

# СИСТЕМЫ ОБРАБОТКИ

## МЕДИАДААННЫХ

### НЕЛИНЕЙНАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

**д.т.н. Вашкевич М. И.**

**[vashkevich@bsuir.by](mailto:vashkevich@bsuir.by)**



Белорусский государственный университет  
информатики и радиоэлектроники

**Кафедра электронных вычислительных средств**

# Зачем нужна нелинейная фильтрация?

## Линейные фильтры:

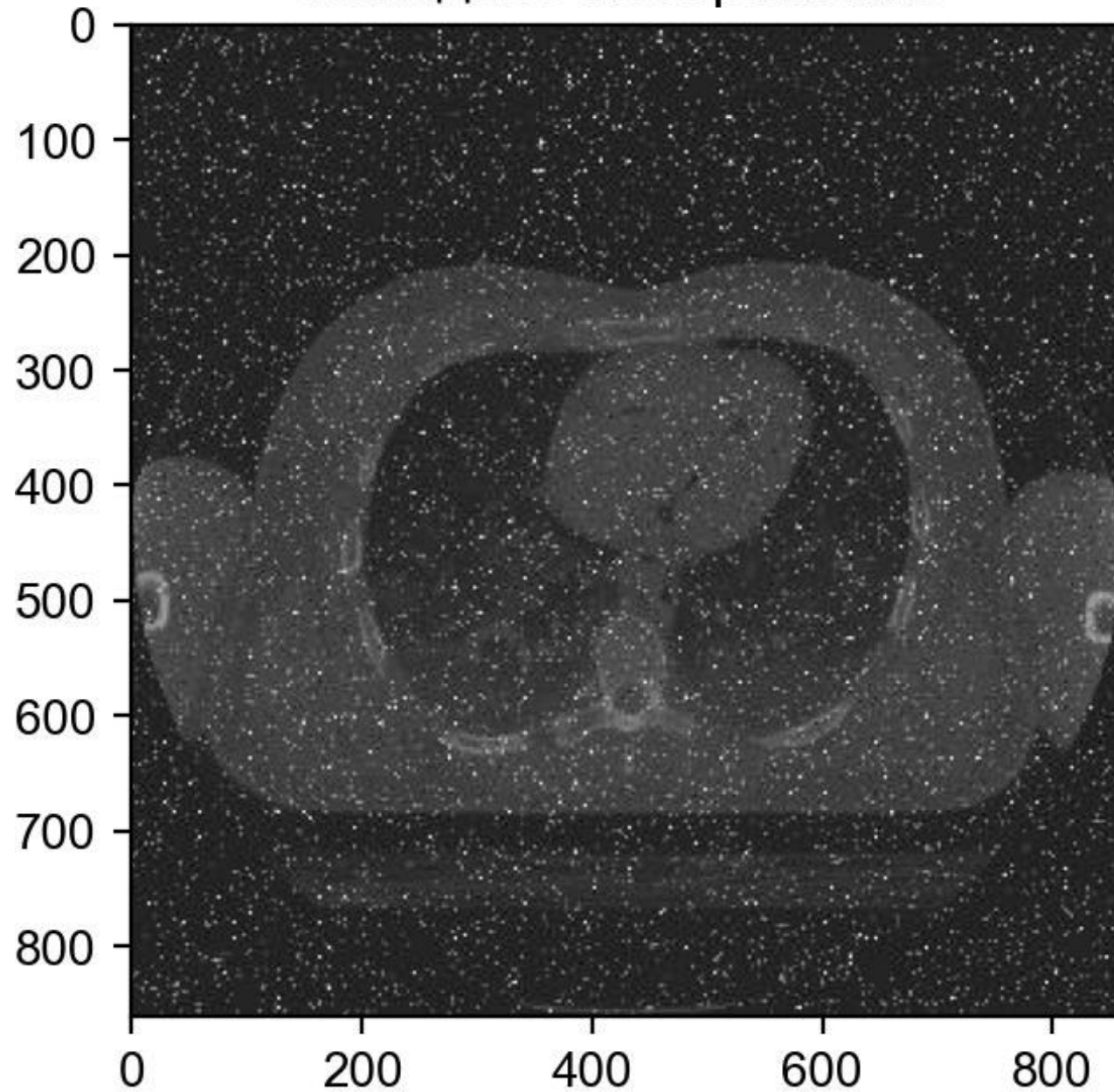
- Имеют тенденцию к **размытию контуров и деталей** изображения;
- Имеют низкую производительность с шумами негауссовского типа;
- Основаны на предположении, что и изображение, и шум имеют гауссово распределение;

## Нелинейные фильтры

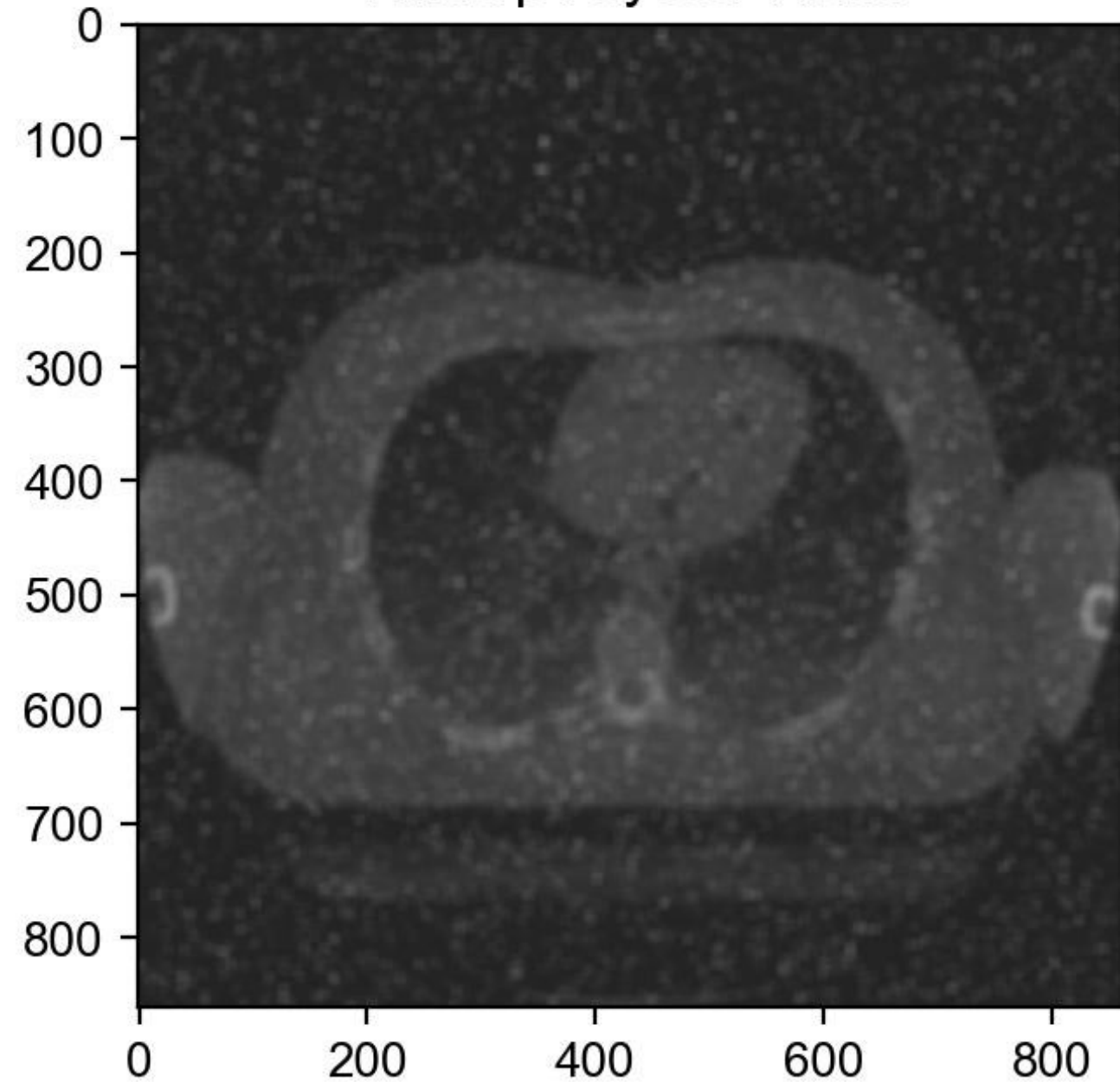
- Очень эффективны для удаления с изображения импульсных шумов;
- Основаны на предположении, что изображение и шум имеют негауссово распределение;
- Форма окна не обязательно должна быть квадратной
- Специальные формы фильтра могут сохранять линейные структуры и контура на изображении.
- Более сложны при проектировании и реализации.

# Удаление импульсного шума гауссовым фильтром

Исходное изображение



Фильтр Гаусса 11x11



В этом случае обычное размытие с помощью фильтра Гаусса не позволяет удалить зашумленные пиксели и вместо этого превращает их в более мягкие (но все еще видимые) пятна.

# Линейные фильтры

✓ Фильтр называется **линейным**, если для любых  $\alpha, \beta \in \mathbb{R}$  выполняется условие

$$\alpha y_1 + \beta y_2 = T\{\alpha x_1 + \beta x_2\},$$

где  $y_1 = T\{x_1\}$  и  $y_2 = T\{x_2\}$ .

✓ Фильтр  $y = T\{x\}$  называется **однородным**, если для любого  $\alpha \in \mathbb{R}$  справедливо:

$$\alpha y = T\{\alpha x\}.$$

- Это более слабое условие, чем линейность.
- Однородность является естественным условием для масштабно-инвариантных систем.

# 1D медианный фильтр

$(2R + 1)$ -точечный медианный фильтр задается выражением:

$$y(n) = \text{median}\{x(n + w), \quad w \in \{-R, \dots, R\}\}$$

✓ Рассмотрим 3-точечный 1D медианный фильтр:

$n$	0	1	2	3	4	5	6	7
$x(n)$	1	2	1000	2	1	3	2	1
$y(n)$	–	2	2	2	2	2	2	–

✓ Является ли медианный фильтр:

- Линейным?
- Однородным?

# Медианный фильтр с квадратным окном

Шаги медианной фильтрации для изображений:

- 1) Представить пиксели окрестности входного изображения в виде одномерного массива.





# Медианный фильтр с квадратным окном

Шаги медианной фильтрации для изображений:

- 1) Представить пиксели окрестности входного изображения в виде одномерного массива.
- 2) Упорядочить по возрастанию значения пикселей внутри окрестности.



# Медианный фильтр с квадратным окном

Шаги медианной фильтрации для изображений:

- 1) Представить пиксели окрестности входного изображения в виде одномерного массива.
- 2) Упорядочить по возрастанию значения пикселей внутри окрестности.
- 3) Найти значение медианы. Для окрестности  $3 \times 3$  элементов медианой будет 5-е значение по величине.





# Медианный фильтр с квадратным окном

Шаги медианной фильтрации для изображений:

- 1) Представить пиксели окрестности входного изображения в виде одномерного массива.
- 2) Упорядочить по возрастанию значения пикселей внутри окрестности.
- 3) Найти значение медианы. Для окрестности  $3 \times 3$  элементов медианой будет 5-е значение по величине.
- 4) Присвоить полученное значение обрабатываемому элементу.



# Задача

Выполнить фильтрацию изображения  $f(x, y)$  медианным фильтром по окрестности  $3 \times 3$ . Перед обработкой изображения выполнить расширение путем добавления на границе нулевых значений.

$f(x, y)$

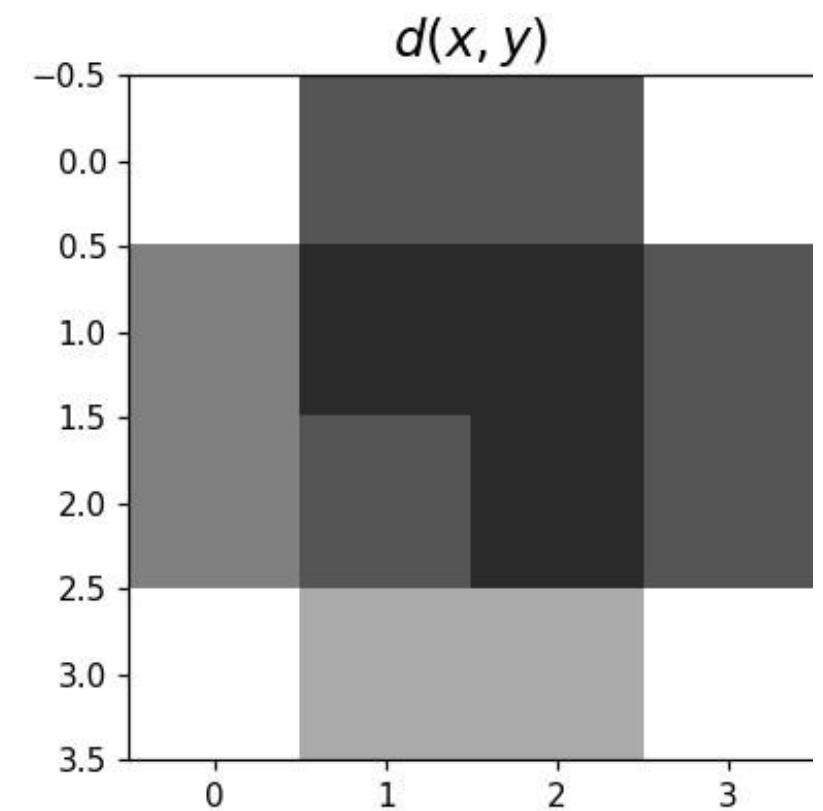
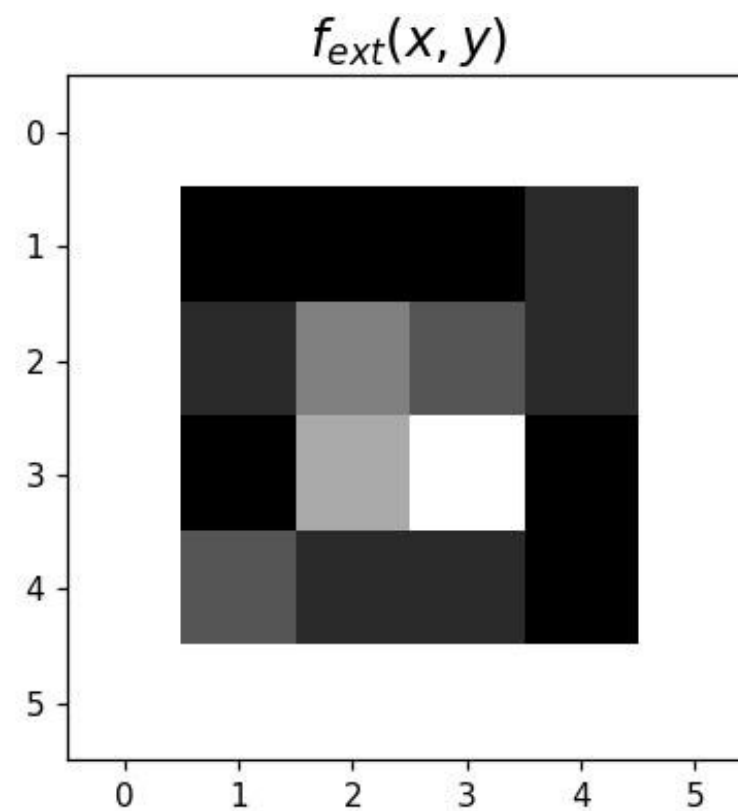
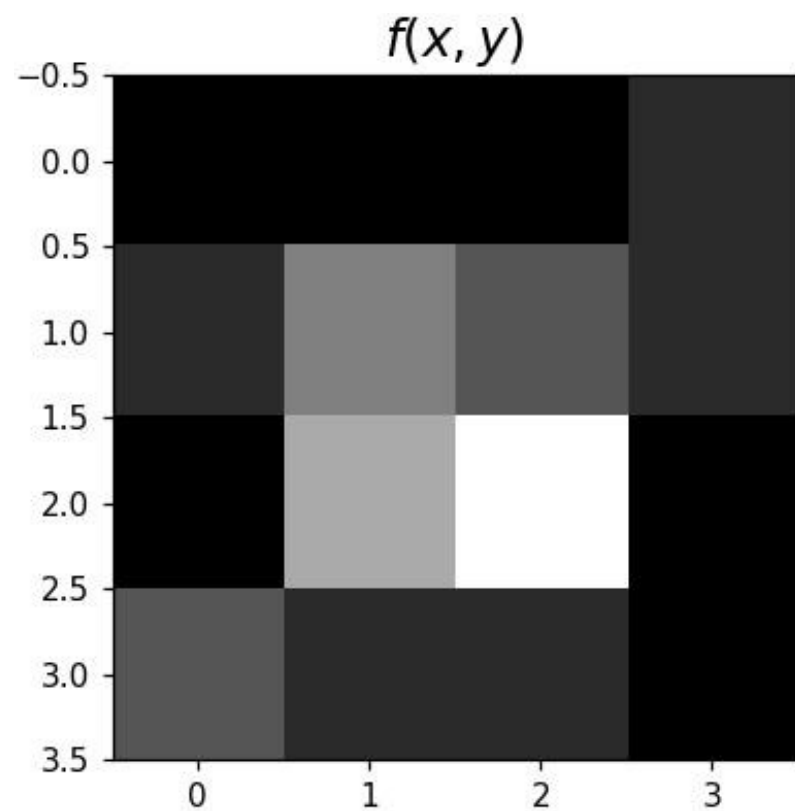
	0	1	2	3	$y$
0	1	1	1	2	
1	2	4	3	2	
2	1	5	7	1	
3	3	2	2	1	

$x$

# Ответ к задаче

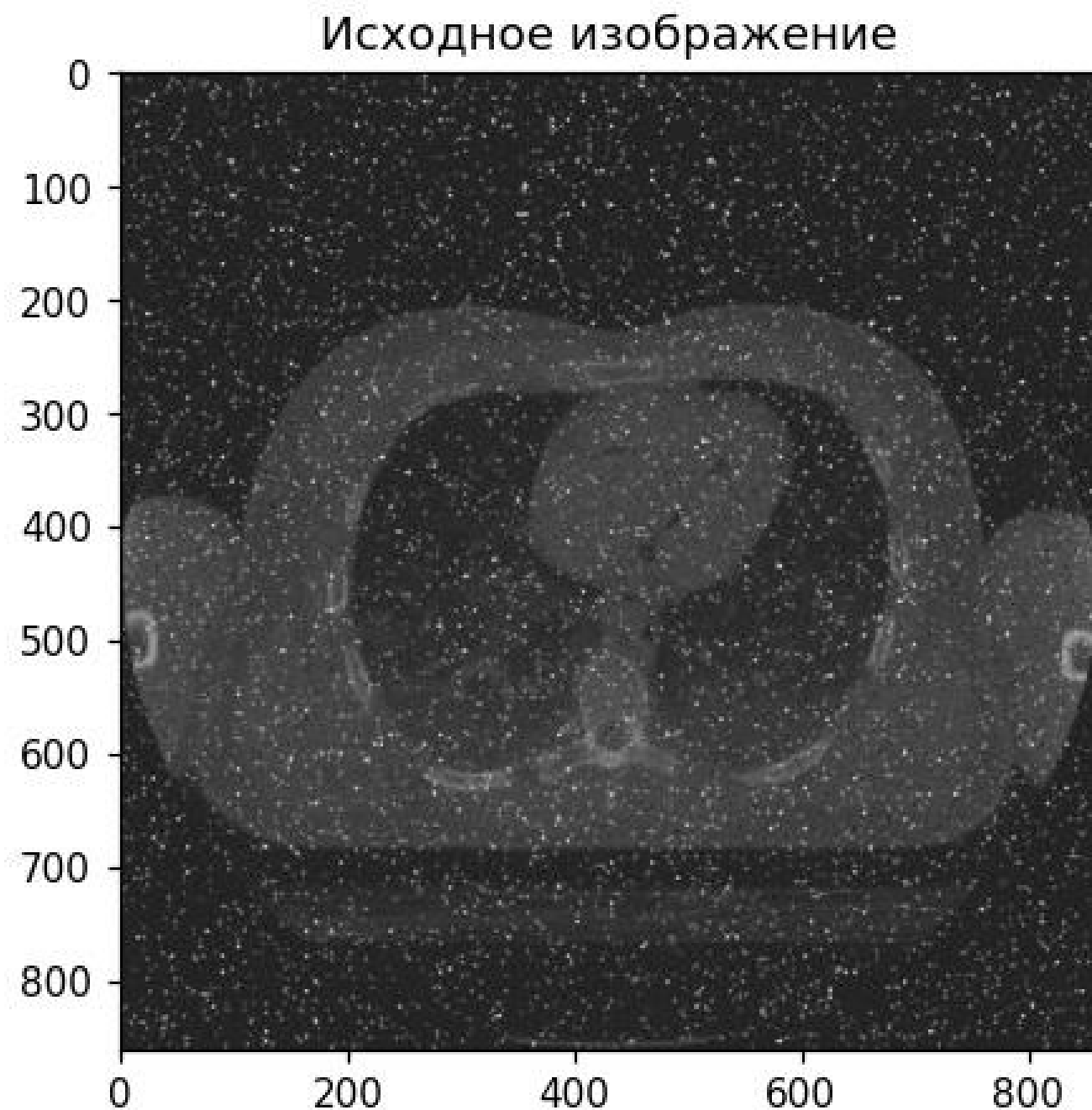
$$d(x, y) =$$

$$\begin{bmatrix} [7 & 3 & 3 & 7] \\ [4 & 2 & 2 & 3] \\ [4 & 3 & 2 & 3] \\ [7 & 5 & 5 & 7] \end{bmatrix}$$



# Пример применения медианного фильтра

Обработка КТ изображения медианным фильтром 11x11



# Недостатки медианного фильтра

- ✓ Более высокая вычислительная сложность в сравнении с гауссовым фильтром и однородным усредняющим фильтром.
- ✓ Медианный фильтр неэффективен для удаления гауссовского шума.

Исходное изображение



После зашумления



После медианного фильтра



После фильтра Гаусса



# Усредняющий фильтр с $\alpha$ -обрезкой

В случае, если на изображении есть гауссовский шум и возможно появление импульсного шума, лучшим выбором может быть использование **усредняющего фильтра с  $\alpha$ -обрезкой**, которое усредняет все пиксели, за исключением доли, которая является наименьшей и наибольшей.

1	2	1	2	4
2	1	3	5	8
1	3	7	6	9
3	4	8	6	7
4	5	7	8	9

$$\text{median} = 4$$

1	2	1	2	4
2	1	3	5	8
1	3	7	6	9
3	4	8	6	7
4	5	7	8	9

$$\alpha\text{-mean} = 4,6$$



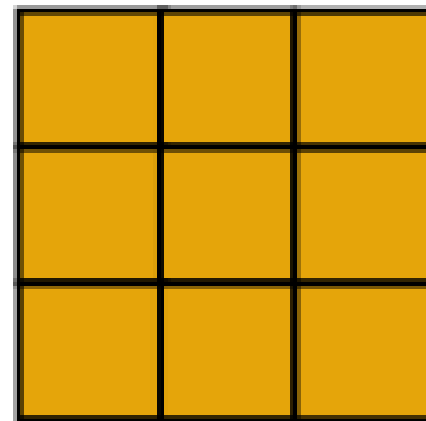
# Медианная фильтрация: квадратное окно

Медианная фильтрация изображения с использованием квадратного окна 3x3

$f(x, y)$

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

Форма окна



$g(x, y)$

100	100	100	100	100
100	100	200	100	100
100	200	200	200	100
100	100	195	100	100
100	100	100	100	100

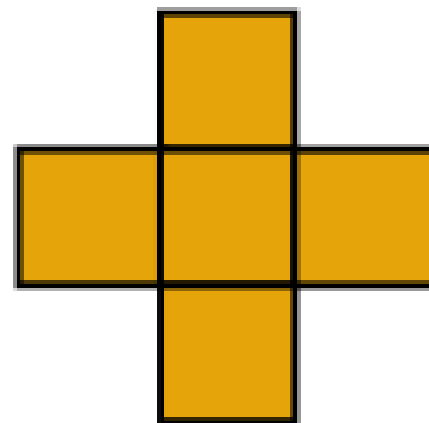
# Медианная фильтрация: перекрестное окно

Медианная фильтрация изображения с использованием перекрестного окна 3x3

$f(x, y)$

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

Форма окна



$g(x, y)$

100	100	100	100	100
100	195	200	200	100
100	200	200	200	100
100	195	200	195	100
100	100	100	100	100

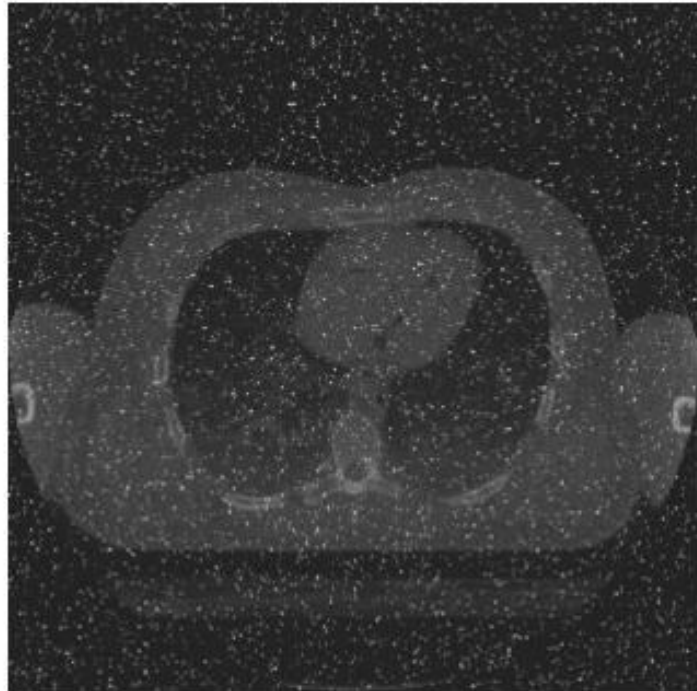
Обратите внимание, что края центрального квадрата лучше сохранились.

# Ранговая фильтрация

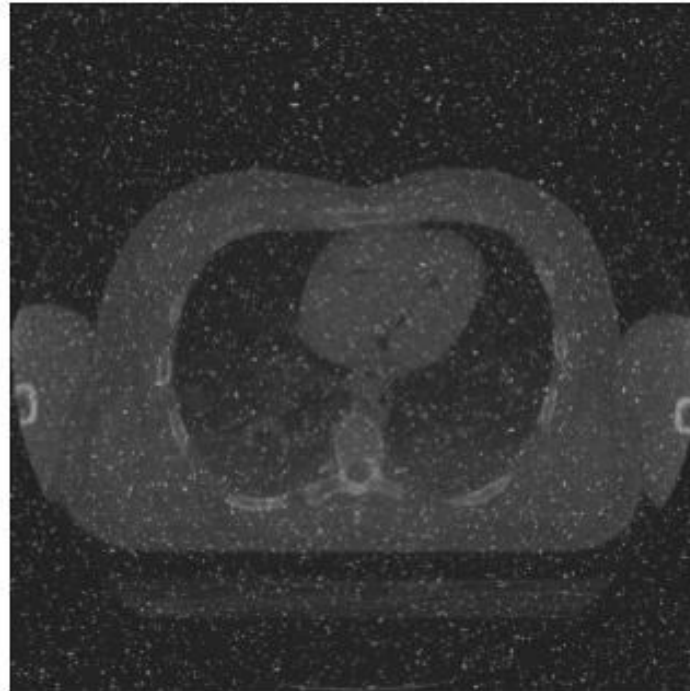
- ✓ Медианный фильтр является частным случаем класса фильтров, называемых **ранговыми** или **порядковыми**.
- ✓ **Ранговый фильтр** порядка  $r$  ( $1 \leq r \leq N$ , где  $N$  – число элементов в окрестности) выбирает из полученного ряда элемент с номером  $r$  и присваивает его значение как результат фильтрации пикселя исходного изображения.
- ✓ Если число  $N$  нечетное и  $r = (N + 1)/2$ , то ранговый 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**.

# Пример ранговой фильтрации

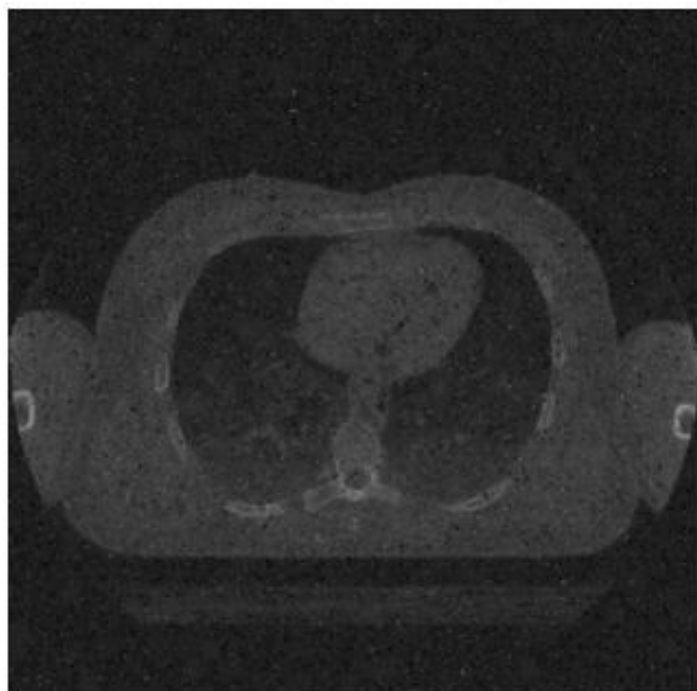
Исходное изображение



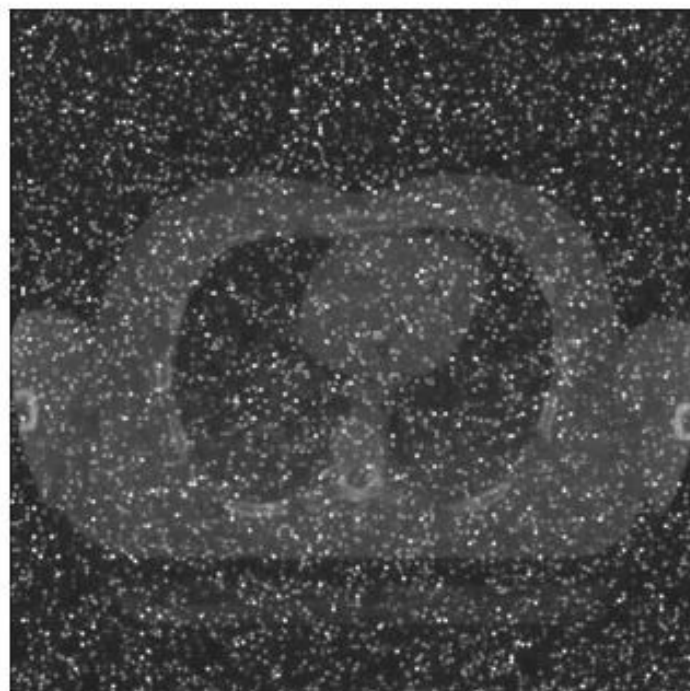
Медианный фильтр



Min-фильтр



Max-фильтр



Размер окна: 3x3

# Билатеральный фильтр

Билатеральный фильтр сочетает в себе идею использования взвешивающего ядра и «мягкого» (*soft*) «отбрасывания» пикселей со значениями, которые сильно отличаются от центрального.

Выход билатерального фильтра является взвешенной комбинацией соседних пикселей:

$$g(x, y) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

Весовые коэффициенты  $w(i, j, k, l)$  зависят от произведения пространственного фильтра (*domain filter*):

# Билатеральный фильтр

$$g(x, y) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

Весовые коэффициенты  $w(i, j, k, l)$  зависят от произведения пространственного фильтра (*domain filter*):

$$d(i, j, k, l) = \exp\left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d}\right),$$



# Билатеральный фильтр

$$g(x, y) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

Весовые коэффициенты  $w(i, j, k, l)$  зависят от произведения пространственного фильтра (*domain filter*):

$$d(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d}\right),$$

и фильтра, зависящего от данных, (*range-filter*):

$$r(i, j, k, l) = \exp\left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r}\right).$$

# Билатеральный фильтр

$$g(x, y) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

Весовые коэффициенты  $w(i, j, k, l)$  зависят от произведения пространственного фильтра (*domain filter*):

$$d(i, j, k, l) = \exp\left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d}\right),$$

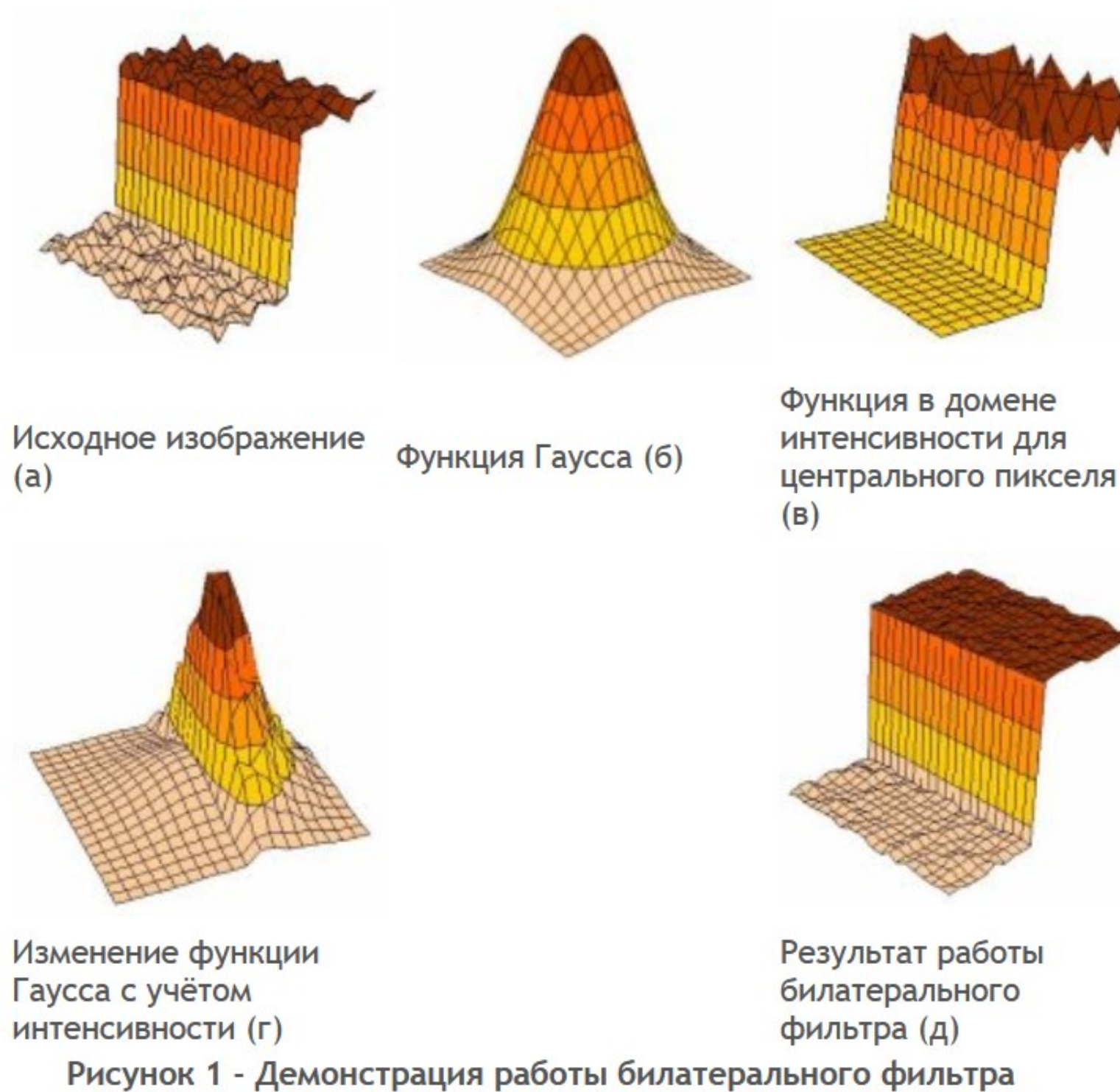
и фильтра, зависящего от данных, (*range-filter*):

$$r(i, j, k, l) = \exp\left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r}\right).$$

После перемножения получается зависящая от данных **билатеральная весовая функция**:

$$w(i, j, k, l) = \exp\left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r}\right).$$

# Билатеральный фильтр



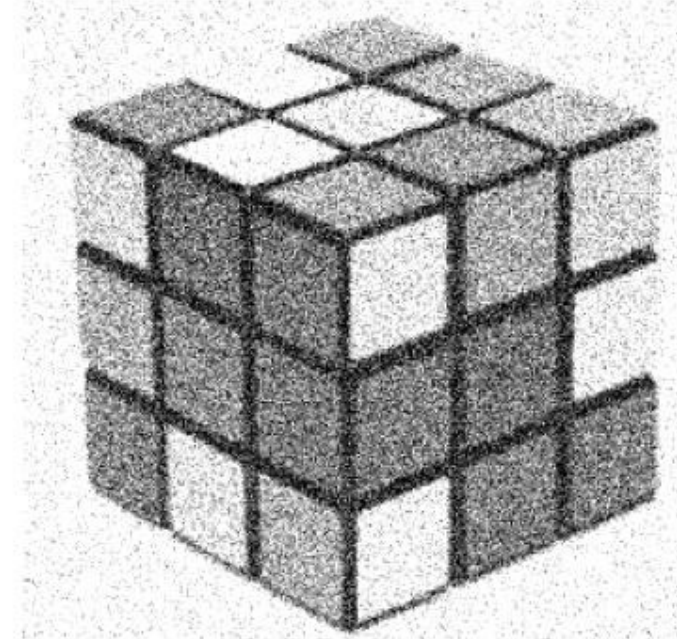
Источник изображения: <http://old.unick-soft.ru/Articles.cgi?id=11>

# Пример билатеральной фильтрации

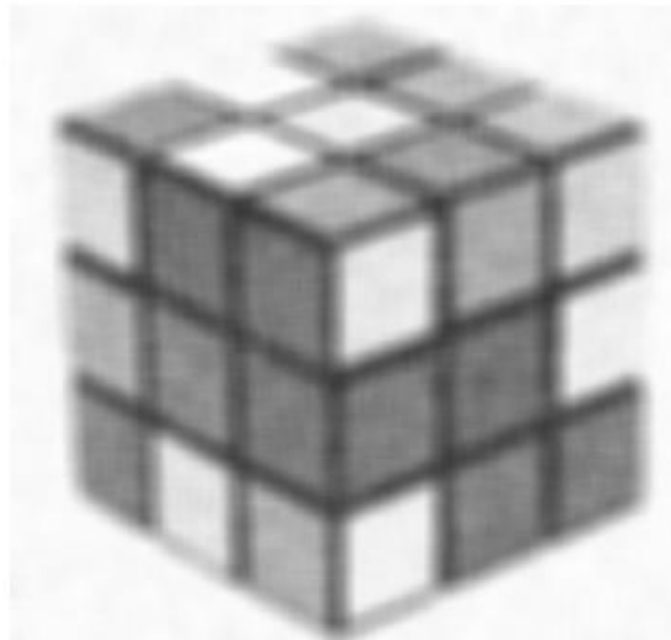
Исходное изображение



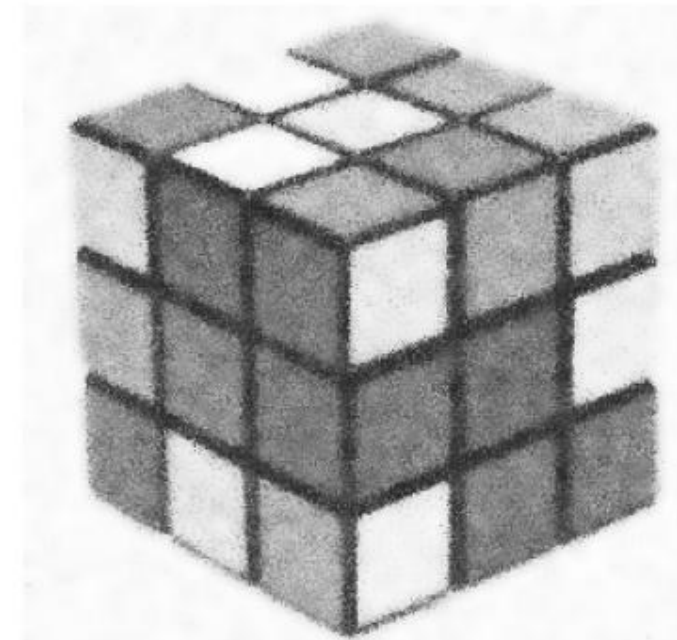
После добавления шума



Фильтр Гаусса



Билатеральный фильтр



# Литература

Richard Szeliski *Computer Vision. Algorithms and Applications. 2<sup>nd</sup> Edition* - Springer, 2022. p.105-109.