ВВЕДЕНИЕ В КОМПЬЮТЕРНОЕ ЗРЕНИЕ

Лекция № 2

Цифровая обработка сигналов

План лекции

1. Частотная область изображения
2. Системы и фильтры
3. Свертки

01

Частотная область изображения

Мотивация к обработке изображений

De-noising

Super-resolution

In-painting

Мотивация к обработке изображений

Бинаризация

Выделение компонент связности

Выделение краев

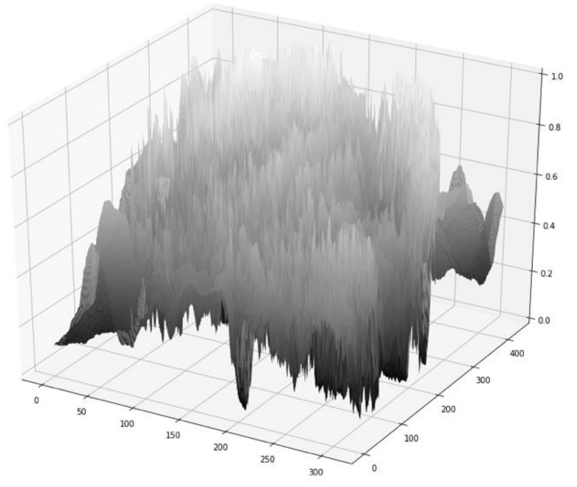
Изображение как дискретная функция

Изображение как функция f от R^2 до R^M:

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

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

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



Pix

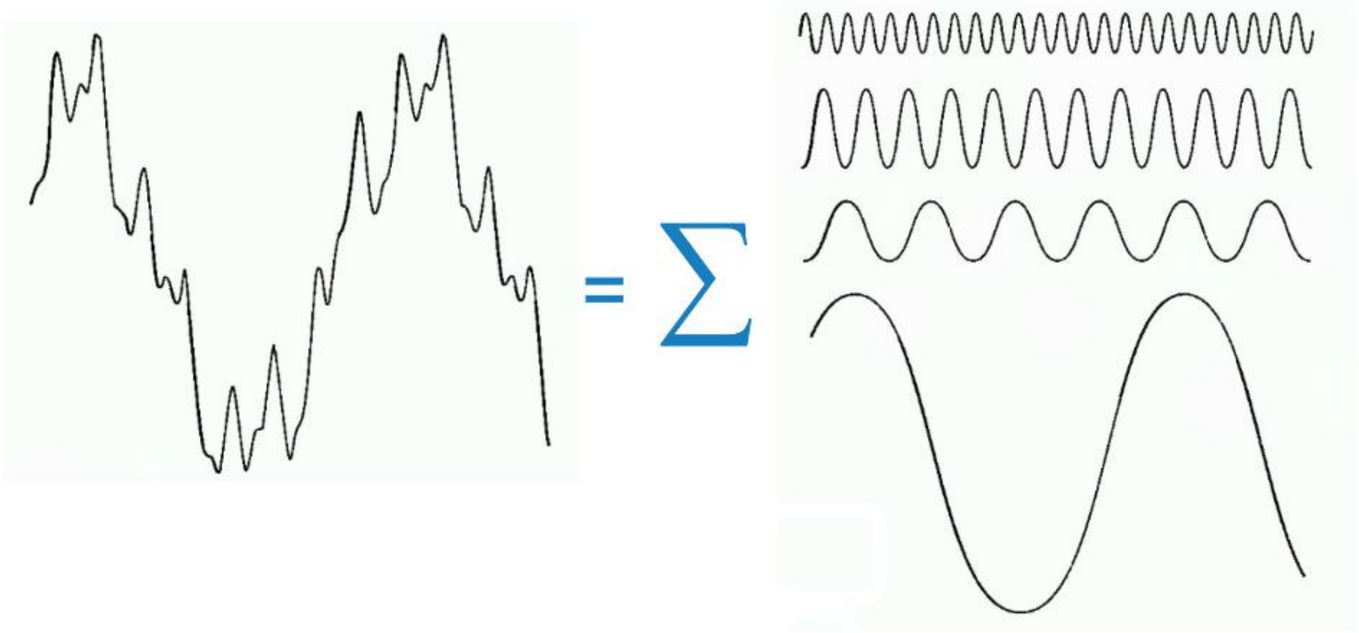
m

n

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

Ряд Фурье

Периодический сигнал может быть представлен в виде суммы



Ряд и преобразование Фурье

Ряд Фурье́ — 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

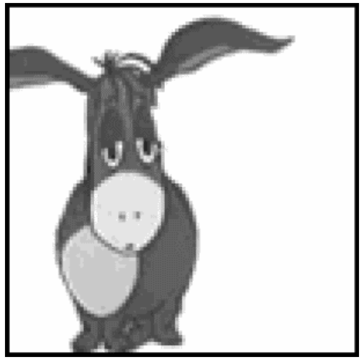
Прямое $$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\omega}\, dx.$$

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

$$S_{mn}(x,y) = \sum_{j=-m}^{j=m} \sum_{k=-n}^{k=n} a_{jk} \cos\frac{\pi j x}{l_1} \cos\frac{\pi k y}{l_2},$$

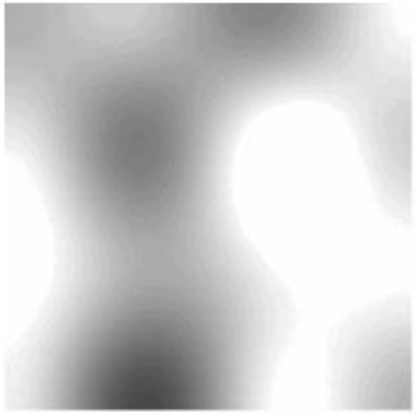


l_1 — ширина, l_2 — высота,

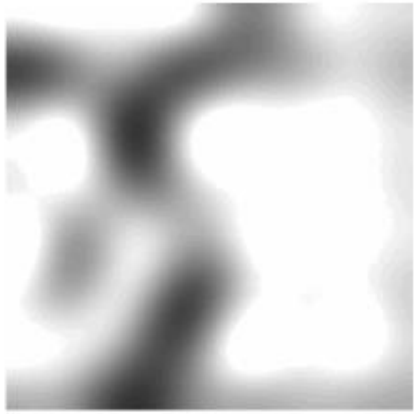
коэффициенты a_{jk} находятся по формулам:

$$a_{jk} = \frac{1}{l_1 l_2} \int_0^{l_1} \int_0^{l_2} f(x,\ y) \cos\frac{\pi j x}{l_1} \cos\frac{\pi k y}{l_2}\, dy dx.$$

Преобразование Фурье для двумерного случая



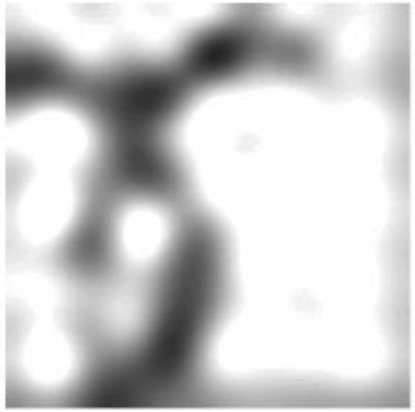
$m = n = 3$



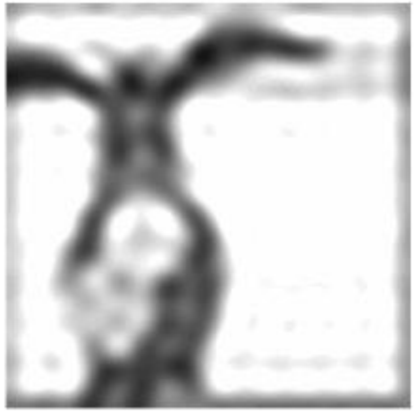
$m = n = 7$

$m = n = 90$

$m = n = 100$



$m = n = 10$



$m = n = 20$

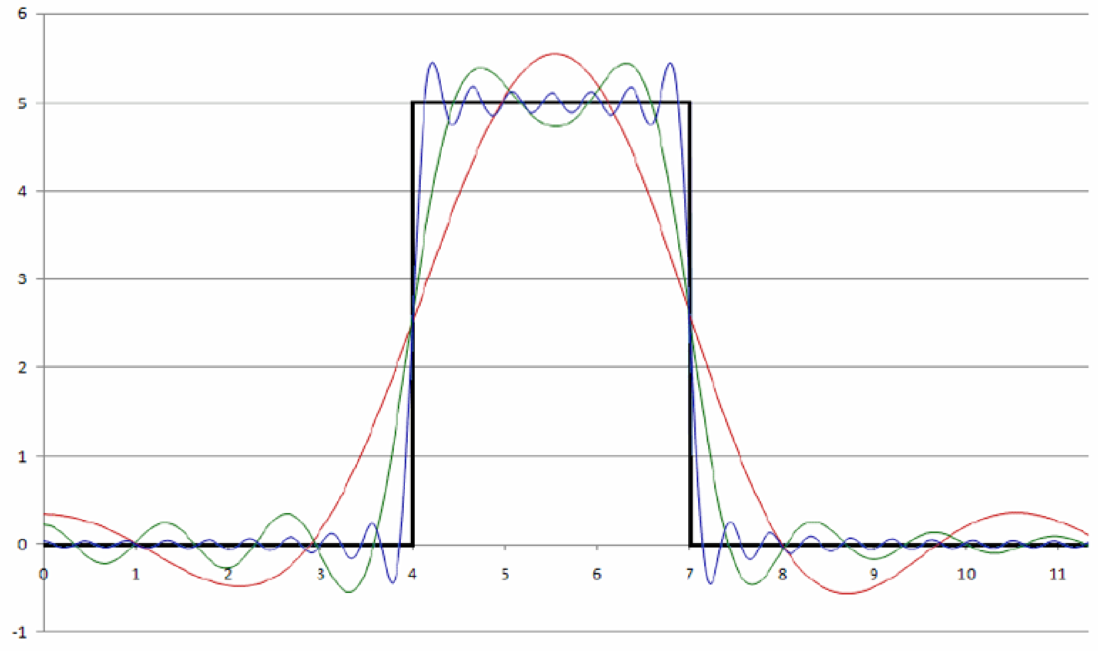
$m = n = 110$

$m = n = 120$

Эффект Гиббса

Частичные суммы ряда Фурье по косинусам для функции с двумя ступеньками:

$$f(x) = \begin{cases} 0, & x \in (0,\ 4); \\ 5, & x \in (4,\ 7); \\ 0, & x \in (7,\ 20). \end{cases}$$



— исходная функция
— 10 слагаемых ряда Фурье
— 30 слагаемых ряда Фурье
— 90 слагаемых ряда Фурье

Сжатие JPEG

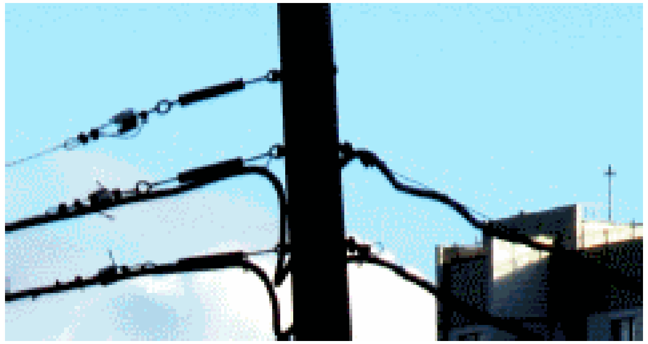
$$F(k,\ l) = \frac{2}{\sqrt{MN}}\, c(k)\, c(l) \sum_{j=0}^{N-1} \cos\left(\frac{\pi k}{N}\left(j+\frac{1}{2}\right)\right) \sum_{m=0}^{M-1} f(j,\ m) \cos\left(\frac{\pi l}{M}\left(m+\frac{1}{2}\right)\right),$$

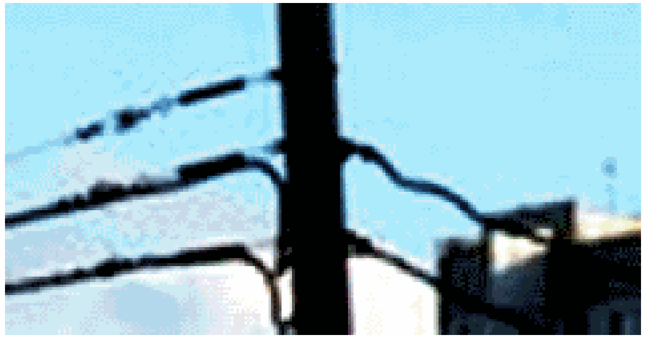
$$k,\ j \in \overline{0,\ N-1}, \quad l,\ m \in \overline{0,\ M-1},$$

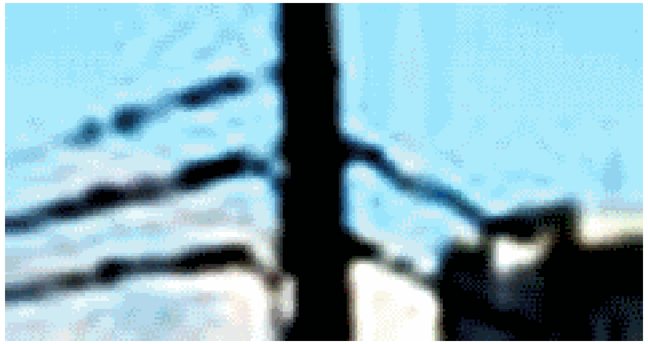
$$\check{f}(j,\ m) = \frac{2}{\sqrt{MN}} \sum_{k=0}^{N-1} c(k) \cos\left(\frac{\pi k}{N}\left(j+\frac{1}{2}\right)\right) \sum_{l=0}^{M-1} c(l)\, F(k,\ l) \cos\left(\frac{\pi l}{M}\left(m+\frac{1}{2}\right)\right),$$

$$c(n) = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0; \\ 1, & n > 0. \end{cases}$$

Формулы написаны в общем виде, хотя при jpeg-компрессии рассматриваются квадраты 8 × 8, поэтому параметры M = N = 8.







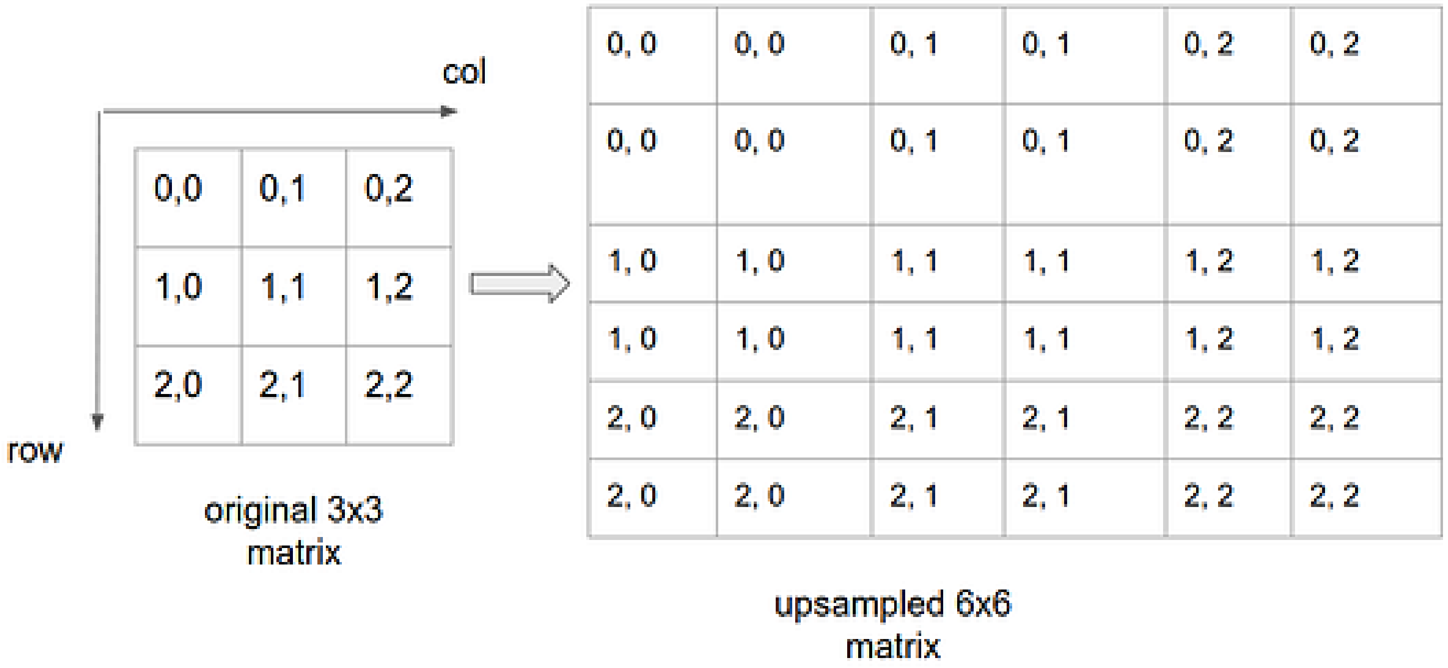
Масштабирование изображений

Как увеличить изображение по ширине и высоте?

Масштабирование изображений

Как увеличить изображение по ширине и высоте?

1. Продублировать значения



Масштабирование изображений

Как увеличить изображение по ширине и высоте?

1. Продублировать значения

2. Интерполировать значения

233	200	100
188	30	130
255	5	34

original 3x3 matrix

233		200		100	
188		30		130	
255		5		34	

upsampled 6x6 matrix

Масштабирование изображений

Как увеличить изображение по ширине и высоте?

1. Продублировать значения

2. Интерполировать значения

3. Фурье преобразование

Алгоритм масштабирования изображения методов Фурье преобразования

1. Пусть X(N1,N2) – массив яркостей пикселей изображения.

2. Вычислить Px = средняя (среднеквадратичная) яркость пикселей в массиве X

3. Вычислить массив Z=FT(X) – прямое двухмерное дискретное преобразование Фурье

4. Вычислить массив Z′=T(Z), где T – либо добавление нулевых строк и столбцов матрицы соответствующих высоким частотам, либо удаление строк и столбцов матрицы соответствующих высоким частотам для получения требуемого размера итогового изображения

5. Вычислить массив Y=RFT(Z′) – обратное двухмерное дискретное преобразование Фурье

6. Вычислить Py = средняя (среднеквадратичная) яркость пикселей в массиве Y

7.Нормировать массив Y(M1,M2) по среднему уровню яркости Px/Py

Алгоритм масштабирования изображения методов Фурье преобразования

Исходное изображение × 4

Ряд Фурье m = n = 100

Подавление эффекта Гиббса

Сглаживающее свойство уравнения теплопроводности. Решение задачи распространения тепла в квадратной мембране с теплоизолированными краями, без источников тепла и с начальным распределением температуры φ(x, y) = яркости изображения (x, y):

$$\begin{cases} u_t - a^2(u_{xx} + u_{yy}) = 0, & x \in (0,\ p),\ \ y \in (0,\ s),\ \ t > 0; \\ u(x,\ y,\ 0) = \varphi(x,\ y), & x \in [0,\ p],\ \ y \in [0,\ s]; \\ u(0,\ y,\ t) = u(p,\ y,\ t) = 0, & y \in [0,\ s],\ \ t \geqslant 0; \\ u(x,\ 0,\ t) = u(x,\ s,\ t) = 0, & x \in [0,\ p],\ \ t \geqslant 0. \end{cases}$$

Решение:

$$u = \sum_{n,k=-\infty}^{+\infty} \varphi_{kn} \cos\left(\frac{\pi n x}{p}\right) \cos\left(\frac{\pi k y}{s}\right) e^{-a^2 \lambda_{kn} t},$$

$$\lambda_{kn} = \left(\frac{\pi n}{p}\right)^2 + \left(\frac{\pi k}{s}\right)^2,$$

$$\varphi_{kn} = \frac{1}{ps} \int\limits_0^s \int\limits_0^p \varphi(x,\ y) \cos\left(\frac{\pi n x}{p}\right) \cos\left(\frac{\pi k y}{s}\right)\, dx\, dy.$$

Алгоритм масштабирования изображения методов Фурье преобразования

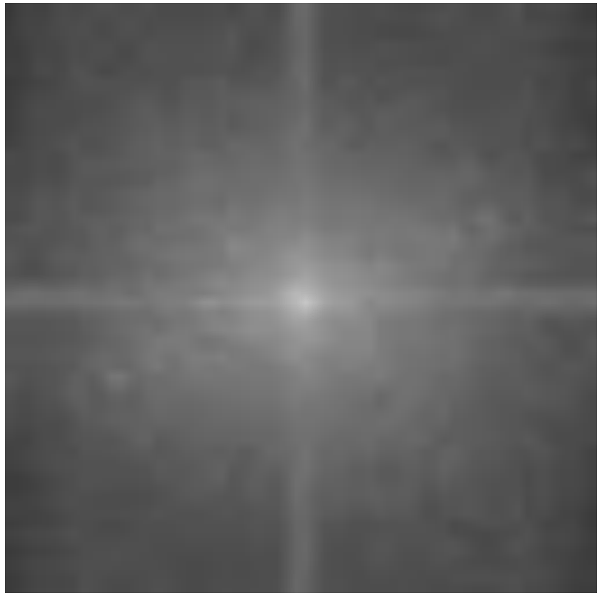
исходное изображение × 4

ряд Фурье m = n = 100

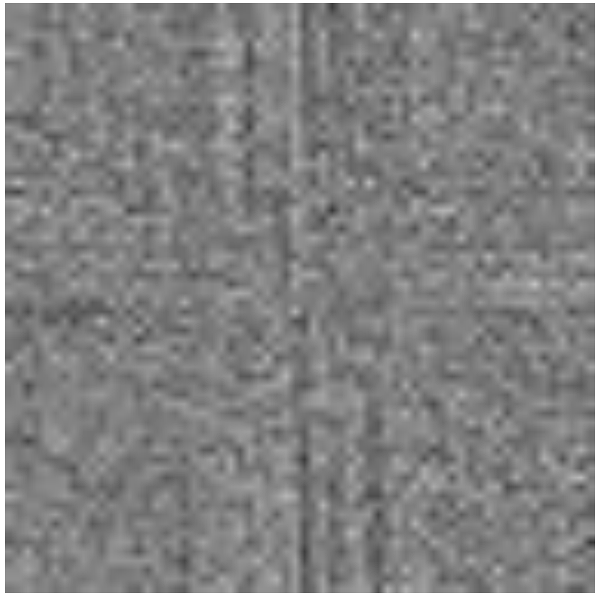
ряд Фурье m = n = 100
после сглаживания

Преобразование Фурье для изображений

Исходное изображение



Центрированная амплитуда спектра изображения

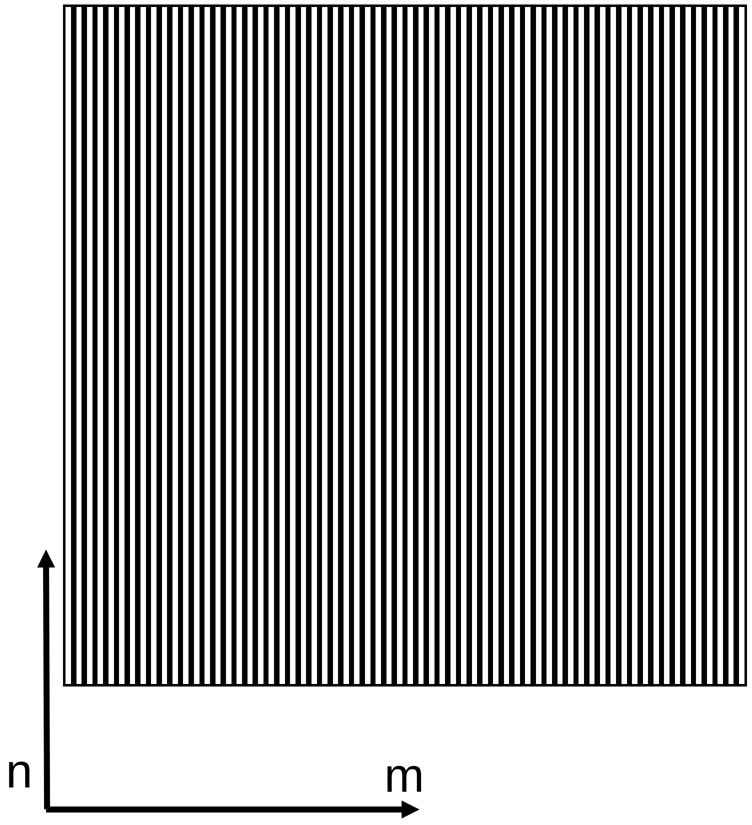


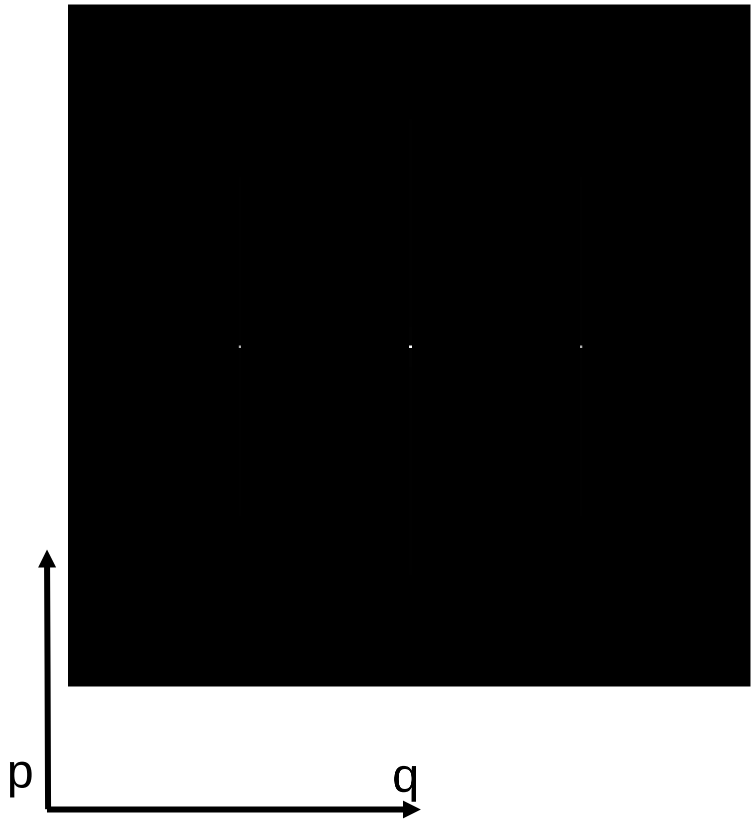
Фаза изображения

Преобразование Фурье для изображений

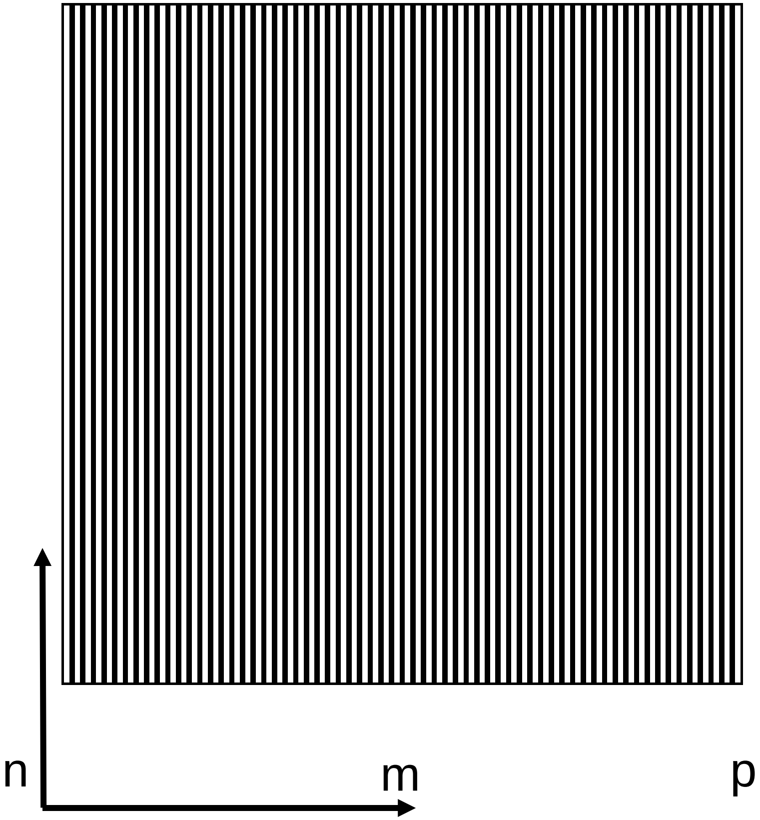
$$f(m,n) = \frac{1}{MN}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} F(p,q)e^{j2\pi pm/M}e^{j2\pi qn/N} \quad \begin{matrix} m = 0,\ 1,\ ...,\ M-1 \\ n = 0,\ 1,\ ...,\ N-1 \end{matrix}$$

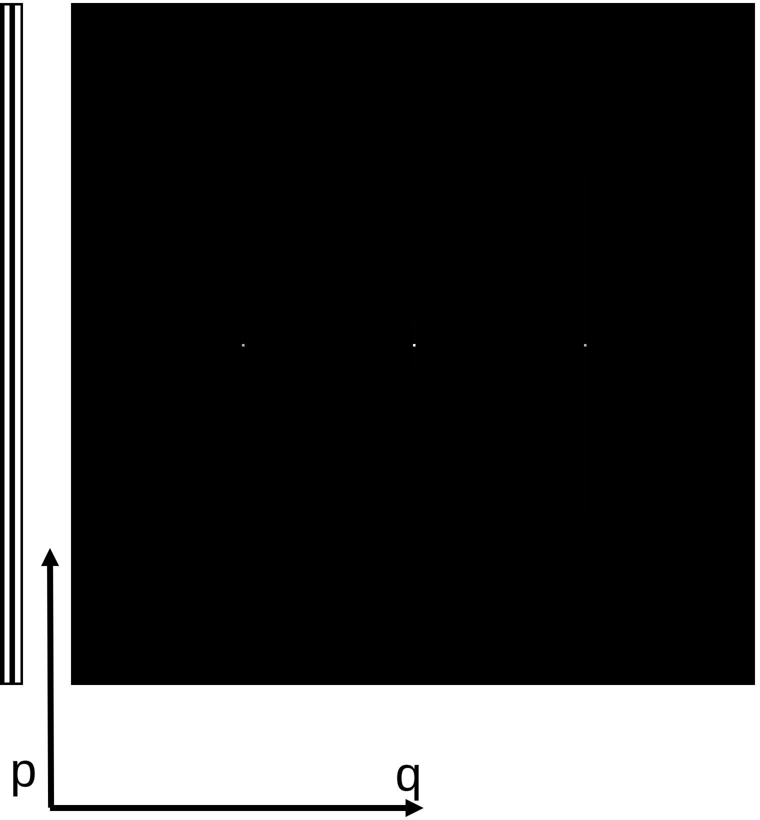
$$F(p,q) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi pm/M}e^{-j2\pi qn/N} \quad \begin{matrix} p = 0,\ 1,\ ...,\ M-1 \\ q = 0,\ 1,\ ...,\ N-1 \end{matrix}$$





Преобразование Фурье для изображений





отображаются все частоты, величина которых составляет не менее 5% от основного пика

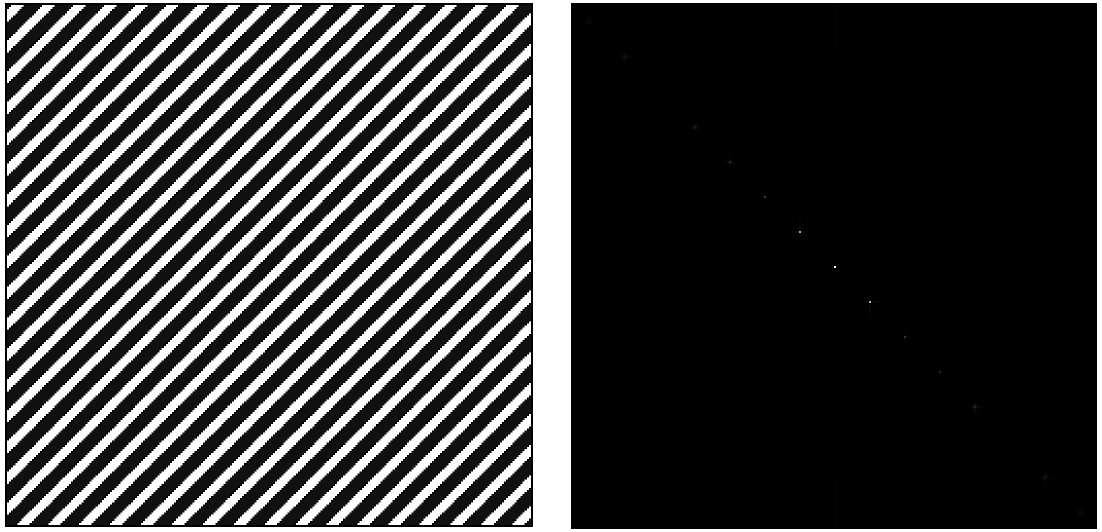
Расстояние точек до центра можно объяснить следующим образом: максимальная частота, которая может быть представлена в пространственной области, равна двум парам полос шириной в пиксель (одна белая, одна черная).

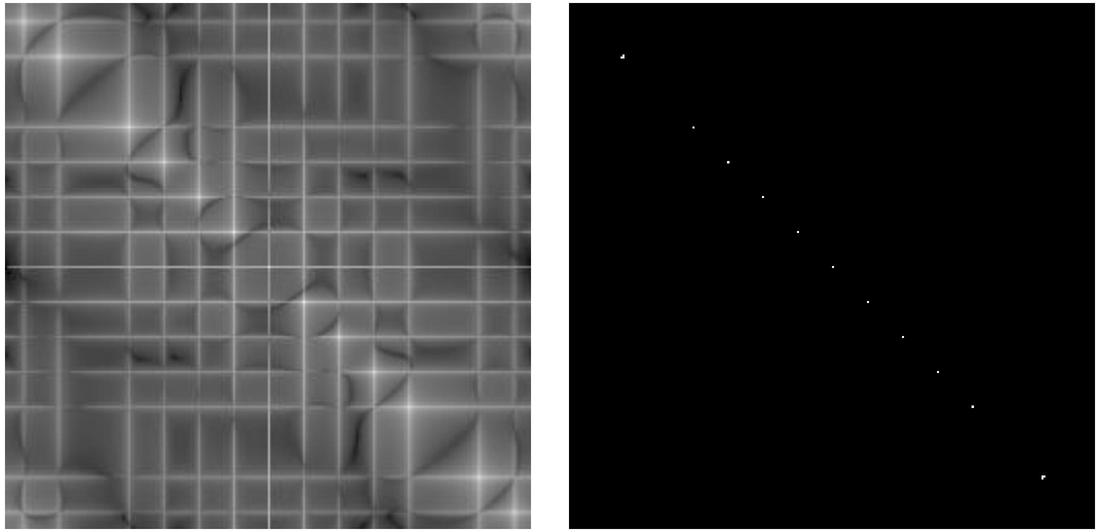
$$f_{max} = \frac{1}{2\ \text{pixels}}$$

Полосы шириной в два пикселя на приведенном выше изображении представляют

$$f = \frac{1}{4\ \text{pixels}} = \frac{f_{max}}{2}$$

Преобразование Фурье для изображений

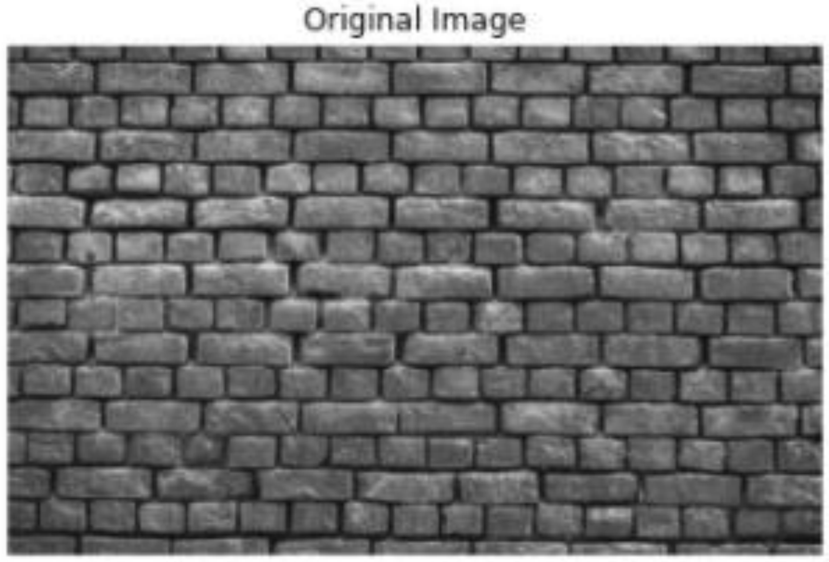


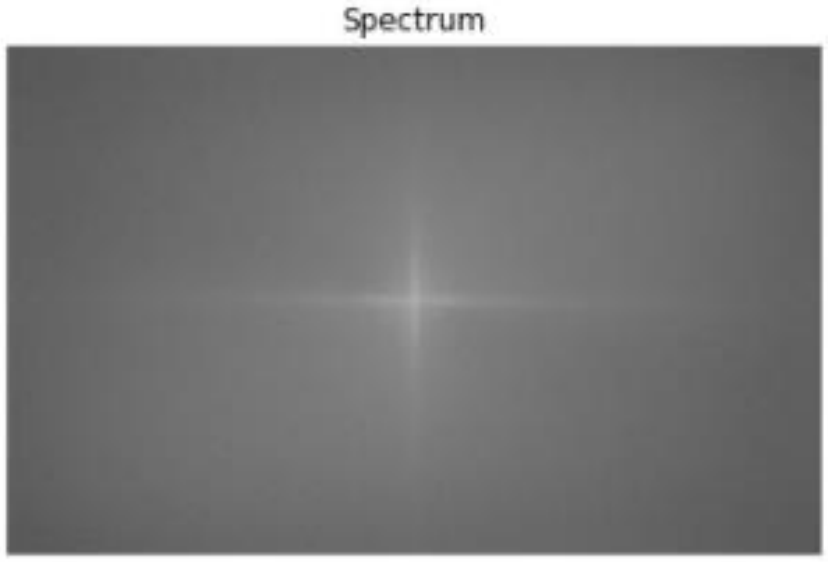


Все представленные частоты кратны базовой частоте полос на изображении пространственной области

отображаются все частоты, величина которых составляет не менее 5% от основного пика

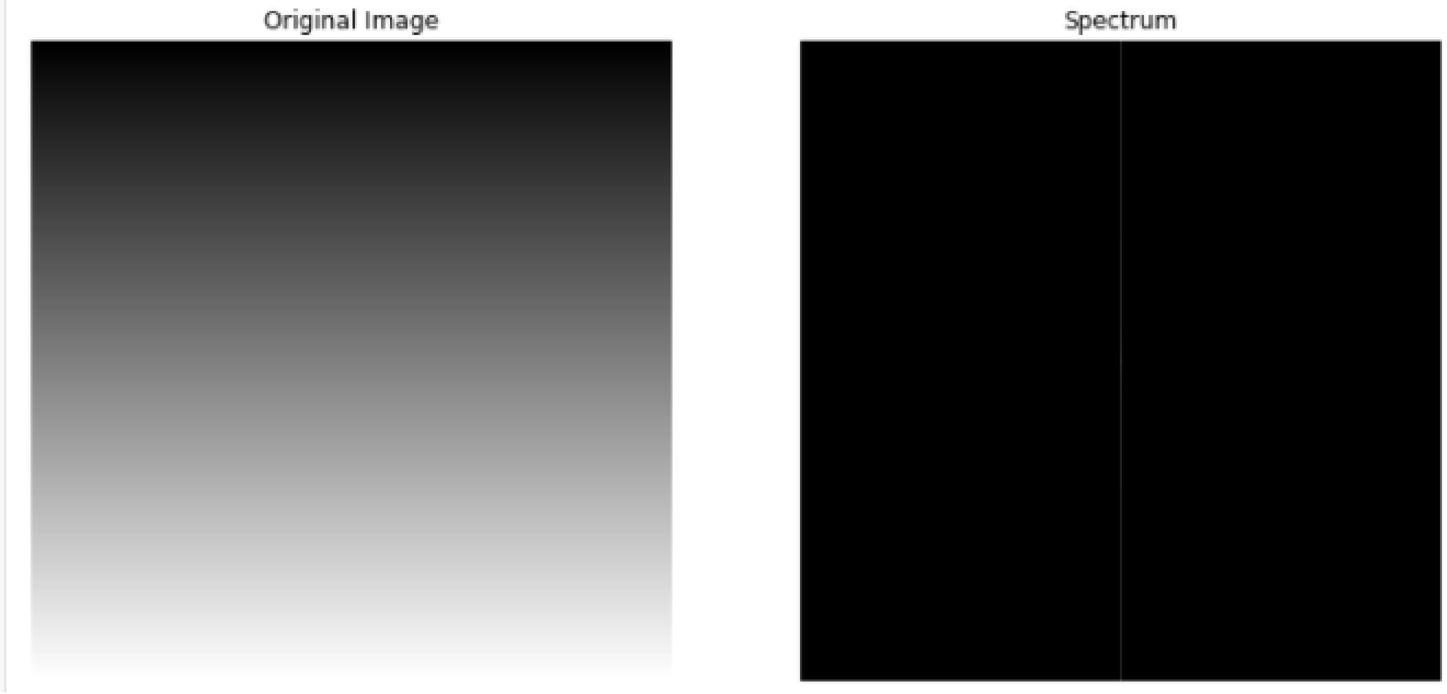
Преобразование Фурье для изображений





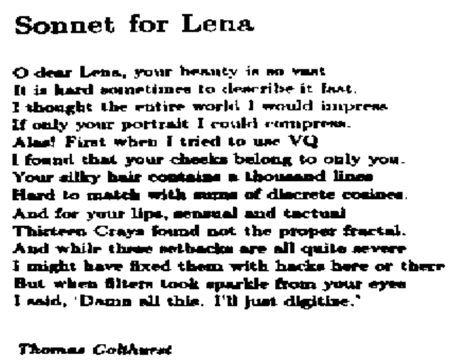
«Высокие» частоты: область с сильными и частыми перепадами значений пикселей

Преобразование Фурье для изображений



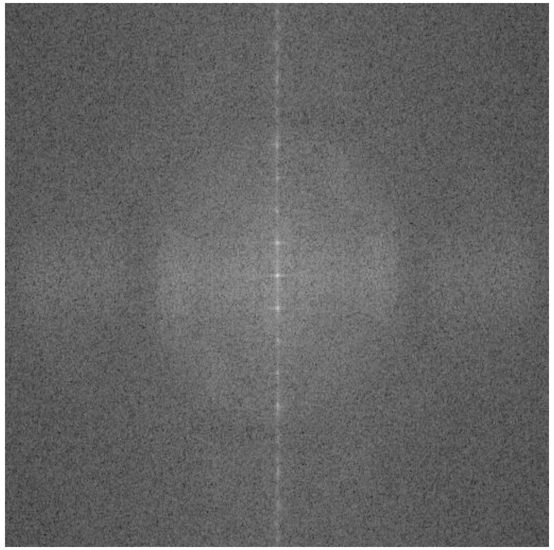
«Низкие» частоты: области с слабыми и редкими перепадами значений пикселей

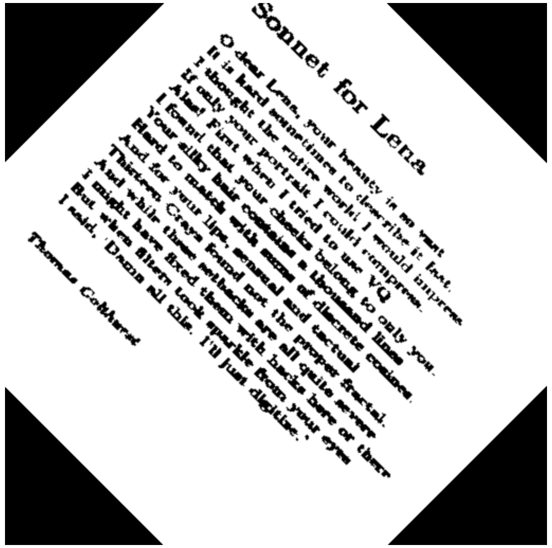
Амплитуда спектра изображения

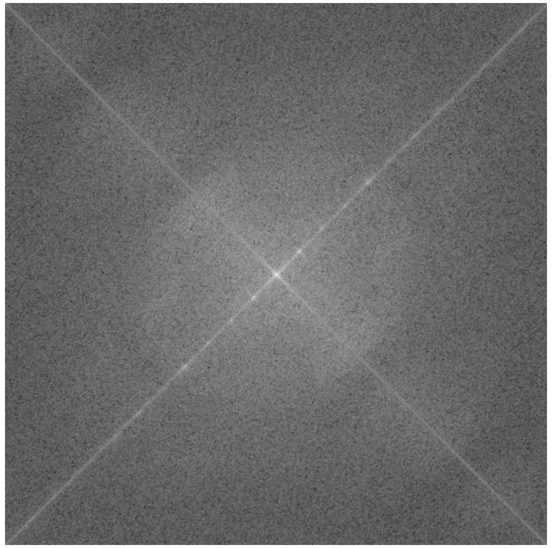

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst







02

Системы и фильтры

Системы и фильтры

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

Зачем?

- Выделить «полезную» информацию

- Изменить или улучшить свойства признаков изображения

Системы и фильтры

Необходимые условия линейности

1. Гомогенность — при изменении амплитуды входного сигнала в k раз также в k раз изменяется и амплитуда выходного сигнала

2. Аддитивность — при суммировании входных сигналов результирующий сигнал на выходе будет равен сумме реакций от исходных сигналов

3. Инвариантность — когда смещение входного сигнала во времени вызывает аналогичное смещение выходного сигнала

4. Статическая линейность — когда основные законы в системе описываются линейными уравнениями

5. Гармоническая верность — если на вход системы подать синусоидальный сигнал, то на выходе будет сигнал той же частоты

Системы и фильтры

Свойства линейных систем

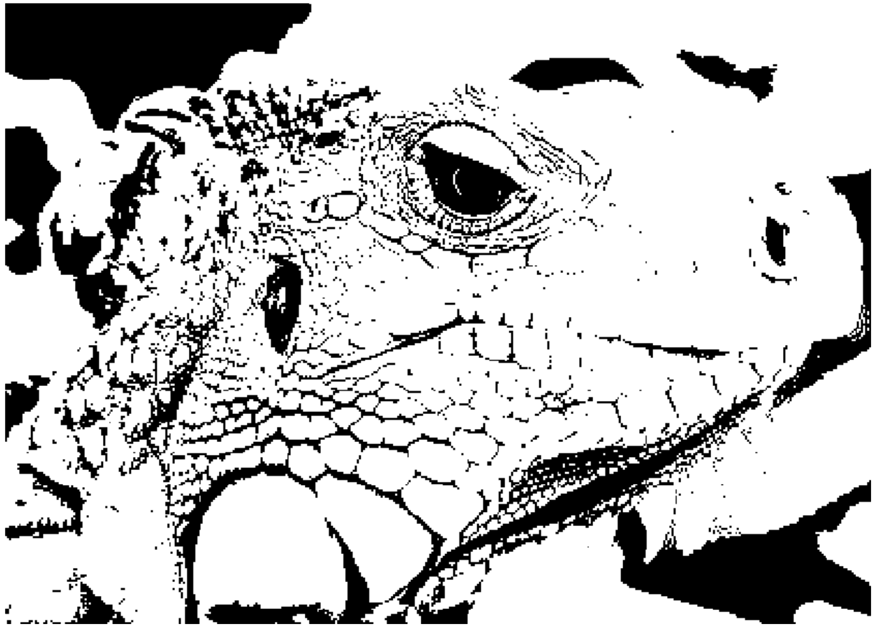
1. Порядок установки линейных систем не влияет на результирующий сигнал

2. Любая сложная система будет линейна, если составлена из линейных систем и блоков суммирования

3. Перемножение сигнала на константу является линейной операцией, а перемножение двух сигналов — нелинейной

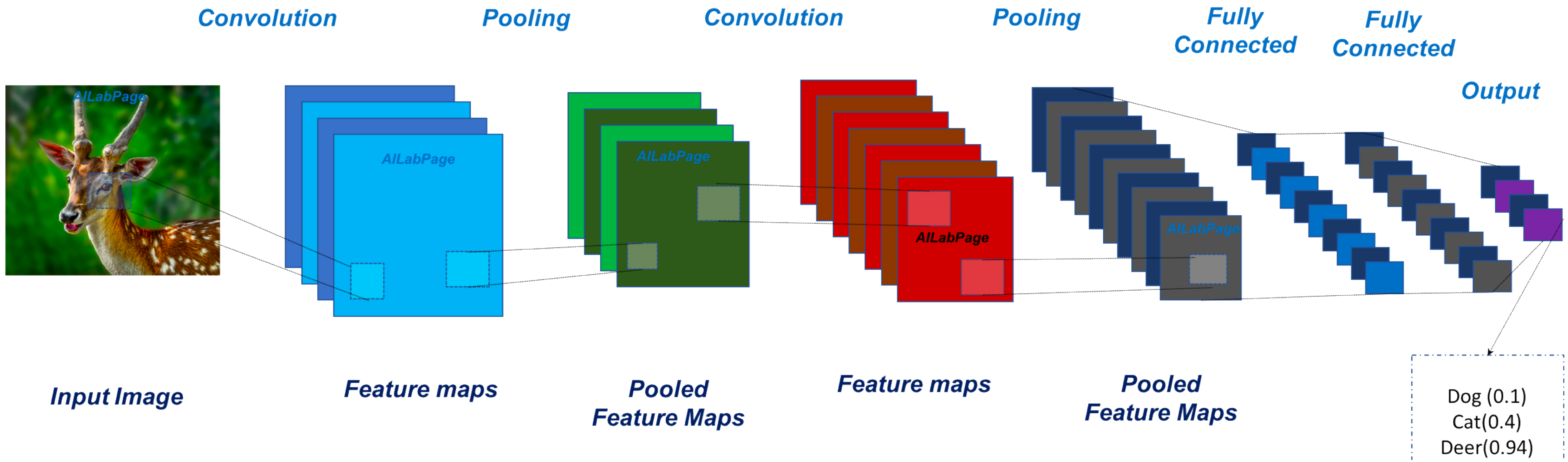
Интуитивное понимание систем

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



Кстати говоря…

Сверточные нейронные сети – это тип системы или нелинейная система, содержащая несколько отдельных линейных подсистем

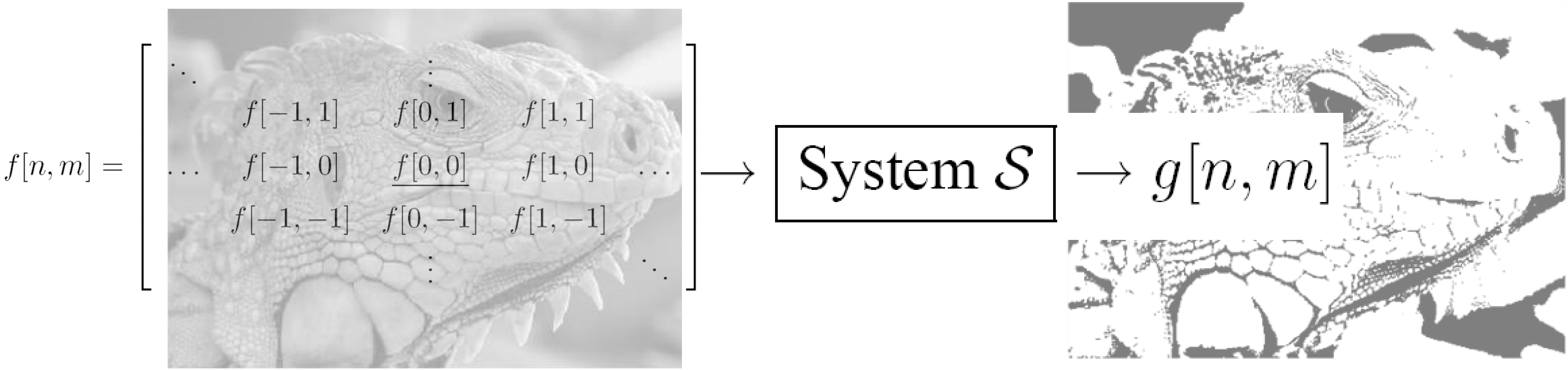


(подробнее об этом в другом курсе)

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



Фильтр «Размытие»

Исходное изображение

Изображение после
размытия

Пример фильтра «Размытие»

2D скользящее среднее размером 3 на 3

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

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

h

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

Пример фильтра «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 «Размытие»

$\mathrm{f}[n, m]$

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

$\mathrm{g}[n, m]$

	0	10							

Пример фильтра «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 «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 «Размытие»

$\mathrm{f}[n, m]$

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

$\mathrm{g}[n, m]$

	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 «Размытие»

Данный фильтр «пересчитывает» каждый пиксель средним значением по окрестности

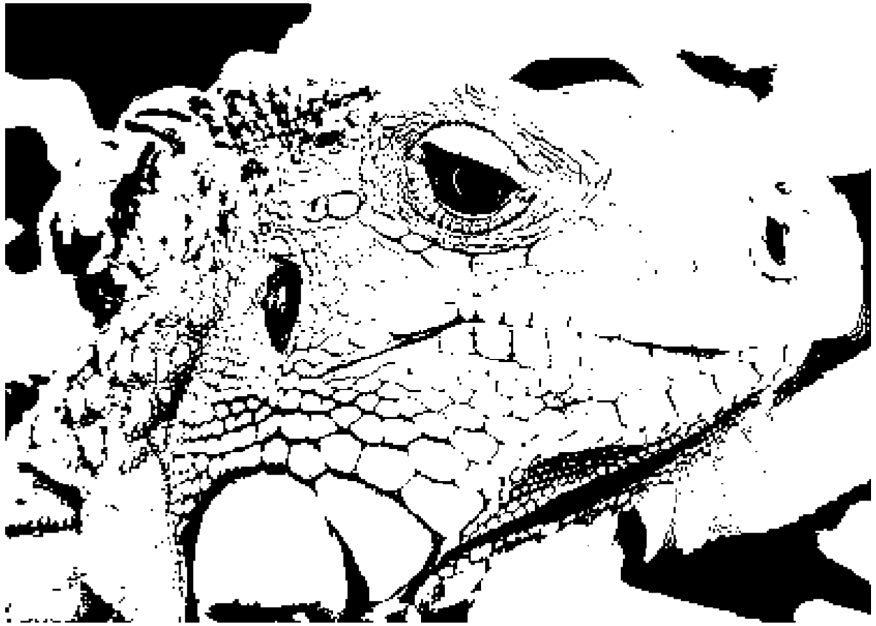
Достигается эффект «сглаживания» (осреднение резких переходов значений пикселей)

h

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

Фильтр «Пороговое правило»

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



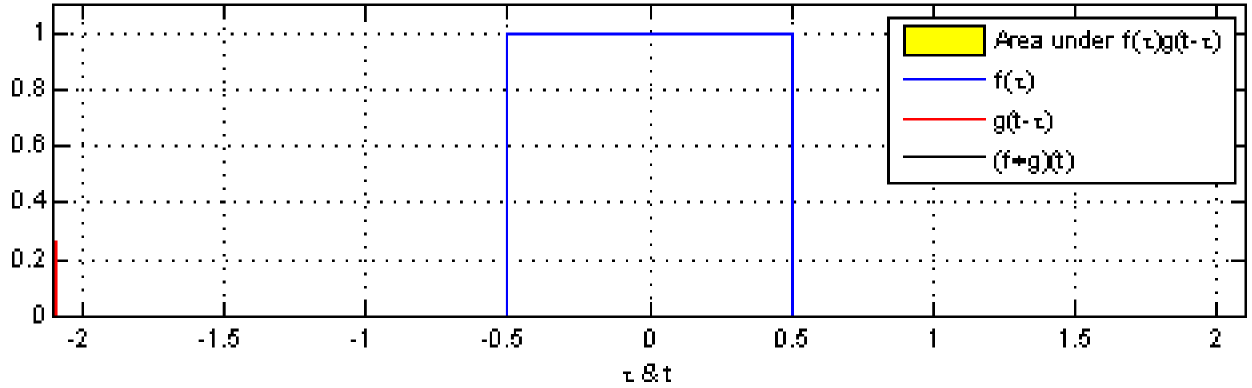
03

Свертки

Свертка

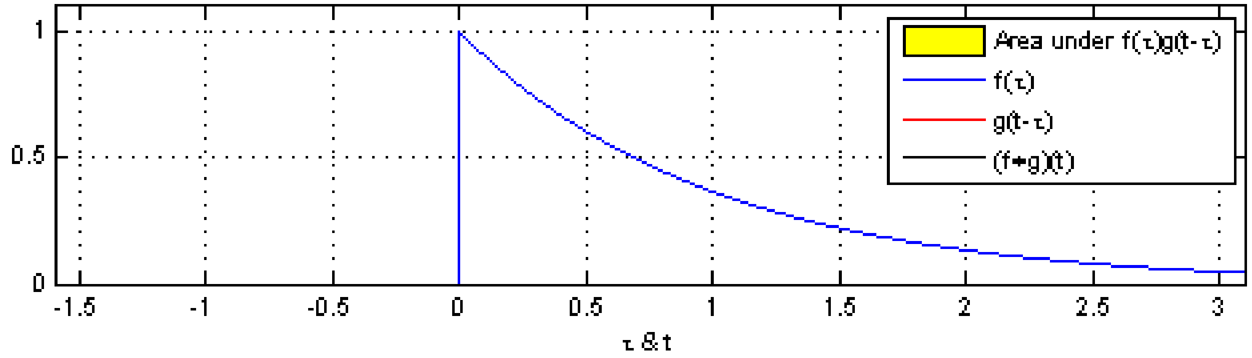
Свёртка, **конволюция** — операция в функциональном анализе, которая при применении к двум функциям f и g возвращает третью функцию, соответствующую взаимнокорреляционной функции f(x) и g(-x)

$$(f * g)(x) \stackrel{\mathrm{def}}{=} \int\limits_{\mathbb{R}^n} f(y)\, g(x-y)\, dy = \int\limits_{\mathbb{R}^n} f(x-y)\, g(y)\, dy.$$



где взаимнокорреляционная функциия f(x) и g(-x):

$$(f \star g)(t) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f^*(\tau)\ g(t+\tau)\, d\tau,$$



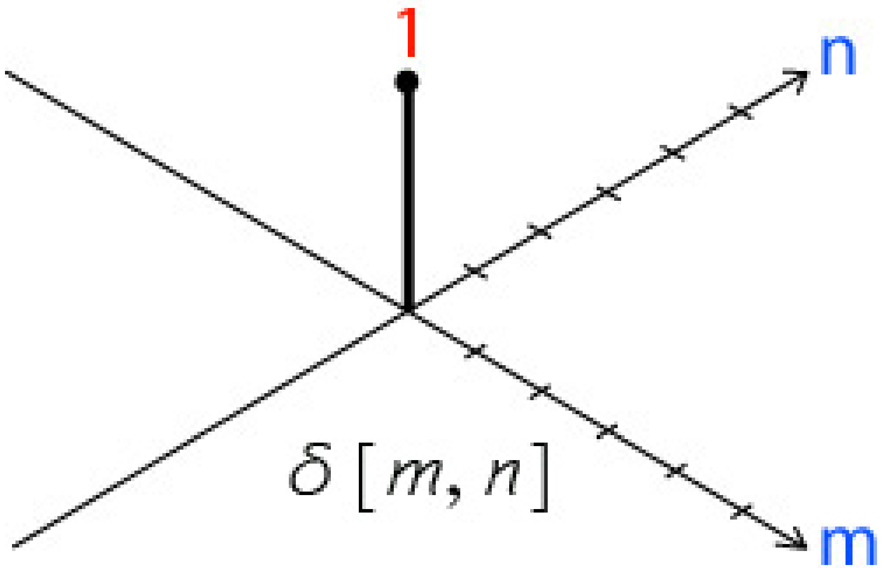
Свойство:

$$\mathcal{F}[f \star g] = (\mathcal{F}[f])^* \cdot (\mathcal{F}[g])$$

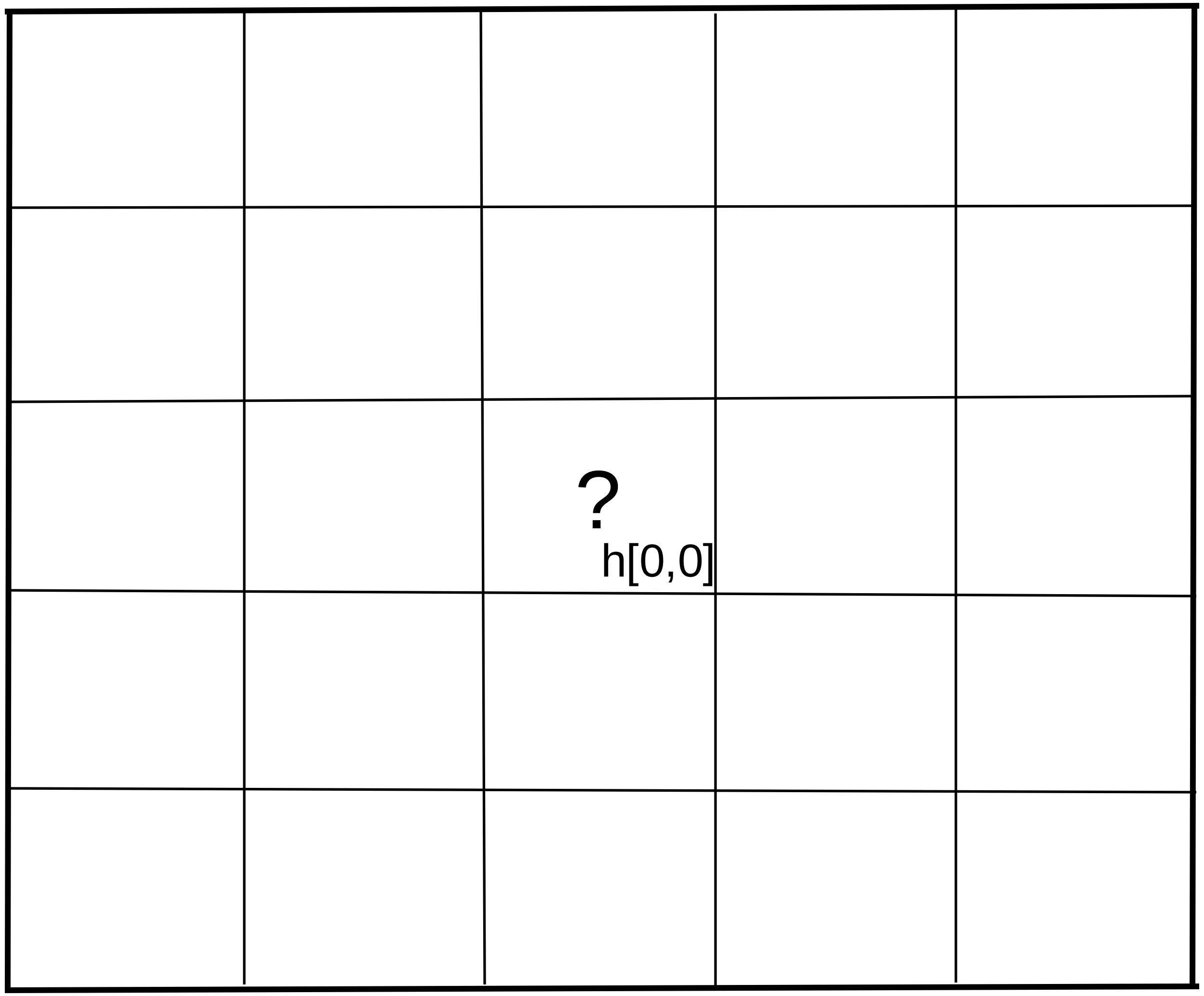
Импульсная функция

Рассмотрим специальную функцию:

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



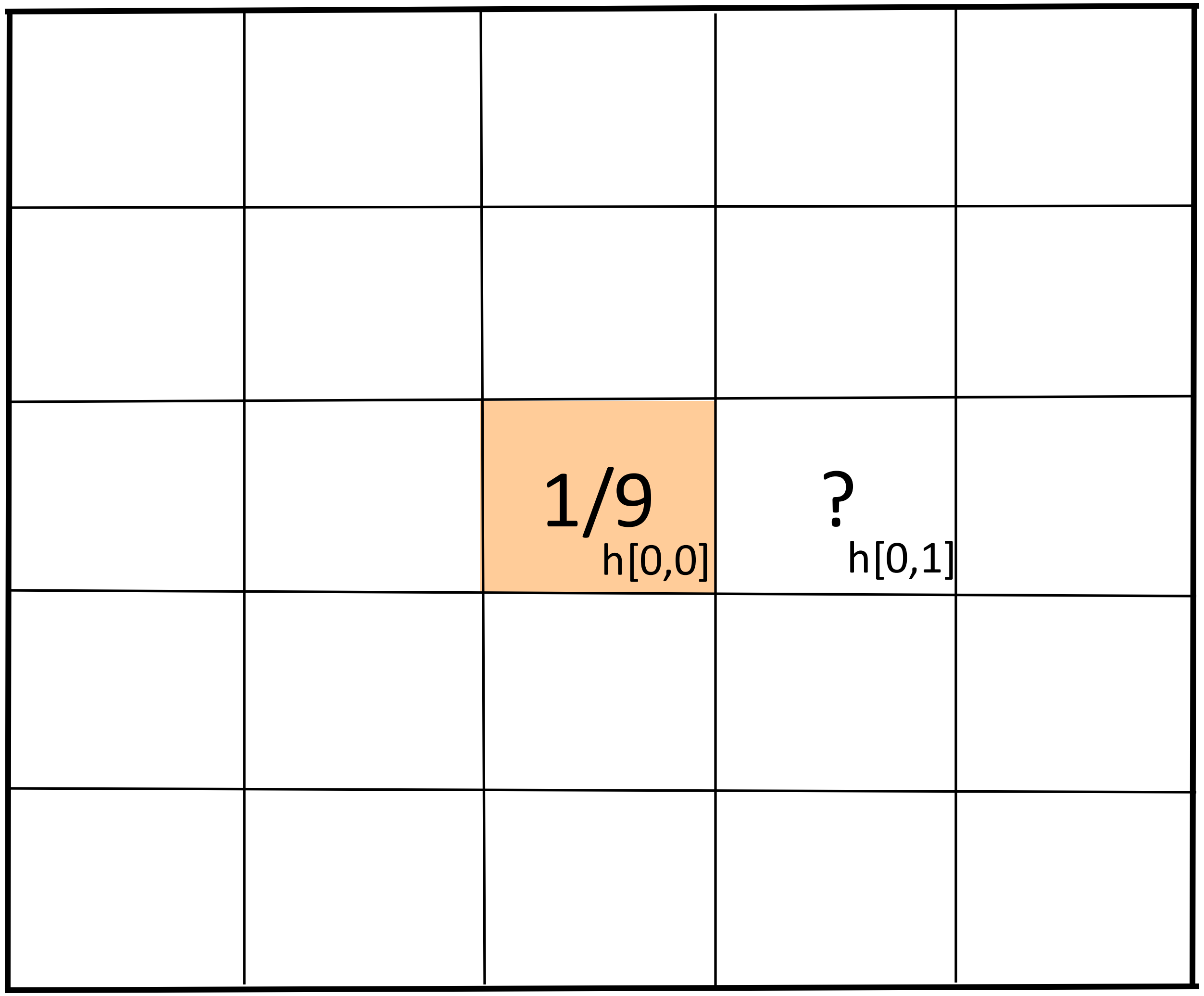
Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

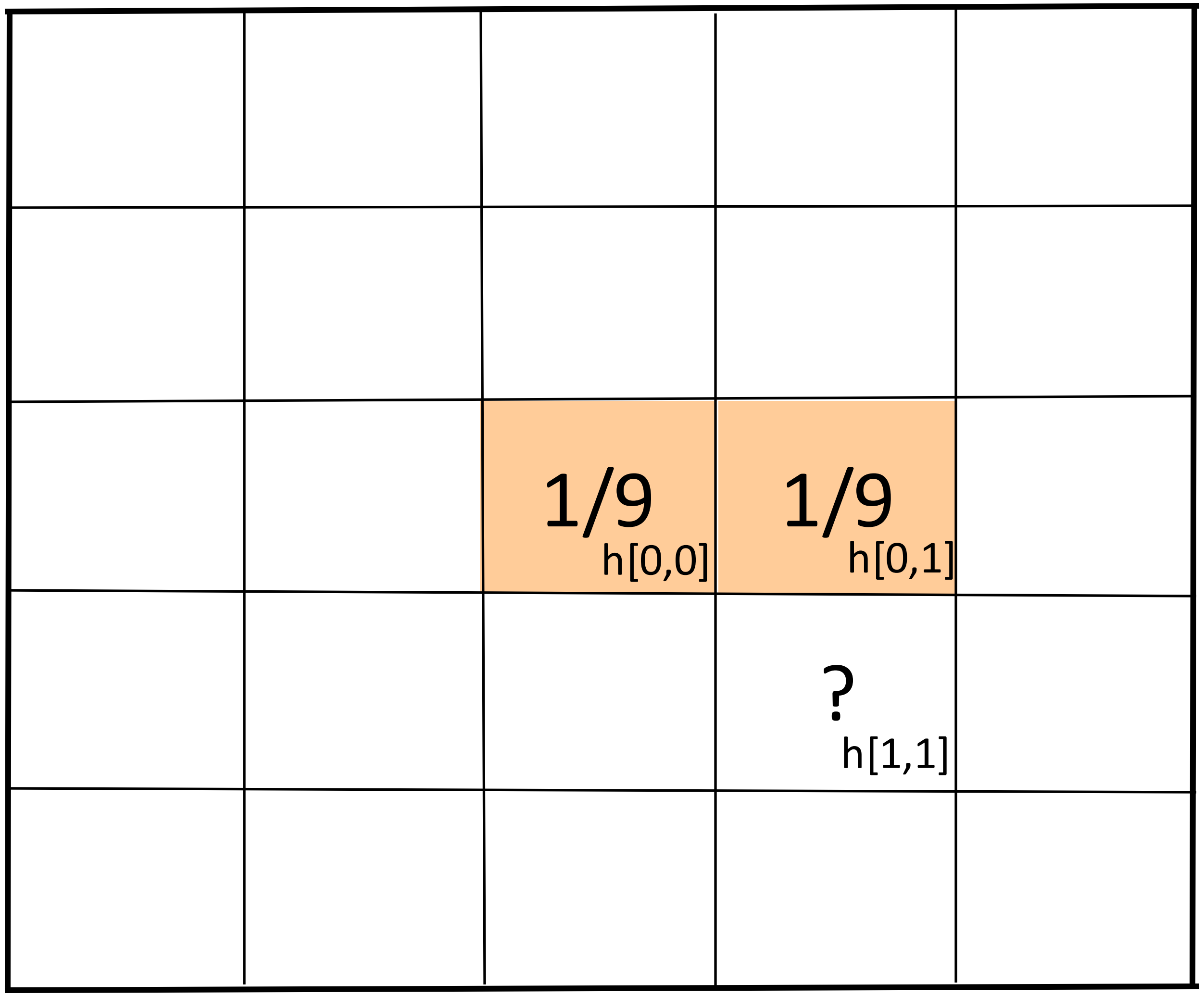
Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

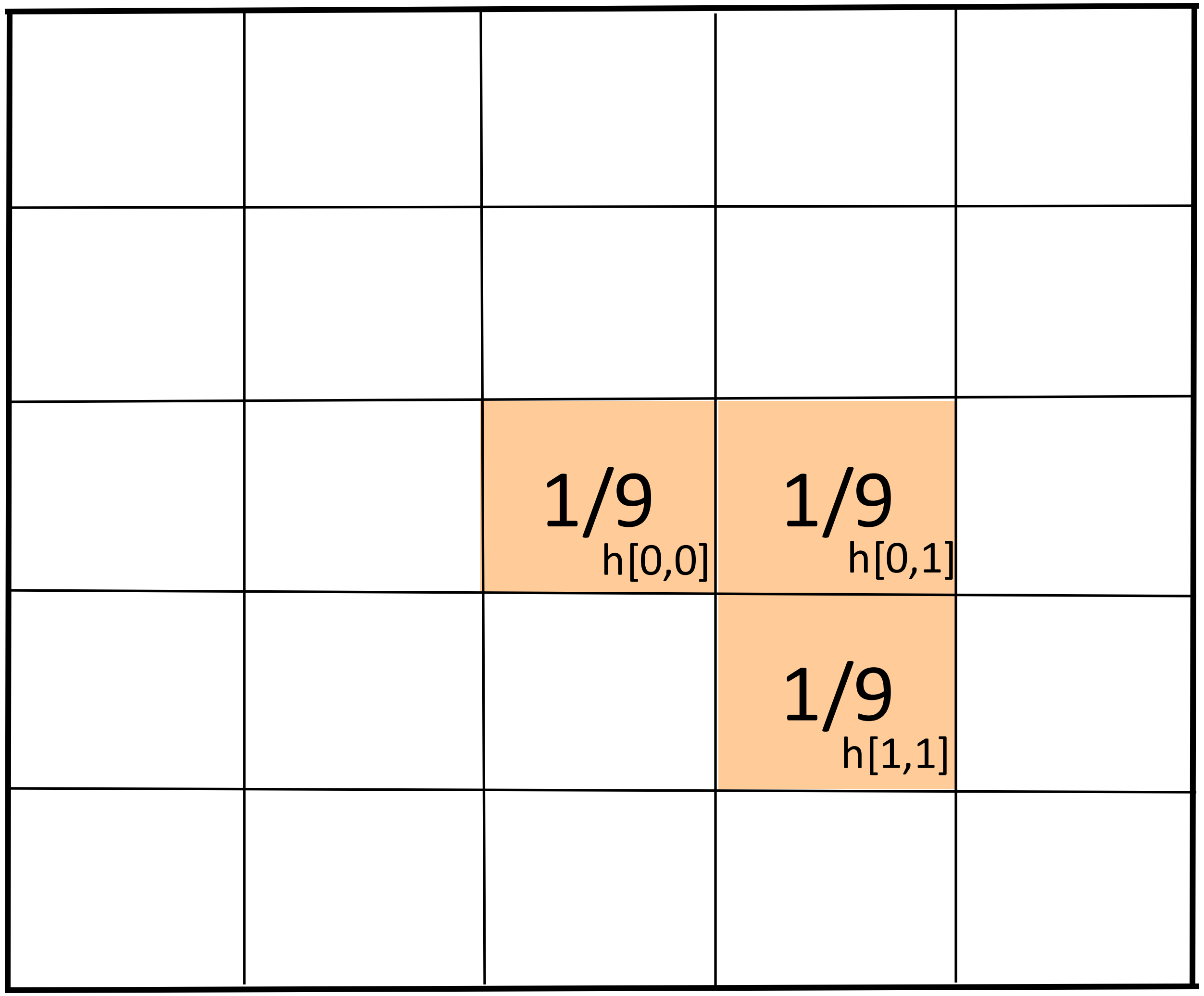
Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

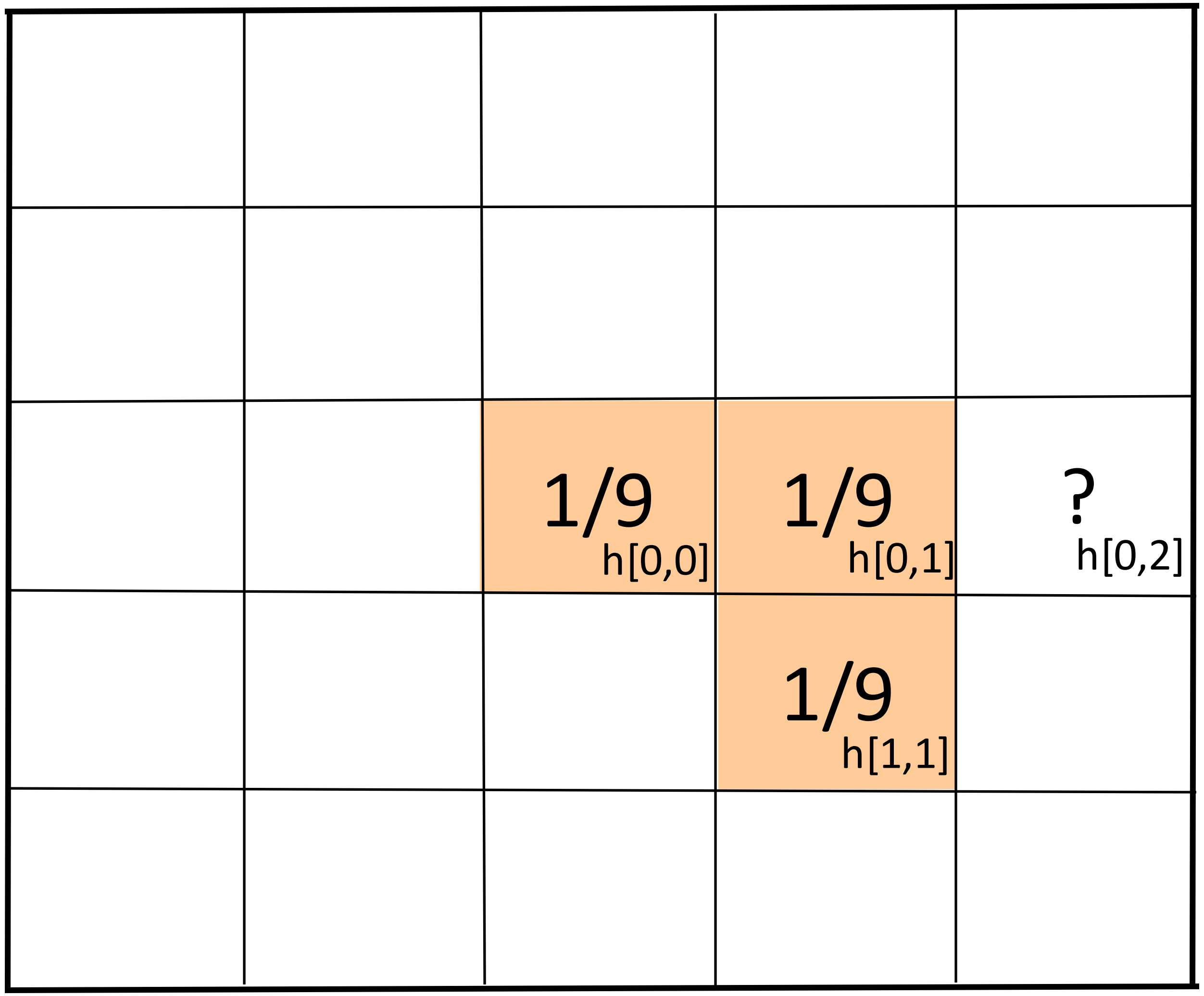
Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

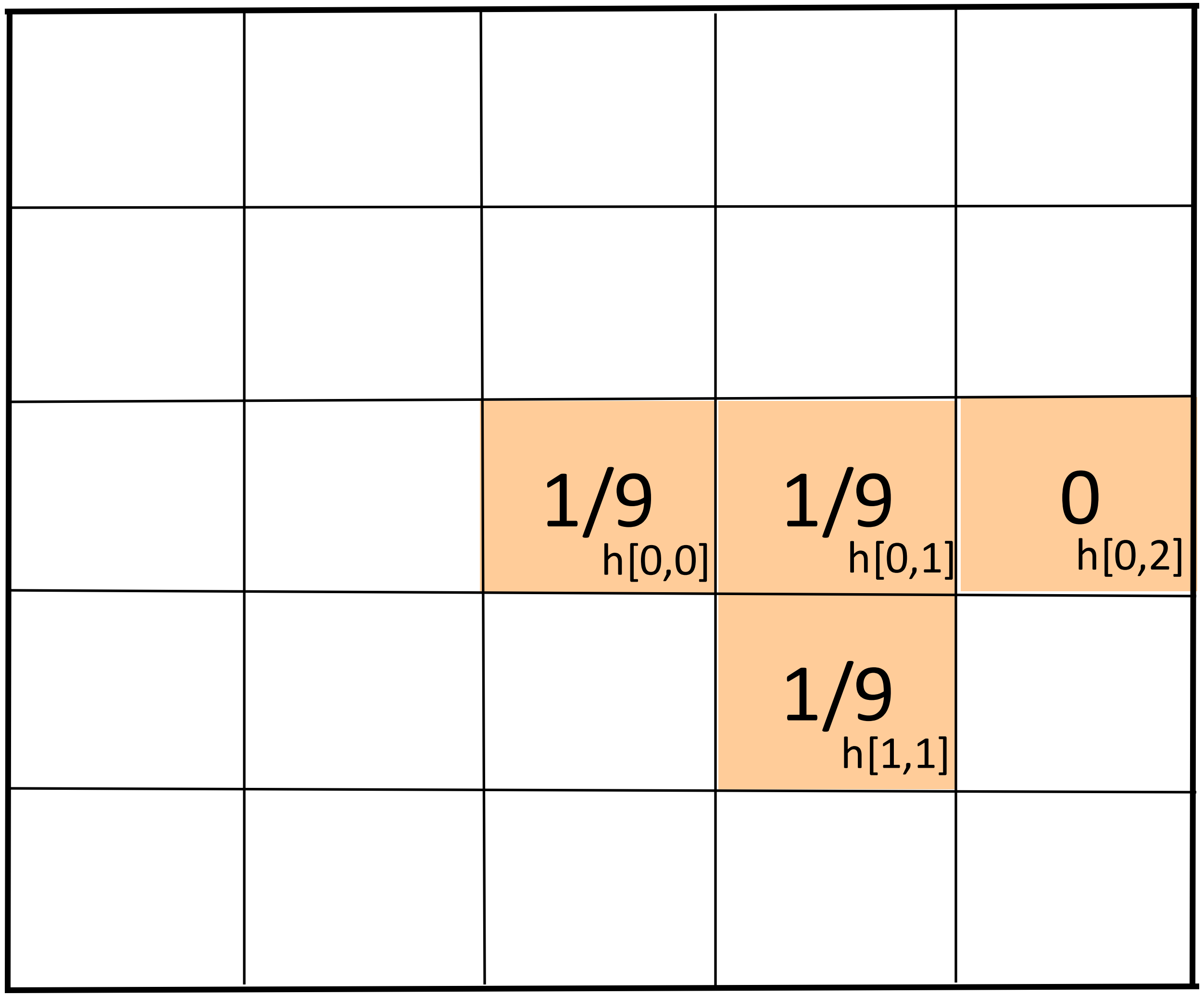
Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

Импульсный отклик от фильтра размытия



$$\delta_2 \xrightarrow{S} h[n,m]$$

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

Импульсный отклик от фильтра размытия

0	0	0	0	0
0	1/9 h[-1,-1]	1/9	1/9	0
0	1/9	1/9 h[0,0]	1/9 h[0,1]	0 h[0,2]
0	1/9	1/9	1/9 h[1,1]	0
0	0	0	0	0

$$\delta_2 \overset{S}{\rightarrow} g[n,m]$$

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

Фильтр размытия через импульсные функции

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

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

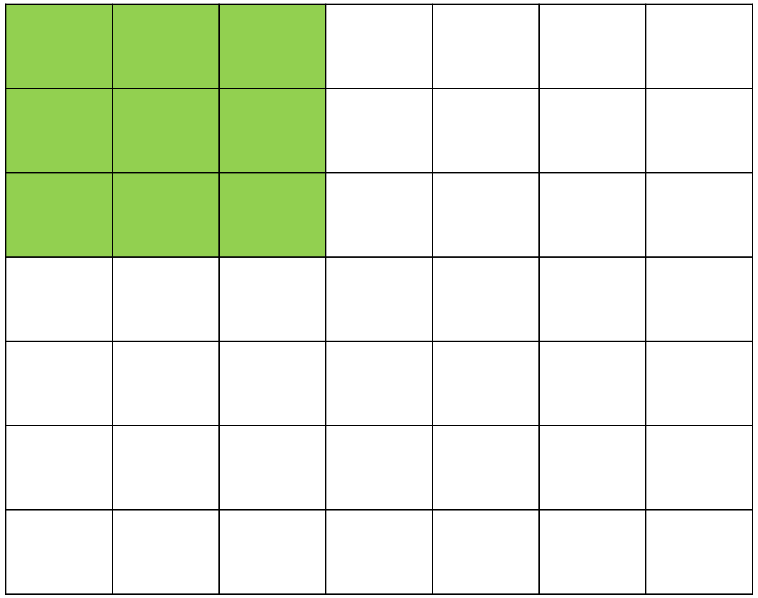
h

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

Двумерная свертка

2D свёртка очень похожа на 1D.
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

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

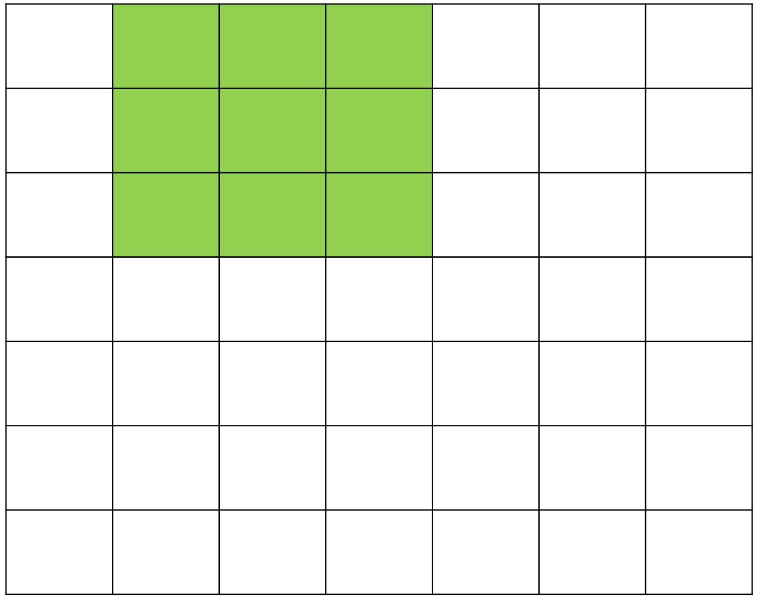


Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

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

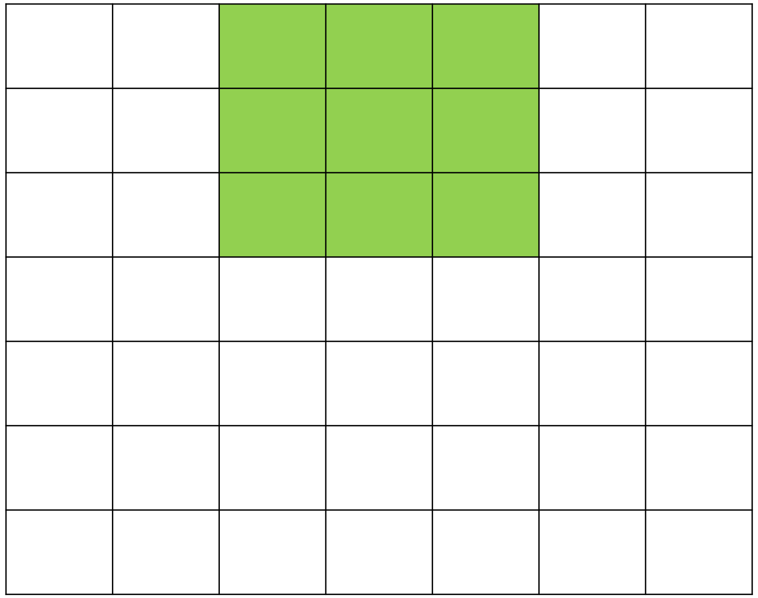


Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

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

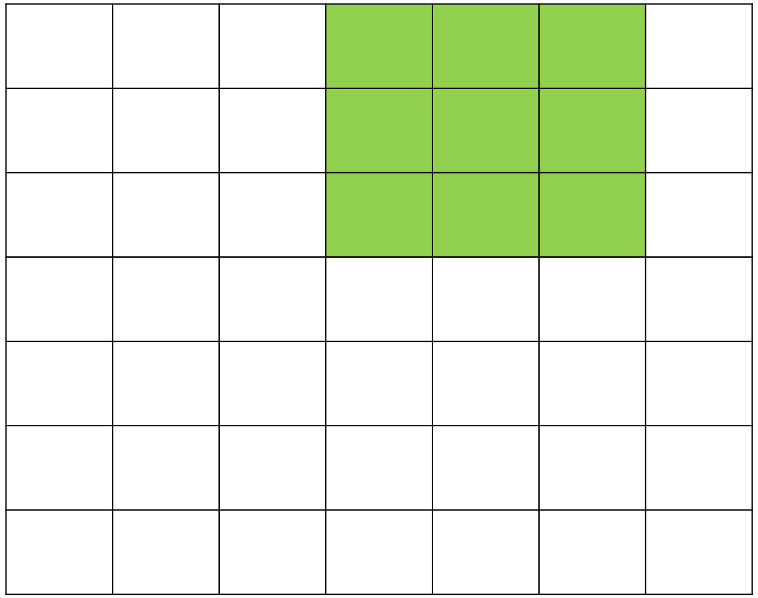


Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

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

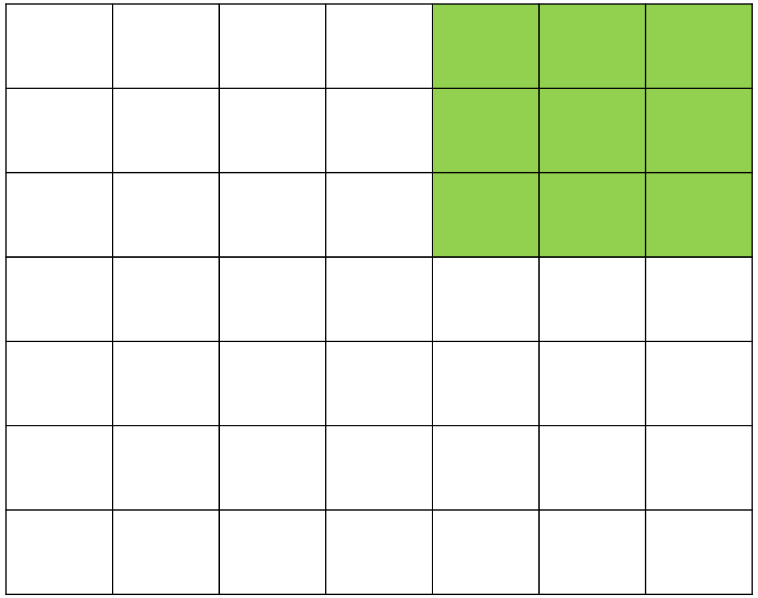


Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

Двумерная свертка

2D свёртка очень похожа на 1D.
Основное отличие состоит в том, что теперь нам приходится проводить итерации по 2 осям вместо 1.

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

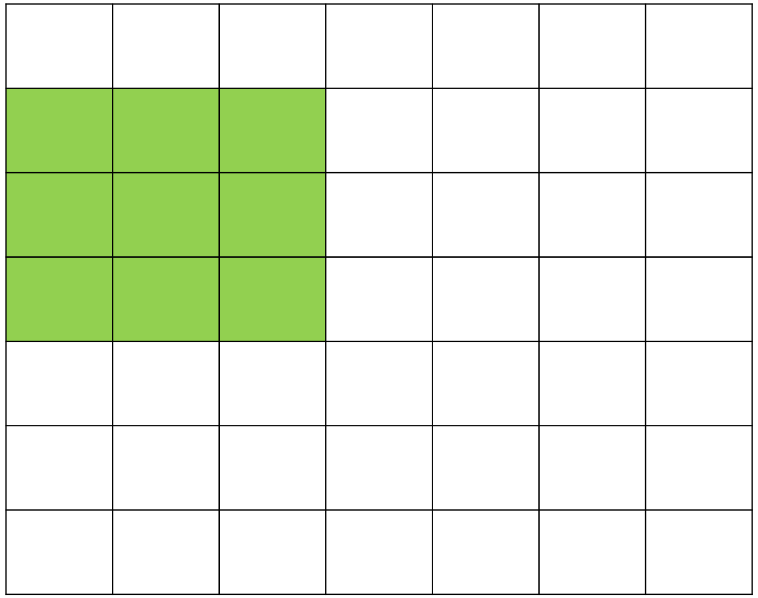


Предположим, что у нас есть фильтр(h[,]) размером 3x3 и изображение (f[,]) размером 7x7.

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

1	2	3
4	5	6
7	8	9

Input

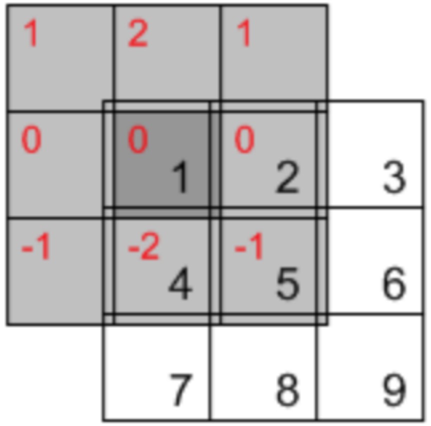
n \ m	-1	0	1
-1	-1	-2	-1
0	0	0	0
1	1	2	1

Kernel

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

Output

Пример двумерной свертки

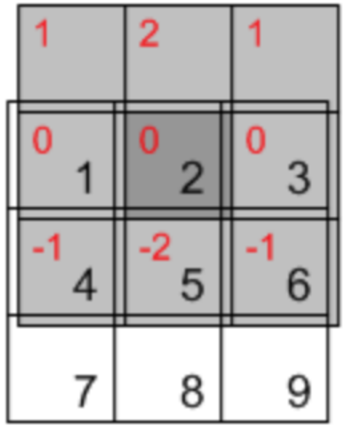


$$
\begin{aligned}
y[0,0] = {} & x[-1,-1]\cdot h[1,1] + x[0,-1]\cdot h[0,1] + x[1,-1]\cdot h[-1,1] \\
& + x[-1,0]\cdot h[1,0] + x[0,0]\cdot h[0,0] + x[1,0]\cdot h[-1,0] \\
& + x[-1,1]\cdot h[1,-1] + x[0,1]\cdot h[0,-1] + x[1,1]\cdot h[-1,-1] \\
= {} & 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 0\cdot 0 + 1\cdot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

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

Output

Пример двумерной свертки

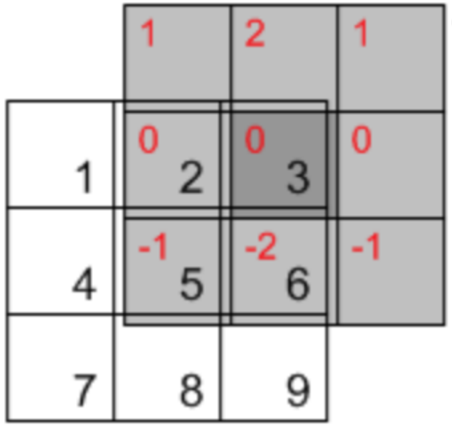


$$
\begin{aligned}
y[1,0] = {} & x[0,-1]\cdot h[1,1] + x[1,-1]\cdot h[0,1] + x[2,-1]\cdot h[-1,1] \\
& + x[0,0]\cdot h[1,0] + x[1,0]\cdot h[0,0] + x[2,0]\cdot h[-1,0] \\
& + x[0,1]\cdot h[1,-1] + x[1,1]\cdot h[0,-1] + x[2,1]\cdot h[-1,-1] \\
= {} & 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 1\cdot 0 + 2\cdot 0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

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

Output

Пример двумерной свертки

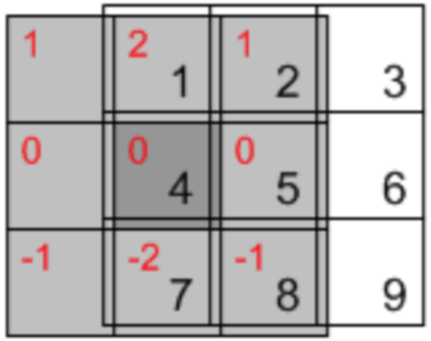


$$
\begin{aligned}
y[2,0] &= x[1,-1]\cdot h[1,1] + x[2,-1]\cdot h[0,1] + x[3,-1]\cdot h[-1,1] \\
&\quad + x[1,0]\cdot h[1,0] + x[2,0]\cdot h[0,0] + x[3,0]\cdot h[-1,0] \\
&\quad + x[1,1]\cdot h[1,-1] + x[2,1]\cdot h[0,-1] + x[3,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 2\cdot 0 + 3\cdot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

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

Output

Пример двумерной свертки



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

-13	-20	-17
$\boxed{-18}$	-24	-18
13	20	17

Output

Пример двумерной свертки

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

Пример двумерной свертки

*

•0	•0	•0
•0	•1	•0
•0	•0	•0

=

Пример двумерной свертки

*

•0	•0	•0
•0	•0	•1
•0	•0	•0

= ?

Пример двумерной свертки

*

•0	•0	•0
•0	•0	•1
•0	•0	•0

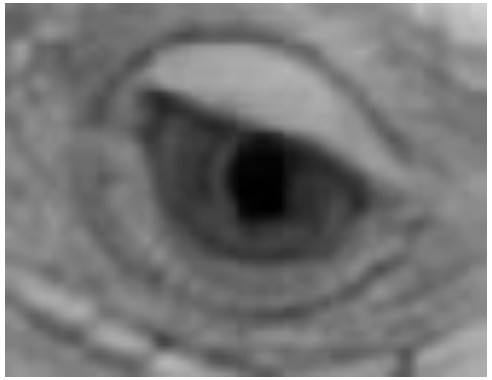
=

Пример двумерной свертки

$$* \ \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} = \ ?$$

Пример двумерной свертки

$$* \ \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} =$$



Пример двумерной свертки

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = ?$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Что отнимает размытость?

Оригинальное изображение

\-

Размытое

=

Детали

Оригинальное изображение

\+

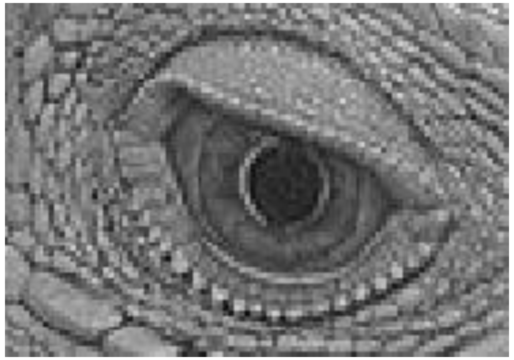
Детали

=

Повышенная резкость

Пример двумерной свертки – фильтр резкости

$$\begin{bmatrix} \bullet 0 & \bullet 0 & \bullet 0 \\ \bullet 0 & \bullet 2 & \bullet 0 \\ \bullet 0 & \bullet 0 & \bullet 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} =$$

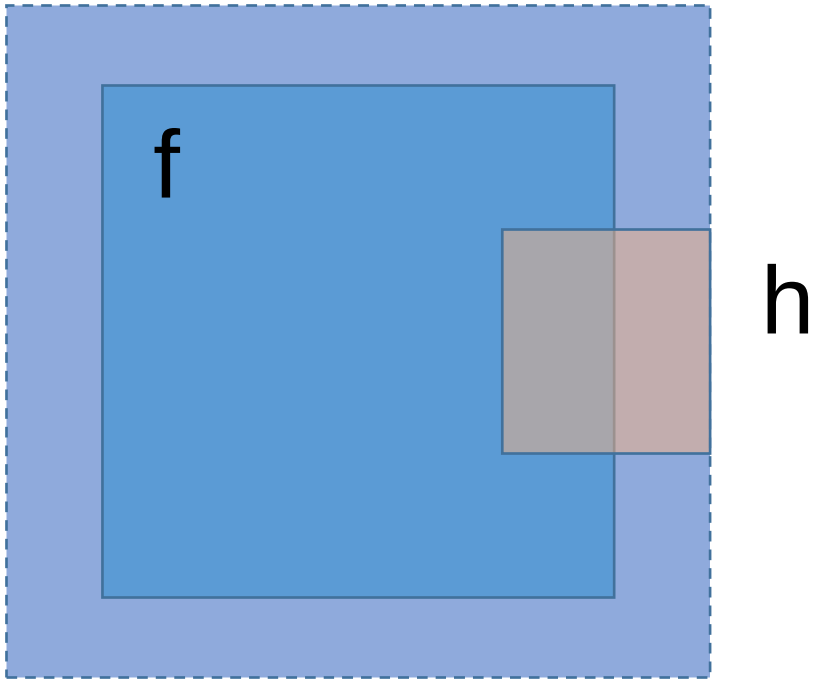


Фильтр резкости: подчеркивает разность со средним местным значениями пикселей

Краевой эффект

Компьютер будет вызывать только **конечные сигналы**

Что происходит на краю?



- нулевой паддинг
- повторение на краях
- отзеркаливание

Кросс-корреляция

Кросс-корреляция (корреляция / взаимнокорреляционная фнукция) в точке (n, m) lkz 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
- «Сходство» между функциями f и h

Разница свертки и кросс-корреляции

Для 1D сигнала

1. свёртка - поэлементное произведение массива (вектор)

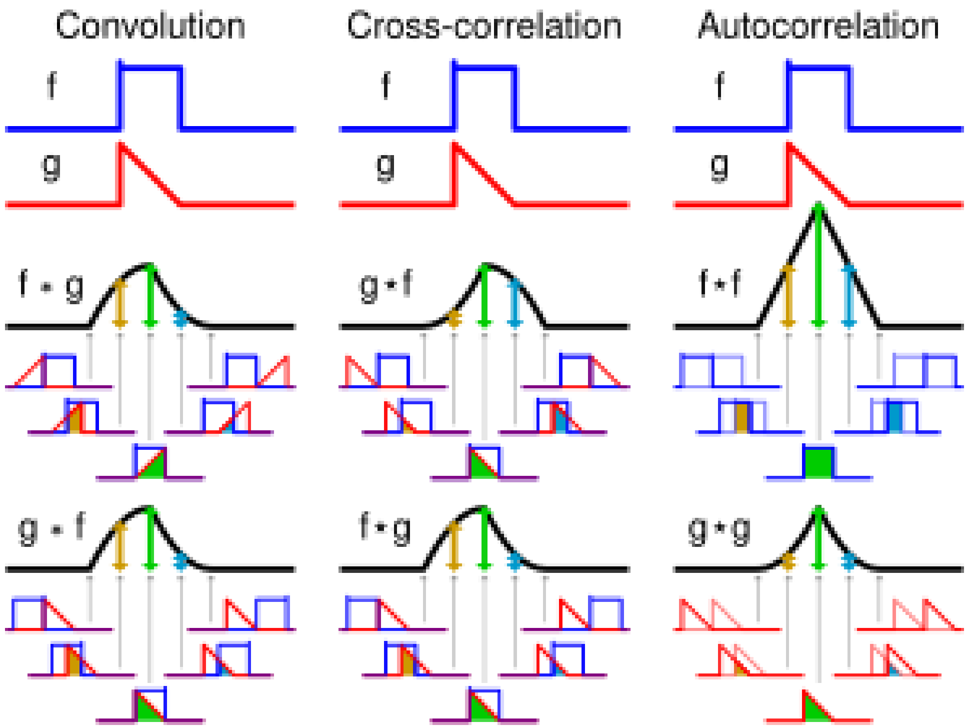
** индексация для суммы отсчётов сигнала и фильтра противоположна*

Пример – фильтрация (размытие)

2. корреляция - сумма этого поэлементного произведения (скаляр)

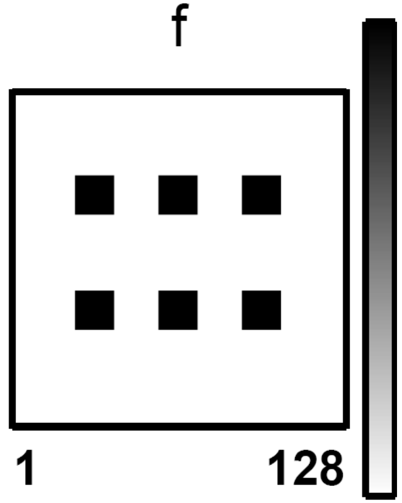
** индексация для суммы отсчётов сигнала и фильтра однонаправленна*

Пример – поиск по шаблону

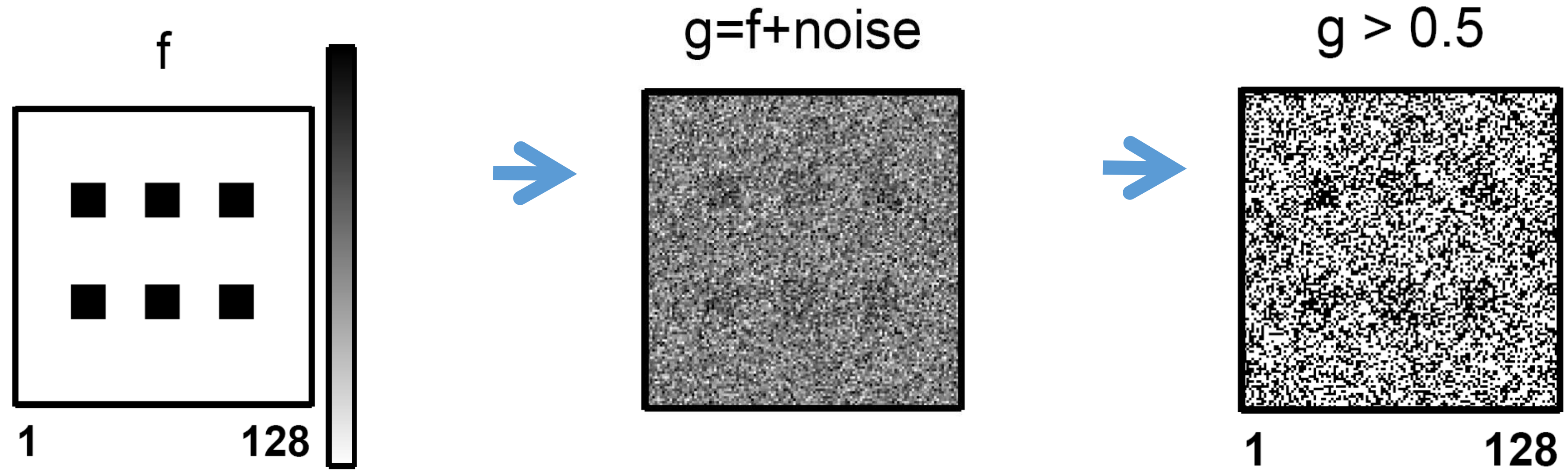


Слева направо: свёртка, взаимная корреляция и автокорреляция

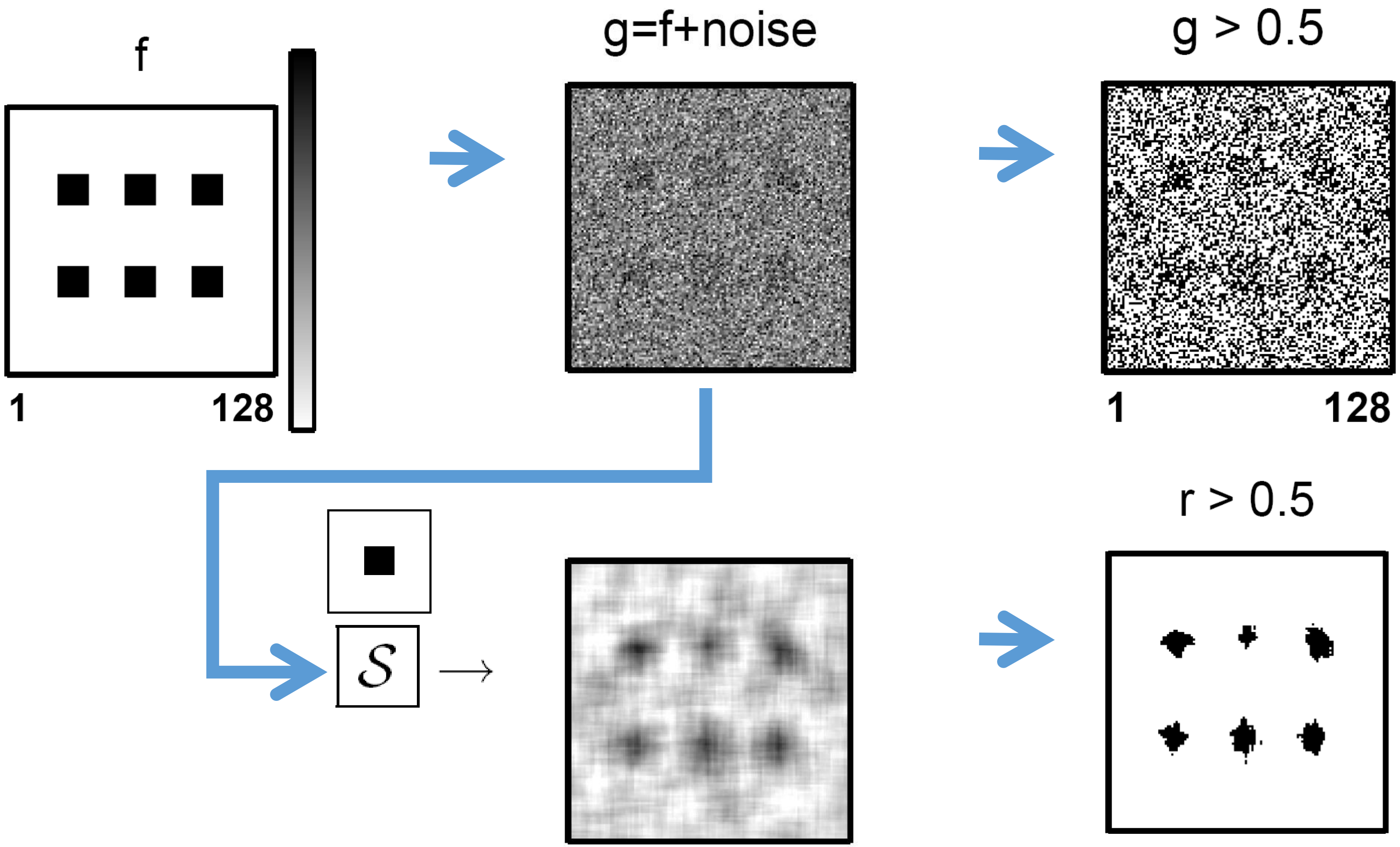
Пример кросс-корреляции



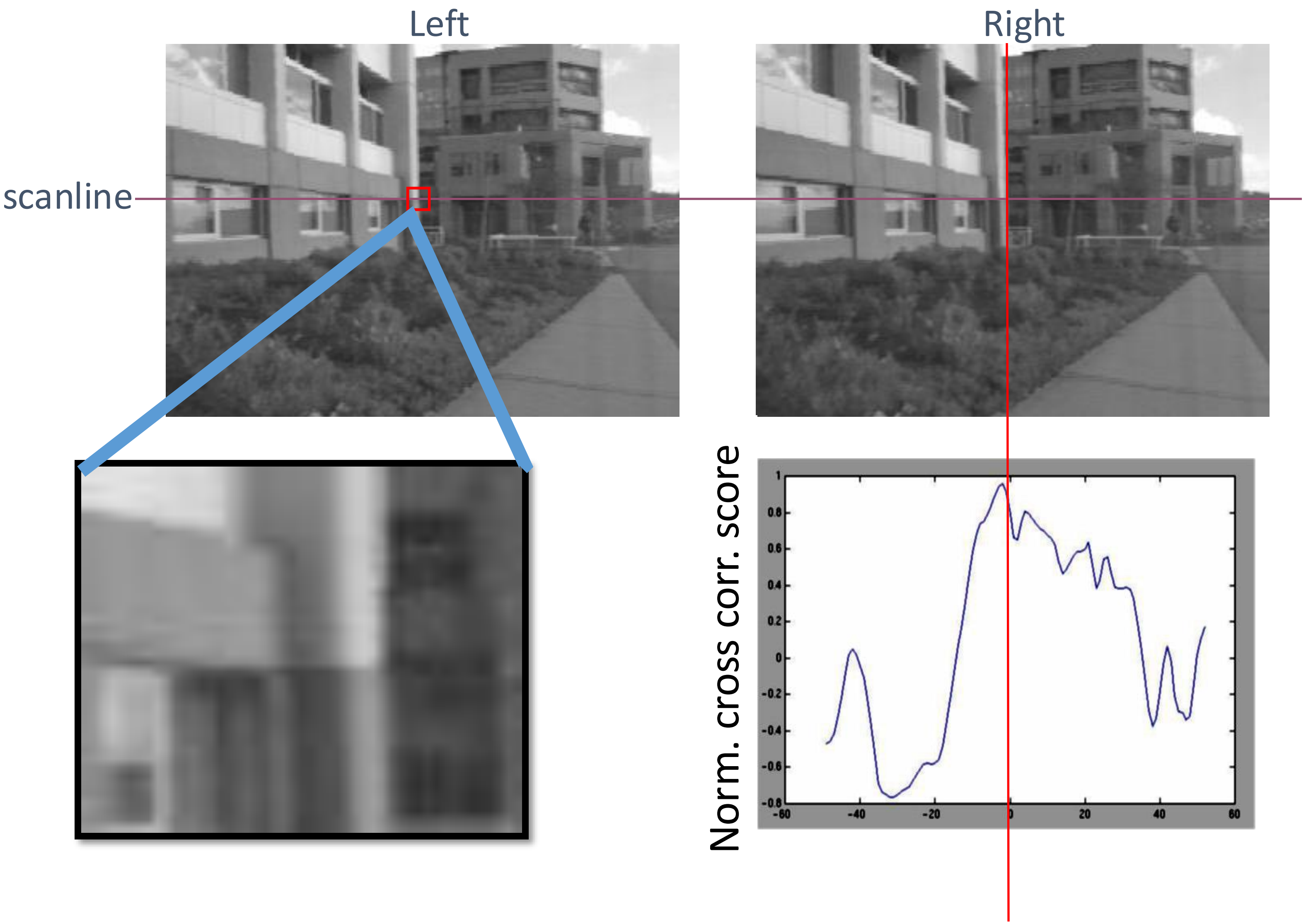
Пример кросс-корреляции



Пример кросс-корреляции



Пример кросс-корреляции



Место для ваших вопросов