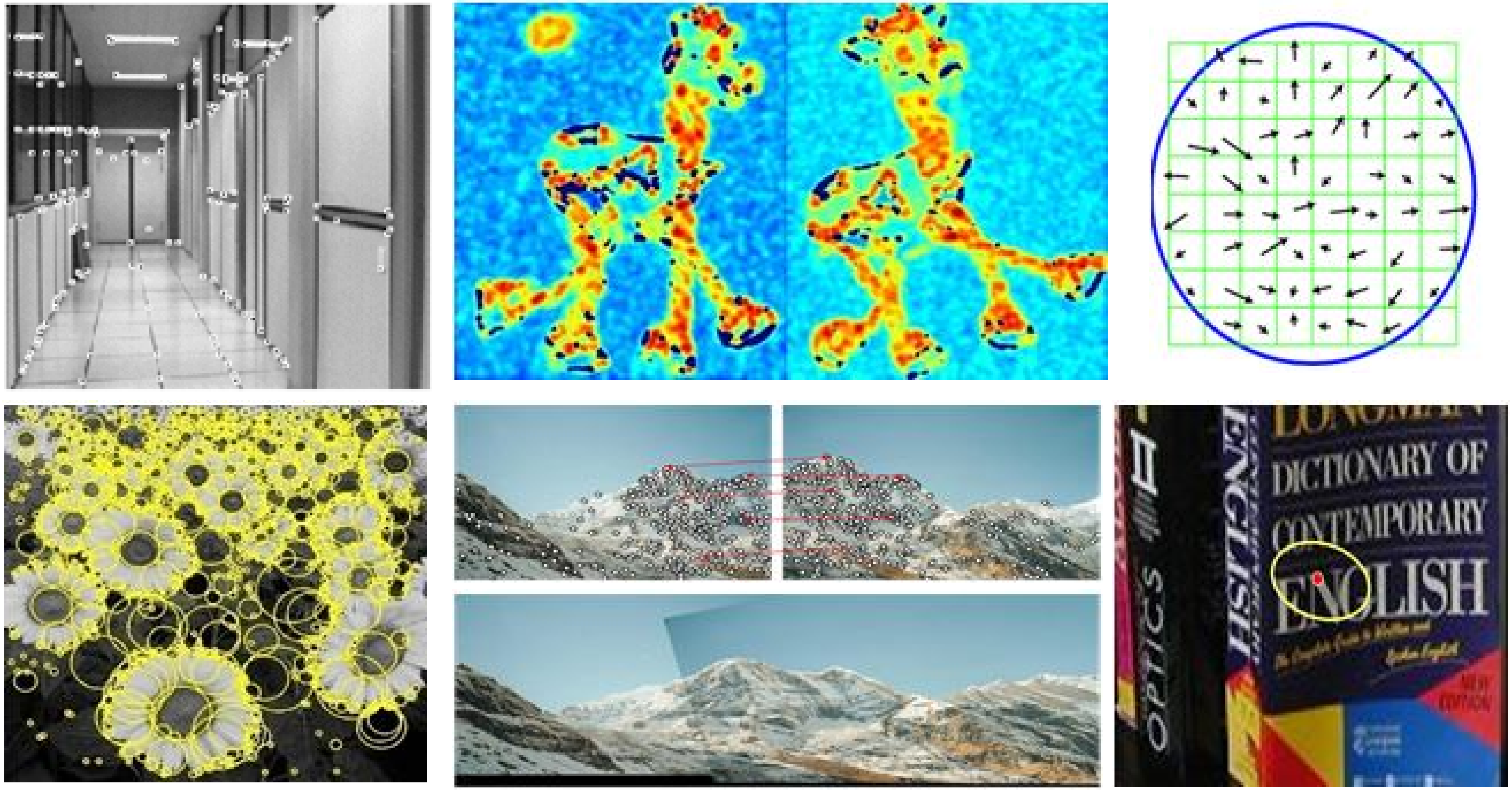
# Сопоставление изображений и локальные особенности

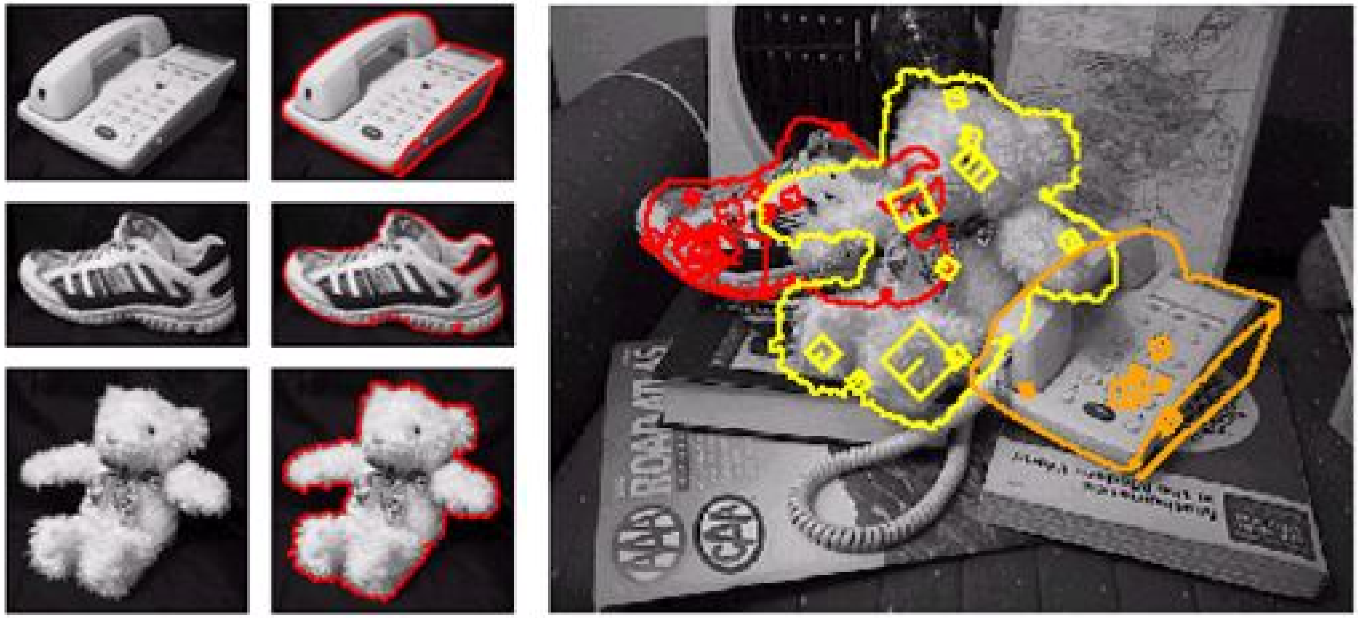



Антон Конушин
http://courses.graphicon.ru/main/vision

Many slides adopted from Svetlana Lazebnik, Steve Seitz and Alexey Efros

# Задача сопоставления изображений

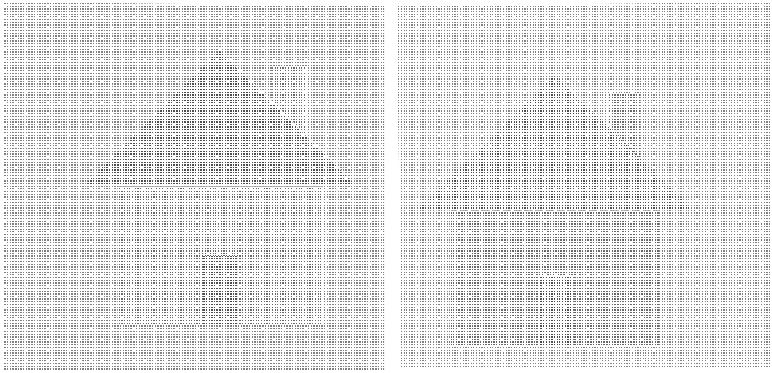
Построение панорам



Распознавание экземпляров объектов

«Image alignment», «Image matching»

# Выравнивание изображений



Есть два изображения одного и того же объекта.

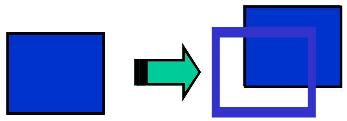
Как нам совместить изображения автоматически?

- Найдём такое преобразование (совмещение изображений), при котором изображения больше всего совпадут
  - Прямое согласование (direct alignment)

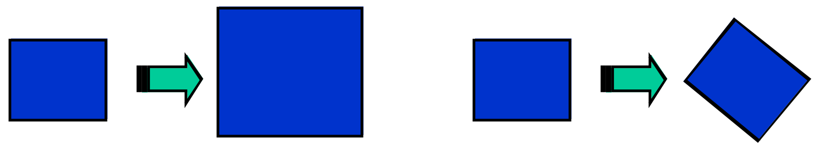
- Что нам нужно определить:
  - Какое преобразование будем использовать?
  - Как оценить совпадение (похожесть изображений)?

# Геометрические преобразования

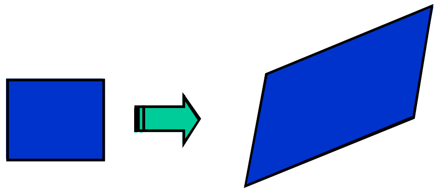
- Параллельный перенос



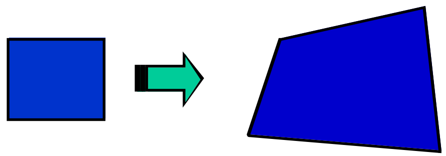
- Подобие (перенос, масштаб, поворот)



- Афинное



- Проективное (гомография)



# Геометрические преобразования

- Параллельный перенос

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

- Евклидово преобразование (M – ортогональная матрица)

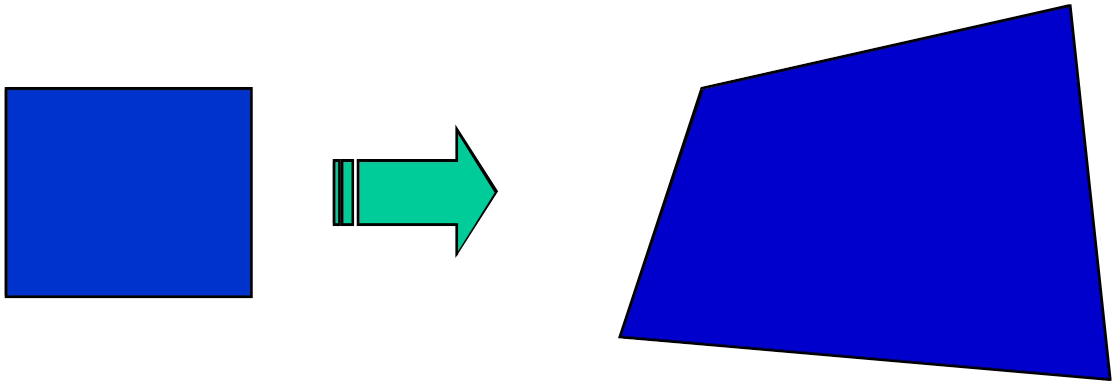
$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

- Аффинное преобразование

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

# Гомография

- Перспективное преобразование плоскости
- Переводит четырехугольник в другой произвольный четырехугольник

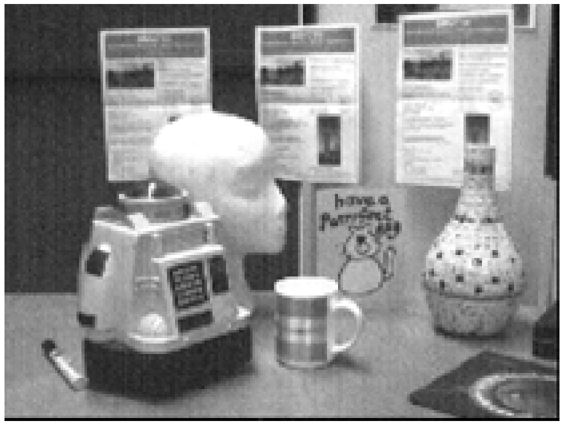


Slide by S. Lazebnik

# Гомография

- Преобразование между 2мя разными видами одной и той же плоскости


- Преобразование между видами с повернутой камеры (центр проекции общий)

Slide by S. Lazebnik

# Однородные координаты

- Однородные координаты

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Перевод в однородные координаты

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

Перевод из однородных координат

Удобнее представлять себе так:

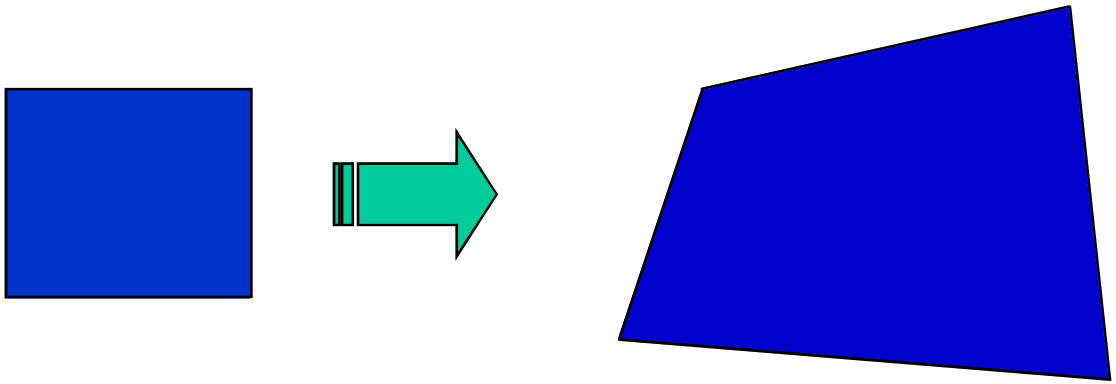
$$\begin{bmatrix} wx & wy & w \end{bmatrix}^T \cong \begin{bmatrix} x & y & 1 \end{bmatrix}^T \cong \begin{bmatrix} x & y \end{bmatrix}^T$$

Подробнее будет в лекциях про геометрические свойства камер!

# Уравнение гомографии

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \qquad \lambda x' = Hx$$

Гомография – линейное преобразование в однородных координатах



# Сравнение изображений

Попиксельное сравнение изображений:

$$\sum_X \sum_Y |I_1(X,Y) - I_2(X,Y)|$$

L1 метрика (SAD - Sum of absolute differences)

$$\sum_X \sum_Y (I_1(X,Y) - I_2(X,Y))^2$$

L2 метрика (SSD - Sum of squared differences)

$$\sum_X \sum_Y I_1(X,Y)\, I_2(X,Y)$$

Кросс-корреляция (CC - Cross-correlation)

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

# Алгоритм

Простейший подход – «грубой силы» (brute force)

- Выберем модель преобразования, определим набор параметров, его описывающих
- Выберем функцию сопоставления изображений
  - SSD, Нормализованная корреляция, и т.д.
- Переберём всевозможные значения параметров в разумных пределах:

Пример – параллельный перенос:

```
for tx=x0:step:x1,
  for ty=y0:step:y1,
     compare image1(x,y) to image2(x+tx,y+ty)
  end;
end;
```

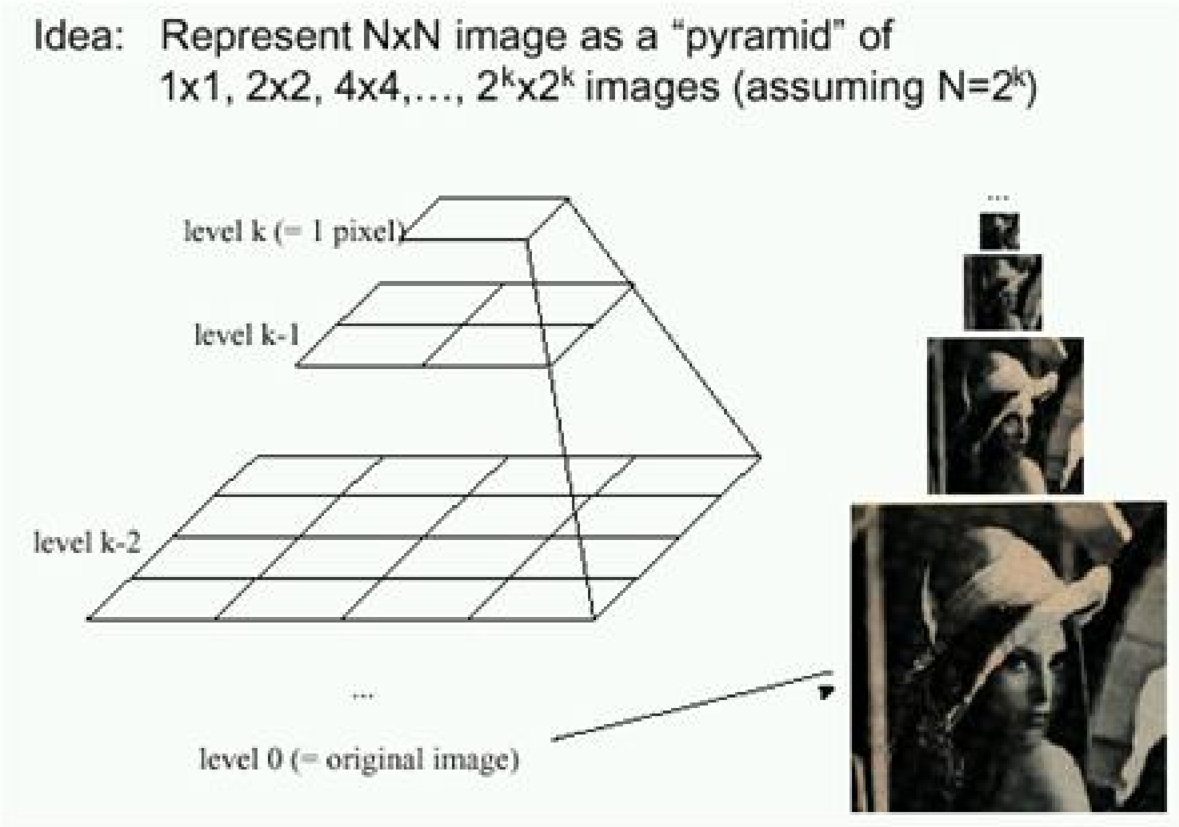
Необходимо заранее выбрать `x0,x1` и `step`

Что мы ещё не учли?

# Задача оптимизации параметров

- Фактически, сопоставление изображений, это задача поиска минимума функции сравнения изображений
- Мы рассмотрели вариант полного перебора параметров с некоторым шагом
  - Grid search
- Можем воспользоваться каким-нибудь другим методом оптимизации, например, градиентным спуском
  - Хорошо сработает только в случае точного начального приближения
  - Ошибка менее 2х пикселей
- Можно улучшить с помощью многомасштабного подхода

# Пирамиды изображений

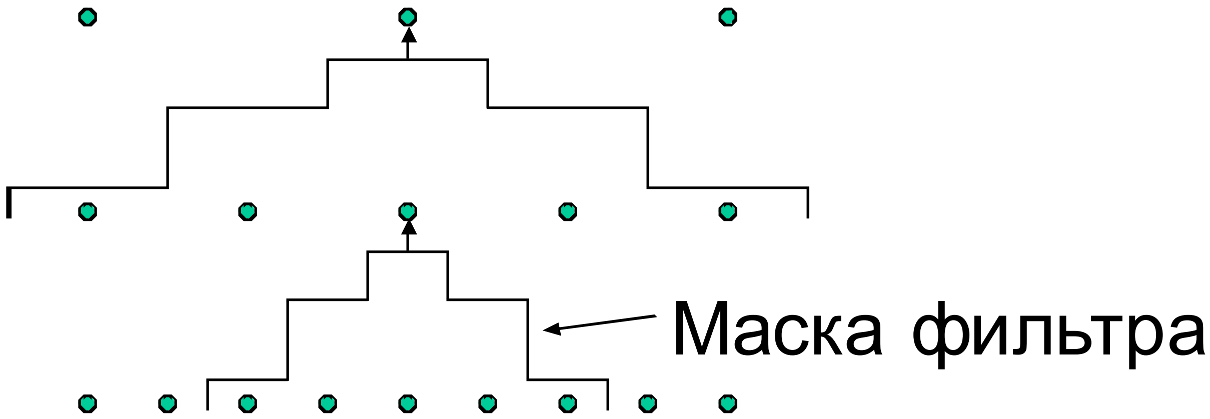



- Известна как Пирамида Гауссиан
- В компьютерной графике – “mip map” [Williams, 1983]

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

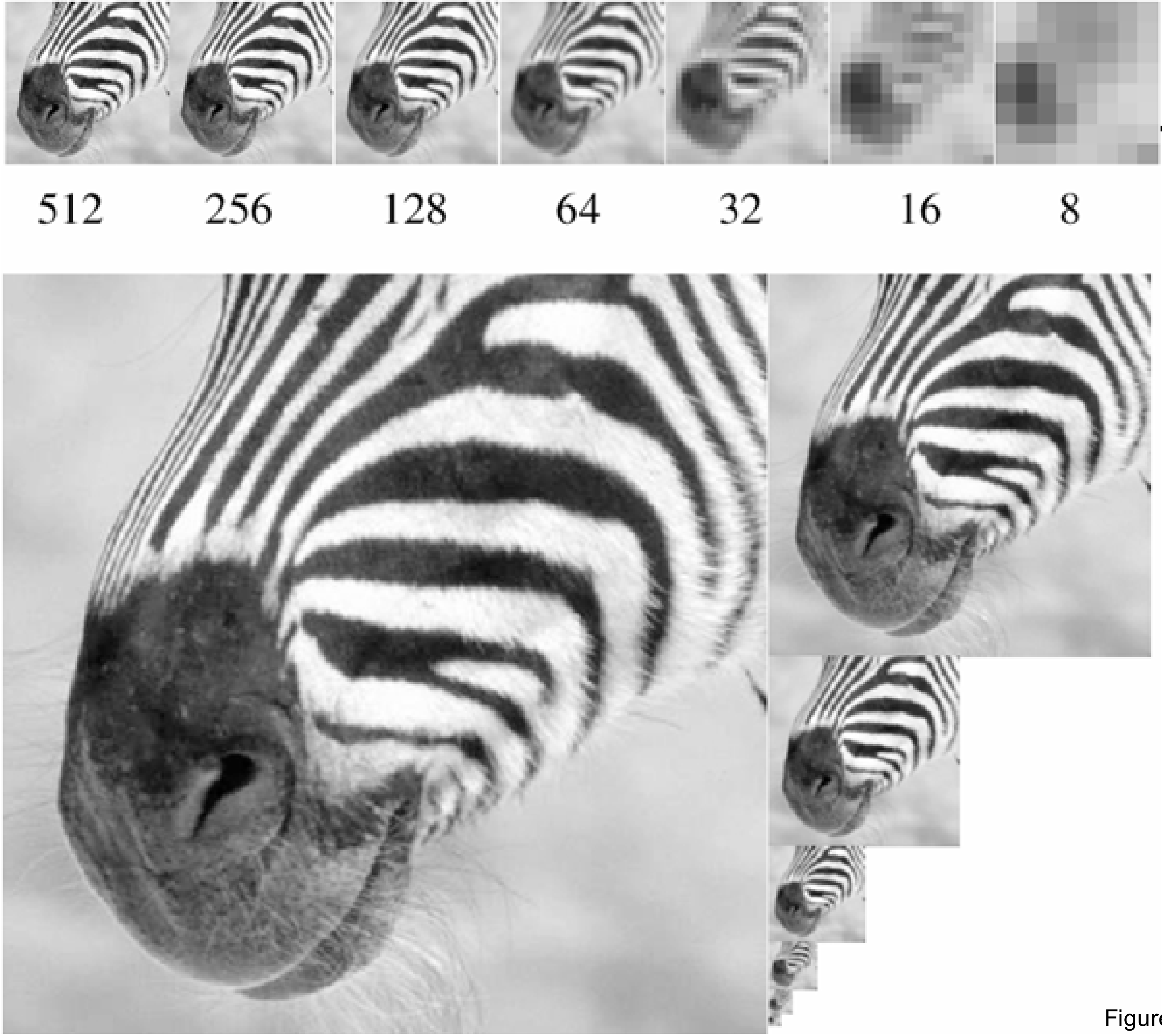
Slide by Steve Seitz

# Построение пирамиды




- Повторяем до достижения минимального разрешения
  - Сглаживаем с помощью фильтра Гаусса текущее изображение
  - Сэмплируем – берём каждый k-ый пиксель (обычно, каждый 2ой)

Slide by Steve Seitz




Microsoft Research

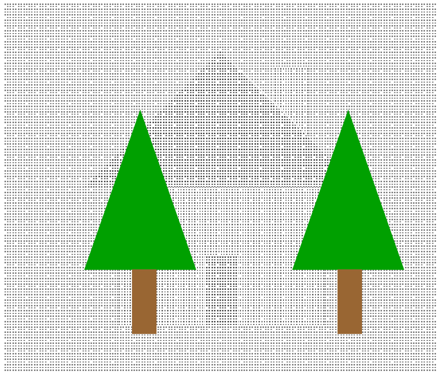
На высшем уровне полоса в пиксель – волос, на среднем – полоска, на нижнем - нос

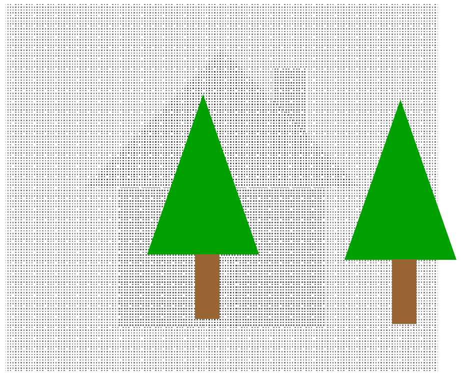
Figure from David Forsyth

# Многомасштабное сопоставление

- Идея метода
  - Строим гауссовы пирамиды для каждого из 2х изображений
  - Ищем преобразование на самом низком разрешении
    - Можно использовать полный перебор
  - Используем как начальное приближение для сопоставления на следующем уровне
    - Для уточнения можем использовать градиентный спуск
- Проблемы
  - Если на нижнем уровне ошиблись, тогда на следующих не можем исправить ошибку
  - Всё равно медленно
  - Проблема устойчивости

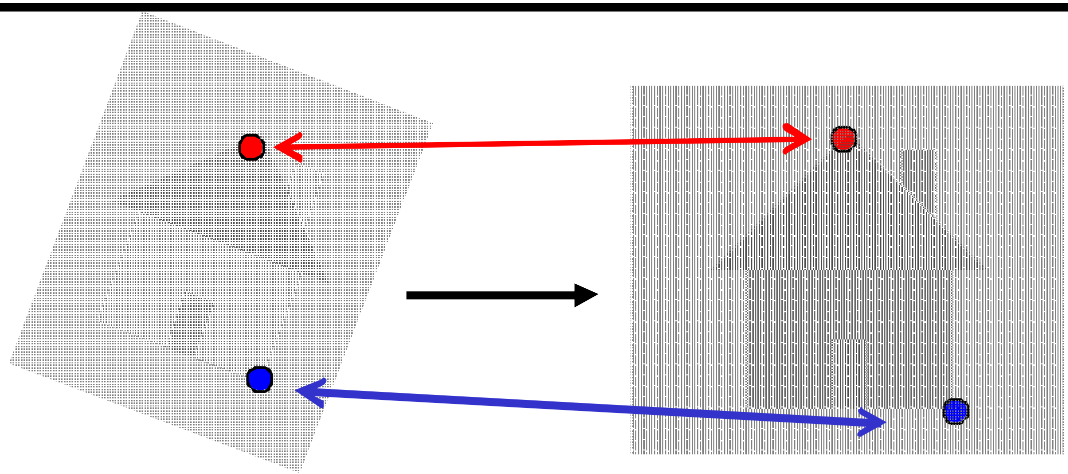
# Перекрытие объекта





- Как быть, если объект («дом») частично перекрыт другими объектами («ёлками»)
- Положение «ёлок» относительно дома разное на разных ракурсах
- Прямое попиксельное сопоставление изображений может не дать хорошего результата

# Локальные особенности



- Найти хорошо различимые точки на изображениях
  - «особенности» (features)
  - «локальный особые точки» (local feature points)
  - «характеристические точки» (characteristic points)
- Сопоставить точки (feature matching)
  - Определить, какой точке на одном изображении соответствуют какая точка на втором изображении
- Найти такое преобразование, которое совмещает найденные точки

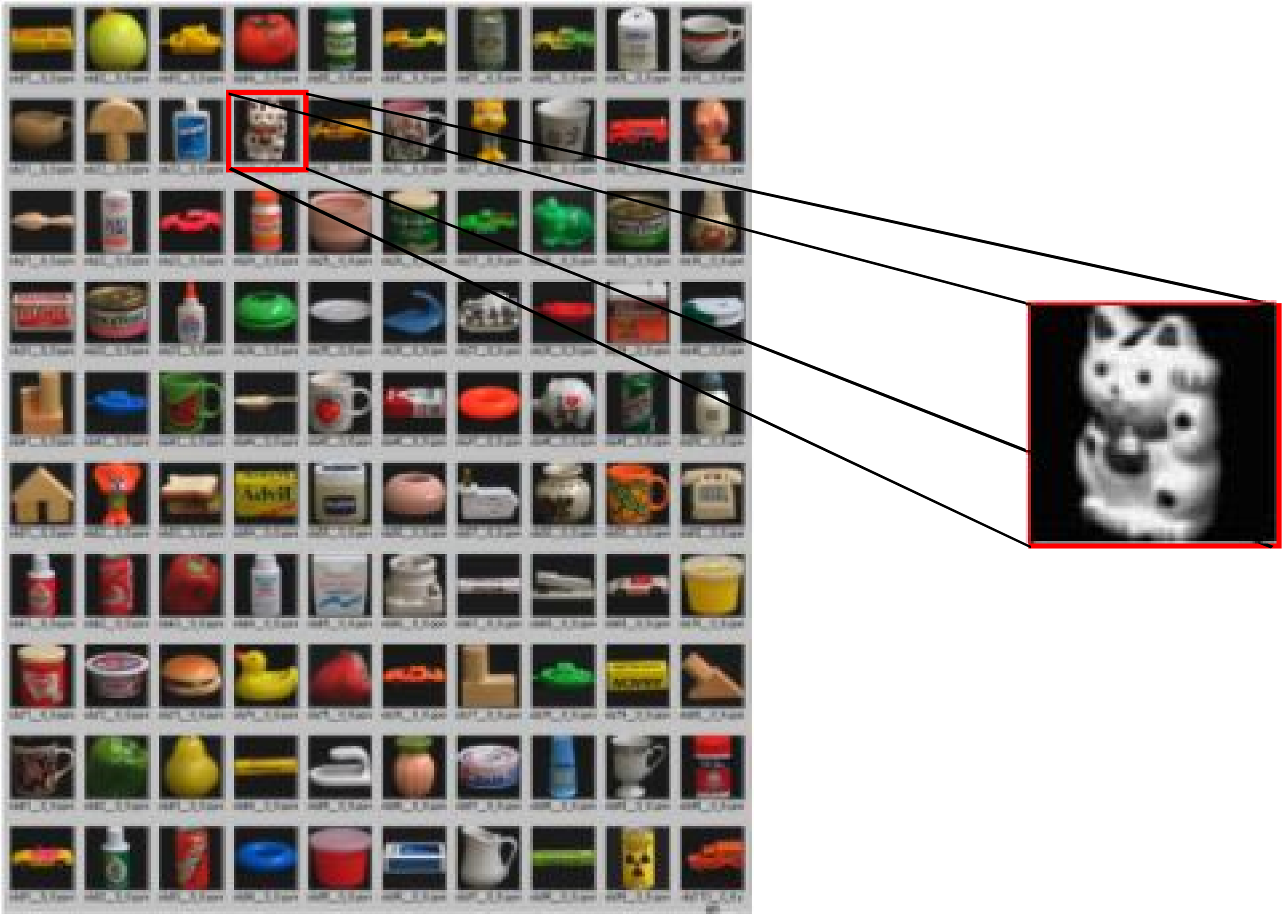
# Применение

Сопоставление изображений и трёхмерная реконструкция

# Применение

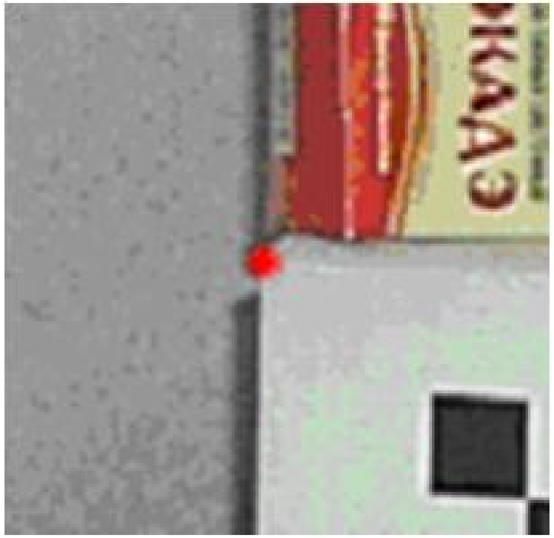

Классификация изображений и выделение объектов

# Применение

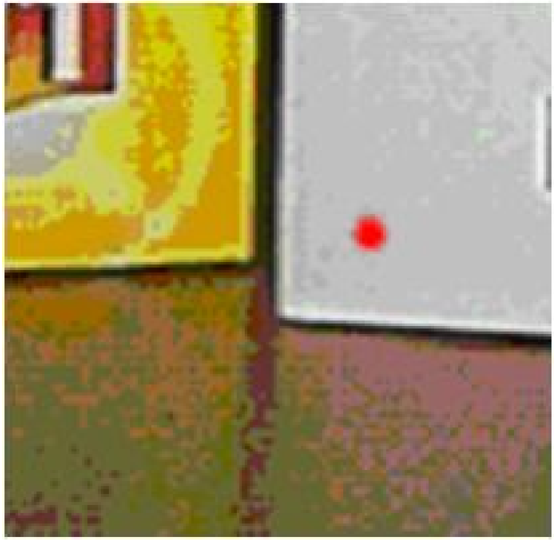


Поиск изображений по содержанию в базе изображений

# Локальные особенности



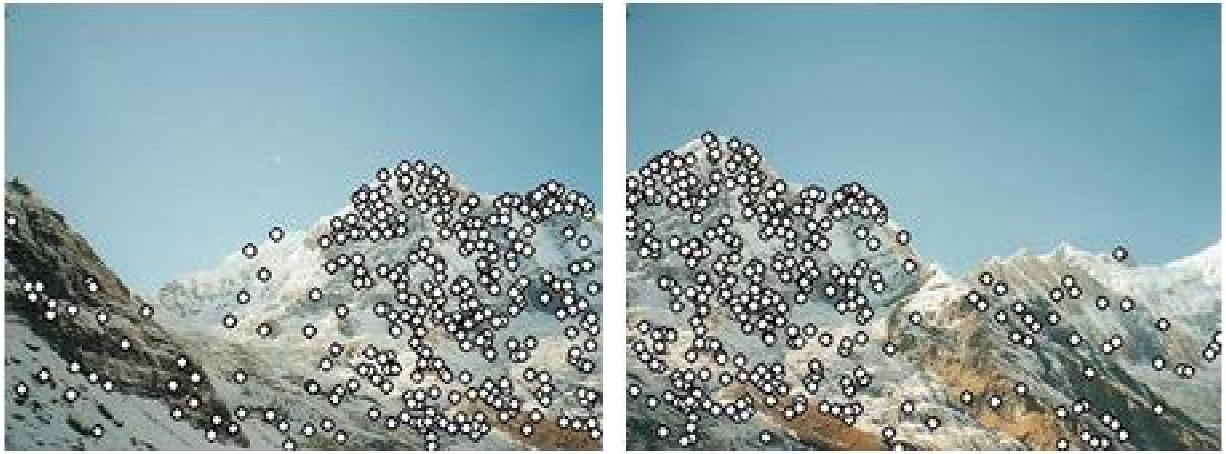
Пример особой точки



Пример точки, не являющейся особой

- Локальная (особая) точка *p* изображения I
    - это точка с характерной (особой) окрестностью , т.е. отличающаяся от всех других точек в некоторой окрестности *p*

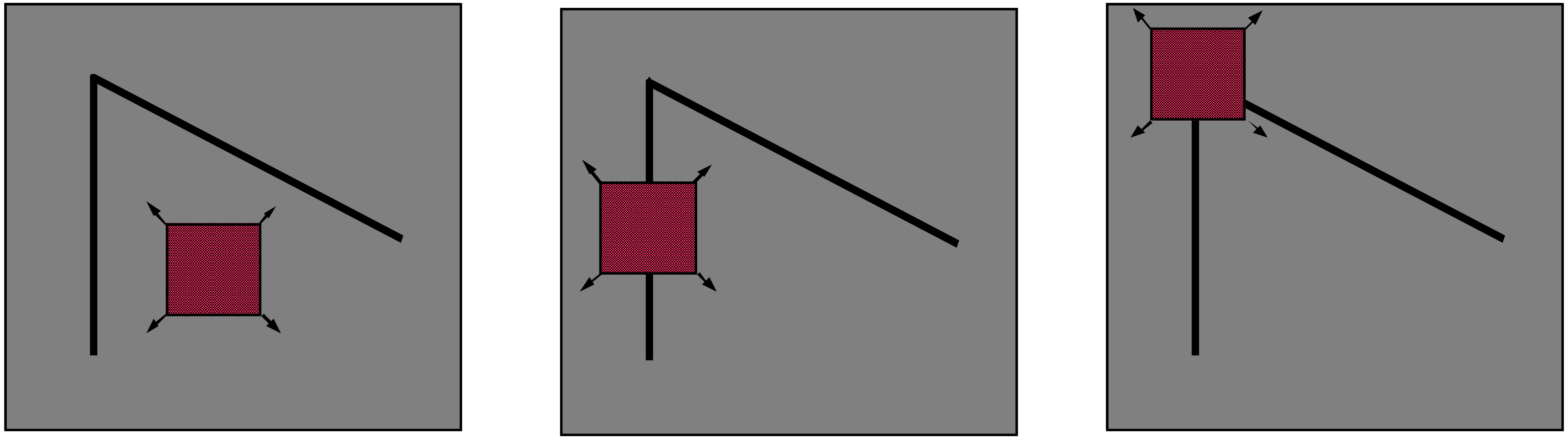
# Требования к особенностям



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

# Локальные особенности

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

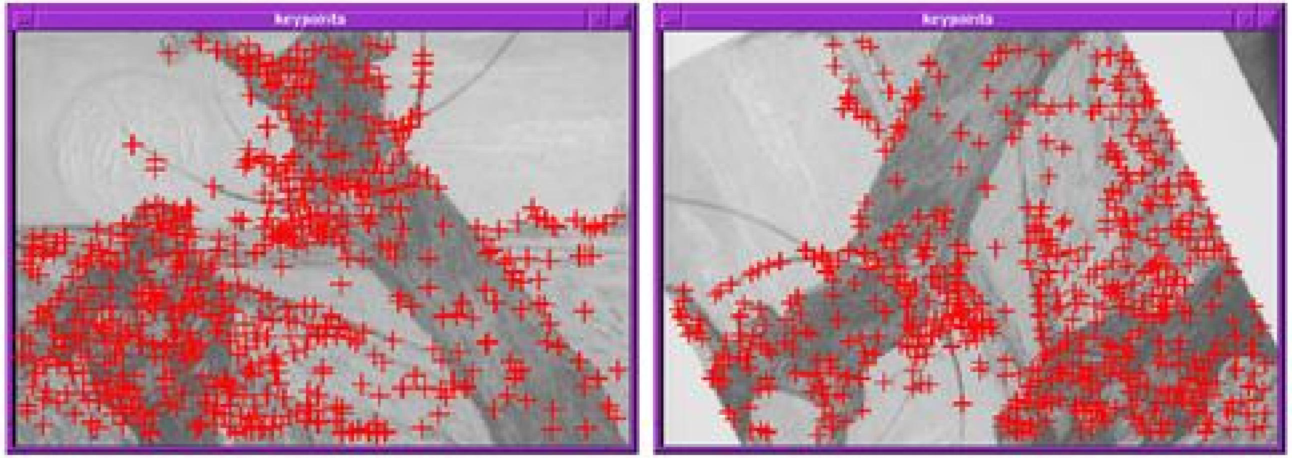


*монотонный* регион: в любом направлении изменений нет

*«край»*: вдоль края изменений нет

*«уголок»*: изменения при перемещении в любую сторону

# Детектор Харриса



- Наиболее популярный детектор локальных особенность точек – детектор Харриса (Harris)
- Главное свойство угла
  - в области вокруг угла у градиента изображения два доминирующих направления
- Уголки хорошо повторимы и различимы

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

# Математические детали

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

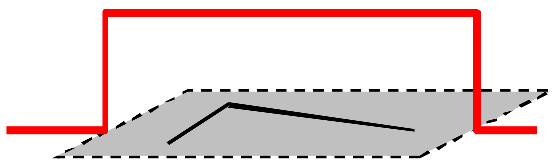
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Функция окна

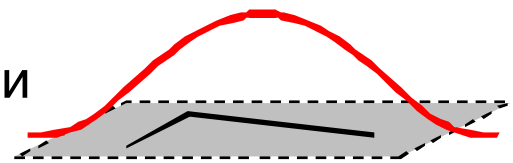
Сдвиг яркости

Яркость

Функция окна $w(x,y) =$



или



1 в окне, 0 вне

Гауссиана

Source: R. Szeliski

# Математические детали

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)

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

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

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

Slide by S. Lazebnik

# Математические детали

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:

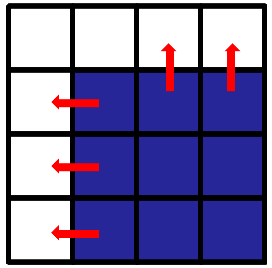
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

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

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:

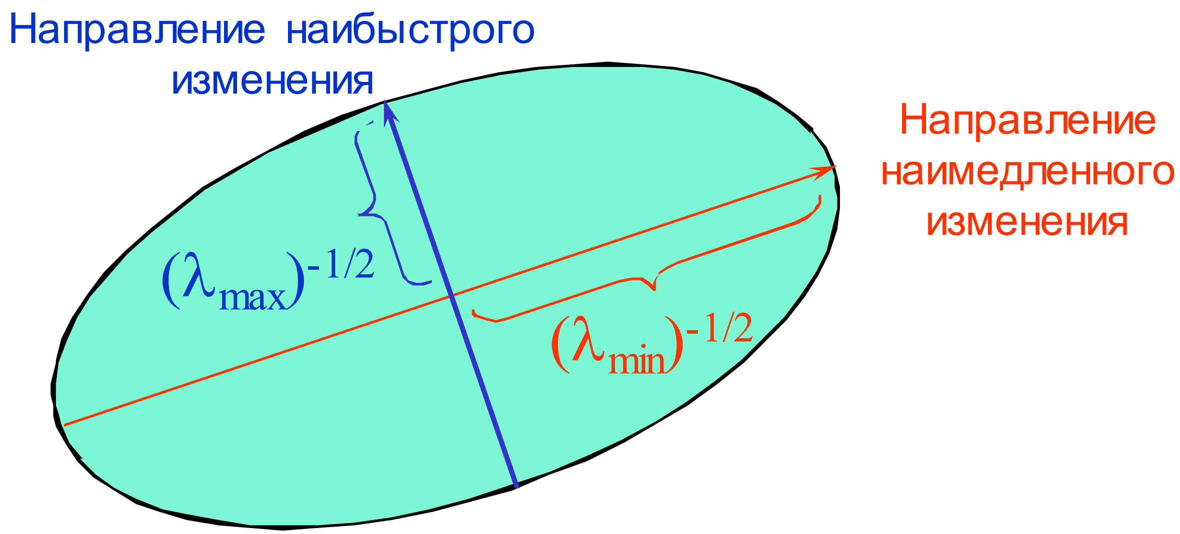
$$M = R^{-1} D R = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

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

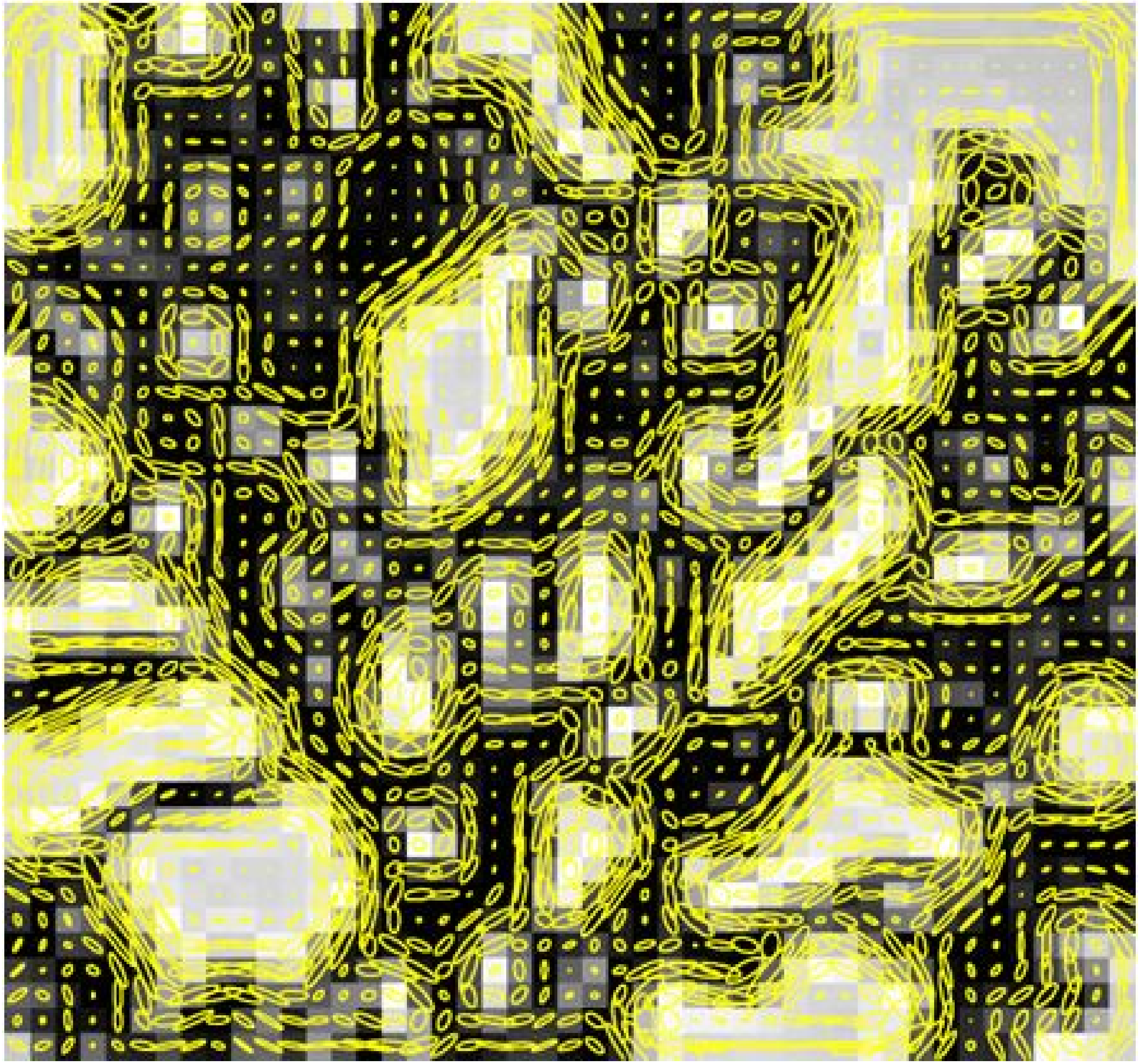
Уравнение эллипса:

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




Slide by S. Lazebnik

# Пример



Визуализация матриц вторых моментов (Гессианов)

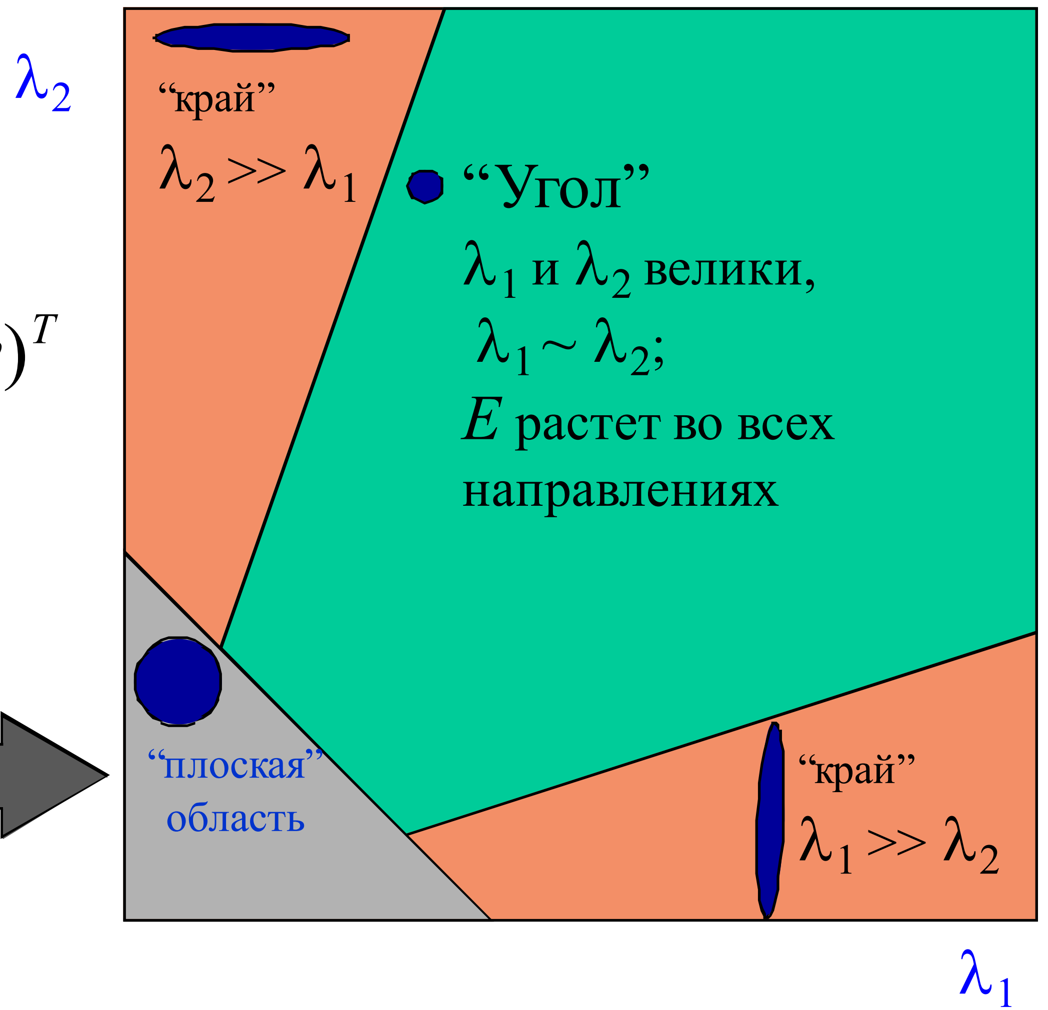
# Детекторы углов

Классификация точек изображения по собственным значениям матрицы производных *M*

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

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

$\lambda_1$ и $\lambda_2$ малы;
$E$ не меняется по всем направлениям



# Детекторы углов

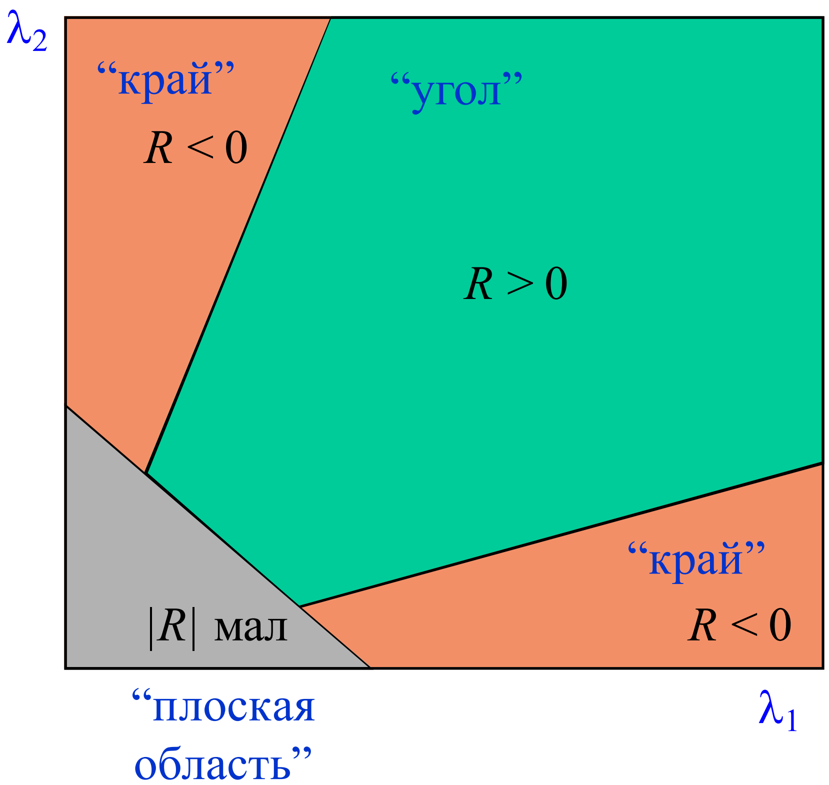
- Мера отклика угла по Харрису:

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

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

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

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




- Мера по Фёрстнеру (Forstner):

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

# Алгоритм детектора Харриса

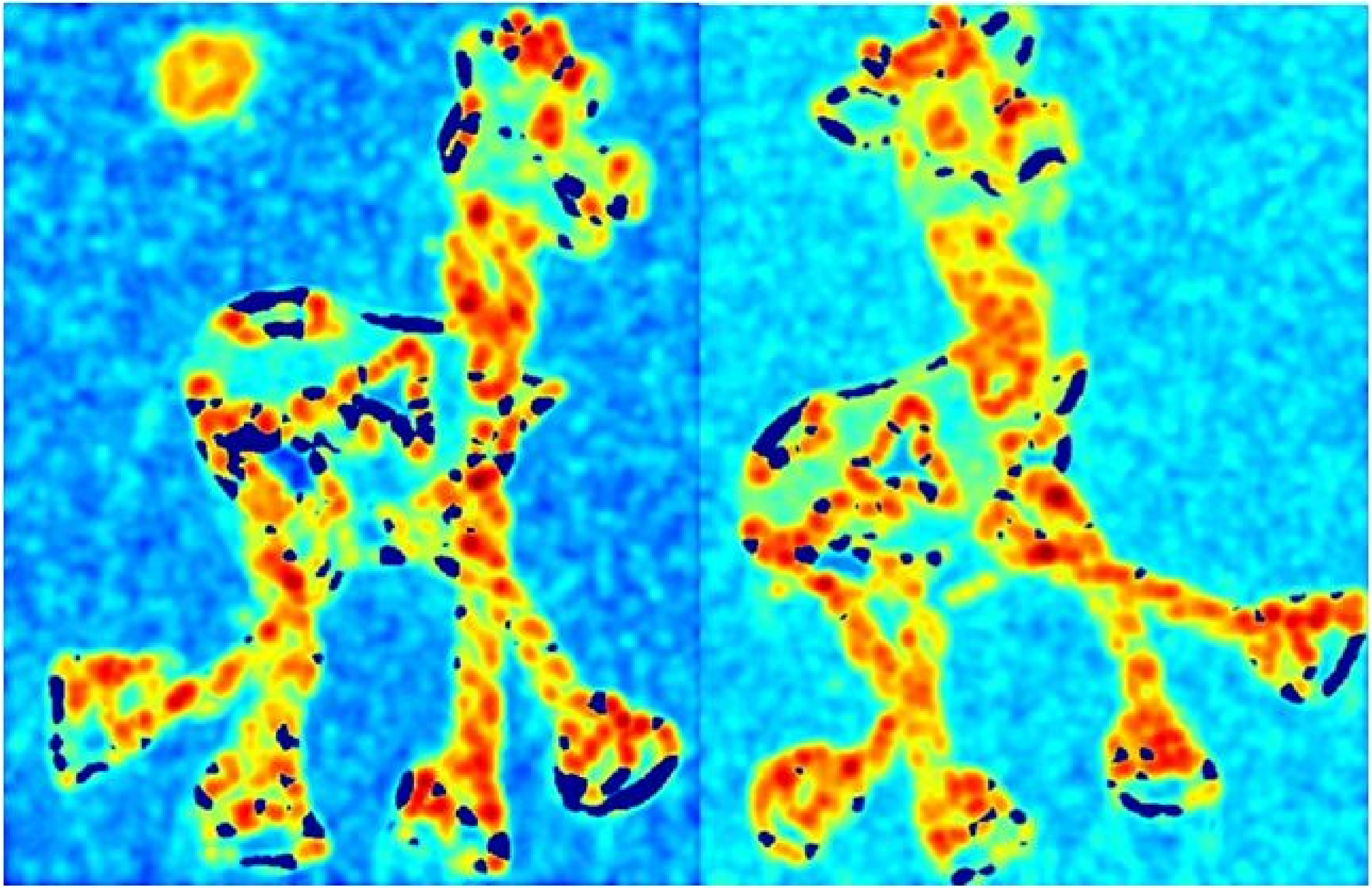
1. Вычислить градиент изображения в каждом пикселе
   - С использованием гауссова сглаживания
2. Вычислить матрицу вторых моментов M по окну вокруг каждого пикселя
3. Вычислить отклик угла *R*
4. Отсечение по порогу *R*
5. Найти локальные максимумы функции отклика (non-maximum suppression) по окрестности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

Slide by S. Lazebnik

# Демонстрация по шагам

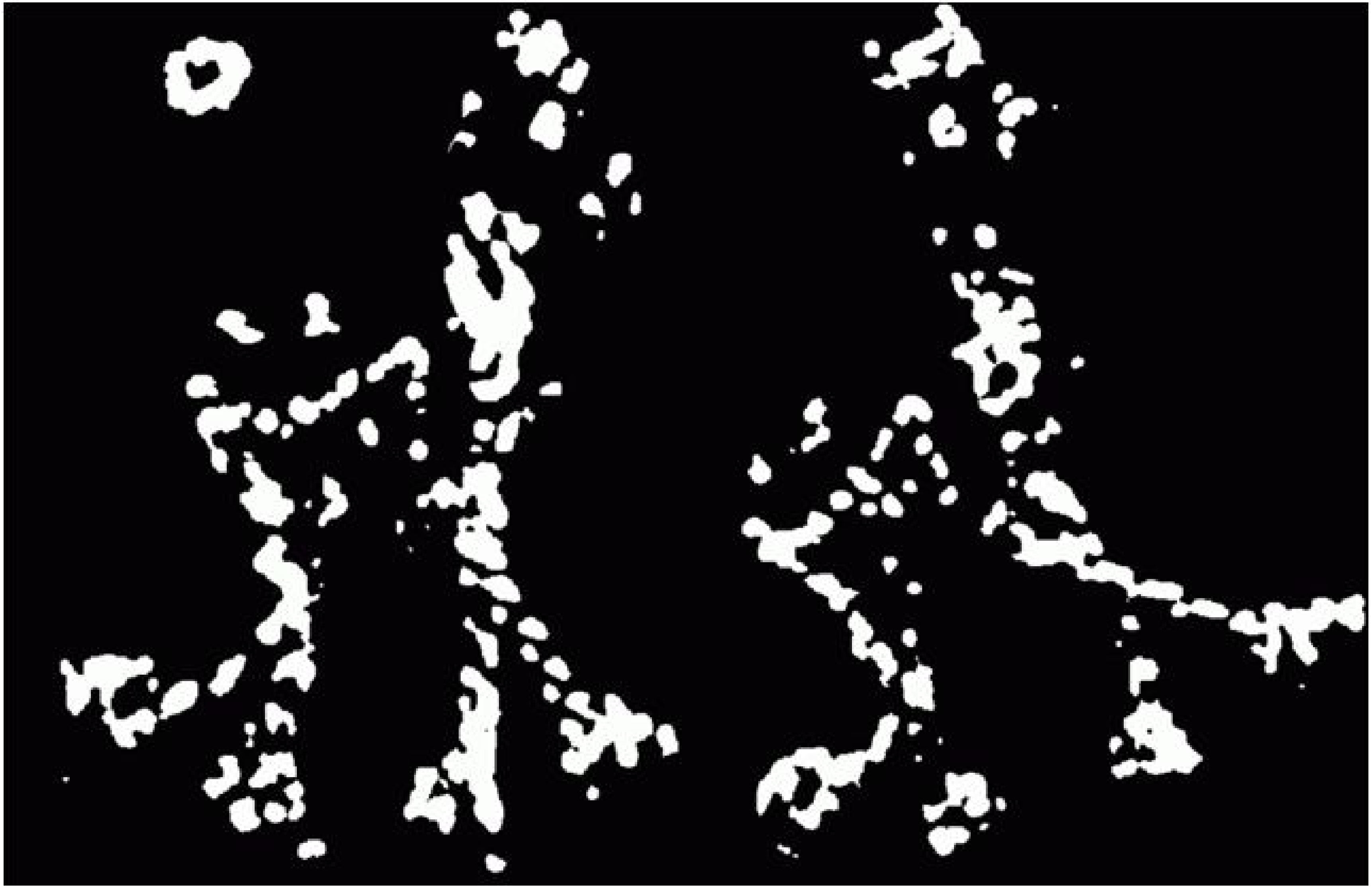
# Демонстрация по шагам

Вычисление отклика угла *R*



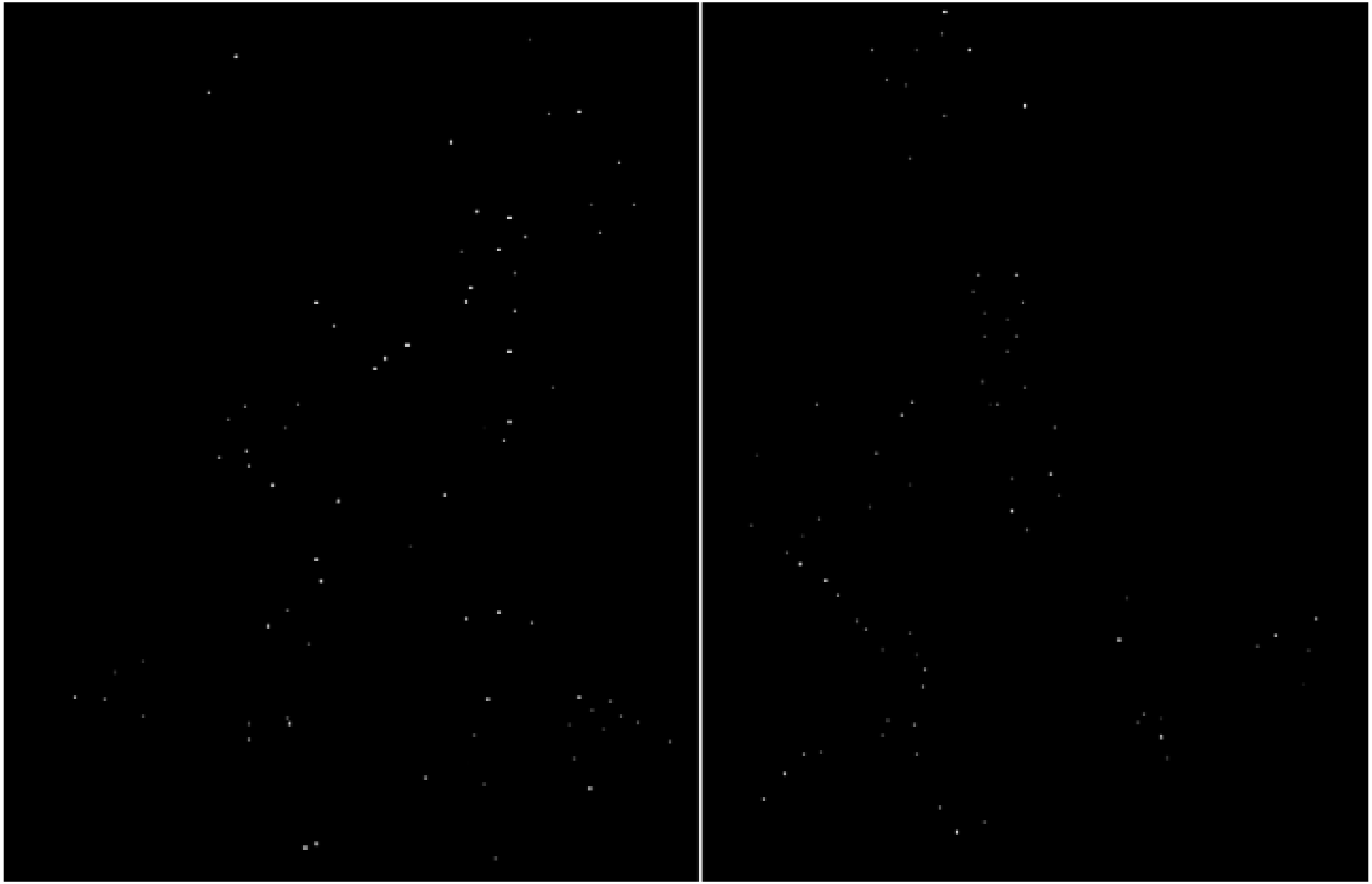
# Демонстрация по шагам

Найдём точки с большим откликом $R$>порог



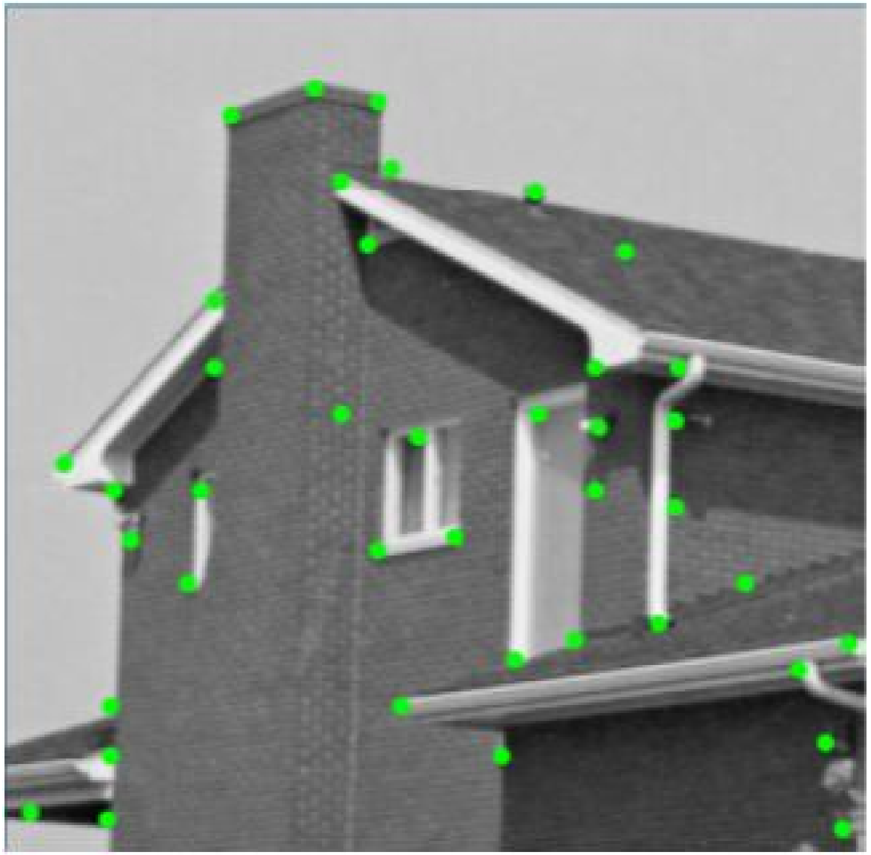
# Демонстрация по шагам

Оставим только точки локальных максимумов *R*

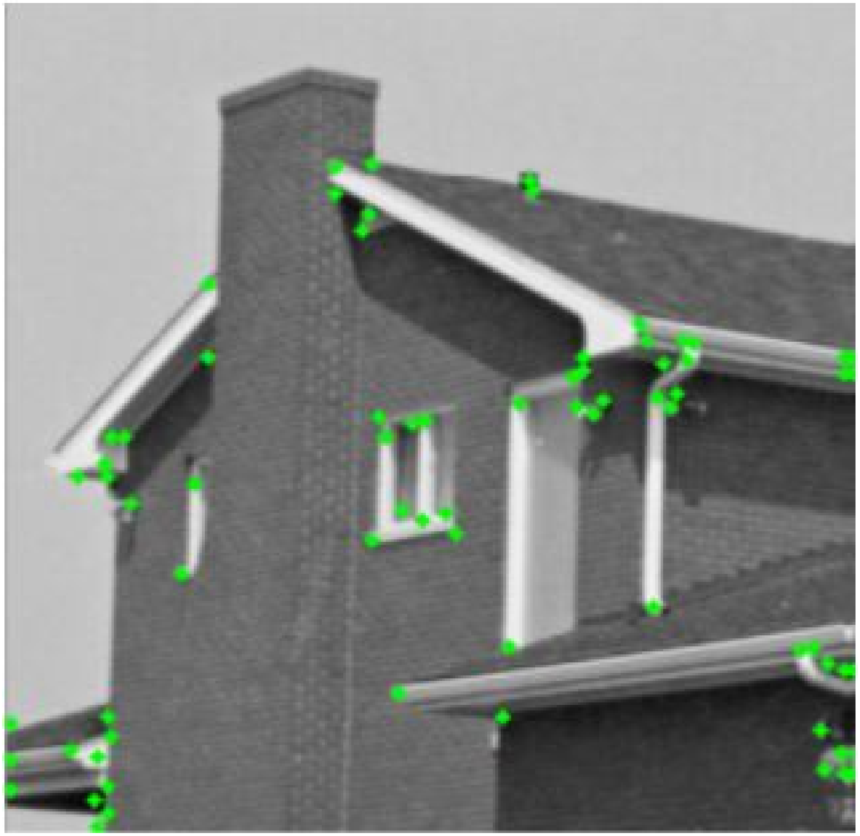


# Демонстрация по шагам

# Результат работы детектора



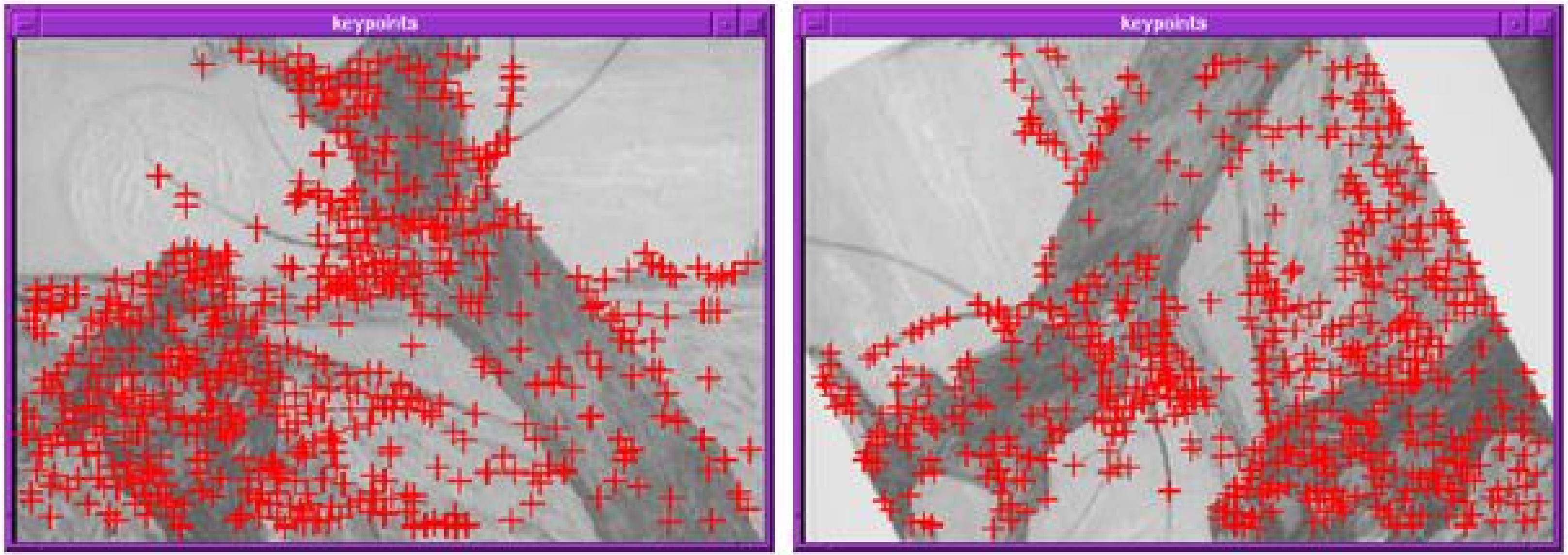
детектор Фёрстнера



детектор Харриса

# Инвариантность

- Хорошо бы чтобы особенности находились в одной и той же точки сцены, на изображениях с разных ракурсов и в разных условиях освещения
- Если применить геометрическое преобразование к изображению, то алгоритм должен найти тот же самый набор точек

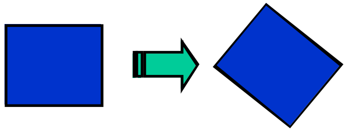



Slide by S. Lazebnik

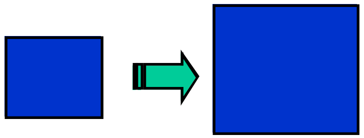
# Модели преобразования

Геометрические

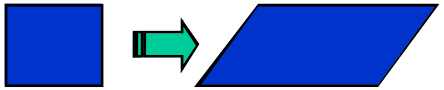
- **Поворот**



- **Масштаб**



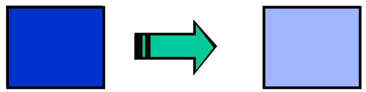
- **Аффинное**



подходит для упрощенных моделей камер (ортографической проекции, локально-плоских объектов)

Фотометрические

- **Аффинное изменение яркости** ($I \rightarrow a\,I + b$)

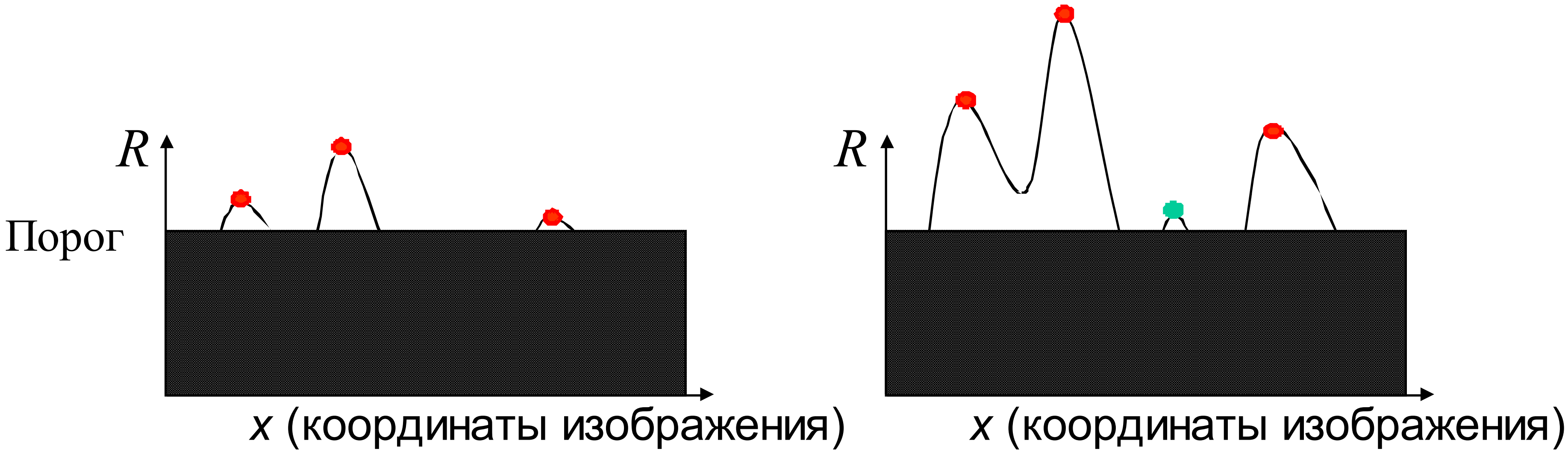


# Детекторы Харриса

- Частичная инвариантность к изменению освещенности

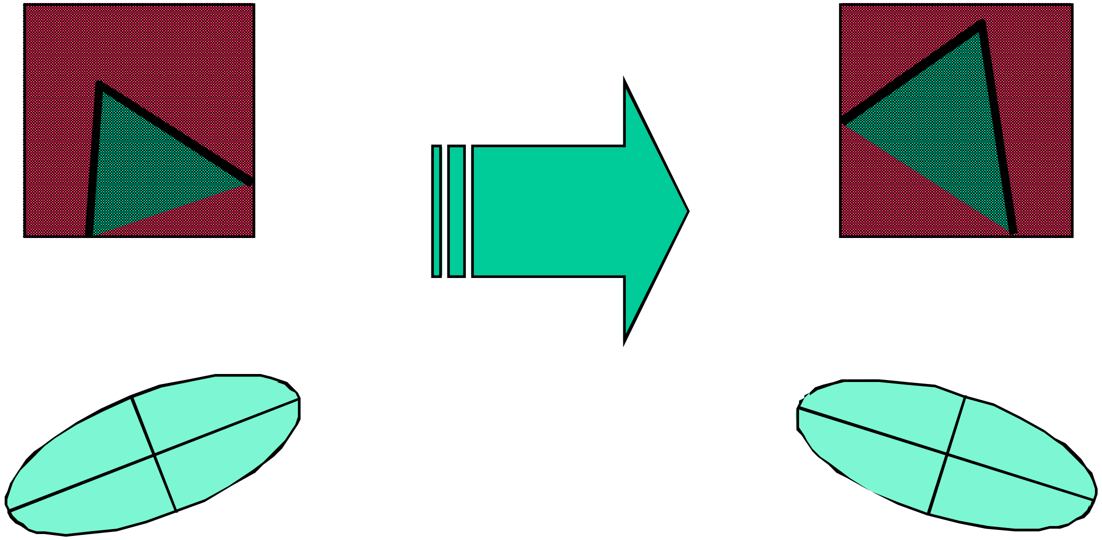
  ✓Используются только производные
  => инвариантность к сдвигу $I \to I + b$

  ✓Масштабирование: $I \to a\,I$



# Детектор Харриса

Инвариантность к вращению изображения:



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

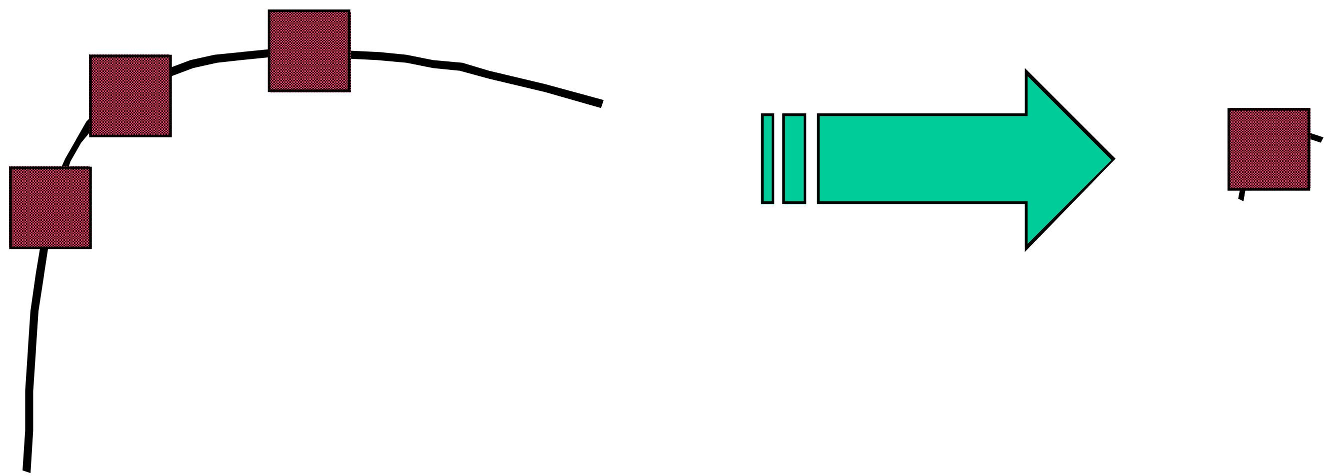
Эллипс вращается, но его форма (собственные значения) остаются неизменными

*Отклик угла $R$* инвариантен относительно вращению изображения

Slide by S. Lazebnik

# Масштабирование?

- Угол или нет? - Зависит от масштаба изображения!

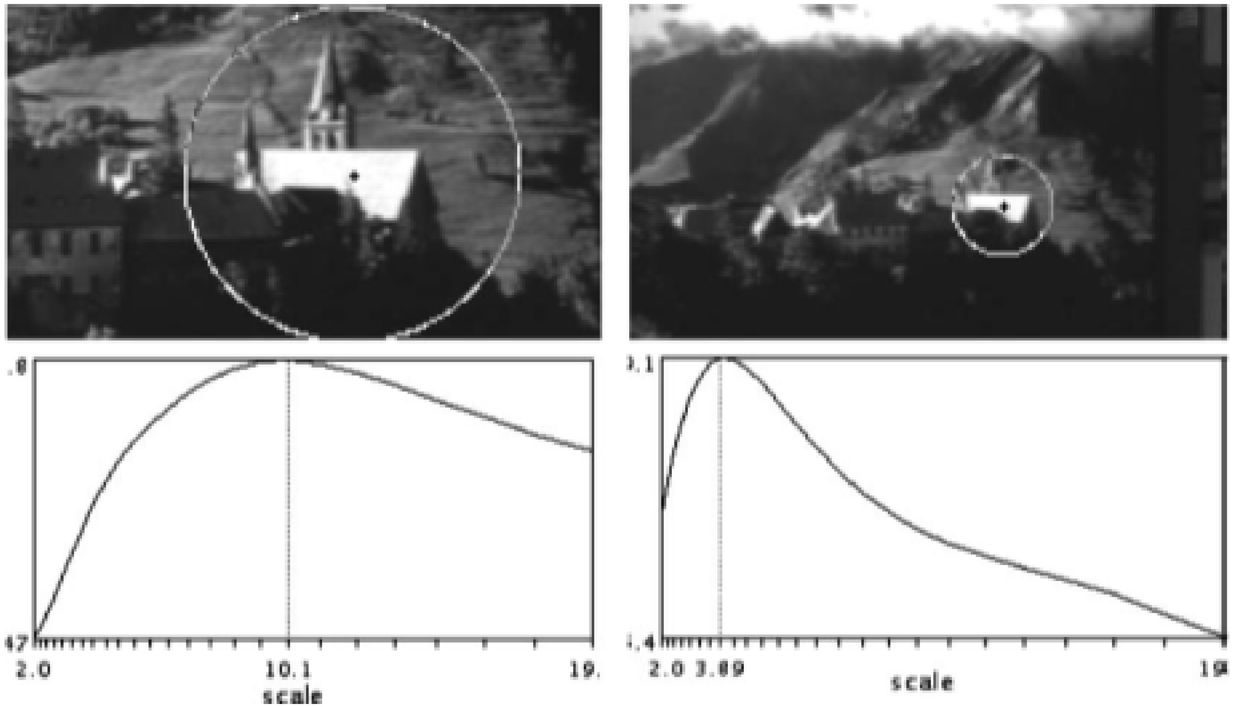


Все эти точки будут помечены как ***края***

**Угол !**

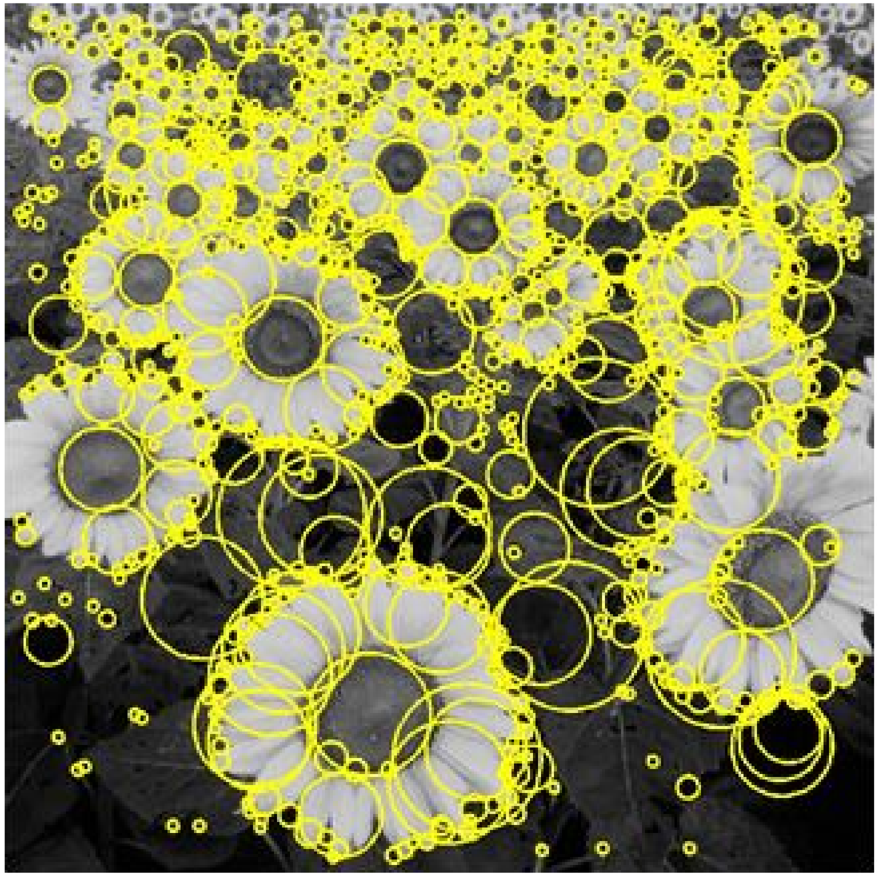
# Инвариантность к масштабированию

- 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



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

# Поиск краев

$$f$$

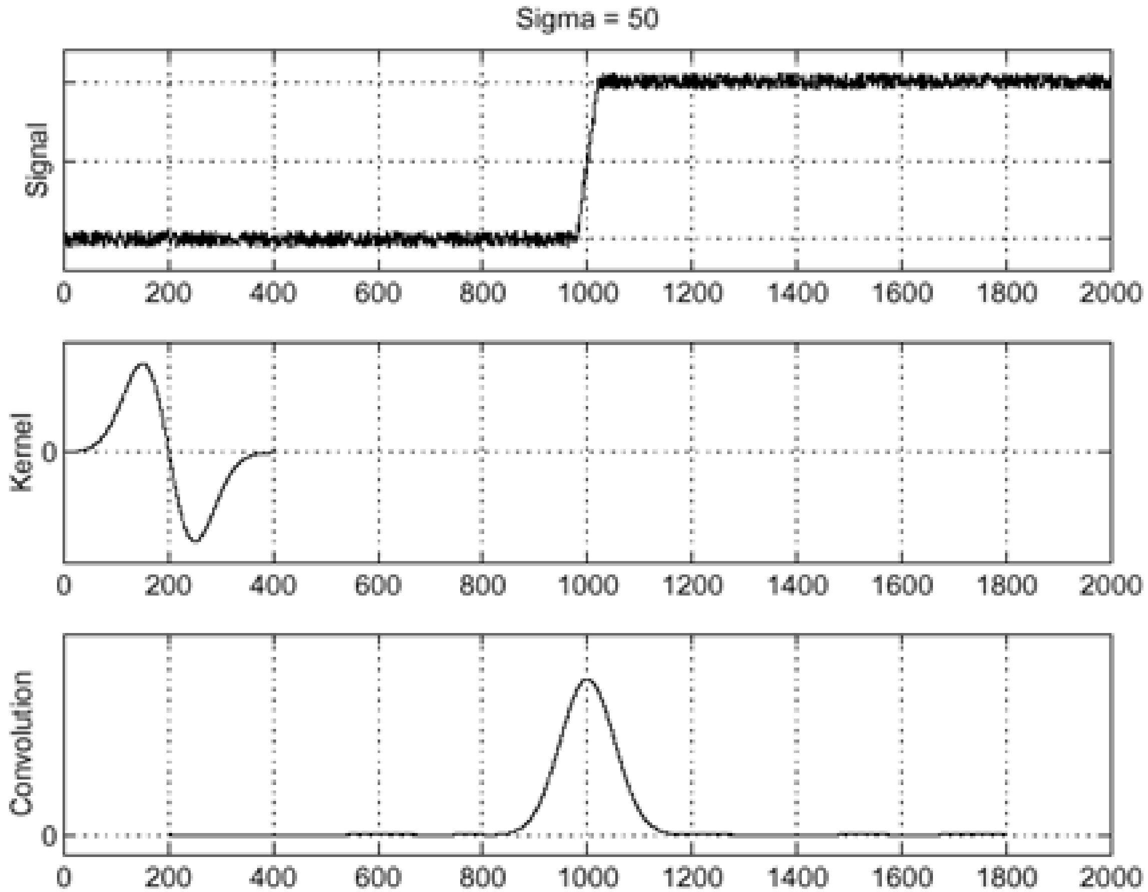
Край

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

Производная гауссианы

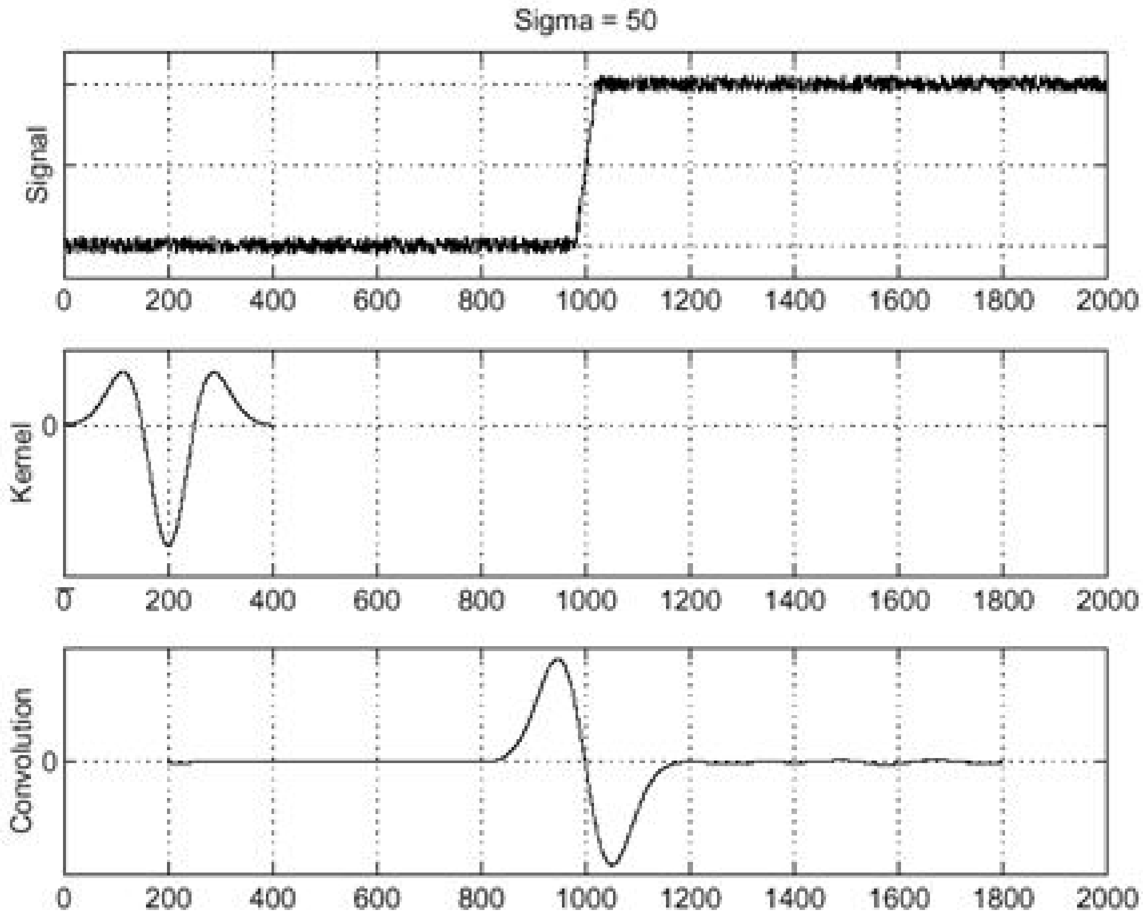
$$f * \frac{d}{dx}g$$

Край = максимум производной




Source: S. Seitz

# Второй проход




$f$ — Край

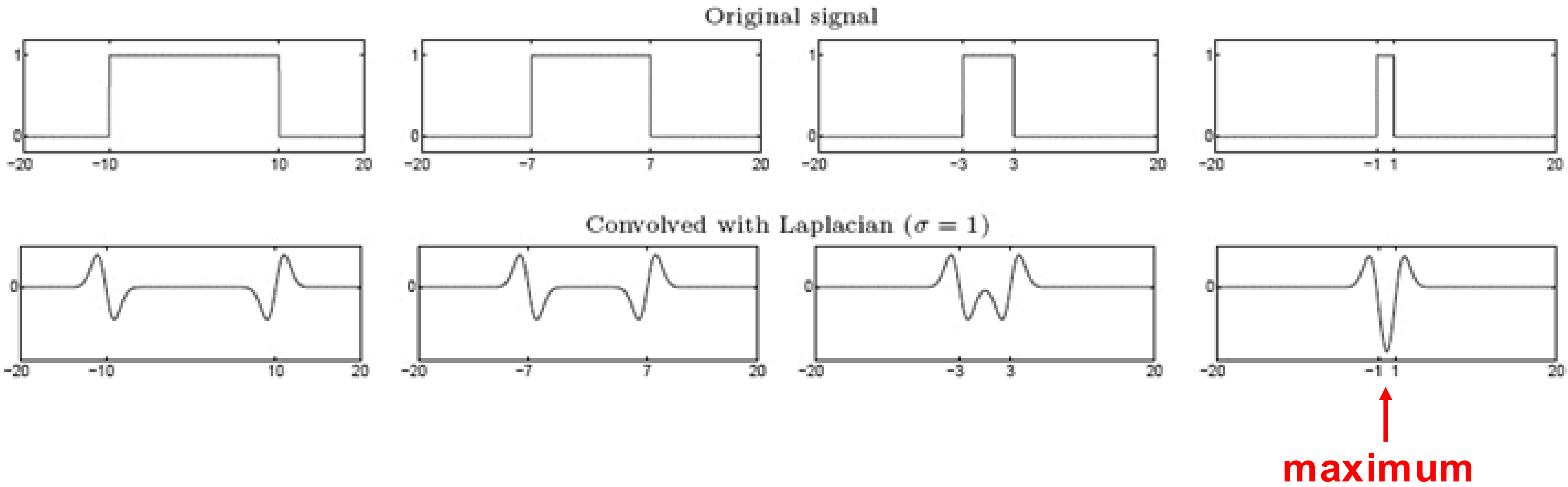
$\frac{d^2}{dx^2} g$ — Вторая производная Гауссианы (Лапласиан)

$f * \frac{d^2}{dx^2} g$ — Край = переход через ноль второй производной

Source: S. Seitz

# От краев к блобам

- Край = «всплеск»
- Блоб = совмещение двух «всплесков»



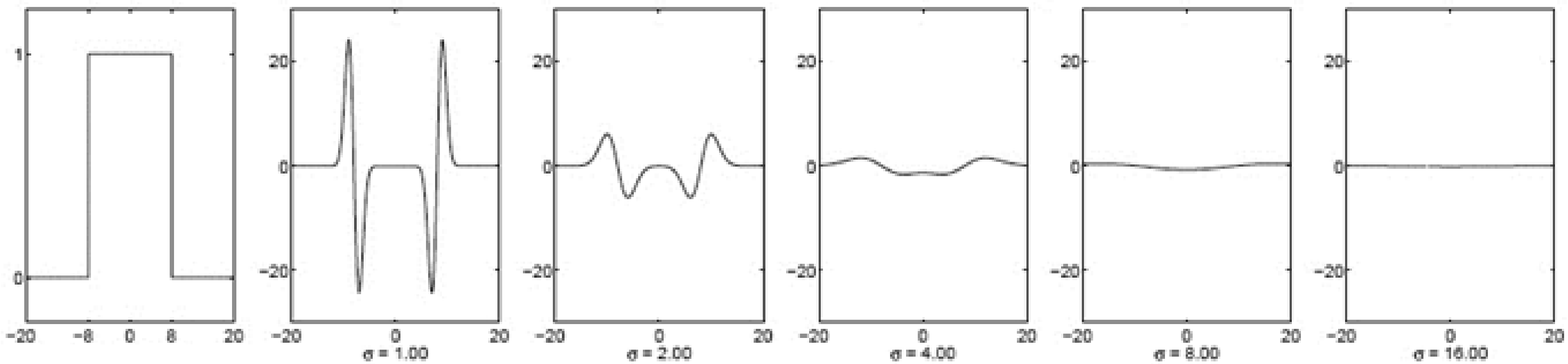

**Выбор масштаба**: величина отклика лапласиана Гауссиана достигает максимума в центре блоба в том случае, если размер лапласиана «соответствует» размеру блоба

Slide by S. Lazebnik

# Выбор масштаба

- Нужно найти характеристический размер блоба путем свертки с Лапласианом в нескольких масштабах и найти максимальные отклики
- Однако, отклик Лапласиана затухает при увеличении масштаба:

Ненормализованный отклик Лапласиана




Исходный сигнал (радиус=8)

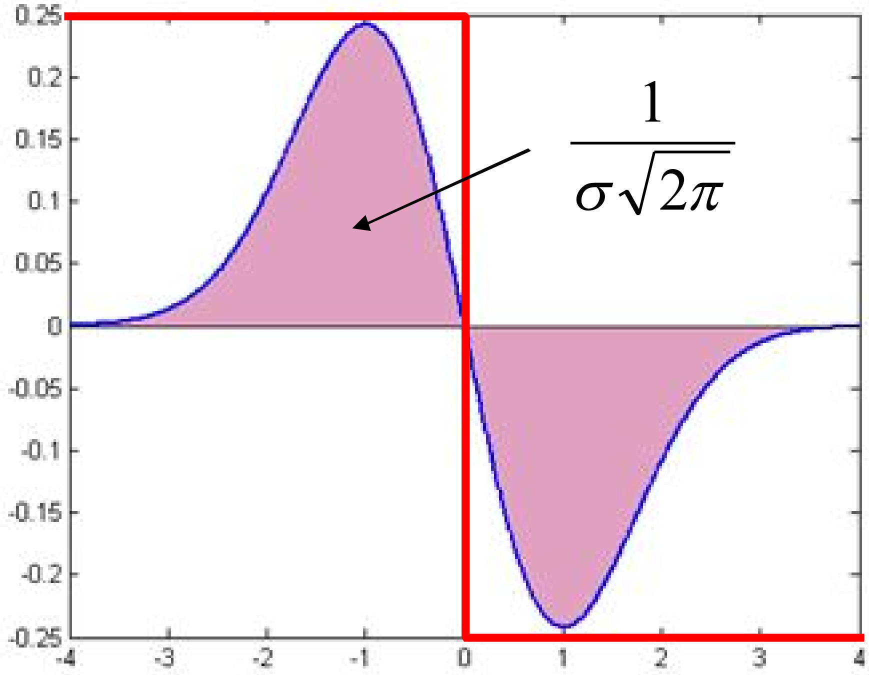
увеличение σ

Почему так происходит?

Slide by S. Lazebnik

# Нормализация масштаба

- Отклик производной фильтра Гаусса на идеальный край затухает с увеличением масштаба $\sigma$




Slide by S. Lazebnik

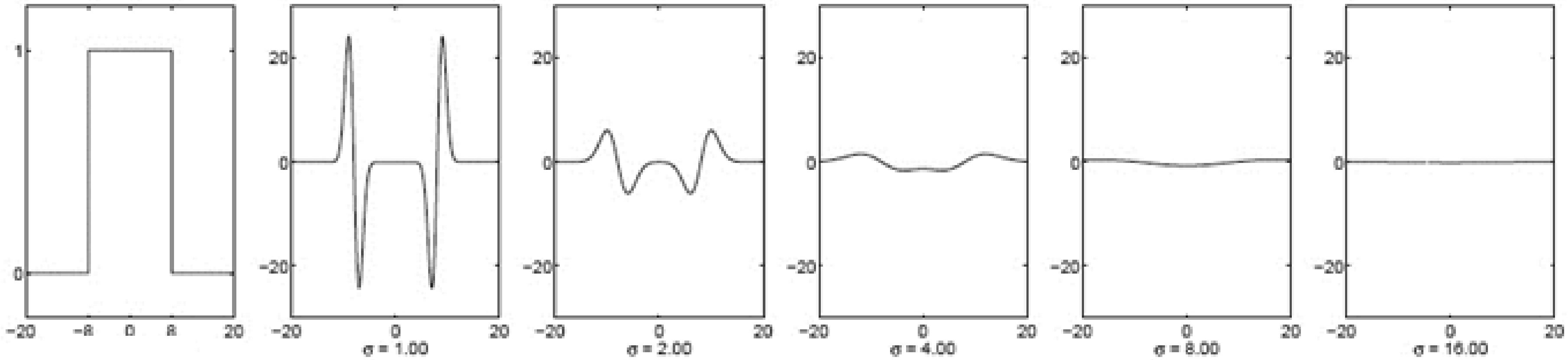
# Нормализация масштаба

- Отклик производной фильтра Гаусса на идеальный край затухает при увеличении σ
- Нужно домножить производную на σ для достижения инвариантности к масштабу
- Лапласиан это вторая производная фильтра гаусса, поэтому домножаем на $\sigma^2$

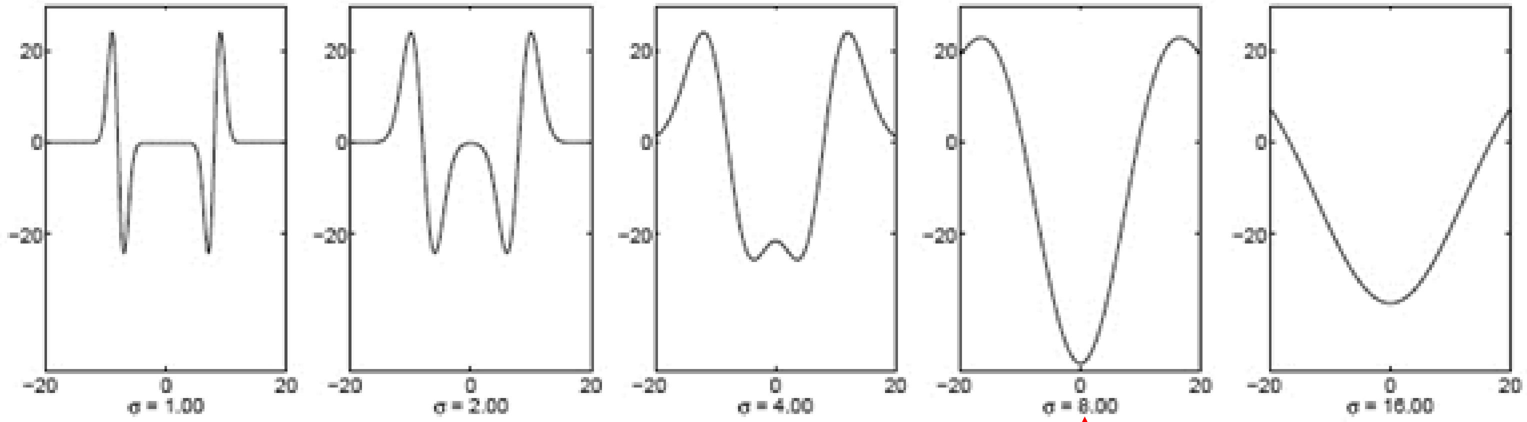
Slide by S. Lazebnik

# Эффект нормализации

Исходный сигнал Ненормализованный отклик Лапласиана




Нормализованный по масштабу отклик Лапласиана

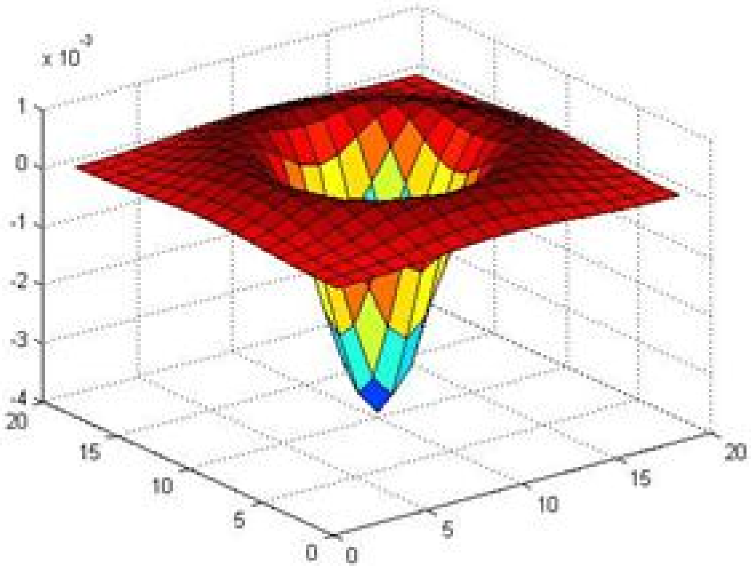


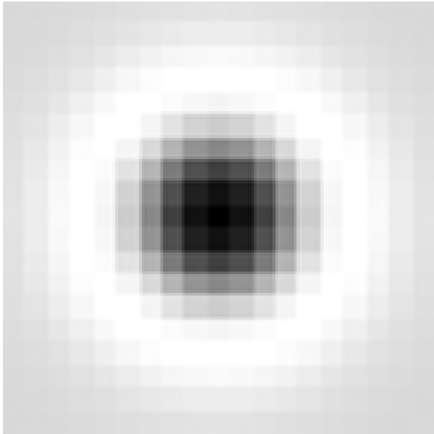

**максимум**

Slide by S. Lazebnik

# Поиск блобов в 2D

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



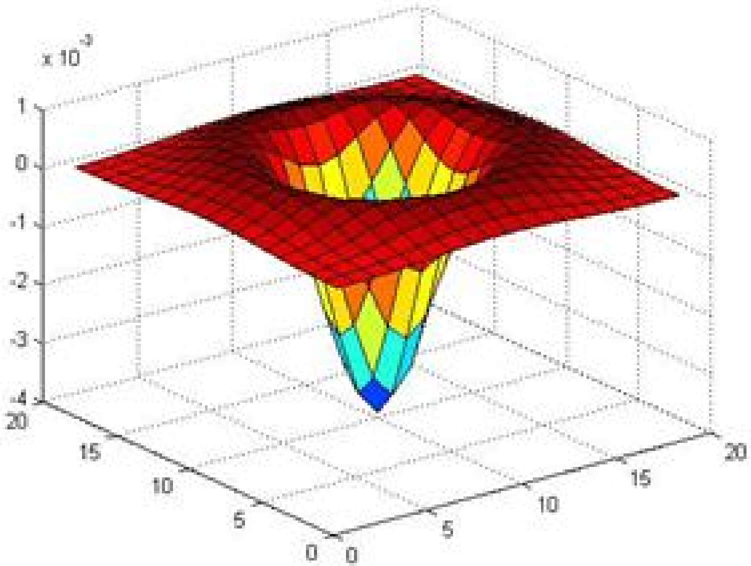



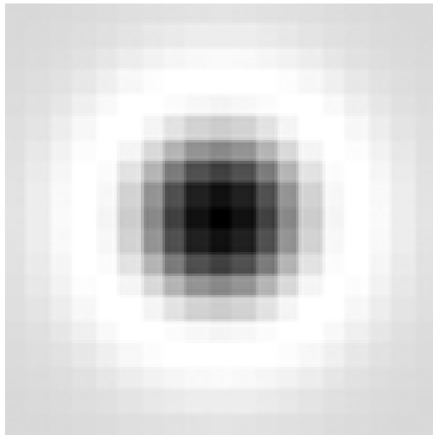
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Slide by S. Lazebnik

# Поиск блобов в 2D

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

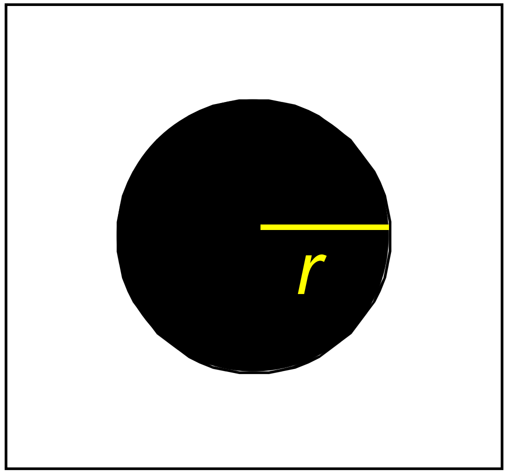


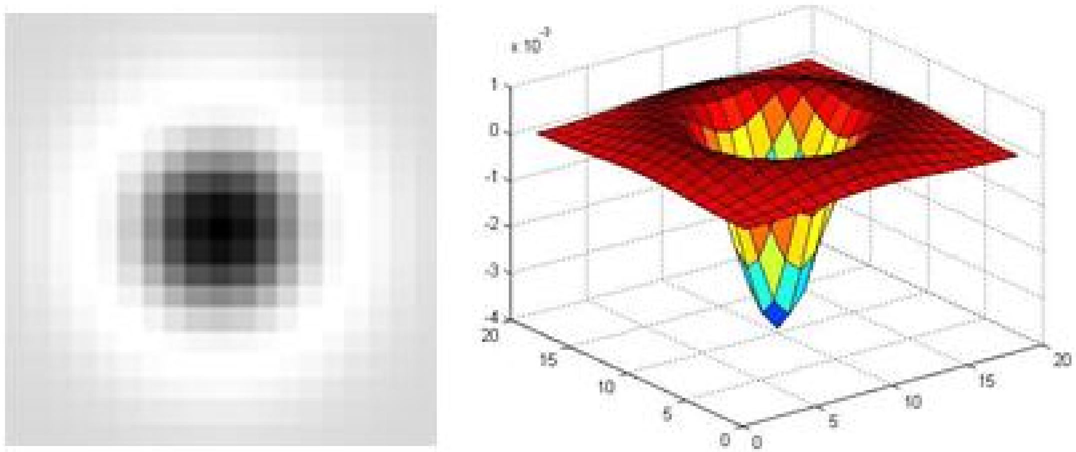
Нормализация: $$\nabla^2_{\text{norm}} g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

# Выбор масштаба

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




изображение



Лапласиан

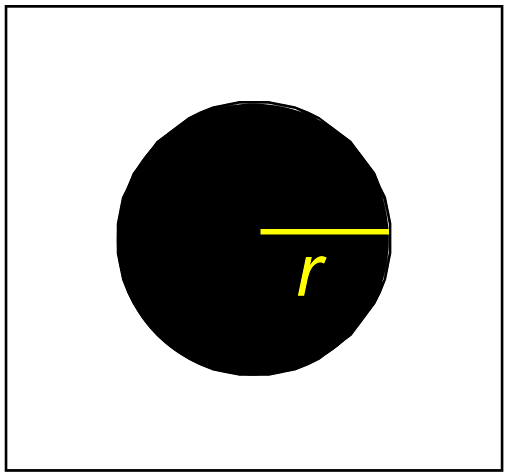
Slide by S. Lazebnik

# Выбор масштаба

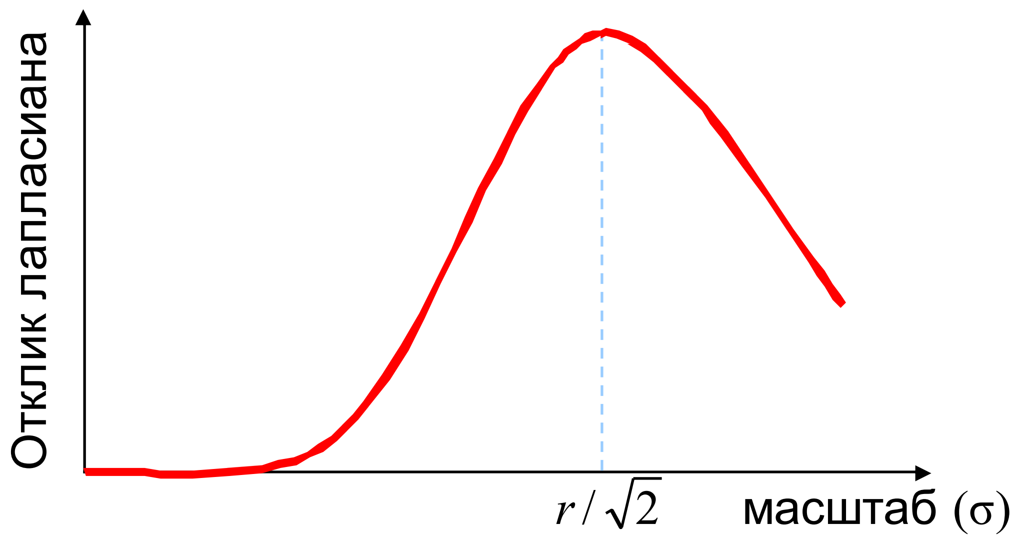
- 2D Лапласиан задается формулой:

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

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



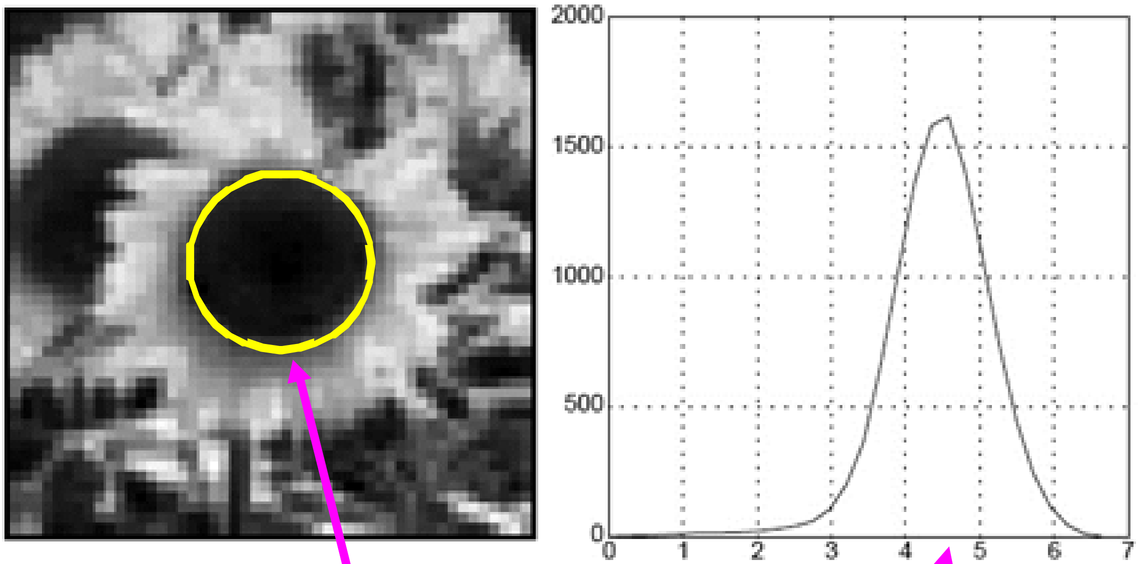

изображение




Slide by S. Lazebnik

# Характеристический размер

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

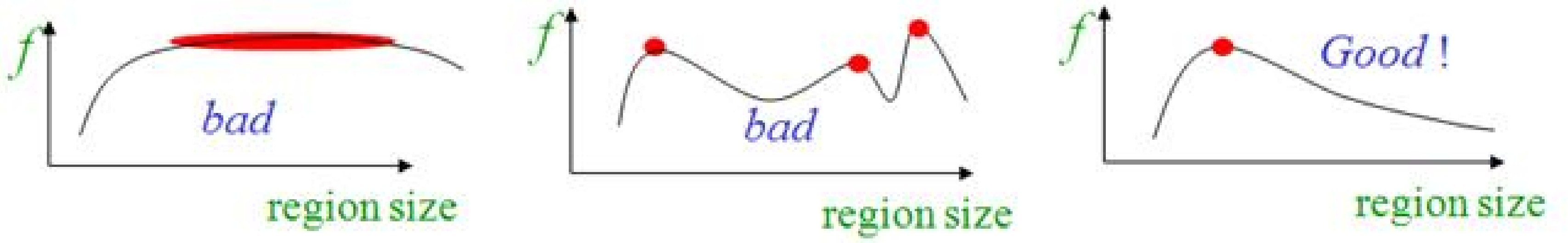



Характеристический масштаб

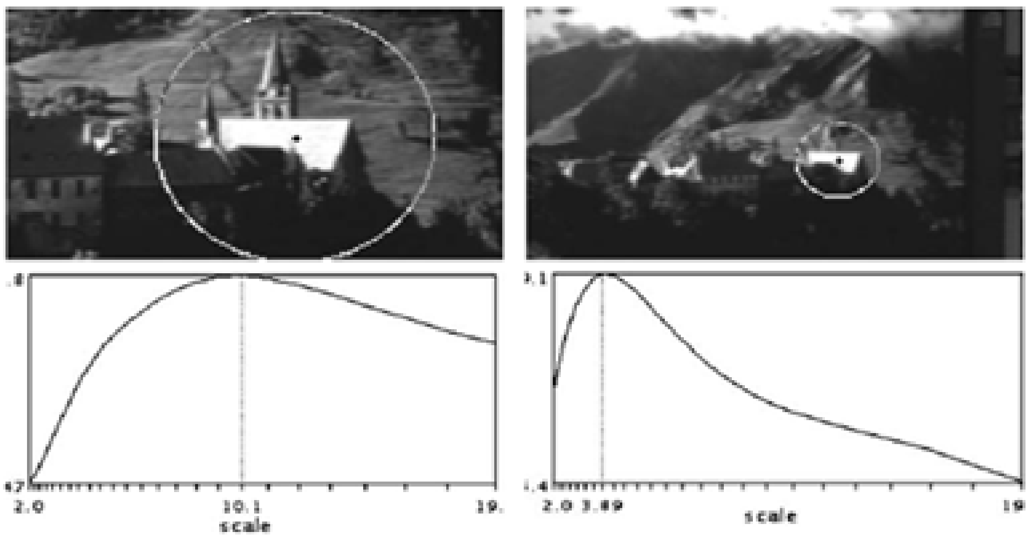
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

Slide by S. Lazebnik

# Характеристический размер



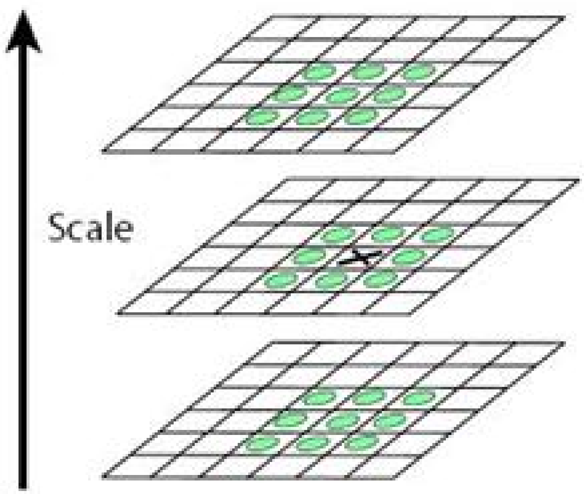

У «хорошего блоба»– один ярко выраженный пик функции




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

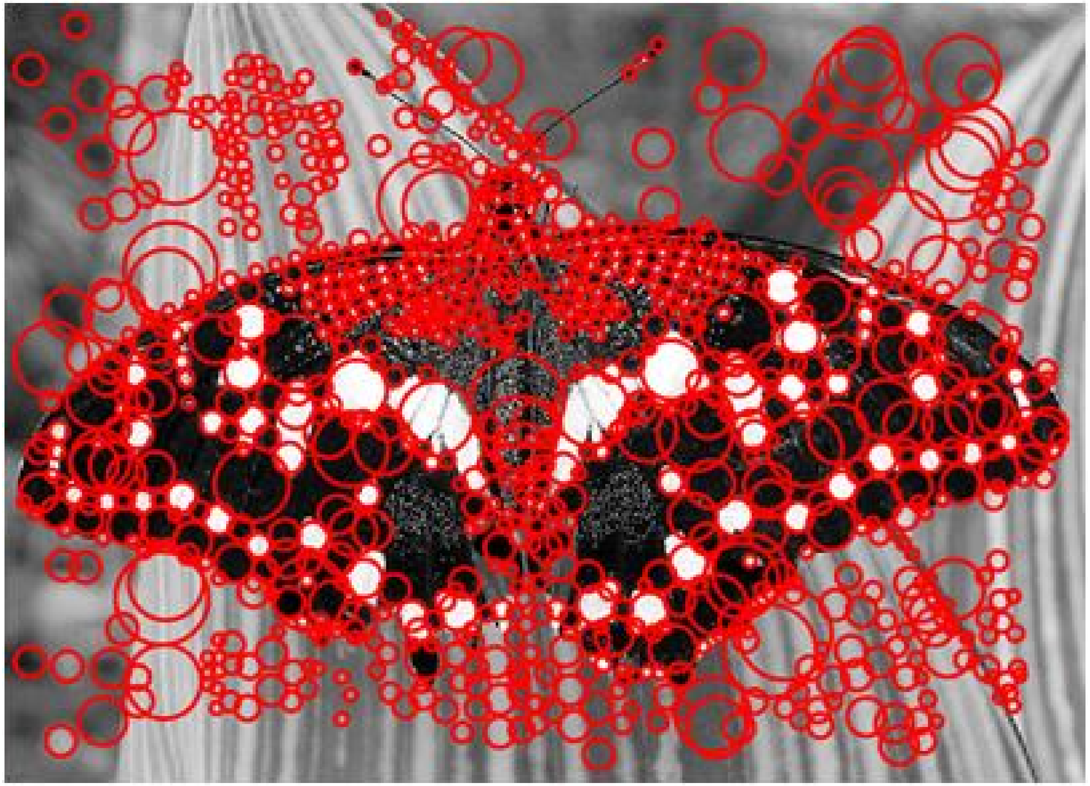
Slide by S. Lazebnik

# Пример

sigma = 11.9912

Slide by S. Lazebnik

# Пример



Slide by S. Lazebnik

# Эффективная реализация (DoG)

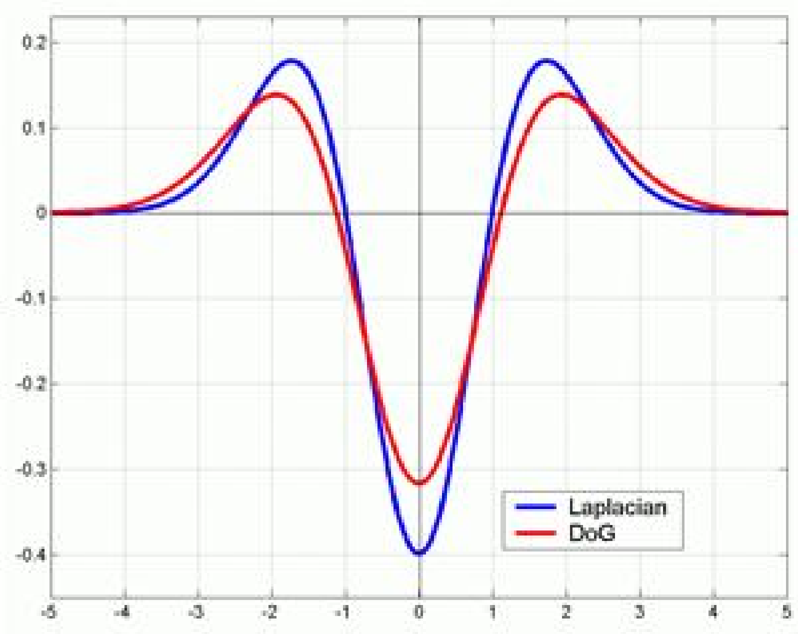
Приближение Лапласиана с помощью разницы гауссиан:

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

(Лапласиан)

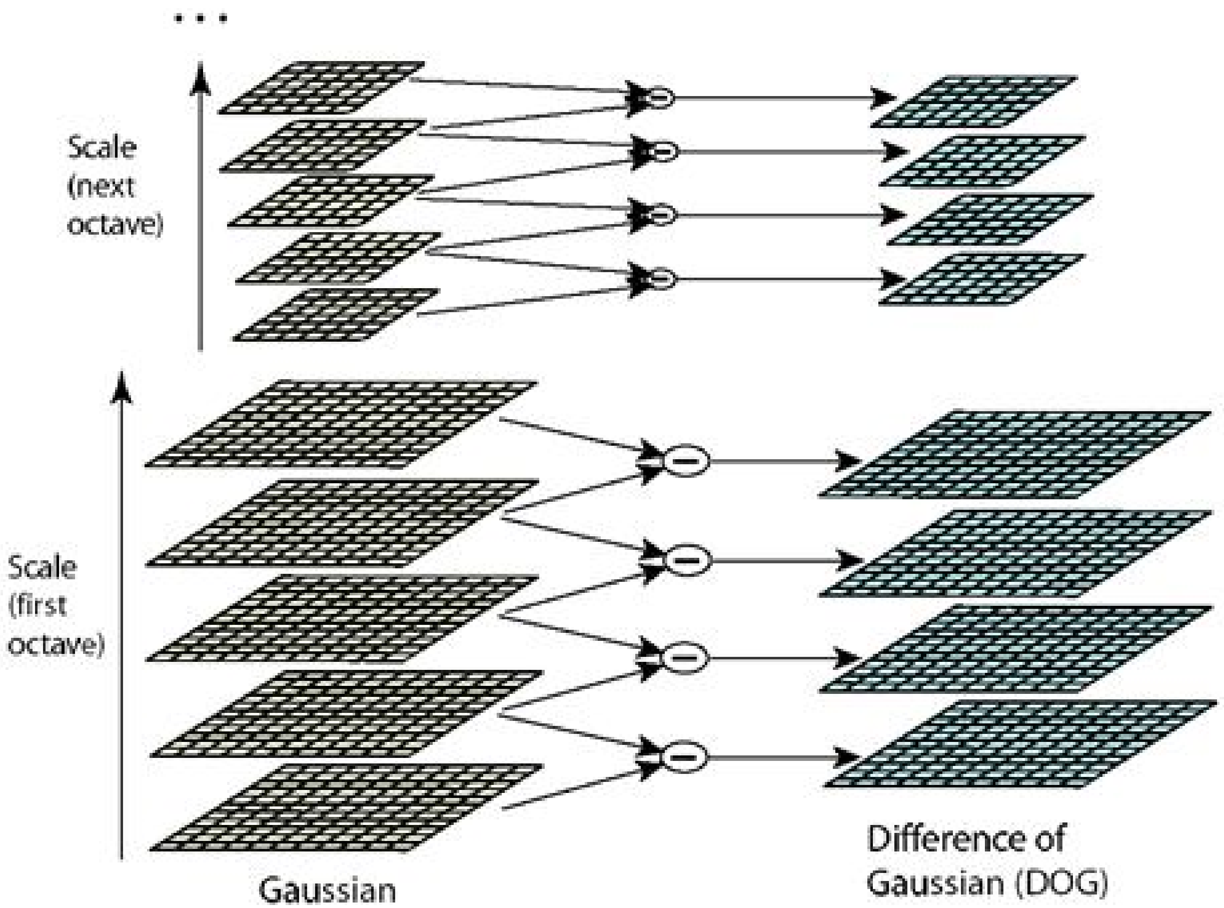
$$DoG=G(x,y,k\sigma)-G(x,y,\sigma)$$

(Разница Гауссиан)




Difference of Gaussian = DoG

# Эффективная реализация (DoG)



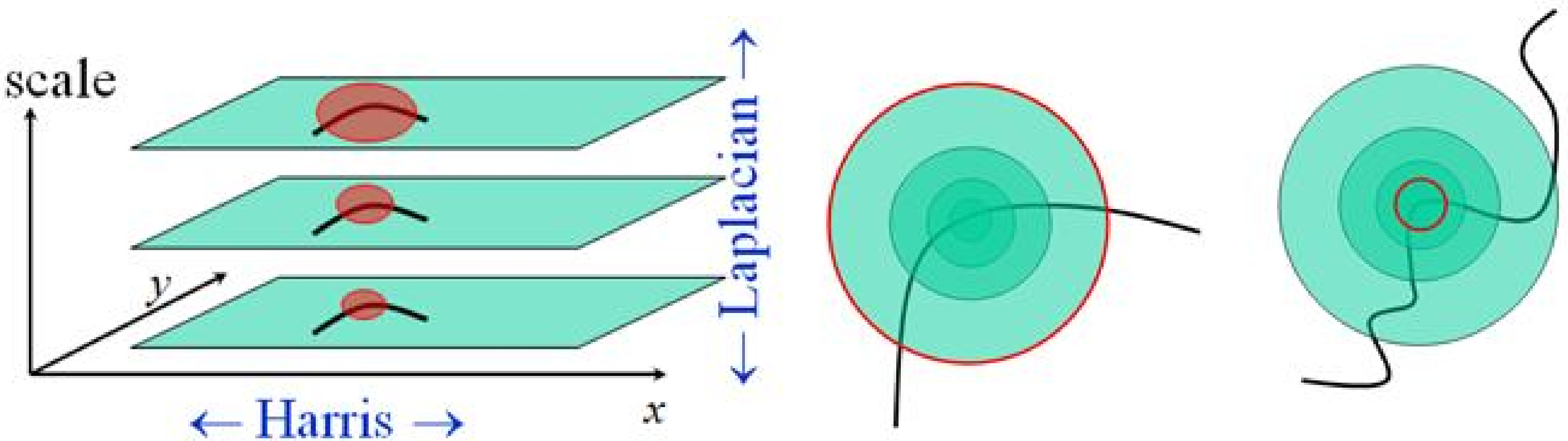

Детектор DoG также выделяет «блобы» на изображении


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


Slide by S. Lazebnik

# Детектор Harris-Laplacian

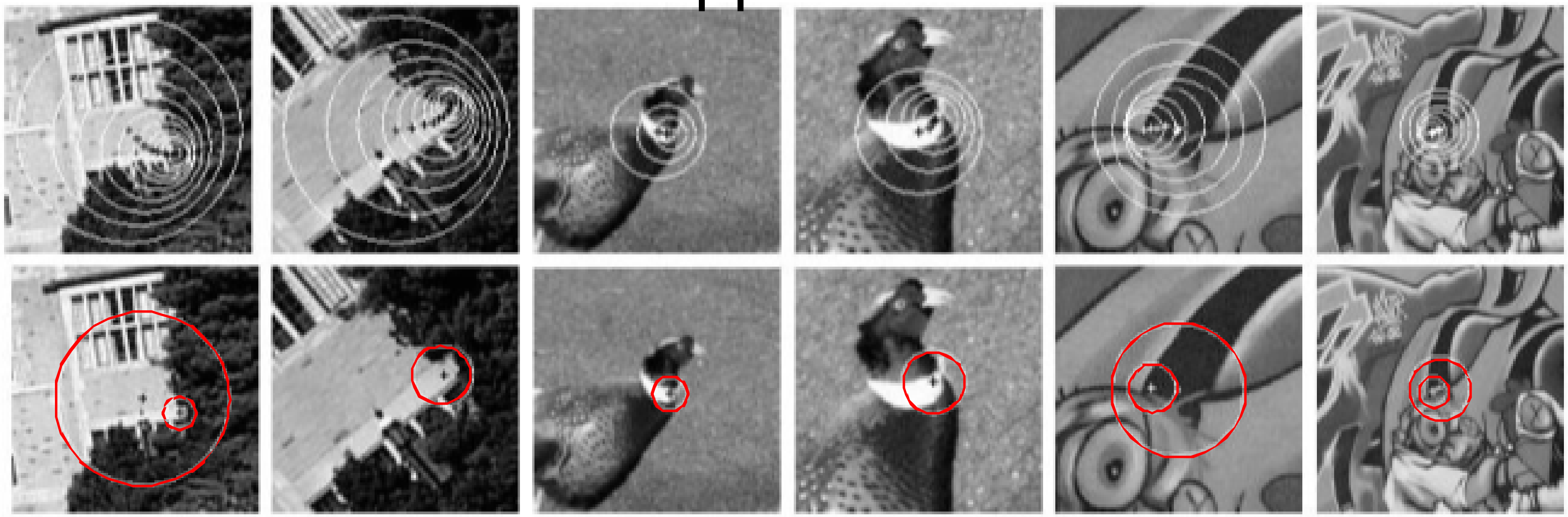
- Выделяем углы на изображении, но с характеристическим размером
- Максимизация:
  - По изображению – откликов углов Харриса
  - По масштабу – Лапласиана
- Разные варианты чередования вычисления функции Харриса и Лапласиана



# Сравнение

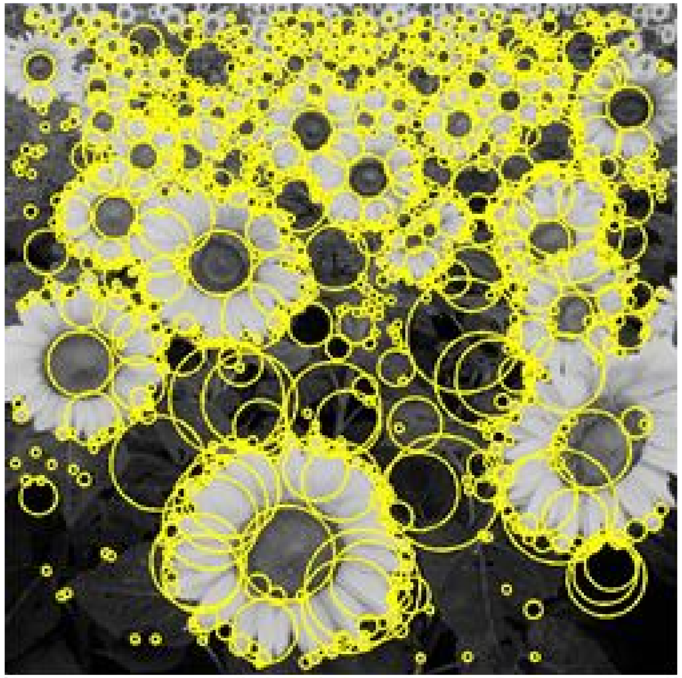
- Сравнение простого детектора Харриса и Харрис-Лапласиана

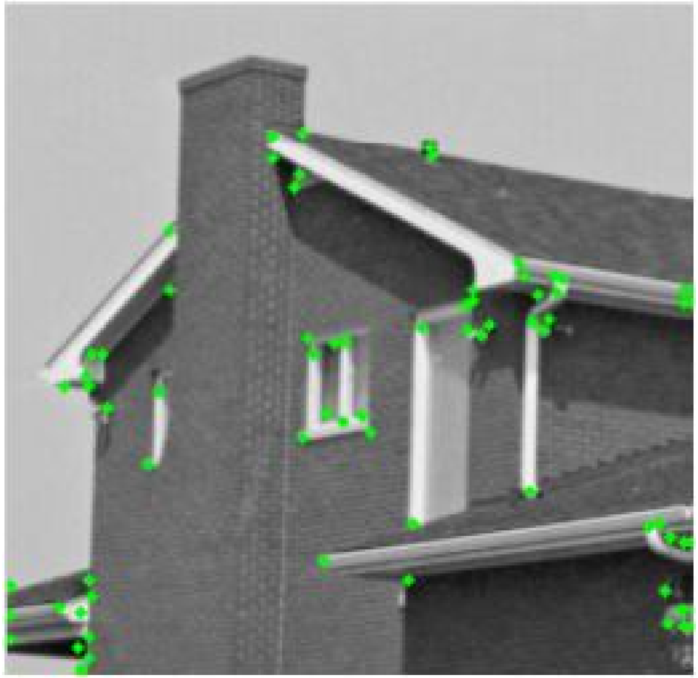
Харрис



Харрис-Лаплас

# Углы и блобы

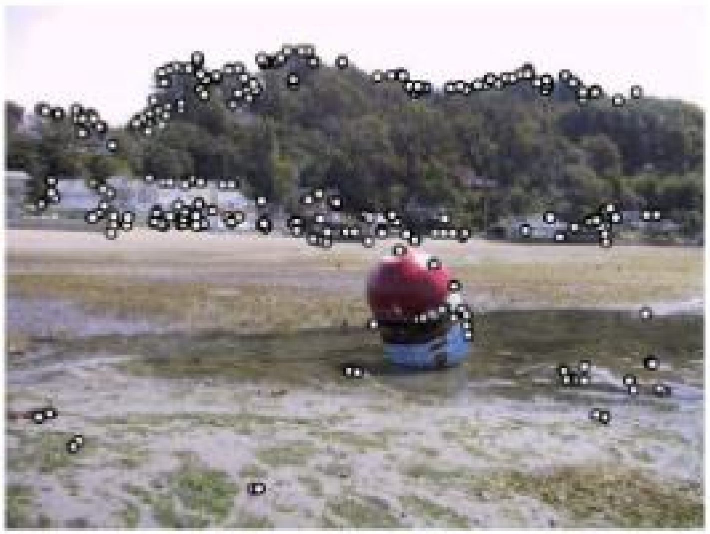




- Углы и блобы – разные виды локальных особенностей
- Детекторы Харрис-Лапласиан и LoG (DoG) выделяют разные множества особенностей
- Можно применять их одновременно

# Выбор точек

- Цель: выбрать фиксированное кол-во точек на изображении
  - Точки должны быть равномерно распределены по изображению
  - Самые сильные отклики обычно расположены в текстурированных областях, неравномерно распределенных по изображению

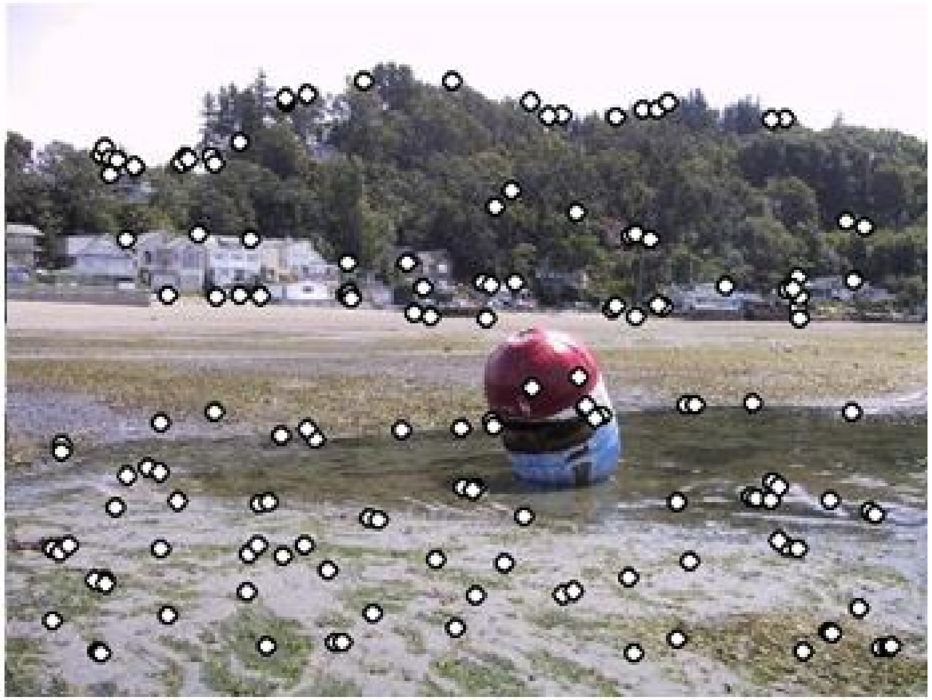


(a) Strongest 250

(b) Strongest 500

# Адаптивный радиус

- Пройдёмся по всем точкам в порядке качества
- Для каждой точки выкинем из списка всех соседей в окрестности радиуса r
- Посчитаем количество оставшихся точек
- Выберем такой радиус r, при котором получим нужное нам количество точек



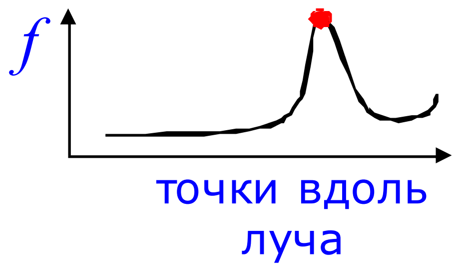
# Детекторы областей

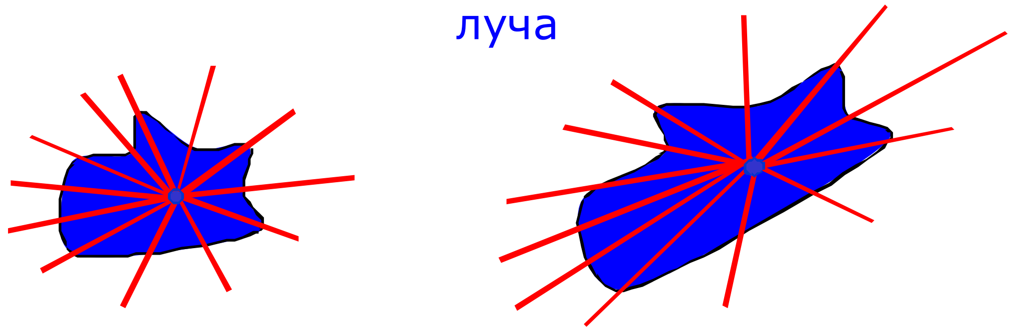
- Стоит попробовать работать с более уникальными характеристиками изображения – с областями
- Интересных областей гораздо меньше, но они более точно характеризуют сцену или объект

# Детекторы областей

- Детектор областей IBR (Intensity-extrema based regions)
- Идти от локального экстремума яркости по лучам, считая некоторую величину f
- Остановка при достижении пика f

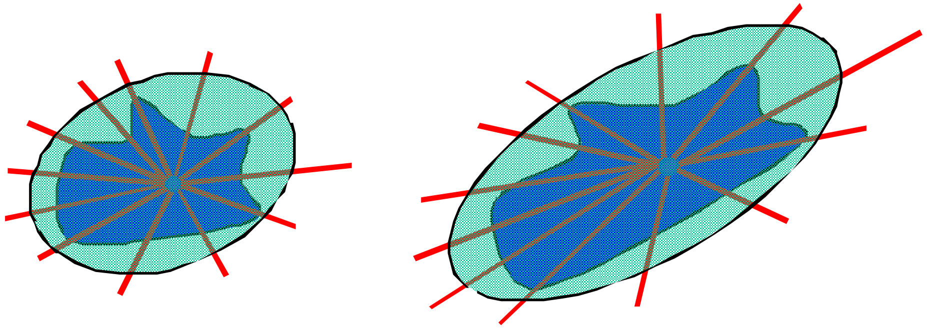
$$f(t) = \frac{\left| I(t) - I_0 \right|}{\frac{1}{t}\int_o^t \left| I(t) - I_0 \right| dt}$$



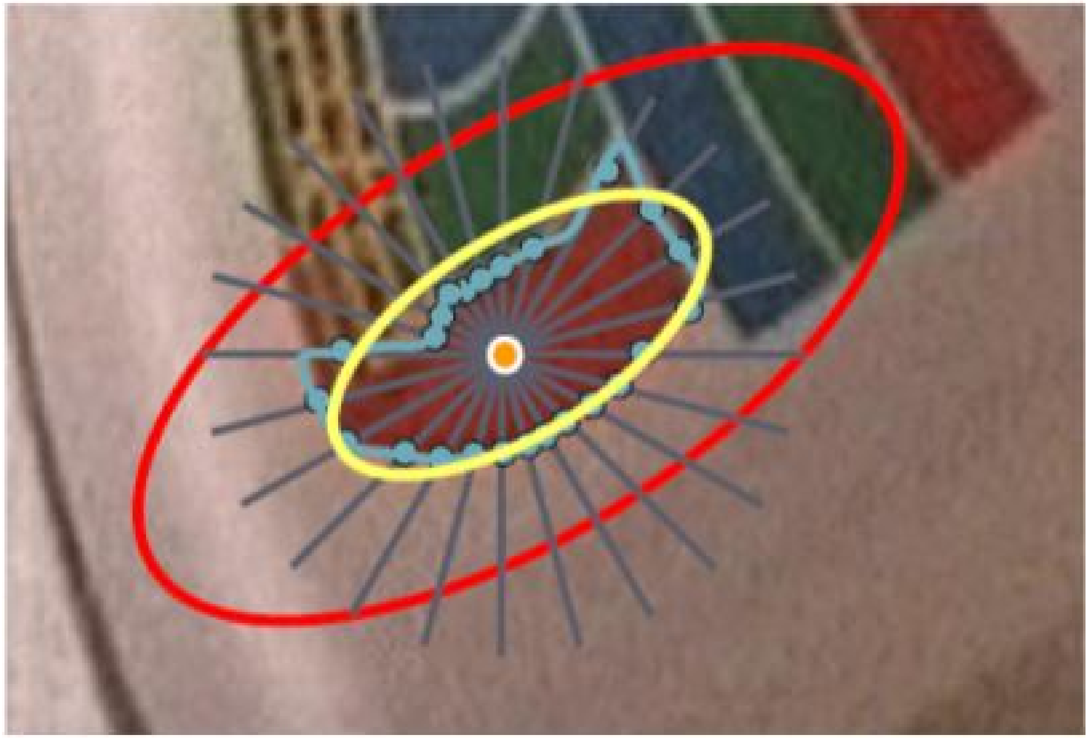
# Детекторы областей

- Области на паре изображений могут различаться, поэтому опишем вокруг них эллипсы



- Если эллипсы превратить в окружности, то получим полное сходство с точностью до поворота (об этом позже)

# Детекторы областей

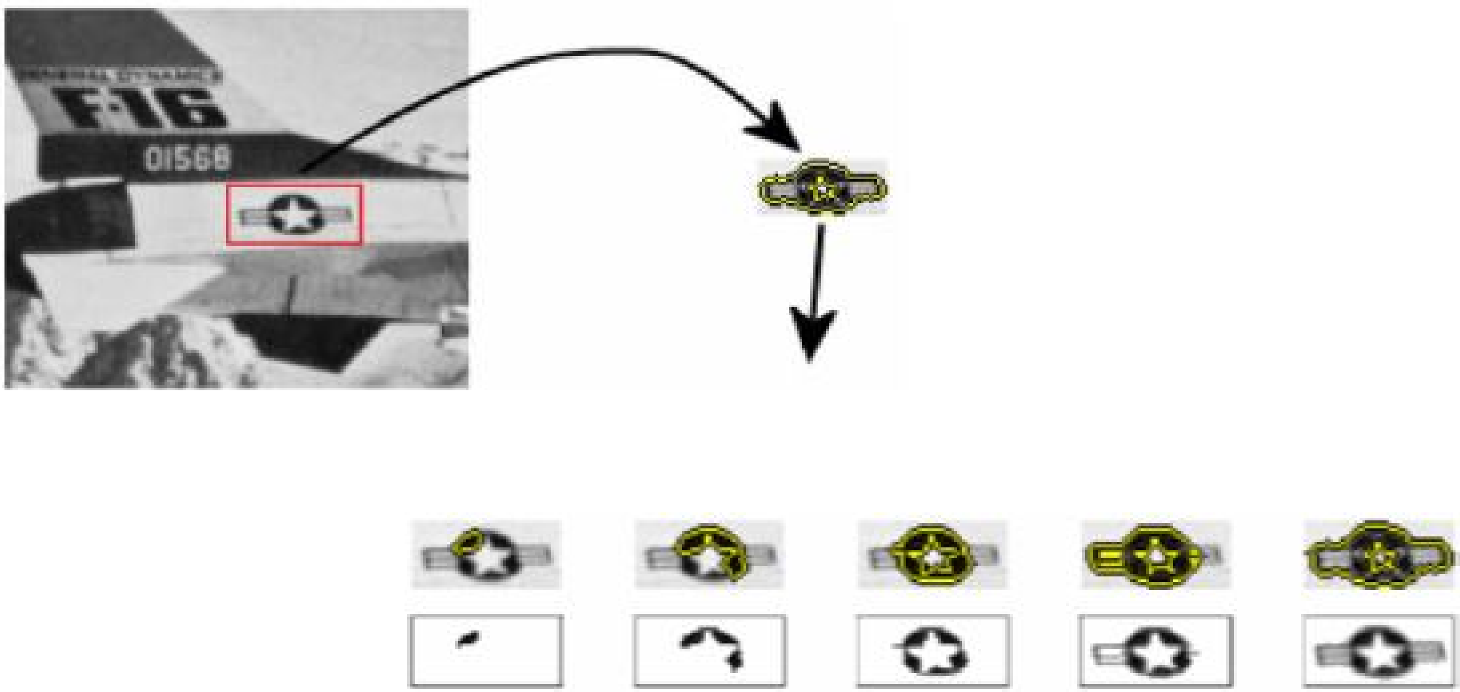


# Детекторы областей

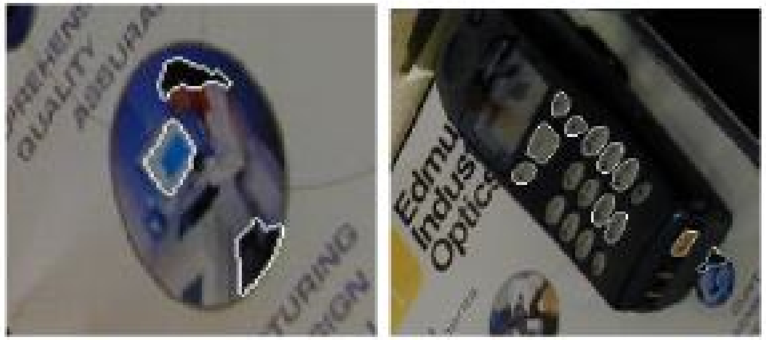
MSER = Maximally Stable Extremal Regions

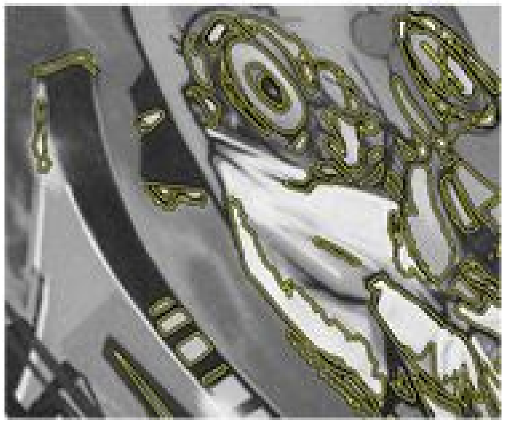
- Задать порог яркости T
- Провести сегментацию
- Извлечь области
- Для каждой области найти порог, при котором рост площади минимален
- Описать вокруг области эллипс

# Детекторы областей



# Детекторы областей



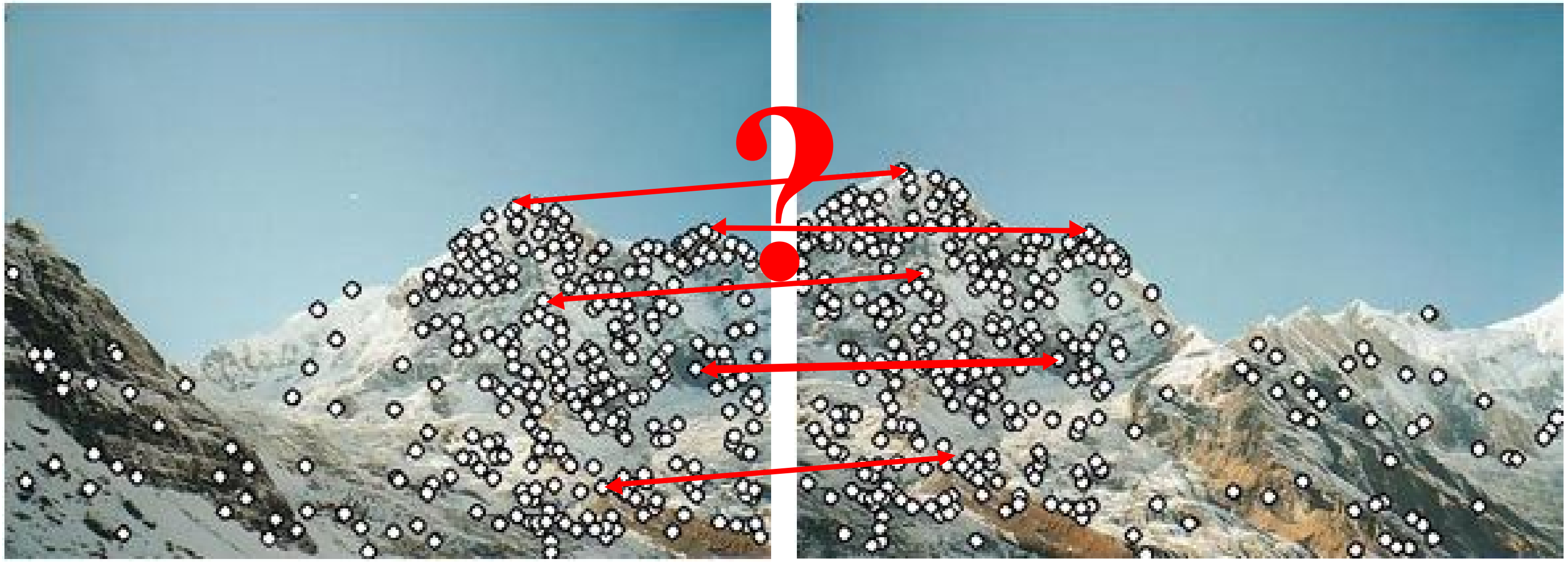


# Резюме локальных особенностей

- Локальные особенности – один из основных инструментов анализа изображений
- Рассмотрели алгоритмы выделения особенностей:
  - Уголки: Harris (Forstner), Harris-Laplace)
  - Блобы: LoG (Laplacian of Gaussian), DoG (Difference of Gaussians)
  - Области: IBR (Intensity-extrema based regions), MSER (Maximally Stable Extremal Regions)

# Дескрипторы

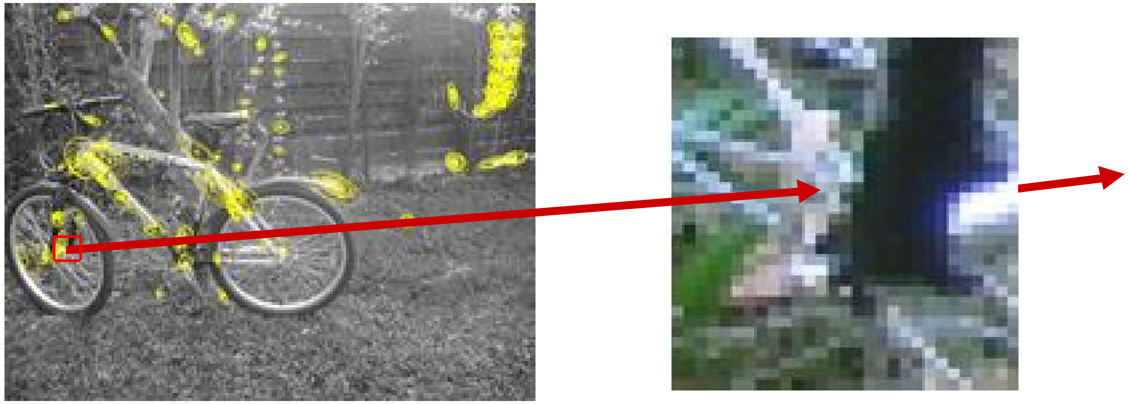
Точки найдены – как их сопоставить?




- 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:

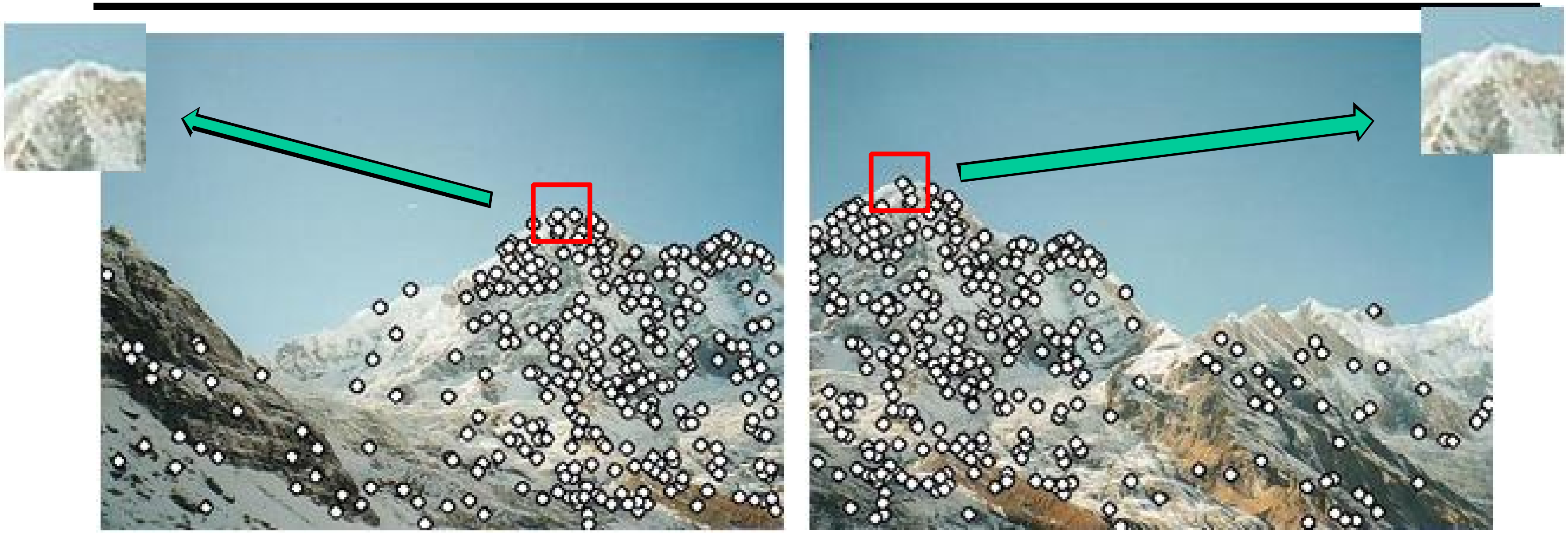


$$\mathbf{f}_n = (f_{n,1}, \cdots, f_{n,j})^T$$

Как будем поступать:

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

# Простейший подход

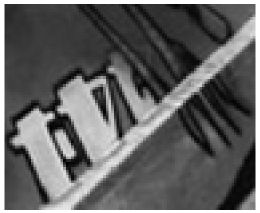


- Возьмём квадратные окрестности, со сторонами, параллельными строкам и столбцами изображения
- Яркости пикселов будут признаками
- Сравнивать будем как изображения попиксельно (SAD, SSD)
- Такая окрестность инвариантна только к сдвигу изображения

# Инвариантность к яркости

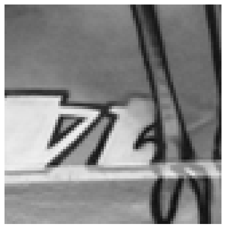
- Можем добиться следующим образом:
  - Локальная нормализация гистограммы
  - Дескрипторы, основанные на градиенте яркости, инвариантны к сдвигу яркости
  - Нормирование яркости - вычесть среднее значение, поделить на дисперсию

$$I' = (I - \mu)/\sigma$$

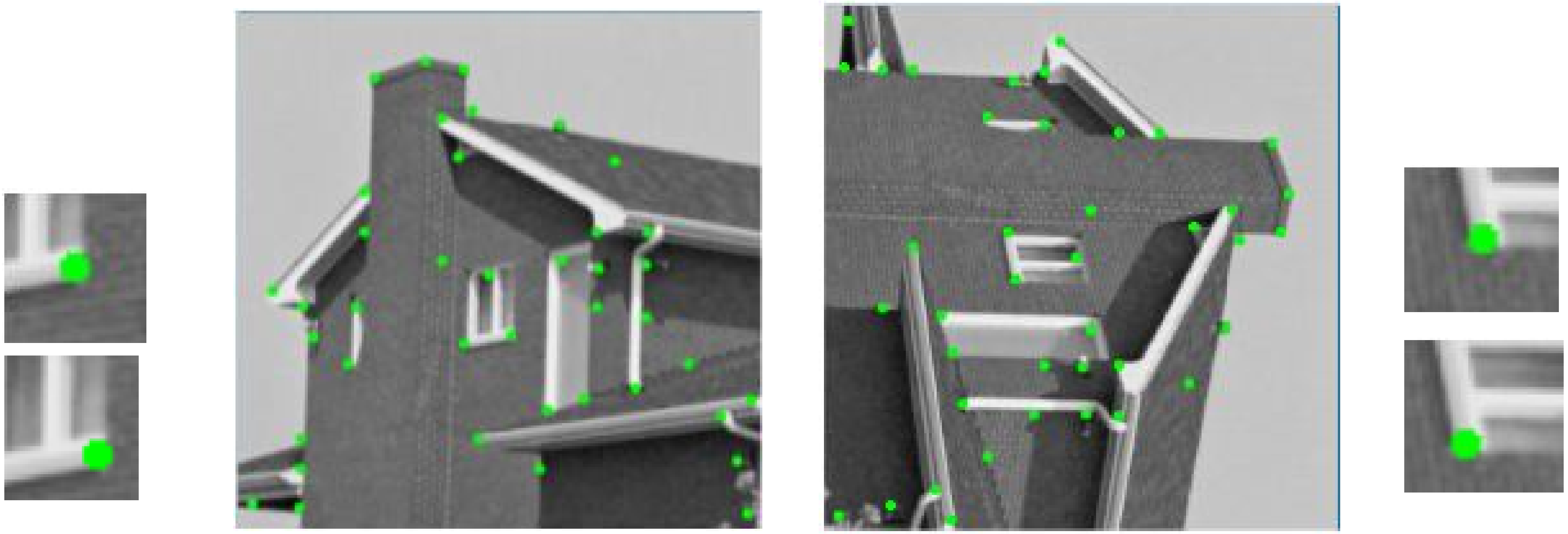


нормализация

$\rightarrow$



# Недостаток простой окрестности



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

# SIFT

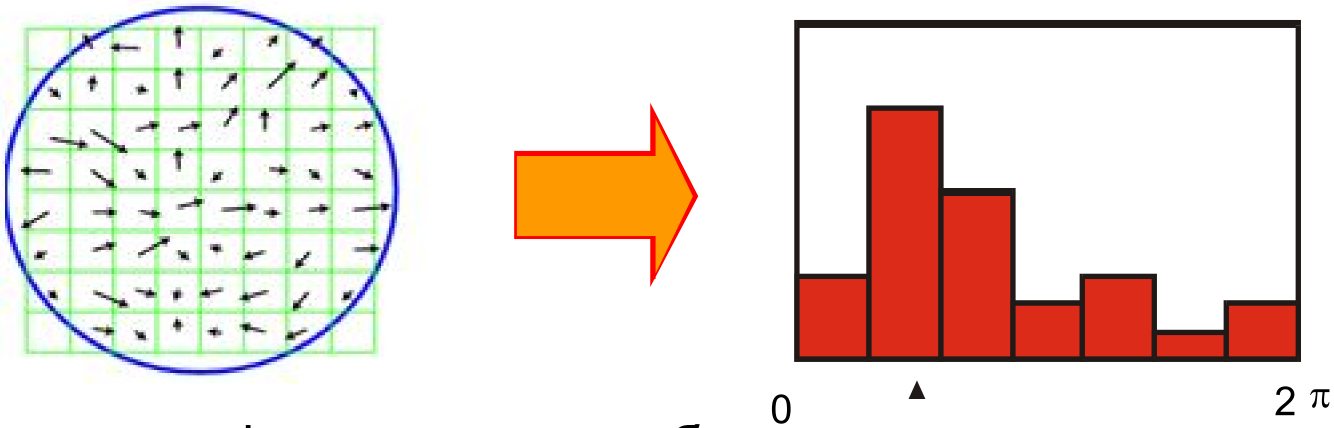
- Scale-Invariant Feature Transform:
  - Детектор DoG
    - Определение положения и масштаба особенности
  - Ориентация
    - Определение доминантной ориентации по градиентам
  - Дескриптор
    - Использование статистик по направлению градиентам

- Устойчив к изменениям освещенности и небольшим сдвигам

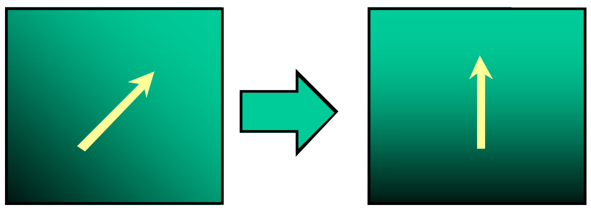
David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

# Ориентация

- Идея: найти основное (доминантное) направление градиентов пикселей в окрестности точки
- Посчитаем гистограмму, взвешивая вклад по гауссиане с центром в точке




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

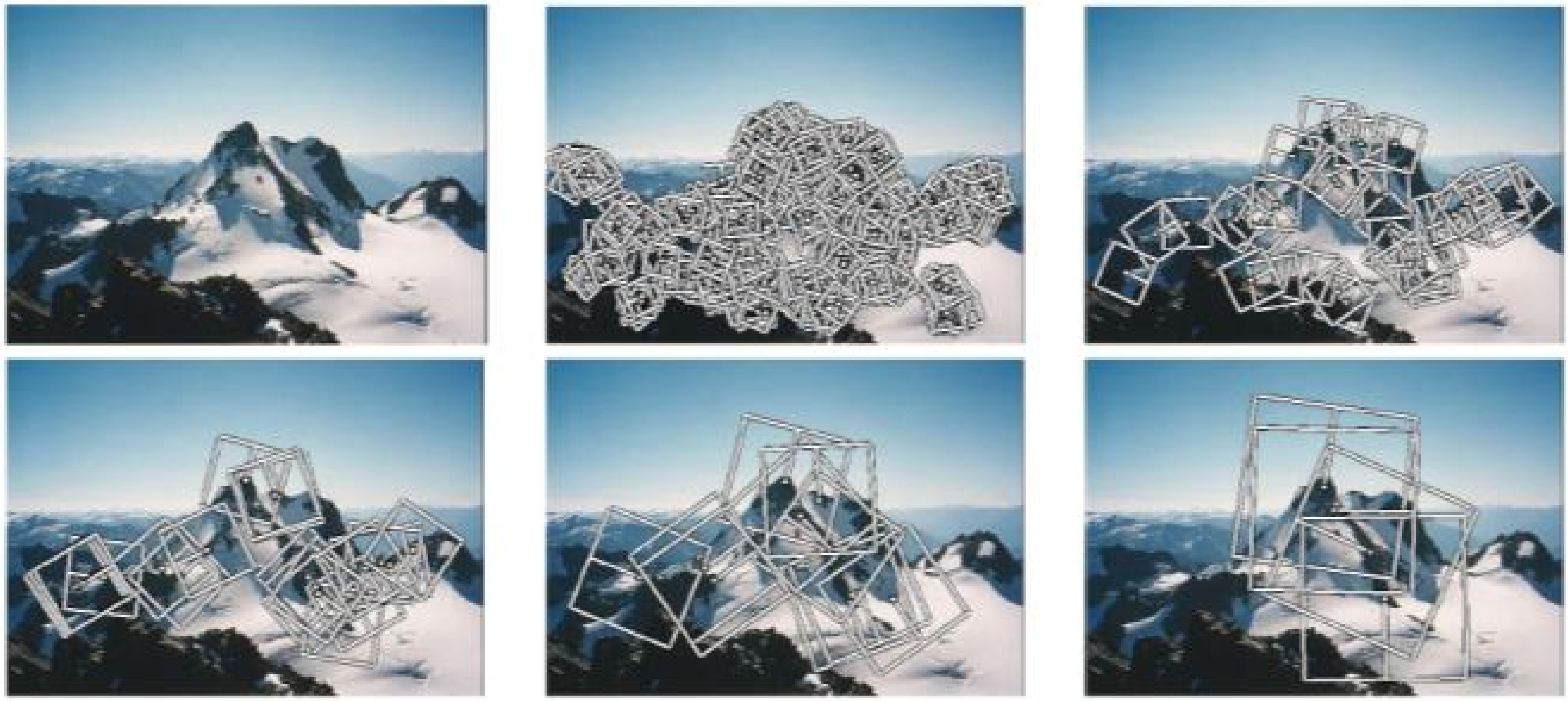


- Если локальных максимумов несколько – считаем, что несколько точек с разной ориентацией

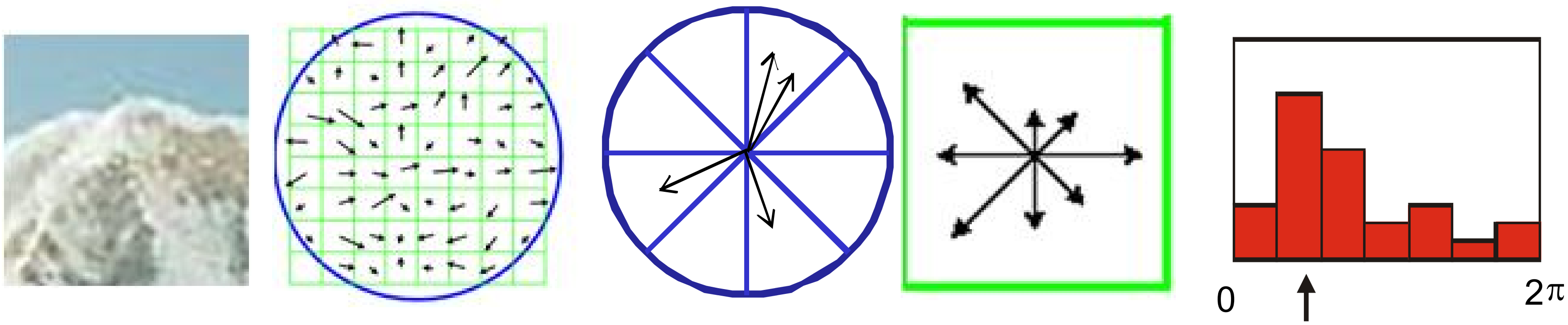
# Окрестность особенности

- 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

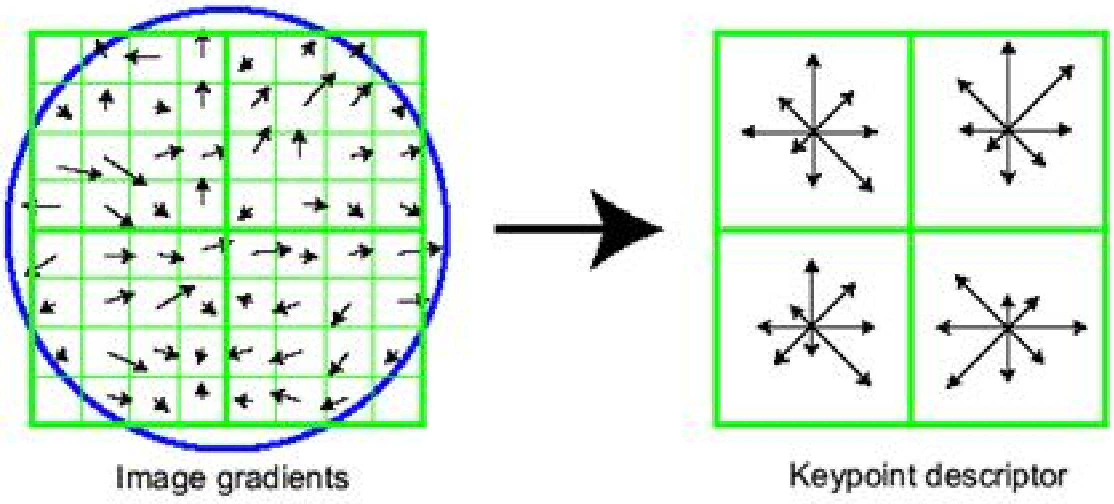


# Гистограмма ориентаций градиентов




- Построим дескриптор SIFT – идея:
  - 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
    - Вклад взвесим по гауссине, с центром в центре окрестности

# Гистограммы градиентов




- Для учета локальных 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)
- Сравниваем как вектор (разные метрики)

# Сравнение SIFT

- Вектор-признак длиной 128, по сути гистограмма

- Стандартные L1, L2
- Специальные для гистограмм:
  - Пересечение гистограмм $$D(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i))$$
  - Расстояние $\chi^2$ $$D(h_1, h_2) = \sum_{i=1}^{N} \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i) + h_2(i)}$$
  - Earth-Mover Distance

# Использование цвета

- RGB-SIFT
  - 3 дескриптора SIFT для каждого канала
- C-SIFT
  - Каналы $0_1$ и $0_2$

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$$

- rgSIFT
  - Каналы r и g

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek, **Evaluating Color Descriptors for Object and Scene Recognition**, *IEEE PAMI*, 2010

# 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

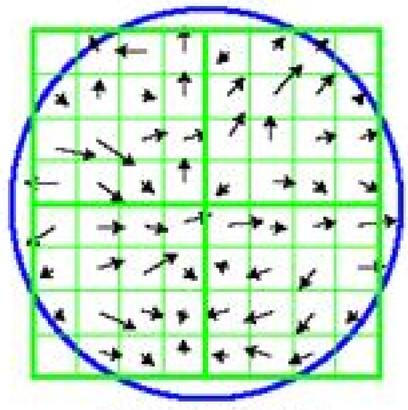
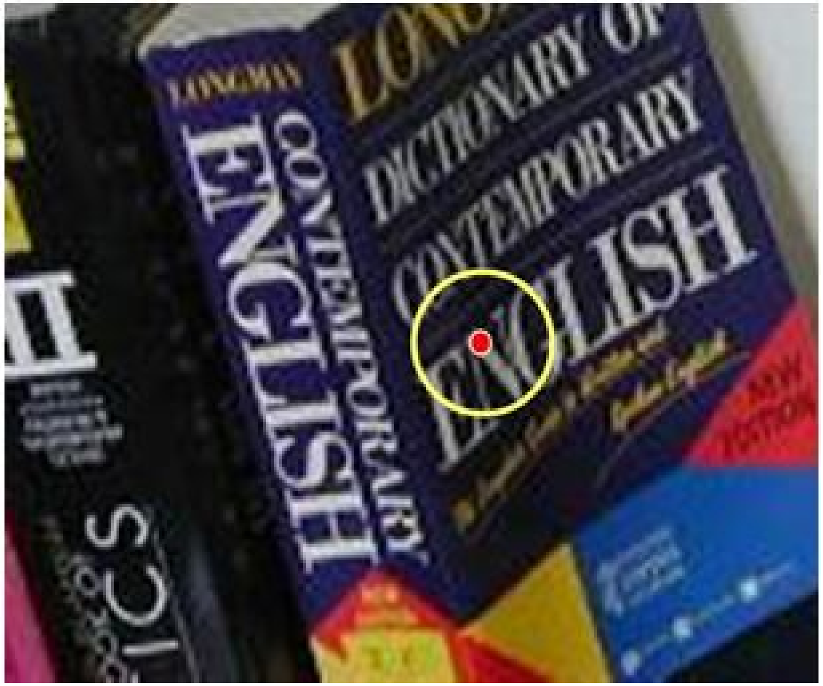
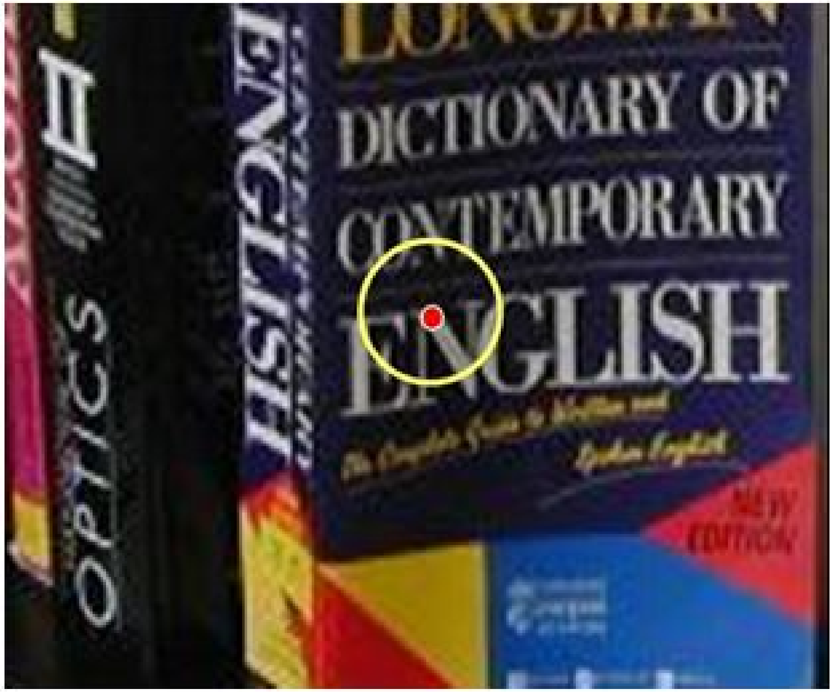


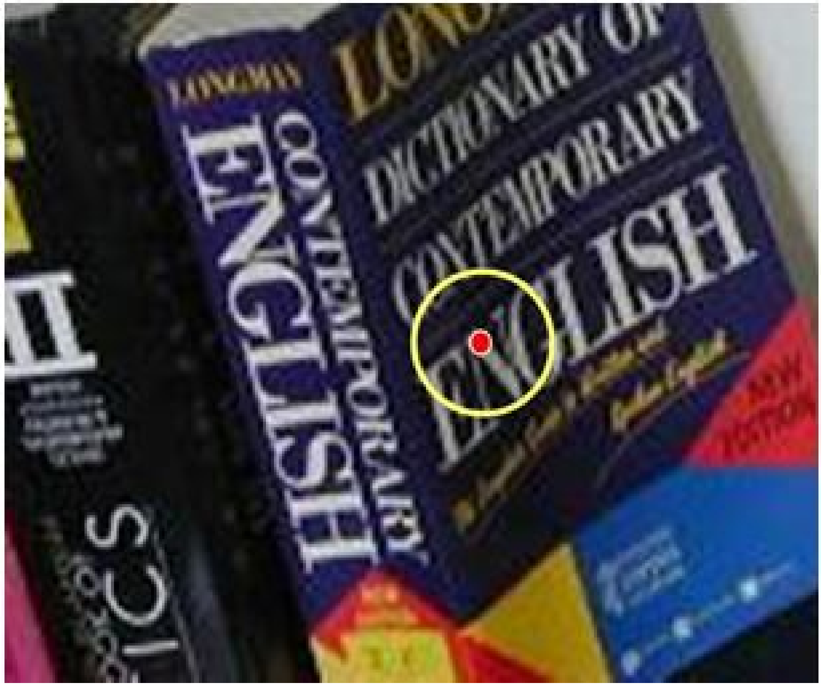
Image gradients

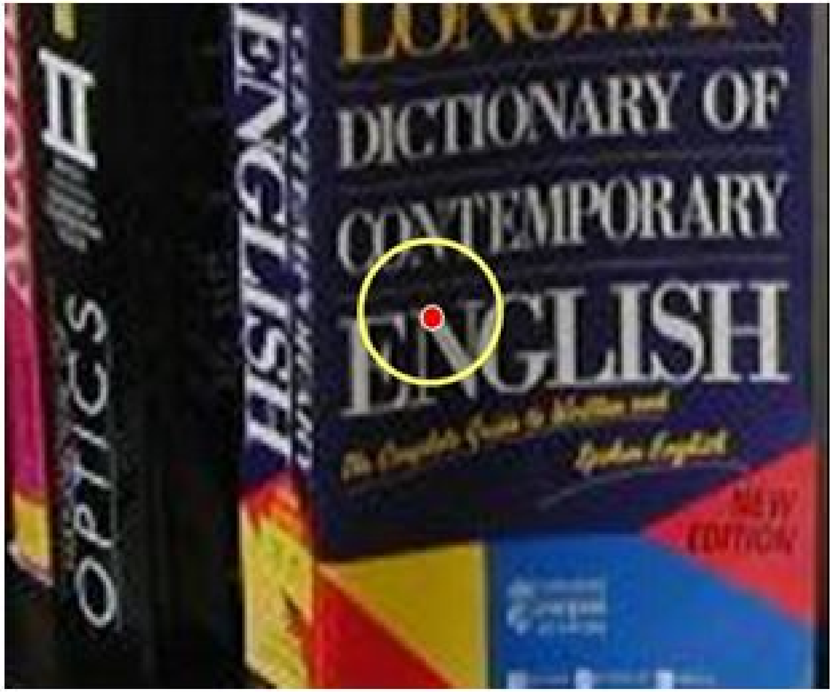
# Перспективные искажения



# Дескрипторы



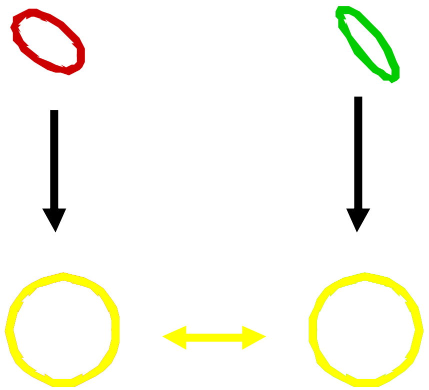



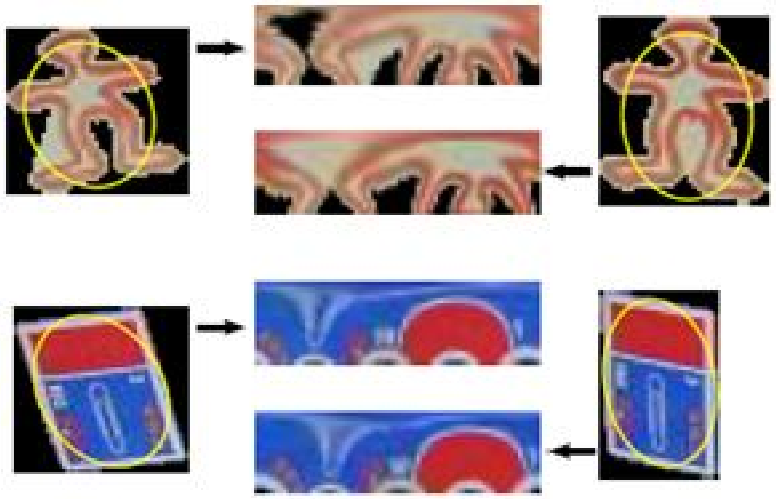

В круглую окрестность попадают разные фрагменты – в левом снимке внутрь окружности попала половина буквы G, в правом он почти не попала

# Дескрипторы


Найти соответствующие окрестности, с учетом аффинных преобразований описав их эллипсом

# Нормализация окрестности





Для облегчения сравнения фрагментов изображения необходимо найти параметры эллипса вокруг интересной точки или области и привести эллипсы к «каноническому» виду – «общему знаменателю»

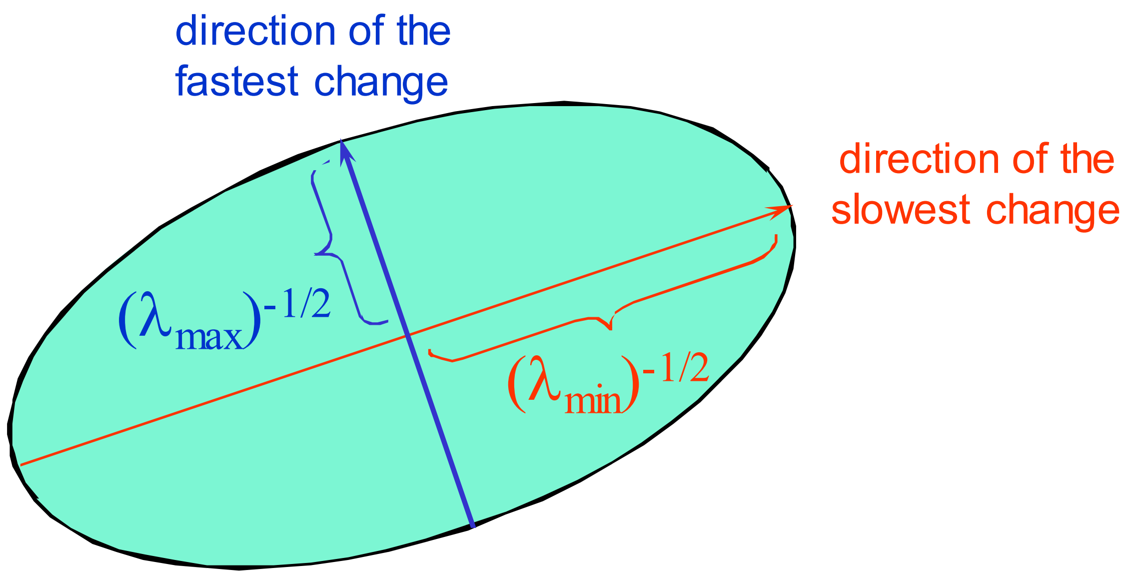
# Аффинная адаптация

Помним: $$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

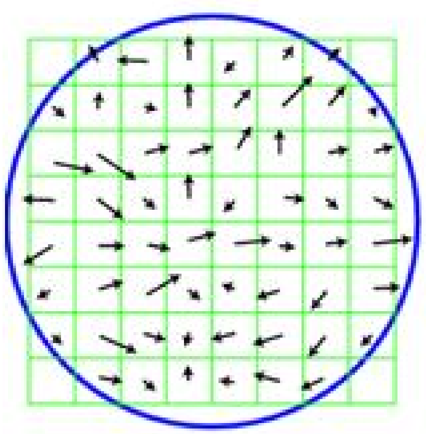
Матрицу *M* можно представить как эллипс, у которого длины осей определены собственными значениями, а ориентация определена матрицей R

Уравнение эллипса:

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



# Афинная адаптация

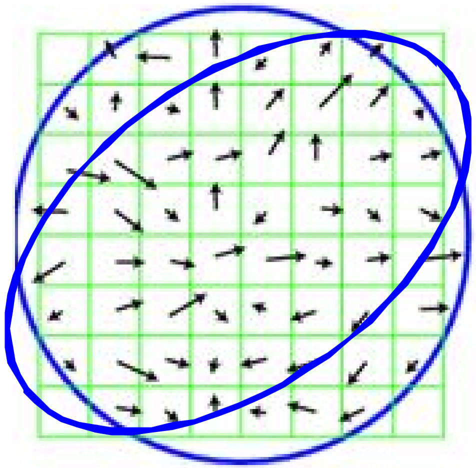



**Проблема:** мы матрицу М считаем по круглой (квадратной) окрестности. На разных изображениях содержимое будет не совпадать, и мы не сможем выделить одинаковые области (эллипсы)

**Решение:** итеративная адаптация

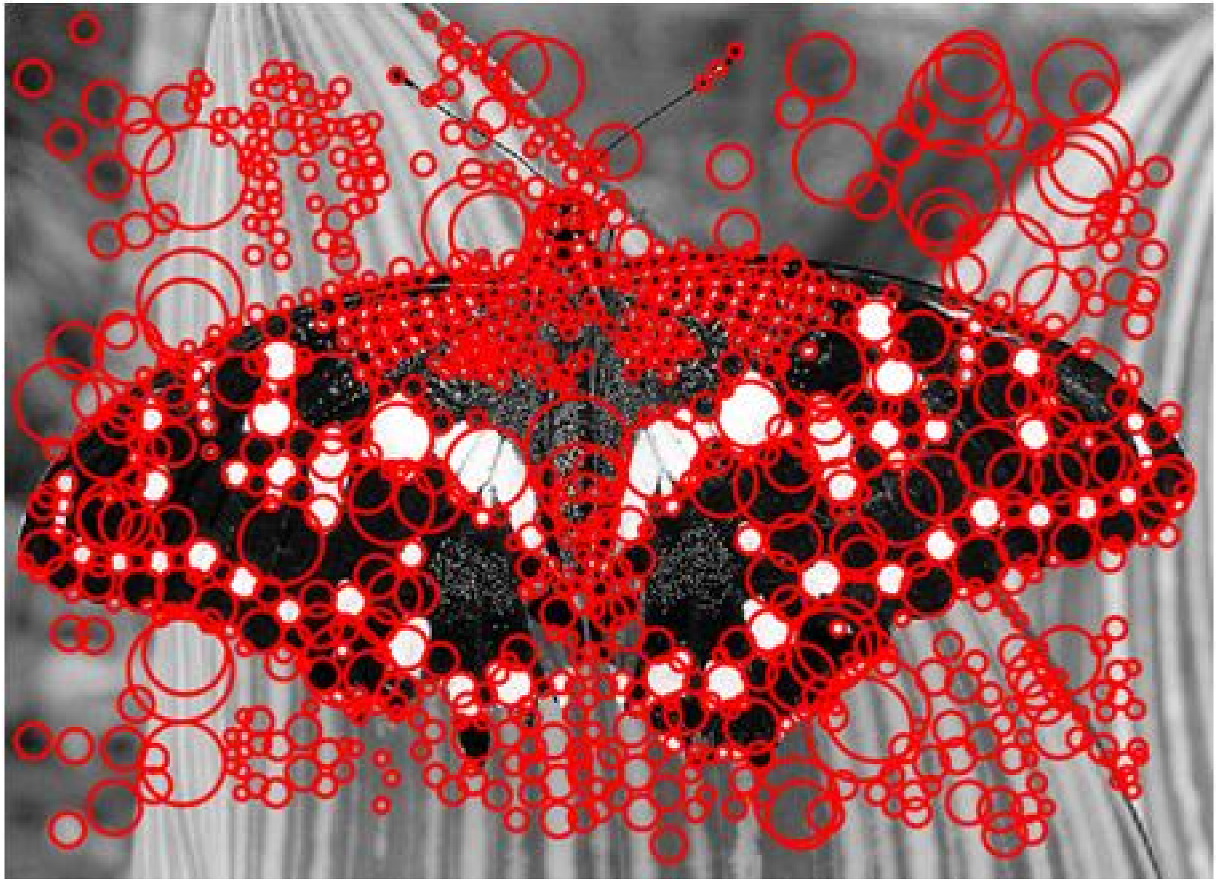
# Аффинная адаптация

- Задача: матрица вторых моментов, определенная весами w(x,y) должна вычисляться по характерной формой области
- Решение: итеративное уточнение
  - Считаем матрицу моментов по круглому окну
  - Применяем аффинную адаптацию для получения эллиптического окна
  - Пересчитываем матрицу моментов по нормализованной окрестности. Повторяем.



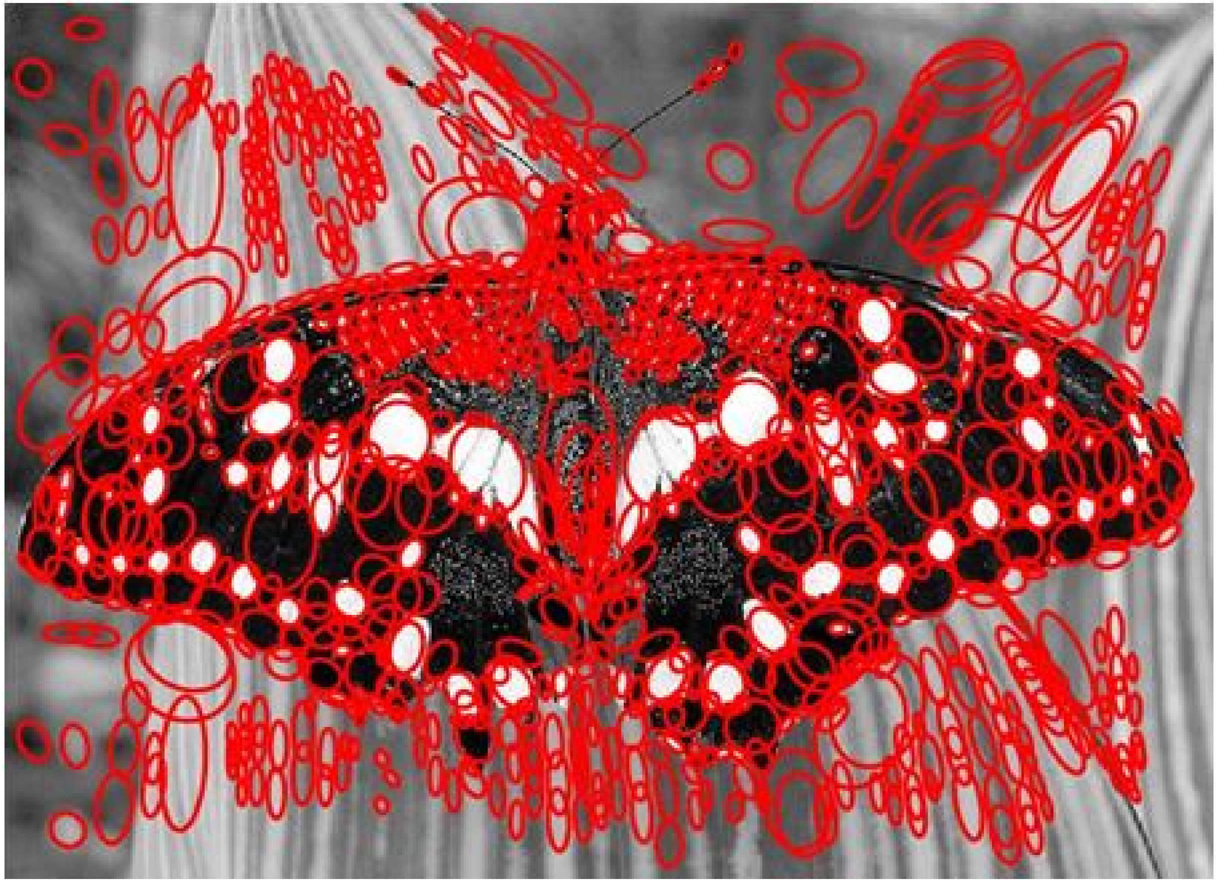
Slide by S. Lazebnik

# Пример аффинной адаптации



Независимые от масштаба области (блобы)

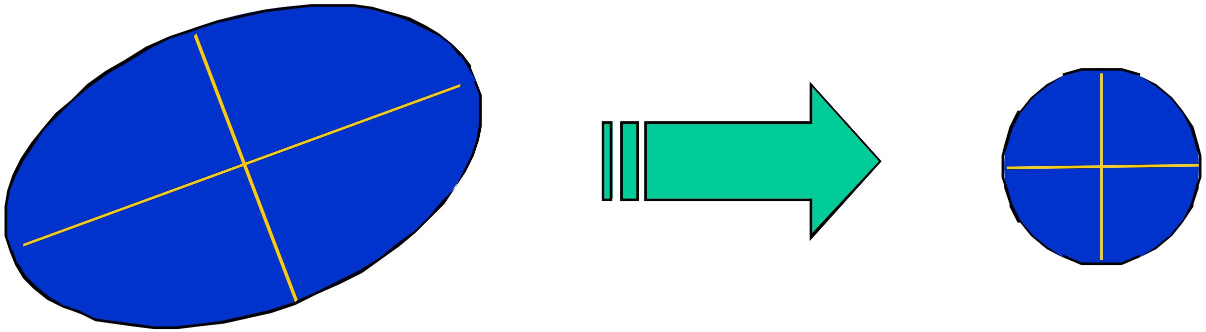
# Пример аффинной адаптации



Уточненные окрестности блобов

# Нормализация

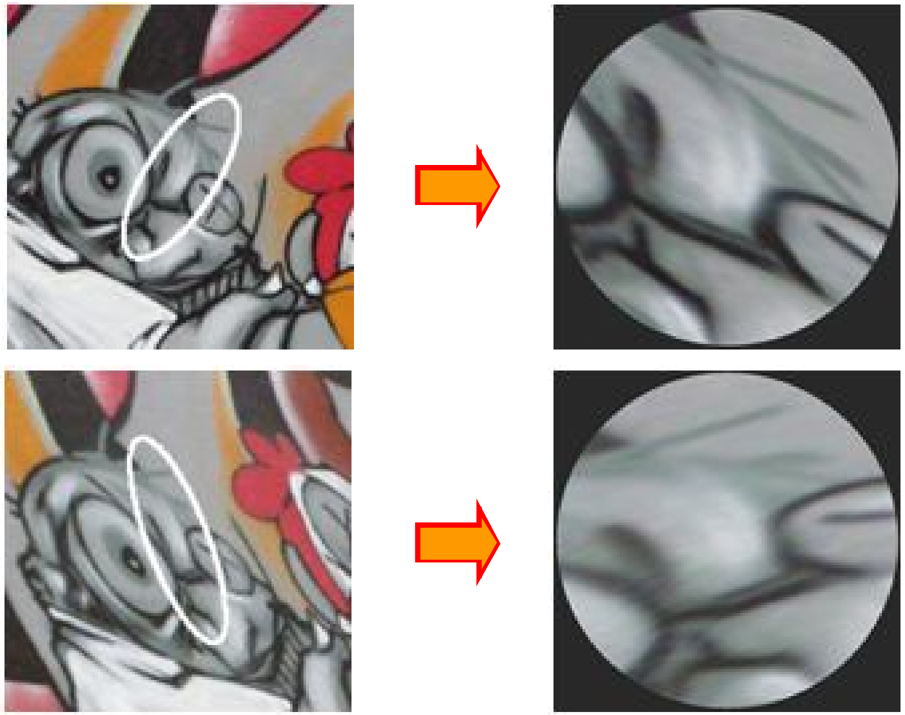
- Эллипс вторых моментов можно считать «характеристической формой» области
- Нормализуем окрестности путем преобразования эллипса в единичный круг



Slide by S. Lazebnik

# Неопределенность ориентации

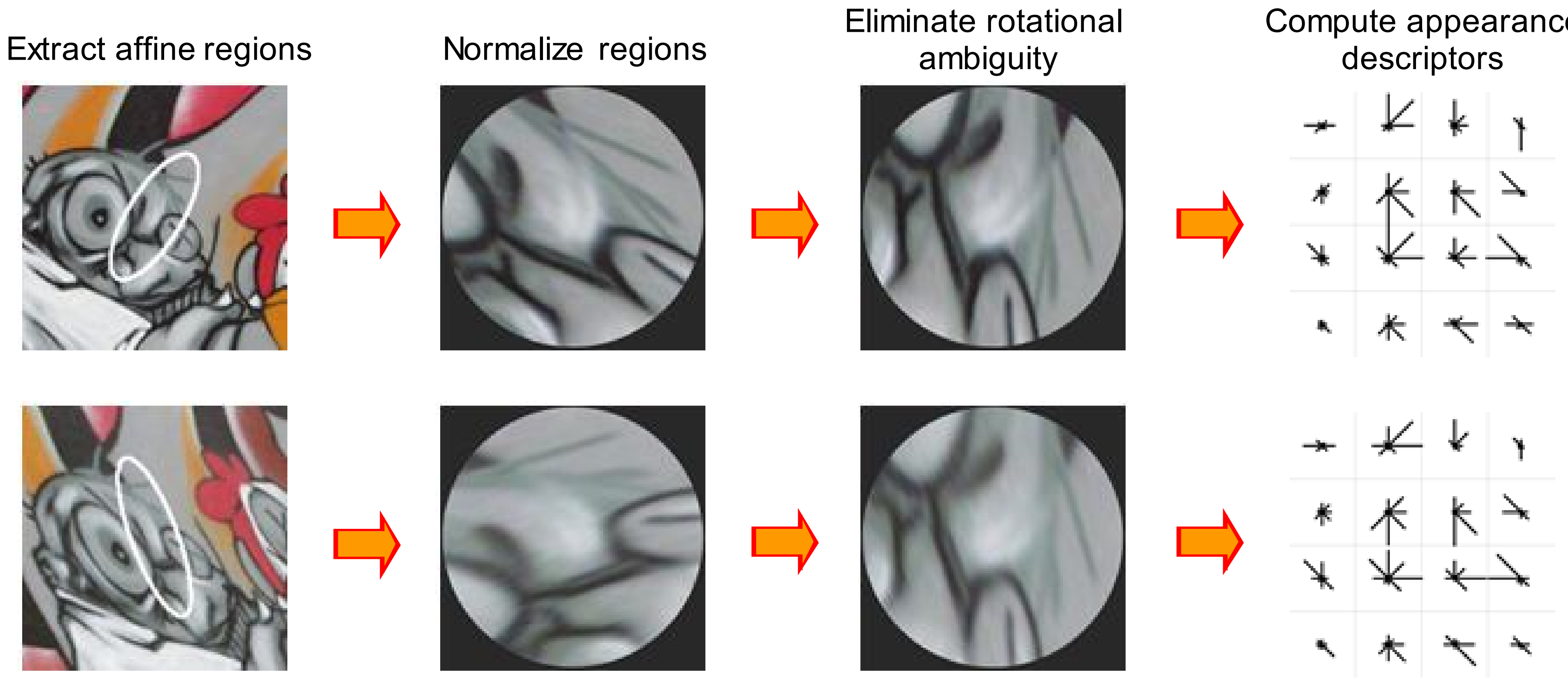
- Нет уникального преобразования из эллипса в единичный круг
  - Мы можем вращать и отражать единичный круг, и он останется единичным кругом



- Посчитаем доминирующий градиент и довернём!

Slide by S. Lazebnik

# Пример



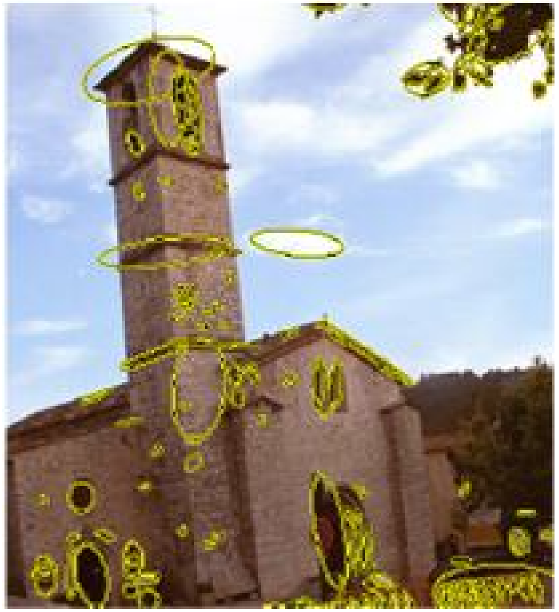

K. Mikolajczyk and C. Schmid, [Scale and Affine invariant interest point detectors](http://www.robots.ox.ac.uk/~vgg/research/affine/), IJCV 60(1):63-86, 2004.

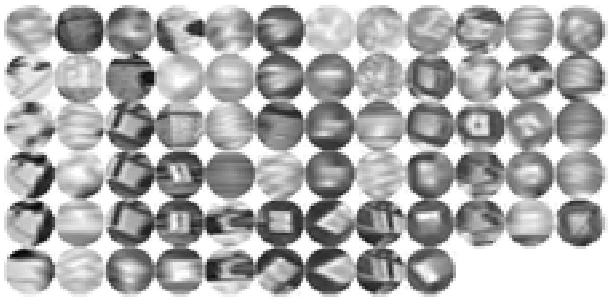
http://www.robots.ox.ac.uk/~vgg/research/affine/

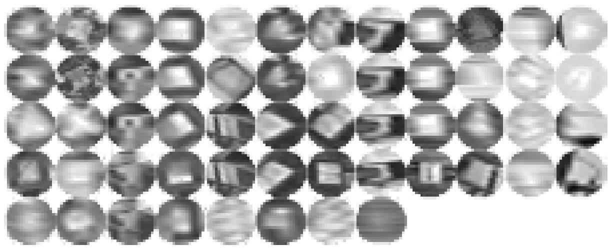
Slide by S. Lazebnik

# Сопоставление

- Имеем набор точек и дескрипторов
- Как будем сопоставлять?

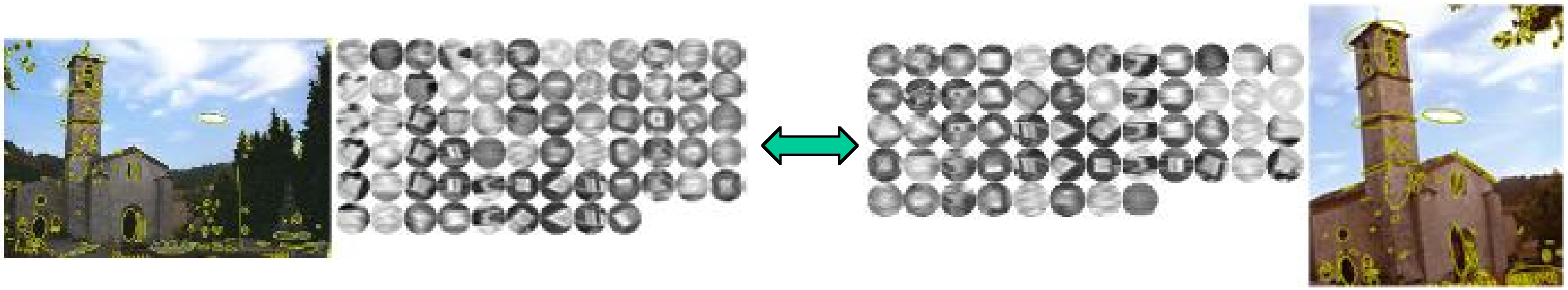






# Сопоставление особенностей

- Генерируем пары-кандидаты: для каждого патча в одном изображении, находим несколько наиболее похожих по выбранной метрики пачтей на другом изображении
- Как выбирать пары?
  - Полный перебор
    - Для каждой особенности вычисляем расстояния до всех особенностей второго изображения и берем лучшую
  - Ускоренные приближенные меры
    - Иерархические структуры (kd-trees, vocabulary trees)
    - Хэширование



# Резюме лекции

- Сопоставление изображений – одна из основных задач в компьютерном зрении

- Прямое сопоставление иногда работает, но страдает от ряда недостатков

- Локальные особенности – один из основных инструментов для анализа изображений
  - Детекторы: Harris, LoG, DoG, Harris-Laplace
  - Дескрипторы: SIFT, C-SIFT
  - Афинная адаптация

# На следующей лекции

- Геометрические модели – прямые, окружности, преобразование изображений
- DLT-метод для оценки их параметров
- Робастные методы оценки параметров
  - Рандомизированные методы
  - Схемы голосования

- Выделение экземпляров объектов через сопоставление