Фильтрация

Техническое зрение

Типы шумов

Импульсный шум

- Математическая модель:

$$x_{i,j} = \begin{cases} d \text{ с вероятностью } p, \\ s_{i,j} \text{ с вероятностью } (1-p), \end{cases}$$

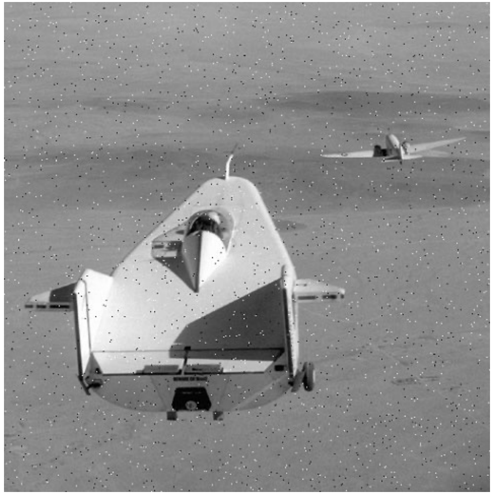
где $\{x_{i,j}\}$ – искаженное изображение,

$s_{i,j}$ – значения яркости исходного изображения,

p – вероятность появление шума в пикселе (i, j),

d – шум:

- Если $d = 0$ – шум типа «перец»,
- если $d = 255$ – шум типа «соль».



Аддитивный шум

- Математическая модель:

$$g(x, y) = f(x, y) + \eta(x, y),$$

где $f(x, y)$ – исходное изображение,

$g(x, y)$ – зашумленное изображение,

$\eta(x, y)$ – аддитивный шум.

Мультипликативный шум

- Математическая модель:

$$g(x, y) = f(x, y)\eta(x, y),$$

где $f(x, y)$ – исходное изображение,

$g(x, y)$ – зашумленное изображение,

$\eta(x, y)$ – мультипликативный шум.

Гауссовский (нормальный) шум

- Математическая модель:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}}$$

$p(z)$ – плотность распределения вероятностей,

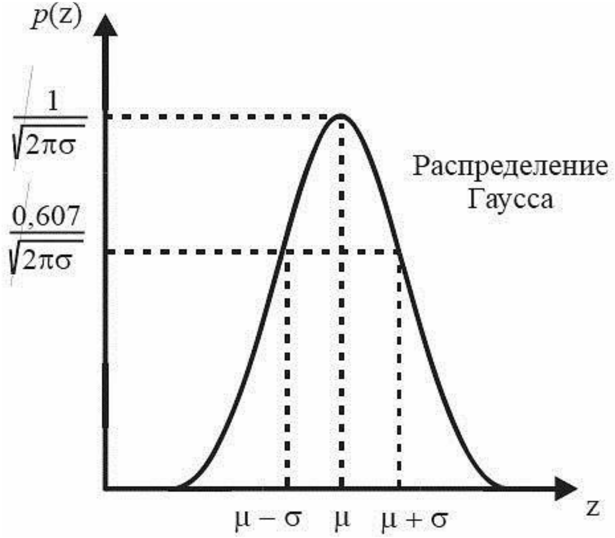
z – случайная величина,

μ – среднее значение (математическое ожидание),

σ – среднеквадратичное отклонение,

σ^2 – дисперсия.

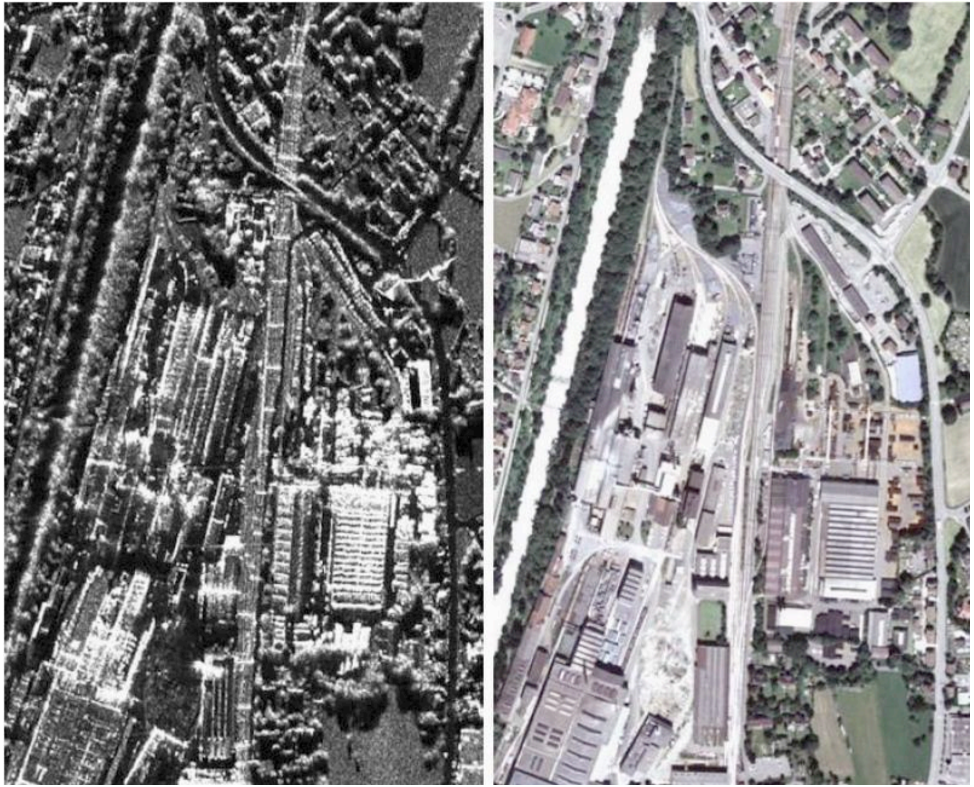
  - 67% z: [(μ − σ), (μ + σ)],
  - 96 % z: [(μ − 2σ), (μ + 2σ)].



Шум квантования

- Появление артефактов
- Не устраняется

Спекл-шум



Фильтрация изображений

Фильтрация изображений

- *Локальные преобразования* учитывают значения яркости в окрестности, называемым «окном».
- Окно описывается матрицей, называемой *маской* (*фильтром, ядром фильтра*).
- Элементы матрицы являются *коэффициентами фильтра*.
- Фильтрация изображения $f(x, y)$ с размерами M×N:

$$g(x, y) = \sum_{s} \sum_{t} w(s, t) f(x + s, y + t),$$

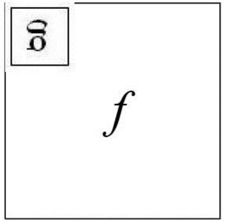
где s и t – координаты элементов маски w относительно ее центра

(в центре $s = t = 0$).

Свертка

- Фильтрацию можно выполнить с помощью операции свертки.
- Свертка показывает «схожесть» одной функции с отражённой и сдвинутой копией другой:

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



f – функция яркости изображения,

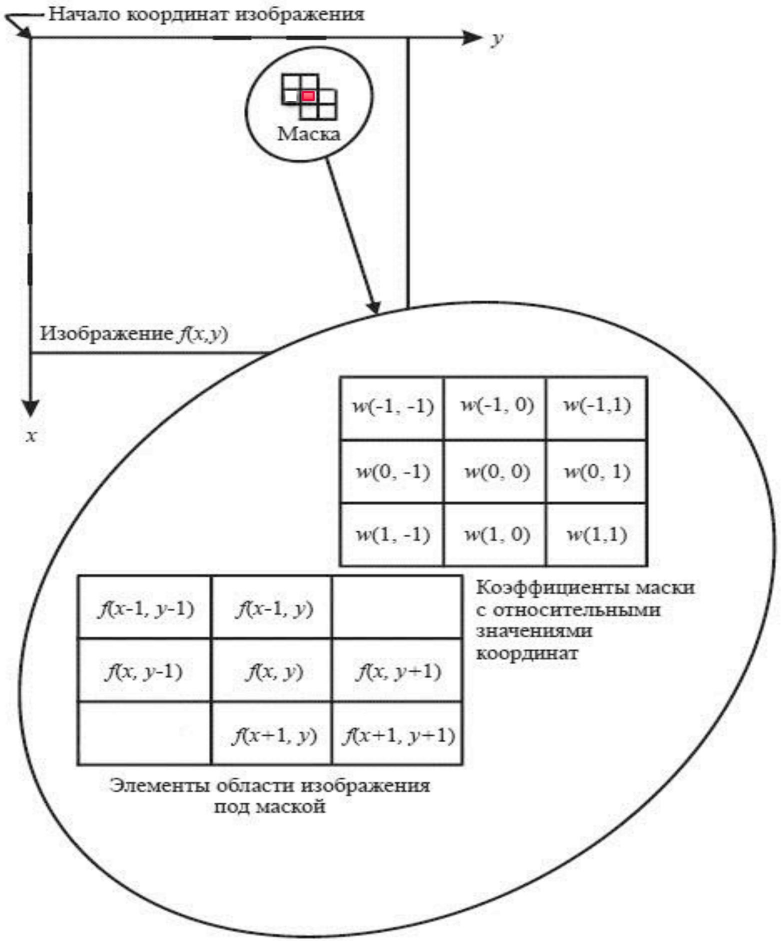
g – маска фильтра.

$(f * g)$ – операция свертки изображения f с помощью g.

Фильтрация

- Если маска у границы изображения:
  - ограничение перемещения центра окна по изображению,
  - расширение изображения путем добавления строк и столбцов с нулевыми значениями,
  - расширение изображения, путем добавления строк и столбцов со значениями симметрично границе.

Фильтрация



$w(1, -1) = w(-1, 1) = 0$

Низкочастотные фильтры

Низкочастотные фильтры

- Результатом низкочастотной фильтрации является размытие изображения.
- Отличительные признаки низкочастотных фильтров:
  - неотрицательные коэффициенты маски;
  - сумма всех коэффициентов равна единице.
- Примеры ядер низкочастотных фильтров:

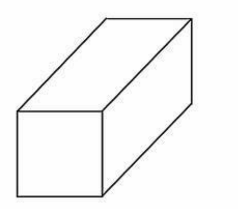
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

- Основную группу низкочастотных фильтров составляют усредняющие (или сглаживающие) фильтры.

Арифметический усредняющий фильтр

- Используются маски с одинаковыми коэффициентами, например:
  - для маски размером 3x3 коэффициенты равны 1/9,
  - при 5x5 – 1/25.

$$g(x, y) = \frac{1}{M \cdot N} \sum_{i=0}^{M} \sum_{j=0}^{N} f(i, j),$$



где $g(x, y)$ – значение пикселя **выходного** изображения,

$f(i, j)$ – **текущее** значение пикселя исходного изображения, соответствующее центру маски,

M и N – **ширина** и **высота** маски соответственно.

- Графическое представление двумерной функции фильтра похоже на параллелепипед (box-фильтр).

Геометрический усредняющий фильтр

- Формула:

$$g(x,y) = \left[\prod_{i=0}^{M}\prod_{j=0}^{N} f(i,j)\right]^{\frac{1}{M\cdot N}},$$

где $g(x,y)$ – значение пикселя **выходного** изображения,

$f(i,j)$ – **текущее** значение пикселя исходного изображения, соответствующее центру маски,

M и N – **ширина** и **высота** маски соответственно.

Гармонический усредняющий фильтр

- Формула:

$$g(x, y) = \frac{M \cdot N}{\sum_{i=0}^{M} \sum_{j=0}^{N} \frac{1}{f(i, j)}},$$

где $g(x, y)$ – значение пикселя **выходного** изображения,

$f(i, j)$ – **текущее** значение пикселя исходного изображения, соответствующее центру маски,

M и N – **ширина** и **высота** маски соответственно.

  - Хорошо работает с шумами типа «соль»,
  - Не работает с шумами типа «перец».

Контргармонический усредняющий фильтр

- Формула:

$$g(x,y) = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)^{Q+1}}{M \cdot N \cdot \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)^{Q}},$$

где Q – порядок фильтра:

- при Q > 0 подавляются шумы типа «перец»,
- при Q < 0 подавляются шумы типа «соль».
- при Q = 0 фильтр превращается в **арифметический**,
- при Q = -1 фильтр превращается в **гармонический**.

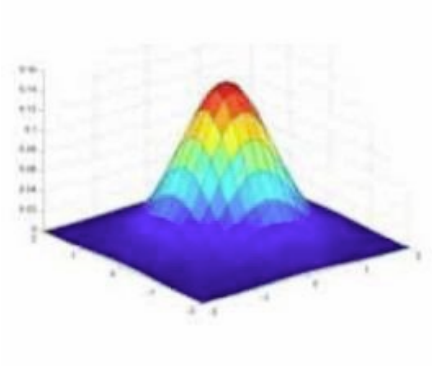
Фильтр Гаусса

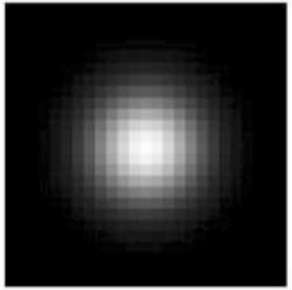
- Формула:

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

где μ – координата центральной точки,

σ – вещественная константа, определяющая ширину «колокола».





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

5 x 5, σ = 1

Свойства фильтра Гаусса

- Является сепарабельным:

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

  - Это позволяет снизить количество вычислений с $(2r+1)^2$до $2(2r+1)$ на каждый пиксель, т.е. примерно в r раз.

- Свертка, выполненная два раза с маской фильтром радиуса r, дает тот же результат, что и один раз с маской радиуса $r\sqrt{2}$.

Свойства фильтра Гаусса

- Чем больше σ, тем больше размывается изображение при применении фильтра.
- Радиус фильтра r выбирается равным 3σ.
- Размер маски равен 2r+1, т.е. она описывается матрицей размером (6σ+1)*(6σ+1).

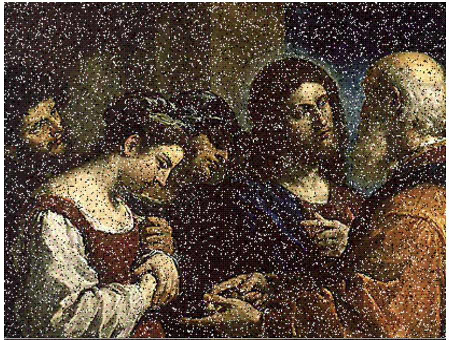
Нелинейная фильтрация

Медианный фильтр

- Окрестность может иметь произвольную форму.
- Значения интенсивности пикселей в окрестности сортируются по возрастанию.
- Результат фильтрации – центральный пиксель.

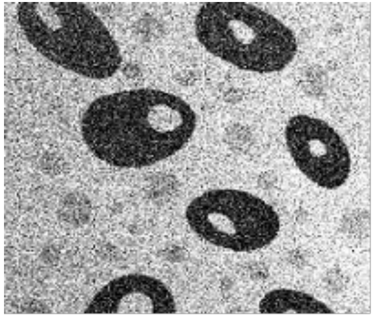
Медианная фильтрация

Исходное изображение

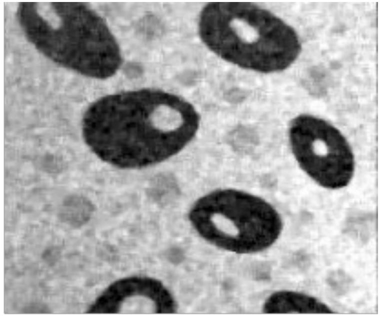


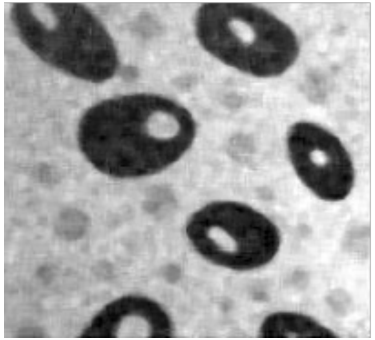
Маска 3x3,
применена 3 раза

Исходное изображение

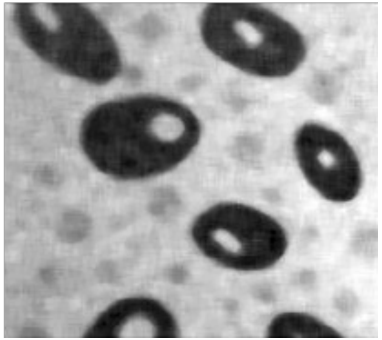


Маска 5x5





Маска 7x7



Маска 9x9

Взвешенный медианный фильтр

- В маске используется веса (2, 3 и т.д.)
- Номер медианного элемента после сортировки равен $(\mathbf{N}+\mathbf{1})/\mathbf{2}$,
  - где N – число значений яркости в сортировке, равно сумме весов маски.
  - При сортировке **повторяется интенсивность** пикселя в соответствии с коэффициентом маски.
- Свойства медианного фильтра:
  - несепарабельный;
  - нелинейный;
  - на полутоновых изображениях не вносит новых значений яркости;
  - качественно удаляет шумы импульсного типа.

Адаптивный медианный фильтр

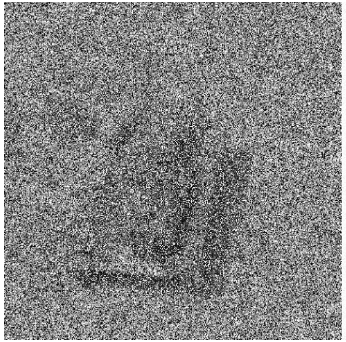
- **Идея:** увеличение размера окна S в процессе фильтрации в зависимости от локальных статистических характеристик.
- **Обозначения:**
  - $S \times S$ – размер окна.
  - Z_{min} – минимальное значения в окне;
  - Z_{max} – максимальное значения в окне;
  - Z_{med} – медианное значения в окне;
  - Z_{ij} – значение пикселя с координатами (i, j);
  - S_{max} – максимально допустимый размер окна.

Алгоритм

1. Задать исходные размеры окна фильтрации S и S_{max}.
2. Для пикселя (i, j) с яркостью Z_{ij}:
   - **Вычислить**: $Z_{min}, Z_{max}, Z_{med}, A_1 = Z_{med} - Z_{min}, A_2 = Z_{med} - Z_{max}$.
   - Если $A_1 > 0$ и $A_2 < 0$, **перейти на шаг 3**.
     - В противном случае, увеличить размер окна.
   - Если текущий размер окна $S \leq S_{max}$, **повторить шаг 2**.
     - В противном случае **результат фильтрации** равен величине $\boldsymbol{Z_{ij}}$.
3. Вычислить: $B_1 = Z_{ij} - Z_{min}, B_2 = Z_{ij} - Z_{max}$.
   - Если $B_1 > 0$ и $B_2 < 0$, **результат фильтрации равен $\boldsymbol{Z_{ij}}$**.
   - В противном случае **результат фильтрации равен $\boldsymbol{Z_{med}}$**.
4. Изменить координаты (i, j).
   - Если не вышли за пределы изображения, перейти на шаг 2.
   - В противном случае фильтрация окончена.

Адаптивная медианная фильтрация

- Преимущества:
    - оптимальное удаление импульсных шумов;
    - сглаживание других типов шумов;
    - уменьшение искажений в виде потери мелких деталей.
- Недостаток:
    - увеличение объема вычислений.



Исходное изображение

Результат фильтрации

Ранговый фильтр

- Ранговый фильтр порядка r ($1 \leq r \leq N$, где N – число элементов в окрестности) выбирает из полученного ряда элемент с номером r и присваивает его значение как результат фильтрации пикселя.
    - Если число N нечетное и $r = (N+1)/2$, фильтр становится медианным.
    - Если $r = 1$, фильтр выбирает минимальное значение яркости в окне и называется min-фильтром.
    - Если $r = N$, фильтр выбирает максимальное значение яркости в окне и называется max-фильтром.
- Ранг можно задать в процентах, тогда выбор минимального значения соответствует 0%, медианного – 50 %, а максимального – 100 %.

Активность

Каков результат медианной фильтрации?

$$\begin{bmatrix} 13 & 11 & 3 \\ 1 & \mathbf{9} & 2 \\ 0 & 2 & 14 \end{bmatrix}$$

Каков результат медианной фильтрации?

$$\begin{bmatrix} 13 & 11 & 3 \\ 1 & \mathbf{9} & 2 \\ 0 & 2 & 14 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 13 & 11 & 3 \\ 1 & \mathbf{3} & 2 \\ 0 & 2 & 14 \end{bmatrix}$$

Каков результат <u>взвешенной</u> медианной фильтрации?

$$\text{Маска } M = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

$$\text{Фрагмент изображения } Im = \begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 2 \\ 2 & 3 & 1 & 0 & 1 \\ 2 & 1 & 2 & 3 & 1 \end{bmatrix}.$$

- <u>Оставить краевые пиксели фрагмента исходными.</u>

Каков результат min/max фильтрации?

$$\text{Фрагмент изображения } Im = \begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 2 \\ 2 & 3 & 1 & 0 & 1 \\ 2 & 1 & 2 & 3 & 1 \end{bmatrix}.$$

- Оставить краевые пиксели фрагмента исходными.

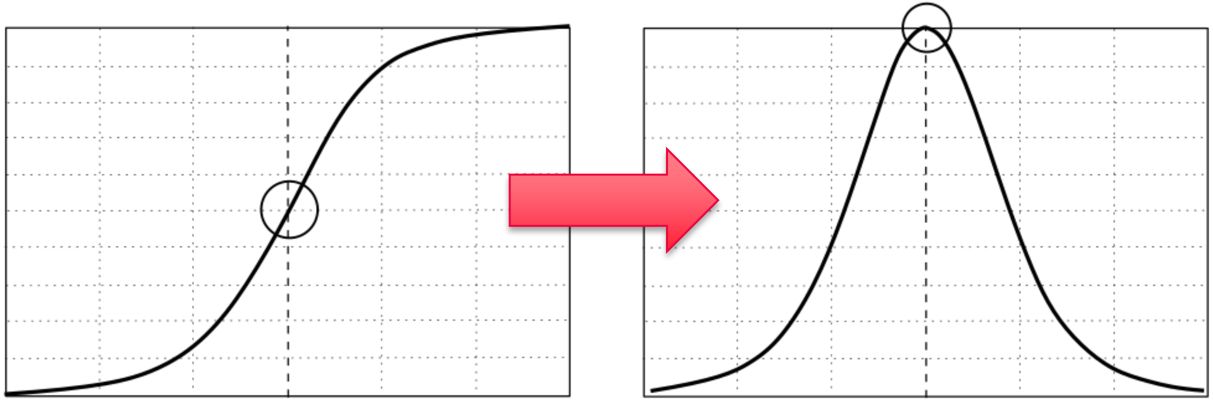
Высокочастотные фильтры

Основные понятия

- Что является высокочастотными компонентами изображения?

Идея детекторов края

- В чем основная идея?
- Производная!

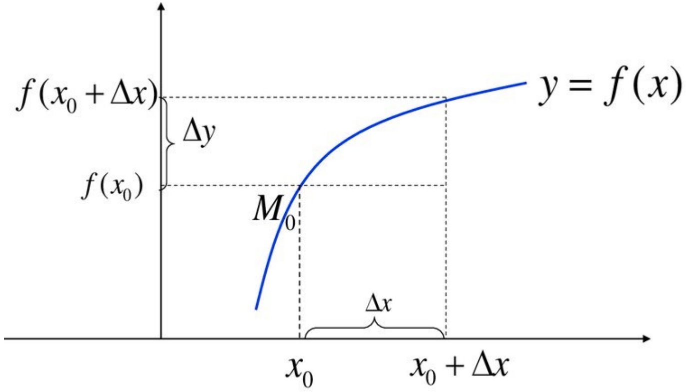


Что такое производная?

- Математическое определение:

$$f'(x) = \lim_{\Delta x \to 0} \left(\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \right),$$

где $f(x_0)$ – функция,
x_0– аргумент,
$f(x_0 + \Delta x)$ – приращение функции,
Δx – приращение аргумента.



Как ее использовать?

- Представим изображение *I* как непрерывную функцию интенсивности (яркости).
- Вычислим производную *I*.
- Мы знаем края! Роял флаш!

- Почти…

Два нюанса

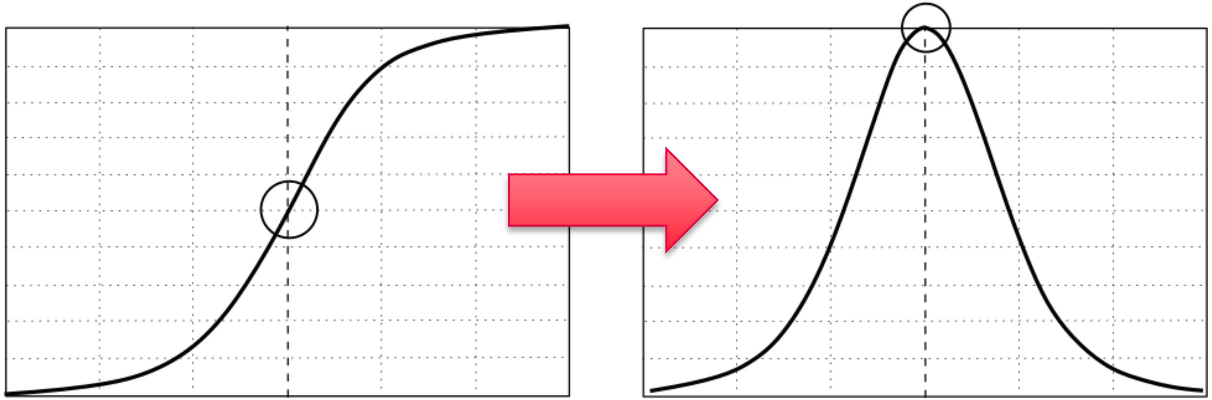
- Изображение дискретно, поэтому...
  - используем фиксированный интервал дискретности!
- У изображения две оси, относительно которой ищем производную?
  - Относительно обеих!

Дискретная реализация

- Какое минимально возможное приращение аргумента Δx?
  - 1 пиксель.
- Как оценить скорость увеличения / уменьшения яркости?
  - Вычислить градиент (производную в каждом пикселе) вдоль осей Ox ($\boldsymbol{G_x}$) и Oy ($\boldsymbol{G_y}$).
- Как найти направление градиента?
  - $\mathbf{arctan}\left(\frac{\boldsymbol{G_y}}{\boldsymbol{G_x}}\right)$
- Фул Хаус!

Свойства высокочастотных фильтров

- Используются для определения перепадов яркости и создания краевых фильтров.
- Резкое изменение яркости можно оценить анализируя первую производную функции яркости.
- Сумма коэффициентов маски должна быть равной нулю.
- Высокочастотные фильтры называют “*дифференциальными операторами*”.



Минимальный размер маски?

2x2

Фильтр Робертса

- Используются маски 2x2:

$$G_x\colon \begin{bmatrix} +1 & -1 \\ 0 & 0 \end{bmatrix}, G_y\colon \begin{bmatrix} +1 & 0 \\ -1 & 0 \end{bmatrix},$$

либо

$$G_x\colon \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}, G_y\colon \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

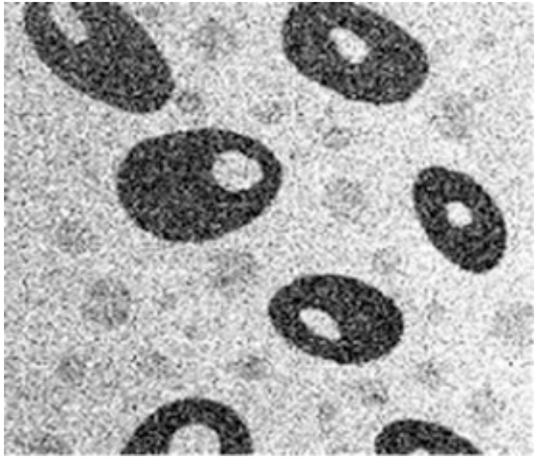
- В результате получим оценку градиента по направлениям G_x, G_y.
- Модуль градиента всех детекторов края:

$$G = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y|.$$

- Направление градиента (максимального перепада яркости):

$$\arctan\left(\frac{G_y}{G_x}\right).$$

Результат работы фильтра Робертса



Исходное изображение

Оператор Робертса – свертка по G_x и G_y

Фильтр Превитта (Преюьитта)

- Используются две матрицы размерностью 3x3:

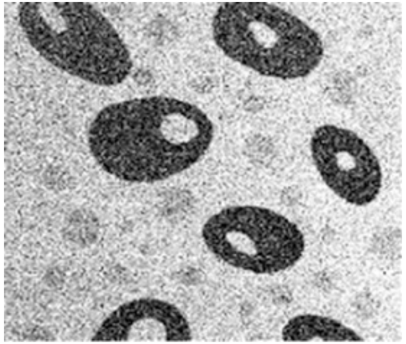
$$G_x: \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, G_y: \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

Фильтр Собела (Собеля)

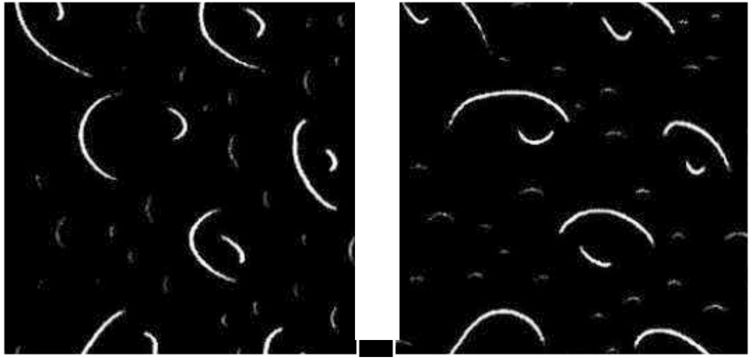
- Используются две матрицы размерностью 3х3:
- Используются разные веса в масках:

$$G_x: \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y: \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

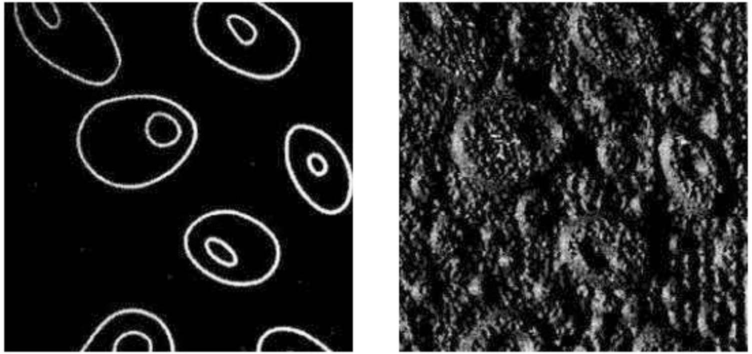
Результат работы фильтра Собела



Исходное изображение



Оператор Собела – свертка по G_x и G_y



Оператор Собела – амплитуда градиента и поле направлений

Фильтр Щарра (Scharr)

- Используются две матрицы размерностью 3x3:
- Используются разные веса в масках:

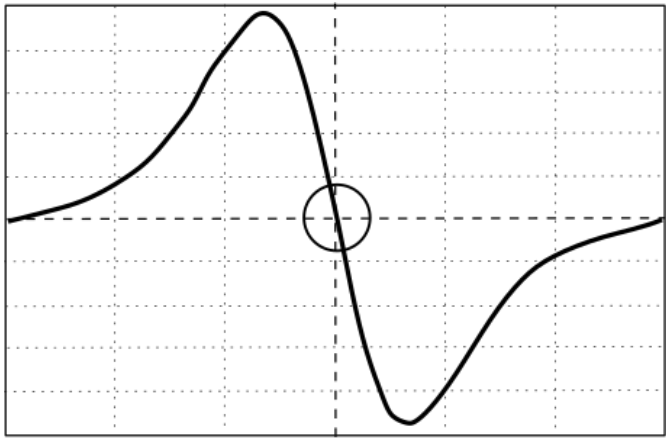
$$G_x: \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}, G_y: \begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

Фильтр Лапласа

- Используется аппроксимация вторых производных по осям *Ox* и *Oy*:

$$L\big(f(x,y)\big) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} \text{ – Лапласиан от изображения } f(x,y).$$

- Градиент вычисляется независимо от направления – границы выделяются точнее.



Фильтр Лапласа

$L\big(f(x,y)\big) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$ – Лапласиан от изображения $f(x,y)$:

$$L\big(f(x,y)\big) = \Big[\big(f(x,y) - f(x-1,y)\big) - \big(f(x+1,y) - f(x,y)\big)\Big] +$$
$$+\Big[\big(f(x,y) - f(x,y-1)\big) - \big(f(x,y+1) - f(x,y)\big)\Big] =$$
$$= -\boldsymbol{f(x,y-1) - f(x-1,y) - f(x,y+1) - f(x+1,y) + 4f(x,y)}$$

Активность

Какова маска фильтра Лапласа?

$$L\big(f(x,y)\big) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$$ – Лапласиан от изображения $f(x,y)$

$$L\big(f(x,y)\big) =$$
$$= -f(x,y-1) - f(x-1,y) - f(x,y+1) - f(x+1,y) + 4f(x,y)$$

$$w(s,t) = ?$$

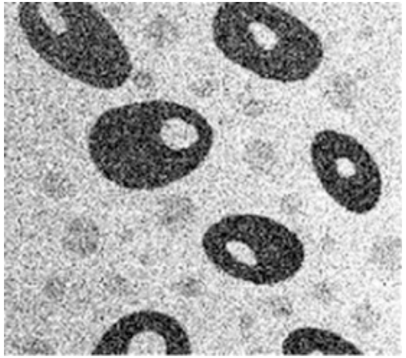
Какова маска фильтра Лапласа?

$$L\big(f(x,y)\big) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$$ – Лапласиан от изображения $f(x,y)$

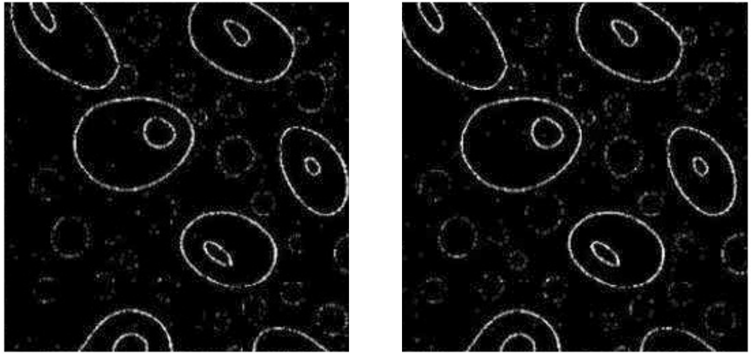
$$L\big(f(x,y)\big) =$$
$$= -f(x,y-1) - f(x-1,y) - f(x,y+1) - f(x+1,y) + 4f(x,y)$$

$$w(s,t) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Результат работы фильтра Лапласа



Исходное изображение



Оператор Лапласа – маски 3*3 и 5*5

Локализация края

- Фильтрация градиента:

$$G(x, y) = \begin{cases} 0, \text{если } G(x, y) \le T, \\ 1, \text{если } \ G(x, y) > T. \end{cases}$$

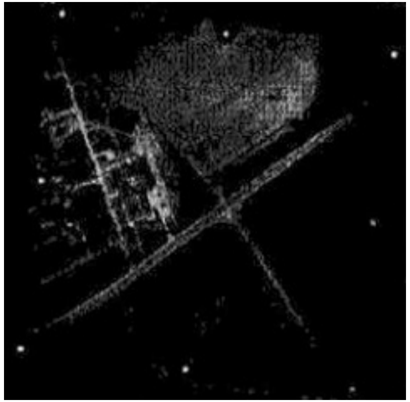
- Сигма-фильтр: $T = M_0 + \alpha\sigma$,

где T – порог,

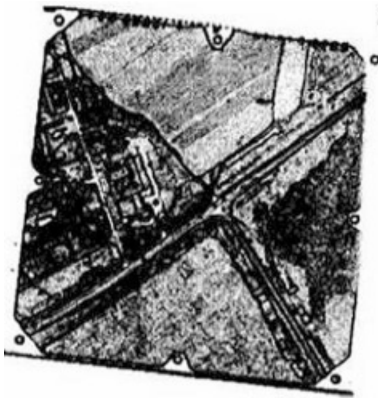
α – параметр фильтрации;

M_0 – среднее значение модуля градиента;

σ – СКО модуля градиента изображения.



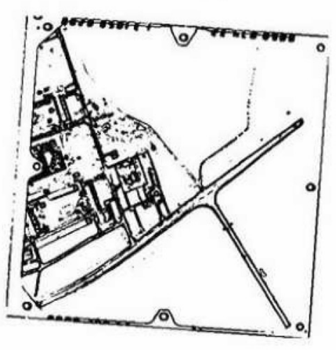
Исходное изображение



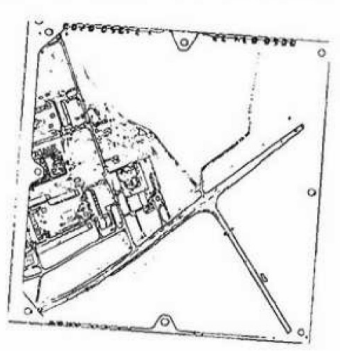
Модуль градиента

Утончение края

- **Задача:** получение контура с единичной шириной.
- **Требования к утончению:**
  - если объект связный, то результат должен быть связным;
  - средняя линия должна проходить через точки с наибольшим значением интенсивности.

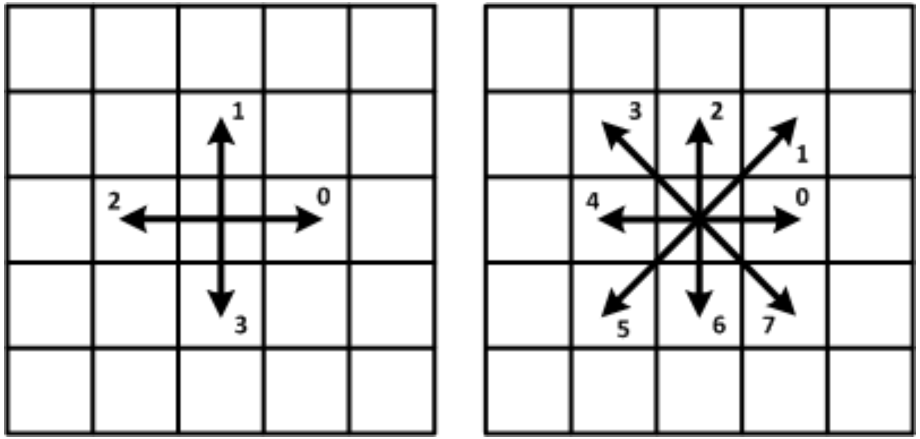


Результат локализации



Результат утончения

Принципиальные направления



По четырехсвязности   По восьмисвязности

Связные области

- Связная область – это область с пикселями одинаковой интенсивности.
- Сколько связных областей в фрагменте бинарного изображения *I*:

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

1. По восьмисвязности;
2. По четырехсвязности;

?

Связные области

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

По восьмисвязности

<u>2 области</u>

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

По четырехсвязности

<u>7 областей</u>

Активность

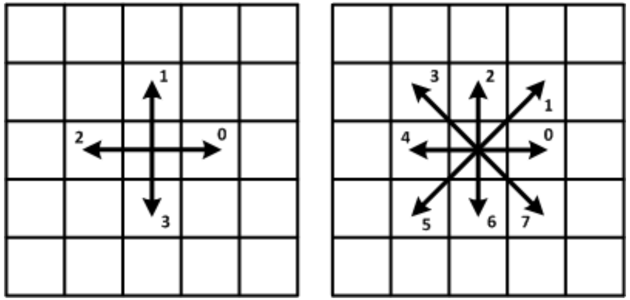
Связные области

- Сколько связных областей во фрагментах бинарного изображения:

$$I1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \qquad I2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

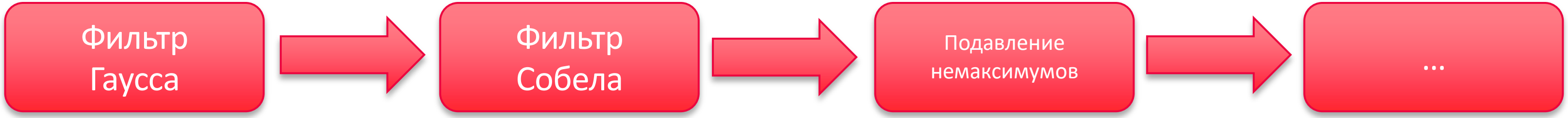
1. По восьмисвязности;
2. По четырехсвязности;

?



Алгоритм Кэнни (Canny)

Цель: получение контура единичной ширины.



Алгоритм:

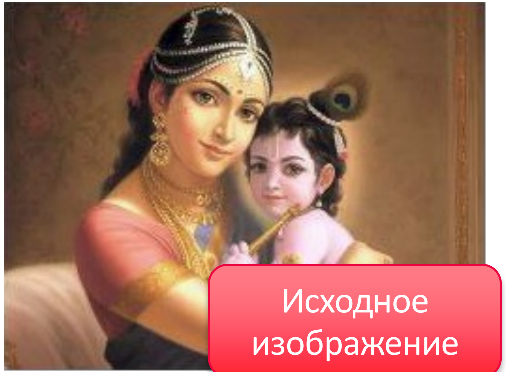
1. **Сглаживание** изображения фильтром Гаусса.
2. **Вычисление градиентов** пикселей фильтром Собела:
   - Направление градиентов округляется с шагом **45 градусов**.
3. **Подавление немаксимумов** модуля градиента.
   - Пиксель является **краем** если его градиент больше градиента соседних пикселей,
   - В противном случае пиксель является **немаксимумом**.

Алгоритм Кэнни (Canny)

4. Выполнение **двойной пороговой фильтрации**:
   - если значение пикселя выше порога T_2, то пиксель **краевой**;
   - если значение пикселя меньше порога T_1, пиксель **не краевой**;
   - если значение между порогами T_1 и T_2, пиксель **неоднозначен**.
5. Уточнение края **трассировкой области неоднозначности**:
   - если пиксель из области неоднозначности связан по восьмисвязности с краем, то пиксель является **краем;**
   - в противном случае пиксель **не краевой.**

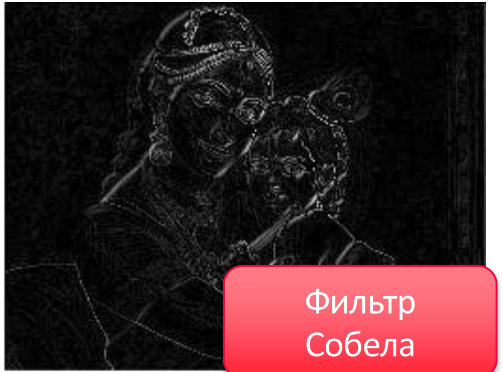
… → Пороговая фильтрация → Уточнение краев → Контур

Результат использования фильтра Кэнни

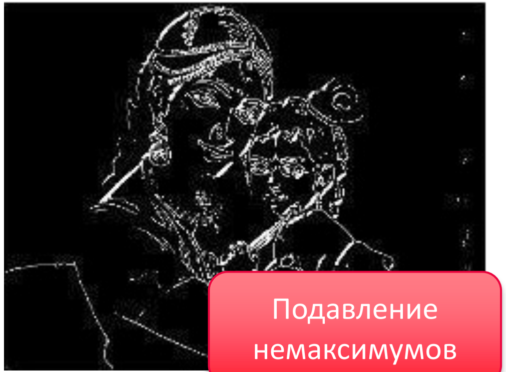


Исходное изображение

RGB->Gray Фильтр Гаусса

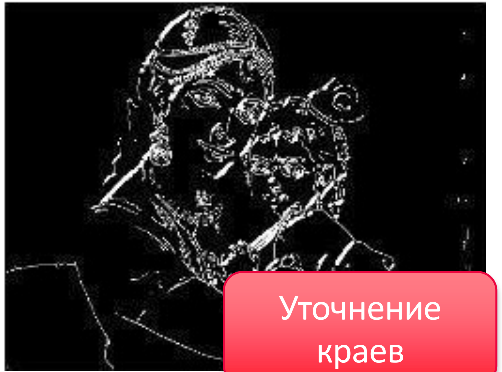


Фильтр Собела



Подавление немаксимумов

Пороговая фильтрация



Уточнение краев

Вопросы?

ITsMOre than a UNIVERSITY

s.shavetov@itmo.ru