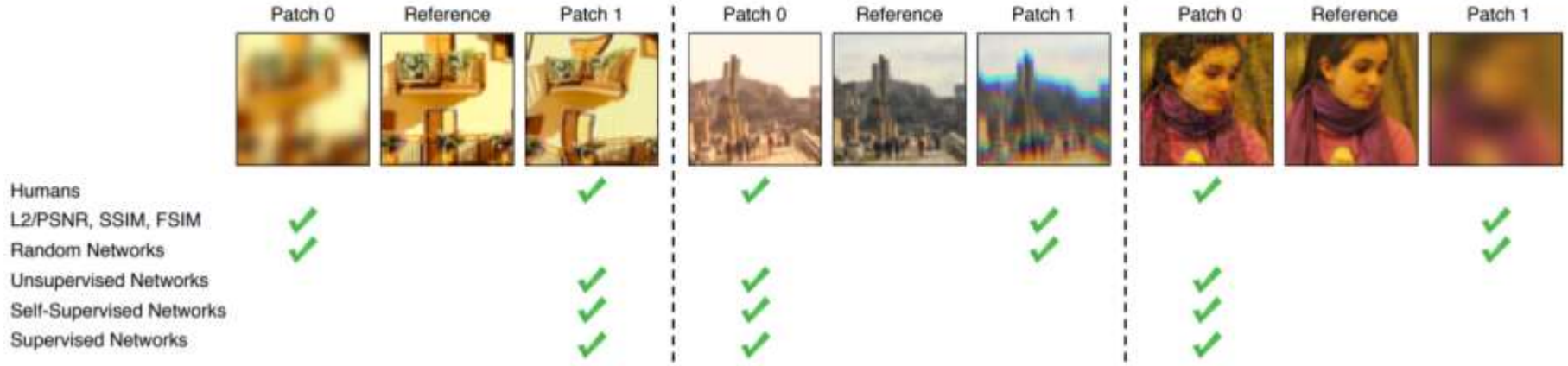


Figure 1: **Which patch (left or right) is "closer" to the middle patch in these examples?** In each case, the traditional metrics (L2/PSNR, SSIM, FSIM) disagree with human judgments. But deep networks, even across architectures (Squeezenet [20], AlexNet [27], VGG [52]) and supervision type (supervised [47], self-supervised [13, 40, 43, 64], and even unsupervised [26]), provide an *emergent embedding* which agrees surprisingly well with humans. We further calibrate existing deep embeddings on a large-scale database of perceptual judgments; models and data can be found at `https://www.github.com/richzhang/PerceptualSimilarity`.

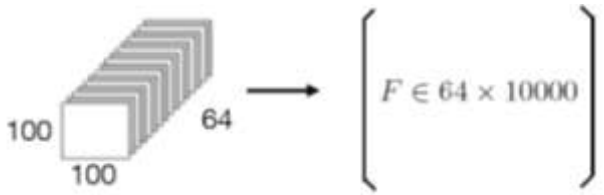
https://arxiv.org/pdf/1801.03924.pdf

Описание стиля

- За описание «стиля» можно взять корреляцию откликов разных фильтров по всему изображению
- Можно вычислить по признакам первых слоёв «стиль» и попробовать реконструировать изображение с тем же стилем

Реконструкция стиля

- Стиль можно описать корреляцией откликов фильтров, записав матрицу Грама $G^l \in \mathcal{R}^{N_l \times N_l}$
- Где G^l_{ij} вычисляется как скалярное произведение откликов i-го и j-го фильтров:



$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

- Генерируем изображения со стилем исходного изображения, минимизируя среднеквадратичную разницу между матрицами Грама исходного изображения G и сгенерированного изображения A (или суммами по L первых слоёв)

$$E_l = \frac{1}{Norm} \sum_{i,j} (G^l_{ij} - A^l_{i,j})^2$$

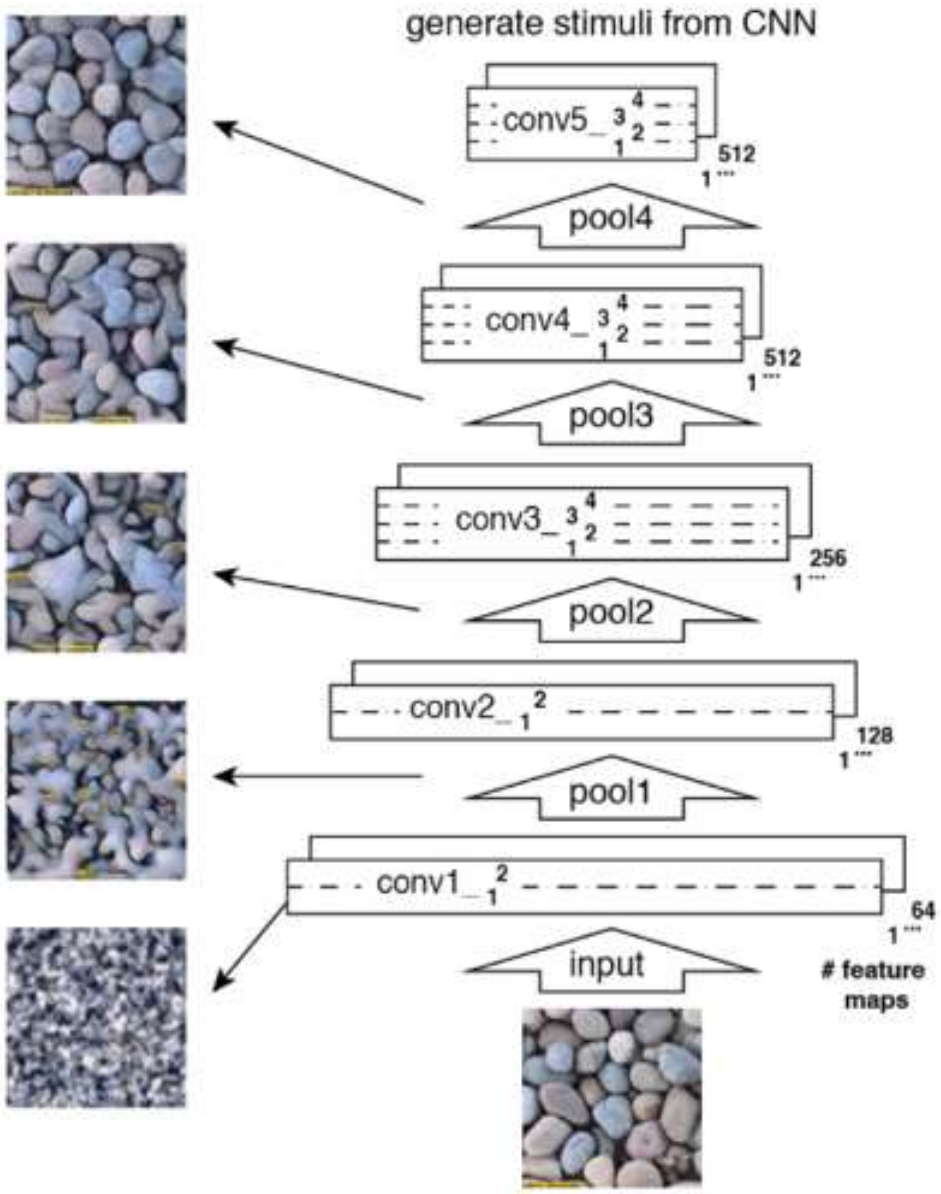
Cost for style reconstruction

$$Loss_{style} = \sum_{l=0}^{L} w_l E_l$$

Accumulate cost for lower layers

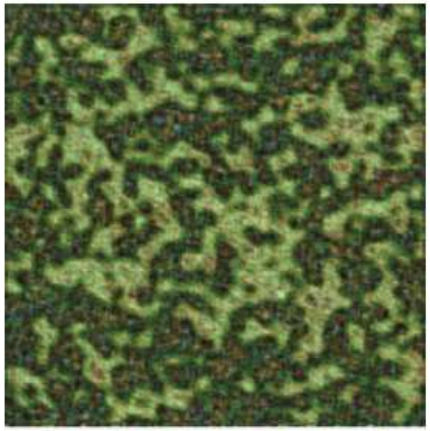
A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. Portilla, J., & Simoncelli, E. P. *Int. J. Comput. Vis.* 40, 49-70 (2000)
Texture Synthesis and The Controlled Generation of Natural Stimuli Using Convolutional Neural Networks. *Gatys, L. A., Ecker, A. S., & Bethge, M. NIPS (2015)*

Реконструкция текстур

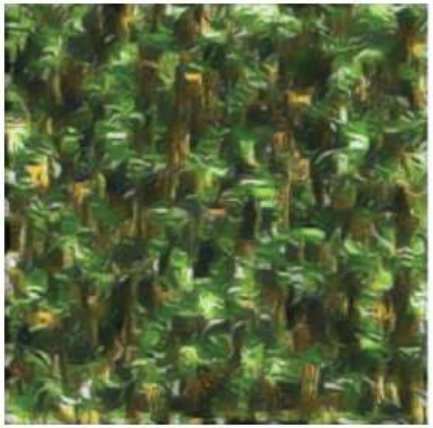


Пример реконструкции стиля

Original



Up to *Conv1_1* layer



Up to *Pool1* layer

Up to *Pool2* layer

Up to *Pool3* layer

Up to *Pool4* layer

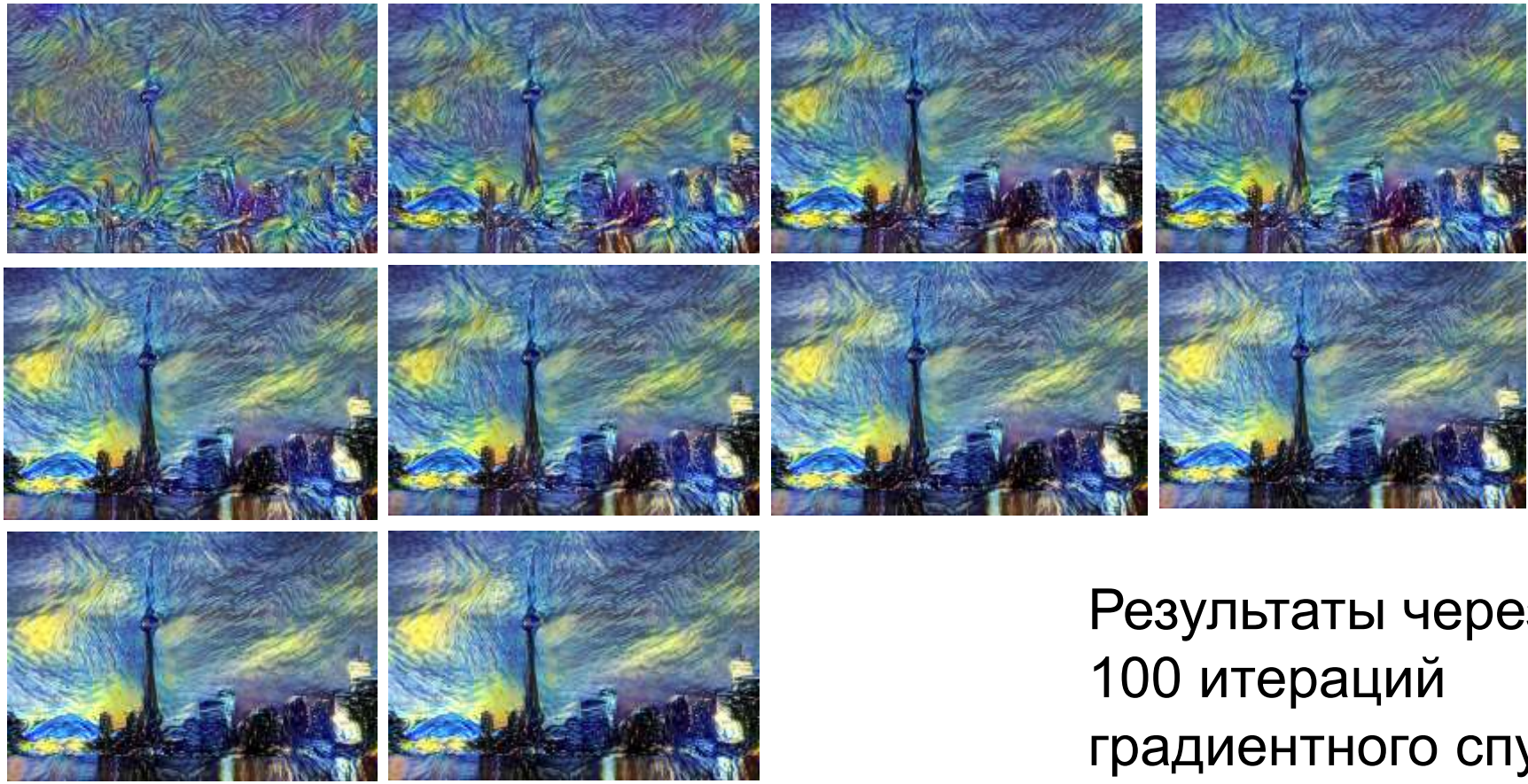
Ключевое наблюдение

- Содержание изображение и стиль оказываются разделимы
- Верхние слои целиком больше описывают содержание изображения
- Корреляция карт нижних слоёв – стиль изображения
- Можем сгенерировать изображения, начав с белого шума, минимизировав градиентным спуском функционал:

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).

Визуализация работы



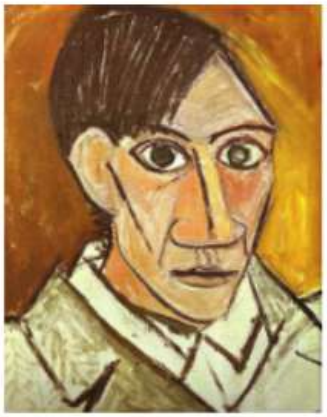
Результаты через
100 итераций
градиентного спуска

Пример работы

Пример работы

Соотношение стиля и содержания

Used for *Content*



Used for *Style*

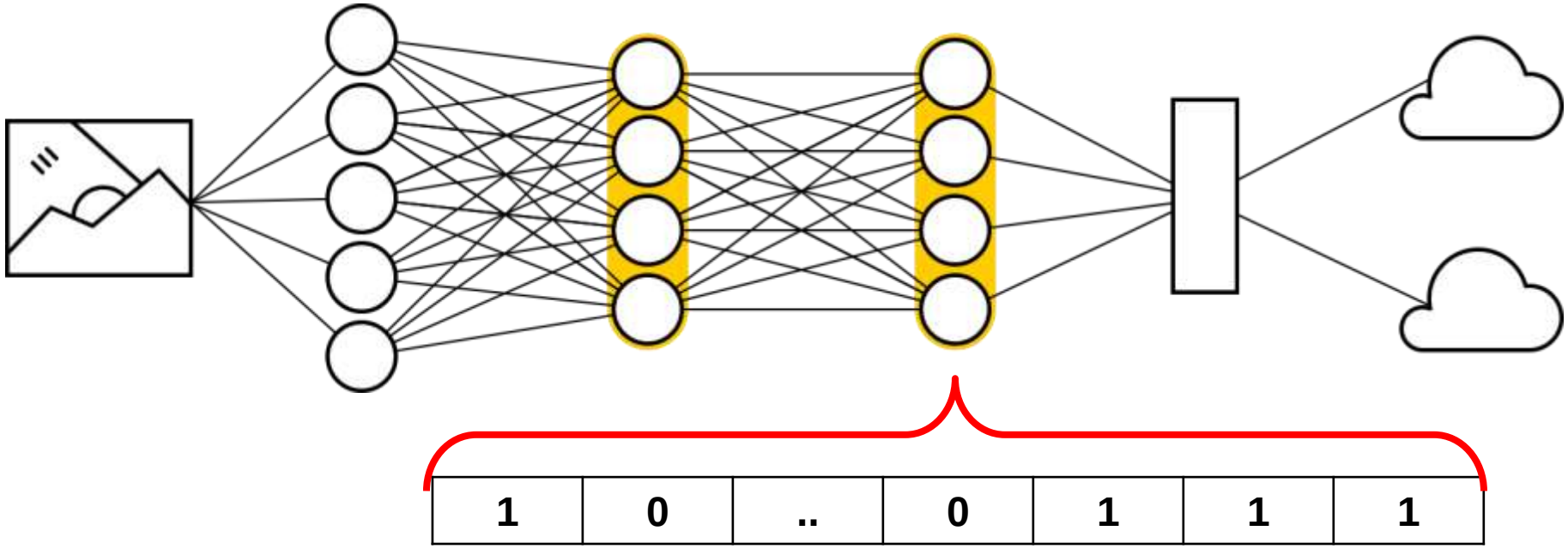
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Decrease α/β

Deep Feature Interpolation

P. Upchurch et. al. Deep Feature Interpolation for Image Content Changes, CVPR 2017

Идея – манипуляция признаками



- Глубокие нейросети показывают отличные результаты на image classification, используя простые линейные классификаторы
- Значит признаковое пространство получается очень хорошее
- М.б. даже Евклидово?
- Можно работать напрямую в признаковом пространстве

Схема алгоритма

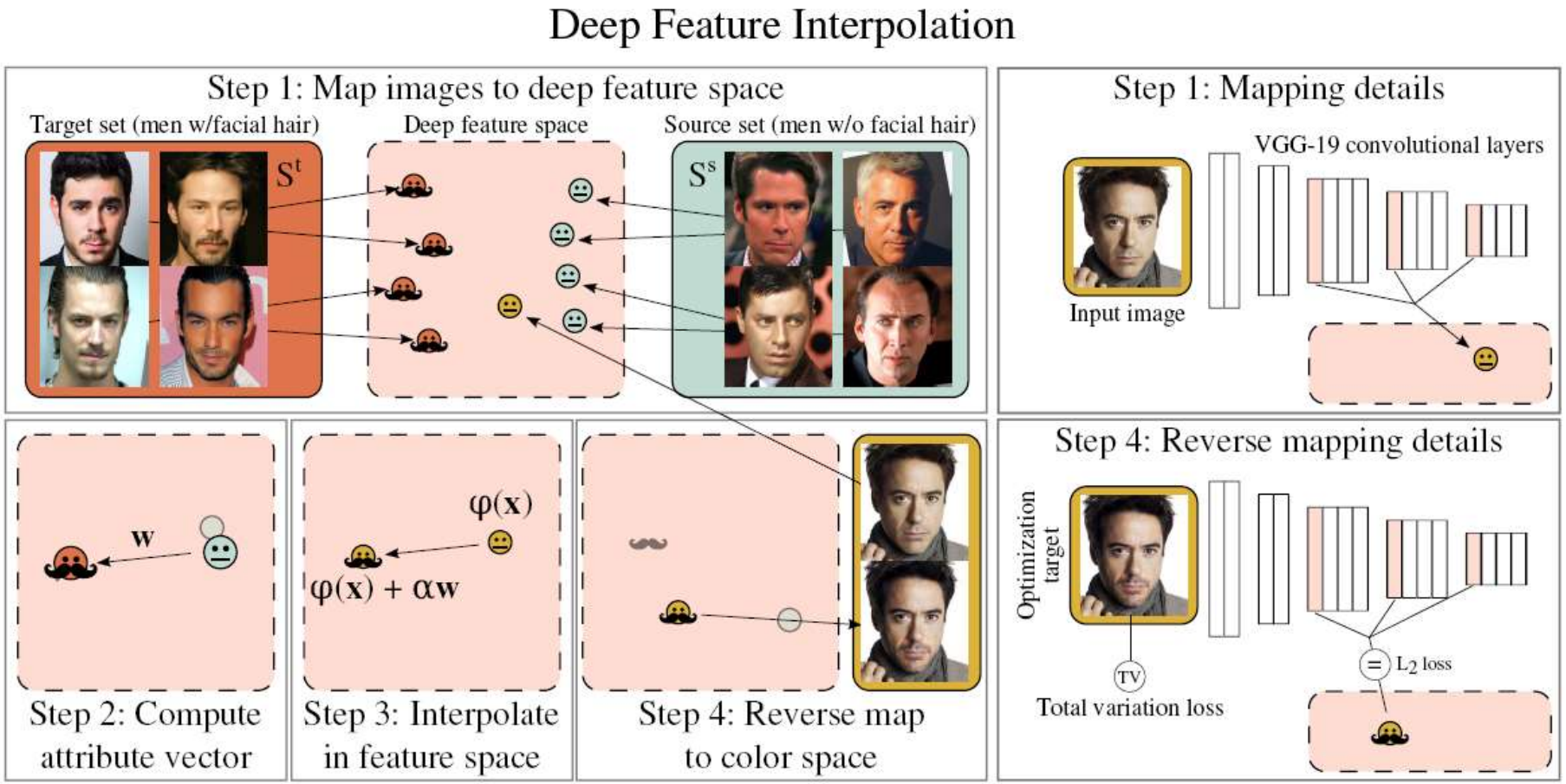
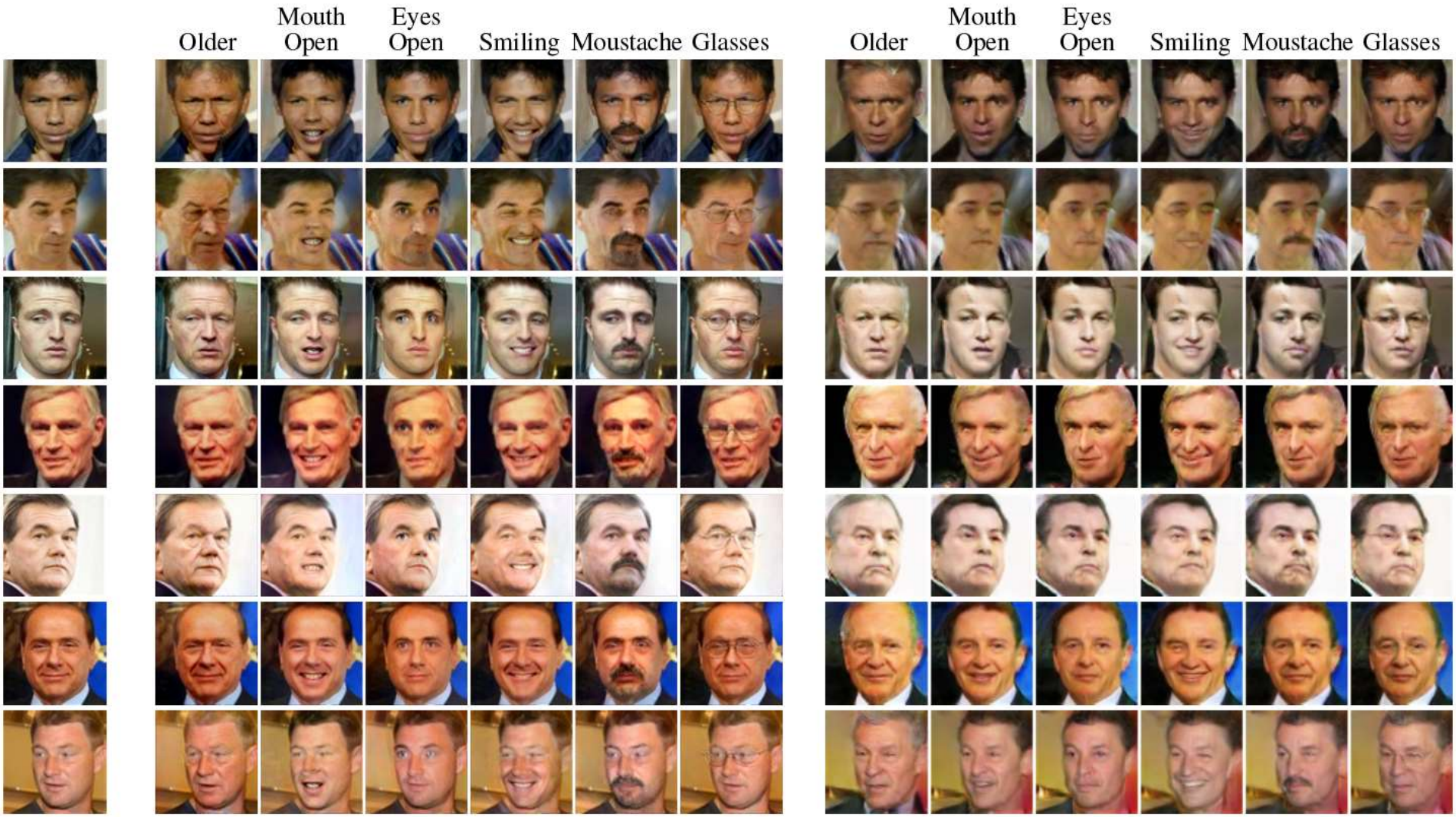


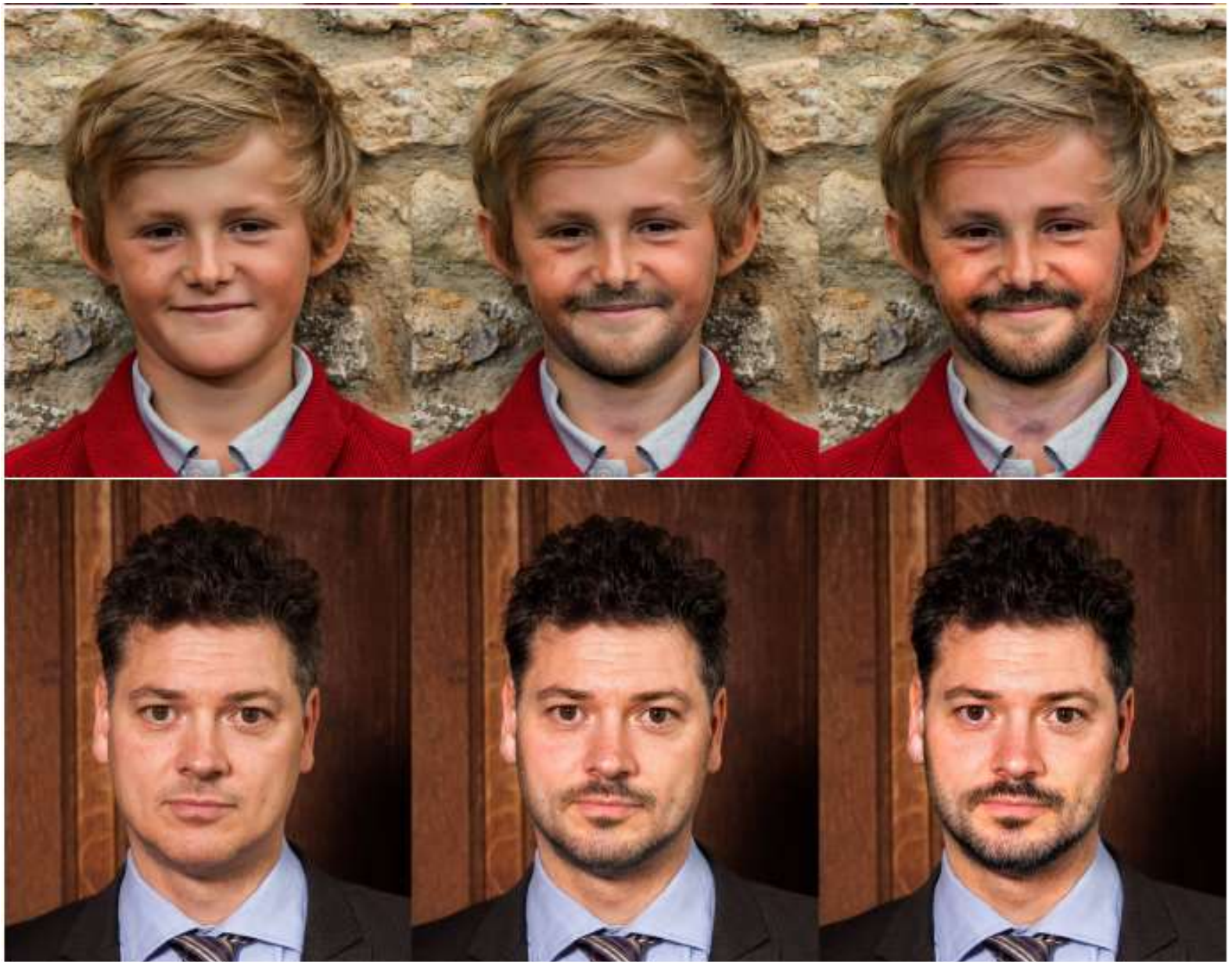
Figure 2. A schematic outline of the four high-level DFI steps.

Используем обычные VGG предобученные на ImageNet

Примеры работы



Пример работы на высоком разрешении



Варьирование параметра

Original | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.4$ | $\beta = 0.6$ | $\beta = 0.8$ | $\beta = 1.0$

Ещё пара моментов

Выравнивание лиц и других атрибутов (мужчина, Caucasian)

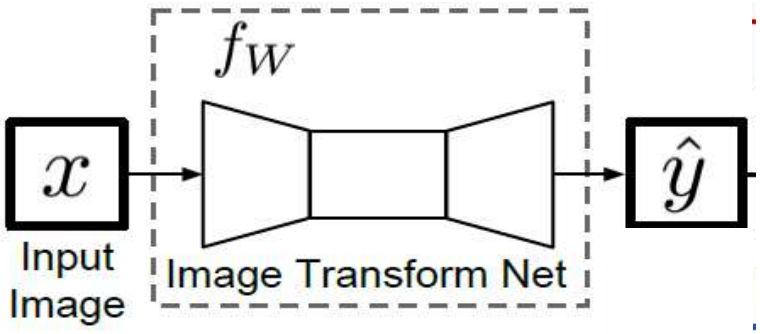
- Низкое разрешение – выравнивание глаз и рта на этапе предобработки коллекции
- Высокое разрешение – подгон лиц к тестовому перед сбором source & target набором для расчета вектора преобразования

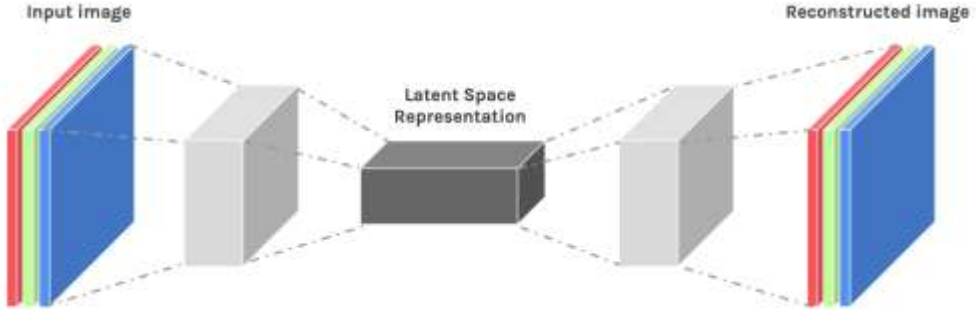
Тип используемой сети для PerceptionLoss

- VGG-face работала хуже VGG на ImageNet. «Внешность», включая атрибуты, она хуже передает
- Если мы хотим сохранить «личность», то можем использовать IdentityLoss, считая признаки на специальной сети, используемой для идентификации объектов/человека

Стилизация с Perception Loss

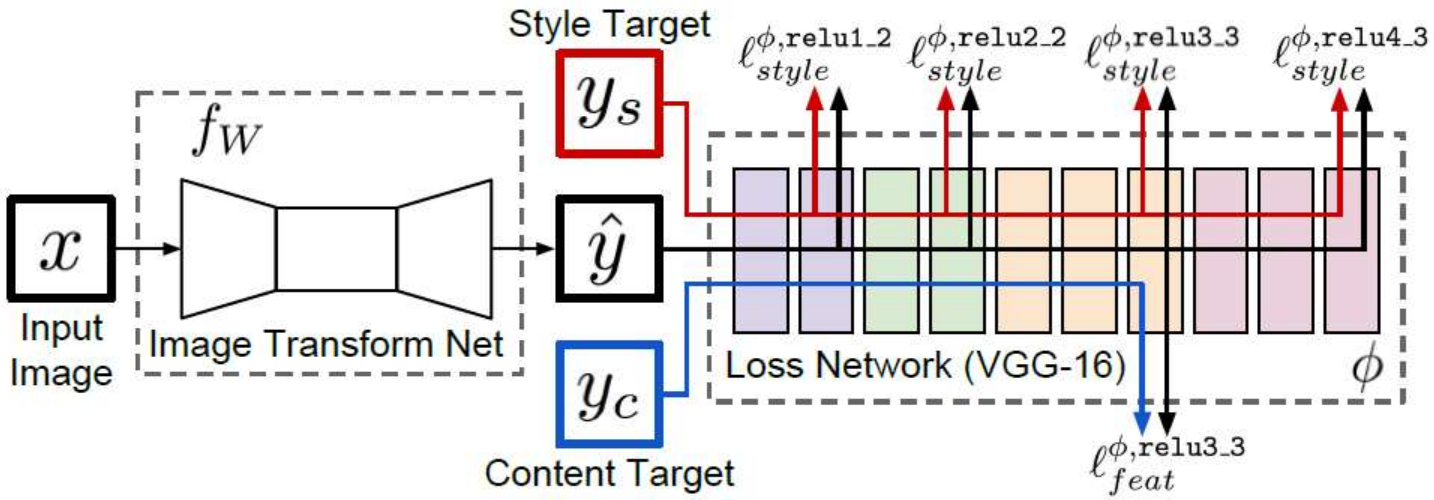
- Оптимизация с белого шума идёт медленно
- Идея: воспользоваться сетью-преобразователем
- Оценивать качество будем через perception loss
- Для каждого стиля нужно обучать свою собственную нейросеть





Johnson et. al. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ECCV 2016

Обучение такой сети



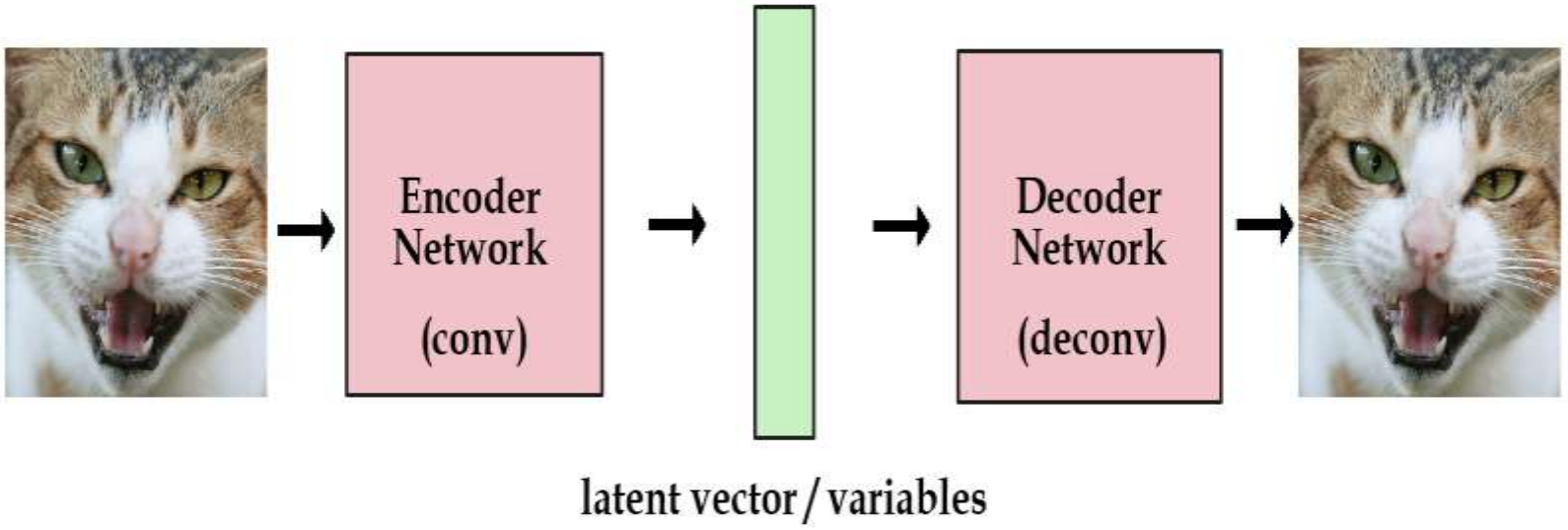
- Воспользуемся предобученной на задаче классификации на imagenet сетью VGG-16
- Будем использовать её для извлечения признаков, и она будет зафиксирована при обучении трансформационной сети
- Через неё прогоняем изображения y_s (стиля), и y_c=x (содержание)

The Muse,
Pablo Picasso,
1935

Original Gatys et al. Ours

Universal Style Transfer [Li et al, 2017]

- Хотим научиться быстро переносить произвольный стиль с помощью сети – преобразователя изображений
- Обучим «автокодировщик» - сеть типа encoder-decoder, которая должна «сжимать» изображение в скрытое (latent) векторное представление F и декодировать его в исходное «без потерь»
- Будем манипулировать скрытым векторным представлением изображения F
- Обозначим признаки двух картинок F_c (содержание) и F_s (стиль). Пусть это матрицы размера [CxA], где C – количество признаков



Преобразование признаков

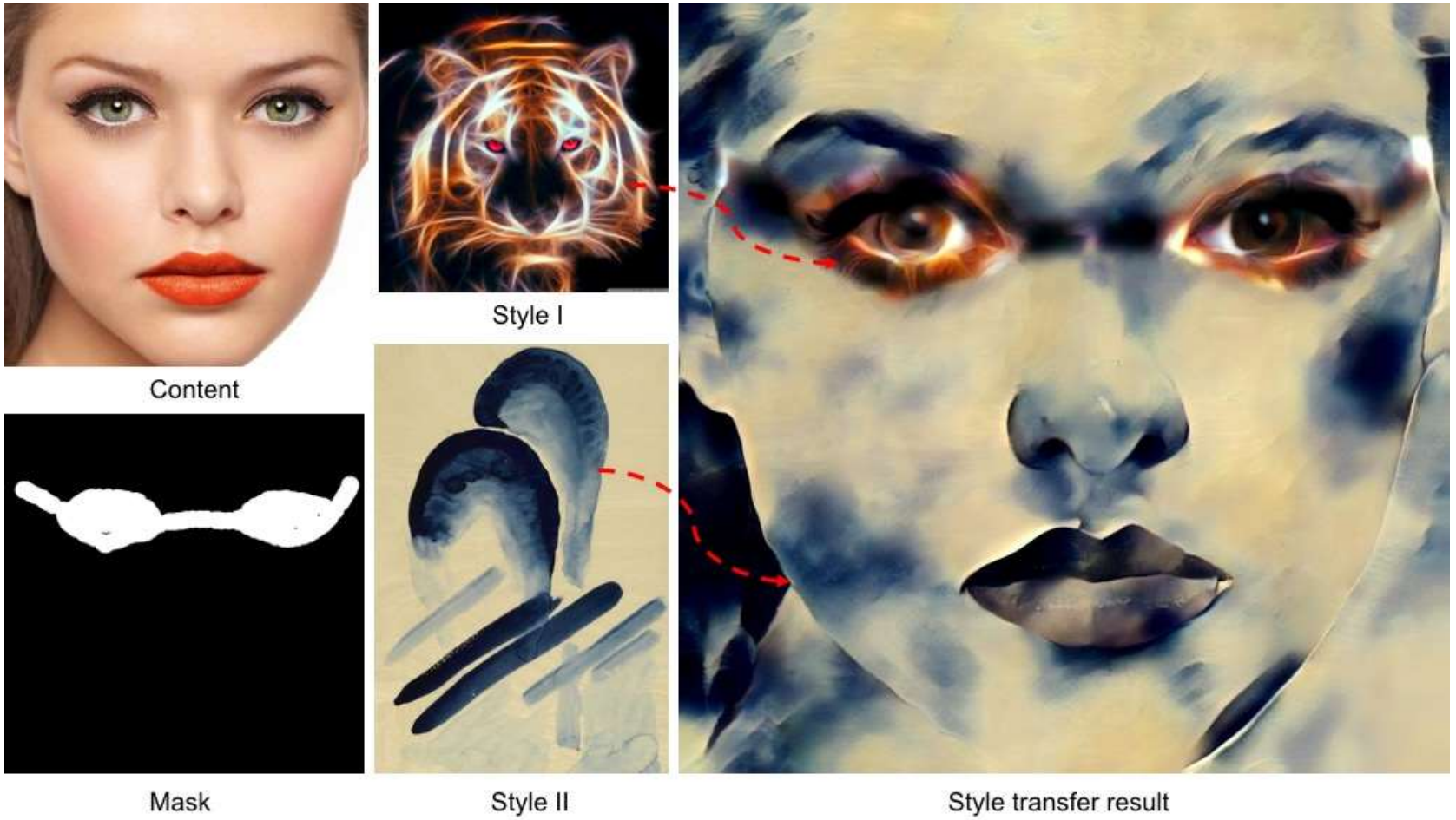
- Мы хотим, чтобы у новой картинки матрица Грама совпадала с $F_S F_S^T$
- Можно показать, что добиться цели можно через линейное преобразование вектор-признака

$FF^T = I$

$FF^T = F_S F_S^T$

- Делаем в 2 этапа:
  - сначала уберем корреляцию исходных признаков
  - затем приравняем нужные матрицы

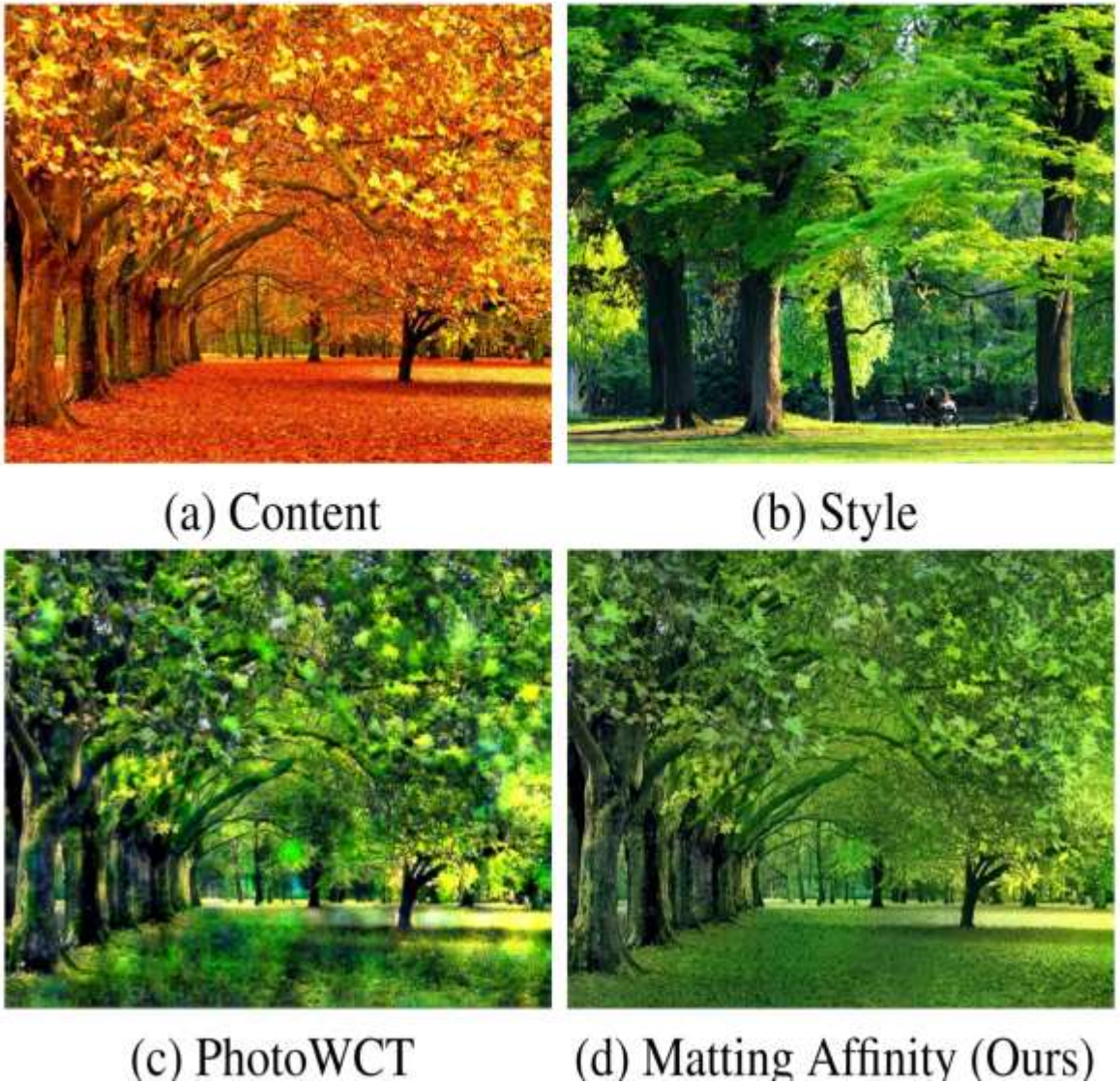
Смешение стилей



Генерация текстур

Замена изображения – содержания на шум

Модификация метода с пост-обработкой

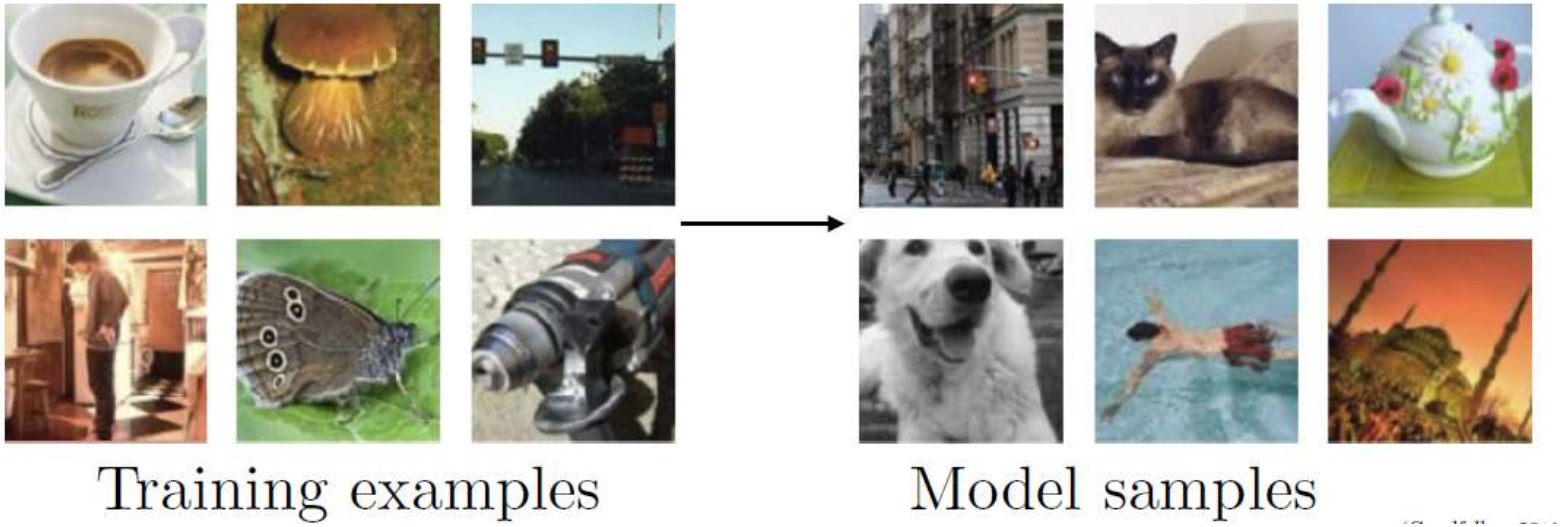


После преобразования стиля применим к изображению фильтр, пришедший из матирования изображений, который делает градиенты итогового изображения похожими на градиенты исходного, и тоже реализуемого в виде линейного преобразования

A Closed-form Solution to Photorealistic Image Stylization [Li et al, 2018]

Синтез изображений

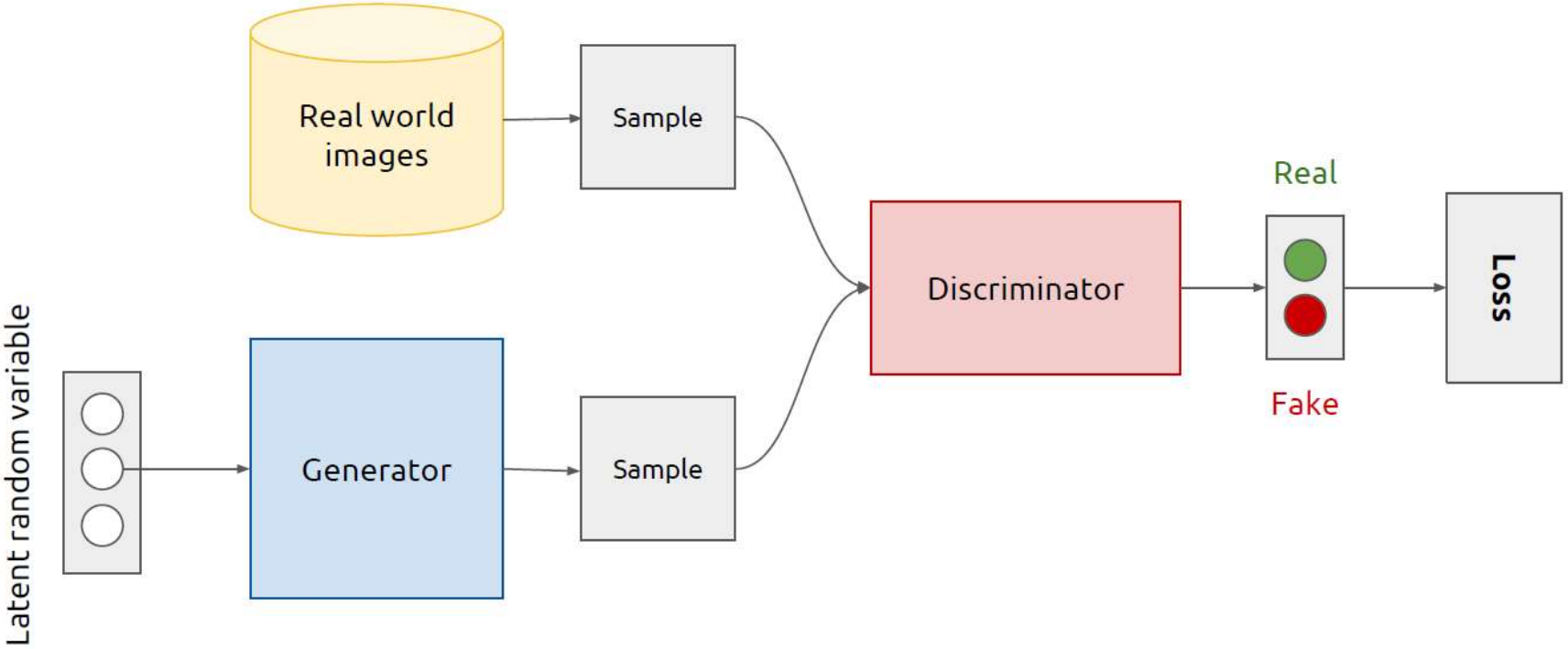
Хотим научиться генерировать изображения, похожие на обучающую выборку



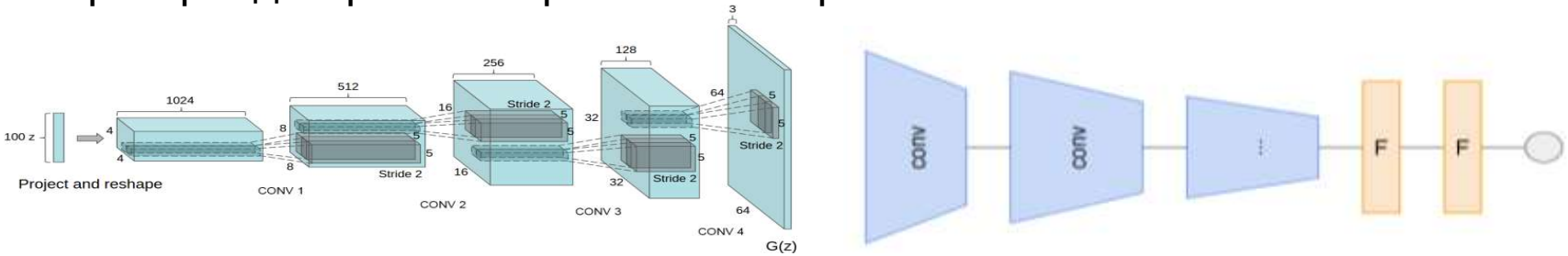
Как проверить, что у нас получилось хорошее, реалистичное изображение?

Общая схема Generative Adversarial Networks (GAN)

Сталкиваем генератор изображений из шума и дискриминатор:



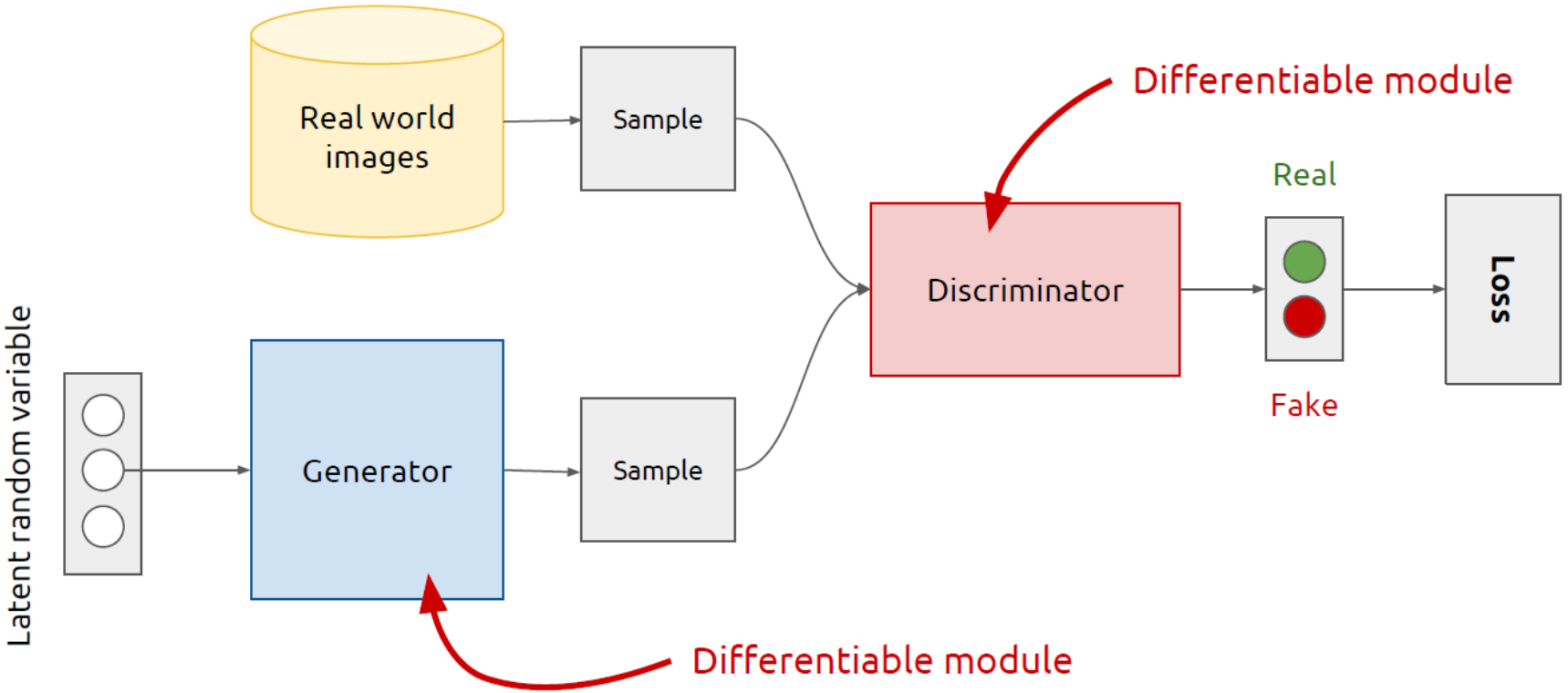
Генератор и дискриминатор обычно нейросети:



Ian Goodfellow et al, “Generative Adversarial Networks”, 2014

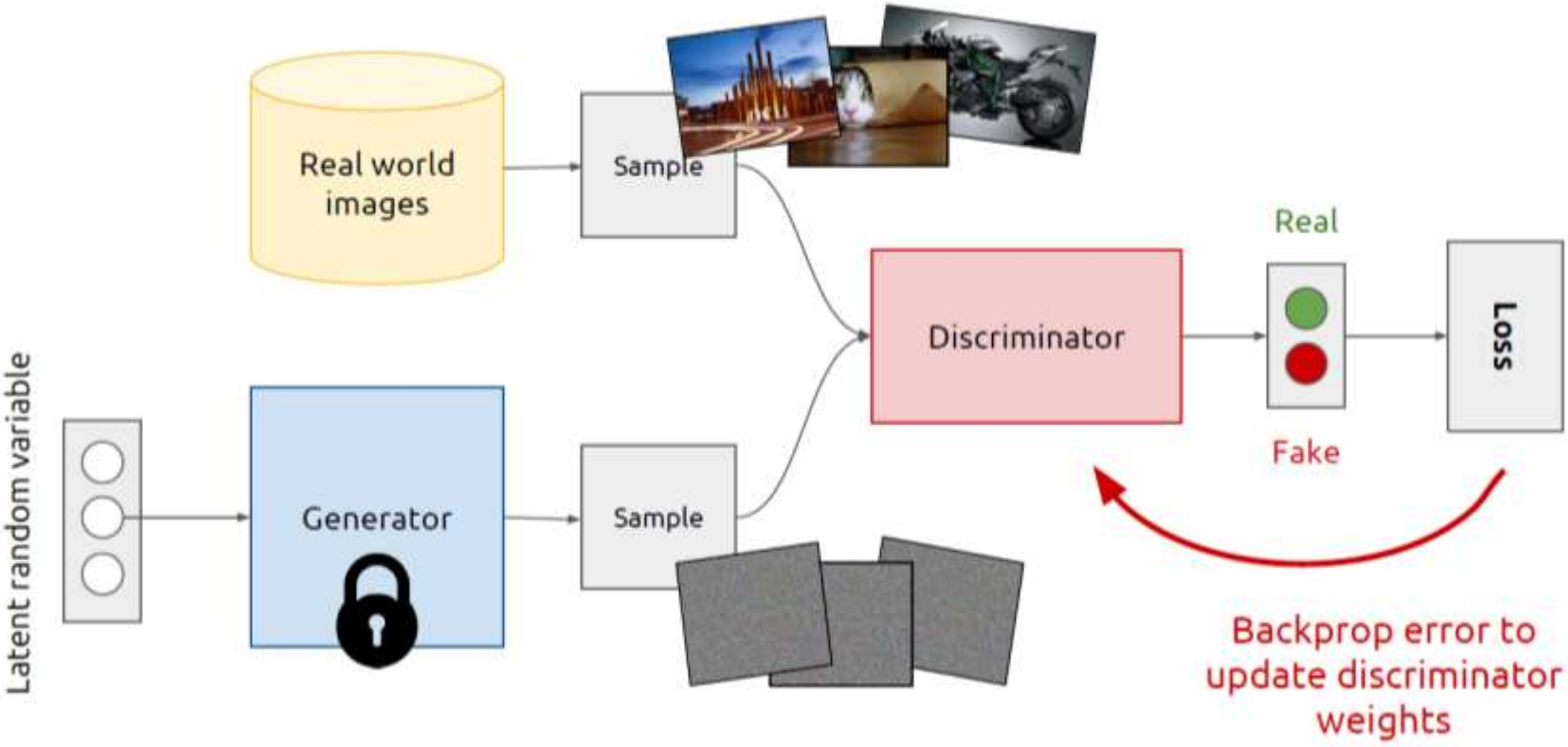
Обучение GAN

Будем обучать совместно, чередуя обучение дискриминатора и генератора



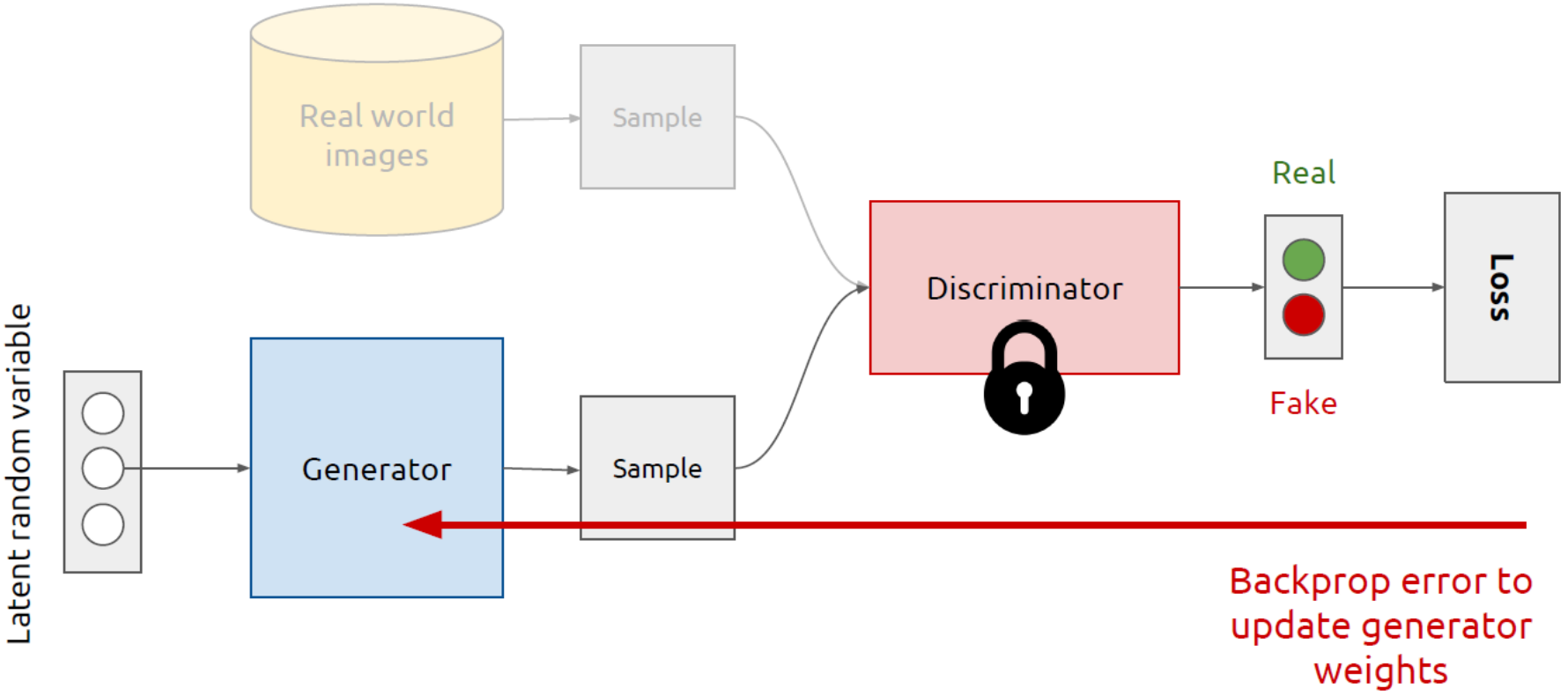
Почему нельзя взять готовый дискриминатор?

Обучение GAN



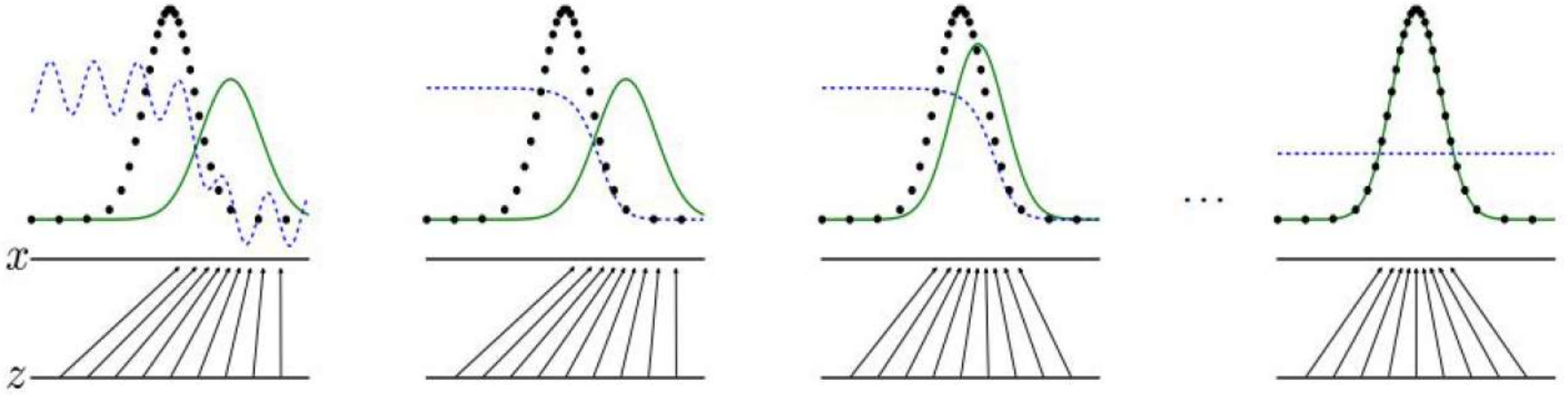
- Фиксируем веса генератора
- Берём примеры реальных изображений и синтезированных изображений
- Учим дискриминатор из различать

Обучение GAN



- Фиксируем веса дискриминатора
- Генерируем примеры генератором
- Распространяем ошибку от дискриминатора

Обучение GAN



- Чередование обучение дискриминатора повторяется до сходимости, которая может и не состояться
- В конце мы надеемся, что генератор научиться делать такие изображения, что дискриминатор не сможет их различить

Результаты

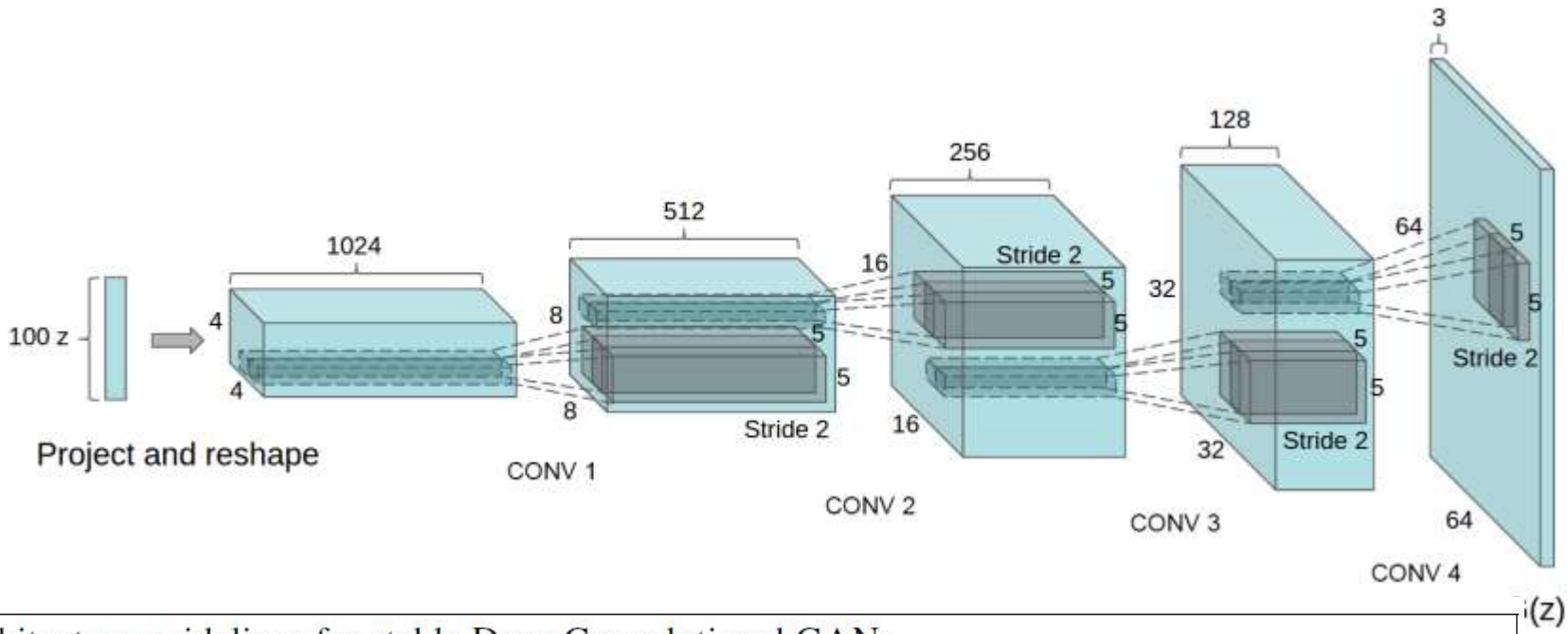
a)

b)

c)

d)

DCGAN



Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

A.Radford, L. Metz, S. Chintala UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS. ICLR 2016

Примеры - лица

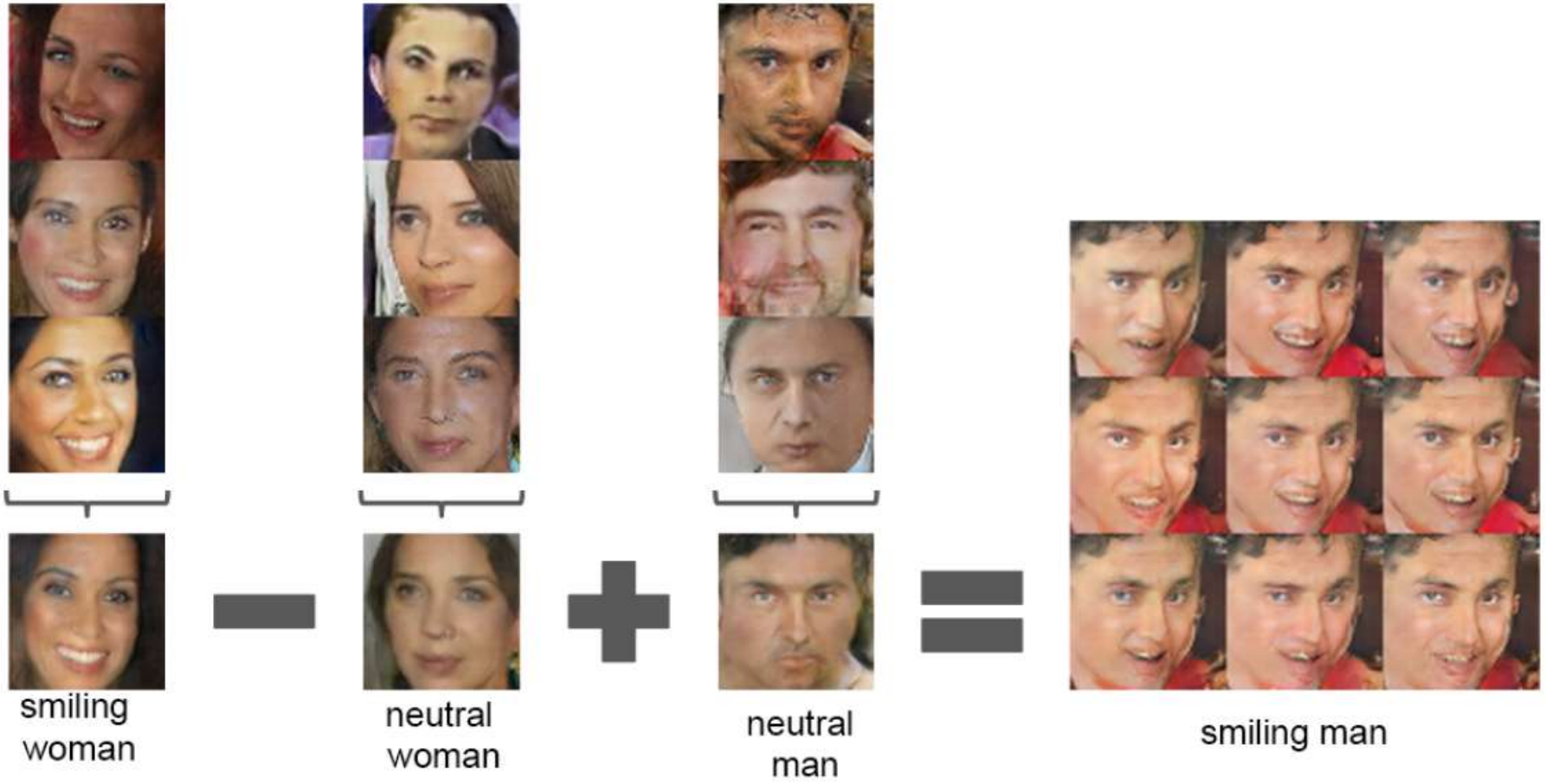
Примеры – обложки альбомов

Примеры - спальни

Изучение представления

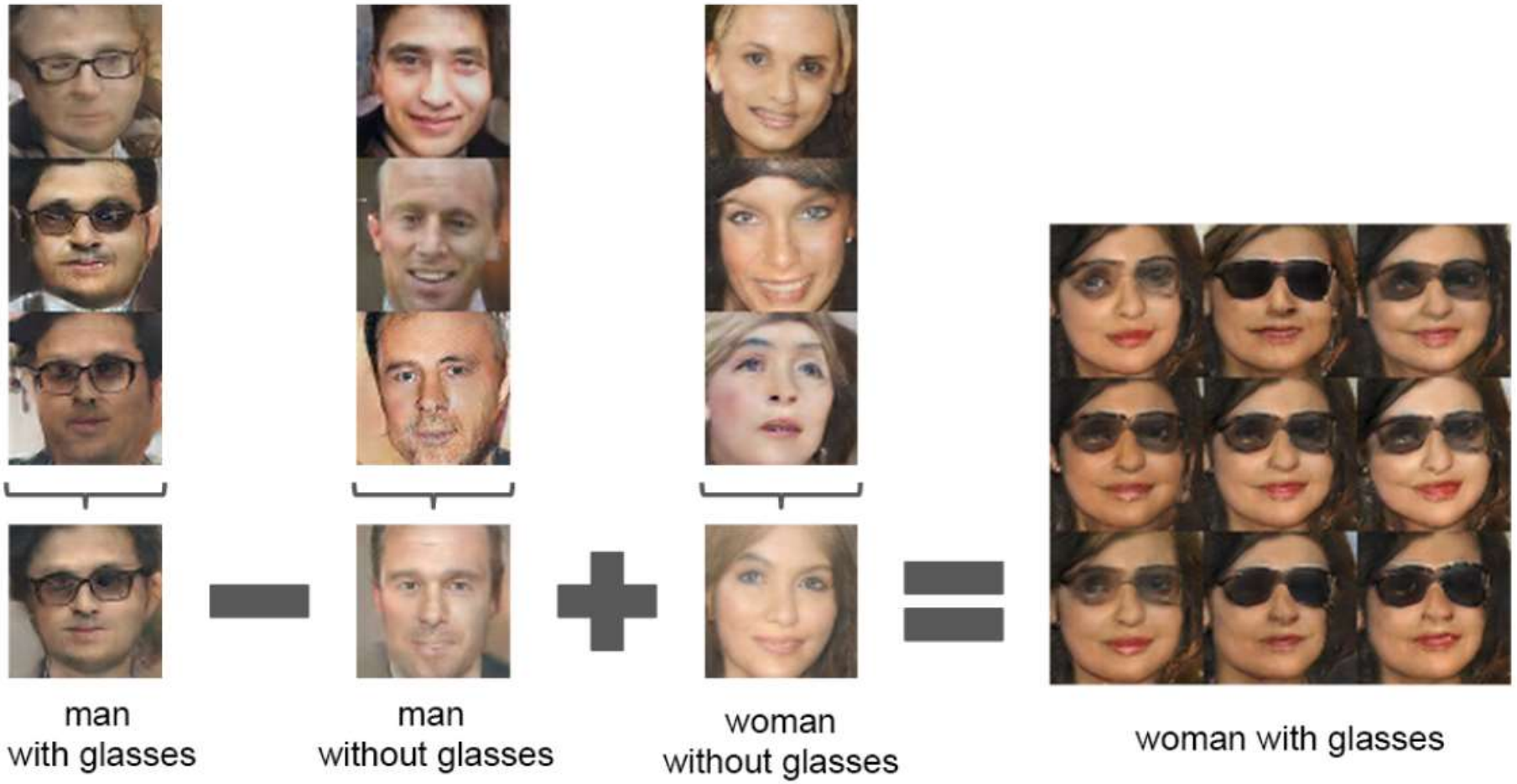
- Поиск фильтров, которые соответствуют определённым объектам
  - Логистическая регрессия по активациям фильтров внутри нарисованных областей
- Обнуление фильтров, которые были выбраны как соответствующие объектам
- На изображениях «исчезли» окна при некотором ухудшении качества
- Мораль – получаем представление, в котором за разные объекты отвечают разные фильтры

Векторная арифметика



- Работа с z-векторами, которые использовались для генерации изображений
- Отдельные вектора нестабильны, поэтому сумма по 3м изображениям

Векторная арифметика

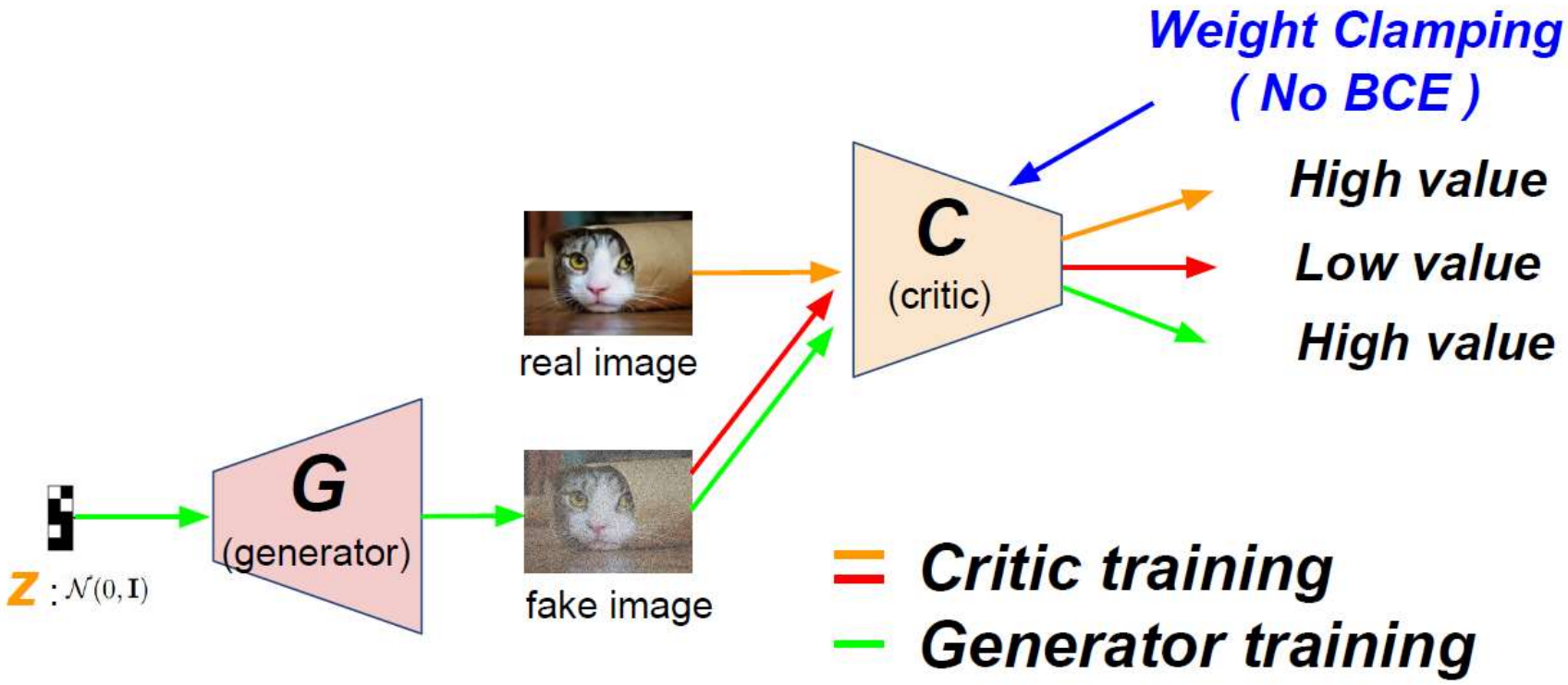


Векторная арифметика

- Вектор «поворота» как разница между усреднёнными векторами 4х изображений с лицами повернутыми налево и 4мя направо

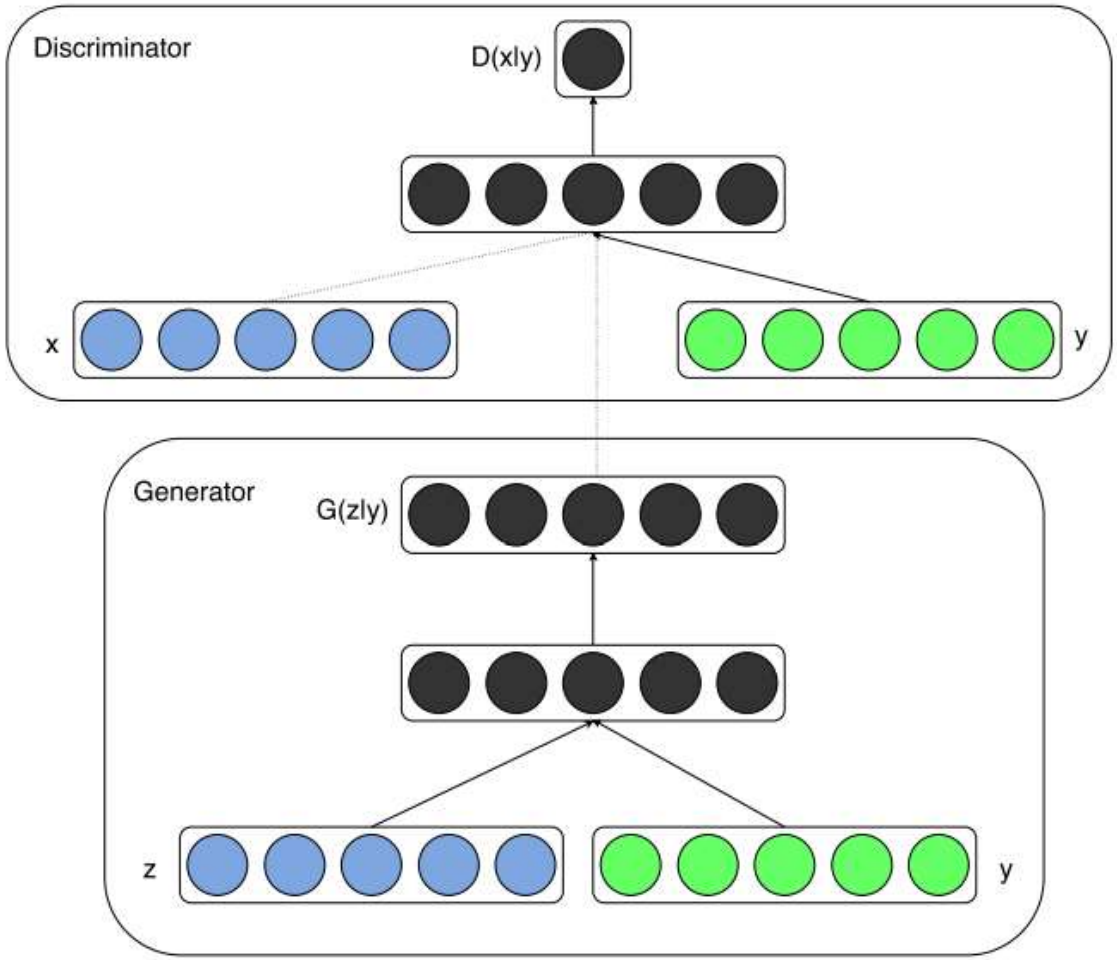
Wasserstein GAN

- Martin et al, Wasserstein GAN, 2017



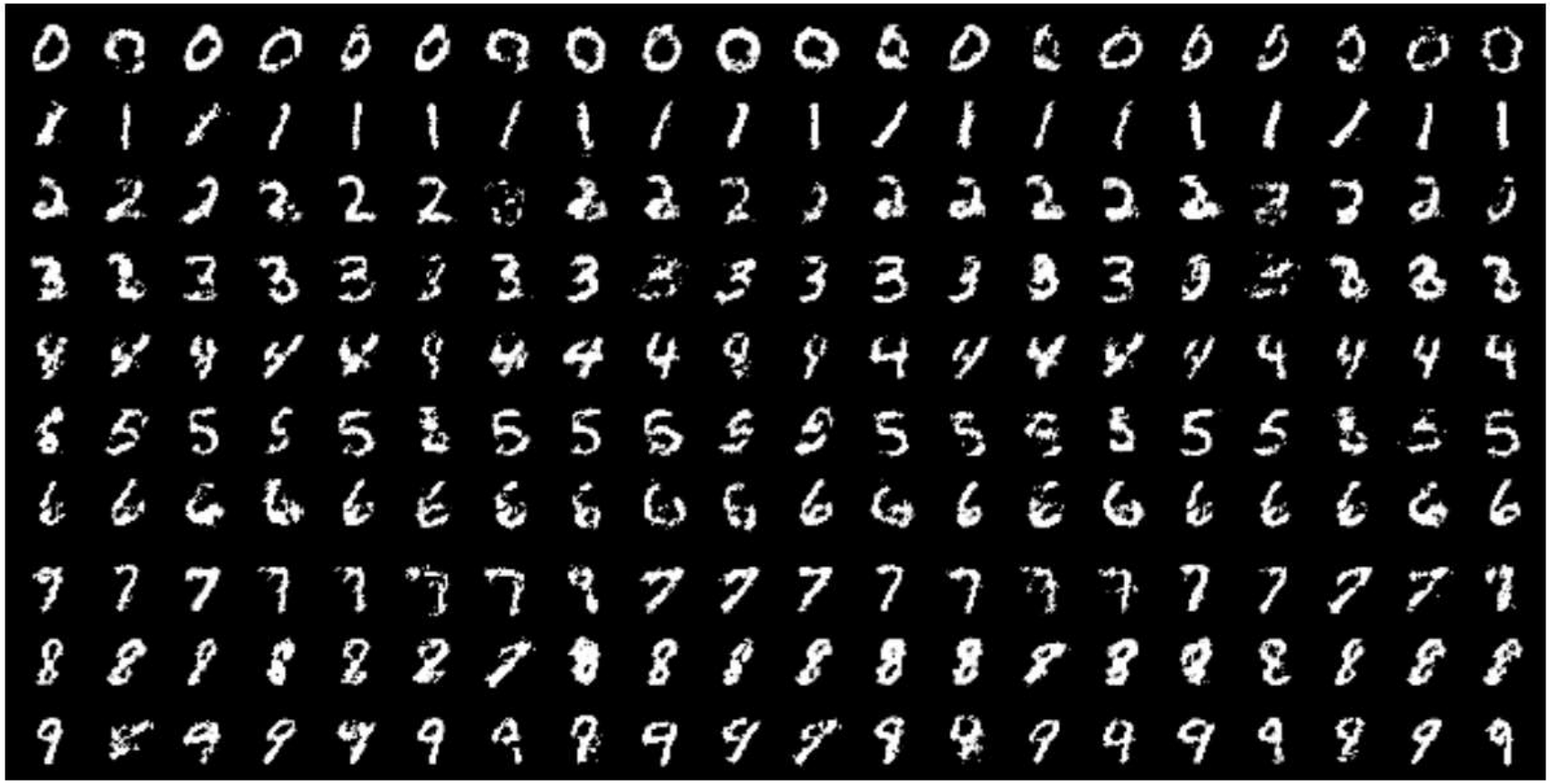
Примеры работы

Conditional GAN

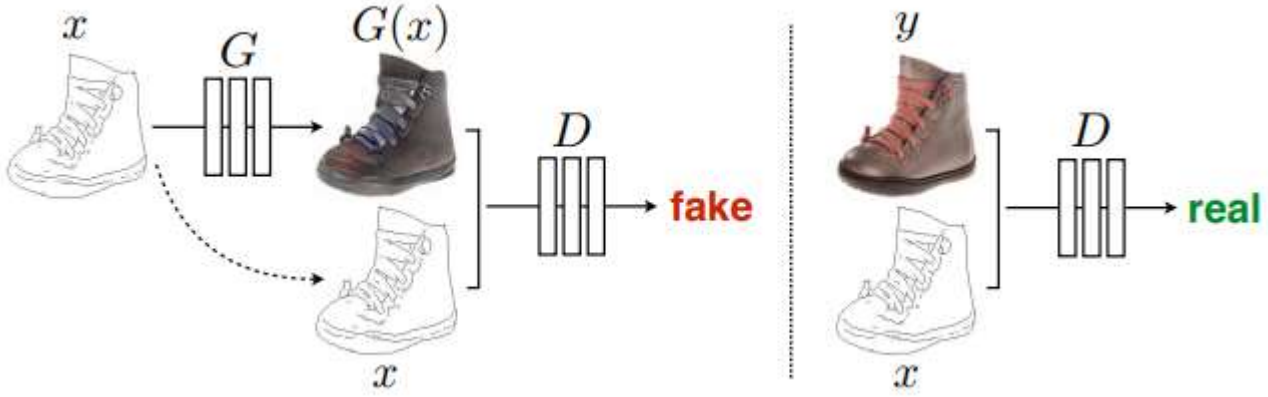


Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).

Conditional GAN на MNIST



Pix2Pix для преобразования изображений



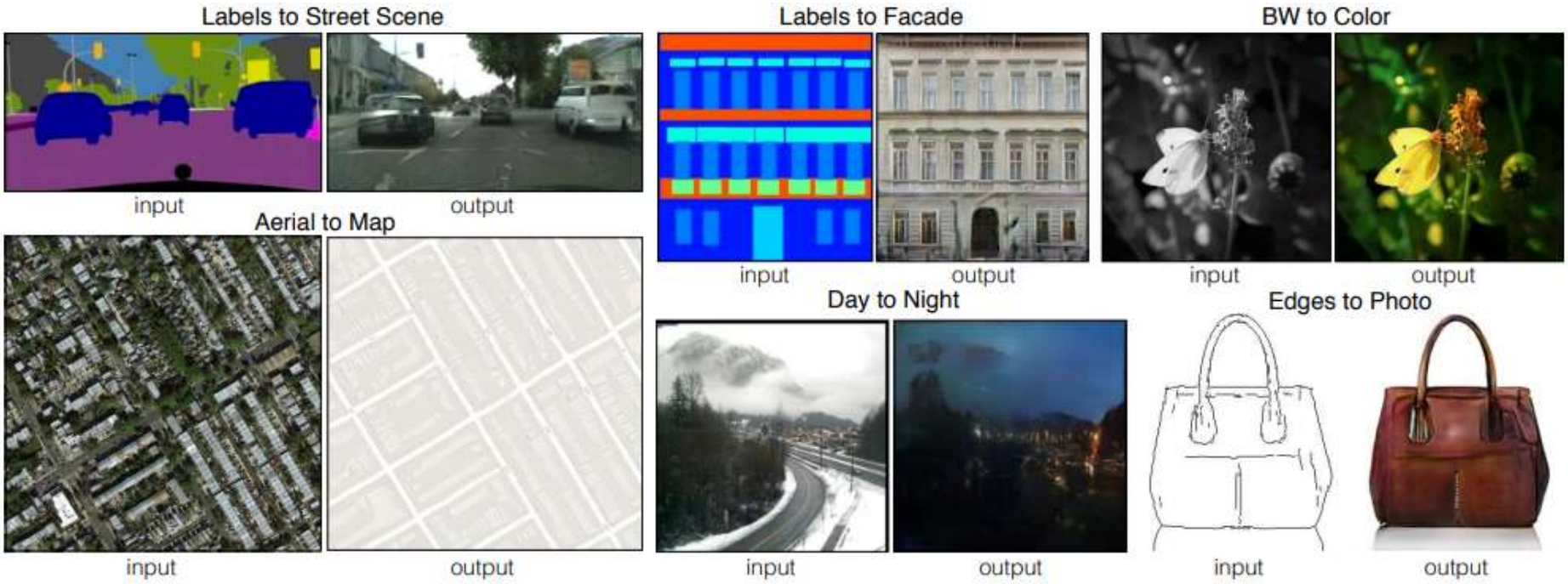
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]$$

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *arXiv preprint* (2017).

Pix2Pix

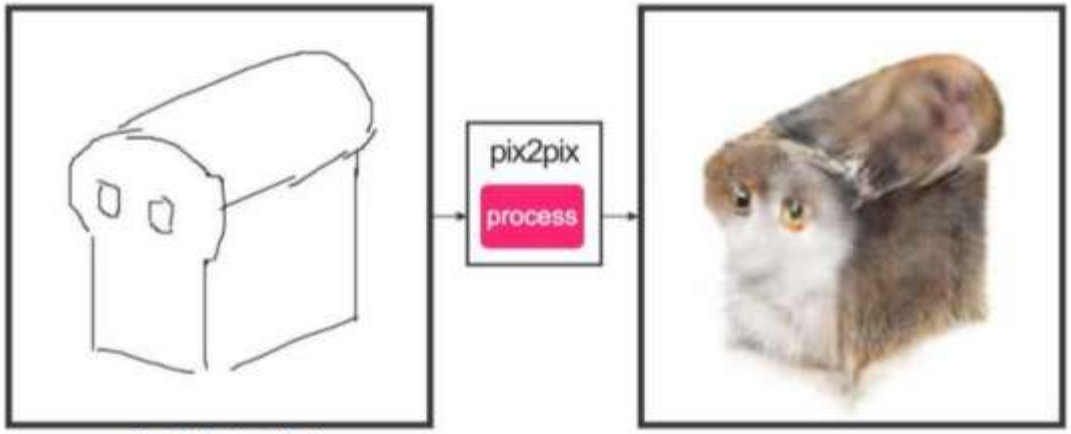

Labels to Street Scene
input
output
Labels to Facade
input
output
BW to Color
input
output
Aerial to Map
input
output
Day to Night
input
output
Edges to Photo
input
output

Pix2Pix

Pix2Pix

Pix2Pix

#edges2cats *by Christopher Hesse*



sketch by Ivy Tsai

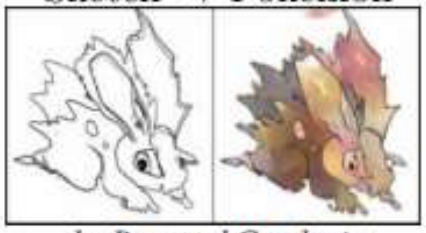
Background removal

by Kaihu Chen

Palette generation

by Jack Qiao

Sketch → Pokemon



by Bertrand Gondouin

"Do as I do"

by Brannon Dorsey

Pix2Pix

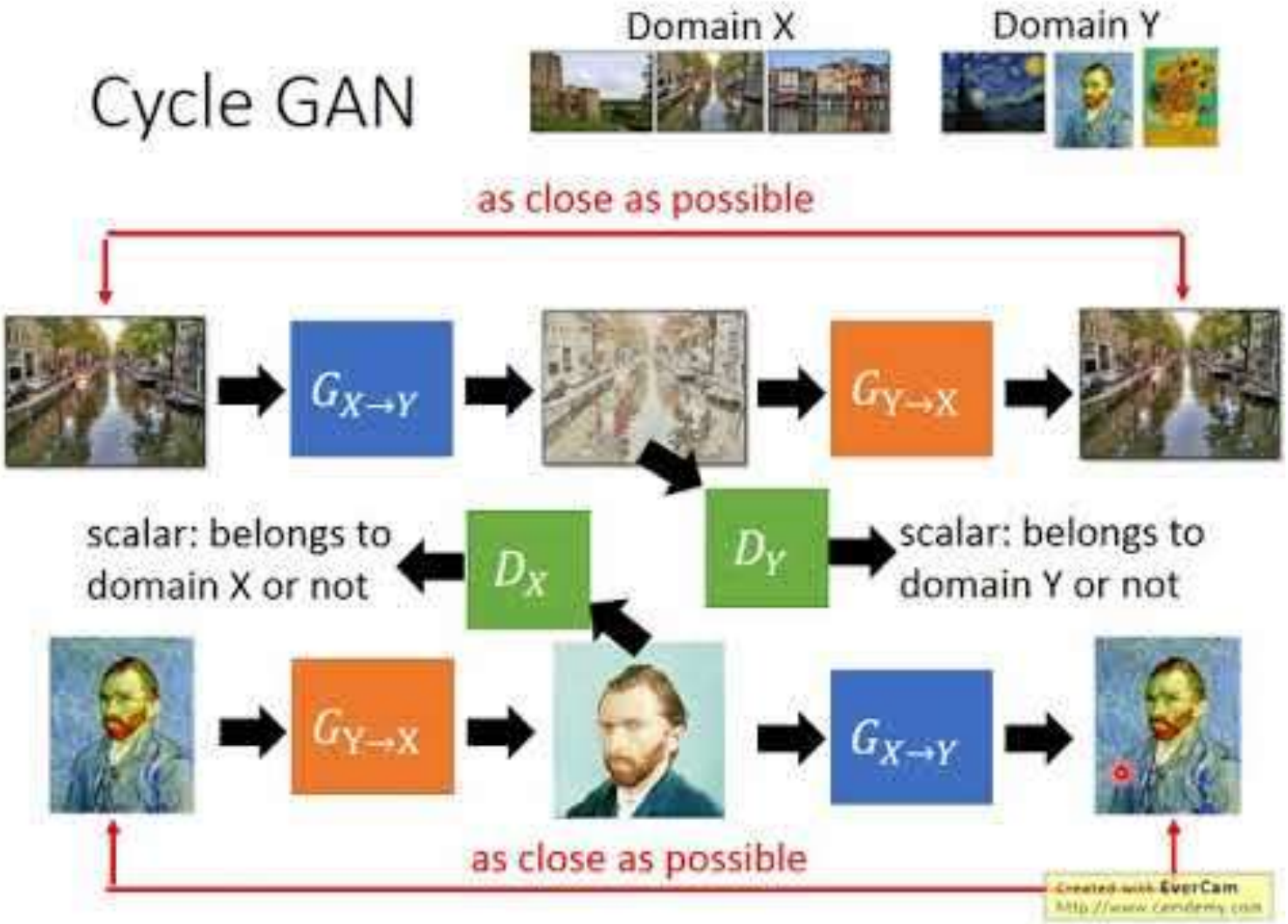
Архитектура:

- Генератор: U-net
- Дискриминатор: PatchGAN

Ограничение:

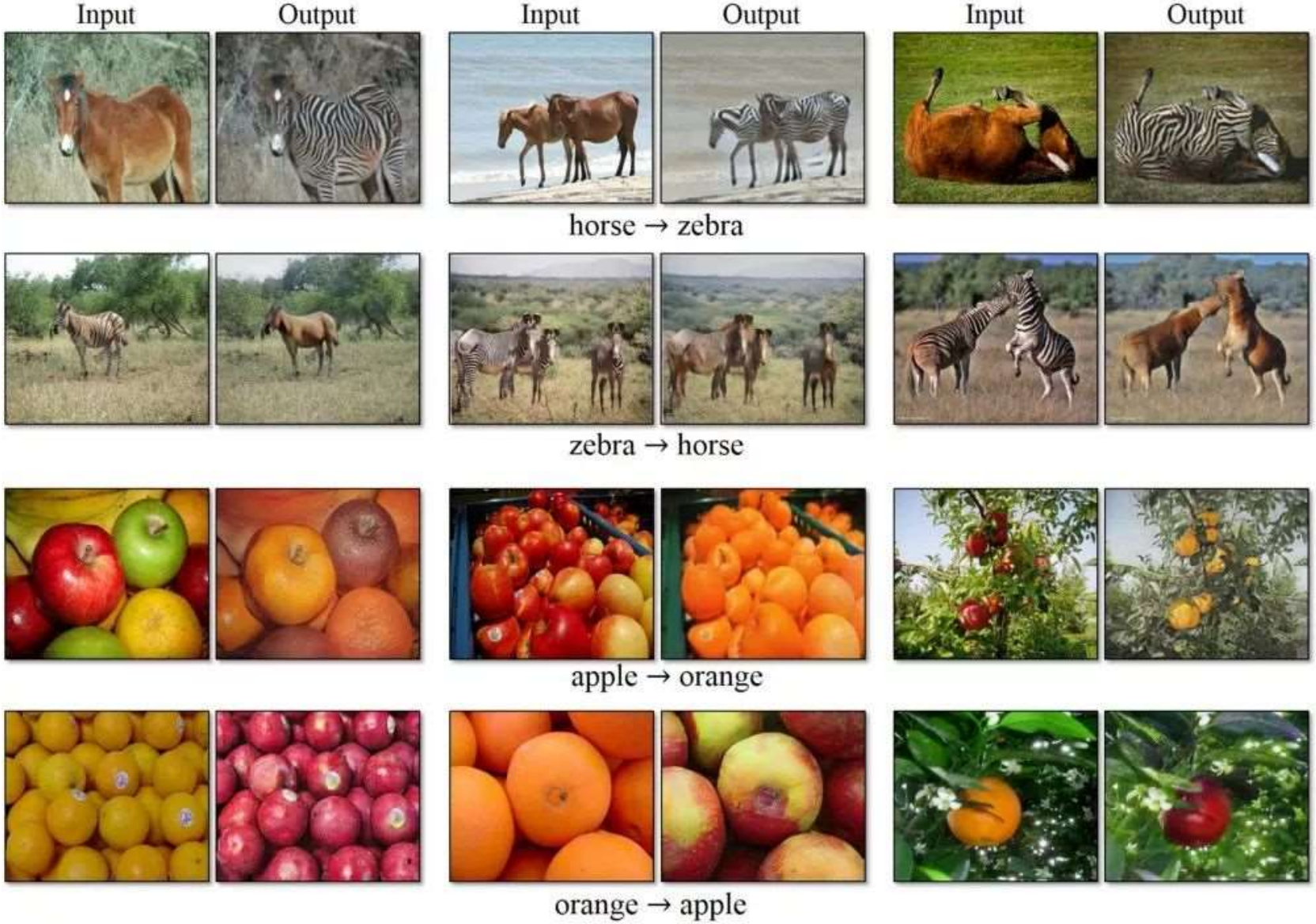
- Требует размеченные попарно данные

CycleGAN



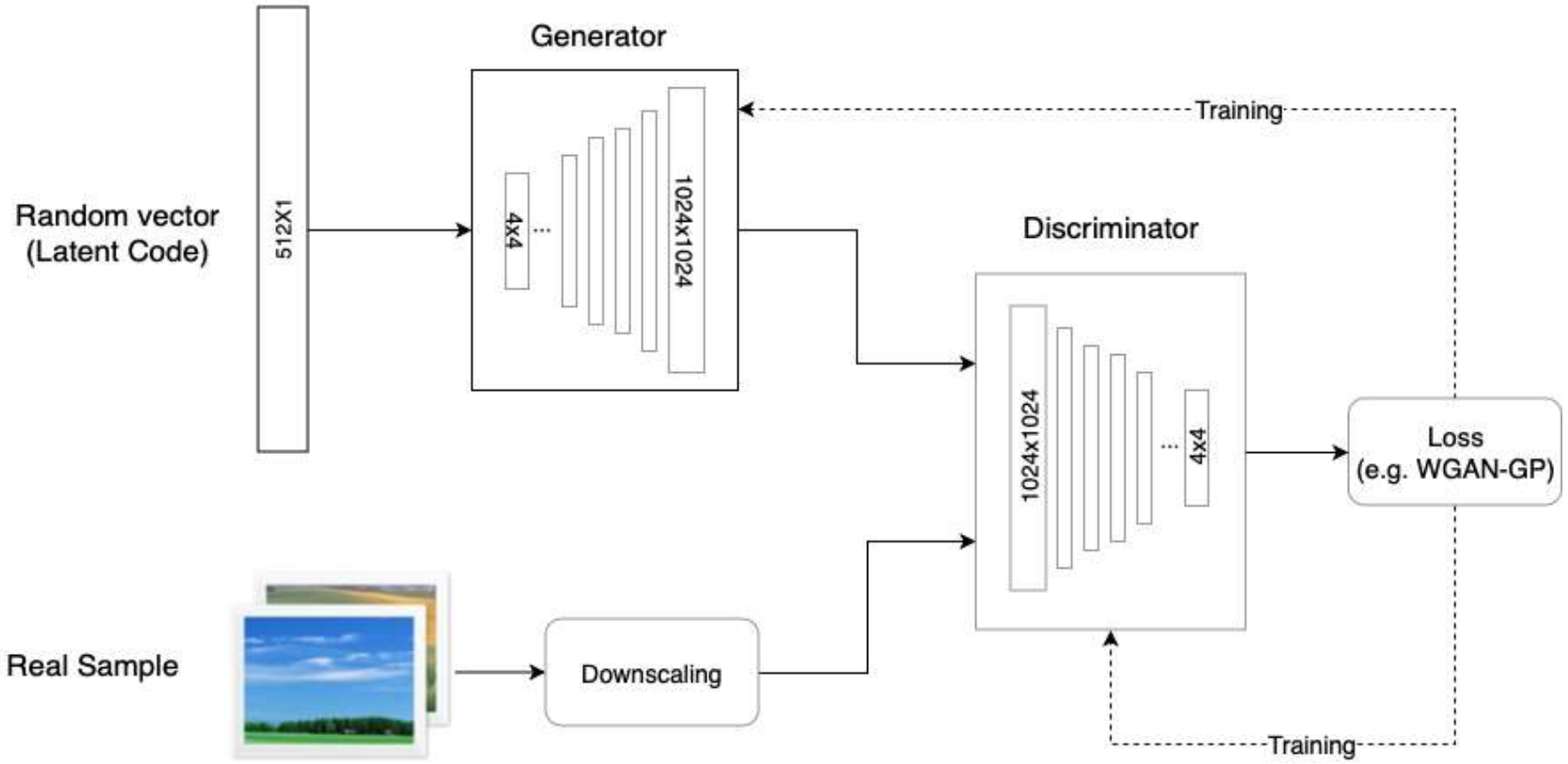
Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *arXiv preprint*(2017).

CycleGAN


Input
Output
Input
Output
Input
Output
horse → zebra
zebra → horse
apple → orange
orange → apple

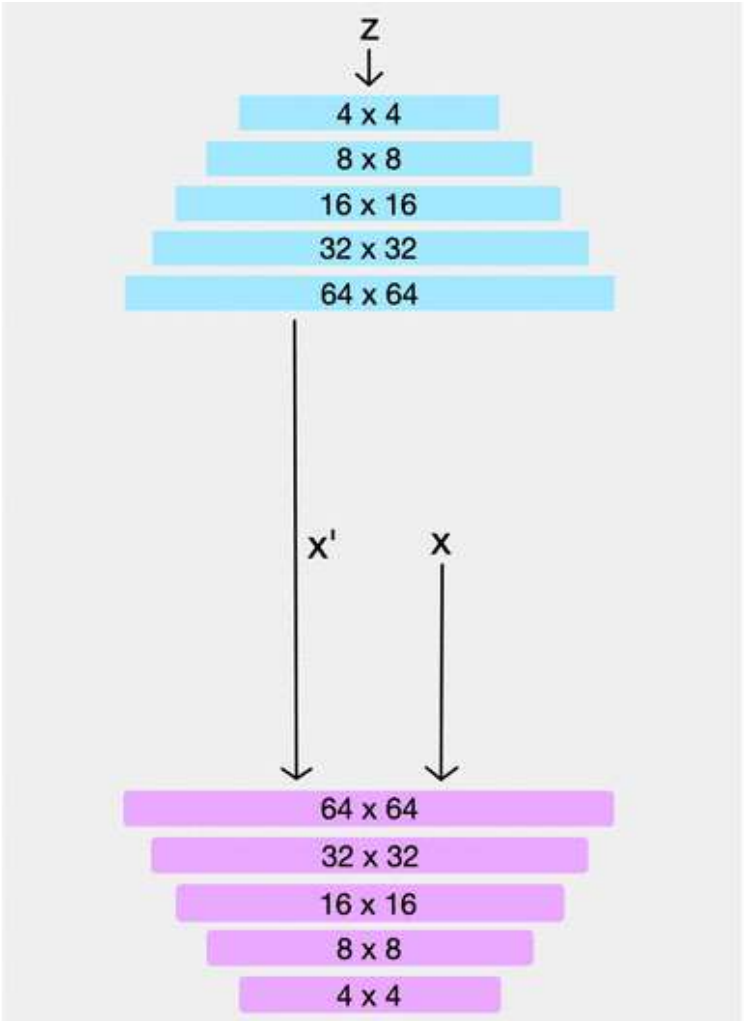
CycleGAN

Progressive GAN



https://arxiv.org/abs/1710.10196

Демонстрация ProGAN

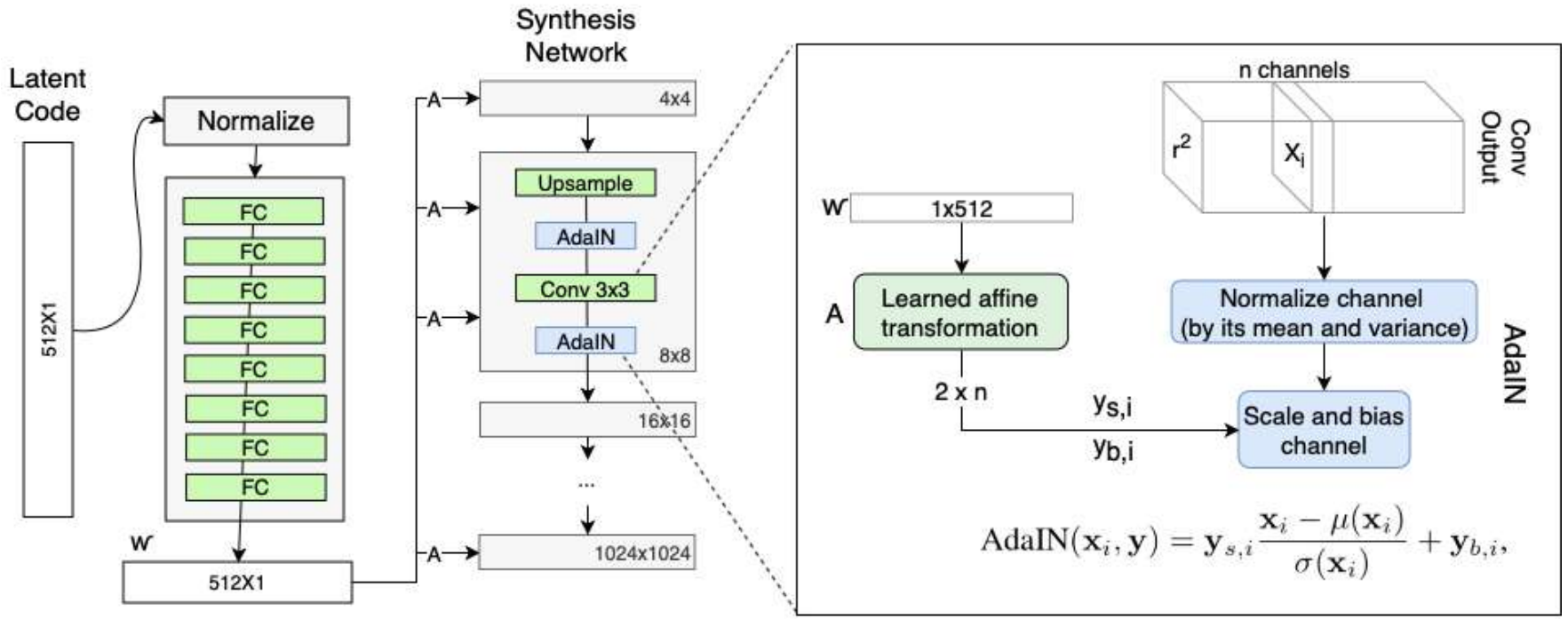


Training time: 2 days
64x64 resolution

Generator
Discriminator

z = random code
x = real image
x' = generated image

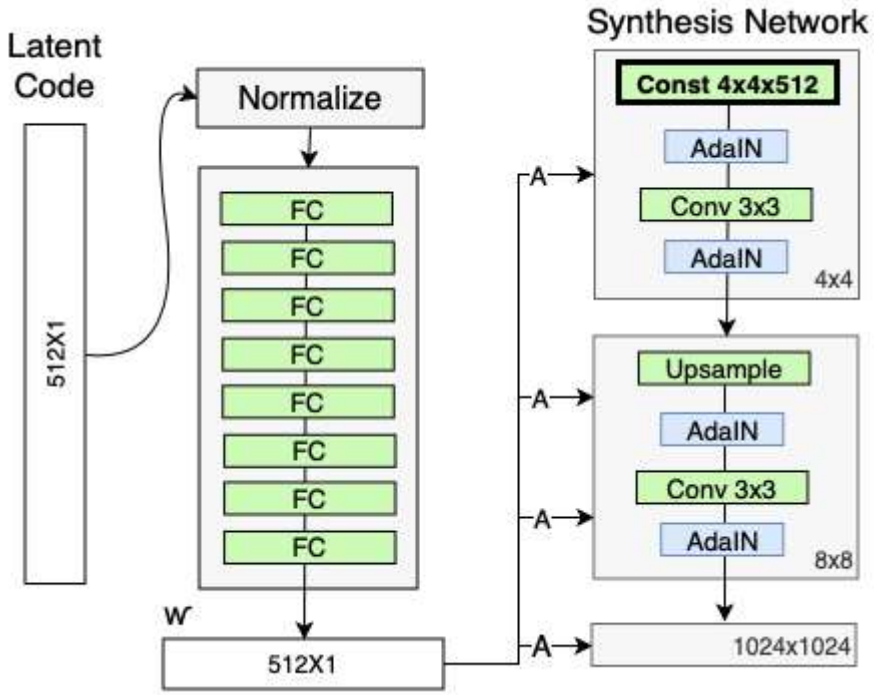
StyleGAN



Пропустим код через специальную сеть, которая сгенерирует «управляющий» сигнал. Сигнал подается на AdaIN слой

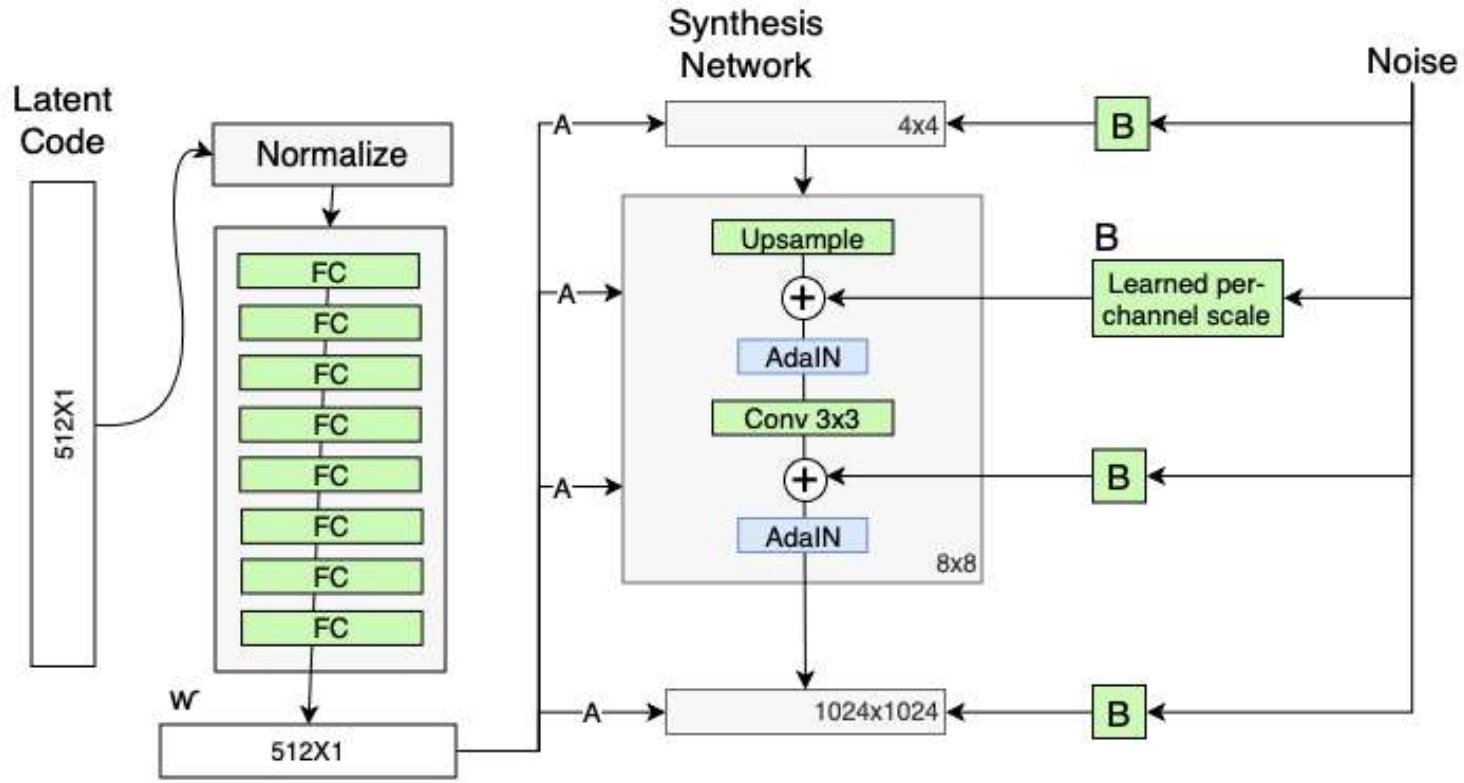
https://arxiv.org/pdf/1812.04948.pdf

StyleGAN



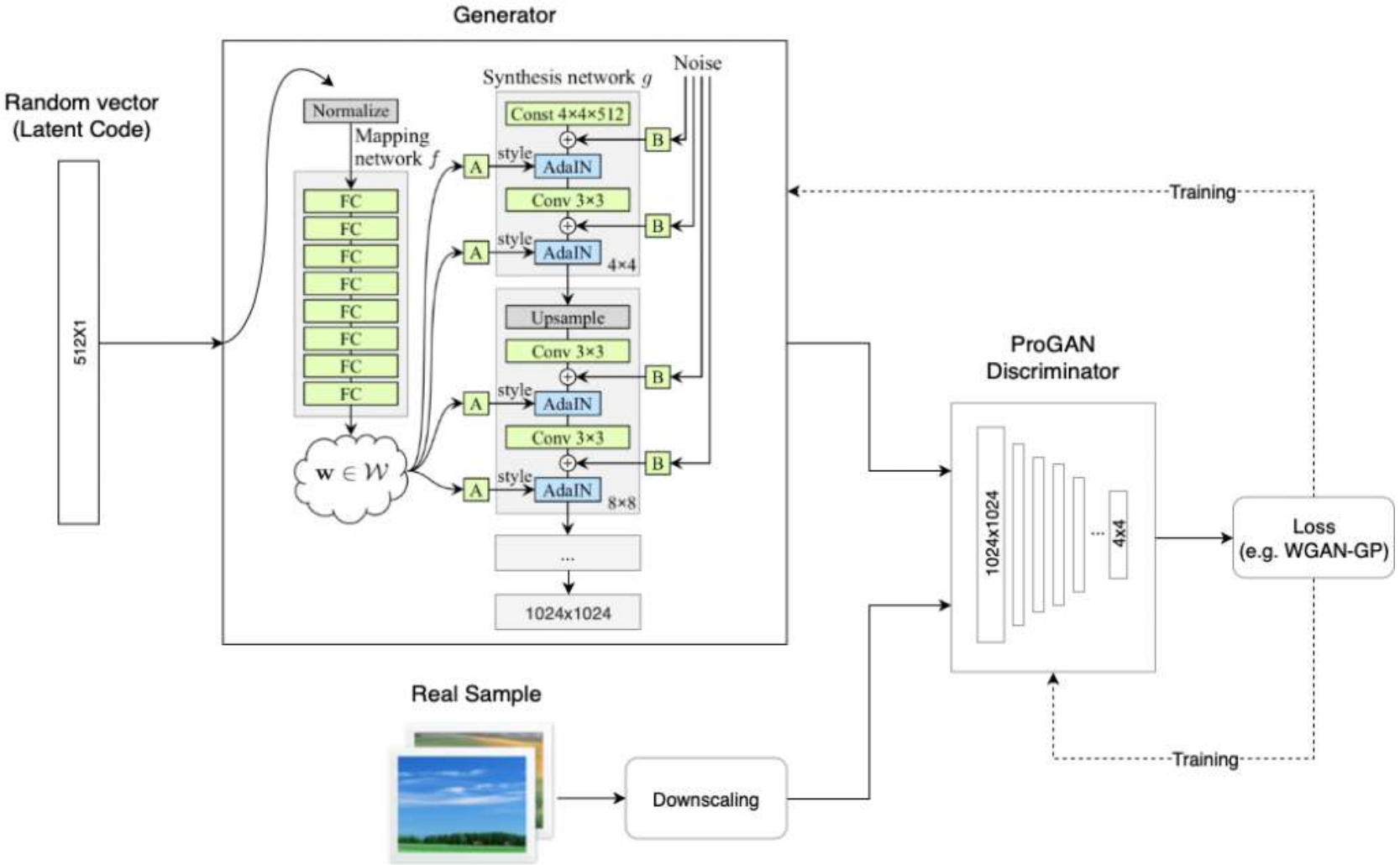
Оказывается, что на вход генератору можно подавать константное мини-изображение. И дальше оно будет преобразовываться через управляющие сигналы в целевое

StyleGAN

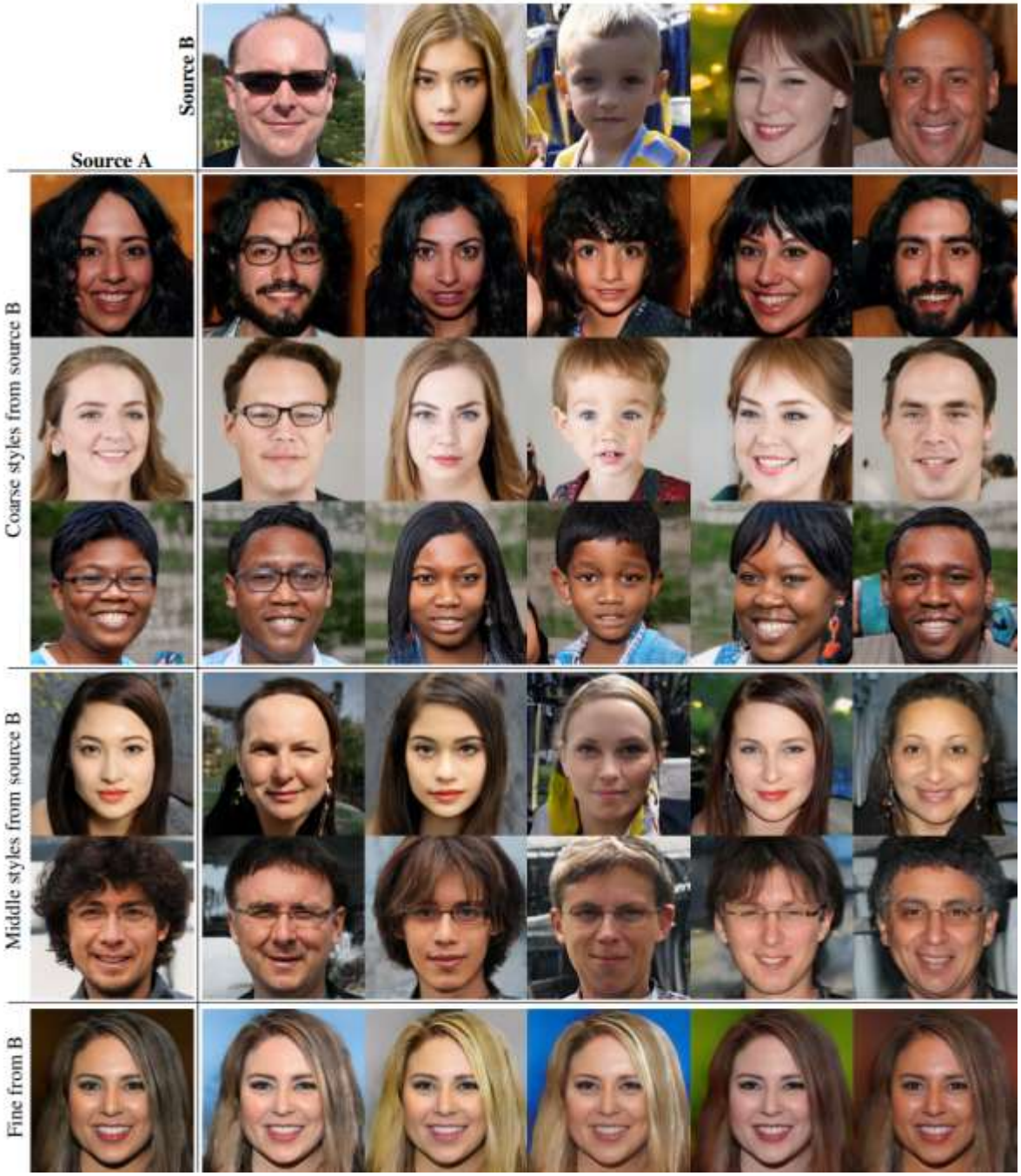


Для генерации случайных мелких деталей и дополнительной рандомизации будем подмешивать на каждом этапе шум

StyleGAN

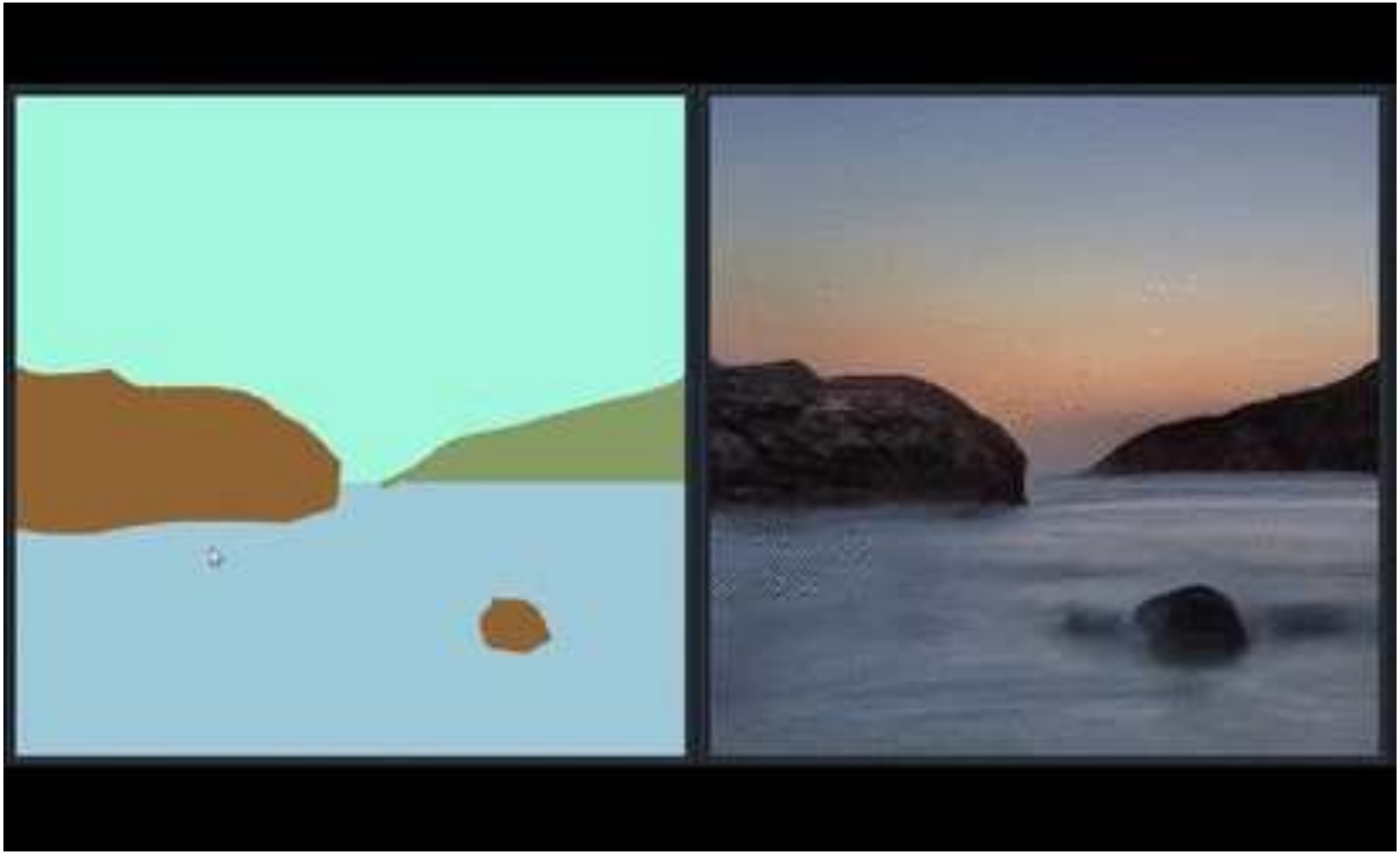


Обучаем в прогрессивном режиме



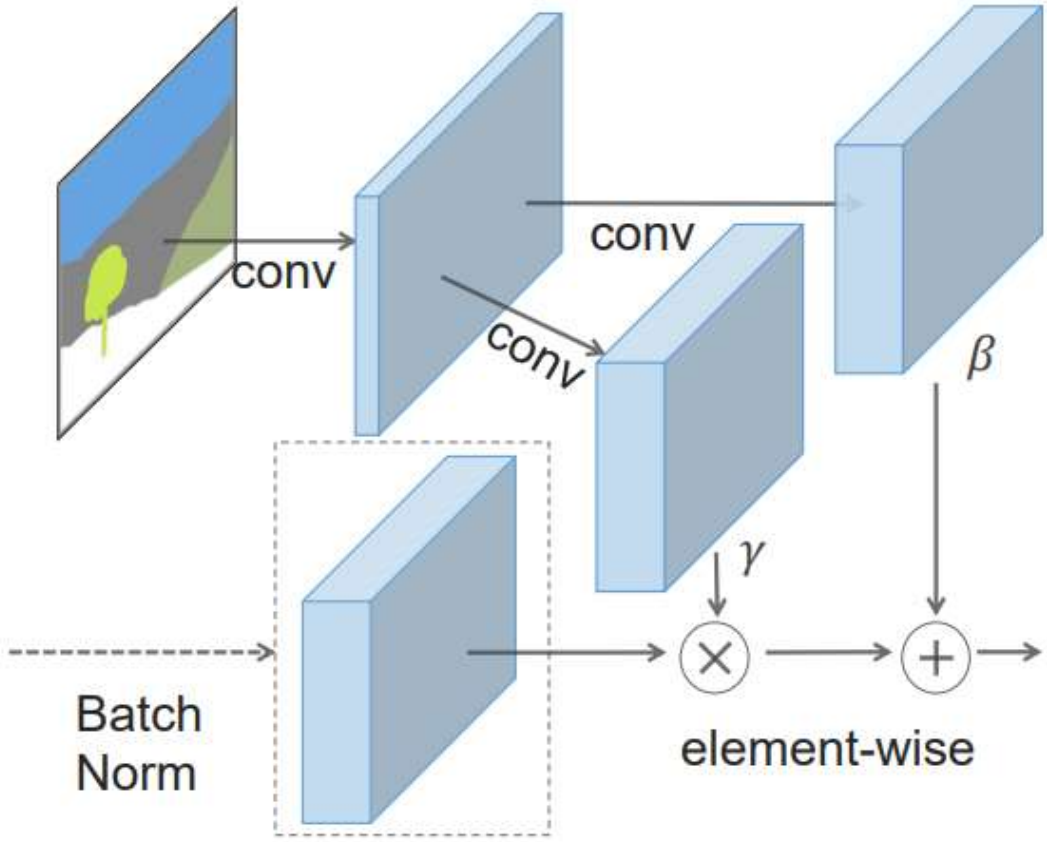
- Разделим «стиль» на компоненты
- Для этого будем учить сеть так, что на разные уровни подаётся управляющий сигнал, сгенерированный от разных кодов
- Можем управлять генерацией, выбирая интересные нам изображения и подмешивая их коды
- В примерах – стиль от B на соответствующем уровне, и всё остальное от A

SPADE (or GauGAN)



https://arxiv.org/pdf/1903.07291.pdf

Spatially-adaptive denormalization



- Похоже на AdaIN, но генерируем параметры для каждого пикселя
- Параметры генерируются управляющей сеткой

Генератор в SPADE

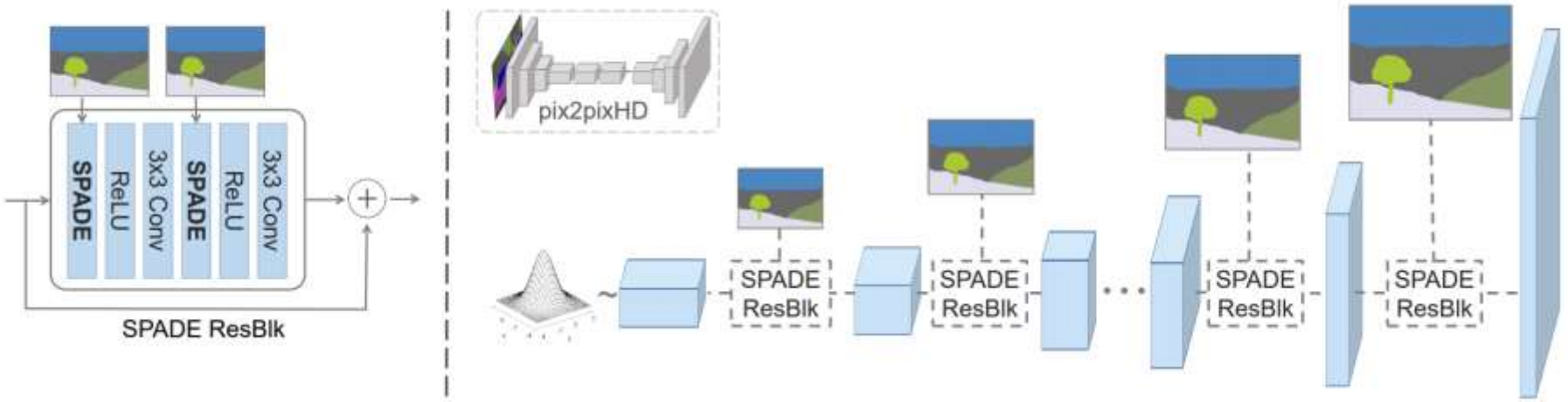
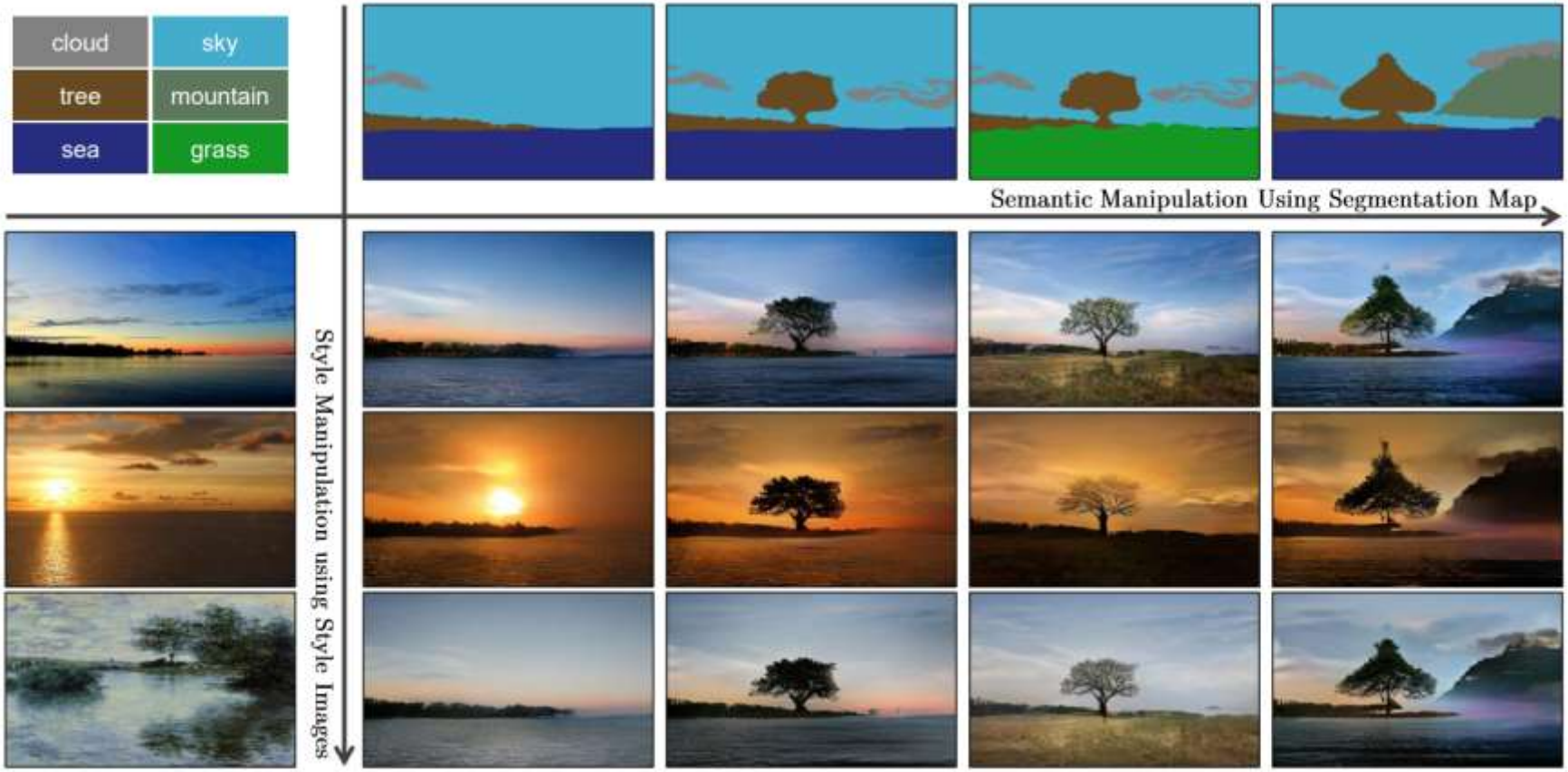


Figure 4: In the SPADE generator, each normalization layer uses the segmentation mask to modulate the layer activations. *(left)* Structure of one residual block with the SPADE. *(right)* The generator contains a series of the SPADE residual blocks with upsampling layers. Our architecture achieves better performance with a smaller number of parameters by removing the downsampling layers of leading image-to-image translation networks such as the pix2pixHD model [48].

Стилизация изображения



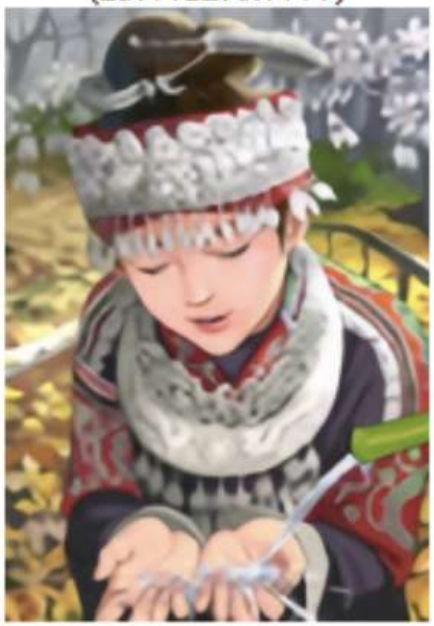
- На вход подаётся случайный вектор
- Вместо случайного вектора можно подавать «сжатое» представление стиля, полученное из Encoder-Decoder сети

Применение - суперразрешение

original

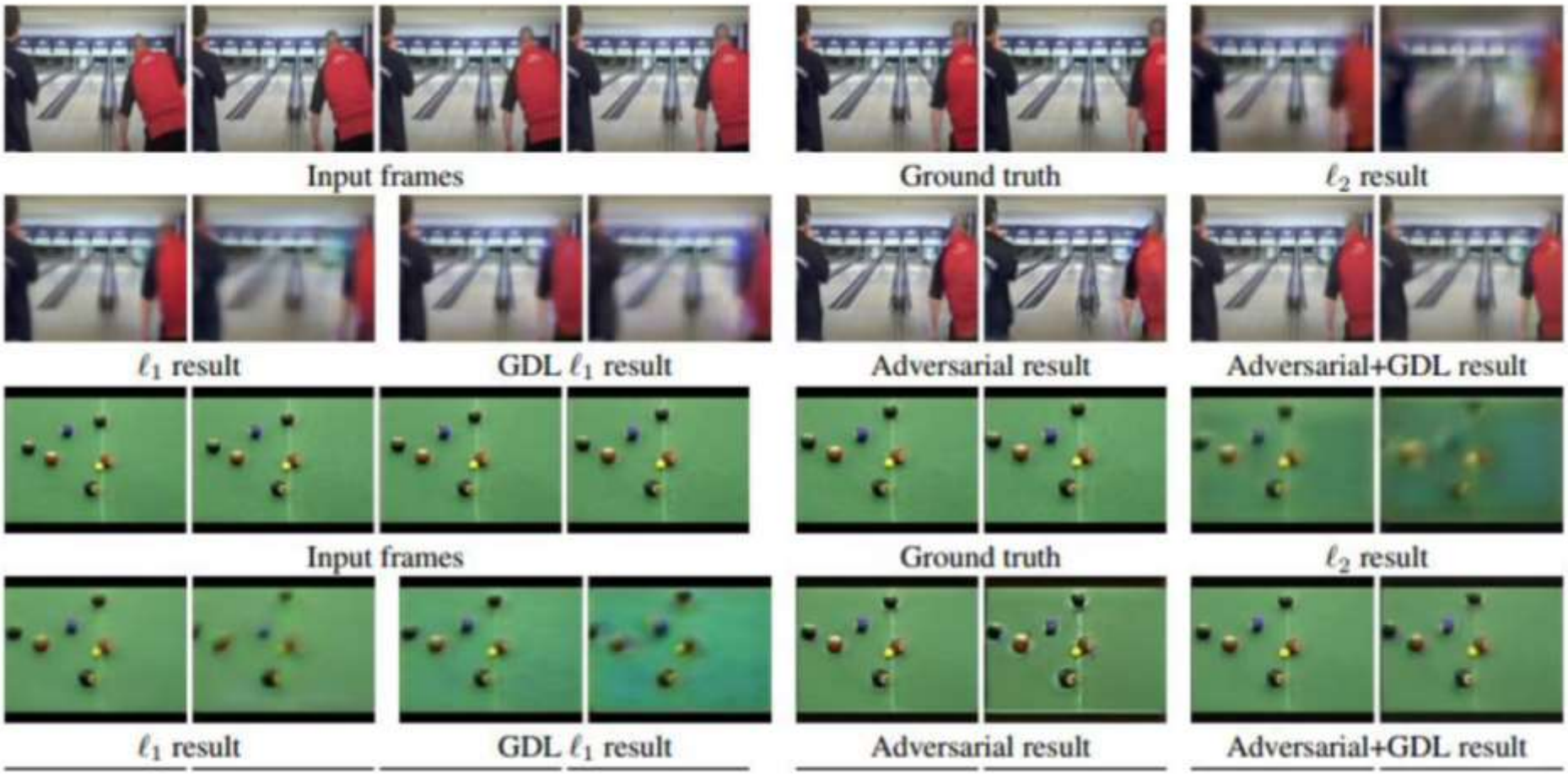
bicubic
(21.59dB/0.6423)

SRResNet
(23.44dB/0.7777)



SRGAN
(20.34dB/0.6562)

Применение - предсказание кадров



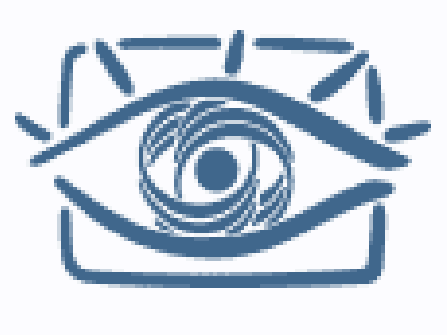
Mathieu et al. **Deep multi-scale video prediction beyond mean square error**, ICLR 2016 (https://arxiv.org/abs/1511.05440)

Применение – Domain Adaptation

Jian Liu, Ajmal Mian. Learning Human Pose Models from Synthesized Data for Robust RGB-D Action Recognition. https://arxiv.org/abs/1707.00823

Резюме лекции

- Реалистичность изображения можно оценить с помощью сети-дескриминатора
- Похожесть изображений хорошо оценивается с помощью Perception Loss – сравнения значений нейросетевых признаков
  - М.б. Identity Loss – если признак кодируют «личность»
- Style Loss – функция от признаков, описывающие статистики распределения признаков
- Можем проводить линейные преобразования в пространстве признаков изображения
- Cycle Loss позволяет работать с произвольными выборками, а не с парами соответствующих изображений
- AdaIN и SPADE слои позволяют управлять синтезом, а параметры для них генерируем управляющими сетями



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №10
«Анализ видео»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

25 ноября 2019 года

Виды видео

Видеопоток – упорядоченная последовательность изображений, полученных с одной камеры через небольшие промежутки времени

Видеопоток подразумевает обработку на лету.
Видеопоследовательность же конечна, можно обрабатывать целиком.

- Пользовательское видео – от 3-5 кадров/сек до 30-50 кадров/сек
- Разрешение – от 320x240 до 1920*1080 (HD) (сейчас ещё 4K)
- В градациях серого (одноканальное) или цветное (3х канальное)
- Поток данных – 2Мб (один канал HD) x 3 (RGB) x 50 кадр/с = 300 Мб/с
  - Больше, чем пропускная способность 1 гигабитной Ethernet сетки!

Что мы хотим от анализа видео?

Извлечь информацию из видео:

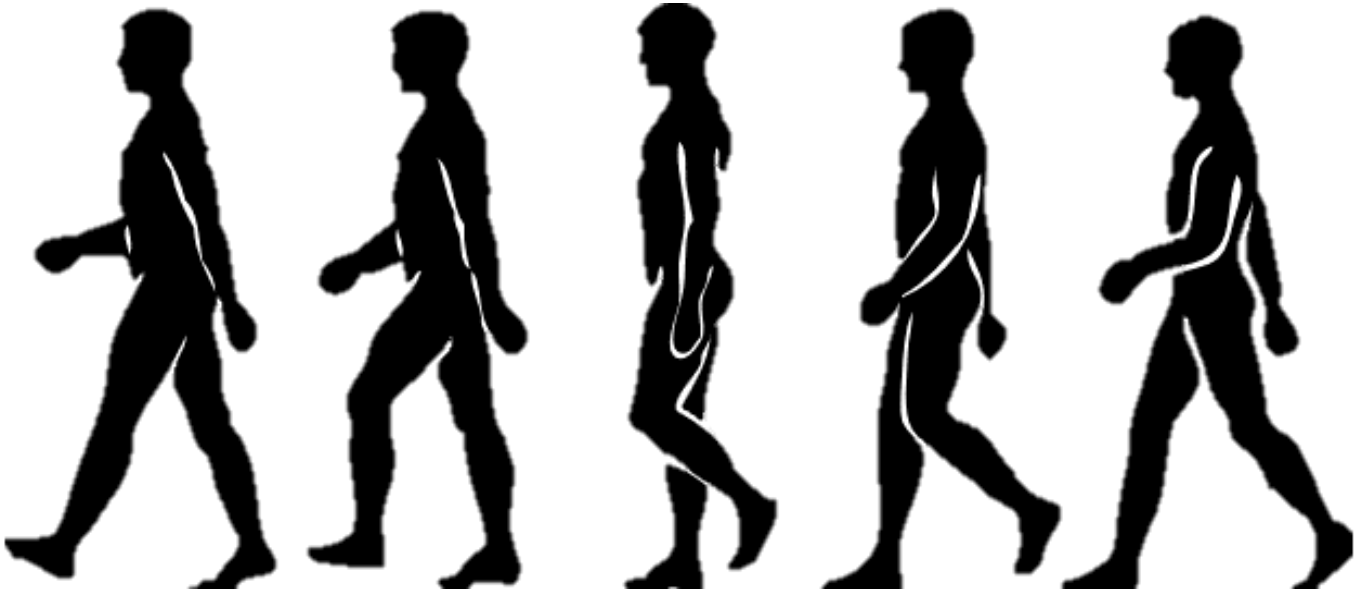
- Найти все объекты
- Отследить их движение
- Определить свойства (позы, атрибуты, личность)
- Распознать действия людей и события

Сценарии съёмки

- Ракурс, вид наблюдаемых объектов и т.д.
- Ракурсы съёмки могут быть крайне различны
- Работающие системы удается создать, «заточившись» на определённый сценарий съёмки

Optical flow (оптический поток)

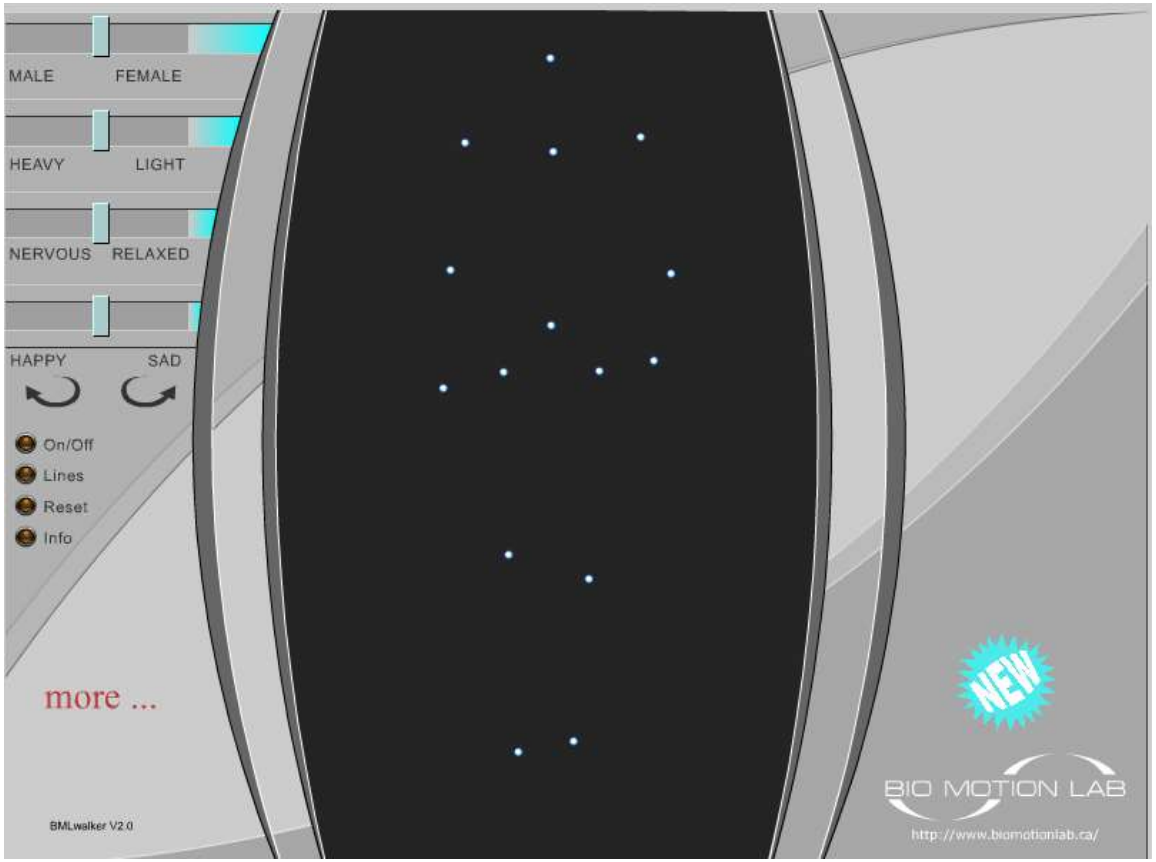
Движение



- Движение – главное отличие видео от изображений
- Движение само по себе является мощной визуальной подсказкой
- Суть многих действий именно в динамике

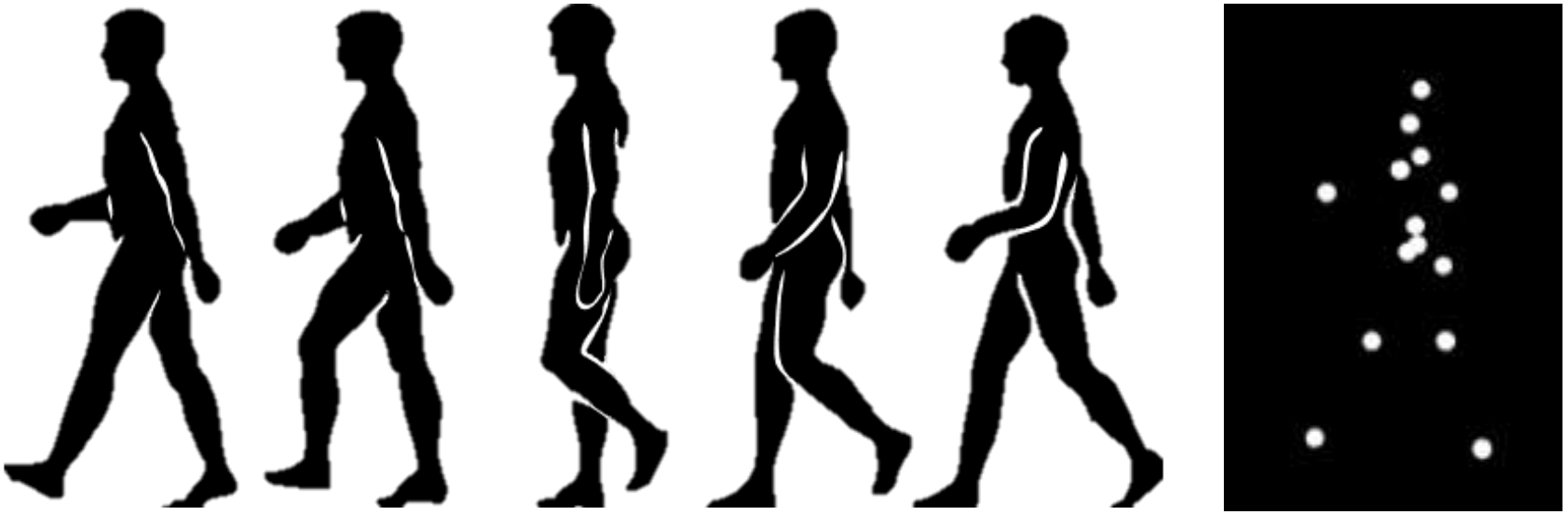
Распознавание по движению

- Иногда достаточно отследить движение отдельных точек, чтобы распознать событие



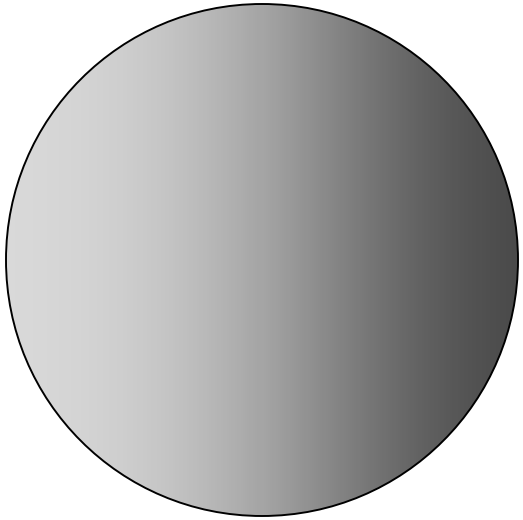
http://www.biomotionlab.ca/Demos/BMLwalker.html

Описание движения



- Точки наблюдаемой сцены движутся относительно камеры / изображения
- Векторное поле движения 2D проекций на изображение 3D точек объектов сцены называется *полем движения* (*motion field*)
- Нужно это движение как-то формализовать, описывать и измерять

Оптический поток

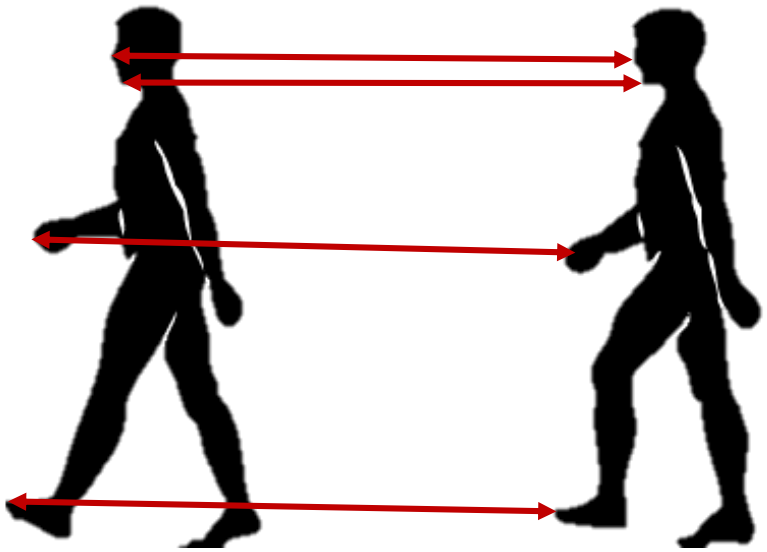


- Движение точек объектов по видео увидеть можно далеко не всегда
- E.g. Серый матовый шар, освещается с одной стороны и вращается вокруг своей оси

•Optical flow (оптический поток) – векторное поле *видимого* (*apparent*) движения пикселей между кадрами

• Вычисление оптического потока – одна из базовых задач анализа видео

Оптический поток

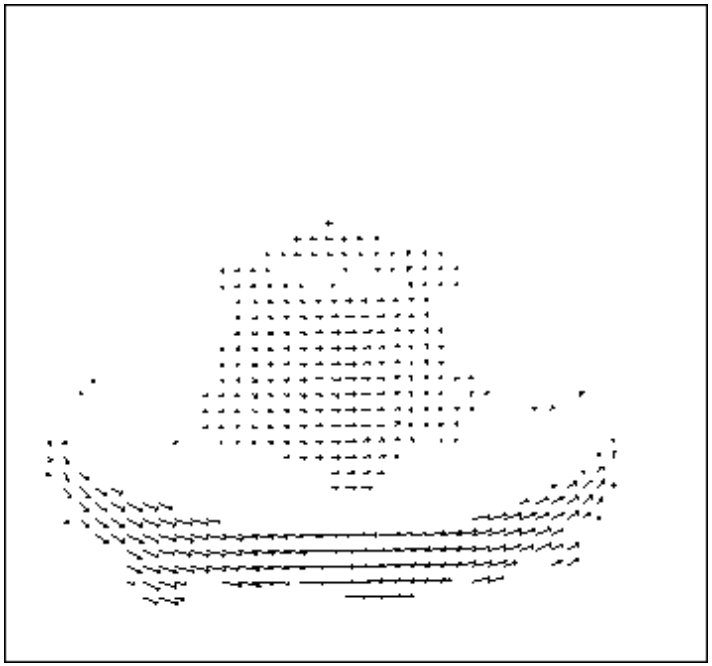


- Optical flow – векторное поле *видимого* (u_{ij}, v_{ij})

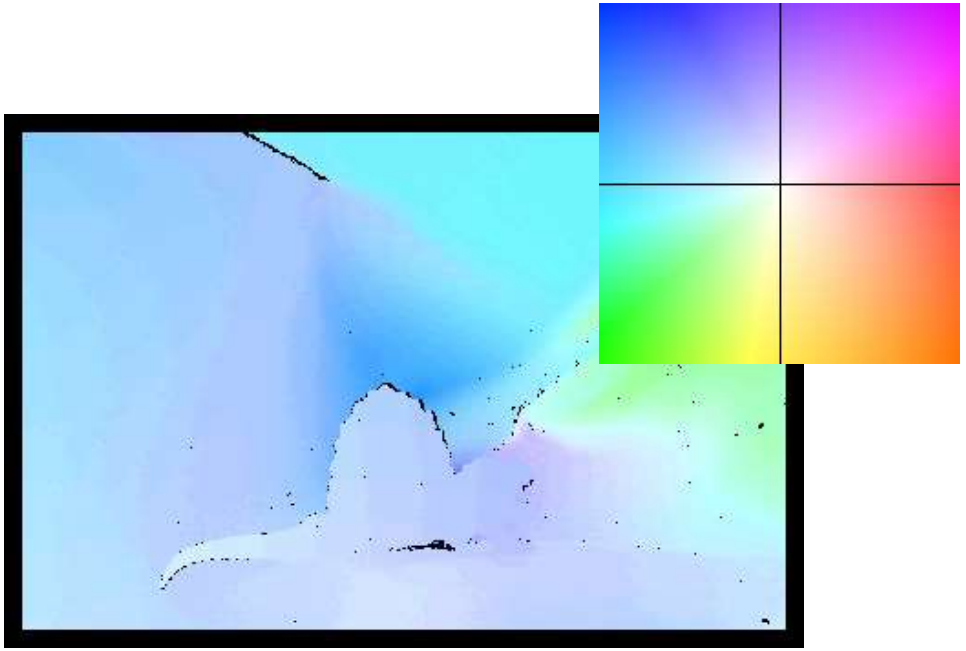
движения пикселей между кадрами

- Это задача **плотного** сопоставления
- Для каждой точки (x_{ij}, y_{ij}) на первом кадре нужно найти точку $(x_{ij} + u_{ij}, y_{ij} + v_{ij})$ на втором кадре

Визуализация



Вектора движения для отдельных точек или всего изображения



Цветовое кодирование вектора движения. Каждому направлению и амплитуде свой цвет и яркость

Оптический поток - векторное поле (u_{ij},v_{ij}) видимого (наблюдаемого) движения пикселей между кадрами

Middleburry optical flow dataset

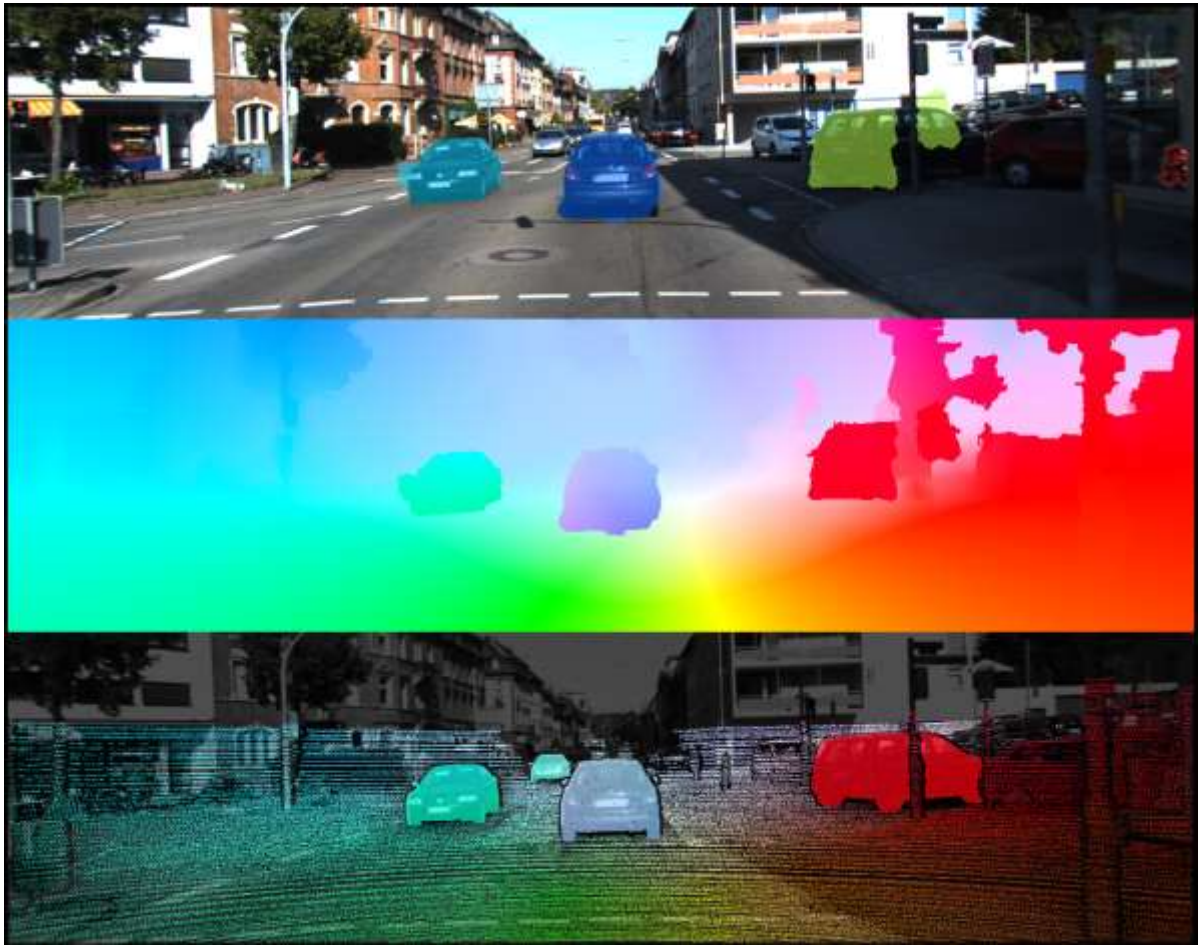
Как получить эталонные данные?

Покадровая съёмка в обычном и флуоресцентном свете

Синтетические данные

Source: S.Baker et. al. A Database and Evaluation Methodology for Optical Flow, IJCV, 2011

KITTI



- 3D сканирование
- Отдельно фон и движущиеся объекты
- Вписывание 3D моделей объектов в движущиеся объекты

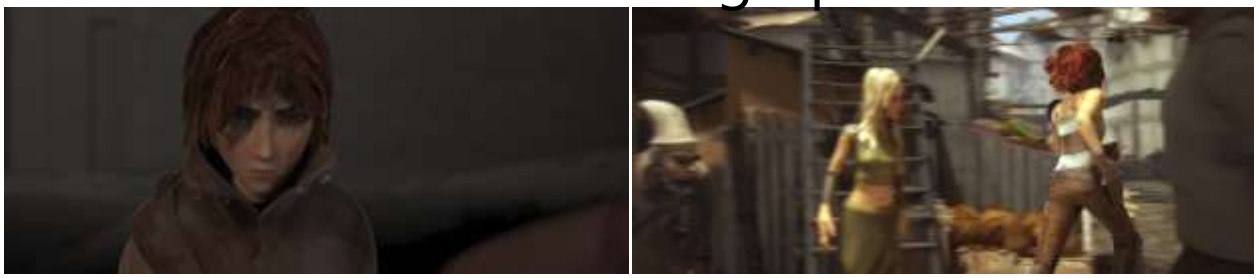
http://www.cvlibs.net/datasets/kitti/eval_flow.php

MPI Sintel

Open source 3D movie “Sintel”

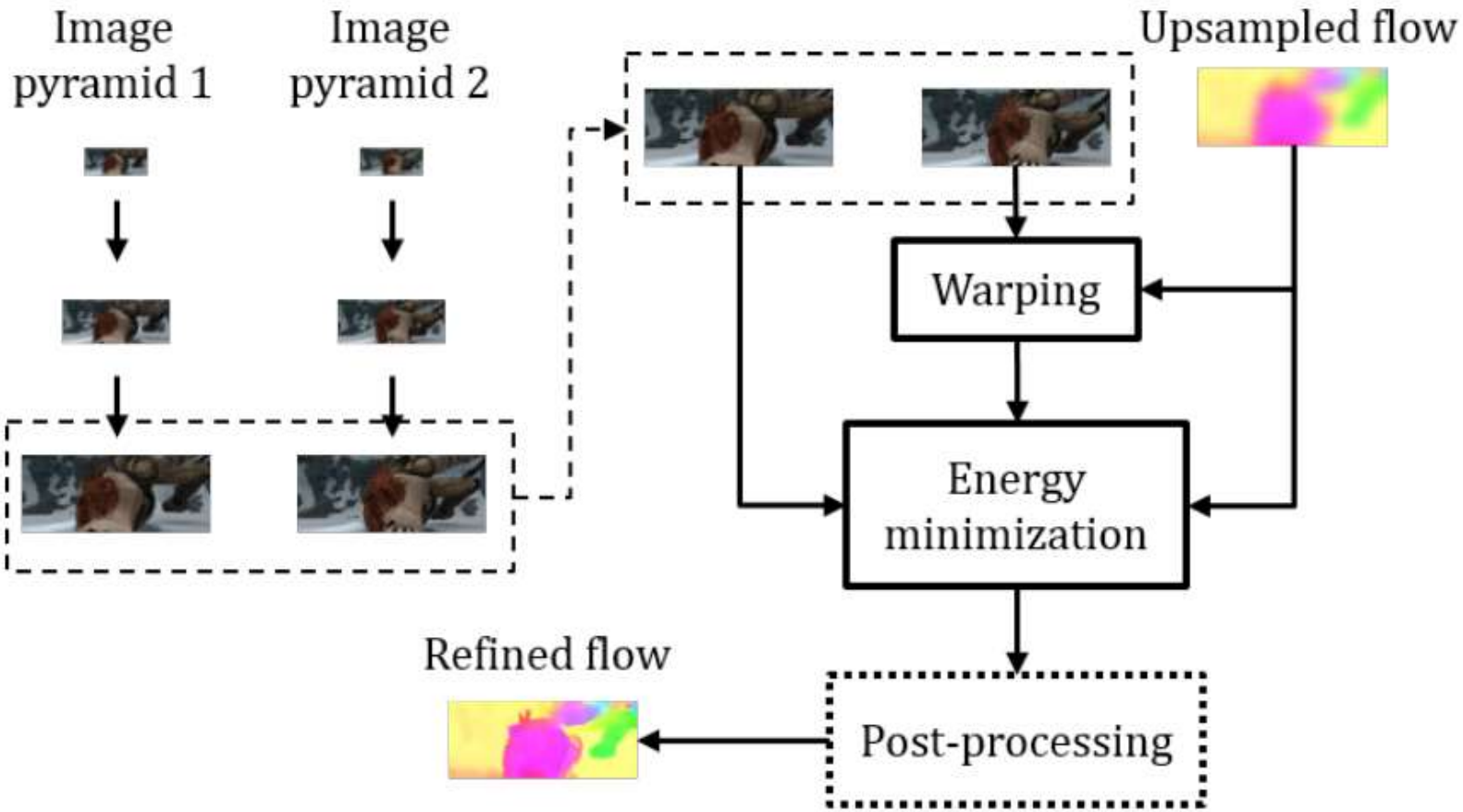
1064 training and 564 test frames

Various rendering options



Source: http://ps.is.tue.mpg.de/research_projects/mpi-sintel-flow
D. Butler et. al. A naturalistic open source movie for optical flow evaluation. ECCV 2012

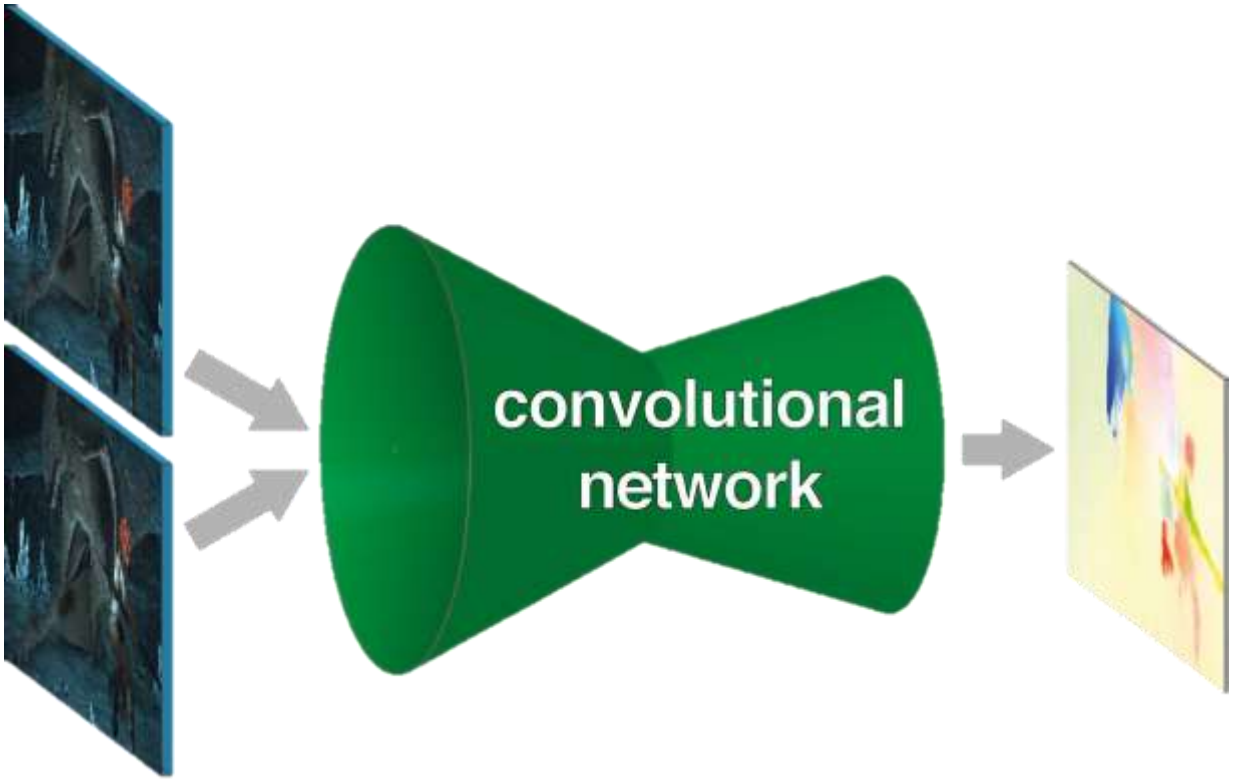
Классический подход



Source: https://arxiv.org/pdf/1709.02371.pdf

FlowNet

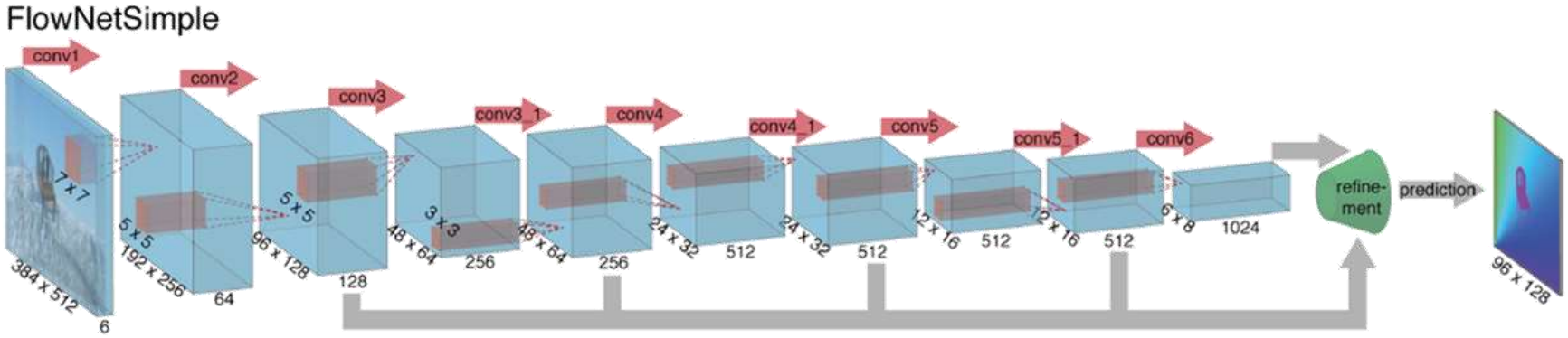
Прямолинейное применение свёрточных нейросетей к вычислению оптического потока



Source: https://arxiv.org/abs/1504.06852

A. Dosovitskiy et. al. FlowNet: Learning Optical Flow with Convolutional Networks. 2015

FlowNet



- Объединяем 2 кадра в 6и канальное изображение
- Применяем нейросеть
- Можно сделать сложнее, объединив признаки с 2х картинок с помощью специального слоя сравнения патчей

Source: arxiv.org/abs/1504.06852

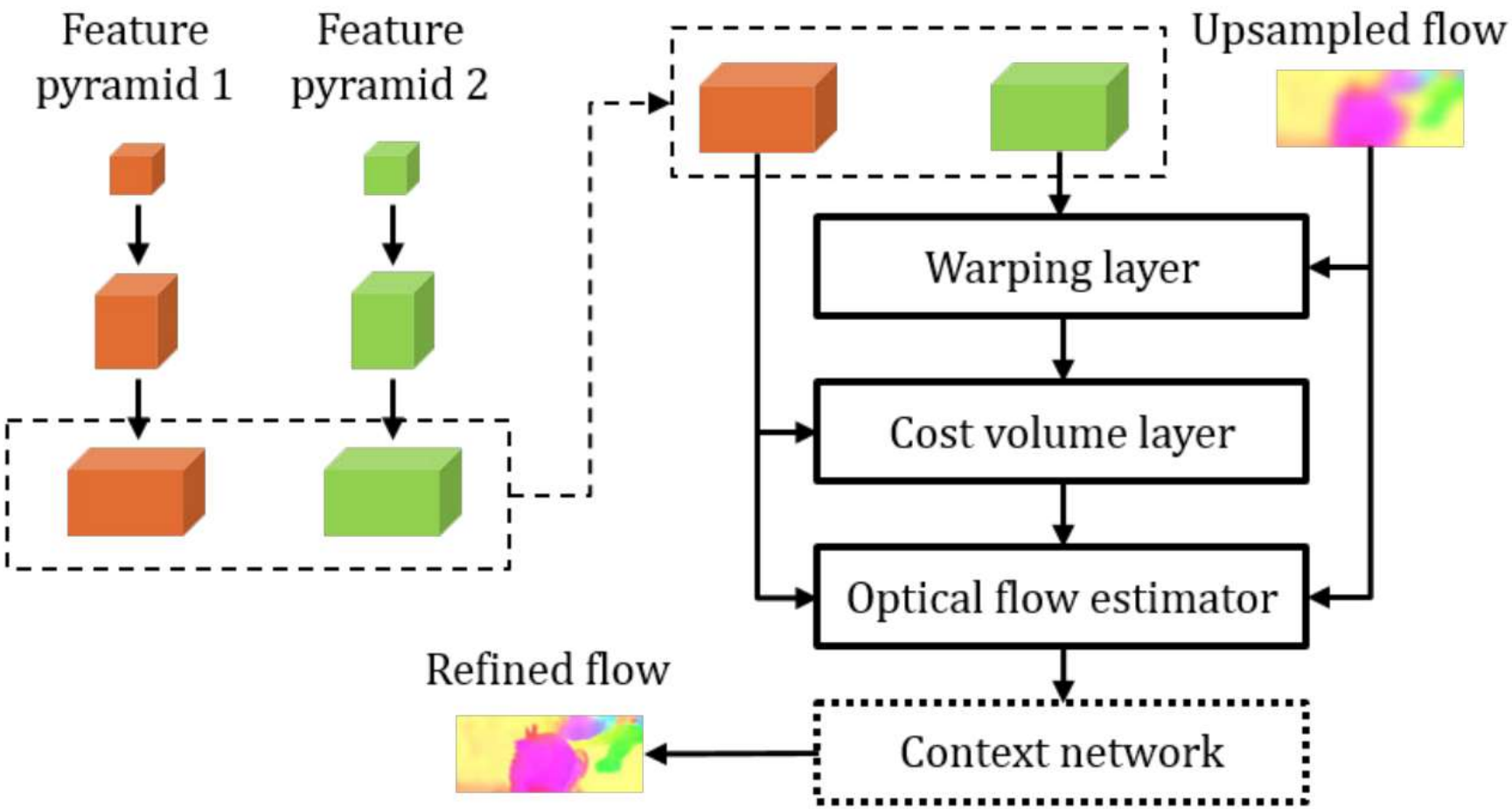
Flying chairs

	Frame pairs	Frames with ground truth	Ground truth density per frame
Middlebury	72	8	100%
KITTI	194	194	~50%
Sintel	1041	1041	100%
Flying Chairs	22872	22872	100%

- Не хватает данных для обучения
- Поэтому синтетический набор из летающих стульев

Source: arxiv.org/abs/1504.06852

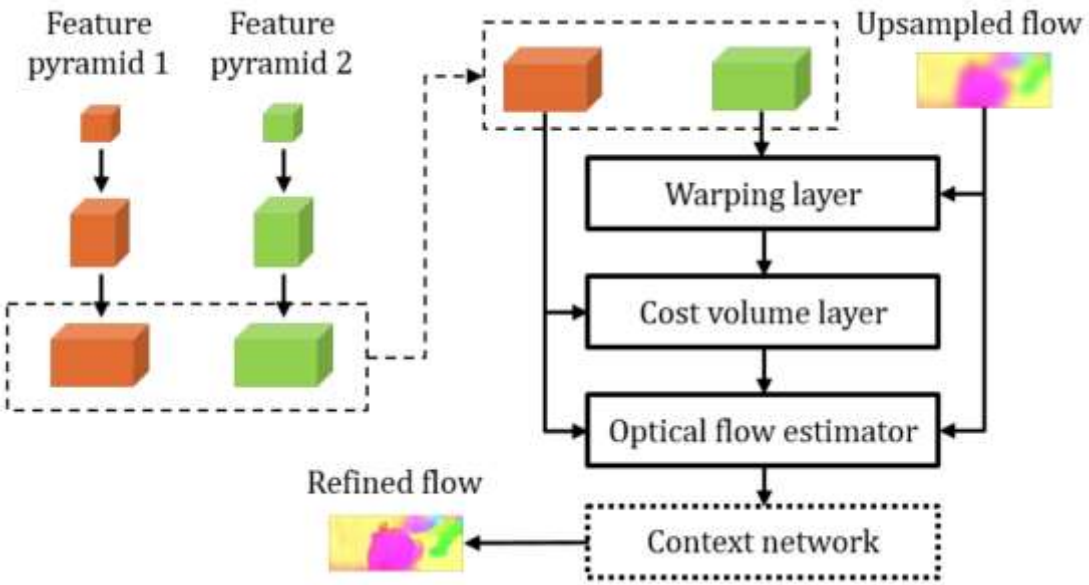
PWC-Net (Pyramid, Warping, Cost volume)



Реализуем классический подход, но через нейросетевые признаки и нейросетевой вывод

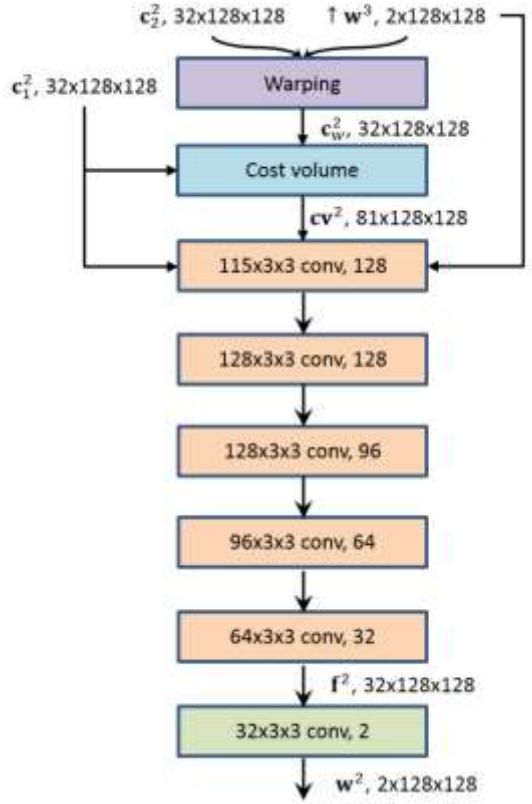
Sun et. al. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. CVPR2018

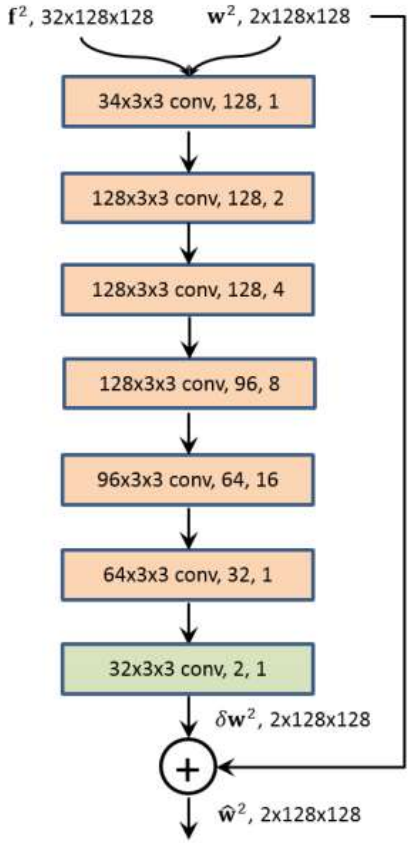
PWC-Net (Pyramid, Warping, Cost volume)



- Cost Volume – считаем корреляцию (произведение) признаков для сдвигов не более d
- Размер CV равен $d^2 * Width * Height$

Некоторые детали





OF Estimator берёт на вход Cost Volume, Optical Flow, и Features

- Сеть Context делает пост-обработку
- На вход признаки и OF

Evaluation ground truth: All pixels ▾ Evaluation area: All pixels ▾

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	PSPO			**4.35 %**	15.21 %	**6.15 %**	100.00 %	5 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
2	ISF			5.40 %	10.29 %	6.22 %	100.00 %	10 min	1 core @ 3 Ghz (C/C++)	
	A. Behl, O. Jafari, S. Mustikovela, H. Alhaija, C. Rother and A. Geiger: Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?. International Conference on Computer Vision (ICCV) 2017.									
3	PRSM		code	5.33 %	13.40 %	6.68 %	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
	C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.									
4	OSF+TC			5.76 %	13.31 %	7.02 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)	
	M. Neoral and J. Šochman: Object Scene Flow with Temporal Consistency. 22nd Computer Vision Winter Workshop (CVWW) 2017.									
5	SSF			5.63 %	14.71 %	7.14 %	100.00 %	5 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
	Z. Ren, D. Sun, J. Kautz and E. Sudderth: Cascaded Scene Flow Prediction using Semantic Segmentation. International Conference on 3D Vision (3DV) 2017.									
6	SOSF			5.42 %	17.24 %	7.39 %	100.00 %	55 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
7	OSF		code	5.62 %	18.92 %	7.83 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)	
	M. Menze and A. Geiger: Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.									
8	PWC-Net			9.66 %	9.31 %	9.60 %	100.00 %	0.03 s	NVIDIA Pascal Titan X	
9	MirrorFlow			8.93 %	17.07 %	10.29 %	100.00 %	11 min	4 core @ 2.2 Ghz (C/C++)	
	J. Hur and S. Roth: MirrorFlow: Exploiting Symmetries in Joint Optical Flow and Occlusion Estimation. ICCV 2017.									
10	FlowNet2			10.75 %	**8.75 %**	10.41 %	100.00 %	0.12 s	GPU Nvidia GeForce GTX 1080	
11	SDF			8.61 %	23.01 %	11.01 %	100.00 %	TBA	1 core @ 2.5 Ghz (C/C++)	
	M. Bai*, W. Luo*, K. Kundu and R. Urtasun: Exploiting Semantic Information and Deep Matching for Optical Flow. ECCV 2016.									
12	UnFlow			10.15 %	15.93 %	11.11 %	100.00 %	0.12 s	GPU @ 1.5 Ghz (Python + C/C++)	
	S. Meister, J. Hur and S. Roth: UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss. AAAI 2018.									
13	FSF+MS			8.48 %	25.43 %	11.30 %	100.00 %	2.7 s	4 cores @ 3.5 Ghz (C/C++)	
	T. Taniai, S. Sinha and Y. Sato: Fast Multi-frame Stereo Scene Flow with Motion Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017) 2017.									
14	CNNF+PMBP			10.08 %	18.56 %	11.49 %	100.00 %	45 min	1 cores @ 3.5 Ghz (C/C++)	
15	MR-Flow		code	10.13 %	22.51 %	12.19 %	100.00 %	8 min	1 core @ 2.5 Ghz (Python + C/C++)	
	J. Wulff, L. Sevilla-Lara and M. Black: Optical Flow in Mostly Rigid Scenes. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017.									

Figure 8. Screenshot of the KITTI 2015 public table. PWC-Net has the lowest percentage of error (Fl-all) among all optical flow methods, only inferior to scene flow methods that use additional stereo input information.

Visual object tracking (VOT)

Отслеживание объектов в видео

- Мы хотим отследить движения объектов по всем кадрам и определить траектории их движения

- Выход – «следы» (track) объектов, последовательность локаций в каждом кадре видео

Visual object tracking (VOT)

- Один объект
- Объект локализован на первом кадре
- Mode-free tracking – мы не знаем ничего про объект, кроме его положения на первом кадре
  - Не можем находить объекты на новых кадрах
- Сопровождение на коротких интервалах
- Только предыдущие кадры могут использоваться

Сложность задачи

- Вычислительная нагрузка
  - Нужно обрабатывать N кадров в секунду
- Изменение по времени
  - Вид объекта меняется от кадра к кадру из-за ракурса, изменения освещения, внутренних изменений (скейтбордист)
- Взаимодействие объектов
  - Перекрытия объектов
  - Визуальное сходство объектов
  - И т.д.

VOT Challenge

Основной конкурс для оценки методов визуального трекинга http://votchallenge.net/

- Открытые реализация
- Matlab-toolkit для оценки
- Оценка качества и скорости
- Небольшой, но разнообразный набор роликов
- Короткие ролики (100 frames)

Развитие:

- Добавление других модальностей (RGB + Depth, RGB + IR)
- Появление «долгого» отслеживания, с пропаданием и появлением объектов

Примеры последовательностей

Most challenging:

Matrix

Rabbit

Butterfly

Least challenging:

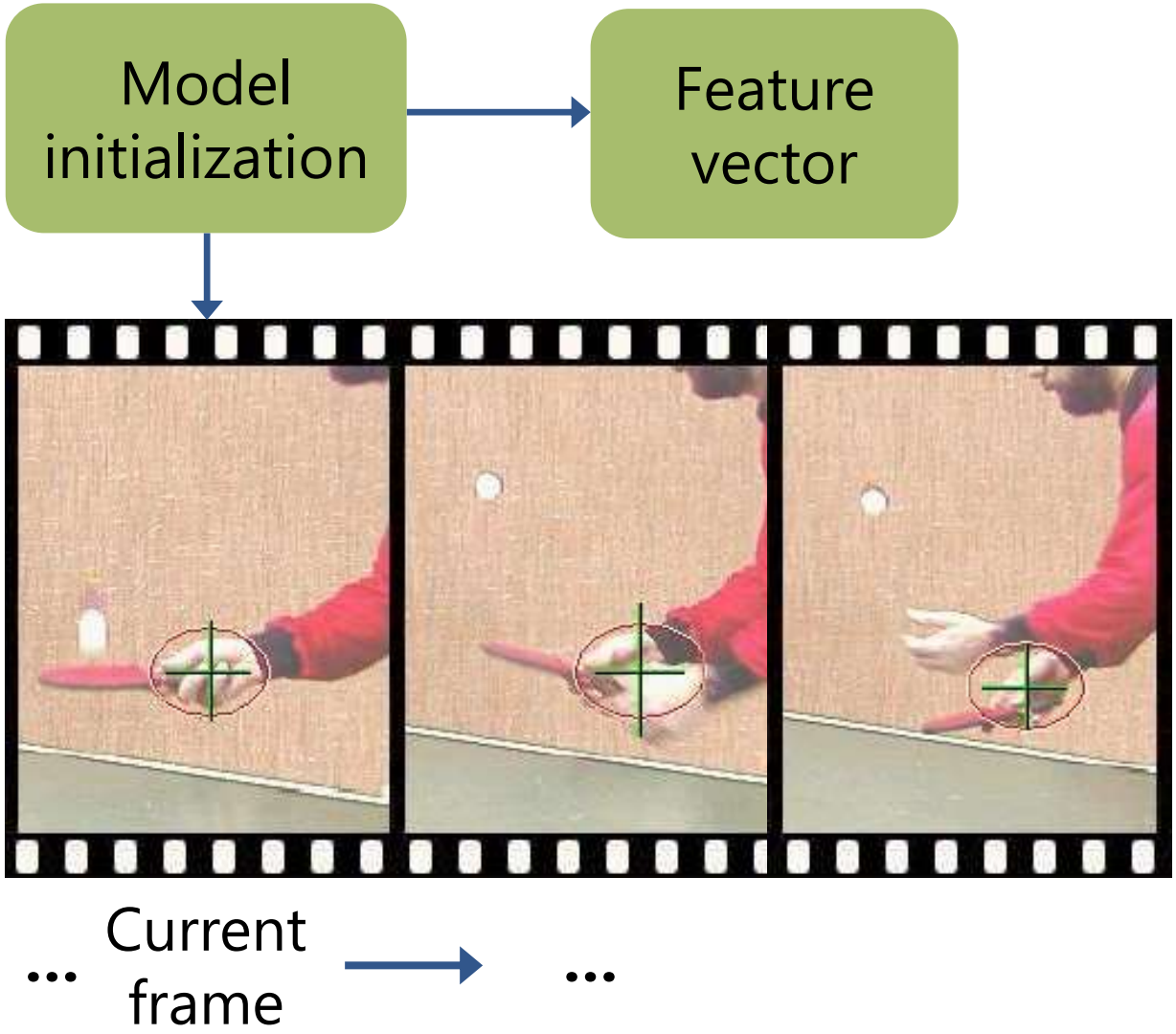
Singer

Octopus

Sheep

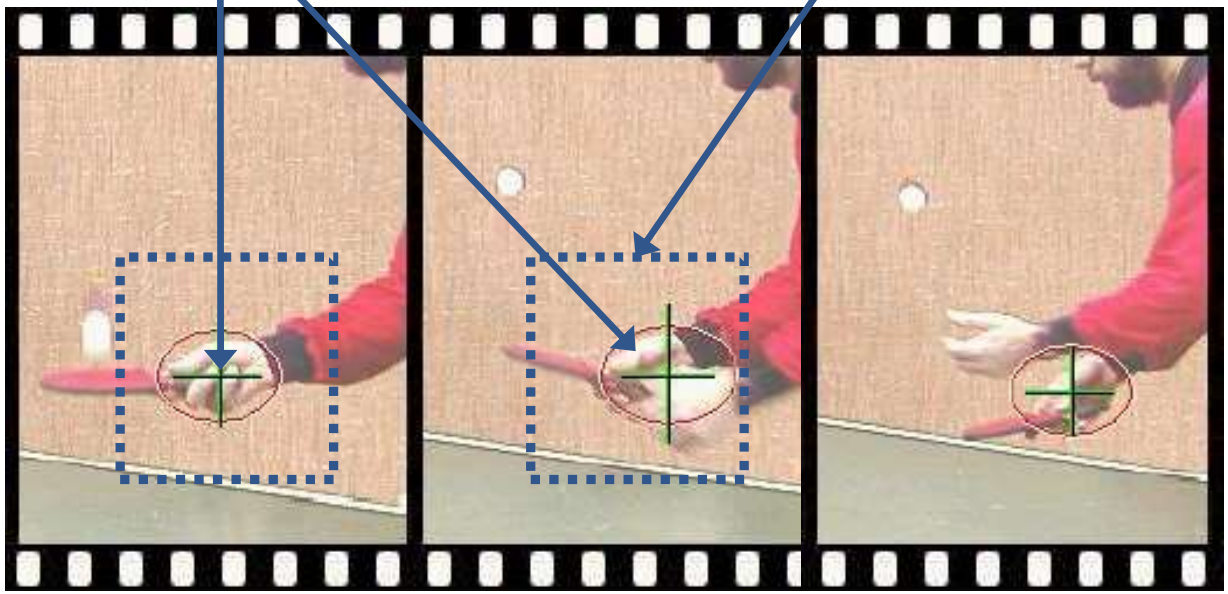
Source: http://votchallenge.net/

Стандартная схема итеративного трекинга



Стандартная схема итеративного трекинга

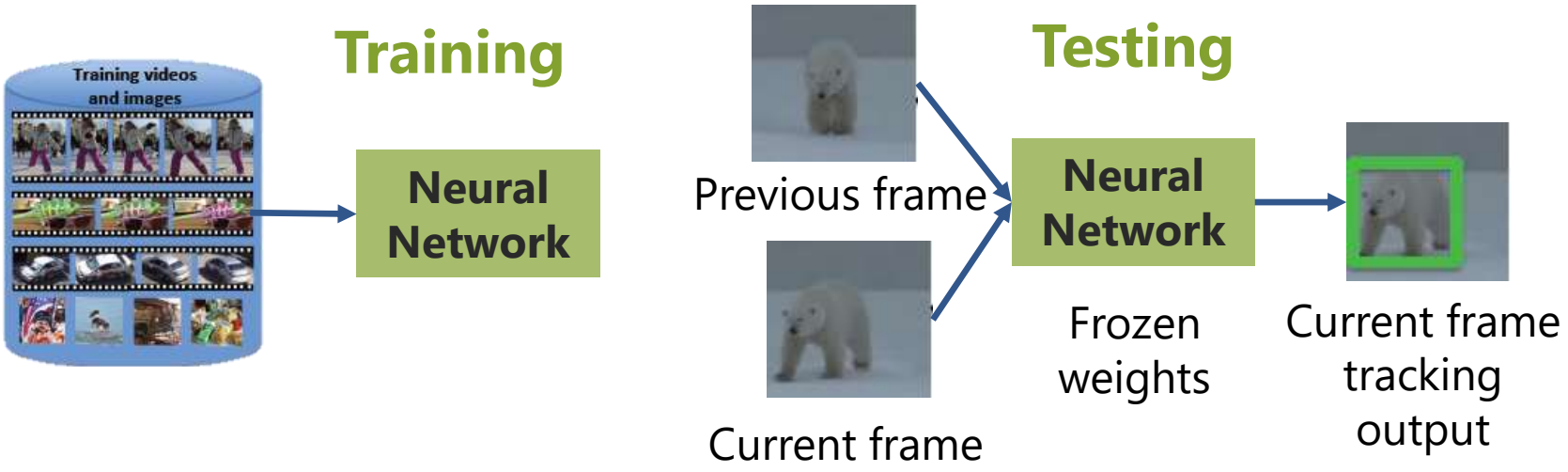
Start from current position → Search in local neighborhood → Maximizing visual similarity



Repeat for next pair of frames

... Previous frame → Current frame → Next frame

GOTURN

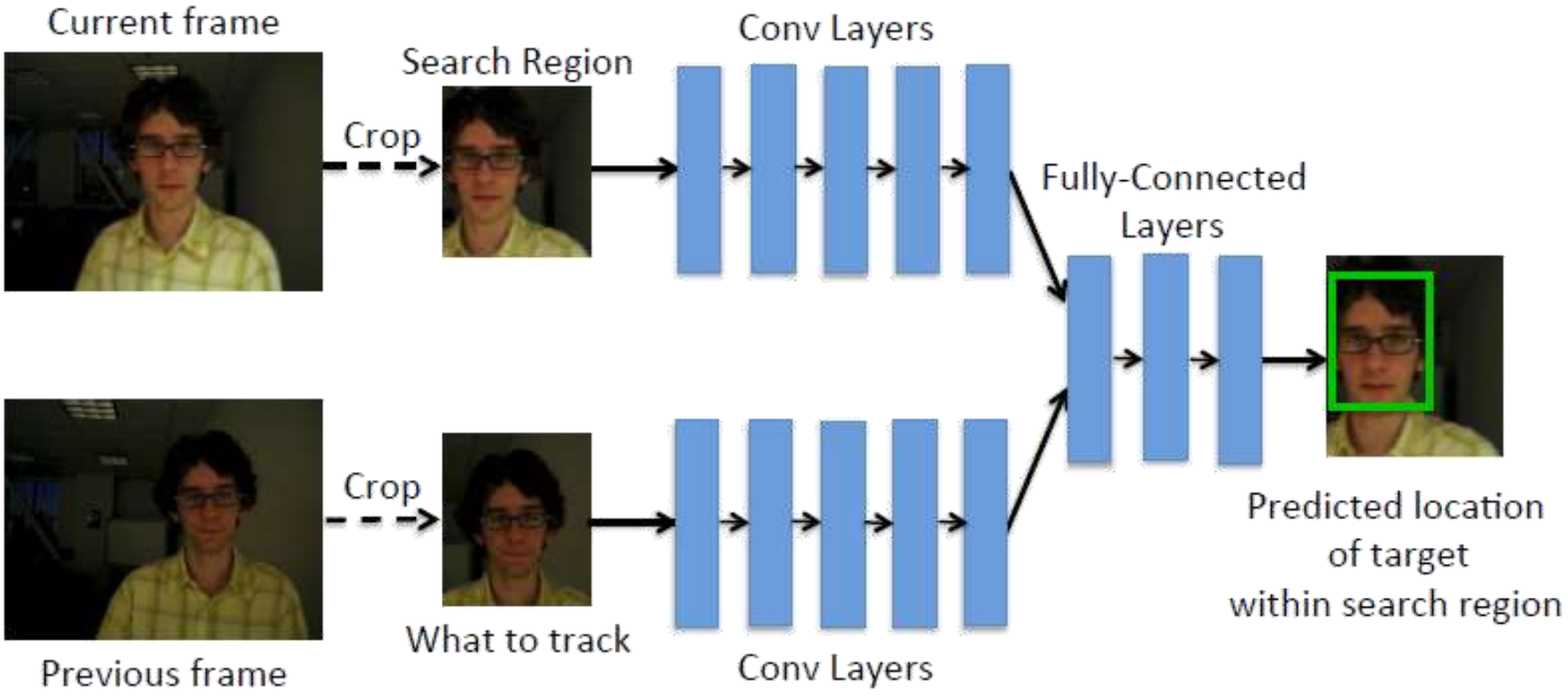


Generic Object Tracking Using Regression Networks (GOTURN):

- Обучаем нейросеть на коллекции видео и изображений с bounding box
- Просто применяем её к паре кадров и за счёт этого получаем скорость 100 кадров/сек

Source: Held et.al. Learning to Track at 100 FPS with Deep Regression Networks, ECCV 2016

GOTURN network



Source: Held et.al. Learning to Track at 100 FPS with Deep Regression Networks, ECCV 2016

Training GOTURN

Previous video frame centered on object

Current video frame, shifted, with ground-truth bounding box

Image centered on object

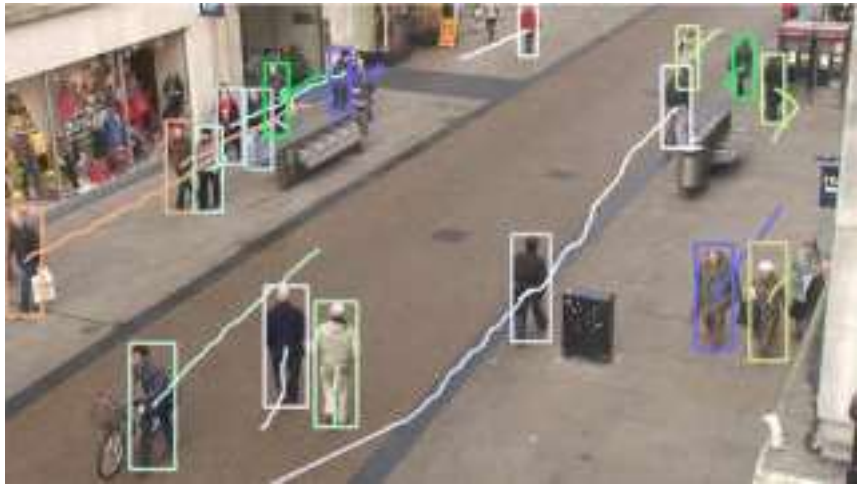
Shifted image with ground-truth bounding box

Small shifts are sampled more often than large shifts

Source: Held et.al. Learning to Track at 100 FPS with Deep Regression Networks, ECCV 2016

Multiple object tracking (MOT)

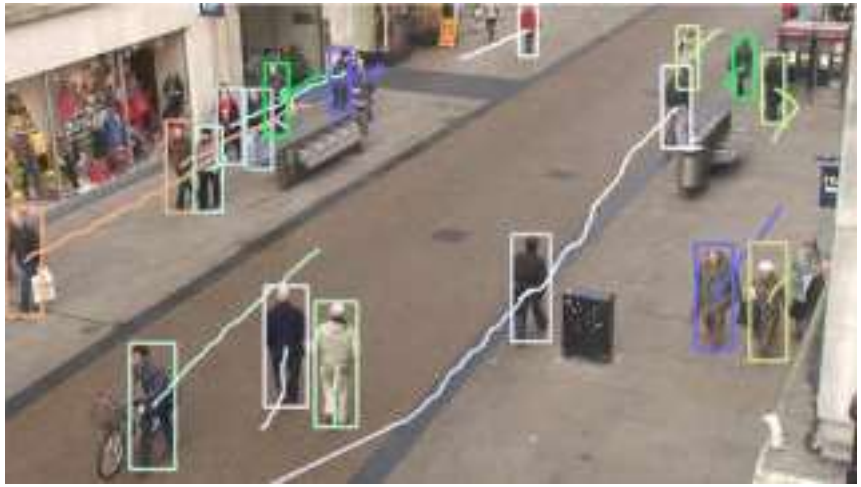
Multiple object tracking (MOT)



- Работаем со множеством объектов
- На длительных промежутках времени
- Варианты :
  - Detection Based Tracking (DBT)
  - Detection Free Tracking (DFT)

Source: http://www.robots.ox.ac.uk/~lav/Papers/benfold_reid_cvpr2011/benfold_reid_cvpr2011.html

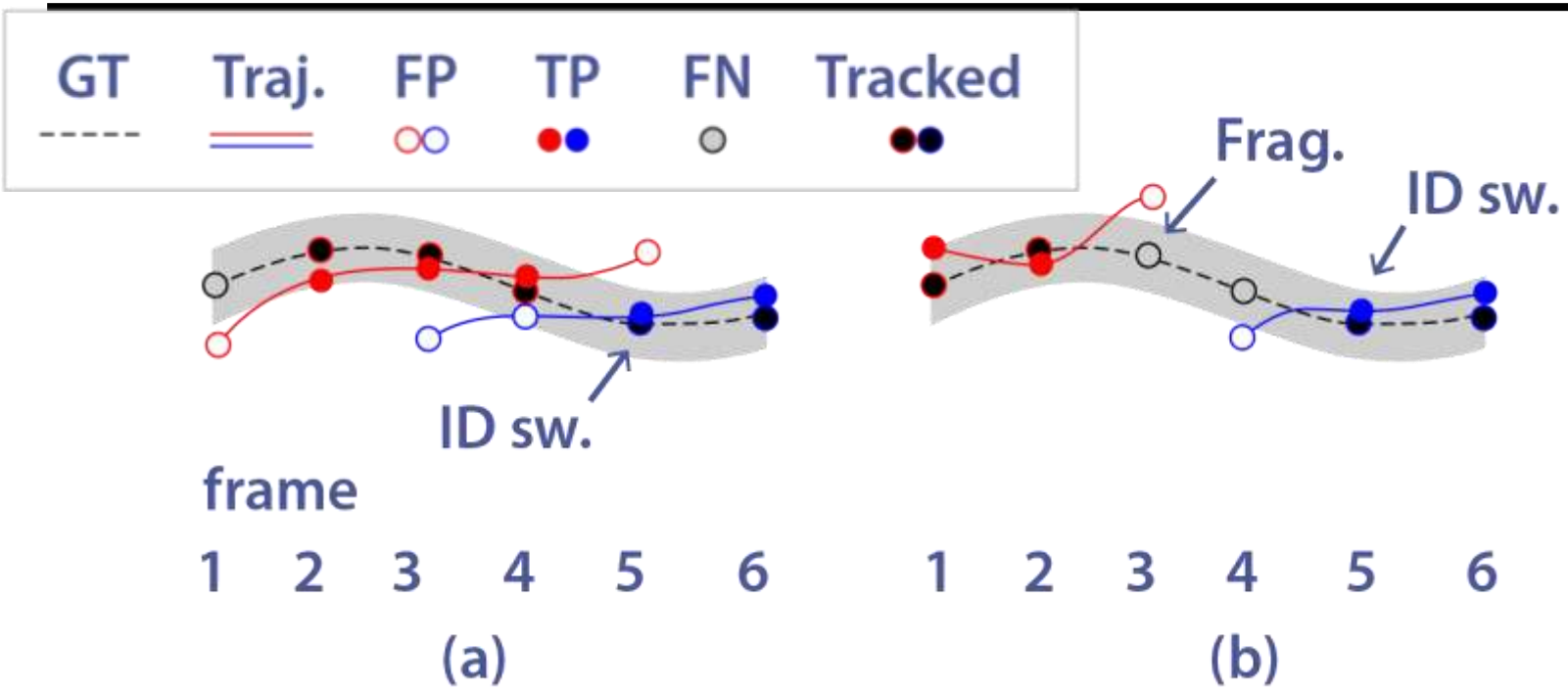
Выход MOT



- Набор траекторий объектов для всех объектов заданного типа, которые видны в видео
- Положение объекта обычно описывается с помощью bounding box

Source: http://www.robots.ox.ac.uk/~lav/Papers/benfold_reid_cvpr2011/benfold_reid_cvpr2011.html

Ошибки MOT



- ID switches – для одной эталонной траектории выдается две и более траектории
- Fragmentations – для одной эталонной траектории выдается две траектории с пропуском между ними, т.е. объект не виден на ряде кадров

Метрики качества

Multiple Object Tracking Accuracy (MOTP) – надежность построения траектории

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}$$

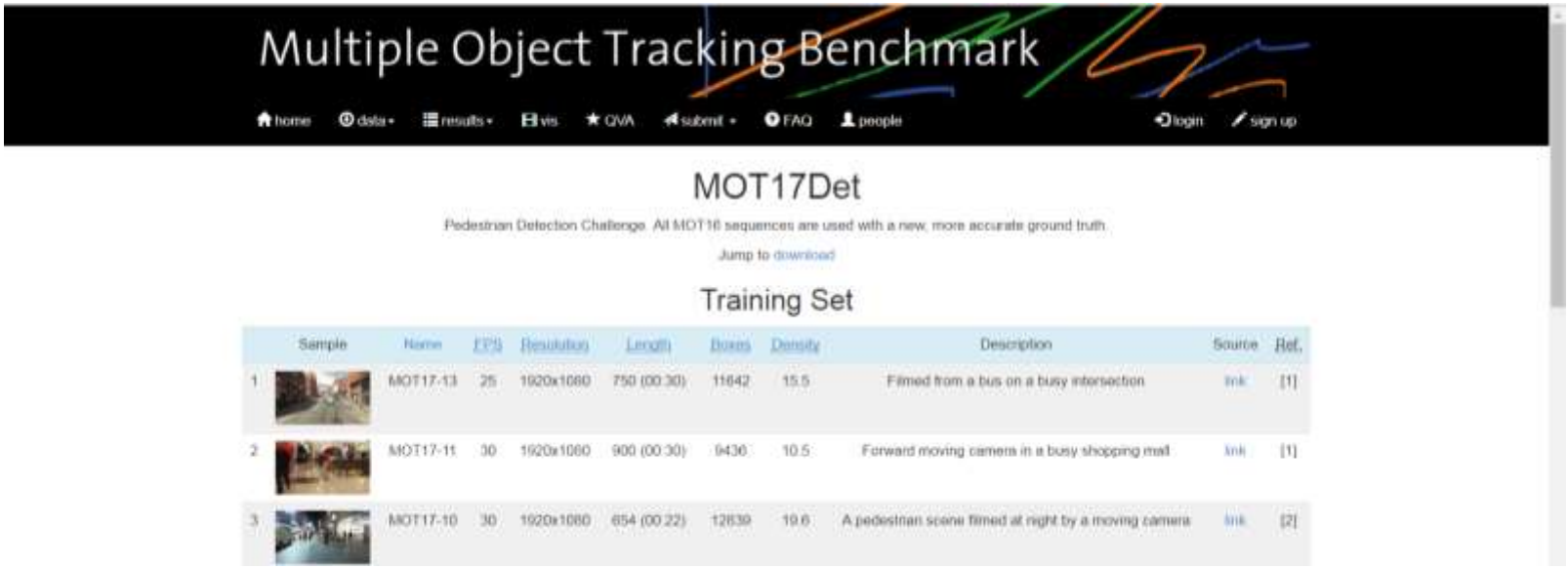
Multiple Object Tracking Precision (MOTP) – точность локализации объектов

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$$

Метрики качества

- Mostly tracked (MT)
  - Объект успешно сопровождался > 80% длины эталонной траектории
- Mostly lost (ML)
  - Объект успешно сопровождался < 20% длины эталонной траектории
- Partially tracked (PT)
  - Все остальные

MOTChallenge 2015 - 2017



- Comprehensive dataset for object detection (MOT17Det) and tracking (MOT17) evaluation
- 15948 frames (645 sec.) for training, 17757 frames (744 sec) for testing
- 112297 bboxes for detector training, 188076 bboxes for detector testing

Source: https://motchallenge.net

Duke Multi-Target, Multi-Camera

- 8 static cameras x 85 minutes of 1080p 60 fps video
- More than 2,000,000 manually annotated frames
- More than 2,000 identities
- Manual annotation by 5 people over 1 year
- More identities than all existing MTMC datasets combined
- Unconstrained paths, diverse appearance

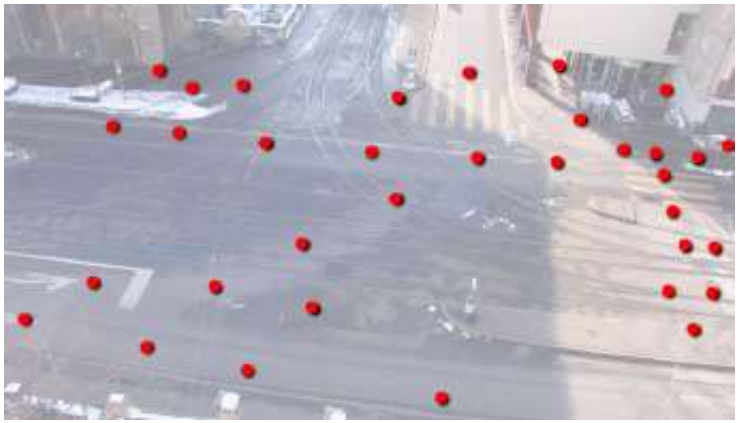
Source: http://vision.cs.duke.edu/DukeMTMC

UA-DETRAC Benchmark

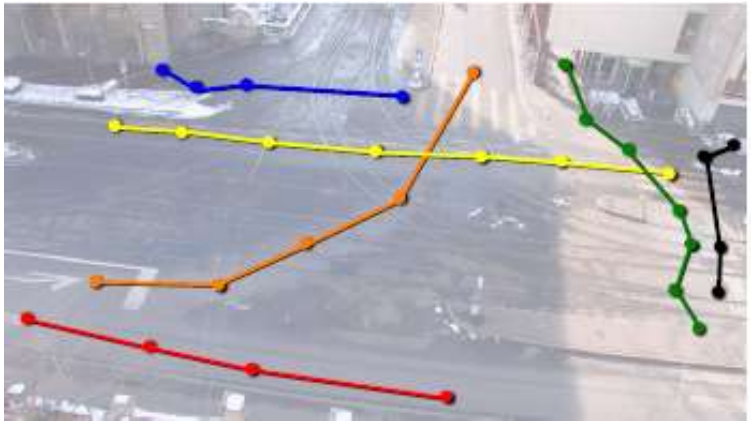
- 10 hours of videos, 25 fps, resolution 960×540 pixels
- 24 different locations at Beijing and Tianjin in China
- > 140000 frames
- 8250 vehicles
- 1.21M labeled boxes

Source: http://detrac-db.rit.albany.edu/

Detection by tracking



Детектирование объектов



Ассоциация обнаружений

Сопоставления обнаружения и траектории (треки)

Source: S.Manen Leveraging single for multi-target tracking using a novel trajectory overlap affinity measure. WACV 2016

Типичные проблемы

- Ошибки детектора – пропуски обнаружений (FN) и ложные срабатывания
- Хороший трекер должен устранять эти проблемы
  - Заполнение пропусков в детекциях
  - Фильтрация ложных срабатываний

Ограничения трекеров

- MOT Challenge findings:
  - 18% of tracks are not covered by detections at all
  - 37% of tracks are covered by low-confidence detections
  - Trackers reduce FP and raise FN
- Detector is a key!

Важность хорошего детектора

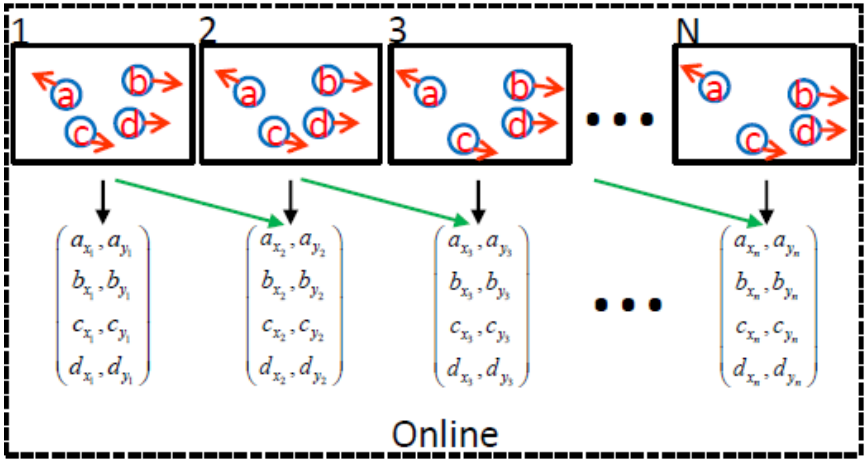
Tracker	Avg.Rank	↑MOTA		MOTP
HCC 1. ✓	9.6	**49.3**	±10.2	79.0
LMP 2. ✓	12.8	48.8	±9.8	**79.0**
FWT 3. ✓	21.7	47.8	±9.4	75.5
				R. Henschel, L. I
DeepSORT_2 13. ○	10.3	61.4	±10.6	79.1
TMO 14. ○	10.6	61.4	±10.1	79.3
SORTwHPD16 15. ○	10.2	59.8	±10.3	79.6

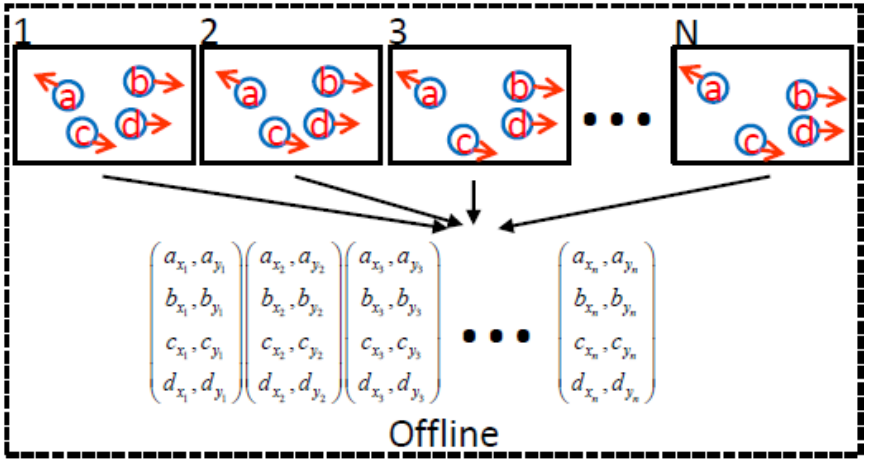
With DPM (Deformable part models) detector

With Faster R-CNN detector

Source: https://motchallenge.net

Online vs offline tracking





On-line tracking

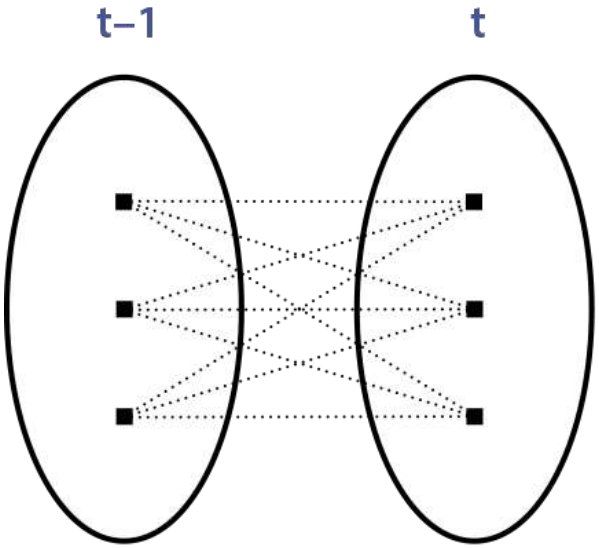
(Only current and previous frames are available)

Off-line (batch) tracking

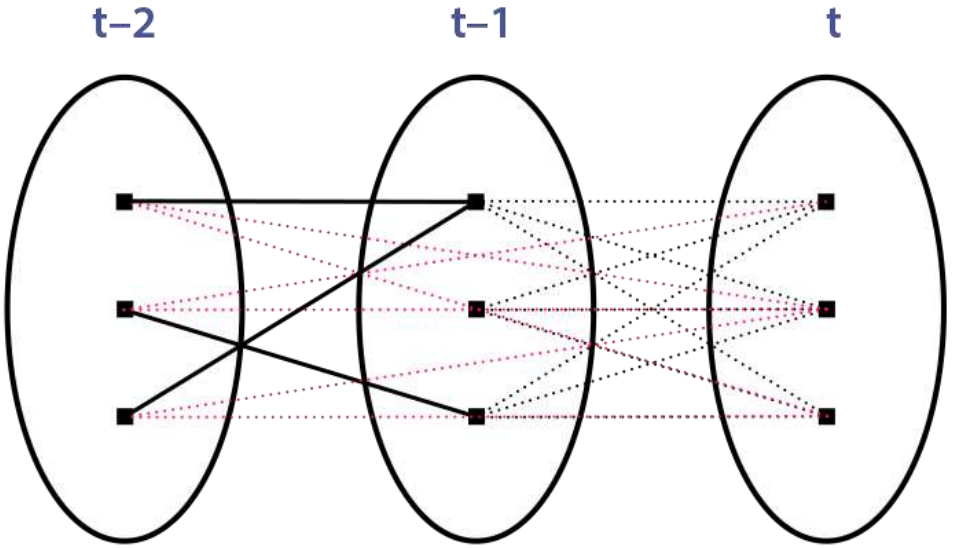
(All frames (including future) are available)

Source: https://arxiv.org/abs/1409.7618

Data association

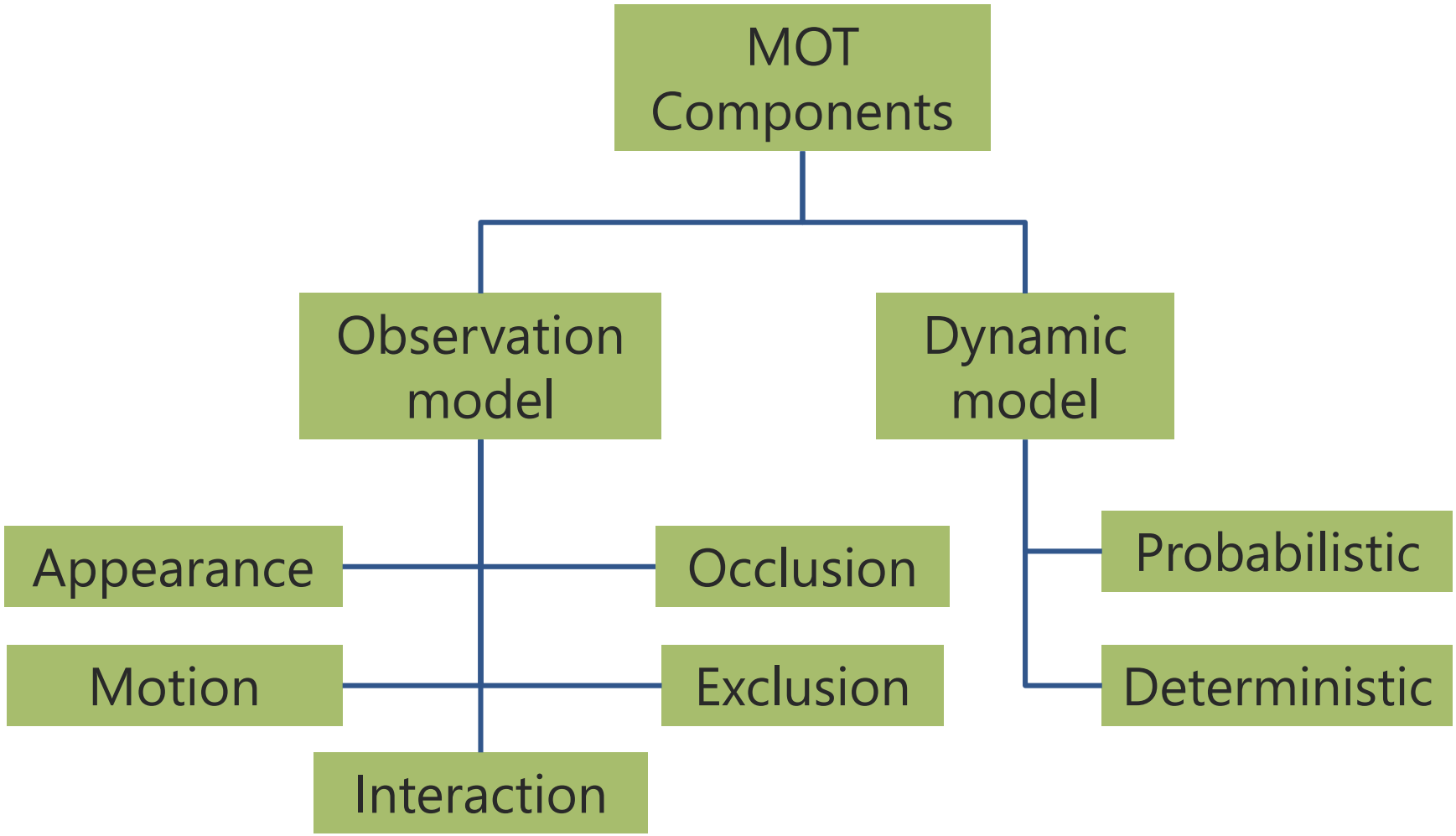


Two-frame methods

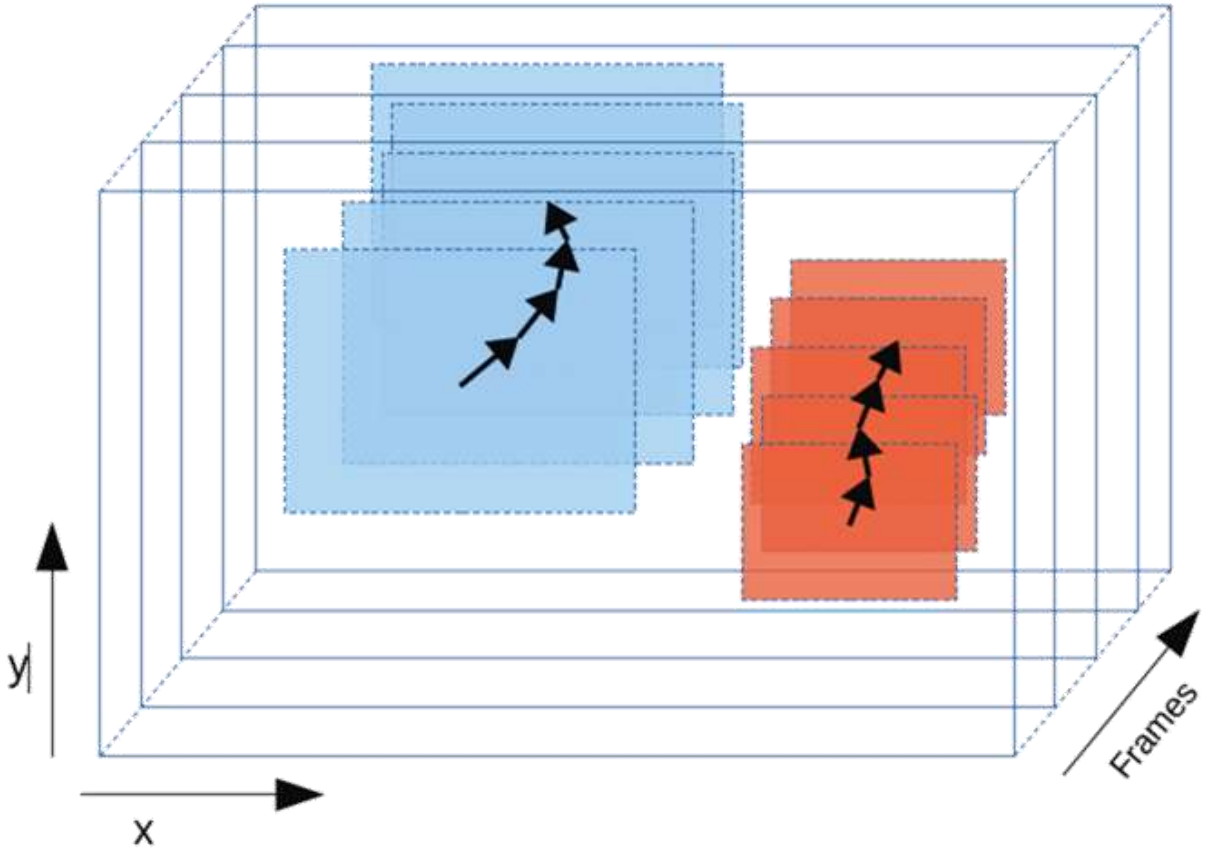


Multi-frame methods

Компоненты функции сходства (affinity)



Ассоциация прямо по IoU



Matching object in neighouring frames by bounding box overlap

Source: Bochinski et. al. High-Speed Tracking-by-Detection Without Using Image Information. AVSS 2017

Algorithm

Algorithm 1 IOU Tracker

Inputs:
$D = \{D_0, D_1, ..., D_{F-1}\} =$
$\{\{d_0, d_1, ..., d_{N-1}\}, \{d_0, d_1, ..., d_{N-1}\}, ...\}$
Initialize:
$T_a = \emptyset, T_f = \emptyset$
$D = \{\{d_i | d_i \in D_j, d_i \geq \sigma_l\} | D_j \in D\}$
for f = 0 to F :
    for $t_i \in T_a$:
        $d_{best} = d_j$ where $max(IOU(d_j, t_i)), d_j \in D_f$
        if $IOU(d_{best}, t_i) \geq \sigma_{IOU}$:
            add d_{best} to t_i
            remove d_{best} from D_f
        else
            if $highest_score(t_i) \geq \sigma_h$
              and $len(t_i) \geq t_{min}$:
                add t_i to T_f
            remove t_i from T_a
    for $d_j \in D_t$:
        start new track t with d_j and insert into T_a
for $t_j \in T_A$:
    if $highest_score(t_i) \geq \sigma_h$ and $len(t_i) \geq t_{min}$:
      add t_i to T_f
return T_f

Source: Bochinski et. al. High-Speed Tracking-by-Detection Without Using Image Information. AVSS 2017

Результаты

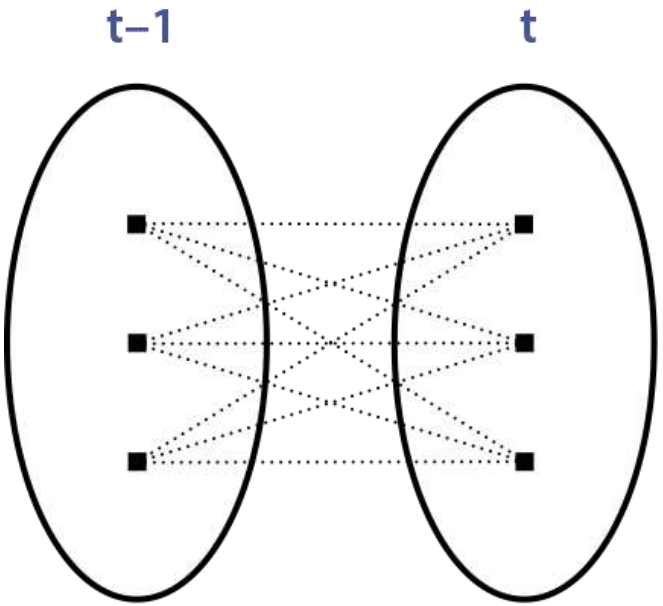
Tracker	Detector	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDs	PR-FM	PR-FP	PR-FN	Speed
Overall (Easy + Medium + Hard)										
CEM [1]	CompACT	5.1%	35.2%	3.0%	35.3%	**267.9**	**352.3**	**12341.2**	260390.4	4.62 fps
CMOT [3]	CompACT	12.6%	36.1%	16.1%	18.6%	285.3	1516.8	57885.9	**167110.8**	3.79 fps
GOG [12]	CompACT	14.2%	37.0%	13.9%	19.9%	3334.6	3172.4	32092.9	180183.8	390 fps
DCT [2]	R-CNN	11.7%	38.0%	10.1%	22.8%	758.7	742.9	336561.2	210855.6	0.71 fps
H^2T [18]	CompACT	12.4%	35.7%	14.8%	19.4%	852.2	1117.2	51765.7	173899.8	3.02 fps
IHTLS [6]	CompACT	11.1%	36.8%	13.8%	19.9%	953.6	3556.9	53922.3	180422.3	19.79 fps
IOU	R-CNN	16.0%	**38.3%**	13.8%	20.7%	5029.4	5795.7	22535.1	193041.9	**100,840 fps**
IOU	EB	**19.4%**	28.9%	**17.7%**	**18.4%**	2311.3	2445.9	14796.5	171806.8	6,902 fps
Easy (AVSS17 Beginner Challenge)										
CEM [1]	Average	7.2%	42.0%	3.4%	42.3%	96.4	119.6	4253.3	63296.3	-
CMOT [3]	Average	16.6%	43.8%	20.0%	23.0%	**68.3**	327.9	16282.2	39467.6	-
GOG [12]	Average	20.1%	44.6%	17.5%	24.0%	1019.0	981.2	8423.5	42065.9	-
DCT [2]	Average	16.3%	44.1%	9.6%	34.3%	73.5	**69.4**	4754.6	51393.3	-
H^2T [18]	Average	17.1%	42.9%	17.8%	24.7%	298.8	305.5	12866.0	42086.5	-
IHTLS [6]	Average	14.8%	43.0%	15.5%	25.3%	299.5	1102.3	13839.9	43954.4	-
IOU	R-CNN	29.3%	**47.2%**	25.0%	**17.3%**	1112.5	1261.0	3457.6	**33394.1**	**117,340 fps**
IOU	EB	**34.0%**	37.8%	**27.9%**	20.4%	573.6	603.7	**1617.0**	33760.8	9,002 fps
Medium + Hard (AVSS17 Experienced Challenge)										
IOU	R-CNN	11.8%	**36.5%**	8.9%	25.0%	3693.1	4228.3	16634.7	168527.2	**87,906 fps**
IOU	EB	**16.4%**	26.7%	**14.8%**	**18.2%**	**1743.2**	**1846.3**	**12627.0**	**136077.8**	6,069 fps

State-of-the-art on DETRACK-test

Source: Bochinski et. al. High-Speed Tracking-by-Detection Without Using Image Information. AVSS 2017

Simple Online and Realtime Tracking (SORT)

- CNN-based object detector
- Kalman filter for prediction of object position in current frame based on positions in previous frames
- Hungarian algorithm for matching object detections in current frames with predicted positions
- IoU of detected and predicted bounding boxes as affinity measure for matching detection and track



A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft. Simple Online and Realtime Tracking. In arXiv:1602.00763, 2016.

Influence of motion prediction

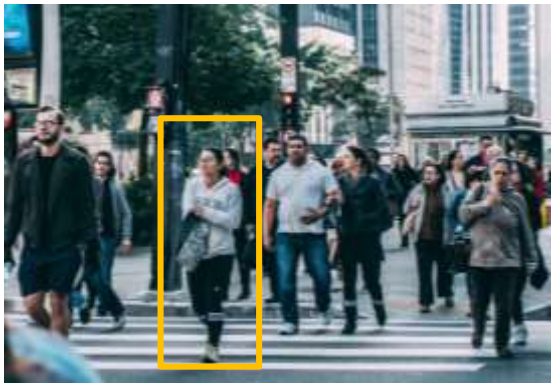
	MOTA	MOTP	MT	ML	FP	FN	ID sw	Frag	Hz
SORT	**59.8**	**79.6**	**25.4%**	**22.7%**	8698	**63245**	**1423**	**1835**	59.5
IOU	57.1	77.1	23.6%	32.9%	**5702**	70278	2167	3028	**3004**

Comparison of SORT to simple IoU tracker:

- Less False Negatives (FN), Fragmentations (Frag), ID Switches (ID Sw), Mostly Tracked (MT)
- Increase in MOTA and MOTP
- Increase in False Positive (bad)

Source: https://motchallenge.net

Re-identification



Detections in video

Probe

Gallery

Matches

SORT + DA

	MOTA	MOTP	MT	ML	FP	FN	ID sw	Frag	Hz
SORT	59.8	79.6	25.4%	22.7%	**8698**	63245	1423	**1835**	**59.5**
Deep SORT	**61.4**	79.1	**32.8%**	**18.2%**	12852	**56668**	**781**	2008	40

Addition of re-identification to affinity between detections and tracks:

- Reduces ID switches (ID sw), False negatives (FN), Mostly Lost (ML) and increases Mostly Tracked (MT)
- Somewhat raises False Positives (FP) and Fragmentations (Frag)

Source: https://arxiv.org/pdf/1703.07402.pdf

Распознавание событий

Action recognition

Human actions are the main content of movies, TV news and shows, home video and video surveillance

- Smart surveillance
  - *Abnormal situation detection*
- Video archive indexing and retrieval
  - *Search for a scene with Putin and Obama handshake*
- Content navigation
  - *Rewind to the next goal in the soccer match*

Source: http://www.di.ens.fr/~laptev/actions/hollywood2/

Human action

The most basic actions are simple body movements like walking or running or clapping hands.

Walking

Jogging

Running

Boxing

Waving

Clapping

Source: http://www.nada.kth.se/cvap/actions/

Actions

Short meaningful movements

Answer phone

Handshake

Actions and events

A set of small actions with a specific common goal can still be called an “action” but we can also call them “events”

Making sandwich

Doing homework

Source: http://trecvid.nist.gov/

Events

An event can include a lot of different actions of different people

Birthday party

Parade

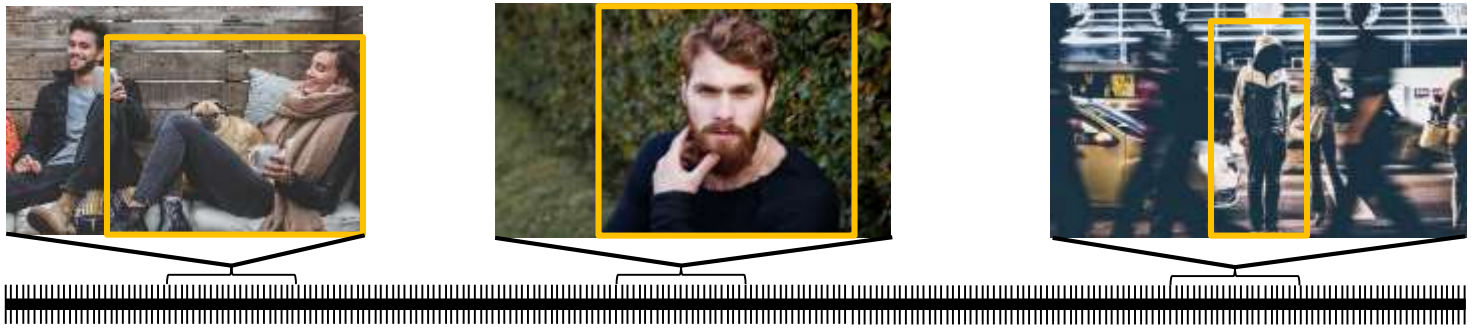
Source: http://trecvid.nist.gov/

Tasks

Action or event classification: assign a label of action or event to a video

Making sandwich: present
Feeding animal: not present
…

Action localization: search for a spatial region and time interval of a specific action in a video



Source: http://trecvid.nist.gov/

Actions and motions

- Actions are usually defined by motions
- But many visually similar motions can have very different meanings and correspond to different actions

Source: http://www.di.ens.fr/willow/teaching/recvis11/slides/lecture10_video.pdf

Datasets

It is much easier to collect and annotate datasets for action recognition compared to object tracking or optical flow estimation

Kiss

Drink

Source: http://www.di.ens.fr/~laptev/actions/hollywood2/

KTH Actions (2004)

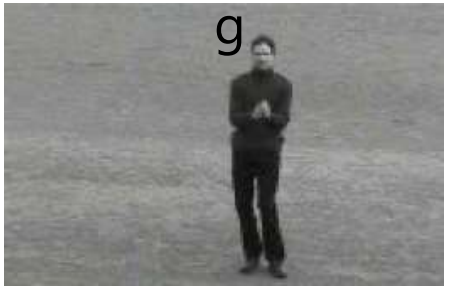
Walking

Jogging

Runnin
g

Boxing

Waving



Clapping

- 25 people, 6 actions, 4 scenes (indoor, outdoor, outdoor with different scale, outdoor with different clothes)
- 2391 video

Source: C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. ICPR, 2004.

UCF101 (2012)

- 13320 videos from Youtube in 101 classes
- Five groups: (1)Human-Object Interaction; (2) Body-Motion Only; (3) Human-Human Interaction; (4) Playing Musical Instruments; (5) Sports.
- Examples: to dye a lips, to tint eyes
- Accuracy from 43.9 (baseline Oct 2013) to 87.9% (Oct 2014)

Source: Khurram Soomro, Amir Roshan Zamir and Mubarak Shah, UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild., CRCV-TR-12-01, November, 2012.

TrecVid MED'13

- 100 positive video clips per event category, 5000 negatives
- Testing on 98000 videos clips, i.e. 4000 hours
- 20 known events, 10 adhoc events
- Videos from publicly available, user-generated content on various Internet sites
- Various descriptors for video, audio, text & speech recognition

Source: http://www-nlpir.nist.gov/projects/tv2013/tv2013.html

Action classification

Goal: to assign a label of action or event to a video

Making sandwich: present
Feeding animal: not present
…

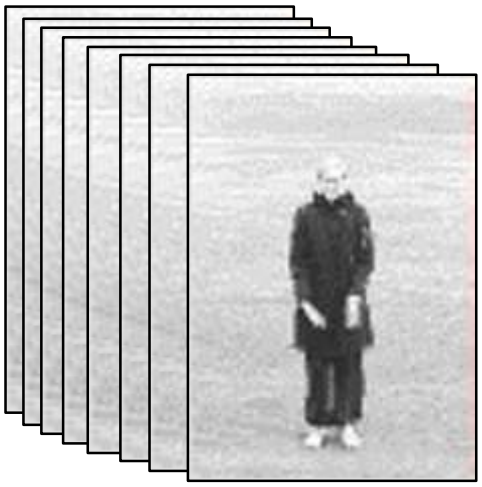
How to recognize actions?

phoning

smoking

hand shaking

drinking



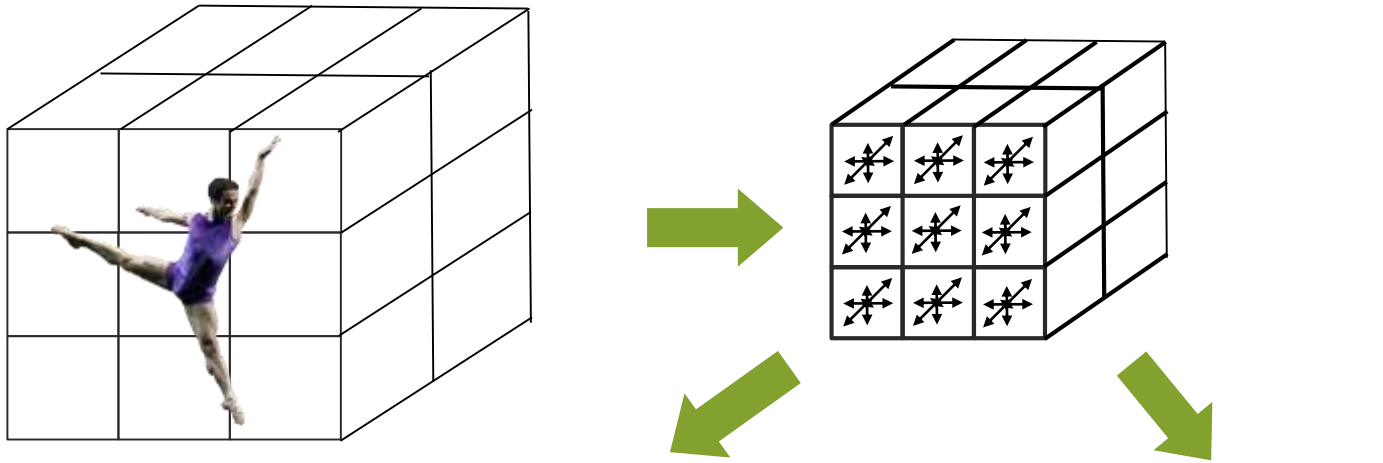
Some actions can be recognized from static image alone.

Some other actions cannot be identified by a single frame.

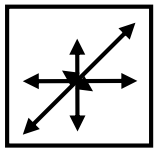
We need to see the motion of the human, so we need to consider space-time volume.

Простейший подход

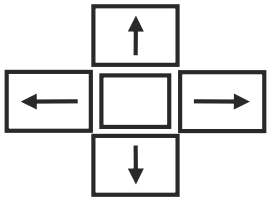
Разобьём решеткой видео на области, в каждой посчитаем гистограмму признаков, и подадим на вход классификатору



Histogram of Gradients (HOG)



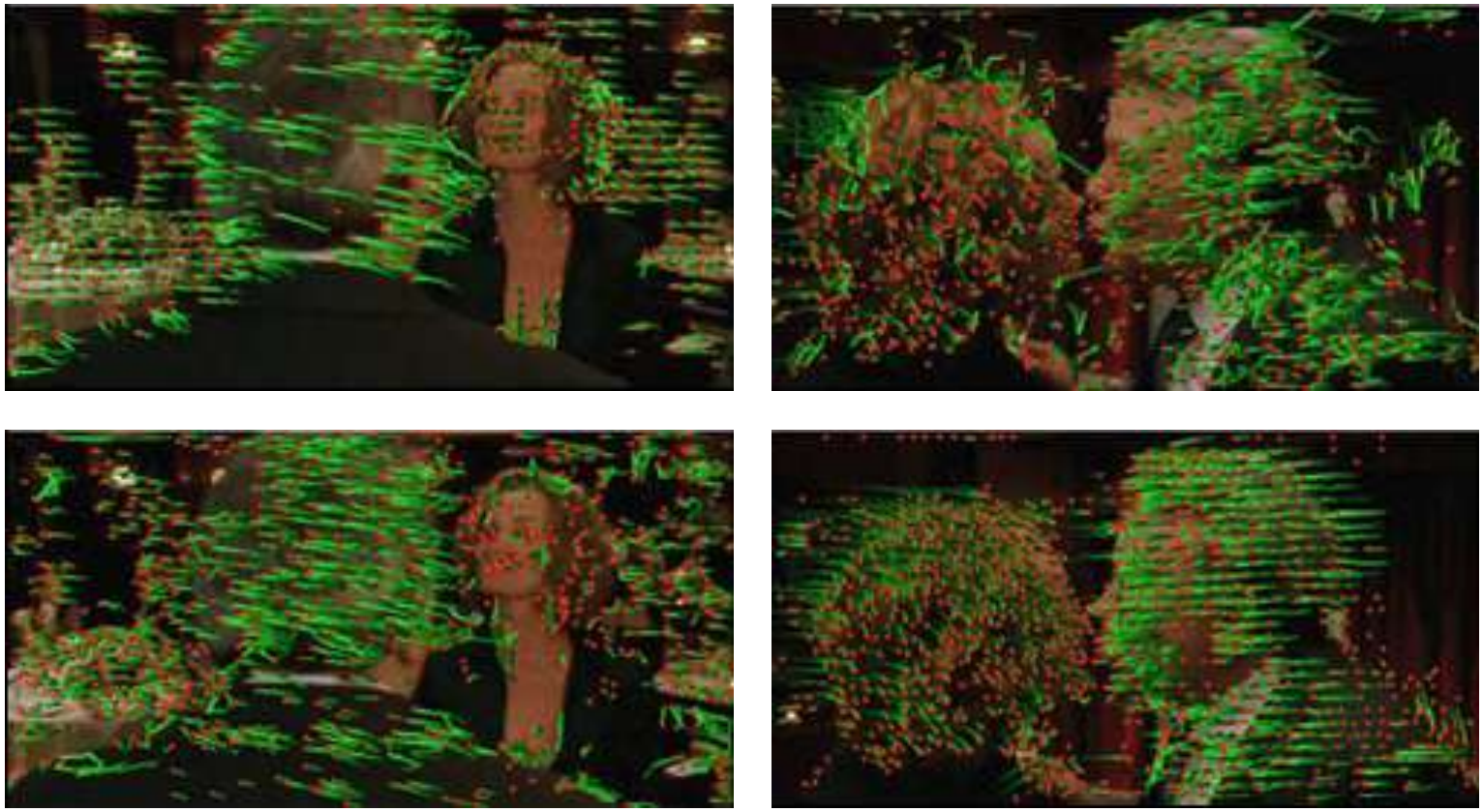
Histogram of optical flow (HOF)



3x3x2x4bins
HOG descriptor

3x3x2x5bins
HOF

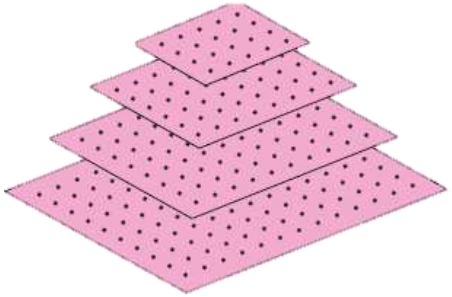
Dense trajectories



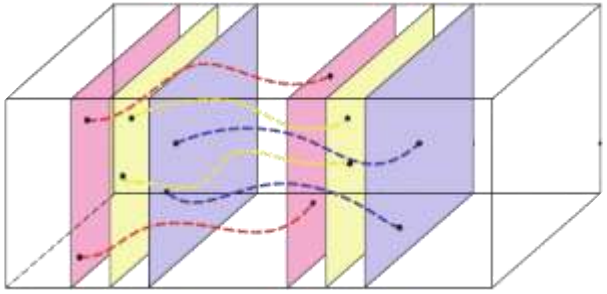
Вместо регулярной сетки можем отследить множество точек

Source: H. Wang, A. Kläser, C. Schmid, C.L.Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition IJCV 2013

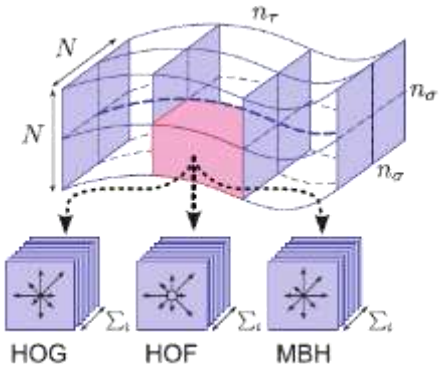
Dense trajectories



Dense sampling in each spatial scale

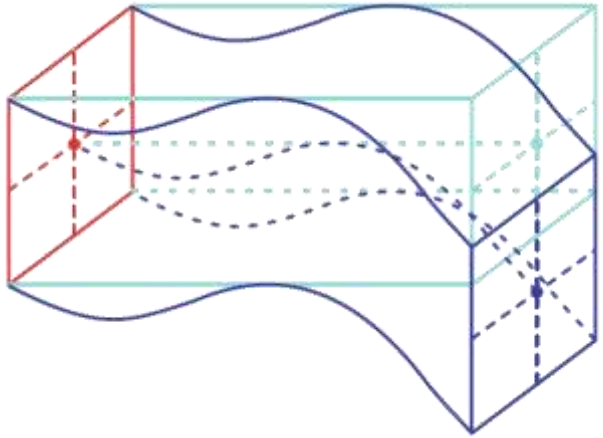


Tracking in each spatial scale separately



Trajectory description

Removal of points in uniform areas



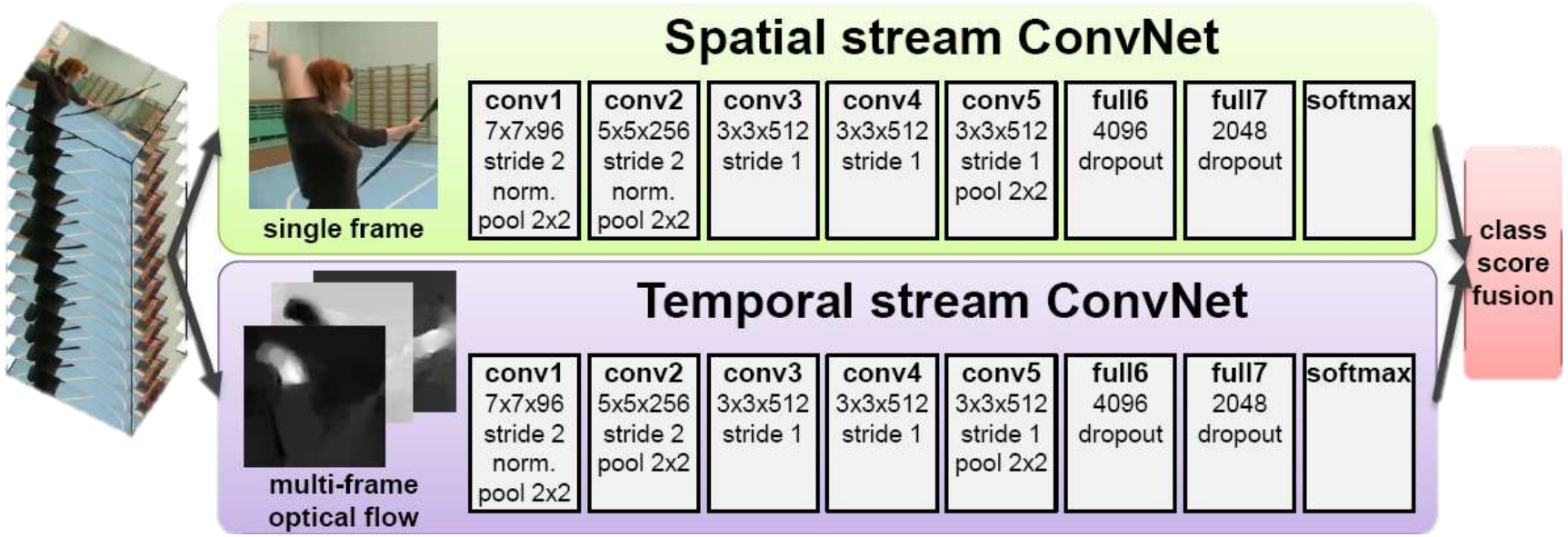
Comparison of dense trajectories with dense cuboids

Source: H. Wang, A. Kläser, C. Schmid, C.L.Liu. Dense Trajectories and Motion Boundary Descriptors for Action Recognition IJCV 2013

Neural features

- We can apply convolutional neural network directly to the space-time volume
- Due to the large size of input the training of such network is prohibitively difficult
- Thus various approaches for simplifying the model are exploited

Two-stream CNN model

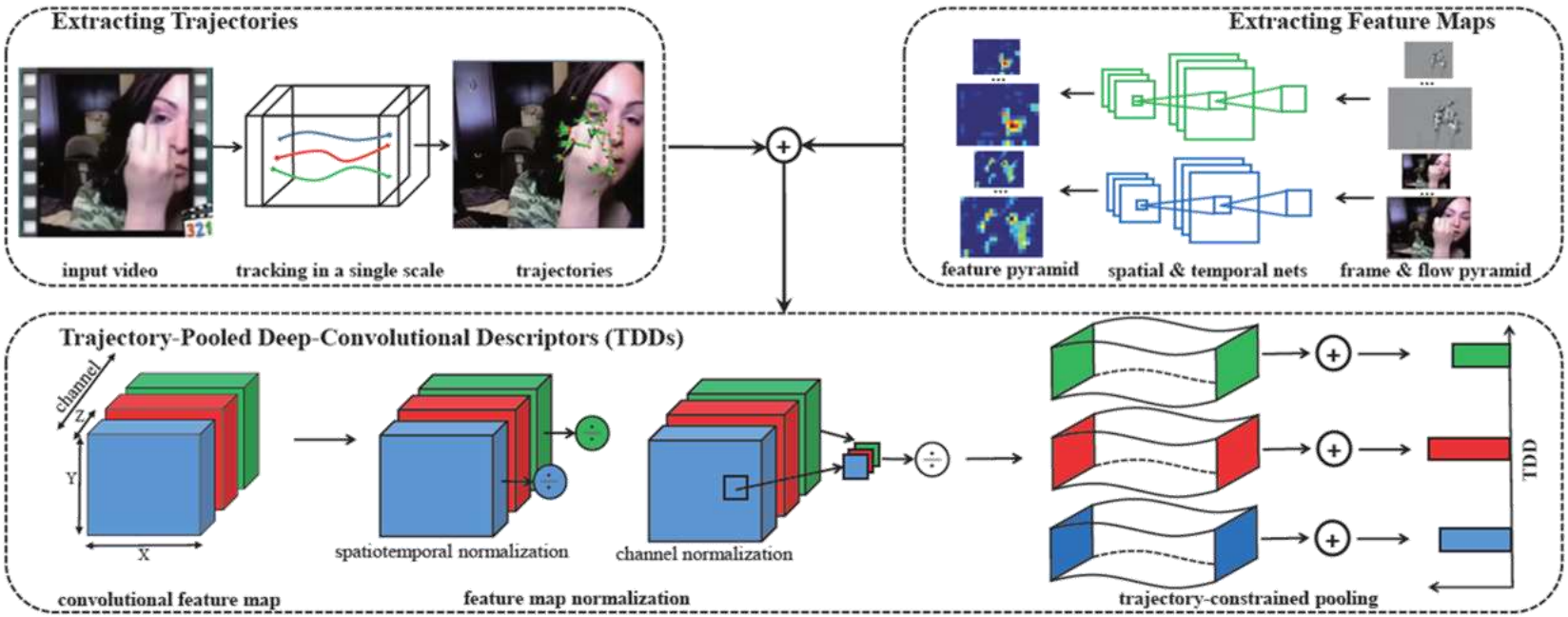


Temporal stream:

- Vertical and horizontal optical flow components
- 5-10 frames (input tensor $w \cdot h \cdot 2L$)
- Mean flow can be subtracted from each frame

Source: Karen Simonyan, Andrew Zisserman Two-Stream Convolutional Networks for Action Recognition in Videos, NIPS 2014

Dense trajectories + Deep features



- Instead of HOG/HOF we will take convolutional features
- We will pool features along dense trajectories

Source: Limin Wang, Yu Qiao, Xiaoou Tang. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors, CVPR 2015

Что сейчас?

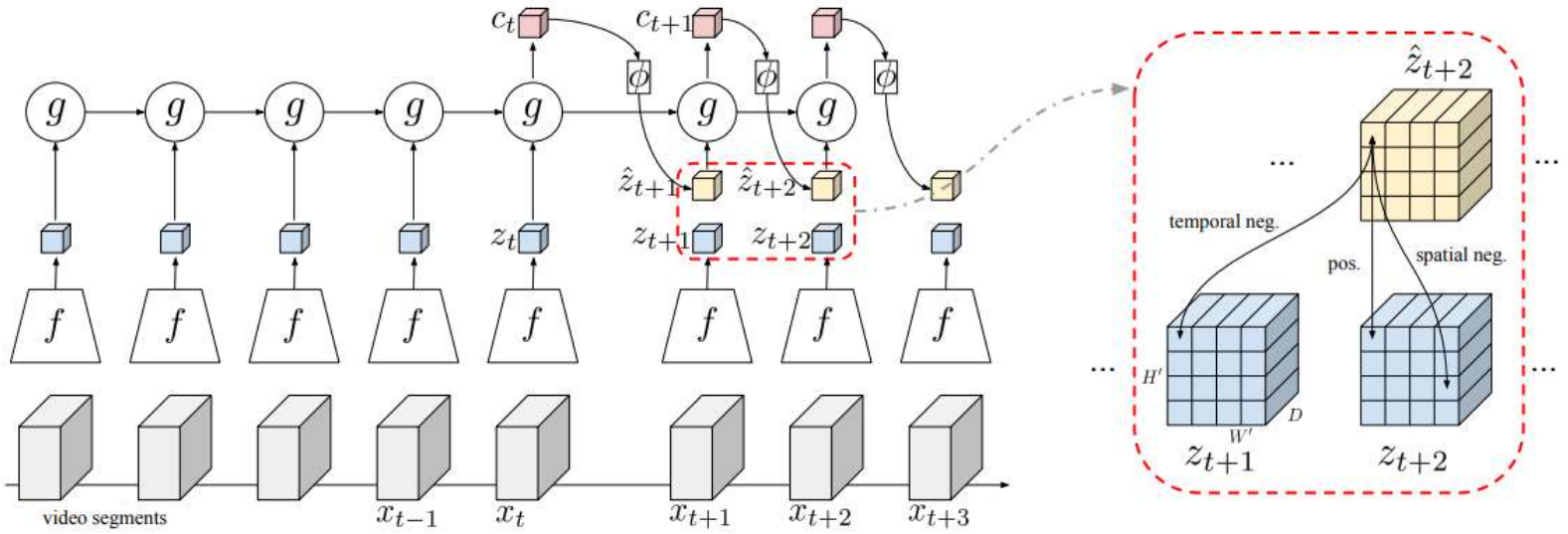


Figure 2: A diagram of **Dense Predictive Coding** method. The left part is the pipeline of the DPC, which is explained in Sec. 3.1. The right part (in the dashed rectangle) is an illustration of the Pred-GT pair construction for contrastive loss, which is explained in Sec. 3.2.

- Развитие self-supervised методов
- Учимся на задаче без разметки – предсказания признаков следующих видеофрагментов по предыдущим
- Нужно строить «похожие» на реальность

https://arxiv.org/pdf/1909.04656.pdf

Пример применения для поиска

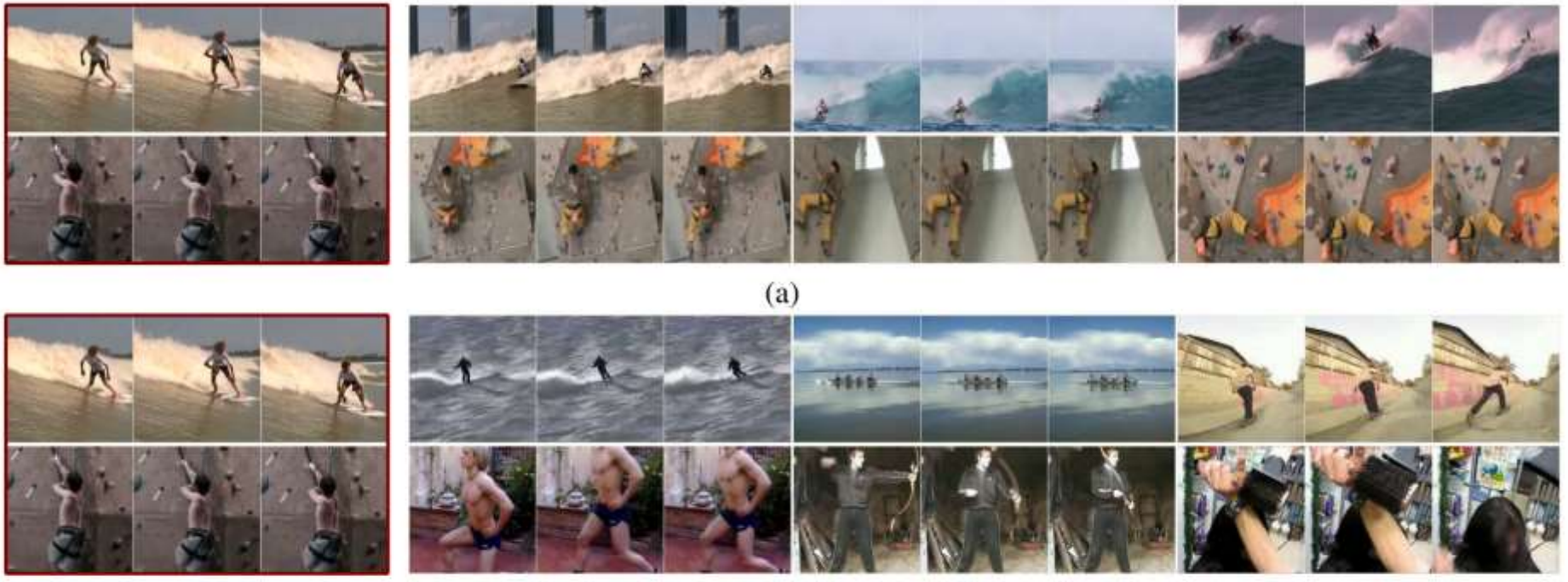


Figure 1: Nearest Neighbour (NN) video clip retrieval on UCF101. Each row contains four video clips, a query clip and the top three retrievals using clip embeddings. To get the embedding, each video is passed to a 3D-ResNet18, average pooled to a single vector, and cosine similarity is used for retrieval. (a) Embeddings obtained by Dense Predictive Coding (DPC); (b) Embeddings obtained by using the inflated ImageNet pretrained weights. The DPC captures the semantics of the human action, rather than the scene appearance or layout as captured by the ImageNet trained embeddings. In the DPC retrievals the actual appearances of frames can vary dramatically, *e.g.* in the change in camera viewpoint for the climbing case.

Обученную сеть для извлечения признаков можно использовать для обучения классификаторов с учителем или как признак для поиска похожих видео

https://arxiv.org/pdf/1909.04656.pdf

Резюме

- Основные задачи обработки видео – оценка оптического потока, сопровождение объектов, распознавание событий
- Сейчас они все более-менее успешно решаются нейросетевыми моделями
- «Движение» обычно нужно учитывать отдельно, и желательно детально (вроде dense trajectories) и учитывать оптический поток
- Сейчас активно развивают self-supervised методы

Школа Анализа
Данных Яндекса

Факультет
компьютерных наук
НИУ ВШЭ

Лаборатория компьютерной графики
и мультимедиа ВМК МГУ

Курс «Компьютерное зрение»

Лекция №11
«Плотная 3-х мерная реконструкция»

Антон Конушин

29 ноября 2019 года

Бинокулярное стерео

Параллакс

- Параллакс - видимое смещение объекта в зависимости от точки обзора
- Чем объект ближе, тем смещение больше

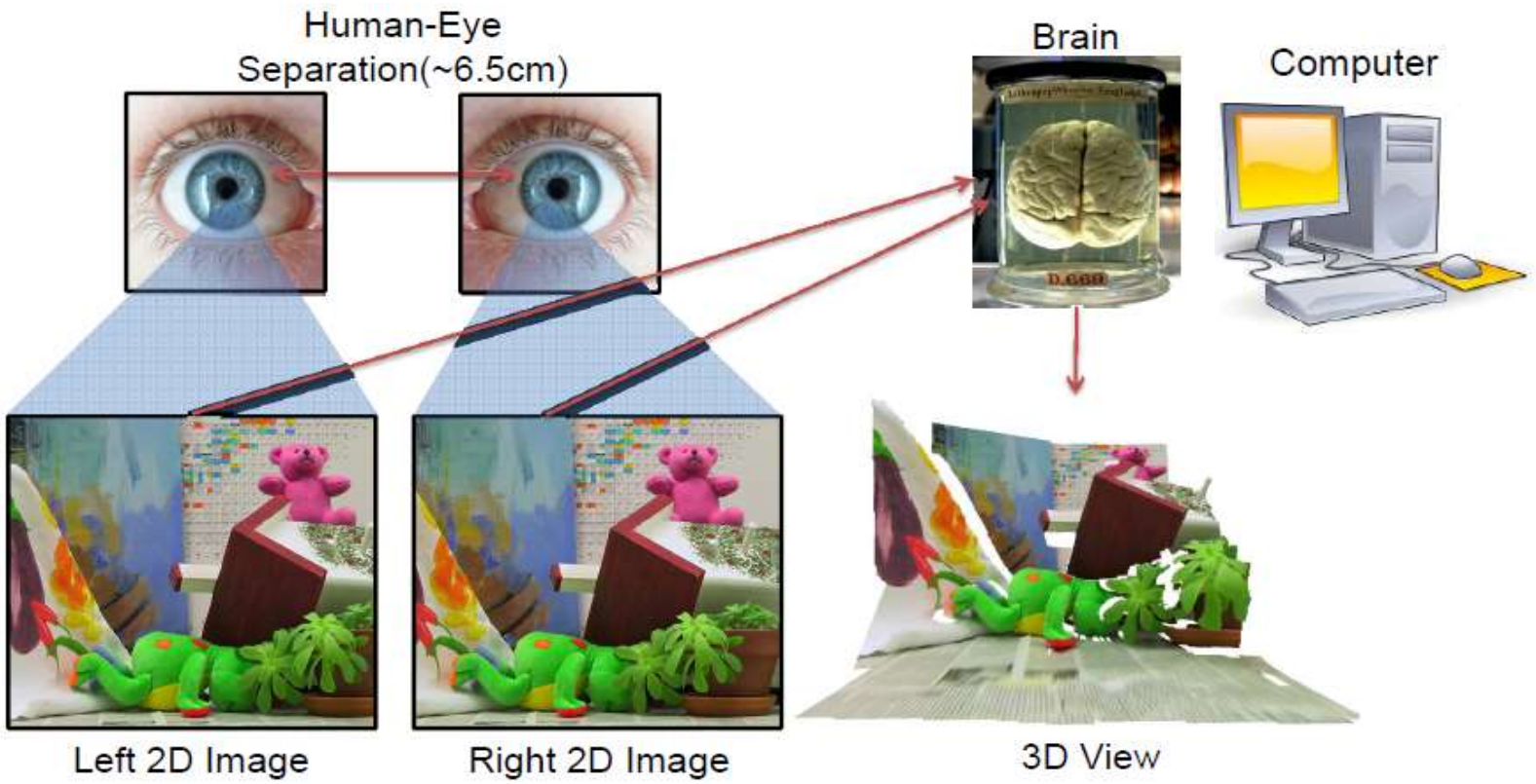
Стереопсис

«**Стереопсис** (англ. stereopsis) - сенсорный процесс, возникающий при бинокулярном зрении как психофизическая реакция на сетчаточную горизонтальную диспаратность. В результате С. субъект переживает специфическое ощущение глубины. <…> ». (Психологическая энциклопедия)

Левое изображение

Правое изображение

Плотное стерео



- Задача плотного стерео: восстановить в 3D все видимые на изображениях точки
  - Если 2 изображения – то бинокулярное стерео
  - Если >2 – многовидовое стерео
- Калибровку камер будем считать известной

Слайд: M. Bleyer

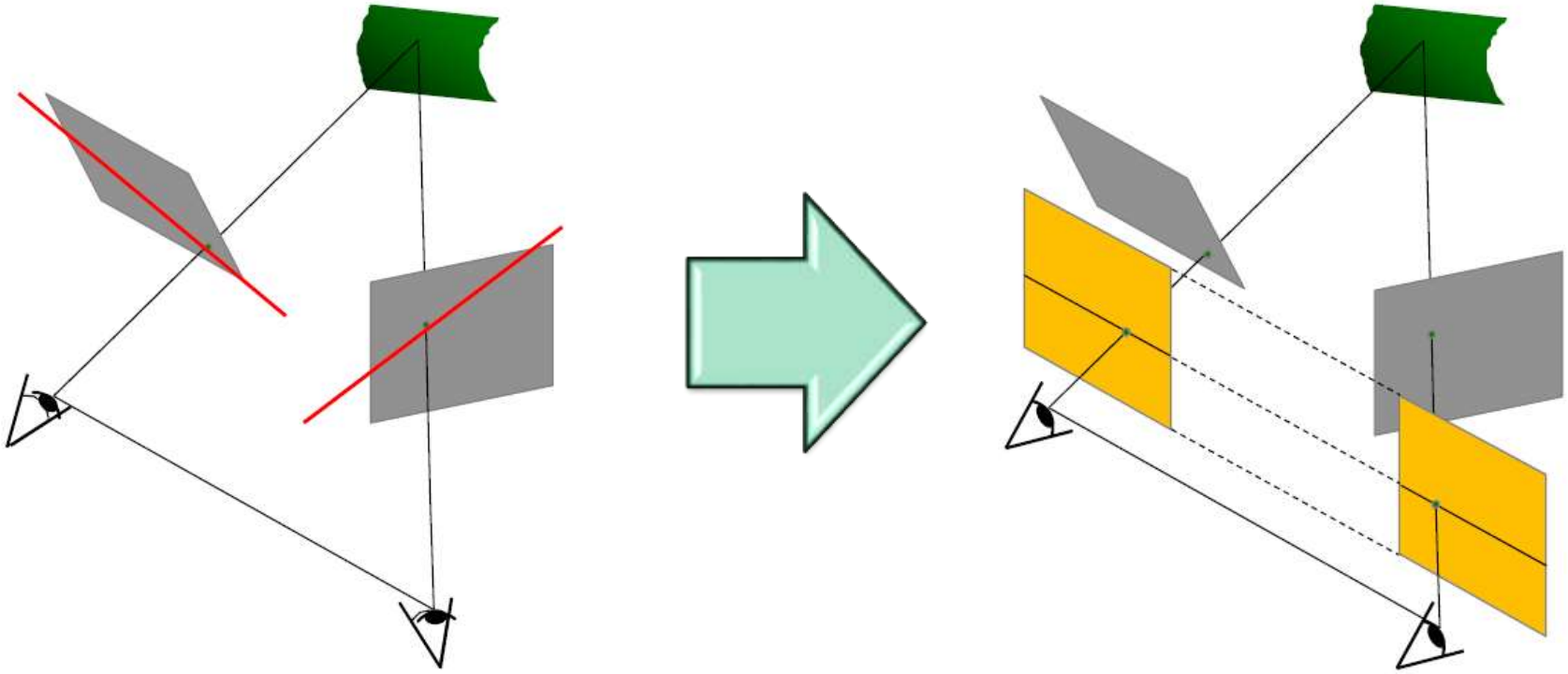
Общая схема

Ректификация изображений

Вычисление соответствующих точек

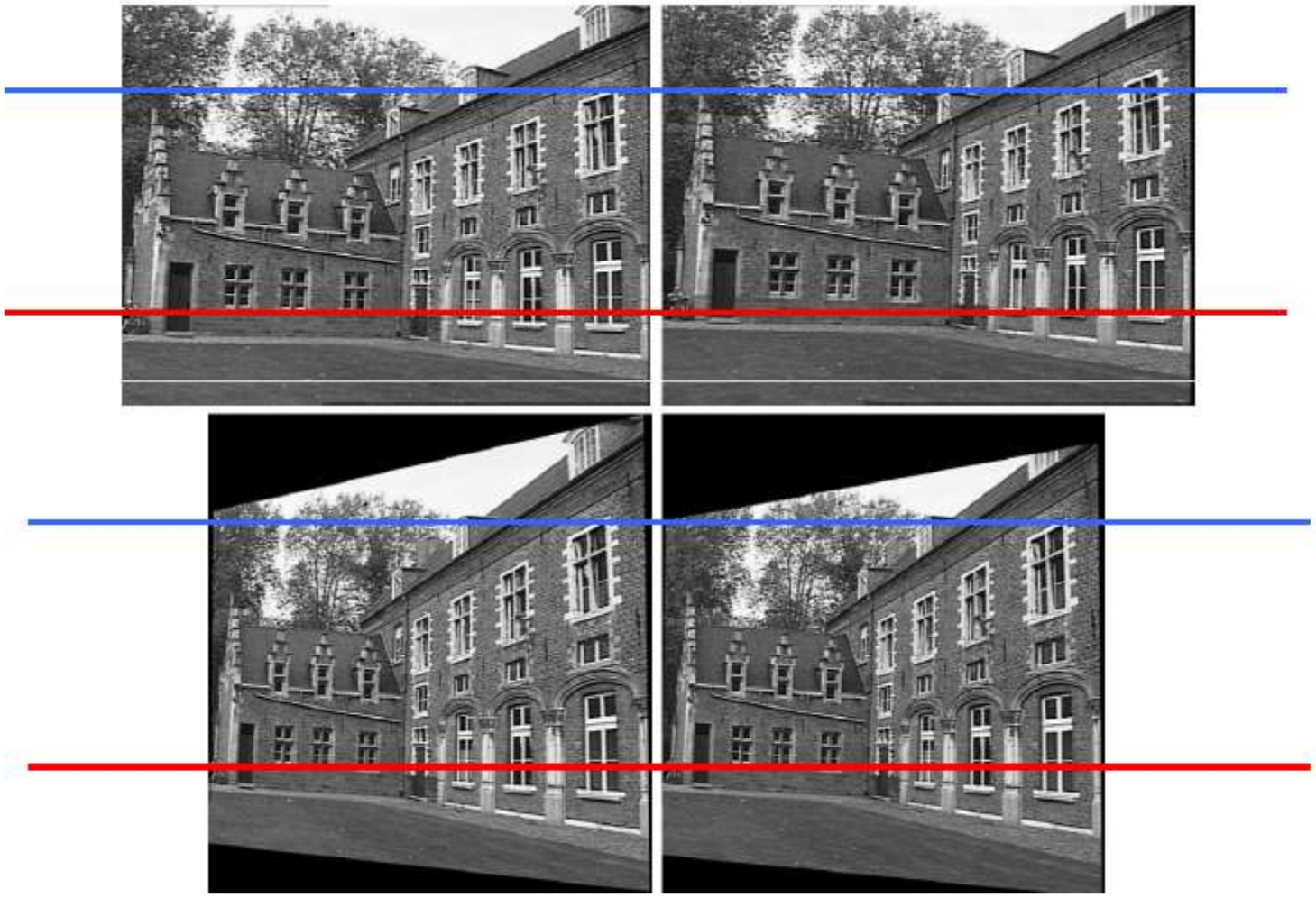
Восстановление 3D путем триангуляции

Ректификация



- «Ректификация» – преобразование стереопары в изображения, в которых соответствующие эпиполярные линии лежат на одной и той же горизонтальной строке
- Первый способ: проецирование на общую плоскость с помощью гомографии

Ректификация через гомографию

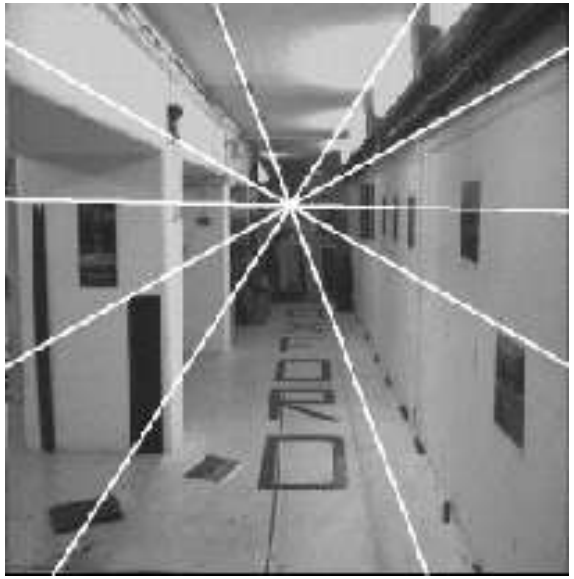


Недостатки:

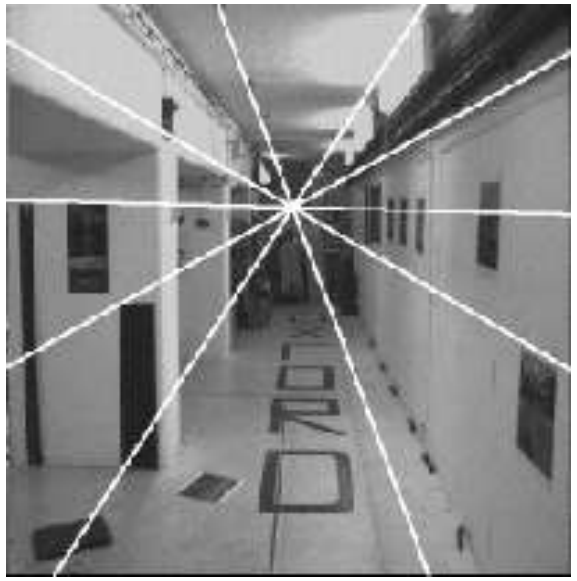
- Не применим, когда камера движется вперёд или назад
- Сильные искажения в некоторых случаях

Радиальная развёртка

- Polar rectification
  - Индексируем эпиполярные линии углом поворота относительно эпиполи
  - Копируем соответствующие пары эпиполярных линии последовательно в соответствующие горизонтали ректифицированных изображений



- Работает в тех случаях, когда ошибается метод проецирования на общую плоскость

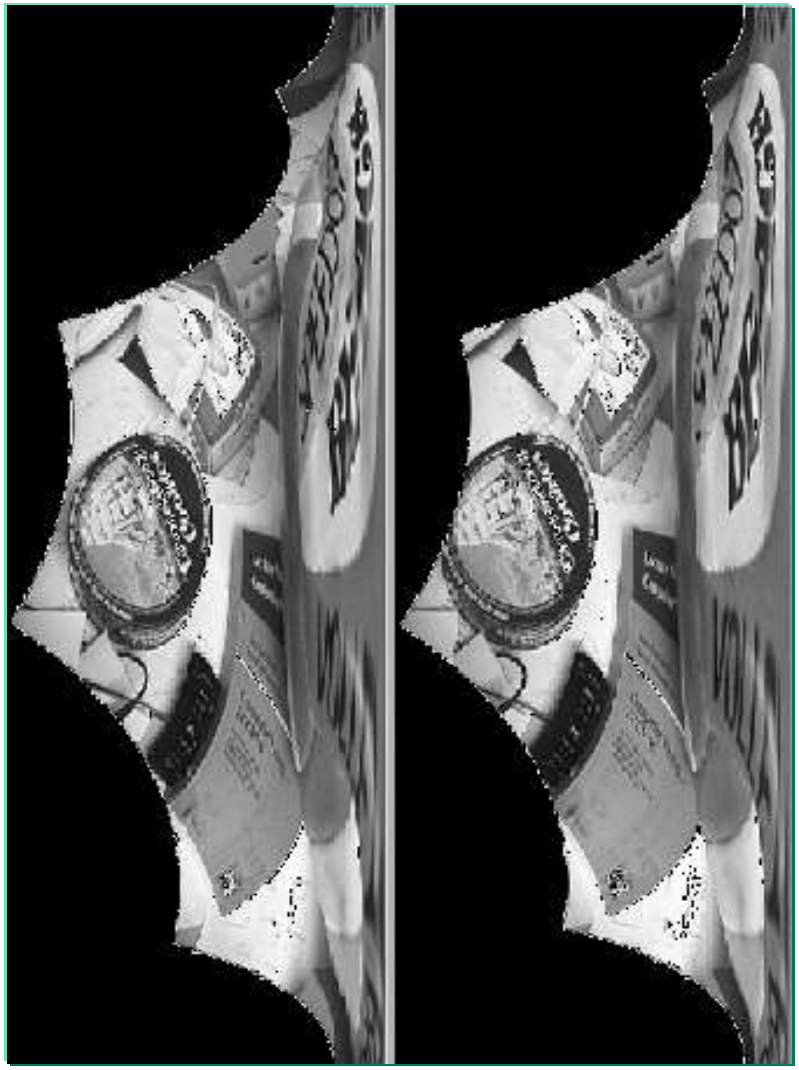


Пример

Проецирование

Полярная ректификация

Пример



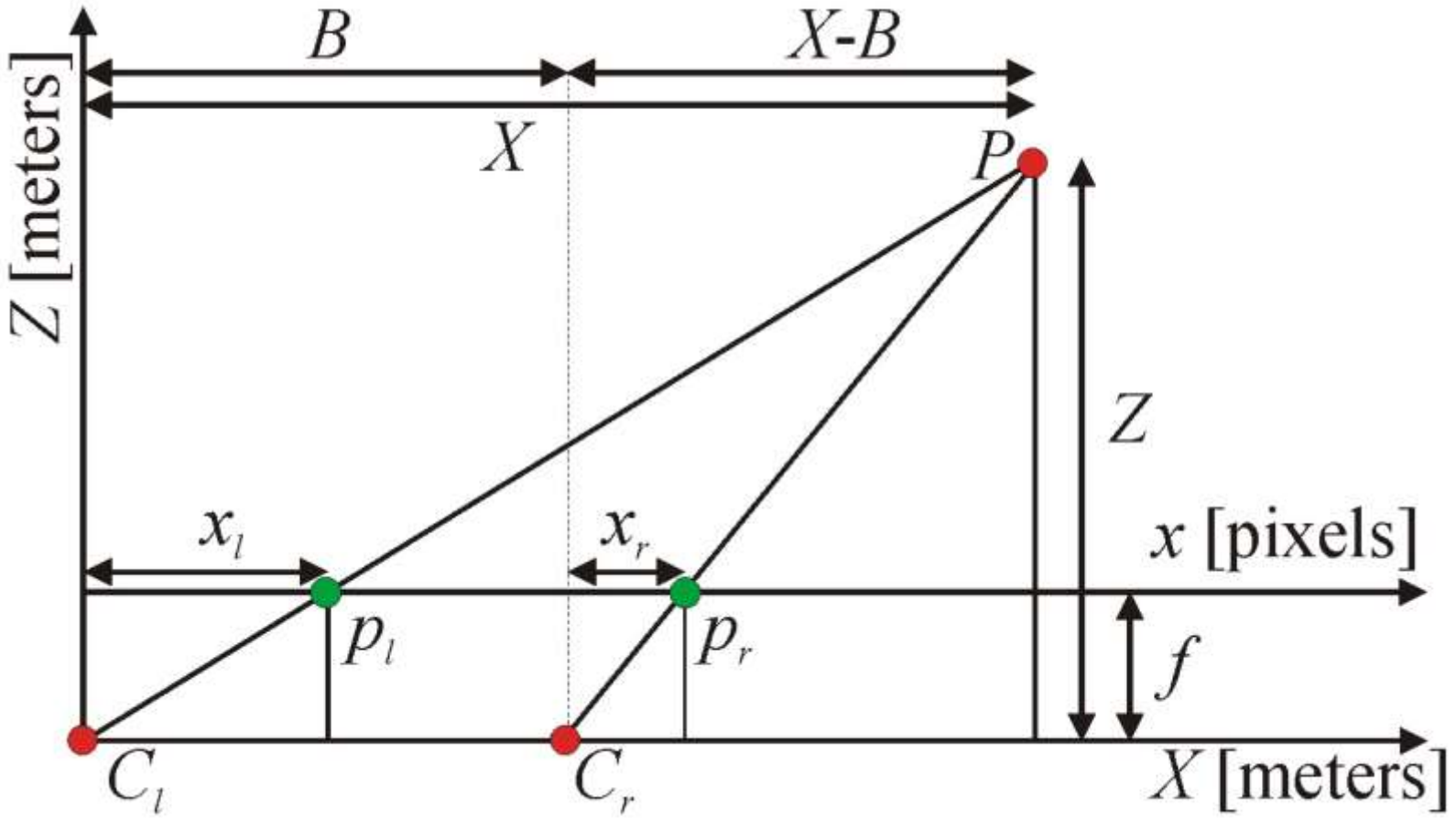
Общая схема

Ректификация изображений

Вычисление соответствующих точек

Восстановление 3D путем триангуляции

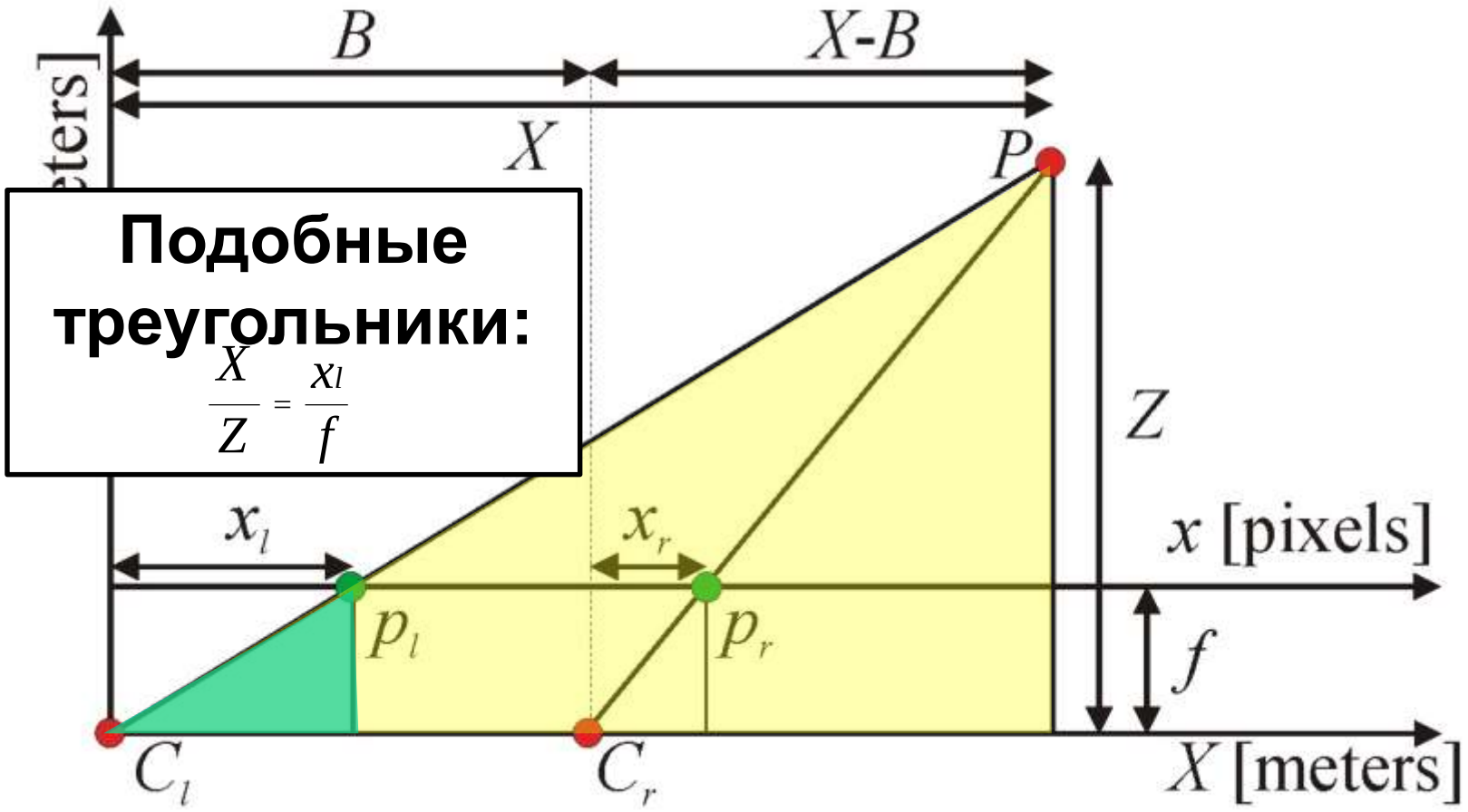
Триангуляция



x_l, x_r - смещения относительно принципиальной точки

Слайд: M. Bleyer

Триангуляция

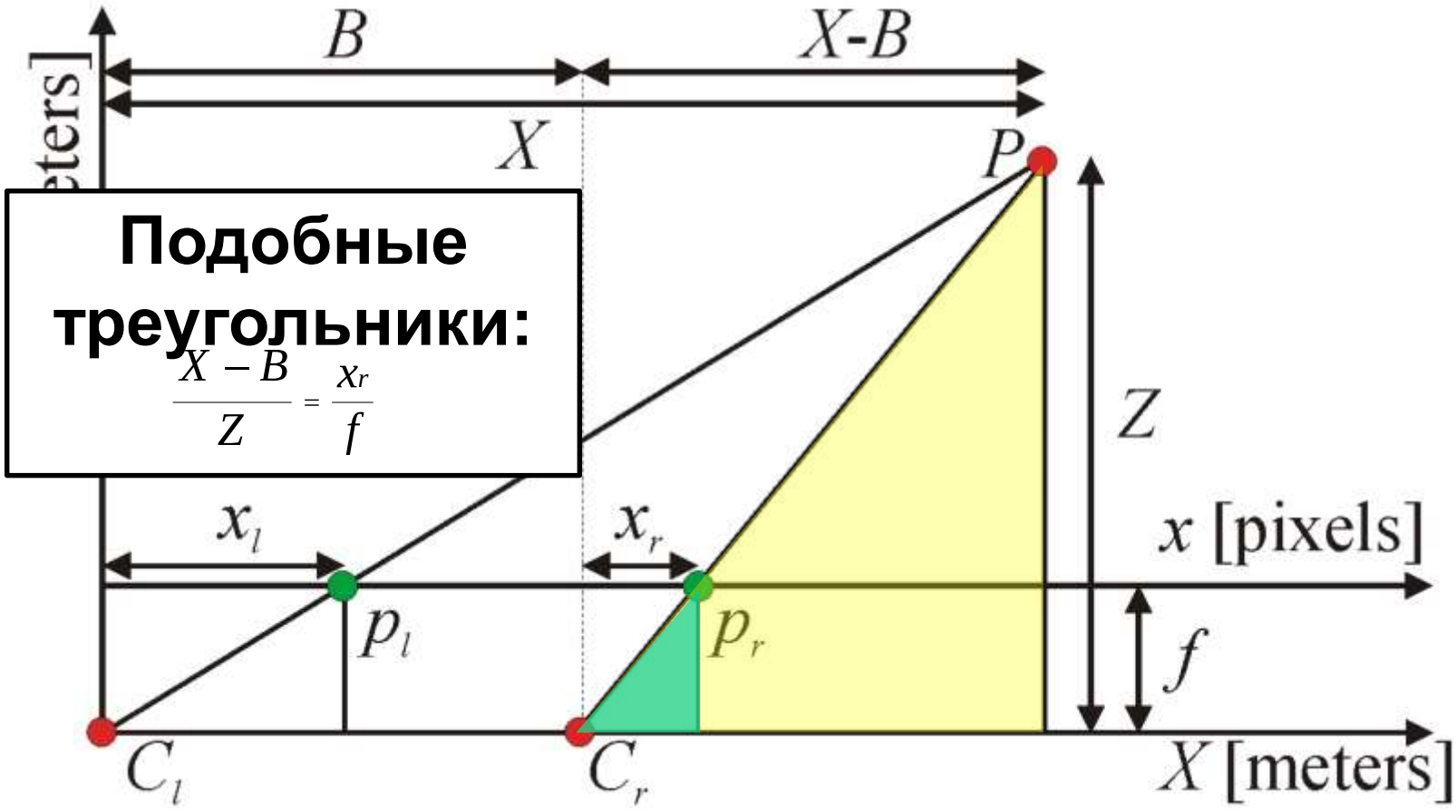


Подобные треугольники:

$$\frac{X}{Z} = \frac{x_l}{f}$$

Слайд: M. Bleyer

Триангуляция



Подобные треугольники:

$$\frac{X - B}{Z} = \frac{x_r}{f}$$

Слайд: M. Bleyer

Триангуляция

Из подобия треугольников:

$$\frac{X}{Z} = \frac{x_l}{f} \qquad \frac{X - B}{Z} = \frac{x_r}{f}$$

Исключаем *X* и выражаем *Z*:

$$Z = \frac{B \cdot f}{x_l - x_r} = \frac{B \cdot f}{d}$$, где $d = x_l - x_r$ - ***диспаритет***

Итоговые координаты 3D-точки:

$$X = \frac{Z \cdot x_l}{f} \qquad Y = \frac{Z \cdot y_l}{f} \qquad Z = \frac{B \cdot f}{d}$$

x_l, x_r - смещения относительно принципиальной точки

Общая схема

Ректификация изображений

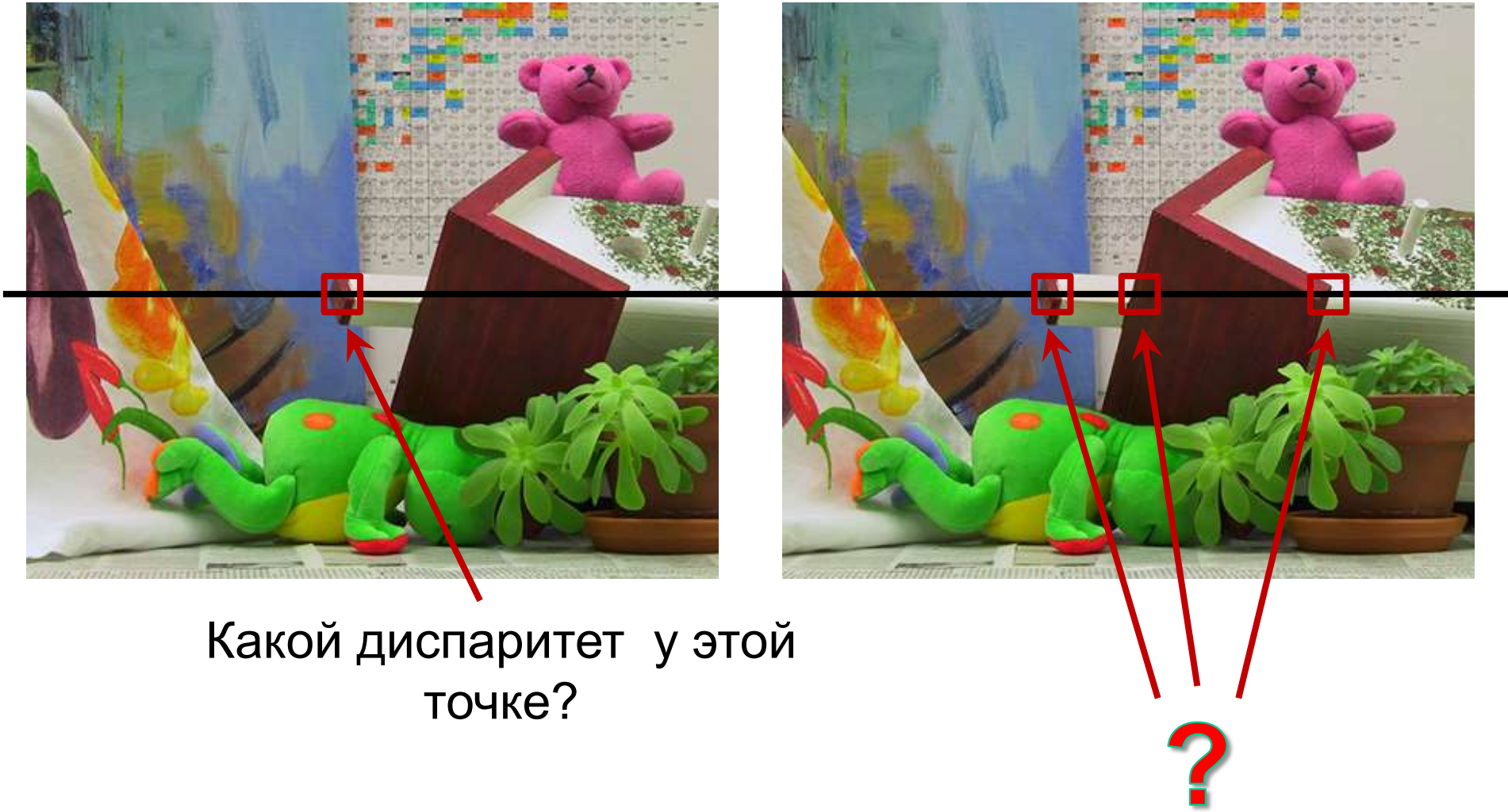
Вычисление карты диспаритета

Восстановление 3D путем триангуляции

Ректификация и триангуляция – технические задачи, основной интерес это стереосопоставление

Стереосопоставление через вычисление диспаритета

Эпиполярное ограничение:
«Соответствующие точки лежат на одной строке»



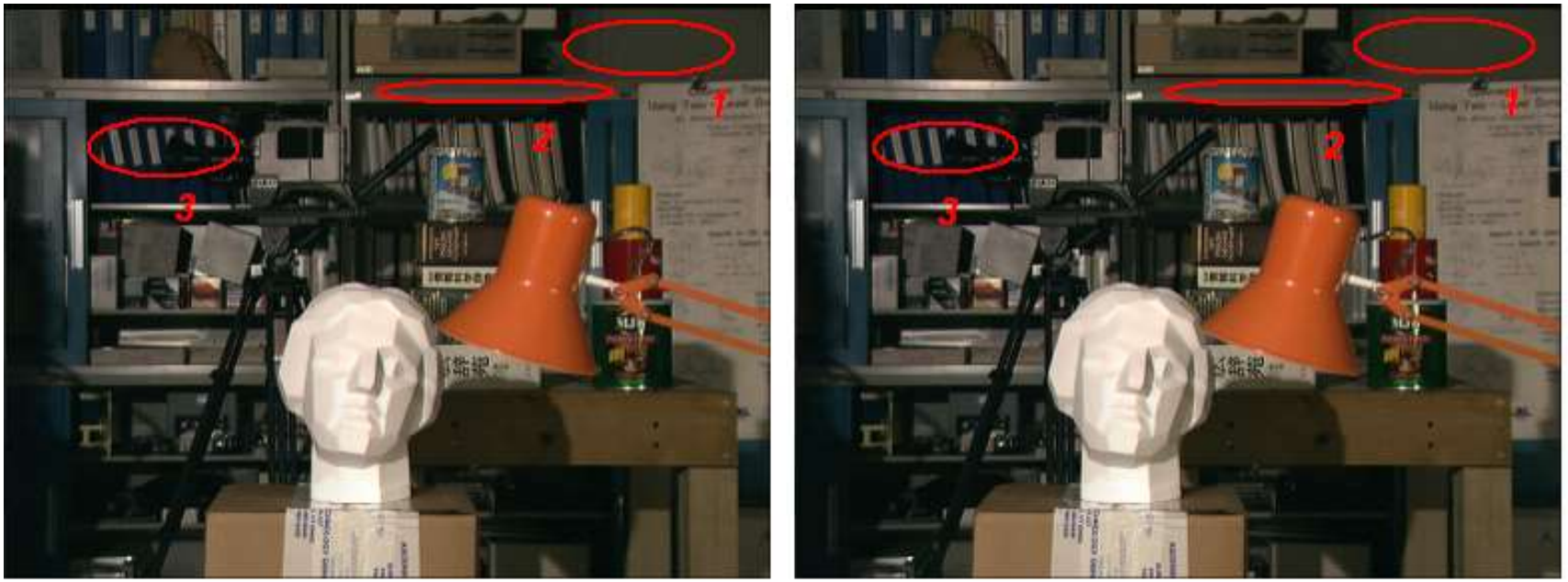
Основные предположения

Соответствие цветов:
«Цвета соответствующих точек совпадают»

Может нарушаться (шум камеры, изменение освещенности, погрешности сэмплирования и т.д.)

Основные трудности

«Плохо» текстурированные области



1 – Однородная область
2 – Текстура неизменна в горизонтальном направлении
3 – Повторяющаяся текстура

Основные трудности

Перекрытия

Некоторые пиксели одного изображения могут быть не видны (перекрыты) на другом.

«Невидимые» области называют областями перекрытия

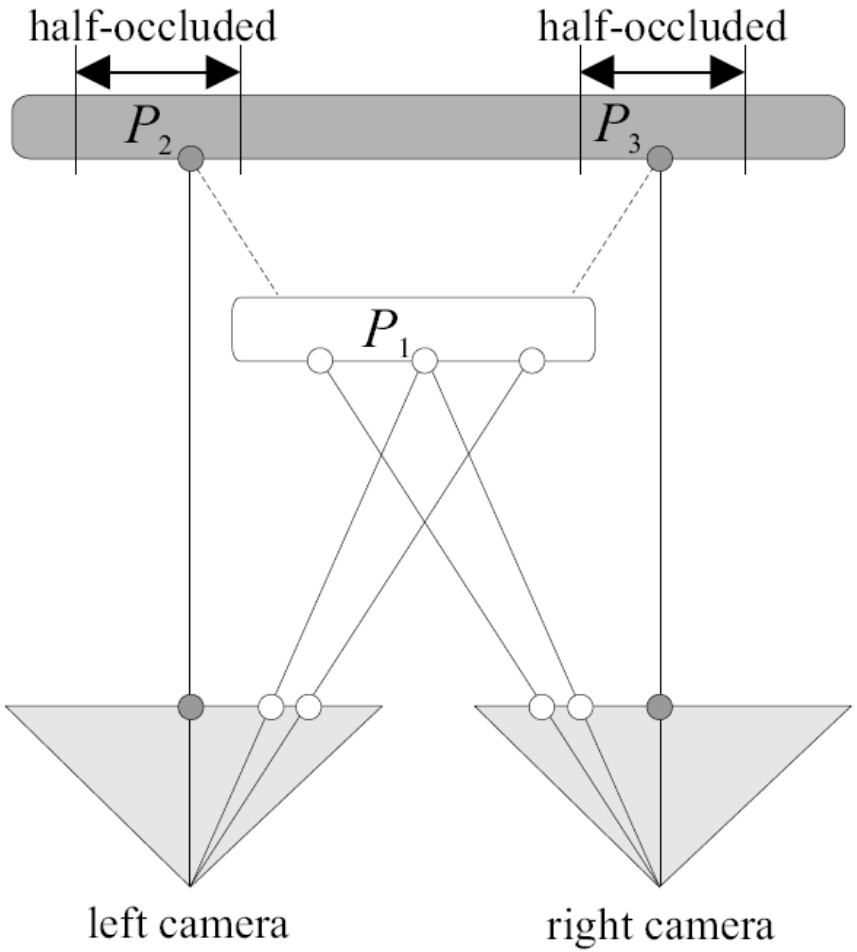
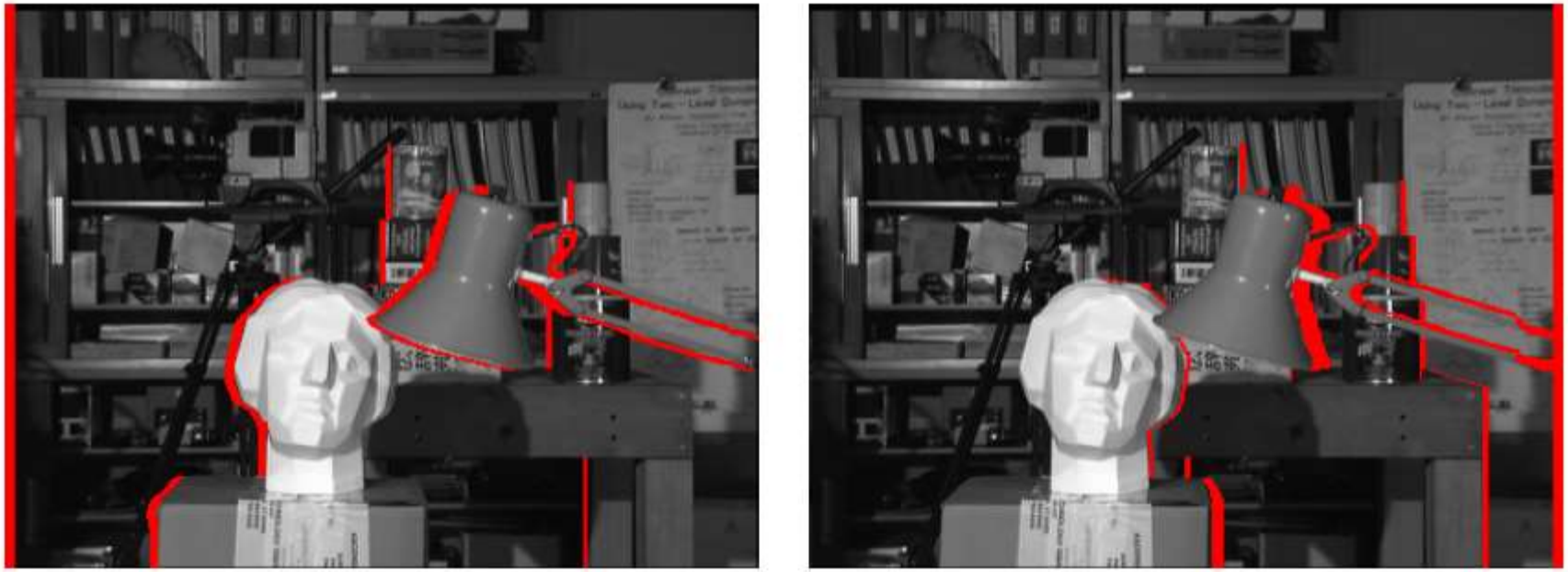


Рисунок: M. Bleyer

Основные трудности

Перекрытия



Пример областей перекрытия (выделены красным)

Рисунок: M. Bleyer

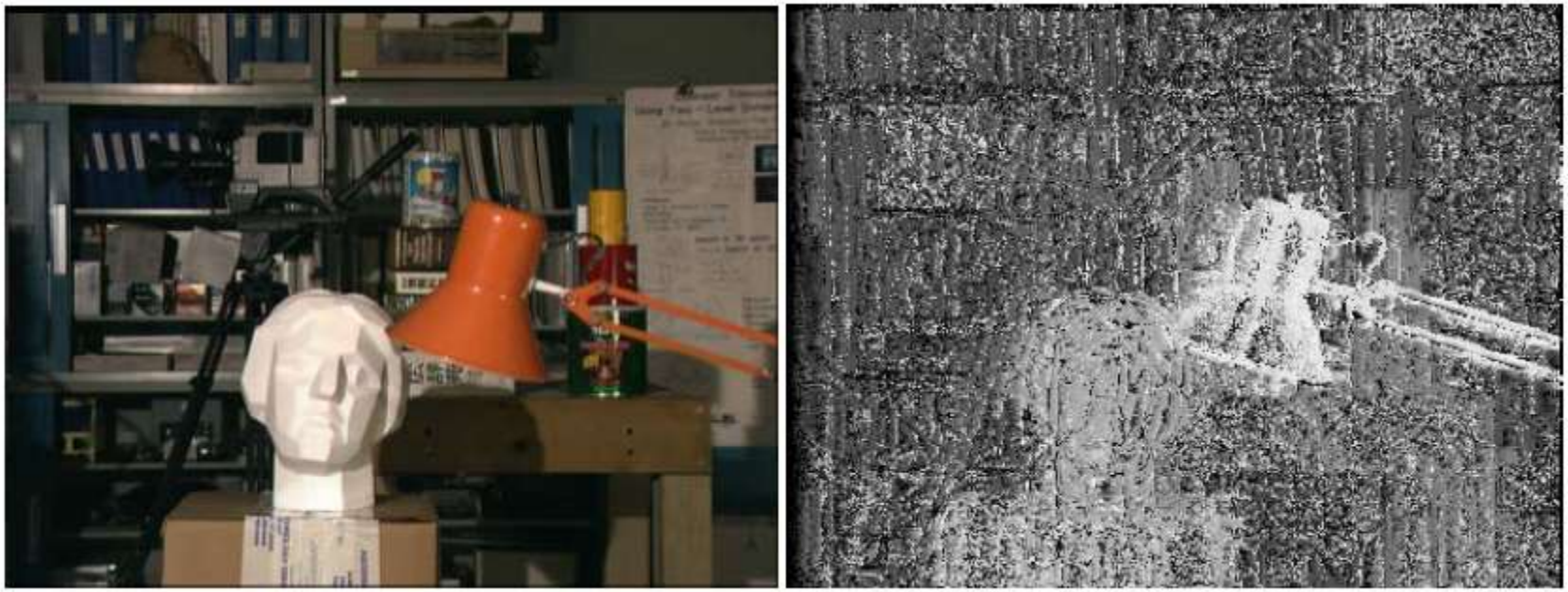
Локальные методы бинокулярного стерео

- Локальные, т.к. диспаритет в каждой точке зависит только от ее локальной окрестности

- Используют стратегию WTA (Winner-Take-All), т.е. в каждой точке «побеждает» соответствие с наименьшей «стоимостью»

Наивный алгоритм

- Стоимость соответствия – разность интенсивностей пикселей
- Результат – слишком много шума



Пример работы наивного алгоритма

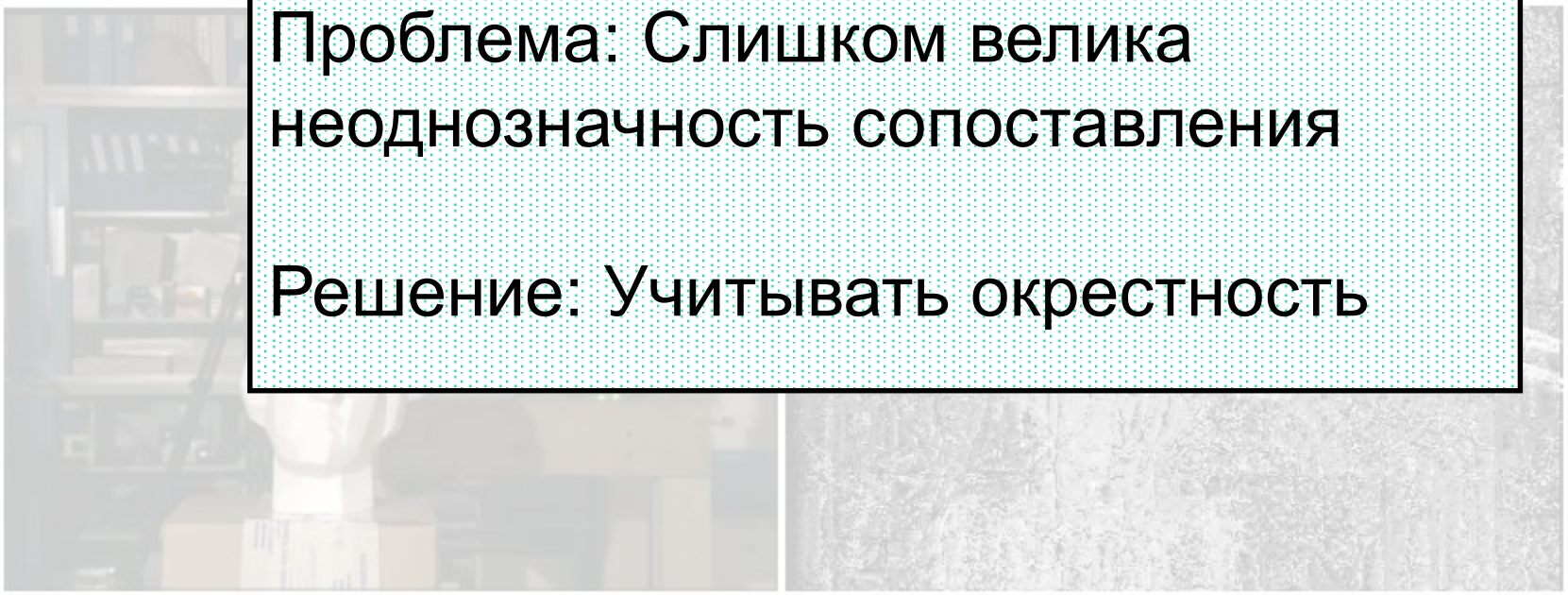
Пример: M. Bleyer

Наивный алгоритм

- Стоимость соответствия – разность интенсивностей пикселей
- Резул

Проблема: Слишком велика неоднозначность сопоставления

Решение: Учитывать окрестность



Пример работы наивного алгоритма

Пример: M. Bleyer

Учет окрестности

- Стоимость соответствия – SAD, SSD, NCC по окну вокруг пикселя

$$d_p = \arg\min_{0 \le d \le d\max} \sum_{q \in Wp} c(q, q-d)$$

где:

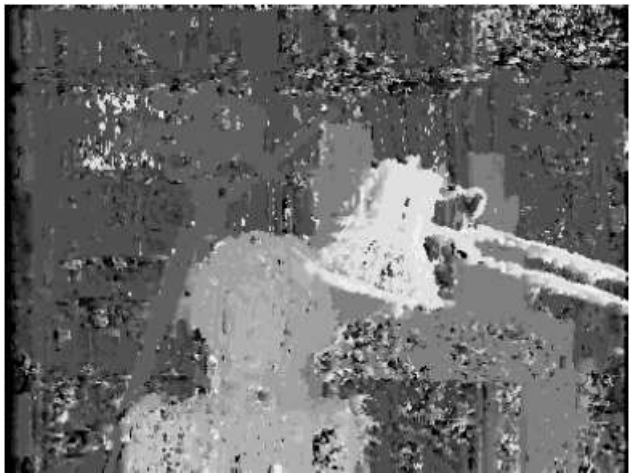
d_p – искомый диспаритет в пикселе p,
$c(p,q)$ – функция стоимости,
W_p – окно вокруг пикселя p,
d_{max} – максимально возможный диспаритет

- Возможна очень эффективная реализация
  - Метод «скользящего окна»

Проблема выбора размера окна

- Маленькие окна – недостаточно текстуры
- Большие окна – эффект «раздувания» объектов переднего плана (foreground fattening)

Ground truth

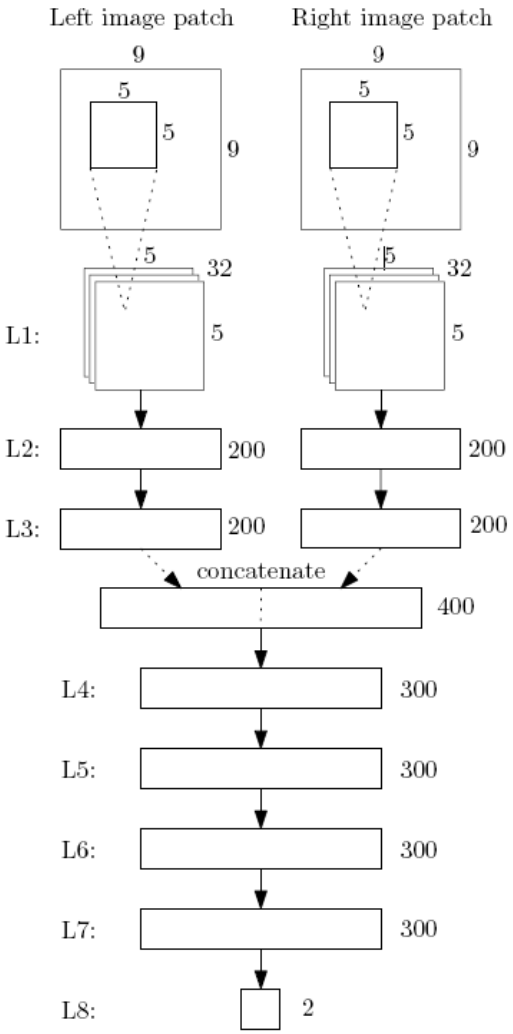


Окно 3x3

Окно 21x21

Примеры: M. Bleyer

Обучение с помощью CNN



- Собираем обучающую выборку $< \mathcal{P}^L_{9\times 9}(\mathbf{p}), \mathcal{P}^R_{9\times 9}(\mathbf{q}) >$
- 1 положительный и 1 отрицательный пример на 1 пиксель с известным диспаритетом
- Отрицательные примеры:

$$\mathbf{q} = (x - d + o_{\text{neg}}, y) \quad \{-N_{\text{hi}}, \ldots, -N_{\text{lo}}, N_{\text{lo}}, \ldots, N_{\text{hi}}\}$$

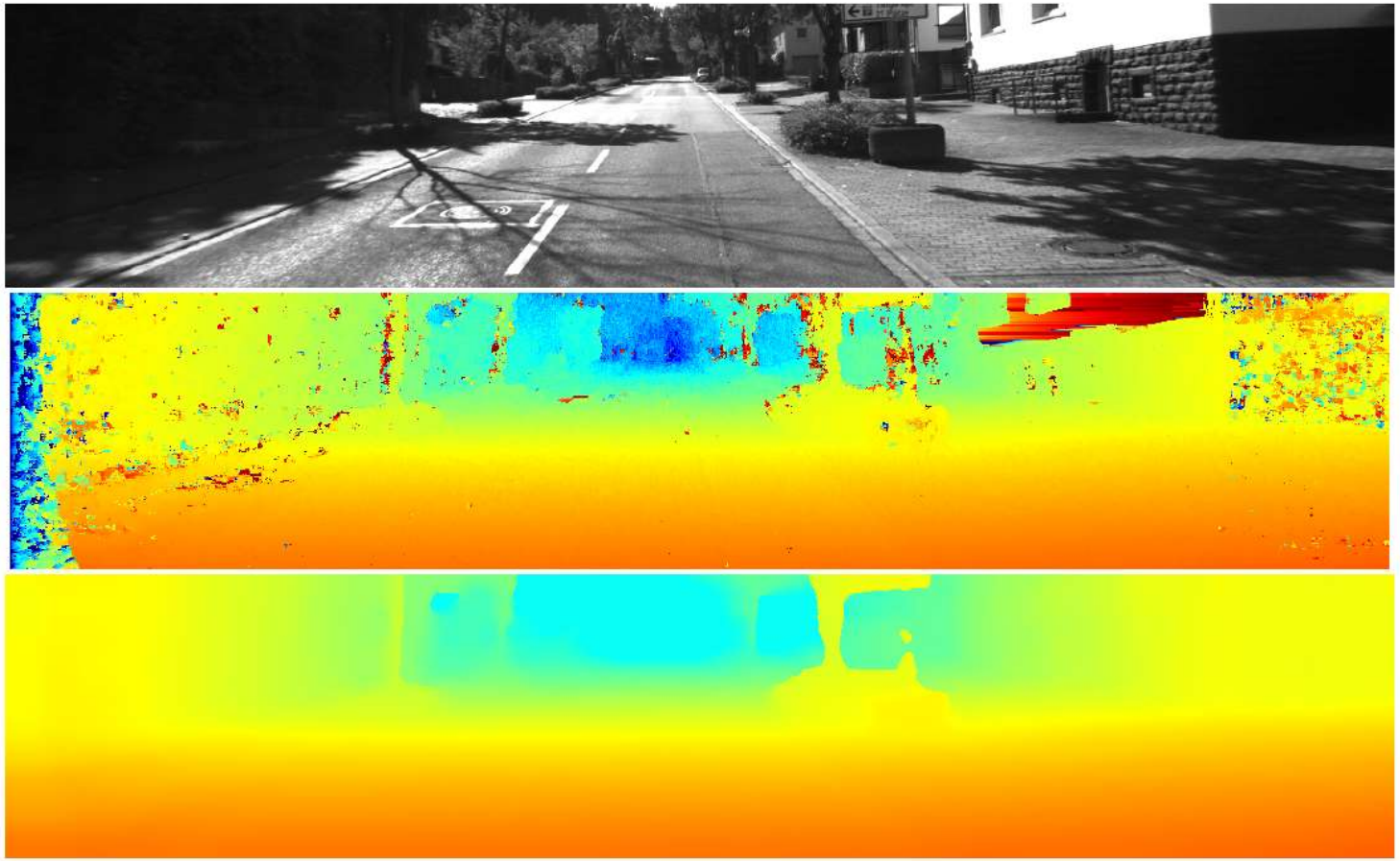
- Положительные примеры:

$$\mathbf{q} = (x - d + o_{\text{pos}}, y) \quad \{-P_{\text{hi}}, \ldots, P_{\text{hi}}\}$$

-

J. Zbontar and Y. LeCun: Computing the Stereo Matching Cost with a Convolutional Neural Network. CVPR 2015

Пример результата



- Ошибка 14.7% для winner-take-all стратегии
- Ошибка 2.61% при использовании метрики внутри semi-global стратегии (идею рассмотрим дальше)

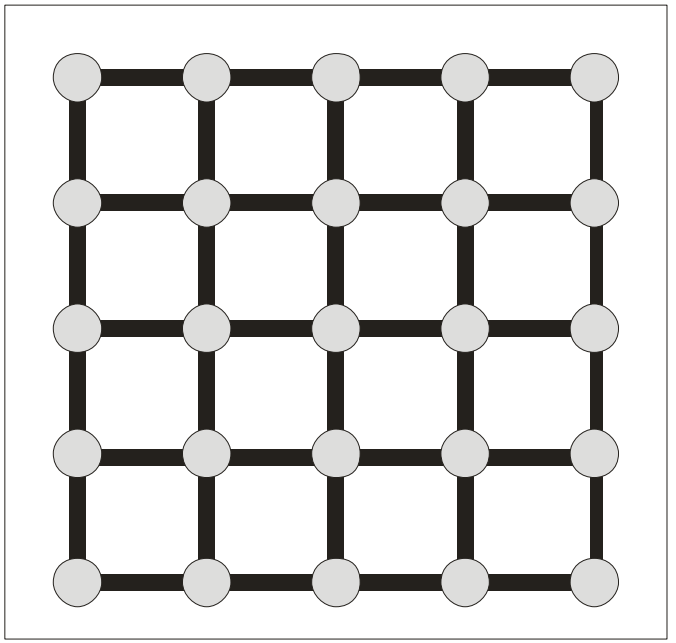
Глобальные методы

- Глобальные, т.к. диспаритет в каждой точке вычисляется при помощи некоторой глобальной процедуры оптимизации, т.е. зависит не только от локальной окрестности

- Как правило, формулируются в терминах разметки графа и минимизации энергии

Глобальные методы

Необходимо найти разметку *D*, минимизирующую функцию энергии *E(D)*.



Метки – значения диспаритета.

Граф – решетка, узлы – пиксели.

$$E(D) = E_{data}(D) + E_{smooth}(D)$$

Соответствие цветов

Гладкость

Рисунок: M. Bleyer

Проблема сохранения границ

Предположим, необходимо восстановить диспаритет в области границы между объектами, как показано на рисунке

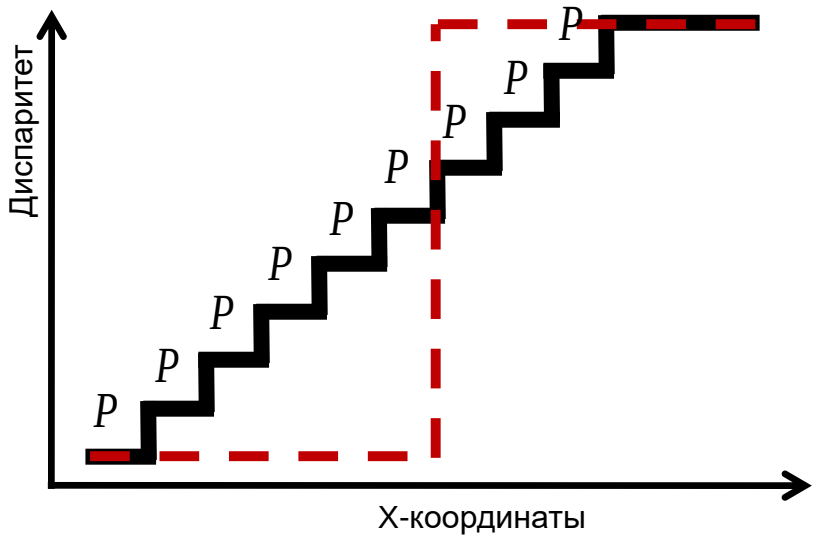
Рисунок: M. Bleyer

Проблема сохранения границ

Случай линейной модели:

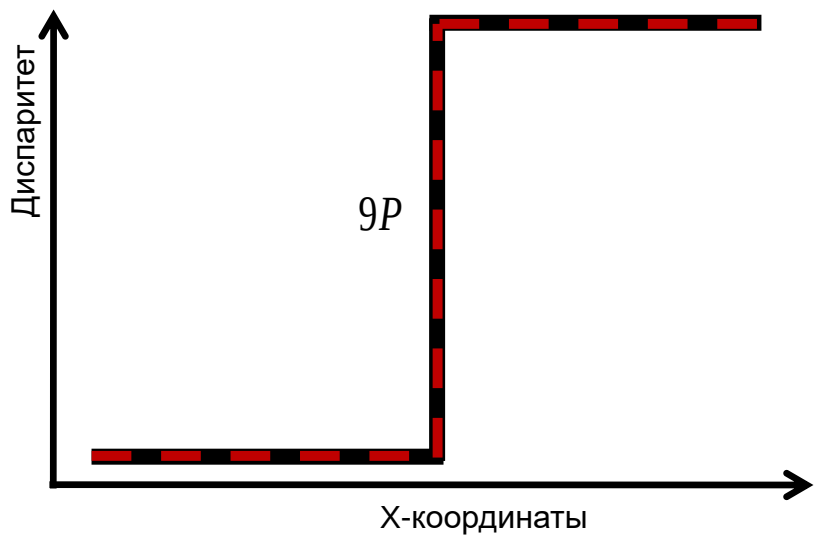
$$s(d_p, d_q) = |d_p - d_q| \cdot P$$

Вклад в энергию:
9P



(неверное решение)

Вклад в энергию:
9P



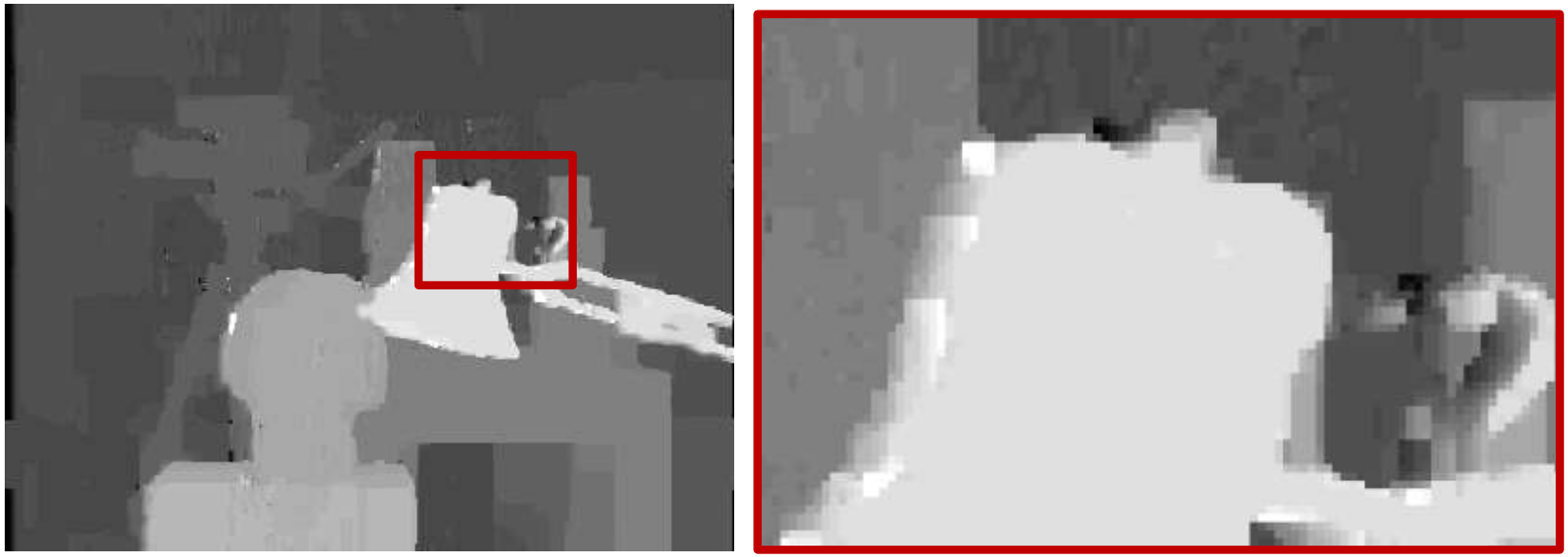
(верное решение)

Линейная модель не поощряет резких разрывов диспаритета. Она чересчур сглаживает решение.

Фадунин М. Bleyer

Проблема сохранения границ

Случай линейной модели: $s(d_p, d_q) = |d_p - d_q| \cdot P$



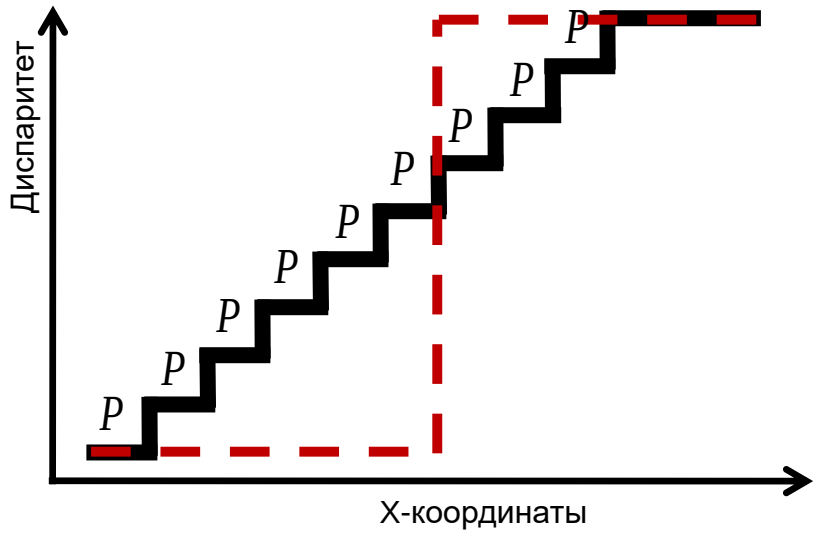
Пример результатов для линейной модели. Справа показан увеличенный фрагмент

Примеры: M. Bleyer

Проблема сохранения границ

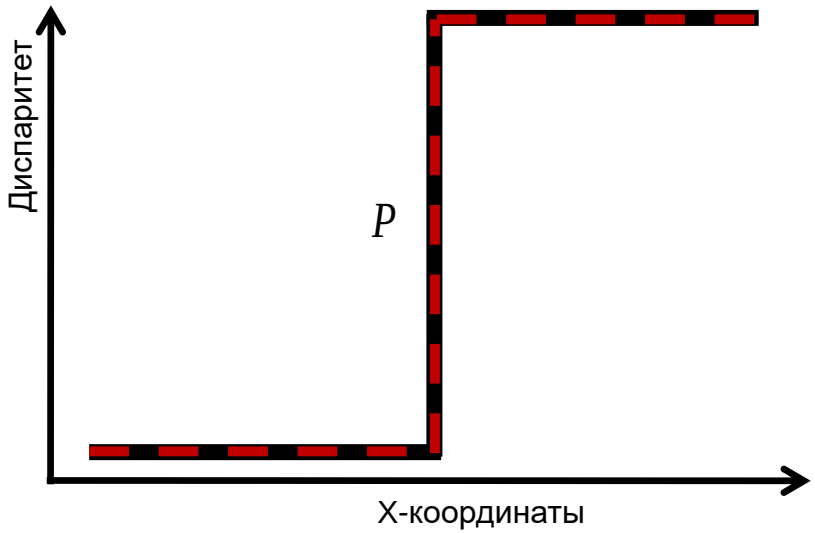
Случай модели Поттса: $$s(d_p, d_q) = \begin{cases} 0, d_p = d_q \\ P, d_p \neq d_q \end{cases}$$

Вклад в энергию: 9P



(неверное решение)

Вклад в энергию: P



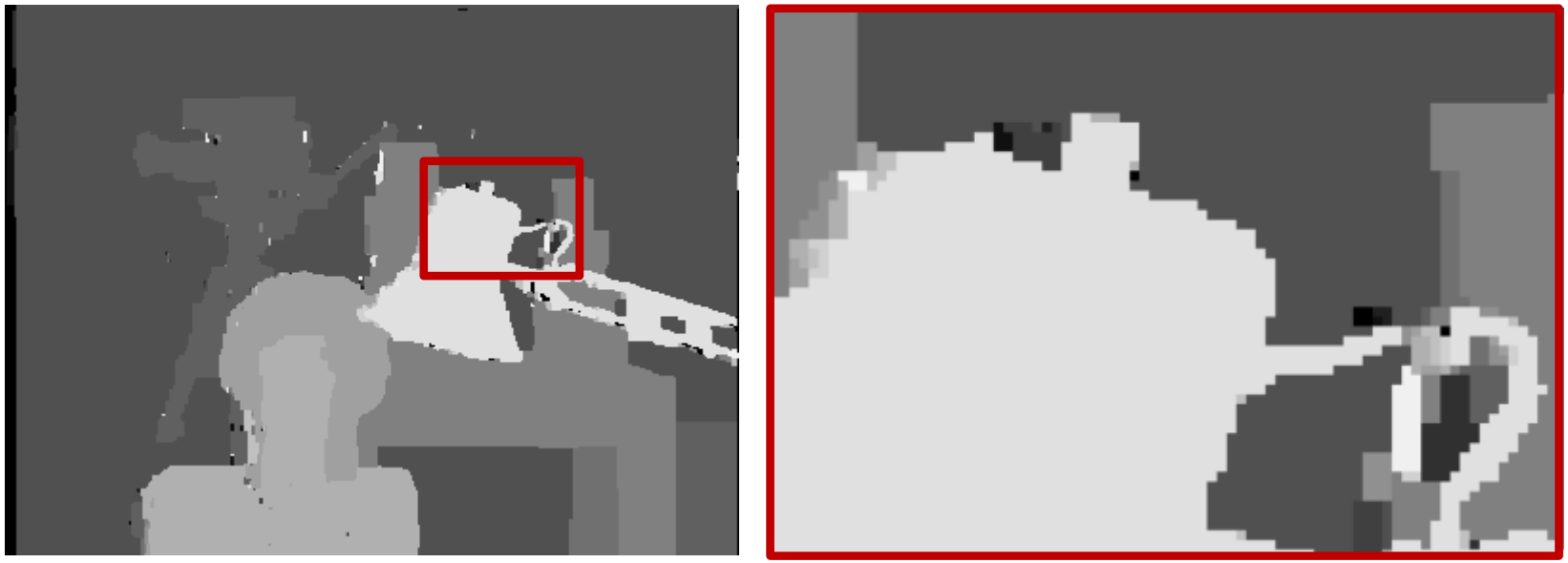
(верное решение)

- Модель Поттса не препятствует резким разрывам диспаритета.
- Такие модели называют сохраняющими разрывы (discontinuity preserving) – как и в оптическом потоке

Рисунки: M. Bleyer

Проблема сохранения границ

Случай модели Поттса: $$s(d_p, d_q) = \begin{cases} 0, d_p = d_q \\ P, d_p \neq d_q \end{cases}$$



Пример результатов для модели Поттса. Справа показан увеличенный фрагмент

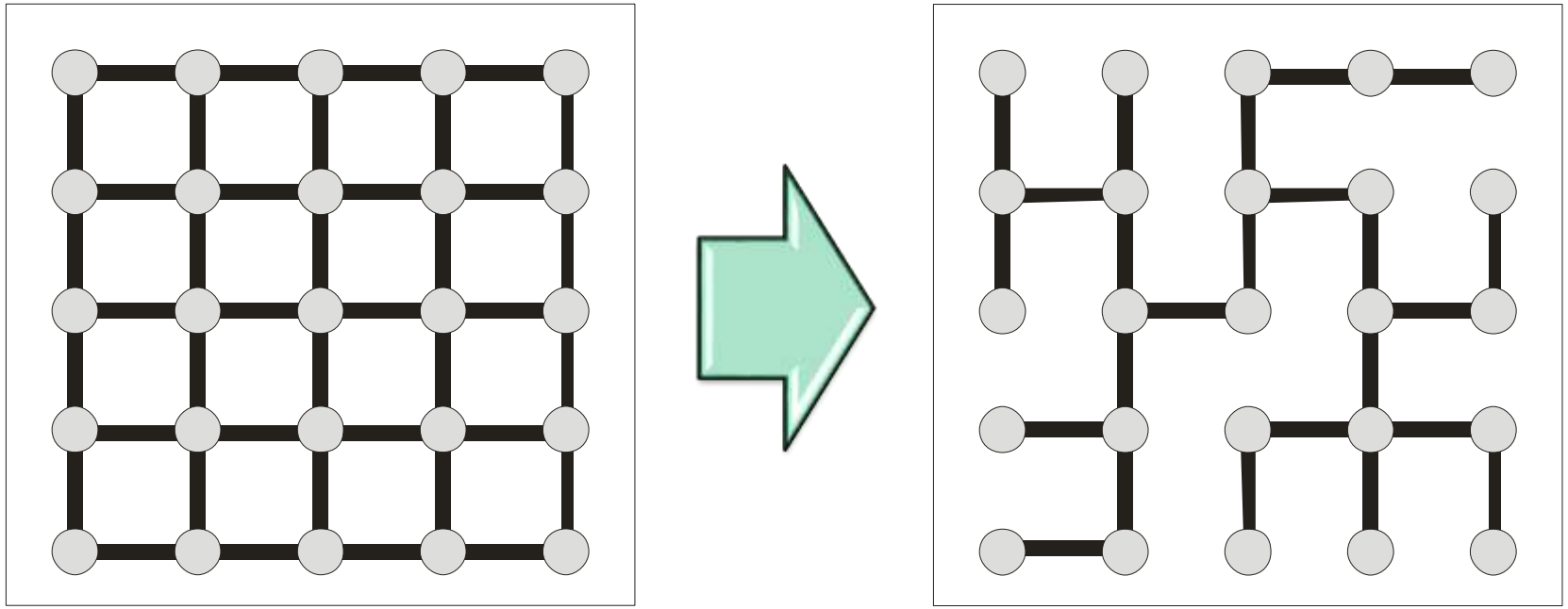
Примеры: M. Bleyer

Минимизация энергии

- Плотное стерео – задача многоклассовой разметки
- Эффективное решение на графе общего вида существует лишь для выпуклых относительно $|d_p - d_q|$ парных потенциалов [Ishikava, 2003]
- Но необходимо использовать модели, сохраняющие границы, а они невыпуклы относительно $|d_p - d_q|$
- Задача становится NP-полной
- Необходимы приближённые алгоритмы
  - Fusion move, Loopy belief propagation, TRW
- Альтернативный вариант – уход от графов общего вида к деревьям

H. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *PAMI*, 2003.

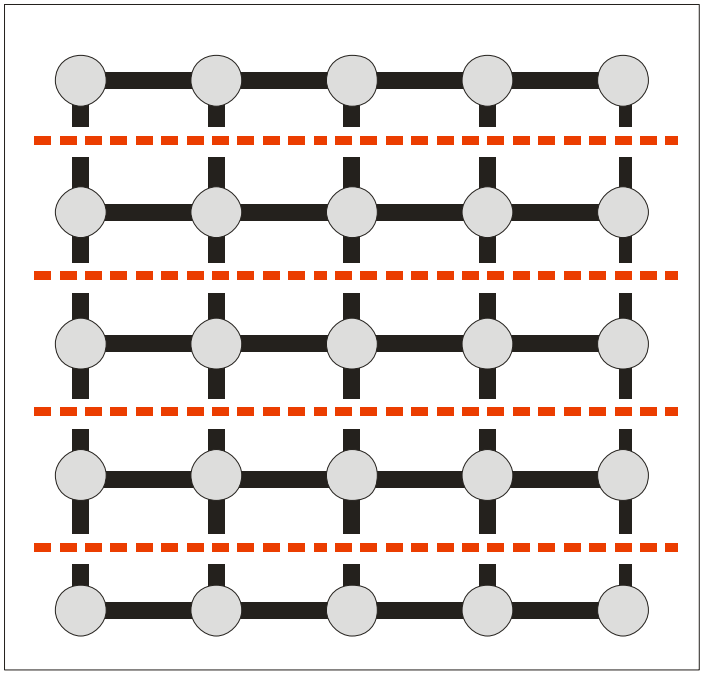
Переход к деревьям

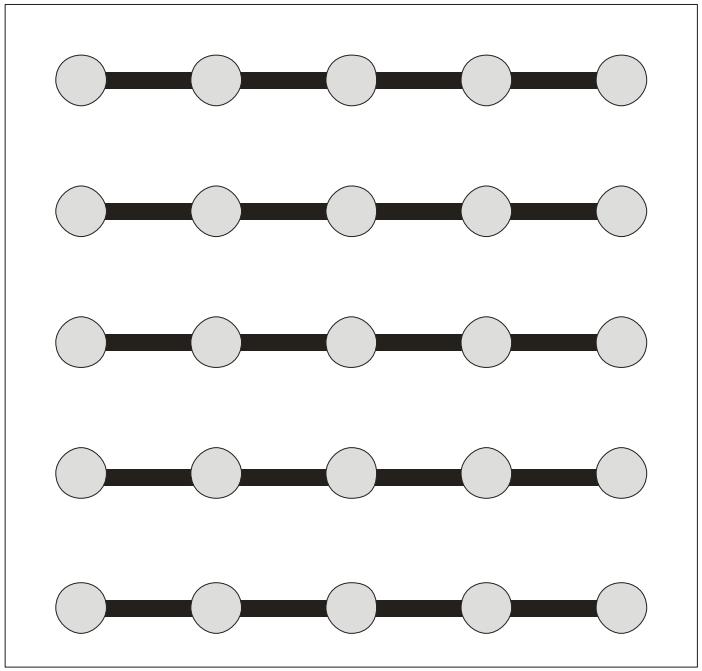


- Отсутствие циклов позволяет использовать метод динамического программирования
  - Глобальный минимум, произвольная энергия, высокая скорость работы
- Главный вопрос – какие ребра убирать?

Рисунки: M. Bleyer

Алгоритм Scanline Optimization





- Удаляются все вертикальные ребра
- Так поступали в первых подобных алгоритмах

Рисунки: M. Bleyer

Алгоритм Scanline Optimization

Очевидная проблема – рассогласованность строк между собой (horizontal streaking)

Пример: M. Bleyer

Алгоритм на основе MST

Идея – не форсировать гладкость между пикселями сильно разного цвета

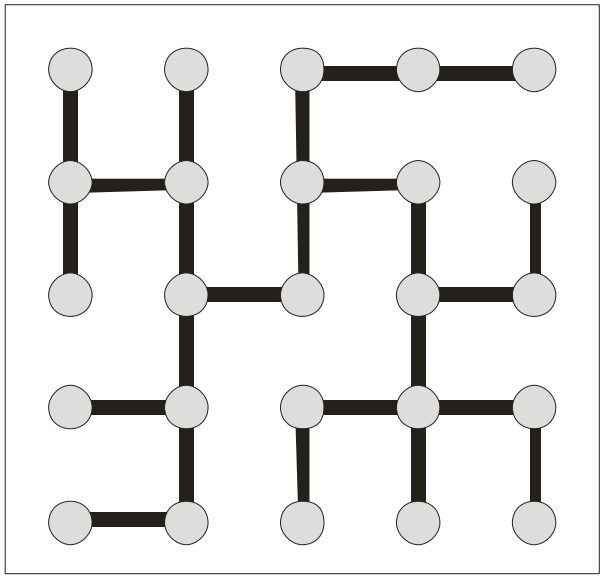


Рисунок: M. Bleyer

Каждому ребру (паре пикселей p и q) присваивается вес:

$$w(p,q) = |I(p) - I(q)|$$

Строится минимальное покрывающее дерево (Minimum Spanning Tree, MST)

O. Veksler. Stereo Correspondence by Dynamic Programming on a Tree. *CVPR*, 2005.

Алгоритм на основе MST

Лучше, чем scanline optimization, но некоторая рассогласованность остается

Алгоритм Semi-Global Matching

В каждом пикселе строится свое дерево

Оптимизация производится вдоль лучей, исходящих из пикселя

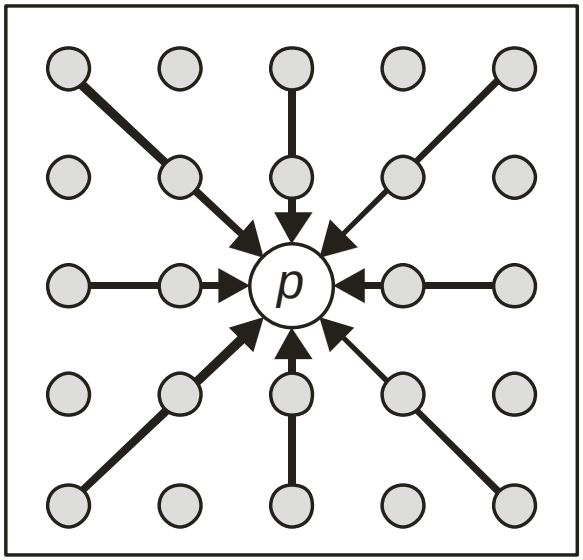
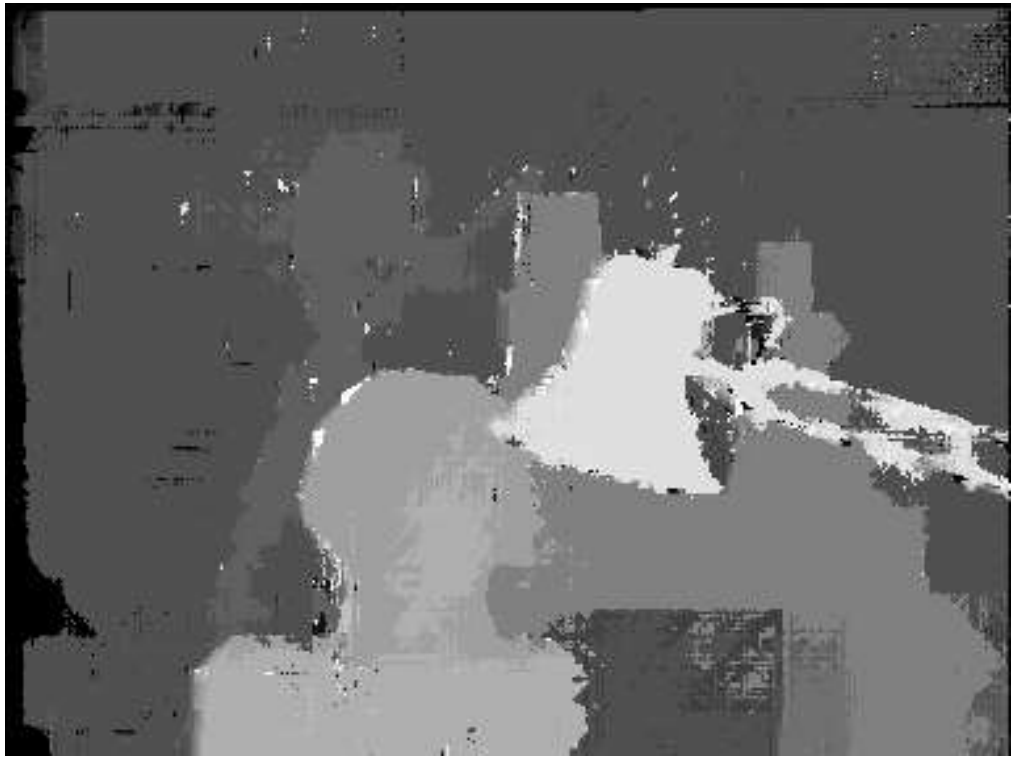


Рисунок: M. Bleyer

Подход не совсем глобальный, но и не локальный

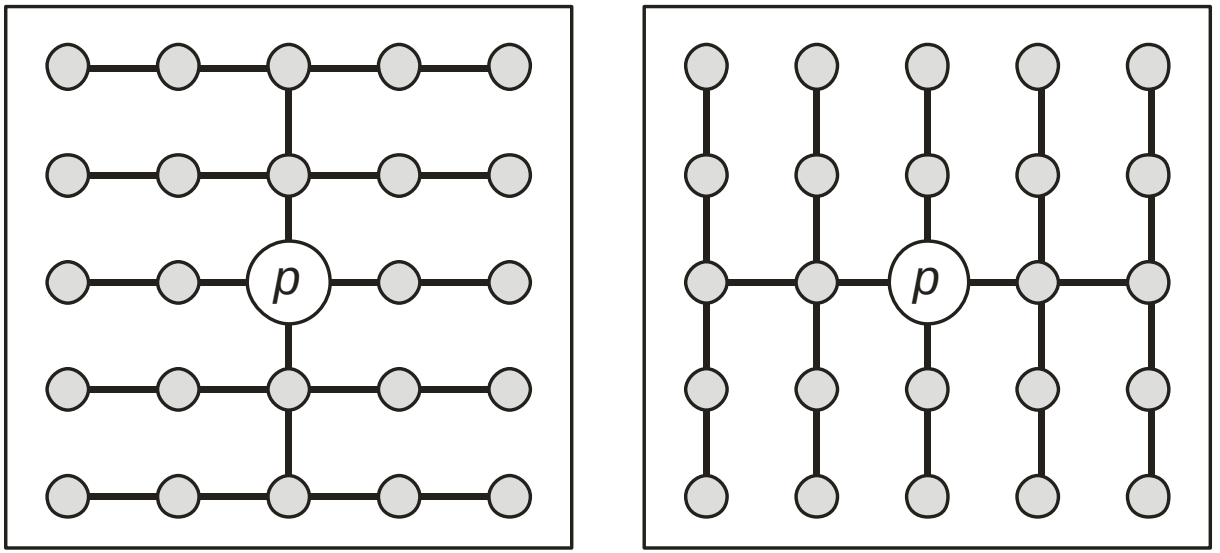
H. Hirschmueller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. *CVPR*, 2005.

Алгоритм Semi-Global Matching



- Рассогласованности нет, но есть «изолированные» пиксели

Алгоритм Simple Tree



Рисунки: M. Bleyer

В каждом пикселе строятся два дерева, совместно покрывающих все изображение. Алгоритм глобальный и лишен недостатка SGM.

M. Bleyer, M. Gelautz. Simple but Effective Tree Structures for Dynamic Programming-based Stereo Matching. *VISAPP*, 2008.

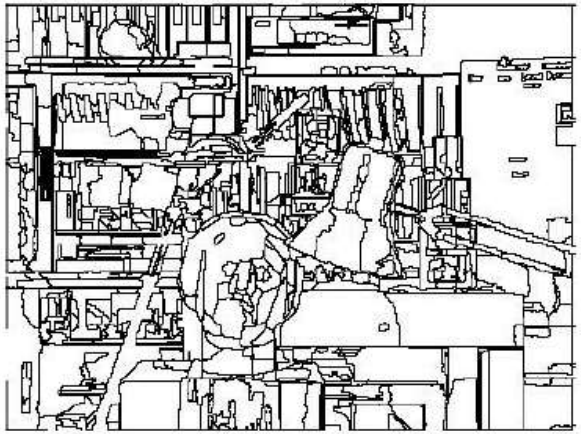
Алгоритм Simple Tree

Пример работы

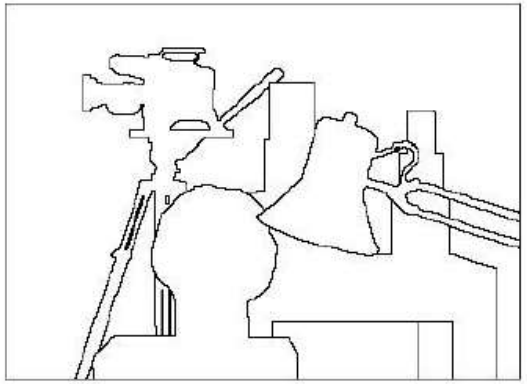
Использование сегментации

- Используется предположение о сегментации («области разрыва диспаритета совпадают с краями на изображении)
- Для его реализации применяют пересегментацию (т.е. сегменты достаточно мелкие, «с запасом»)

Исходное изображение



Результат сегментации



Границы объектов

- Происходит переход из пиксельного пространства в пространство сегментов. Гладкость внутри сегментов форсируется
- Этот подход сейчас показывает наилучшие результаты
  - Возможно, переобучение на Middleburry

Примеры: M. Bleyer

Базовый алгоритм с сегментацией

- Пересегментация
- Инициализация решения
  - Любой локальный алгоритм на пикселях
- Аппроксимация сегментов гладкими поверхностями
  - Модель: плоскость, B-сплайн
  - Средство: RANSAC, голосование и т.д.
- Уточнение разметки сегментов
  - Iterated Conditional Modes (ICM), Cooperative Optimization и др.

Использование сегментации

- Преимущества
  - Надежность в областях со слабой текстурой
  - Снижение размерности задачи (оптимизация на уровне сегментов)
- Недостатки
  - Нет защиты от нарушения предположения о сегментации
  - Сложность выбора модели, описывающей изменение диспаритета внутри сегмента
  - Проблему перекрытий все равно необходимо решать на пиксельном уровне

Stereo Evaluation

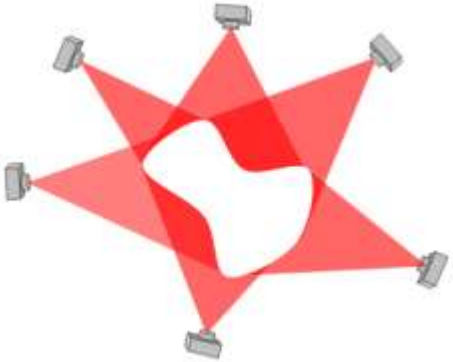
Rank	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	Displets		code	**2.47 %**	3.27 %	0.7 px	0.9 px	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)	
F. Guney and A. Geiger: Displets: Resolving Stereo Ambiguities using Object Knowledge. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
2	MC-CNN			2.61 %	3.84 %	0.8 px	1.0 px	100.00 %	100 s	Nvidia GTX Titan (CUDA, Lua/Torch7)	
J. Zbontar and Y. LeCun: Computing the Stereo Matching Cost with a Convolutional Neural Network. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
3	PRSM		code	2.78 %	**3.00 %**	**0.7 px**	0.7 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.											
4	SPS-StFl			2.83 %	3.64 %	0.8 px	0.9 px	100.00 %	35 s	1 core @ 3.5 Ghz (C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. ECCV 2014.											
5	VC-SF			3.05 %	3.31 %	0.8 px	0.8 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, S. Roth and K. Schindler: View-Consistent 3D Scene Flow Estimation over Multiple Frames. Proceedings of European Conference on Computer Vision. Lecture Notes in, Computer Science 2014.											
6	OSF		code	3.28 %	4.07 %	0.8 px	0.9 px	99.98 %	50 min	1 core @ 3.0 Ghz (Matlab + C/C++)	
M. Menze and A. Geiger: Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
7	CoR		code	3.30 %	4.10 %	0.8 px	0.9 px	100.00 %	6 s	6 cores @ 3.3 Ghz (Matlab + C/C++)	
A. Chakrabarti, Y. Xiong, S. Gortler and T. Zickler: Low-level Vision by Consensus in a Spatial Hierarchy of Regions. CVPR 2015.											
8	SPS-St		code	3.39 %	4.41 %	0.9 px	1.0 px	100.00 %	2 s	1 core @ 3.5 Ghz (C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. ECCV 2014.											
9	PCBP-SS			3.40 %	4.72 %	0.8 px	1.0 px	100.00 %	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Robust Monocular Epipolar Flow Estimation. CVPR 2013.											
10	DDS-SS			3.83 %	4.59 %	0.9 px	1.0 px	100.00 %	1 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
D. Wei, C. Liu and W. Freeman: A Data-driven Regularization Model for Stereo and Flow. 3DTV-Conference, 2014 International Conference on 2014.											
11	StereoSLIC			3.92 %	5.11 %	0.9 px	1.0 px	99.89 %	2.3 s	1 core @ 3.0 Ghz (C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Robust Monocular Epipolar Flow Estimation. CVPR 2013.											
12	PR-Sf+E			4.02 %	4.87 %	0.9 px	1.0 px	100.00 %	200 s	4 cores @ 3.0 Ghz (Matlab + C/C++)	
C. Vogel, S. Roth and K. Schindler: Piecewise Rigid Scene Flow. International Conference on Computer Vision (ICCV) 2013.											
13	PCBP			4.04 %	5.37 %	0.9 px	1.1 px	100.00 %	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)	
K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun: Continuous Markov Random Fields for Robust Stereo Estimation. ECCV 2012.											
14	CSPMS			4.13 %	5.92 %	1.2 px	1.6 px	100.00 %	6 s	4 cores @ 2.5 Ghz (C/C++)	
Anonymous submission											
15	MBM			4.35 %	5.43 %	1.0 px	1.1 px	100.00 %	0.2 s	1 core @ 3.0 Ghz (C/C++)	
Anonymous submission											
16	PR-Sceneflow			4.36 %	5.22 %	0.9 px	1.1 px	100.00 %	150 sec	4 core @ 3.0 Ghz (Matlab - C/C++)	
C. Vogel, S. Roth and K. Schindler: Piecewise Rigid Scene Flow. International Conference on Computer Vision (ICCV) 2013.											
17	CoR Conf		code	4.40 %	5.26 %	1.0 px	1.2 px	96.27 %	6 s	6 cores @ 3.3 Ghz (Matlab + C/C++)	

Резюме бинокулярного стерео

- Бинокулярное стерео разбивается на 3 шага – ректификация, стереосопоставление и триангуляция
- Сейчас для стереосопоставления используют комбинацию нейросетей для вычисления похожести пикселов и оптимизации по изображению
- Можно также использовать карты сегментации и локальную оценку параметров поверхностей
- Методы продолжают активно развиваться

Многовидовая реконструкция

Многовидовое стерео - формально



Вход:

- N фотографий (в один момент времени) объекта с известной калибровкой

Выход:

- 3D модель объекта – геометрическая модель и текстура

Данные

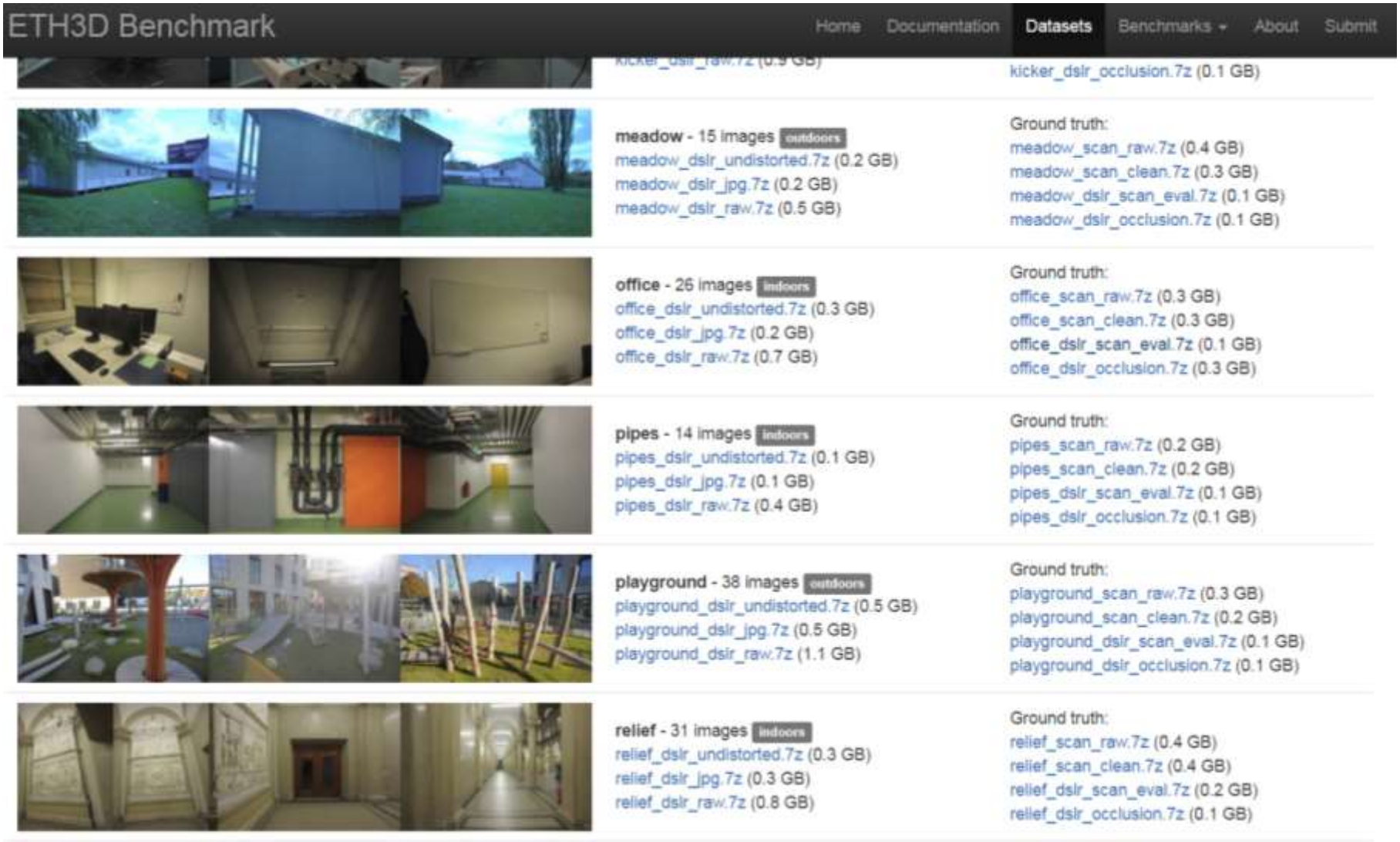
- Данные сложно получить, поэтому их ограниченное количество
- Вот обзор из работы 2017 года:

Benchmark	Setting	Resolution	Online Eval.	6DoF Motion	MVS	Stereo	Video	Varying FOV
Middlebury MVS [38]	Laboratory	0.3 Mpx	✓		✓			
Middlebury [32, 33]	Laboratory	6 Mpx	✓			✓		
DTU [17]	Laboratory	2 Mpx			✓			
MPI Sintel [1]	Synthetic	0.4 Mpx	✓	✓		✓	✓	
KITTI [6, 24]	Street scenes	0.5 Mpx	✓		✓	✓	✓	
Strecha [40]	Buildings	6 Mpx		✓	✓			
ETH3D (Proposed)	Varied	0.4 / 24 Mpx	✓	✓	✓	✓	✓	✓

Table 1. Comparison of existing state-of-the-art benchmarks with our new dataset. Among other factors, we differentiate between different scene types (*e.g.*, staged scenes captured in a laboratory vs. synthetic scenes), whether the camera undergoes a restricted or a full 6 degrees-of-freedom (DoF) motion, or whether cameras with different fields-of-view (FOV) are used.

Thomas Schops et. al. Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos, CVPR 2017

ETH3D



https://www.eth3d.net

Фото-согласование

Основной принцип многовидового стерео

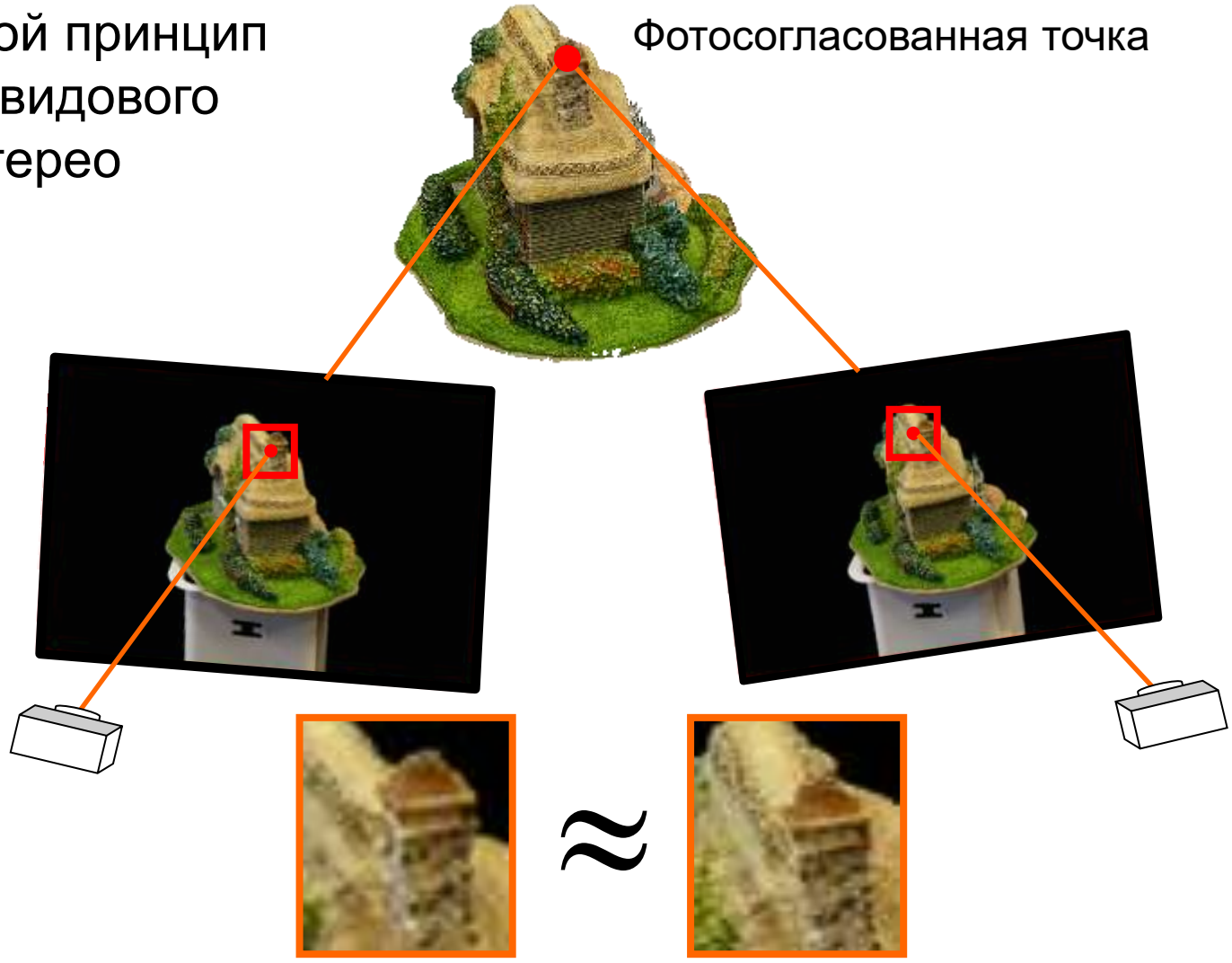
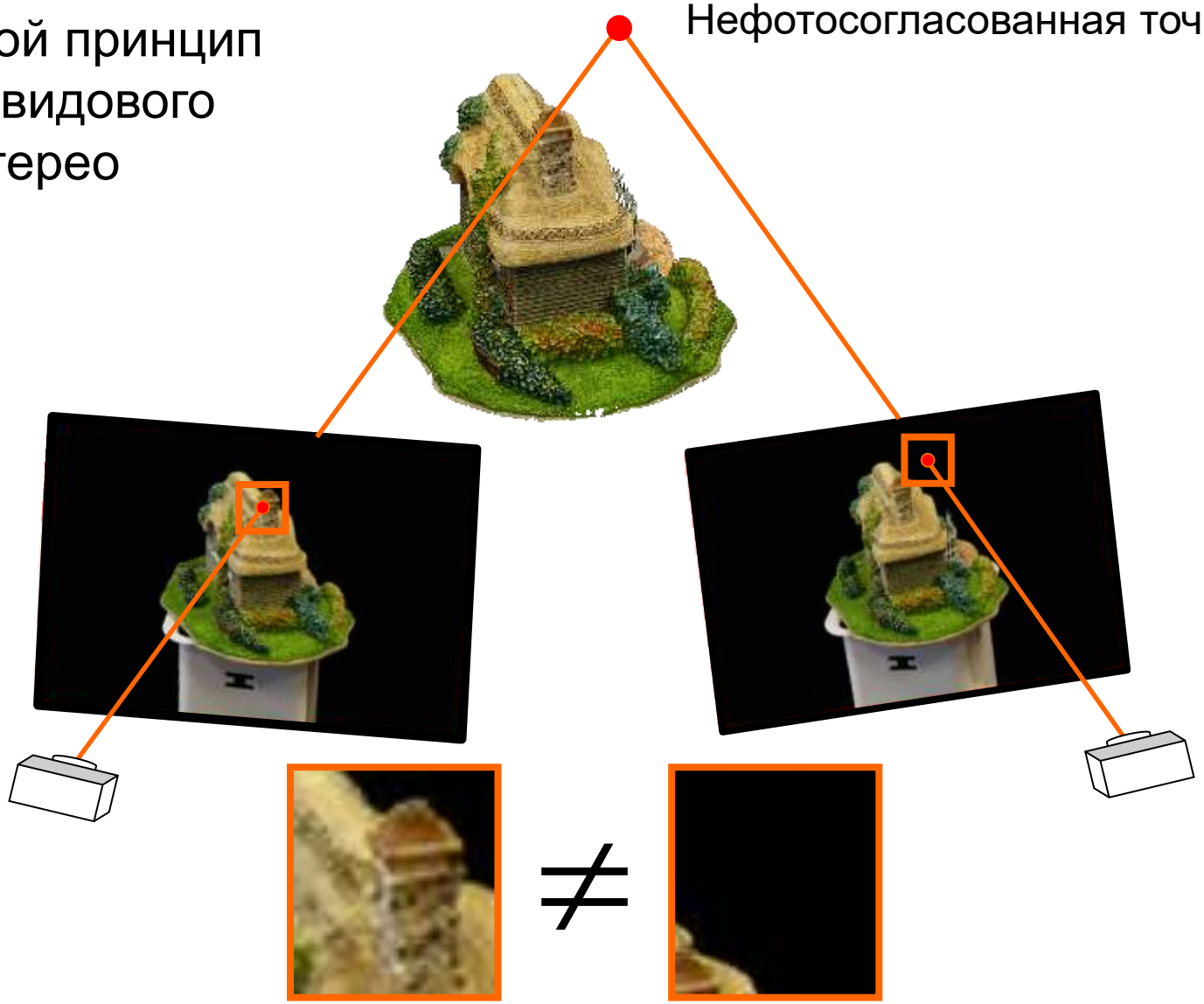


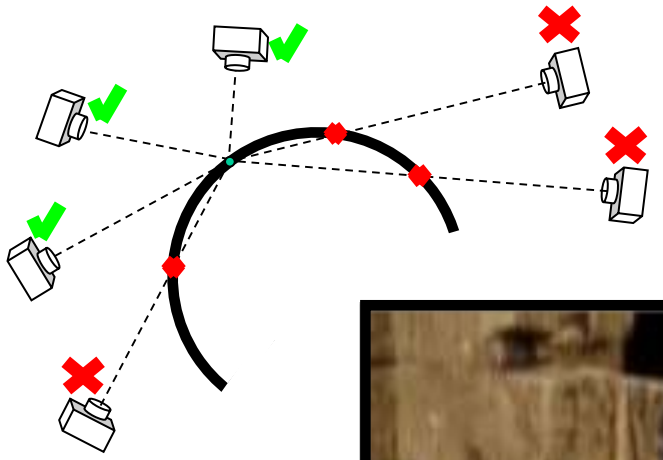
Фото-согласование

Основной принцип многовидового стерео



Проблемы фото-согласования

- Видимость камеры



- Ошибки метрики
  - Повтор текстуры
  - Отсутствие текстуры
  - Блики

Разные методы по разному подходят к решению указанных проблем

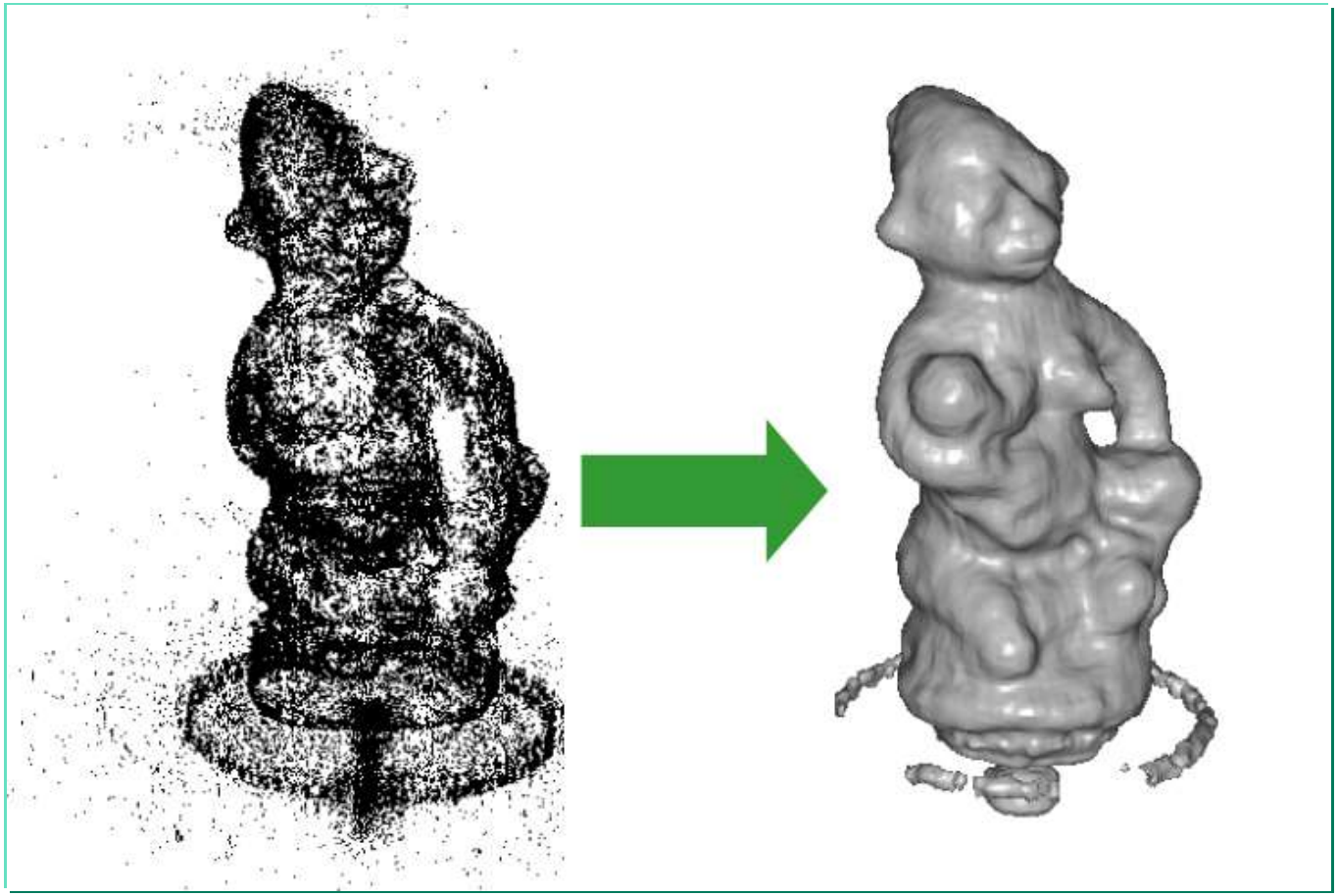
Схема объединения карт глубины

1. Вычисляем карту глубины

2. Объединяем карты глубины в объеме

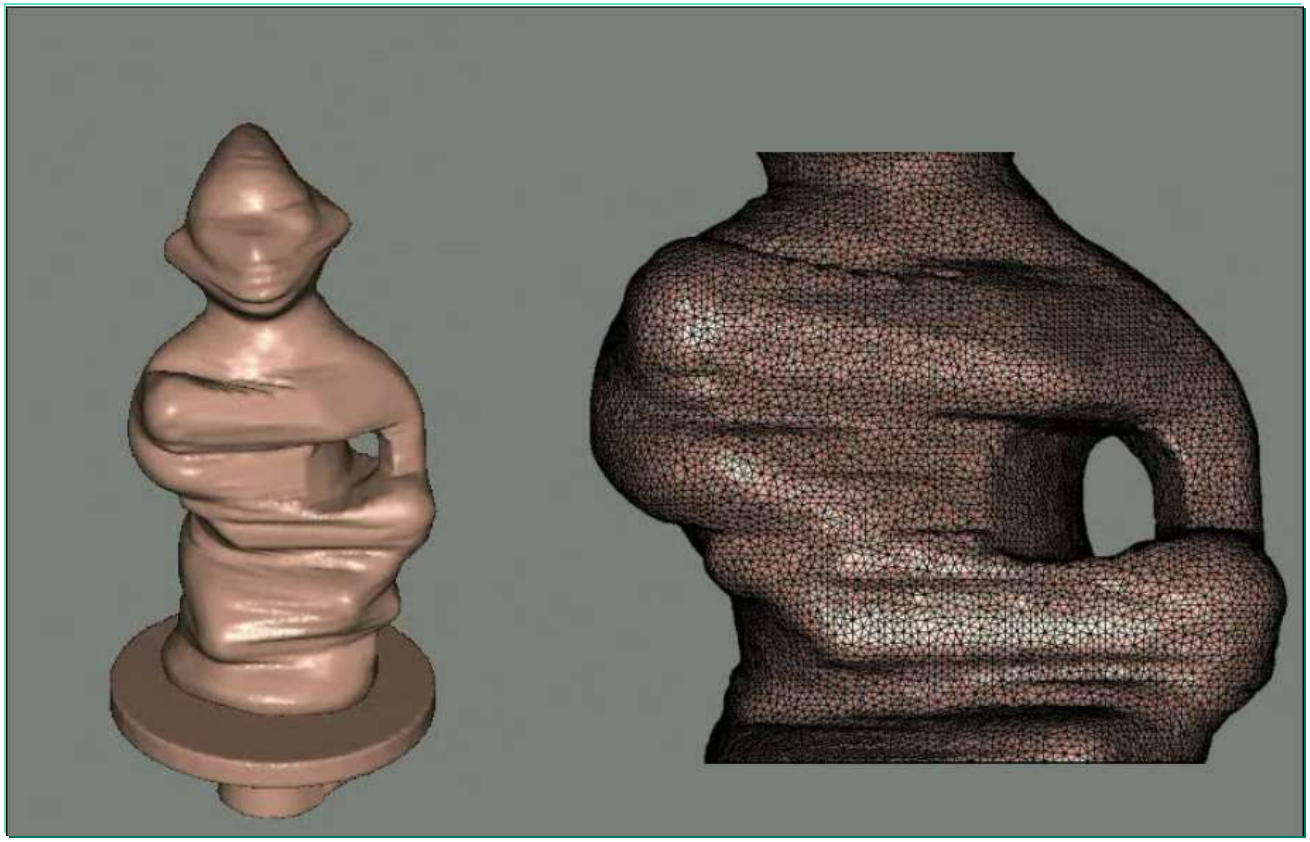
3. Извлекаем 3д поверхность

Извлечение поверхности



- Объединяем облака точек от карт плотного стерео от всех опорных изображений
- Триангулируем сетку и фильтруем выбросы

Деформируемые поверхности



- Оптимизируем сетку, добиваясь фото-согласованности поверхности
- Очень медленно, бывают локальные минимумы

Пример работы

Оптимизация сетки позволяет удалить ошибки из облака точек, и получить гладкие и точные поверхности

Методы на основе фрагментов

[Lhuillier and Quan, PAMI 05]

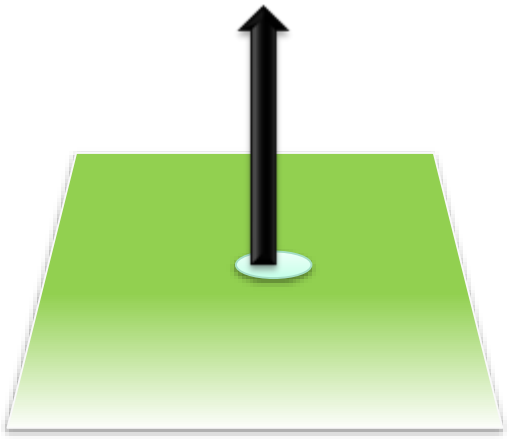
[Furukawa and Ponce, CVPR 07]

[Habbecke and Kobbelt, CVPR 07]

Одни из самых точных современных алгоритмов основаны на фрагментах

PMVS (Patch-based Multi-View Stereo)

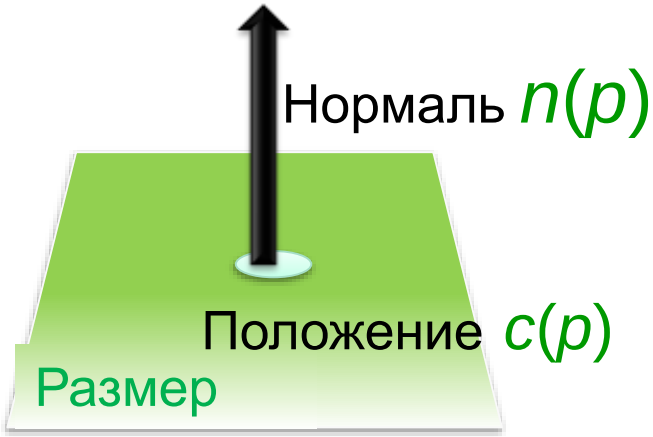
- «Фрагмент» (patch) – это модель небольшого участка поверхности объекта
- Поскольку участок небольшой, то можно описать его плоской поверхностью (квадратом)



Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. PAMI, 32(8):1362–1376, 2010

Определение фрагмента

- Фрагмент p определен
  - Положением $c(p)$
  - Нормалью $n(p)$
  - Видимыми изображениями $V(p)$
- Размер выбирается так, чтобы p был примерно 9x9 пикселей в $V(p)$

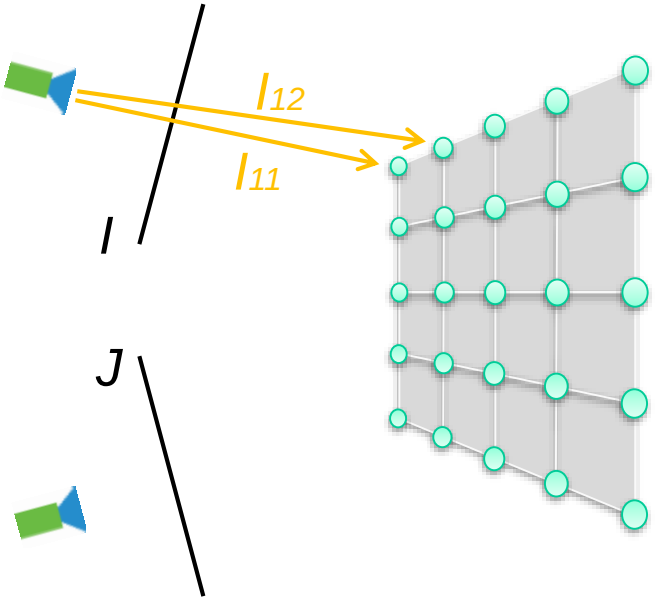


Видимые изображения $V(p)$

Фотосогласованность

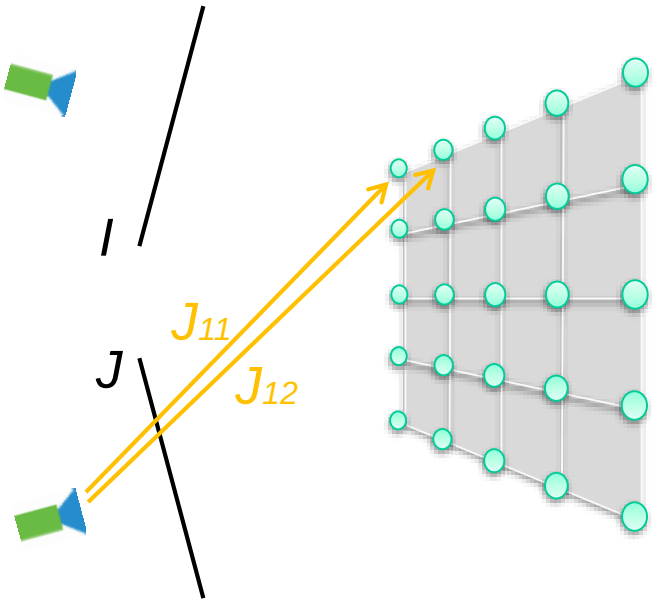
- Вычислим фотосогласованность $N(I, J, p)$ фрагмента p между изображениями I и J

I_{xy}: цвет в изображении I



Фотосогласованность

- Вычислим фотосогласованность $N(I, J, p)$ фрагмента p между изображениями I и J



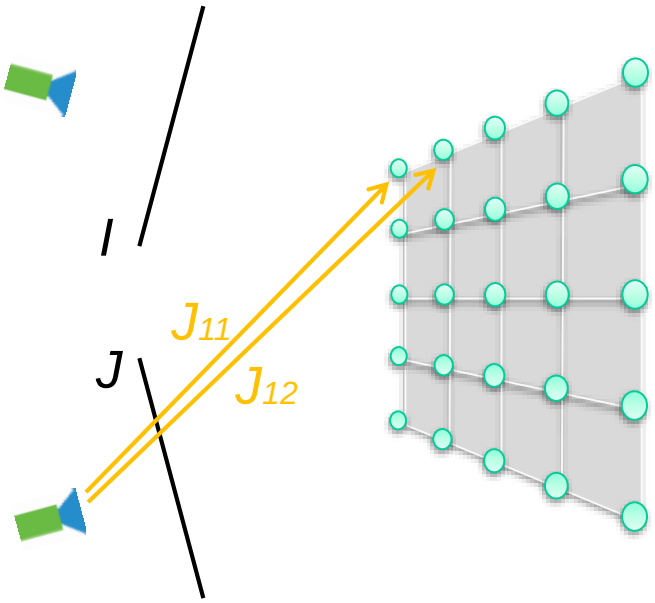
I_{xy}: цвет в изображении I

J_{xy}: цвет в изображении J

$$N(I,J,p)=\frac{\sum(I_{xy}-\overline{I_{xy}})\cdot(J_{xy}-\overline{J_{xy}})}{\sqrt{(I_{xy}-\overline{I_{xy}})^2}\sqrt{(J_{xy}-\overline{J_{xy}})^2}}$$

Фотосогласованность

- Перейдём от случаях 2х изображений к случаю многих изображений:



$$N(I,J,p)=\frac{\sum(I_{xy}-\overline{I_{xy}})\cdot(J_{xy}-\overline{J_{xy}})}{\sqrt{(I_{xy}-\overline{I_{xy}})^2}\sqrt{(J_{xy}-\overline{J_{xy}})^2}}$$

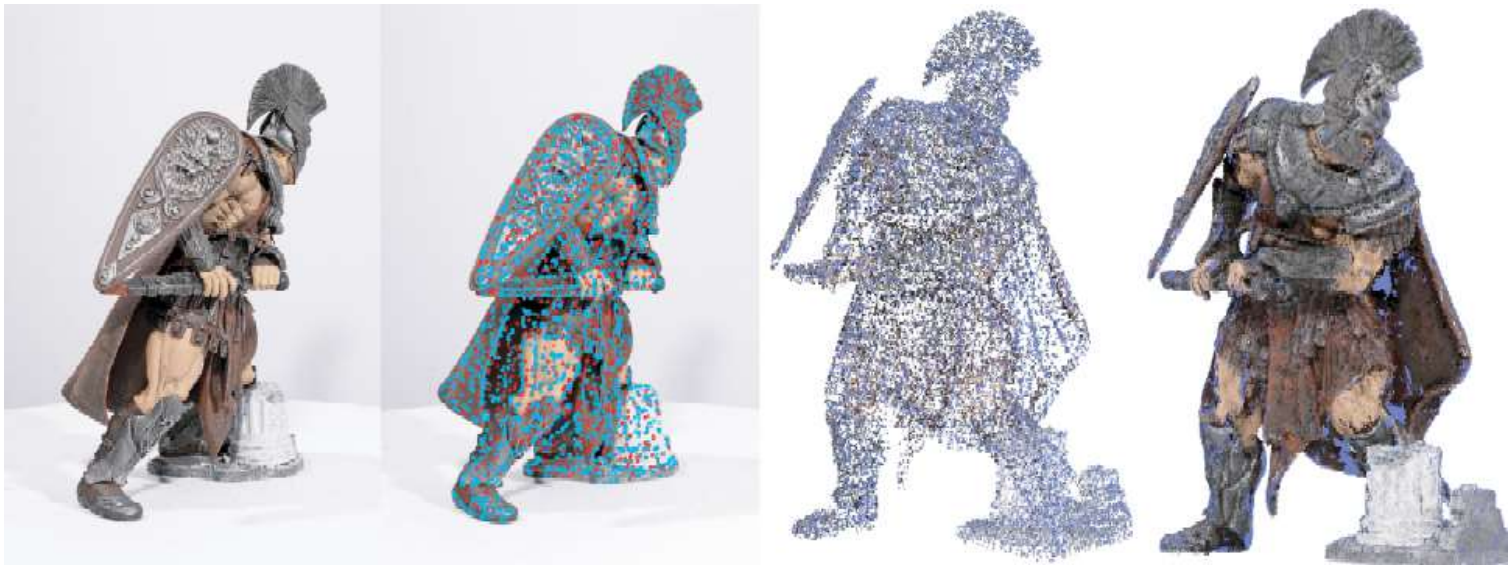
$$N(p)=\frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}N(I_i,I_j,p)}{(n+1)n/2}$$

Алгоритм на основе фрагментов

#1. Поиск особых точек

#2. Инициализация точек

#3. Расширение фрагментов и фильтрация



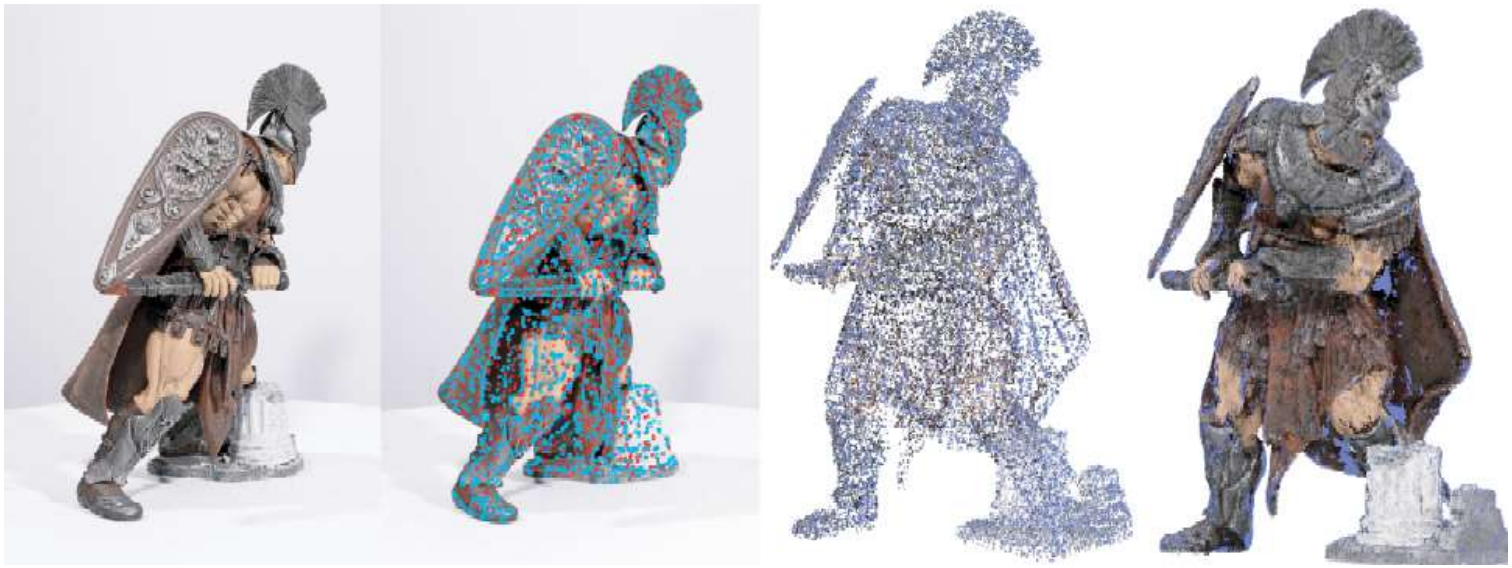
Input image #1 #2 #3

Алгоритм на основе фрагментов

#1. Поиск особых точек

#2. Инициализация фрагментов

#3. Расширение фрагментов и фильтрация



Input image #1 #2 #3

Поиск особых точек

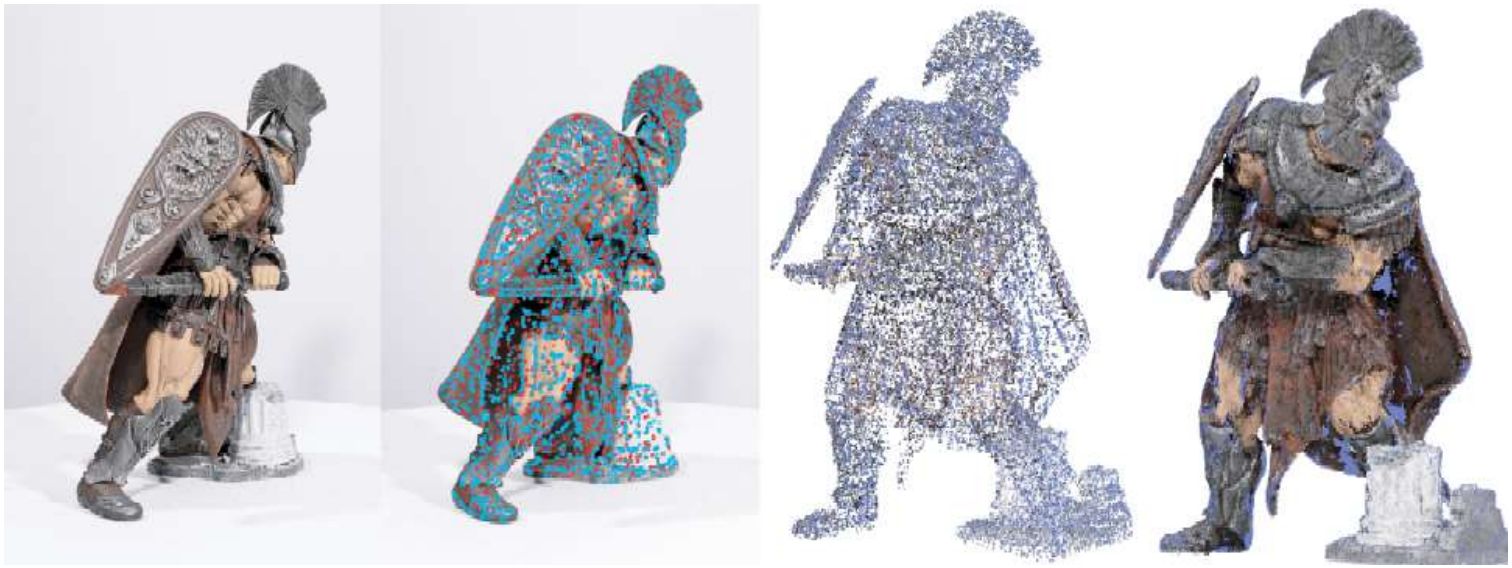
- Harris corner detector (corners)
- Difference of Gaussian (blobs)

Алгоритм на основе фрагментов

#1. Поиск особых точек

#2. Инициализация фрагментов

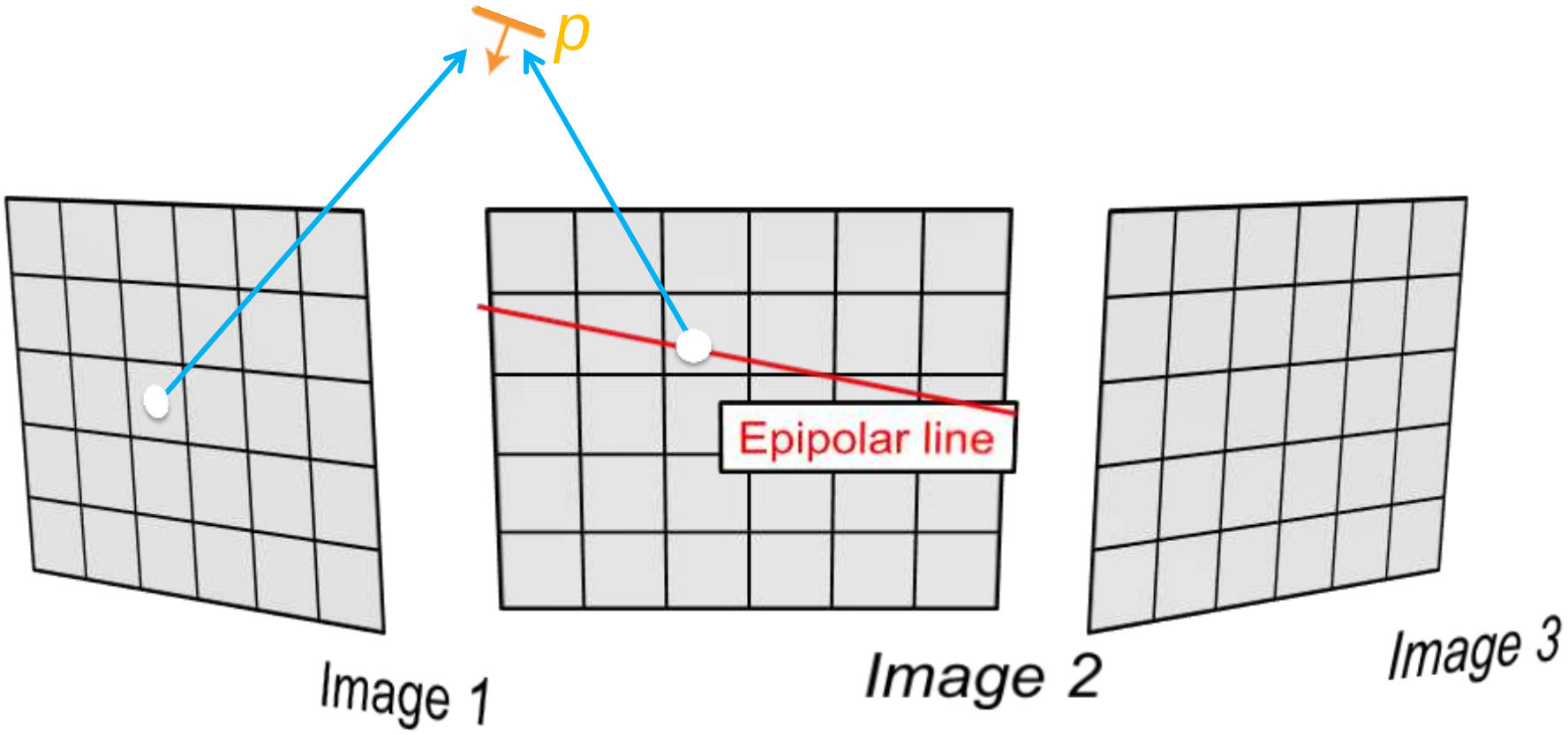
#3. Расширение фрагментов и фильтрация



Input image #1 #2 #3

Инициализация фрагментов

$c(p)$: триангуляция
$n(p)$:
$V(p)$:

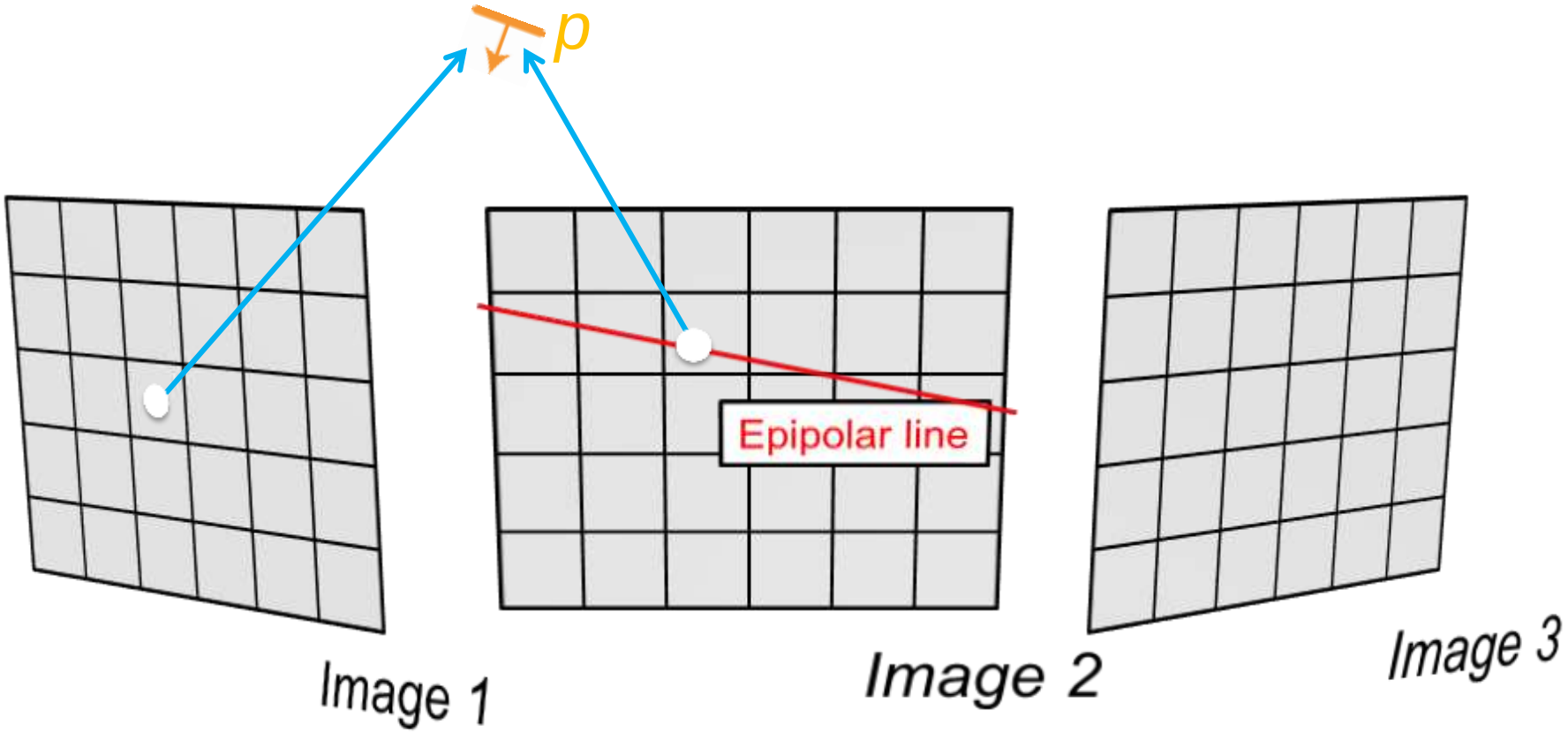


Инициализация фрагментов

c(*p*): триангуляция

n(*p*): параллельно к *Image1*

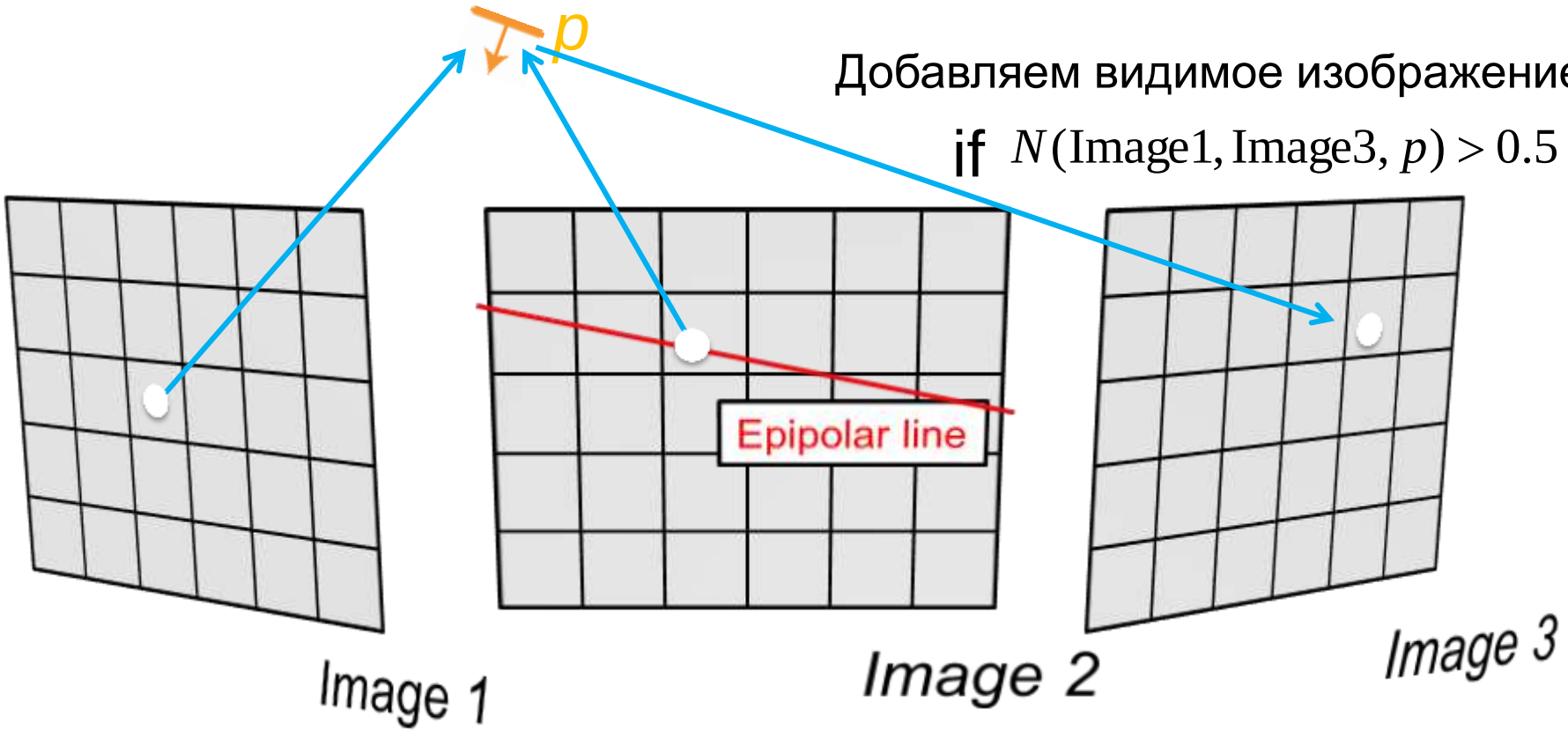
V(*p*): {*Image1*, *Image2*}



Инициализация фрагментов

c(*p*): триангуляция
n(*p*): параллельно к *Image1*
V(*p*): {*Image1*, *Image2*}

c(*p*): не меняем
n(*p*): параллельно к *Image1*
V(*p*): {*Image1*, *Image2*, *Image3*}

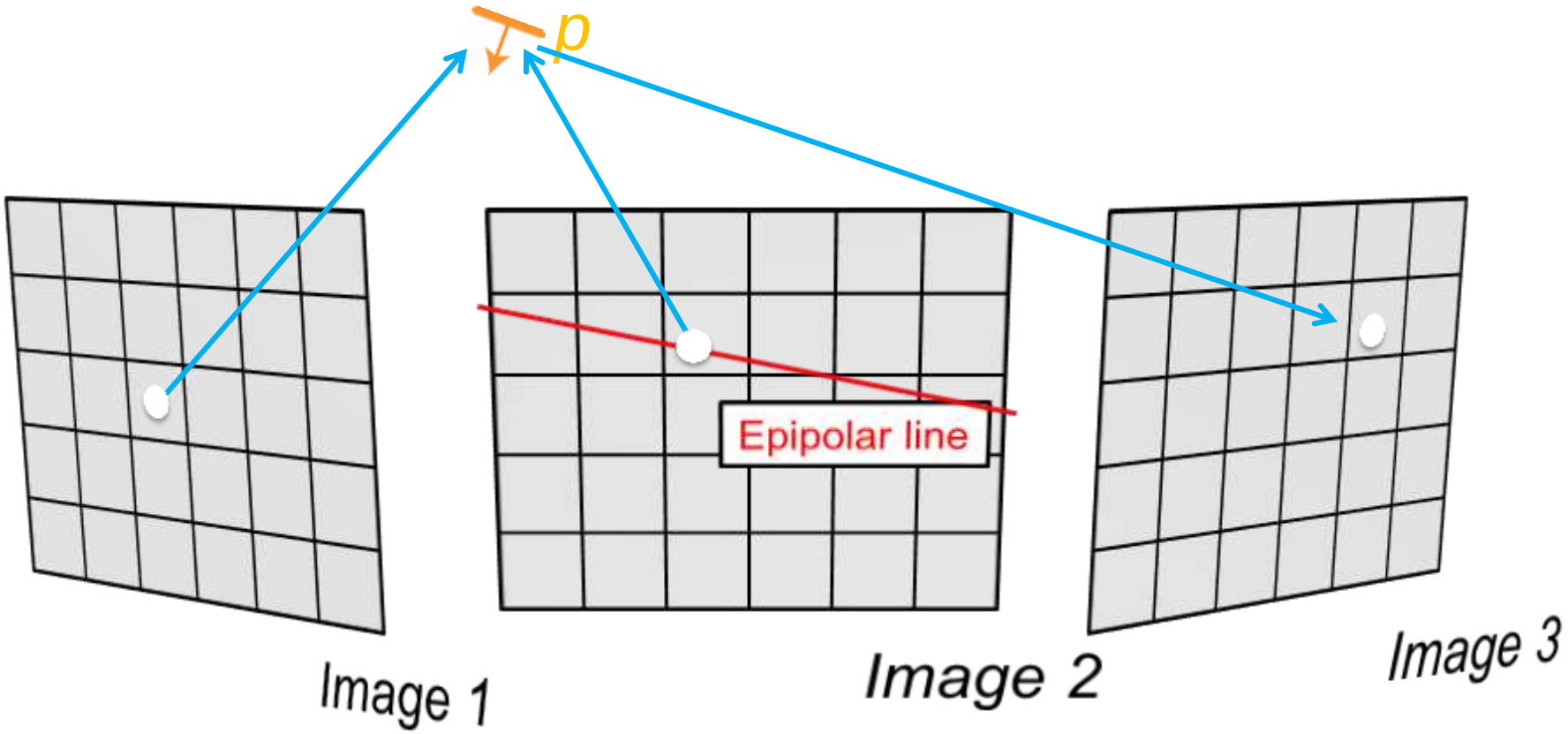


Добавляем видимое изображение

if $N(\text{Image1}, \text{Image3}, p) > 0.5$

Инициализация фрагментов

c(*p*): триангуляция
n(*p*): параллельно к *Image1*
V(*p*): {*Image1*, *Image2*, *Image3*}



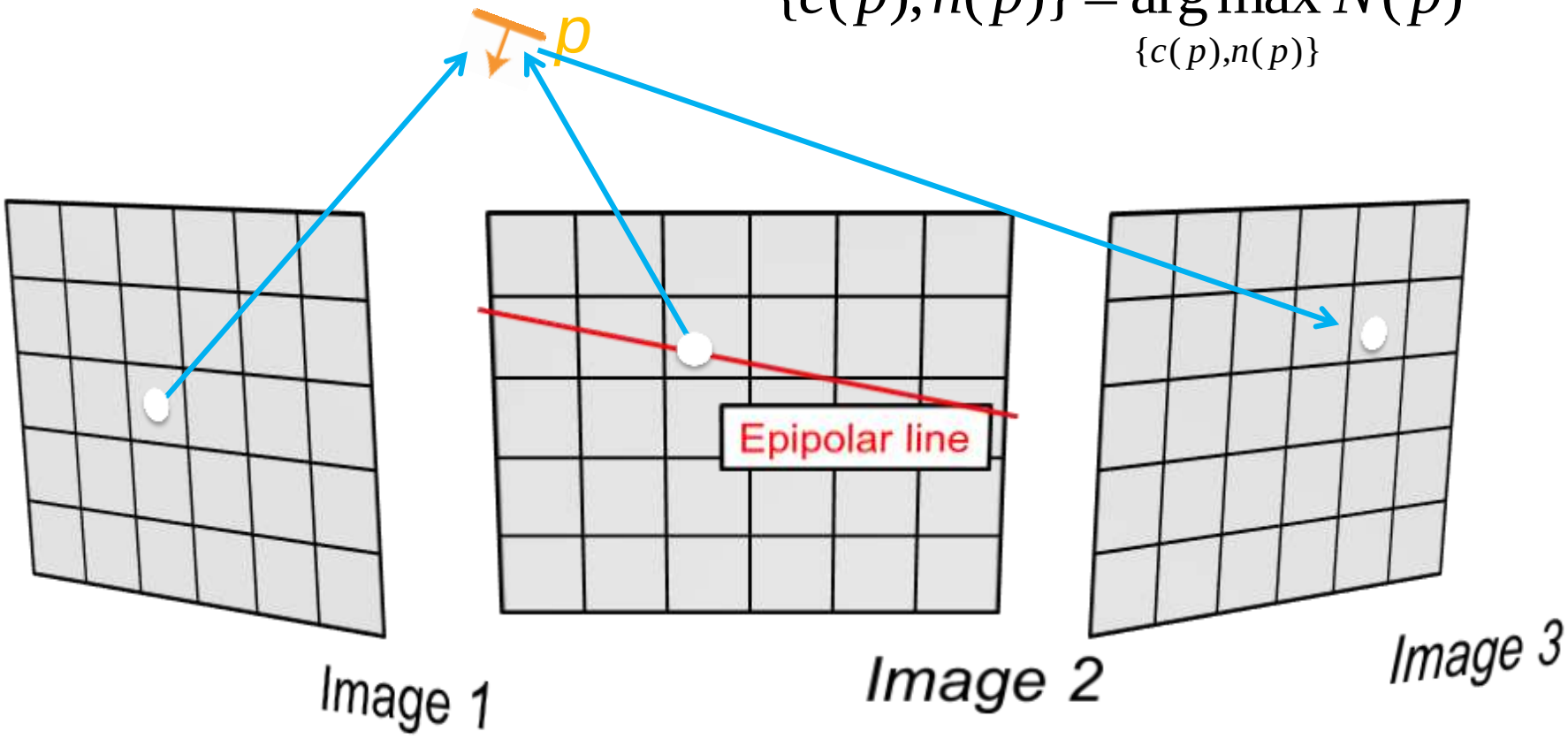
Инициализация фрагментов

c(*p*): уточним

n(*p*): уточним

V(*p*): {*Image1*, *Image2*, *Image3*}

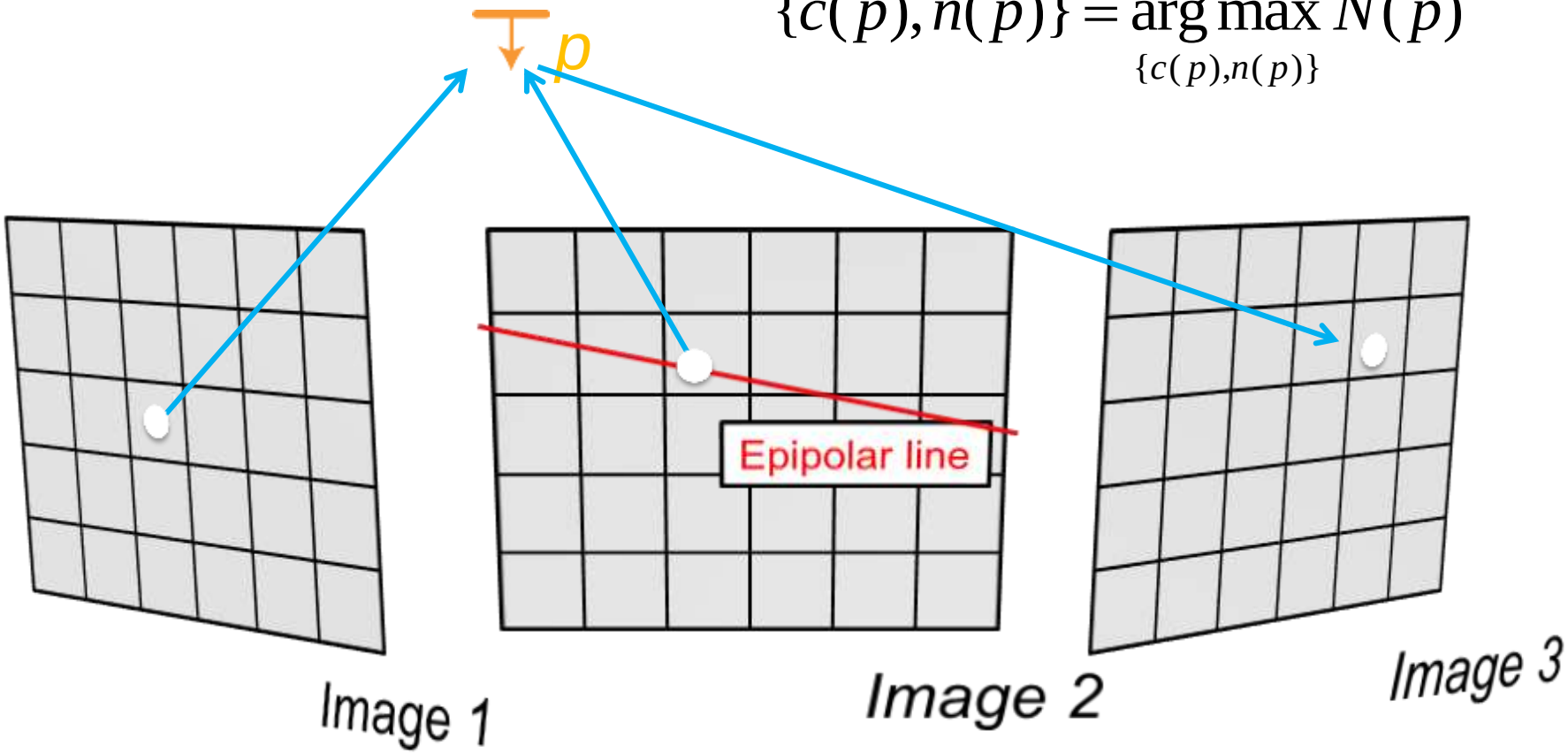
$$\{c(p), n(p)\} = \arg\max_{\{c(p),n(p)\}} N(p)$$



Инициализация фрагментов

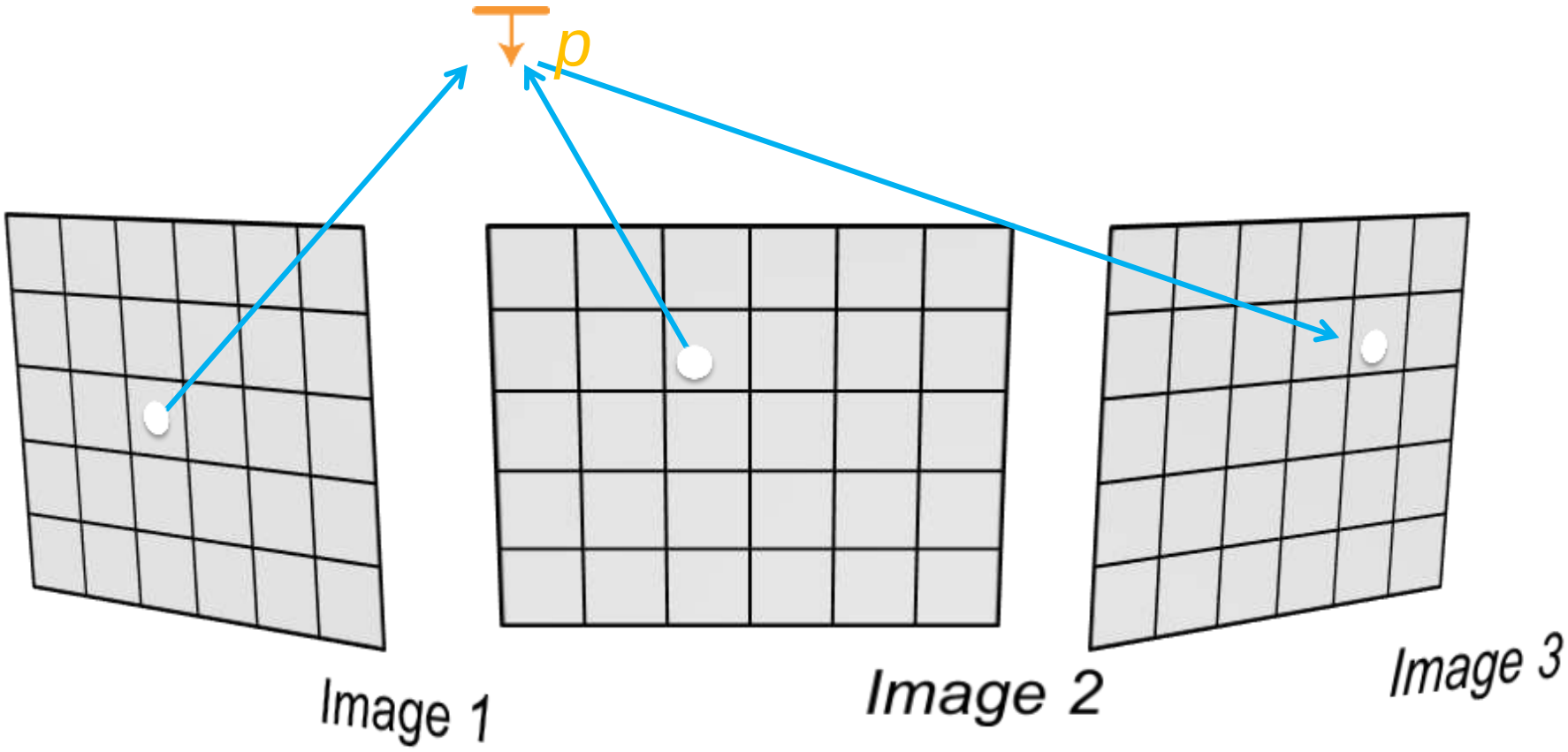
c(*p*): уточним
n(*p*): уточним
V(*p*): {*Image1*, *Image2*, *Image3*}

$$\{c(p), n(p)\} = \arg\max_{\{c(p),n(p)\}} N(p)$$

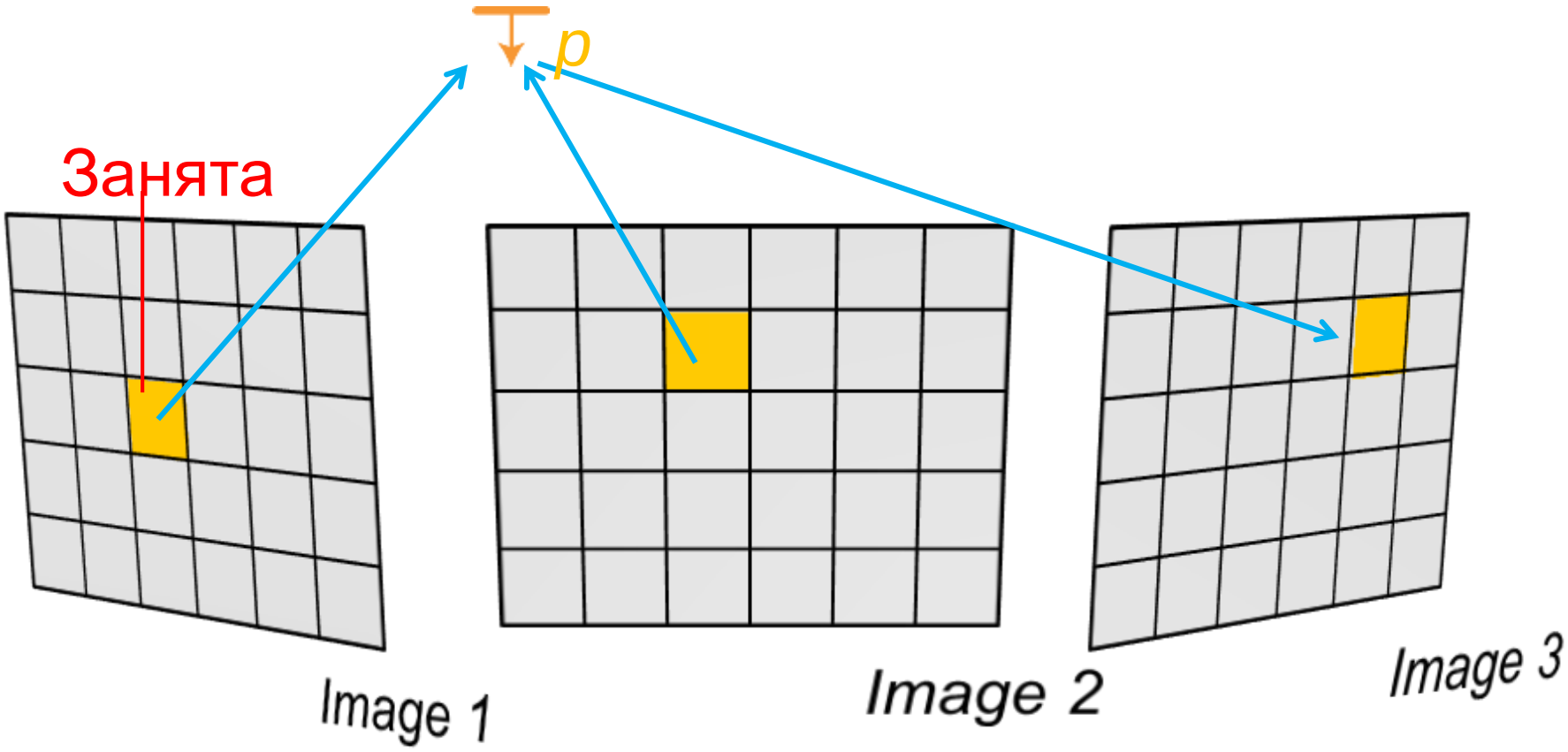


Инициализация фрагментов

Верификация
(обновим $V(p)$ и проверим $|V(p)|\geq 3$)

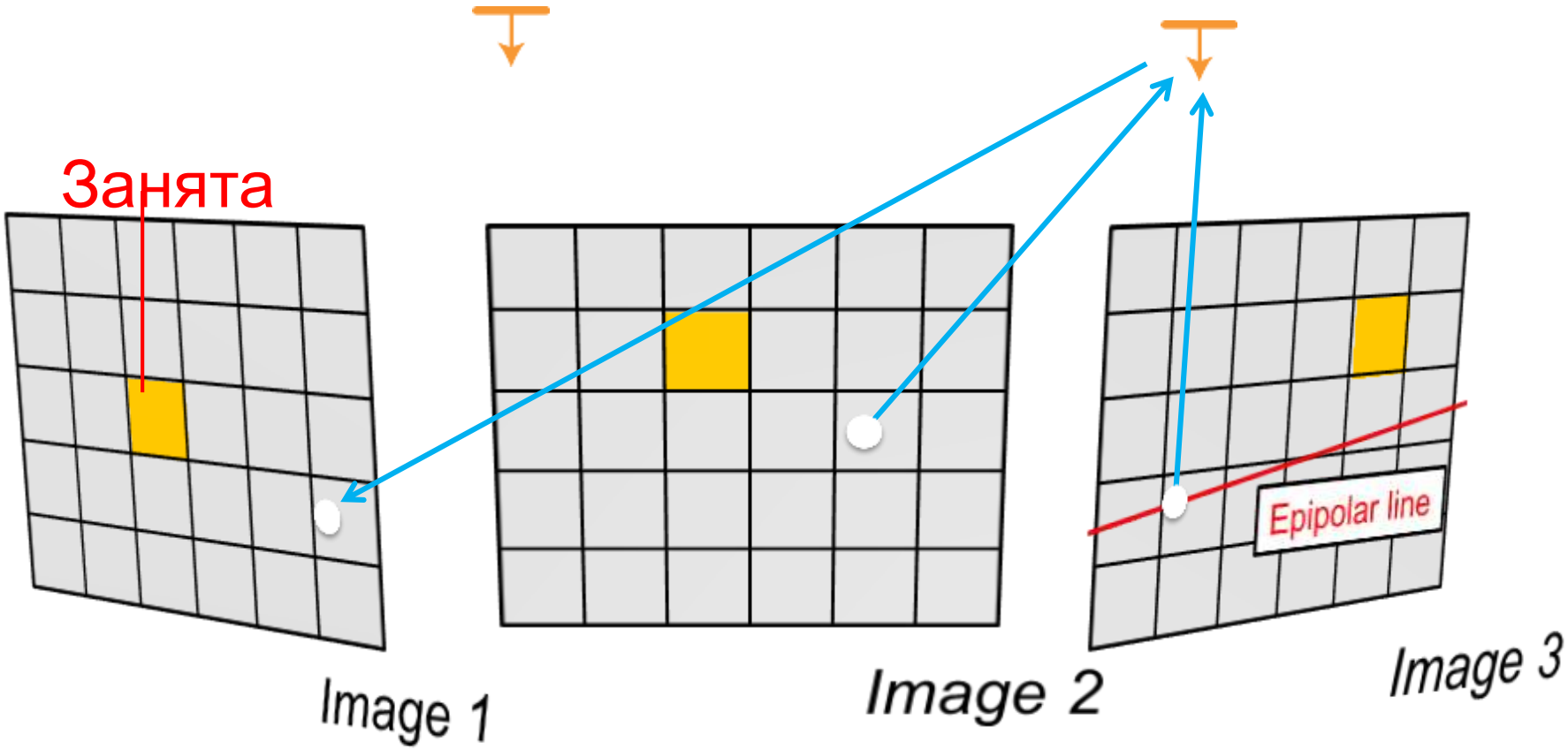


Инициализация фрагментов



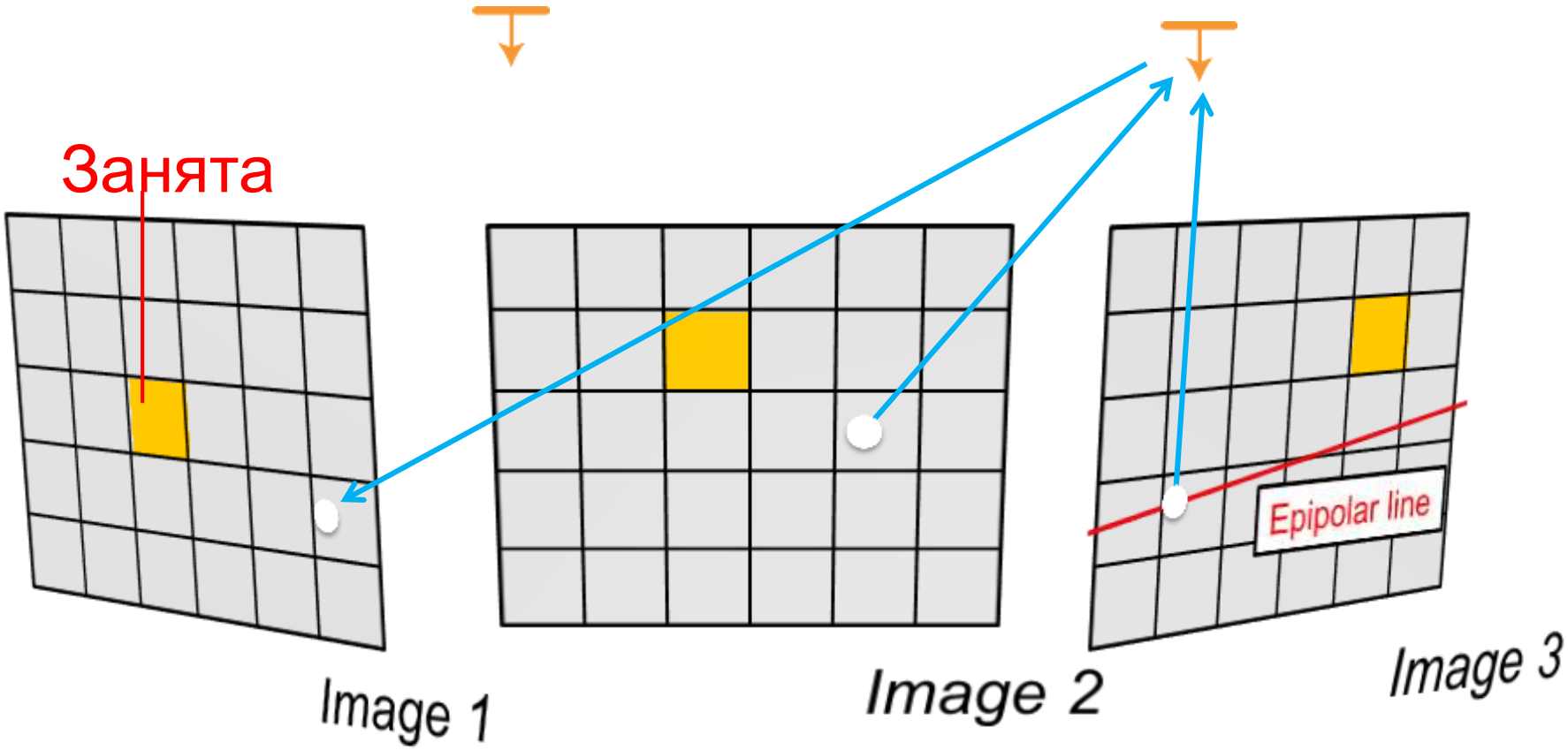
Инициализация фрагментов

- Повторяем для всех найденных особых точек



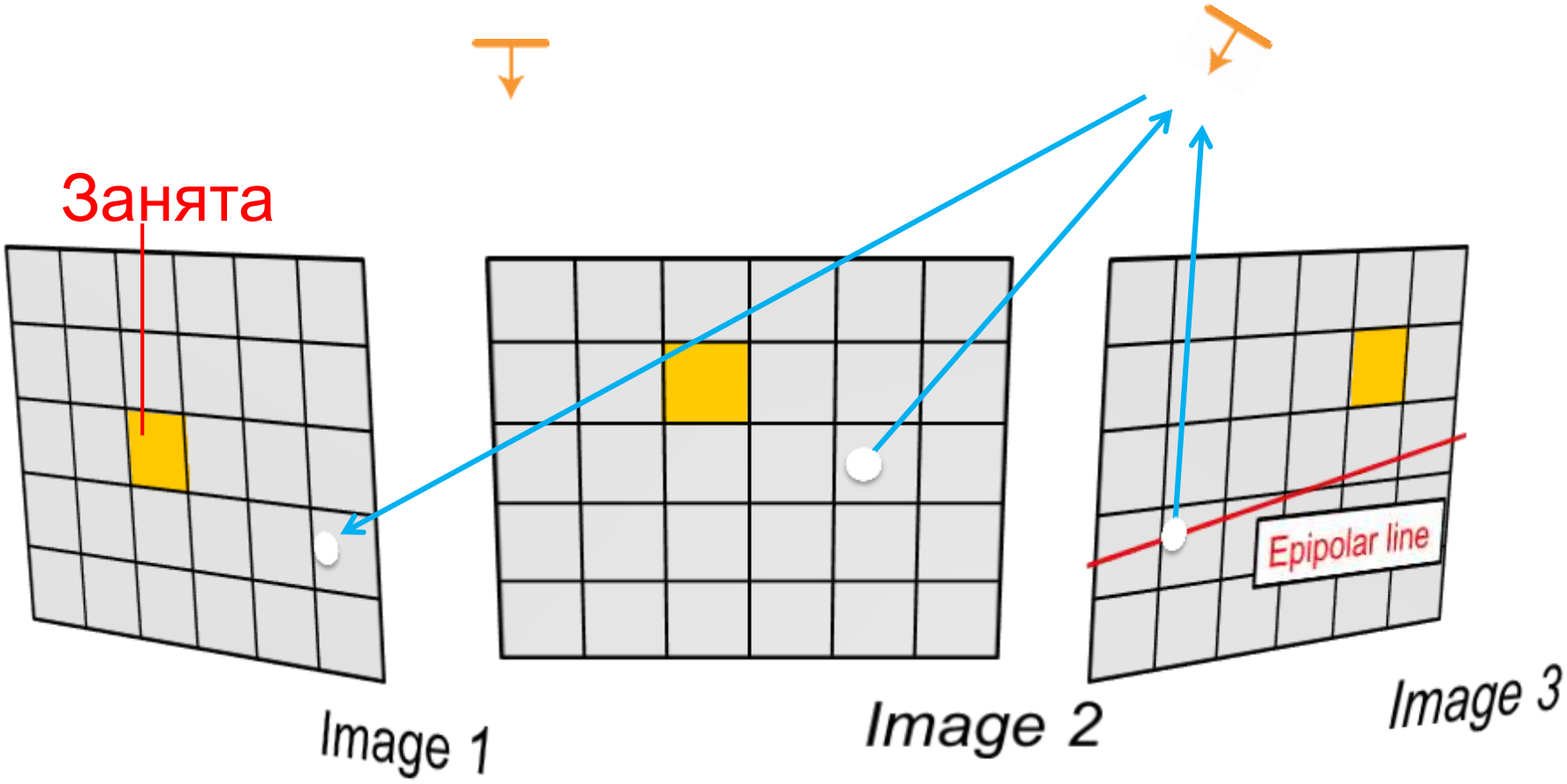
Инициализация фрагментов

- Повторяем для всех найденных особых точек



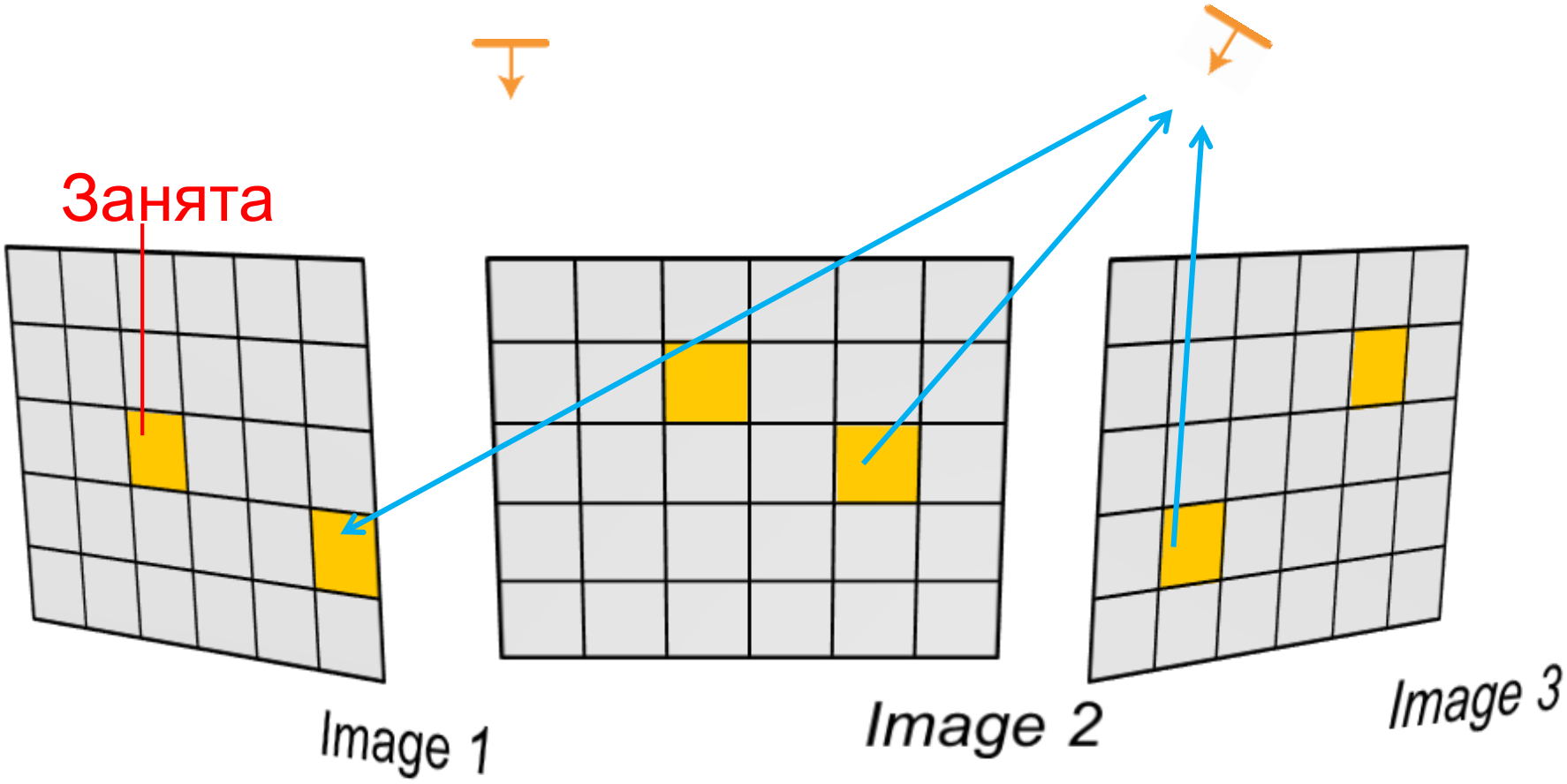
Инициализация фрагментов

- Повторяем для всех найденных особых точек



Инициализация фрагментов

- Повторяем для всех найденных особых точек

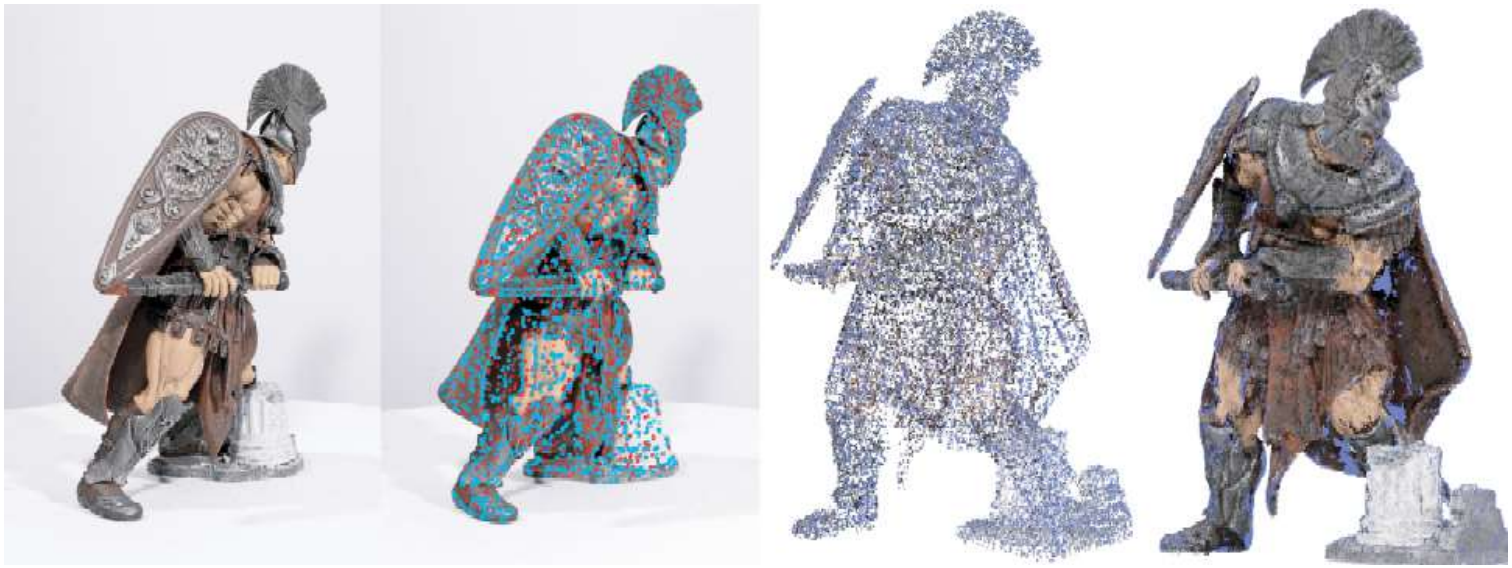


Алгоритм на основе фрагментов

#1. Поиск особых точек

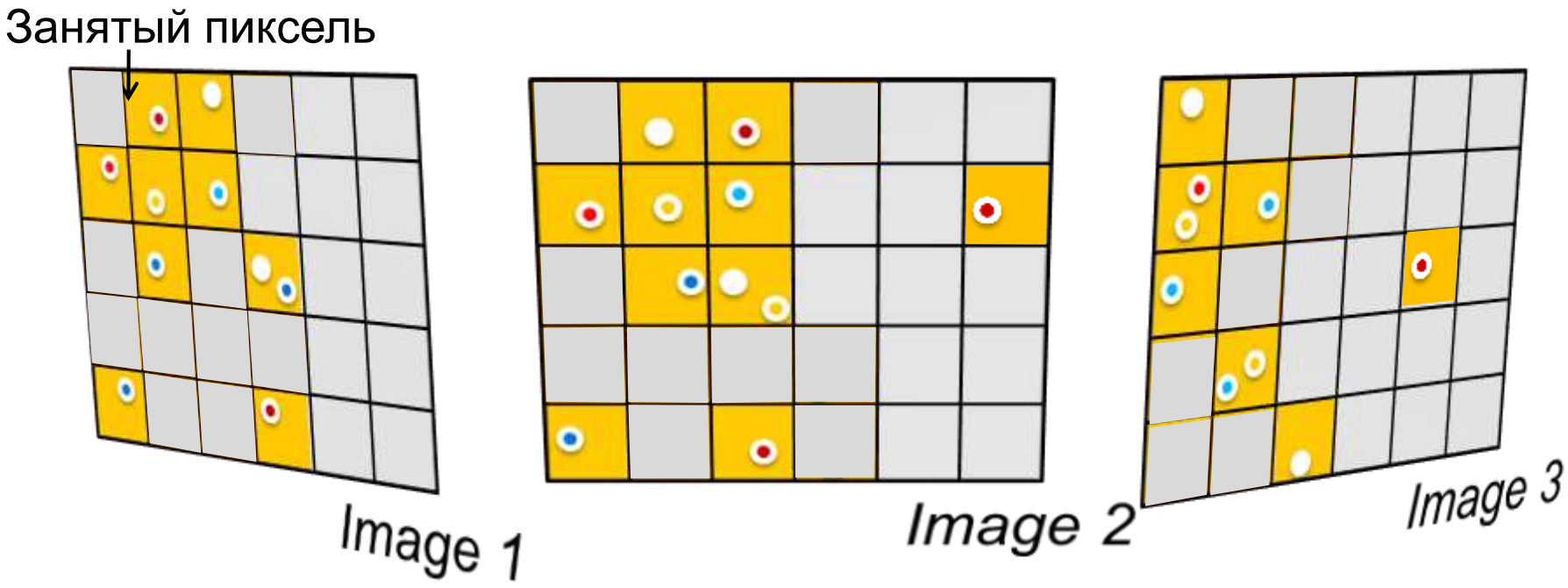
#2. Инициализация фрагментов

#3. Расширение фрагментов и фильтрация

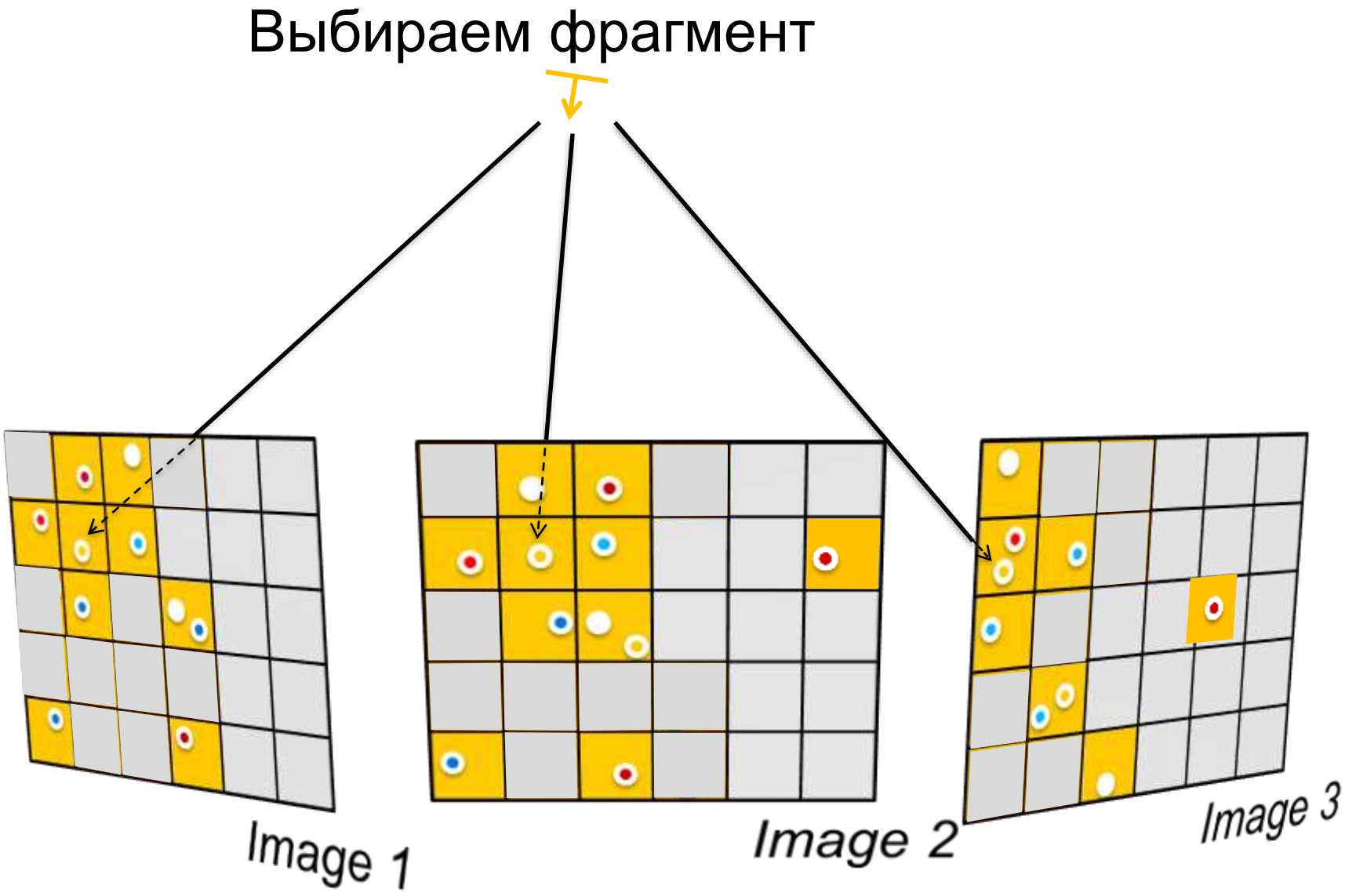


Input image #1 #2 #3

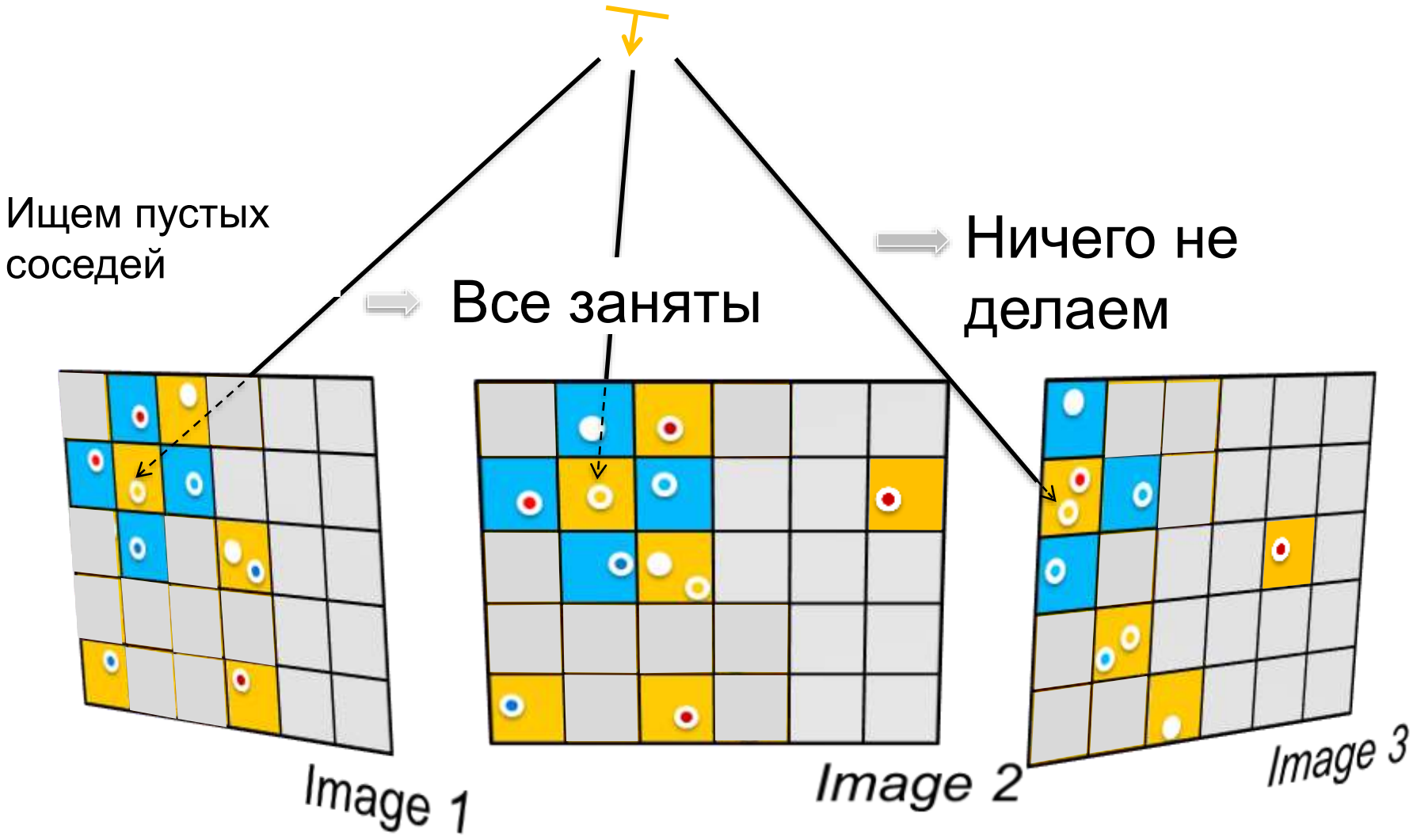
Расширение фрагментов



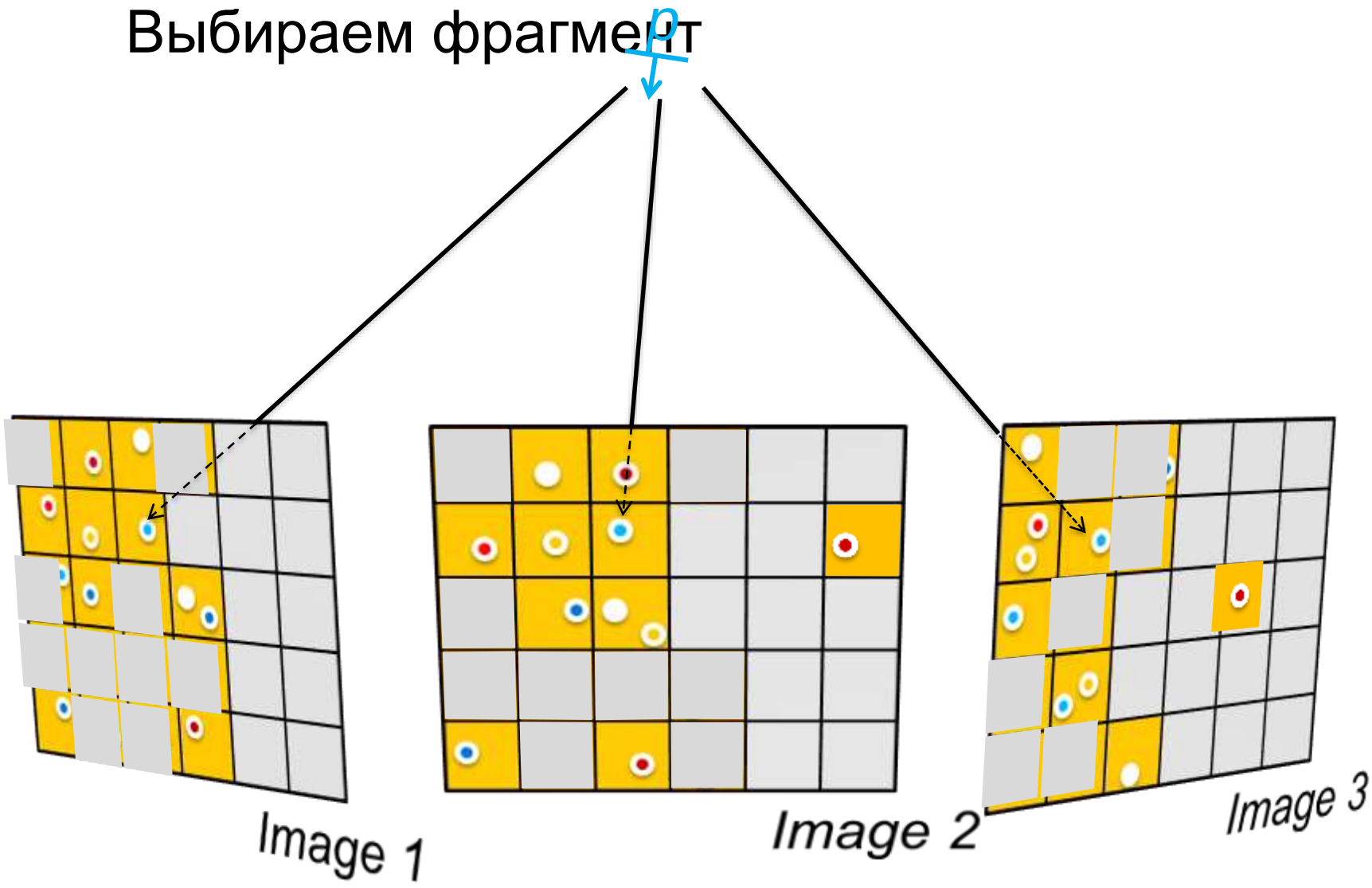
Расширение фрагментов



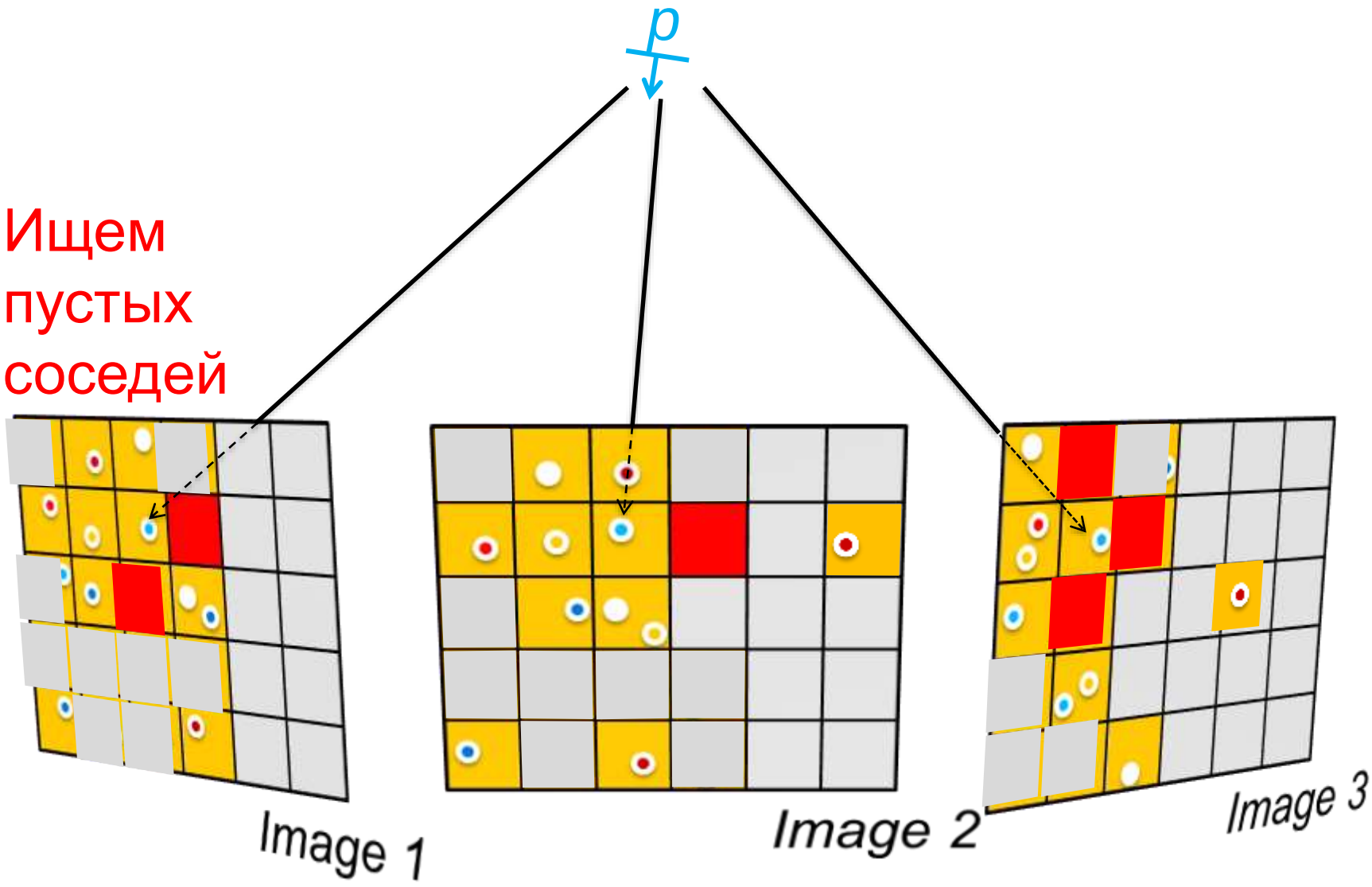
Расширение фрагментов



Расширение фрагментов

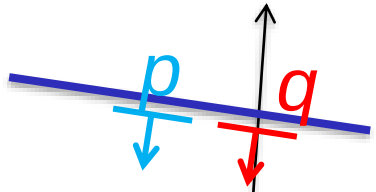


Расширение фрагментов



Расширение фрагментов

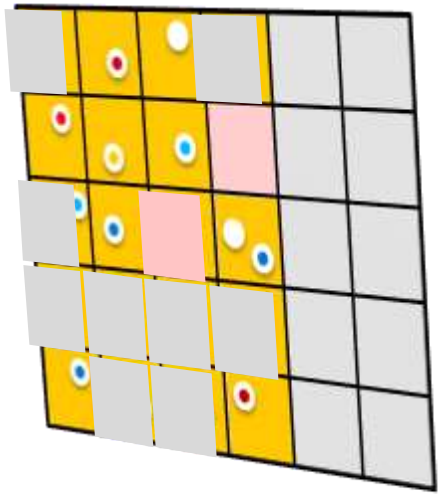
Реконструируем фрагменты во всех соседних пикселях

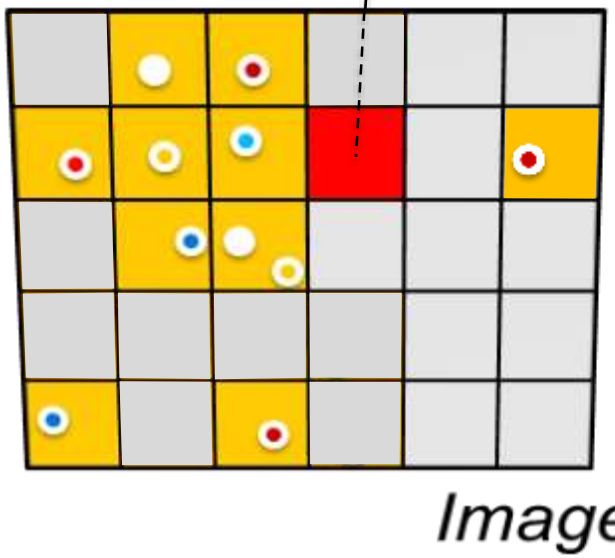


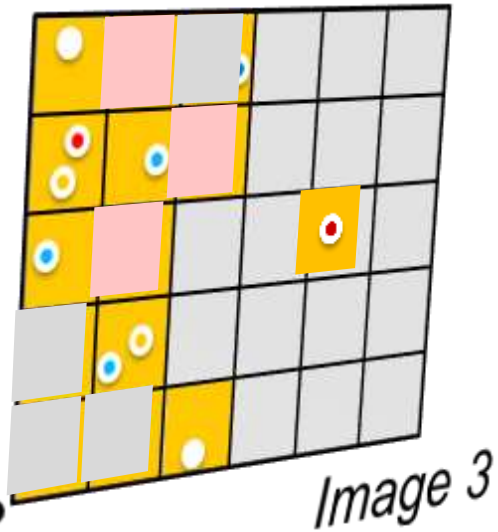
$c(q)$:
{пересечение касательной плоскости и луча}

$n(q)$:

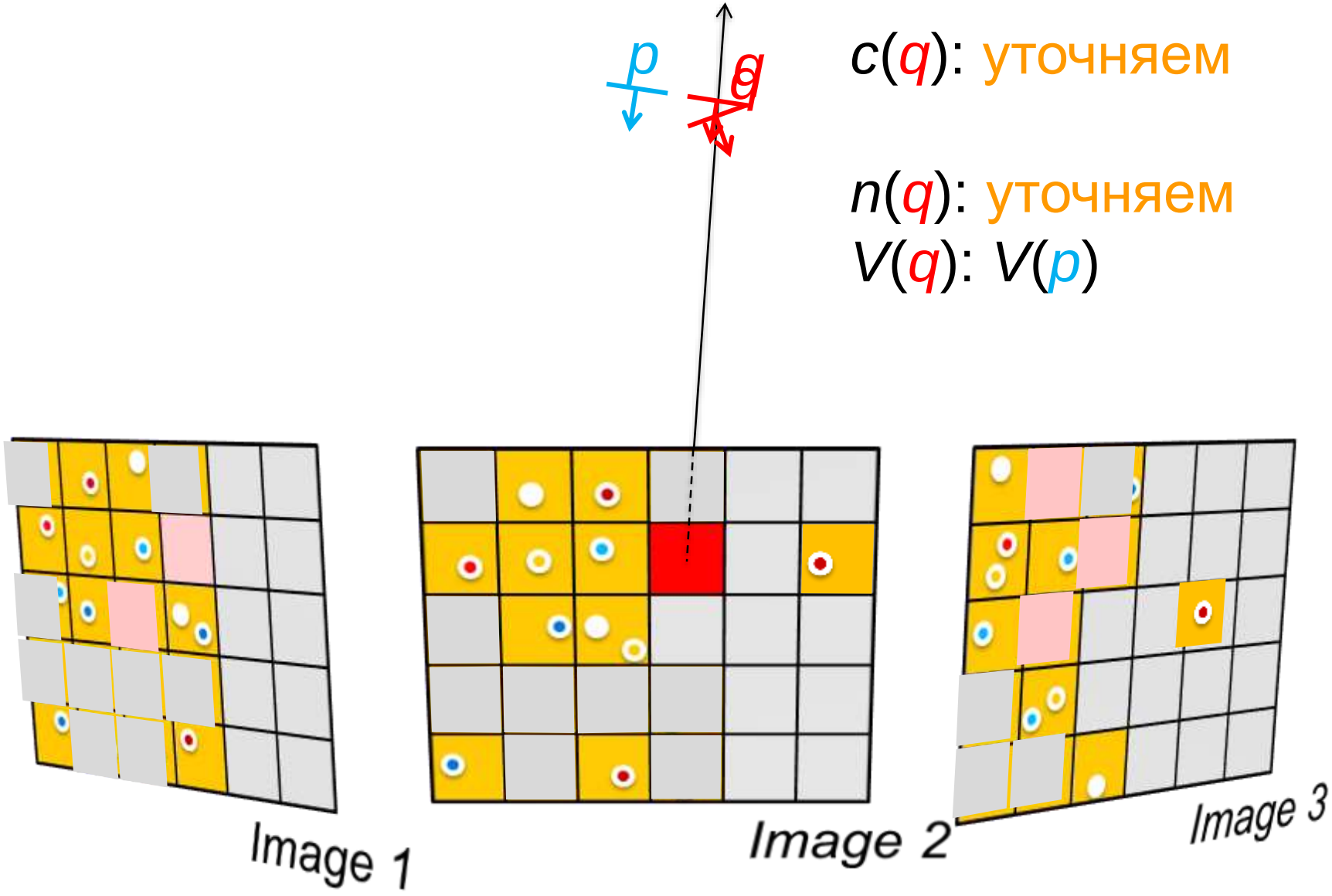
$V(q)$:



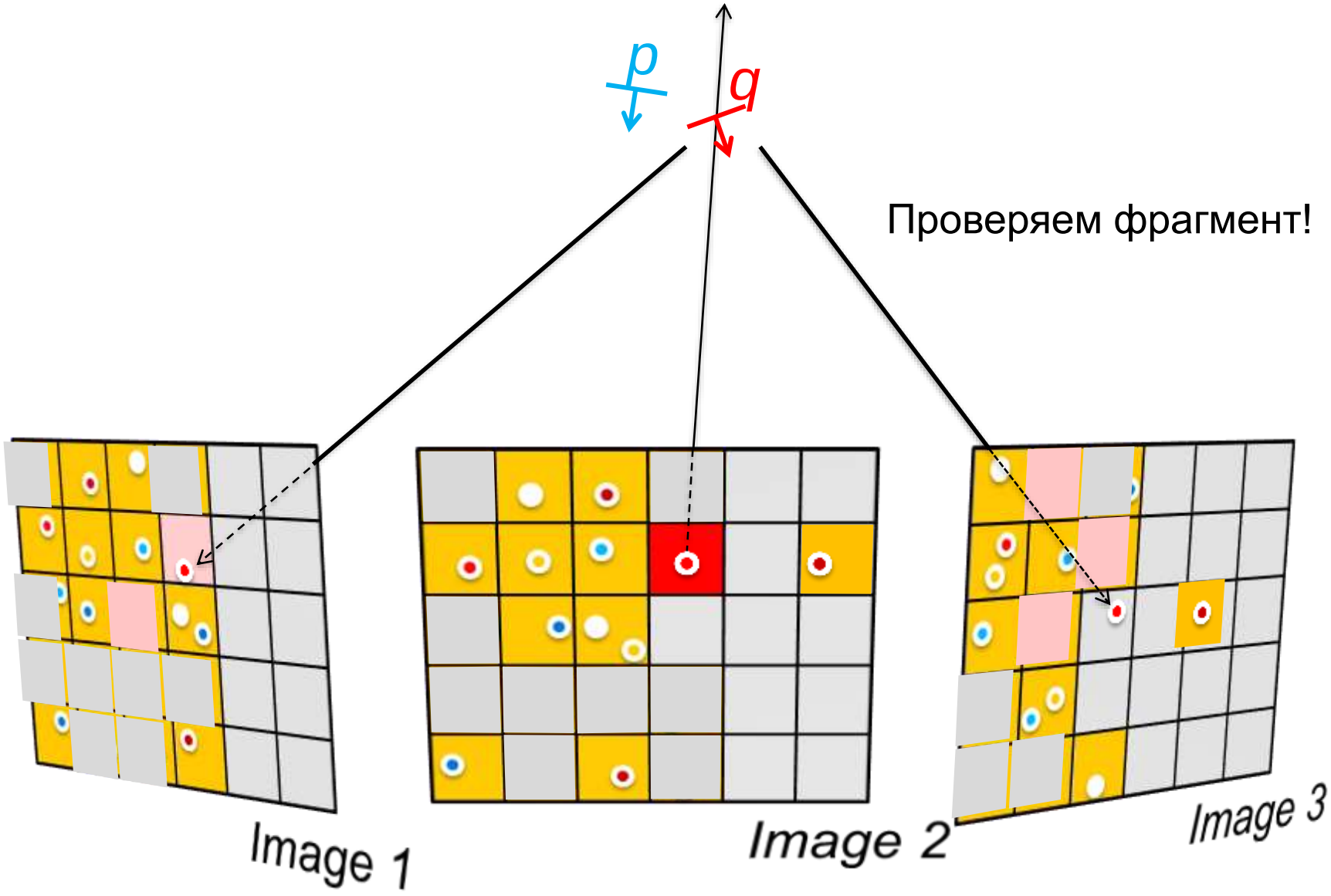




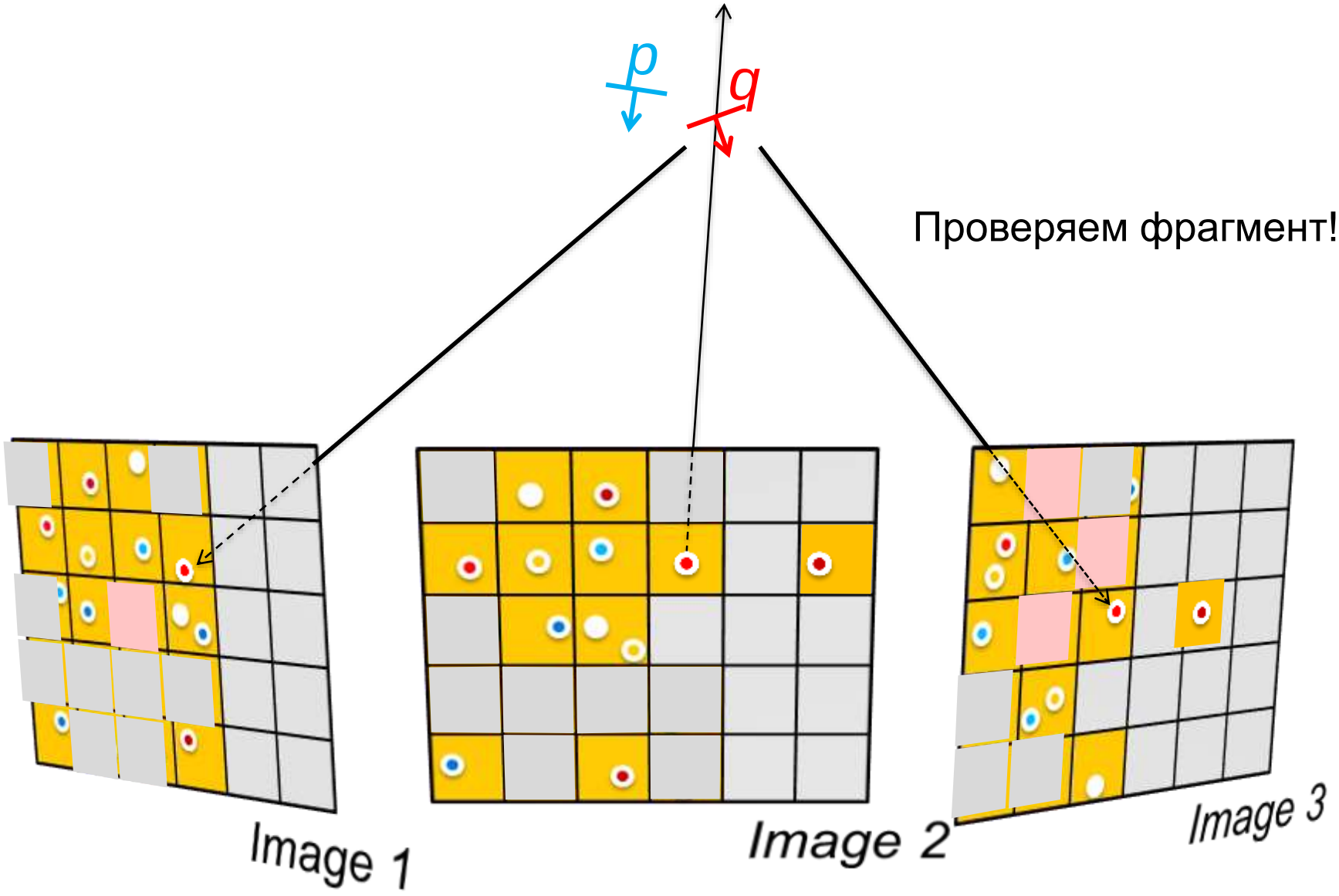
Расширение фрагментов



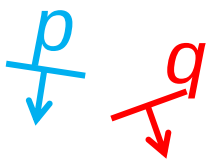
Расширение фрагментов



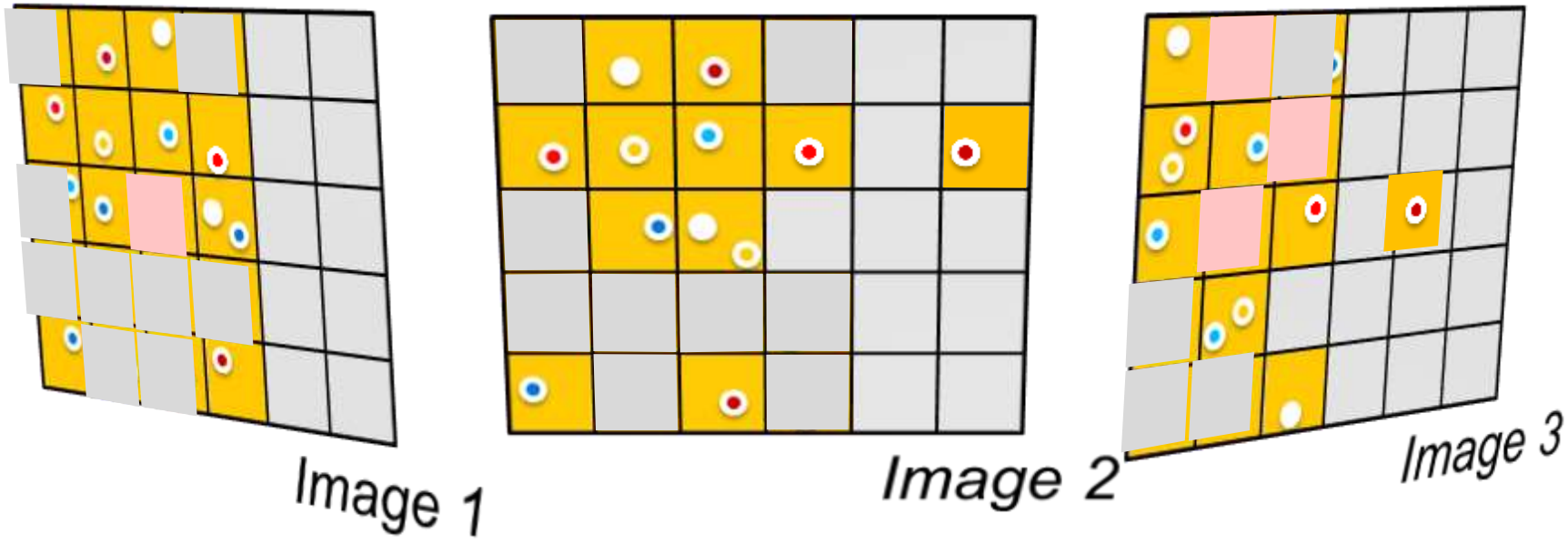
Расширение фрагментов



Расширение фрагментов



Повторяем
- Для каждого фрагмента
- Для каждого соседнего пустого пикселя

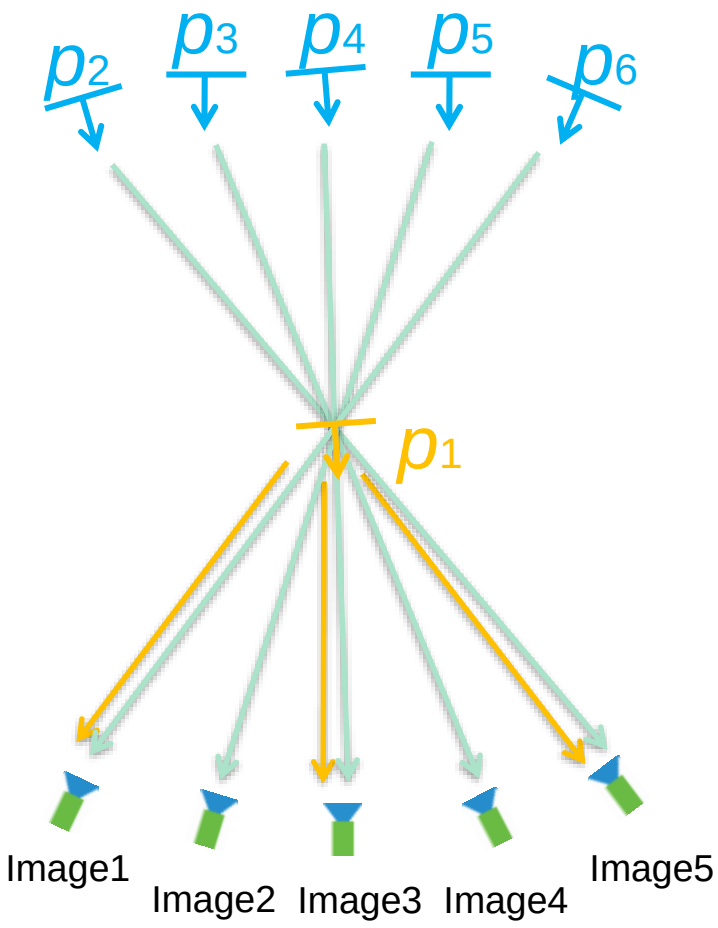


Фильтрация фрагментов

- Согласование видимости

Убираем p_1 если

$$|V(p_1)|\,N(p_1) < \sum_{i=2}^{6} N(p_i)$$

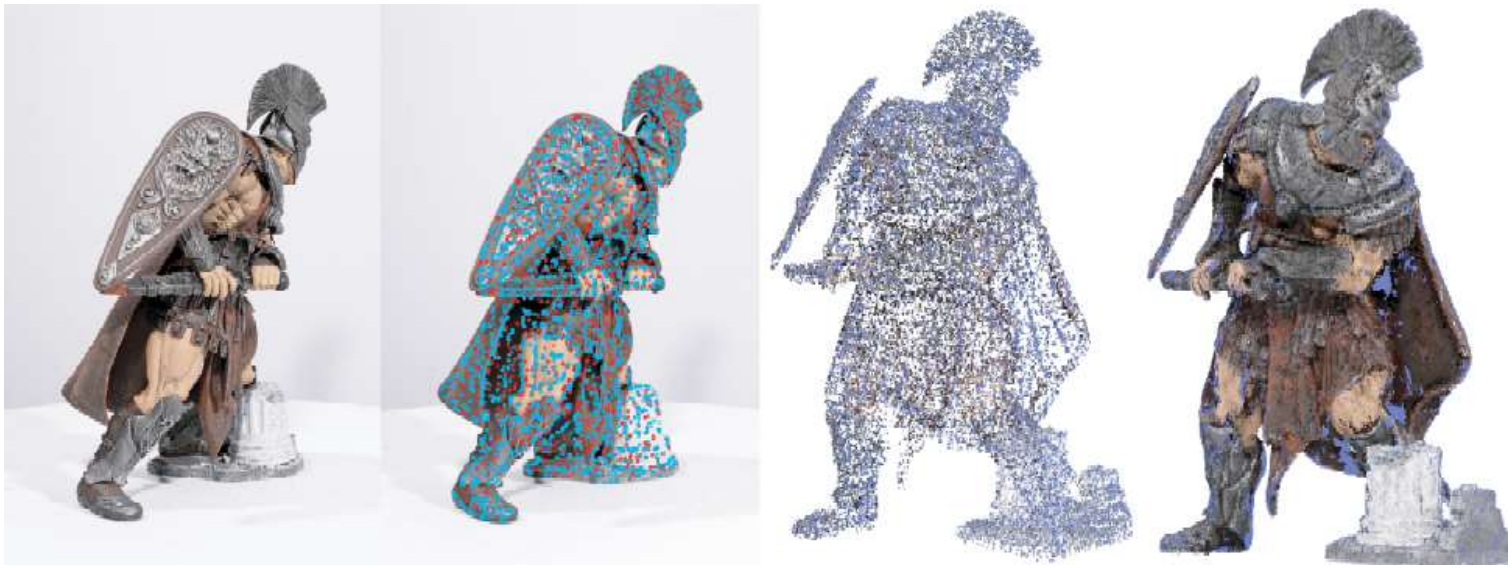


Алгоритм на основе фрагментов

#1. Поиск особых точек

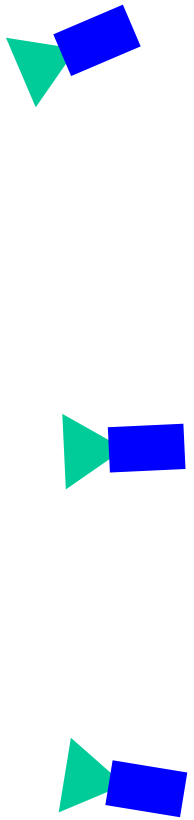
#2. Инициализация фрагментов

#3. Расширение фрагментов и фильтрация

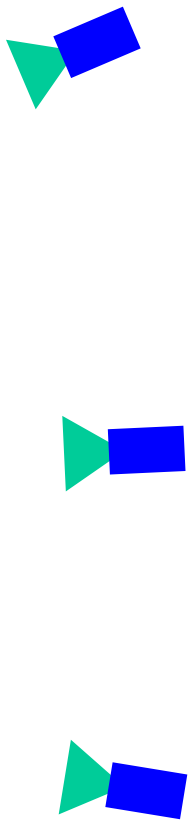


Input image #1 #2 #3

Резюме алгоритма

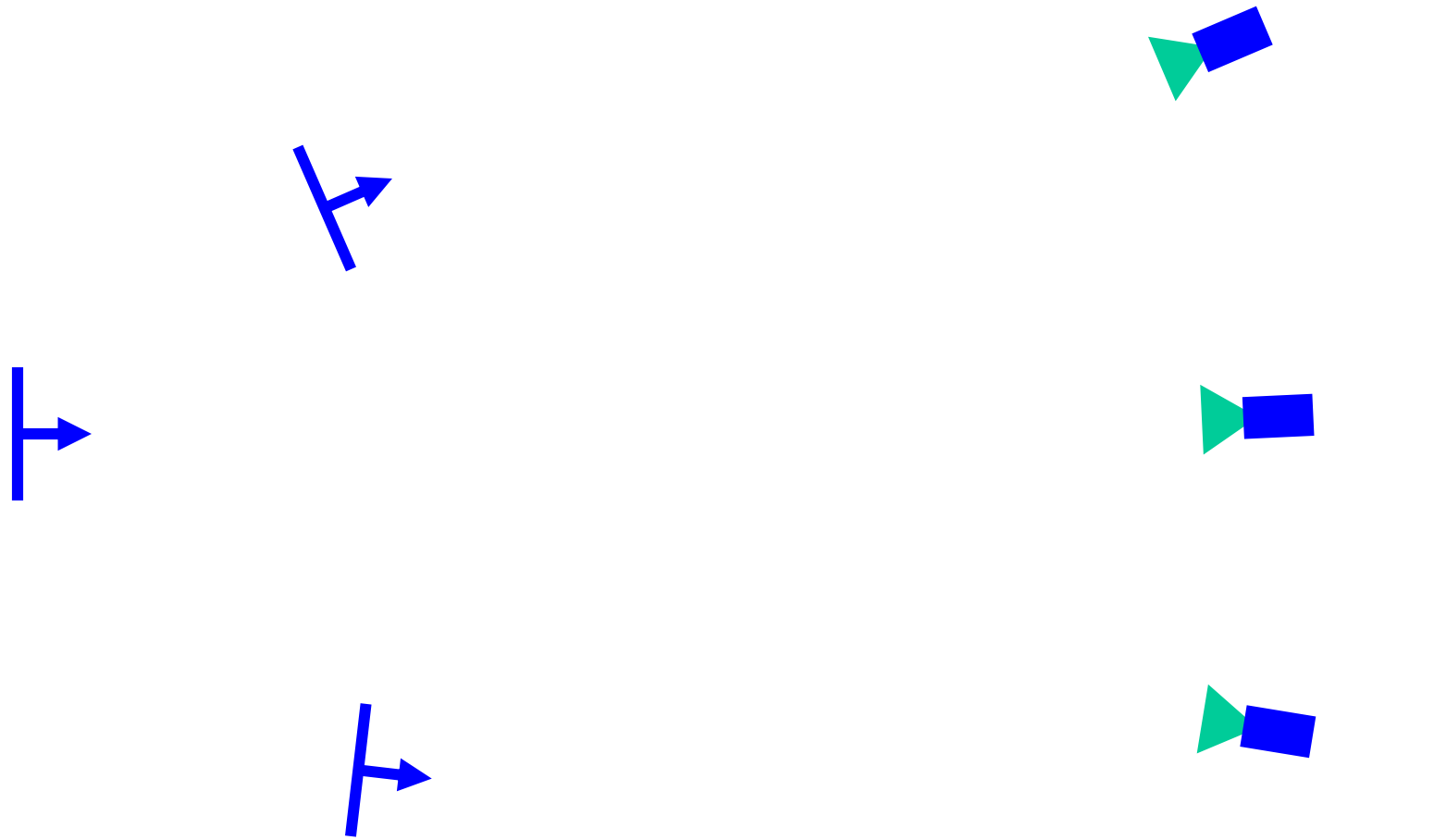


Резюме алгоритма



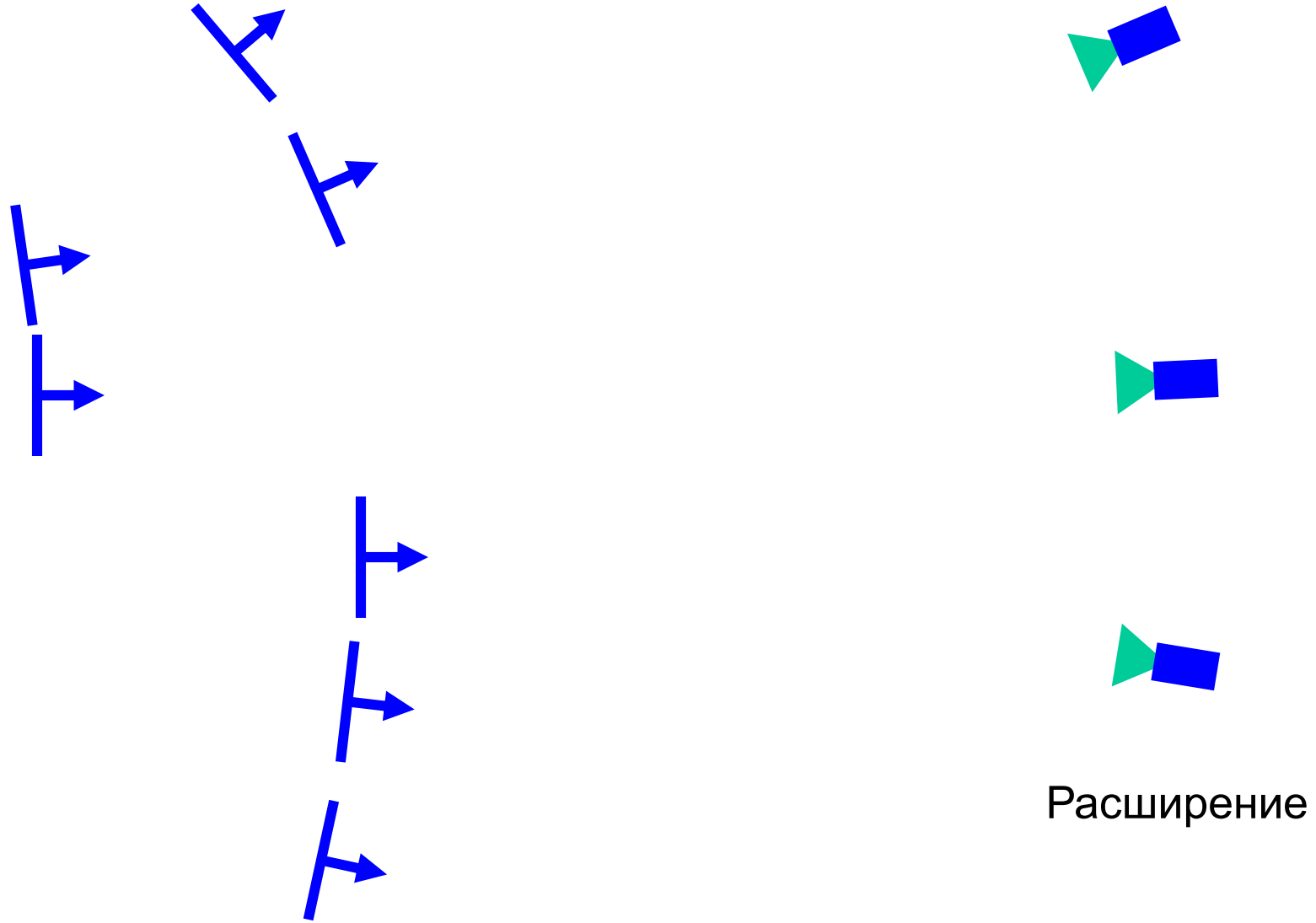
Поиск
особенностей

Резюме алгоритма



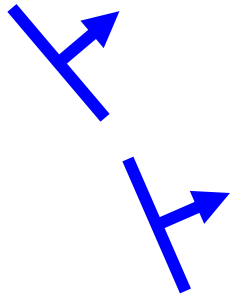
Инициализация
фрагментов

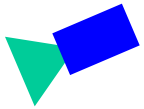
Резюме алгоритма

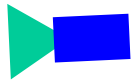


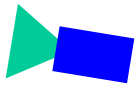
Расширение

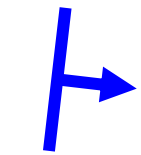
Резюме алгоритма

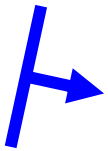






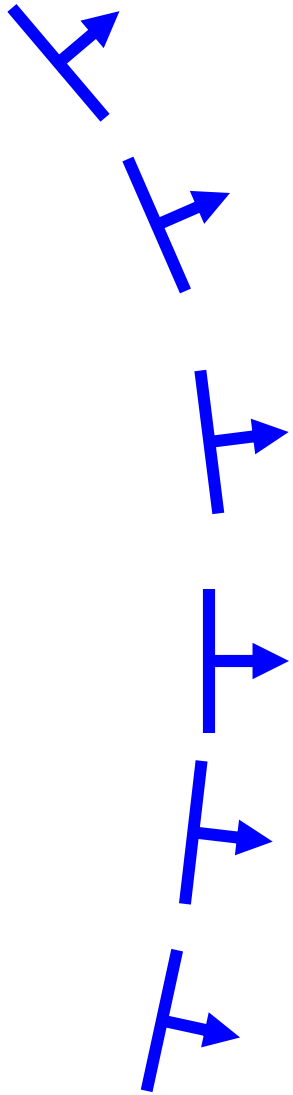


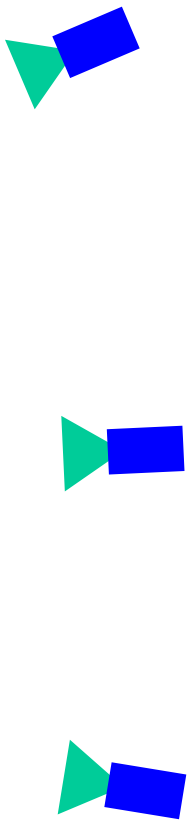




Фильтрация

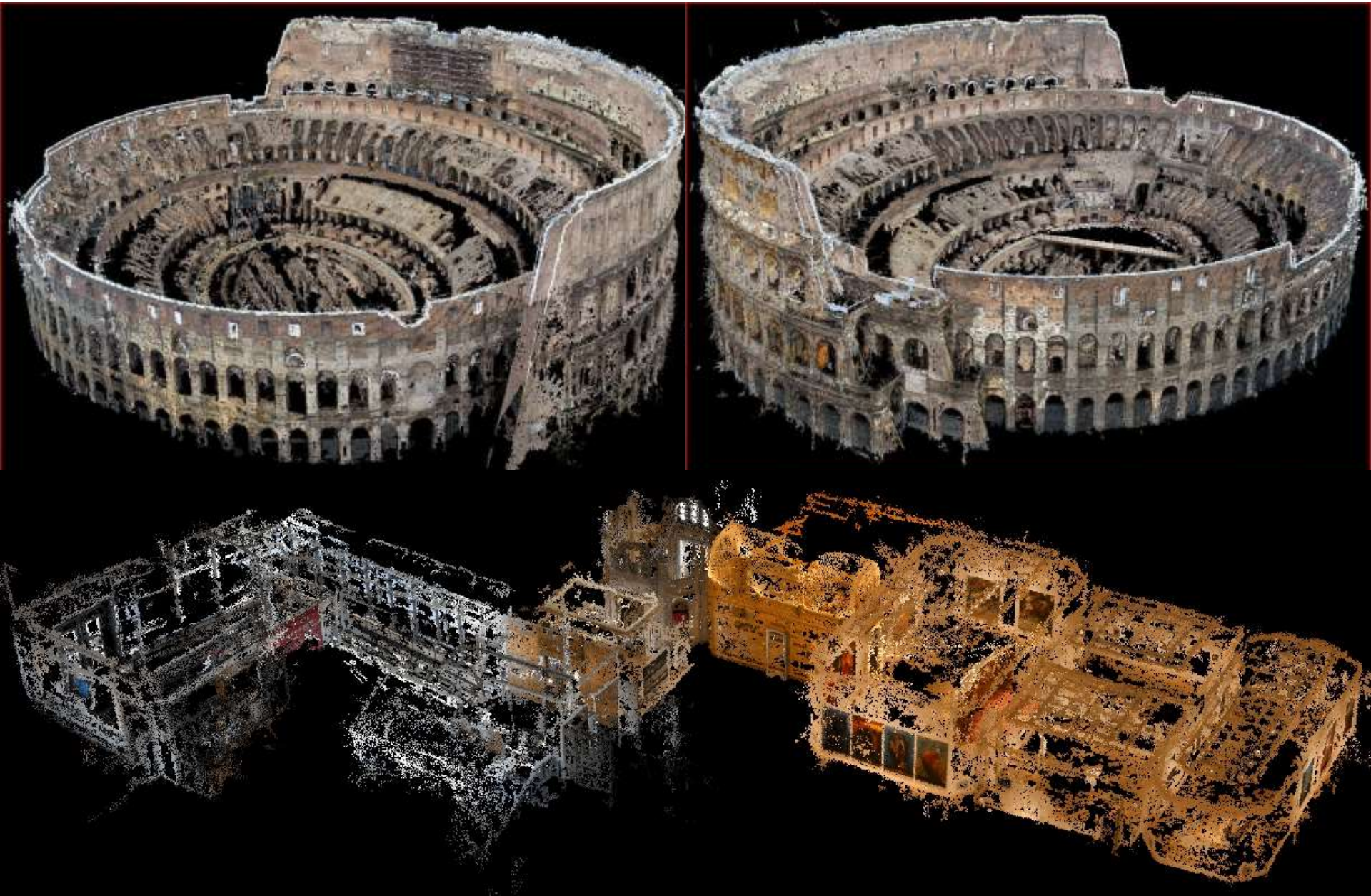
Резюме алгоритма





Расширение

Примеры работы



Оценка качества на ETH3D (2017)

Method	Indoor	Outdoor	Mobile	DSLR
CMPMVS	67.2 / 47.3 / 55.5	44.2 / 40.0 / 42.0	14.4 / 7.4 / 9.8	71.6 / **57.6** / 63.8
COLMAP	**90.2 / 51.1 / 65.2**	**80.9 / 53.1 / 64.1**	**69.5 / 41.2 / 51.8**	**91.7** / 56.2 / **69.7**
Gipuma	74.9 / 24.0 / 36.3	52.8 / 20.8 / 29.9	31.1 / 13.4 / 18.7	76.5 / 25.9 / 38.7
PMVS	85.1 / 28.0 / 42.1	72.2 / 27.8 / 40.1	48.7 / 18.8 / 27.2	90.1 / 31.3 / 46.5

Table 3. Category-based MVS evaluation showing accuracy / completeness / F_1 score (in %) at a 2cm evaluation threshold.

По прежнему один из самых точных, но не лучший по «полноте» метод MVS

Масштабирование

Кластеризация видов и объединение реконструкций

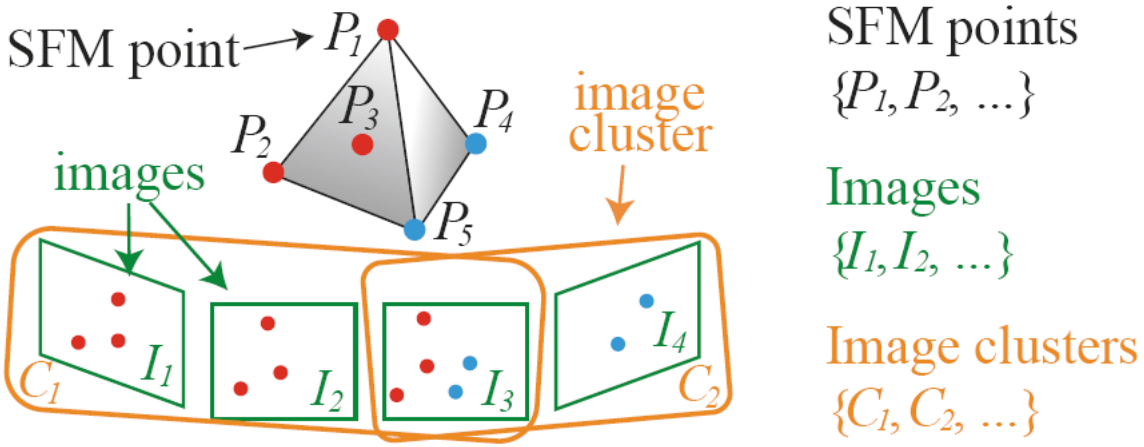


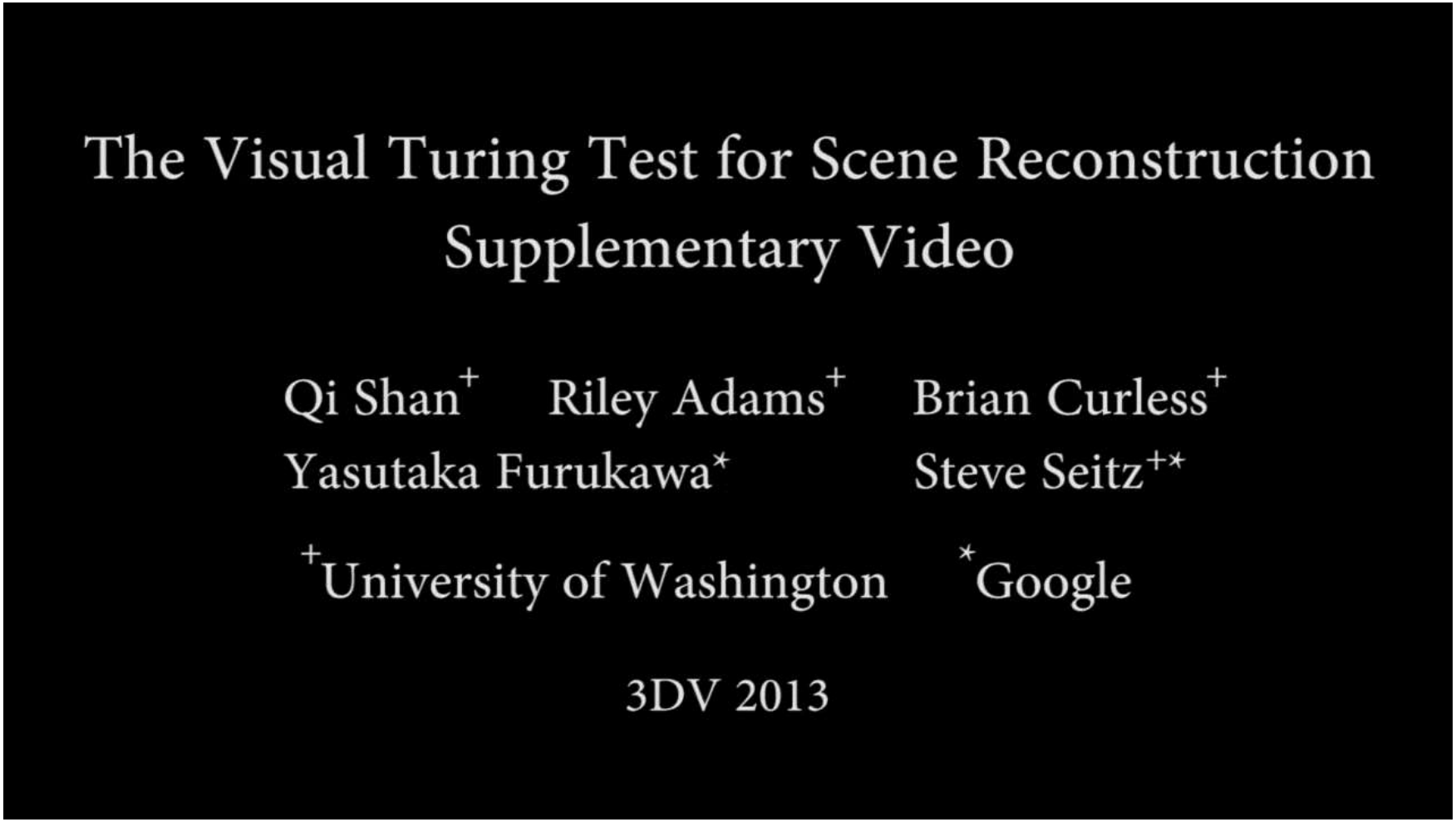
Figure 2. Our view clustering algorithm takes images $\{I_i\}$, SFM points $\{P_j\}$, and their associated visibility information $\{V_j\}$, then produces overlapping image clusters $\{C_k\}$.

Towards Internet-scale Multi-view Stereo. Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. CVPR 2010

Визуальный тест Тьюринга

The Visual Turing Test for Scene Reconstruction. Qi Shan, Riley Adams, Brian Curless, Yasutaka Furukawa, and Steven M. Seitz. 3D Vision 2013, June 2013 (Best Paper Award).

Видео


The Visual Turing Test for Scene Reconstruction
Supplementary Video
Qi Shan[+] Riley Adams[+] Brian Curless[+]
Yasutaka Furukawa[*] Steve Seitz[+*]
[+]University of Washington [*]Google
3DV 2013

Учёт контуров

До

После

Qi Shan and Brian Curless and Yasutaka Furukawa and Carlos Hernandez and Steven M. Seitz **Occluding Contours for Multi-View Stereo**. CVPR 2014

Учёт контуров и дополнение реконструкции

Дополнение реконструкции точками

COLMAP

COLMAP
3.6

Search docs

- Installation
- Tutorial
- Database Format
- Camera Models
- Output Format
- Datasets
- Graphical User Interface
- Command-line Interface
- Frequently Asked Questions
- Changelog
- Contribution
- License
- Bibliography

Docs » COLMAP View page source

COLMAP

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

https://colmap.github.io/

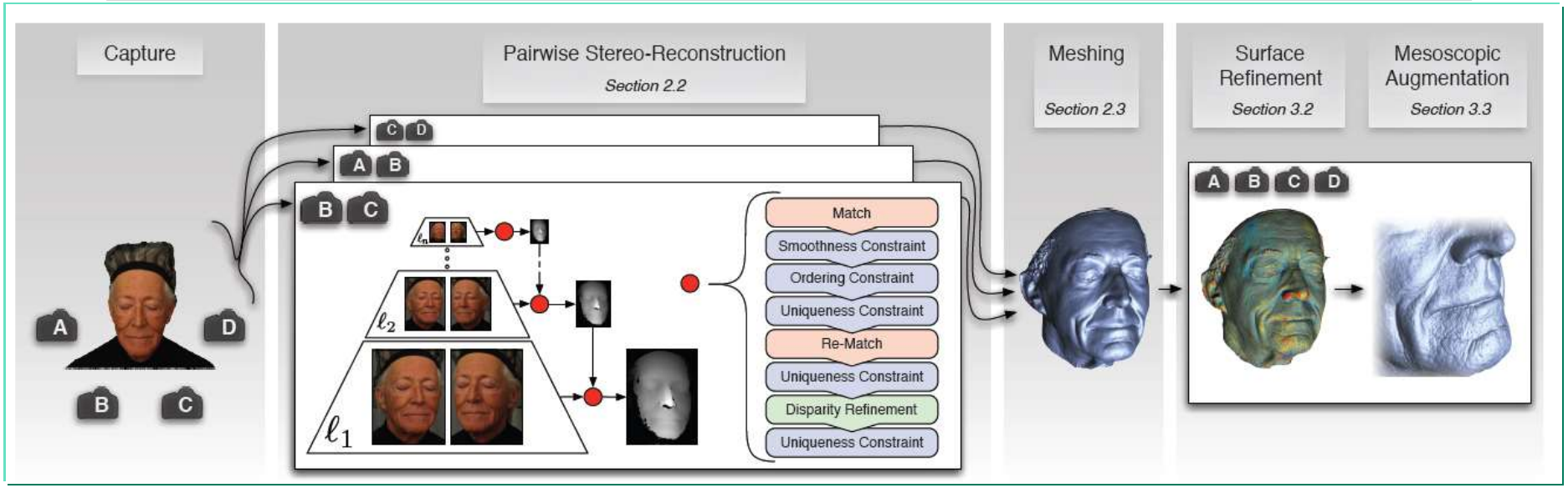
Немного об отдельном use case – реконструкции головы человека

Моментальное 3Д

- Построение высокодетализированной 3Д модели лица человека
- От одной стереокамеры до нескольких стереокамер

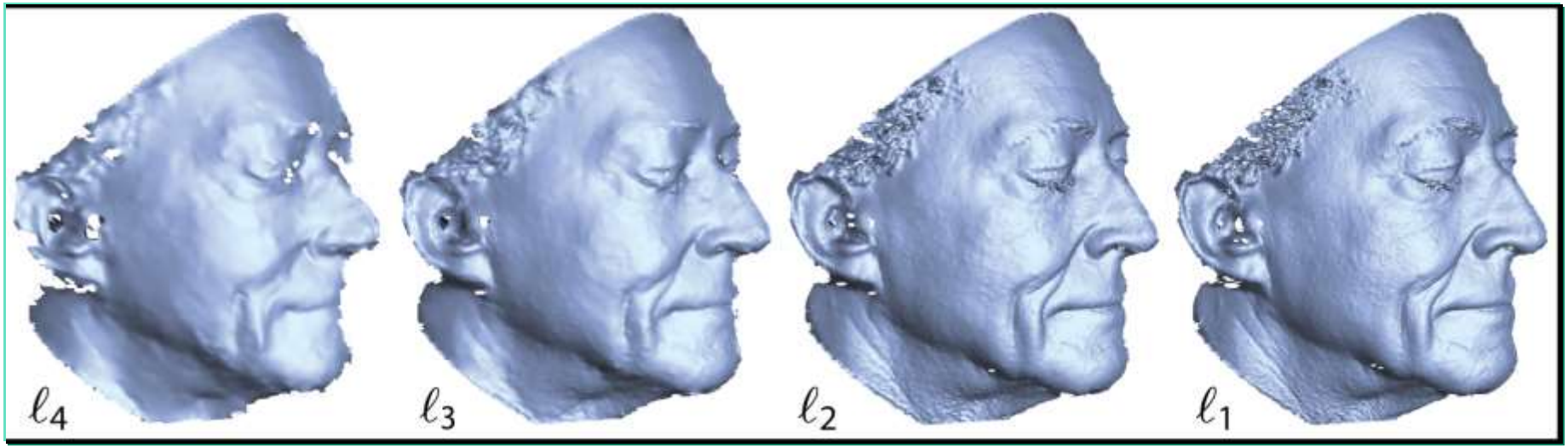
T. Beeler, B. Bickel, P. Beardsley, R. Sumner, M. Gross High-Quality Single-Shot Capture of Facial Geometry, SIGGRAPH 2010

Схема системы



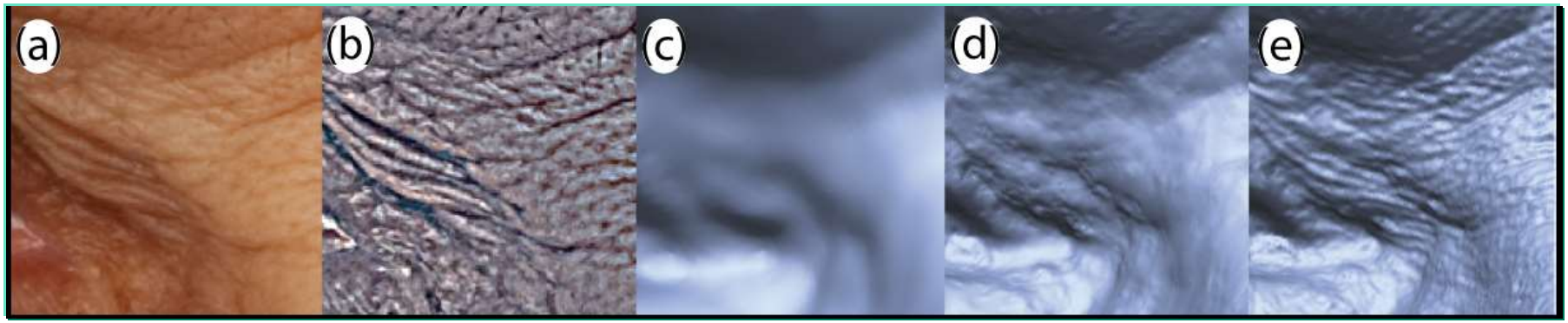
- Иерархическое бинокулярное стерео с итеративным уточнением
  - От разрешения 160*160 до 1280*1280
  - Локальные методы - без использования глобальной оптимизации, сегментации и т.д.

Результат стерео



- Последовательное уточнение геометрии модели
- Субпиксельное уточнение диспаритета

Mesoscopic Augmentation



- Вычисляем высокочастотную составляющую изображения
- Высокие частоты соответствуют мелким деталям на коже человека
- «Искажаем» геометрию согласно специальной формуле

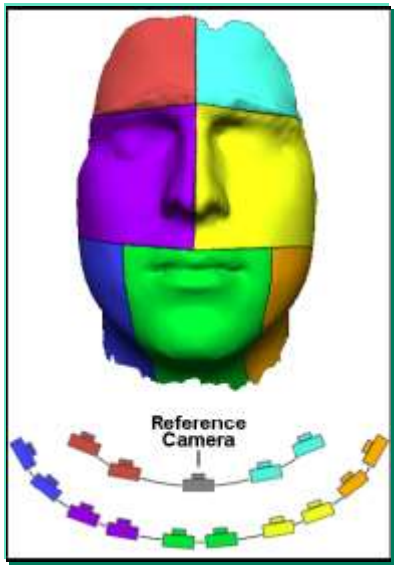
3Д захват лица

- Пассивная многокамерная система (14 камер)

D. Bradley, W. Heidrich, T. Popa, A. Sheffer **High Resolution Passive Facial Performance Capture** , SIGGRAPH 2010

Устройство системы

- 7 стереопар высокого разрешения Sony HDR-SR7
- Система подсветки на основе LED
- Каждая стереопара видит свою зону на лице

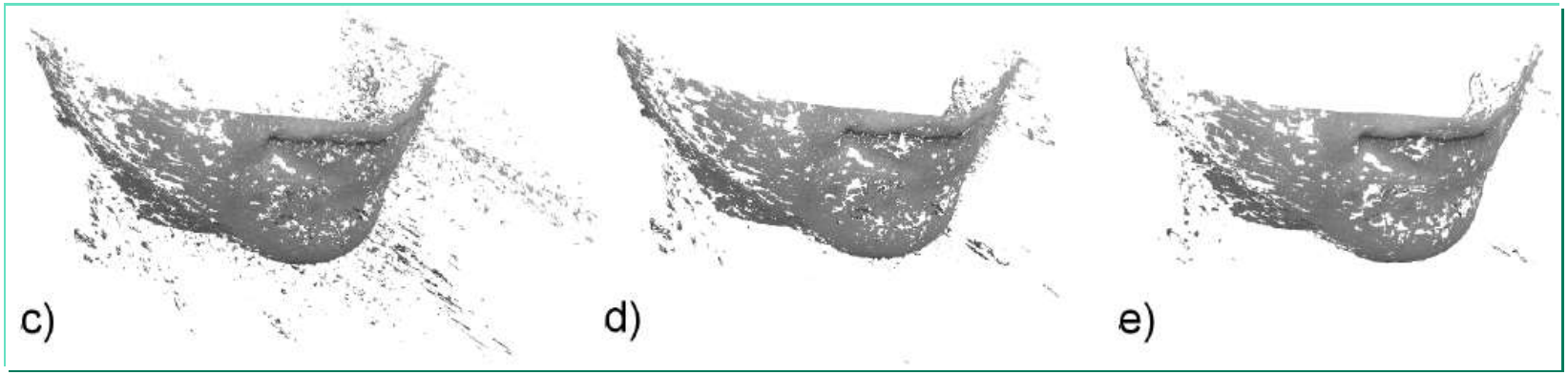


Разрешение изображений

- При таком высоком разрешении и равномерной подсветке поры на коже дают достаточно текстуры для вычисления пассивного стерео

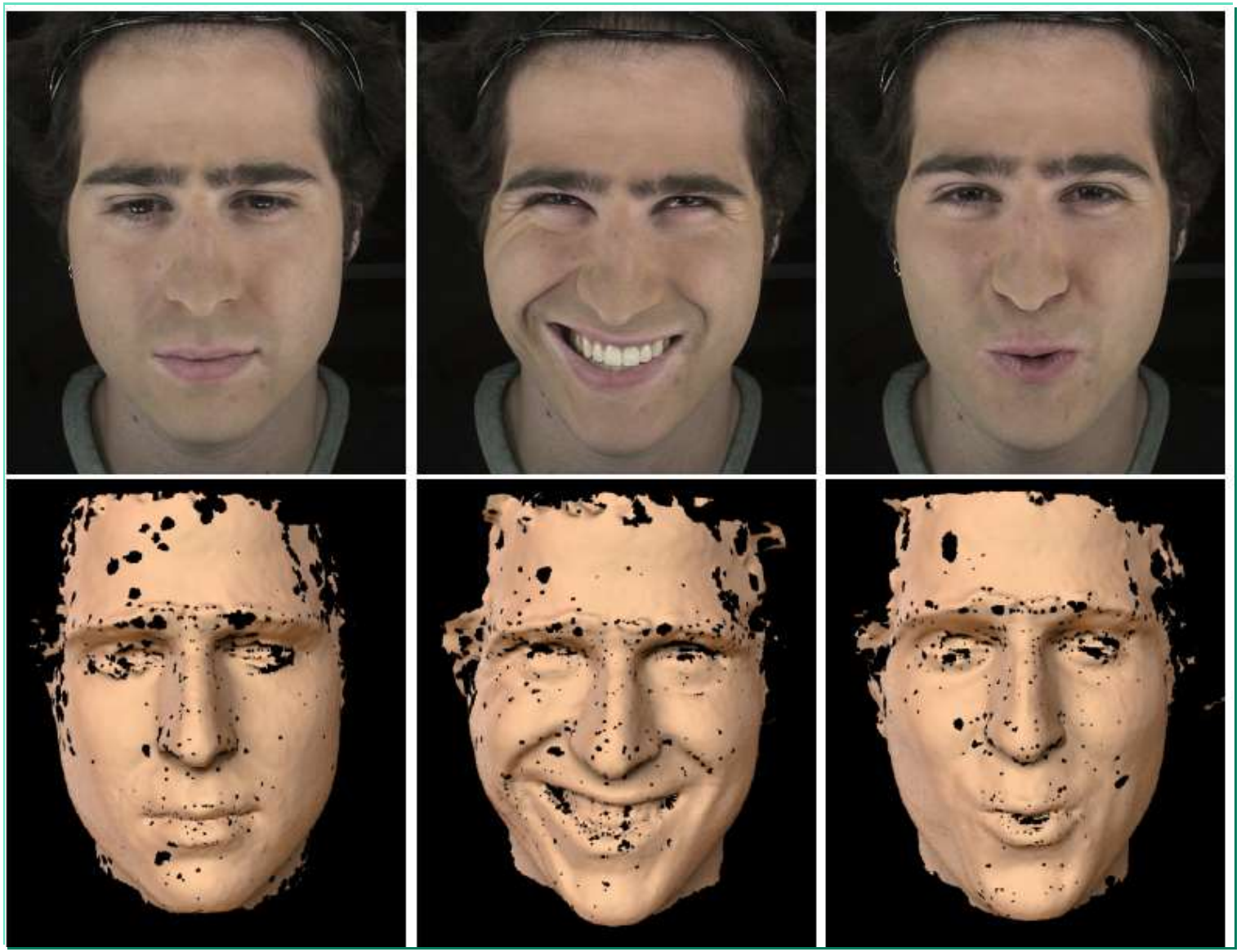
BRADLEY, D., BOUBEKEUR, T., AND HEIDRICH, W. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. CVPR. 2008.

Пассивное стерео и обработка



- Каждая стереопара даёт карту глубины и облако точек (шумное)
- Итеративное сглаживание и вычисление порогов для отброса шумов

Результат по 1 кадру



Отслеживание

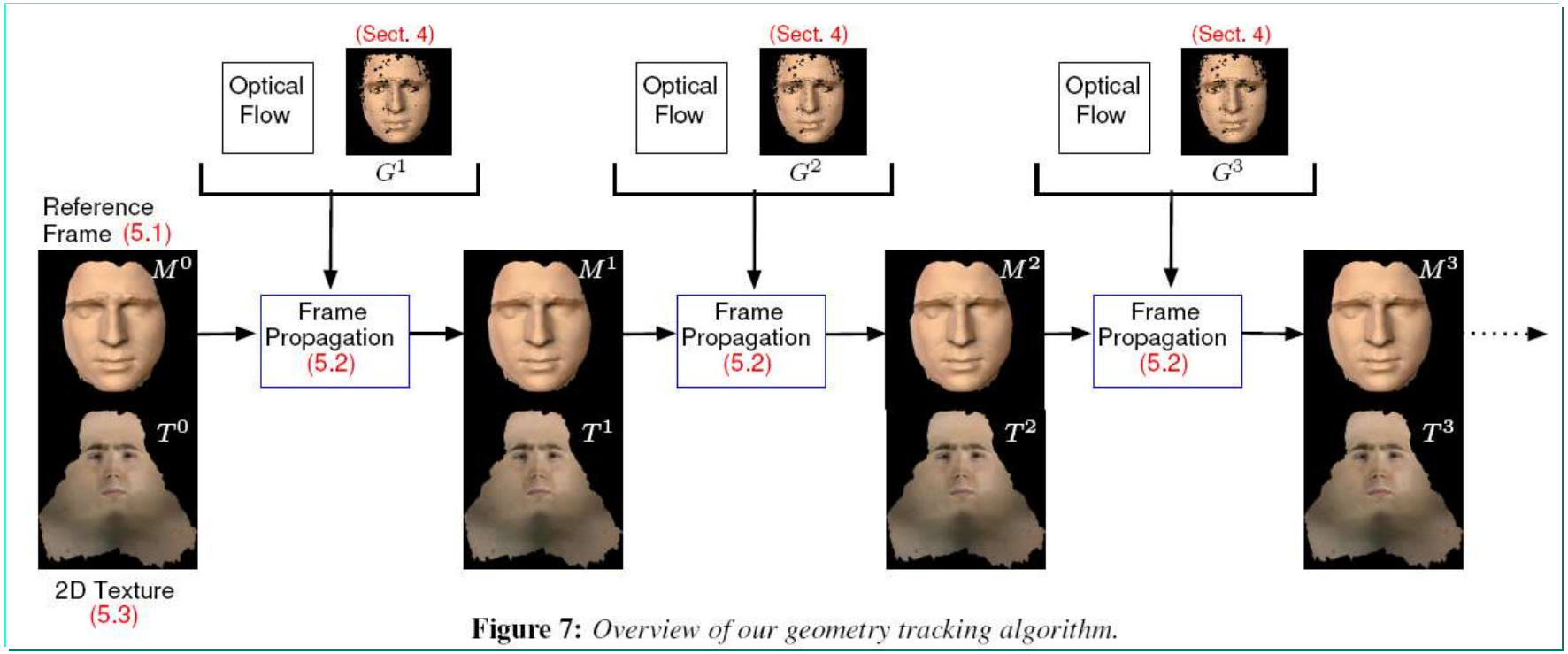
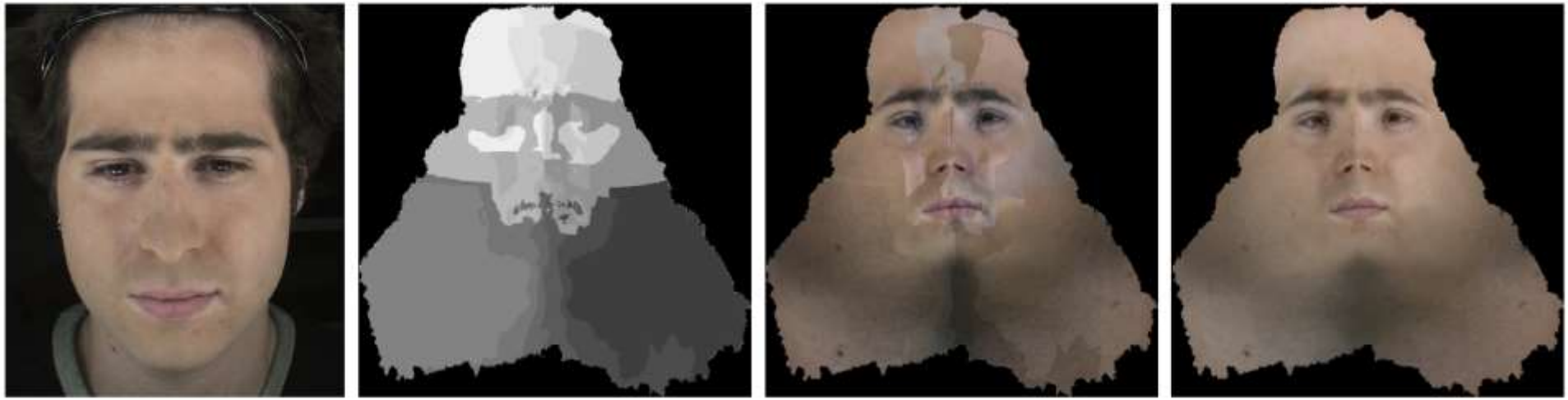


Figure 7: *Overview of our geometry tracking algorithm.*

- Первое изображение выбирается как опорная модель
- Нужно построить такую деформацию опорной модели, чтобы она совпадала с моделями по следующим кадрам
- Для направления используется оптический поток

Синтез текстуры

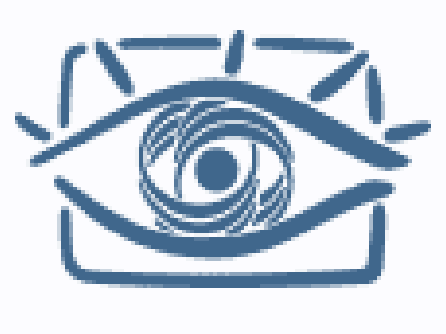


- Участки текстуры копируются со своих камер
- Поскольку камеры без цветовой калибровки, при прямом копировании текстуры видны резкие цветовые переходы (артефакты)
- Для обработки используется композиция по Пуассону

Пример результатов

Резюме

- Все методы многовидового пассивного стерео опираются на принцип фотосогласования
- Два классических подхода:
  - Объединение сеток
  - Распространение фрагментов
- Поверх можно накрутить пост-обработку
- На визуальное качество сильно влияет рендеринг и качество материалов



Лаборатория компьютерной графики и мультимедиа ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

Лекция №11
«Разреженная 3-х мерная реконструкция»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа ВМК МГУ

2 декабря 2019 года

3D реконструкция по изображениям

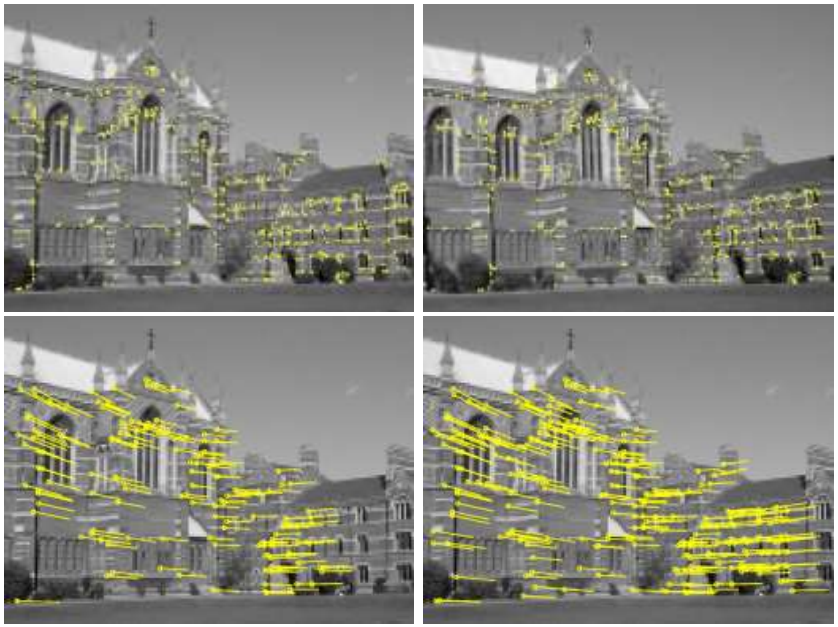
Набор изображений

Трёхмерная (3D) модель

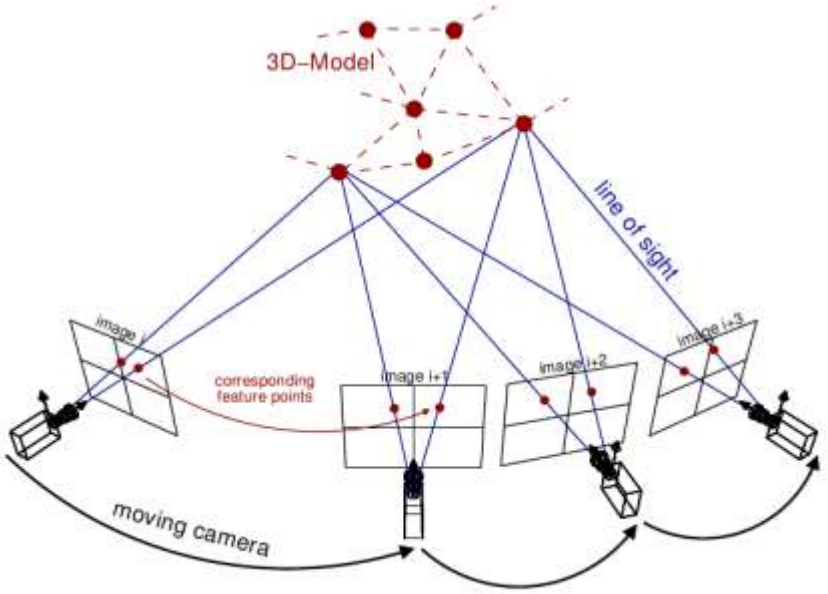
Пример задачи: построение 3D модели реального объекта по набору фотографий из интернета

Source: Agarwal et. al.Building Rome in a Day. ICCV 2009

Разреженная реконструкция



Point features and matching

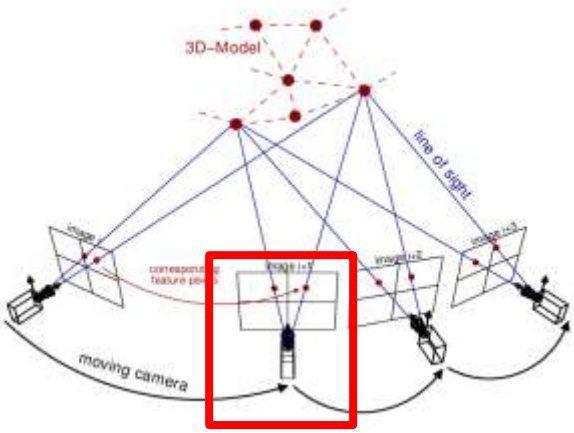


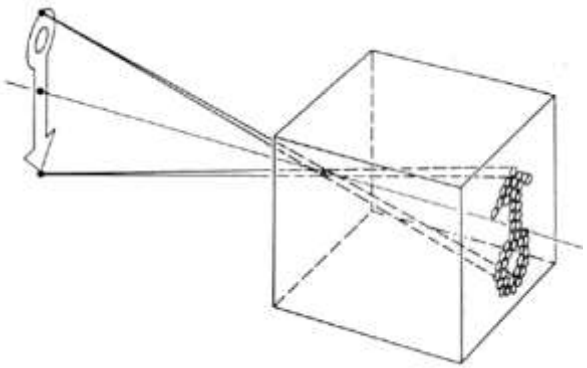
Camera localization and structure estimation

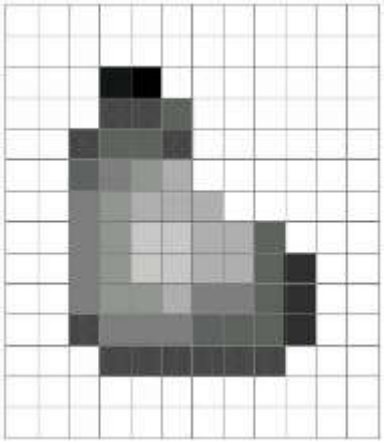
- Опираемся на сопоставление ключевых точек между изображениями
- 3D реконструкция в форме облака точек, соответствующих найденным ключевым точкам на изображениях
- Решаем совместно определение 3Д точек, соответствующих ключевым, и положение камер в пространстве относительно них
- 3D реконструкция – обратная задача к задаче компьютерной графики

Модель формирования изображения

- Начнём с модели 1 камеры





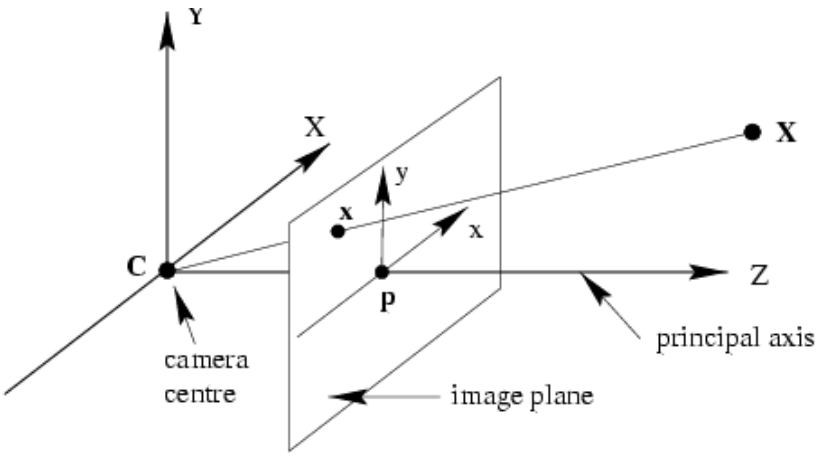


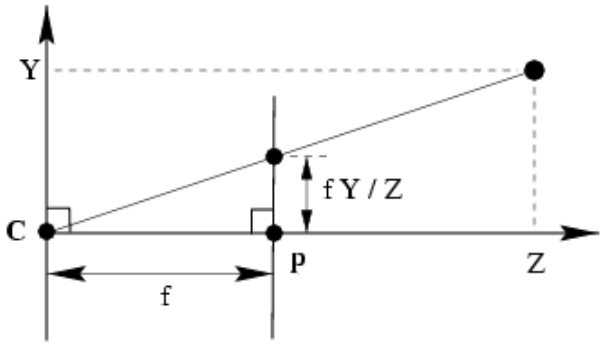
- Преобразование из мировой (общей) системы координат в камеру
- Определяется положением камеры в пространстве и ориентацией

- Перспективная проекция
- Важно фокусное расстояние

- Дискретное изображение на матрице

Перспективная проекция





- **C** = (0,0,0) – центр камеры (проекции)
- **X**=**(**X,Y,Z) – точка в 3х мерном пространстве
- **x**=(x,y,f) – проекция **X** на картинную плоскость

C, **x**, **X** лежат на одной прямой

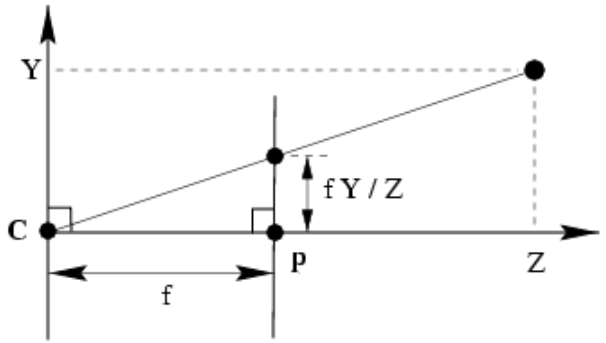
Тогда простейшее уравнение перспективной проекции:

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z}$$

Перспективная проекция – нелинейное преобразование!

Матрица проекции и внутренняя калибровка

Перейдём к однородным координатам и запишем всё в матричном виде



$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$P = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{[I\,|\,0]}$$

Уравнение перспективной проекции

В матричном виде x=PX , где **X**- точка в 3D, x **–** точка на цифровом изображении (в пикселях), **P –** матрица проекции

$$P = K \cdot [I \mid 0] \cdot C^{-1}$$

Матрица проекции камеры

Внутренняя калибровка

Центральная проекция

Внешняя калибровка

$$K = \begin{bmatrix} a_x & s & c_x \\ 0 & a_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f/pix & s & c_x \\ 0 & f/pix & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Внутренняя калибровка отображает точки с идеальной картинной плоскости в пиксели

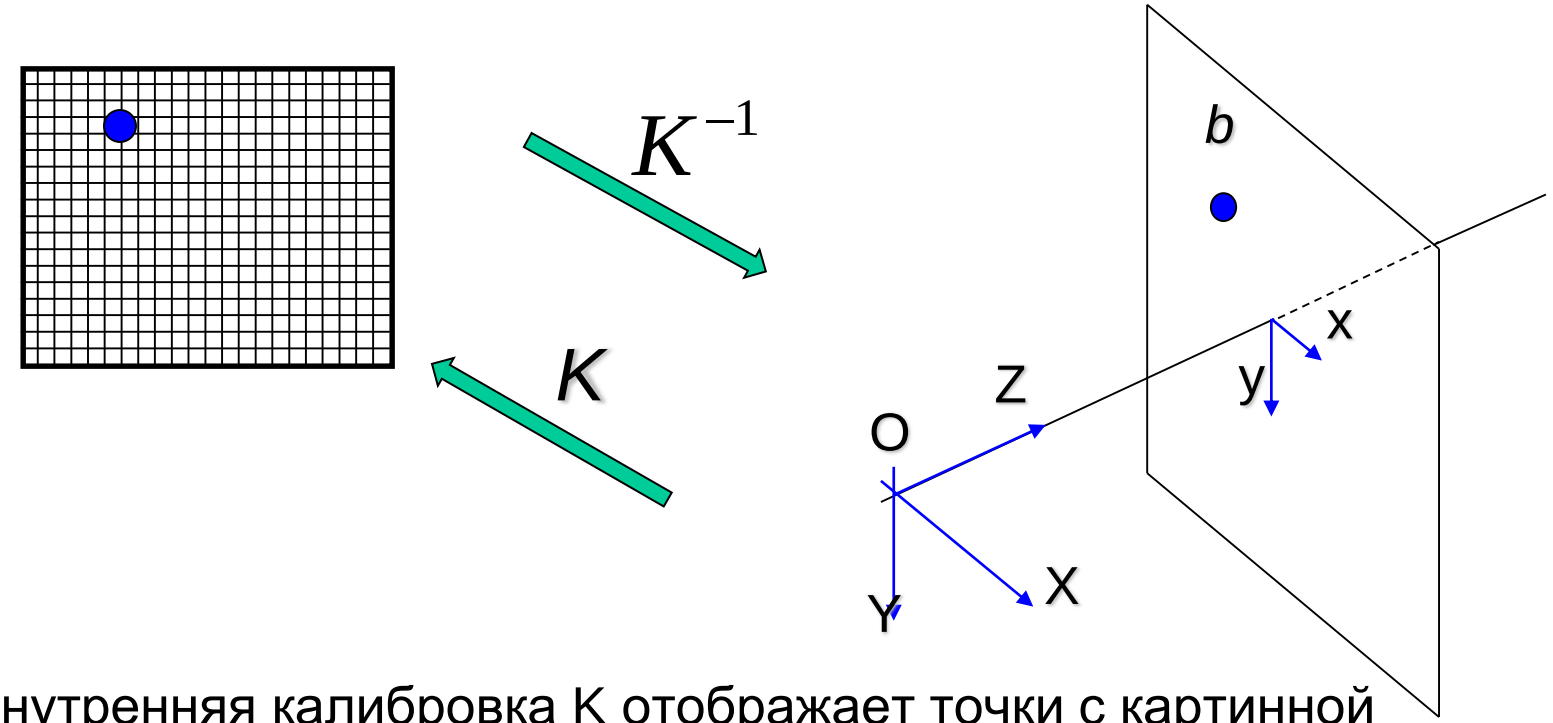
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Проекция на идеальную картинную плоскость

$$C^{-1} = \left(\begin{bmatrix} R & T \\ [0,0,0] & 1 \end{bmatrix} \right)^{-1} = \begin{bmatrix} R^T & -R^T T \\ [0,0,0] & 1 \end{bmatrix}$$

Матрица внешней калибровки определяется положением и ориентацией камеры в мировой системе координат

Полезно о внутренней калибровке



- Внутренняя калибровка K отображает точки с картинной плоскости (плоскости на расстоянии 1, в системе координат камеры) на изображение (в пиксели)
- Обратная матрица K^{-1} отображает точки изображения (в пикселях) на картинную плоскость (в систему координат камеры, с f=1)
  - Координаты в пространстве с центром в C будут (x,y,1,1)
  - Отображение точек на изображении в лучи

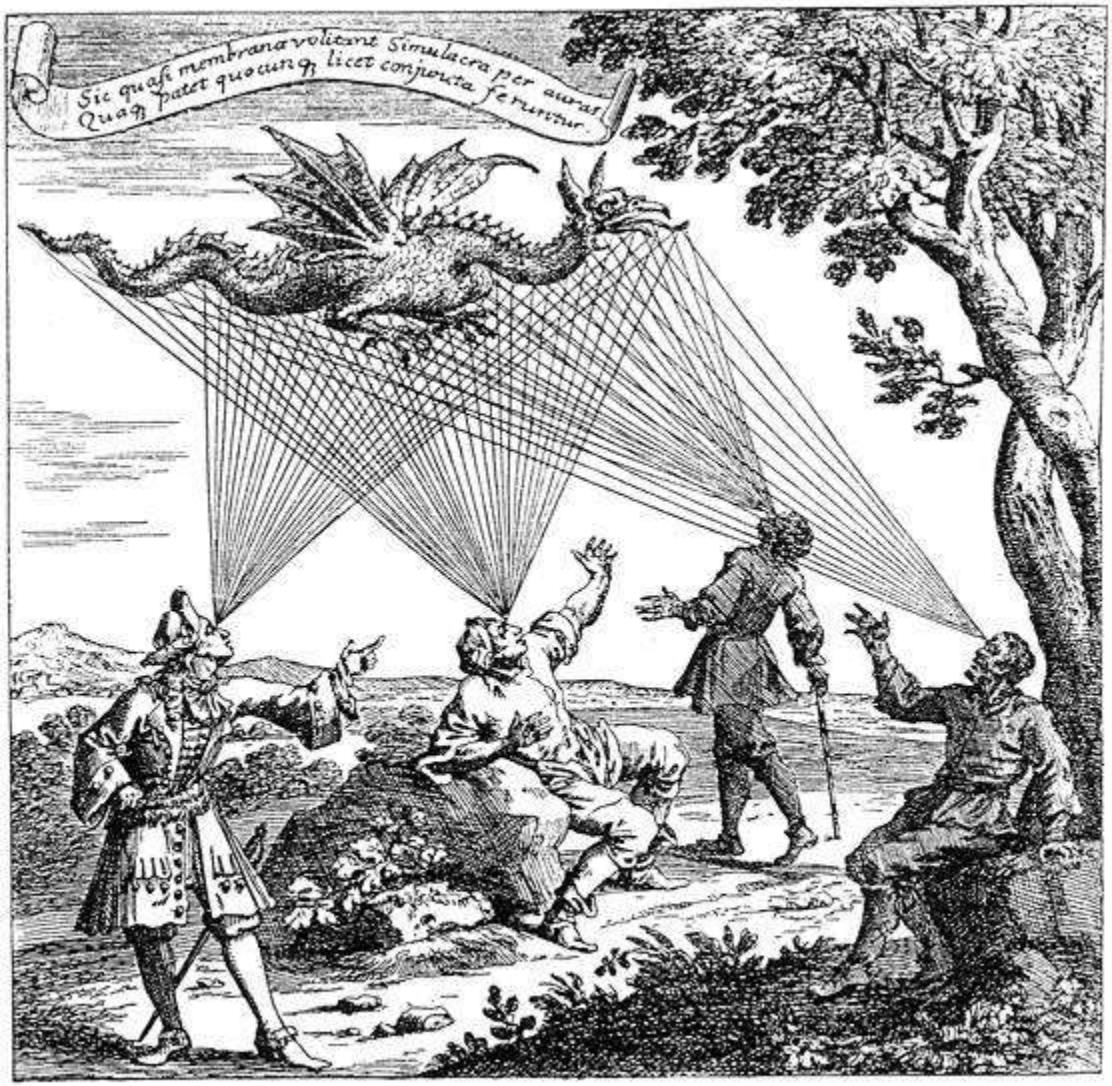
Радиальная дисторсия

- Прямые линии по краям изображения превращаются в кривые
- Точки смещаются вдоль радиуса от принципиальной точки
- Модель дисторсии Tsai:

$$\hat{x} = x + L(r)x \qquad r^2 = x^2 + y^2$$

$$\hat{y} = y + L(r)y \qquad L(r) \approx k_1 r^2 + k_2 r^4$$

Разреженная 3D реконструкция



Драконъ, видимый подъ различными углами зрѣнія
По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus" Цана. 1702 года.

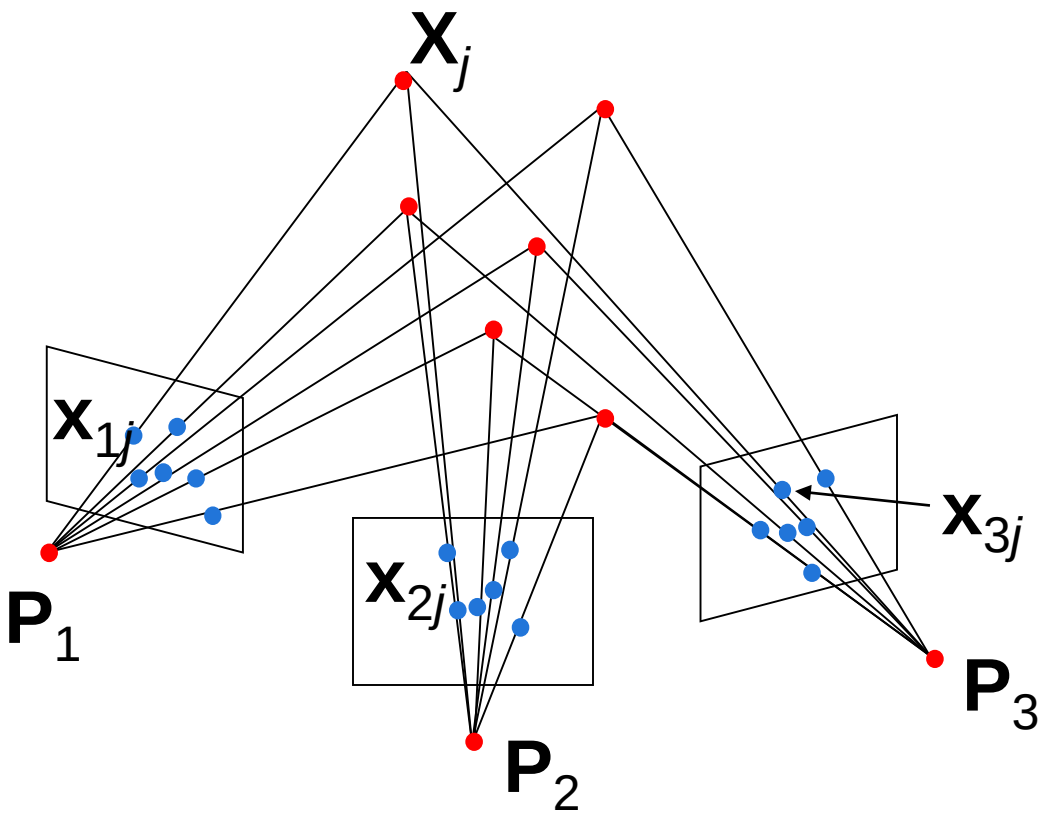
- Структура из движения
- Structure from Motion (SfM)

Структура из движения

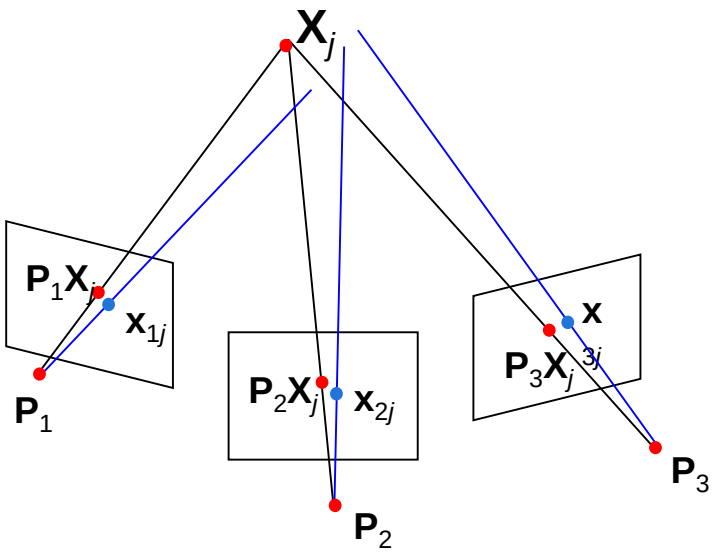
- Дано: *m* изображений *n* фиксированных 3D точек

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- Задача: оценить *m* матриц проекции $\mathbf{P}_i$ и *n* 3D точек $\mathbf{X}_j$ из *mn* соответствий $\mathbf{x}_{ij}$



Целевая функция

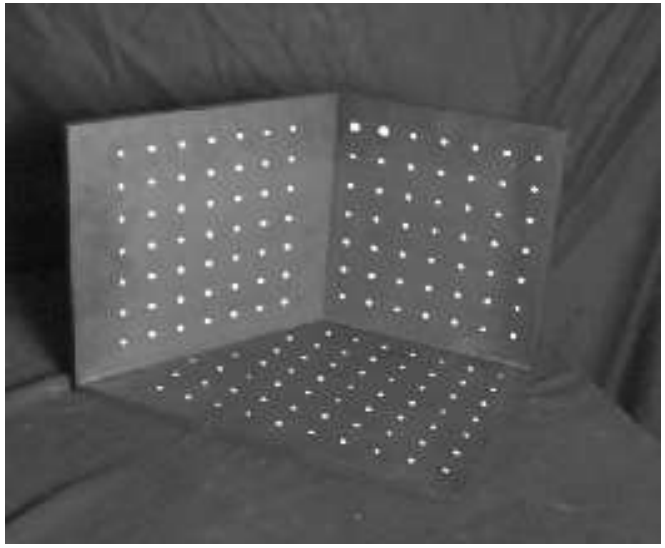


Оптимальная целевая функция – сумма ошибок репроекций 3D точек X на все изображения:

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

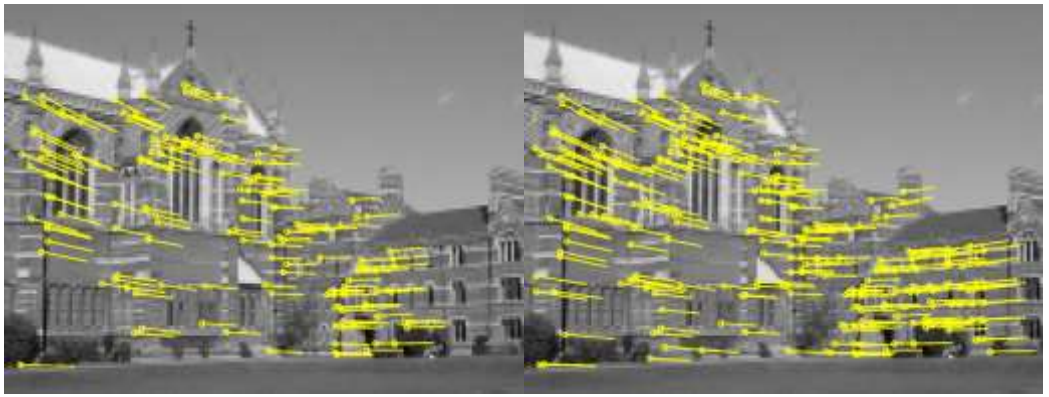
- Нелинейная целевая функция с большим количеством параметров
  - 3n точек
  - 6m параметров для калиброванного случая, 11m для некалиброванного
- Решается градиентным спуском, но плохо – масса локальных минимумов
- Поэтому нужно получить хорошее начальное приближение, в этом основная проблема

Что мы можем знать об изображениях?



Геометрию объекта и характерные точки

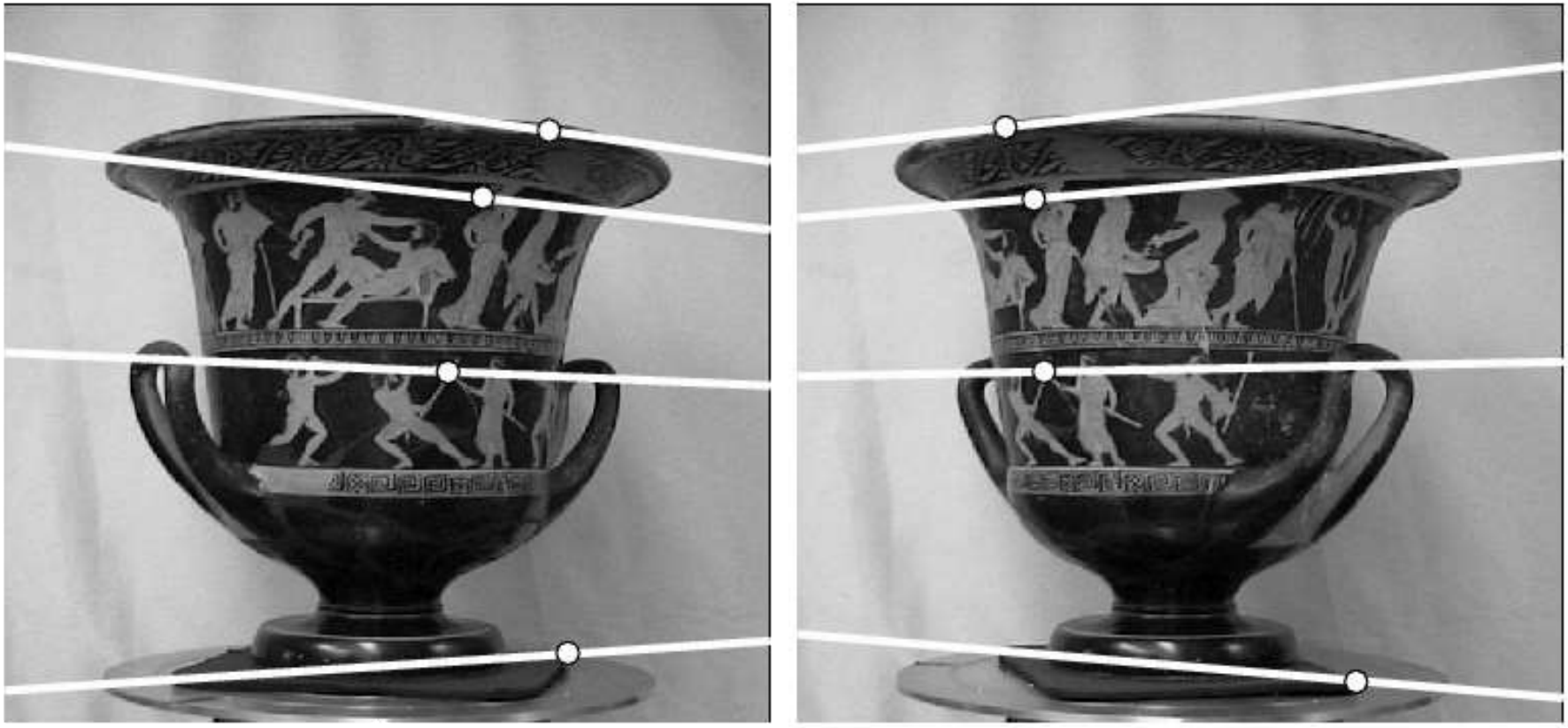
2D – 3D соответствия



Набор предположительных соответствий особых точек

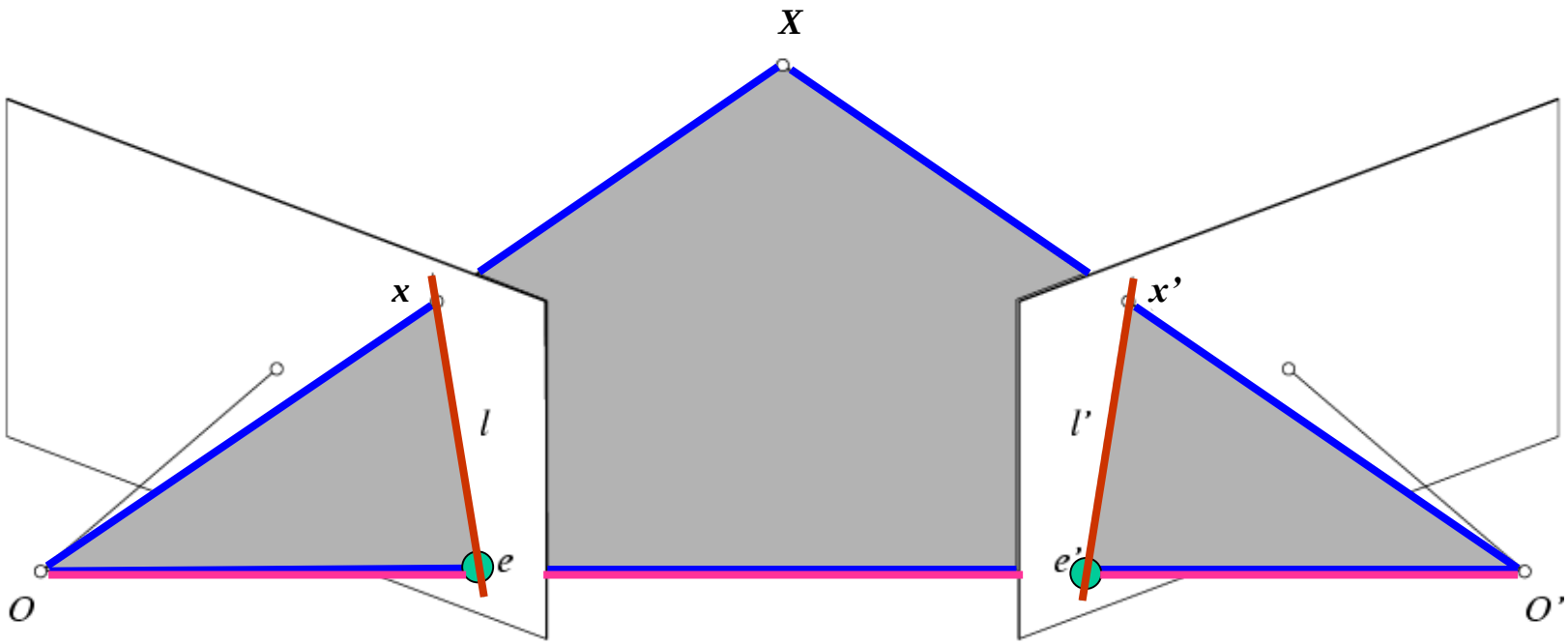
2D – 2D соответствия

Геометрия 2х камер



Рассмотрим случай, когда два снимка одной и той же сцены получены с разных ракурсов (разные центры проекций)

Эпиполярная геометрия



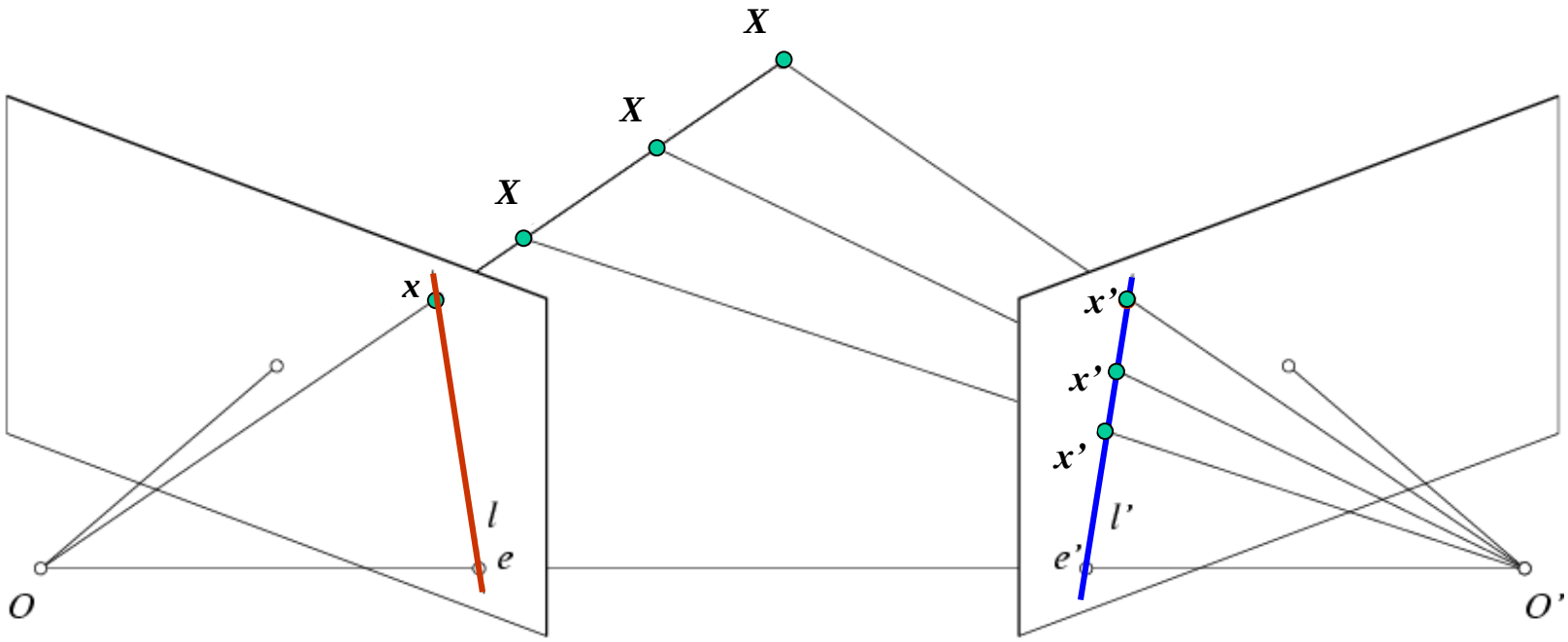
• **Базовая линия (Baseline)** – линия, соединяющая центры камер

• **Эпиполярная плоскость (Epipolar Plane)** – плоскость, проходящая через базовую линию (пучок плоскостей))

• **Эпиполи/Эпиполяры (Epipoles)**
= пересечение базовой линии с картинными плоскостями
= проекции центра проекций второй камеры

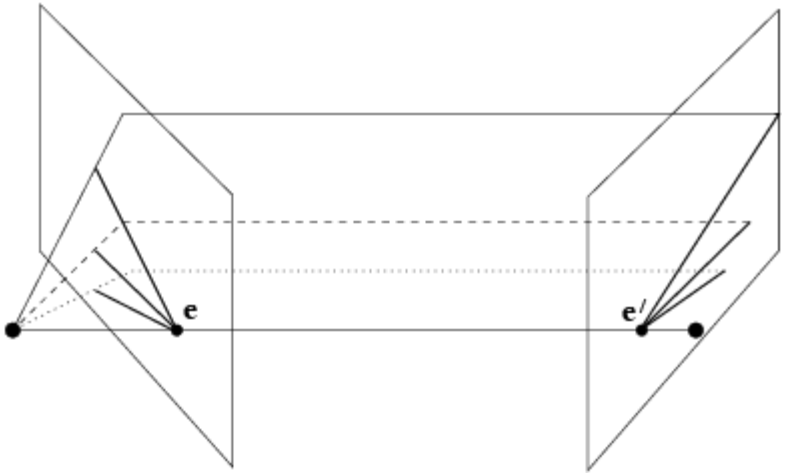
• Эпиполярные линии (**Epipolar Lines)** – пересечение эпиполярной плоскости с картинной плоскостью (дают соответствующие пары)

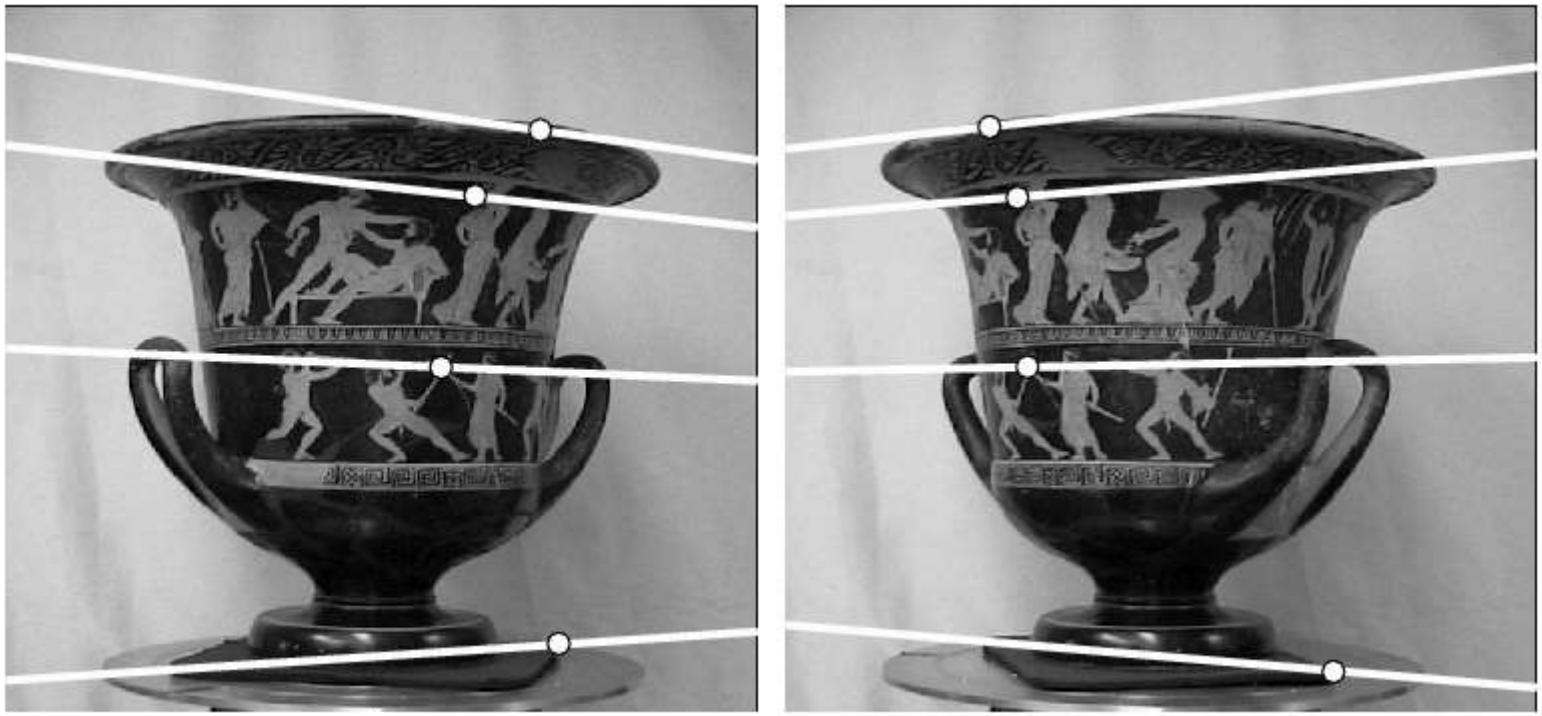
Эпиполярное ограничение



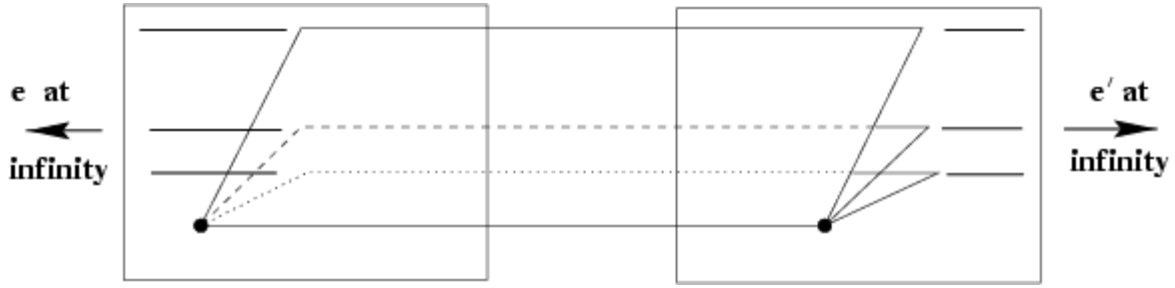
- Пусть даны изображения I и I’ с центрами проекции O и O’, и проекция **x** точки **X** (неизвестной) на изображение I.
- Что мы можем сказать про точку x’ – проекцию **X** на I’?
  - x’ должна лежать на эпиполярной линии l’
- Эпиполярное ограничение:
  - Проекции x и x’ точки сцены X на изображения I и I’ должны лежать на соответствующих эпиполярных линиях l и l’

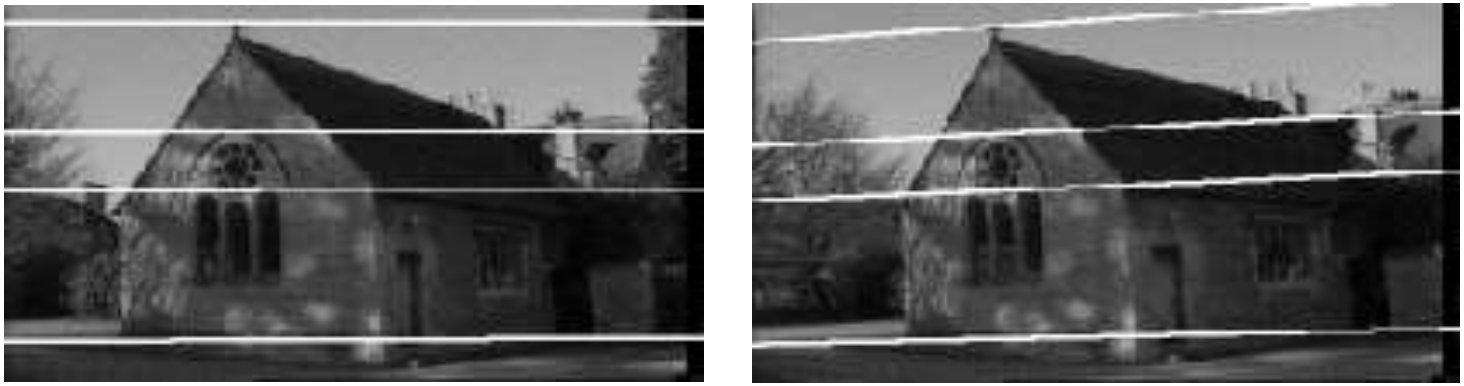
Пример: Сближающиеся камеры





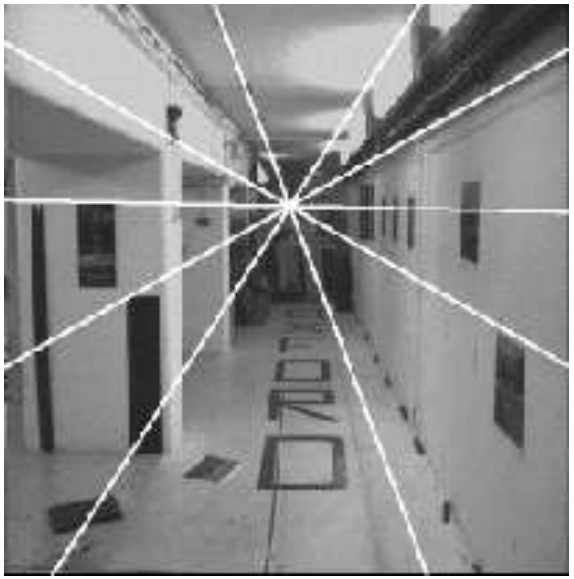
Пример: стереопара

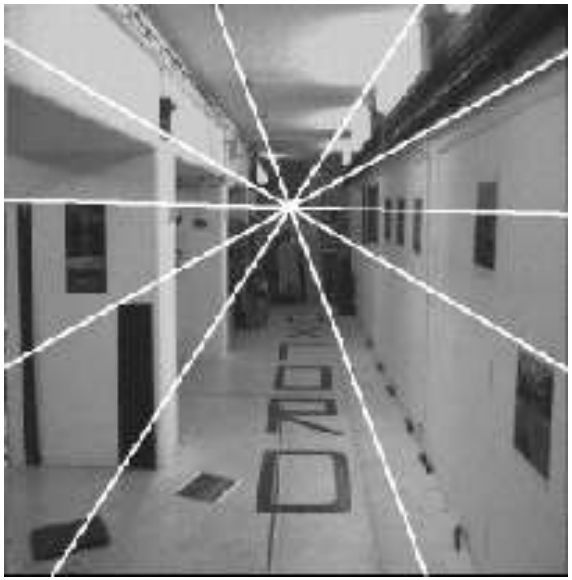


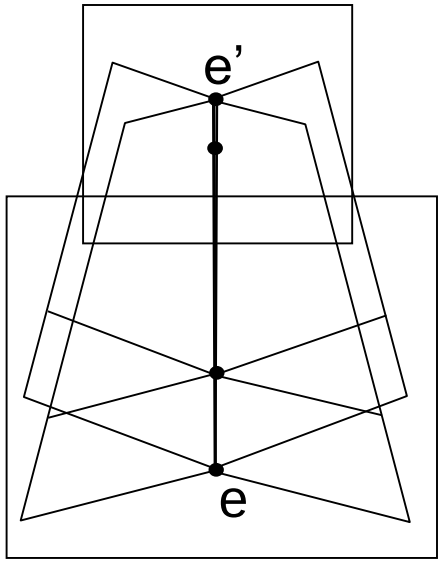


Движение, параллельное картинной плоскости

Пример: движение вперед

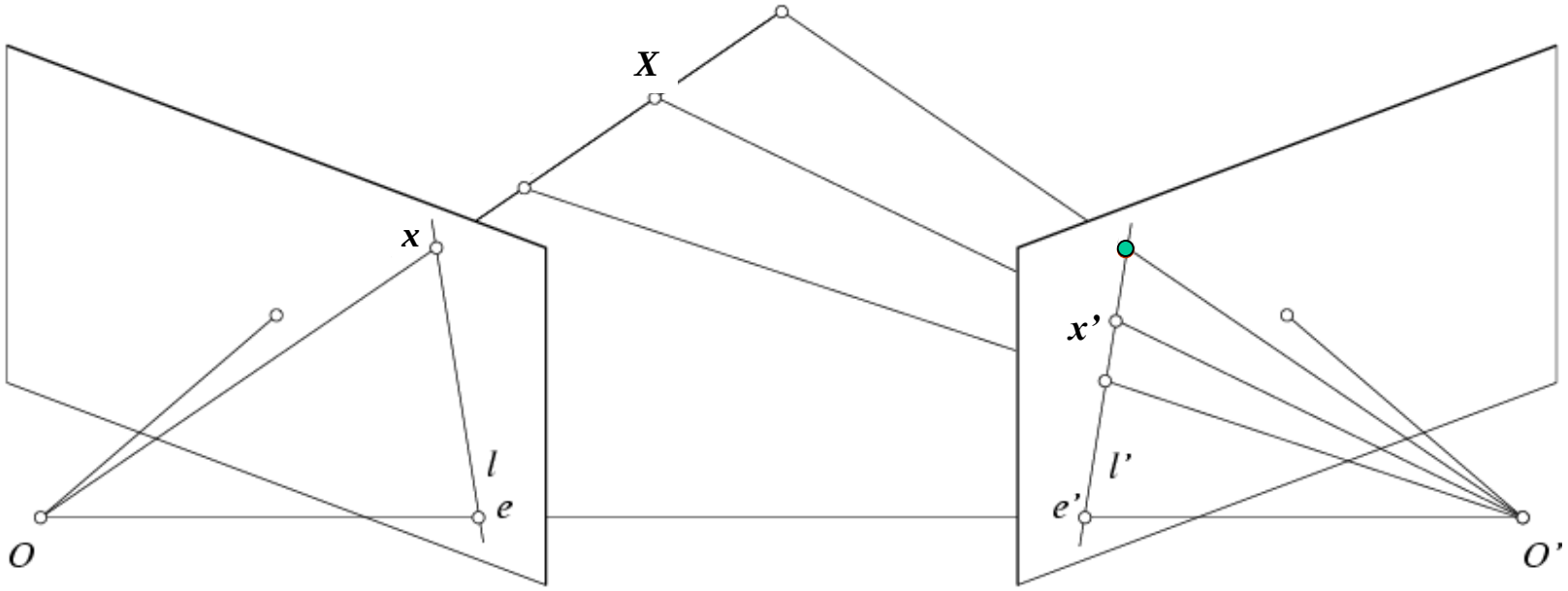






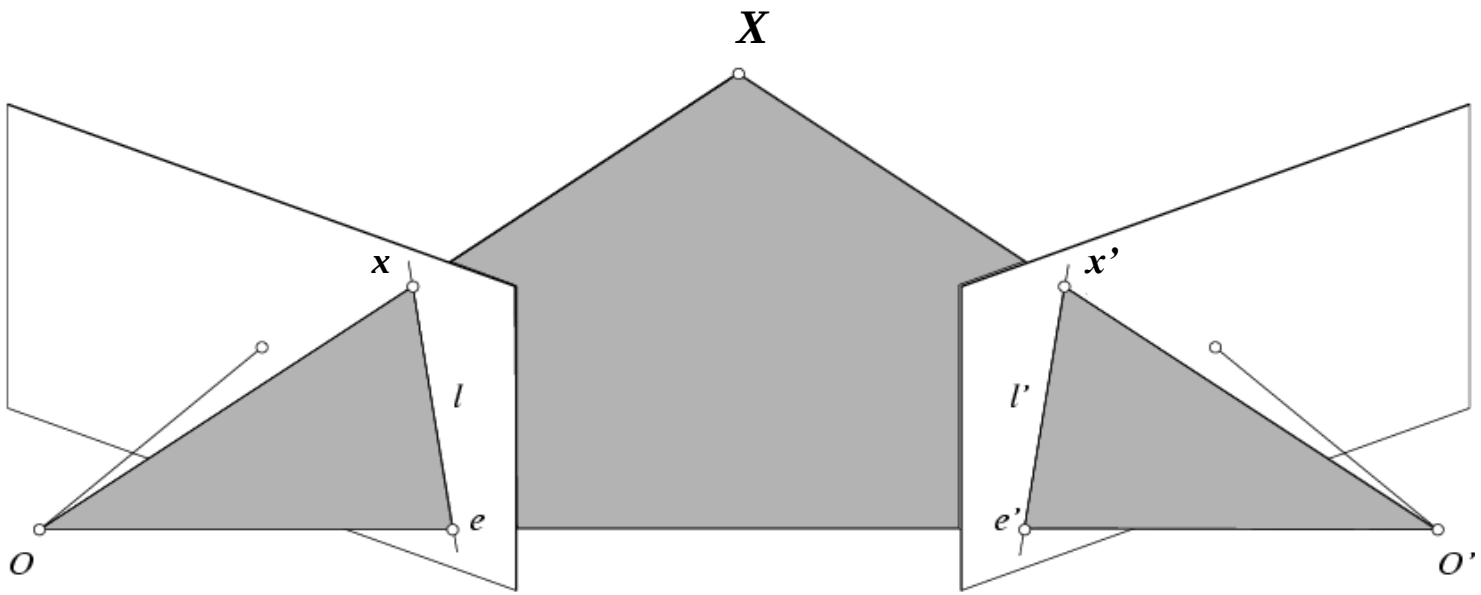
У эпиполей одинаковые координаты в обоих изображениях.
Точки движутся вдоль линий, исходящих из e – «фокуса расширения»

Эпиполярное ограничение



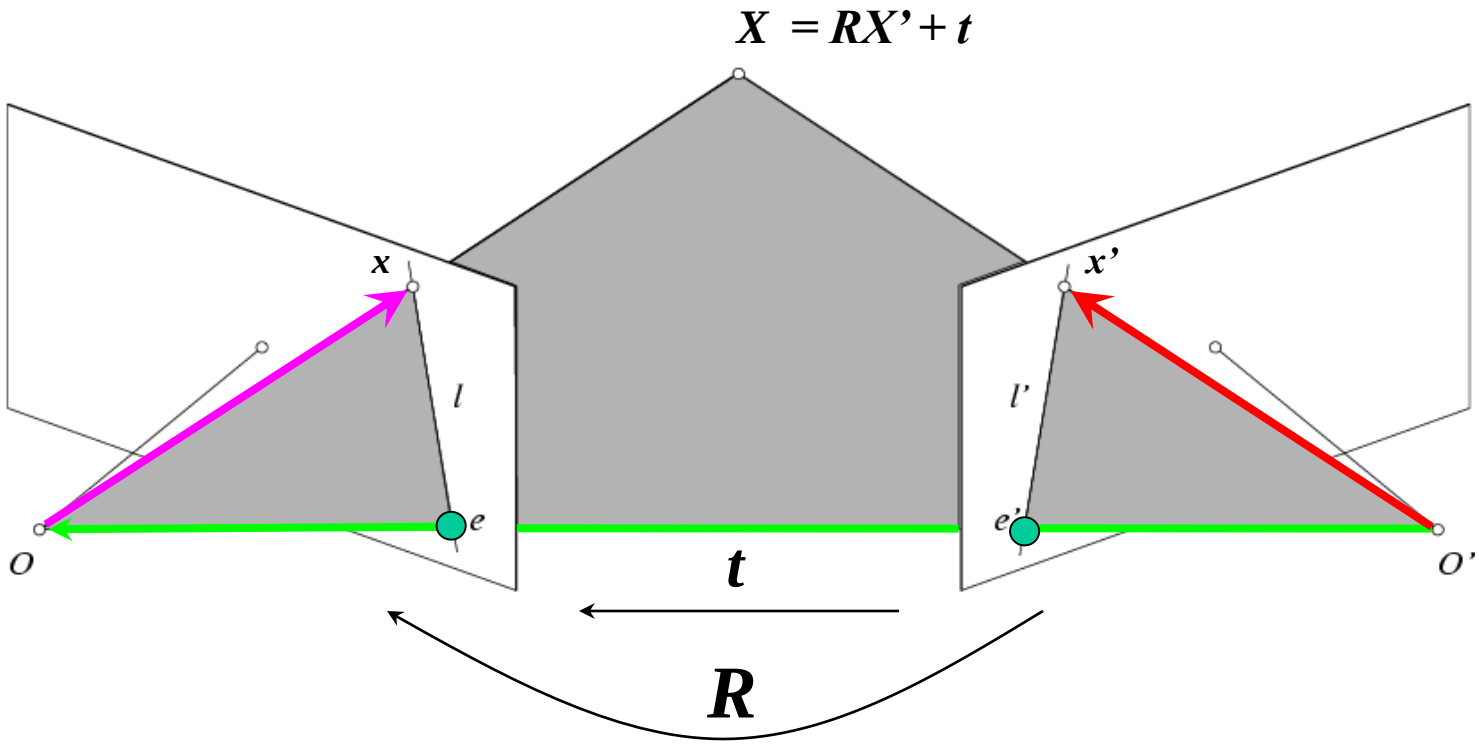
- Интересный вывод:
  - Линия l' не зависит от X, если x и калибровка камер известна
  - Линия l не зависит от X, если x' и калибровка камер известна
  - Между x, x' и калибровкой камер существует зависимость
  - Можем вывести эту зависимость

Калиброванный случай



- Совместим глобальную систему координат с координатной системой первой камеры
- Допустим, известны внутренние калибровки камер K, K’ и внешняя калибровка второй камеры R и T
- Умножим матрицу проекции каждой камеры на обратную матрицу калибровки K^{-1}, K'^{-1}
  - Также отобразим точки x, x’ с изображения на картинную плоскость

Калиброванный случай



Матрица камеры [I|0]

$\boldsymbol{X} = (u, v, w, 1)^T$

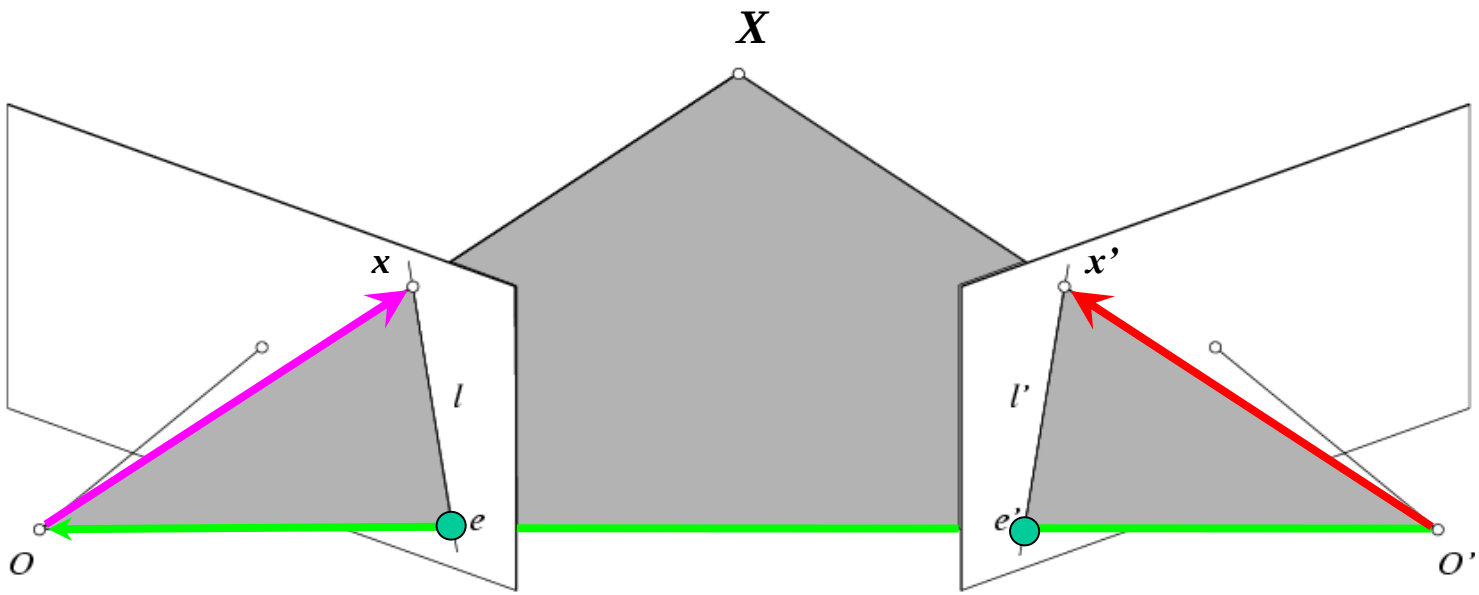
$\boldsymbol{x} = (u, v, w)^T$

Матрица камеры: $[\boldsymbol{R}^T \mid -\boldsymbol{R}^T\boldsymbol{t}]$

Вектор $\boldsymbol{x}$’ во второй системе координат имеет координаты $\boldsymbol{Rx}$’ в первой

Вектора $\boldsymbol{x}$, $\boldsymbol{t}$, и $\boldsymbol{Rx}$’ копланарны

Калиброванный случай

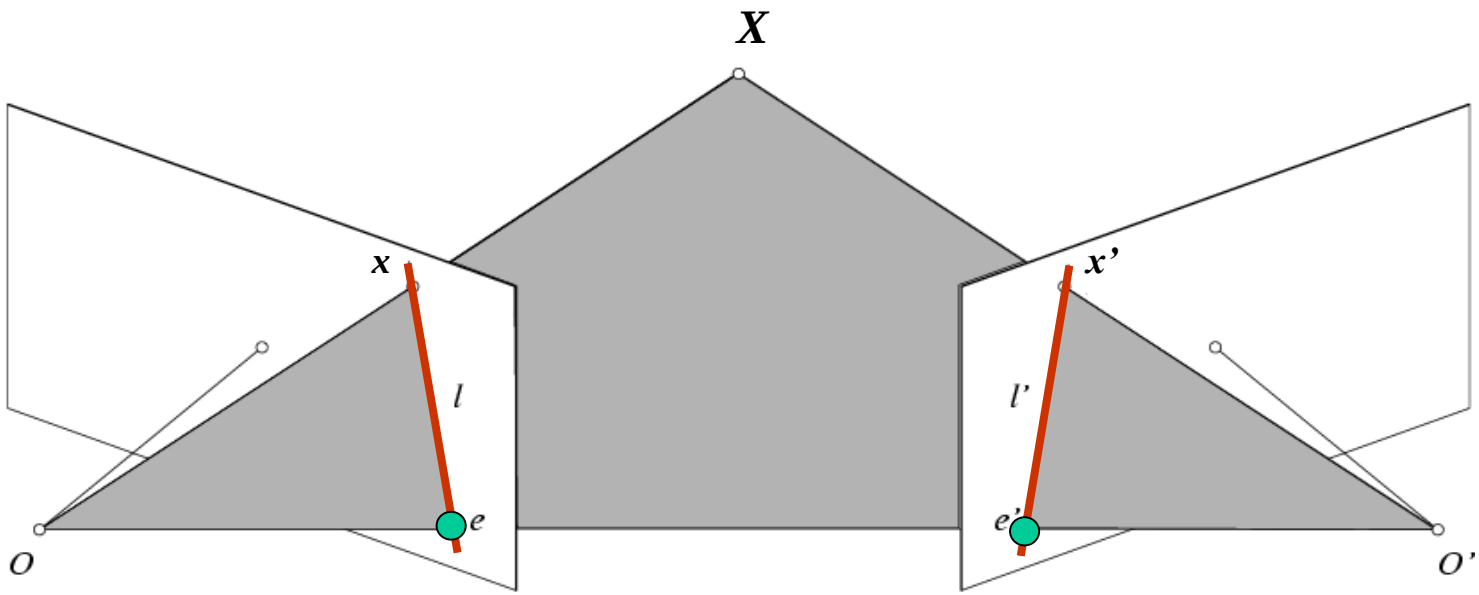


$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{где} \quad E = [t_\times]R$$

Существенная матрица (Essential Matrix)
(Longuet-Higgins, 1981)

Вектора ***x***, ***t***, и ***Rx'*** копланарны

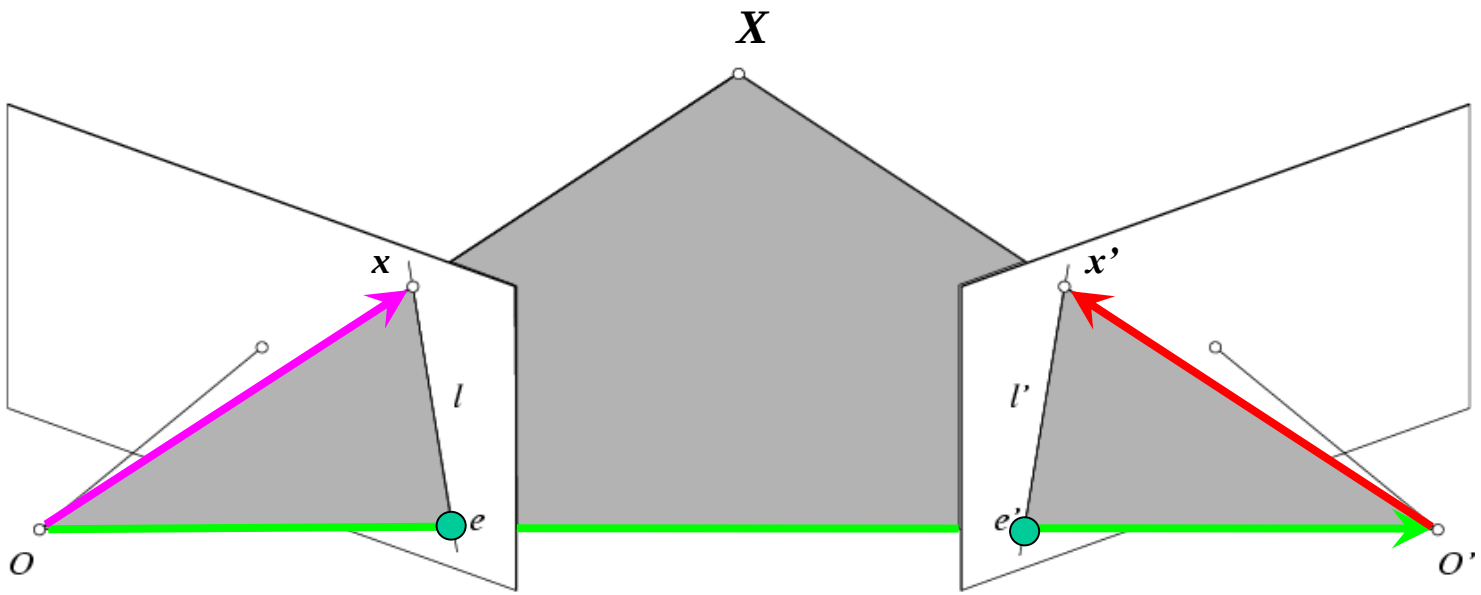
Свойства существенной матрицы



$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{где} \quad E = [t_\times]R$$

- $E\,x'$ - эпиполярная линия, соответствующая x' ($l = E\,x'$)
- $E^T x$ - эпиполярная линия, соответствующая x ($l' = E^T x$)
- $E\,e' = 0$ и $E^T e = 0$
- E вырождена (ранг 2)
- E имеет 5 степеней свободы

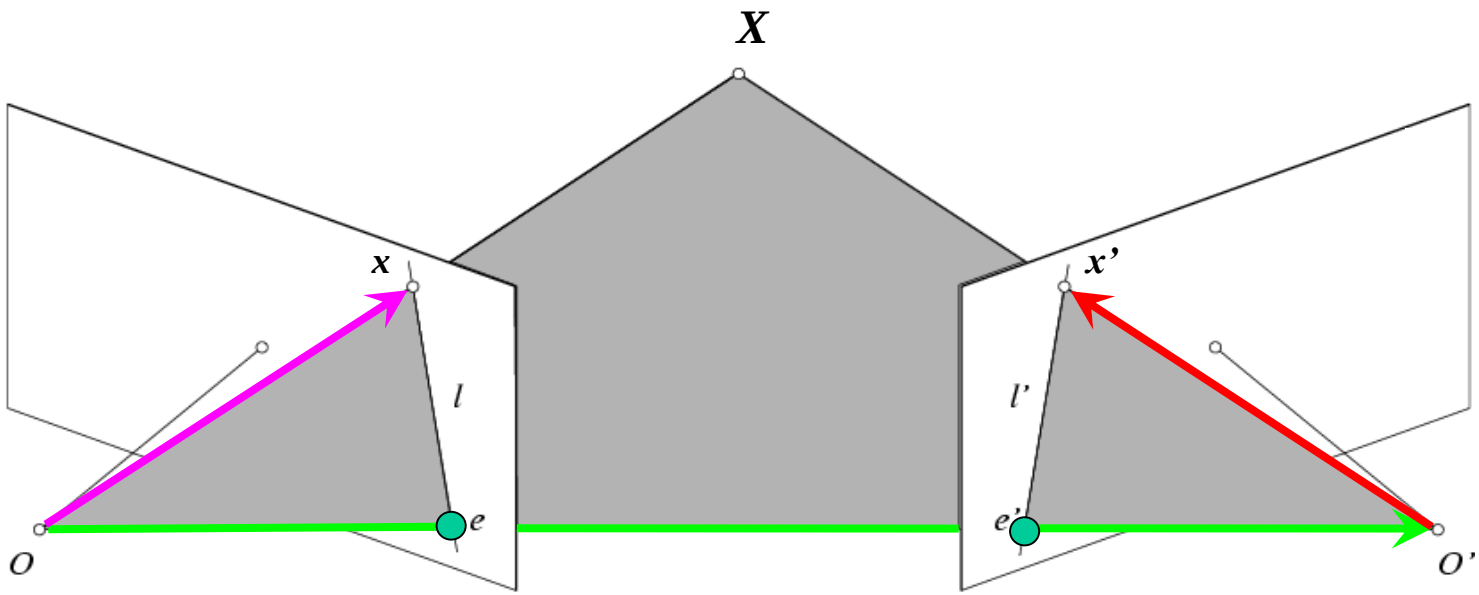
Некалиброванный случай



- Пусть матрицы калибровки K и K' для обеих камер неизвестны
- Запишем эпиполярное ограничение через *неизвестные* нормализованные координаты:

$$\hat{x}^T E \hat{x}' = 0 \qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Некалиброванный случай



$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{где} \quad F = K^{-T} E K'^{-1}$$

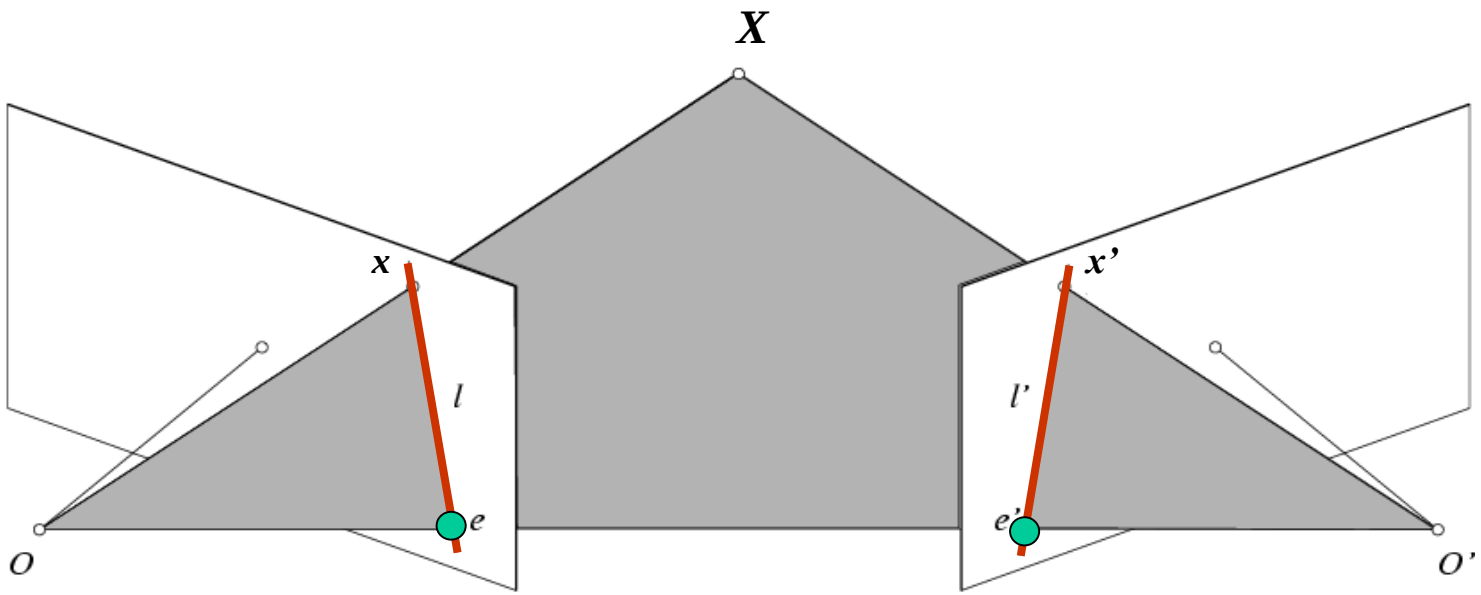
$$x = K\hat{x}$$

$$x' = K'\hat{x}'$$

Фундаментальная матрица (Fundamental Matrix)

(Faugeras and Luong, 1992)

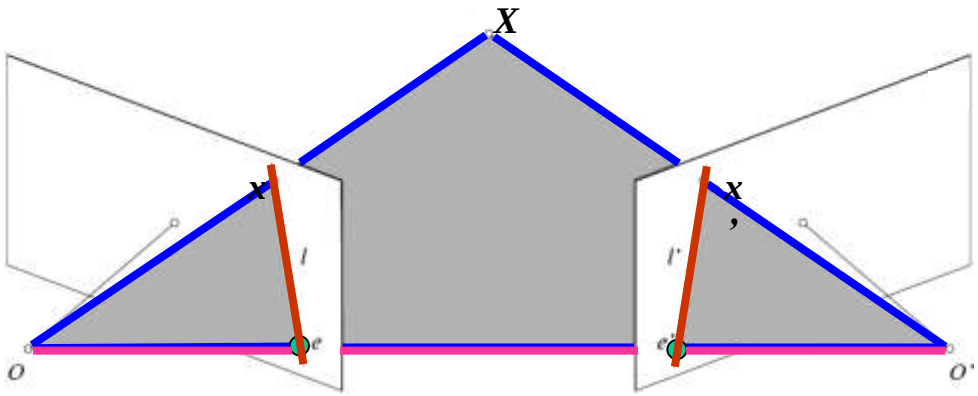
Свойства



$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{где} \quad F = K^{-T} E K'^{-1}$$

- $F\,x$ - эпиполярная линия, соответствующая x' ($l = F\,x'$)
- $F^T x$ - эпиполярная линия, соответствующая x ($l' = F^T x$)
- $F\,e' = 0$ и $F^T e = 0$
- F вырождена (ранг 2)
- F имеет семь степеней свободы

Резюме эпиполярной геометрии



- Базовая линия (Baseline)
- Эпиполярная плоскость (Epipolar Plane)
- Эпиполи/Эпиполяры (Epipoles)
- Эпиполярные линии (Epipolar Lines)

Пара камер порождает «эпиполярную геометрию», т.е. зависимость между парами соответствующих точек и калибровкой камер

Существенная матрица (калиброванный случай):

$$x^T E x' = 0 \quad где \quad E = [t_\times]R$$

Фундаментальная матрица (некалиброванный случай):

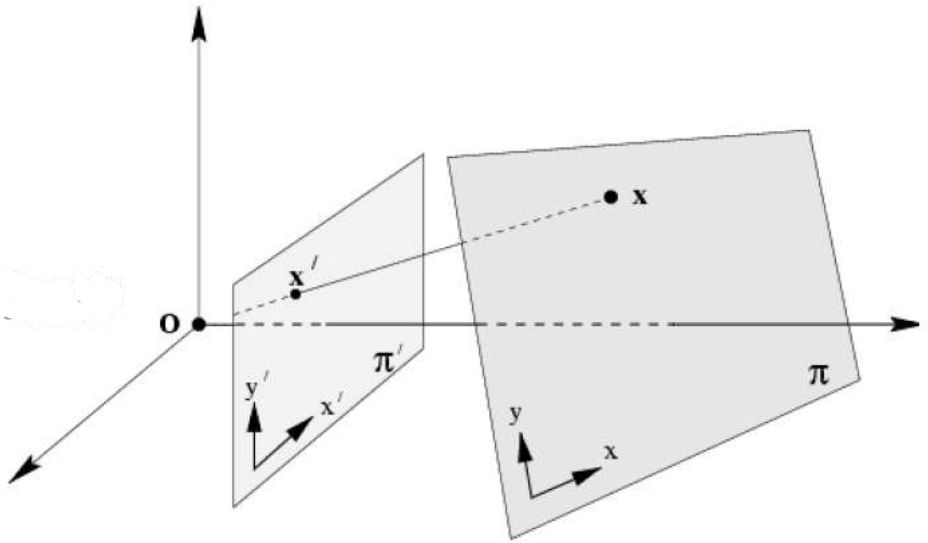
$$x^T F x' = 0 \quad где \quad F = K^{-T} E K'^{-1}$$

Существенная и фундаментальная матрицы описывают зависимости между калибровкой камер и парам соответствующих точек на изображениях

Гомография как вырожденный случай

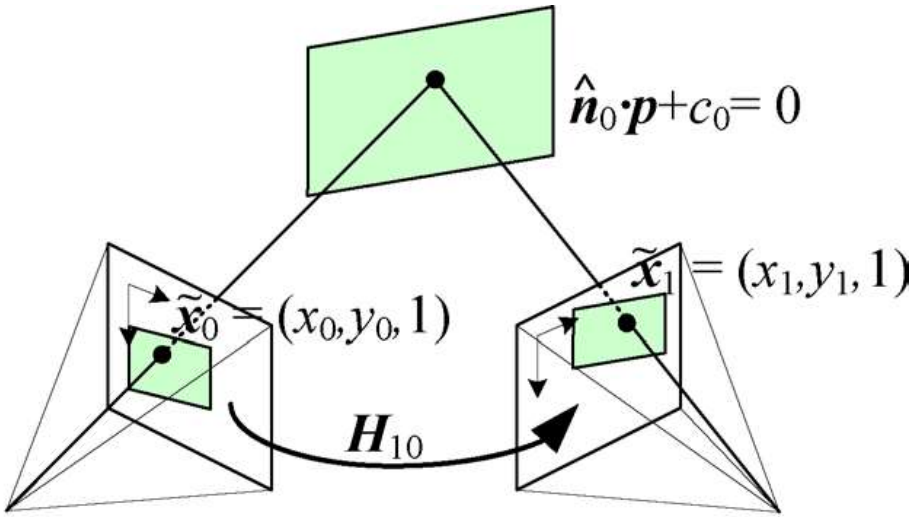
- Перспективное преобразование плоскости (гомография)

- 8 степеней свободы

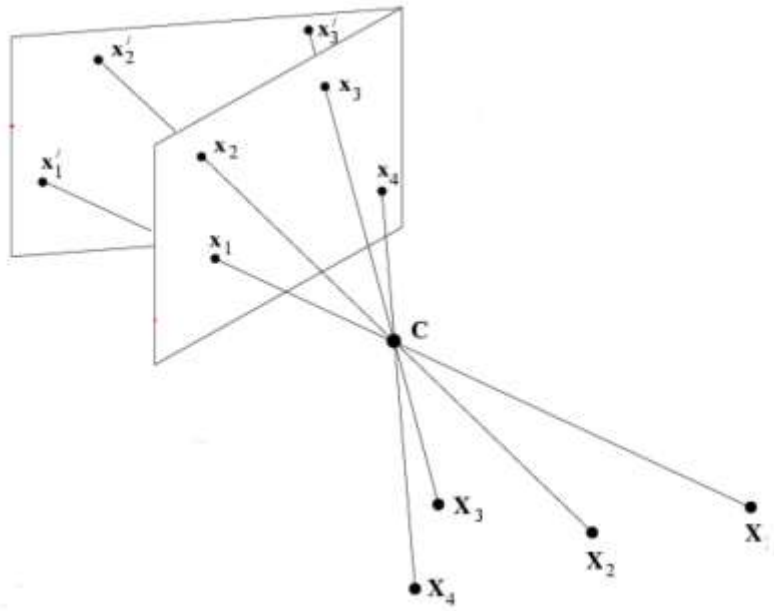


$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{12} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Применение гомографии



Плоская сцена



Камера на штативе

Вывод: если строим панораму или карту местности (считая её «плоской»), то одно изображение преобразуется в другое с помощью гомографии

Подзадачи структуры и движения

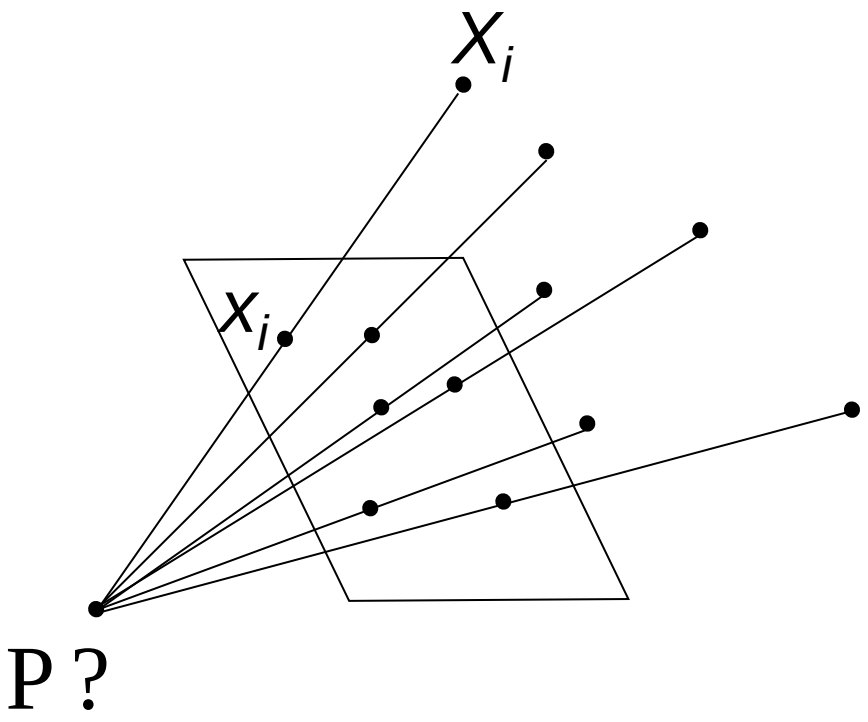
- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2х и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2х и более изображениях. Определить 3D координаты точки.

Подзадачи структуры и движения

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2х и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2х и более изображениях. Определить 3D координаты точки.

Калибровка камеры (PnP-проблема)

Perspective-*n*-point problem (PnP) - даны n точек с известными 3D координатами X_i и известными проекциями x_i, оценить параметры камеры



DLT-метод

Direct Linear Transformation (прямое линейное преобразование)

$$\lambda \mathrm{x}_i = \mathrm{P}\mathrm{X}_i \qquad \lambda \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} \mathrm{P}_1^T \\ \mathrm{P}_2^T \\ \mathrm{P}_3^T \end{bmatrix} \mathrm{X}_i \qquad \mathrm{x}_i \times \mathrm{P}\mathrm{X}_i = 0$$

$$\begin{bmatrix} 0 & -w_i \mathrm{X}_i^T & y_i \mathrm{X}_i^T \\ w_i \mathrm{X}_i^T & 0 & -x_i \mathrm{X}_i^T \\ -y_i \mathrm{X}_i^T & x_i \mathrm{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0$$

- Два линейно независимых уравнения (берут обычно первые два)

Калибровка камеры

$$\begin{bmatrix} 0^T & \mathrm{X}_1^T & -y_1\mathrm{X}_1^T \\ \mathrm{X}_1^T & 0^T & -x_1\mathrm{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathrm{X}_n^T & -y_n\mathrm{X}_n^T \\ \mathrm{X}_n^T & 0^T & -x_n\mathrm{X}_n^T \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0 \qquad \mathrm{Ap} = 0 \qquad \|\mathrm{p}\| = 1$$

- У P 11 степеней свободы (12 параметров, но масштаб произвольный)
- Каждая пара (x,X) даёт нам уравнения
- 6 соответствий необходимо для решения (p6p)
- Однородные наименьшие квадраты

Калибровка камеры

- Когда мы получили матрицу проекции, нам необходимо извлечь из неё внутренние и внешние параметры калибровки
- Представим матрицу проецирования в следующем виде:

$$P = KR^T \left[I \mid -T \right] = \left[M \mid -MT \right]$$

- Для разложения M в KR^T воспользуемся RQ-факторизацией
  - Q – ортогональная матрица
  - K – верхнетреугольная
- Потом найдем T

“Gold Standard”

- Метод «Золотого стандарта» – метод, оптимизирующий параметры по наиболее корректной, оптимальной метрике
- В случае калибровки камеры, это расстояния от проекций 3D точек до их измеренных значений (ошибка репроекции):

$$\min_{P} \sum_{i} d^2(x_i, P\,X_i)$$

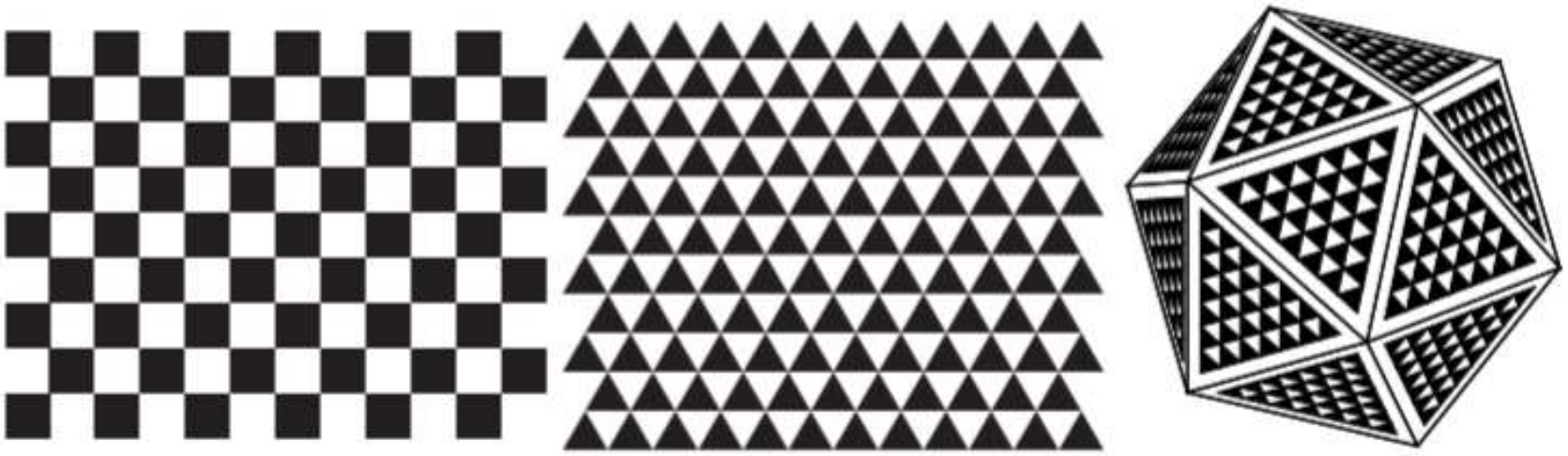
- Требует использования нелинейных методов оптимизации (градиентный спуск и т.д.)

Схема калибровки камеры

- DLT-метод для получения матрицы проецирования
- RQ-факторизация для извлечения матрицы внутренней калибровки K и внешней калибровки R и T
- Уточнение параметров калибровки с помощью нелинейной оптимизации ошибки репроекции:

$$\min_{P} \sum_{i} d^2(x_i, P\,X_i)$$

Калибровочный шаблон



(a) Checkerboard (b) Deltille Grid (c) Icosahedron

Figure 1: Regular tiling with uniform but alternating colorings. (a) Checkerboard pattern, (b) a Deltille pattern with the same edge length as (a), (c) a 3D calibration object textured with a deltille grid.

Подзадачи

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2х и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2х и более изображениях. Определить 3D координаты точки.

Калибровка камер (движение)

Дан набор соответствующих точек в 2х и более изображениях, определить матрицы калибровки камер для этих изображений (видов)

- Существенная матрица:

$$x^T E x' = 0 \quad где \quad E = [t_\times]R$$

- Фундаментальная матрица:

$$x^T F x' = 0 \quad где \quad F = K^{-T} E K'^{-1}$$

- Если известна фундаментальная матрица и матрицы внутренней калибровки, тогда мы можем вычислить матрицу $E = K^T F K'$ и из неё извлечь внешнюю калибровку камеры

8-и точечный алгоритм

• Фундаментальная матрица накладывает ограничения на пары соответствующих точек
• Попробуем её вычислить напрямую из соответствий, без калибровки камеры

• Даны соответствия (x,x'): $\boldsymbol{x} = (u, v, 1)^T$, $\boldsymbol{x'} = (u', v', 1)^T$

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

• 1 соответствие даёт 1 уравнение на F

• Если точек много, можем свести к обычной задаче однородных наименьших квадратов

8-и точечный алгоритм

$$(uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Минимизируем:

$$\sum_{i=1}^{N} (x_i^T F x'_i)^2$$

При условии

$$|F|^2 = 1$$

Приведение к рангу 2

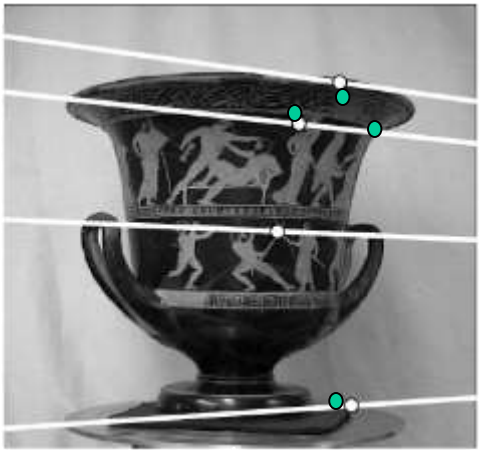
- Фундаментальная матрица F имеет ранг 2
- В общем случае, результат DLT-оценки F имеет ранг 3 (из-за влияния шума)
- Необходимо привести к рангу 2
- Найдем F’, минимизирующую норму Фробениуса:

$$\|F - F'\|, где \ \mathrm{rang}(\mathrm{F}') = 2$$

- Воспользуемся SVD-разложением:
- $F=UDV^T$, или $F = Udiag(r,s,t)V^T$, где $r \geq s \geq t$
- $F'= = Udiag(r,s,0)V^T$

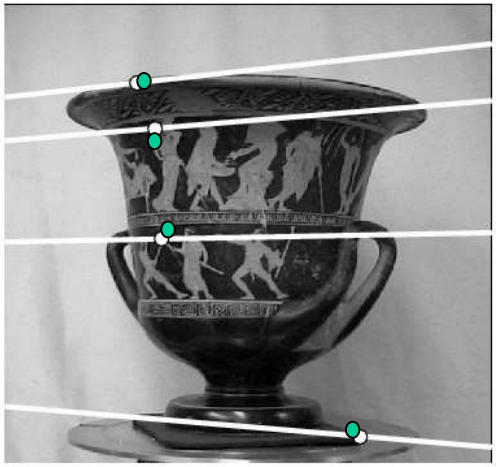
“Gold standard”

- Ошибка $\sum_{i=1}^{N}(x_i^T F x_i')^2$ - алгебраическая
  - Не имеет физического смысла



- «Оптимальная» ошибка:

$$\sum_{i=1}^{N}\left[\mathrm{d}^2(x_i', \hat{x}_i') + \mathrm{d}^2(x_i, \hat{x}_i)\right], \text{ где } (\hat{x}_i')^T F \hat{x}_i = 0$$



  - Т.е. сумма квадратов расстояний от измеренных точек, до «идеальных»
  - Требует нелинейной оптимизации

Если данные зашумлены?

- В соответствиях могут быть ошибки!
  - Ложные соответствия
- Нужно использовать робастные методы
- Обычно используют метод RANSAC с ошибкой:

$$\sum_{i=1}^{N}\left[\mathrm{d}^2(x_i', Fx_i) + \mathrm{d}^2(x_i, F^T\hat{x}_i)\right]$$

  - Ошибка переноса (transfer error)

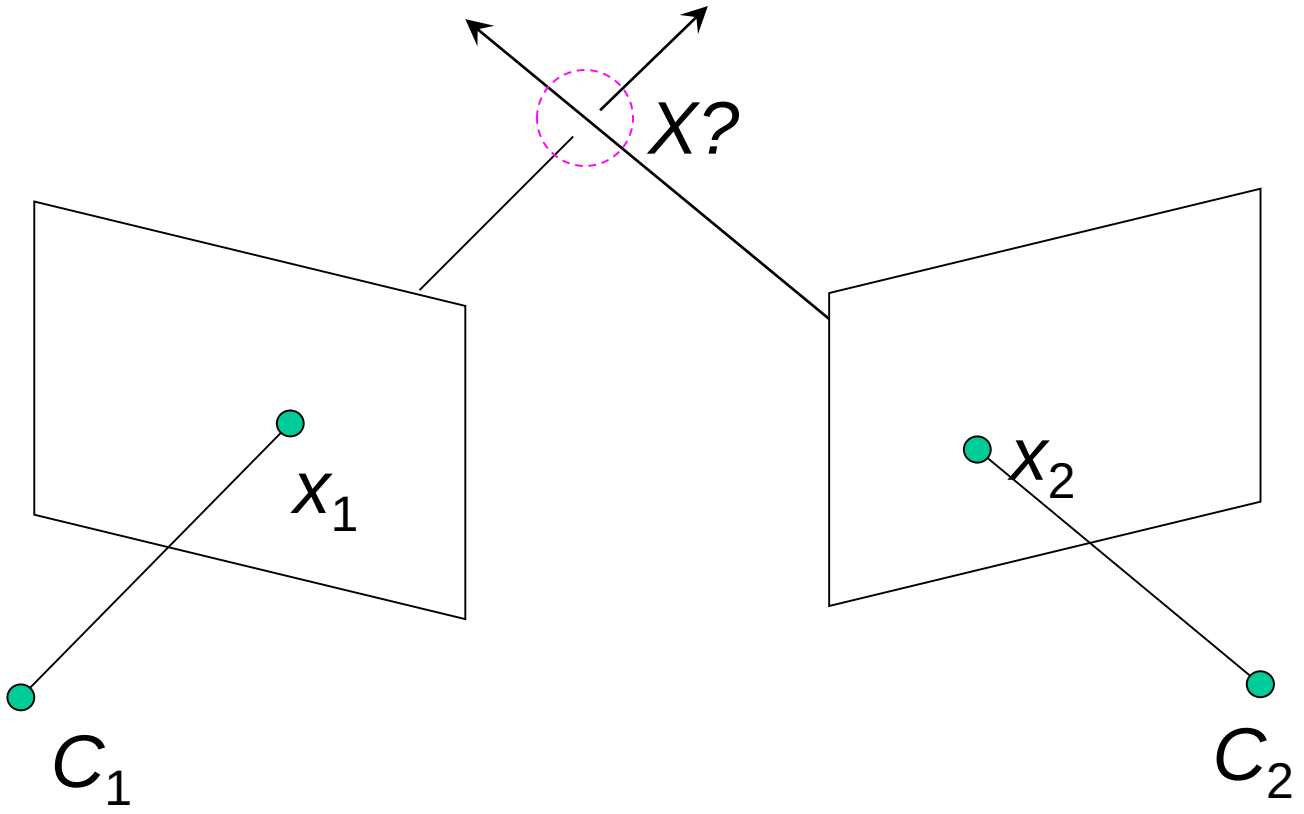
•Затем уточняют по всем найденным корректным соответствиям

Подзадачи

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2х и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2х и более изображениях. Определить 3D координаты точки.

Триангуляция («структура» сцены)

Даны проекции x_1, x_2 точки **X**=(X,Y,Z) на 2 или более изображения (с известными матрицами калибровки), найти координаты точки



Линейный подход

Проекция на каждую камеру даст 2 уравнения на X:

$$\lambda_1 \mathrm{x}_1 = \mathrm{P}_1 \mathrm{X} \qquad \mathrm{x}_1 \times \mathrm{P}_1 \mathrm{X} = 0 \qquad [\mathrm{x}_{1\times}]\mathrm{P}_1 \mathrm{X} = 0$$

$$\lambda_2 \mathrm{x}_2 = \mathrm{P}_2 \mathrm{X} \qquad \mathrm{x}_2 \times \mathrm{P}_2 \mathrm{X} = 0 \qquad [\mathrm{x}_{2\times}]\mathrm{P}_2 \mathrm{X} = 0$$

$$x_1(p_1^{3T}X) - (p_1^{1T}X) = 0$$

$$y_2(p_1^{3T}X) - (p_1^{2T}X) = 0$$

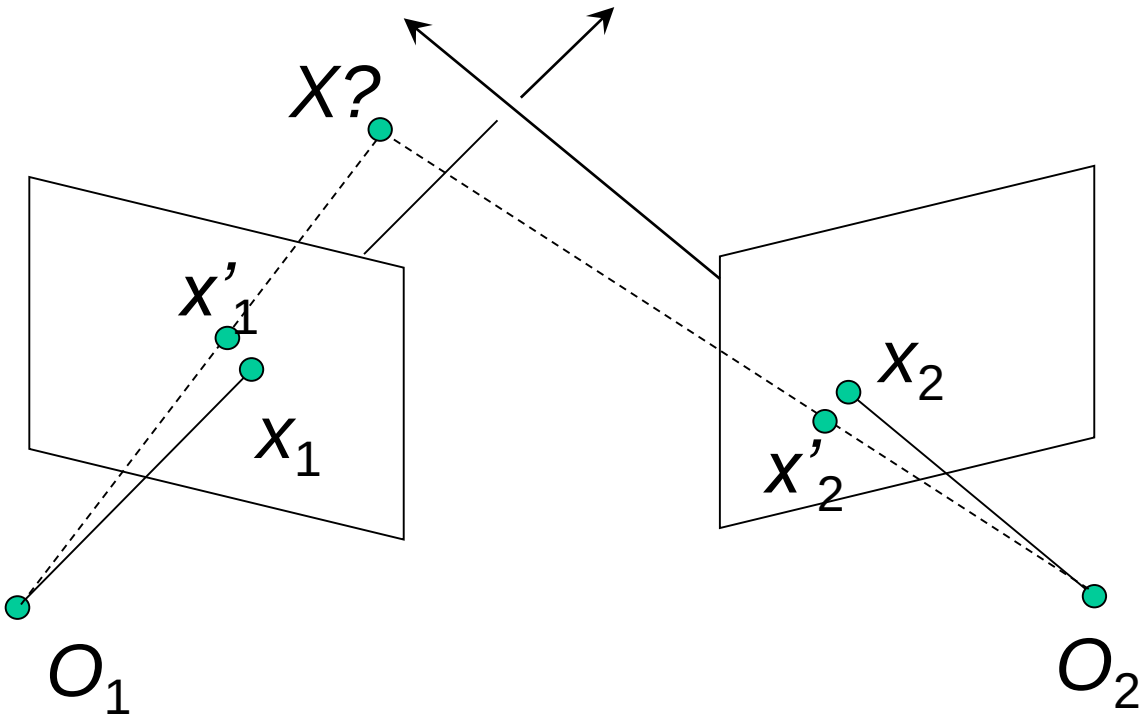
Проекция на обе камеры приведет к знакомой задаче на однородные наименьшие квадраты

$$A = \begin{bmatrix} x_1 p_1^{3T} - p_1^{1T} \\ y_1 p_1^{3T} - p_1^{2T} \\ x_2 p_2^{3T} - p_2^{1T} \\ y_2 p_2^{3T} - p_2^{2T} \end{bmatrix} x = 0 \qquad \begin{matrix} AX = 0 \\ \|X\| = 1 \end{matrix}$$

Метод «золотого стандарта»

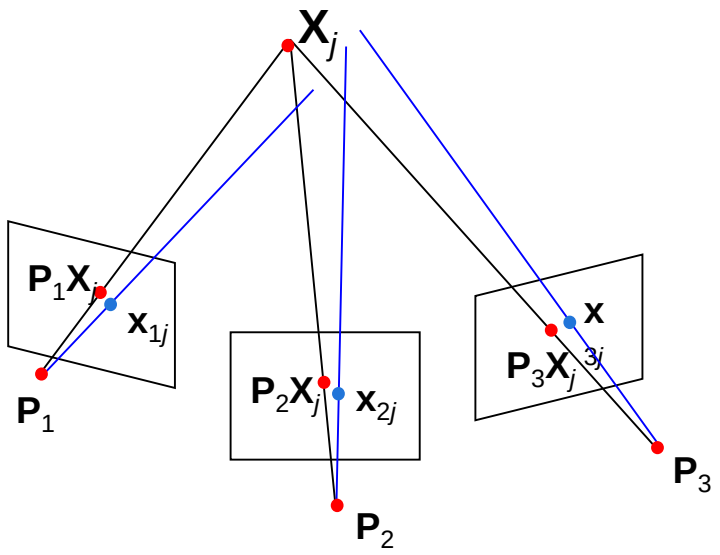
Найти точку X, минимизирующую ошибку репроекции:

$$d^2(x_1, P_1X) + d^2(x_2, P_2X)$$



- Результат DLT используем как начальное приближение
- Требует применение нелинейной оптимизации

Вернёмся к структуре и движению



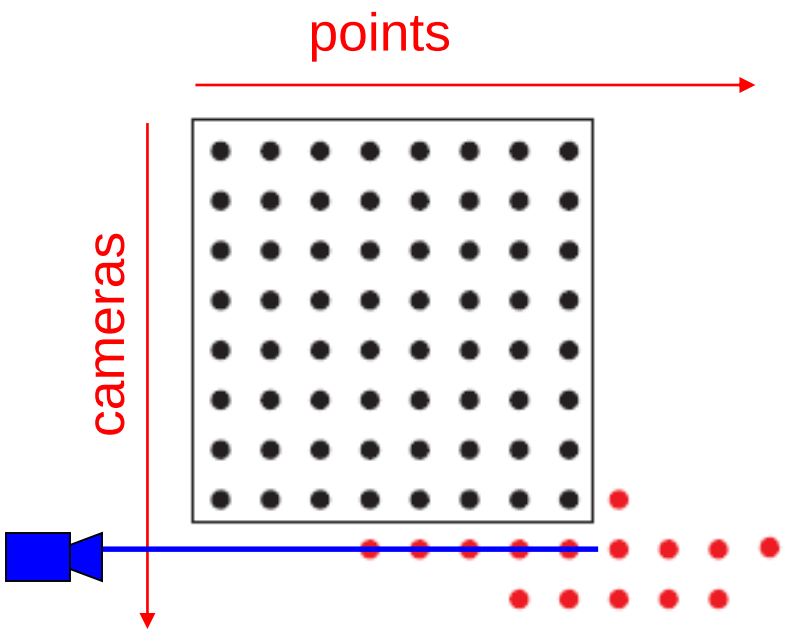
Оптимальная целевая функция – сумма ошибок репроекций 3D точек X на все изображения:

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

- Будем рассматривать случай, когда известны внутренние калибровки всех камер K_i
- Тогда мы можем вычислять существенную матрицу между двумя видами по соответствиям, т.е. оценивать напрямую внешнюю калибровку камеры
- Составим алгоритм для получения начального приближения и уточним результат градиентным спуском

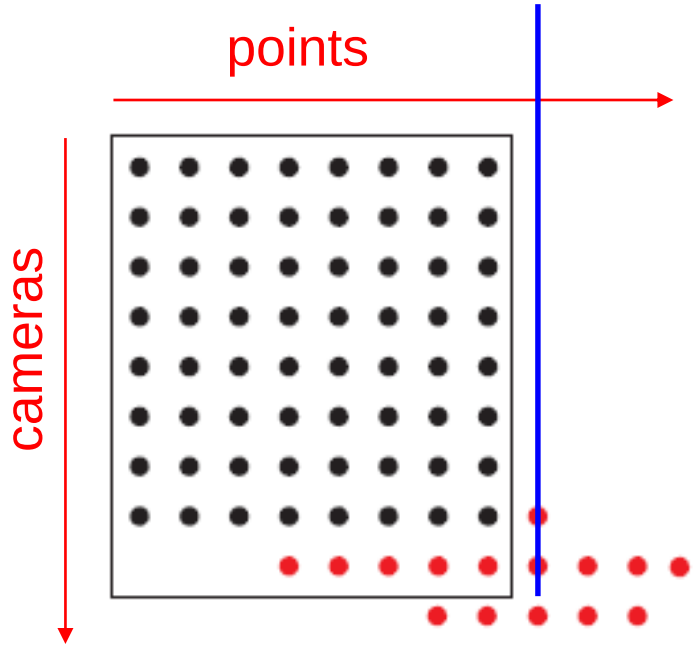
Последовательный подход

- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого вида:
  - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка



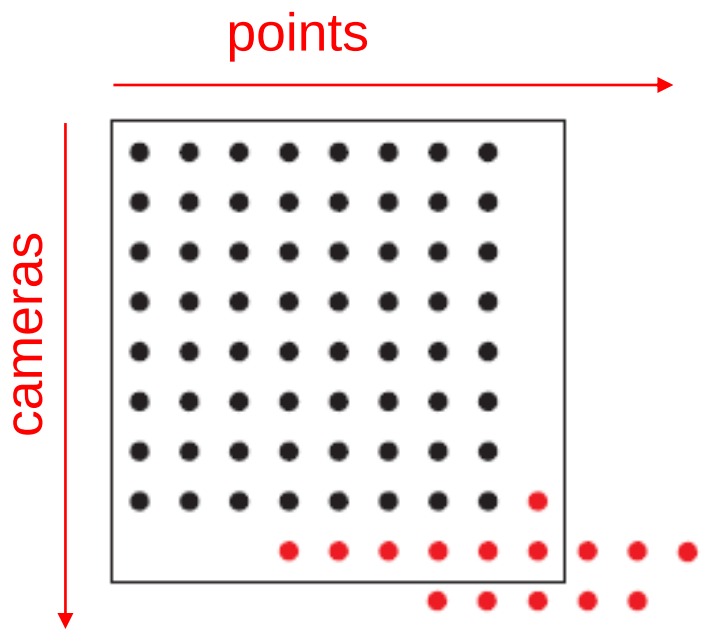
Последовательный подход

- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого вида:
  - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка
  - Уточняем и дополняем структуру, вычисляем новые 3D точки, уточняем существующие точки, видимые на камере - триангуляция



Последовательный подход

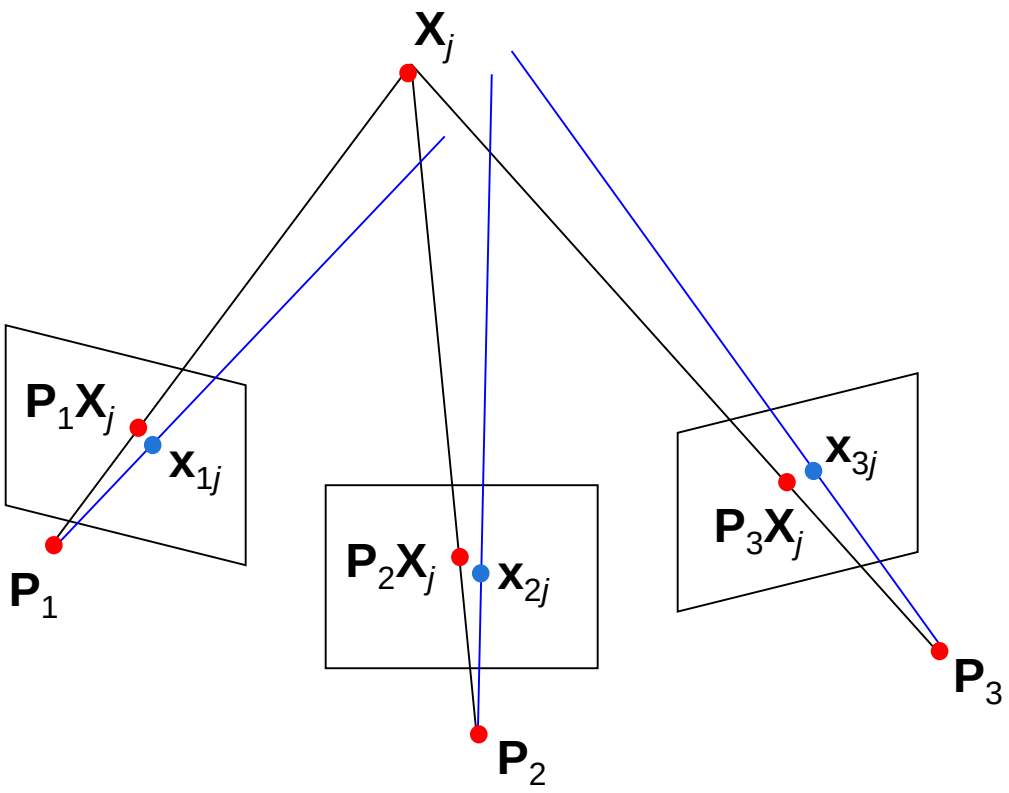
- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого вида:
  - •Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка
  - •Уточняем и дополняем структуру, вычисляем новые 3D точки, уточняем существующие точки, видимые на камере - триангуляция
- •Уточняем структуру и в движение методом связок



Метод связок (Bundle adjustment)

- Нелинейный метод для уточнения структуры и движения
- Минимизируем сумму ошибок проекций всех точек на все камеры:

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$



Неоднозначность решения

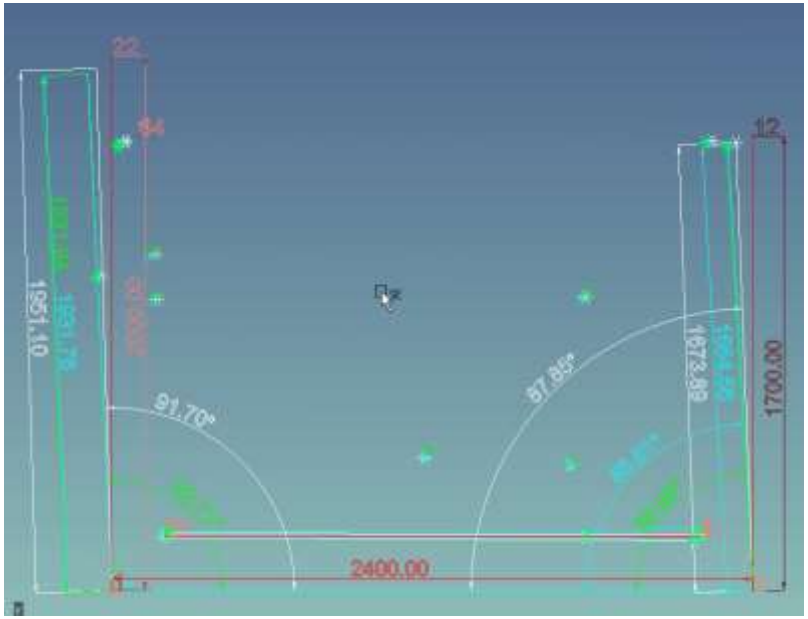
•Если мы умножим всю сцену на некоторый коэффициент *k* и в то же время умножим матрицы камер на 1/*k*, проекции точек сцены на изображения не изменятся:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\mathbf{X})$$

Вывод: оценить абсолютный масштаб (размеры) сцены только по изображениям невозможно!

Нужна дополнительная информация

Фотограмметрия



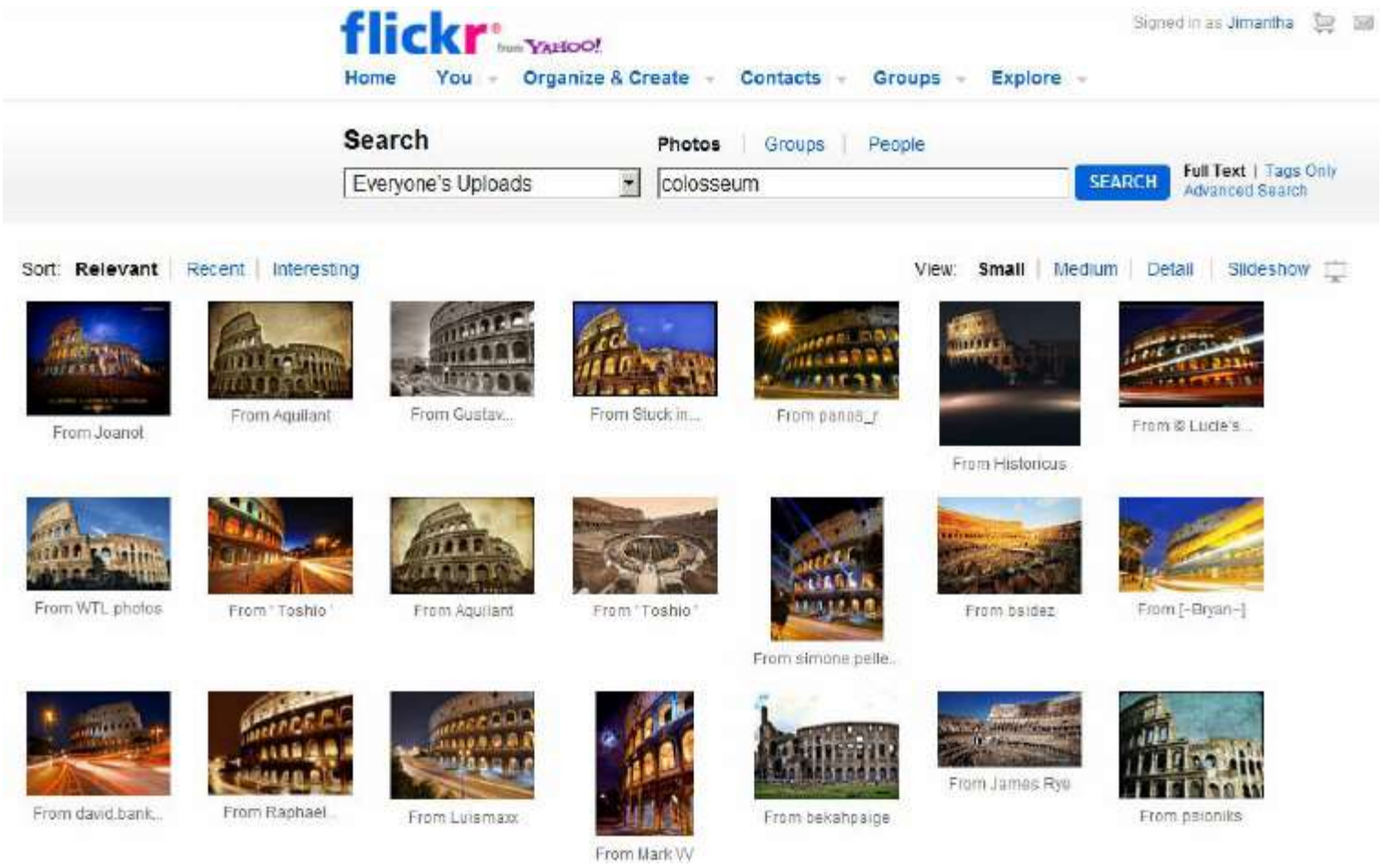
Чтобы провести измерения по изображениям (задача фотограмметрии), необходимо знать расстояние между минимум 2мя точками для нормализации сцены

Разреженная 3D реконструкция

Коллекция изображений из интернета

Трёхмерная модель

Коллекции изображений



Несколько миллионов изображений по тегу «Rome»

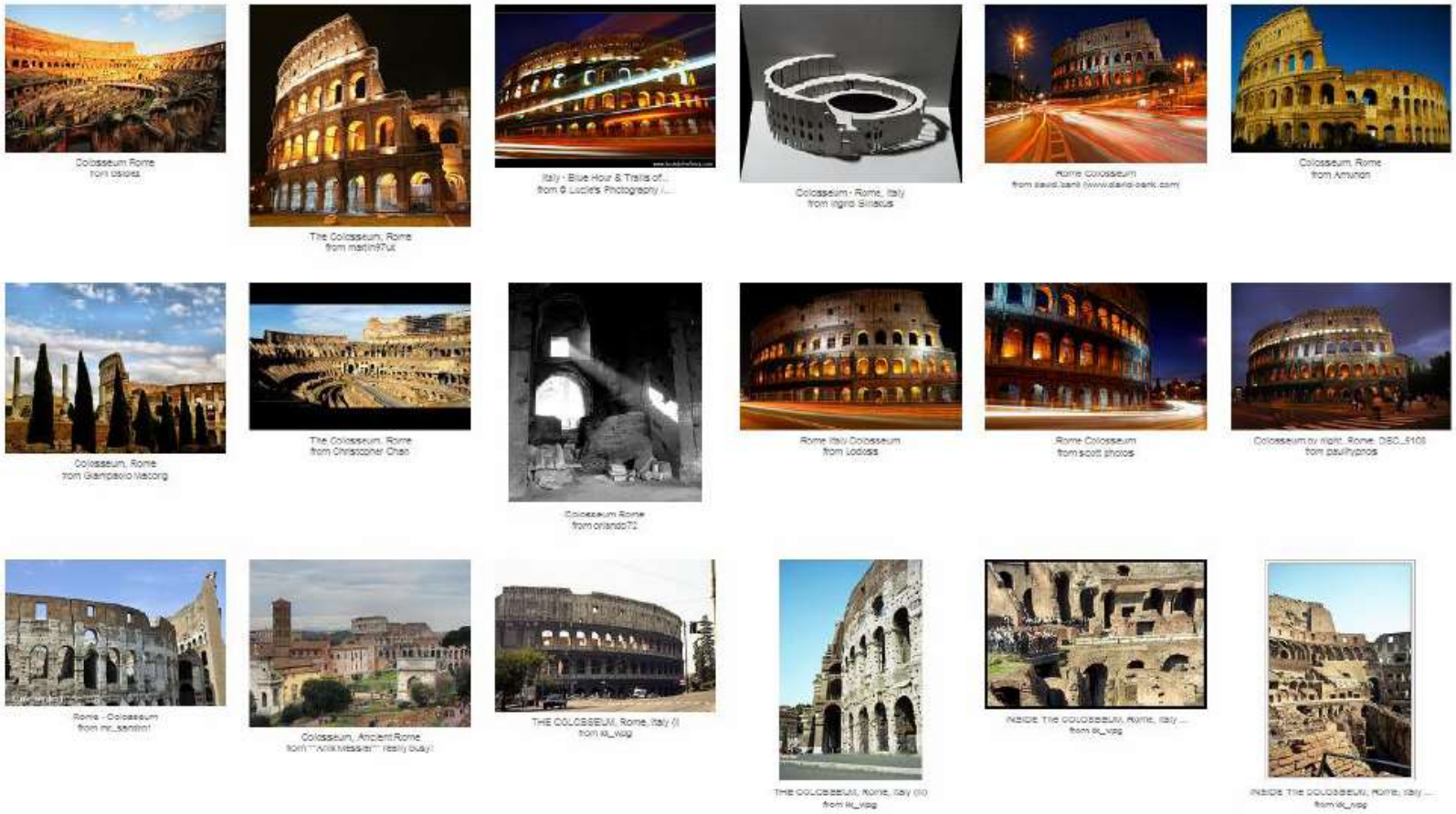
Обычные фотографии

- Получены разными пользователями с разных фотоаппаратов

iPhone 3G

Nikon D3

Обычные фотографии



- Разное время года, разное время суток
- Случайный порядок изображений

Внутренняя калибровка

```
File size     : 85111 bytes
File date     : 2005:12:16 04:17:12
Camera make   : Panasonic
Camera model  : DMC-FZ20
Date/Time     : 2005:03:19 12:52:33
Resolution    : 450 x 600
Flash used    : No
Focal length  : 6.0mm
Exposure time: 0.0012 s  (1/800)
Aperture      : f/5.6
ISO equiv.    : 80
Whitebalance  : Auto
Metering Mode: matrix
Exposure      : program (auto)
```

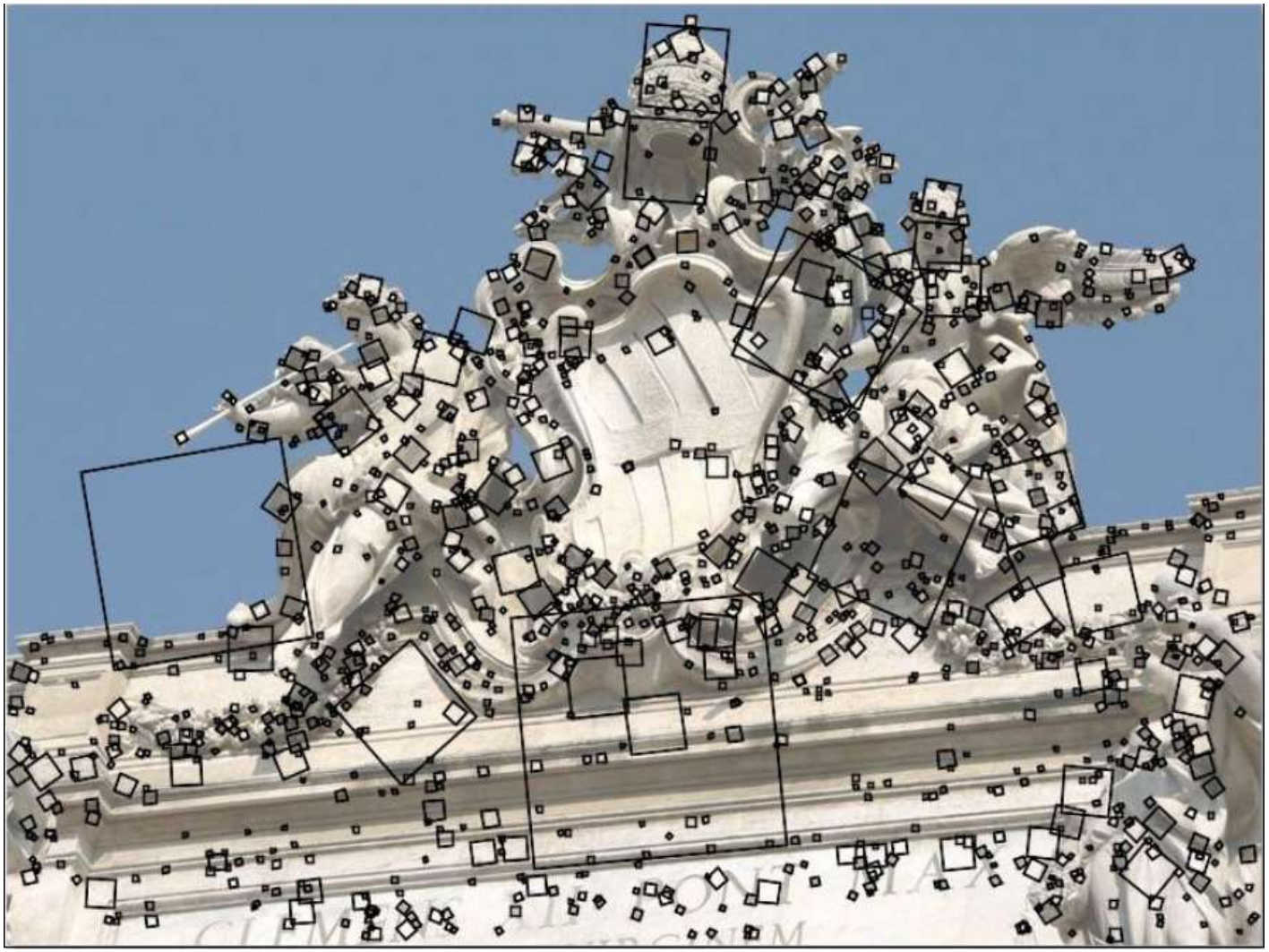
Структура EXIF-данных

Параметры камеры

Найдем соответствующие точки

Поиск особых точек (SIFT)

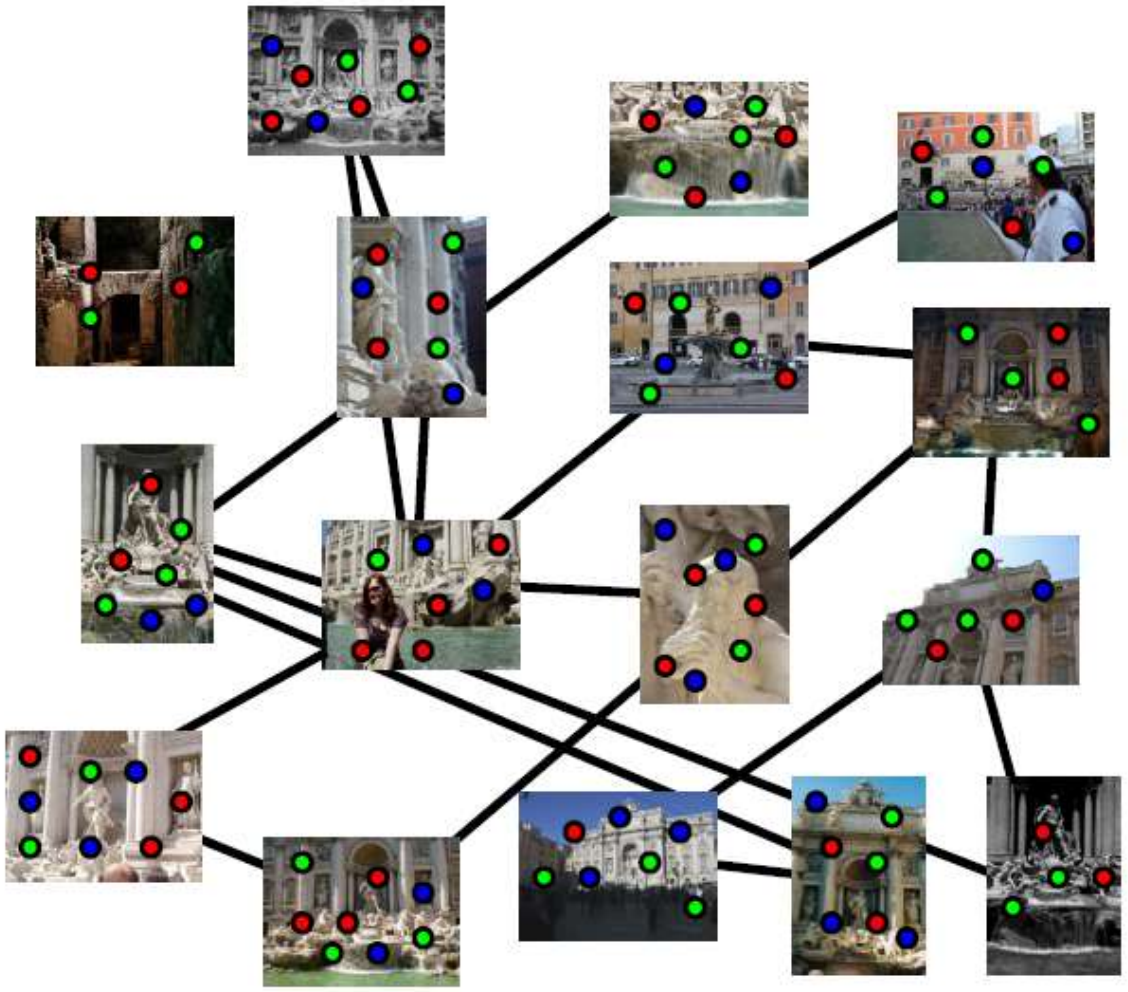
Найденные особые точки



Сопоставление изображений

- Сопоставляем точки по дескрипторам SIFT
- Фильтруем ложные соответствия по эпиполярной геометрии
- Для этого будем с помощью RANSAC вычислять фундаментальную матрицу
- Пары, между которыми много соответствий, будем считать «связанными»

Граф связанности изображений

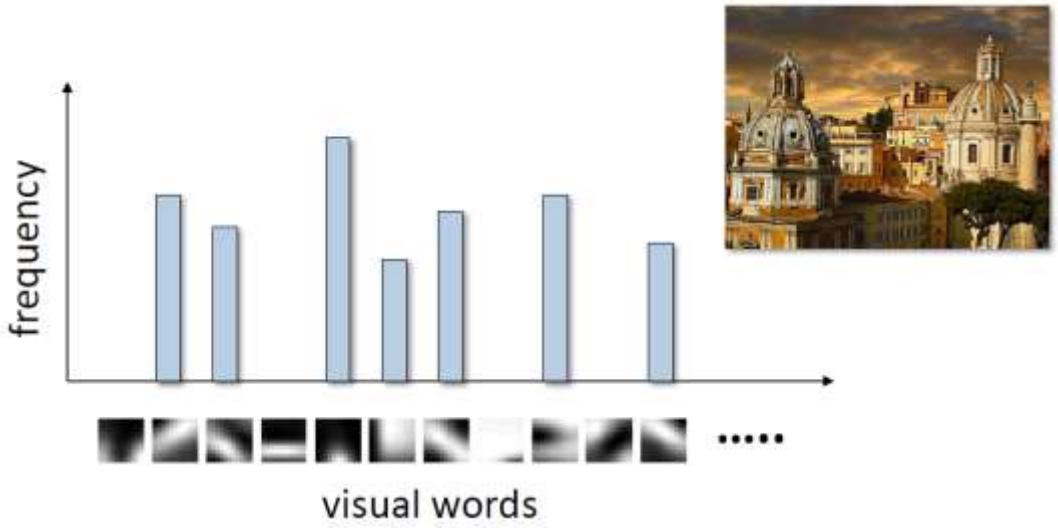


Объем задачи

- 250К изображений -> 31М пар изображений
- 2 пары изображений в секунду
  - 1 год на 500 машинах
- 1М изображений -> 500 000М пар изображений
  - 15 лет на 500 машинах

Поиск похожих изображений

- Метод «Мешок слов»
  - •Обучаем словарь визуальных слов размером 100К
  - •Строим гистограмму частот для каждого изображения:

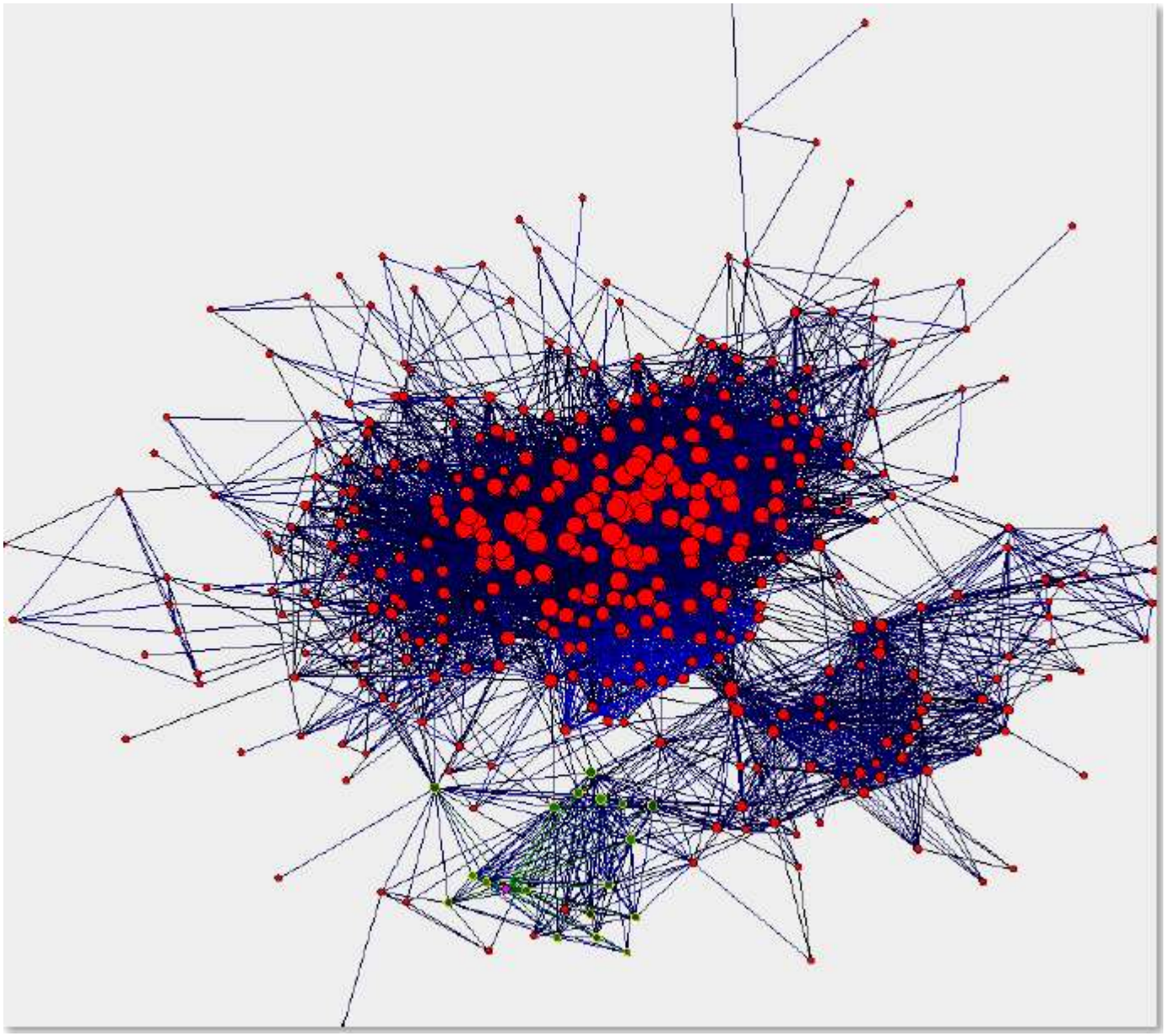


  - • Для каждого изображения ищем 40 наиболее похожих
  - • Сопоставляем точки между этими парами изображений
  - • Выбираем хорошие пары

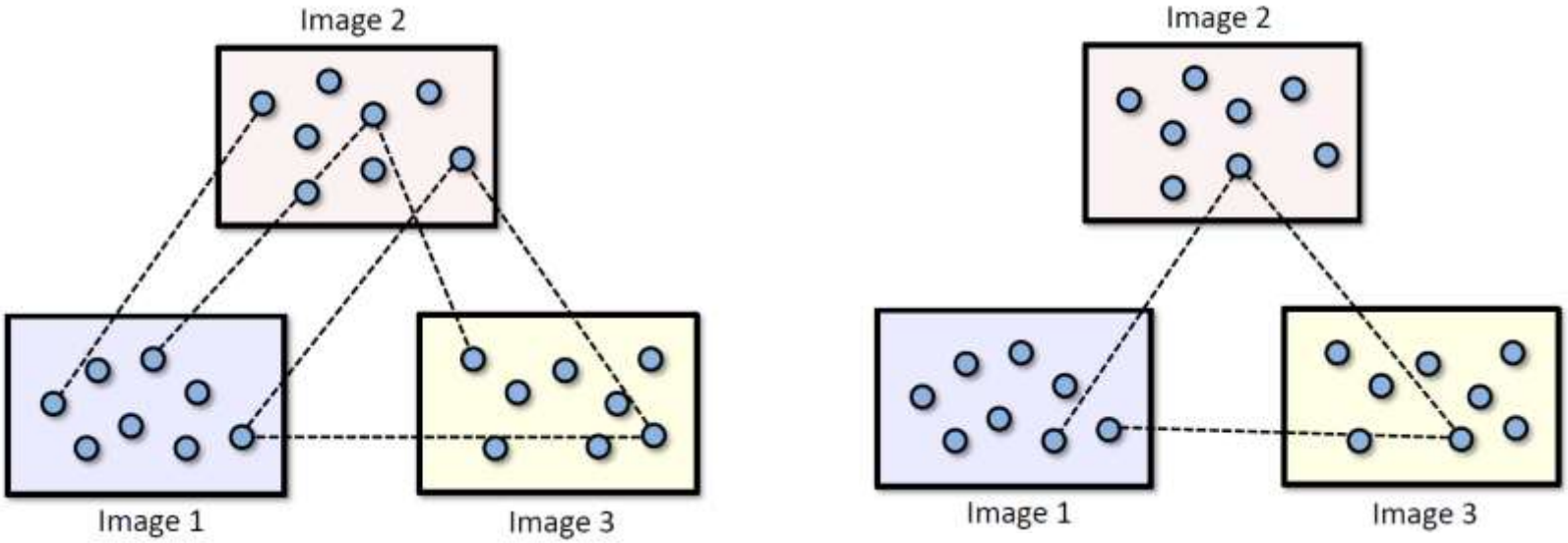
Результат сопоставлений

Мощность SIFT!

Граф связанности изображений

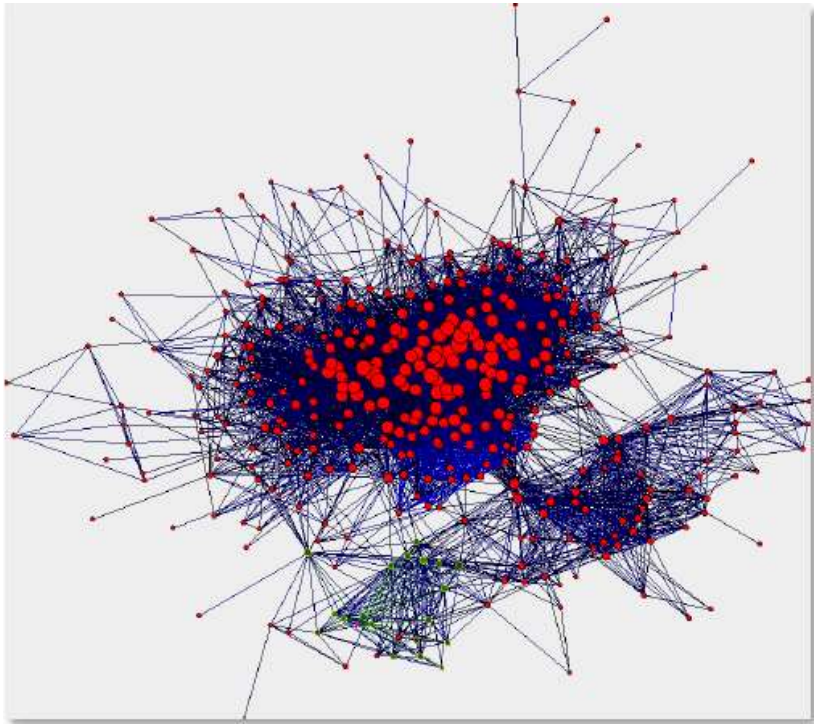


Построение «следов»

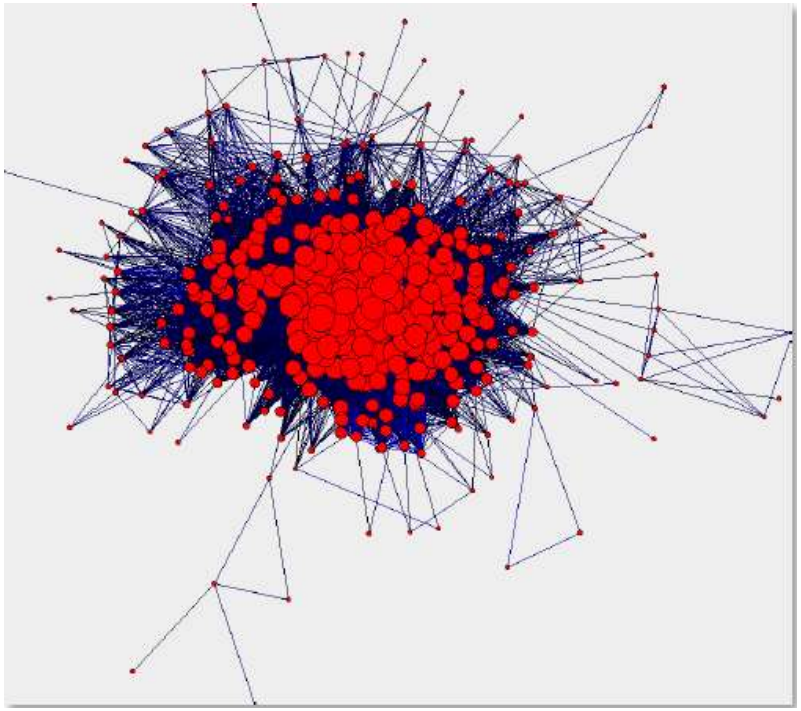


- Объединяем соответствия в «следы»
  - Отбрасываем незамкнутные следы, как потенциальные ошибки
- Пример (3000 изображений), всего 1.5M+ следов
  - 79% длины 2
  - 90% длины <=3
  - 98% длины <= 10
  - Самый длинный след 385 точек

Граф связанности изображений

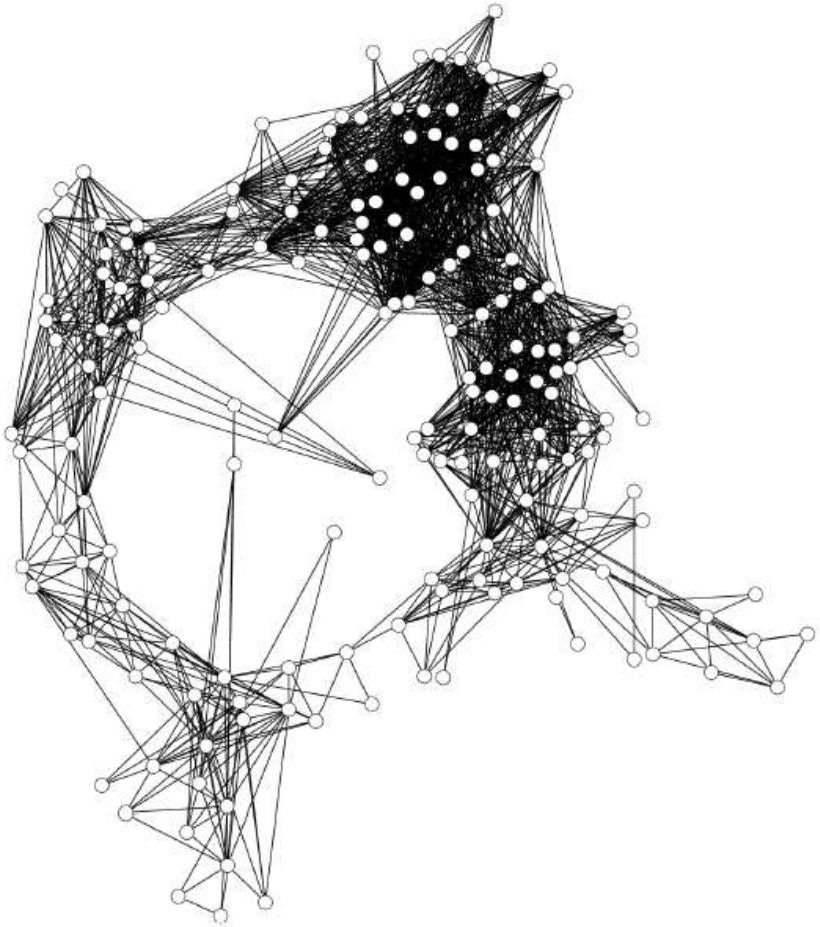


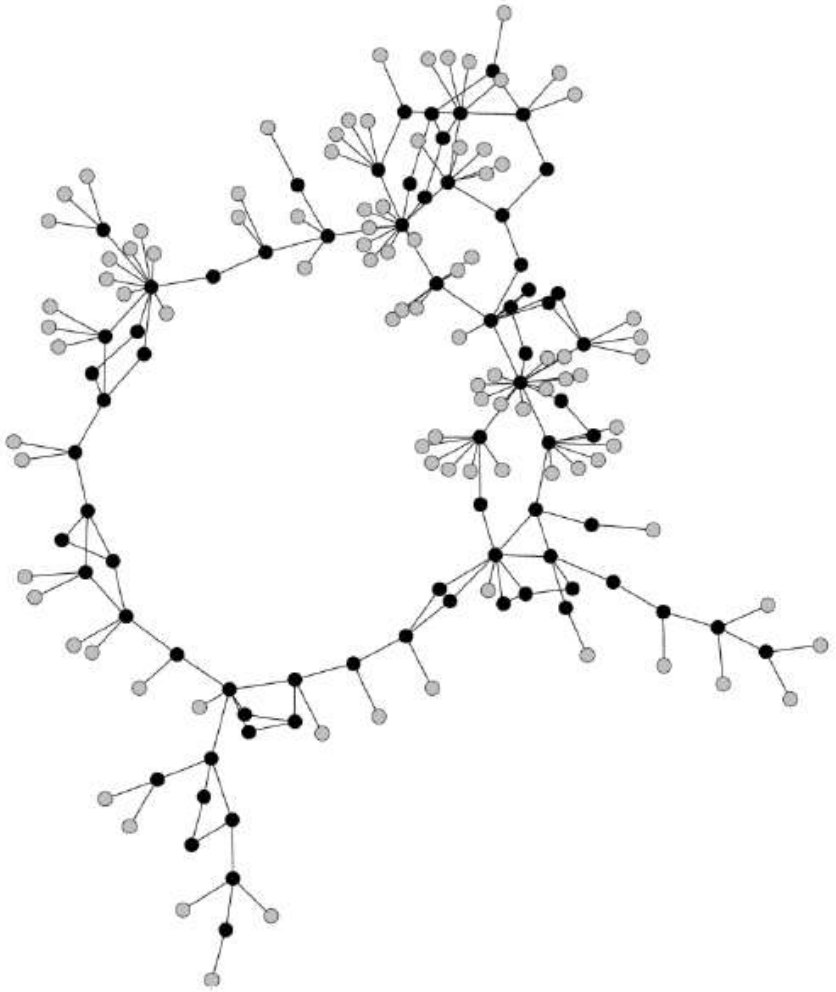
Исходные соответствия



После построения следов

«Скелет» графа





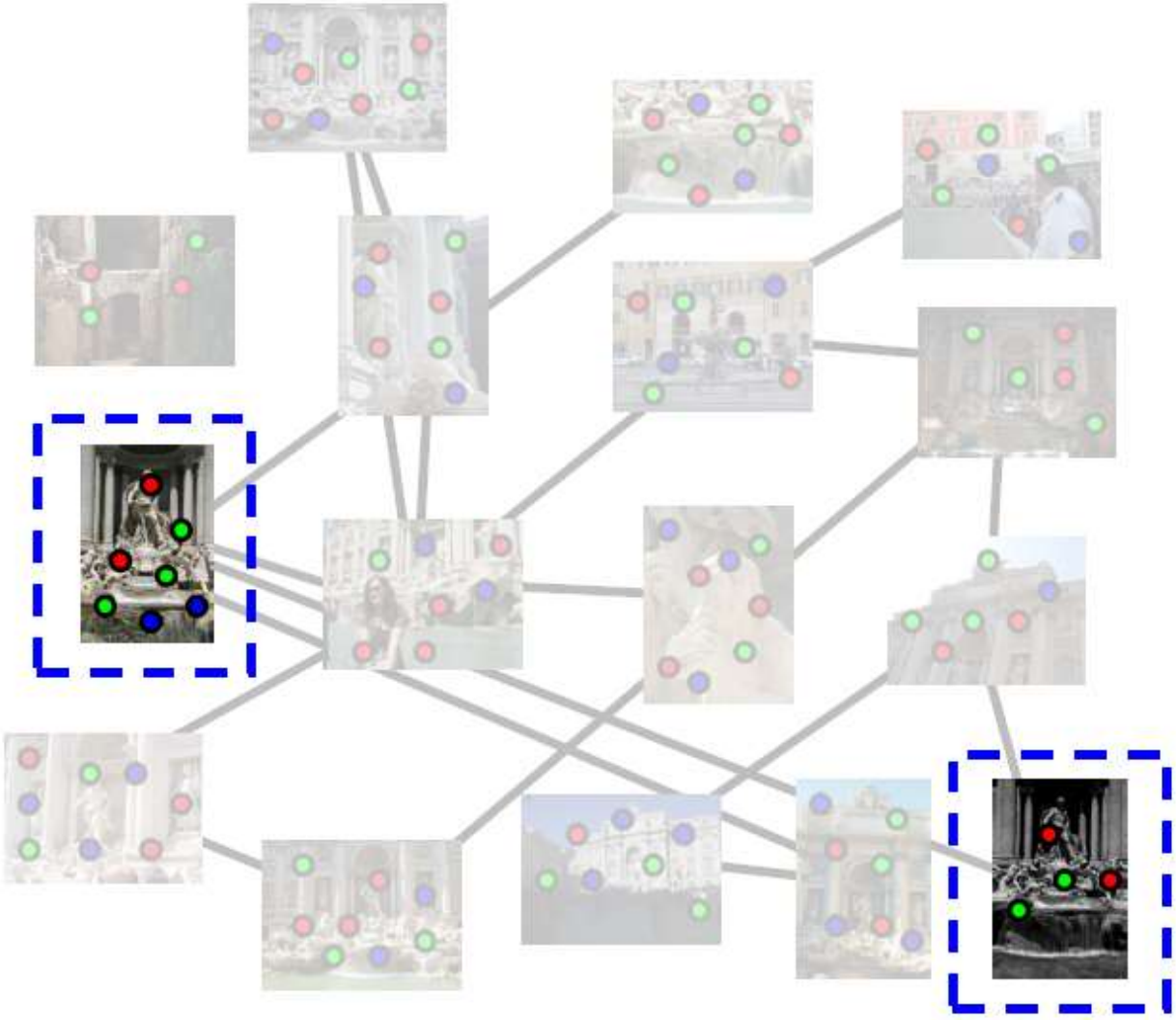
Фильтруем изображения, оставляя только «ключевые» для упрощения последующей реконструкции

[Snavely, *et al.*, “Skeletal sets for efficient structure from motion,” CVPR 2008]

Стандартная схема

- Выбор опорных пар изображений
- Вычисление структуры по паре изображений
  - Построение матрицы внутренней калибровки
  - Вычисление фундаментальной и существенной матрицы
  - Триангуляция точек
  - Уточнение методом связок
- Добавление новых изображений
  - Калибровка камеры по известным 2D/3D соответствиям
  - Триангуляция новых точек
  - Уточнение методом связок

Выбор опорной пары



Выбор опорной пары

(+) много точек

(-) малая база

(+) большая база

(-) мало точек

(+) много точек

(+) большая база

Выбор опорной пары

- Возможны разные эвристики
- Гомография плохо описывает сцену, когда сцена не плоская, и существенный параллакс (большая база)
- Пример:
  - Возьмем пары, в которых > 100 соответствий
  - Вычислим гомографию и фундаментальную матрицу
  - Выберем пару изображений, минимизирующее отношение:

$$\frac{\mathrm{numInliers}_{\mathrm{homography}}}{\mathrm{numInliers}_{\mathrm{Fmatrix}}}$$

Примеры реконструкций

Примеры реконструкций

Дубровник

BigSFM

BigSFM: Reconstructing the World from Internet Photos

Our group is working on building a 3D model of the world from online photo collections, and our research spans several areas, from image features, to large-scale image matching, to structure-from-motion optimization, to applications such as location recognition. This page summarizes our work, has links to **code** and **datasets** we have made available, and has a description of each **project**.

Cornell Participants (and alumni): Noah Snavely (Cornell), David Crandall (Indiana University), Daniel Hauagge (Cornell), Kyle Wilson (Cornell), Song Cao (Cornell), Yin Lou (Cornell), Yunpeng Li (EPFL), Andrew Owens (MIT), Johannes Gehrke (Cornell) Dan Huttenlocher (Cornell)

Other Collaborators: Sameer Agarwal (University of Washington and Google), Brian Curless (UW), Yasutaka Furukawa (Google), Steve Seitz (UW), Ian Simon (UW), Rick Szeliski (Microsoft Research)

We gratefully acknowledge the support of the following sponsors:

National Science Foundation

Amazon Web Services in Education

MIT Lincoln Labs

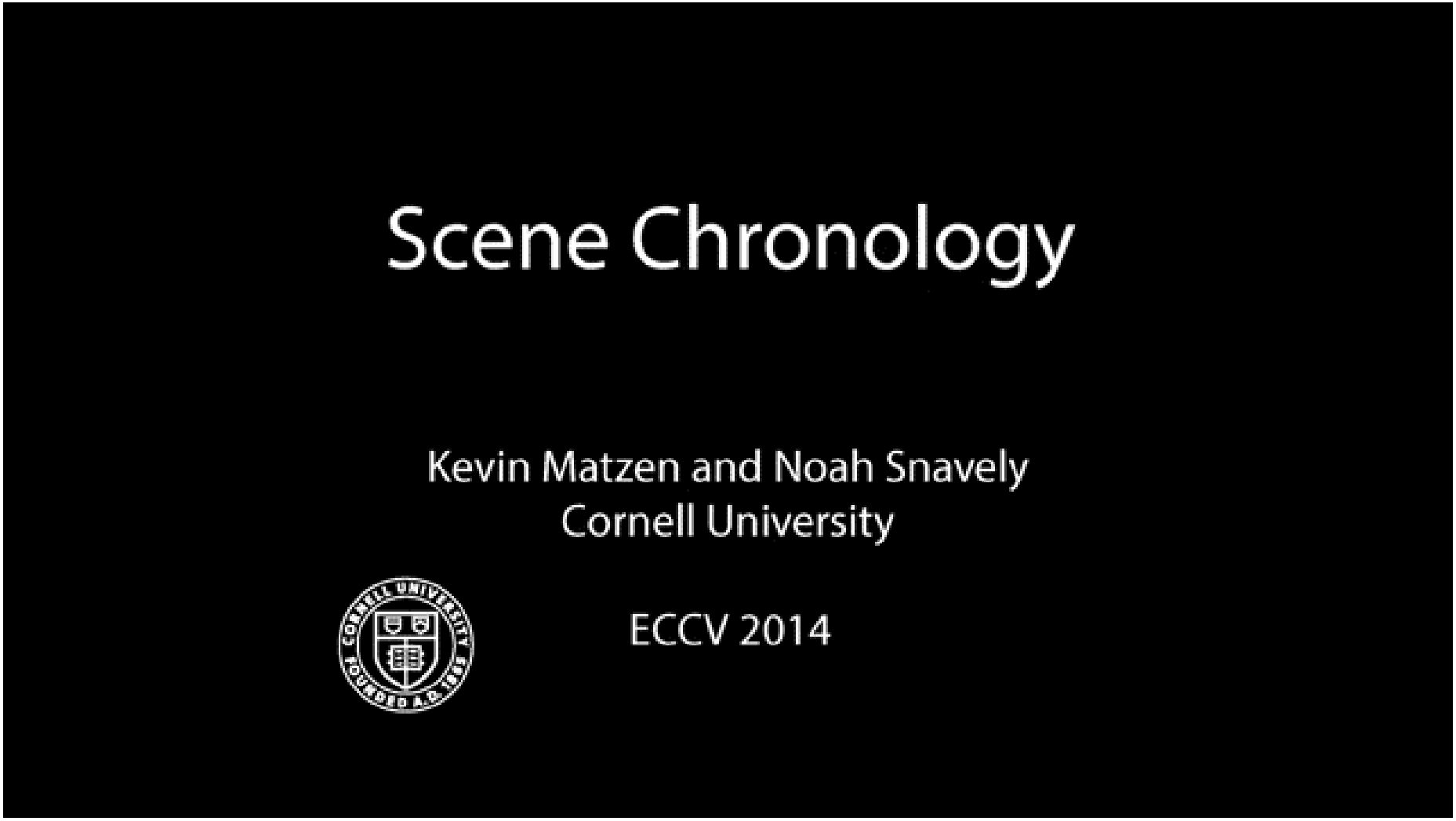
Microsoft

IU Data to Insight Center

http://www.cs.cornell.edu/projects/bigsfm/

Глобальное позиционирование

Визуальная хронология



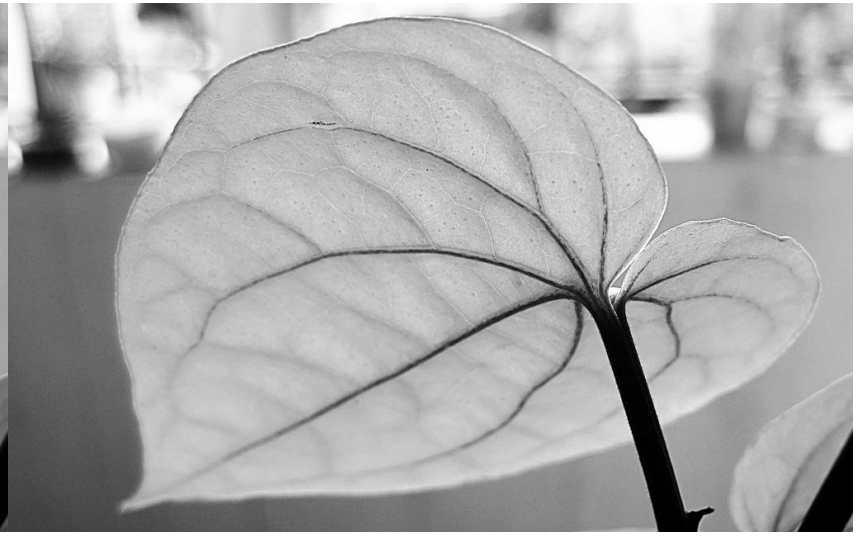
Резюме

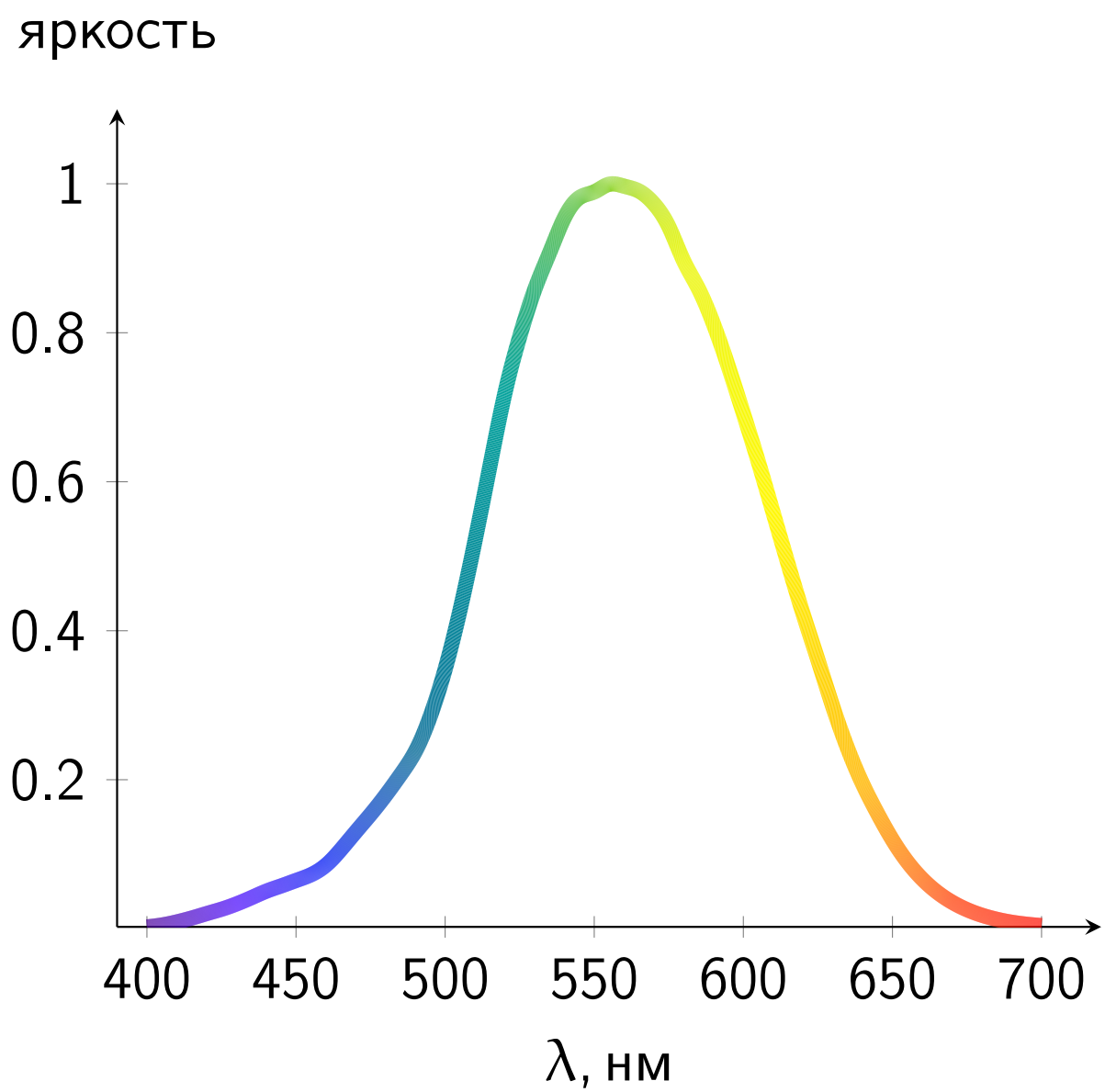
- Разреженная трёхмерная реконструкция – один из наиболее проработанных разделов компьютерного зрения
- Умеем:
  - Калибровать камеру по шаблону
  - Определять взаимное расположение камер
  - Триангулировать точки
  - Определять положение камеры относительно объекта по 2D-3D соответствиям
  - Сопоставлять изображения на больших коллекциях
- Направление развития:
  - Использование нейросетевых методов для решения отдельных подзадач и задачи в целом

Преобразование Фурье. JPEG

Влад Шахуро

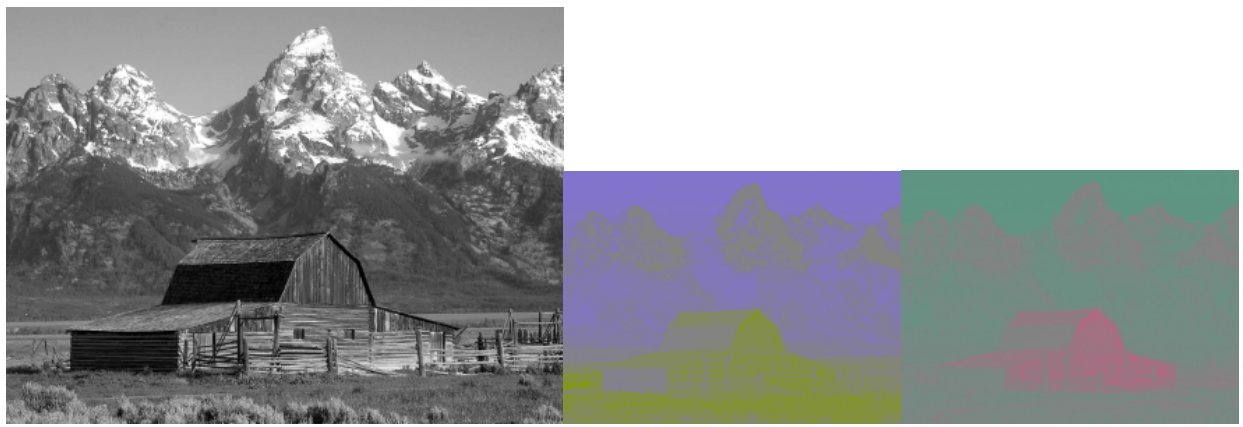
16 сентября 2019 г.




яркость
1
0.8
0.6
0.4
0.2
400
450
500
550
600
650
700
λ, нм

RGB → YCbCr

$$
\begin{aligned}
Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\
C_b &= -0.169 \cdot R - 0.331 \cdot G + 0.500 \cdot B + 128 \\
C_r &= 0.500 \cdot R - 0.419 \cdot G - 0.081 \cdot B + 128
\end{aligned}
$$

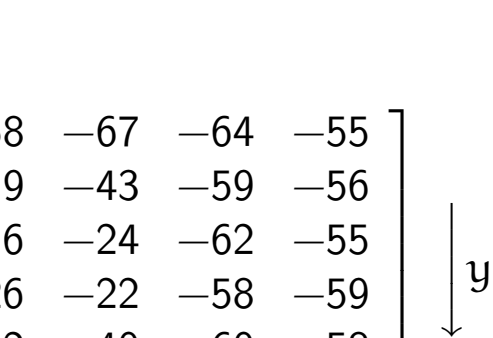


Сдвиг на 128

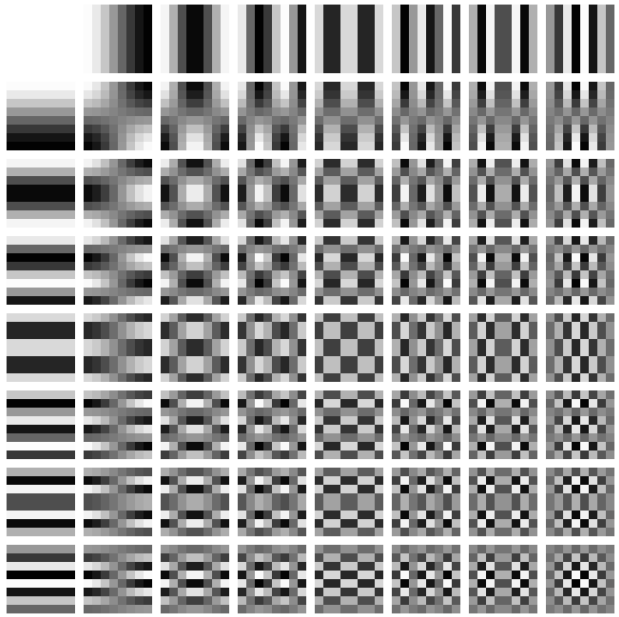
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

исходное изображение

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$



Дискретное косинусное преобразование



u

$\longrightarrow$

$$\begin{bmatrix}
-415.3 & -30.1 & -61.2 & 27.2 & 56.1 & -20.1 & -2.3 & 0.4 \\
4.4 & -21.8 & -60.7 & 10.2 & 13.1 & -7.0 & -8.5 & 4.8 \\
-46.8 & 7.3 & 77.1 & -24.5 & -28.9 & 9.9 & 5.4 & -5.6 \\
-48.5 & 12.0 & 34.1 & -14.7 & -10.2 & 6.3 & 1.8 & 1.9 \\
12.1 & -6.5 & -13.2 & -3.9 & -1.8 & 1.7 & -2.7 & 3.1 \\
-7.7 & 2.9 & 2.3 & -5.9 & -2.3 & 0.9 & 4.3 & 1.8 \\
-1.0 & 0.1 & 0.4 & -2.4 & -0.8 & -3.0 & 4.1 & -0.6 \\
-0.1 & 0.1 & -1.0 & -4.1 & -1.1 & -0.1 & 0.5 & 1.6
\end{bmatrix} \quad \downarrow v$$

амплитуды

Квантование амплитуд

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

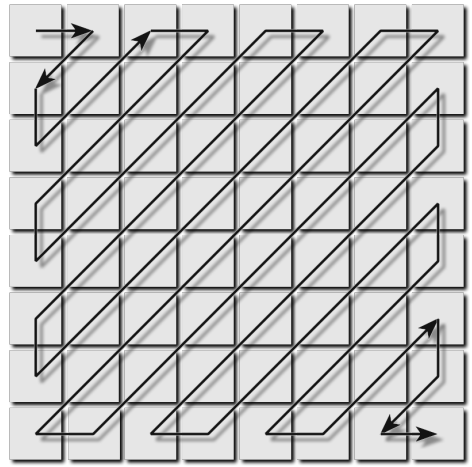
матрица квантования

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

квантованные амплитуды

Сжатие квантованных амплитуд

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Кодирование серий (run-length encoding), затем кодирование Хаффмана

Алгоритм JPEG

1. Конвертируем изображение в пространство YCbCr
2. Уменьшаем разрешение цветовых компонент в 2 раза
3. Делим изображение на блоки 8×8, вычитаем 128
4. Для каждого блока:
   - 4.1 применяем дискретное косинусное преобразование
   - 4.2 квантуем коэффициенты (амплитуды)
   - 4.3 сжимаем квантованные коэффициенты

Преобразование Фурье

$$\underbrace{G(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} g(x) e^{-i\omega x} dx}_{\text{прямое преобразование Фурье, } \mathcal{F}}$$

$$\underbrace{g(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} G(\omega) e^{i\omega x} d\omega}_{\text{обратное преобразование Фурье, } \mathcal{F}^{-1}}$$

Дискретное преобразование Фурье

$$G(m) = \frac{1}{\sqrt{M}} \sum_{u=0}^{M-1} g(u) e^{-i2\pi \frac{mu}{M}}, \quad 0 \leqslant m < M$$

$$g(u) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} G(m) e^{i2\pi \frac{mu}{M}}, \quad 0 \leqslant u < M$$

Дискретное преобразование Фурье

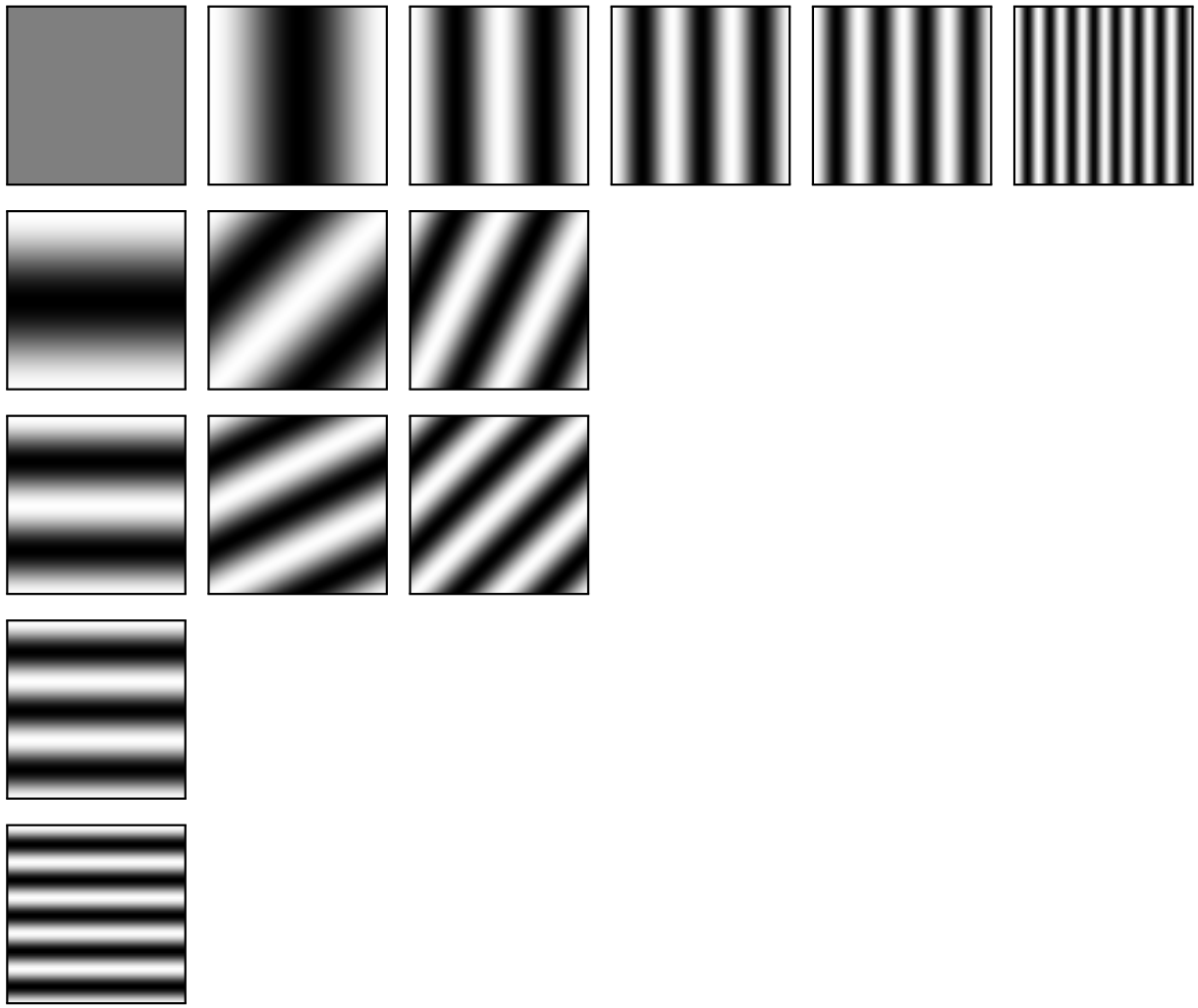
$$G(m) = \frac{1}{\sqrt{M}} \sum_{u=0}^{M-1} g(u) e^{-i2\pi \frac{mu}{M}}, \quad 0 \leqslant m < M$$

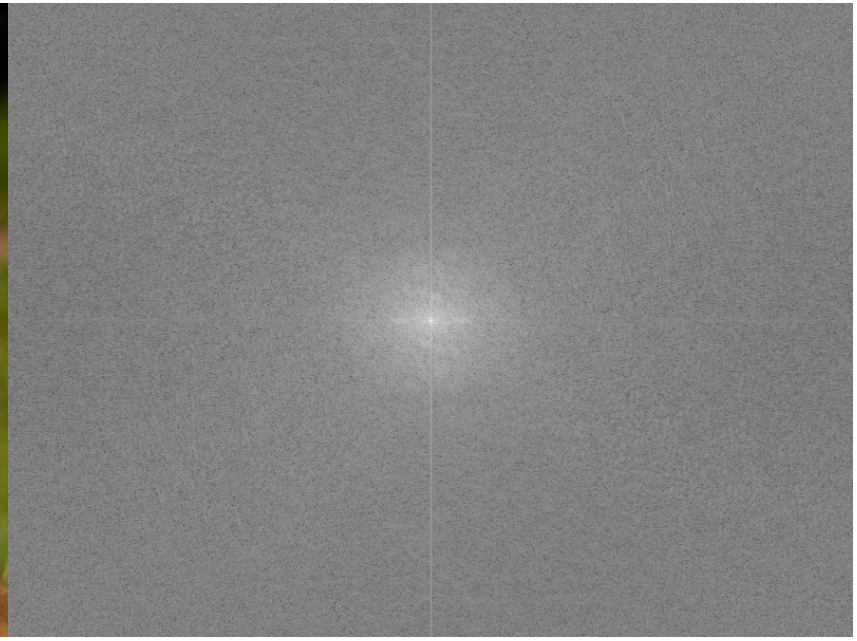
$$g(u) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} G(m) e^{i2\pi \frac{mu}{M}}, \quad 0 \leqslant u < M$$

Наивная реализация: $O(M^2)$

Быстрая реалиация: $O(M \log M)$

Базис Фурье для изображений

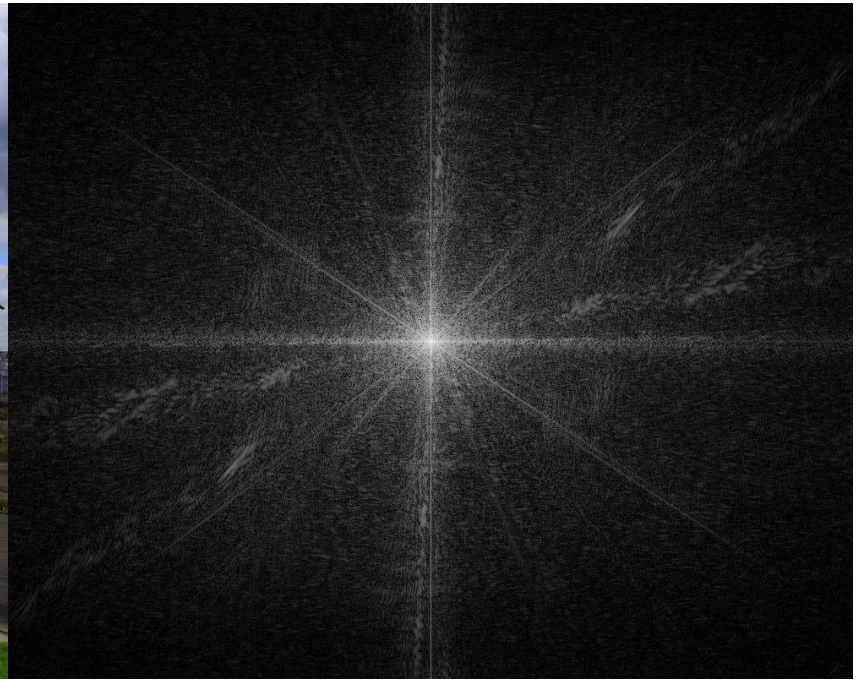




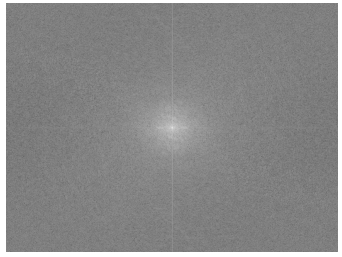
пространственный домен

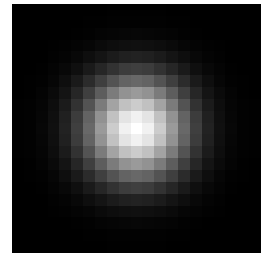
частотный домен

```
freq = log(1 + abs(fftshift(fft2(img))))
```

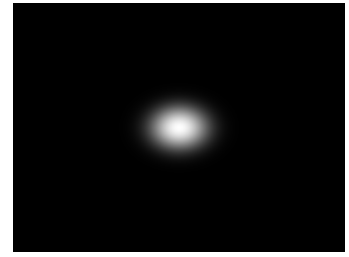


$\downarrow \mathcal{F}$



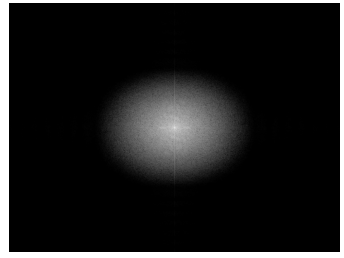


$\downarrow \mathcal{F}$



$*$

$\downarrow$



$\downarrow \mathcal{F}^{-1}$

Теорема о свертке

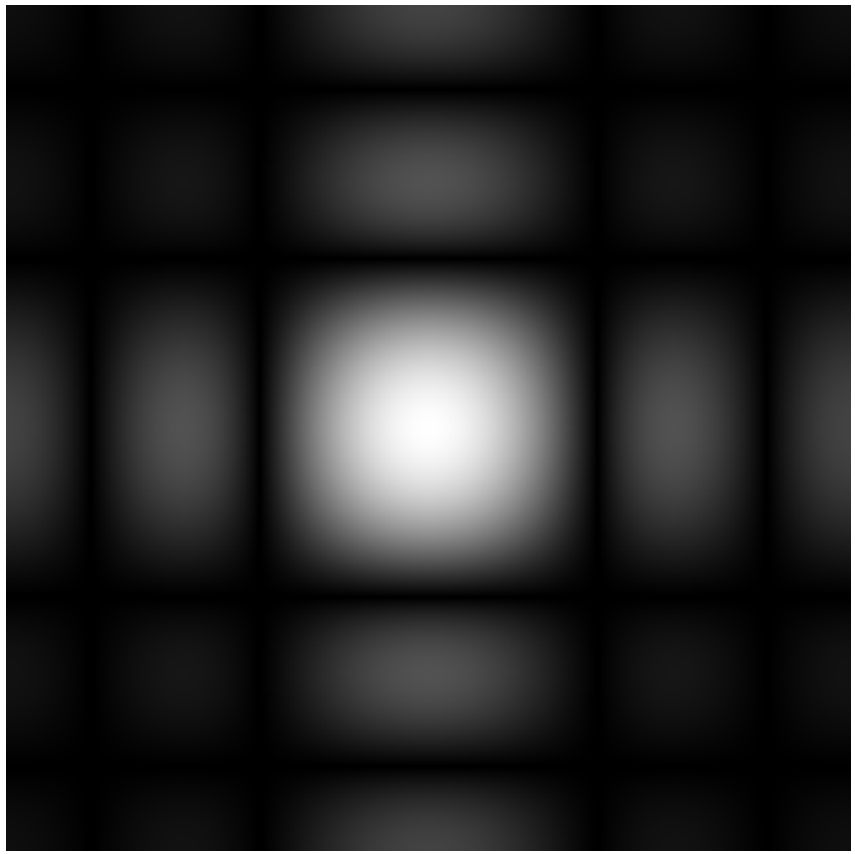
Преобразование Фурье свертки двух функций — это произведение преобразований Фурье этих функций

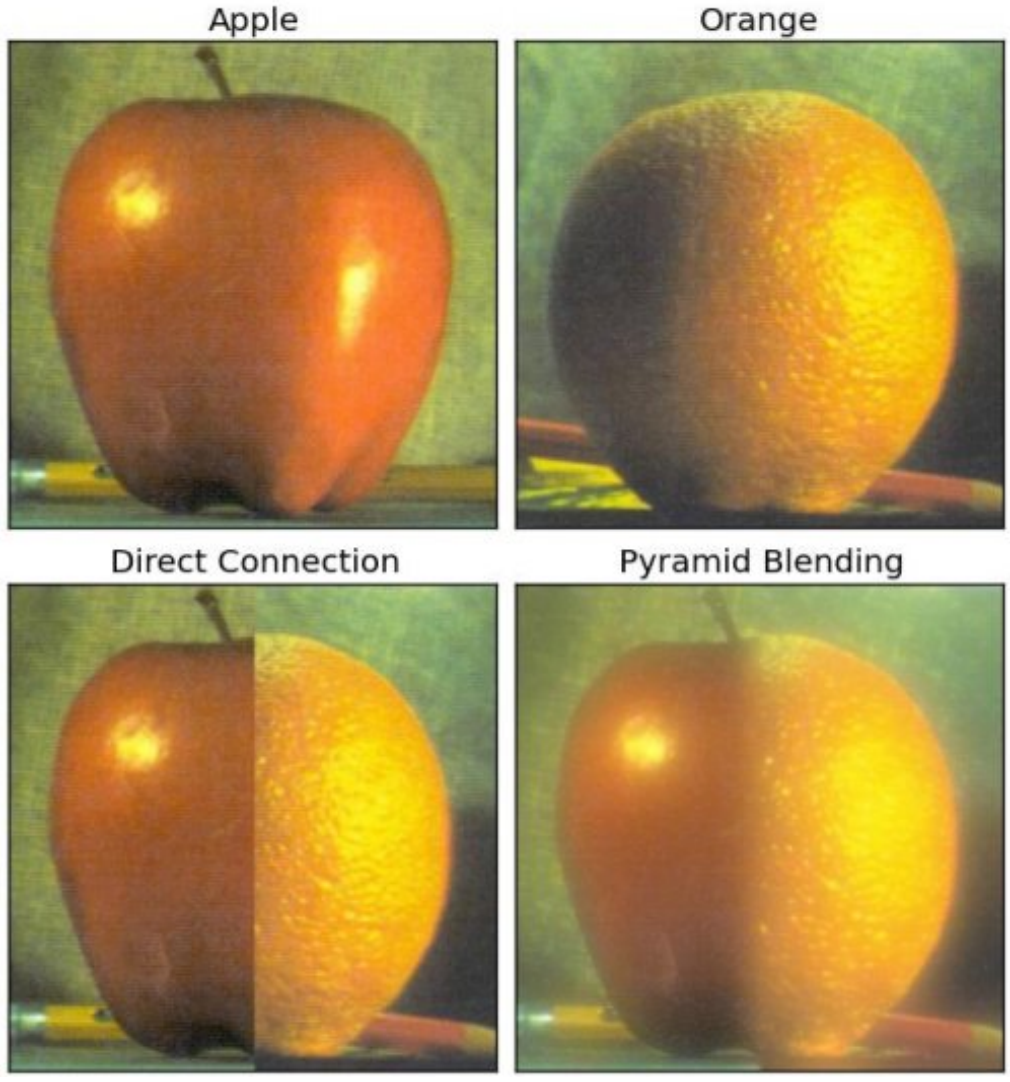
$$\mathcal{F}[\mathrm{g} * \mathrm{h}] = \mathcal{F}[\mathrm{g}] \cdot \mathcal{F}[\mathrm{h}],$$

т.е. свертка эквивалента произведению частот:

$$\mathrm{g} * \mathrm{h} = \mathcal{F}^{-1}[\mathcal{F}[\mathrm{g}] \cdot \mathcal{F}[\mathrm{h}]]$$

Фурье-образ box фильтра




Apple
Orange
Direct Connection
Pyramid Blending

Лапласовская пирамида

Вход:

Изображения A, B, маска M

Алгоритм склейки:

1. Строим лапласовские пирамиды LA и LB из изображений A и B
2. Строим гауссовскую пирамиду GM
3. Комбинируем лапласовские пирамиды: $\mathrm{LS} = \mathrm{GM} * \mathrm{LA} + (1 - \mathrm{GM}) * \mathrm{LB}$
4. Строим результат из пирамиды LS

Резюме

Алгоритм сжатия изображений JPEG работает на основе дискретного косинусного преобразования, квантования и сжатия амплитуд

Преобразование Фурье — мощный инструмент анализа изображений и поведения фильтров

С помощью Фурье-анализа можно теоретически обосновать применимость лапласовской пирамиды для фотомонтажа