## Выделение границ

Светлана Носова nsa.lector@gmail.com

#### Способы выделения границ

- Восстановить аналоговую функцию (изображение), найти градиент.
- Приближенно вычислить производную при помощи численных методов.

Производные. Численные производные 
$$f(x) - f(x - \Delta x) = f'(x) = f$$

Производные. Численные производные 
$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

изводные. Численные производные 
$$f(x) - f(x - \Delta x)$$

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

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

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

## Дискретные производные $\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$ f(x) - f(x-1) = f'(x)f(x) - f(x+1) = f'(x)f(x+1)-f(x-1)=f'(x) [1 0 -1] f(x) = 10 15 10 10 25 20 20 20f'(x) = 0 5 - 5 0 15 - 5 0 0

#### **2D** производные

Функция

Вектор градиента

Норма градиента

Направление градиента

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

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

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

#### Примеры фильтрации

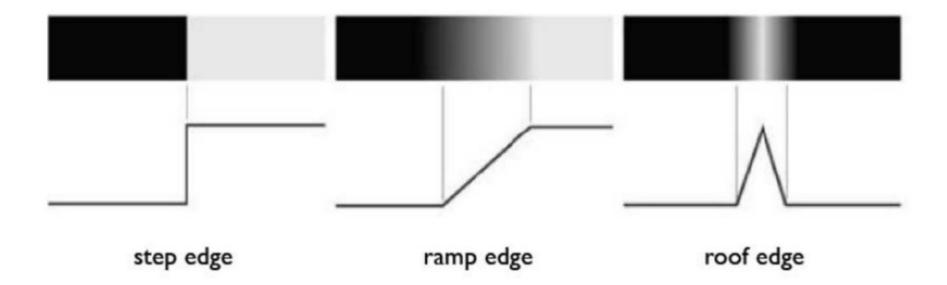
$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

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

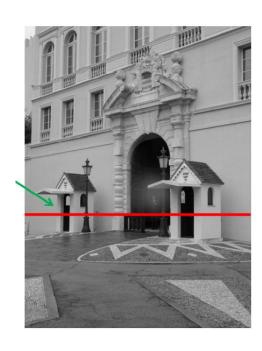
$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

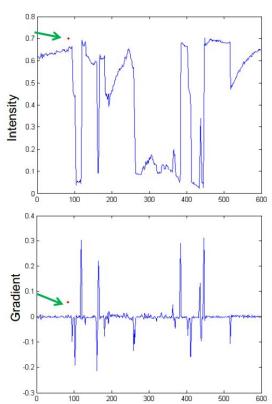
$$\frac{1}{3} \begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}$$

#### Что такое границы?

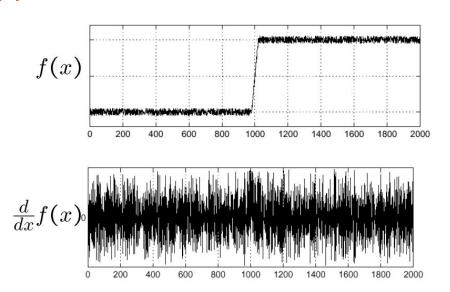


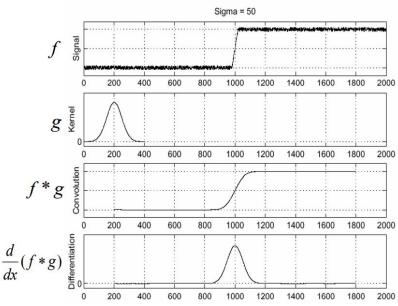
### Профиль интенсивностей (!!!)



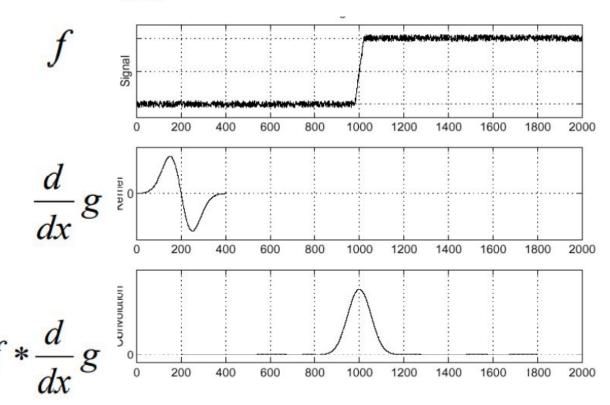


Влияние шума на результат вычисления вектора градиента. Фильтрация как первый шаг обработки данных.

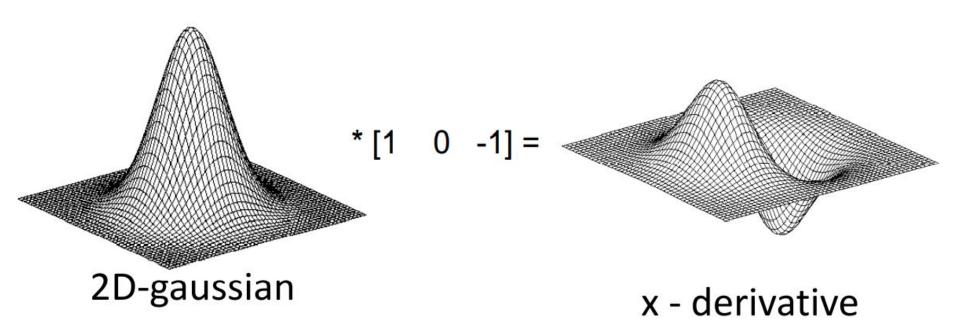




$$\frac{d}{dx}(f*g) = f*\frac{d}{dx}g$$



### Производная гауссиана (Derivative of Gaussian)



#### Оператор Собеля

- слабая локализация;
- границы толщиной в несколько пикселей.

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

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

$$oldsymbol{\Theta} = atanigg(rac{\mathbf{G}_y}{\mathbf{G}_x}igg)$$

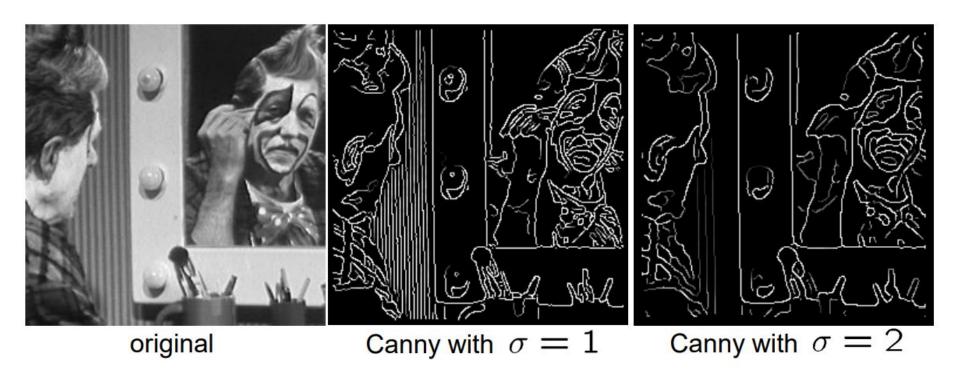
 $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$ 

$$\mathbf{G}_x = egin{bmatrix} +1 & 0 & -1 \ +2 & 0 & -2 \ +1 & 0 & -1 \end{bmatrix} = egin{bmatrix} 1 \ 2 \ 1 \end{bmatrix} [+1 & 0 & -1]$$
 Gaussian smoothing differentiation

#### Детектор границ Кэнни

- удалить шум;
- вычислить норму и направление градиента;
  - квантование величин углов: 0, 45, 90, 135.
- подавить не-максимумы (позволяет получить границу толщиной 1 пиксель):
  - только локальные максимумы вдоль направления градиента отмечаются как возможные граничные точки.
- трассировка границ:
  - Выбрать порог Т1 (для сильных) и Т2(для слабых границ);
  - Пиксели из Т1 отметить как граничные;
  - Пиксели из Т2 отмечены как граничные в том случае, если они связаны с граничными пикселями.

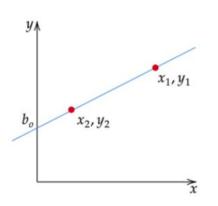
#### Детектор границ Кэнни

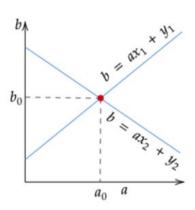


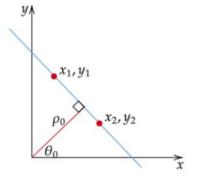
#### Преобразование Хафа

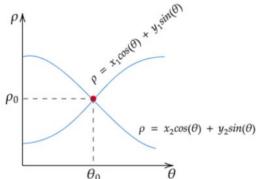
- детектирование примитивов (линий, окружностей и тп) на изображении, заданных в параметрическом виде
- цель: найти локализацию примитивов с заданными параметрами на изображении

#### От пространства точек к пространству параметров





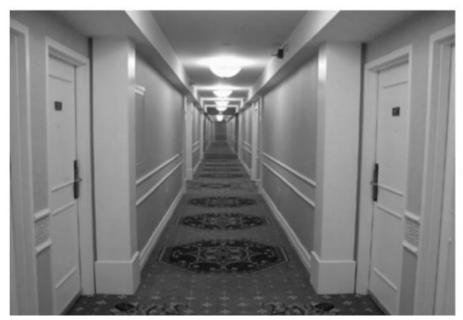


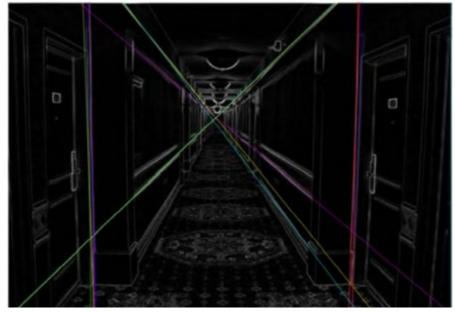


#### Пространство Хафа

- 1. Определить разброс  $\rho$  and  $\theta$ .
  - а. диапазон  $\theta$  [ 0, 180 ] градусов;
  - b. диапазон  $\rho$  [ -d, d ] где d длина диагонали изображения.
  - с. квантовать значения каждого диапазона.
- 2. Создать 2D аккумуляторный массив (пространство Хафа)с размерностью (num\_rhos, num\_thetas) и инициализировать его нулями.
- 3. Выполнить детектирование границ на изображении.
- 4. Для каждого граничного пикселя:
  - а. для всех возможных значений  $\theta$  в данной точке:
    - і. вычислить соответствующее значение ρ;
    - ii. найти точку(  $\theta$ ,  $\rho$ ) в аккумуляторном массиве(массив голосования);
    - ііі. увеличить значение, записанное в аккумуляторном массиве ( $\theta$ ,  $\rho$ ) (этап голосования).
- 5. Найти максимумы в аккумуляторном массиве. Это и будут параметры найденного примитива.

# 20 линий, за которое проголосовало наибольшее количество граничных точек





#### Метод Хафа для детектирования окружностей

- три параметра:
  - (х,у) координаты центра окружности
  - o R радиус окружности
- Пространство (аккумуляторный массив)
   "голосования" - трехмерное
- Каждый пиксель границы "голосует" за множество окружностей, которые теоретически могут быть проведены через него на изображении
  - квантованный диапазон изменения х,у,R задан как параметр поиска



- "голосование" увеличение соответствующей ячейки (Xcur,Ycur, Rcur) аккумуляторного массива на 1.
- по завершении этапа голосования происходит поиск максимумов в полученном аккумуляторном массиве.
  - дополнительно: отфильтровать локальные максимумы по порогу
- множество точек
   (X\_loc\_max, Y\_loc\_max, R\_loc\_max)
   определяют найденные на изображении окружности

$$x = a + R\cos(\theta)$$

$$y = b + R\sin(\theta)$$

$$r^2 = (x - x_0)^2 + (y - y_0)^2$$

### Трассировка границ. Метод "Паука"

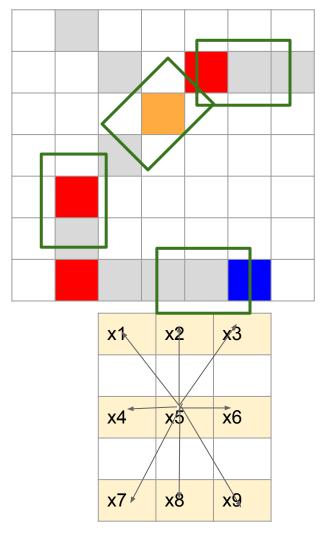
Идея метода: "паук" продолжает движение вдоль границы, если при продвижение вдоль границы (х2,х5,х8) остаются граничными, а пиксели (х1,х4,х7), (х3,х6,х9) - фоновыми. При этом при геометрических преобразованиях (смещение влево/вправо, поворот) данное правило нарушается.

#### Старт трассировки:

- конечный пиксель границы.
- произвольный пиксель границы.

#### Возможные варианты:

- пиксели (x1,x2,x3) стали фоновыми конец трассировки.
- (x2,x5,x8) граничные, но в наборе пикселей (x1,x4,x7) или (x3,x6,x9) появились граничные разветвление. Сохранить точку разветвления для последующего старта трассировки.
- пиксели (x2/x5/x8) стали фоновыми, но граничными оказались пиксели (x1/x4/x7) или (x3/x6/x9) поворот. Выполнить поворот паука вдоль границы.



#### Кодирование границ

- 1. Последовательно хранить тройку (x,y,C\_type), в которых:
  - а. (x,y) координаты пикселей, в которых происходит изменение поведения границы.
  - b. C\_type тип изменения поведения границы(поворот на угол+величина угла поворота, окончание границы).
- 2. Каждый пиксель границы кодируется соответствующим числом:
  - а. 0 отсутствие изменения поведения границы.
  - b. 1,2,3,4,5,6 поворот на 45, 90, 135, -45, -90,-135 градусов.
  - с. 7,8 начальный/конечный пиксель границы.
  - d. \* данное разбиение пример квантования