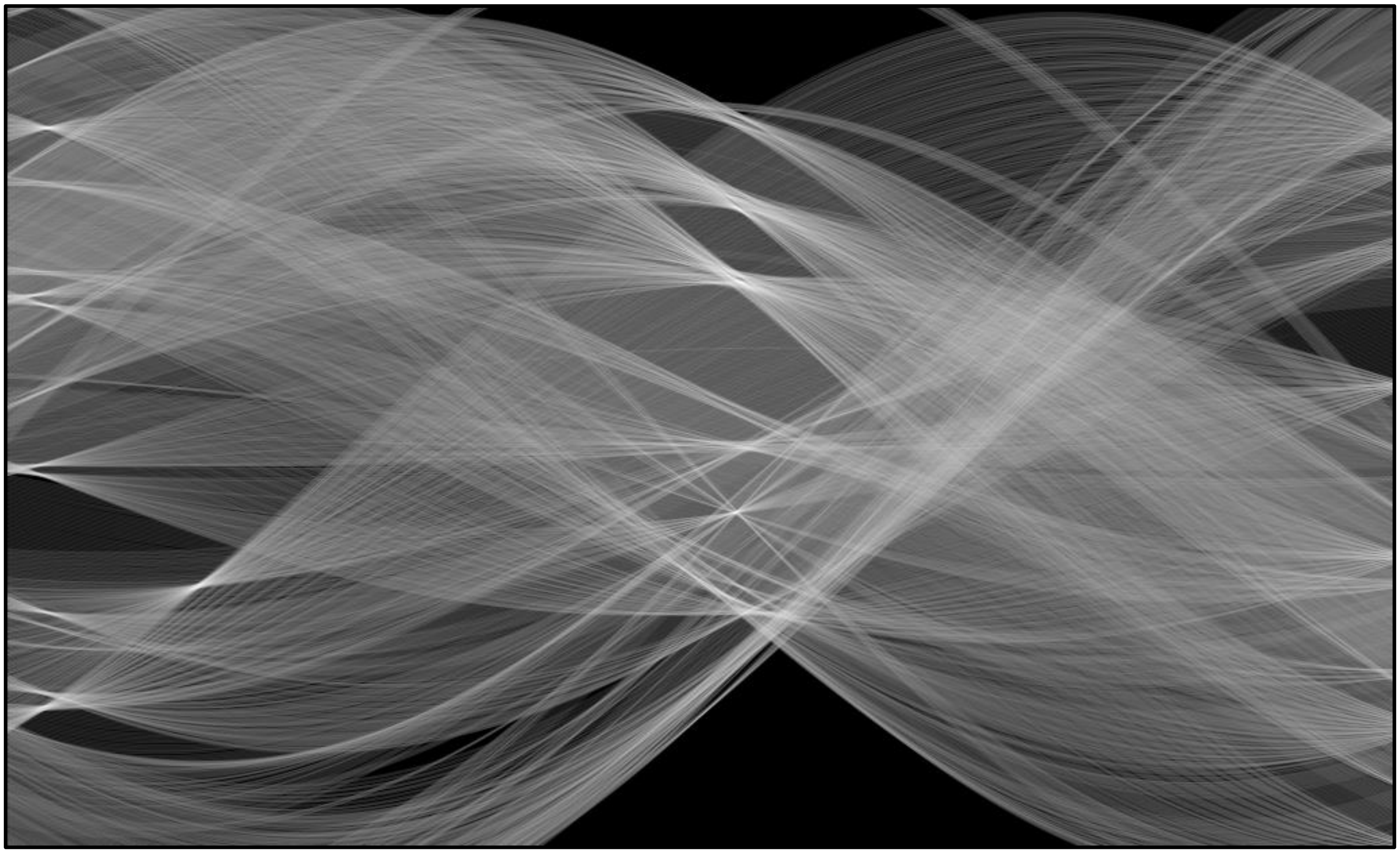
Hough transform



ICS 483 Computer Vision
Fall 2025, Lecture 5

Class announcement

- Quiz 1 already graded will be returned soon
- It's great some of you have discussed with your classmates and noted that.
- Discussion is encouraged. Share your questions on Discord. **Ask questions and answer questions**! No questions are stupid questions.
- But please remember, **copying others'** answers from others is **prohibited**.
- If you used LLMs, please also note in your submission, and briefly describe how you used LLMs and how you learn from or used the LLMs' answer. (e.g. "I cannot derive the Taylor series expansion of function g(x) correctly, and ChatGPT helped me found the terms I missed …. ")
- There are 10pts extra credit opportunities (presentations about CV-related papers, software, news, etc). Feel free to talk to me.

Recap

Fourier transform

	Fourier transform	inverse Fourier transform
continuous	$F(k) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi kx} \, dx$	$f(x) = \int_{-\infty}^{\infty} F(k) e^{j2\pi kx} \, dk$
discrete	$F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi kx/N}$ $k = 0, 1, 2, \dots, N-1$	$f(x) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) e^{j2\pi kx/N}$ $x = 0, 1, 2, \dots, N-1$

Where is the connection to the *"summation of sine waves"* idea?

Fourier transform

Where is the connection to the *"summation of sine waves"* idea?

$$f(x) = \frac{1}{N}\sum_{k=0}^{N-1} F(k) e^{j2\pi kx/N}$$

Euler's formula

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

sum over frequencies

$$f(x) = \frac{1}{N}\sum_{k=0}^{N-1} F(k) \left\{ \cos\left(\frac{2\pi kx}{N}\right) + j\sin\left(\frac{2\pi kx}{N}\right) \right\}$$

scaling parameter

wave components

Recap: Computing the discrete Fourier transform (DFT)

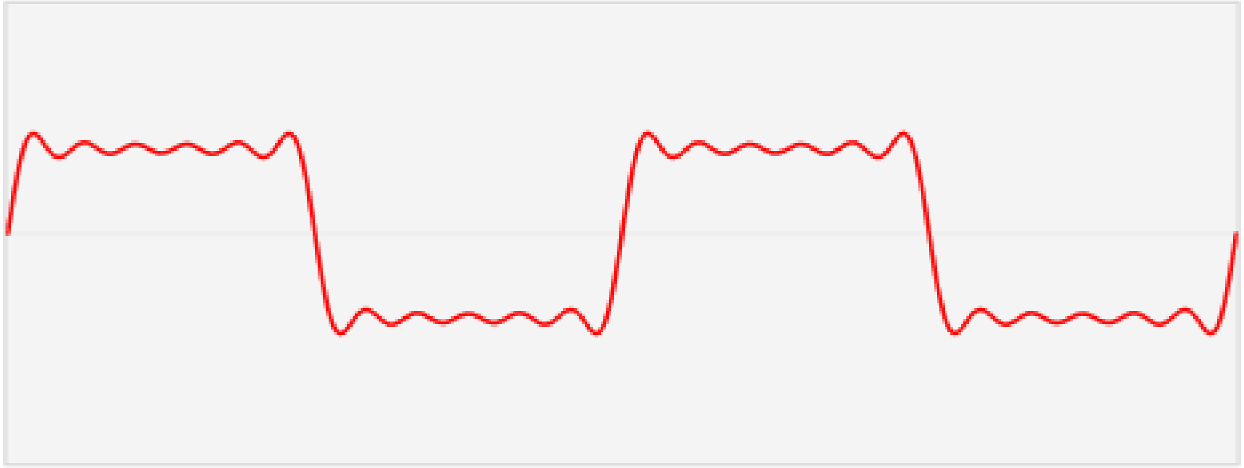
$F(k) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi kx/N}$ is just a matrix multiplication:

$$\boldsymbol{F} = \boldsymbol{W}\boldsymbol{f}$$

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix} \qquad W = e^{-j2\pi/N}$$

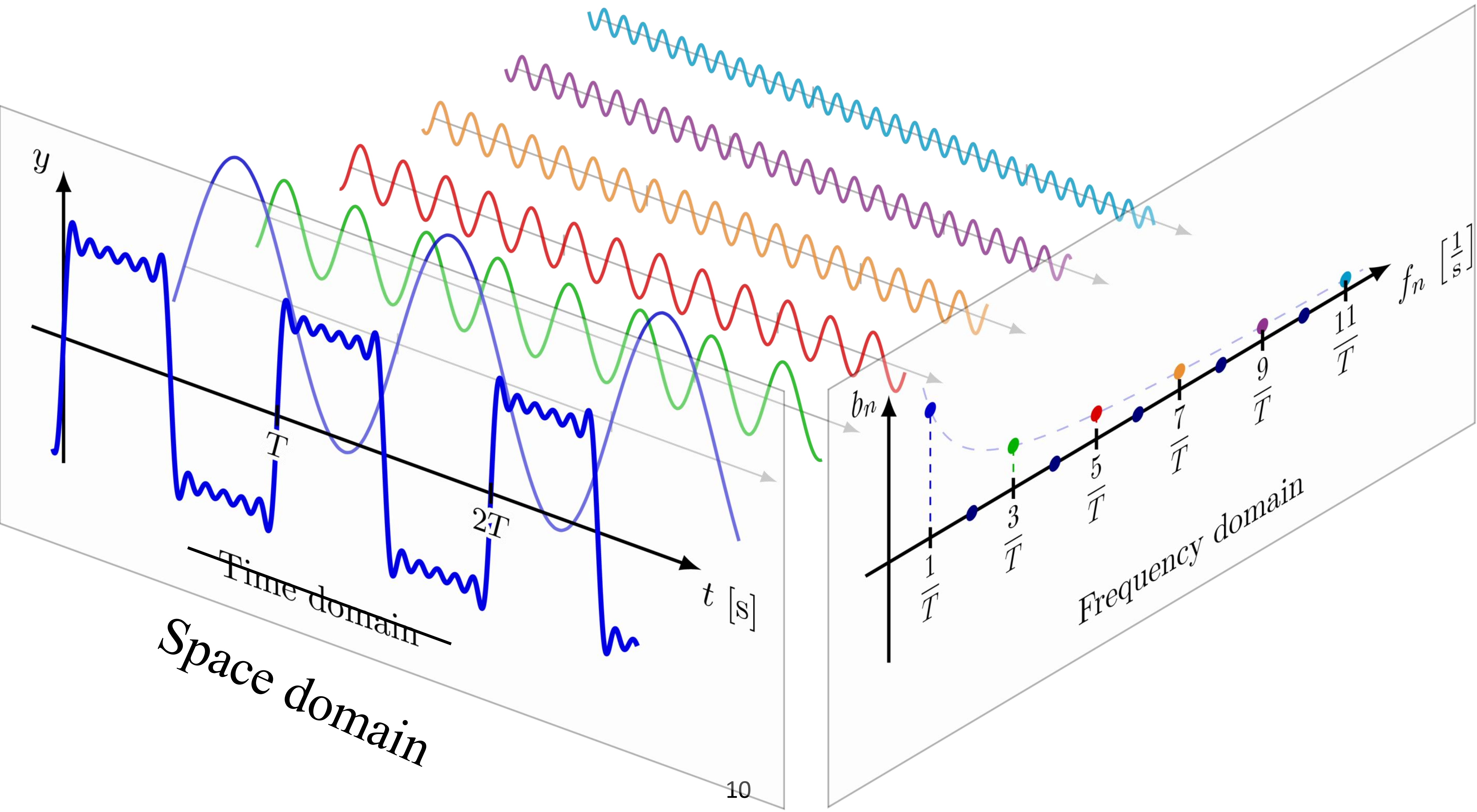
In practice this is implemented using the *fast Fourier transform* (FFT) algorithm.

Recap: FT



Wikipedia: Fourier Transform

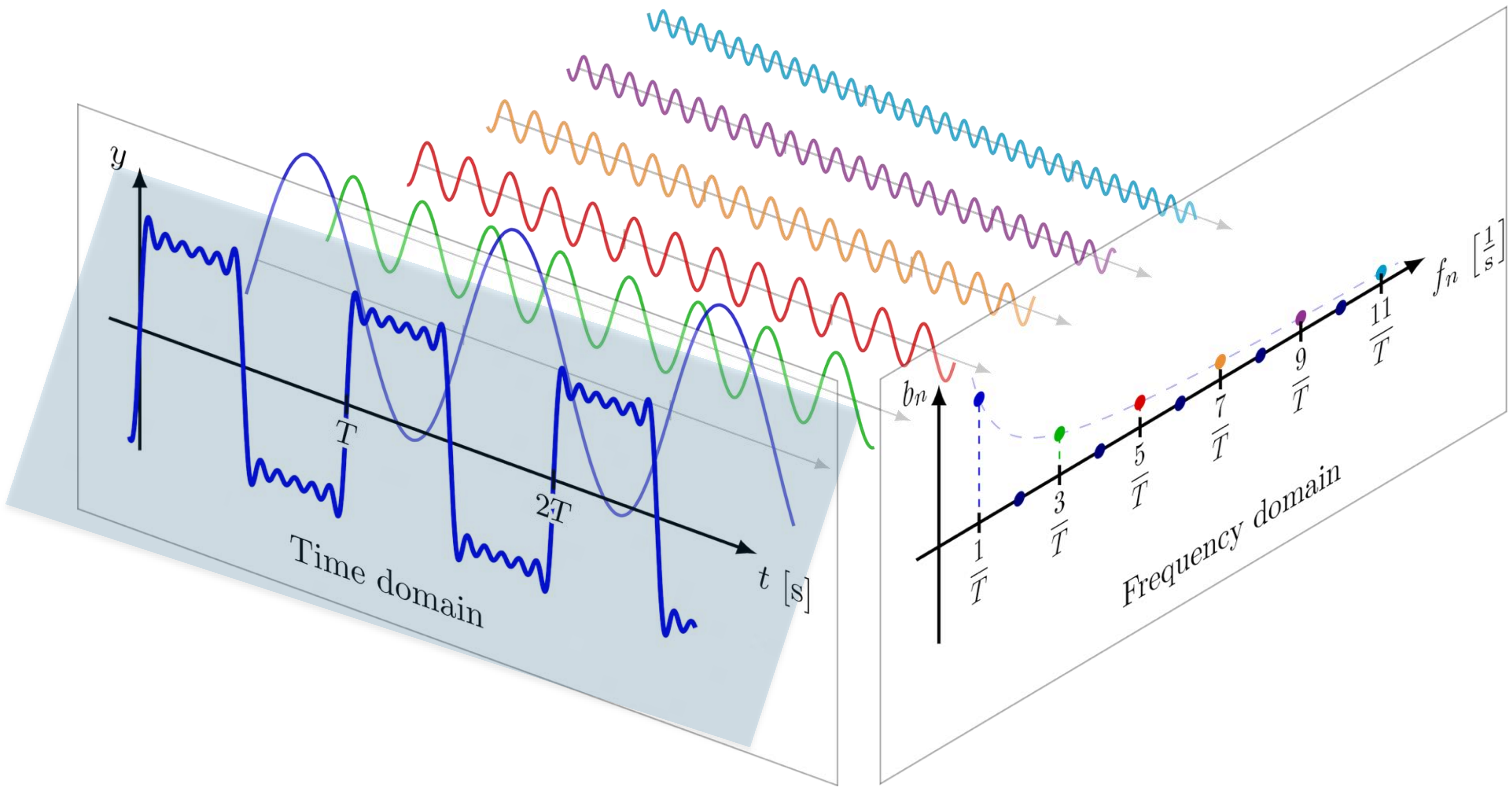
Recap: FT



y
T
$2T$
t [s]
~~Time domain~~
Space domain
b_n
$\frac{1}{T}$
$\frac{3}{T}$
$\frac{5}{T}$
$\frac{7}{T}$
$\frac{9}{T}$
$\frac{11}{T}$
$f_n\ \left[\frac{1}{\mathrm{s}}\right]$
Frequency domain

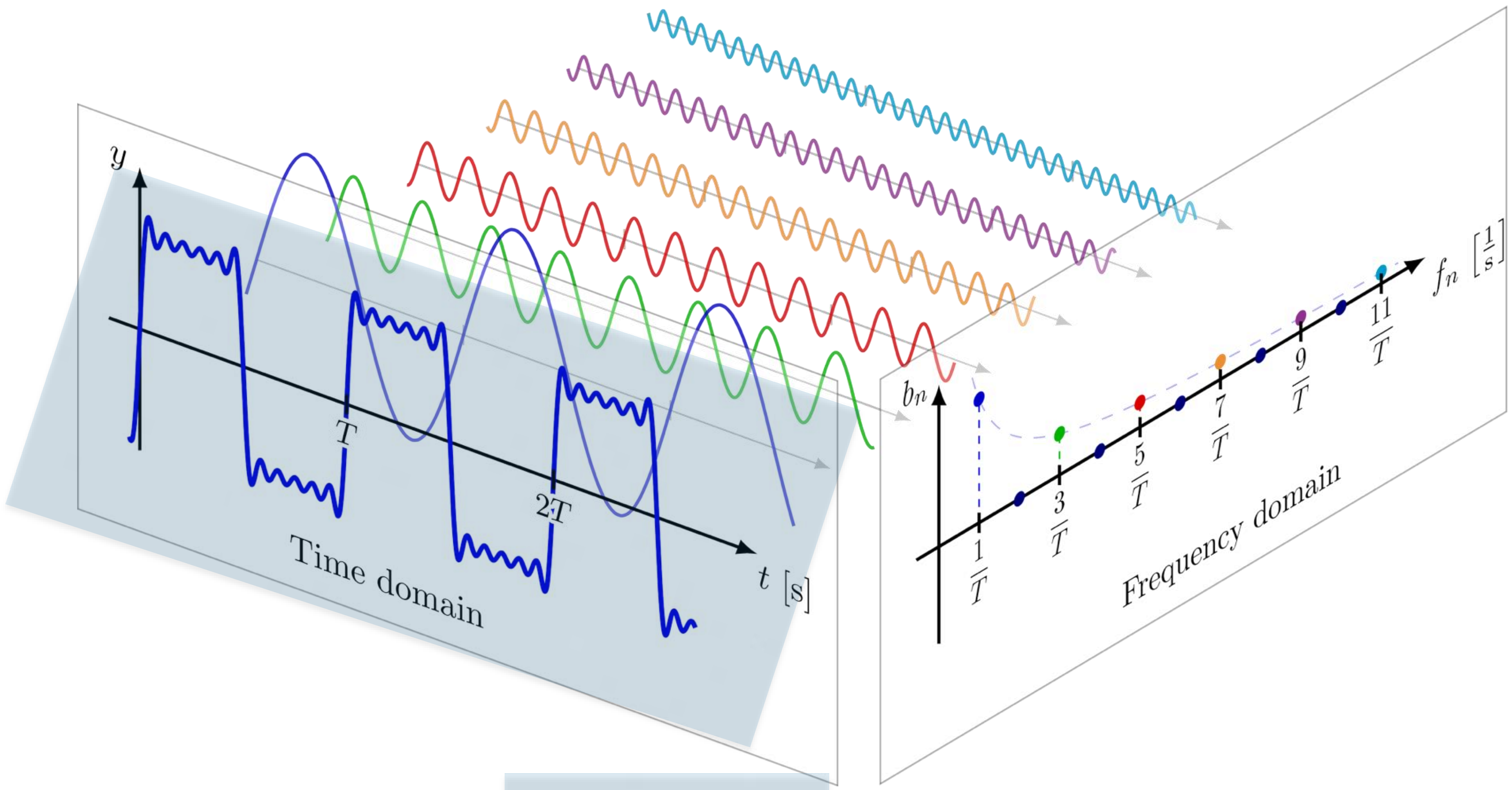
Recap: FT

What's this in the matrix?



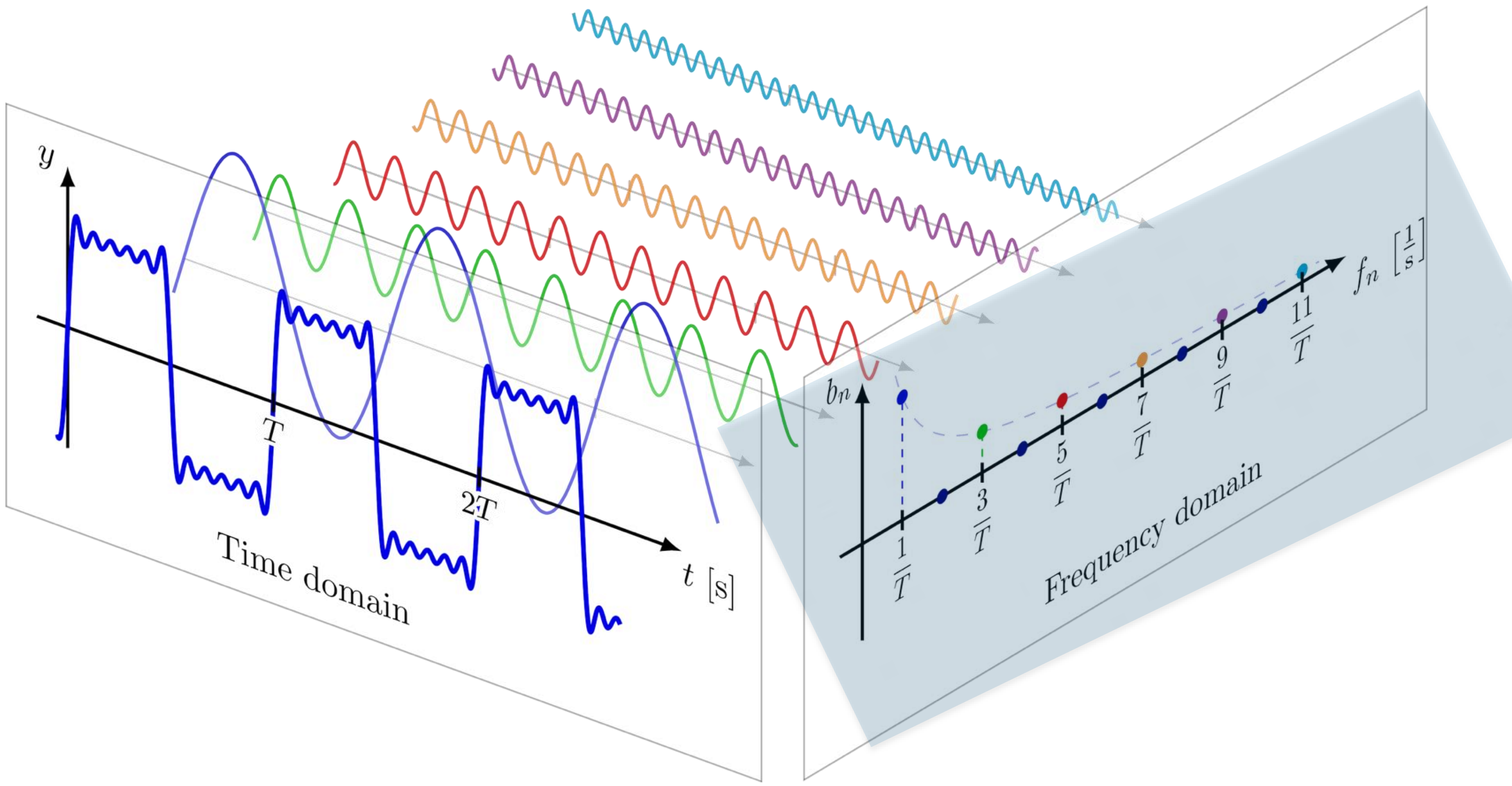
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

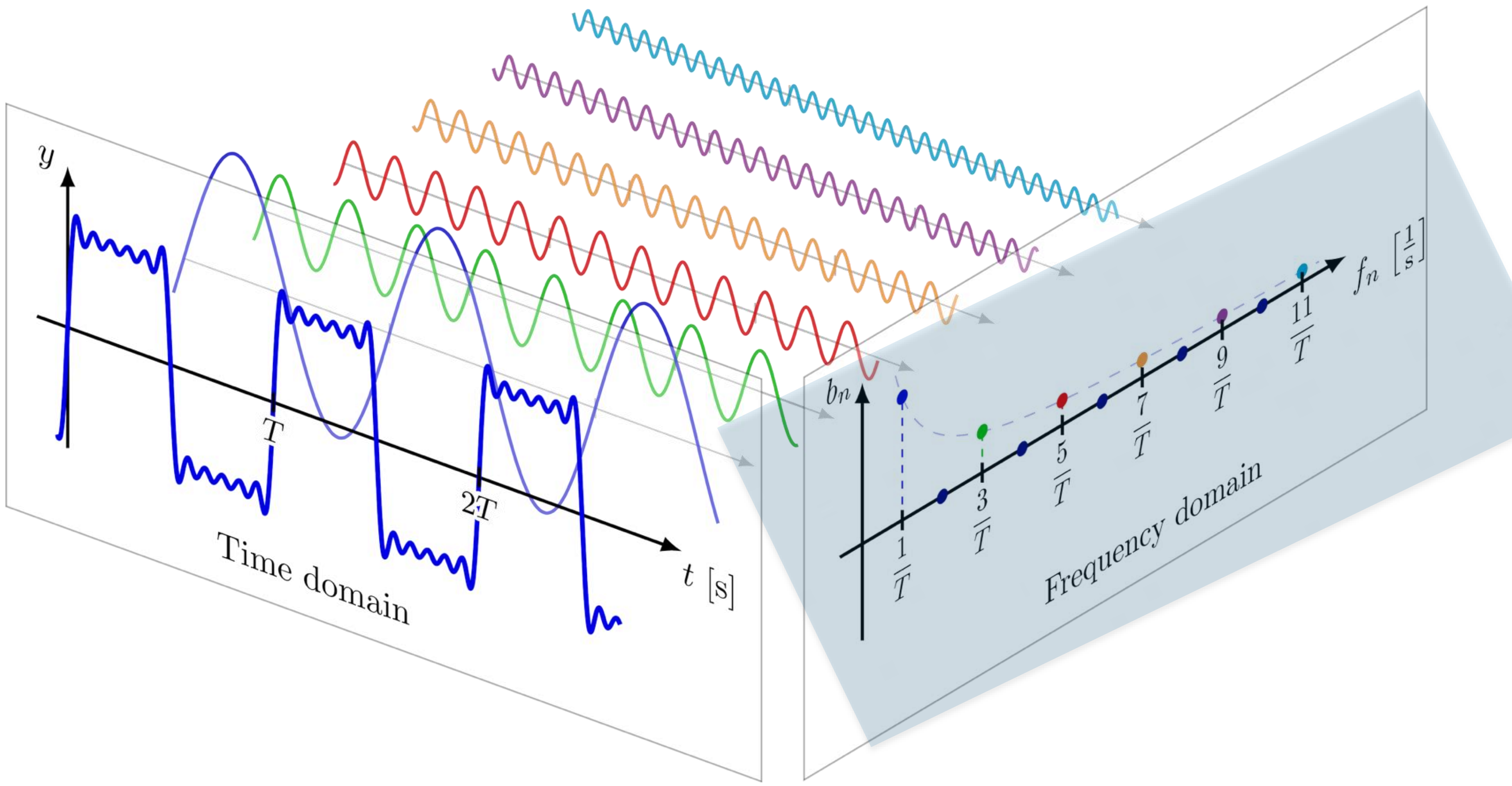
Recap: FT



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

What’s this in the matrix?

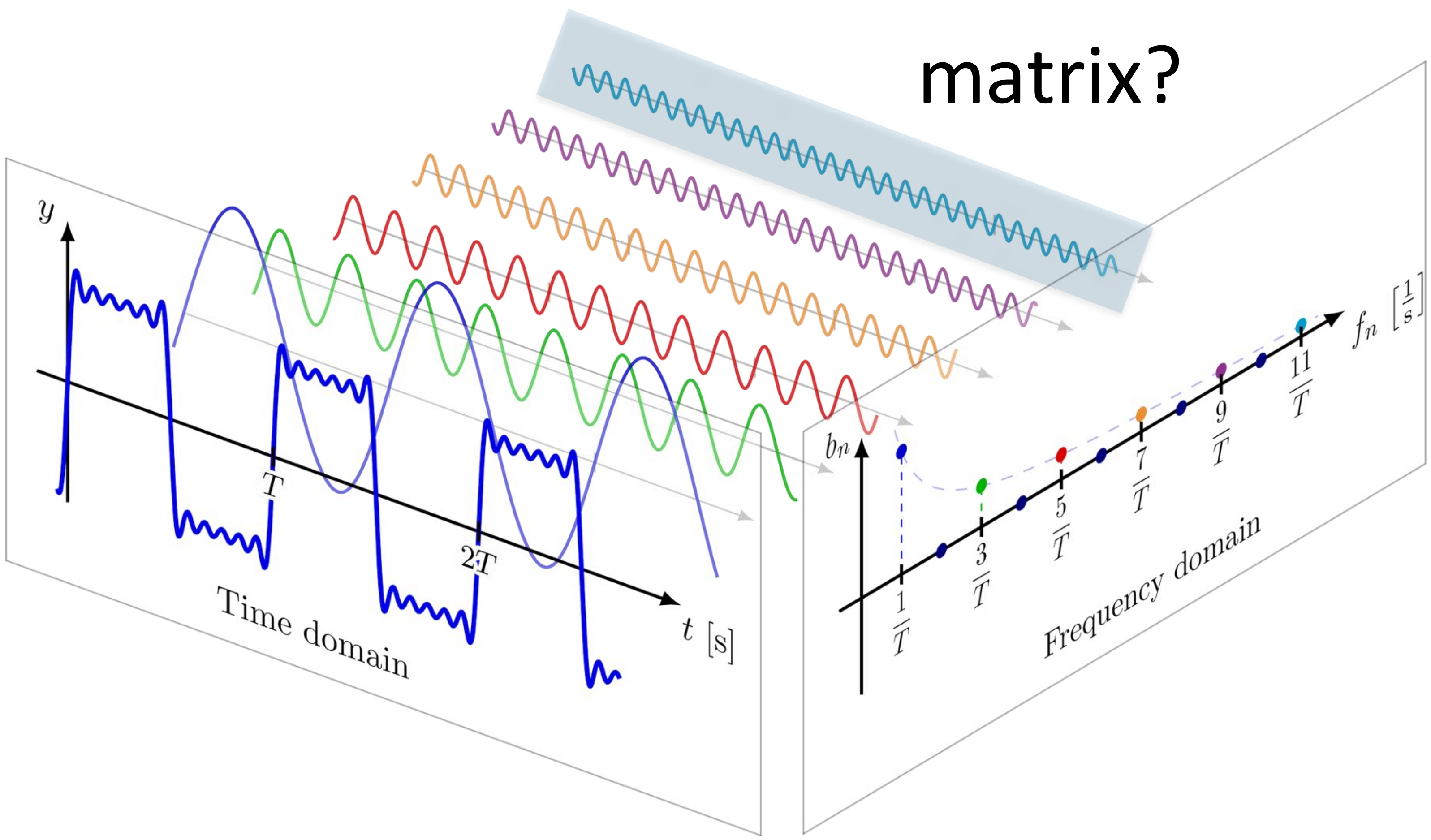
Recap: FT



$$
\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix}
=
\begin{bmatrix}
W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\
W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\
W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\
W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\
\vdots & & & & \ddots & \vdots \\
W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1
\end{bmatrix}
\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}
$$

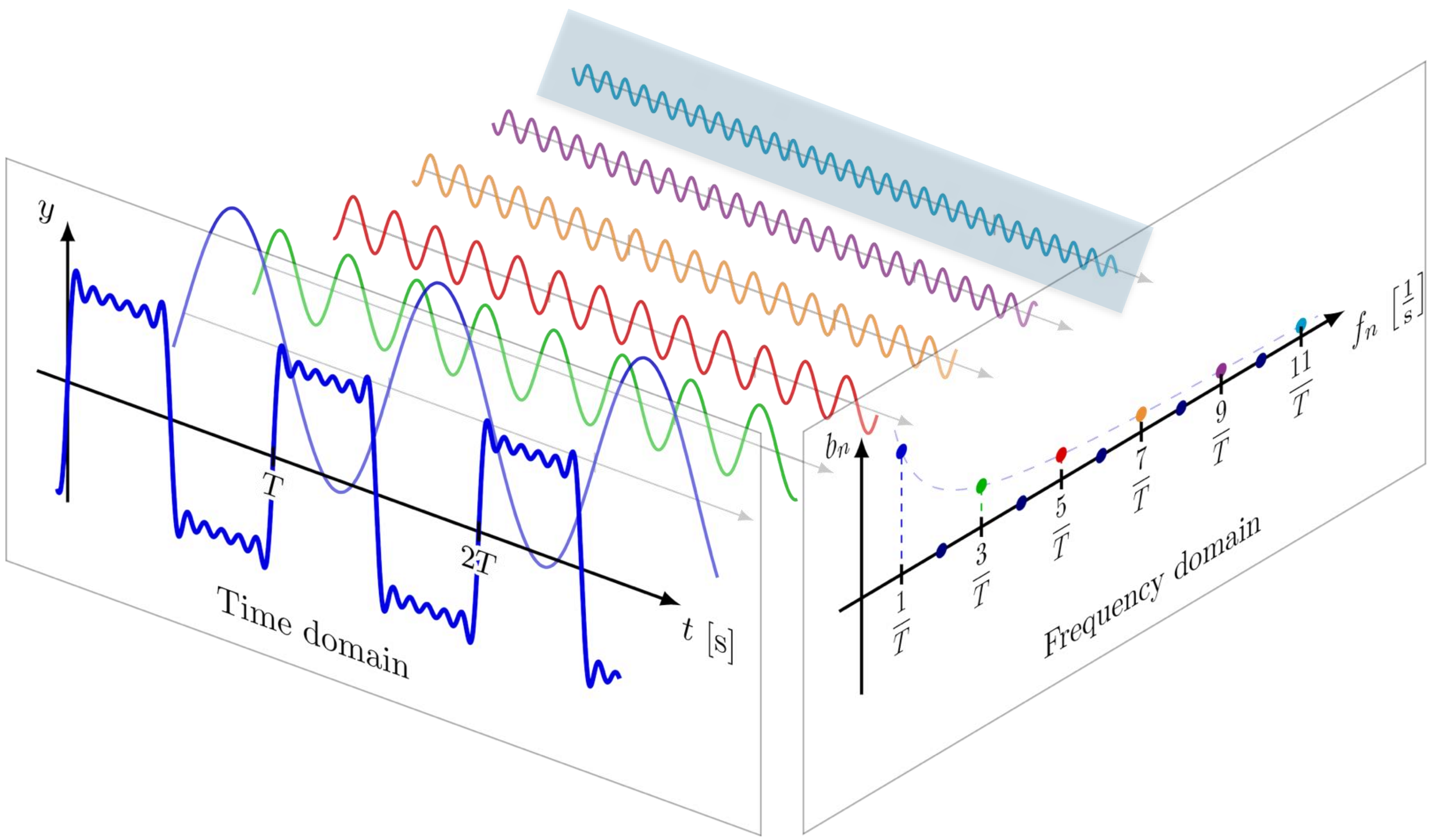
Recap: FT

What's this in the matrix?



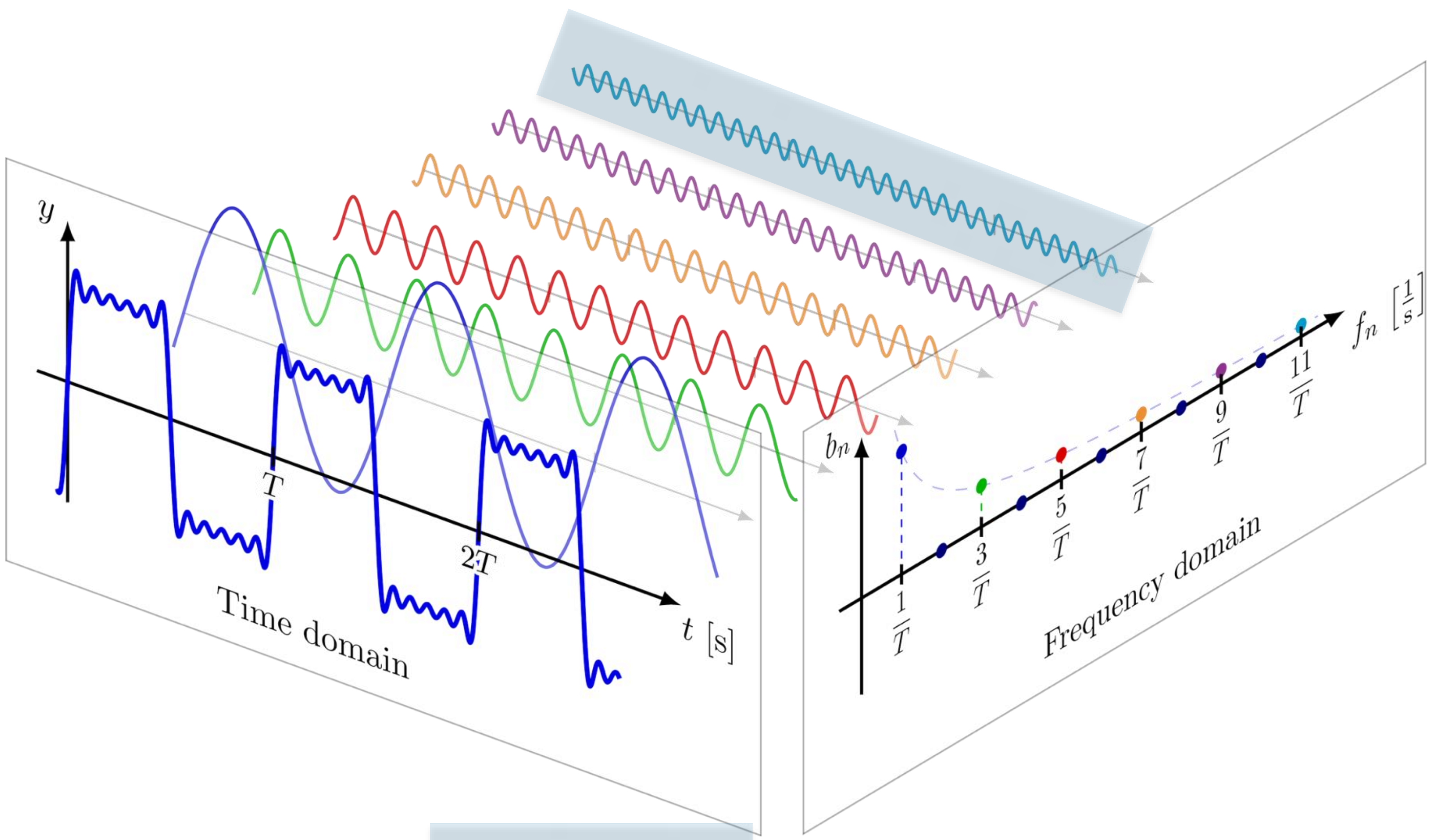
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT



$$
\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}
$$

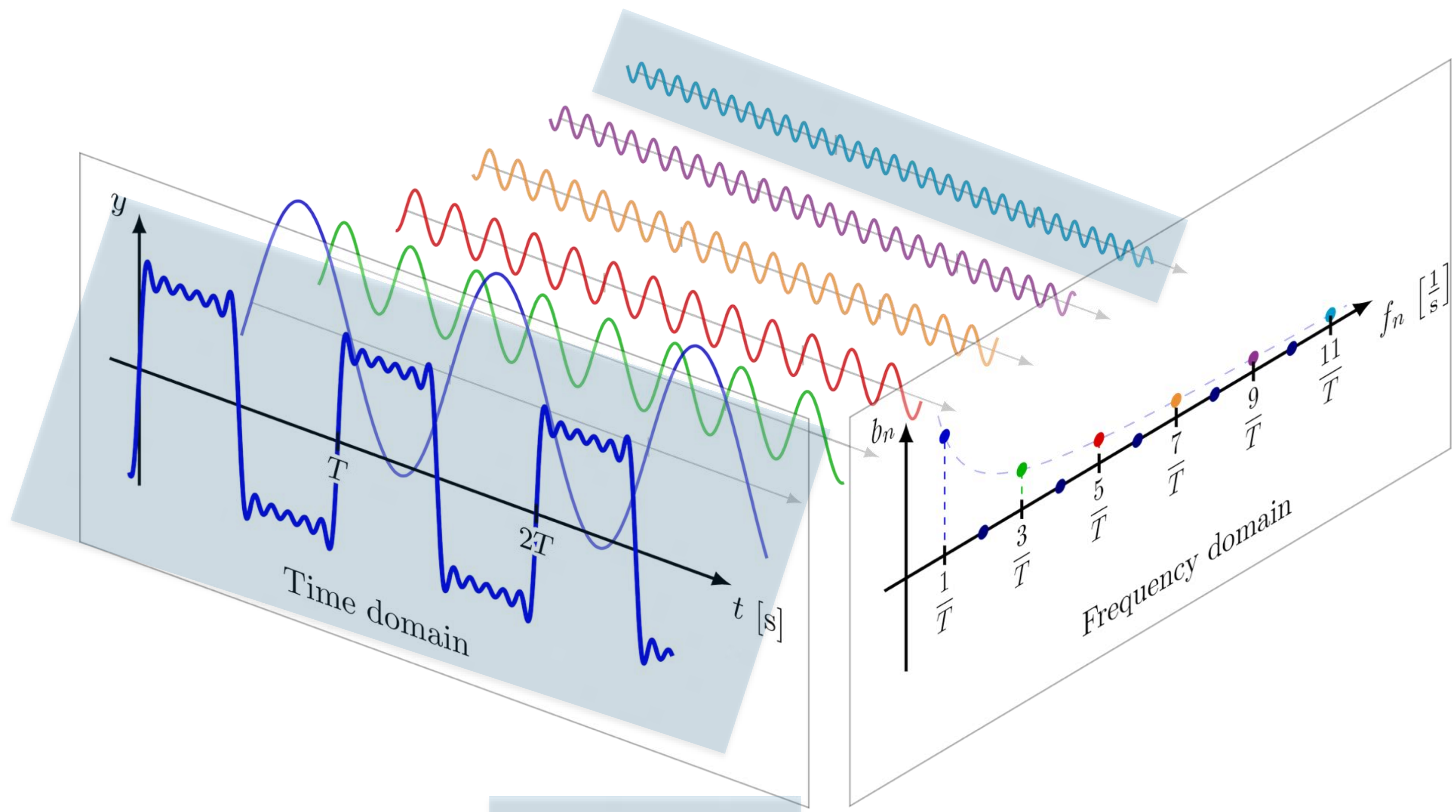
Recap: FT



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

What’s this in the diagram?

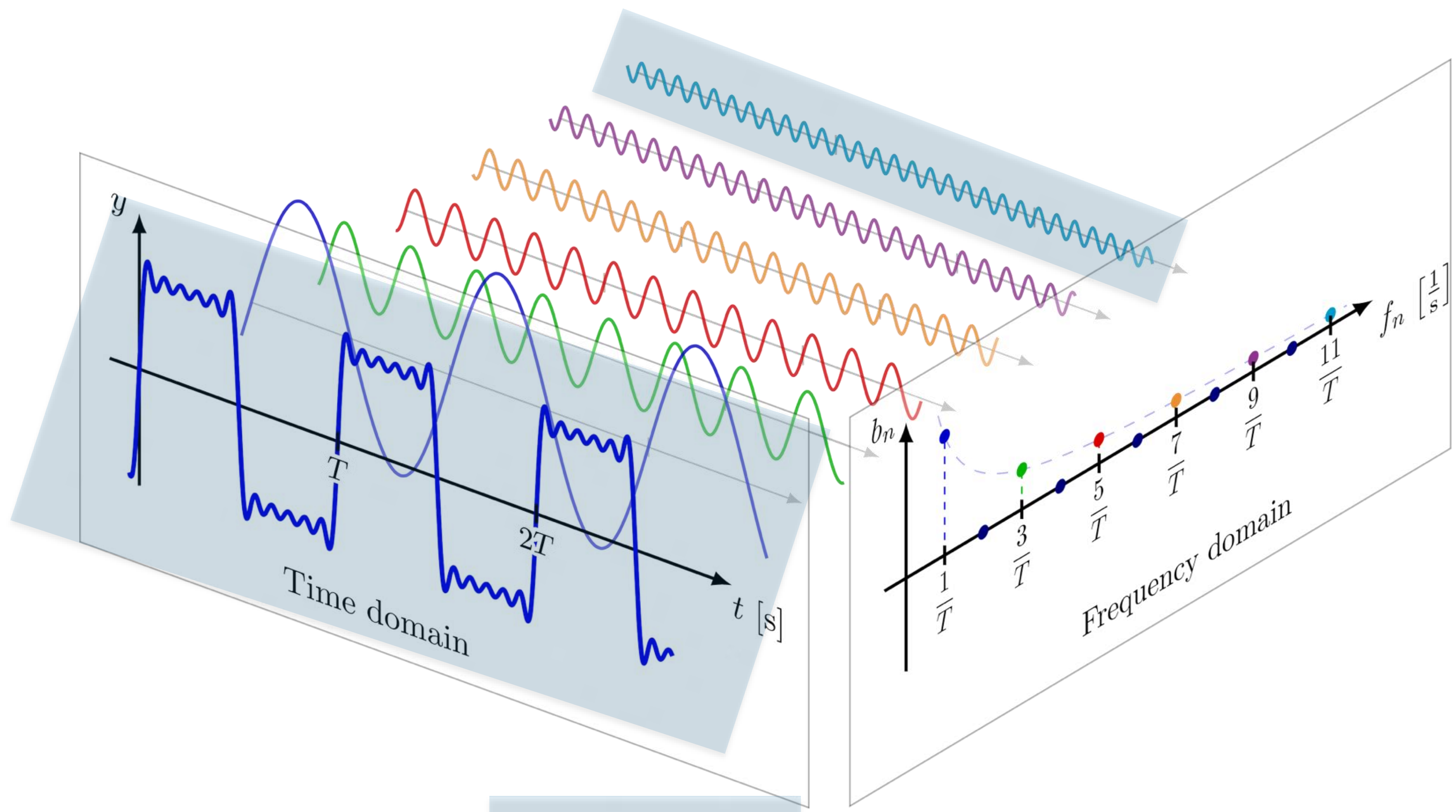
Recap: FT



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

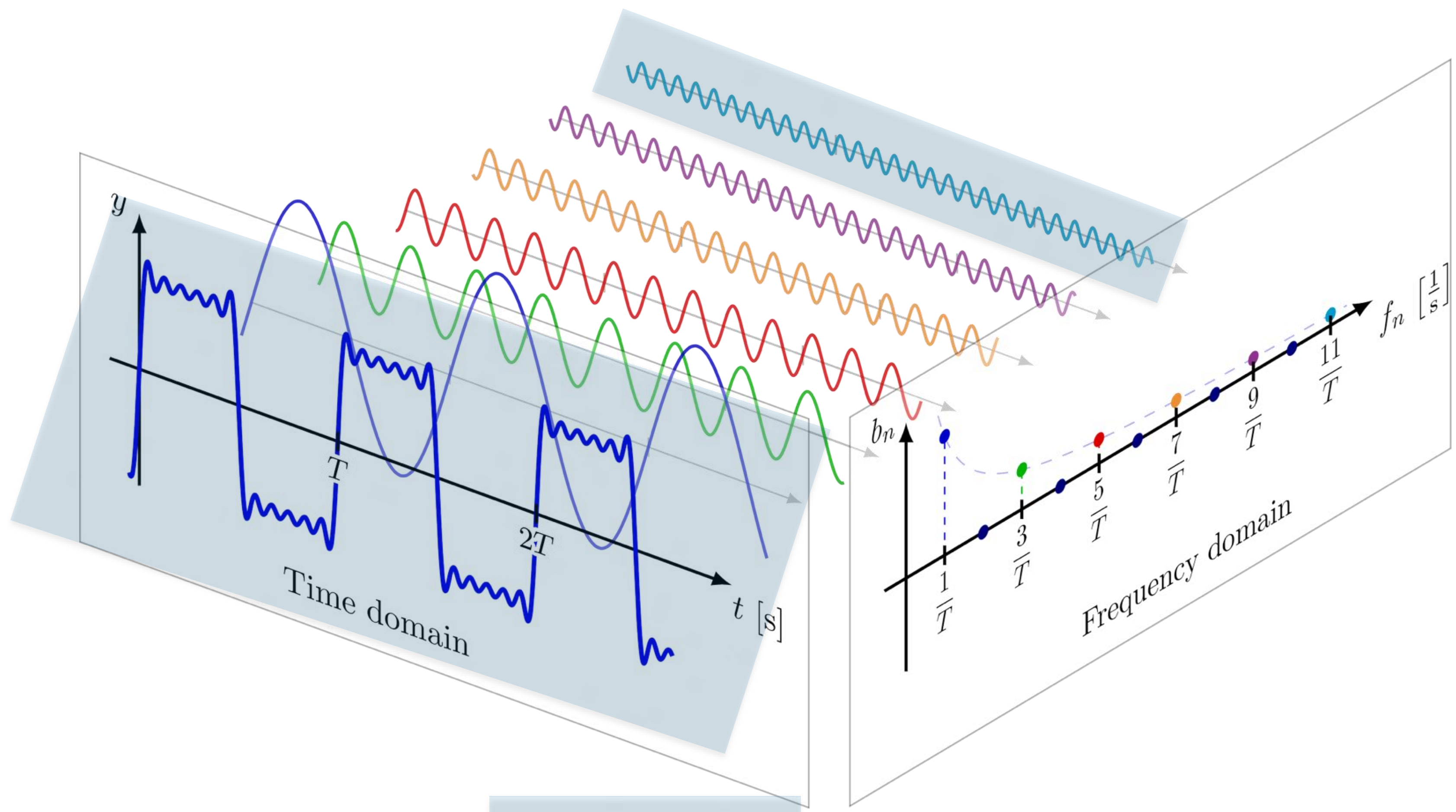
Multiplying it with this row, what do you get?

Recap: FT



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

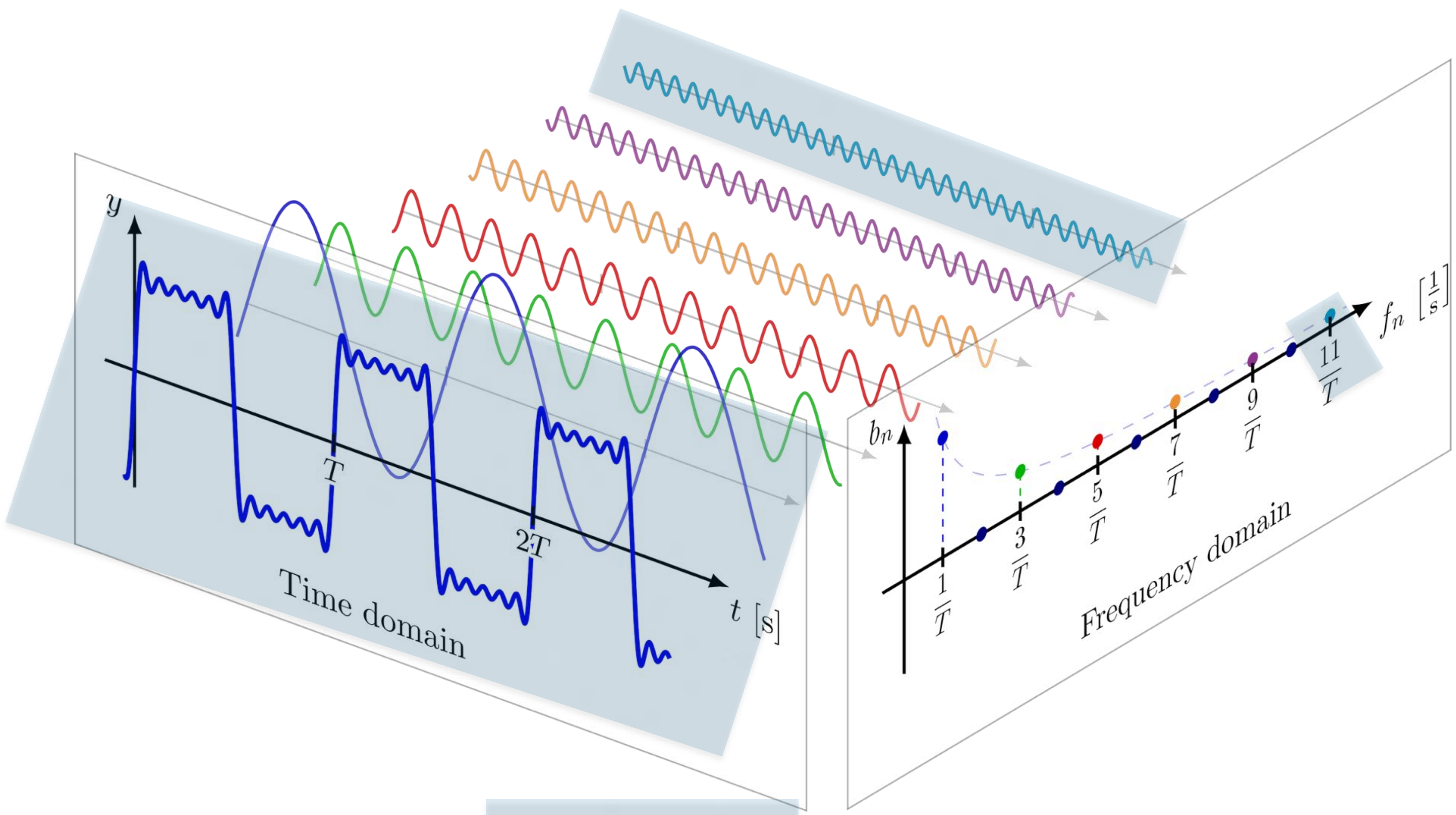
Recap: FT



What's this item in the diagram?

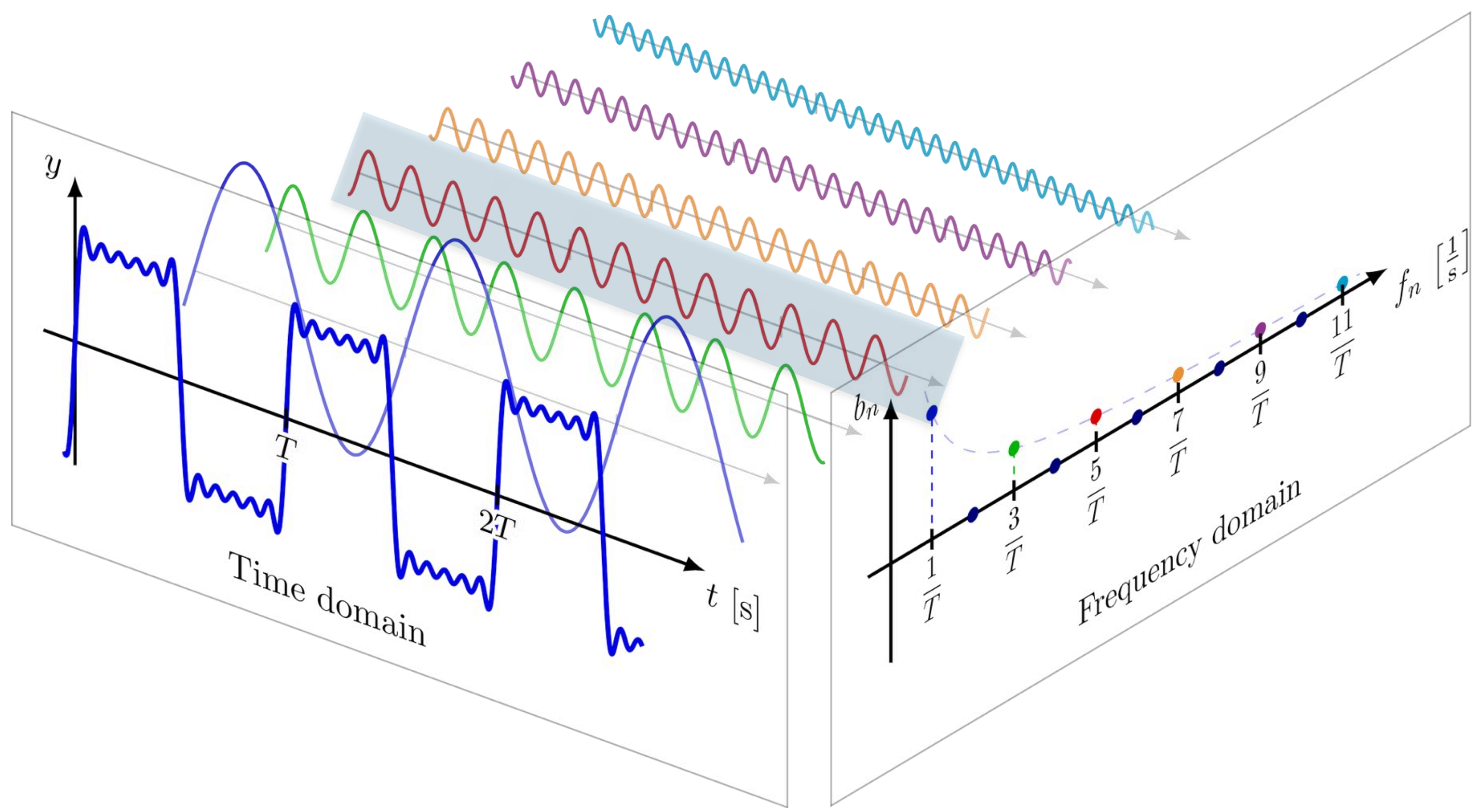
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT



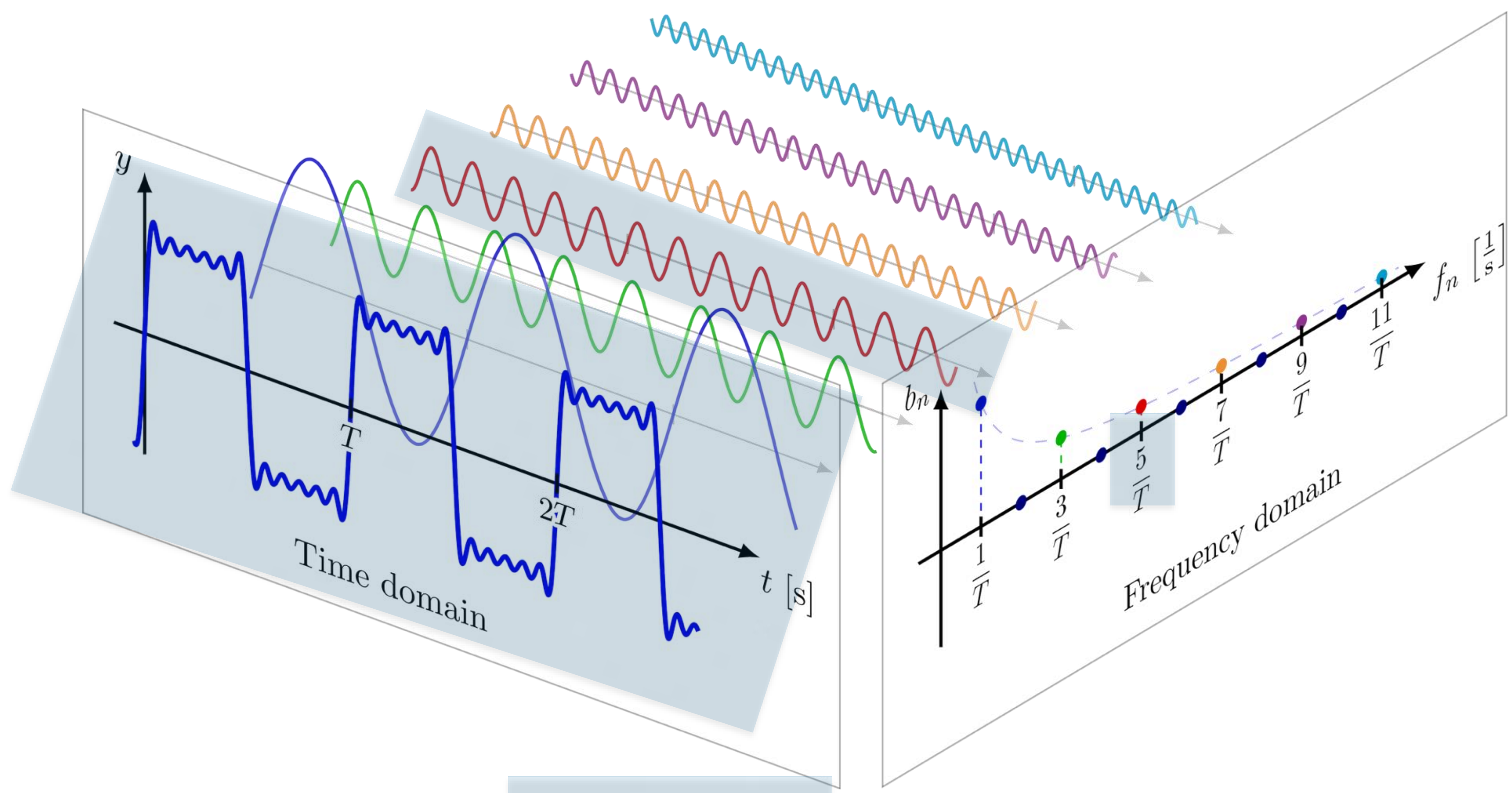
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT



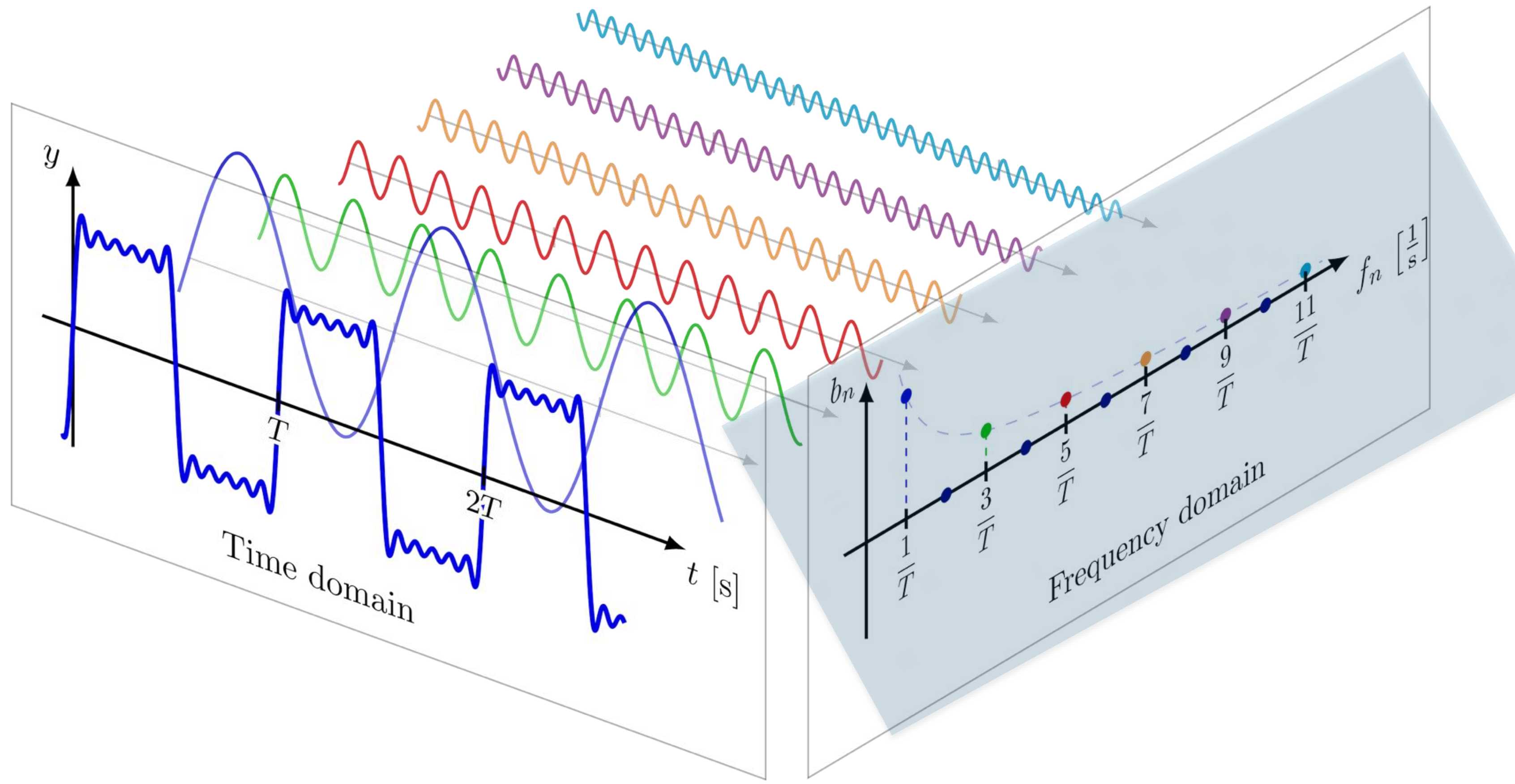
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT



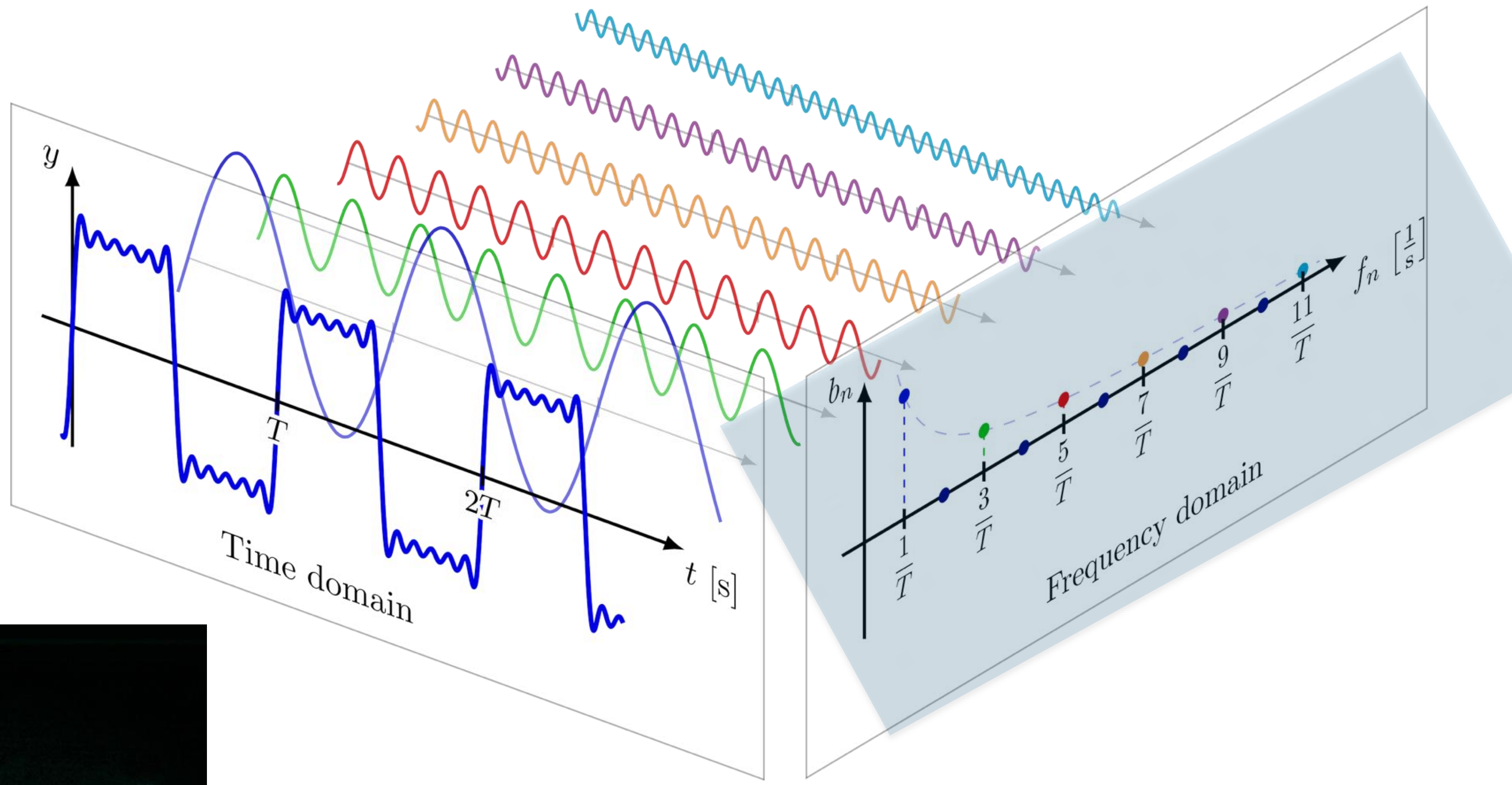
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \ddots & \vdots \\ W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{bmatrix}$$

Recap: FT

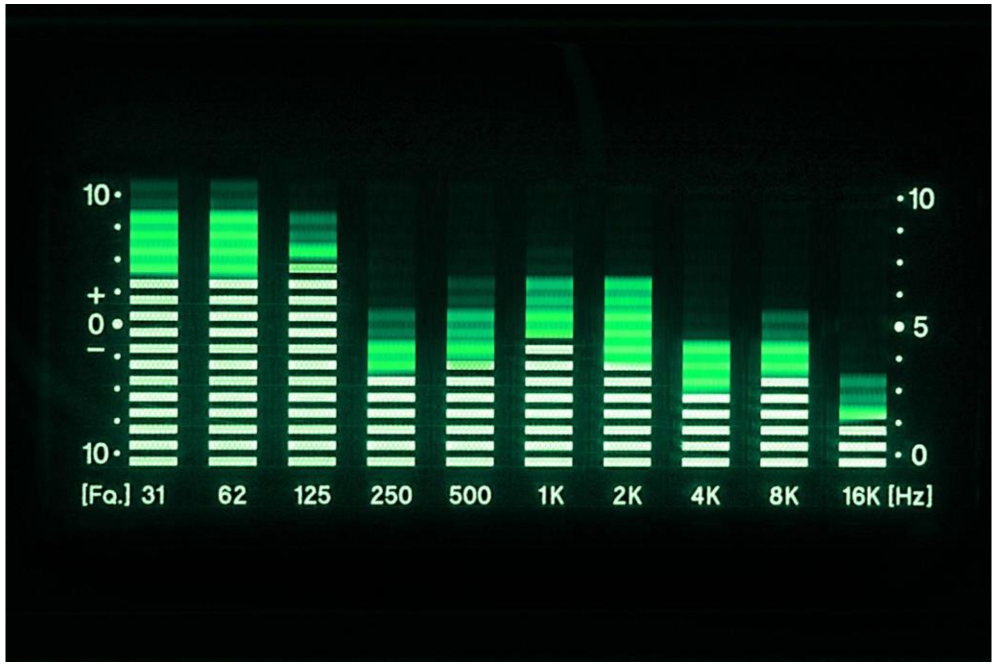


y
T
$2T$
t [s]
Time domain
b_n
f_n $\left[\frac{1}{s}\right]$
$\frac{1}{T}$
$\frac{3}{T}$
$\frac{5}{T}$
$\frac{7}{T}$
$\frac{9}{T}$
$\frac{11}{T}$
Frequency domain

Recap: FT

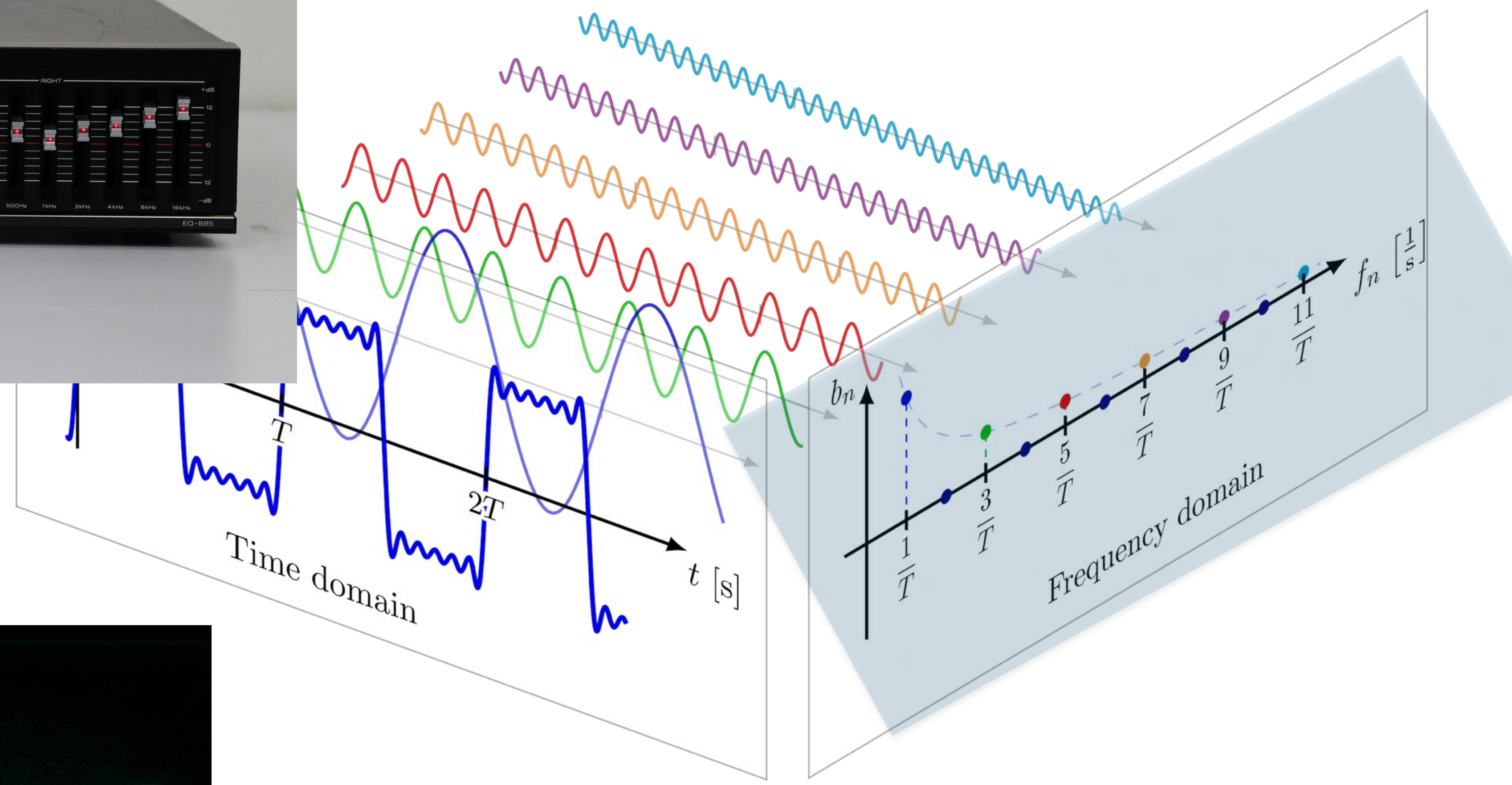


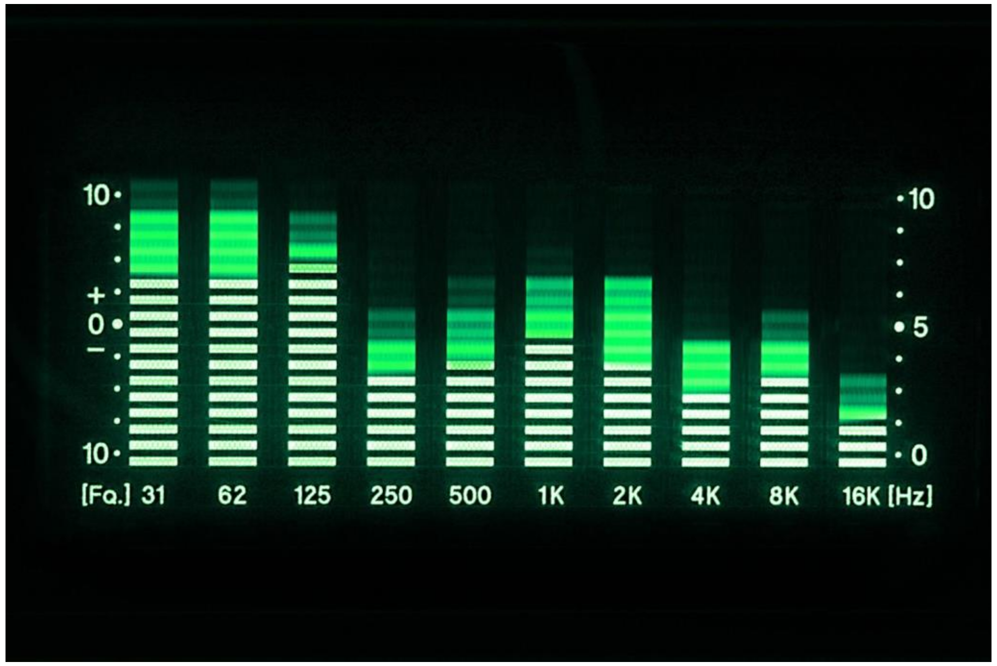
y
T
$2T$
t [s]
Time domain
b_n
$\frac{1}{T}$
$\frac{3}{T}$
$\frac{5}{T}$
$\frac{7}{T}$
$\frac{9}{T}$
$\frac{11}{T}$
f_n $[\frac{1}{s}]$
Frequency domain



10
+
0
−
10
10
5
0
[Fq.] 31 62 125 250 500 1K 2K 4K 8K 16K [Hz]

Recap: FT





What happens when you tune the nob?

Bass
Low-freq

Treble
High-freq

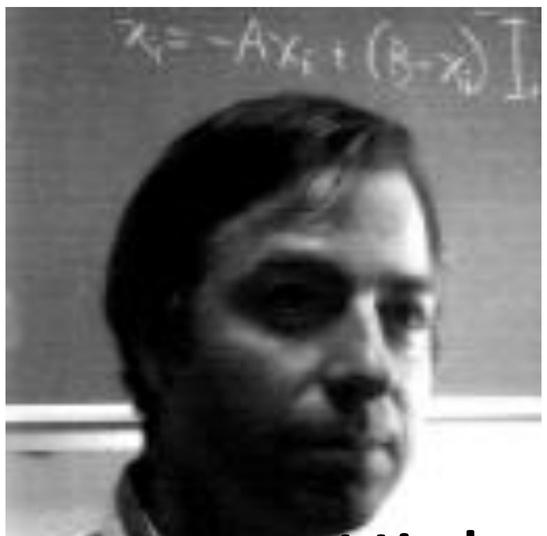
Bass
Low-freq

Treble
High-freq

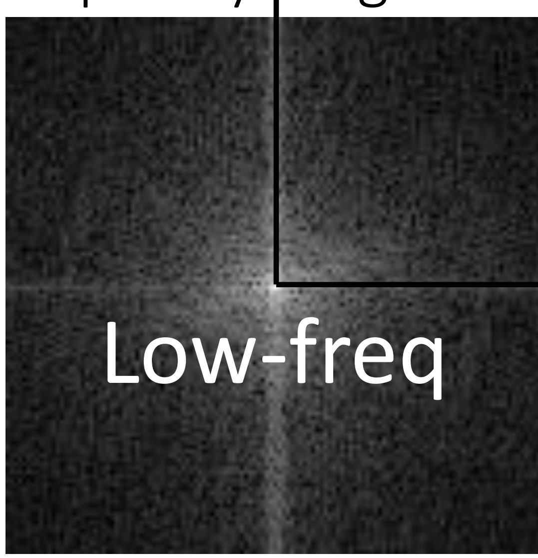
What are “bass” and ”treble” in images?

More filtering examples

original image



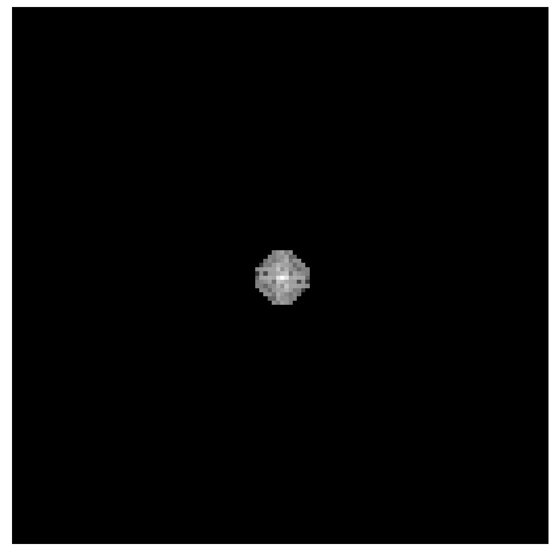
frequency magnitude



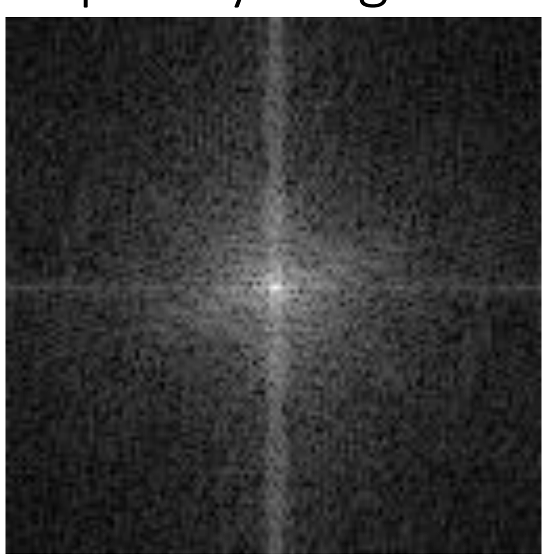
More filtering examples

original image

low-pass filter



frequency magnitude



Code for 1D DFT

- https://colab.research.google.com/drive/174q_UO9lslWQ_YRw7A4oJhf5HQrNckCD#scrollTo=m8EU348tXVUC

- **2pts bonus points: extend the code to 2D**

More resourses

- https://gru.stanford.edu/doku.php/tutorials/fouriertransform
- https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT4/node2.html

Frequency-domain filtering

The convolution theorem

The Fourier transform of the convolution of two functions is the product of their Fourier transforms:

$$\mathcal{F}\{g * h\} = \mathcal{F}\{g\}\mathcal{F}\{h\}$$

The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms:

$$\mathcal{F}^{-1}\{gh\} = \mathcal{F}^{-1}\{g\} * \mathcal{F}^{-1}\{h\}$$

Convolution in spatial domain is equivalent to multiplication in frequency domain!

What do we use convolution for?

Convolution for 1D continuous signals

Definition of linear shift-invariant filtering as convolution:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)dy$$

filtered signal

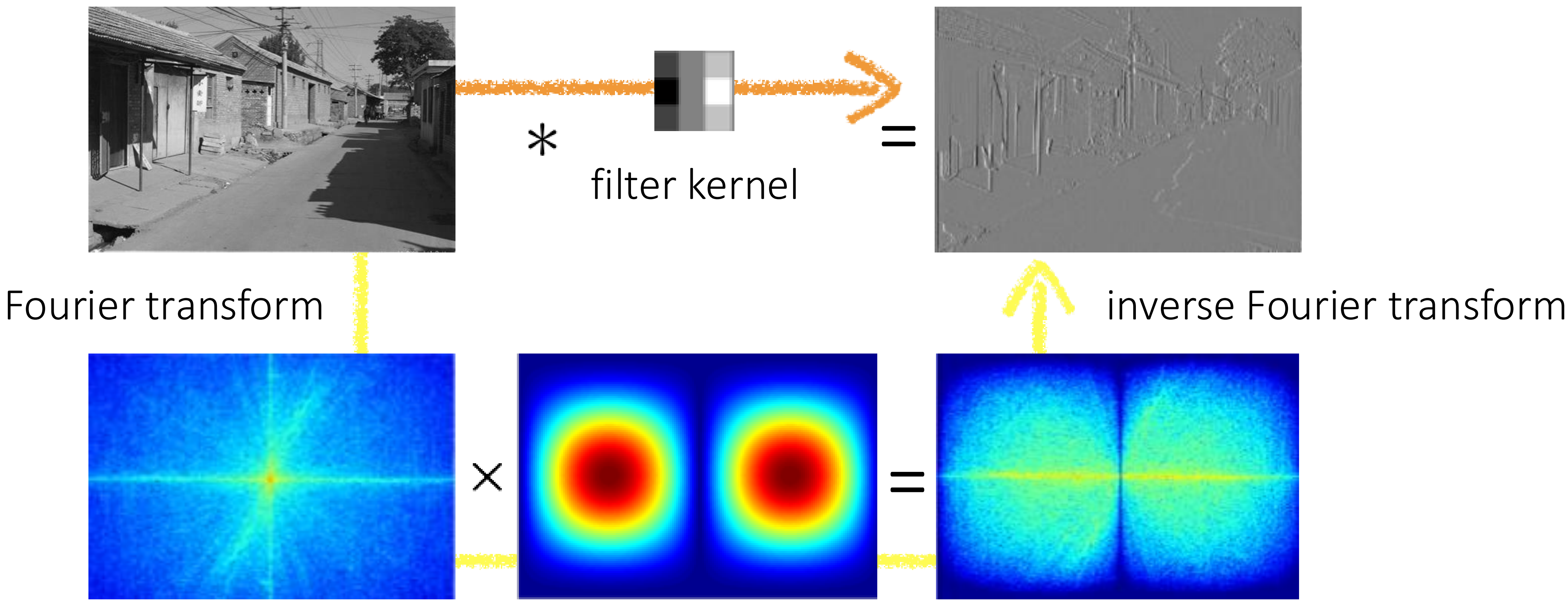
filter

input signal

Using the convolution theorem, we can interpret and implement all types of linear shift-invariant filtering as multiplication in frequency domain.

Why implement convolution in frequency domain?

Spatial domain filtering



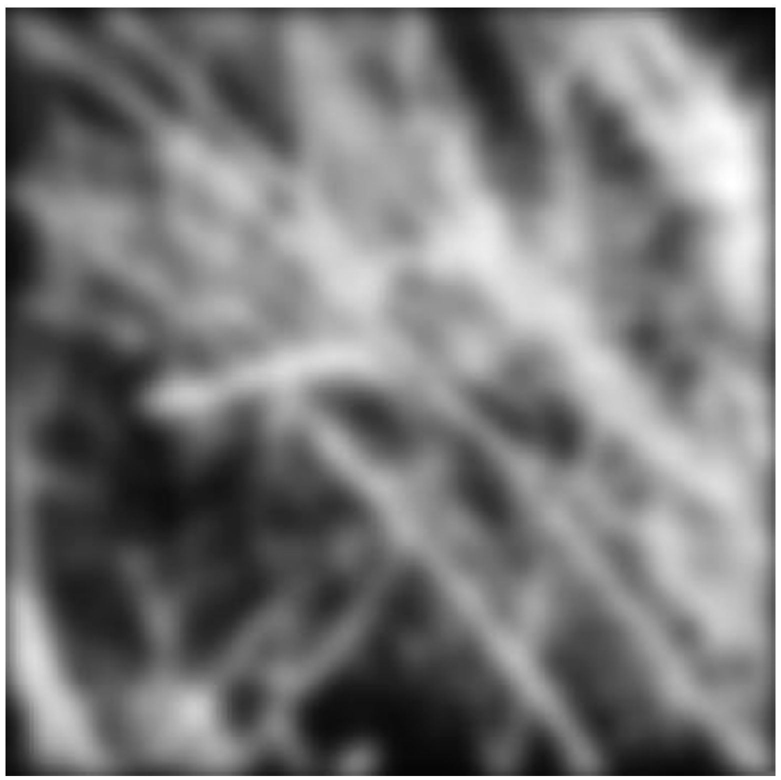
Frequency domain filtering

Frequency-domain filtering in Python

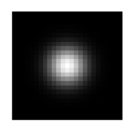
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft2, ifft2, fftshift

# Create a sample image
image = np.zeros((100, 100))
image[40:60, 40:60] = 1

# Compute the Fourier Transform
fft_image = fft2(image)
fft_shifted = fftshift(fft_image)

# Create a low-pass filter
rows, cols = image.shape
crow, ccol = rows // 2, cols // 2
mask = np.zeros((rows, cols), np.uint8)
mask[crow - 10: crow + 10, ccol - 10: ccol + 10] = 1

# Apply the filter
fft_filtered = fft_shifted * mask

# Inverse Fourier Transform
filtered_image = np.real(ifft2(fftshift(fft_filtered)))

# Display the results
plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(np.log(np.abs(fft_shifted)), cmap='gray')
plt.title('Fourier Spectrum')

plt.subplot(1, 3, 3)
plt.imshow(filtered_image, cmap='gray')
plt.title('Filtered Image')

plt.tight_layout()
plt.show()
```

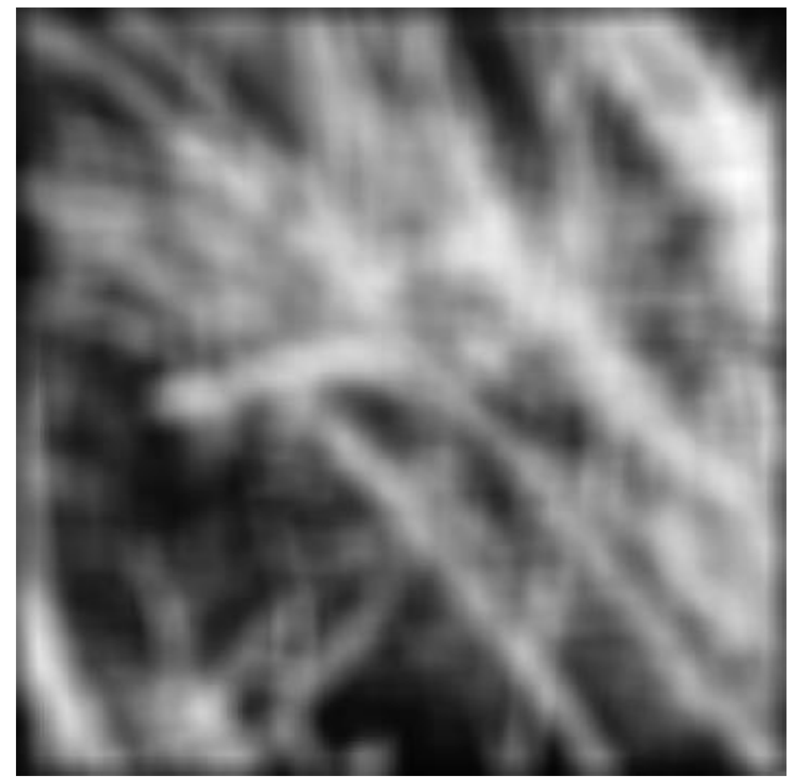
Revisiting blurring

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

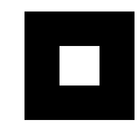


Gaussian filter

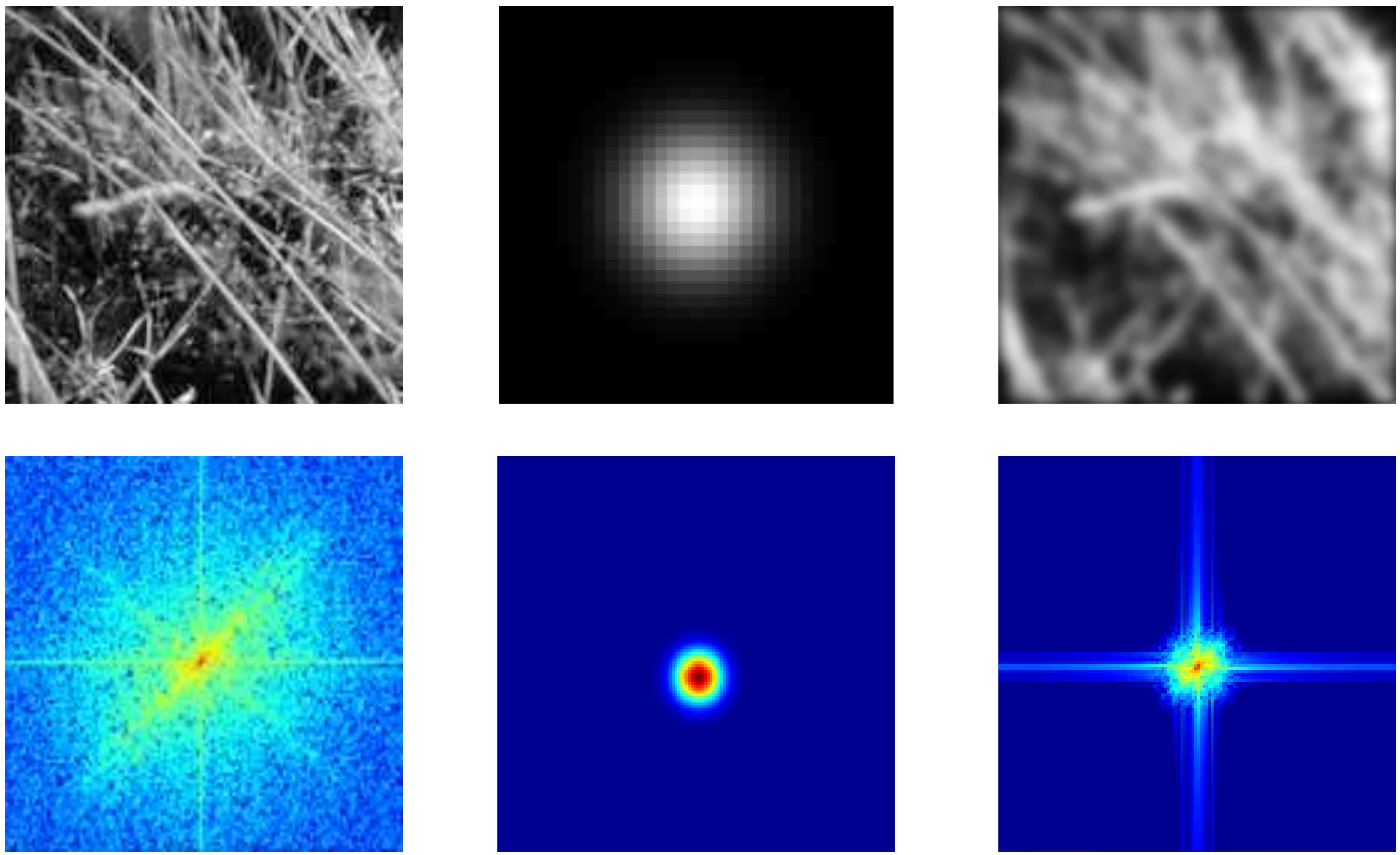




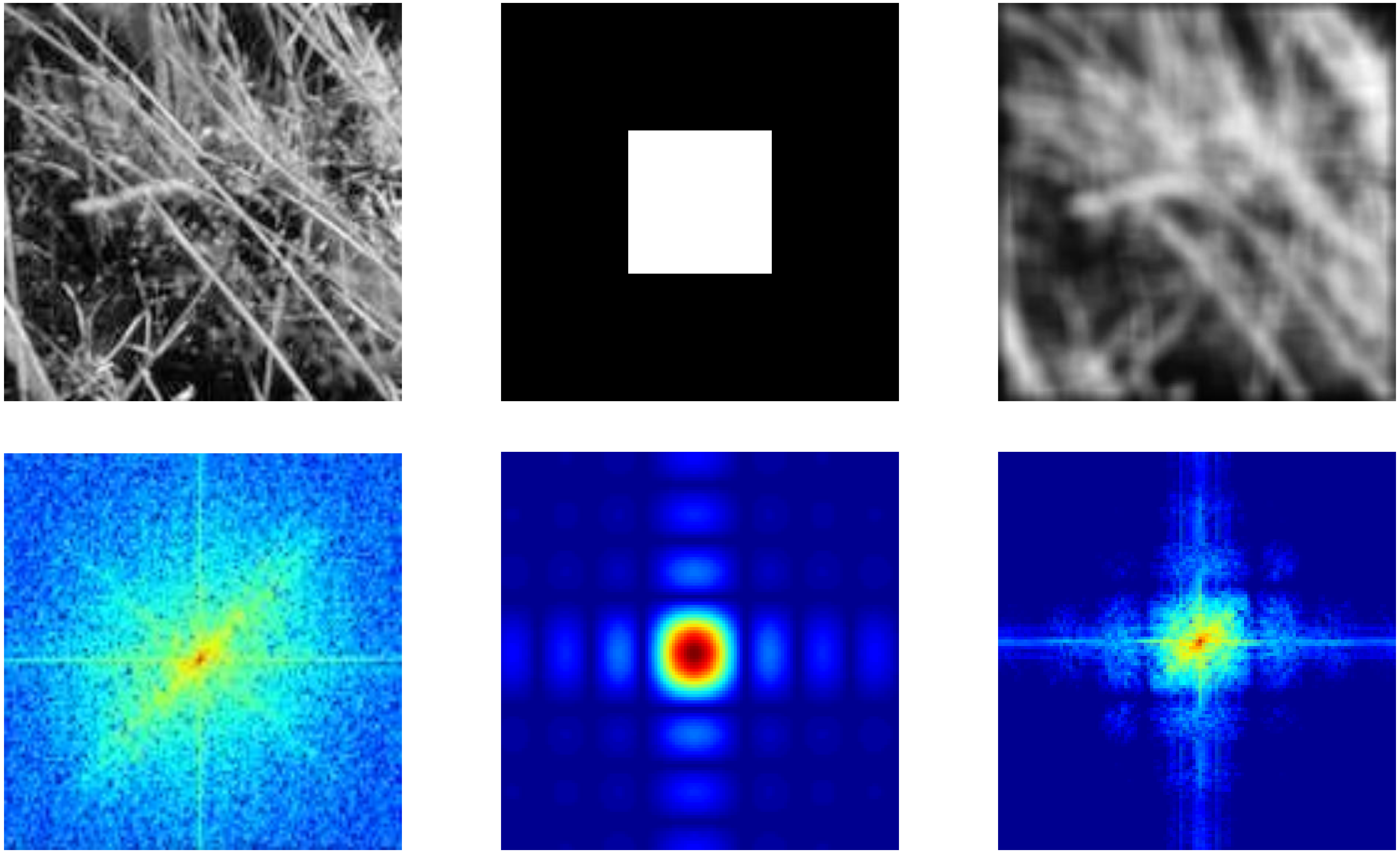
Box filter



Gaussian blur

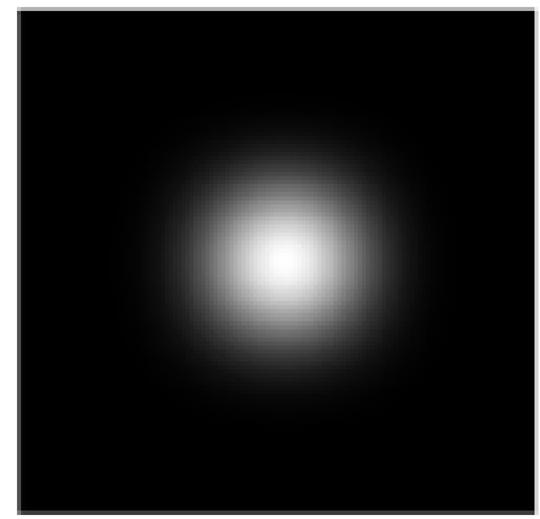


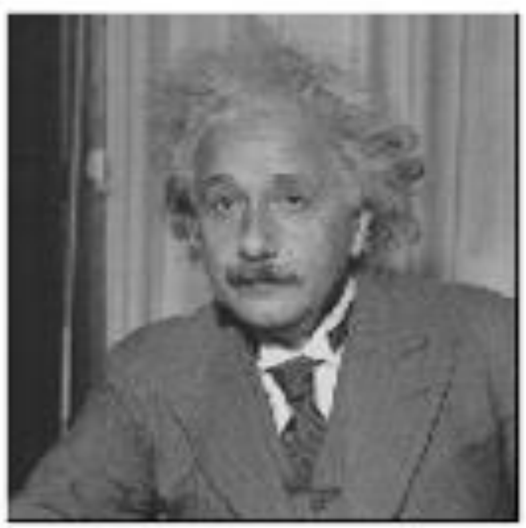
Box blur



More filtering examples

?



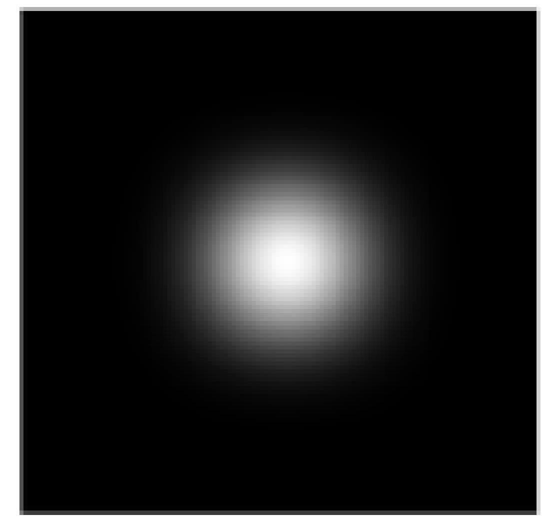


?

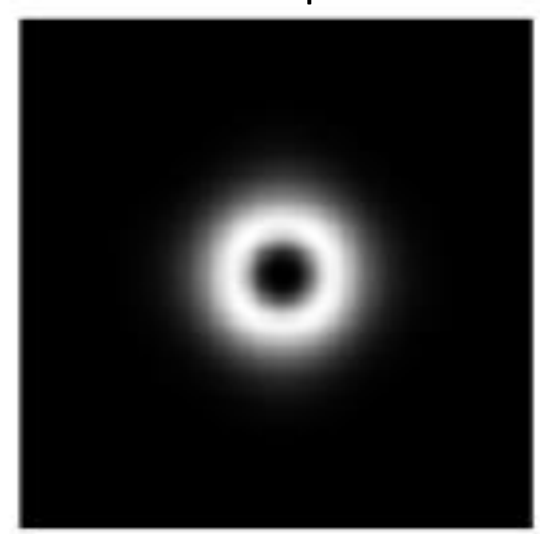
filters shown in frequency-domain

More filtering examples

low-pass



band-pass

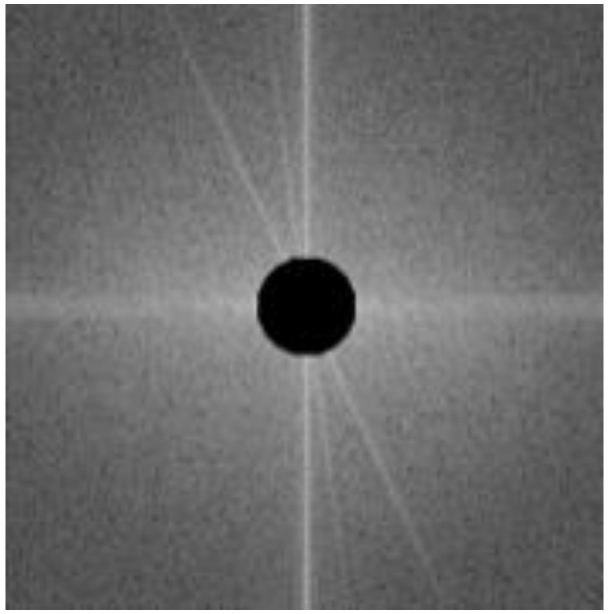


filters shown in frequency-domain

More filtering examples

?

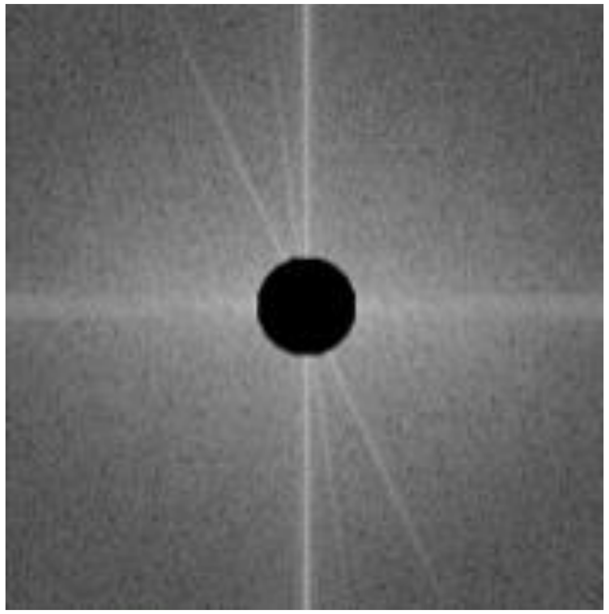
high-pass



More filtering examples

high-pass

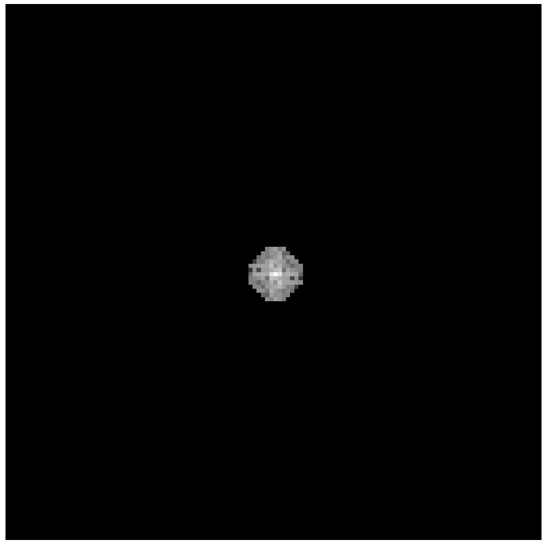




More filtering examples

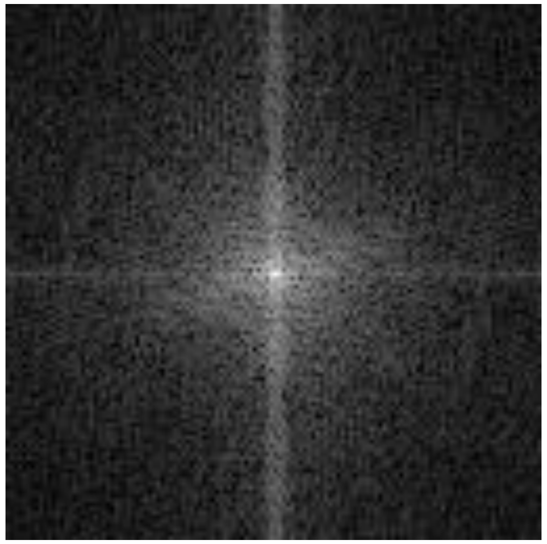
original image

low-pass filter



?

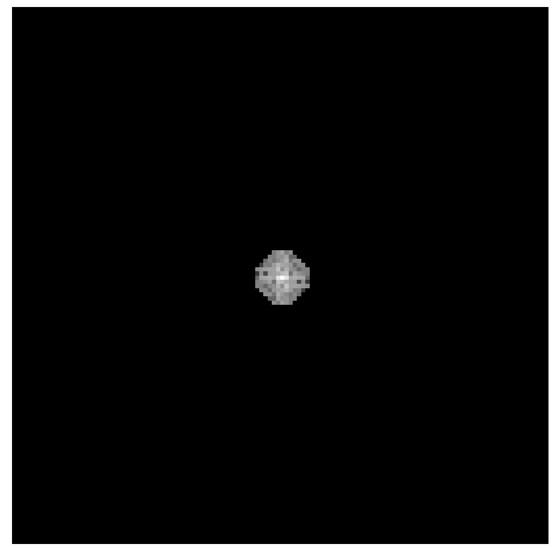
frequency magnitude



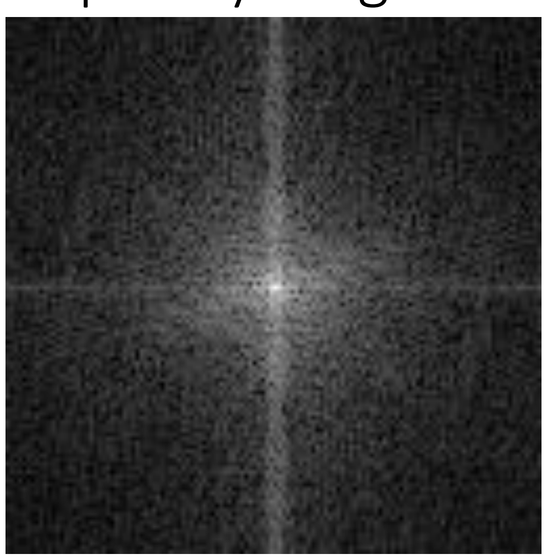
More filtering examples

original image

low-pass filter



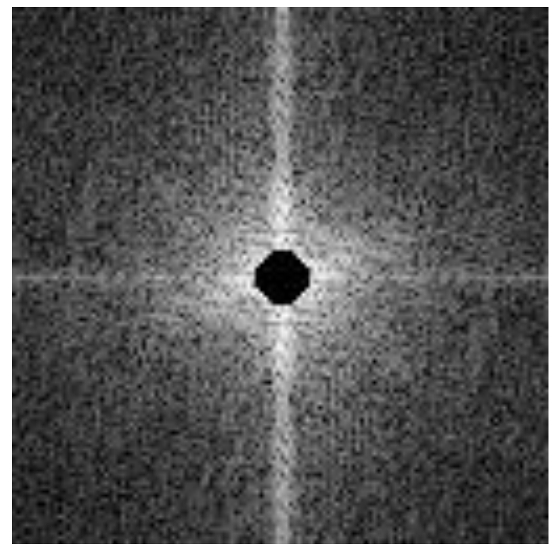
frequency magnitude



More filtering examples

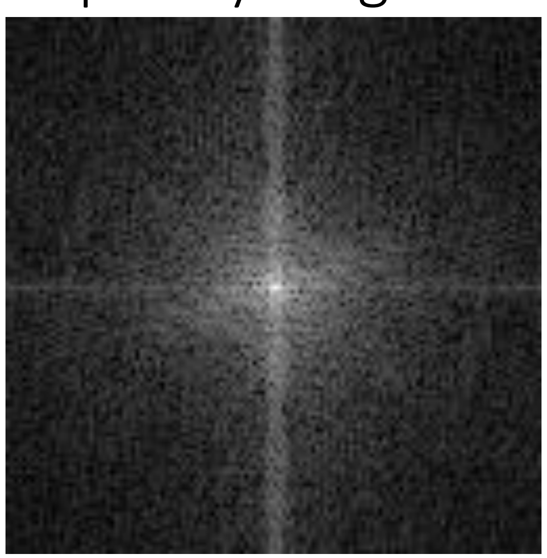
original image

high-pass filter



?

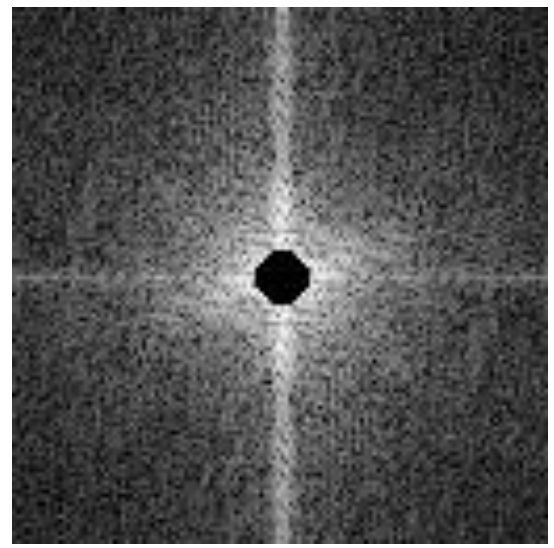
frequency magnitude



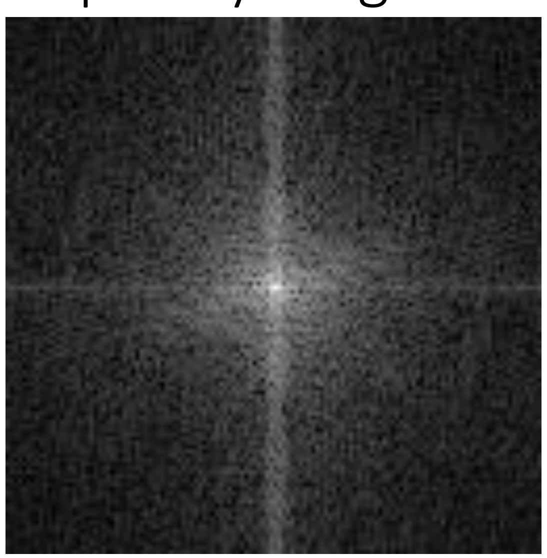
More filtering examples

original image

high-pass filter



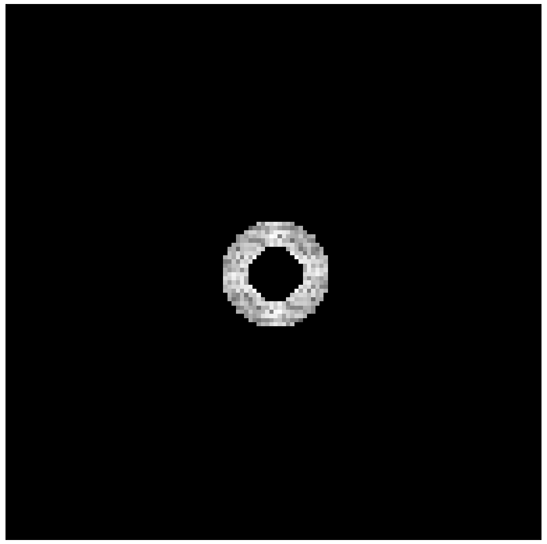
frequency magnitude

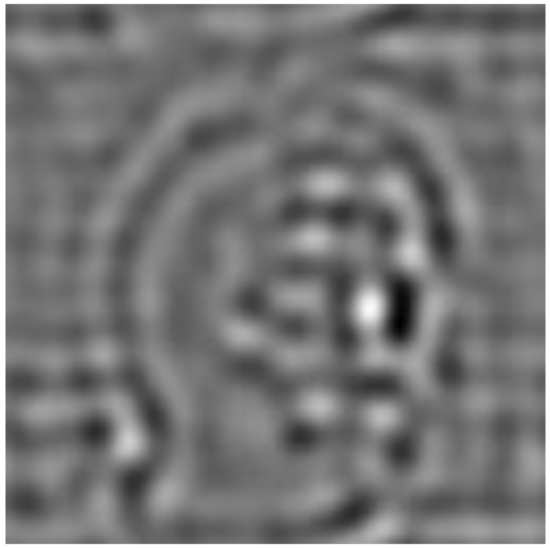


More filtering examples

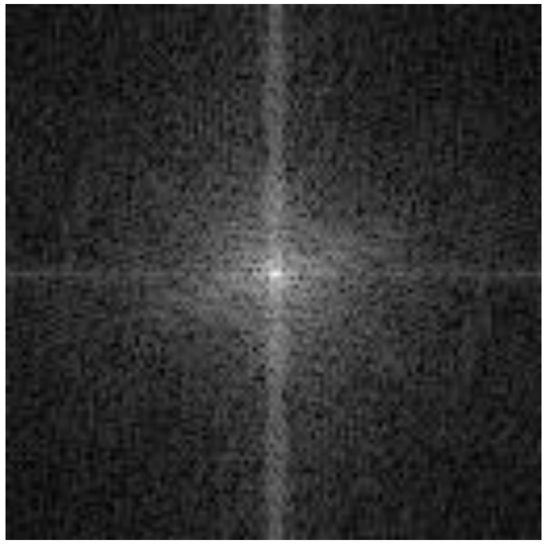
original image

band-pass filter





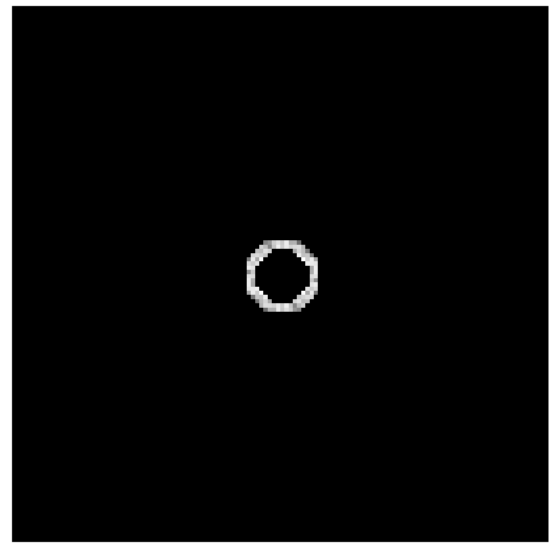
frequency magnitude

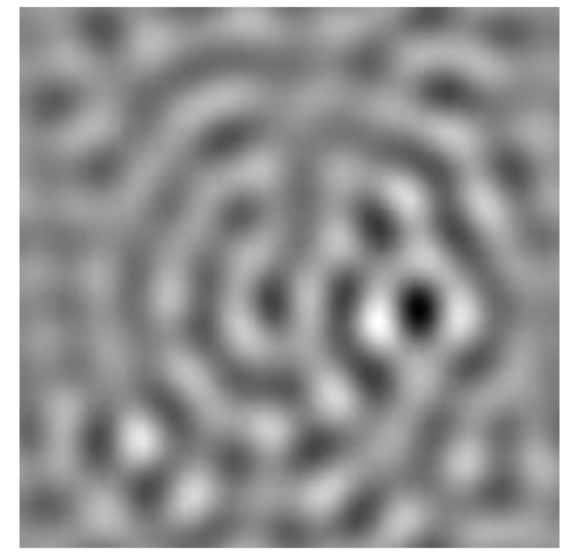


More filtering examples

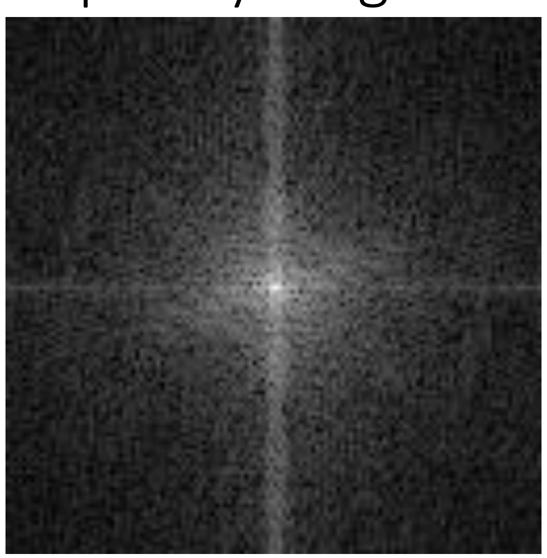
original image

band-pass filter





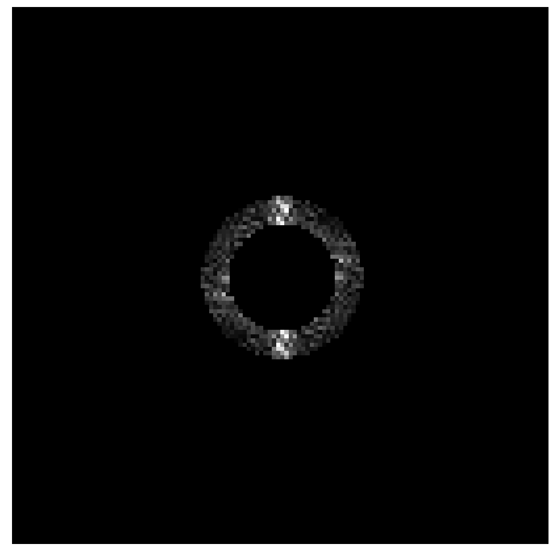
frequency magnitude

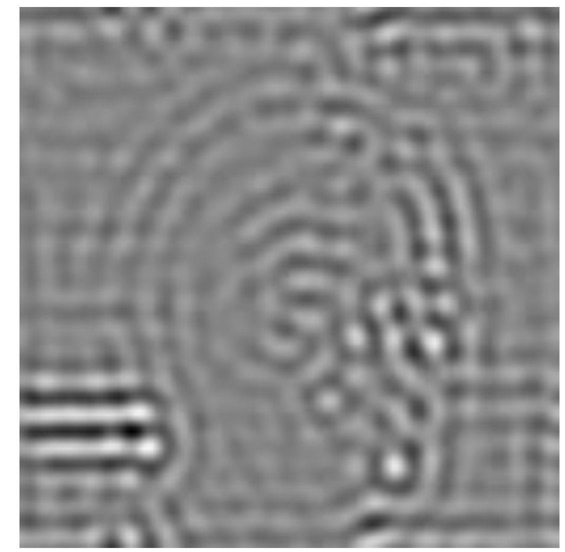


More filtering examples

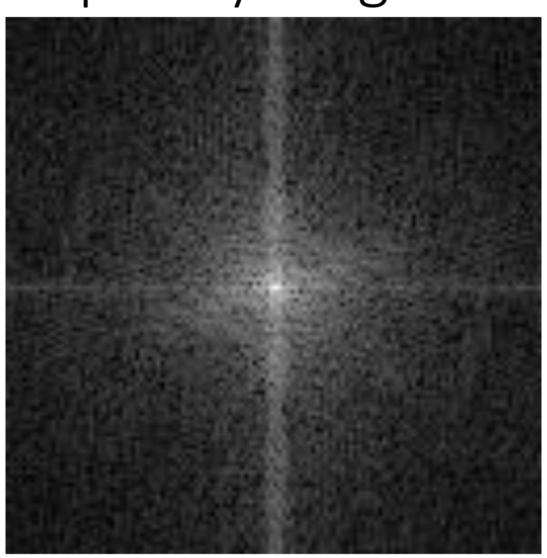
original image

band-pass filter





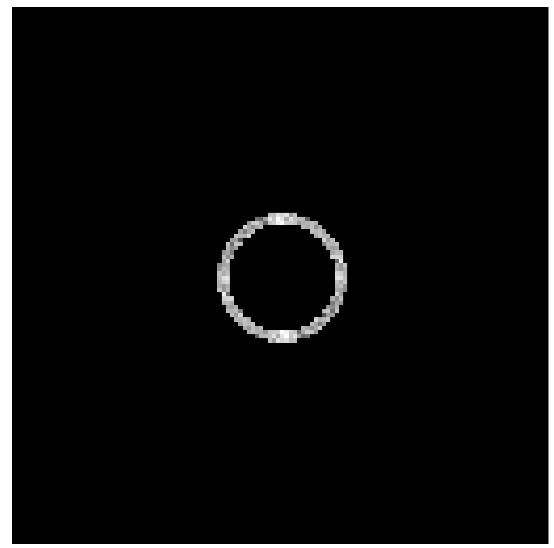
frequency magnitude

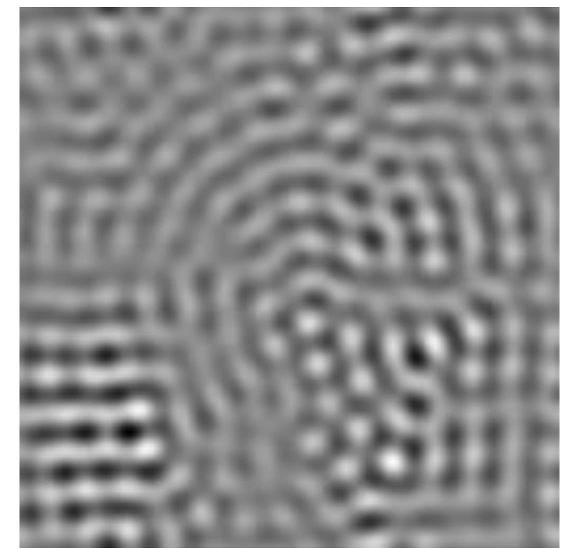


More filtering examples

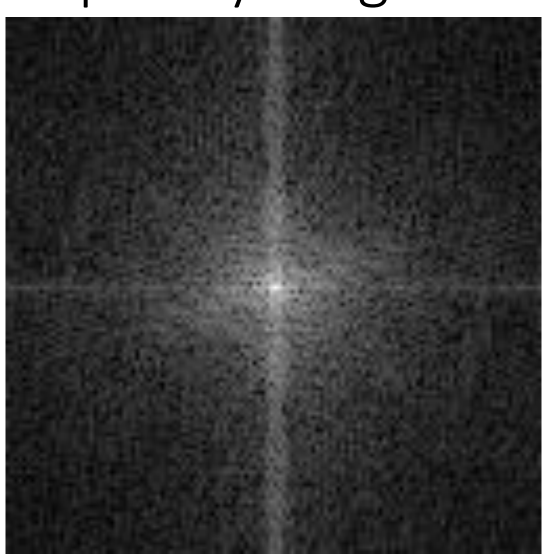
original image

band-pass filter





frequency magnitude



Revisiting sampling

The Nyquist-Shannon sampling theorem

A continuous signal can be perfectly reconstructed from its discrete version using linear interpolation, if sampling occurred with frequency:

$$f_s \geq 2 f_{\max}$$

This is called the Nyquist frequency

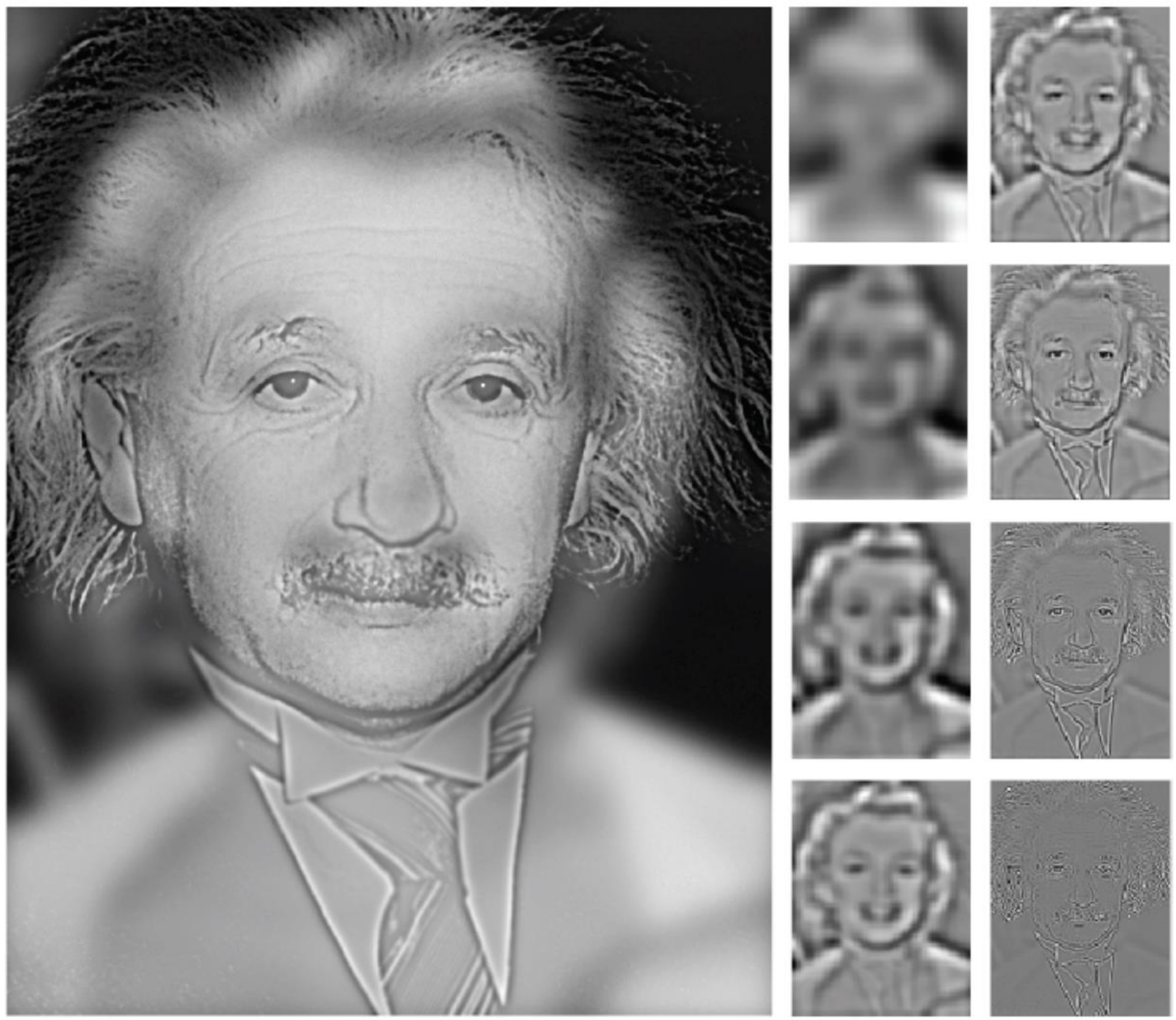
Equivalent reformulation: When downsampling, aliasing does not occur if samples are taken at the Nyquist frequency or higher.

Frequency-domain filtering in human vision

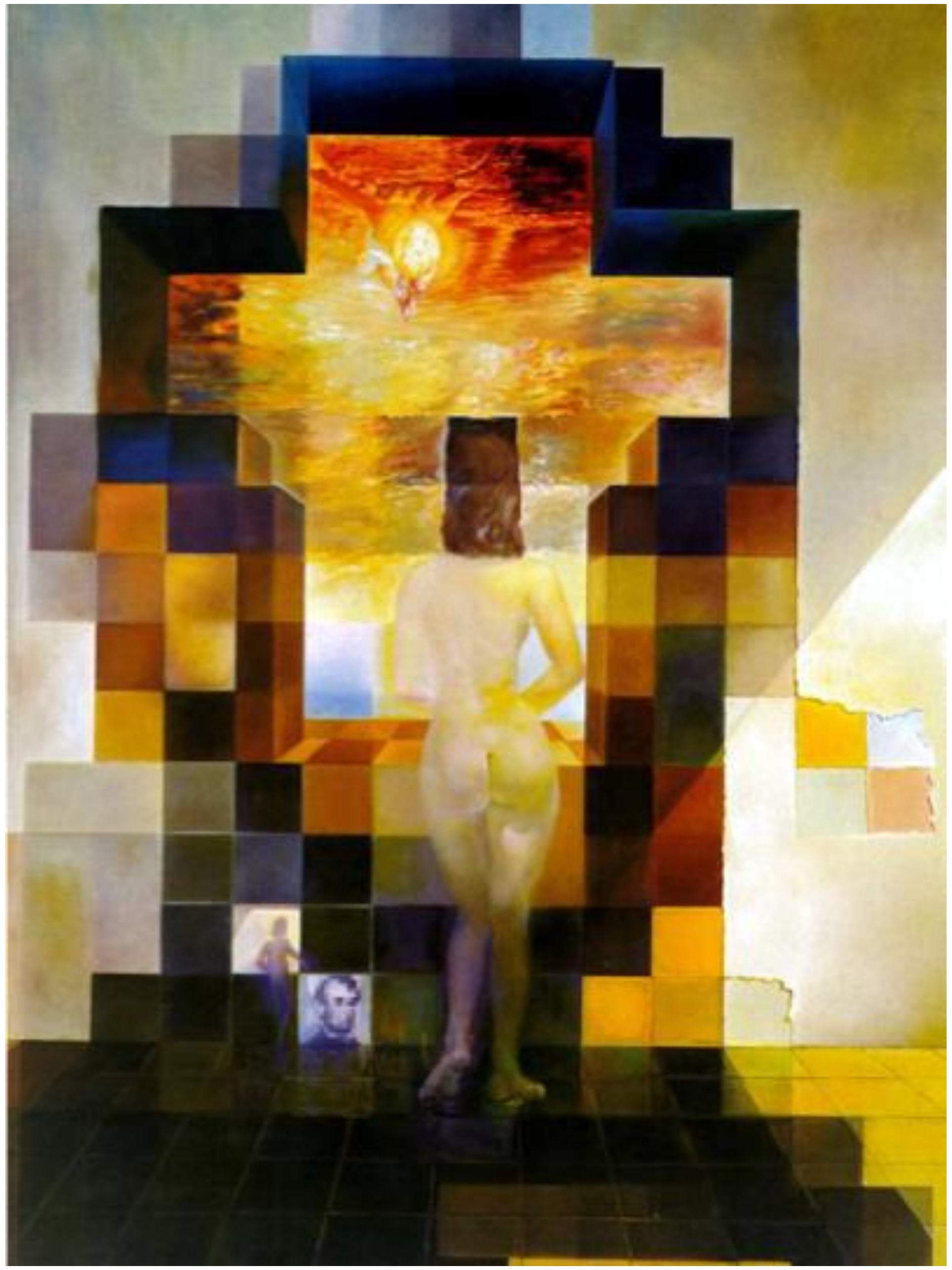
“Hybrid image”

Aude Oliva and Philippe Schyns

Frequency-domain filtering in human vision



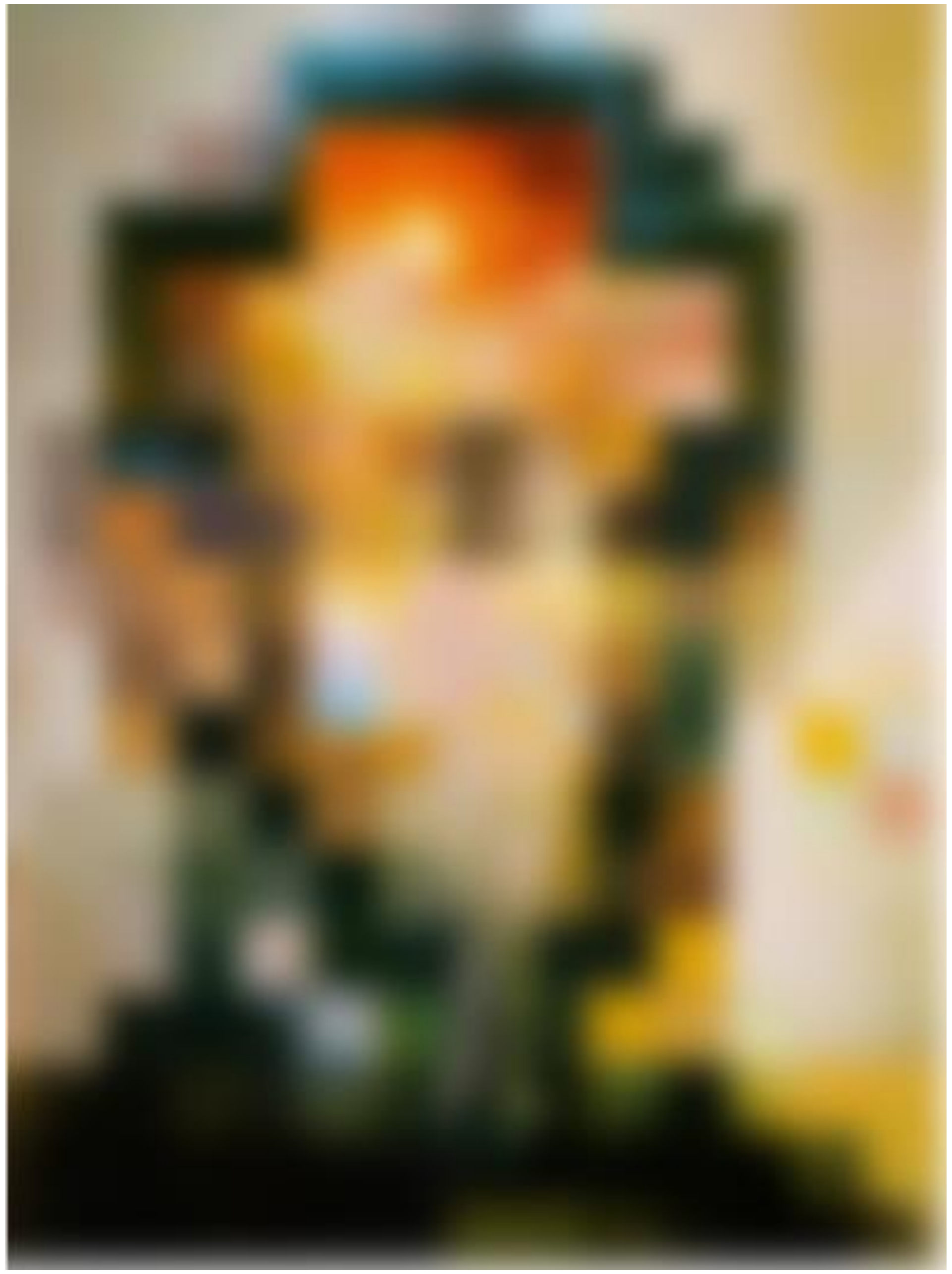
Frequency-domain filtering in human vision



Gala Contemplating the Mediterranean Sea Which at Twenty Meters Becomes the Portrait of Abraham Lincoln (Homage to Rothko)

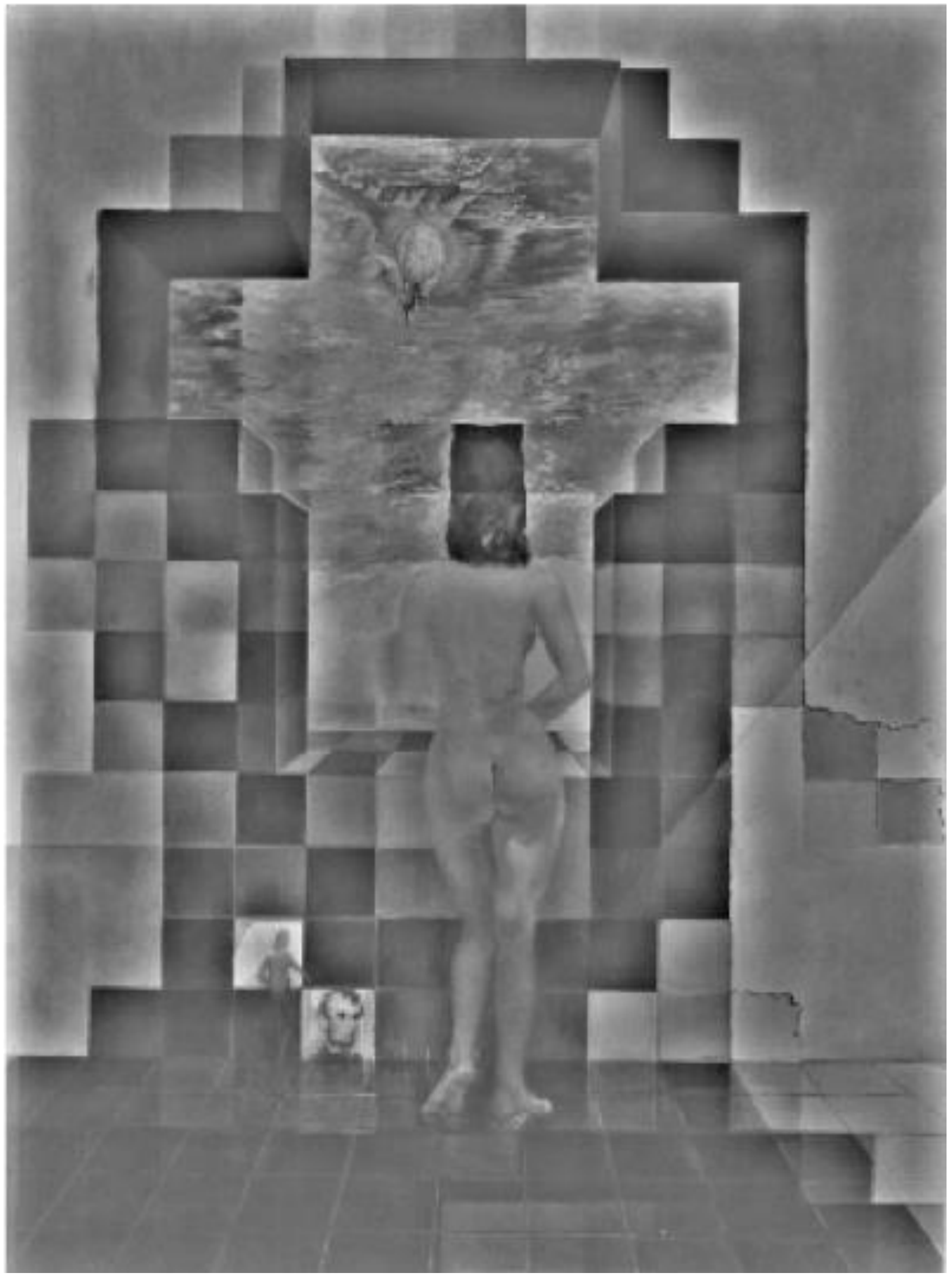
Salvador Dali, 1976

Frequency-domain filtering in human vision



Low-pass filtered version

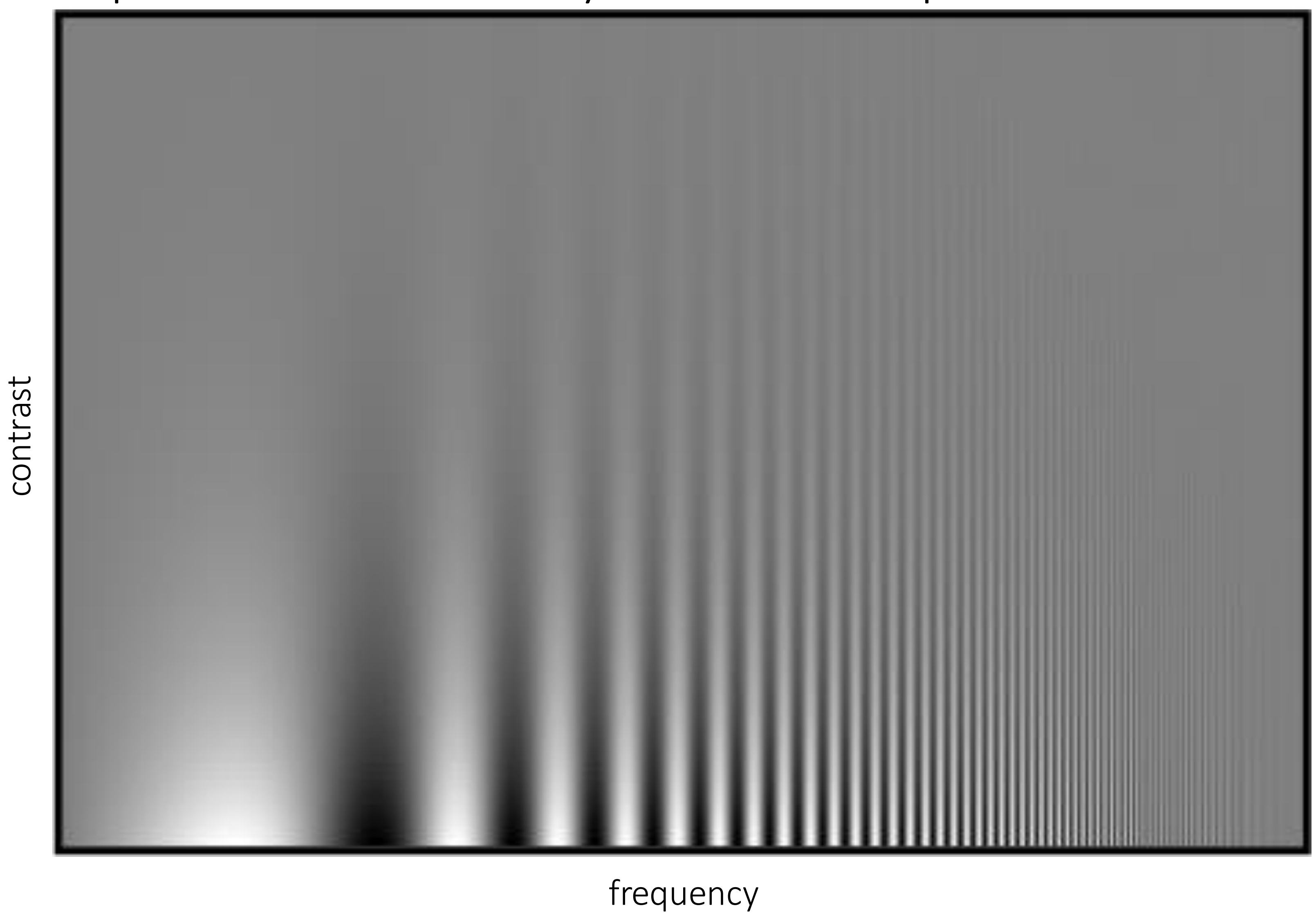
Frequency-domain filtering in human vision



High-pass filtered version

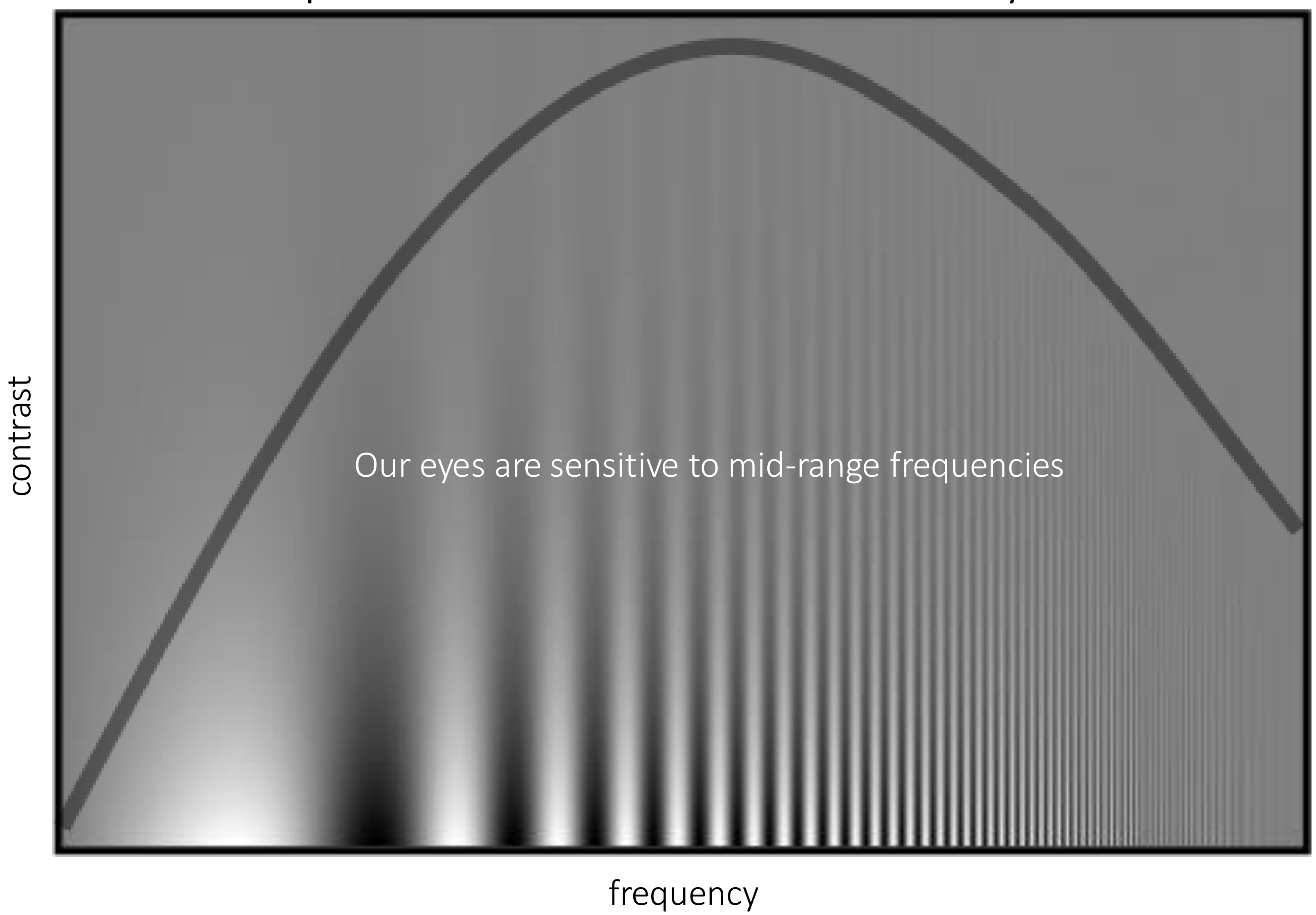
Variable frequency sensitivity

Experiment: Where do you see the stripes?



Variable frequency sensitivity

Campbell-Robson contrast sensitivity curve



- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid frequencies dominate perception

Other Properties of FT

- https://dspillustrations.com/pages/posts/misc/properties-of-the-fourier-transform.html

Properties of FT: Convolution theorem

The Fourier transform of the convolution of two functions is the product of their Fourier transforms:

$$\mathcal{F}\{g * h\} = \mathcal{F}\{g\}\mathcal{F}\{h\}$$

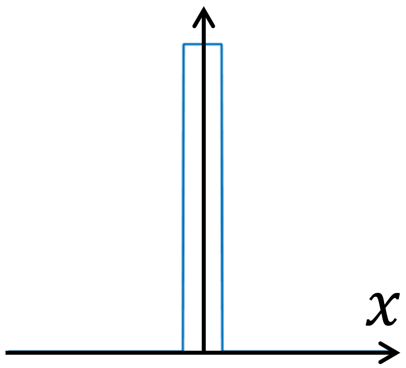
The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms:

$$\mathcal{F}^{-1}\{gh\} = \mathcal{F}^{-1}\{g\} * \mathcal{F}^{-1}\{h\}$$

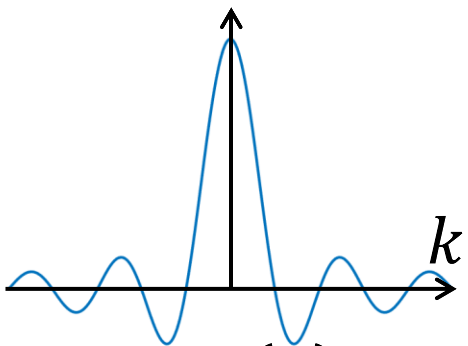
Convolution in spatial domain is equivalent to multiplication in frequency domain!

Properties of FT

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi kx} dx \qquad f(x) = \int_{-\infty}^{\infty} F(k) e^{j2\pi kx}\ dk$$

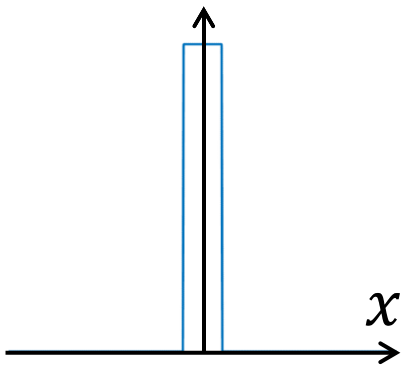


$$f(x) \xleftrightarrow{\text{Fourier}} F(k)$$



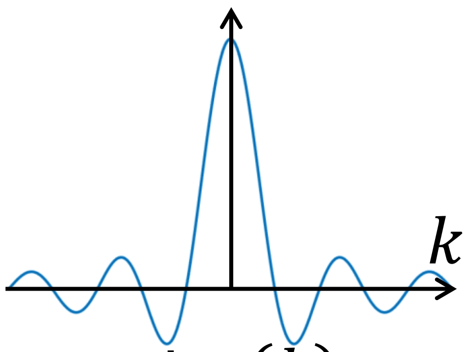
Properties of FT: scaling

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi k x} dx \qquad f(x) = \int_{-\infty}^{\infty} F(k) e^{j 2\pi k x}\, dk$$



$\mathrm{rect}(x)$

$$f(x) \overset{\text{Fourier}}{\longleftrightarrow} F(k)$$

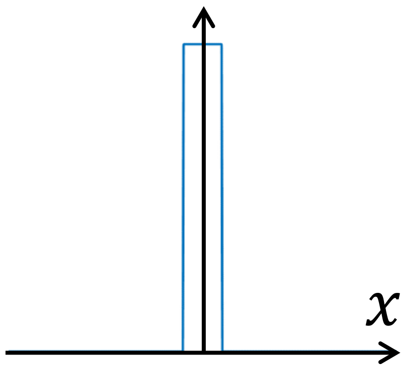


$\mathrm{sinc}(k)$

$$f(sx) \overset{\text{Fourier}}{\longleftrightarrow} ??$$

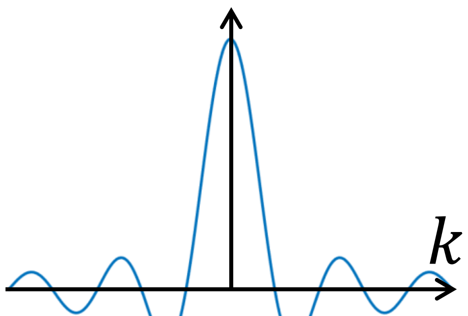
Properties of FT: scaling

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi k x} dx \qquad f(x) = \int_{-\infty}^{\infty} F(k) e^{j2\pi k x}\, dk$$



rect(x)

$$f(x) \xleftrightarrow{\text{Fourier}} F(k)$$



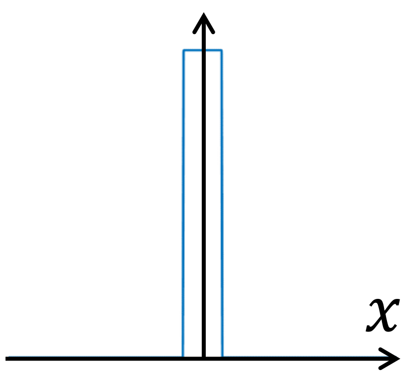
sinc(k)

??

$$f(sx) \xleftrightarrow{\text{Fourier}} F(k/s)$$

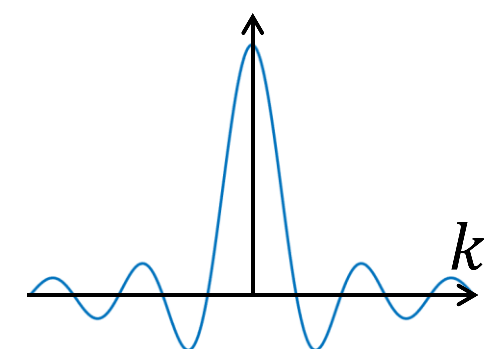
Properties of FT: scaling

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi k x} dx \qquad f(x) = \int_{-\infty}^{\infty} F(k) e^{j 2\pi k x} \, dk$$

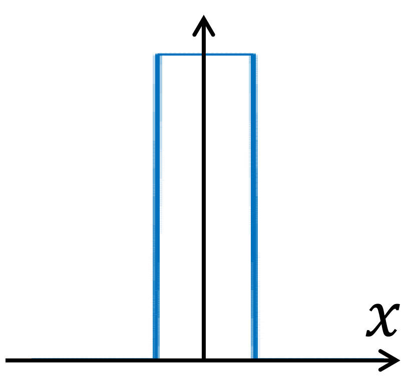


rect(x)

$$f(x) \xleftrightarrow{\text{Fourier}} F(k)$$



sinc(k)



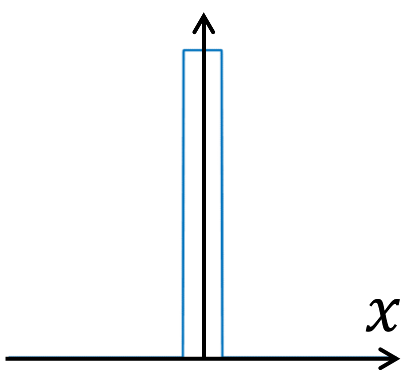
rect(sx)

$$f(sx) \xleftrightarrow{\text{Fourier}} F(k/s)$$

??

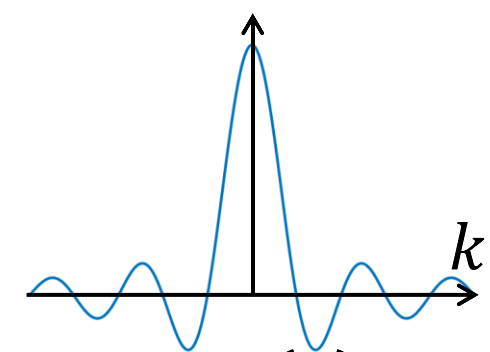
Properties of FT: scaling

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi k x} dx \qquad f(x) = \int_{-\infty}^{\infty} F(k) e^{j2\pi k x} \, dk$$

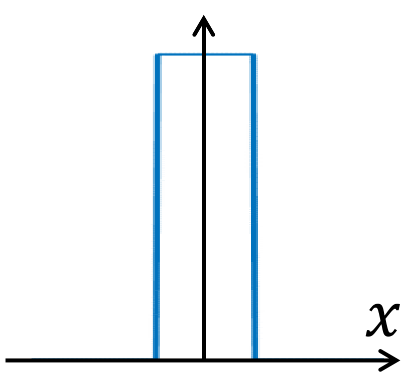


rect(x)

$$f(x) \xleftrightarrow{\text{Fourier}} F(k)$$

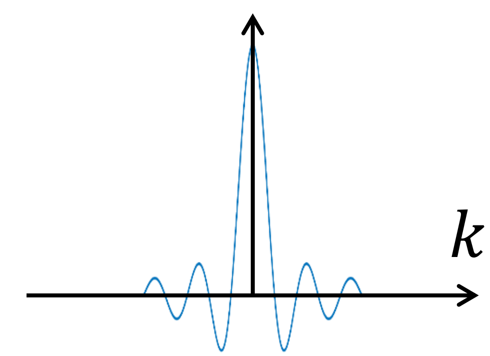


sinc(k)



rect(sx)

$$f(sx) \xleftrightarrow{\text{Fourier}} F(k/s)$$



sinc(k/s)