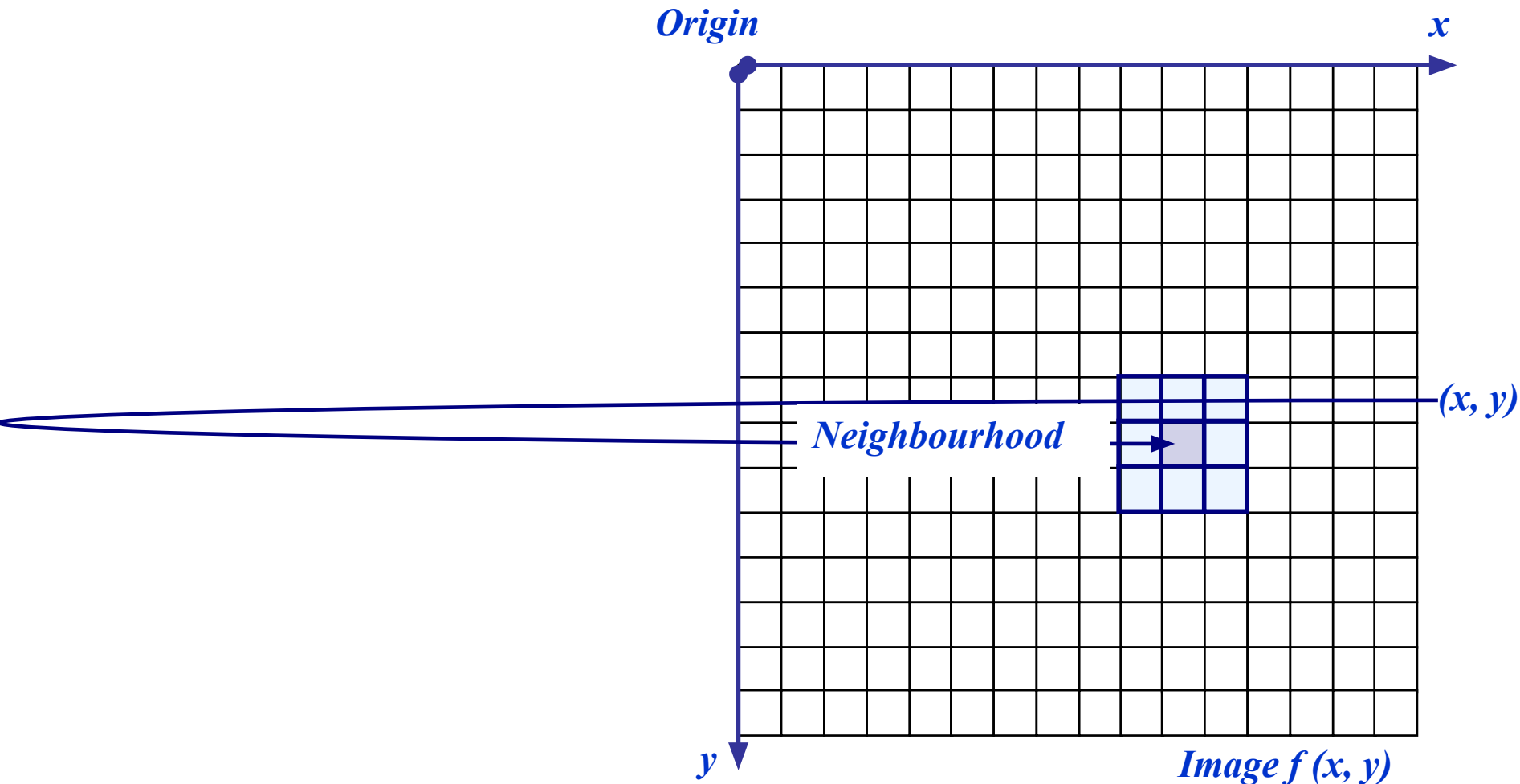
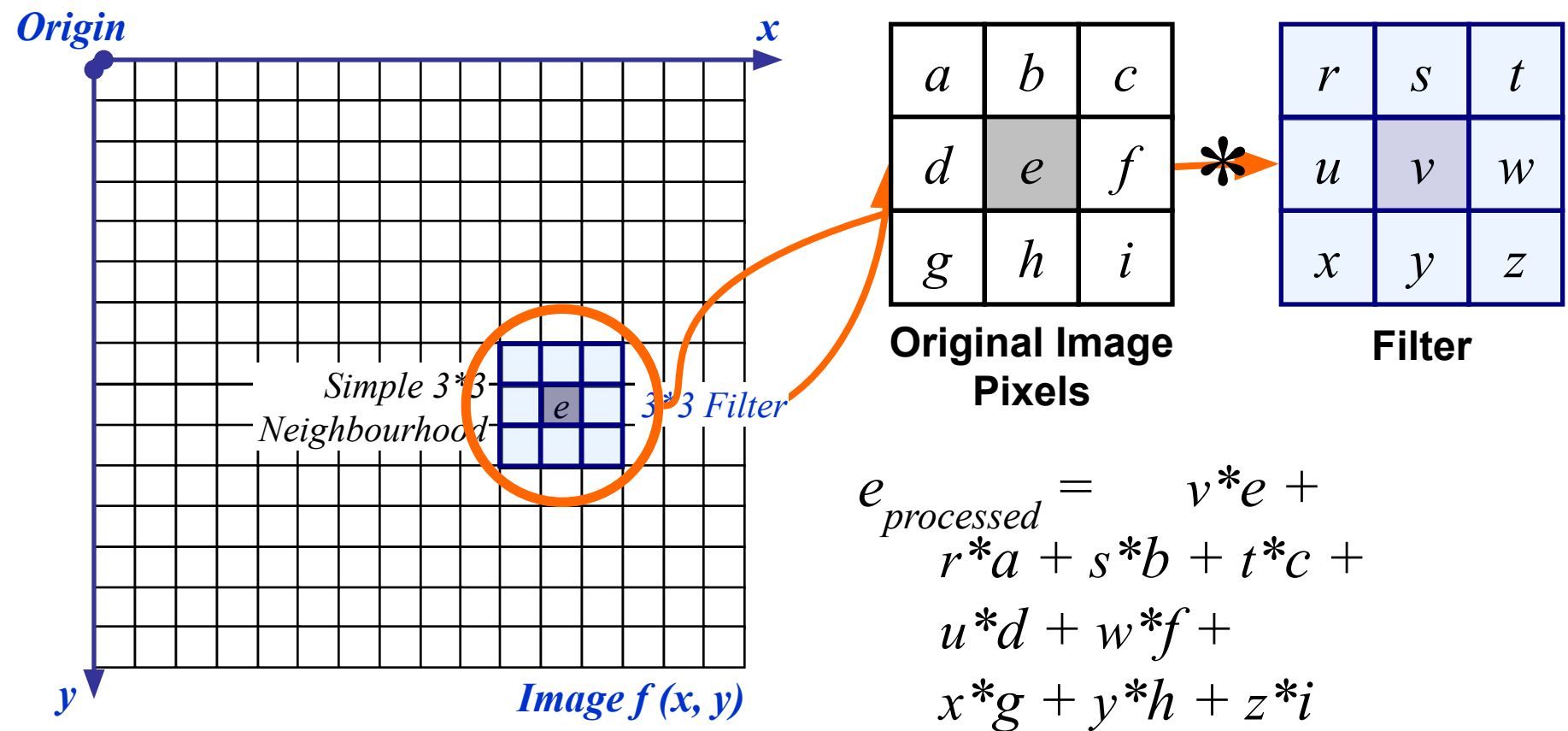


Локальные преобразования

фильтр, маска, шаблон, окно

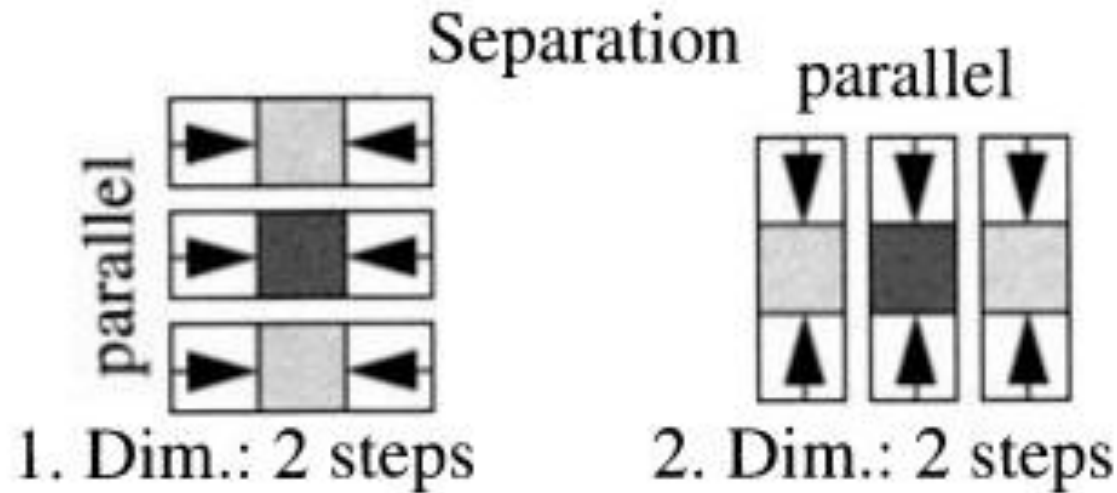


Пространственная фильтрация



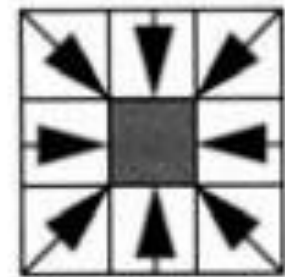
Процесс повторяется для каждого пиксела изображения

Пространственная фильтрация



$$2*(n-1)$$

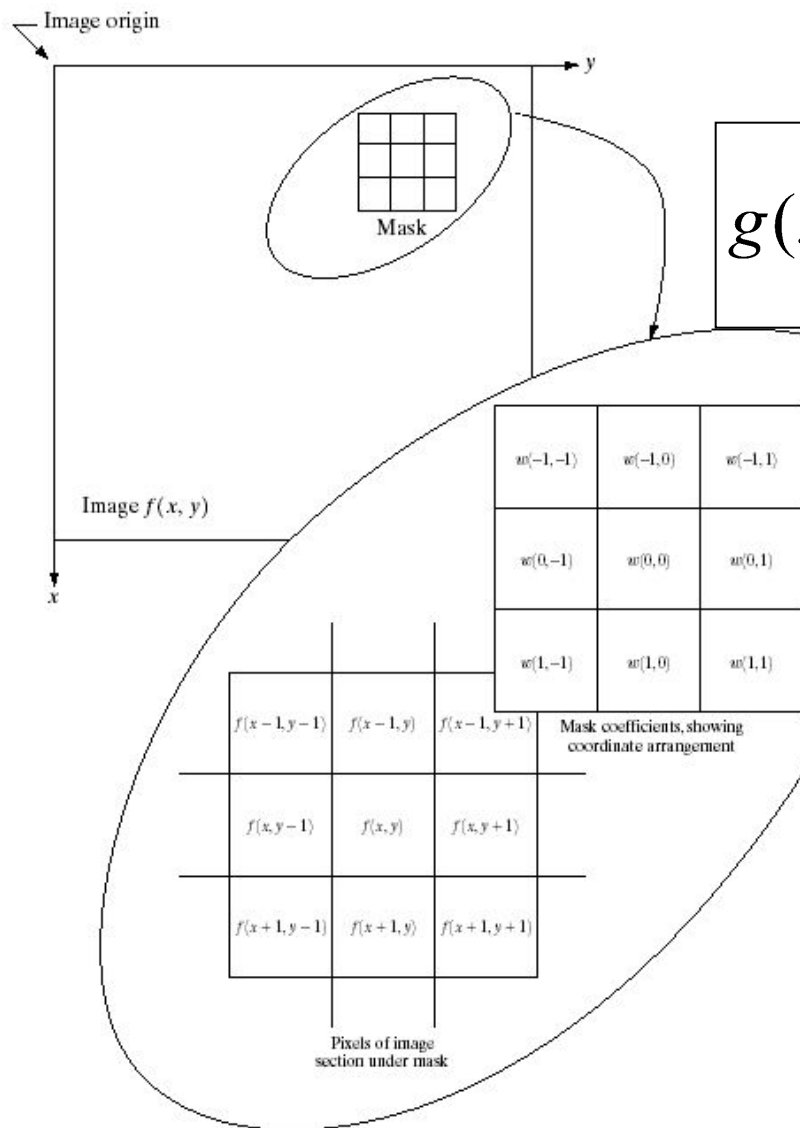
Naive Method



8 steps

$$n^2-1$$

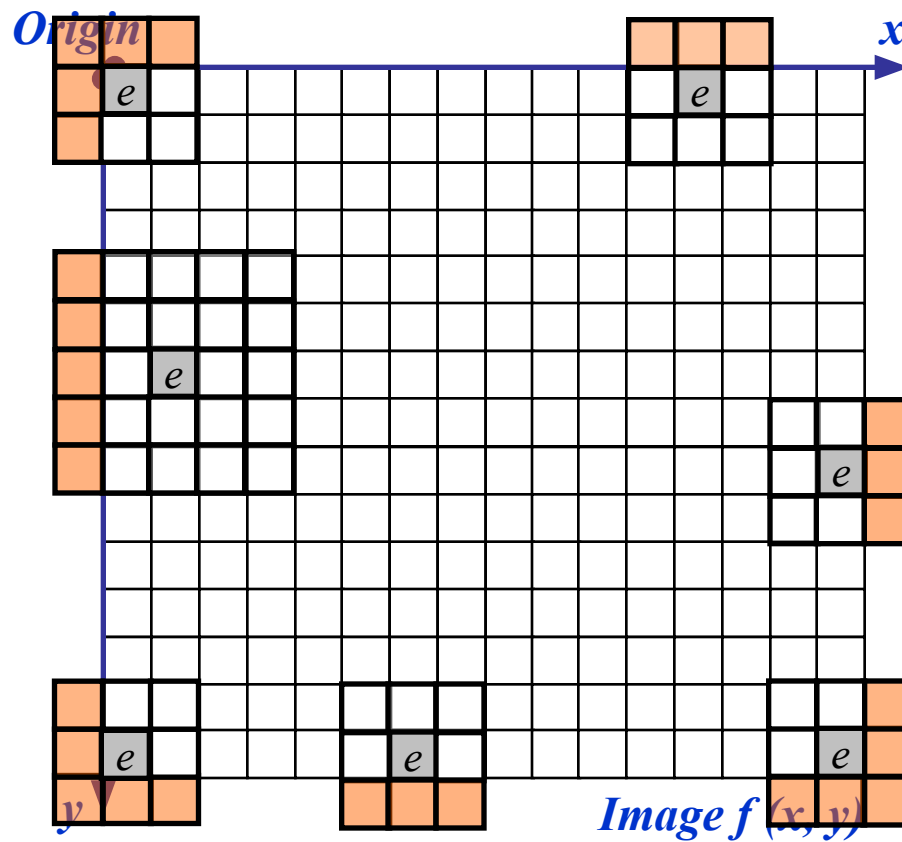
Линейная пространственная фильтрация общего вида



$$m \times n, \quad m = 2a + 1, \quad n = 2b + 1$$

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

Обработка краев изображения



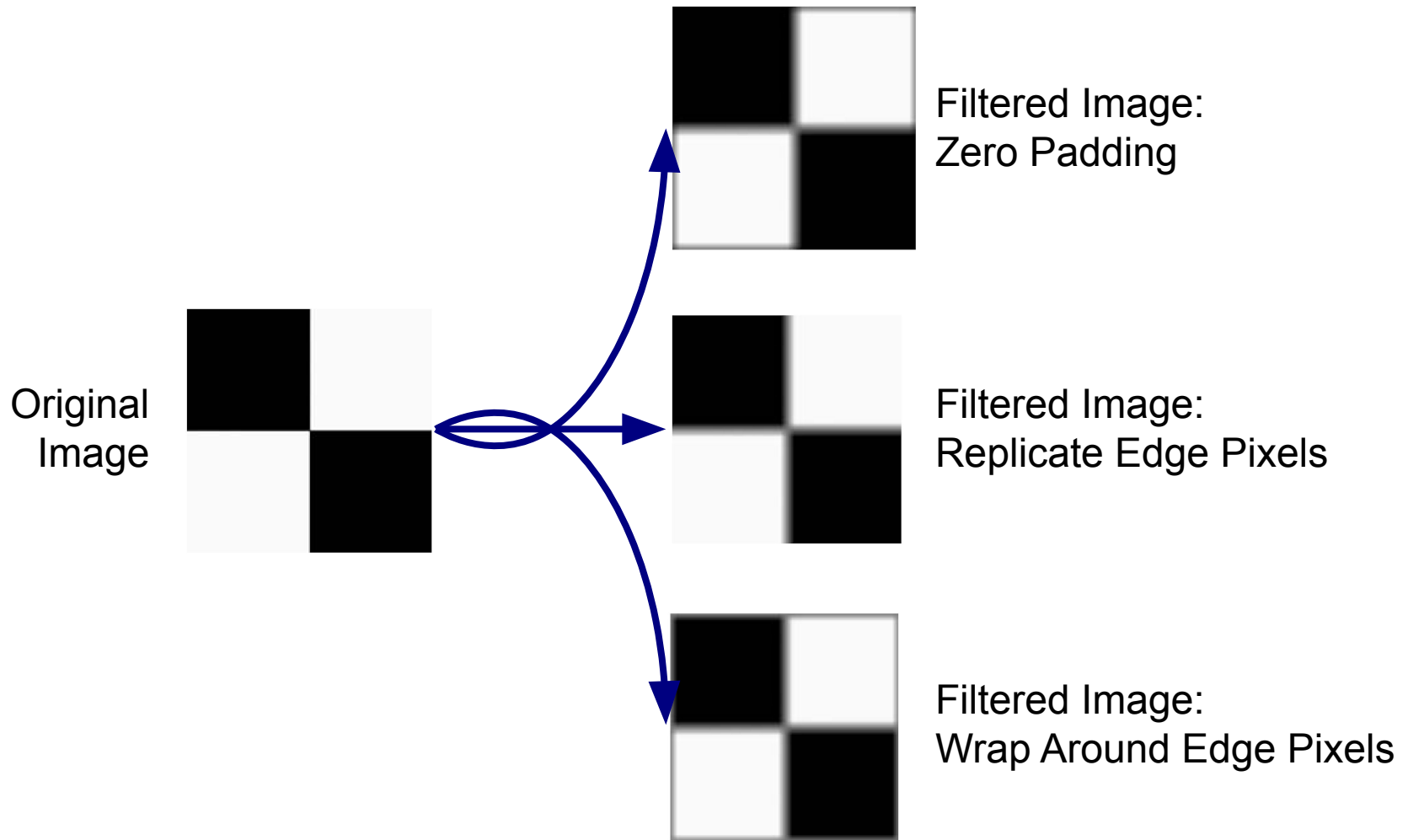
Strange Things Happen At The Edges!

(cont...)

There are a few approaches to dealing with missing edge pixels:

- Omit missing pixels
 - Only works with some filters
 - Can add extra code and slow down processing
- Pad the image
 - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image
- Allow pixels *wrap around* the image
 - Can cause some strange image artefacts

Strange Things Happen At The Edges! (cont...)

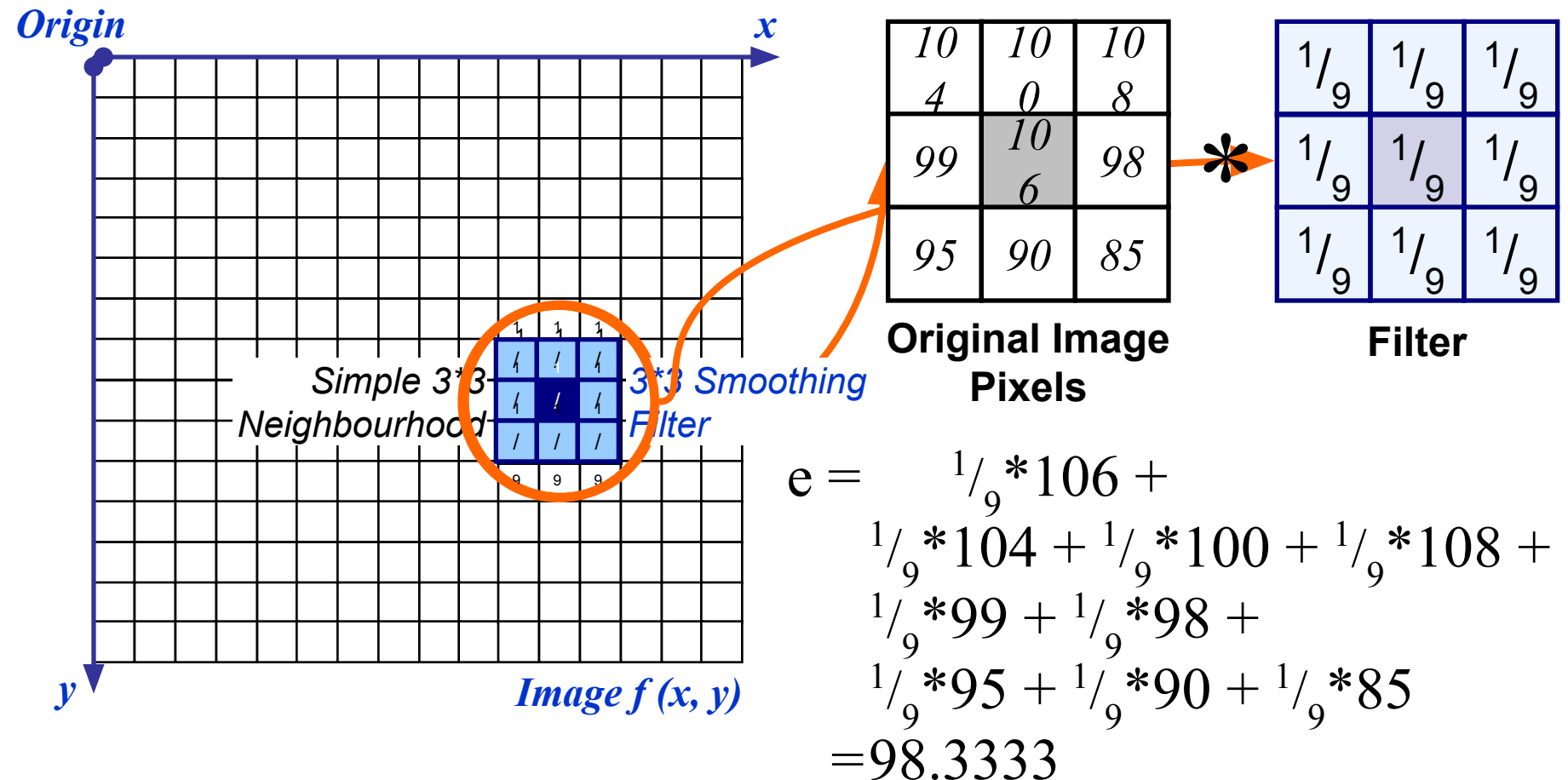


Сглаживающие пространственные фильтры

- Удаление шума
- Расфокусировка изображения

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Сглаживающие пространственные фильтры



Процесс повторяется для каждого пиксела изображения

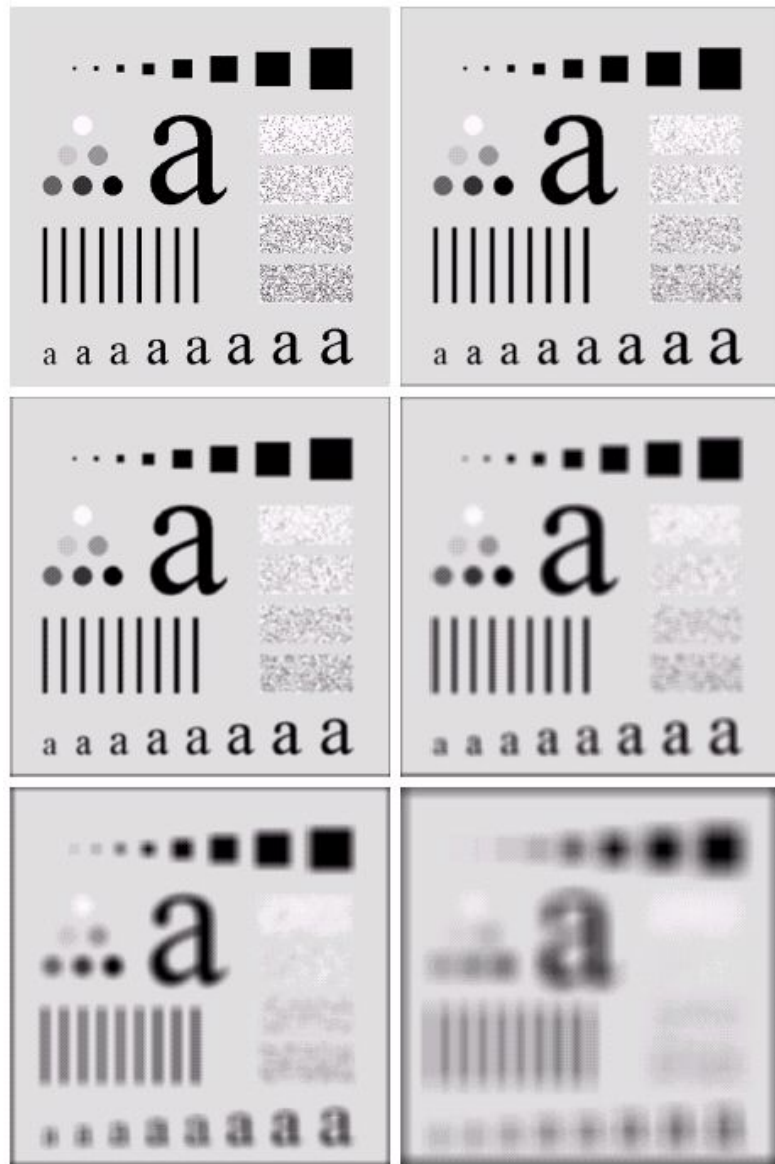
Пример сглаживания

Исходное изображение
500*500 пикселей

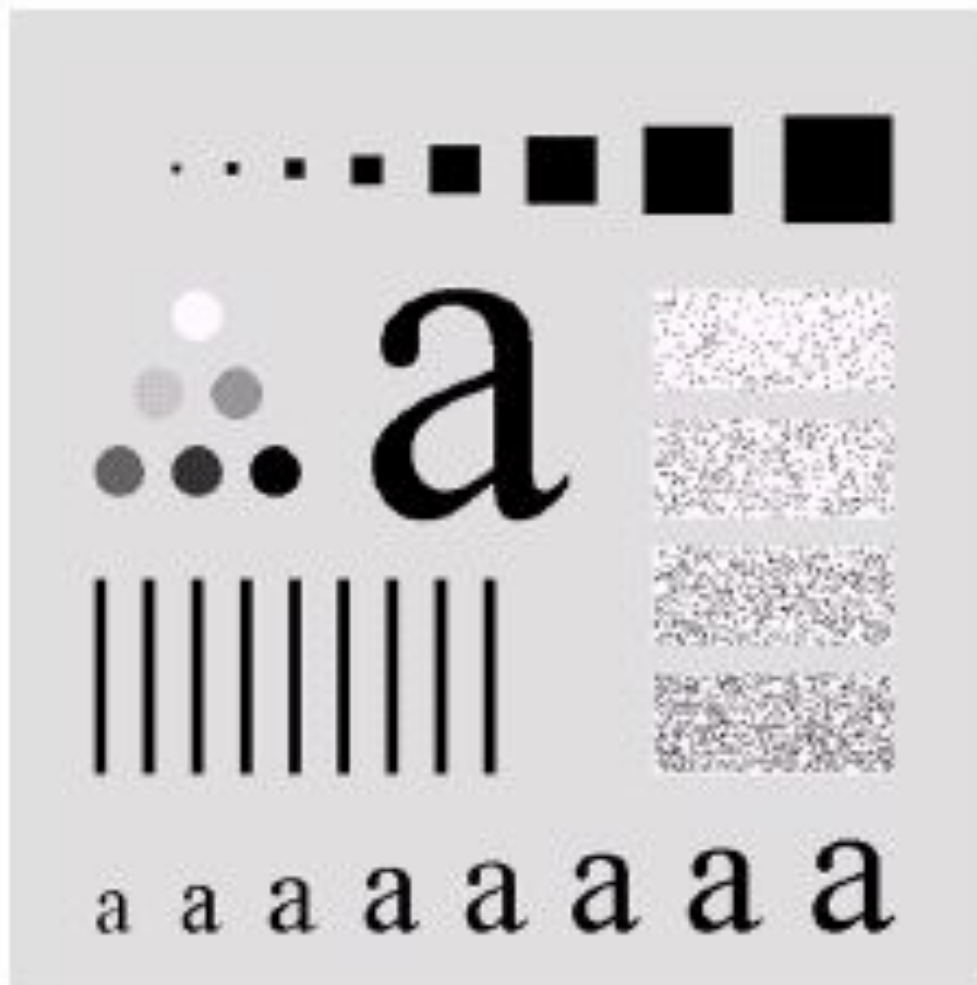
Размер фильтра

– 3, 5, 9, 15 and 35

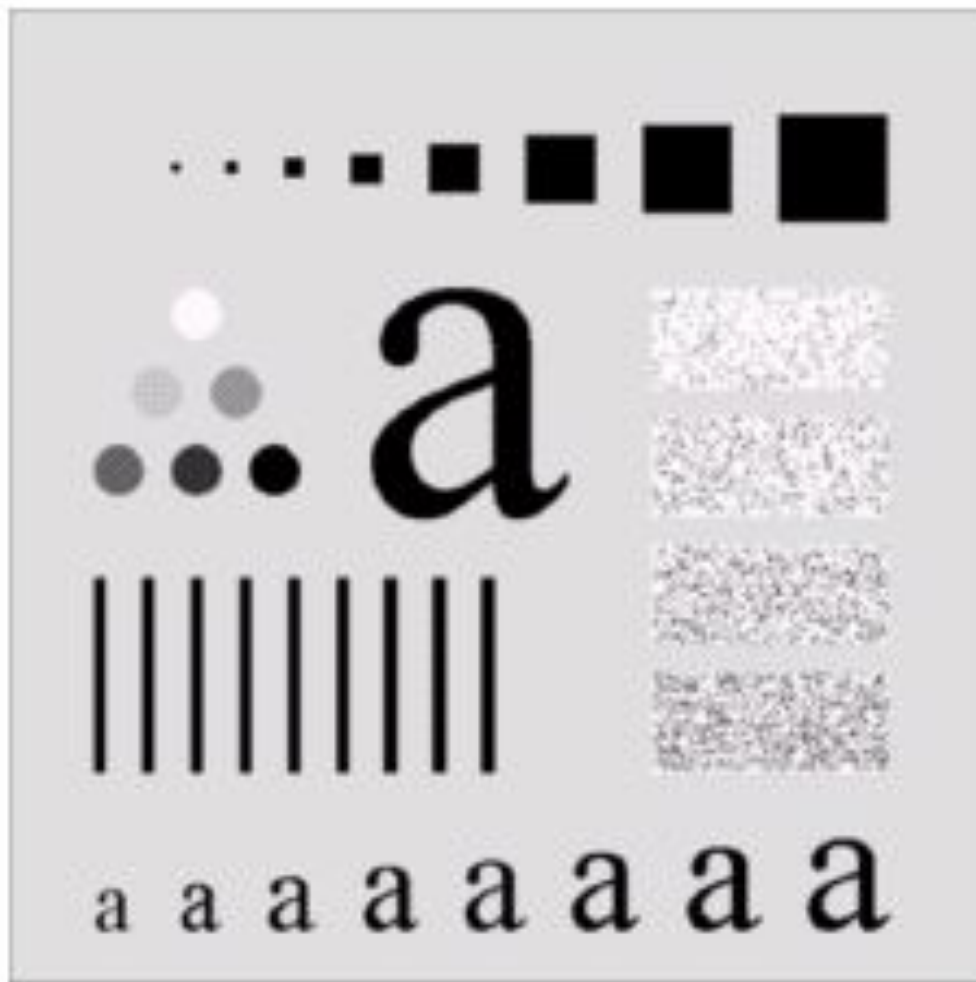
Исчезание деталей

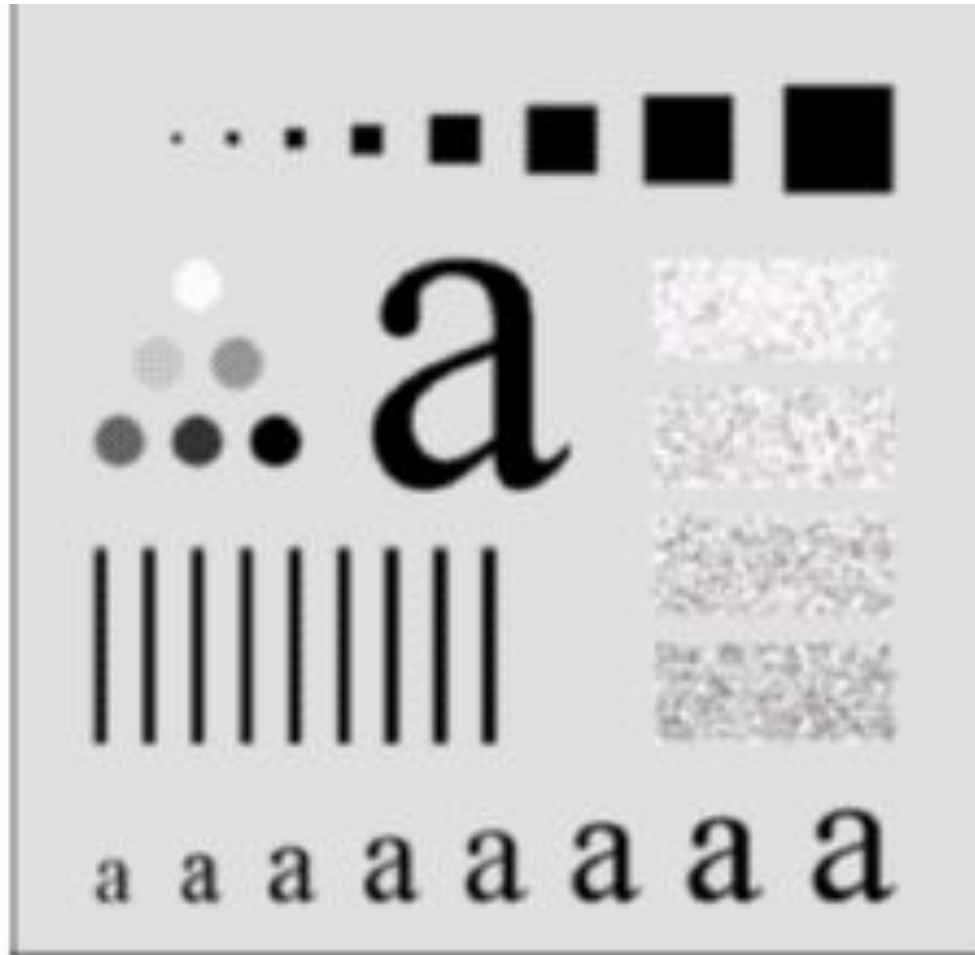


Пример сглаживания

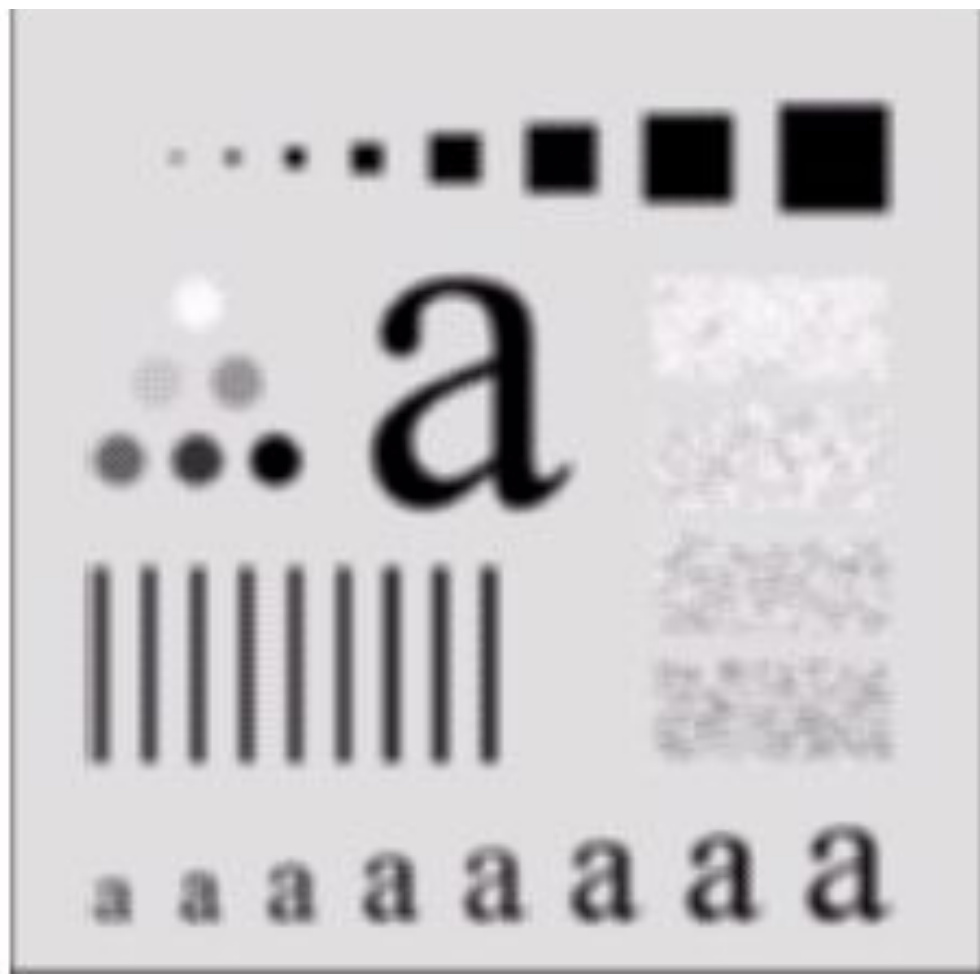


Пример сглаживания

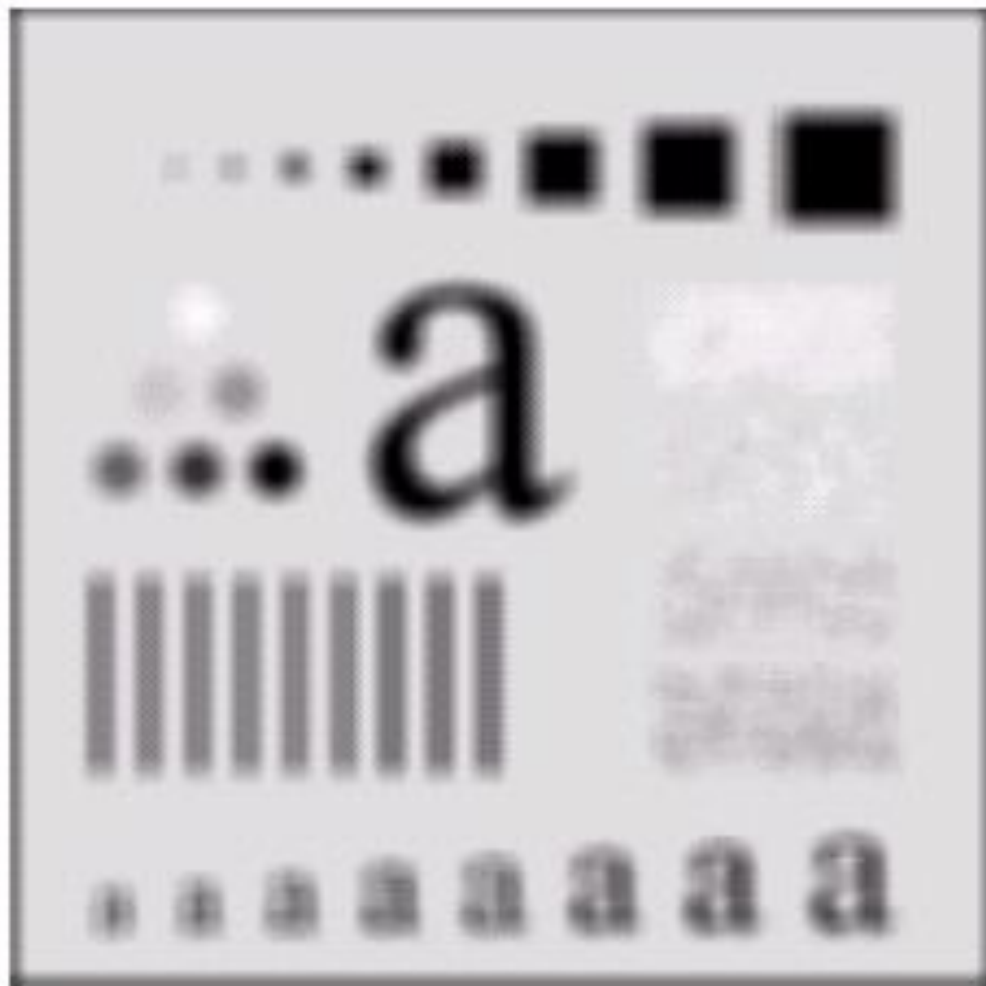




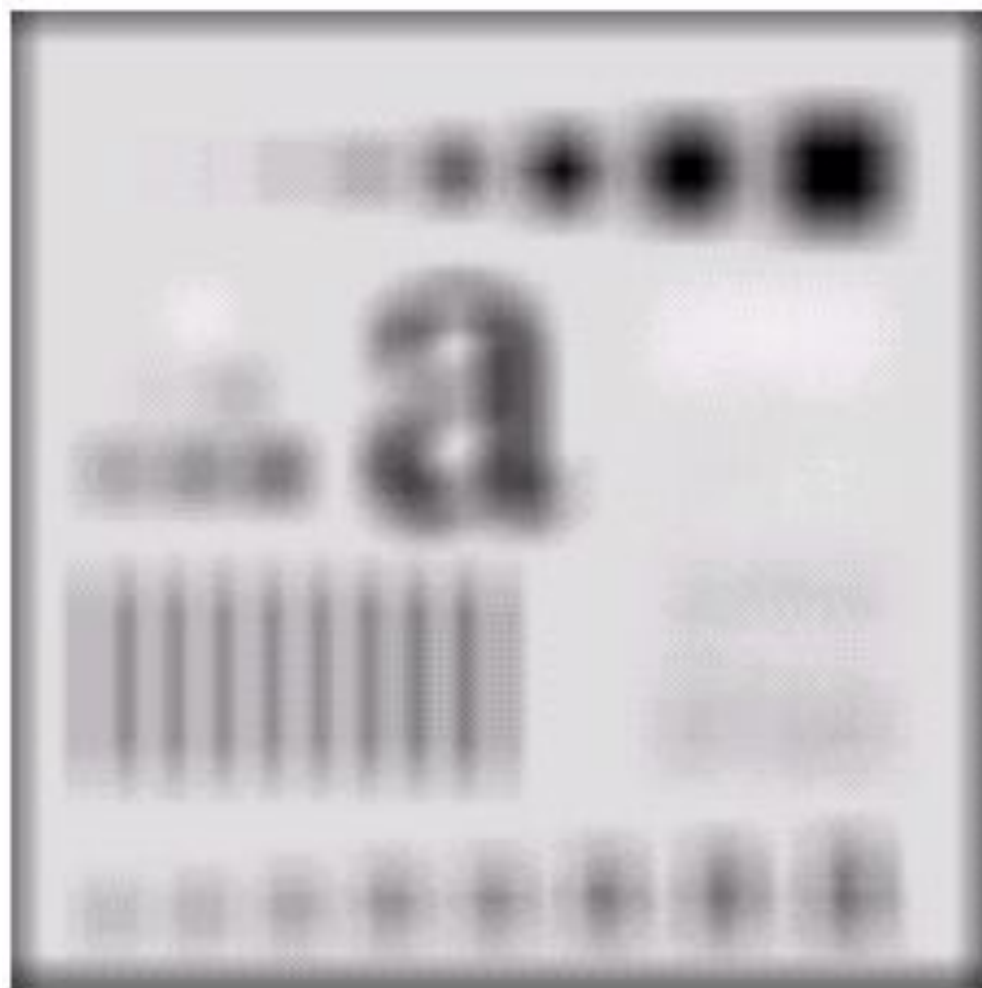
Пример сглаживания



Пример сглаживания



Пример сглаживания

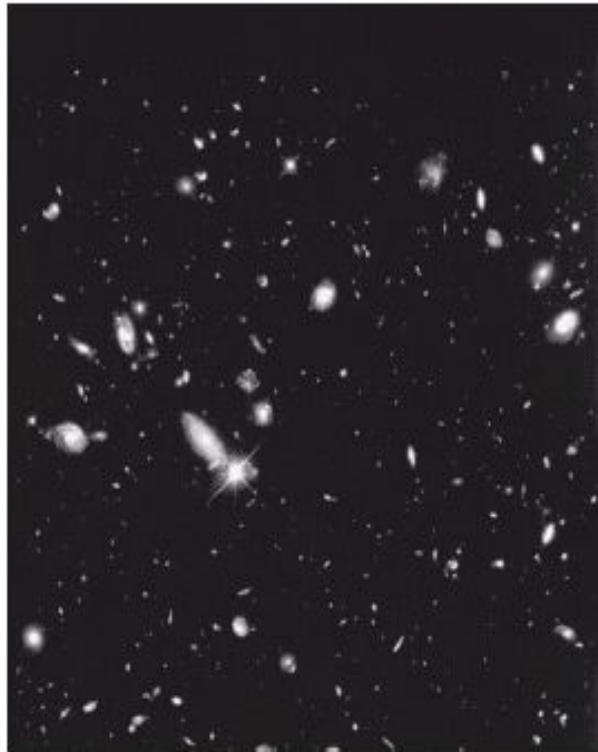


Фильтр взвешенного среднего

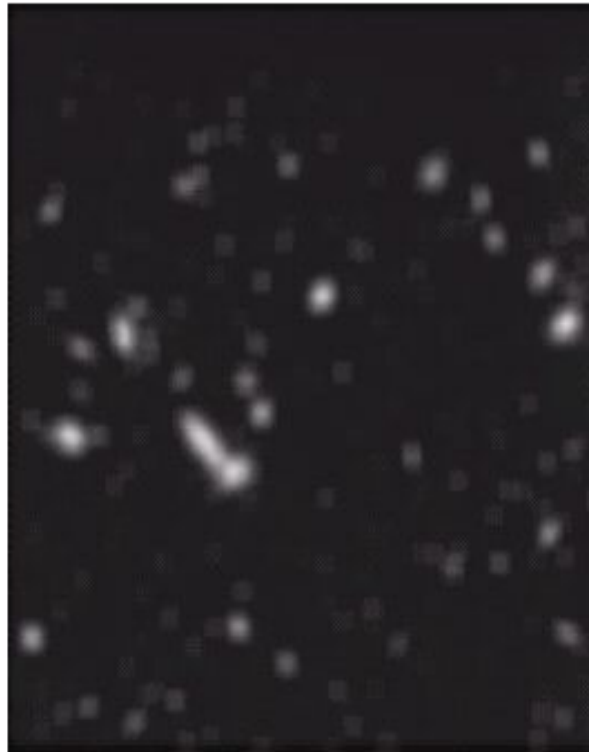
$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

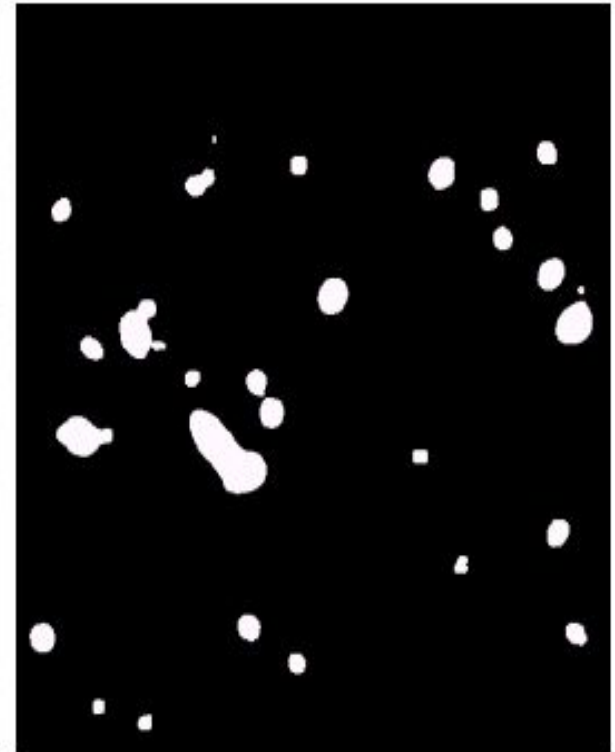
Пример сглаживания



Original Image



Smoothed Image



Thresholded Image

Важные локальные операции:

- **Min:** Set the pixel value to the minimum in the neighbourhood
- **Max:** Set the pixel value to the maximum in the neighbourhood
- **Median:** The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average

Медианный фильтр

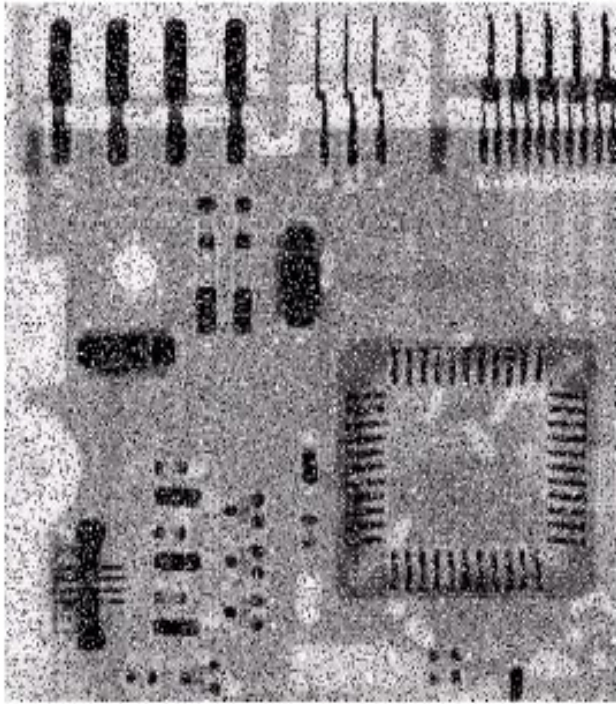
1	7	4
3	1	99
6	0	2

Mean: $(1+7+4+3+1+99+6+0+2) / 9 \approx \mathbf{14}$

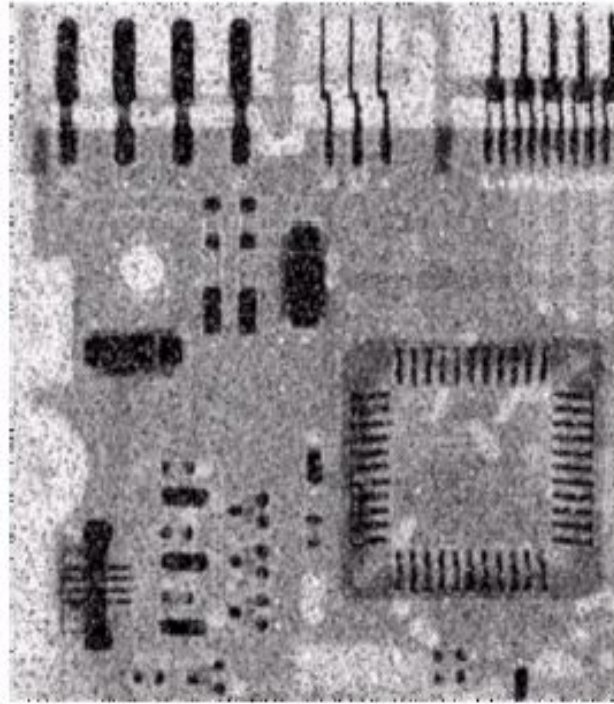
Median: $\text{middle}(0, 1, 1, 2, 3, 4, 6, 7, 99) = \mathbf{3}$

Fast-Median: $\text{middle}(\text{middle}(1,7,4), \text{middle}(3,1,99), \text{middle}(6,0,2))$
 $= \text{middle}(4, 3, 2) = \mathbf{3}$

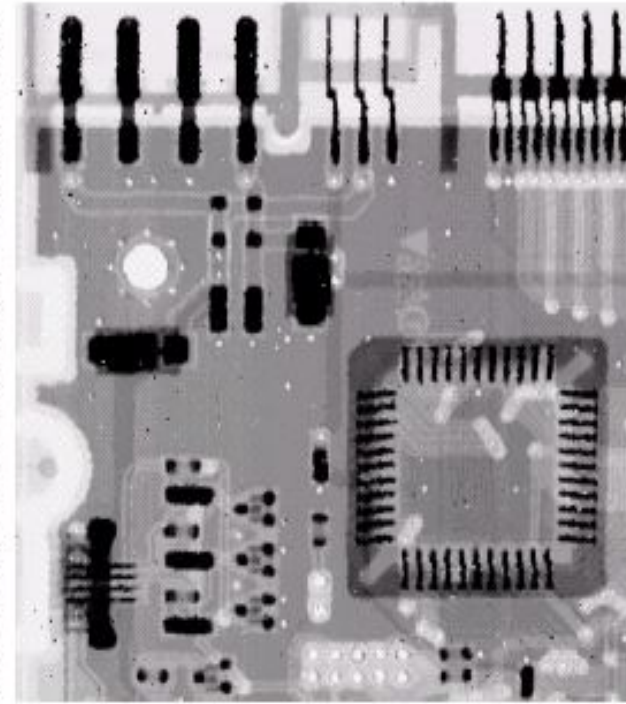
Сглаживающий против медианного



**Original Image
With Noise**

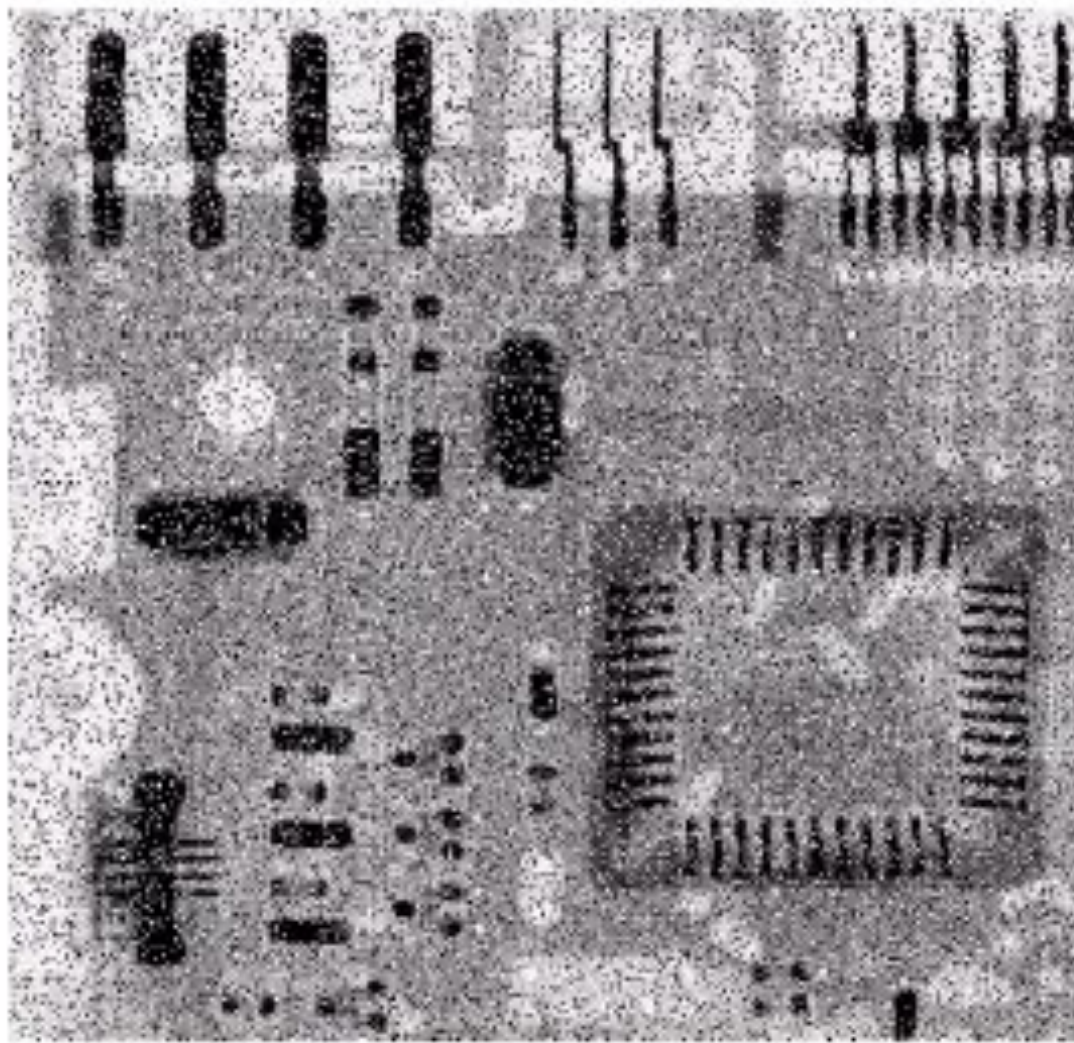


**Image After
Averaging Filter**

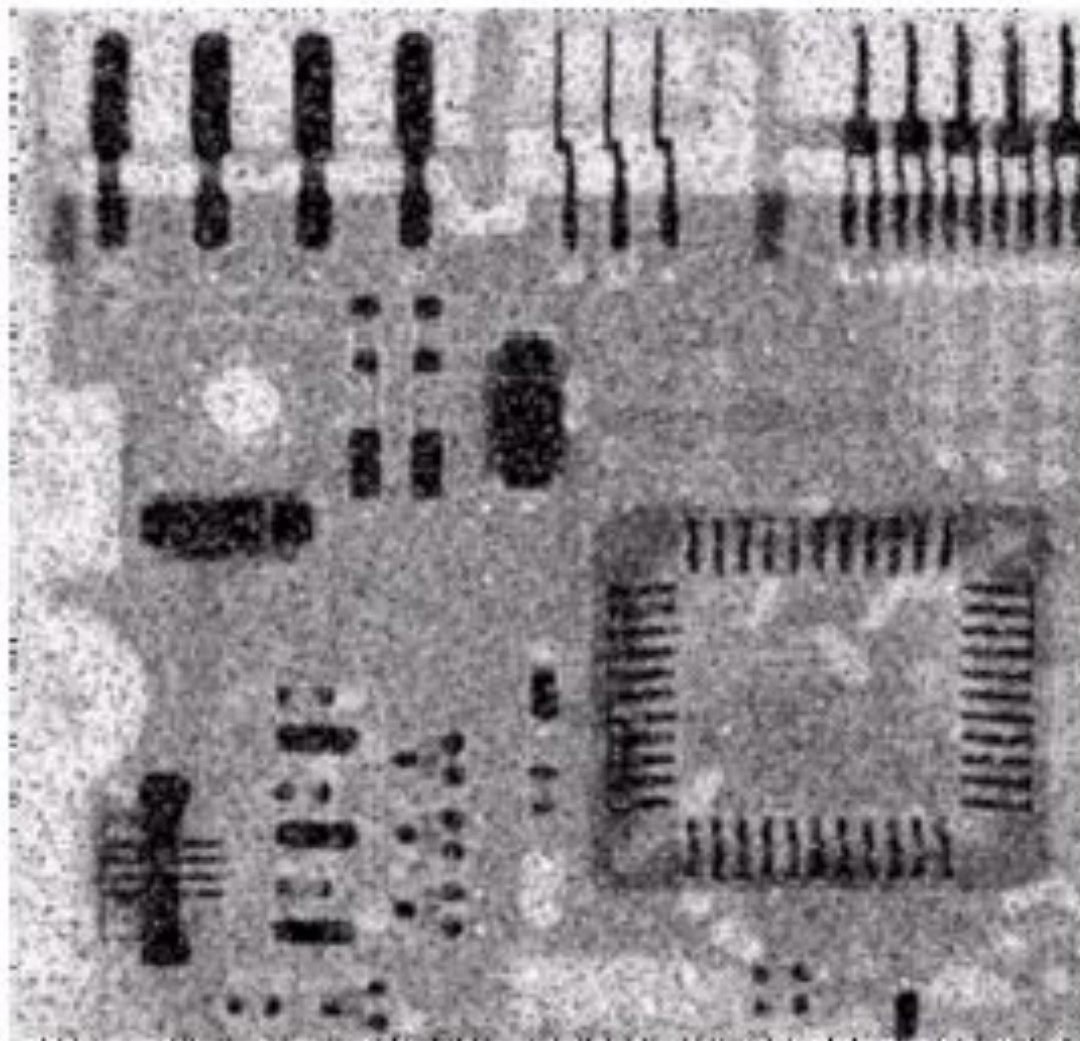


**Image After
Median Filter**

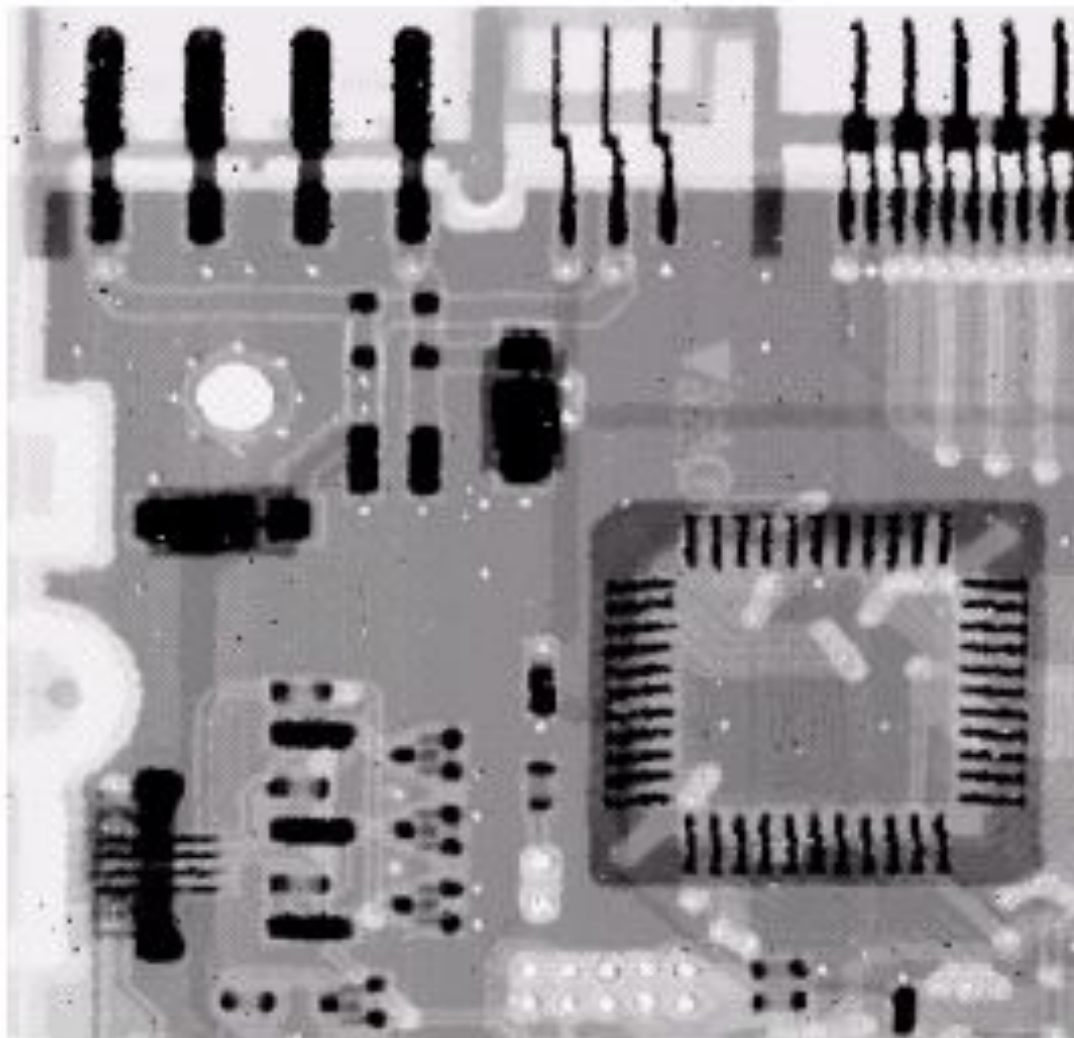
Сглаживающий против медианного



Сглаживающий против медианного



Сглаживающий против медианного



Correlation & Convolution

The filtering we have been talking about so far is referred to as *correlation* with the filter itself referred to as the *correlation kernel*

Convolution is a similar operation, with just one subtle difference

a	b	c
d	e	e
f	g	h

Original Image
Pixels

$*$

r	s	t
u	v	w
x	y	z

Filter

$$e_{processed} = v * e + z * a + y * b + x * c + w * d + u * e + t * f + s * g + r * h$$

For symmetric filters it makes no difference