

Компьютерное Зрение
Лекция №2, осень 2022

Обработка сигналов



Кафедра
технологий
проектирования
сложных
технических
систем

Мотивация к обработке изображений

De-noising



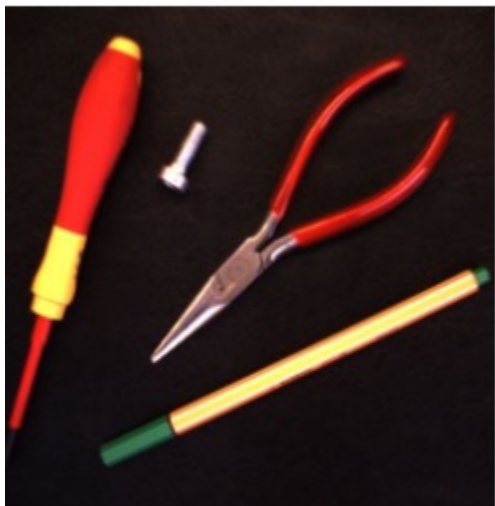
Super-resolution



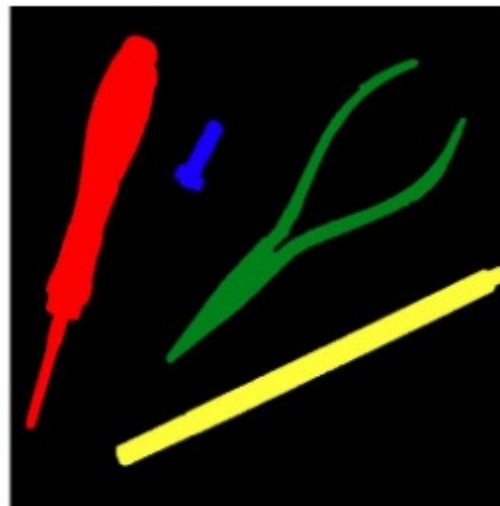
In-painting



Мотивация к обработке изображений



Бинаризация



Выделение
компонент
связности



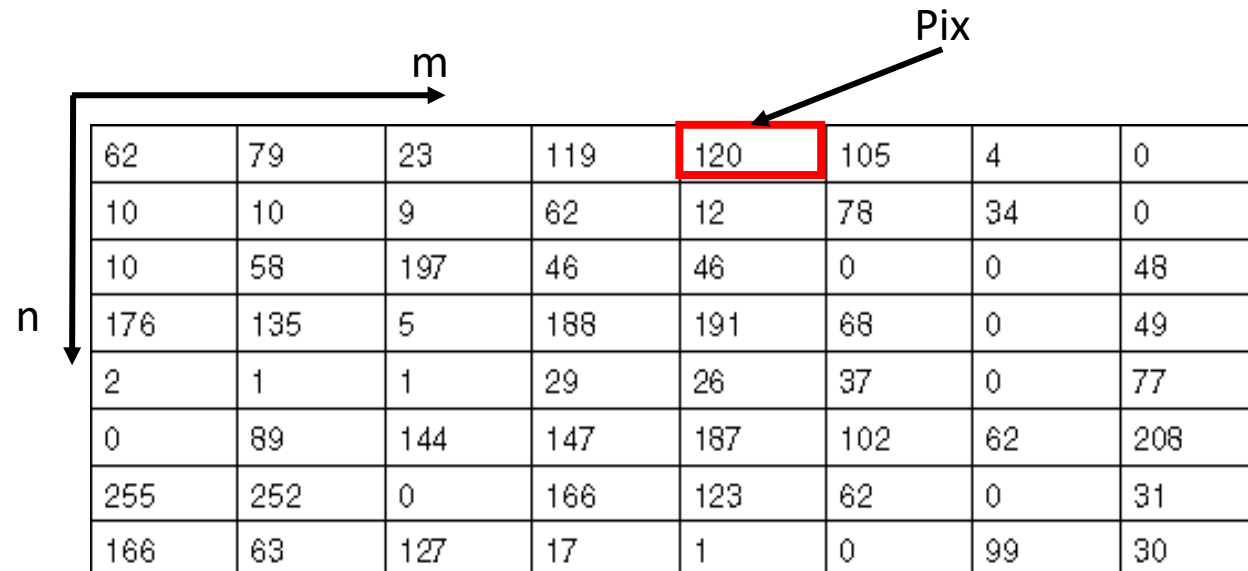
Выделение
краев

План лекции

- Представление изображения в частотной области.
Преобразование Фурье
- Системы и фильтры
- Свертки

Изображение как дискретная функция

- Изображения обычно цифровые (дискретные):
 - Пример 2D пространства на регулярной сетке
- Представлено в виде матрицы целочисленных значений



The diagram illustrates a 2D pixel grid. A horizontal arrow labeled 'm' points to the right, indicating the column index. A vertical arrow labeled 'n' points downwards, indicating the row index. The grid consists of 8 rows and 8 columns of numerical values. The value 120, located at row 1 and column 5, is highlighted with a red border. An arrow labeled 'Pix' points to this highlighted cell.

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Изображение как дискретная функция

Декартовы координаты

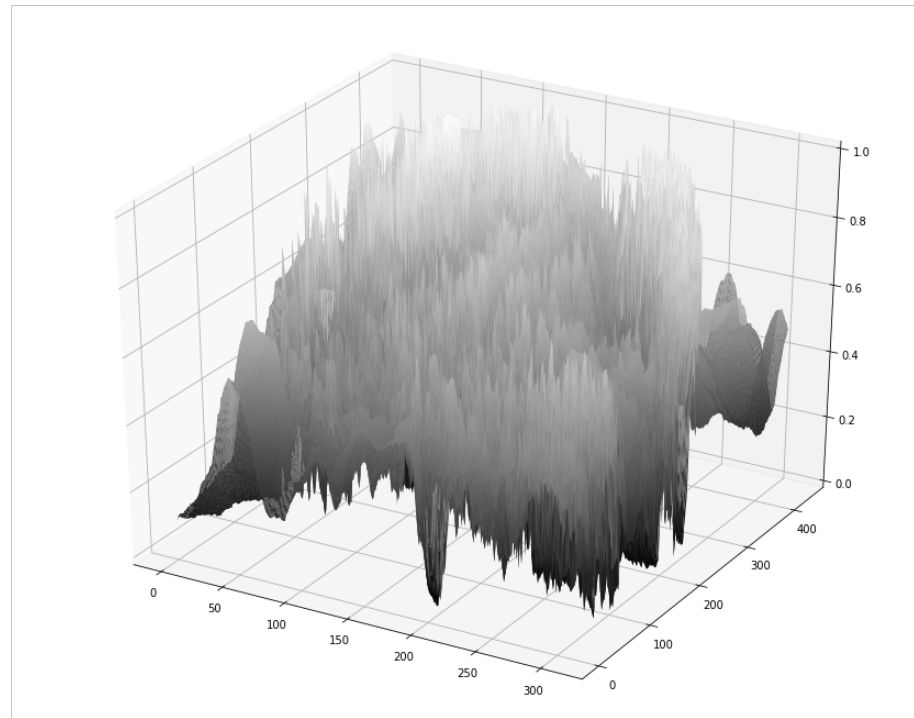
$$f[n, m] = \begin{bmatrix} \ddots & & \vdots & & \\ & f[-1, 1] & f[0, 1] & f[1, 1] & \\ \dots & f[-1, 0] & \underline{f[0, 0]} & f[1, 0] & \dots \\ & f[-1, -1] & f[0, -1] & f[1, -1] & \\ & & \vdots & & \ddots \end{bmatrix}$$

Изображение как дискретная функция

Изображение как функция f от \mathbb{R}^2 до \mathbb{R}^M :

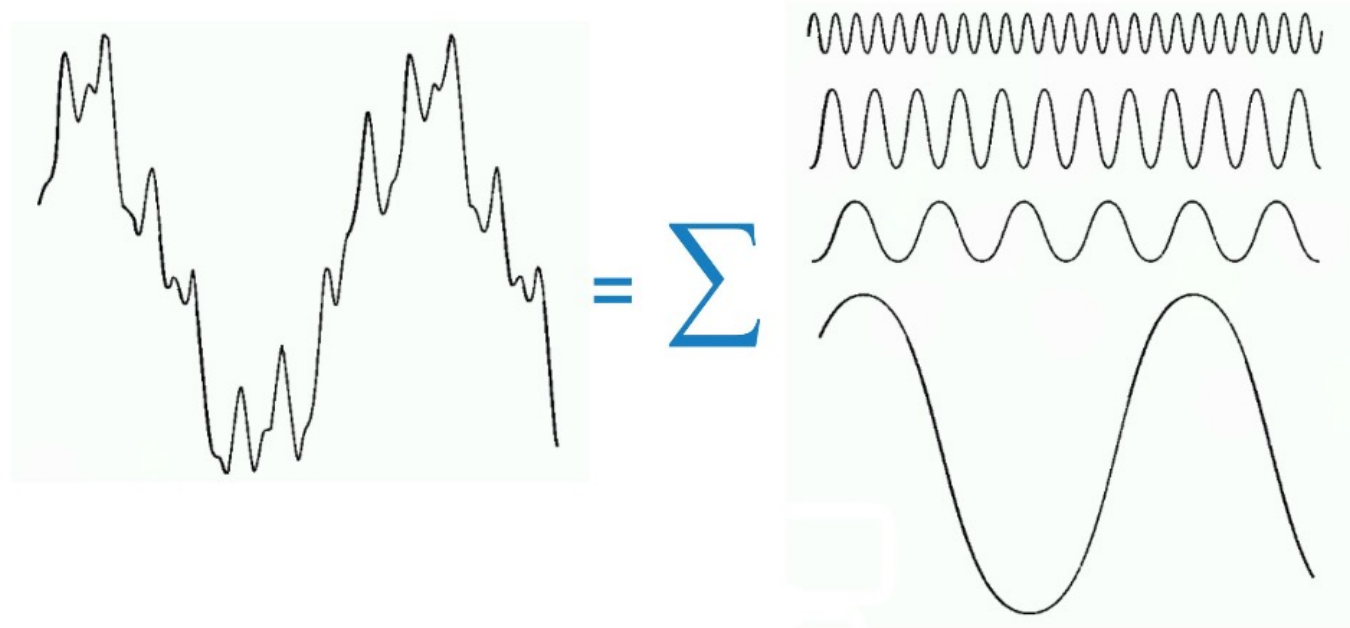
- $f(x, y)$ дает интенсивность в позиции (x, y)
- Определяется через прямоугольник, с конечным диапазоном:

$$f: [a, b] \times [c, d] \rightarrow [0, 255]$$



Ряд Фурье

Периодический сигнал может быть представлен в виде суммы



Ряд и преобразование Фурье

Ряд Фурье — представление функции f с периодом τ в виде ряда

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} A_k \cos\left(k \frac{2\pi}{\tau} x + \theta_k\right)$$

Этот ряд может быть также записан в виде

$$f(x) = \sum_{k=-\infty}^{+\infty} \hat{f}_k e^{ik \frac{2\pi}{\tau} x},$$

где

A_k — амплитуда k -го гармонического колебания,

$k \frac{2\pi}{\tau} = k\omega$ — круговая частота гармонического колебания,

θ_k — начальная фаза k -го колебания,

\hat{f}_k — k -я комплексная амплитуда

Преобразование Фурье

Прямое

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\omega} dx.$$

Обратное

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{ix\omega} d\omega$$

Преобразование Фурье для двумерного случая

Прямое
преобразование

$$F(k, l) = \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} f(p, q) e^{-2i\pi(\frac{kp}{N} + \frac{lq}{N})}$$

Обратное
преобразование

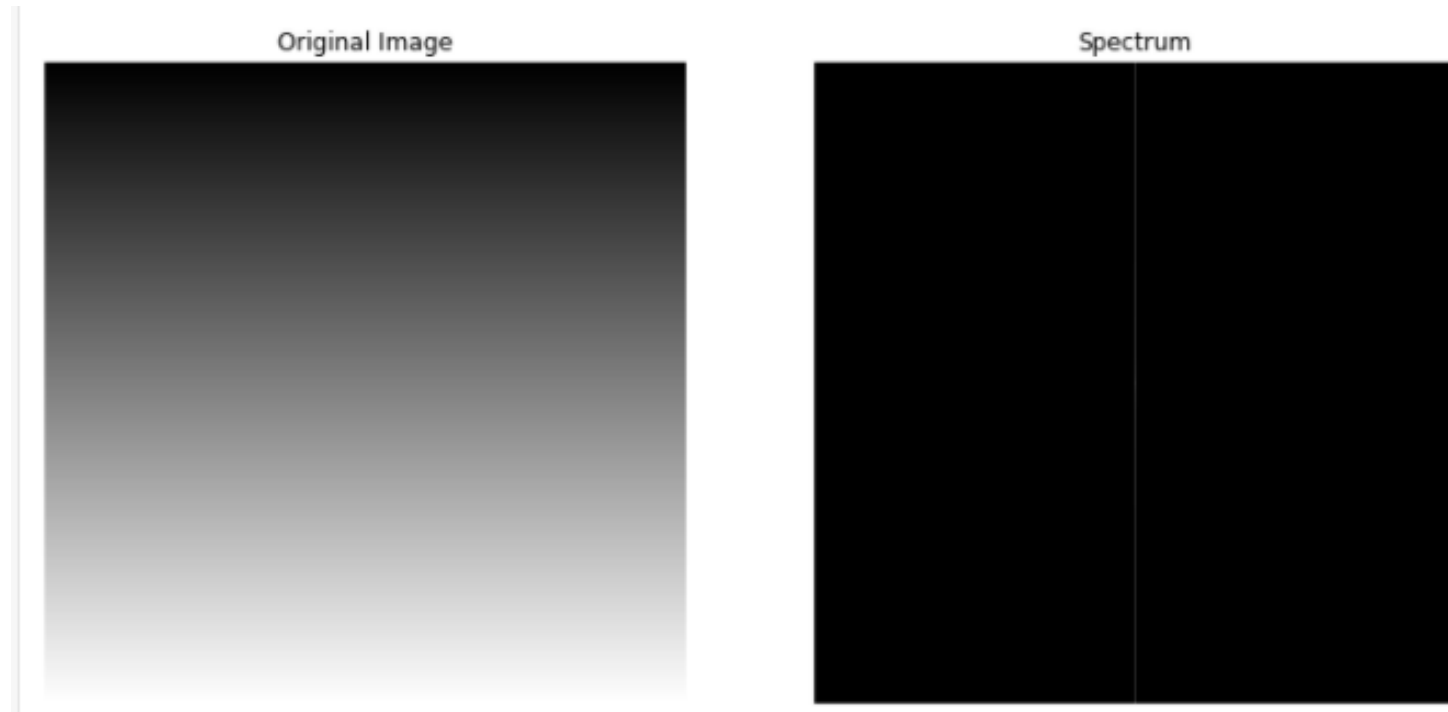
$$f(p, q) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{-2i\pi(\frac{kp}{N} + \frac{lq}{N})}$$

Преобразования Фурье для изображений



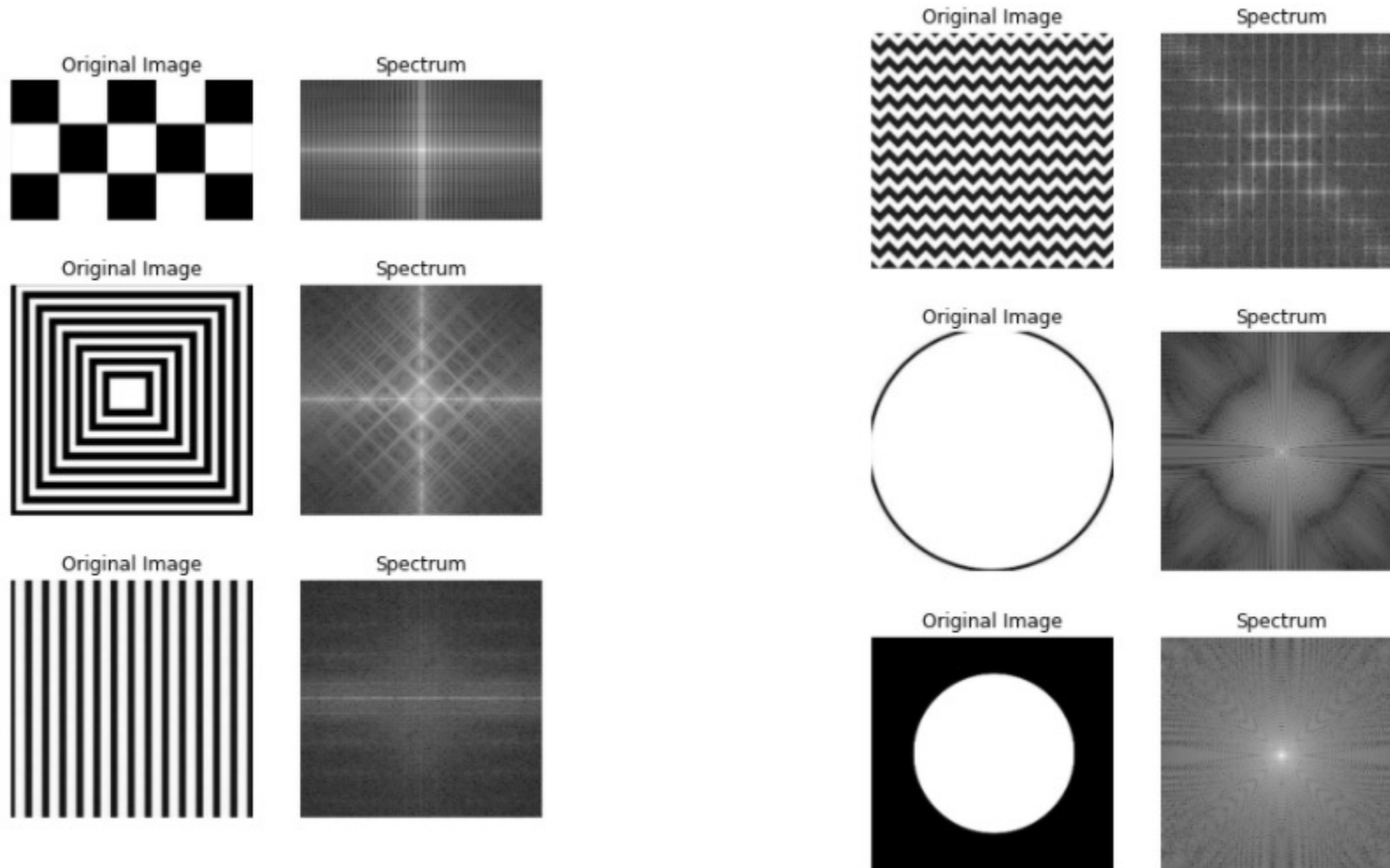
«**Высокие**» частоты: область с сильными и частыми перепадами значений пикселей

Преобразования Фурье для изображений



«Низкие» частоты: области с слабыми и редкими перепадами значений пикселей

Интерпретация спектра изображения



План лекции

- Представление изображения в частотной области.
Преобразование Фурье
- **Системы и фильтры**
- Свертки

Системы и фильтры

Фильтрация – формирование нового изображения, значения пикселей которого трансформируются из исходных значений пикселей.

Мотивация:

- Выделить полезную информацию
- Изменить или улучшить свойства полезных признаков на изображении

Интуитивное понимание систем

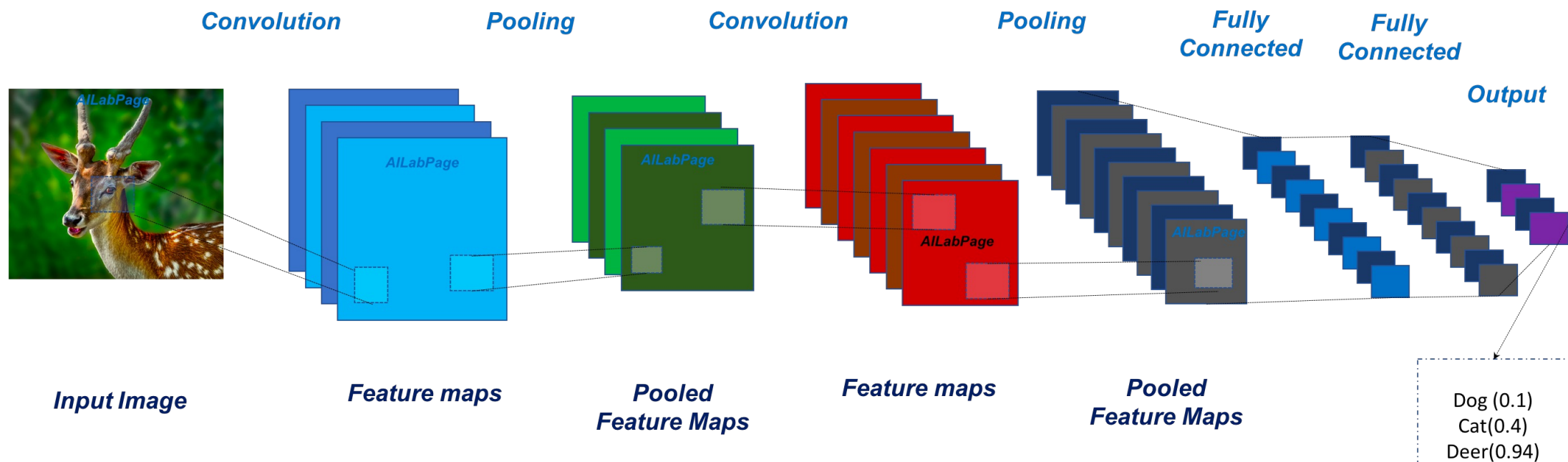
Мы рассмотрим линейные системы как вид функции, которая применяется к изображениям, как двумерным функциям.

Преобразование изображения или его умножение на константу оставляет семантическое содержание нетронутым – можно выделить некоторые закономерности.



Кстати говоря...

Нейронные сети и, в частности, сверточные нейронные сети – это тип системы или нелинейная система, содержащая несколько отдельных линейных подсистем.



(подробнее об этом в другом курсе)

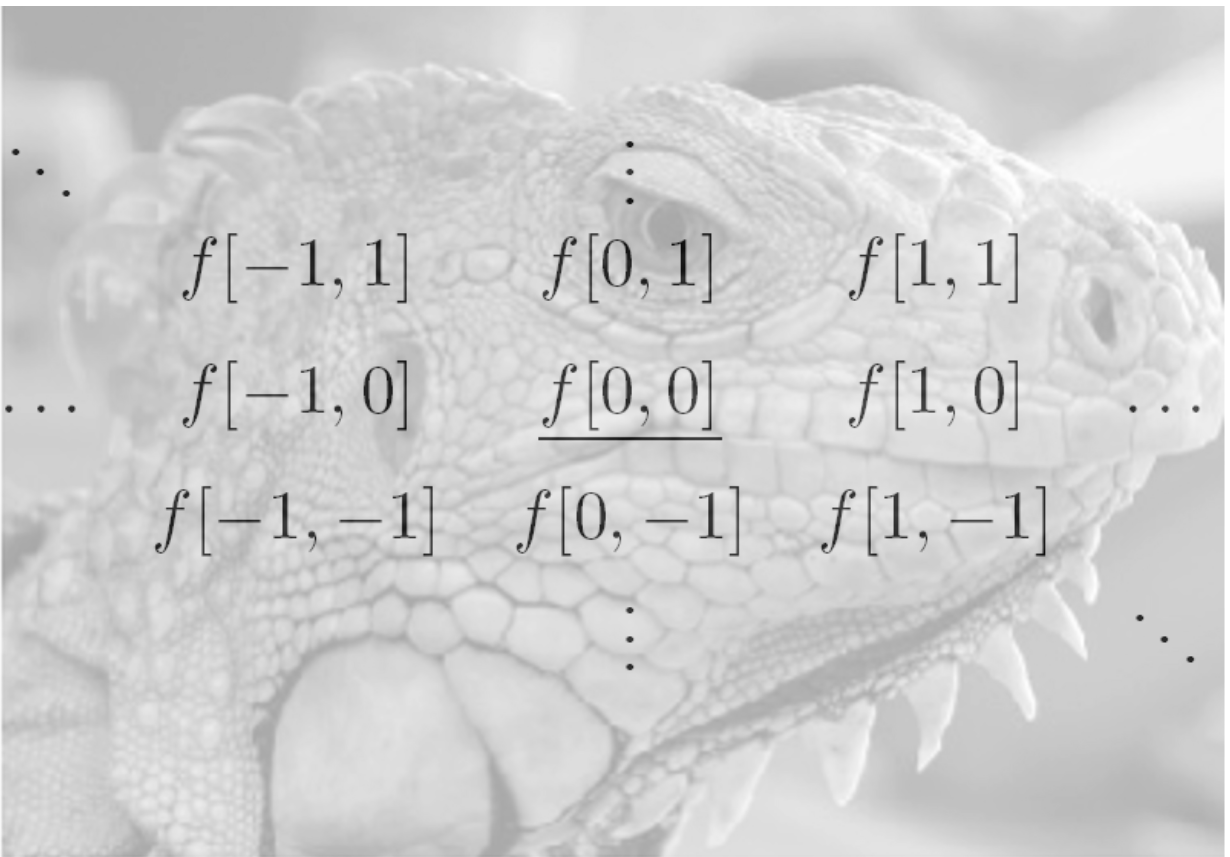
Системы и фильтры

Определим **систему** как единицу, которая преобразует входную функцию $f[n, m]$ в выходную (или ответную) функцию $g[n, m]$, где (n, m) являются независимыми переменными.

В случае изображений (n, m) представляет пространственное положение на изображении.

$$f[n, m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n, m]$$

Изображение как дискретная функция

$$f[n, m] = \left[\begin{array}{ccccc} \ddots & & \vdots & & \\ & f[-1, 1] & f[0, 1] & f[1, 1] & \\ \dots & f[-1, 0] & \underline{f[0, 0]} & f[1, 0] & \dots \\ & f[-1, -1] & f[0, -1] & f[1, -1] & \\ & & \vdots & & \ddots \end{array} \right]$$


Пример фильтра №1: Размытие

Original image



Smoothed image



Пример фильтра №1: Размытие

2D DS moving average over a 3×3 window of neighborhood

$$g[n, m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k, l]$$
$$= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 f[n - k, m - l]$$

h

1	1	1
1	1	1
1	1	1

$\frac{1}{9}$

Пример фильтра №1: Размытие

$f[n, m]$

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

$g[n, m]$

	0								

Пример фильтра №1: Размытие

$f[n, m]$

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

$g[n, m]$

	0	10							

Пример фильтра №1: Размытие

$f[n, m]$

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

$g[n, m]$

	0	10	20						

Пример фильтра №1: Размытие

$f[n, m]$

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

$g[n, m]$

	0	10	20	30					

Пример фильтра №1: Размытие

$f[n, m]$

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

$g[n, m]$

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

Пример фильтра №1: Размытие

Подводя итог:

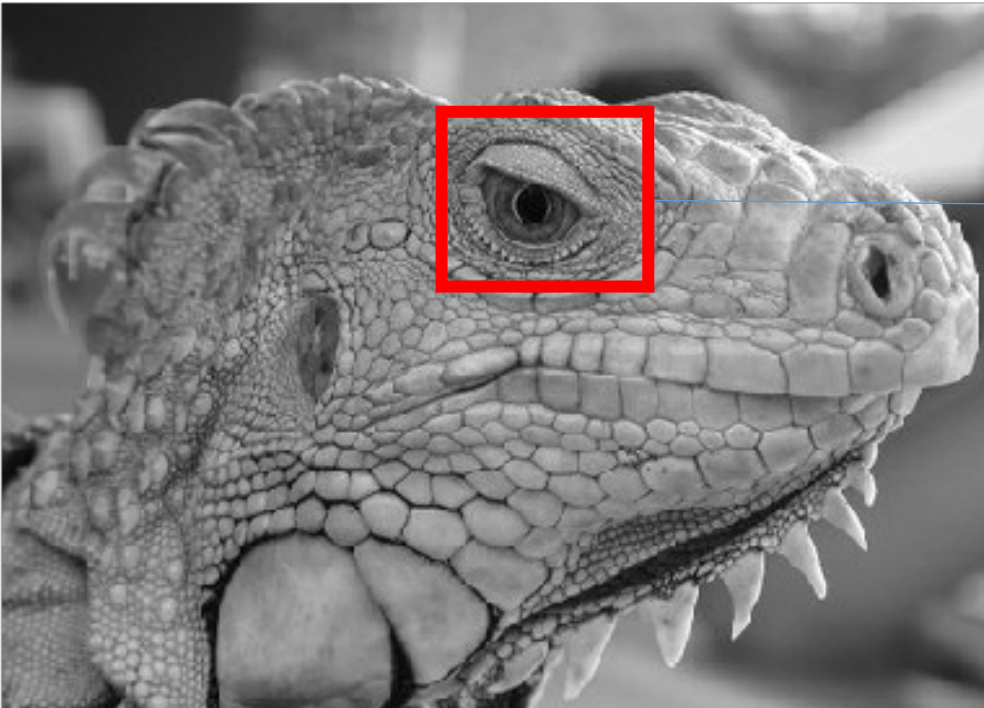
- Данный фильтр "Заменяет" каждый пиксель средним значением по окрестностям.
- Достигается эффект сглаживания (осреднение резких переходов значений пикселей).

h

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

Пример фильтра №1: Размытие

Original image



Smoothed image



Пример фильтра №2: Пороговое правило

$$g[n, m] = \begin{cases} 1, & f[n, m] > 100 \\ 0, & \text{otherwise.} \end{cases}$$



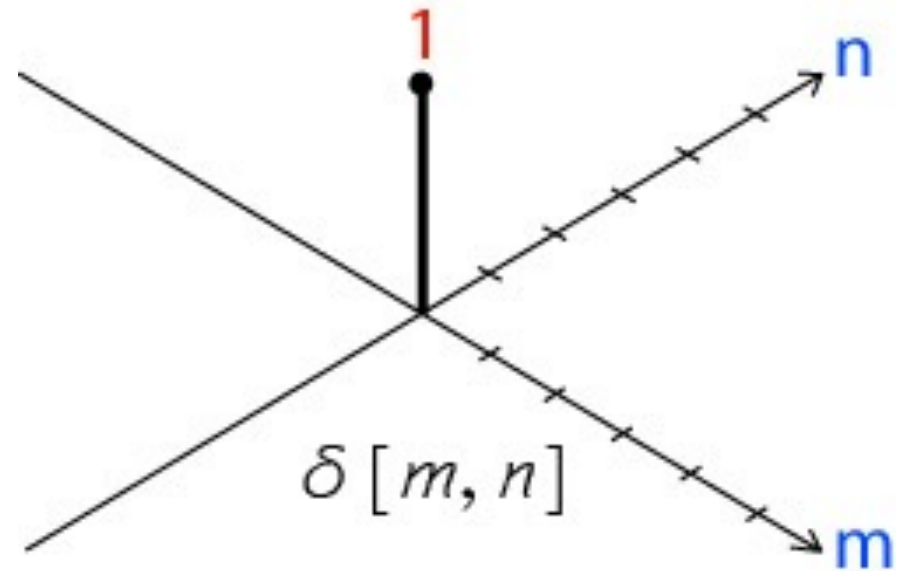
План лекции

- Представление изображения в частотной области.
Преобразование Фурье
- Системы и фильтры
- **Свертки**

Импульсная функция

Рассмотрим специальную функцию:

- равна 1, в точке $[0,0]$.
- равна 0, во всех остальных точках



Импульсный отклик от фильтра размытия

		?		
		h[0,0]		

$$\begin{aligned}\delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Импульсный отклик от фильтра размытия

		$\frac{1}{9}$ $h[0,0]$	$?$ $h[0,1]$	

$$\begin{aligned}\delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Импульсный отклик от фильтра размытия

		$\frac{1}{9}$ $h[0,0]$	$\frac{1}{9}$ $h[0,1]$	
			? $h[1,1]$	

$$\begin{aligned}\delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Импульсный отклик от фильтра размытия

		$\frac{1}{9}$ $h[0,0]$	$\frac{1}{9}$ $h[0,1]$	
			$\frac{1}{9}$ $h[1,1]$	

$$\begin{aligned}\delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Импульсный отклик от фильтра размытия

		$\frac{1}{9}$ $h[0,0]$	$\frac{1}{9}$ $h[0,1]$? $h[0,2]$
			$\frac{1}{9}$ $h[1,1]$	

$$\begin{aligned}\delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Импульсный отклик от фильтра размытия

		$\frac{1}{9}$ $h[0,0]$	$\frac{1}{9}$ $h[0,1]$	0 $h[0,2]$
			$\frac{1}{9}$ $h[1,1]$	

$$\begin{aligned} \delta_2 &\xrightarrow{s} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l] \end{aligned}$$

Импульсный отклик от фильтра размытия

0	0	0	0	0
0	$\frac{1}{9}$ $h[-1,-1]$	$\frac{1}{9}$	$\frac{1}{9}$	0
0	$\frac{1}{9}$	$\frac{1}{9}$ $h[0,0]$	$\frac{1}{9}$ $h[0,1]$	0 $h[0,2]$
0	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$ $h[1,1]$	0
0	0	0	0	0

$$\begin{aligned} \delta_2 &\xrightarrow{s} g[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l] \end{aligned}$$

Фильтр размытия через импульсные функции

$$h[n, m] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

$$= \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

h

1	1	1
1	1	1
1	1	1

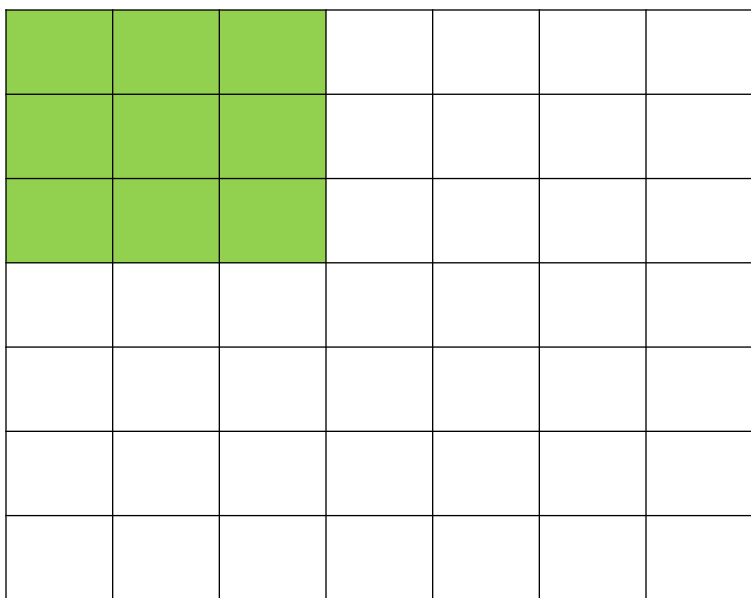
$\frac{1}{9}$

Двумерная свертка

2D свёртка очень похожа на 1D.

Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



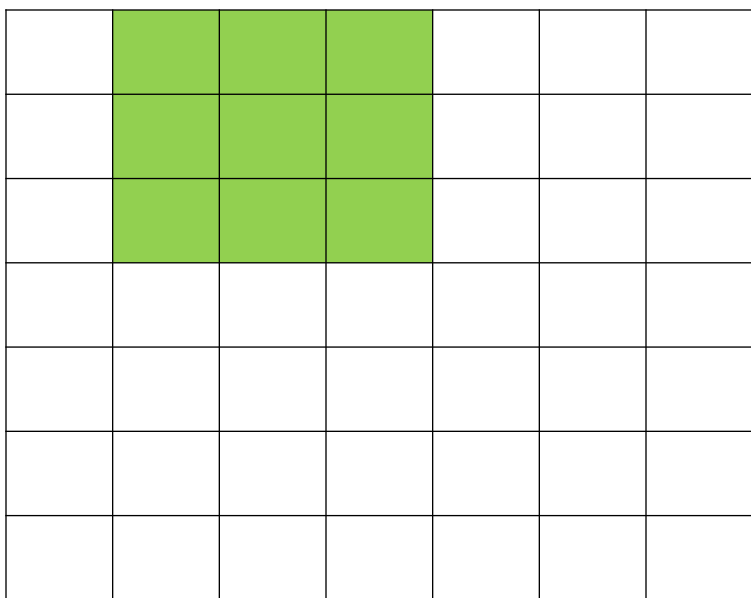
Предположим, что у нас есть фильтр($h[,]$) размером 3x3 и изображение ($f[,]$) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.

Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



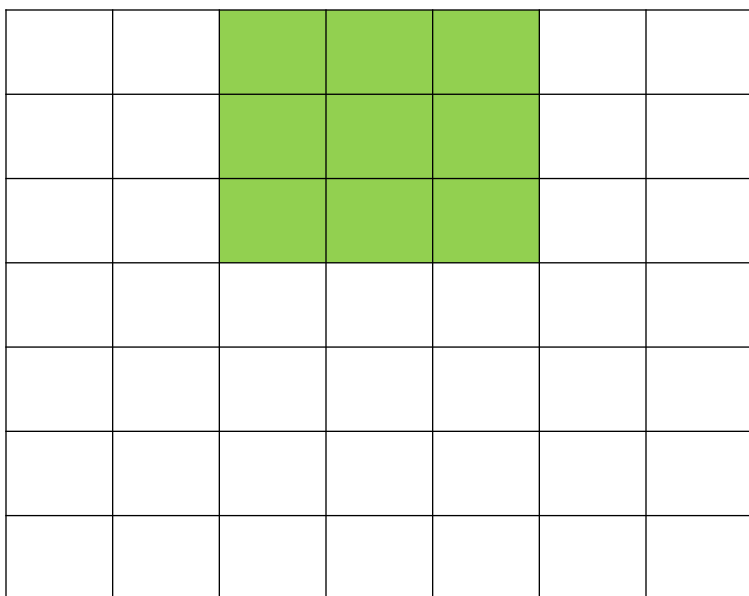
Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.

Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



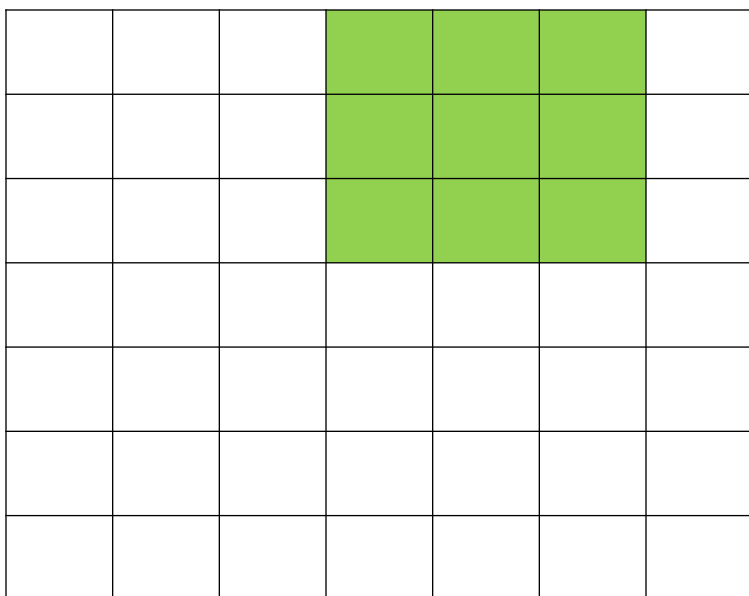
Предположим, что у нас есть фильтр($h[,]$) размером 3x3 и изображение ($f[,]$) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.

Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



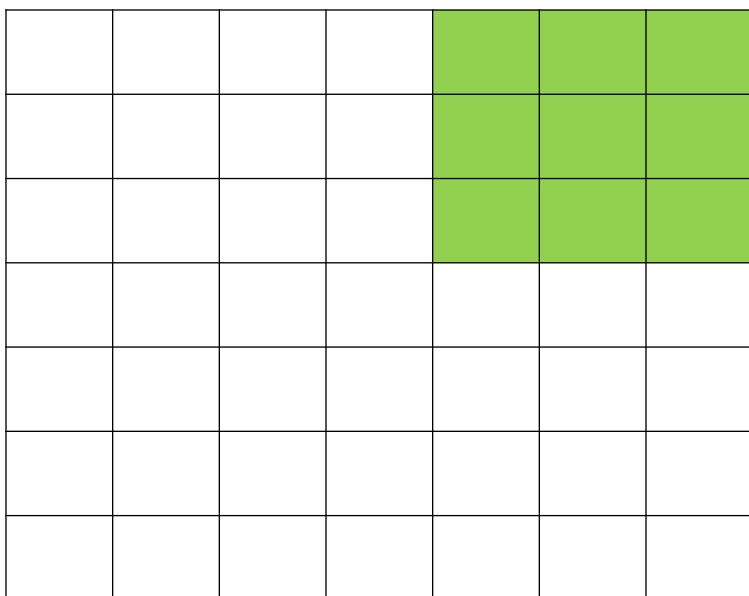
Предположим, что у нас есть фильтр($h[,]$) размером 3x3 и изображение ($f[,]$) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.

Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



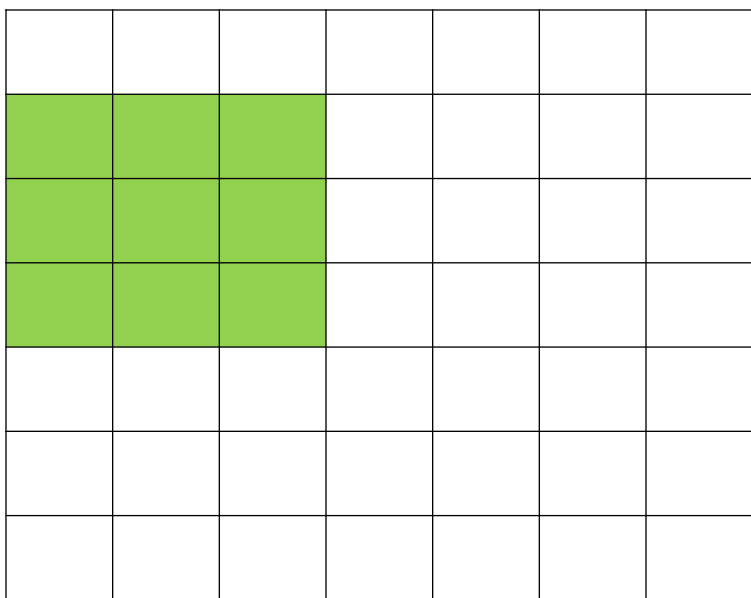
Предположим, что у нас есть фильтр($h[,]$) размером 3x3 и изображение ($f[,]$) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.

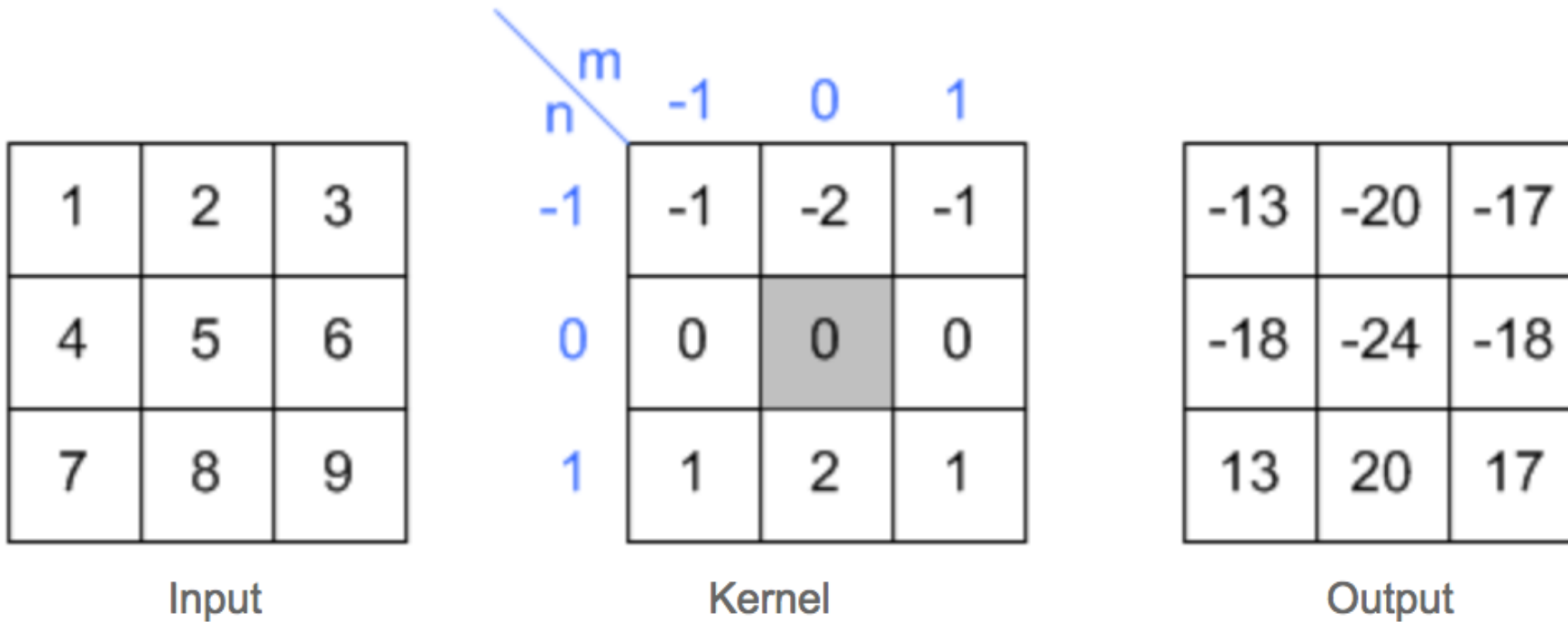
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



Предположим, что у нас есть фильтр($h[.,.]$) размером 3x3 и изображение ($f[.,.]$) размером 7x7.

Пример двумерной свертки



Пример двумерной свертки

1	2	1	
0	0	0	3
-1	-2	-1	6
	7	8	9

$$\begin{aligned}y[0,0] &= x[-1,-1] \cdot h[1,1] + x[0,-1] \cdot h[0,1] + x[1,-1] \cdot h[-1,1] \\&\quad + x[-1,0] \cdot h[1,0] + x[0,0] \cdot h[0,0] + x[1,0] \cdot h[-1,0] \\&\quad + x[-1,1] \cdot h[1,-1] + x[0,1] \cdot h[0,-1] + x[1,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot (-1) + 4 \cdot (-2) + 5 \cdot (-1) = -13\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Пример двумерной свертки

1	2	1
0	0	0
-1	-2	-1
7	8	9

$$\begin{aligned}y[1,0] &= x[0,-1] \cdot h[1,1] + x[1,-1] \cdot h[0,1] + x[2,-1] \cdot h[-1,1] \\&\quad + x[0,0] \cdot h[1,0] + x[1,0] \cdot h[0,0] + x[2,0] \cdot h[-1,0] \\&\quad + x[0,1] \cdot h[1,-1] + x[1,1] \cdot h[0,-1] + x[2,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 0 + 4 \cdot (-1) + 5 \cdot (-2) + 6 \cdot (-1) = -20\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Пример двумерной свертки

		1	2	1
	0	2	0	0
1			3	
	-1	5	-2	-1
4			6	
	7	8	9	

$$\begin{aligned}y[2,0] &= x[1,-1] \cdot h[1,1] + x[2,-1] \cdot h[0,1] + x[3,-1] \cdot h[-1,1] \\&\quad + x[1,0] \cdot h[1,0] + x[2,0] \cdot h[0,0] + x[3,0] \cdot h[-1,0] \\&\quad + x[1,1] \cdot h[1,-1] + x[2,1] \cdot h[0,-1] + x[3,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 2 \cdot 0 + 3 \cdot 0 + 0 \cdot 0 + 5 \cdot (-1) + 6 \cdot (-2) + 0 \cdot (-1) = -17\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Пример двумерной свертки

1	2	1	3
0	0	0	6
-1	-2	-1	9

$$\begin{aligned}y[0,1] &= x[-1,0] \cdot h[1,1] + x[0,0] \cdot h[0,1] + x[1,0] \cdot h[-1,1] \\&\quad + x[-1,1] \cdot h[1,0] + x[0,1] \cdot h[0,0] + x[1,1] \cdot h[-1,0] \\&\quad + x[-1,2] \cdot h[1,-1] + x[0,2] \cdot h[0,-1] + x[1,2] \cdot h[-1,-1] \\&= 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 + 0 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 + 0 \cdot (-1) + 7 \cdot (-2) + 8 \cdot (-1) = -18\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Пример двумерной свертки





*

•0	•0	•0
•0	•1	•0
•0	•0	•0

=

?

Пример двумерной свертки


$$* \begin{array}{|c|c|c|} \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \bullet 0 & \bullet 1 & \bullet 0 \\ \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \end{array} =$$


The diagram illustrates a 2D convolution operation. On the left is a 3x3 grayscale image of a snake's eye. This is followed by a convolution kernel, represented by a 3x3 grid of orange cells. The kernel contains the values: top row [•0, •0, •0], middle row [•0, •1, •0], and bottom row [•0, •0, •0]. The kernel is multiplied by the input image, as indicated by the asterisk symbol. The result is shown on the right, separated by an equals sign, as a 3x3 grayscale image of a snake's eye that is identical to the input image.

Пример двумерной свертки




*

•0	•0	•0
•0	•0	•1
•0	•0	•0


=

?

Пример двумерной свертки


$$\ast$$

•0	•0	•0
•0	•0	•1
•0	•0	•0

$$=$$


Пример двумерной свертки



*

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

=

?

Пример двумерной свертки

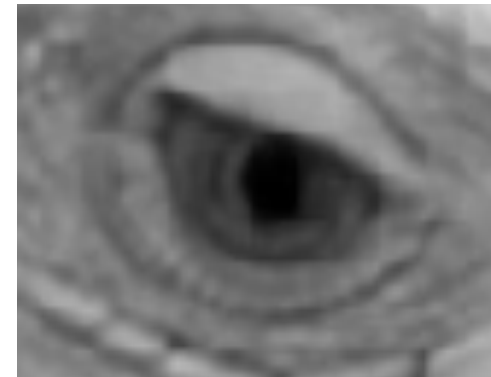


*

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

=



Пример двумерной свертки



•0	•0	•0
•0	•2	•0
•0	•0	•0

-

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

= ?

•0	•0	•0
•0	•1	•0
•0	•0	•0

+

•0	•0	•0
•0	•1	•0
•0	•0	•0

-

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

Что отнимает размытость?



Оригинальное
изображение

-



Размытое

=



Детали



Оригинальное
изображение

+




Детали

=



Повышенная резкость

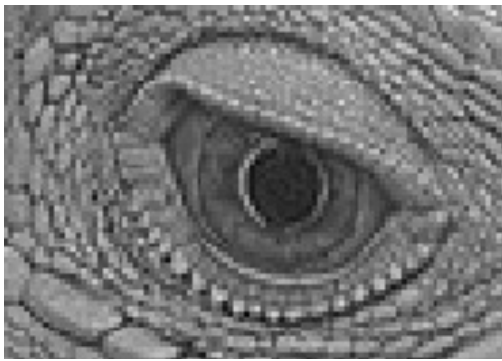
Пример двумерной свертки – фильтр резкости



•0	•0	•0
•0	•2	•0
•0	•0	•0

 $-$

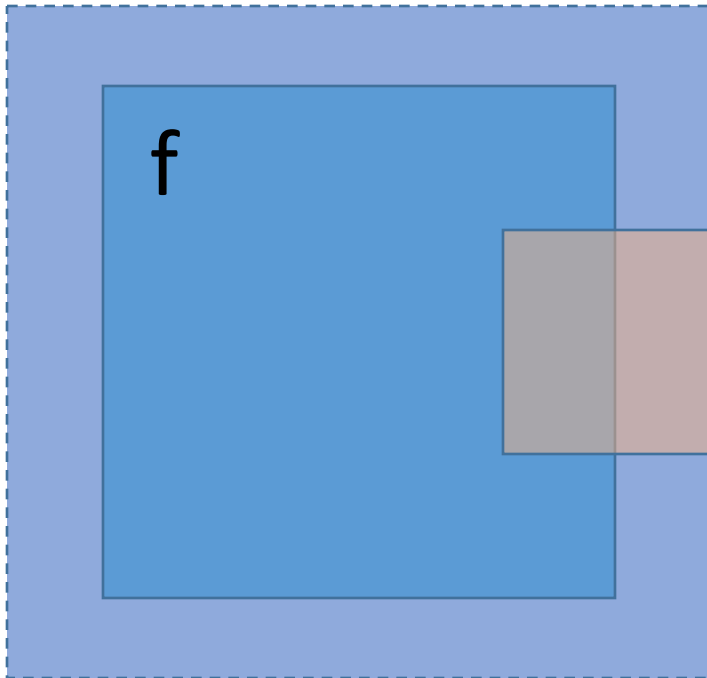
•1	•1	•1
•1	•1	•1
•1	•1	•1

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

Фильтр резкости: подчеркивает разность со средним местным значениями пикселей

Краевой эффект

- Компьютер будет вызывать только **конечные сигналы**.
- Что происходит на краю?



h

- нулевой паддинг
- повторение на краях
- отзеркаливание

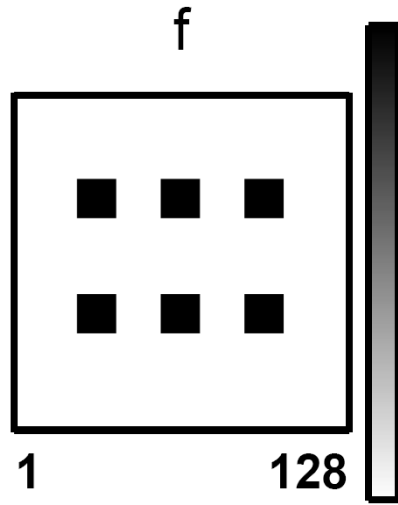
Кросс-корреляция

Кросс-корреляция двух 2D сигналов $f[n,m]$ и $h[n,m]$.

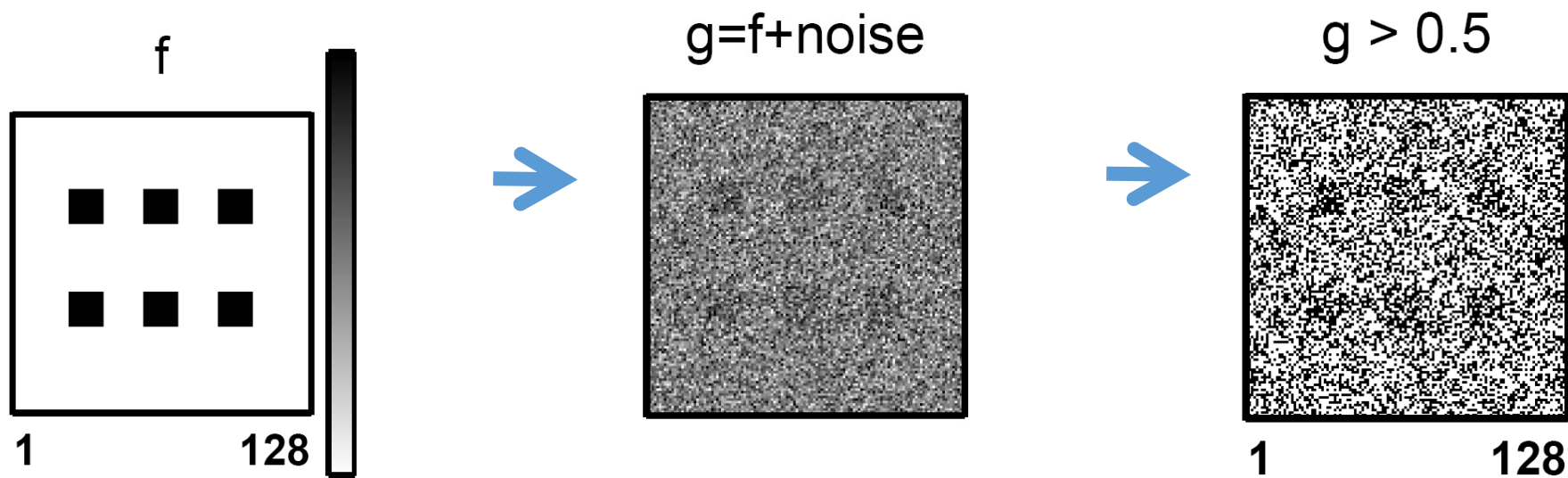
$$f[n, m] * h[n, m] = \sum_k \sum_l f[k, l] h[n - k, m - l]$$

- Эквивалент свертывания без переворачивания
- Измерения "сходства" между f и h .

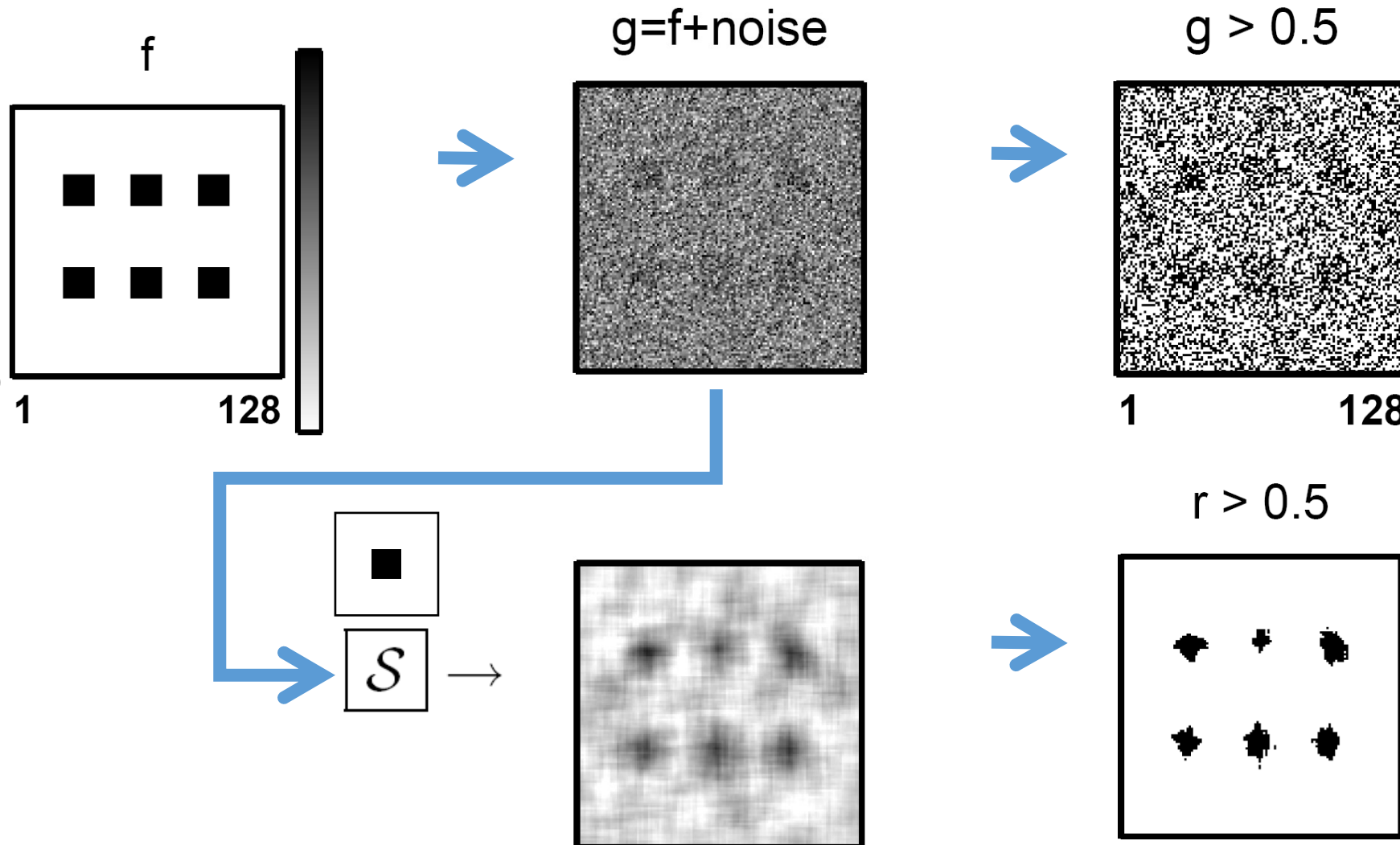
Пример кросс-корреляции



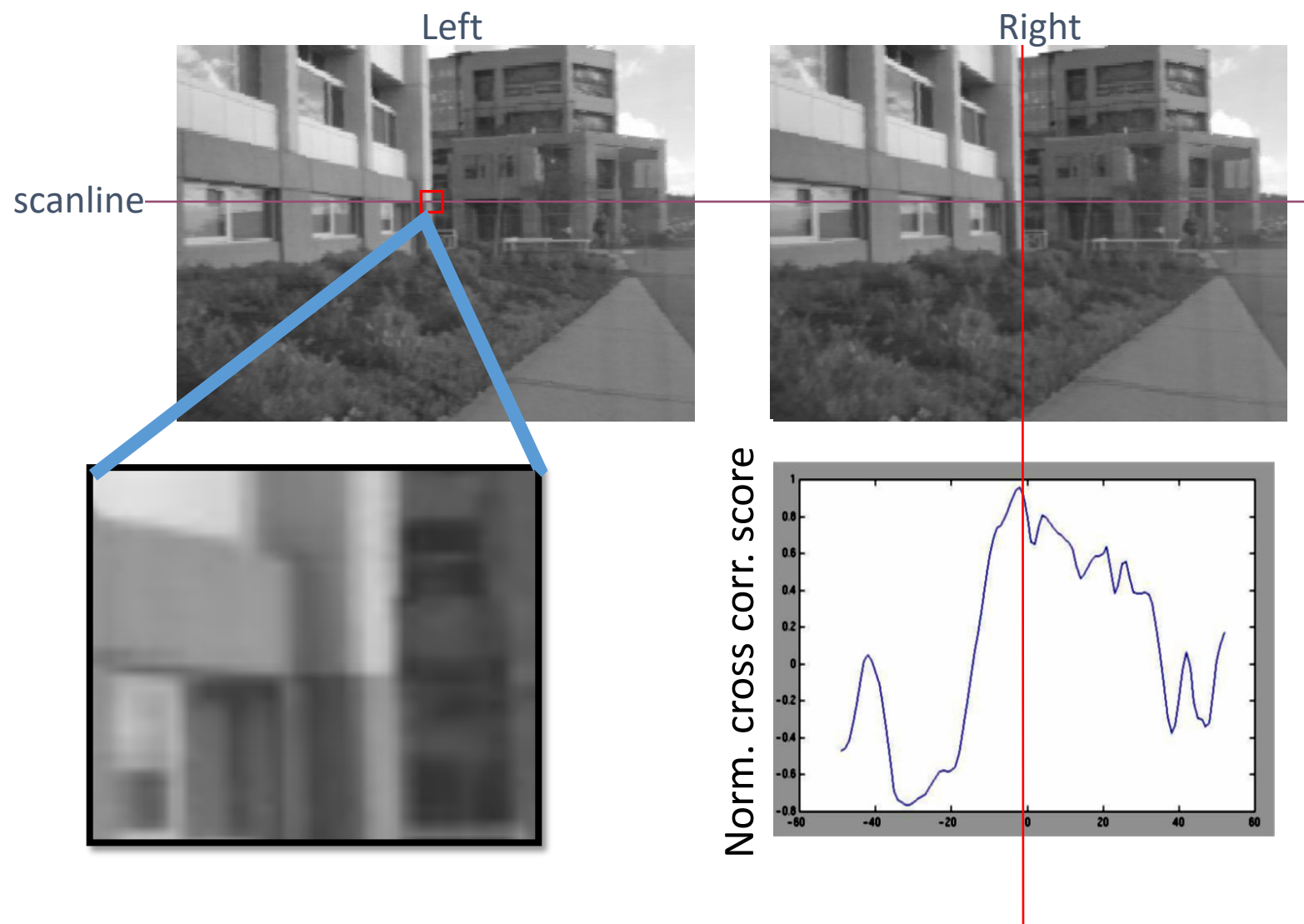
Пример кросс-корреляции



Пример кросс-корреляции



Пример кросс-корреляции



Свертка vs кросс-корреляция

- **Свертка** – это интеграл, выражающий величину перекрытия одной функции при ее смещении по другой.
 - свертка – это операция фильтрации
- **Корреляция** сравнивает **сходство** двух **наборов данных**. Корреляция рассчитывает меру сходства двух входных сигналов при их смещении друг от друга. Результат корреляции достигает максимума в тот момент, когда два сигнала совпадают наилучшим образом .
 - корреляция является мерой сходства двух сигналов.

ИТОГИ

- Рассмотрено частотное представление изображения
- Показаны методы фильтрации в пространственной и частотной областях
- Изучено понятие свертки и кросс-корреляции