Системы обработки медиаданных

Пространственная фильтрация

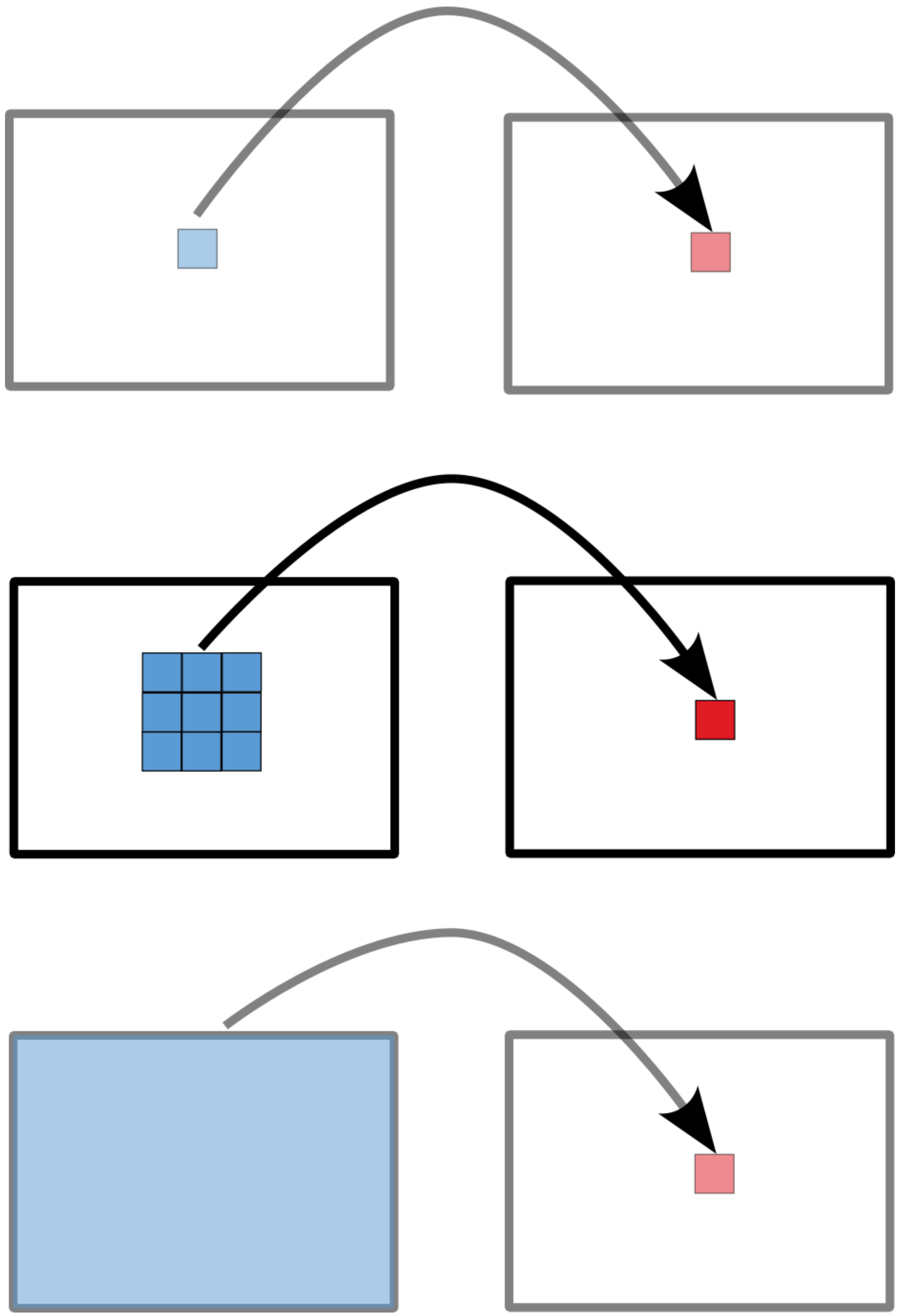
д.т.н. Вашкевич М. И.

vashkevich@bsuir.by

Белорусский государственный университет
информатики и радиоэлектроники

Кафедра электронных вычислительных средств

Типы операций обработки изображений



Точечные операции

Выходное значение в точке (x, y) зависит только от входного значения в точке (x, y).

Окрестностная обработка

Выходное значение в точке (x, y) зависит от входных значений в окрестности точки (x, y).

Глобальные операции

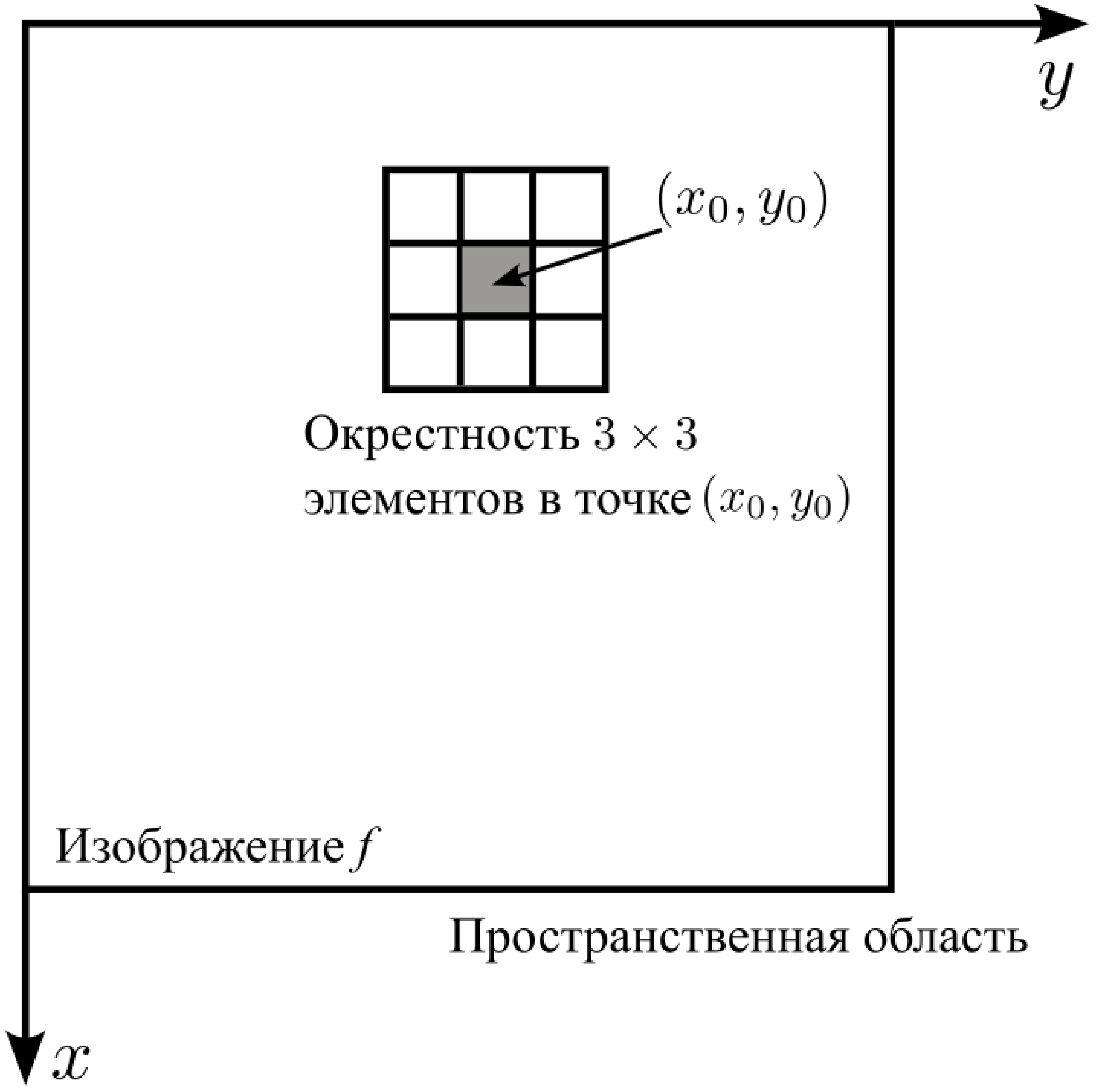
Выходное значение в точке (x, y) зависит от всех значений во входном изображении.

Линейная инвариантная к сдвигу фильтрация изображений

- ✓ В процессе обработки каждый пиксель на изображении заменяется ***линейной*** комбинацией данного пикселя и находящихся в его окрестности.

- ✓ Характер получаемого результата определяется ядром фильтра (*filter kernel*).

- ✓ Ядро фильтра скользит (сдвигается) по изображению таким образом, что ядро фильтра поочередно проходит через все позиции пикселей на изображении.



Свертка 1D сигналов

$$\underbrace{y(n)}_{\text{выход}} = \sum_{k=-\infty}^{\infty} \underbrace{h(\mathrm{k})}_{\text{фильтр}} \underbrace{x(n-k)}_{\text{вход}} = x(n) * h(n).$$

Пример: усреднение

Рассмотрим усредняющий фильтр $h = \frac{1}{3}[1, \underbrace{1}_{n=0}, 1]$, тогда свертка превратиться в

$$y(n) = \frac{1}{3}(x(n+1) + x(n) + x(n-1))$$

Т.е. выходной сигнал будет сглаженной версией входного сигнала.

Свертка 2D сигналов

$$\underbrace{\mathrm{g}(x,y)}_{\substack{\text{выходное}\\\text{изображение}}} = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} \underbrace{w(i,j)}_{\substack{\text{фильтр}\\\text{(ядро)}}} \cdot \underbrace{\mathrm{f}(x-i,y-j)}_{\substack{\text{входное}\\\text{изображение}}},$$

Обычно фильтр $w(i,j)$ имеет ненулевые значения только для $-1 \le i,j \le 1$, тогда

$$\mathrm{g}(x,y) = \sum_{\mathrm{i},j=-1}^{1} w(i,j) \cdot \mathrm{f}(x-i,y-j).$$

Для данного примера фильтр может быть представлен матрицей 3x3.

Свертка vs корреляция

Обратите внимание на минус

Определение свертки для 2D сигналов

$$g(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} w(i,j) \cdot f(x-i, y-j).$$

Минуса нет

Определение корреляции для 2D сигналов

$$g(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} w(i,j) \cdot f(x+i, y+j).$$

- ✓ В большинстве случаев это не играет роли, поскольку ядро фильтра имеет симметричный вид.
- ✓ Отличие между сверткой и корреляцией важно в контексте обработки изображений в частотной области.

Однородный фильтр (box filter)

- ✓ Известен, как 2D прямоугольный фильтр.

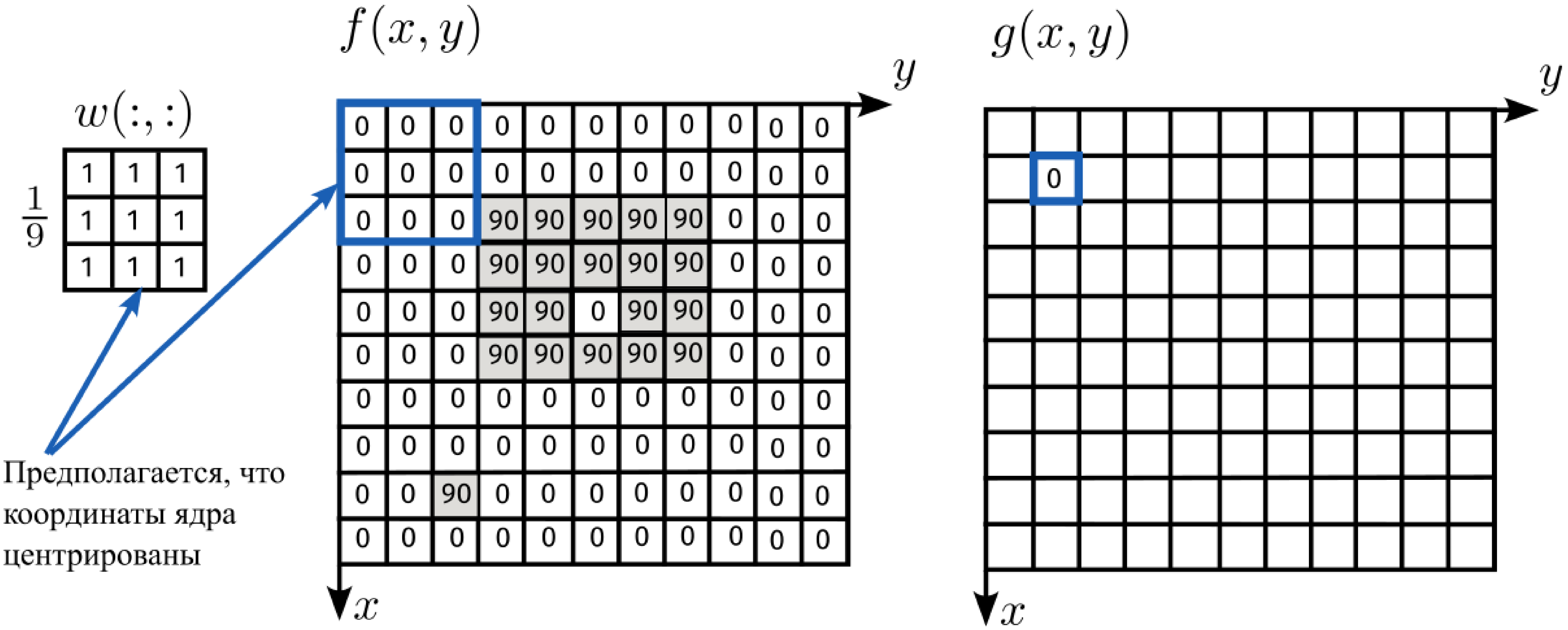
- ✓ Иногда называют ***усредняющий фильтром***, поскольку каждый пиксель на изображении заменяется локальным усредненным значением.

- ✓ Применение фильтра приводит к появлению эффекта «размытия»

- ✓ Ядро фильтра

$$\mathbf{w} = [w(i,j)] = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Работа усредняющего фильтра



1	1	1
1	1	1
1	1	1

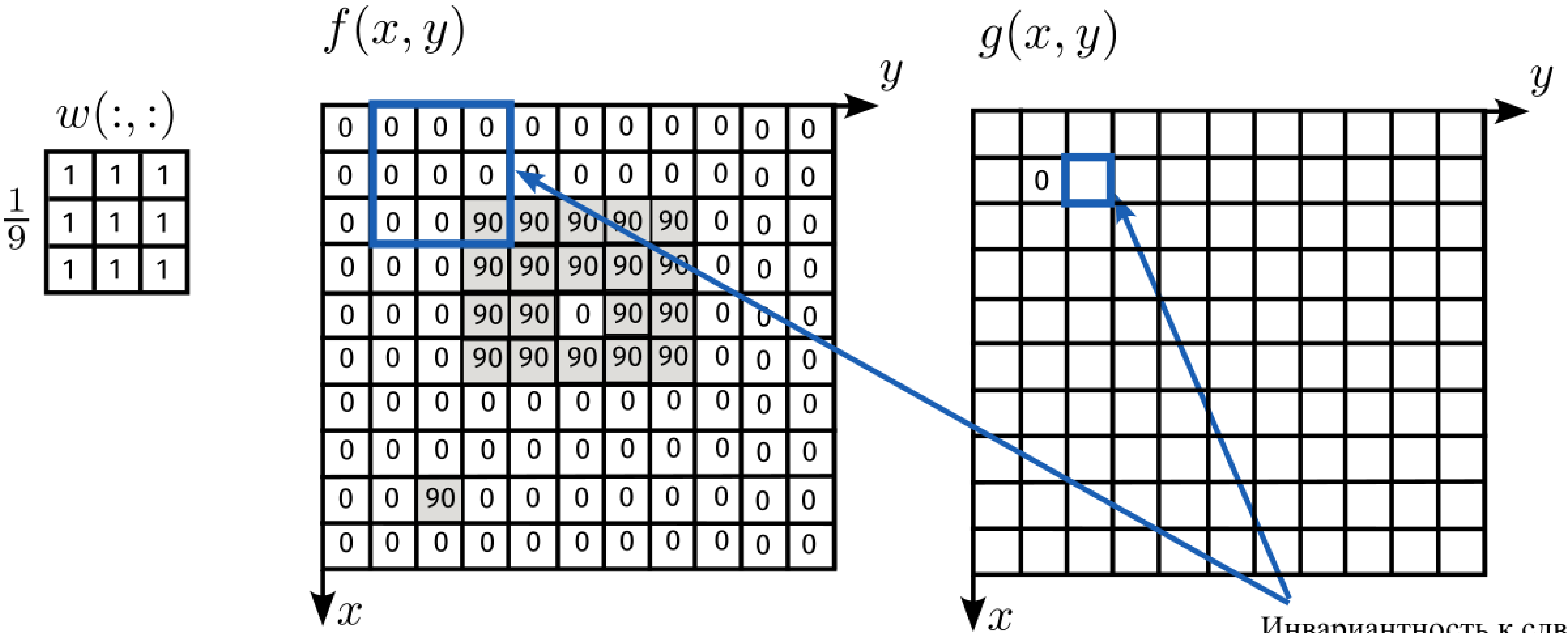
Предполагается, что координаты ядра центрированы

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

	0									

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



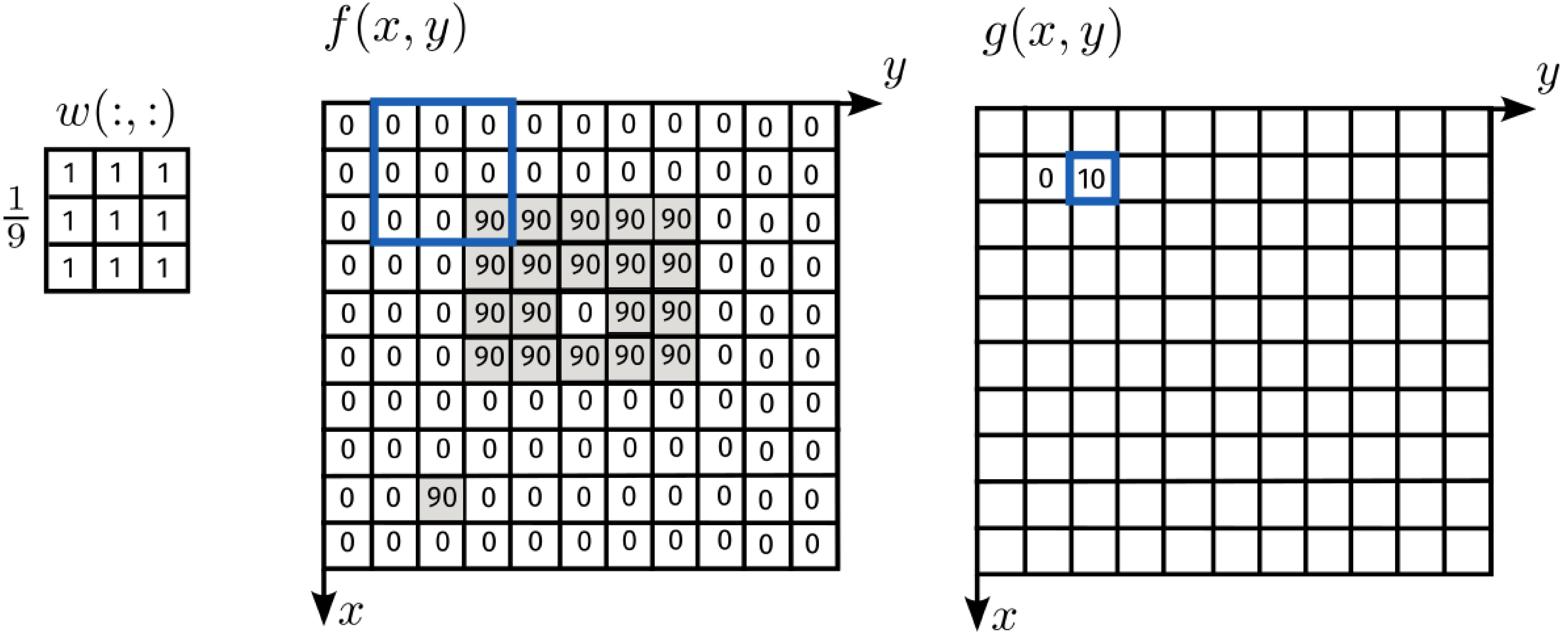
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Инвариантность к сдвигу:
поскольку сдвигается пиксель
сдвигается и ядро

Работа усредняющего фильтра



$w(:,:)$ = $\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

$f(x,y)$

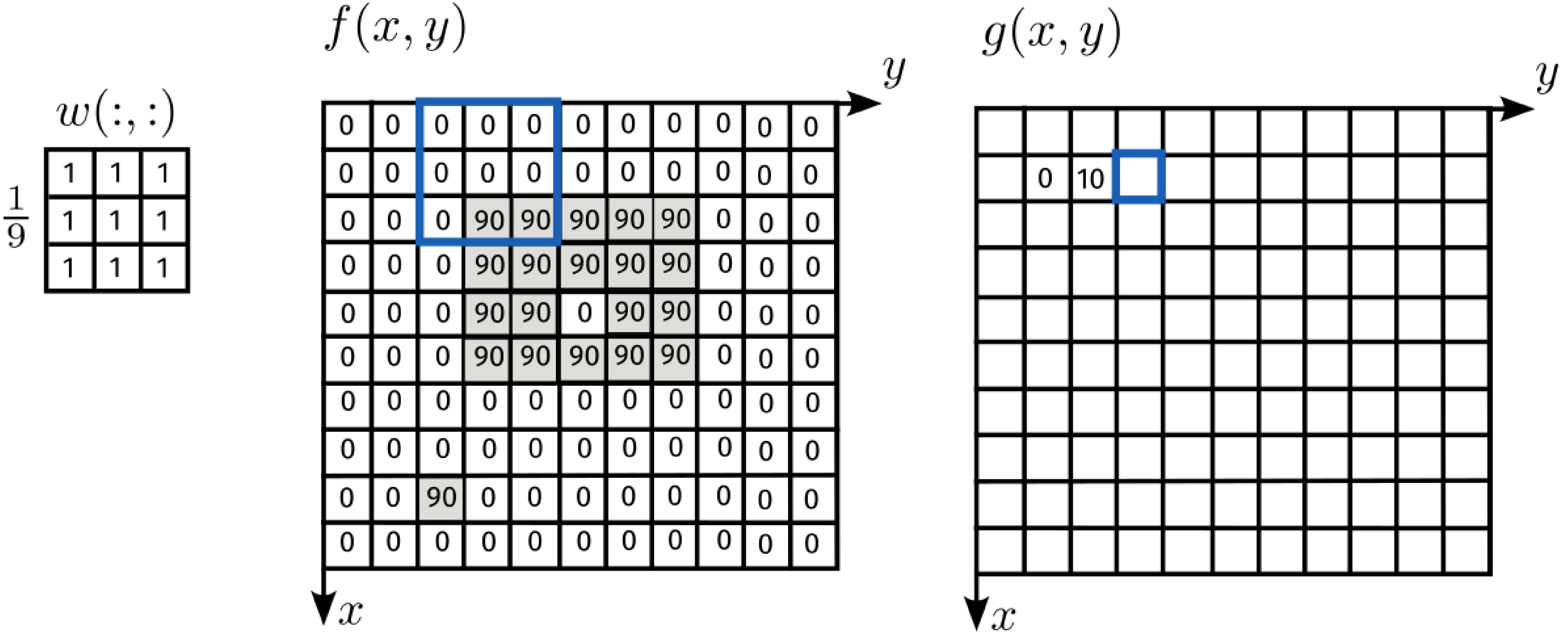
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10								

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



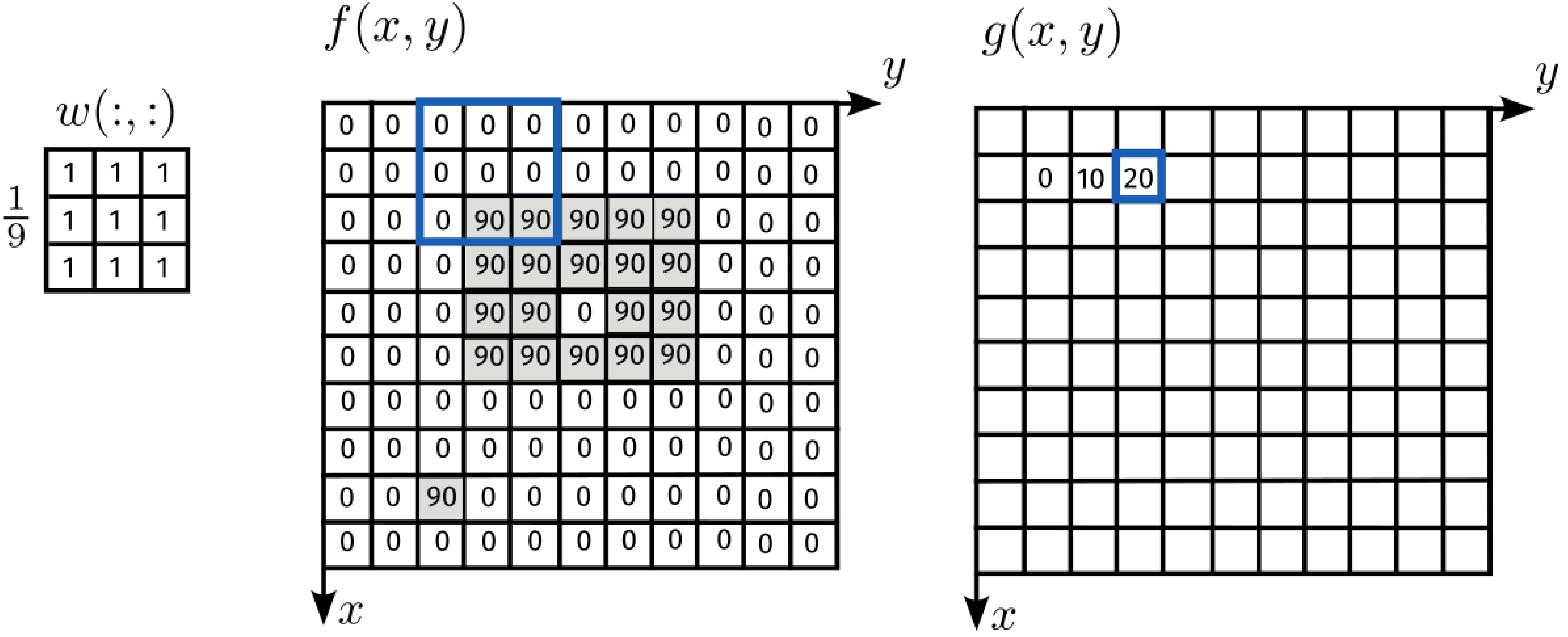
$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

	0	10								

$$f(x,y)=\sum_{i,j} w(i,j)f(x+i,y+j)$$

Работа усредняющего фильтра



$$w(:,:) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f(x,y)$

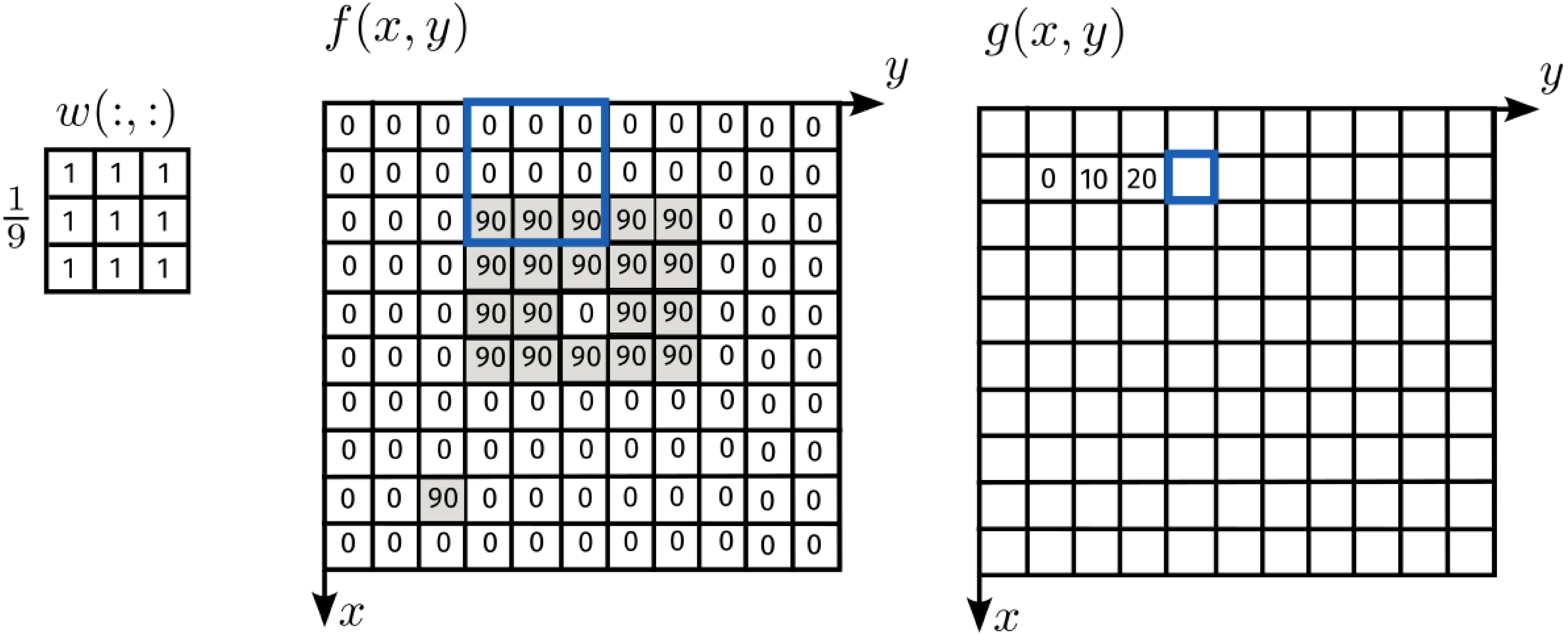
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20							

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

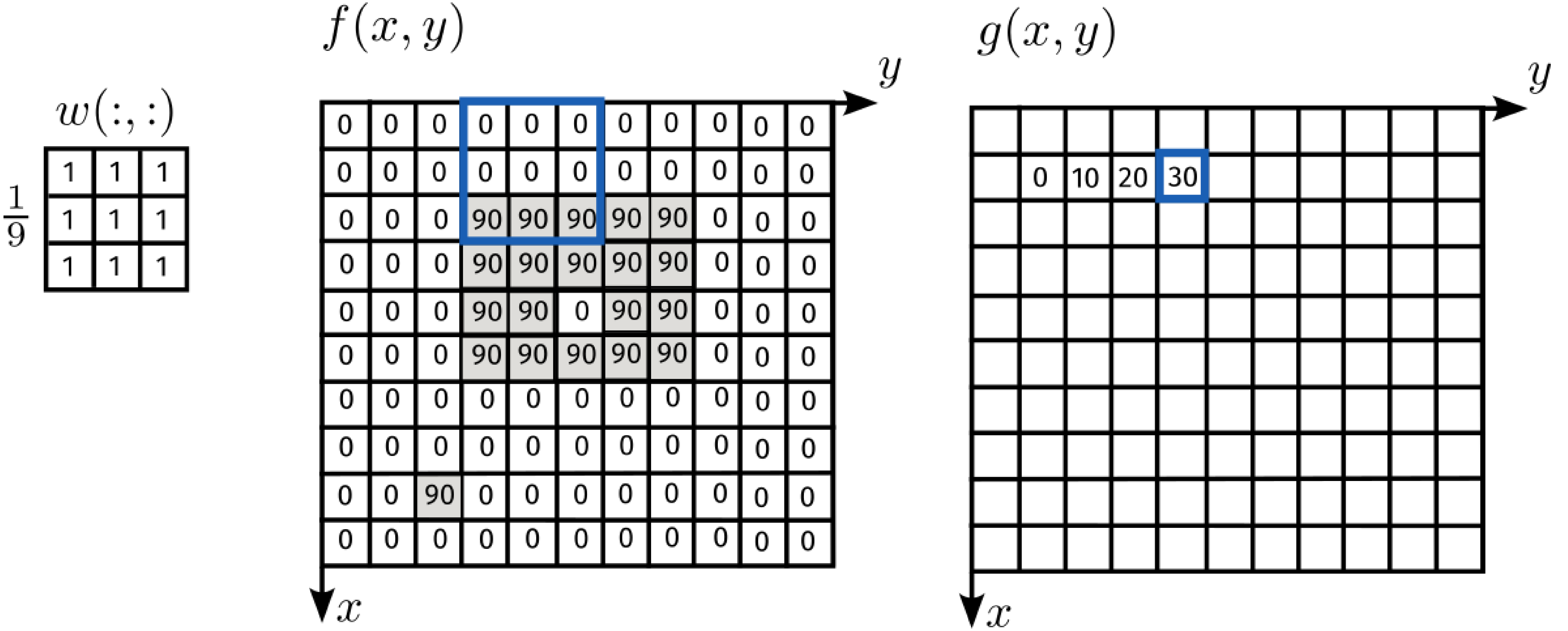
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20							

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

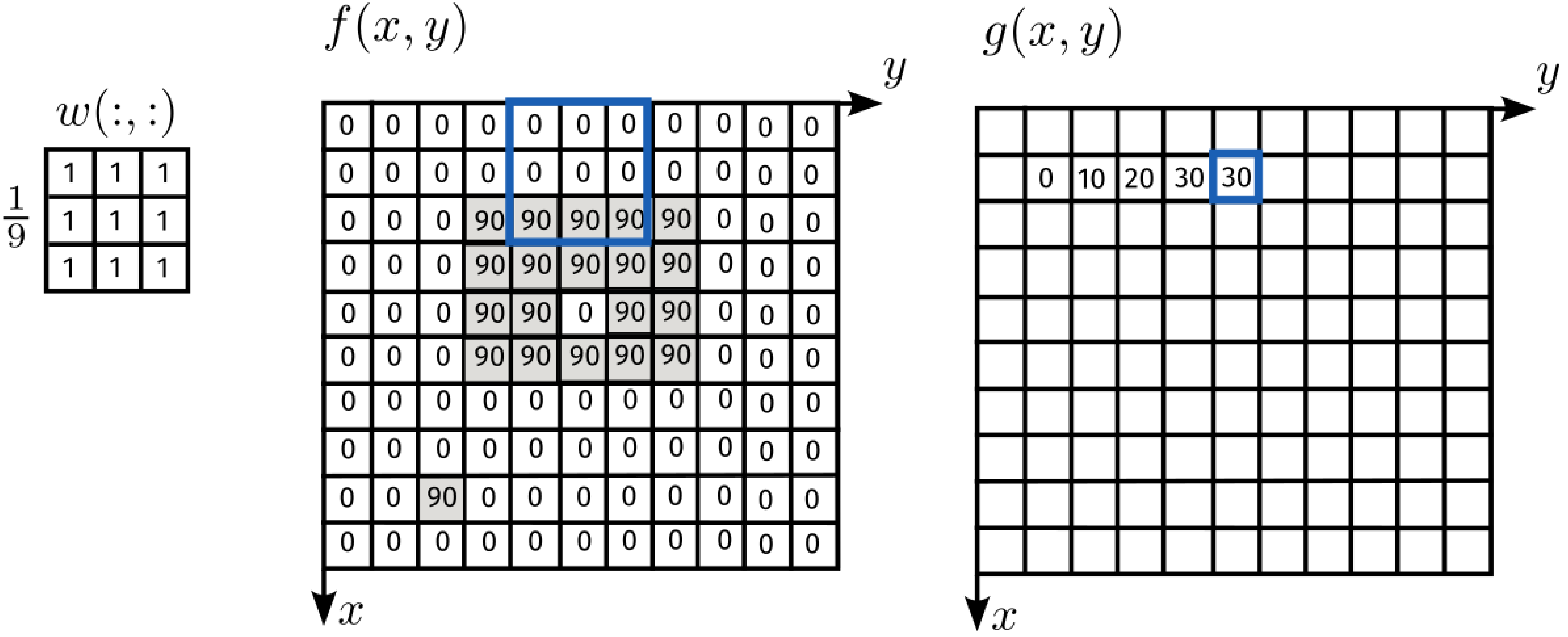
$f(x,y)$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30						

Работа усредняющего фильтра



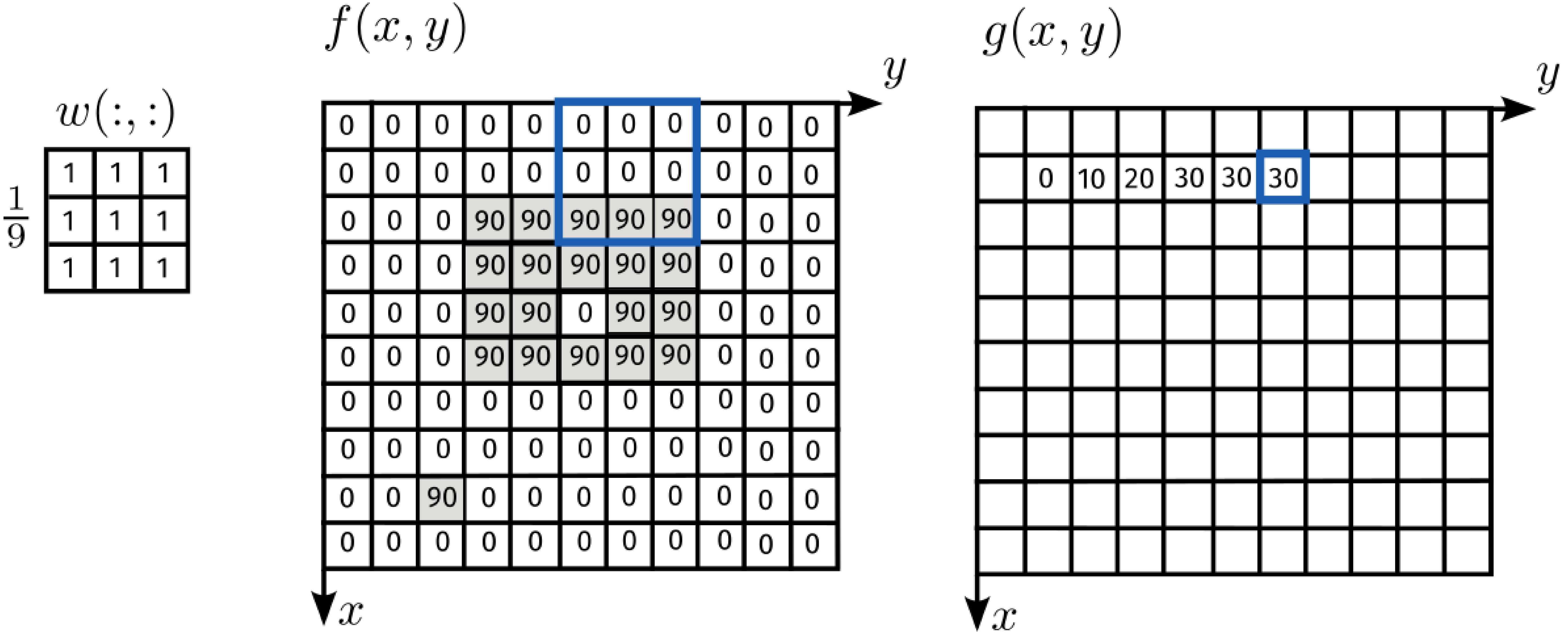
1	1	1
1	1	1
1	1	1

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30					

$$f(x,y)=\sum_{i,j} w(i,j) f(x+i,y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

$f(x,y)$

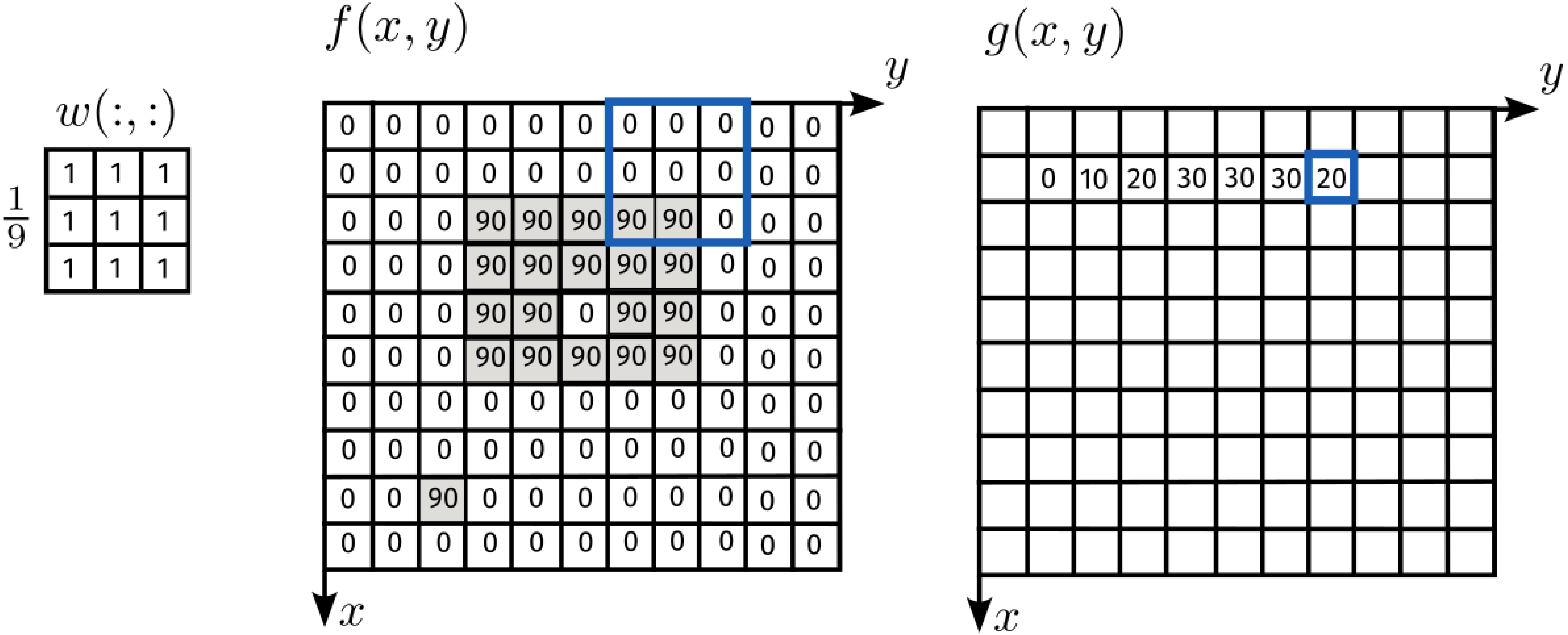
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30				

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

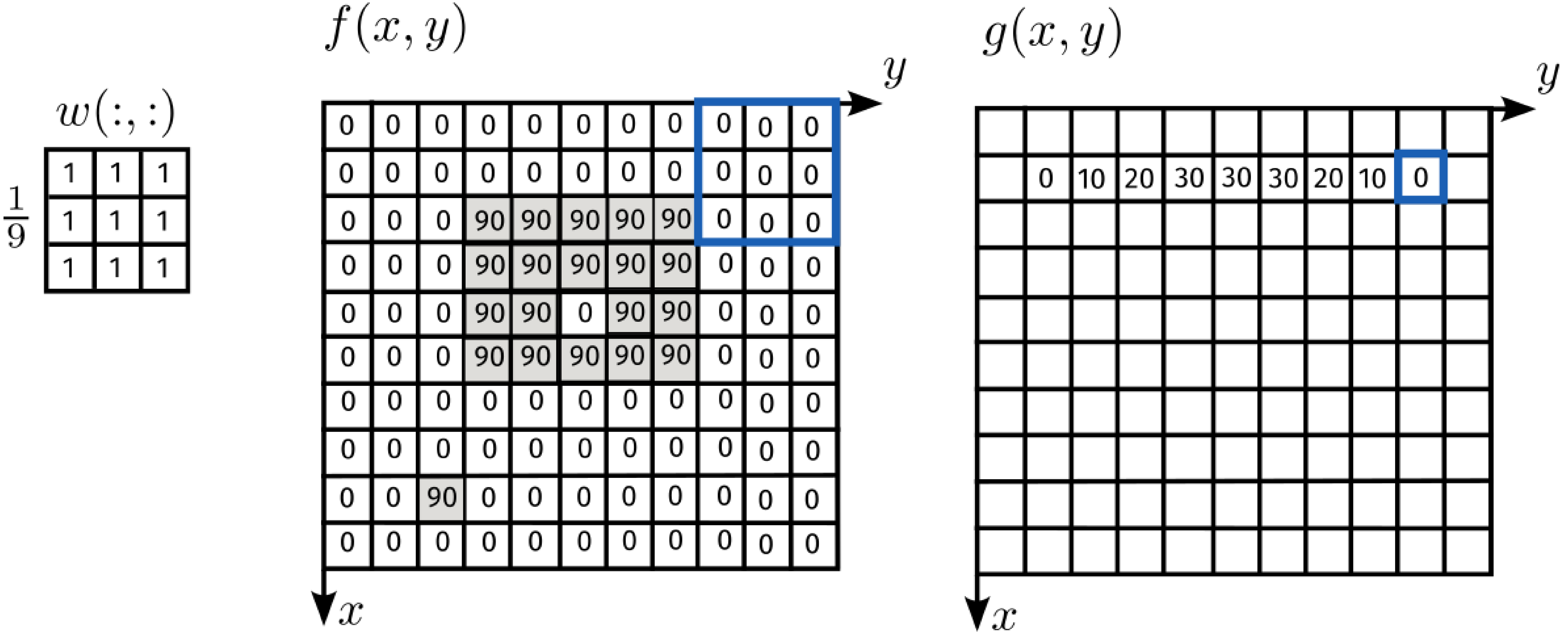
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20			

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

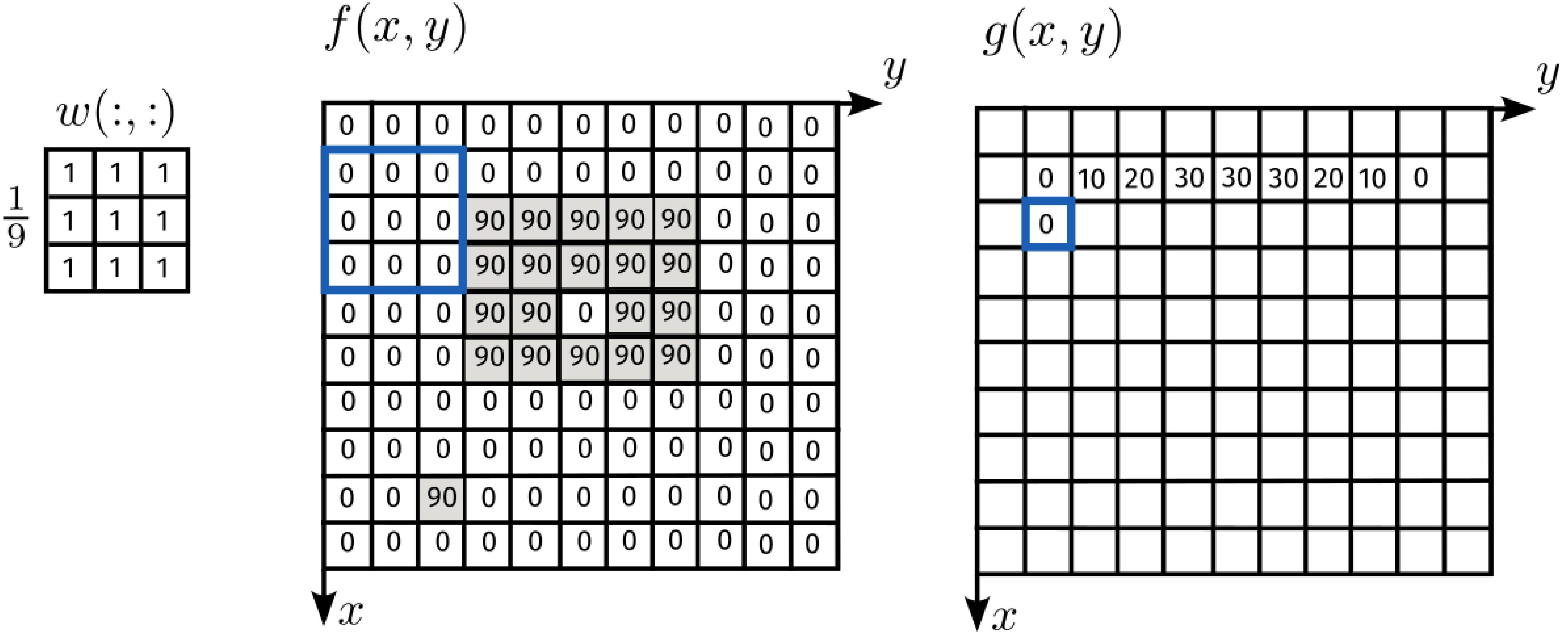
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20	10	0	

$$f(x,y)=\sum_{i,j} w(i,j)f(x+i,y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

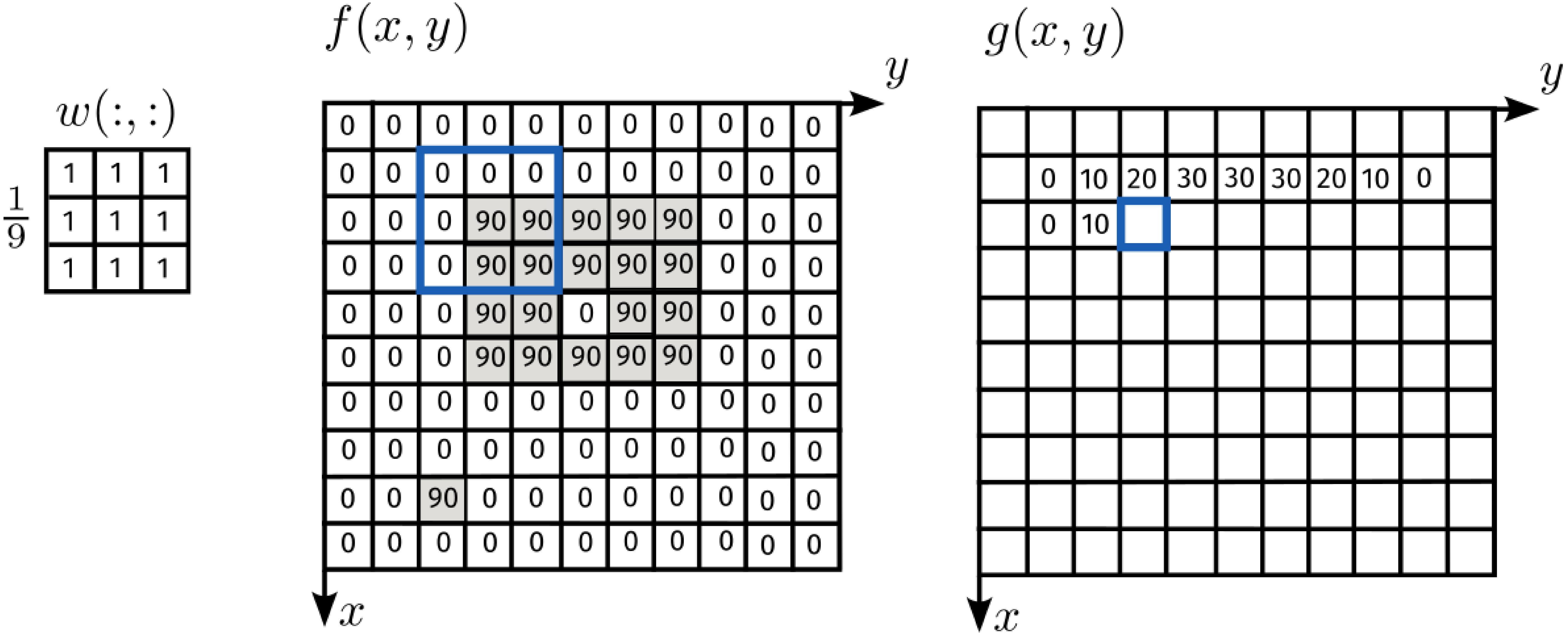
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20	10	0	
	0									

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра

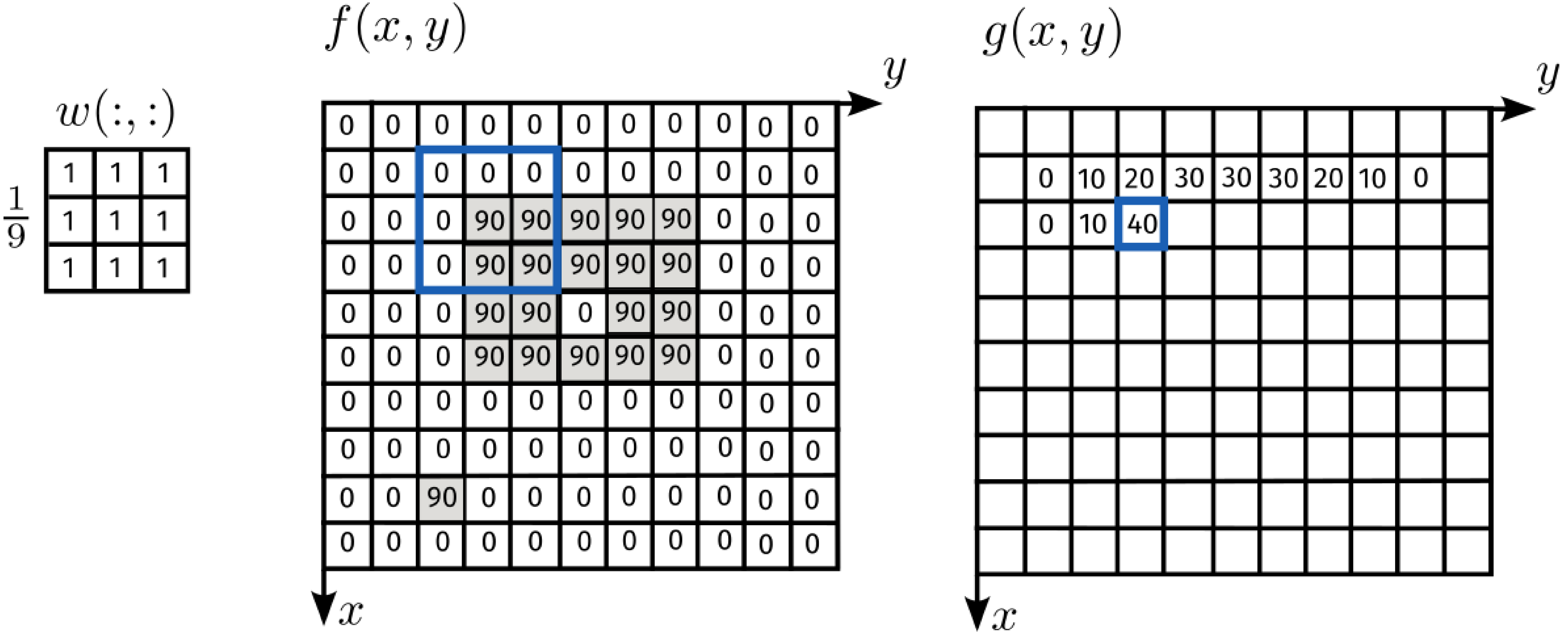


$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	0	
	0	10								

Работа усредняющего фильтра



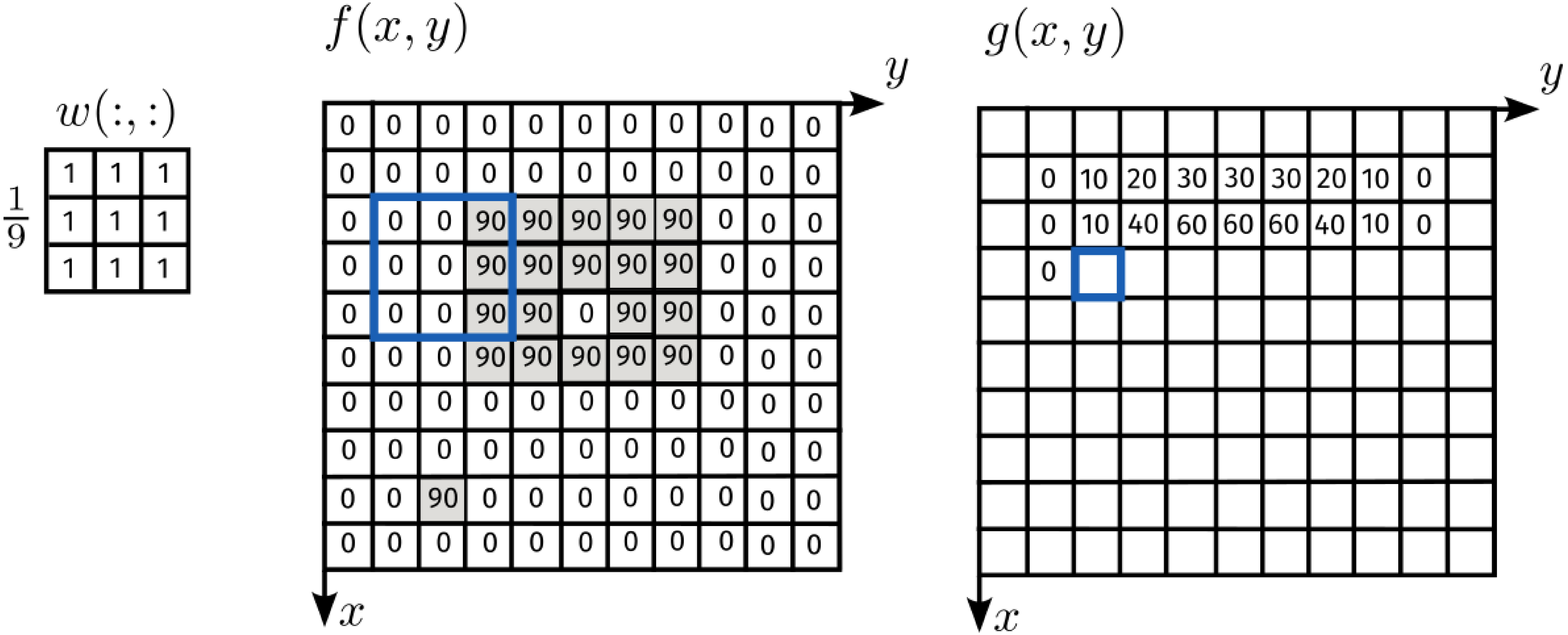
1	1	1
1	1	1
1	1	1

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	0	
	0	10	40							

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$$w(:,:) = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f(x,y)$

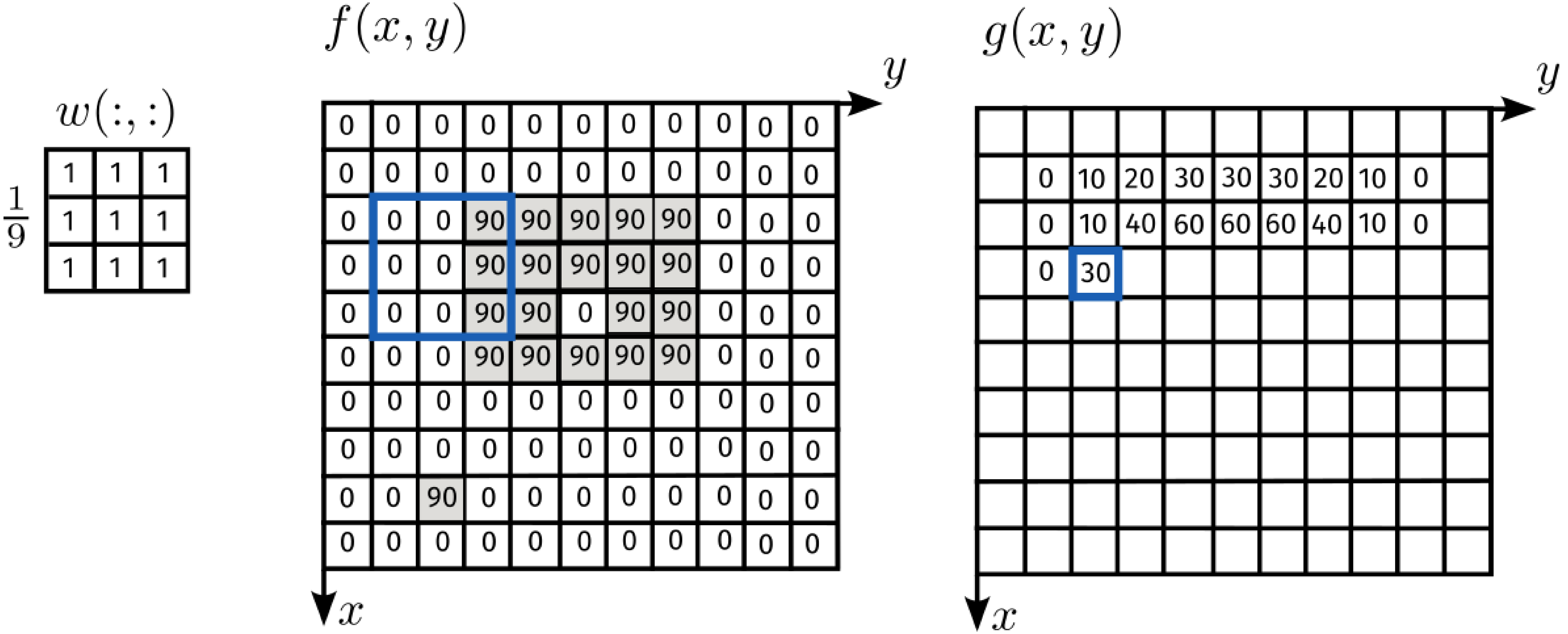
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20	10	0	
	0	10	40	60	60	60	40	10	0	
	0									

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

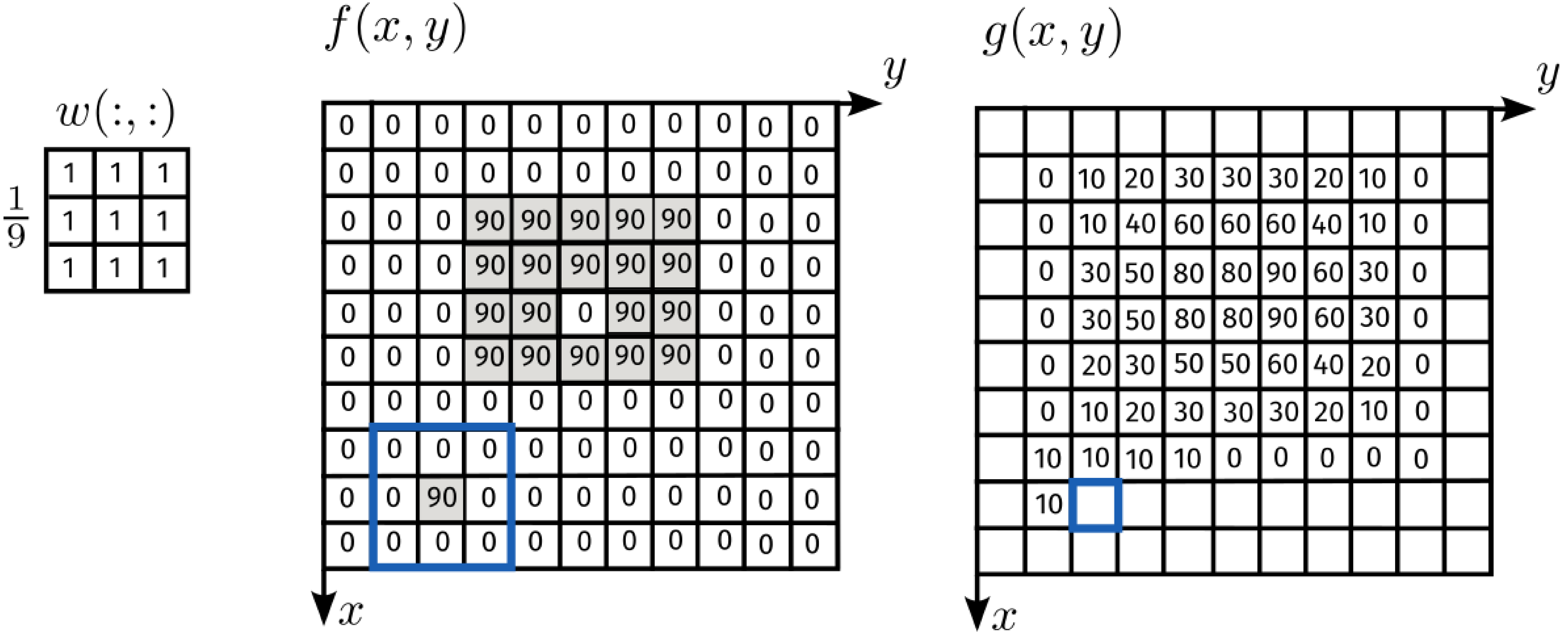
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20	10	0	
	0	10	40	60	60	60	40	10	0	
	0	30								

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра



$$w(:,:) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f(x, y)$

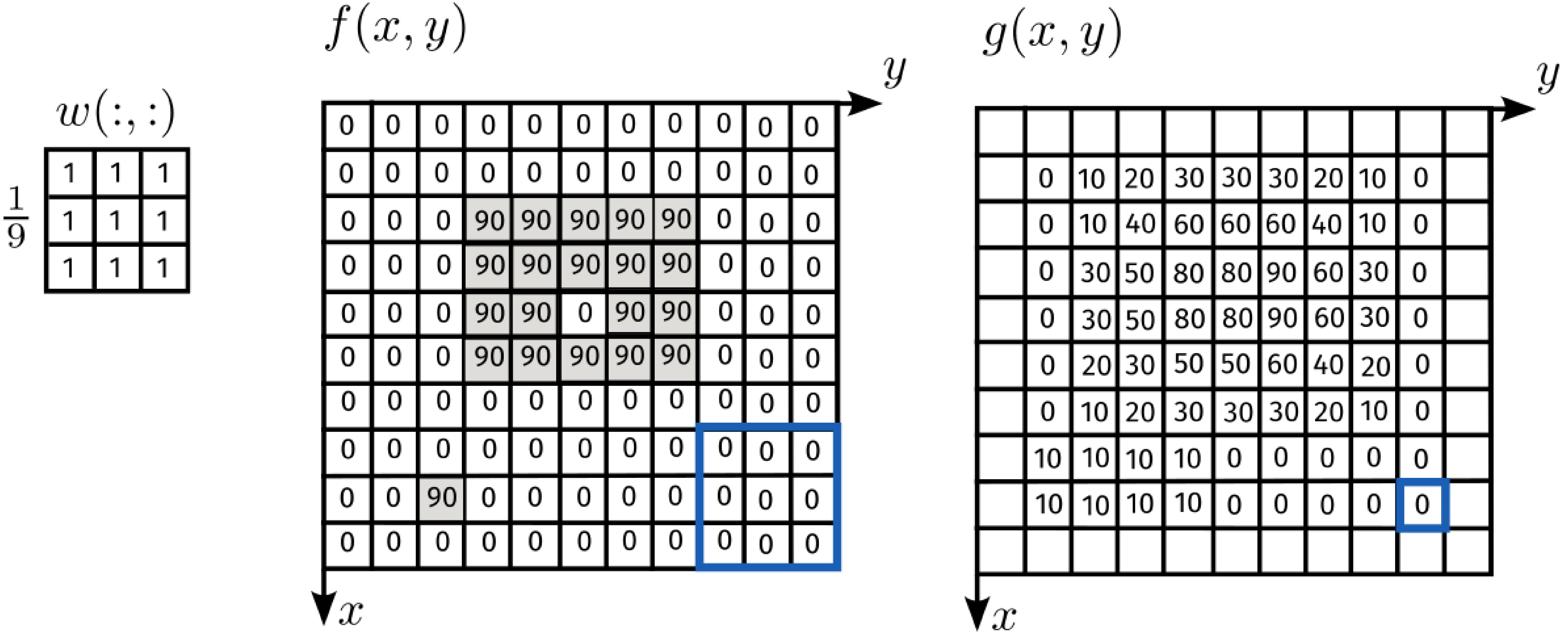
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x, y)$

	0	10	20	30	30	30	20	10	0	
	0	10	40	60	60	60	40	10	0	
	0	30	50	80	80	90	60	30	0	
	0	30	50	80	80	90	60	30	0	
	0	20	30	50	50	60	40	20	0	
	0	10	20	30	30	30	20	10	0	
	10	10	10	10	0	0	0	0	0	
	10									

$$f(x, y) = \sum_{i,j} w(i, j) f(x + i, y + j)$$

Работа усредняющего фильтра



$w(:,:)$, $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(x,y)$

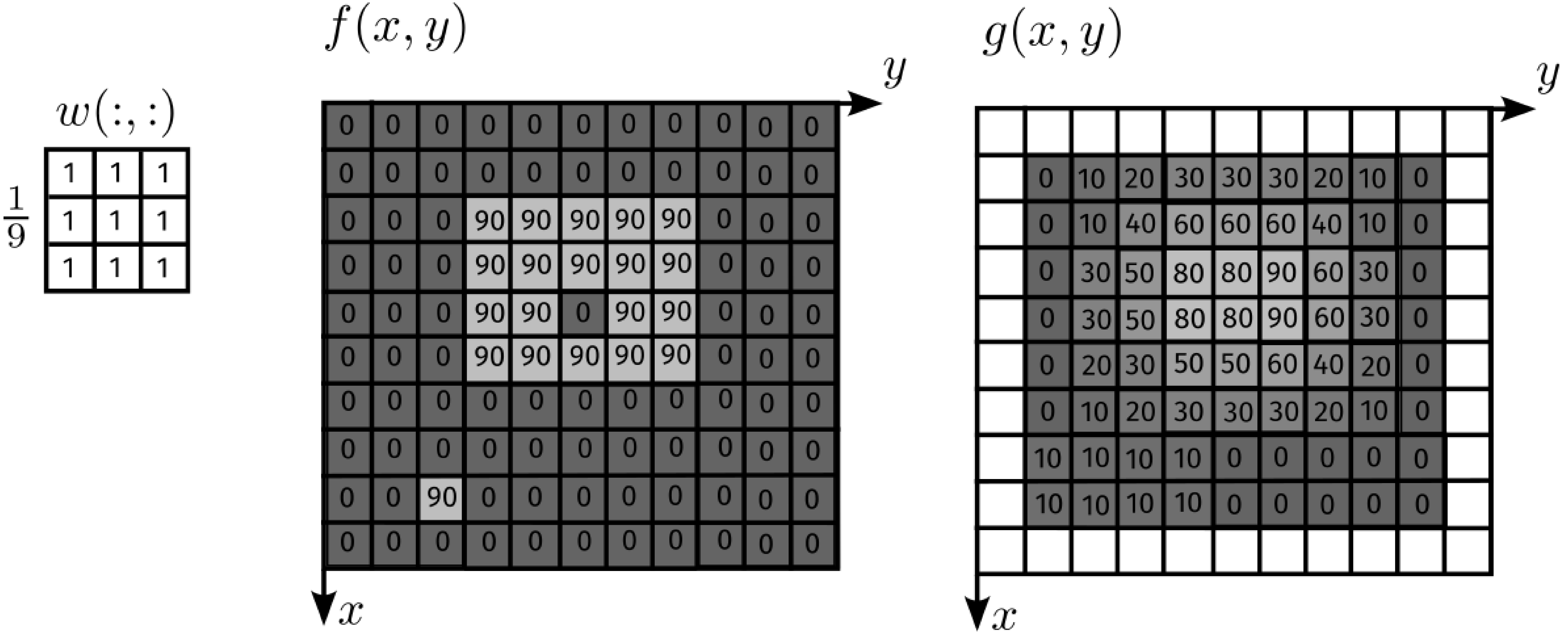
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$g(x,y)$

	0	10	20	30	30	30	20	10	0	
	0	10	40	60	60	60	40	10	0	
	0	30	50	80	80	90	60	30	0	
	0	30	50	80	80	90	60	30	0	
	0	20	30	50	50	60	40	20	0	
	0	10	20	30	30	30	20	10	0	
	10	10	10	10	0	0	0	0	0	
	10	10	10	10	0	0	0	0	0	

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Работа усредняющего фильтра: результат



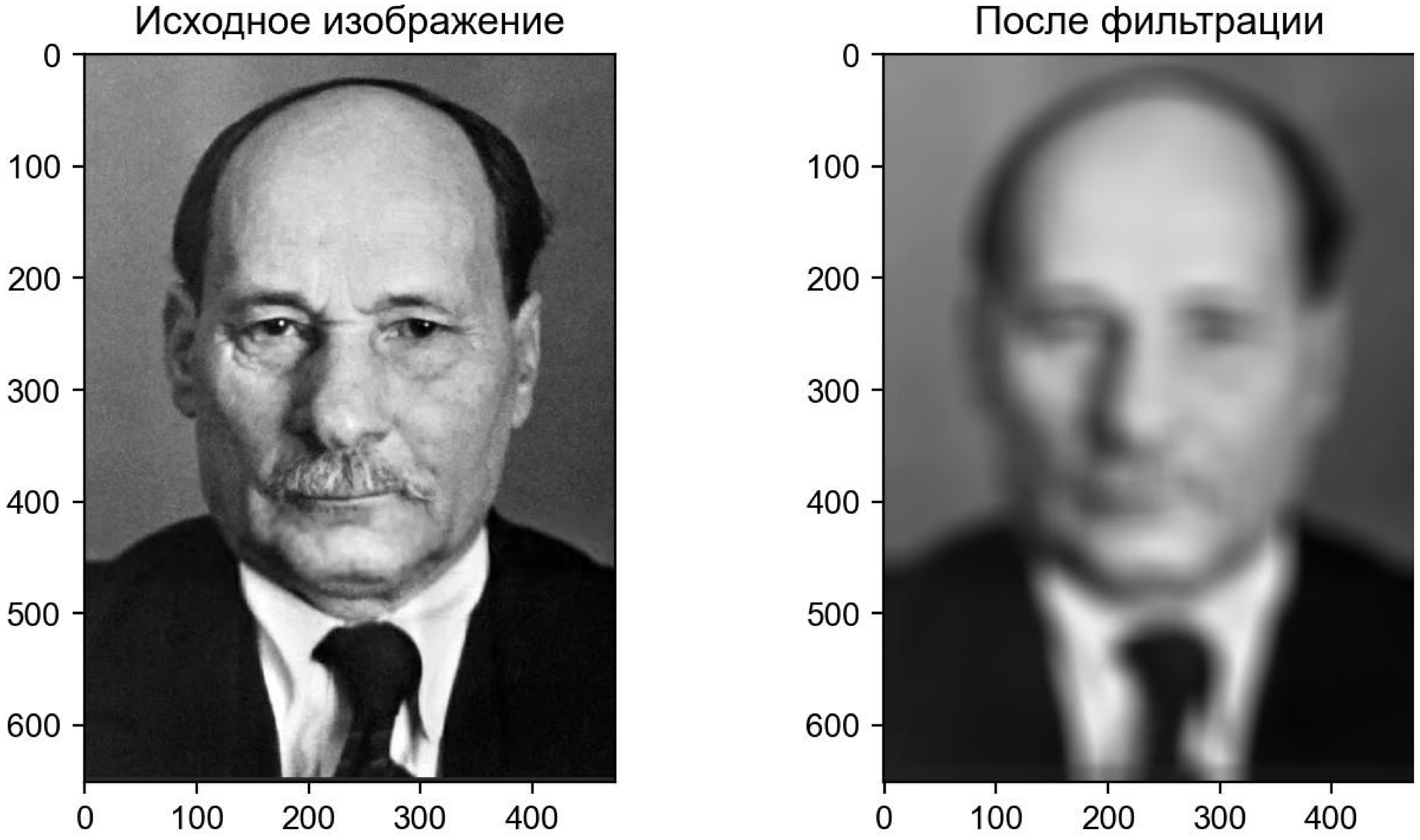
1	1	1
1	1	1
1	1	1

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	0	90	90	0	0	0
0	0	0	90	90	90	90	90	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
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

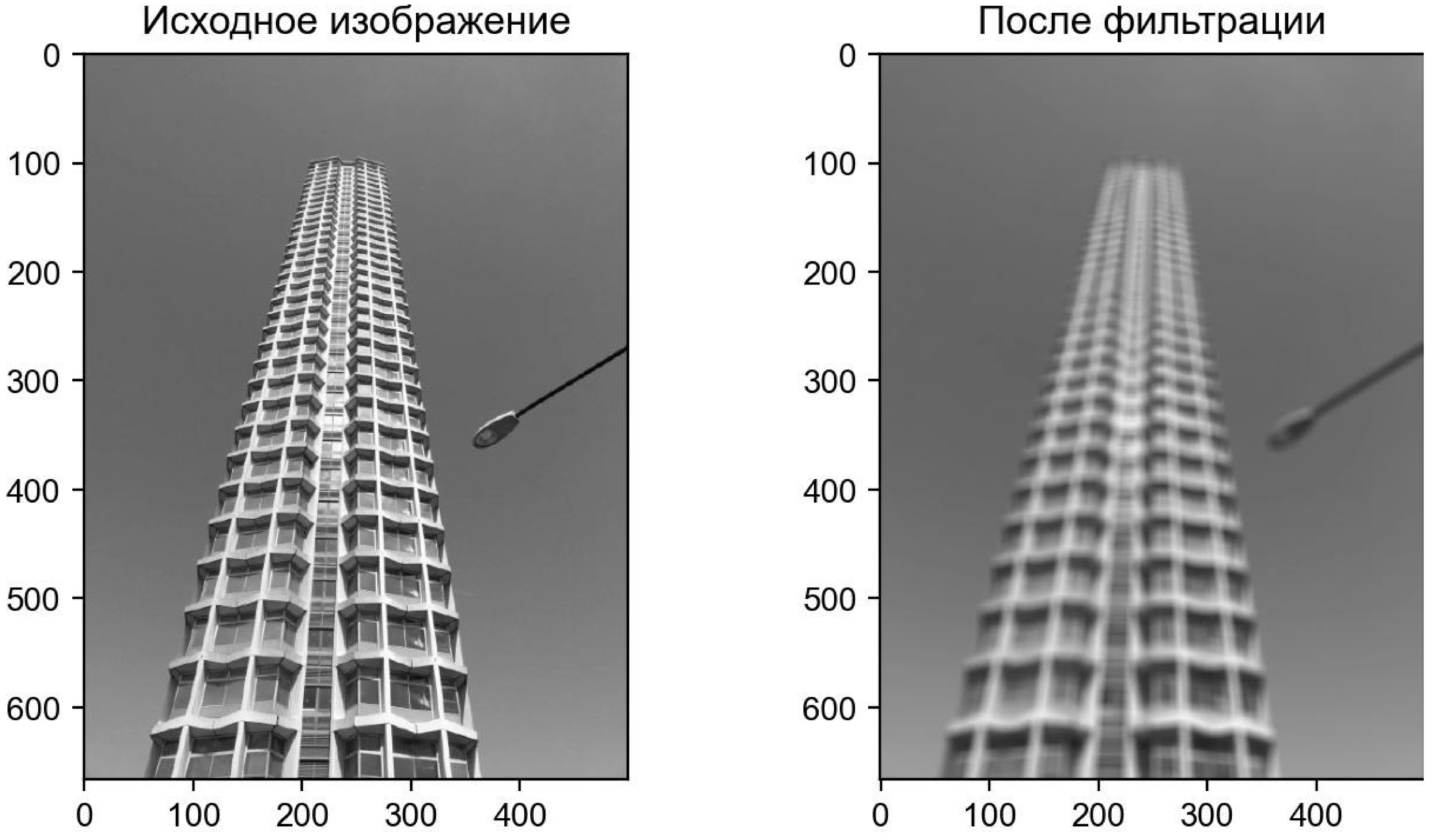
	0	10	20	30	30	30	20	10	0	
	0	10	40	60	60	60	40	10	0	
	0	30	50	80	80	90	60	30	0	
	0	30	50	80	80	90	60	30	0	
	0	20	30	50	50	60	40	20	0	
	0	10	20	30	30	30	20	10	0	
	10	10	10	10	0	0	0	0	0	
	10	10	10	10	0	0	0	0	0	

$$f(x,y) = \sum_{i,j} w(i,j) f(x+i, y+j)$$

Пример работы фильтра на реальном изображении



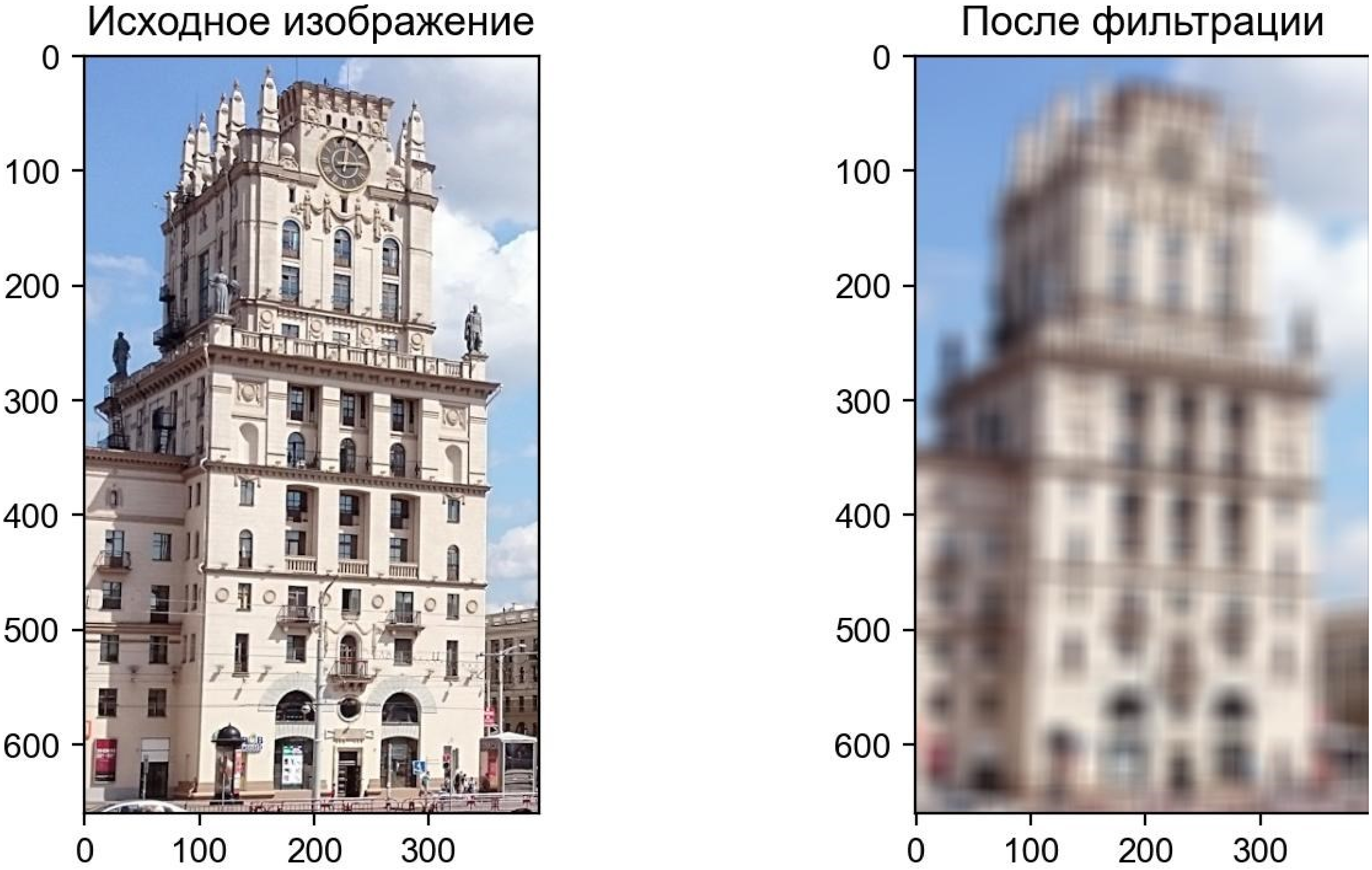
Пример работы фильтра на реальном изображении



Исходное изображение

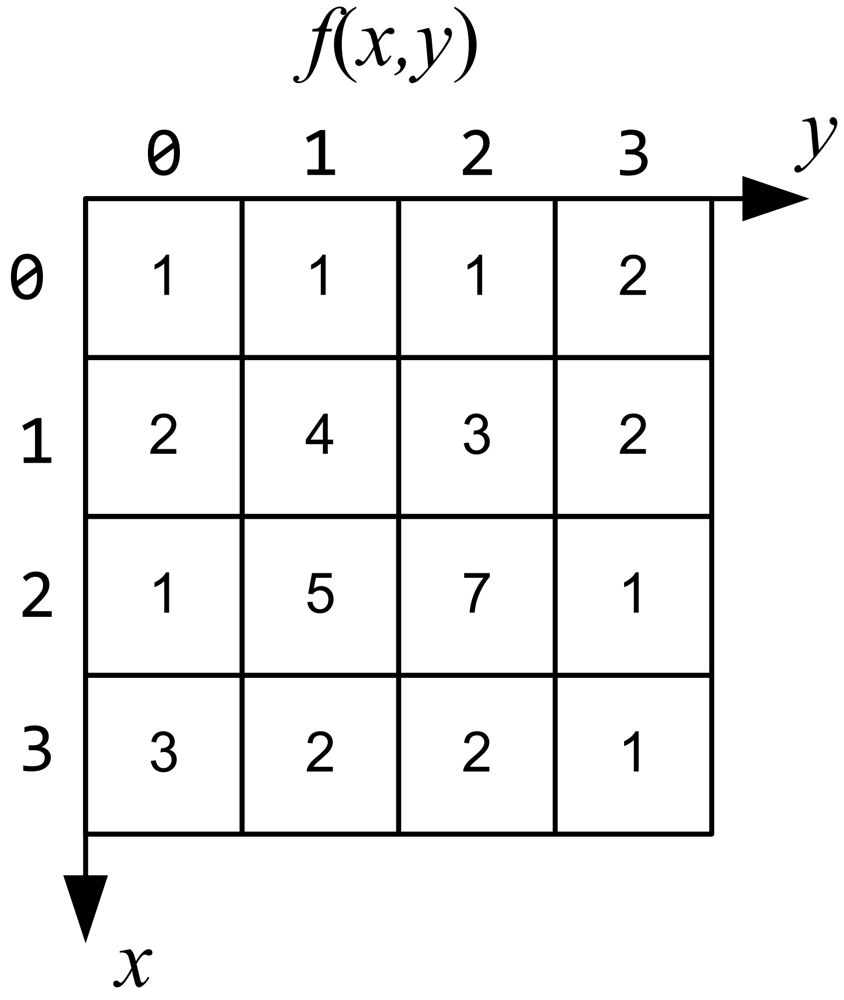
После фильтрации

Пример работы фильтра на реальном изображении



Задача 1 (пример экзаменационной задачи)

Выполнить фильтрацию следующего монохромного изображения $f(x, y)$, однородным усредняющим фильтром по окрестности 3×3. Перед обработкой изображения выполнить расширение путем зеркального отображения.

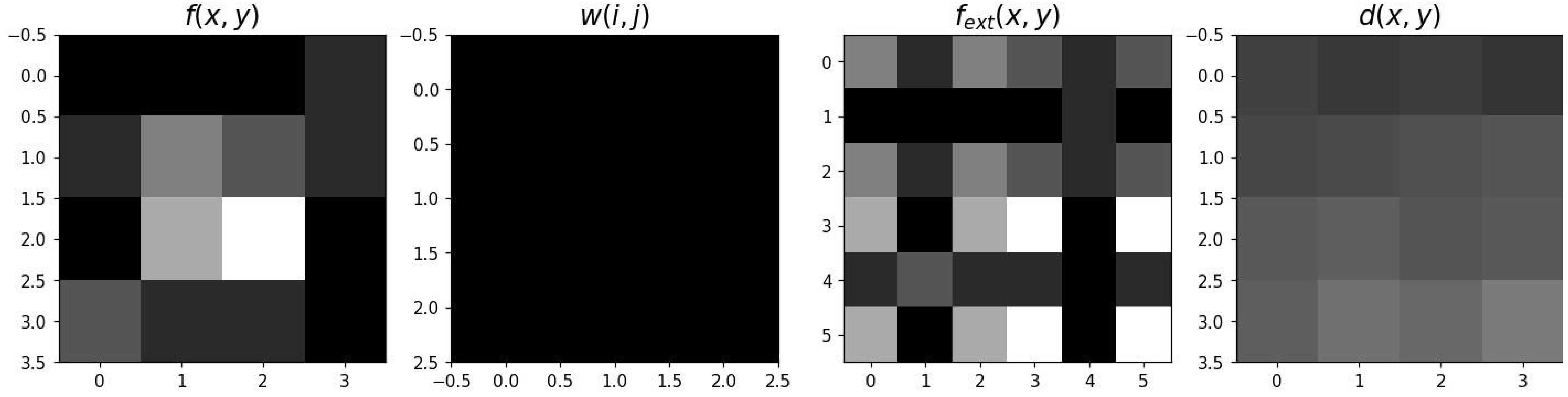


	0	1	2	3
0	1	1	1	2
1	2	4	3	2
2	1	5	7	1
3	3	2	2	1

Ответ к задаче 1

$d(x, y) =$

```
[[2.55   2.33   2.44   2.22]
 [2.66   2.77   2.88   3.00]
 [3.11   3.22   3.00   3.11]
 [3.22   3.66   3.44   3.88]]
```



Вычислительные аспекты 2D свертки

✓ Свертка с ядром размером $K \times K$ требует K^2 MAC-операций на один пиксель.

✓ Во многих случаях эту операцию можно значительно ускорить, выполнив сначала одномерную горизонтальную свертку, за которой следует одномерная вертикальная свертка, что требует в общей сложности $2K$ операций на пиксель.

✓ Ядро свертки, для которого это возможно, называется разделяемым.

Разделяемые фильтры

Двумерный фильтр называется ***разделяемым***, если он может быть представлен в виде произведения «столбца» на «строку»:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \mathbf{v} \times \mathbf{h}^T.$$

✓ Двумерная свертка с разделяемым фильтром эквивалентна двум одномерным сверткам (одна по вертикали и одна по горизонтали).

Разделяемые фильтры

Двумерный фильтр называется ***разделяемым***, если он может быть представлен в виде произведения «столбца» на «строку»:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \mathbf{v} \times \mathbf{h}^T.$$

✓ Двумерная свертка с разделяемым фильтром эквивалентна двум одномерным сверткам (одна по вертикали и одна по горизонтали).

✓ Если мы выполняем свертку изображения размером $M \times M$ с фильтром, ядро которого имеет размер $K \times K$:

- Сколько потребуется операций в случае, если фильтр будет неразделяемым (non-separable)?

Разделяемые фильтры

Двумерный фильтр называется ***разделяемым***, если он может быть представлен в виде произведения «столбца» на «строку»:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \mathbf{v} \times \mathbf{h}^T.$$

✓ Двумерная свертка с разделяемым фильтром эквивалентна двум одномерным сверткам (одна по вертикали и одна по горизонтали).

✓ Если мы выполняем свертку изображения размером $M \times M$ с фильтром, ядро которого имеет размер $K \times K$:

- Сколько потребуется операций в случае, если фильтр будет неразделяемым (non-separable)? → $M^2 \times K^2$
- Сколько потребуется операций в случае, если фильтр будет разделяемым (separable)?

Разделяемые фильтры

Двумерный фильтр называется ***разделяемым***, если он может быть представлен в виде произведения «столбца» на «строку»:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \mathbf{v} \times \mathbf{h}^T.$$

✓ Двумерная свертка с разделяемым фильтром эквивалентна двум одномерным сверткам (одна по вертикали и одна по горизонтали).

✓ Если мы выполняем свертку изображения размером $M \times M$ с фильтром, ядро которого имеет размер $K \times K$:

- ○ Сколько потребуется операций в случае, если фильтр будет неразделяемым (non-separable)? → $M^2 \times K^2$
- ○ Сколько потребуется операций в случае, если фильтр будет разделяемым (separable)? → $M^2 \times 2K$

Как узнать является ли фильтр разделяемым?

✓ Установить, что фильтр является разделимым часто можно, изучив аналитическое выражение, которым описываются коэффициенты фильтра.

✓ Более прямой подход заключается в том, чтобы применить к ядру (матрице) фильтра сингулярное разложение (*SVD – singular value decomposition*):

$$\mathbf{w} = \sum_i \sigma_m \mathbf{u}_i \mathbf{v}_i^T$$

Если в результате разложения окажется, что только значение σ_0 является ненулевым, то ядро w является разделимым причем

$$\mathbf{v} = \sqrt{\sigma_0}\mathbf{u}_0, \qquad \mathbf{h} = \sqrt{\sigma_0}\mathbf{v}_0^T$$

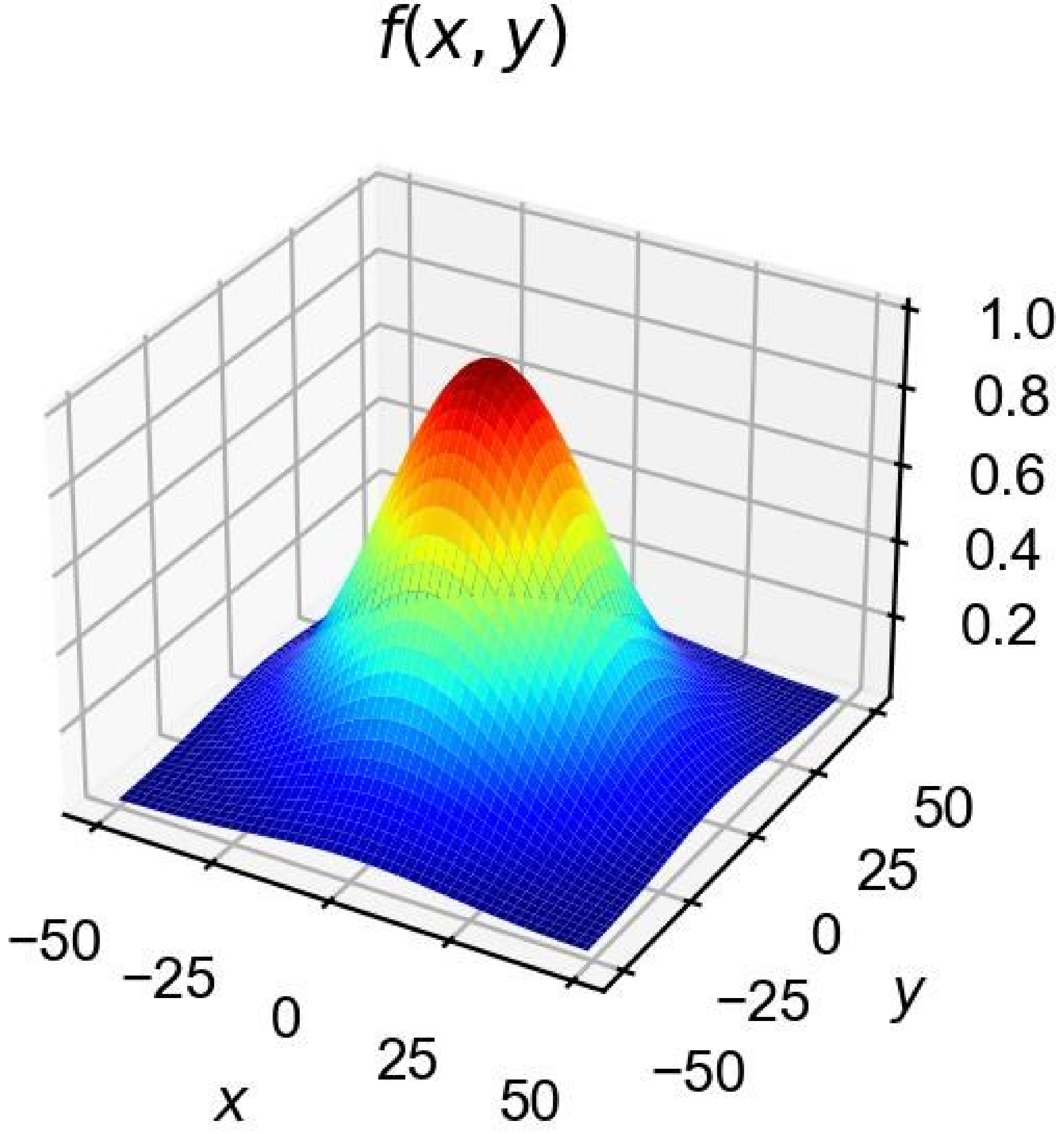
✓ Даже, если ядро фильтра является неразделимым применение SVD может быть использовано для ускорения свертки.

Фильтр Гаусса

- ✓ Ядро фильтра получается путем дискретизации двумерной функции Гаусса

$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$$

- ✓ Коэффициенты ядра спадают по мере удаления пикселей от центра.



- ✓ На практике используют ядра небольшого размера

$$\mathbf{w} = \frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}.$$

- ✓ Является ли такой фильтр разделимым?

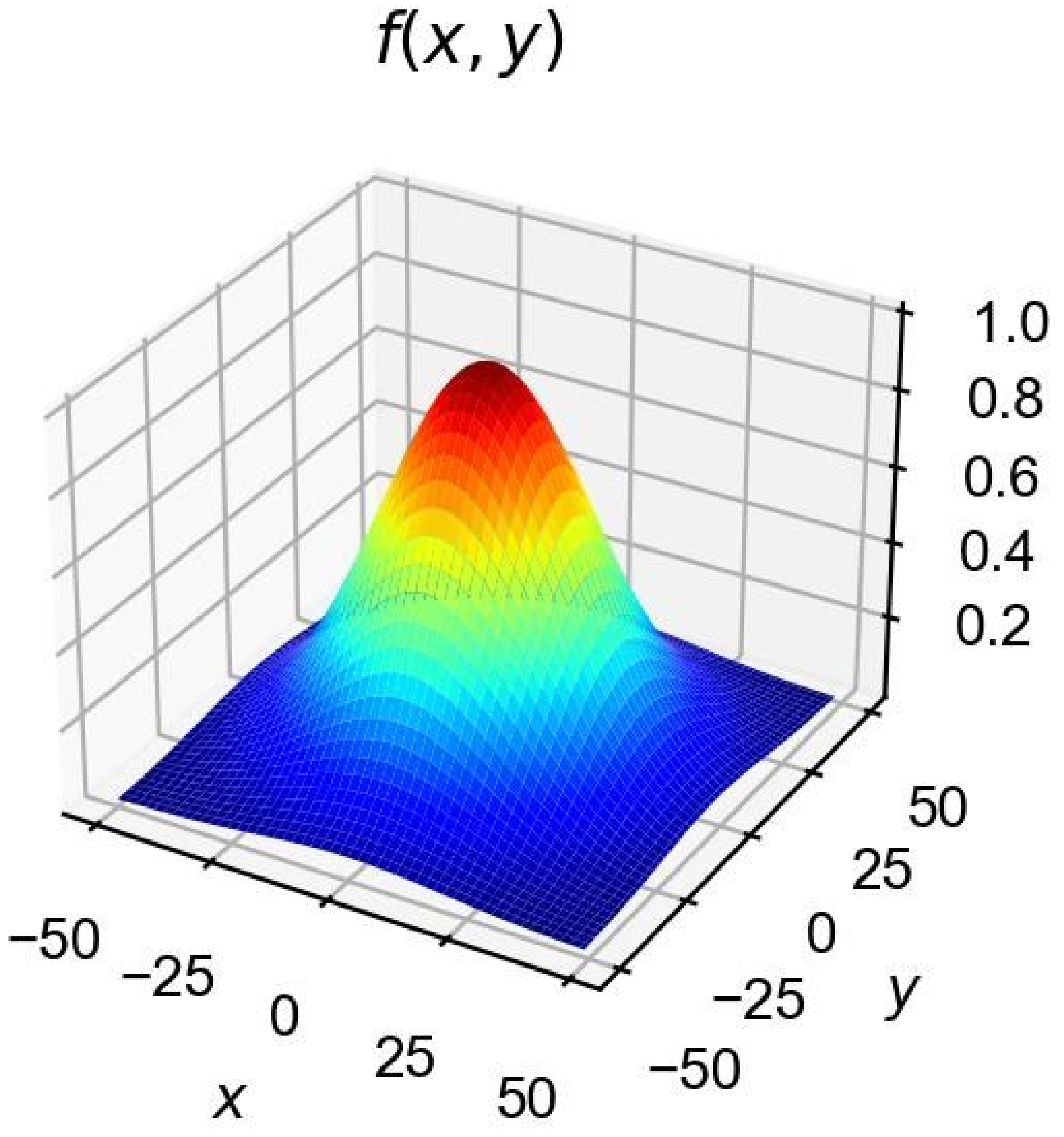
Фильтр Гаусса

- ✓ Ядро фильтра получается путем дискретизации двумерной функции Гаусса

$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$$

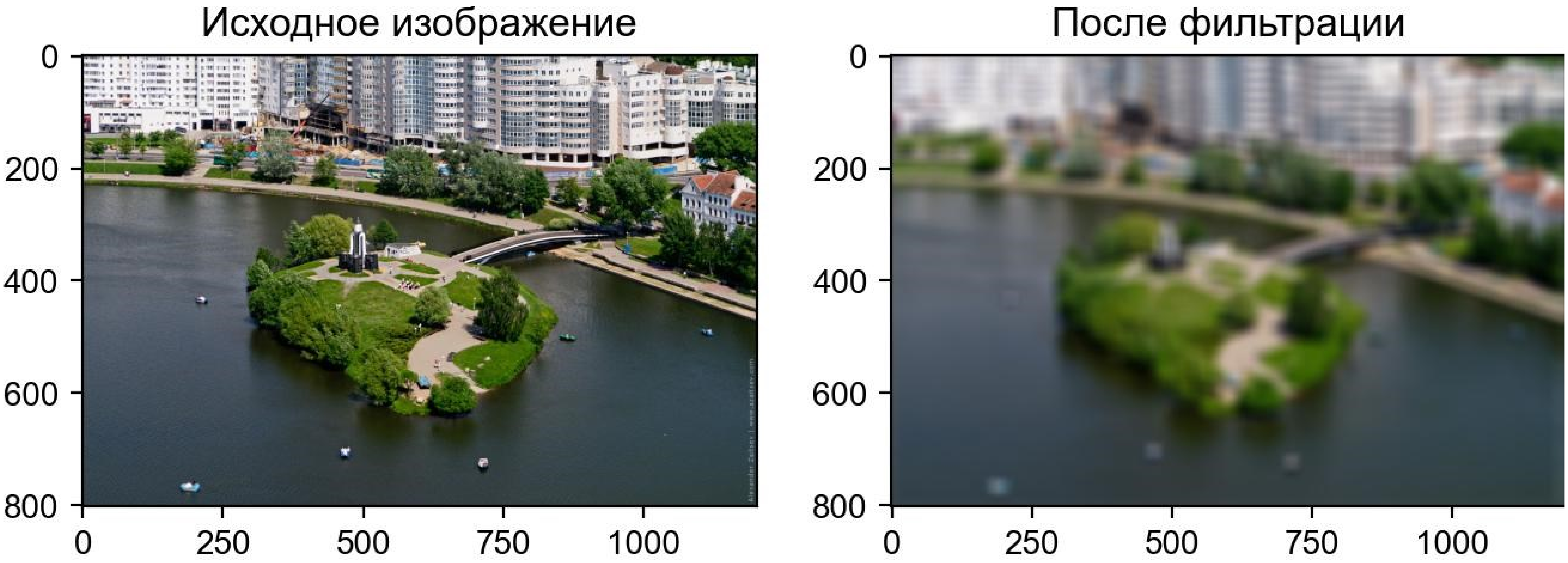
- ✓ Коэффициенты ядра спадают по мере удаления пикселей от центра.
- ✓ На практике используют ядра небольшого размера

$$\mathbf{w} = \frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}.$$

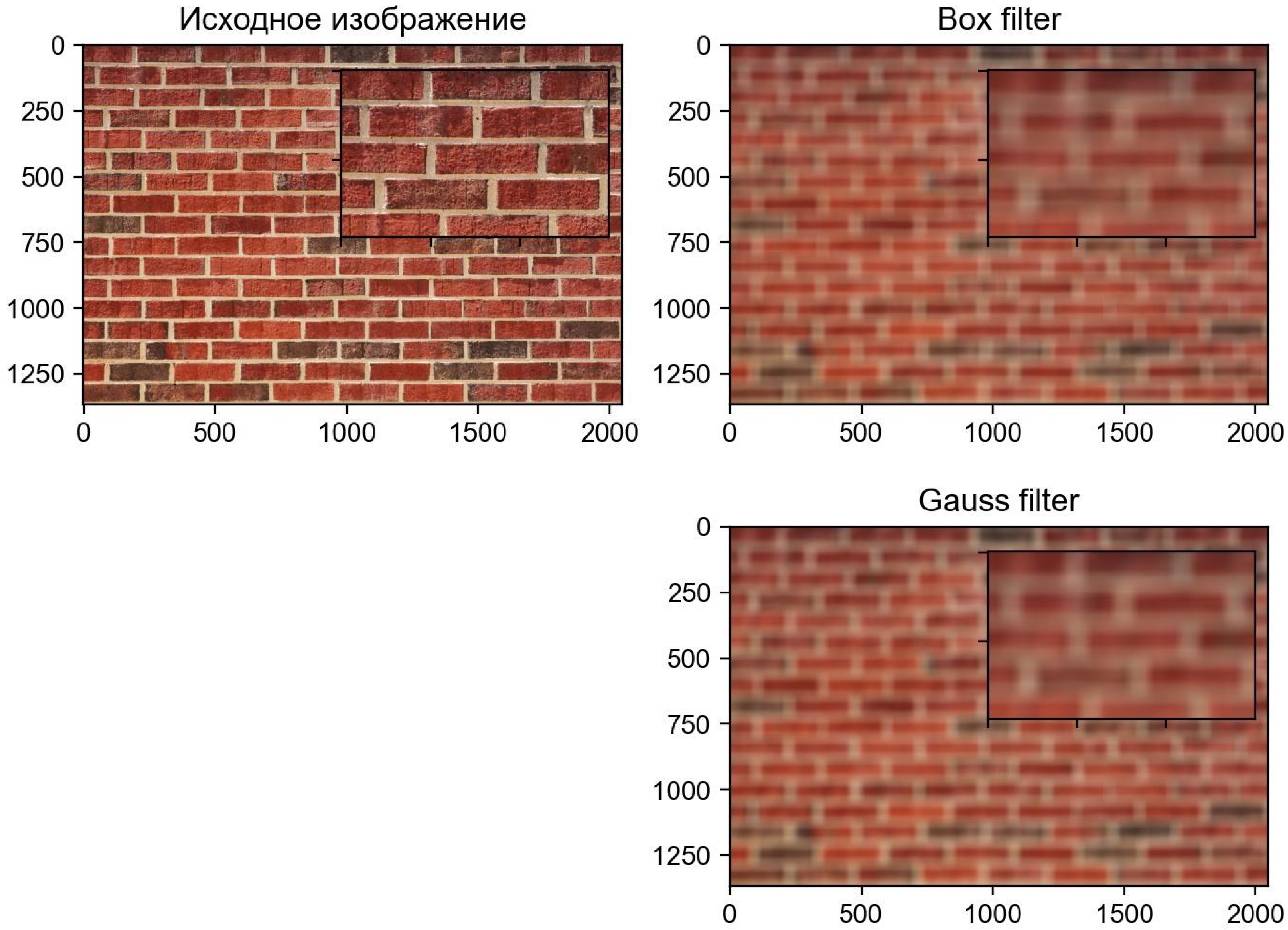


- ✓ Является ли такой фильтр разделимым? → Да!
- ✓ Размер фильтра обычно равен 2-3σ.

Применение фильтра Гаусса



Сравнение фильтра Гаусса и усредняющего фильтра


Исходное изображение

Box filter

Gauss filter

Сравнение фильтра Гаусса и усредняющего фильтра

Оригинал

Усредняющий фильтр

Ядро 5x5

Фильтр Гаусса

Ядро 5x5

Сравнение фильтра Гаусса и усредняющего фильтра

Оригинал

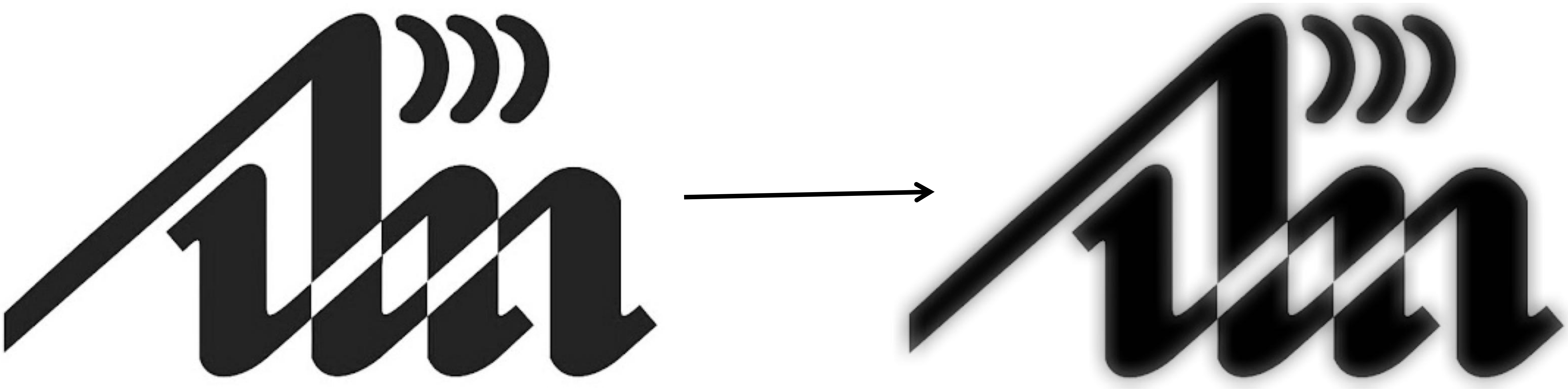
Усредняющий фильтр

Ядро 5x5

Фильтр Гаусса

Ядро 5x5

Как создать эффект мягкой тени?



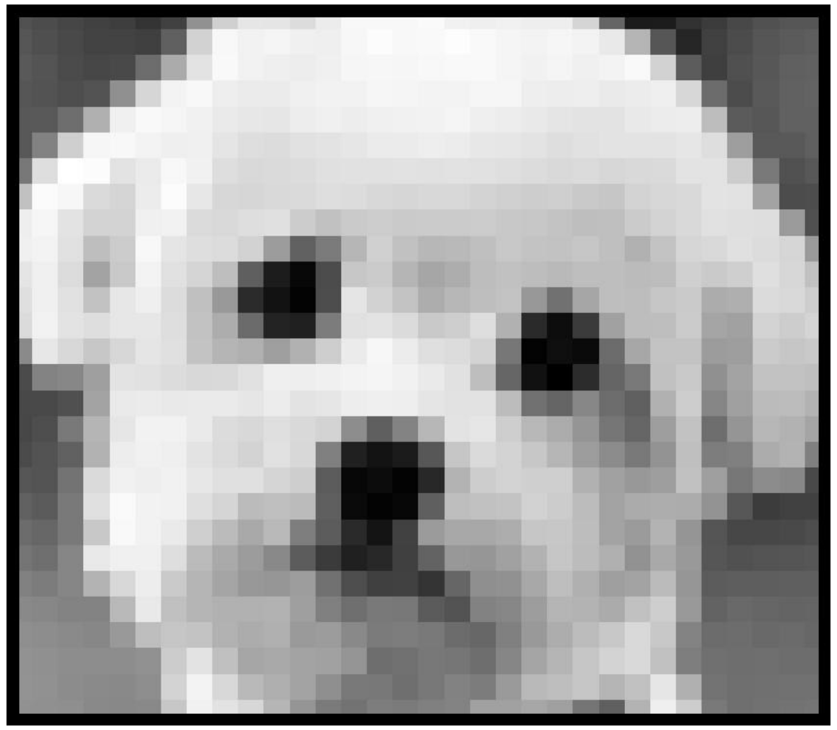
Как создать эффект мягкой тени?

Наложение

Применение фильтра Гаусса
(гауссово размытие)

Применение свертки

Вход	Фильтр	Выход

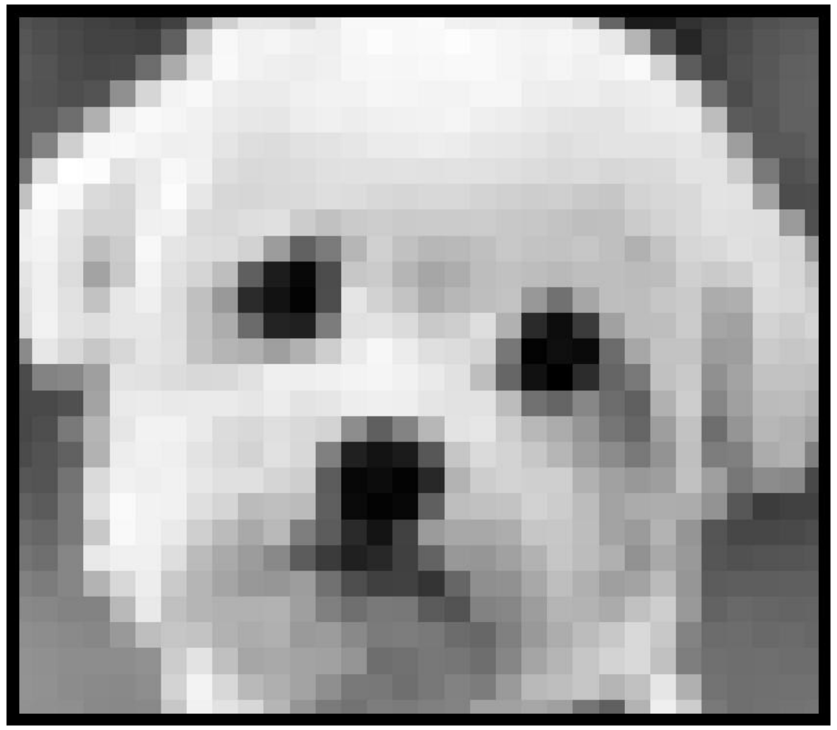


0	0	0
0	1	0
0	0	0

?

Применение свертки

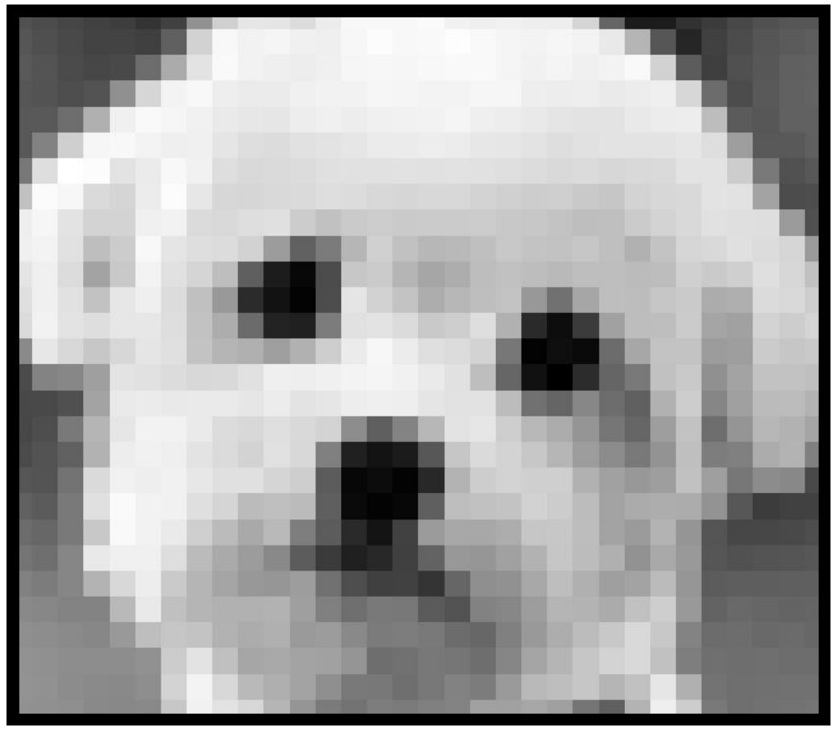
Вход



Фильтр

0	0	0
0	1	0
0	0	0

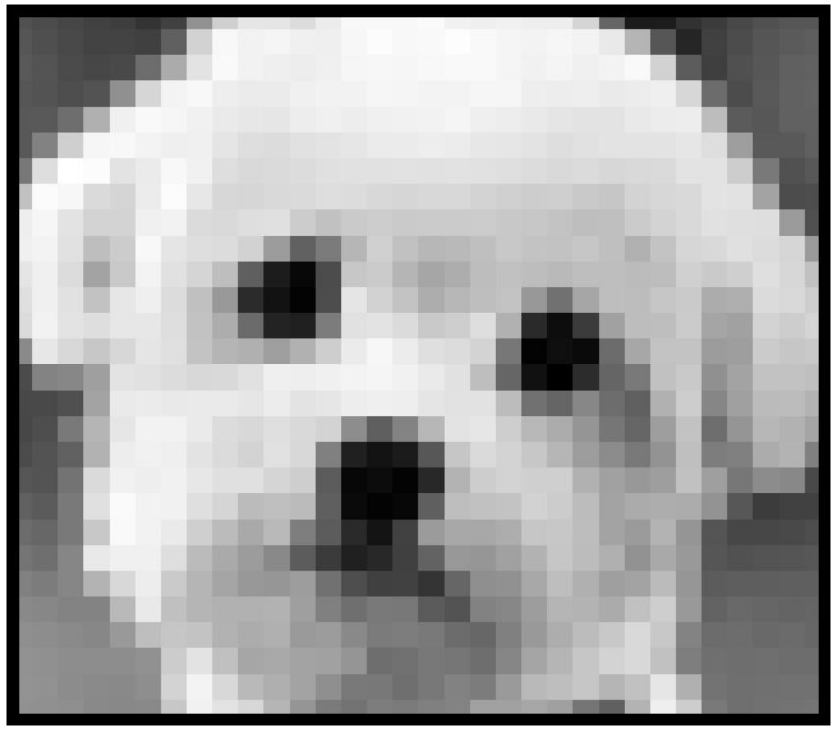
Выход



Без изменений

Применение свертки

Вход



Фильтр

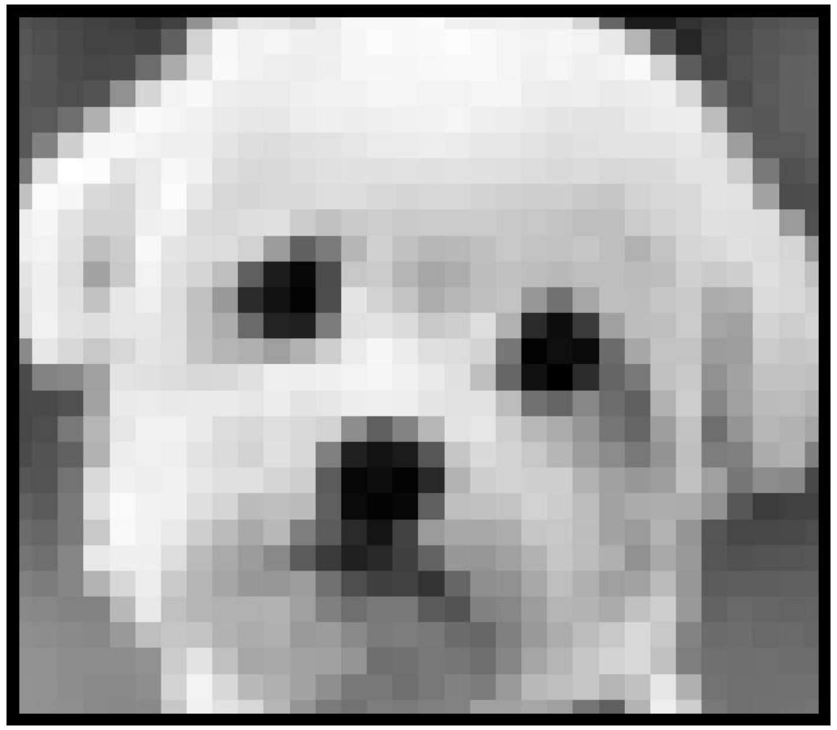
0	0	0
0	0	1
0	0	0

Выход

?

Применение свертки

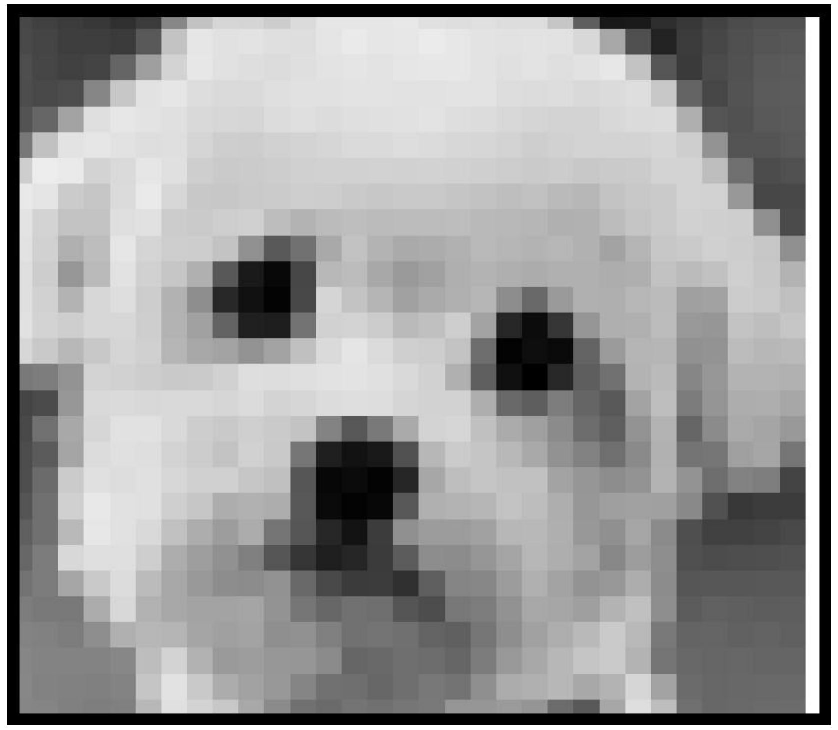
Вход



Фильтр

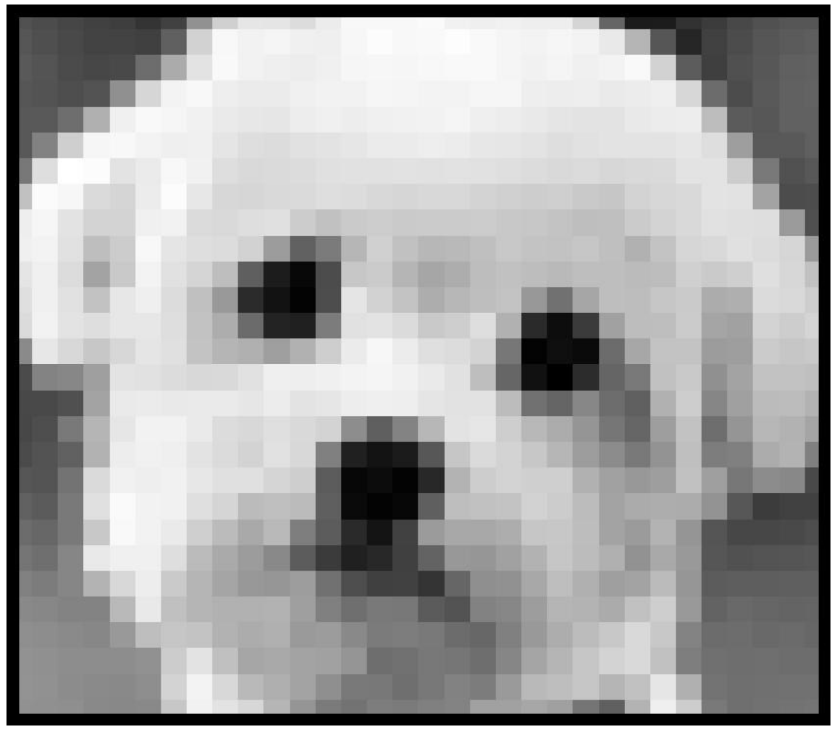
0	0	0
0	0	1
0	0	0

Выход



Применение свертки

Вход



Фильтр

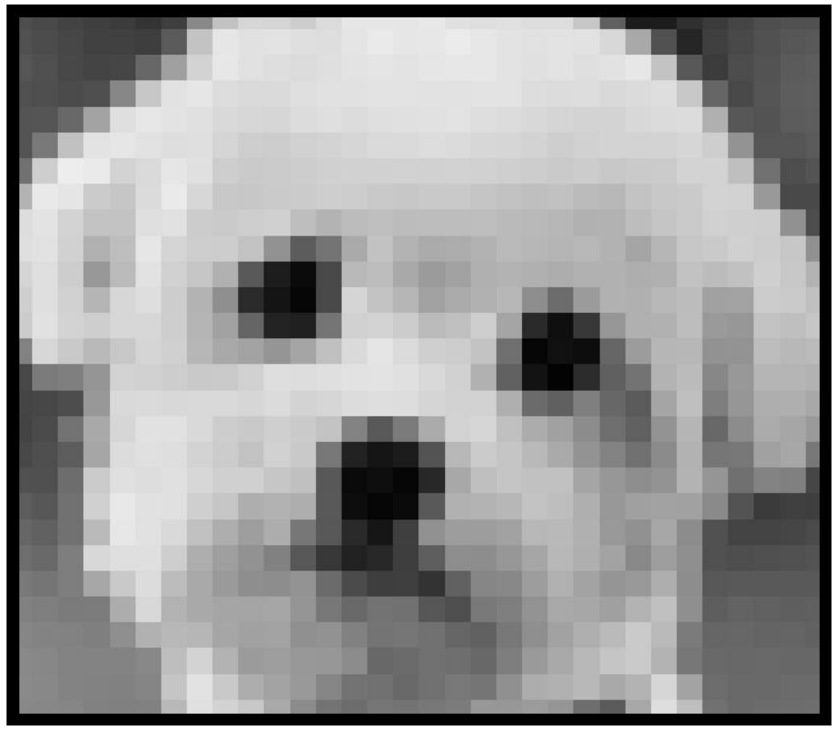
$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Выход

?

Применение свертки

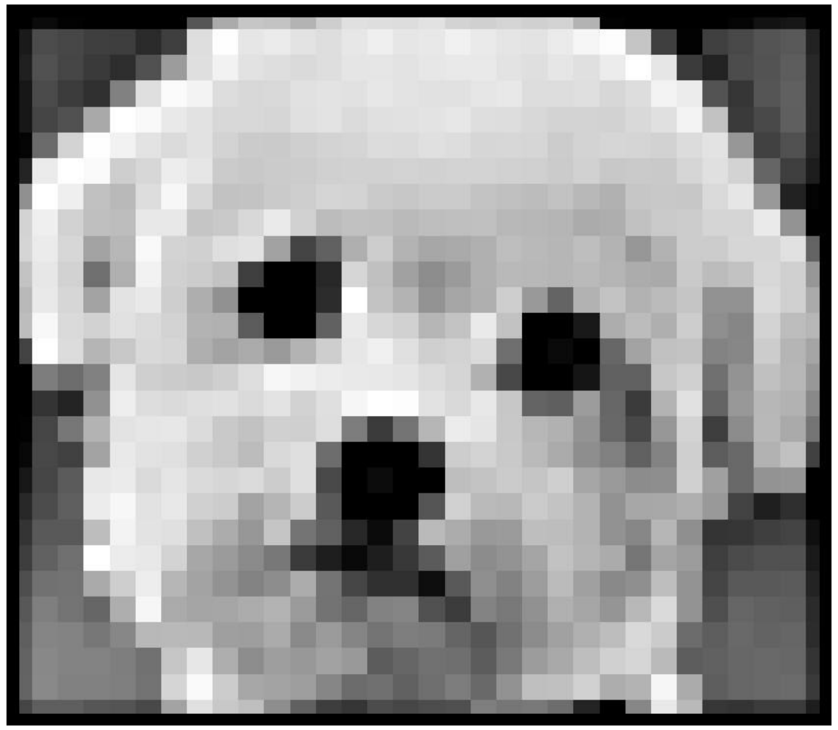
Вход



Фильтр

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Выход



Повышение резкости

- Фильтр не изменяет областей с постоянной интенсивностью
- Фильтр усиливает перепады интенсивности

Повышение резкости

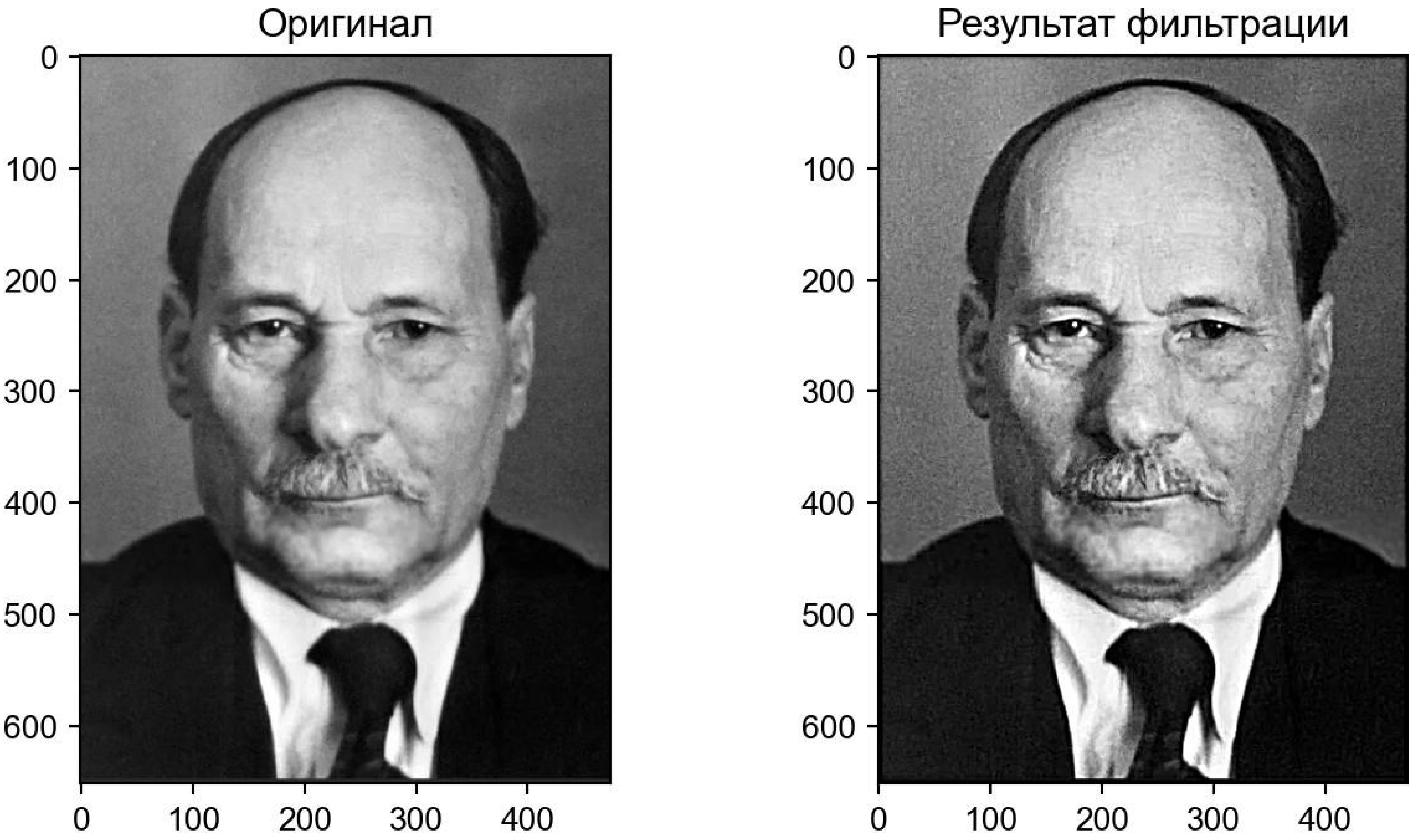
Удивительно, но ядра сглаживающих фильтров также можно использовать для повышения резкости изображений, используя метод, называемый нерезкое маскирование (*unsharped masking*). Поскольку размытие изображения уменьшает высокие частоты, добавление некоторой разницы между исходным и размытым изображением делает его более четким

$$g_{sharpen} = f + \gamma(f - w_{blur} * f).$$

В предыдущем примере было использовано $\gamma = 1$, поэтому

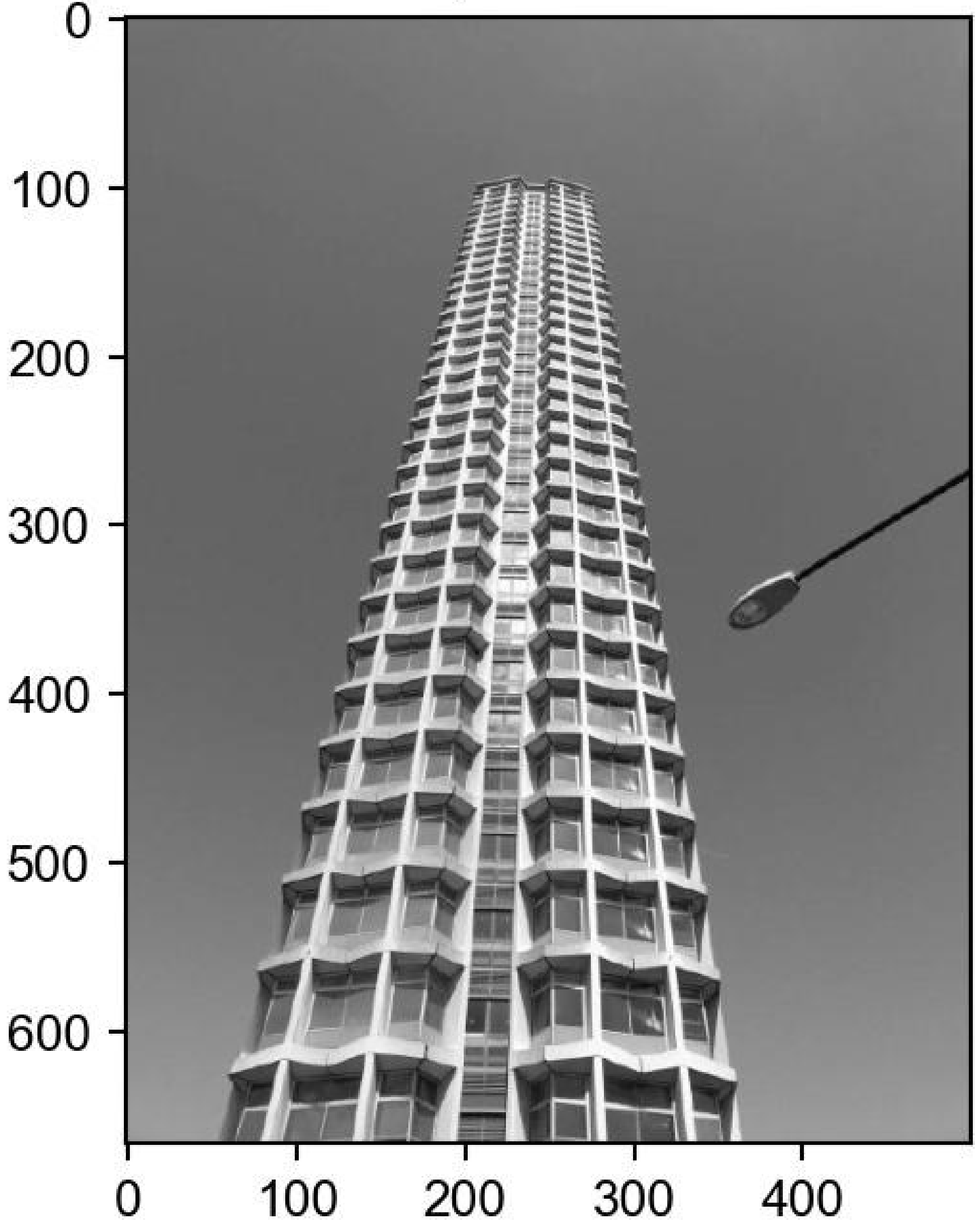
$$g_{sharpen} = 2f - w_{blur} * f = (2 - w_{blur}) * f.$$

Пример повышения резкости

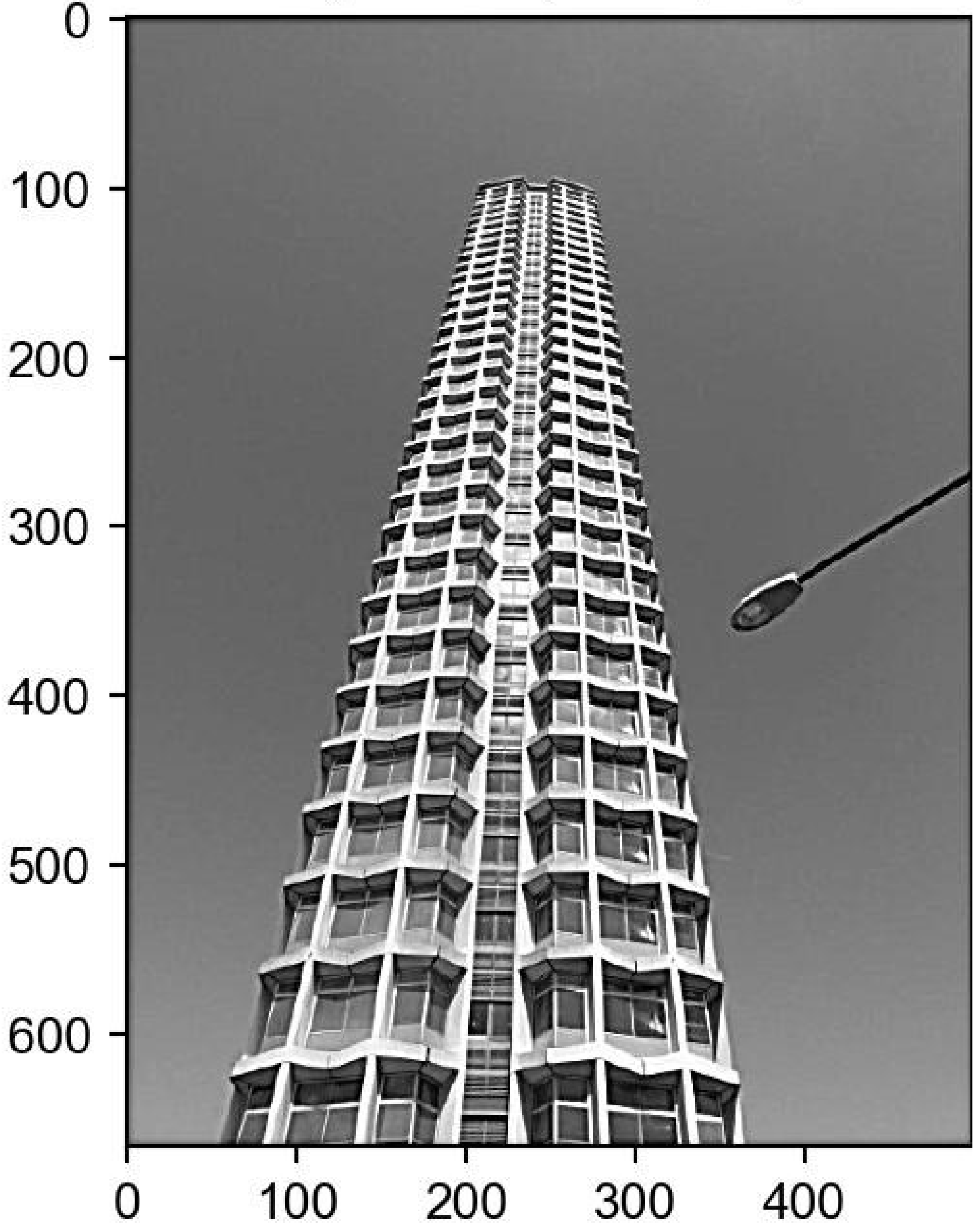


Пример повышения резкости

Оригинал



Результат фильтрации



Фильтры повышения резкости

Цель повышения резкости – подчеркнуть мелкие детали, которые оказались расфокусированными вследствие ошибок или несовершенства метода съемки.

Ранее было показано, что расфокусировка (размытие) изображения достигается путем усреднения точек, лежащих в окрестности. Усреднение аналогично интегрированию, поэтому логично предположить, что повышения резкости можно достичь действием обратным к интегрированию, т.е. дифференцированием.

Величина отклика оператора взятия производной в точке изображения будет пропорциональна «степени разрывности» изображения в данной точке, т.е. вместе с перепадами интенсивности, относящимися к объектам сцены могут быть усилены и шумы, присутствующие на изображении.

Градиент изображения

Рассмотрим вначале одномерный случай. Производная дискретной функции $f(x)$ определяются, как конечная разность

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

Вторая производная определяется как разность соседних значений первой производной:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x-1).$$

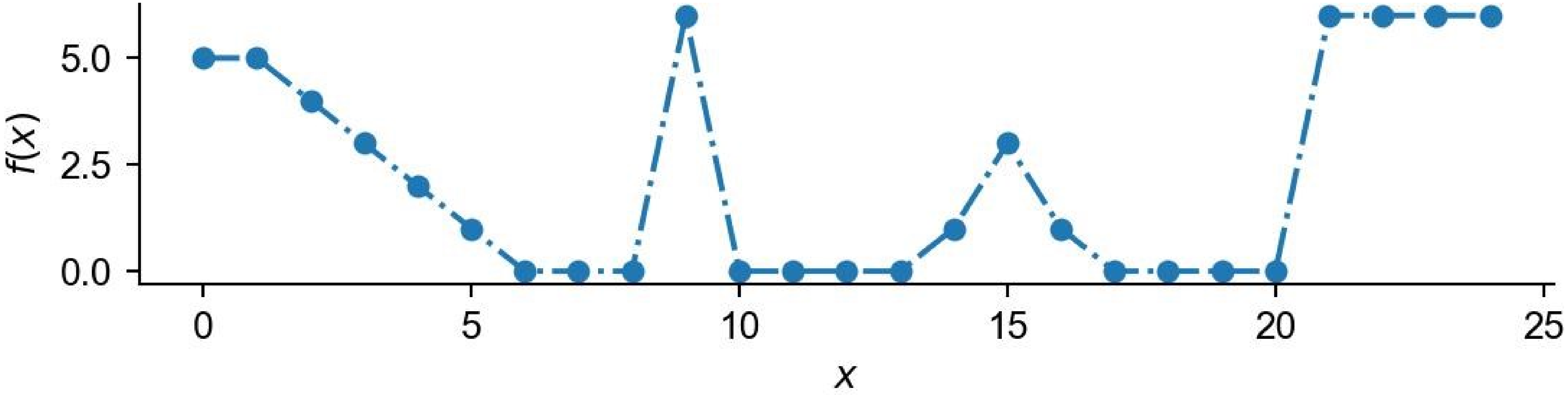
Градиент изображения

Первая производная

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

Вторая производная

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x-1).$$



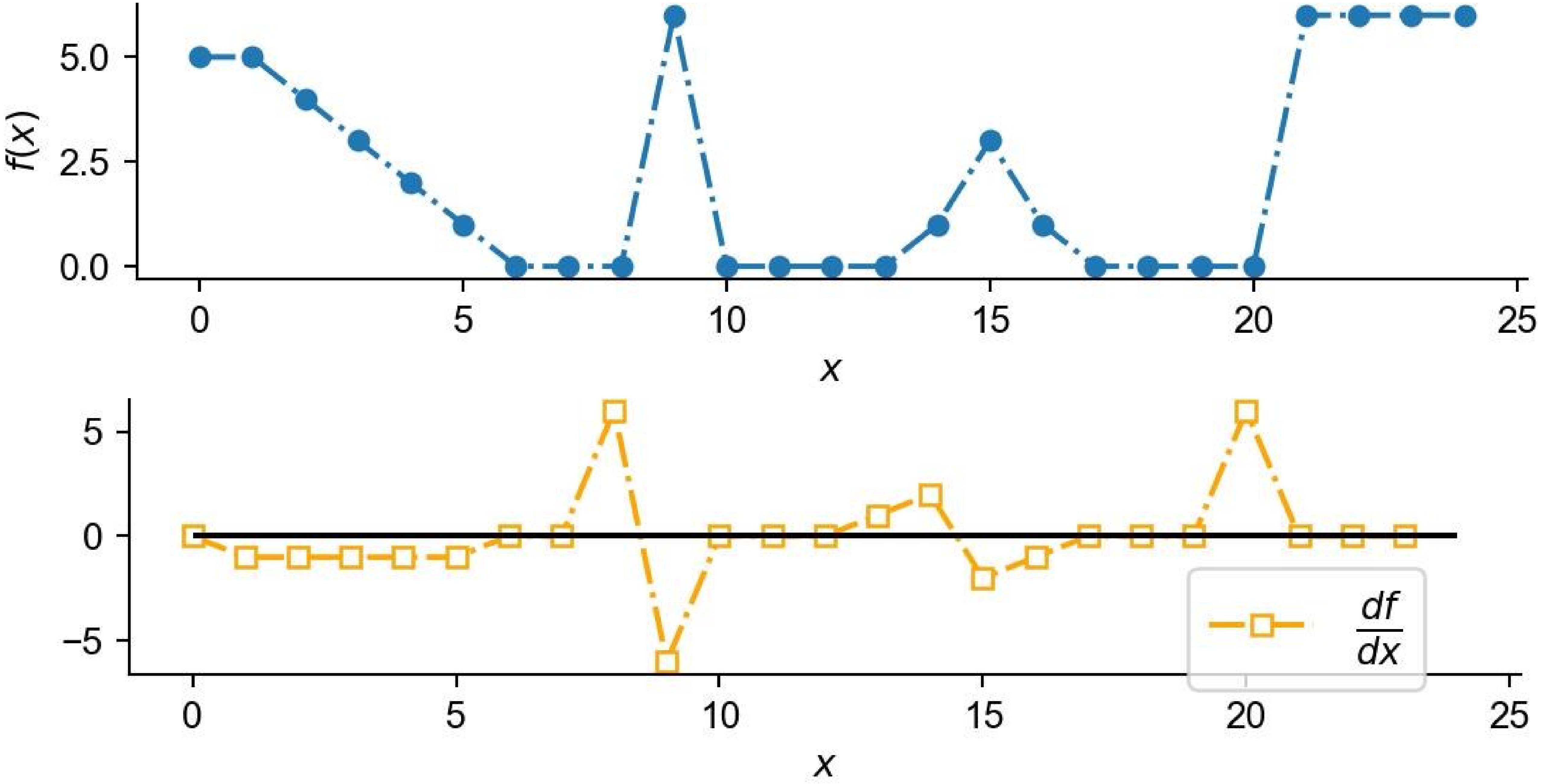
Градиент изображения

Первая производная

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

Вторая производная

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x+1).$$



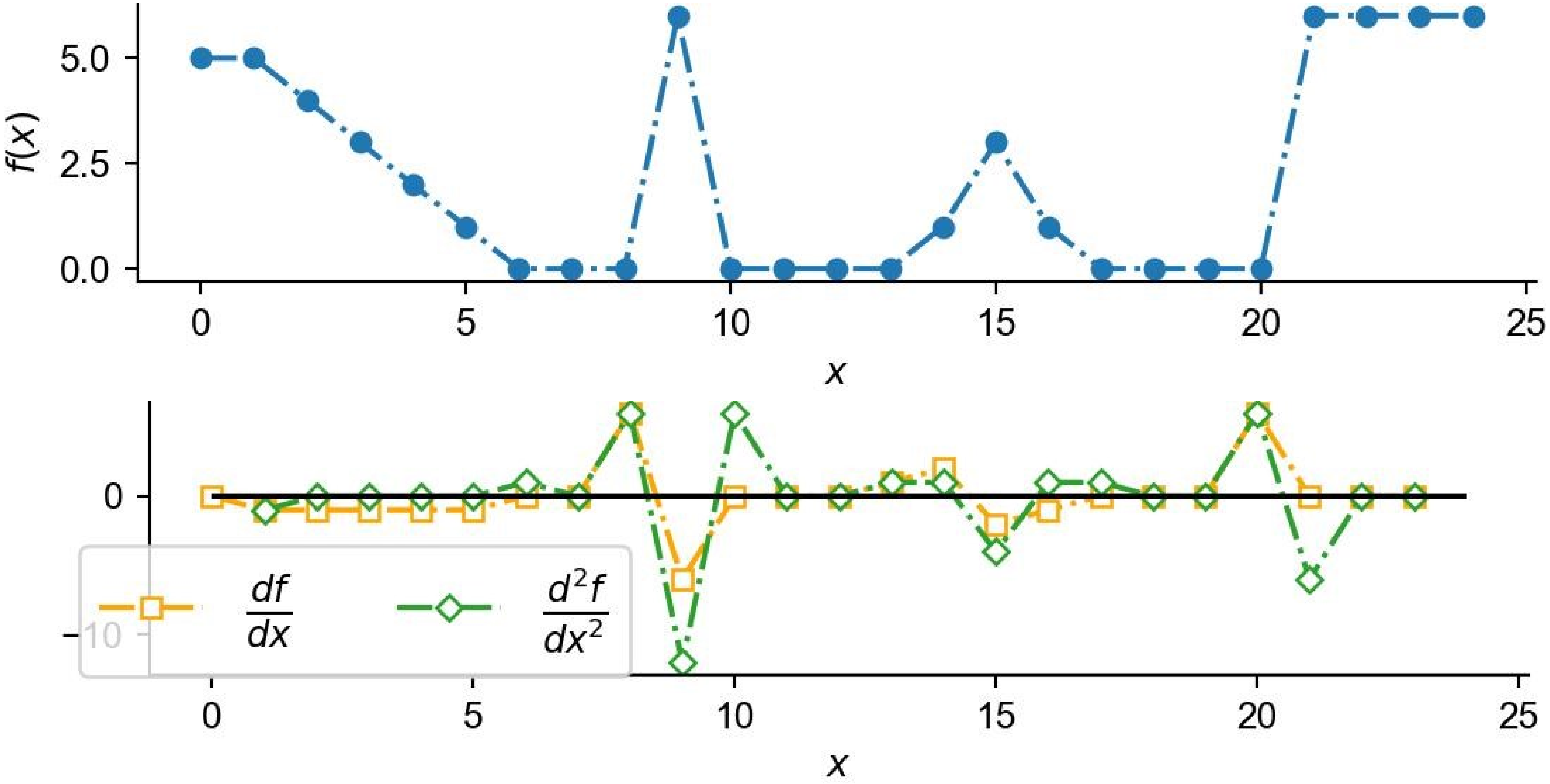
Градиент изображения

Первая производная

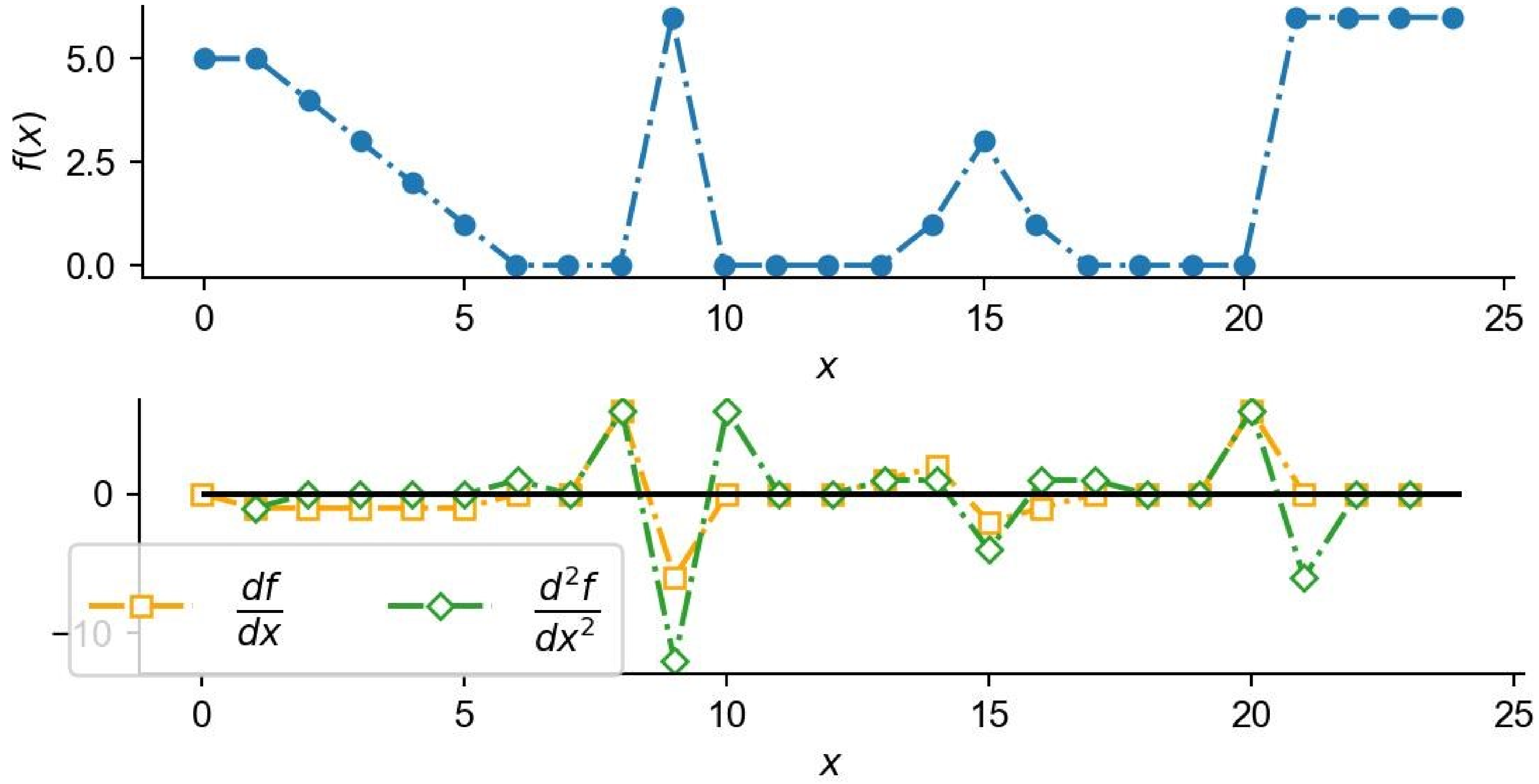
$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

Вторая производная

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x+1).$$



Градиент изображения



- ➢ Первая производная дает «толстые» контуры, а вторая – более тонкие.
- ➢ Отделанная точка (при $x = 9$) имеет больший отклик для второй производной.
- ➢ На ступеньке отклики обеих производных совпадают.
- ➢ Значение второй производной имеют колебания из положительных значений в отрицательные, на изображении это выглядит, как двойная линия.

Первая и вторая производные: вывод

- В методах улучшения качества изображений вторая производная оказывается более предпочтительной, поскольку лучше усиливает мелкие детали.

- Первая производная используется в методах выделения контуров (*edge detection*).

Повышение резкости изображения: лапласиан

Изотропный фильтр – это фильтр, отклик которого не зависит от направления неоднородностей на обрабатываемом изображении.

Если повернуть изображение на 90 градусов и обработать его изотропным фильтром, то результат будет тот же, как и в случае обработки этим фильтром изображения без поворота.

Оператор Лапласа

Простейшим изотропным оператором является лапласиан:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

Вторая производная по x:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

Вторая производная по y:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Оператор Лапласа

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} =$$

$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)$$

$$- 4f(x, y)$$

Данное уравнение может быть записано в виде свертки

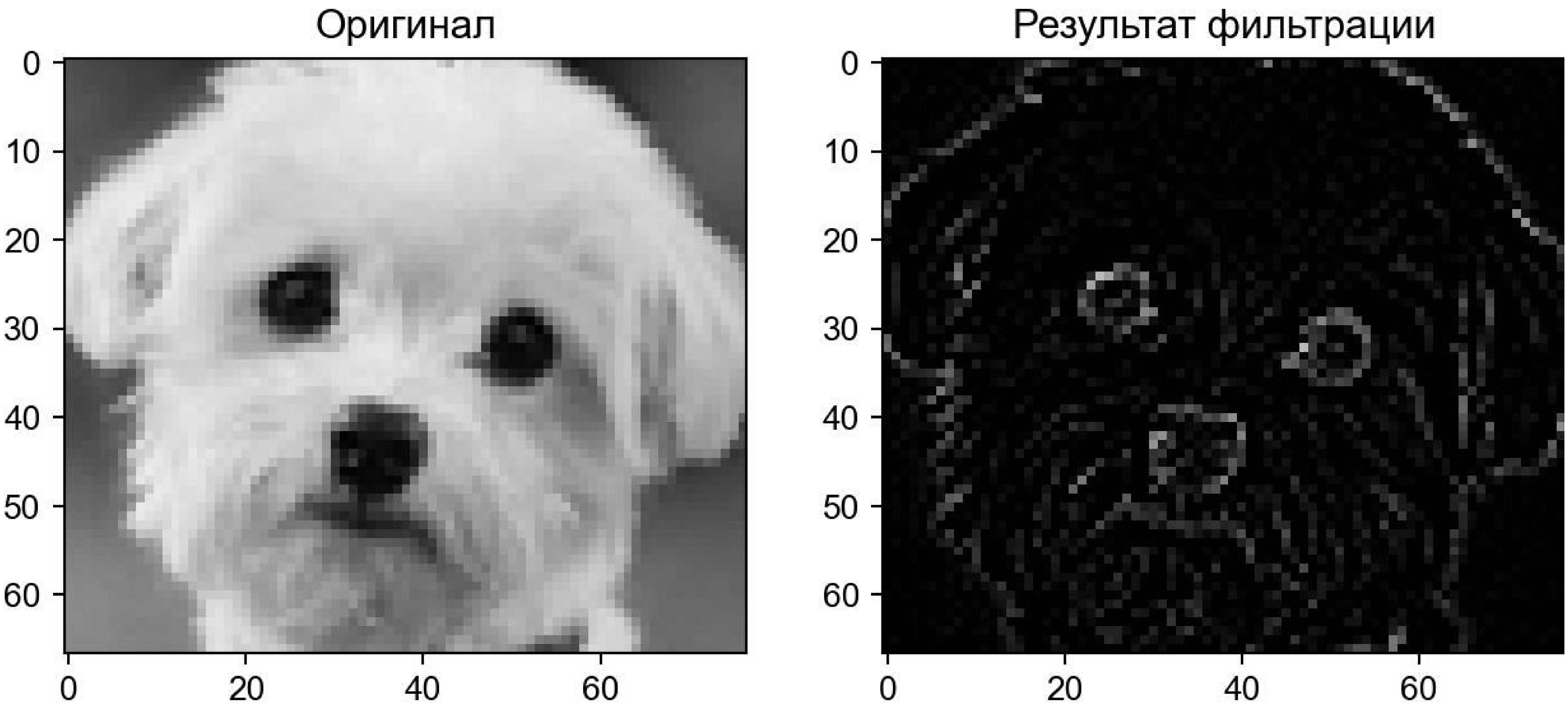
$$\nabla^2 f = f * \mathbf{w}$$

где $\mathbf{w}$ – ядро фильтра:

$$\mathbf{w} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

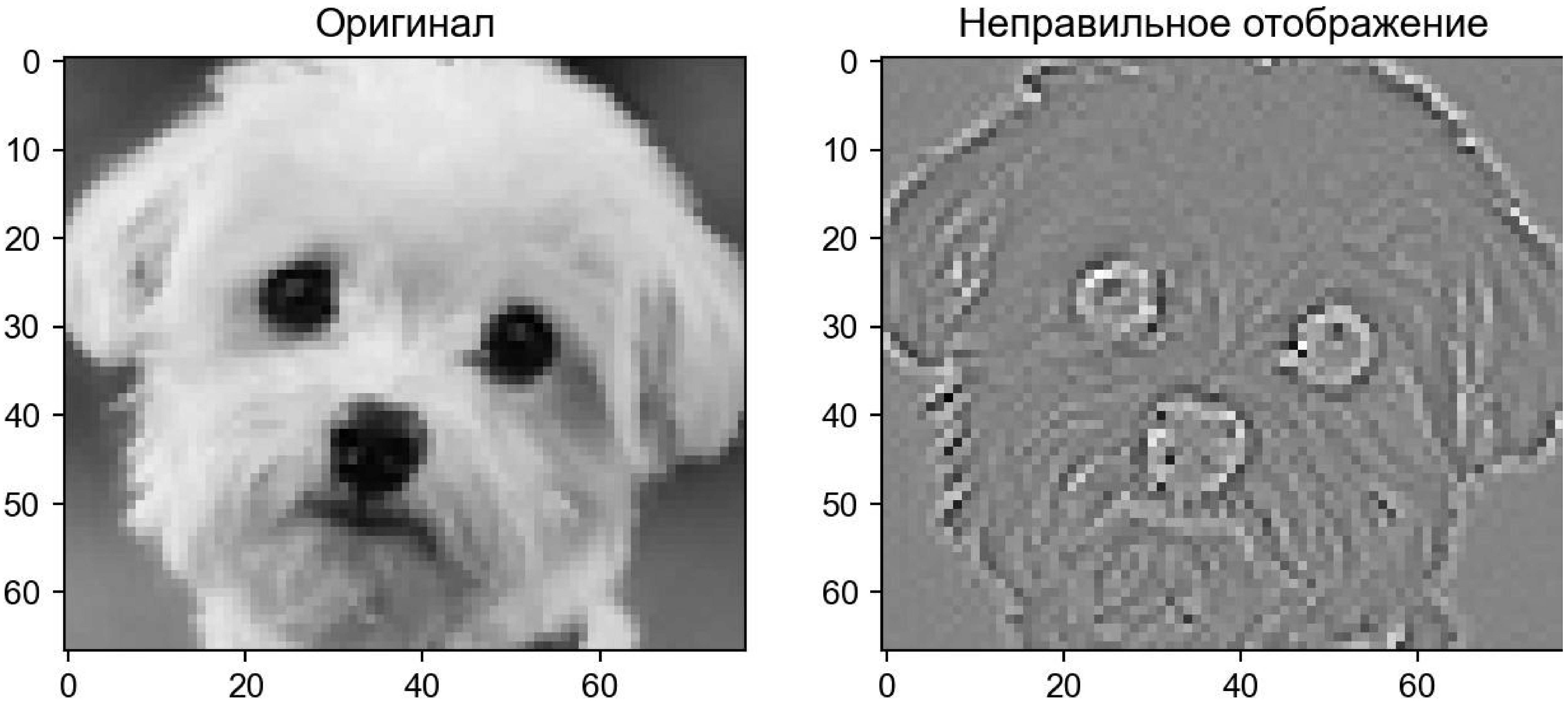
Применение лапласиана подчеркивает разрывы уровней яркости на изображении и подавляет области со слабыми изменениями яркости.

Применение оператора Лапласа



$$\mathbf{w} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Применение оператора Лапласа


Оригинал

Неправильное отображение

```
plt.imshow(img_dst, cmap='gray')   # неправильное отображение
plt.imshow(img_dst, cmap='gray', vmin=0, vmax=255) # правильное отображение
```

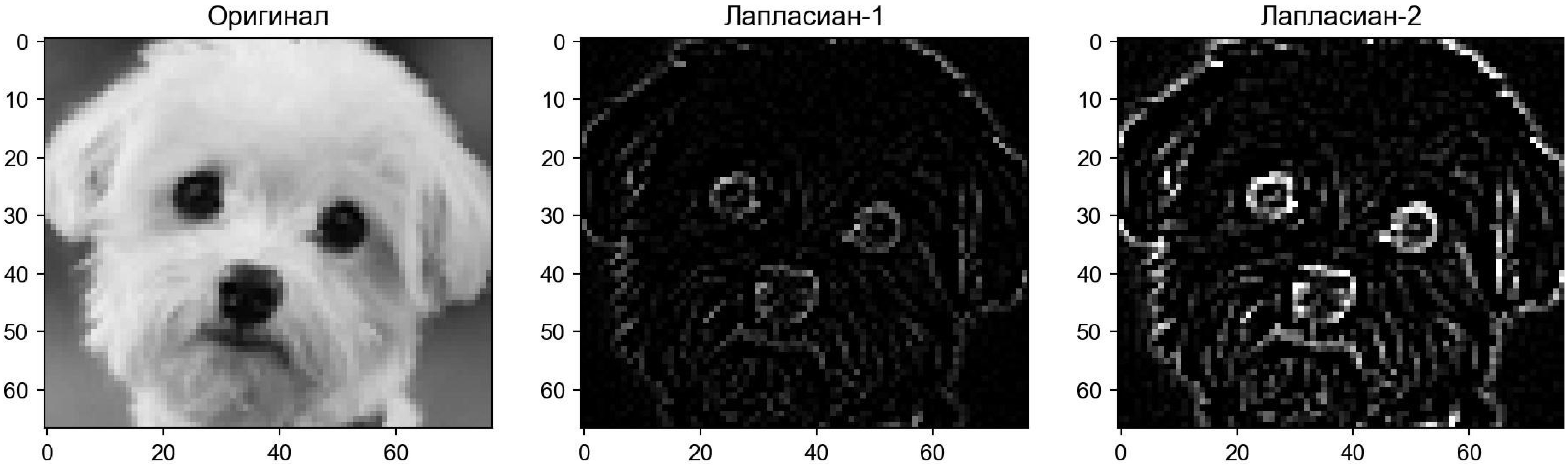
Лапласиан-2

Диагональные направления могут быть включены в формулу дискретного лапласиана:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Такая ядро фильтра является изотропным для поворотов на углы, кратные 45°.

Пример обработки

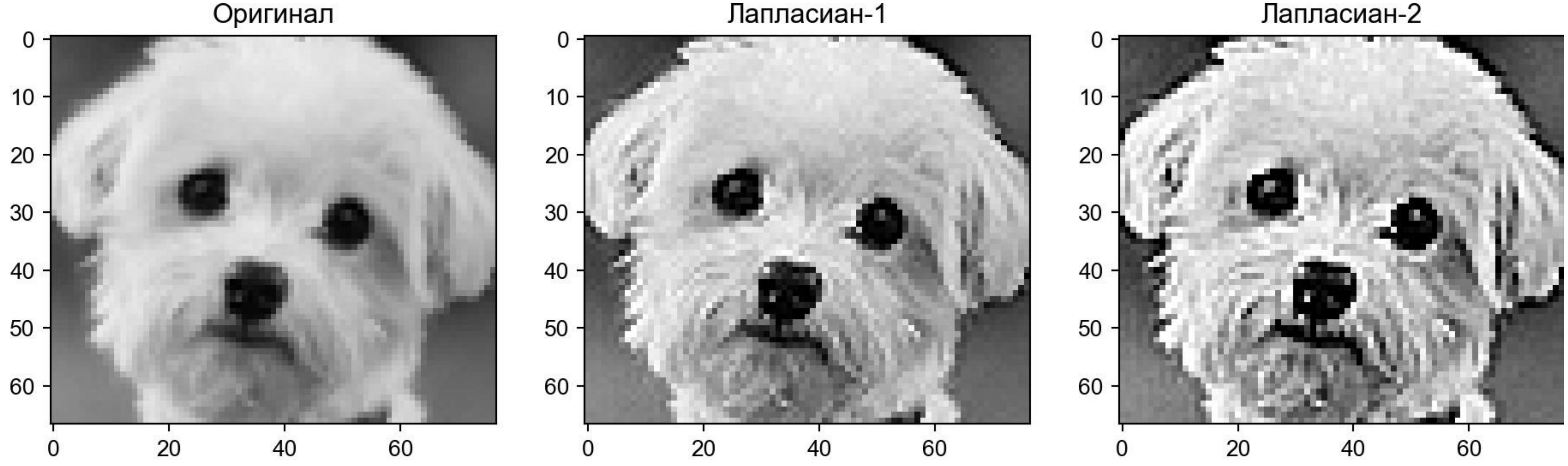


Применение лапласиана на повышения резкости

Поскольку оператор Лапласа по сути является второй производной, его применение подчеркивает разрывы уровней яркостей на изображении и подавляет области со слабыми изменениями яркостей. Это приводит к получению изображения, содержащего сероватые линии на месте контуров и других разрывов, наложенные на темный фон без особенностей. Но фон можно «восстановить», сохранив эффект повышения резкости:

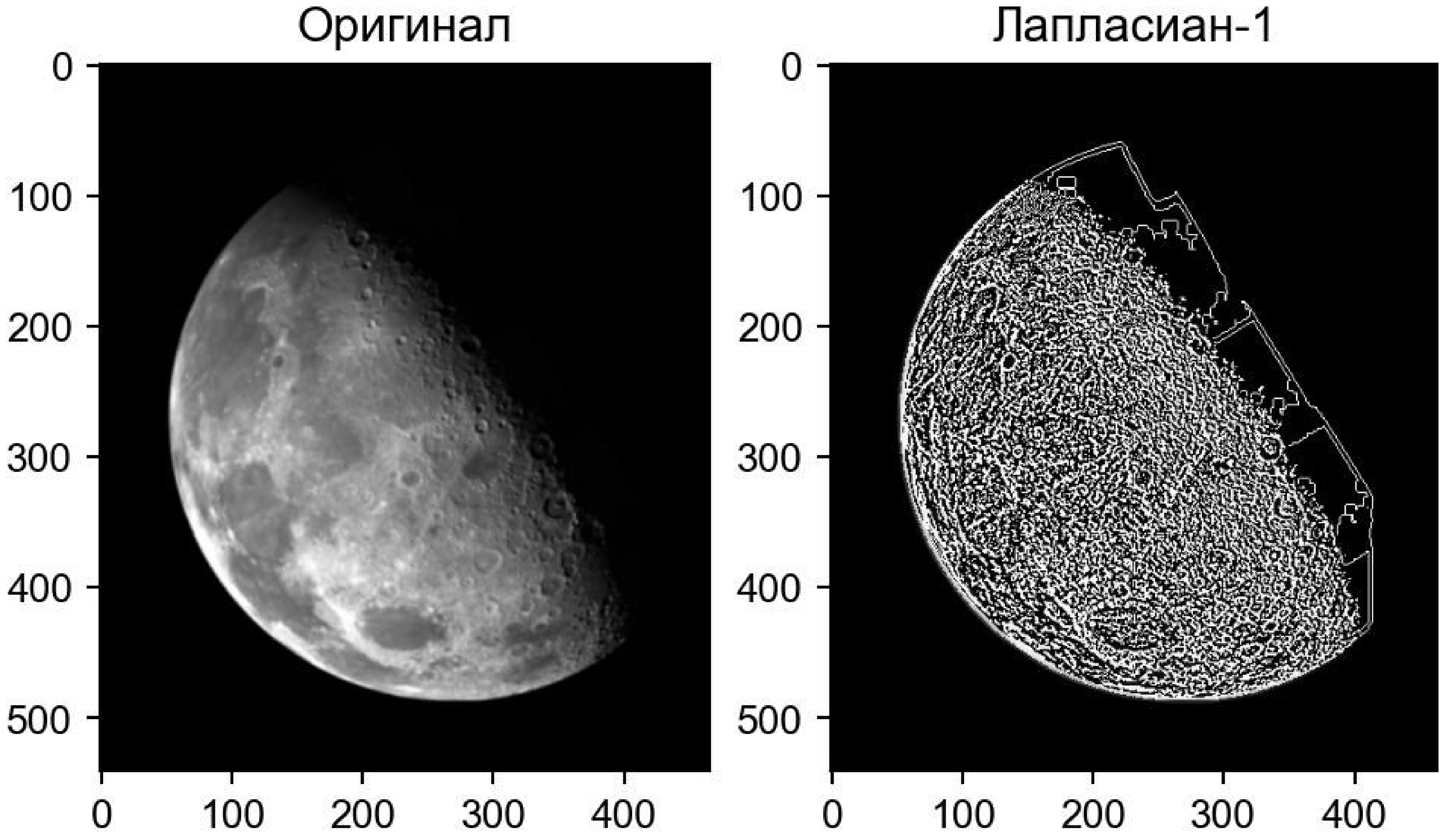
$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)].$$

$c \in [-1,\ 0]$ – отрицательная константа, определяющая степень повышения резкости.



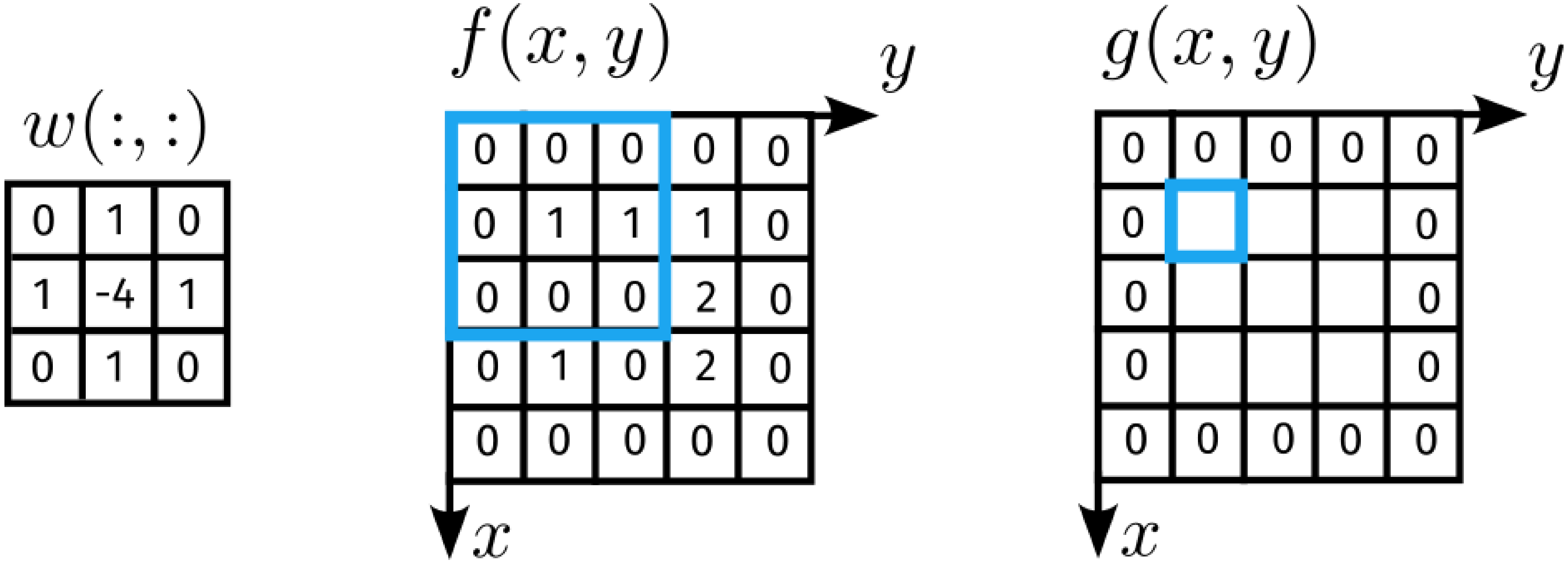
Особенности вычисления лапласиана

```
img_src = plt.imread("img\\blurry_moon.tif")
w1 = np.array([[0,1,0], [1,-4, 1],[0,1,0]])
img_laplase_1 = conv(img_src, w1)
```



Особенности вычисления лапласиана

Рассмотрим пример вычисления Лапласиана



$w(:,:)$

0	1	0
1	-4	1
0	1	0

$f(x,y)$

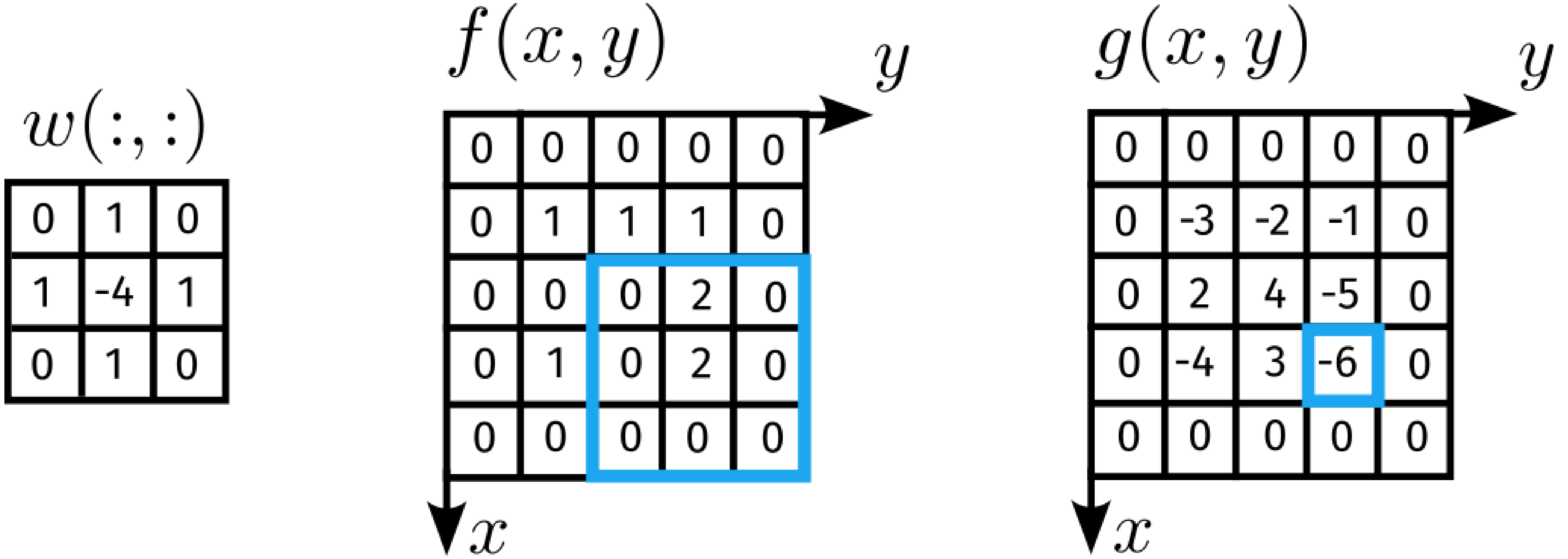
0	0	0	0	0
0	1	1	1	0
0	0	0	2	0
0	1	0	2	0
0	0	0	0	0

$g(x,y)$

0	0	0	0	0
0				0
0				0
0				0
0	0	0	0	0

Особенности вычисления лапласиана

Рассмотрим пример вычисления Лапласиана



$w(:,:)$

0	1	0
1	-4	1
0	1	0

$f(x,y)$

0	0	0	0	0
0	1	1	1	0
0	0	0	2	0
0	1	0	2	0
0	0	0	0	0

$g(x,y)$

0	0	0	0	0
0	-3	-2	-1	0
0	2	4	-5	0
0	-4	3	-6	0
0	0	0	0	0

Особенности вычисления лапласиана

Рассмотрим пример вычисления Лапласиана

$w(:,:)$

0	1	0
1	-4	1
0	1	0

$f(x,y)$ int

y

0	0	0	0	0
0	1	1	1	0
0	0	0	2	0
0	1	0	2	0
0	0	0	0	0

x

$g(x,y)$ int

y

0	0	0	0	0
0	-3	-2	-1	0
0	2	4	-5	0
0	-4	3	-6	0
0	0	0	0	0

x

$-1_{10}=FF_{16}$

Но, на самом деле...

Особенности вычисления лапласиана

Рассмотрим пример вычисления Лапласиана

$w(:,:)$

0	1	0
1	-4	1
0	1	0

$f(x,y)$ uint8

y

0	0	0	0	0
0	1	1	1	0
0	0	0	2	0
0	1	0	2	0
0	0	0	0	0

x

$g(x,y)$ uint8

y

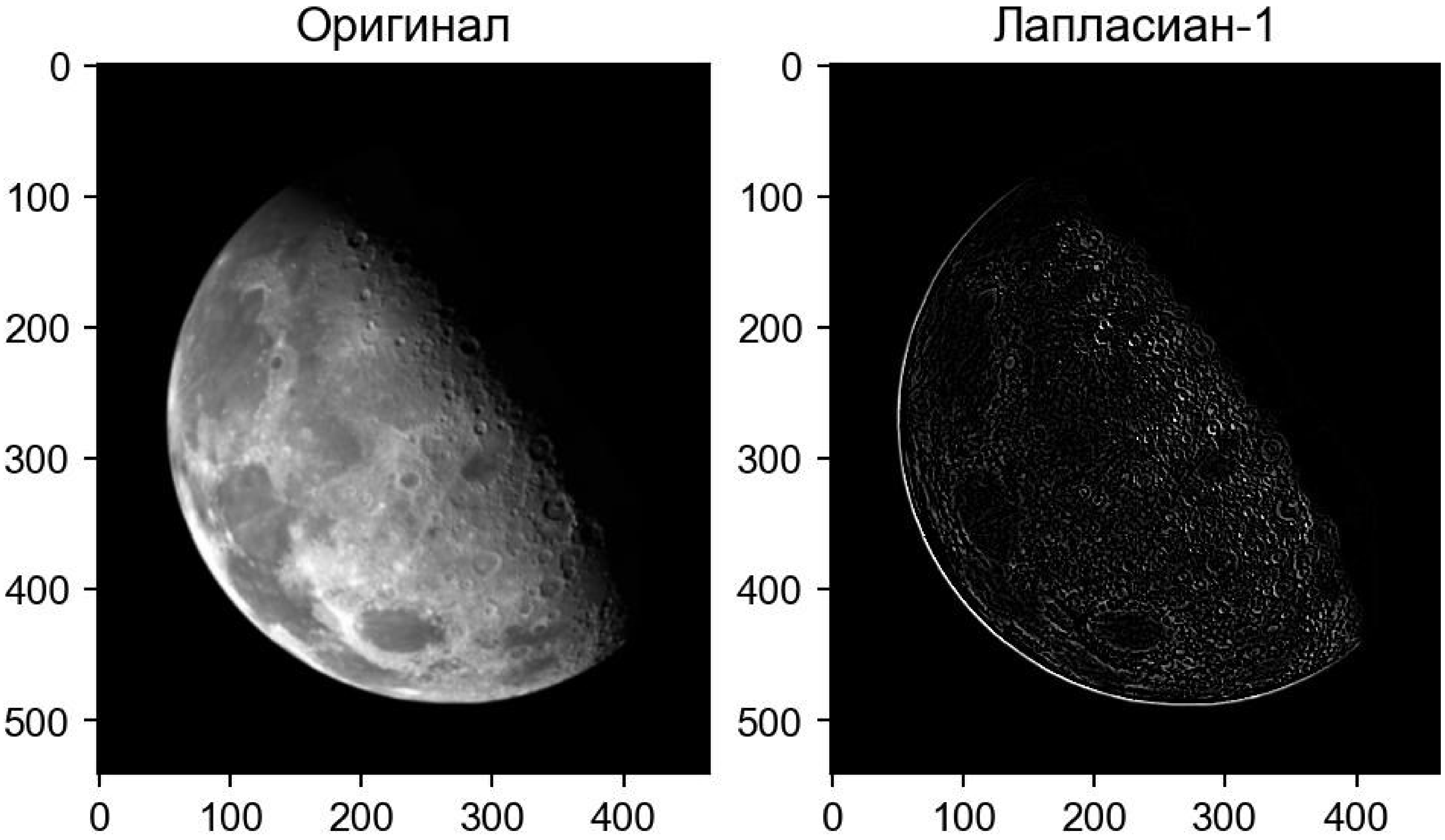
0	0	0	0	0
0	253	254	255	0
0	2	4	251	0
0	252	3	250	0
0	0	0	0	0

x

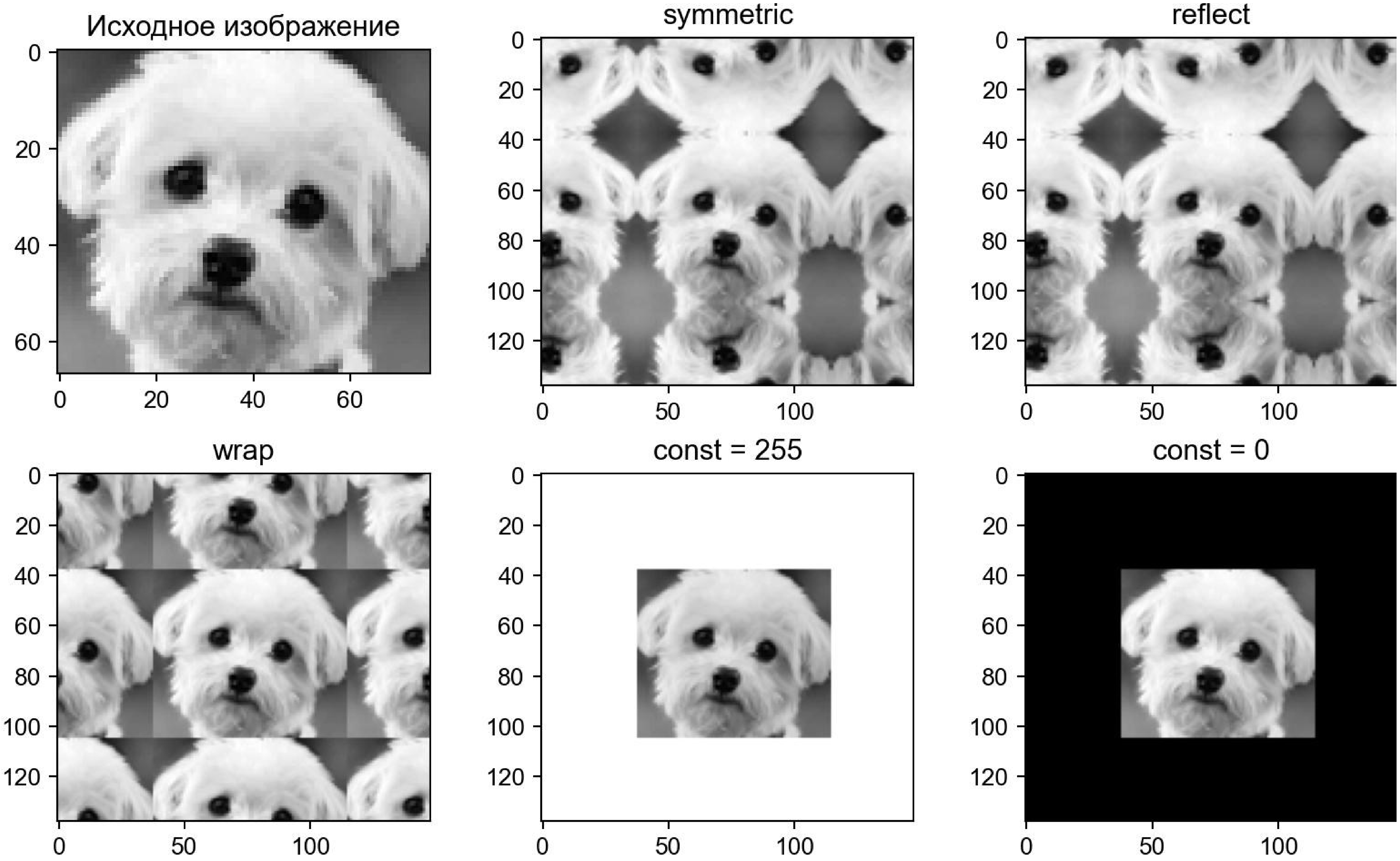
$FF_{16}=255_{10}$

Особенности вычисления лапласиана

```
img_src = plt.imread("img\\blurry_moon.tif").astype(np.float32)
w1 = np.array([[0,1,0], [1,-4, 1],[0,1,0]])
img_laplase_1 = conv(img_src, w1)
```



Расширение изображения



Расширение изображения

```
img_padded_const_255 = np.pad(img, (w//2,h//2),'constant',
constant_values=(255, 255))


img_padded_const_0 = np.pad(img, (w//2,h//2),'constant',
constant_values=(0, 0))


img_padded_sym = np.pad(img, ((w//2,h//2),(w//2,h//2)),'symmetric')


img_padded_ref = np.pad(img, (w//2,h//2),'reflect')


img_padded_wrap = np.pad(img, (w//2,h//2),'wrap')
```

Взвешенное среднее

$$\mathbf{w} = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Задача 2

Выполнить фильтрацию монохромного изображения $f(x, y)$, взвешенным усредняющим фильтром $\mathbf{w}$. Перед обработкой изображения выполнить расширение путем зеркального отображения.

Ответ к задаче 2

$d(x, y) =$

```
[[2.000  2.125  2.1250  2.000]
 [2.500  3.000  3.0625  2.625]
 [2.875  3.625  3.6875  3.000]
 [2.750  3.375  3.3750  2.750]]
```