

Лекция 3

Edge detection. Особые точки.
Детектор Харриса. SIFT, HoG.

21.01.2020
Руслан Алиев

План занятия

- Границы объектов

Основные свойства, оператор Собеля, детектор Кэнни

- Локально-инвариантные признаки

Зачем это нужно, что такое инвариантность, преобразования изображения

- Особые точки изображения

Почему углы это круто, Детектор Харриса

Edge detection



Source: D. Lowe

Edge detection

Задача: выделить резкие изменения (разрывы) в изображении

Интуитивно понятно, что большая часть информации об изображении содержится в краях:

- Соответствует человеческому восприятию мира
- Компактнее, чем пиксели

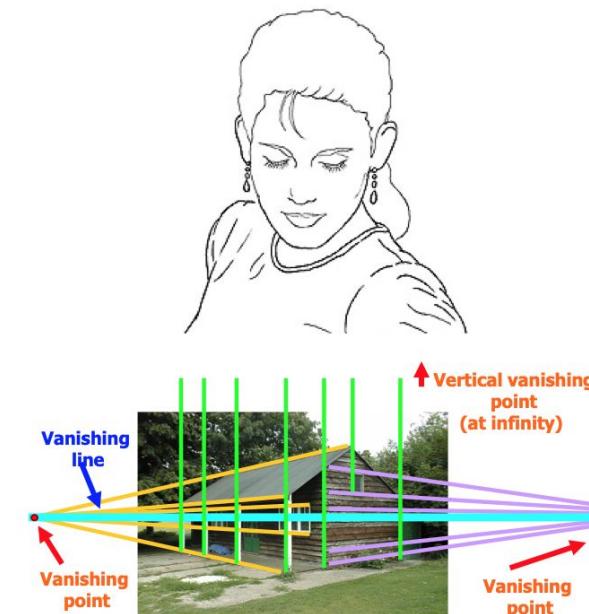
Идеал: рисунок художника (но ему легче - он уже знает, какой объект рисует, есть априорная информация)



Source: D. Lowe

Зачем нам нужны границы изображения?

- Выделять признаки, распознавать/детектировать/сегментировать объекты
- Геометрические преобразования: находить точку съемки, восстанавливать 3D модель изображения



Откуда берутся границы объектов



Source: D. Hoiem

Откуда берутся границы объектов



Разрыв поверхности

Разрыв глубины

Изменение цвета

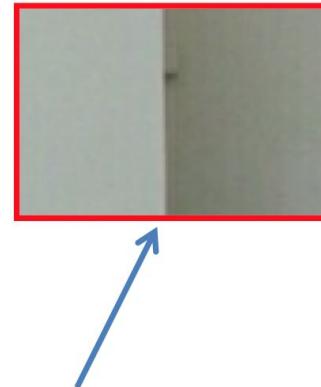
Изменения освещения

Source: D. Hoiem

Откуда берутся границы объектов



Разрыв глубины

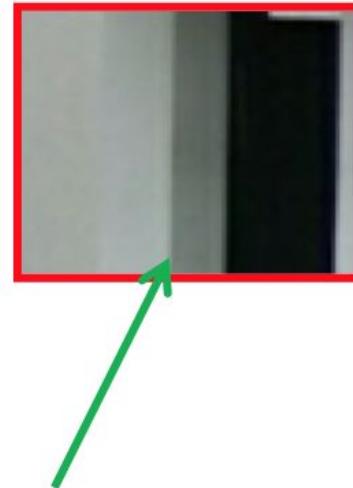


Source: D. Hoiem

Откуда берутся границы объектов



Разрыв поверхности

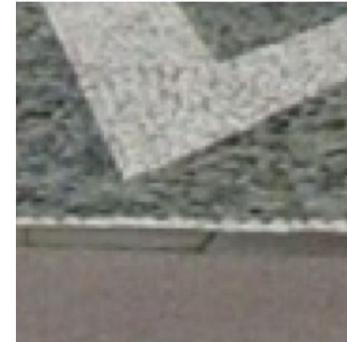


Source: D. Hoiem

Откуда берутся границы объектов



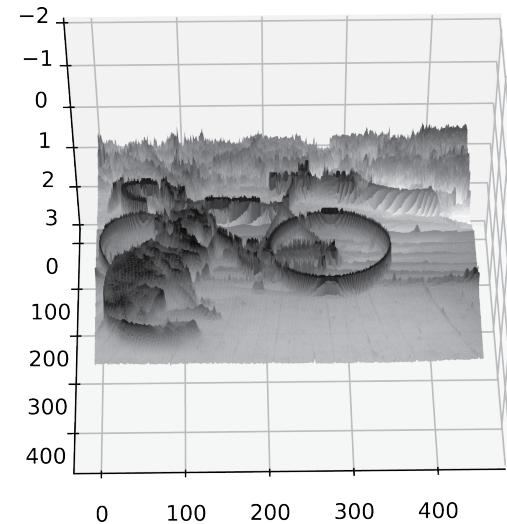
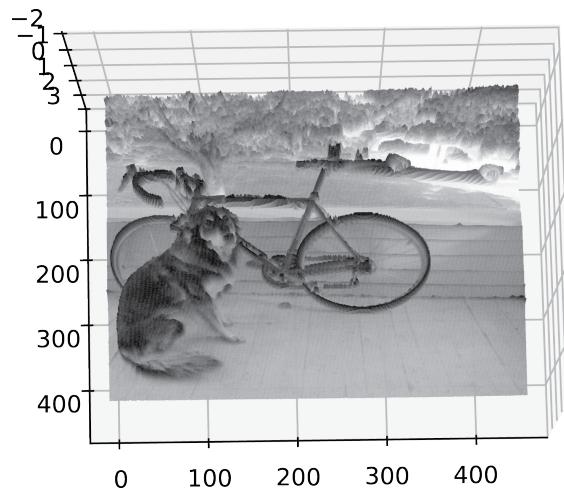
Изменение цвета



Source: D. Hoiem

Что такое граница

- Изображения - это функция
- Граница - резкое изменение значения этой функции
- Изменение функции - производная!



Повторим производные
(на всякий случай)

Производная 1D

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

Производная 1D

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

Производная 1D

$$y = x^2 + x^4$$

$$y = \sin x + e^{-x}$$

$$\frac{dy}{dx} = 2x + 4x^3$$

$$\frac{dy}{dx} = \cos x + (-1)e^{-x}$$

Дискретная производная 1D

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x - 1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x - 1) = f'(x)$$

Дискретная производная 1D

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x - 1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x - 1) = f'(x)$$

Дискретная производная 1D

Backward $\frac{df}{dx} = f(x) - f(x-1) = f'(x)$

Forward $\frac{df}{dx} = f(x) - f(x+1) = f'(x)$

Central $\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$

Дискретная производная 1D

- Backward filter:

$$[0 \quad 1 \quad -1]$$

$$f(x) - f(x-1) = f'(x)$$

- Forward:

$$[-1 \quad 1 \quad 0]$$

$$f(x) - f(x+1) = f'(x)$$

- Central:

$$[1 \quad 0 \quad -1]$$

$$f(x+1) - f(x-1) = f'(x)$$

Дискретная производная 1D (пример)

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

Дискретная производная 1D (пример)

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

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

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

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

?

Дискретная производная 2D

Given function

$$f(x, y)$$

Дискретная производная 2D

Given function

$$f(x, y)$$

Gradient vector

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Дискретная производная 2D

Given function

$$f(x, y)$$

Gradient vector

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude

$$|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$$

Gradient direction

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

Фильтр дискретной производной 2D

Что делает этот фильтр?

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Фильтр дискретной производной 2D

Что делает этот фильтр?

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

А этот?

$$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Фильтр дискретной производной 2D

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Фильтр дискретной производной 2D

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Что получим, если применим этот фильтр к изображению слева?

$$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Фильтр дискретной производной 2D

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Фильтр дискретной производной 2D

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Что получим, если применим этот фильтр к изображению слева?

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Фильтр дискретной производной 2D

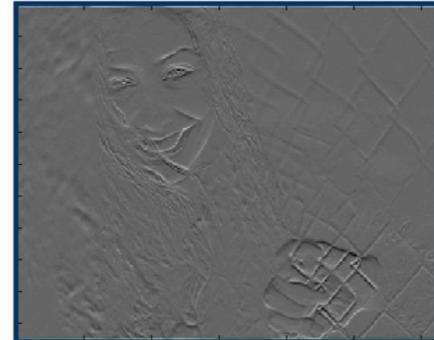
$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

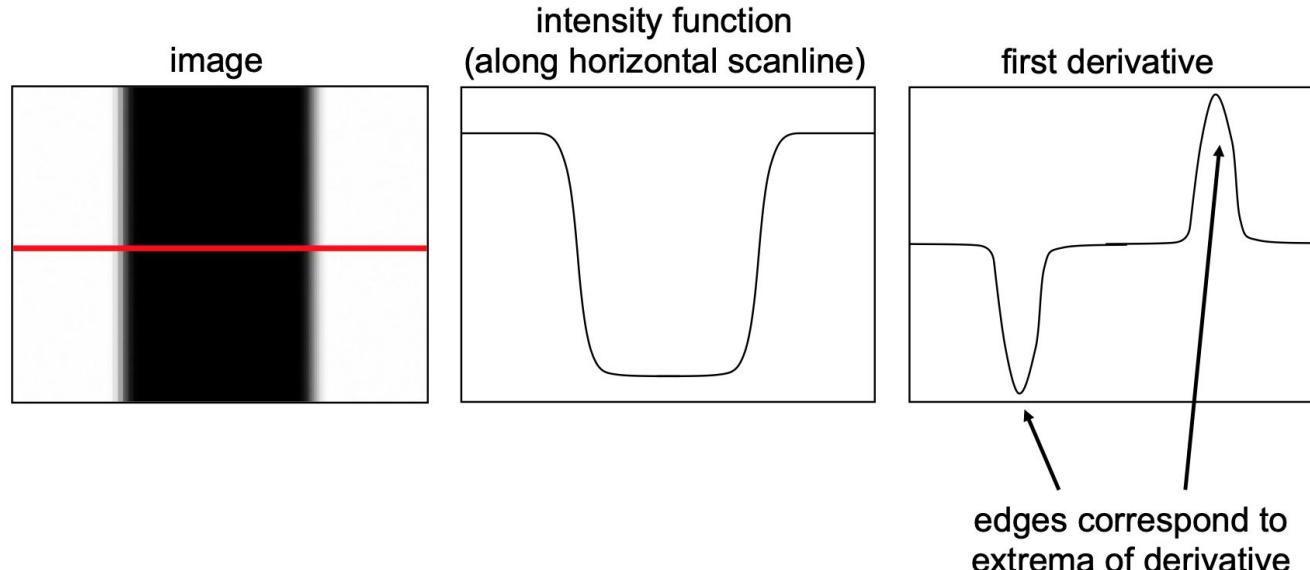
Пример

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



Разрыв функции

Край - место быстрого изменения функции яркости изображения.

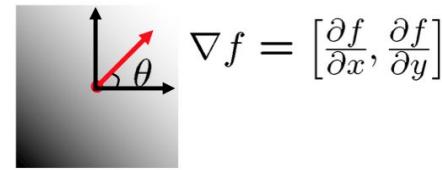
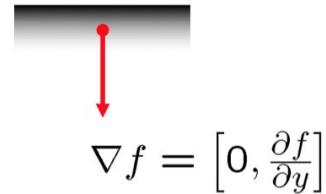
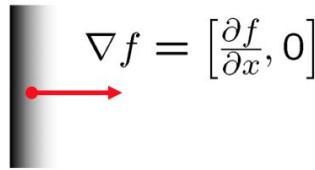


Source: cs231

Разрыв функции

Градиент изображения:

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



Градиент в каждой точке указывает направление наибольшего возрастания функции

Это направление дается формулой

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

- А как это связано с направлением края?

“Сила” границы - это магнитуда градиента

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

Source: Steve Seitz

Пример

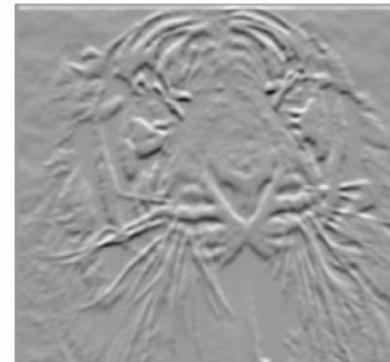
Исходное
изображение



?



Магнитуда
градиента



?

Пример

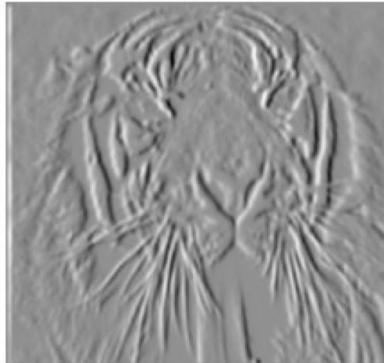
Исходное изображение



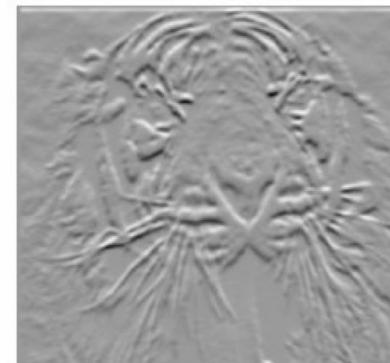
Магнитуда градиента



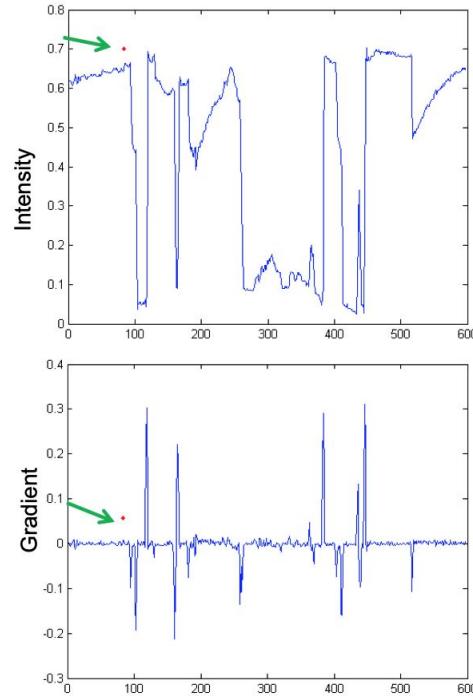
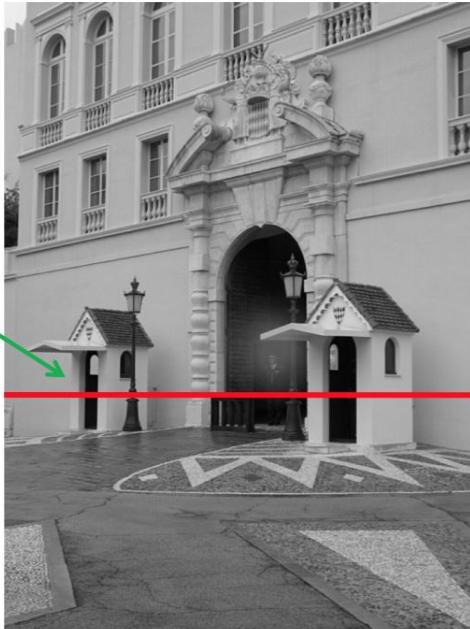
Градиент X



Градиент Y



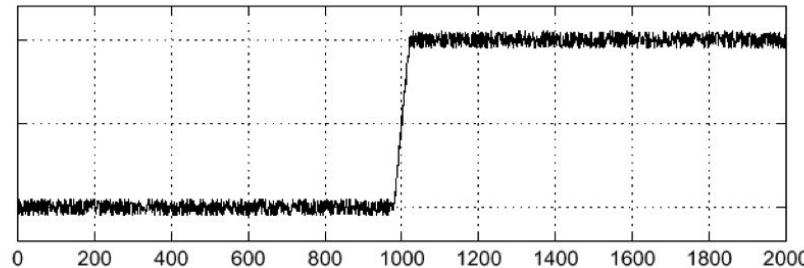
Профиль (разрез) яркости



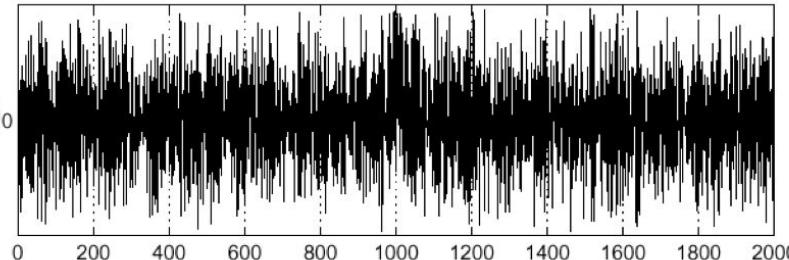
Source: D. Hoiem

Границы в зашумленных изображениях

$f(x)$



$\frac{d}{dx} f(x)_0$



Где край?

Границы в зашумленных изображениях

Фильтры для нахождения производной очень чувствительны к шуму

- Приводит к появлению “неправильных” краев
- Чем больше шум, тем больше ошибка

Что делать?

Границы в зашумленных изображениях

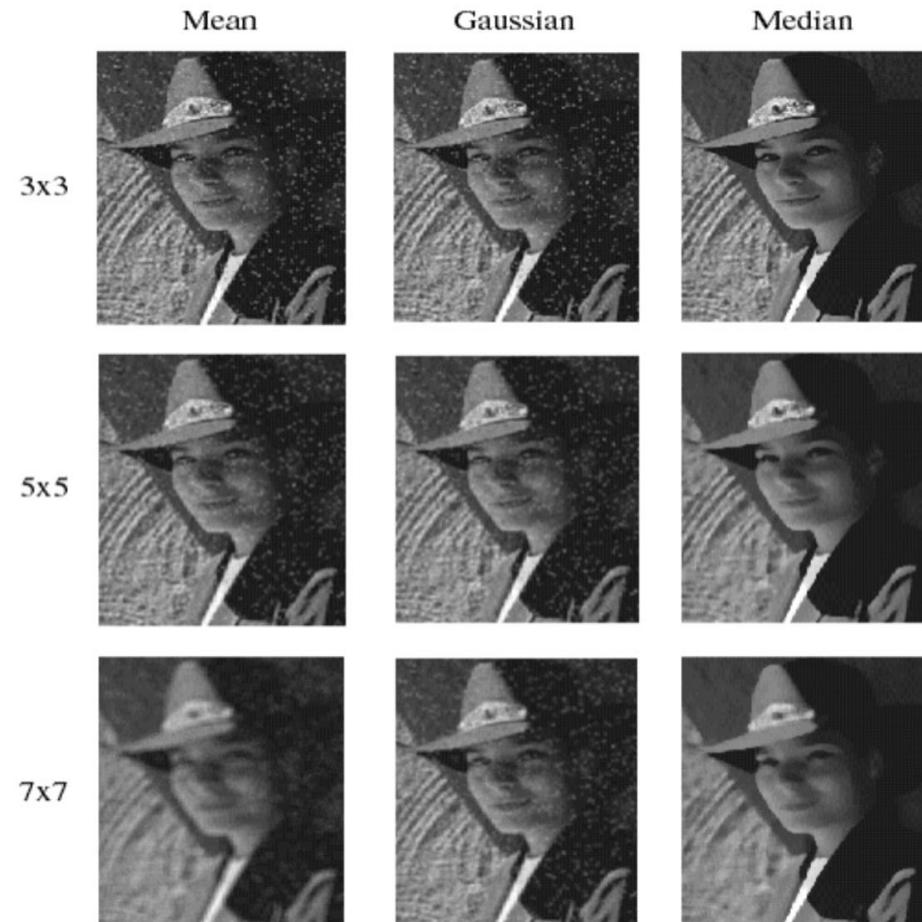
Фильтры для нахождения производной очень чувствительны к шуму

- Приводит к появлению “неправильных” краев
- Чем больше шум, тем больше ошибка

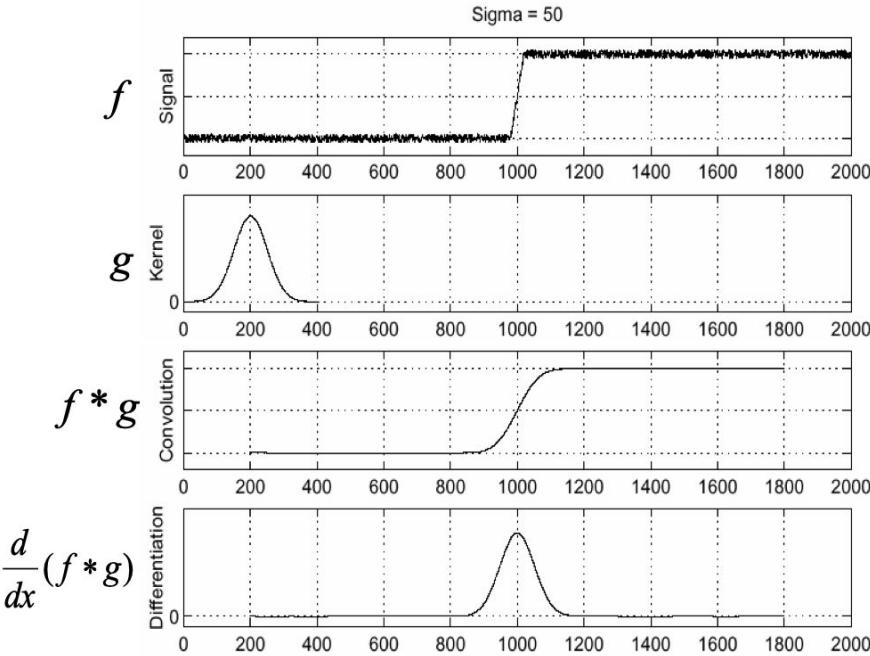
Что делать?

- Сглаживать изображение (это мы уже умеем!)

Границы в зашумленных изображениях



Границы в зашумленных изображениях



Чтобы найти края, ищем пики в

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

Границы в зашумленных изображениях

Из теории обработки сигналов известна теорема:

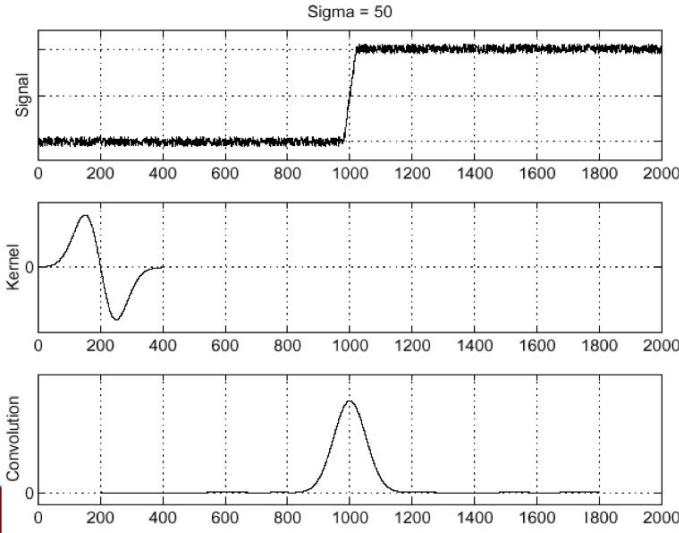
$$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$

Это позволяет делать на одну операцию меньше:

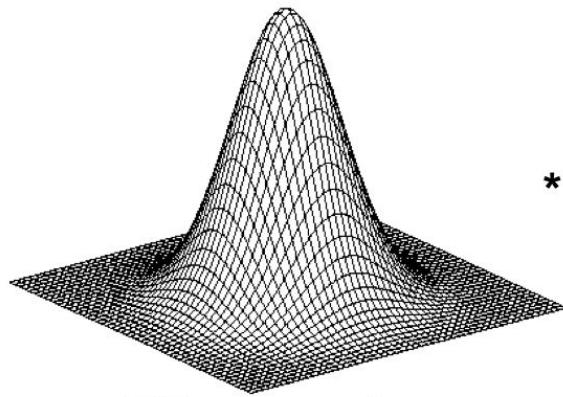
f

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

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

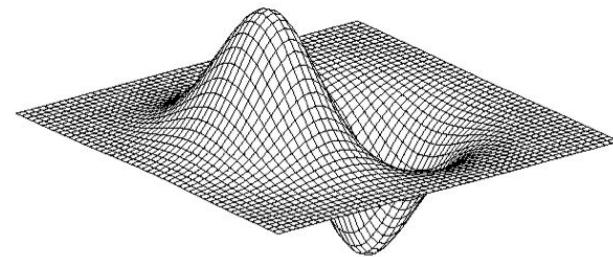


Производная фильтра Гаусса



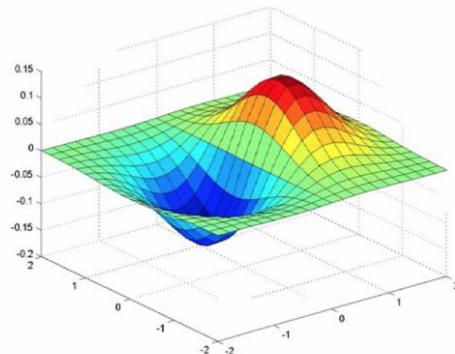
2D-gaussian

$$* [1 \ 0 \ -1] =$$

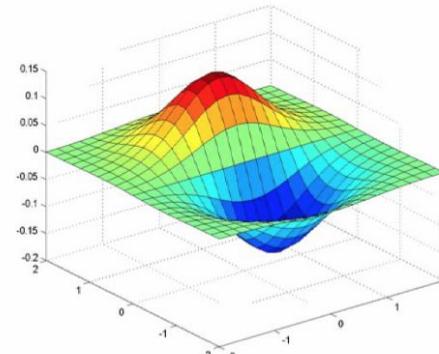


x - derivative

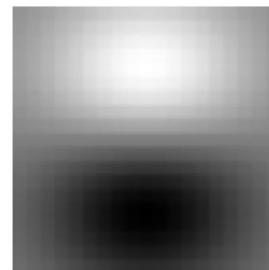
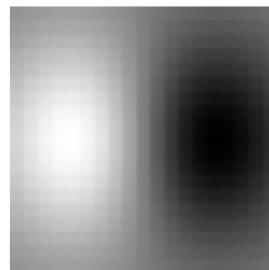
Производная фильтра Гаусса



x-direction



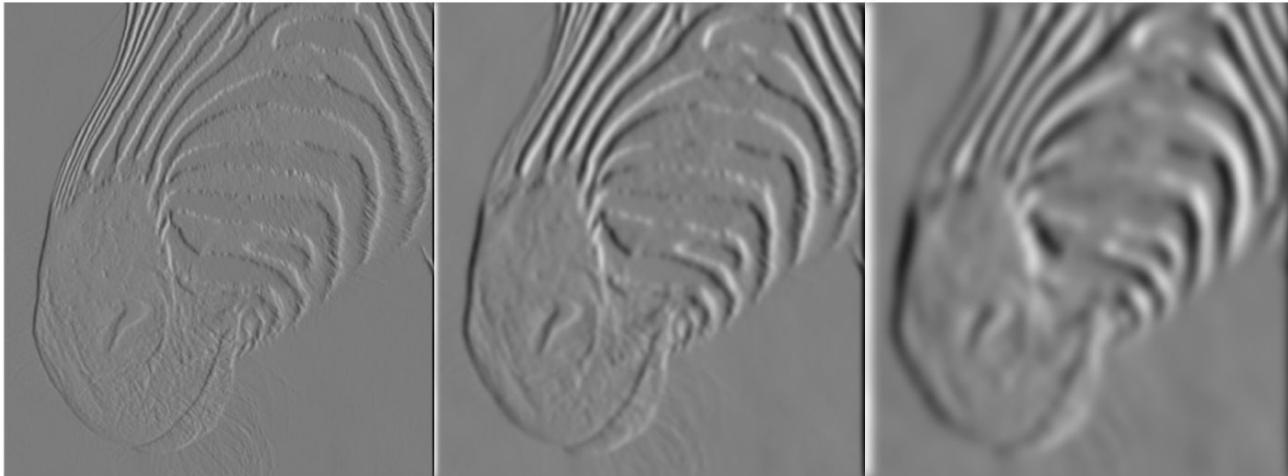
y-direction



Производная фильтра Гаусса



Сглаживание различного масштаба



1 pixel

3 pixels

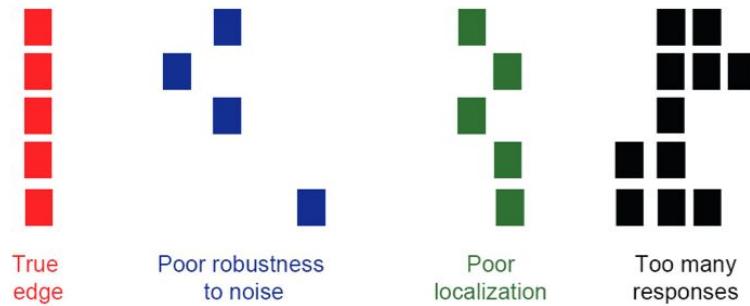
7 pixels

- Фильтрация Гаусса убирает шум, но также размывает границы.
- Позволяет находить производные на различных уровнях масштаба

Обнаружение границ

Критерий для оптимального определителя краев

- **Точность** - мало False Positive (не принимать шум за границы) и False Negative (не пропускать реальные края)
- **Локализация** - обнаруженные границы должны быть как можно ближе к реальным границам
- **Single response** - граница не должна быть широкой



Source: cs231

Оператор Собеля

Оператор Собеля

- Используются ядра размера 3x3, свертываются с исходным изображением, чтобы посчитать аппроксимацию градиента
- Один - горизонтальный, другой вертикальный

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Оператор Собеля

- Сглаживание + производная

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} [+1 \quad 0 \quad -1]$$

Сглаживание

Производная

Оператор Собеля

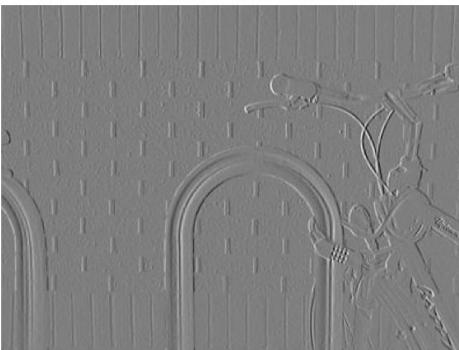
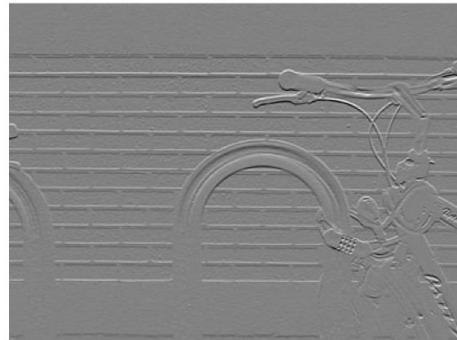
- Магнитуда:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

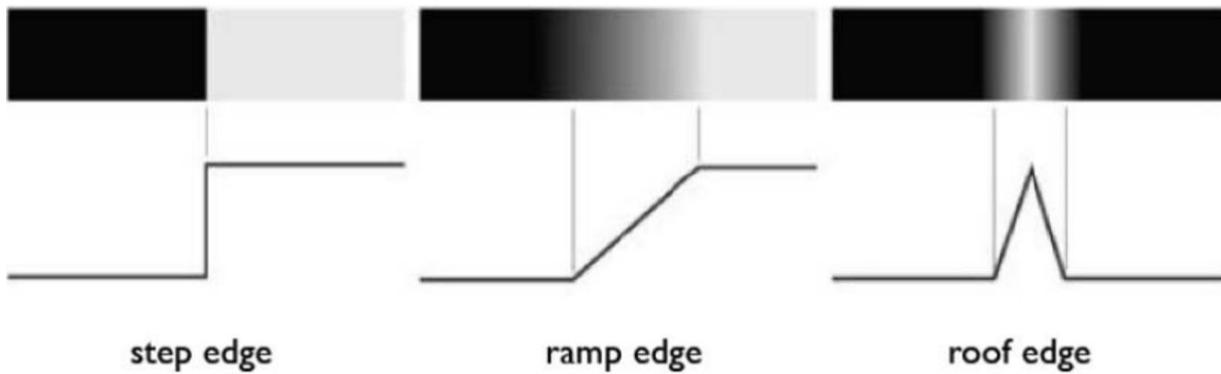
- Направление:

$$\Theta = \text{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Оператор Собеля (пример)



Оператор Собеля - минусы



- Плохая локализация (в одной области - множество пикселей принимаются за границу)
- Не все направления равнозначны из-за порогового значения
Диагональные границы чаще пропускаются чем горизонтальные или вертикальные
False Negatives

Детектор Кэнни (Canny detector)

Детектор границ Кэнни

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

Наиболее широко используемый детектор границ в компьютерном зрении



Детектор границ Кэнни

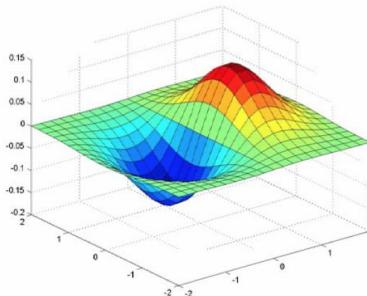
- Удаление шума
- Вычисление магнитуды и направление градиента в каждом пикселе
- Применение Non-maximum suppression
- Отбор методом гистерезиса (анализ связности)

Детектор границ Кэнни

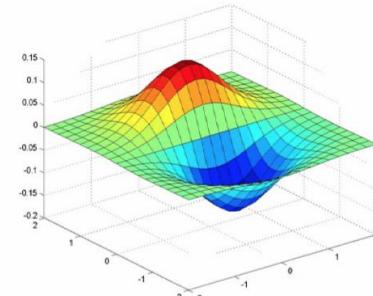


Исходное изображение

Детектор границ Кэнни

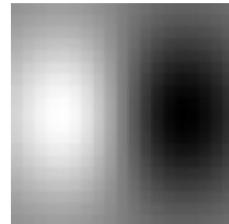


x-direction

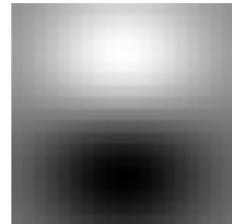


y-direction

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

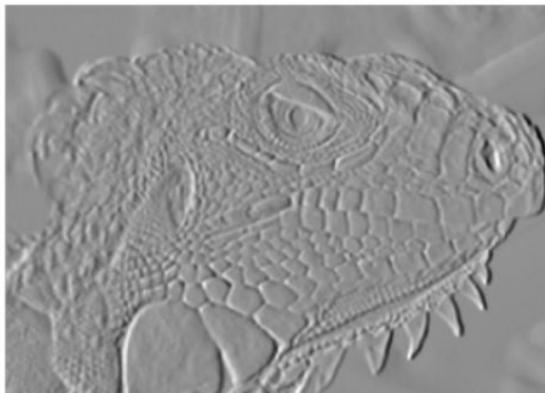


$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

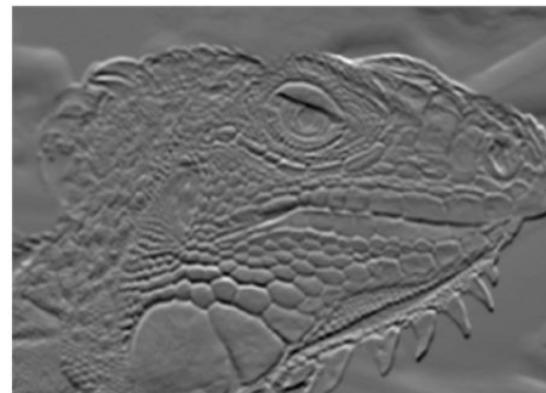


Производная фильтра Гаусса

Подсчет градиентов



X-Derivative of Gaussian



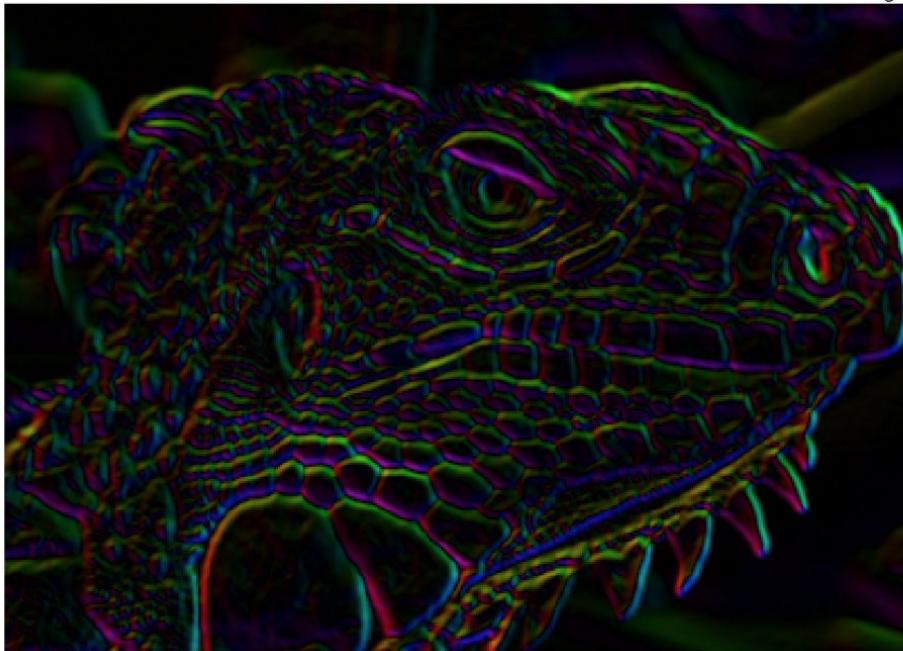
Y-Derivative of Gaussian



Gradient Magnitude

Направление градиента

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



Source: J. Hayes

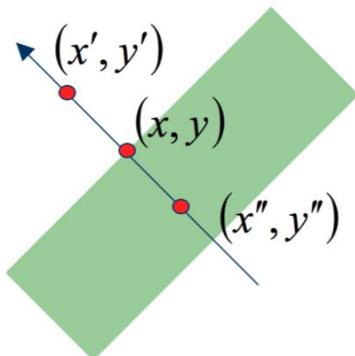
Non-maximum suppression

- Края возникает, когда градиент достигает максимума
- Не будем рассматривать точку как границу, если ее градиент не максимальен (даже если проходит пороговое значение)
- Убираем пиксели как границы во всех направлениях, если их градиенты не максимальны

Позволяет добиться Single response!

Non-maximum suppression

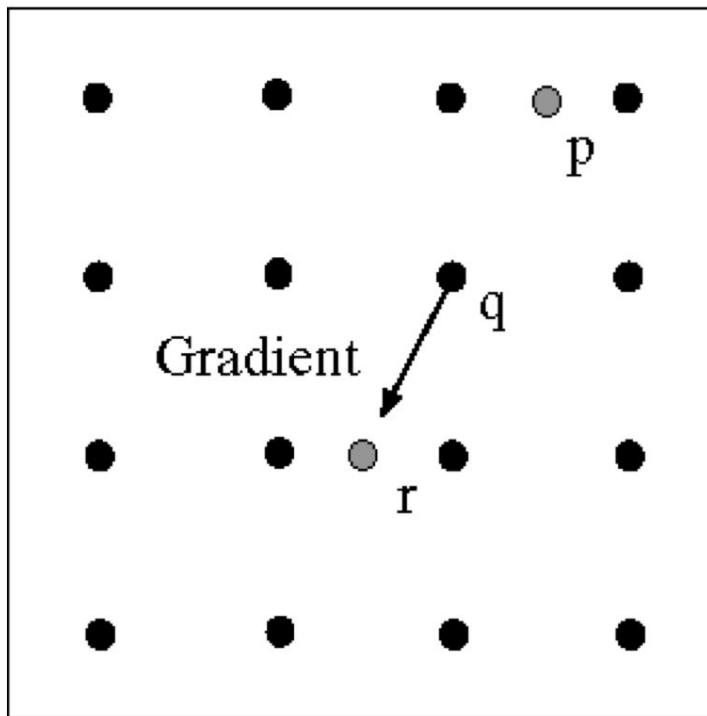
$|\nabla G|(x, y)$ это градиент в пикселе (x, y)



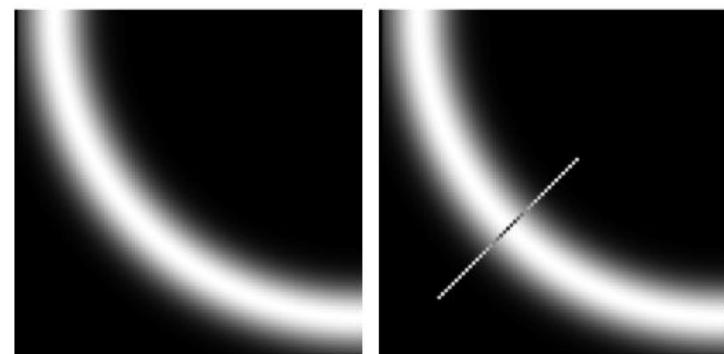
$$M(x, y) = \begin{cases} |\nabla G|(x, y) & \text{if } |\nabla G|(x, y) > |\nabla G|(x', y') \\ & \& |\nabla G|(x, y) > |\nabla G|(x'', y'') \\ 0 & \text{otherwise} \end{cases}$$

x' and x'' are the neighbors of x along
normal direction to an edge

Non-maximum suppression



Интерполируем, чтобы
получить значение градиента в
р и г.



Non-maximum suppression



Before



After

Гистерезис

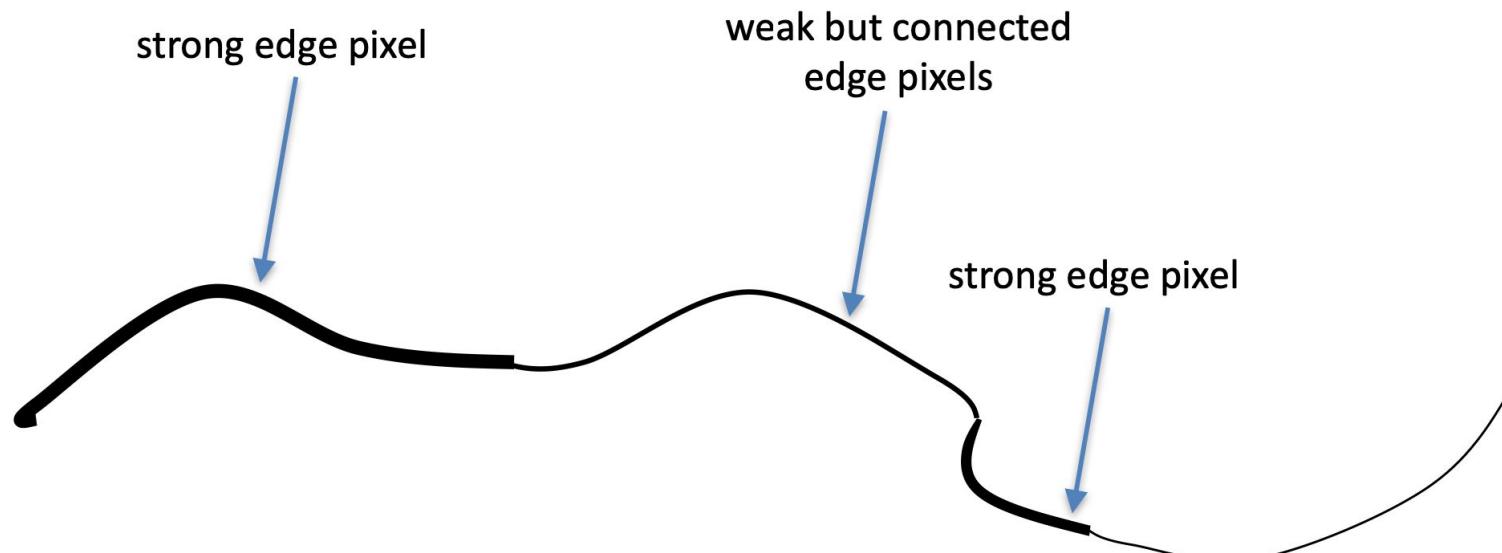
Сделаем два пороговых значения: низкий и высокий

- Если меньше низкого - точно не граница (non-edge-pixel)
- Если больше высокого - точно граница (strong-edge-pixel)
- Если между низким и высоким - слабая граница (weak-edge-pixel)

Смотрим на соседей weak-edge-pixel.

Если соединен с strong-edge-pixel напрямую, или через других weak-edge-pixel, тогда это тоже граница.

Гистерезис



Source: S. Seitz

Результат



Алгоритм

1. Фильтруем изображение x,y -производной фильтра Гаусса
2. Находим производную и магнитуду градиента
3. Non-maximum suppression
 - Границы становятся тонкими
4. Пороговое отсечение и связывание (гистерезис)
 - Два порога - низкий и высокий
 - Высокий порог позволяет начать инициализировать кривую, обозначающую границу, низкий позволяет продолжать ее

Влияние размера и ширины фильтра Гаусса



original



Canny with $\sigma = 1$

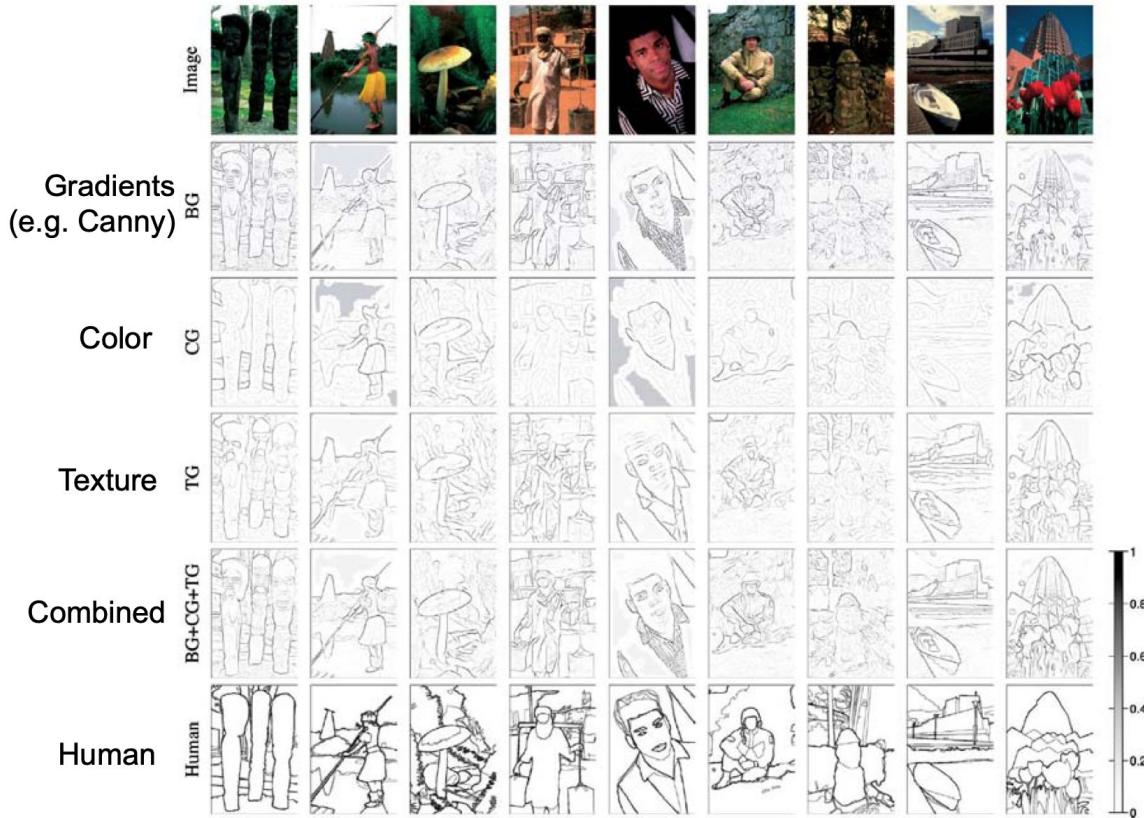


Canny with $\sigma = 2$

Выбор сигмы зависит от того, что хочется получить:

1. Большая сигма позволяет находить границы больших объектов
2. Маленькая сигма позволяет находить небольшие границы

Различные методы детекции границ



Извлечение признаков из изображений

Матчинг изображений



Матчинг изображений



by [Diva Sian](#)



by [swashford](#)

Источник: Steve Seitz

Матчинг изображений



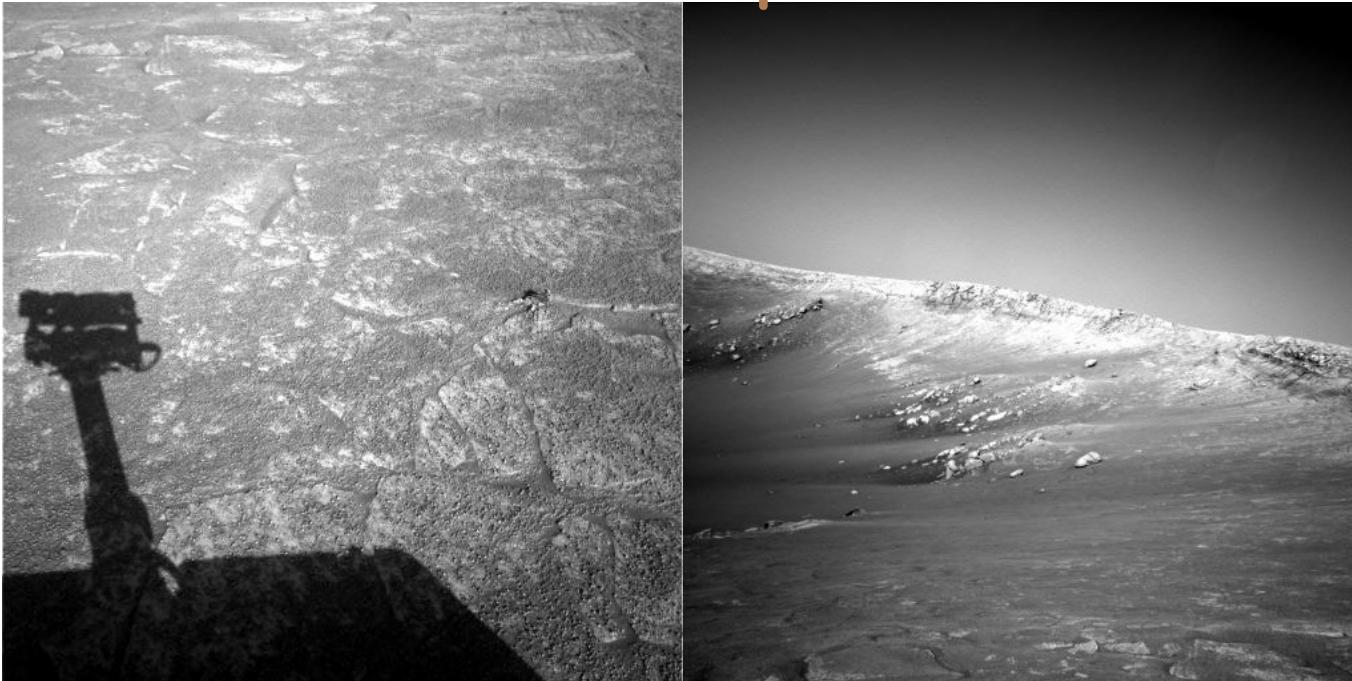
by [Diva Sian](#)



by [scgbt](#)

Источник: Steve Seitz

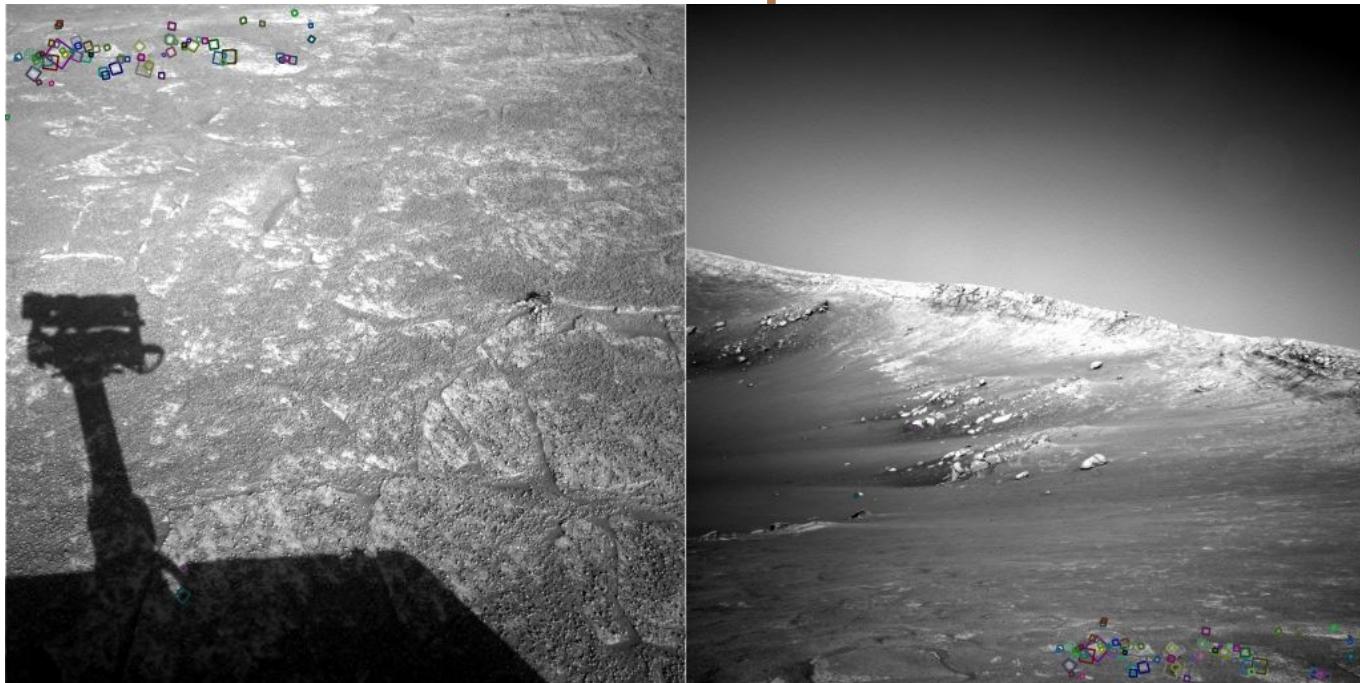
Матчинг изображений



NASA Mars Rover images

Источник: Steve Seitz

Матчинг изображений



Ответ!

Источник: Steve Seitz

Зачем использовать локальные признаки?

Глобальные признаки слишком ненадежны для использования

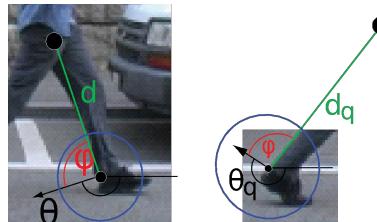
Вместо этого, будем использовать небольшие области в изображении

Устойчиво к:

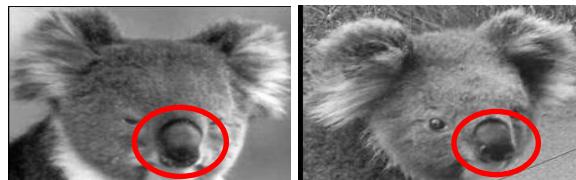
- перекрытию



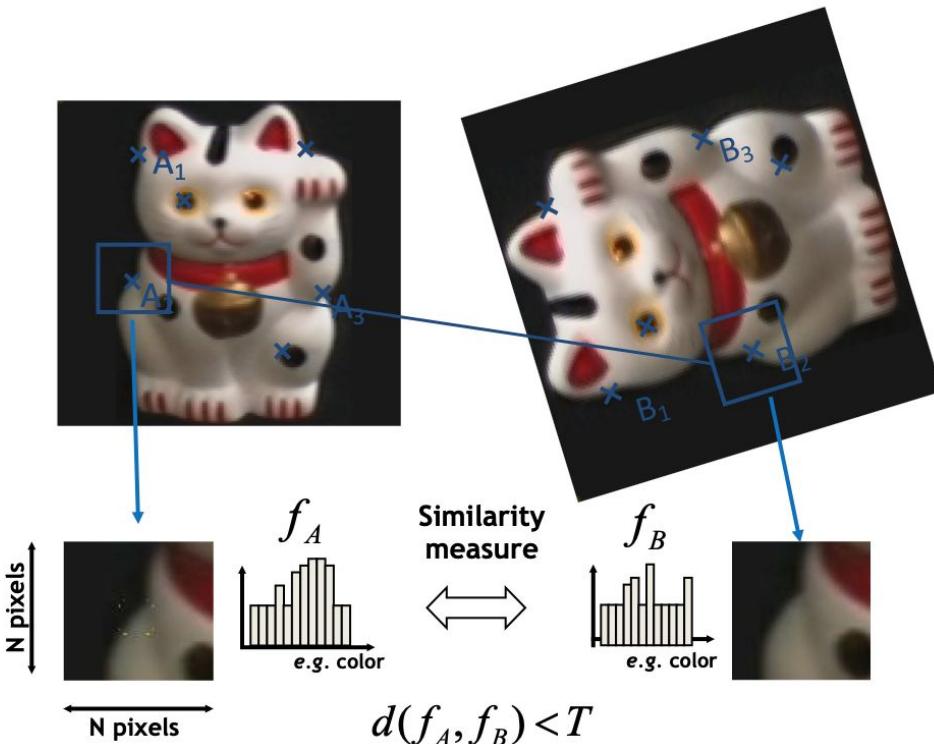
- поворотам



- внутриклассовой изменчивости



Общий подход



1. Найти набор различных особых точек на изображениях.
2. Выделить область вокруг каждой из этих точек
3. Нормализовать каждую из этих областей
4. Посчитать локальный дескриптор из нормализованной области
5. Смэтчить локальные дескрипторы

Требования

Проблема 1:

Считать точки независимо на каждом изображении



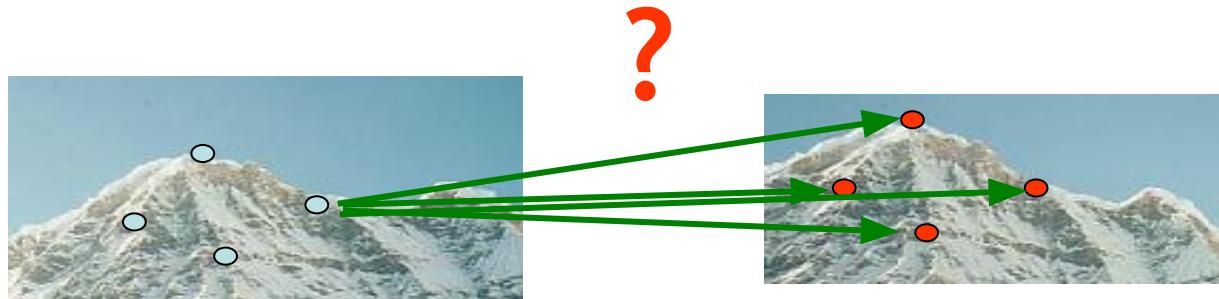
Не матчается!

Нужен повторяемый (repeatable) детектор!

Требования

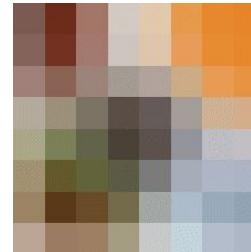
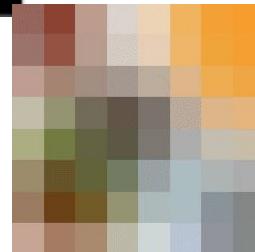
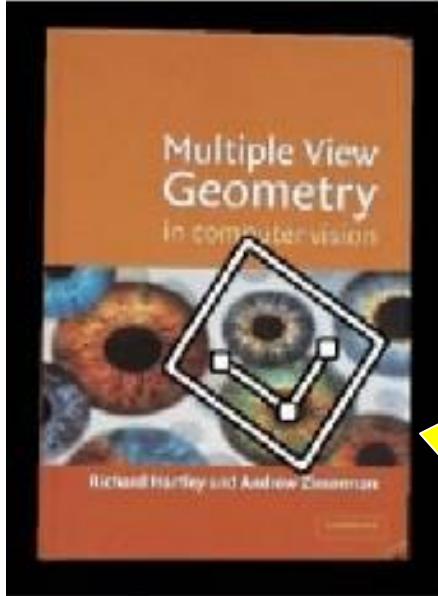
Проблема 2:

Для каждой точки корректно найти соответствующую ей на другом изображении

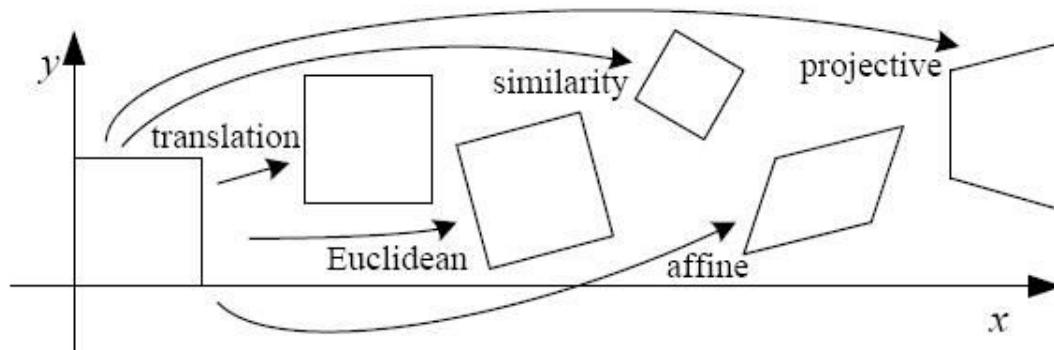


Нужен устойчивый к преобразованиям дескриптор!

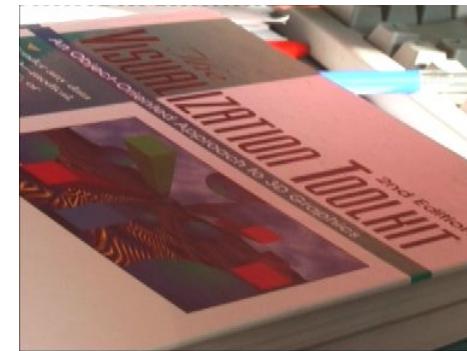
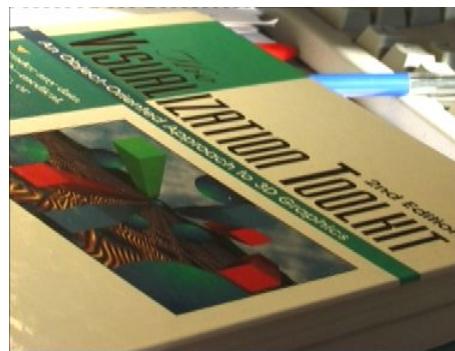
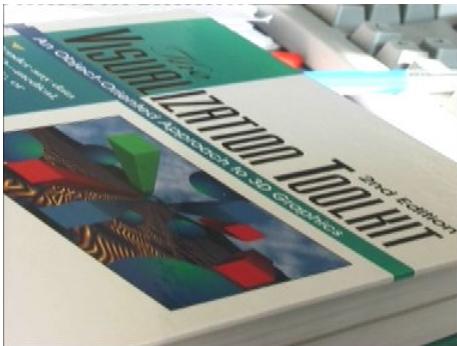
Инвариантность: геометрические преобразования



Геометрические преобразования



Инвариантность: фотометрические преобразования



Тоже линейное преобразование:
Умножение + смещение

Требования

Выделение регионов должно быть точными и повторяемым:

- **Инвариантно** к сдвигам, поворотам, масштабированию
- **Устойчиво** к аффинным преобразованиям
- **Устойчиво** к изменению освещению, шуму, размытию
- **Локальность**: признаки локальны, а поэтому устойчивы к перекрытию
- **Количество**: нужно достаточно большое количество областей для описания объекта
- **Уникальность**: локальные области должны быть “необычны” для изображения
- **Эффективность**: должно работать быстро (real-time performance)

Различные детекторы

Hessian & Harris	[Beaudet '78], [Harris '88]
Laplacian, DoG	[Lindeberg '98], [Lowe '99]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid '01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid '04]
EBR and IBR	[Tuytelaars & Van Gool '04]
MSER	[Matas '02]
Salient Regions	[Kadir & Brady '01]
другие...	

Эти детекторы основа большинства алгоритмов компьютерного зрения

Нахождение ключевых точек

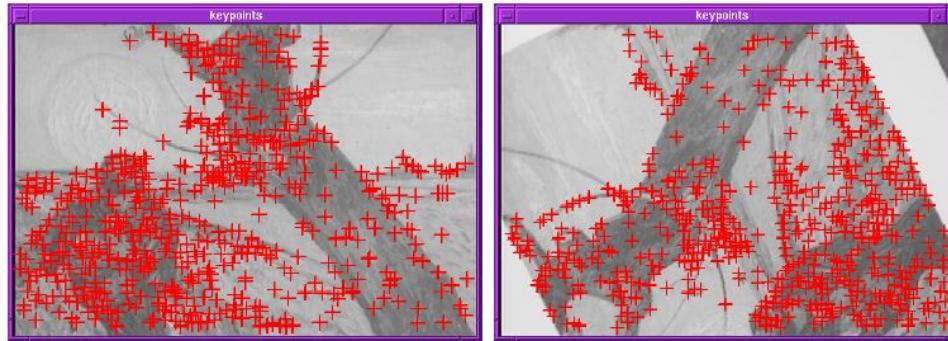
Что хотим:

- Устойчивость к трансформациям
- Точная локализация
- Интересное содержание в изображении

Помним, что изображение - функция!



Нахождение углов



Ключевое свойство:

В области вокруг угла, градиент изображения имеет два или более доминантных направления

Углы - уникальны!

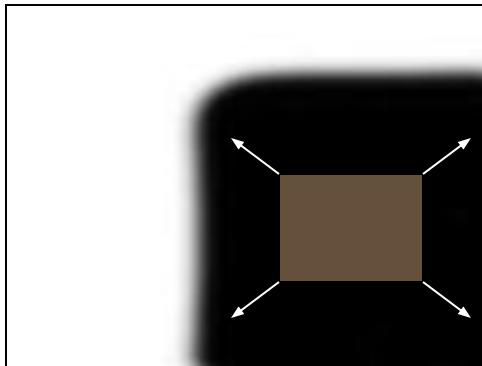
C.Harris and M.Stephens. ["A Combined Corner and Edge Detector."](#) *Proceedings of the 4th Alvey Vision Conference*, 1988.

Углы как ключевые точки

Критерий:

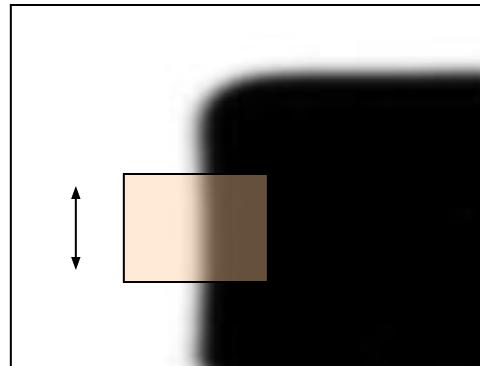
Хотим чтобы было легко распознать точку, по небольшому окну вокруг нее.

Смещение окна в любом направлении дает большое изменение яркости



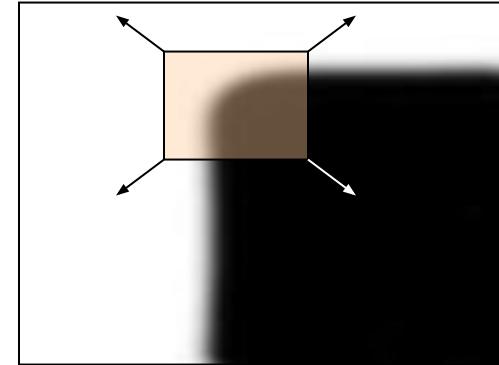
“flat”:

Не изменяется ни в
каком направлении



“edge”:

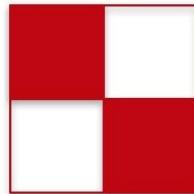
Не изменяется в
направлении
границы



“corner”:

Изменение в
любом
направлении

Углы, края, плато



$$\sum I_x^2 \longrightarrow \text{Large}$$

$$\sum I_y^2 \longrightarrow \text{Large}$$

Corner



$$\sum I_x^2 \longrightarrow \text{Small}$$

$$\sum I_y^2 \longrightarrow \text{Large}$$

Edge

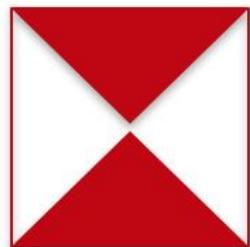


$$\sum I_x^2 \longrightarrow \text{Small}$$

$$\sum I_y^2 \longrightarrow \text{Small}$$

Nothing

Углы, края, плато



$$\sum I_x^2 \longrightarrow ??$$

$$\sum I_y^2 \longrightarrow ??$$

Corner

Детектор Харриса

Детектор Харриса

Изменение интенсивности при сдвиги в направлении $[u,v]$:

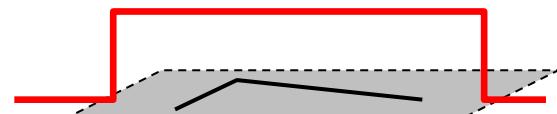
$$E(u,v) = \sum_{x,y} w(x,y) [I(x+u, y+v) - I(x, y)]^2$$

Функция веса

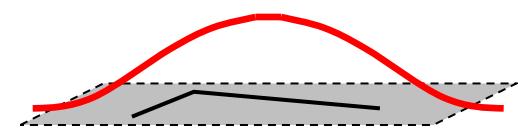
Смещенная
интенсивность

Интенсивность

Функция веса $w(x,y) =$



1 in window, 0 outside



Gaussian

Детектор Харриса

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

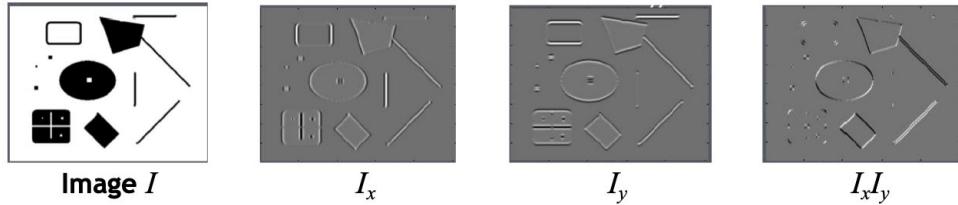
$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Производная по x,
умноженная на
производную по y

Сумма по всем пикселям
рассматриваемой области

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

Детектор Харриса



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

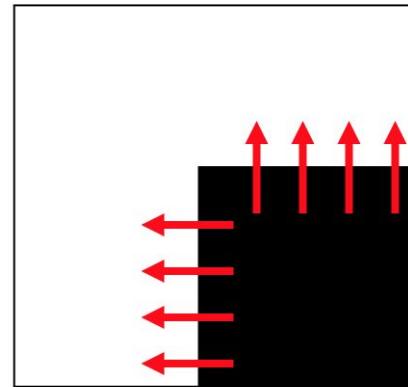
Производная по x,
умноженная на
производную по y

Сумма по всем пикселям
рассматриваемой области

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

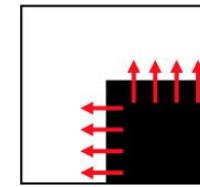
Зачем нужна эта матрица?

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



Зачем нужна эта матрица?

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



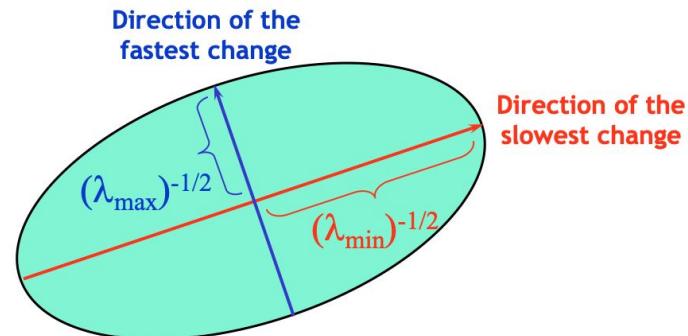
- Доминирующие направления градиента направления вдоль осей x или y
- Если хоть одна из lamda близка к нулю, значит это не угол
- Что делать, если наш угол не сонаправлен осям изображения?

Общий случай

Поскольку M - симметрична, мы можем факторизовать ее:
(eigenvalue decomposition)

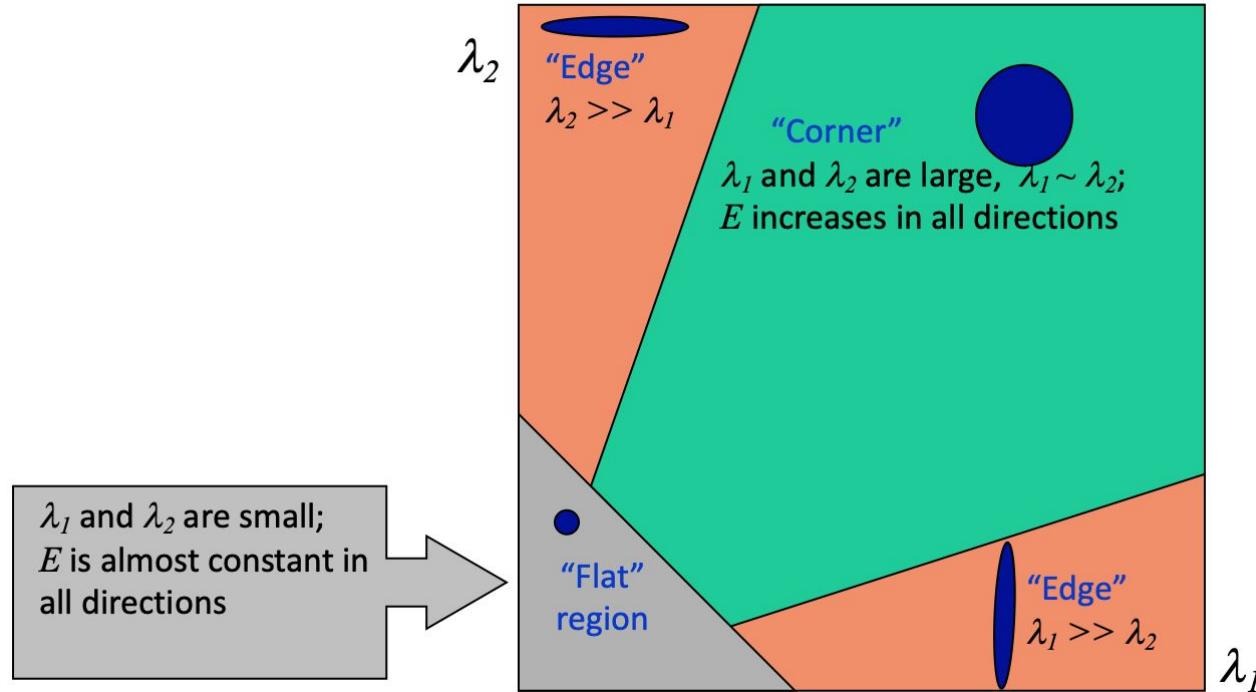
$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Мы можем визуализировать матрицу M как эллипс, с длинами осей равными собственным числам и направлением поворота R



Собственные значения

Классификация точек изображения с помощью матрицы M



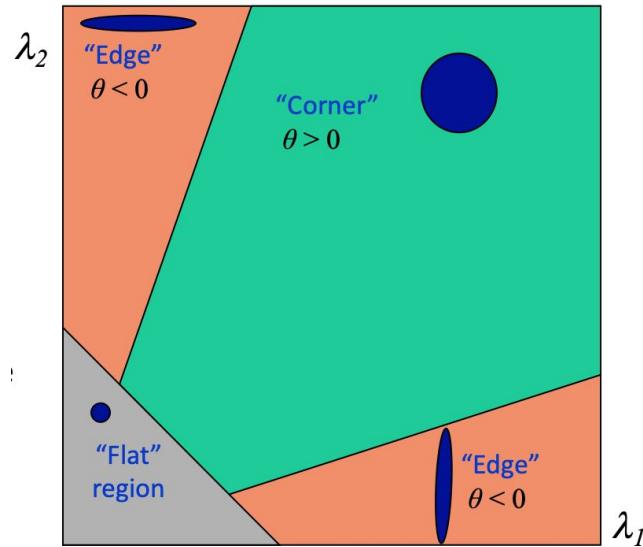
Функция ответа угла (Corner response function)

Классификация точек изображения с помощью матрицы M

$$\theta = \det(M) - \alpha \operatorname{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

Эффективная аппроксимация!

- Не нужно считать собственные значения
- Alpha - константа (0.04-0.06)

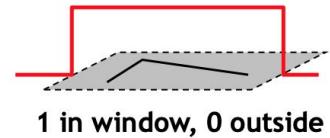


ФУНКЦИЯ ОКНА

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

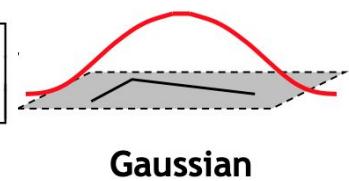
1. Box window
 - сумма по всему окну
 - проблема: неинвариантна к поворотам

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



2. Gaussian window
 - взвешенная сумма в зависимости от расстояния до центра
 - результат: инвариантность к поворотам

$$M = g(\sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Детектор Харриса - summary

$$M(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Вычисляем производные по x и y

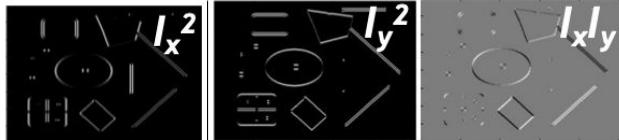
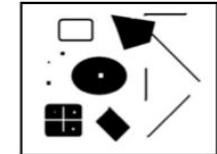
2. Квадраты производных

3. Фильтруем Гауссом

4. Corner response function

$$\begin{aligned}\theta &= \det[M(\sigma_I, \sigma_D)] - \alpha[\text{trace}(M(\sigma_I, \sigma_D))]^2 \\ &= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2\end{aligned}$$

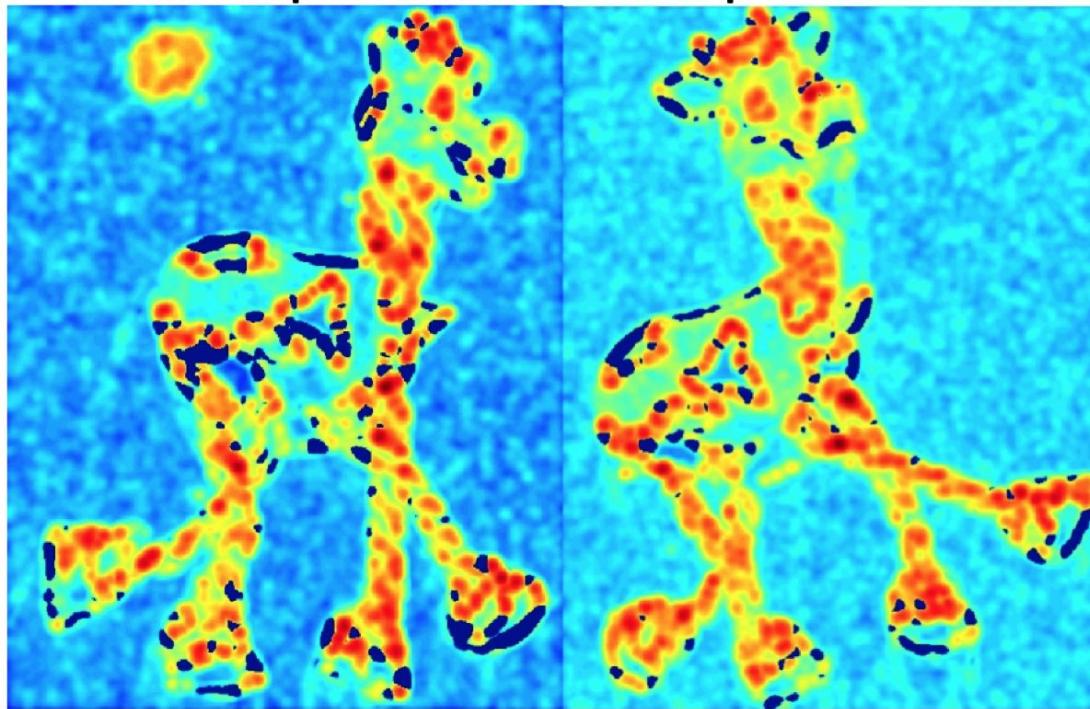
5. Non maximum suppression



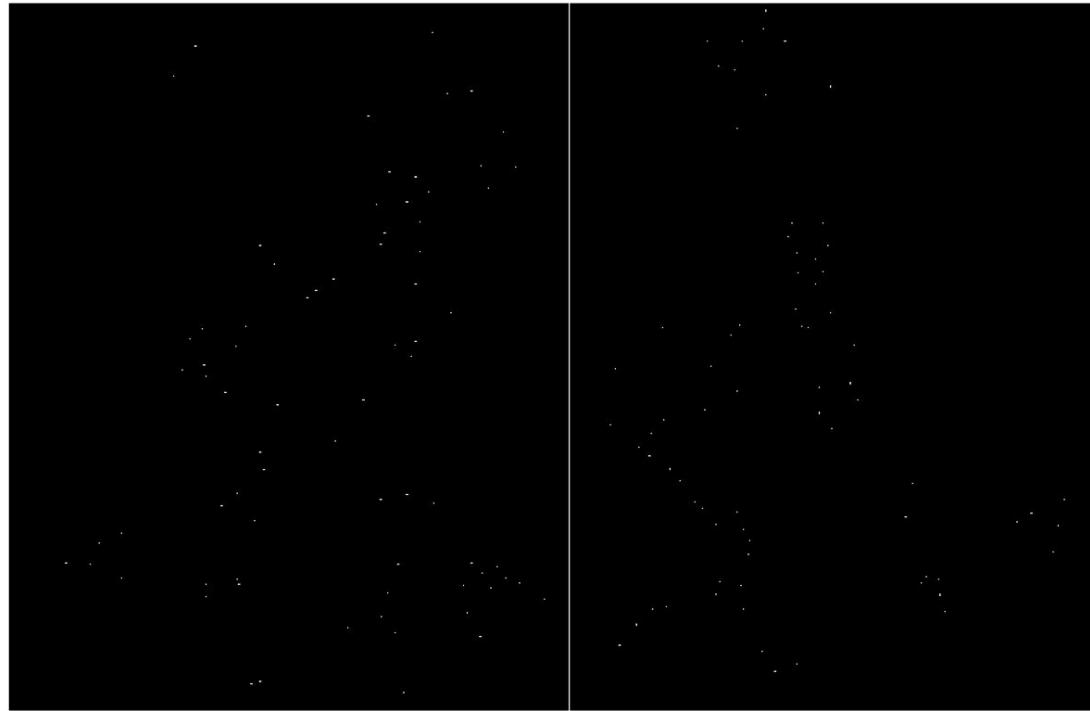
Детектор Харриса - пример



Детектор Харриса - пример



Детектор Харриса - пример



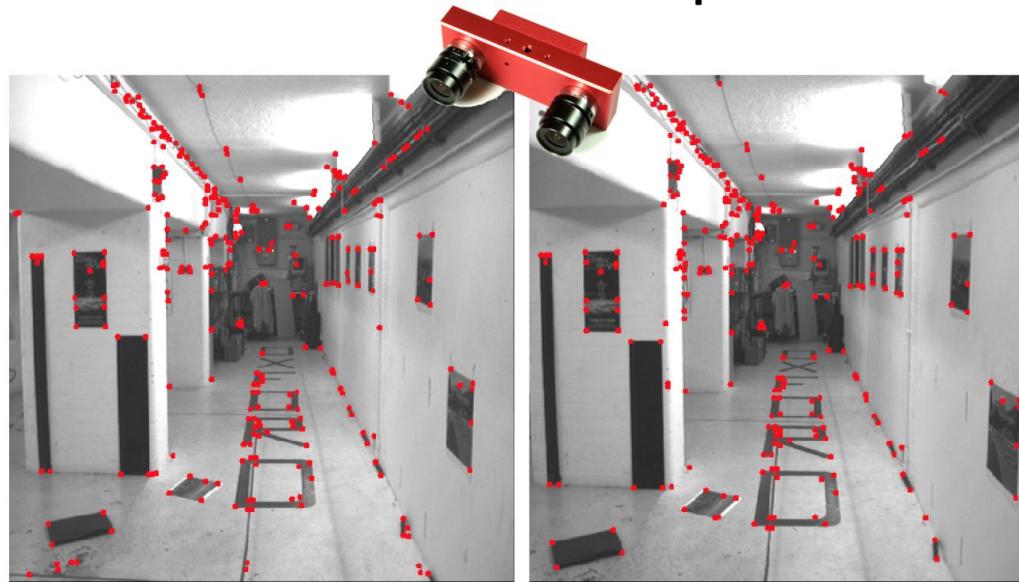
Берем локальные максимумы theta, где $\theta > \text{threshold}$

Детектор Харриса - пример



Результат!

Детектор Харриса - пример



Хорошо применимы для стереопреобразований.

Детектор Харриса - свойства

Инвариантность к сдвигу?

Детектор Харриса - свойства

Инвариантность к сдвигу? **ДА!**

Детектор Харриса - свойства

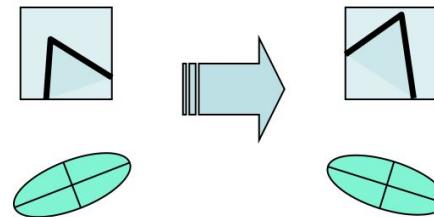
Инвариантность к сдвигу? **ДА!**

Инвариантность к повороту?

Детектор Харриса - свойства

Инвариантность к сдвигу? **ДА!**

Инвариантность к повороту?



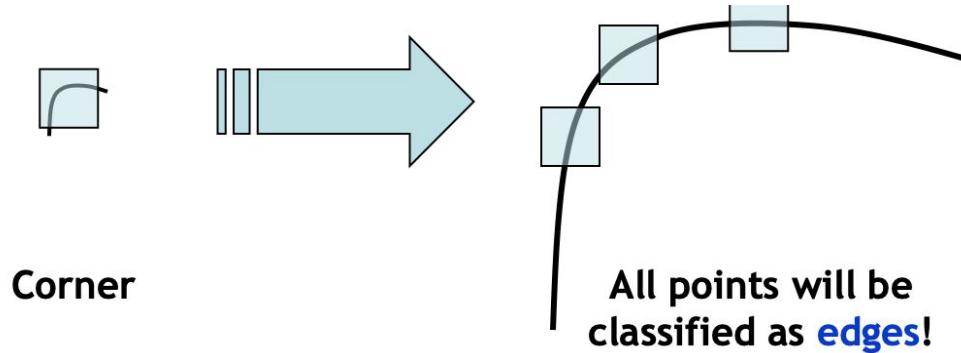
Эллипс поворачивается, но
его форма (т.е. собственные
значения) не меняется

Детектор Харриса - свойства

Инвариантность к сдвигу? **ДА!**

Инвариантность к повороту? **ДА!**

Инвариантность к масштабированию?

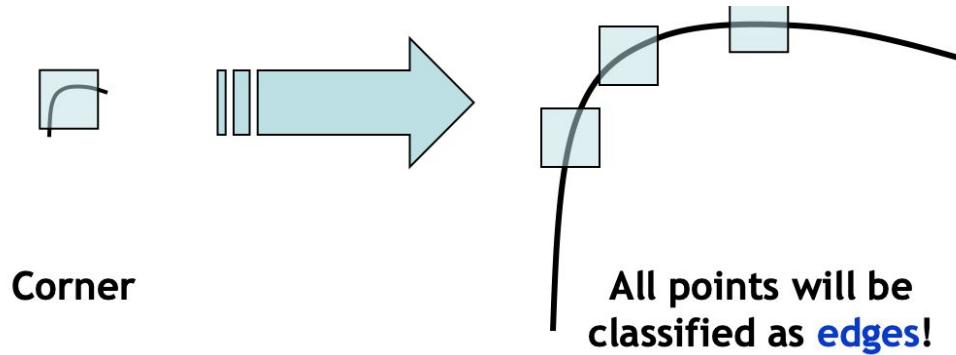


Детектор Харриса - свойства

Инвариантность к сдвигу? **ДА!**

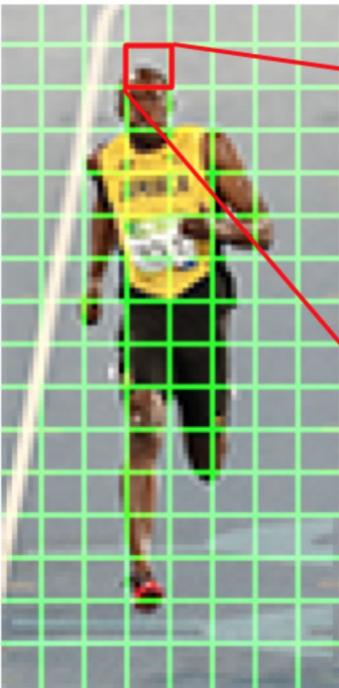
Инвариантность к повороту? **ДА!**

Инвариантность к масштабированию? **НЕТ :(**



HOG

HOG



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

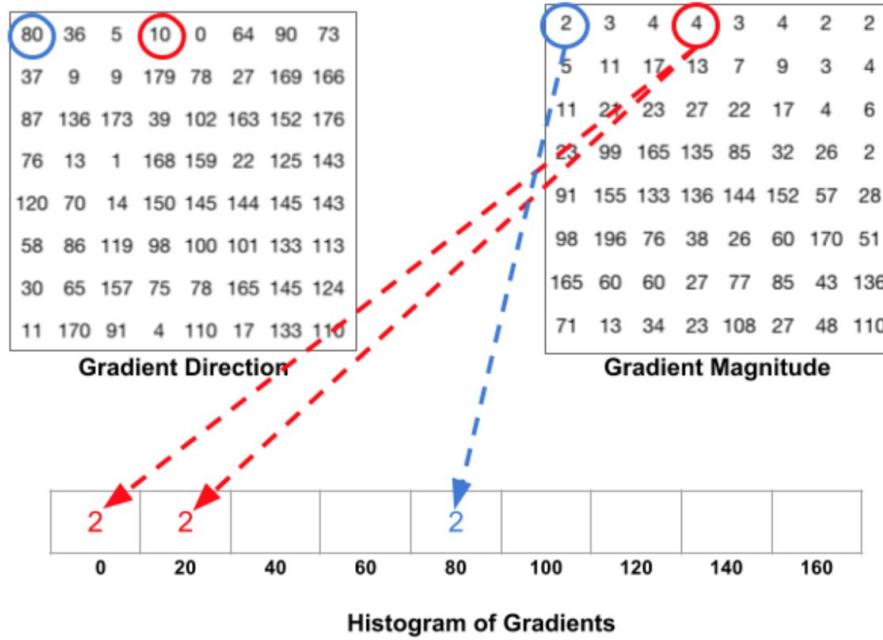
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Общий алгоритм

- Находим градиент
- Переводим в полярные координаты
- Оцениваем гистограмму:
 - Разбиение на бины по углу
 - Норма градиента дает вклад в столбец

Построение гистограммы



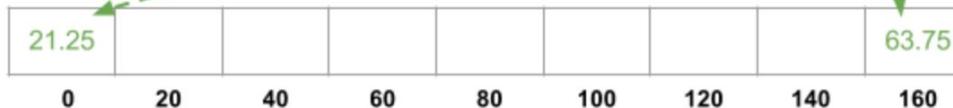
Построение гистограммы

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude



Histogram of Gradients

Let's code