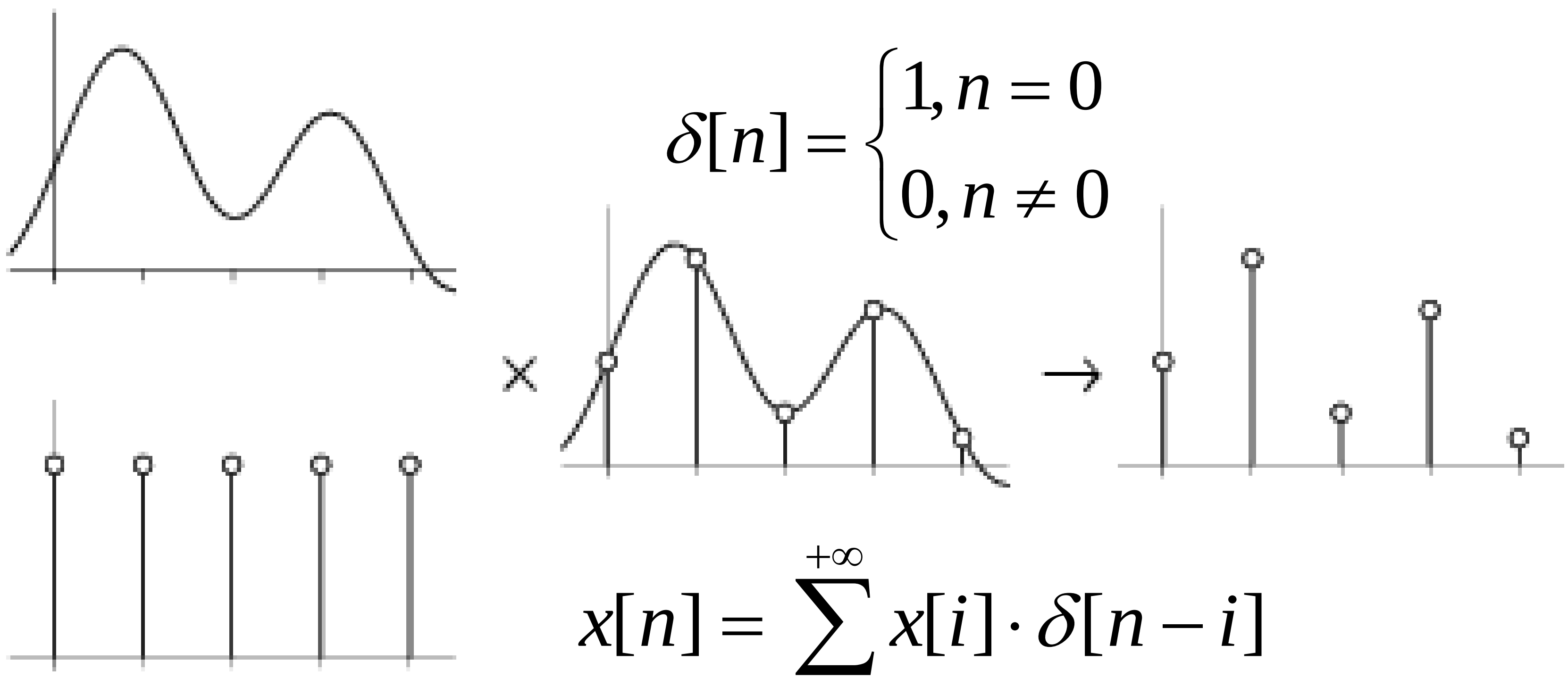
Частотные и пространственные преобоазования

The Discrete Fourier Transform

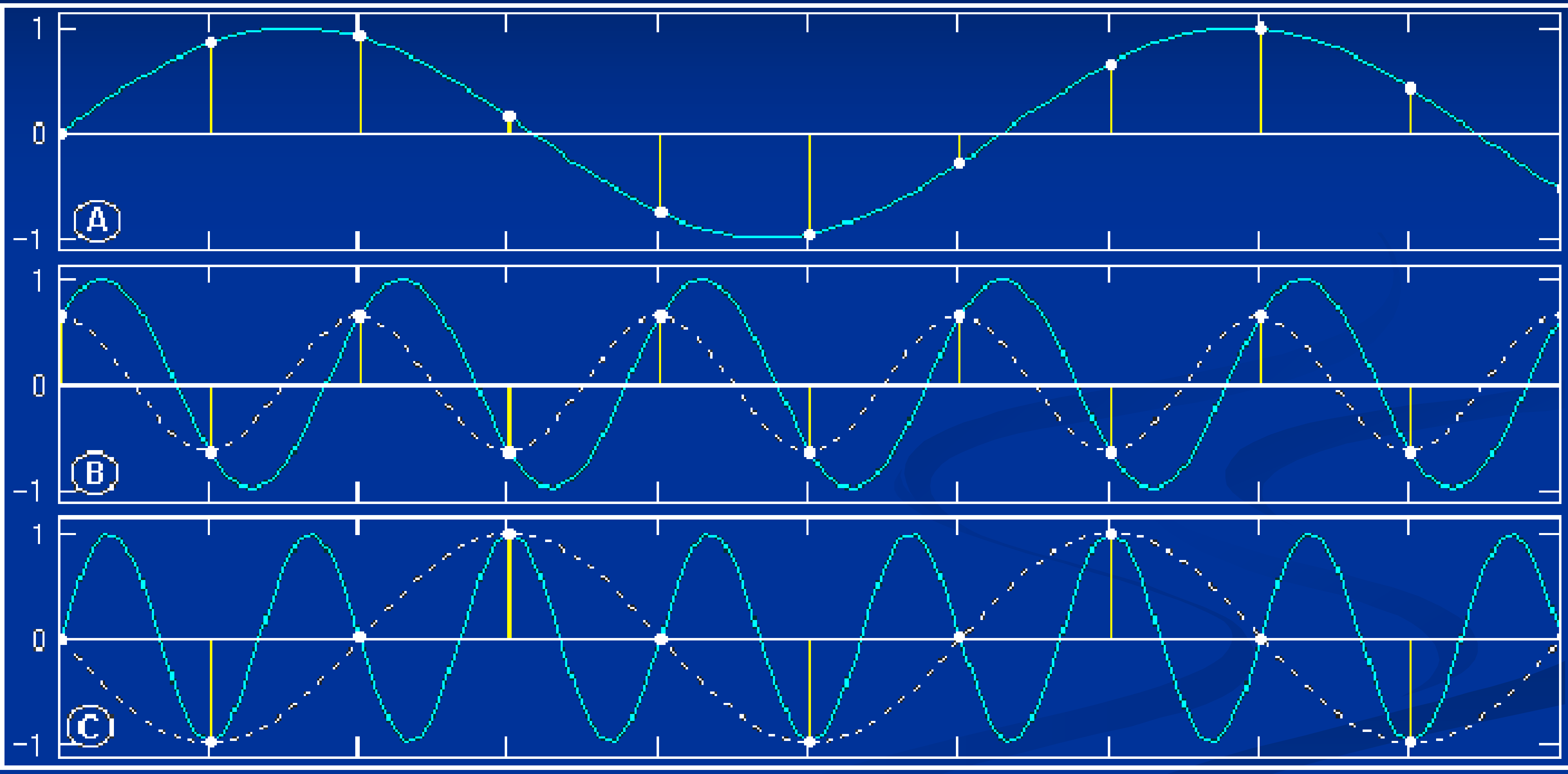
Дискретизация во временной области



$$\delta[n] = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases}$$

$$x[n] = \sum_{i=-\infty}^{+\infty} x[i] \cdot \delta[n-i]$$

Дискретизация гармоник с разной частотой



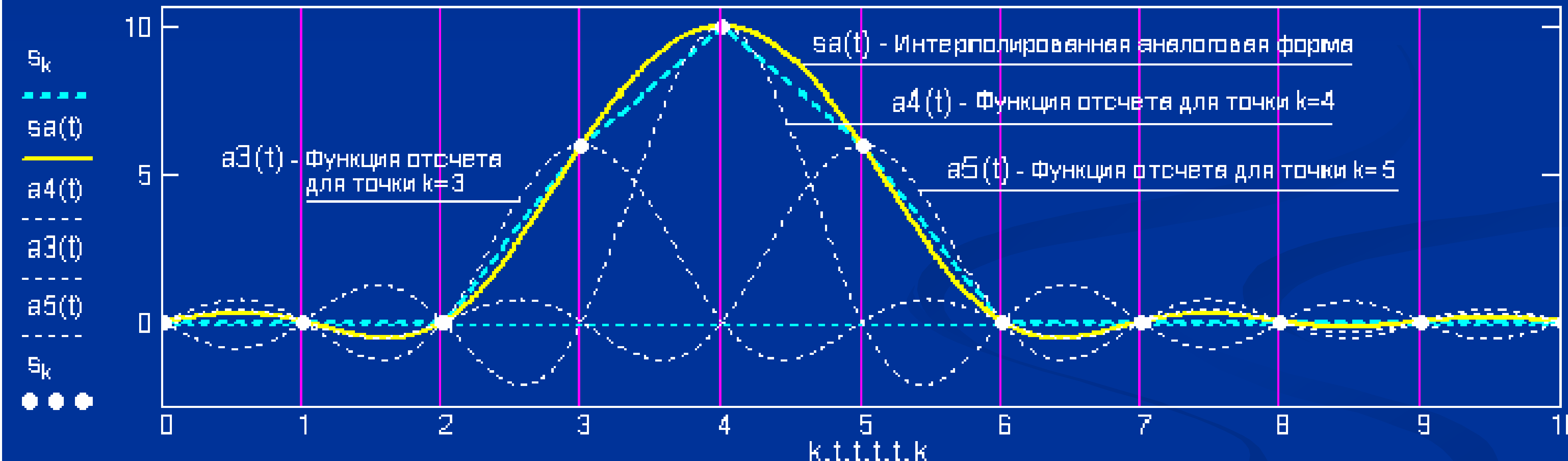
MATHCAD

Интерполяция функции рядом Котельникова-Шеннона

$K := 10 \qquad k := 0..K \qquad s_k := 0 \qquad s_4 := 10 \qquad s_3 := 6 \qquad s_5 := 6 \qquad t := 0, 0.01..K$

$\mathrm{sinc}(x) := \mathrm{if}\left(x = 0, 1, \dfrac{\sin(\pi \cdot x)}{\pi \cdot x}\right) \qquad sa(t) := \sum_k s_k \cdot \mathrm{sinc}(t - k)$ <= Функция отсчетов и ряд Котельникова-Шеннона

$a4(t) := s_4 \cdot \mathrm{sinc}(t - 4) \qquad a3(t) := s_3 \cdot \mathrm{sinc}(t - 3) \qquad a5(t) := s_3 \cdot \mathrm{sinc}(t - 5)$ <= Значимые члены ряда



MATHCAD

Интерполяционный ряд Котельникова-Шеннона

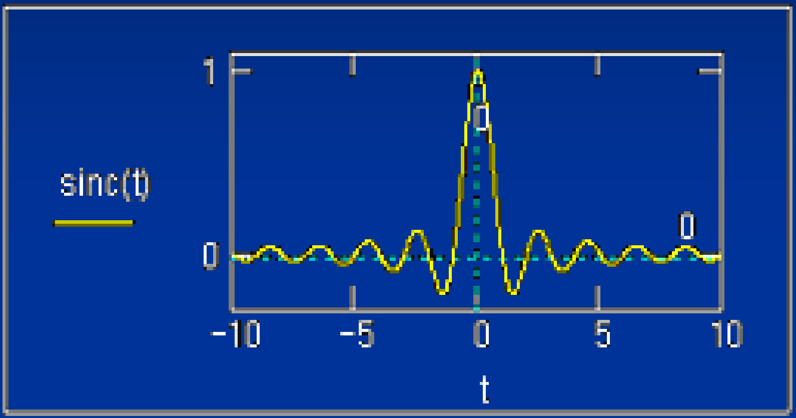
$K = 10 \quad k = 0..K \quad y^T =$

	0	1	2	3	4	5	6	7	8	9	10
0	0	1.17	2.91	6	10.1	7.84	4.23	5.73	4.32	0.7	0

<= Входные данные

$$z(t) = \frac{\sin(\pi \cdot t)}{\pi \cdot t} \qquad \mathrm{sinc}(t) = \mathrm{if}(t = 0, 1, z(t))$$

<= Функция отсчетов (интегральный синус).



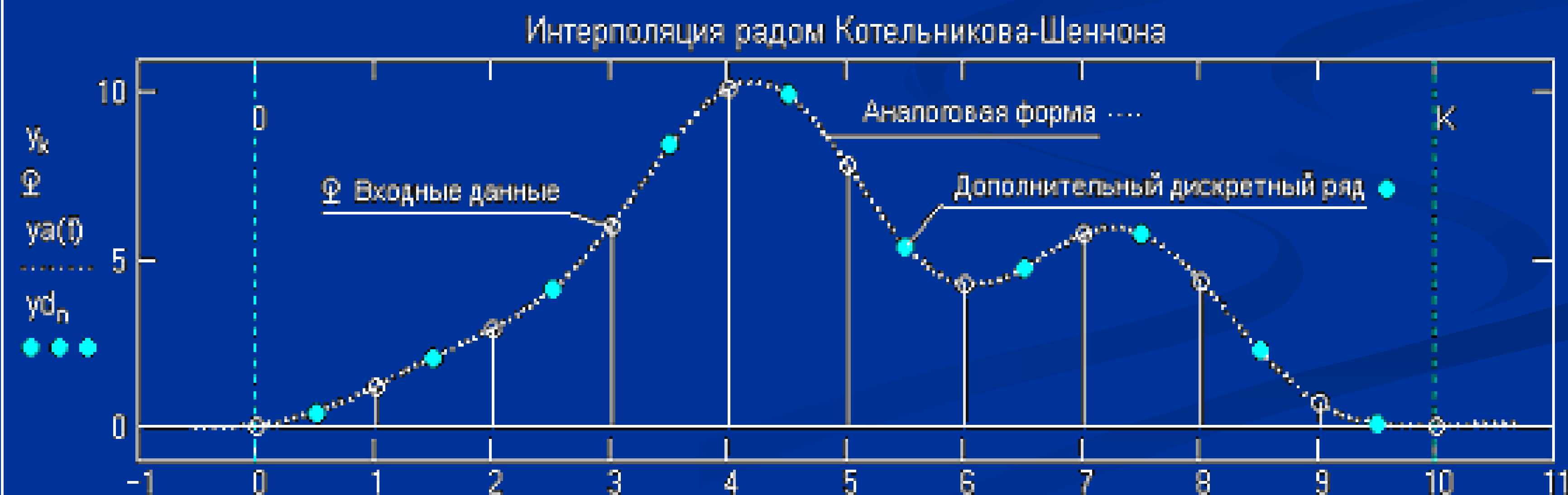
$$ya(t) = \sum_{k=0}^{K} y_k \cdot \mathrm{sinc}(t - k)$$

<= Восстановление аналоговой формы

$$\Delta t = 0.5 \qquad n = 1, 3.. \frac{K}{\Delta t}$$

$$yd_n = \sum_{k=0}^{K} y_k \cdot \mathrm{sinc}(n \cdot \Delta t - k)$$

<= Изменение шага дискретных данных

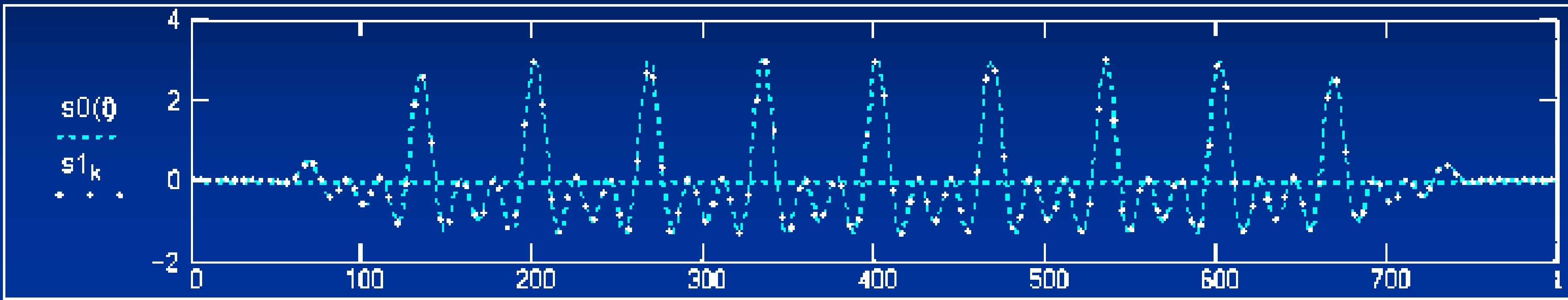


MATHCAD

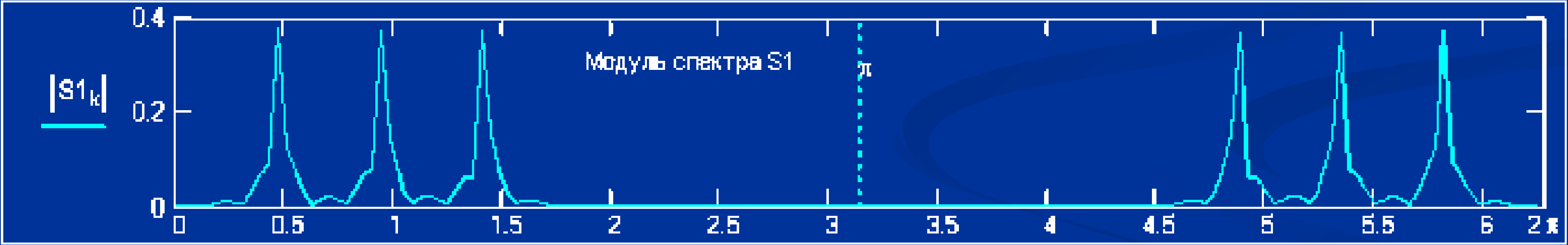
Быстрое преобразование Фурье

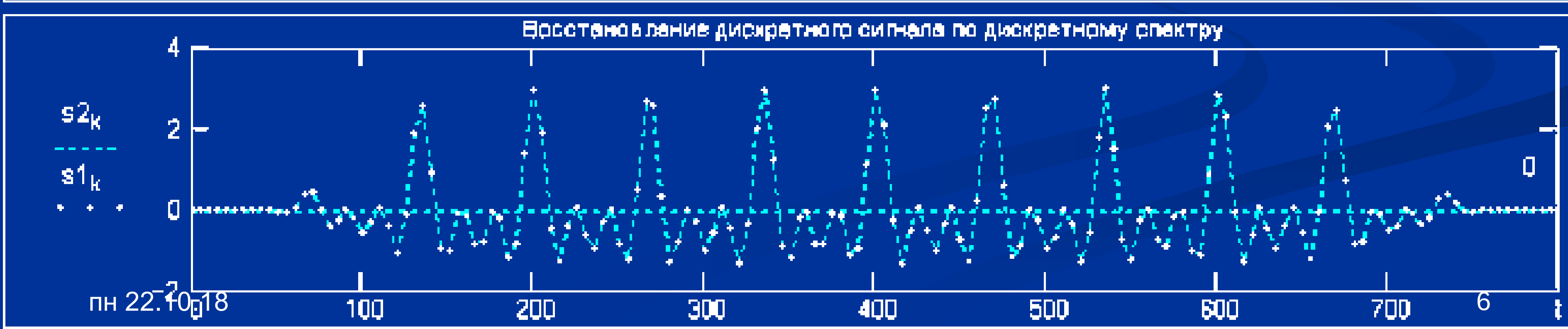
$T := 800 \quad t := 0..T \quad u1(t) := \exp\left[-1\cdot 10^{-30}\cdot(t-400)^{12}\right] \quad v1(t) := \sum_{k=1}^{3}\cos(0.094\cdot t\cdot k) \quad s0(t) := u1(t)\cdot v1(t)$ <= модель

$\Delta t := 5 \quad K := 2\cdot \mathrm{floor}\left(\frac{T}{\Delta t\cdot 2}\right) \quad K = 160 \quad N := K+1 \quad N = 161 \quad k := 0..K \quad s1_k := s0(k\cdot\Delta t)$ <= Дискретизация



$S1 := \mathrm{CFFT}(s1) \quad s2 := \mathrm{ICFFT}(S1) \quad \Delta\omega := \frac{2\pi}{N}$ <= Прямое и обратное быстрое преобразование Фурье





Свертка

$$f(x)\otimes g(x)=\int_{-\infty}^{+\infty} f(x-t)\,g(t)\,dt$$

- Свертка (перемножение) двух функций эквивалентна перемножению (свертке) их образов Фурье
- Фильтрация: умножение на фильтр в частотной области или свертка в пространственной.

Свертка

$$h(x) = f(x) * g(x) = \int_{-\infty}^{+\infty} f(\tau) g(x-\tau) d\tau$$

For all pixels i falling under filter support centered at x

$$h(x) = \sum_i P_i g(x-i) = \sum_i P_i g(i-x)$$

Filter value at pixel location i

Pixel value at i

Ядро свертки

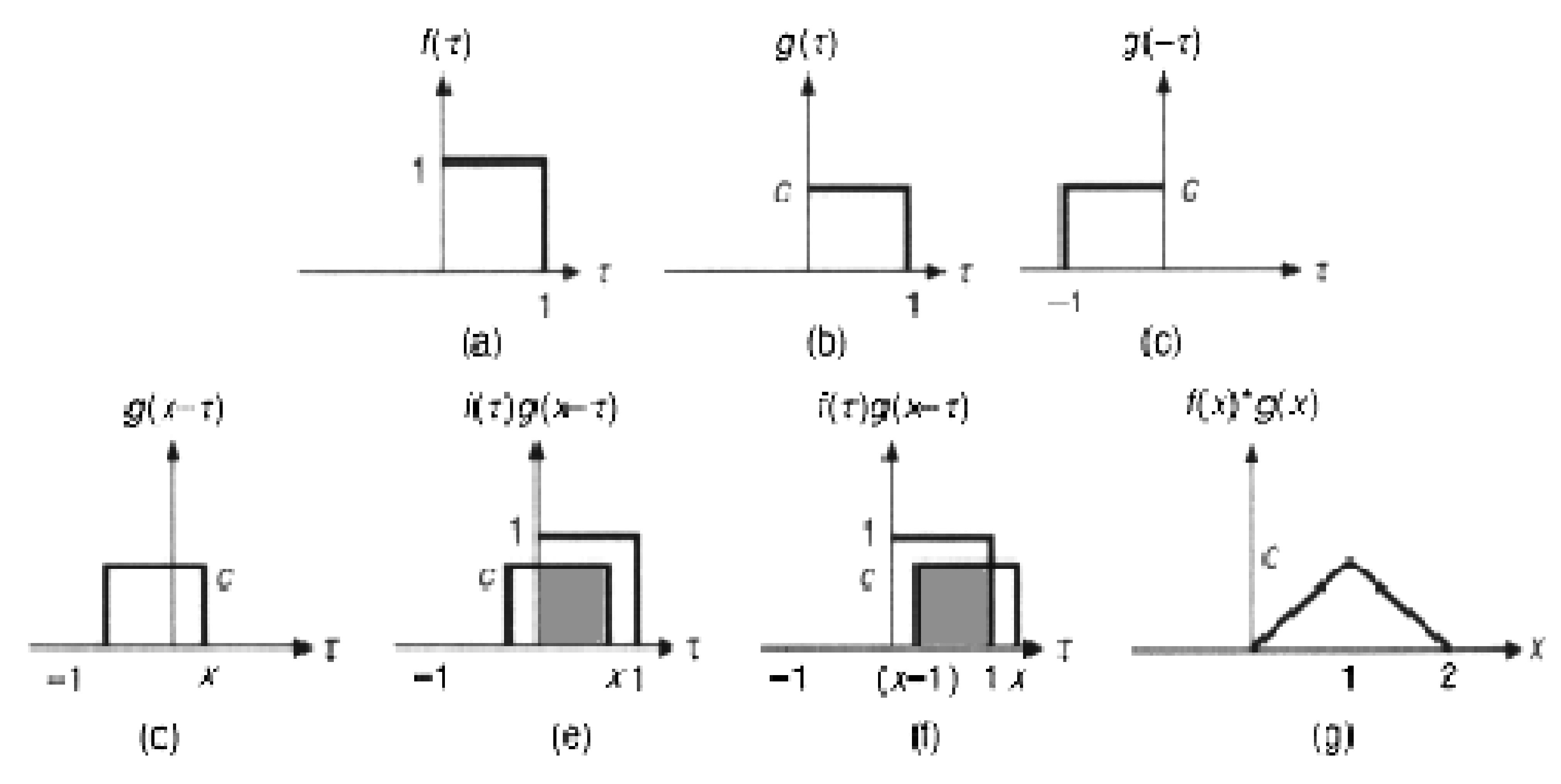
$$h(x, a) = \sum_i P_i g(i - x, a)$$

for all pixels *i*
where *i* is in support of *g*

Pixel at integer *i*

filter *g*, centered at *x*,
evaluated at *i*

$$\hat{h}(k, a) = h\left(\frac{k}{a}, a\right) = \sum_i P_i g\left(i - \frac{k}{a}, a\right)$$



f(τ)
1
τ
1
(a)
g(τ)
c
τ
1
(b)
g(−τ)
c
τ
−1
(c)
g(x−τ)
c
τ
−1
x
(c)
f(τ)g(x−τ)
1
c
τ
−1
x 1
(e)
f(τ)g(x−τ)
1
c
τ
−1
(x−1) 1 x
(f)
f(x)*g(x)
c
x
1
2
(g)

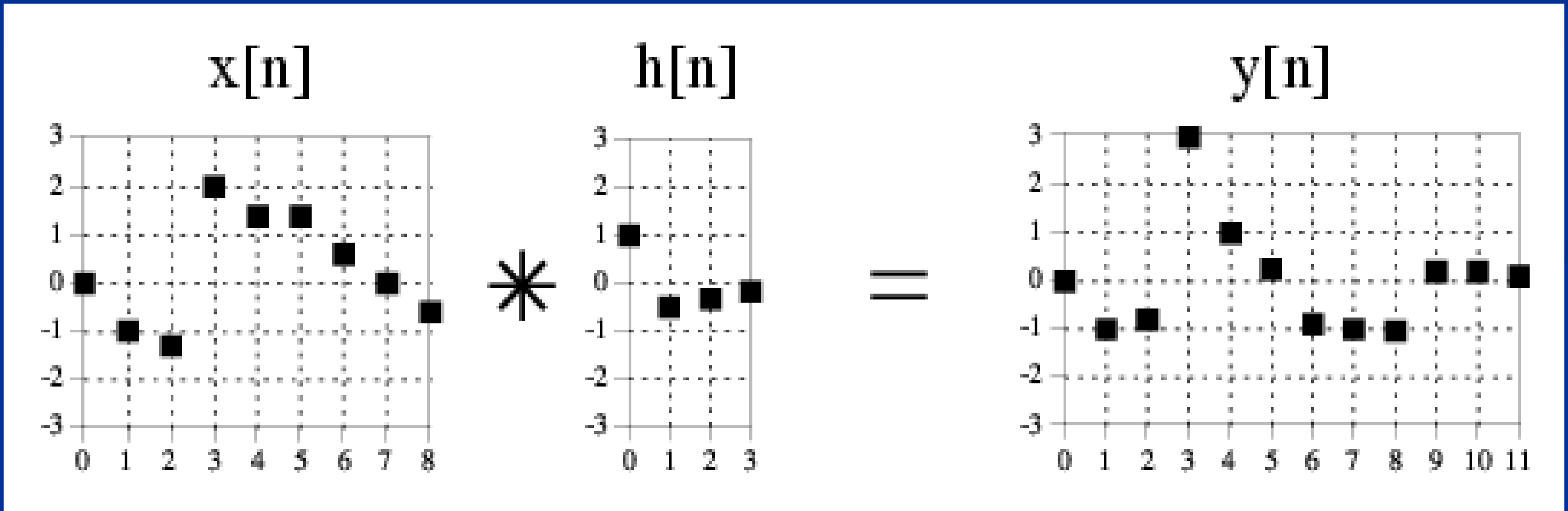
Дискретная свертка

- Двумерный случай

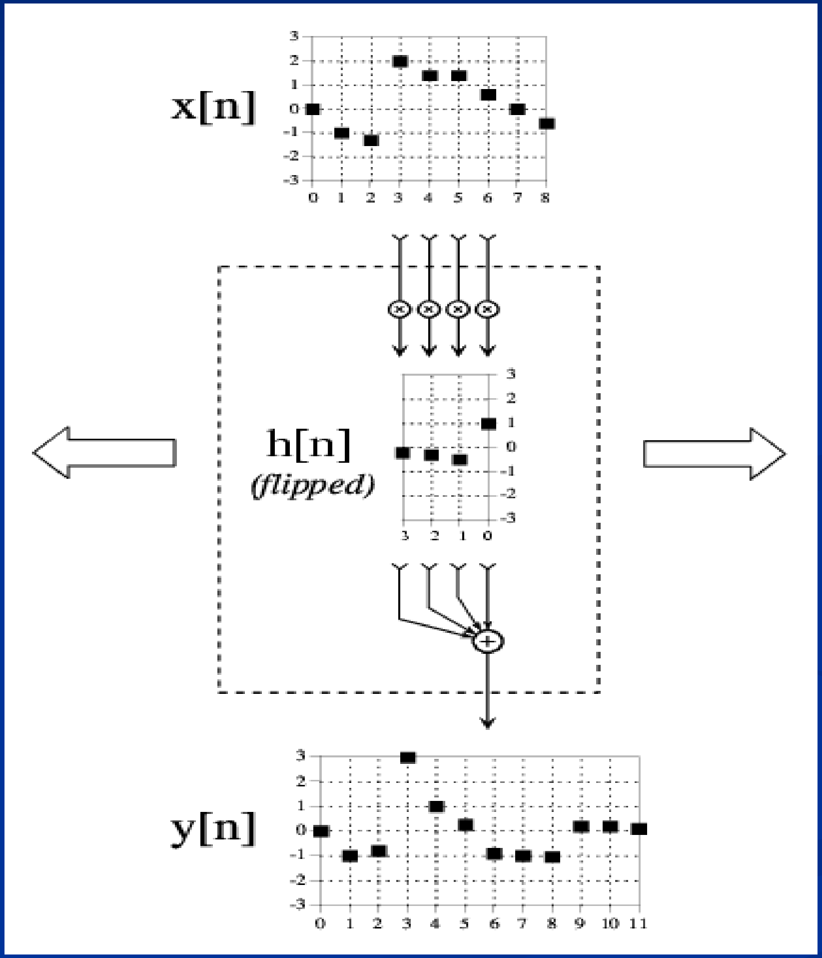
$$h = f \otimes g: \quad h_{i,j} = \sum_{l=L_0}^{L_1} \sum_{m=M_0}^{M_1} f_{i-l,j-m} g_{l,m}$$

- g называется *фильтром* или *ядром свертки*
- Замечание: с точки зрения математики f и g абсолютно равноправны...

Пример свертки



Машина свертки



Преобразование Фурье

- Преобразование Фурье (образ Фурье):

$$F(\omega) = \int_{-\infty}^{+\infty} f(x) \exp(-i\omega x)\, dx$$

$$A(\omega) = \left| F(\omega) \right|, \quad \operatorname{tg} \alpha(\omega) = \arg F(\omega)$$

- Обратное преобразование Фурье:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \exp(i\omega x)\, d\omega$$

Дискретный случай

$$x[n] = \sum_{k=0}^{N/2} A_k \cos\frac{2\pi kn}{N} + \sum_{k=0}^{N/2} B_k \sin\frac{2\pi kn}{N} = \sum_{k=0}^{N/2} C_k \cos\frac{2\pi k(n+\varphi_k)}{N}$$

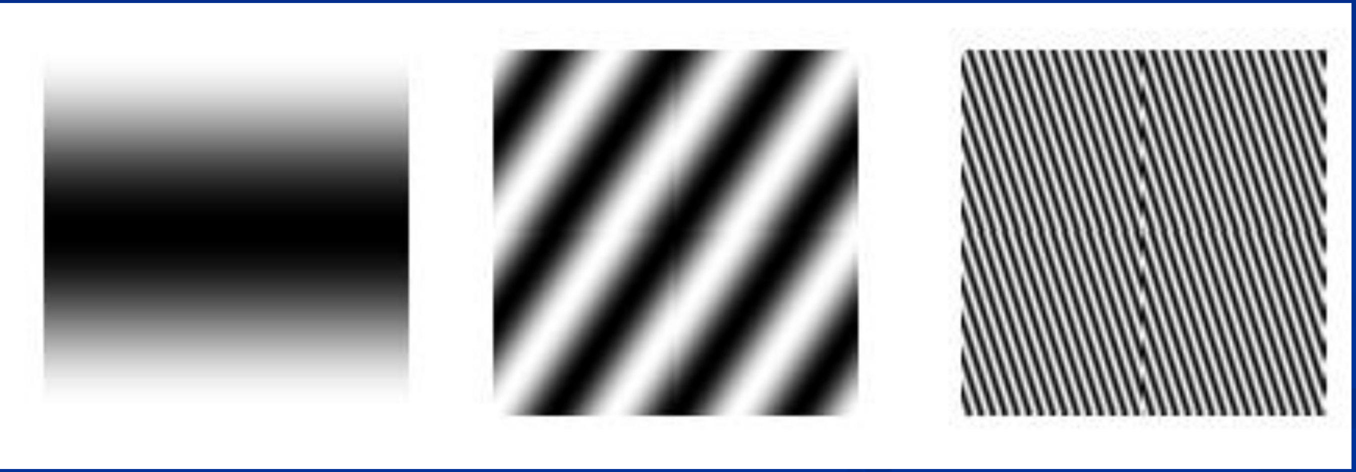
$$A_k = \frac{2}{N}\sum_{i=0}^{N-1} x[i]\cos\frac{2\pi ki}{N} \qquad k = 1,\ldots,\frac{N}{2}-1$$

$$A_k = \frac{1}{N}\sum_{i=0}^{N-1} x[i]\cos\frac{2\pi ki}{N} \qquad k = 0,\frac{N}{2}$$

$$B_k = \frac{2}{N}\sum_{i=0}^{N-1} x[i]\sin\frac{2\pi ki}{N} \qquad k = 0,\ldots,\frac{N}{2}$$

2D преобразование

Базисные функции имеют вид двумерных синусоид с разными углами наклона и фазами

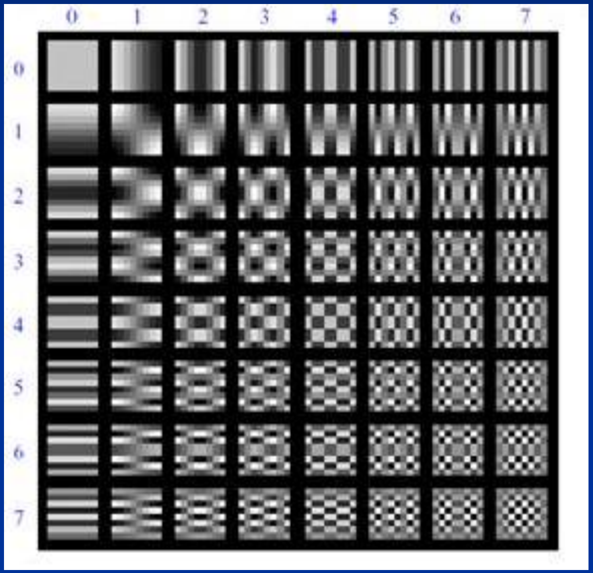


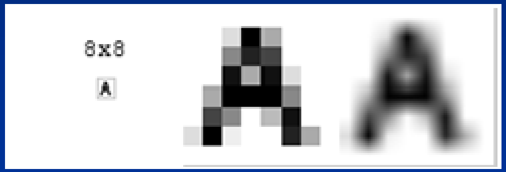
Вычисление двумерного ДПФ (ДКП, ДСП)

• Прямой способ – скалярные произведения со всеми базисными функциями. Очень много операций.

• Быстрый способ – декомпозиция на одномерные ДПФ, затем быстрое преобразование Фурье

Дискретное косинусное преобразование



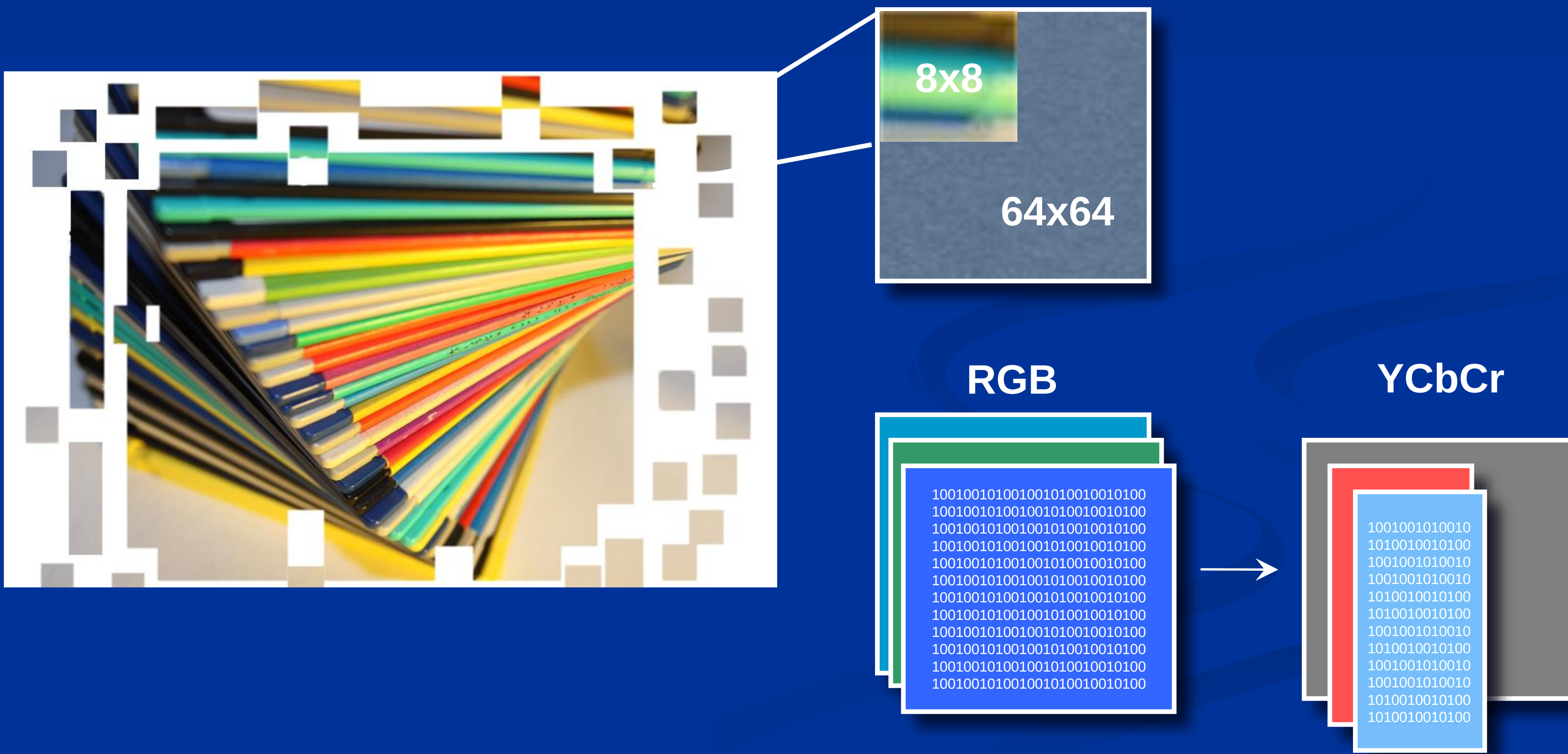


Базис для дискретного косинусного
преобразования (ДКП)

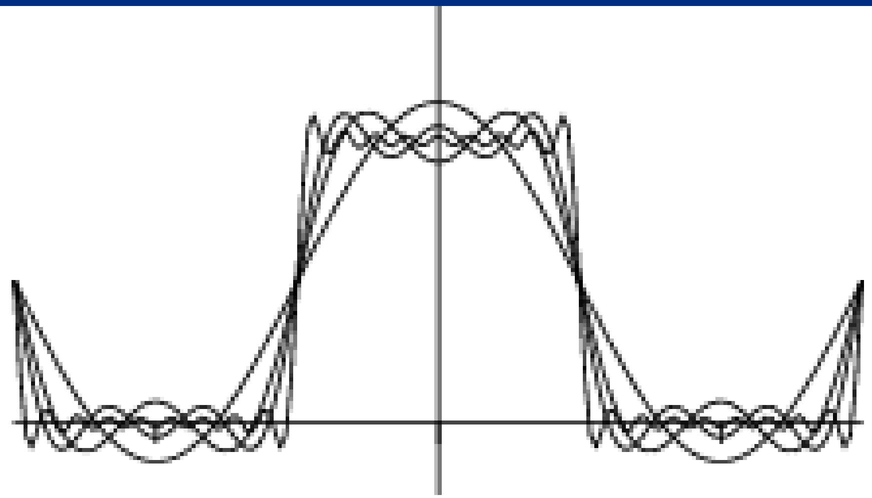
- Поблочное Дискретное Косинусное Преобразование (ДКП)
- Discrete Cosine Transform (DCT)

Описание формата JPEG

Трансформация цветового пространства со сжатием



Ряд Фурье для прямоугольного импульса



$$S_n(x) = \frac{1}{2} + \frac{2}{\pi}\left(\cos\omega x - \frac{1}{3}\cos 3\omega x + \frac{1}{5}\cos 5\omega x + \ldots\right)$$

$$= \frac{1}{2} + \frac{2}{\pi}\sum_{k=1}^{n}(-1)^k \frac{\cos(2k-1)\omega x}{2k-1}$$

Быстрое преобразование Фурье

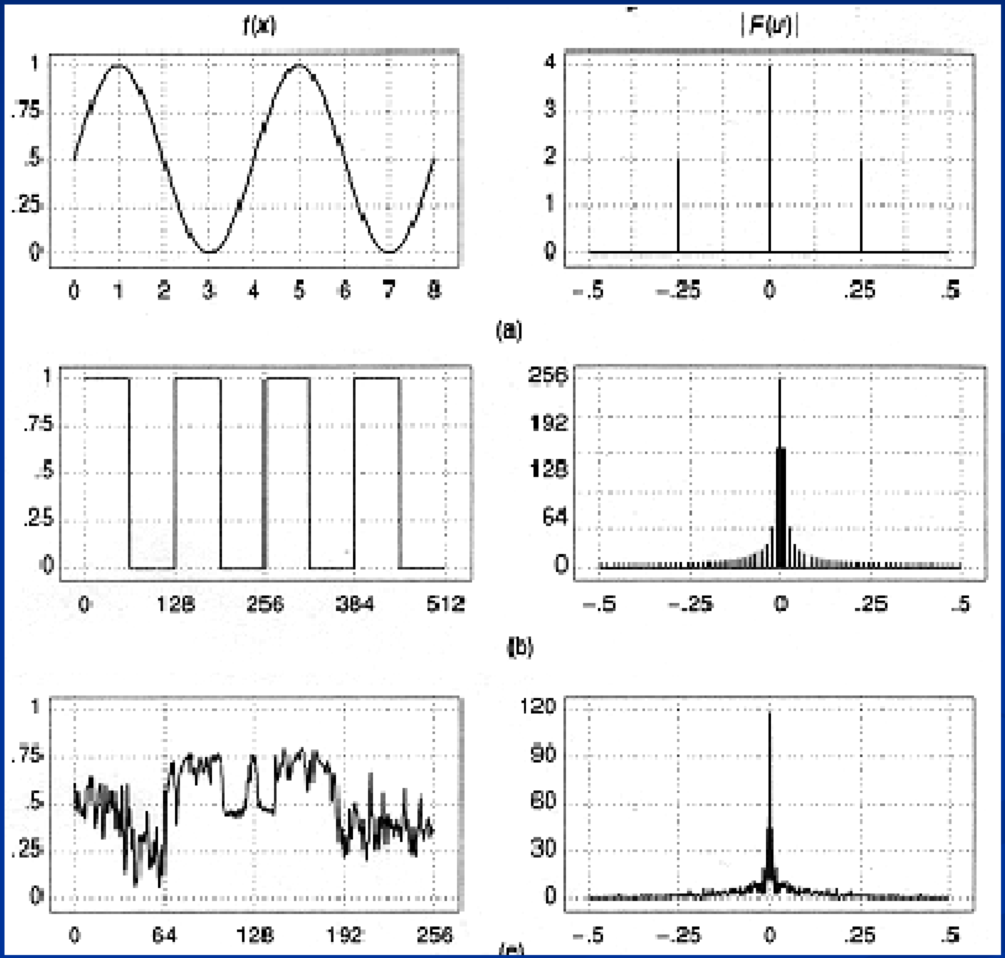
Для вычисления всех коэффициентов через скалярное произведение требуется примерно N^2 умножений: очень много при больших длинах сигнала N.

Быстрое преобразование Фурье (БПФ, FFT) – ускоренный алгоритм вычисления ДПФ

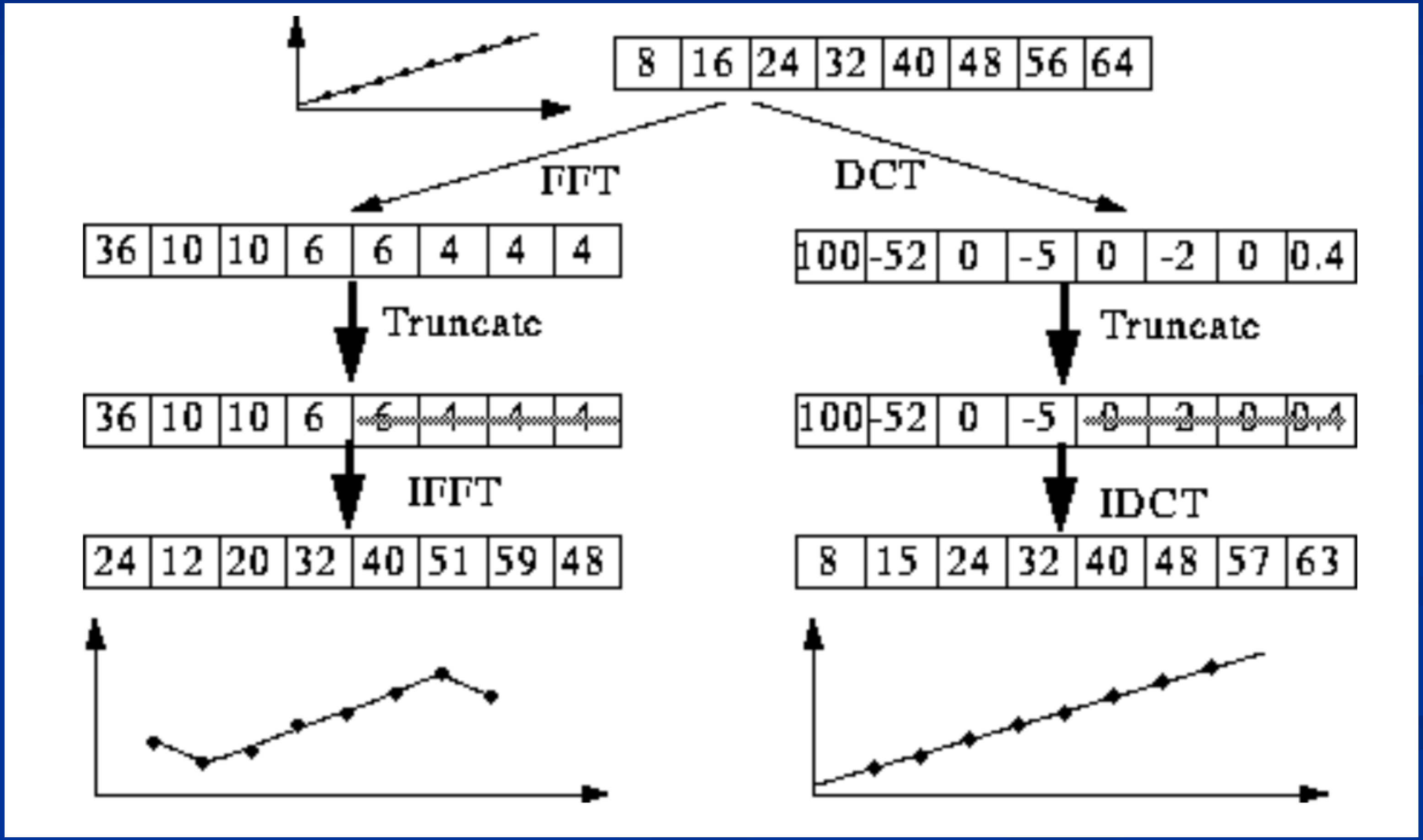
- Основан на периодичности базисных функций (много одинаковых множителей)
- Математически точен (ошибки округления даже меньше, т.к. меньше число операций)
- Число умножений порядка $N \log_2 N$, намного меньше, чем N^2
- Ограничение: большинство реализаций FFT принимают только массивы длиной $N = 2^m$

Есть и быстрое обратное преобразование

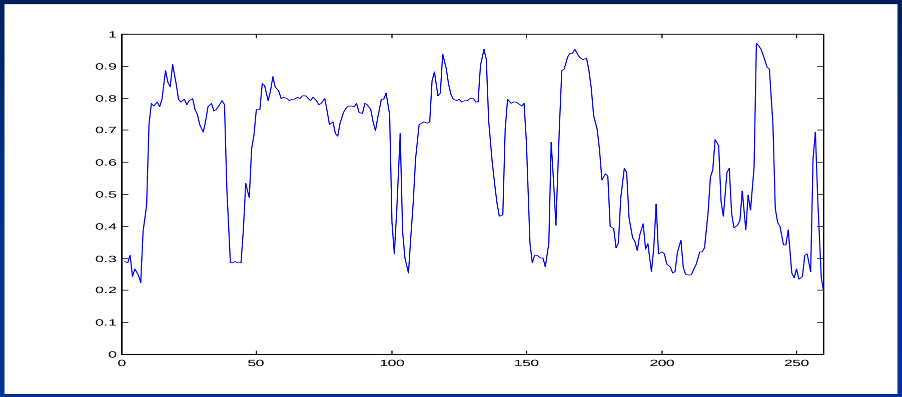
Спектр некоторых функций



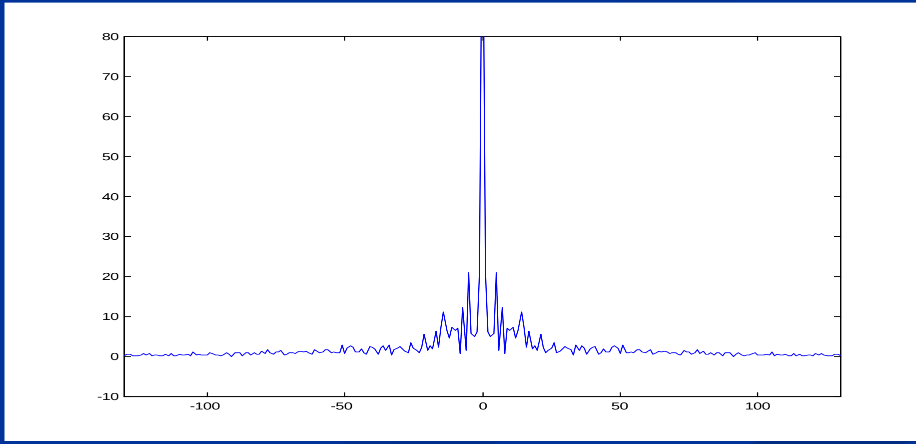
ДИСКРЕТНОЕ КОСИНУСНОЕ И ФУРЬЕ ПРЕОБРАЗОВАНИЕ



Пример преобразования Фурье

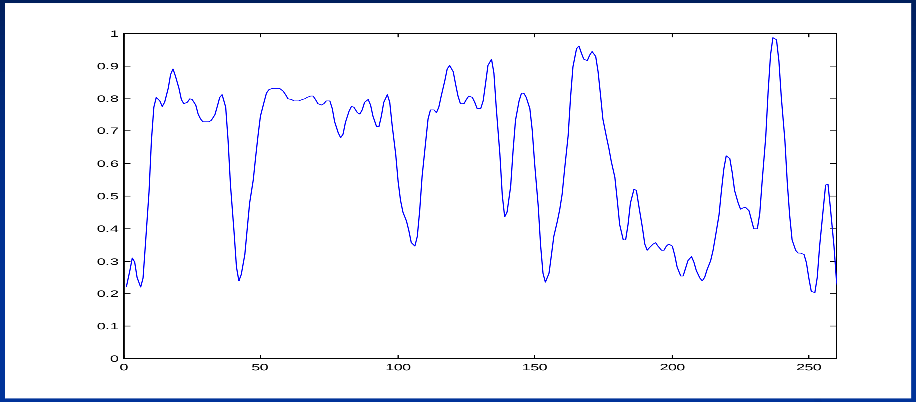


$f(x)$

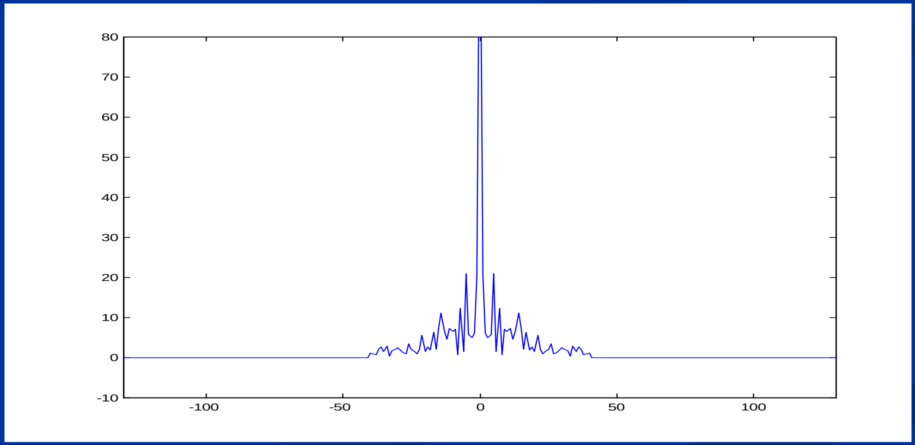


$|F(\omega)|$

Низкочастотная фильтрация



$f(x)$

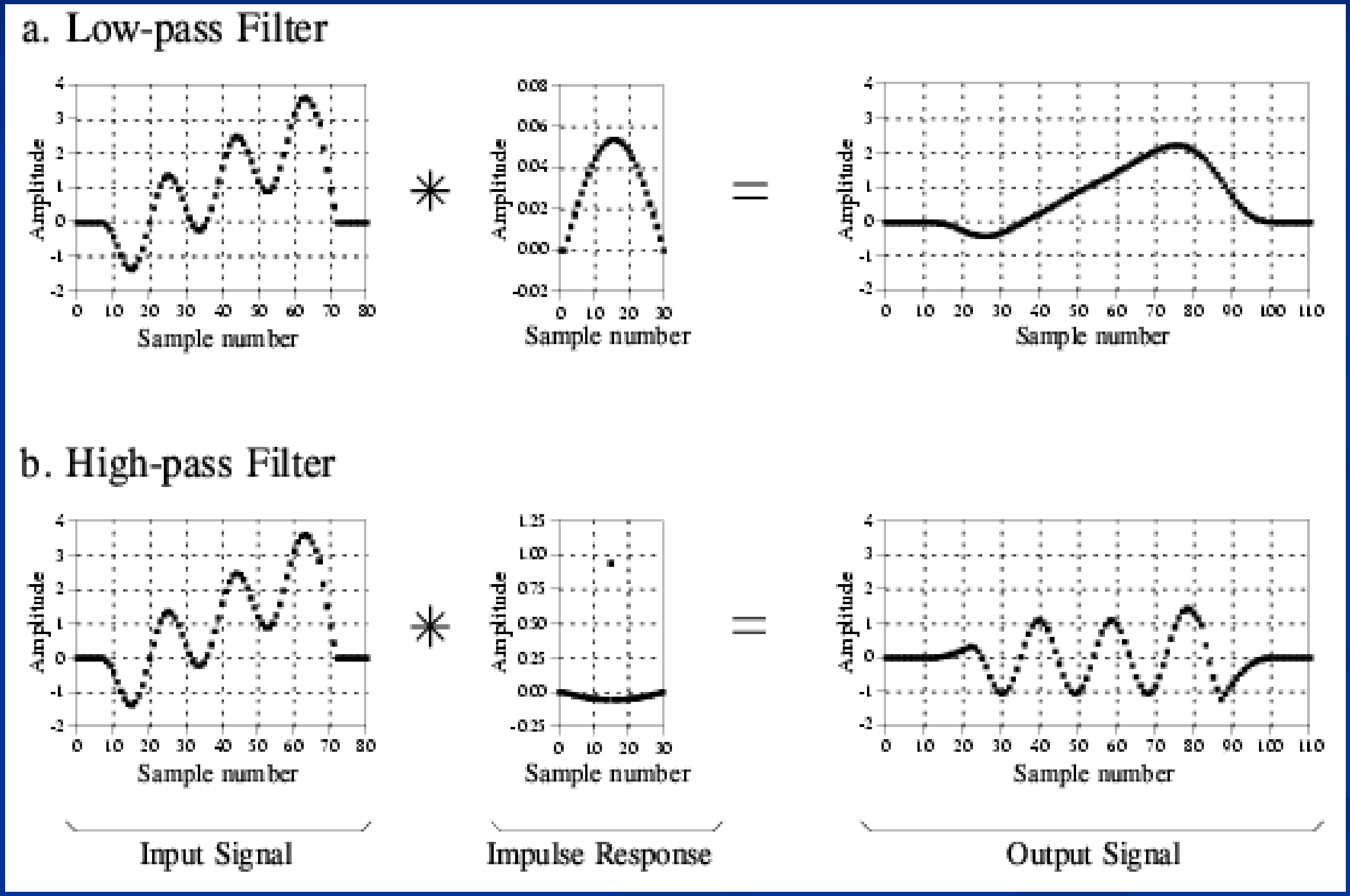


$|F(\omega)|$

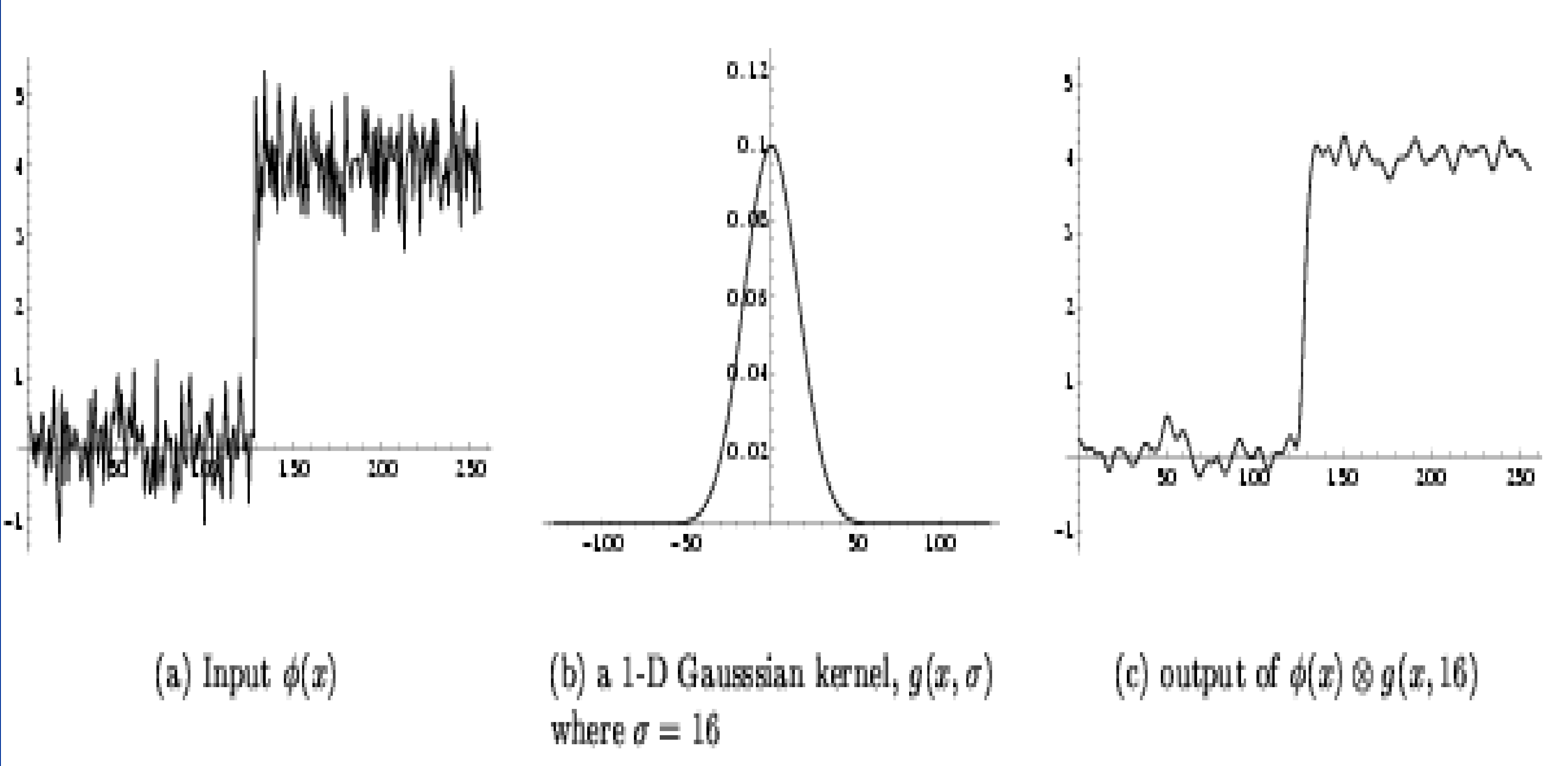
Фильтрация

- *Фильтрация* — выделение (или подавление) частотных составляющих сигнала.
- *Фильтр* — функция, осуществляющая фильтрацию.
- *Высоко-* и *низкочастотные* фильтры. (High-pass and low-pass filters)

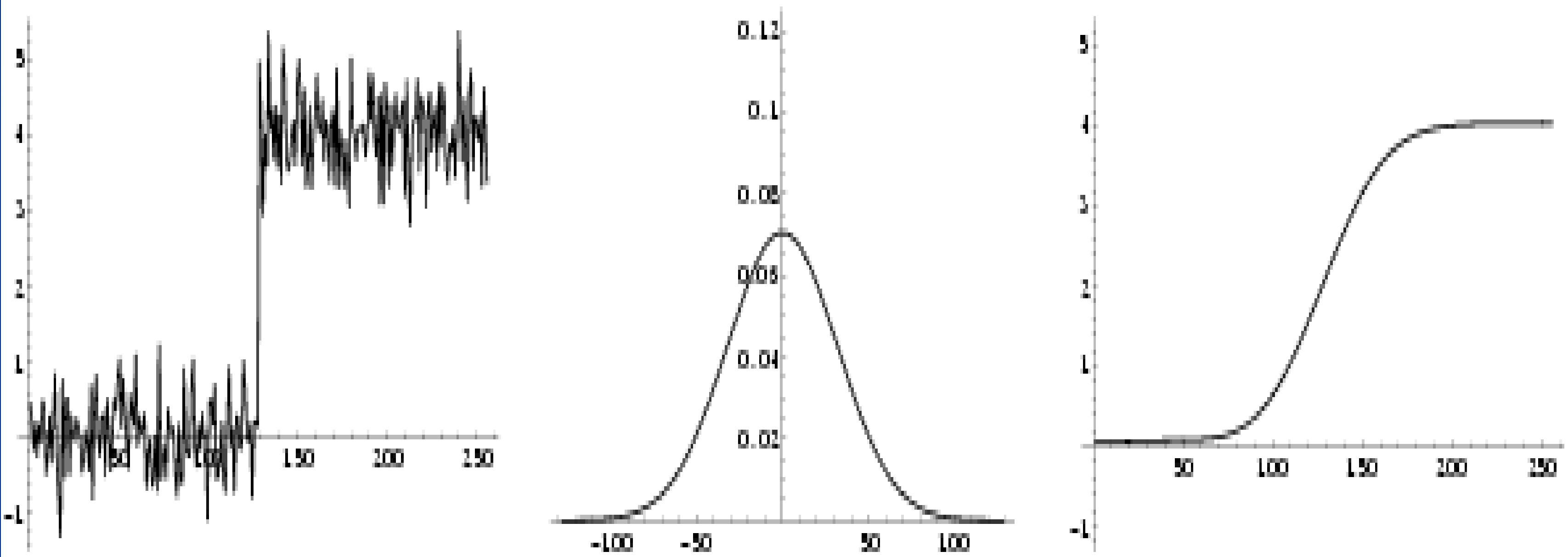
Фильтры низкой и высокой частоты



Гауссиан *(sigma=16)*



Гауссиан ***(sigma=32)***

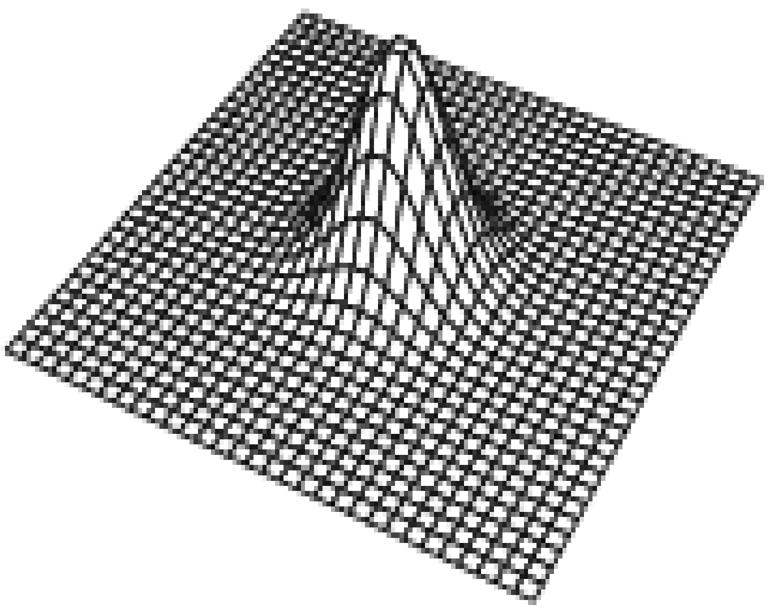


(a) Input $\phi(x)$

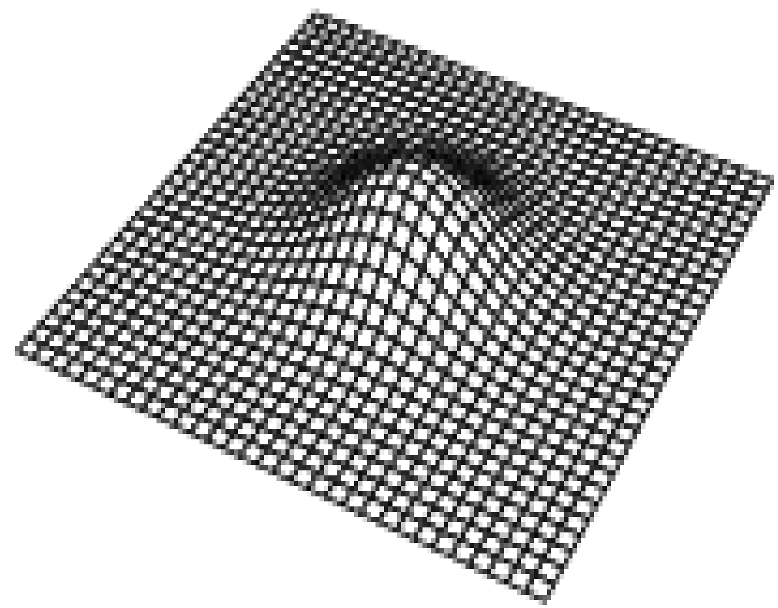
(b) a 1-D Gaussian kernel, $g(x, \sigma)$ where $\sigma = 32$

(c) output of $\phi(x) \otimes g(x, 32)$

2-D Gaussian kernel

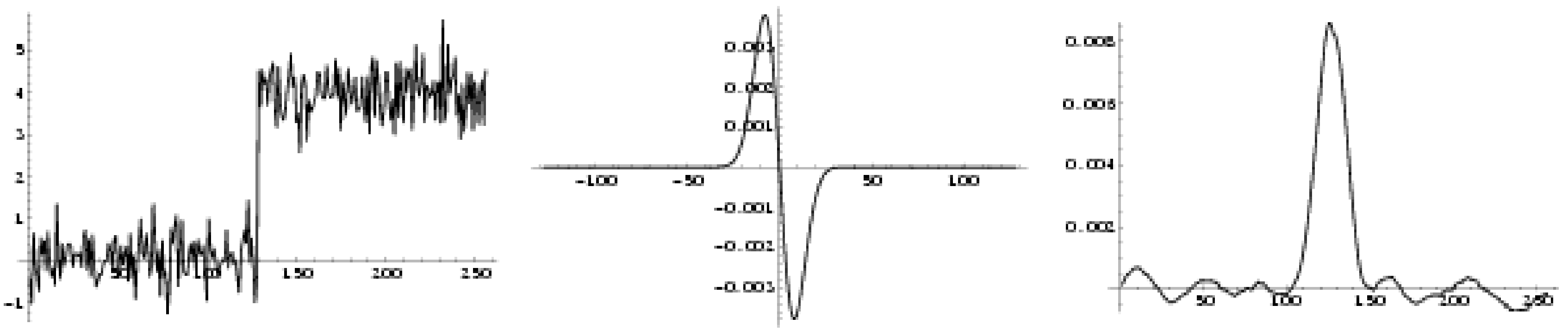


(a) 2-D Gaussian kernel $g(x, y, \sigma_1)$ with $\sigma_1 = 8$



(b) 2-D Gaussian kernel $g(x, y, \sigma_2)$ with $\sigma_2 = 12$

Свертка - дифференцирование



(a) 1-D Input $\phi(x)$ (b) $\frac{\partial}{\partial x} g(x,\sigma)$ where $\sigma = 3$ (c) $\phi(x) \otimes \frac{\partial}{\partial x} g(x,\sigma)$

$$h(x,y,z) \otimes \frac{\partial}{\partial x}\phi(x,y,z) = \frac{\partial}{\partial x}h(x,y,z) \otimes \phi(x,y,z)$$