Fourier Analysis

Computer Vision - Lecture 03

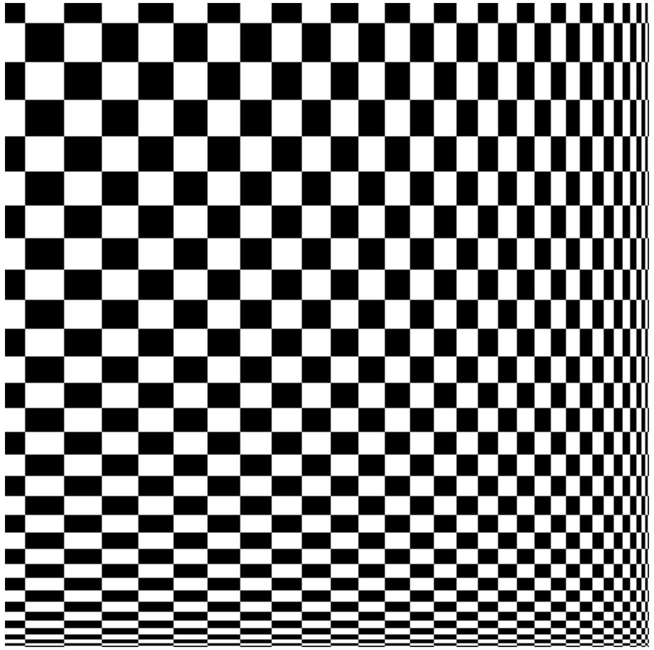
Further Reading

- Slides borrowed and adapted from S. Lazebnik, S. Seitz, A. Efros, D. Hoiem, B. Freeman, A. Zisserman

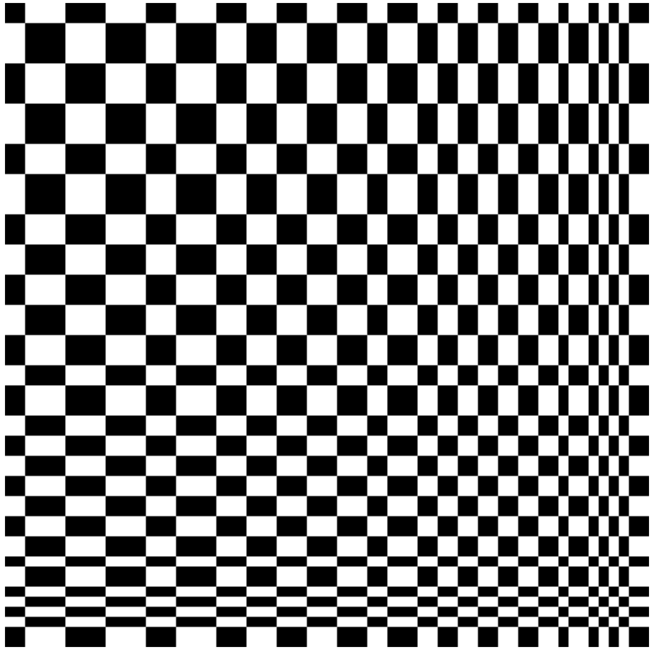
- Code for the examples in the slides is on the course website.

Observation 1

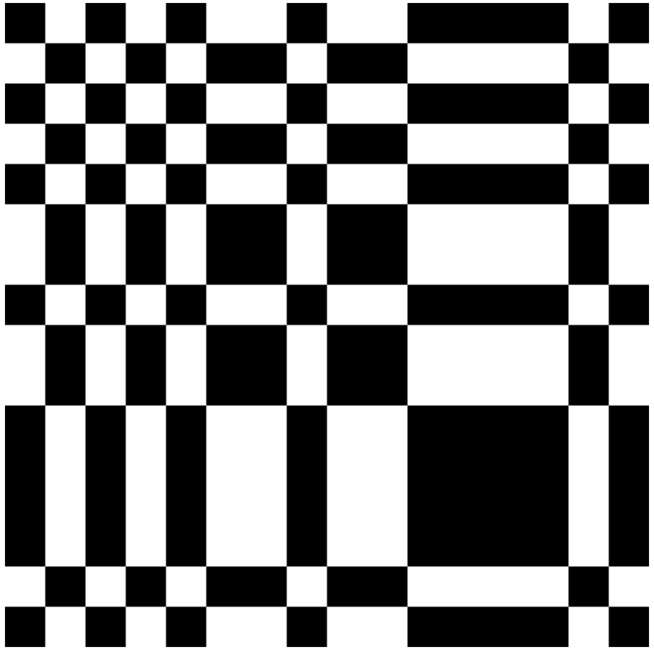
- From lecture 02: down-sampling leads to aliasing



Original: 512x512



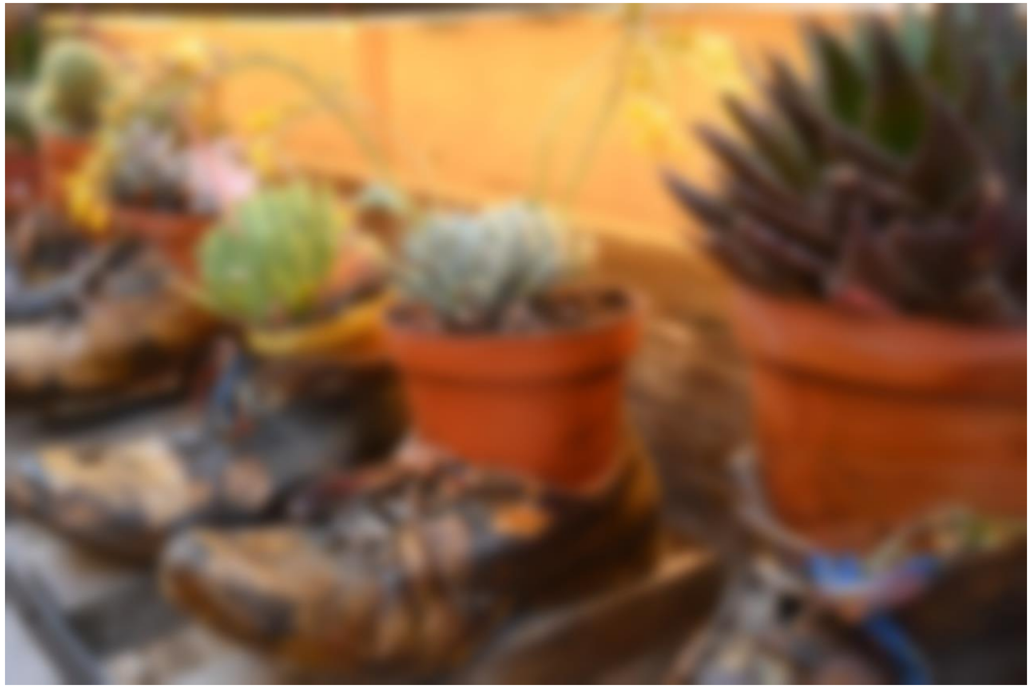
64x64



16x16

Observation 2

- From lecture 02: Gaussian Blur is smoother than average filtering



Gaussian Blur

Averaging in a square

Observation 3

- Hybrid Images (A. Oliva, A. Torralba, P.G. Schyns, *Hybrid Images*, SIGGRAPH 2006)

Source: twitter.com/NerdyRodent Sep 24,'23

Fourier Analysis

- Intuition: all these phenomena relate to fast and slow changing components of the image.

- To understand these better, we need a tool to analyse the frequency components of an image.

- For better understanding, we will start with 1D signals before moving to 2D.

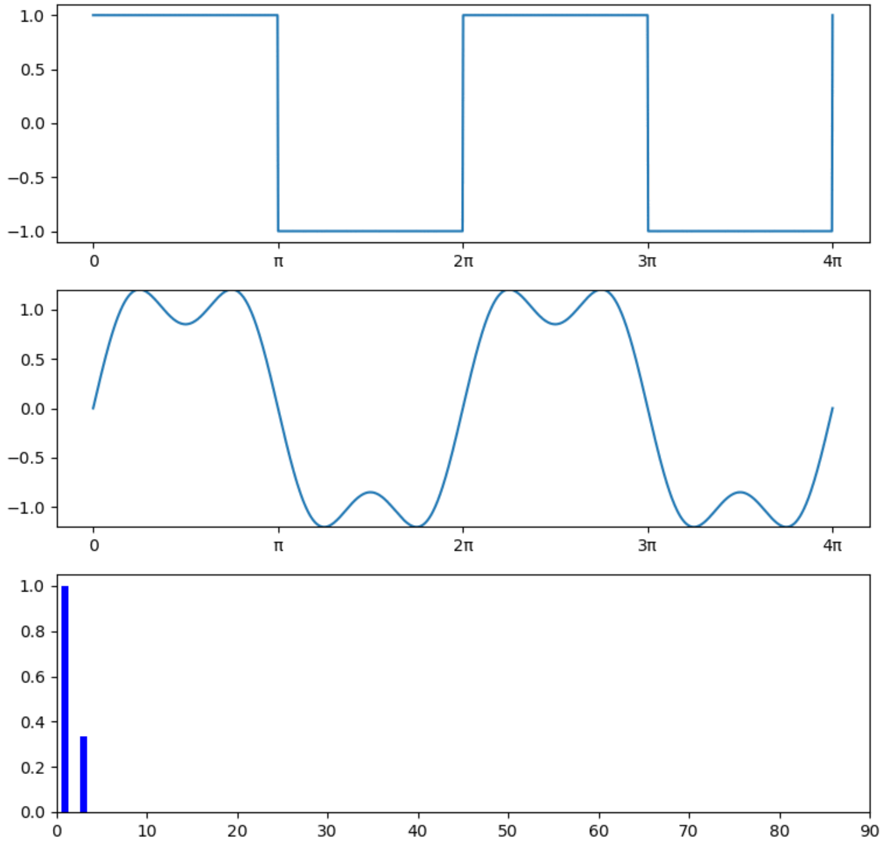
Fourier Transform

Any(**) univariate function can be expressed as a weighted sum of sinusoids of different frequencies (1807)

Jean-Baptiste Joseph Fourier (1768-1830)

Example: series for a square wave

$$\sum_{k=1,3,5,\ldots}^{\infty} \frac{1}{k}\sin(kt)$$



Slide credit: S. Lazebnik

Fourier analysis

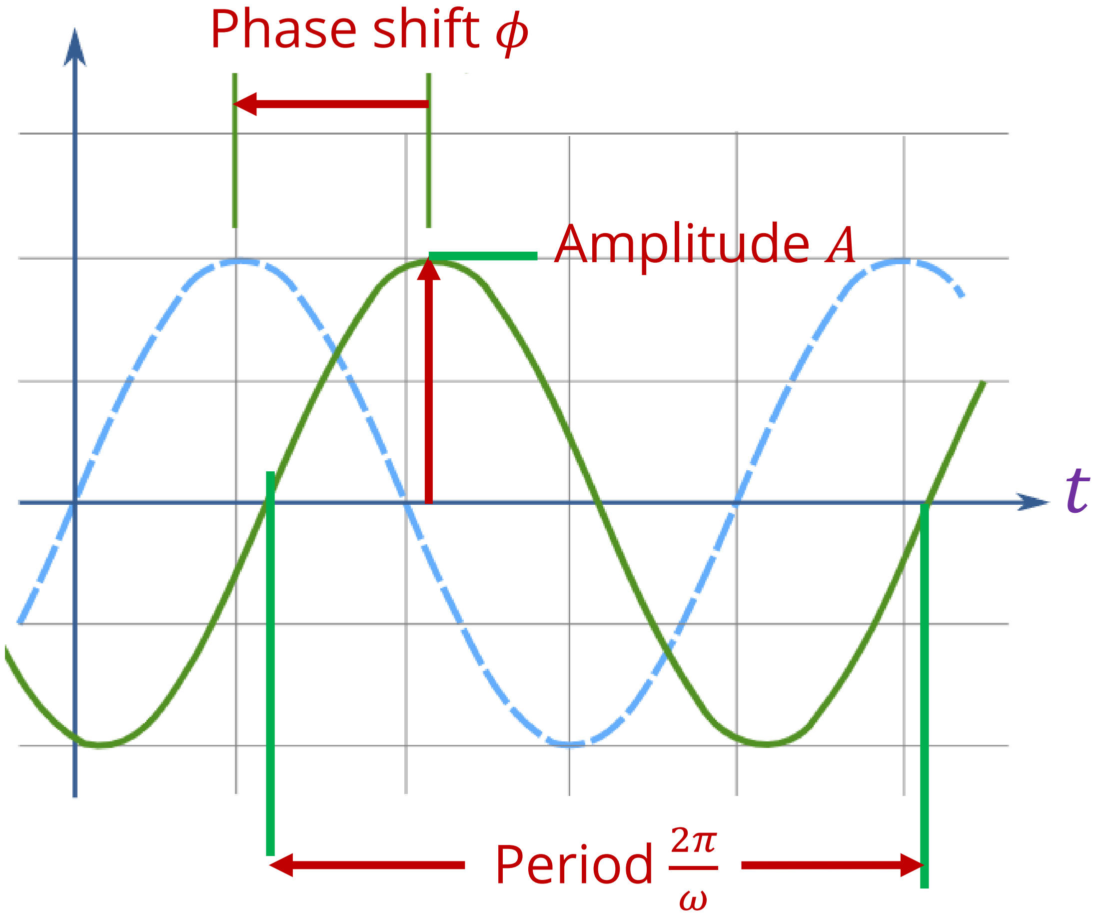
Our building block:

Frequency

$$A \sin(\omega t + \phi)$$

Amplitude

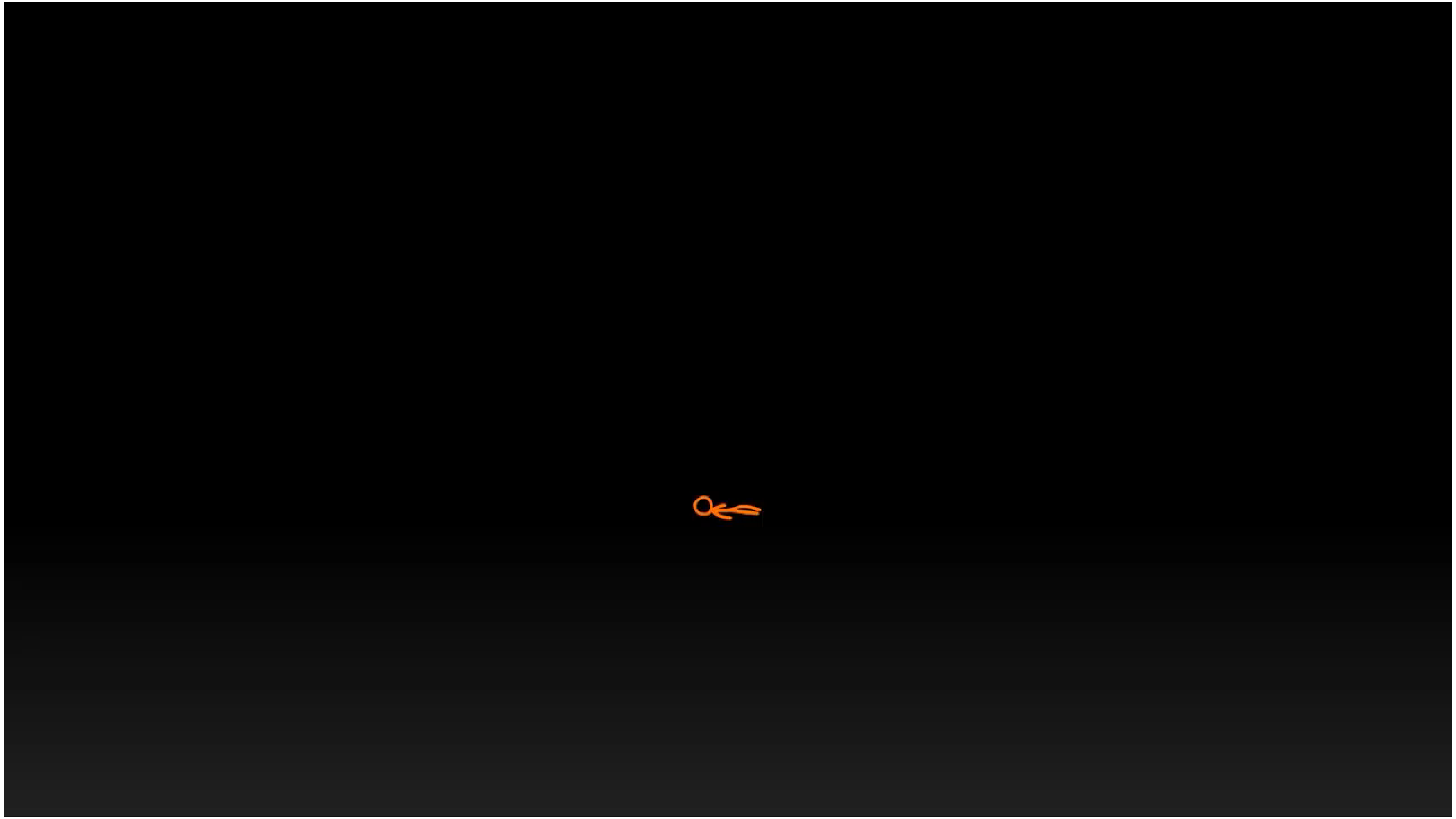
Phase



Add enough of these to get any signal you want!

Slide credit: S. Lazebnik

Complex Exponentials



Animation vs. Math, https://www.youtube.com/watch?v=B1J6Ou4q8vE

Complex Exponentials

- Euler's Identity: $e^{i\pi} + 1 = 0$
- Euler's Formula: $e^{i\phi} = \cos\phi + i\sin\phi$
- Identity: $e^{i\pi} = \cos\pi + i\sin\pi = -1 + 0$

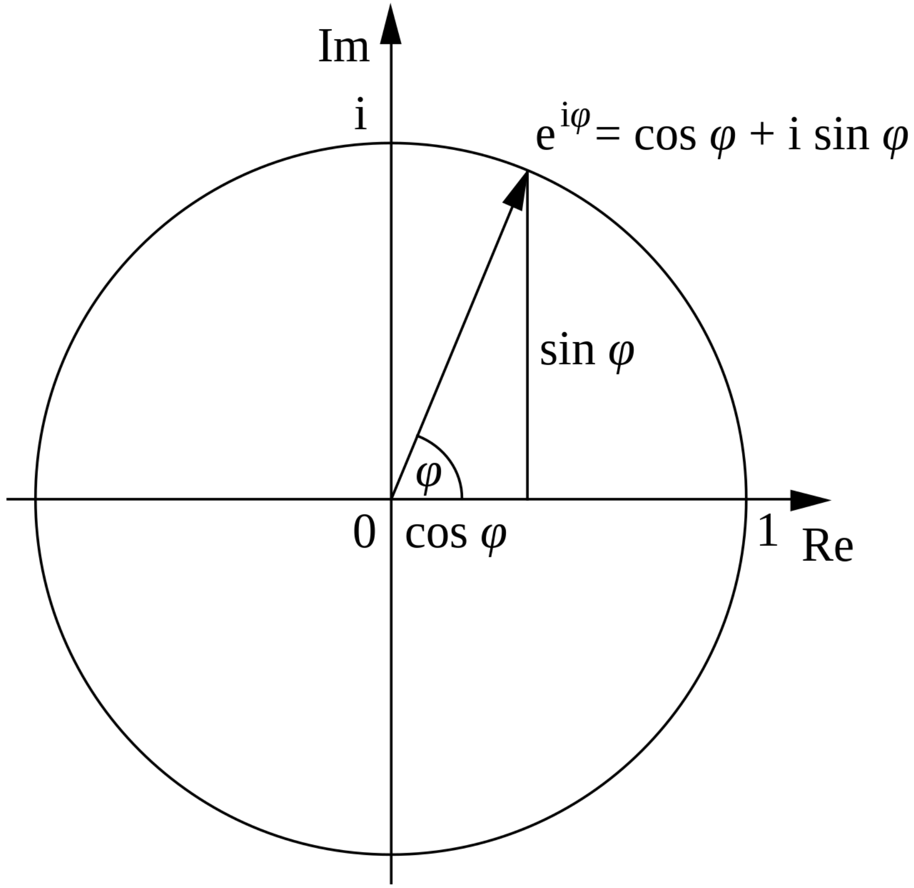


Image source

Basis Functions

Define a set of functions to use as a basis:

$$\psi_u(t) = e^{i2\pi ut}, \qquad u \in (-\infty, \infty)$$

Given a signal $f(t)$, we can represent it as a weighted combination of the basis functions with weights $F(u)$:

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ut} du$$

Basis functions

Inner product for complex functions is:

$$\langle g, h \rangle = \int_{-\infty}^{\infty} g(t) h^*(t) dt$$

Complex conjugate:
real part stays the same,
imaginary part is flipped

$$(a + ib)^* = a - ib$$

Our basis is orthonormal:

$$\langle \psi_{u_1}, \psi_{u_2} \rangle = \begin{cases} 1 & \text{if } u_1 = u_2 \\ 0 & \text{otherwise} \end{cases}$$

Finding the weights

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ut} du$$

To express f with the basis functions ψ_u, we need to find the weights $F(u)$.

$$F(u) = \langle f, \psi_u \rangle = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ut} dt$$

Fourier Transform

- *Analysis* process, decomposing a complex-valued function $f(t)$ into its constituent frequencies $F(u)$.

- The inverse process is *synthesis*, which recreates $f(t)$ from $F(u)$.

Fourier Transform

For each u, $F(u)$ is a *complex number* that encodes both the *amplitude* A and *phase* ϕ of the sinusoid $A\sin(2\pi ut + \phi)$ in the decomposition of $f(t)$:

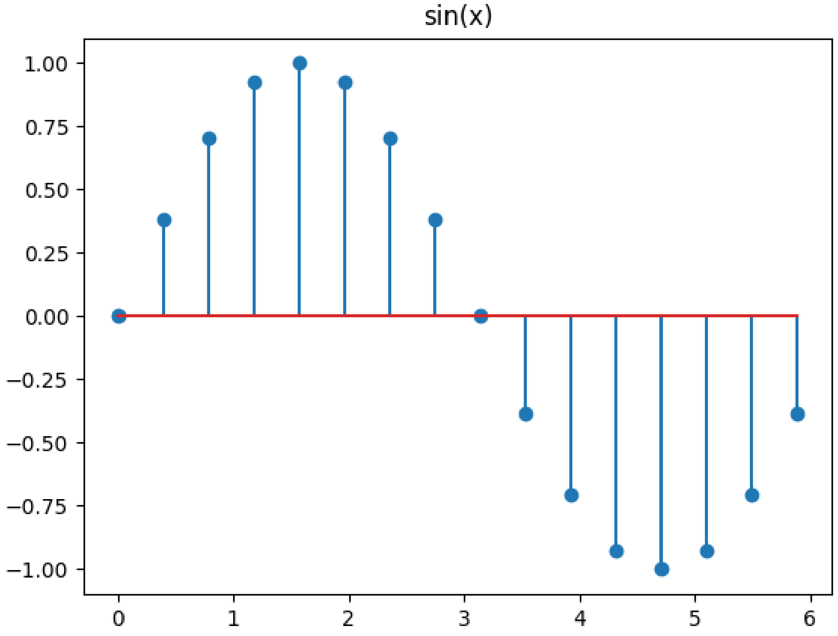
$$F(u) = \mathrm{Re}(F(u)) + i\,\mathrm{Im}(F(u))$$

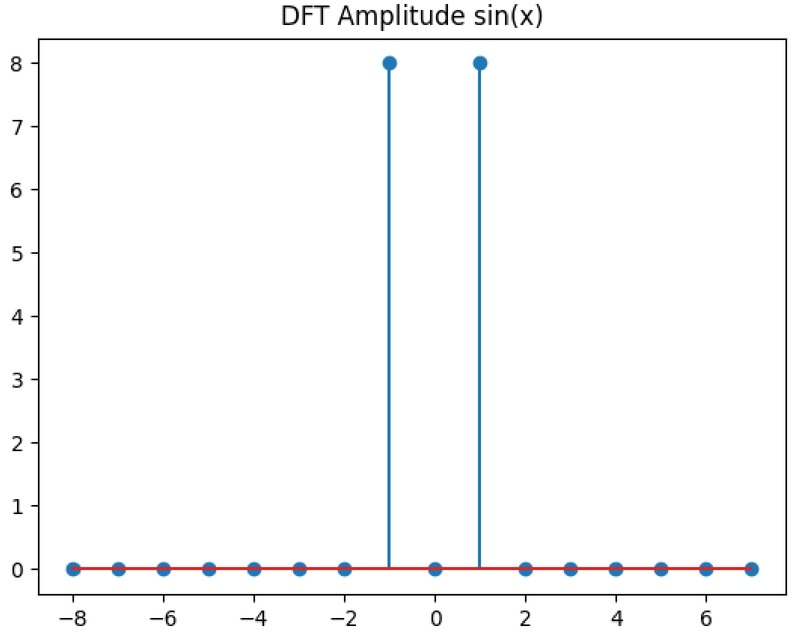
$$A = |F(u)| = \sqrt{\mathrm{Re}(F(u))^2 + \mathrm{Im}(F(u))^2}, \qquad \phi = \tan^{-1}\frac{\mathrm{Im}(F(u))}{\mathrm{Re}(F(u))}$$

If $f(t)$ is real, then $\mathrm{Re}(F(u)) = \mathrm{Re}(F(-u))$
$\mathrm{Im}(F(u)) = -\mathrm{Im}(F(-u))$

Discrete Fourier Transform

When we only have N discrete (evenly spaced) samples from a signal, we also only need a discrete set of N basis functions.



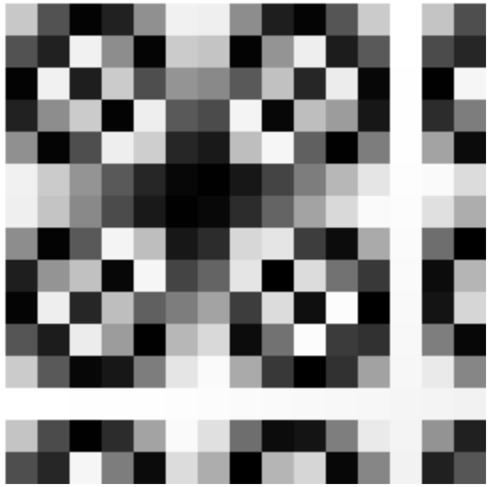
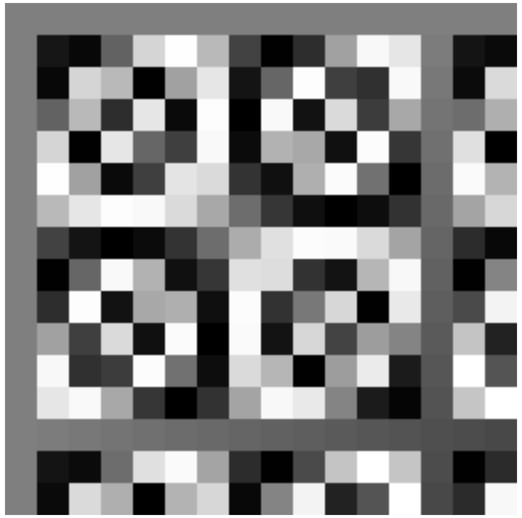


Discrete Fourier Transform

$$F(k) = \langle f, \psi_k \rangle = \sum_{n=0}^{N-1} f(n) e^{-i\frac{2\pi k}{N}n}$$

- For each k we compute the dot-product between the discrete signal f and a discrete basis function ψ_k.
- This is just a matrix-vector multiplication!

Basis

$F =$  $* ($  $+i$  $)$

f^T

Inverse DFT

We will use U for the basis matrix.

- Forward DFT:

$$F(k) = \sum_{n=0}^{N-1} f(n)\exp\left(-i\frac{2\pi}{N}kn\right), \text{ or } F = Uf$$

- Inverse DFT:

$$f(n) = \frac{1}{N}\sum_{k=0}^{K-1} F(k)\exp\left(i\frac{2\pi}{N}kn\right), \text{ or } f = \frac{1}{N}U^{-1}F$$

U^{-1} is the transpose of the *complex conjugate* of U

Periodicity of DFT and inverse DFT

The result of DFT is periodic: because $F(k)$ is obtained as a sum of complex exponentials with a common period of N samples:

$$F(k + aN) = \sum_{n=0}^{N-1} f(n) \exp\left(-i \frac{2\pi}{N} n(k + aN)\right)$$

$$= \sum_{n=0}^{N-1} f(n) \exp\left(-i \frac{2\pi n}{N} k\right) \exp(-i2\pi an) = F(k)$$

Likewise, the result of the inverse DFT is a periodic signal: $f(t + aN) = f(t)$ for any integer a.

2D Fourier Analysis

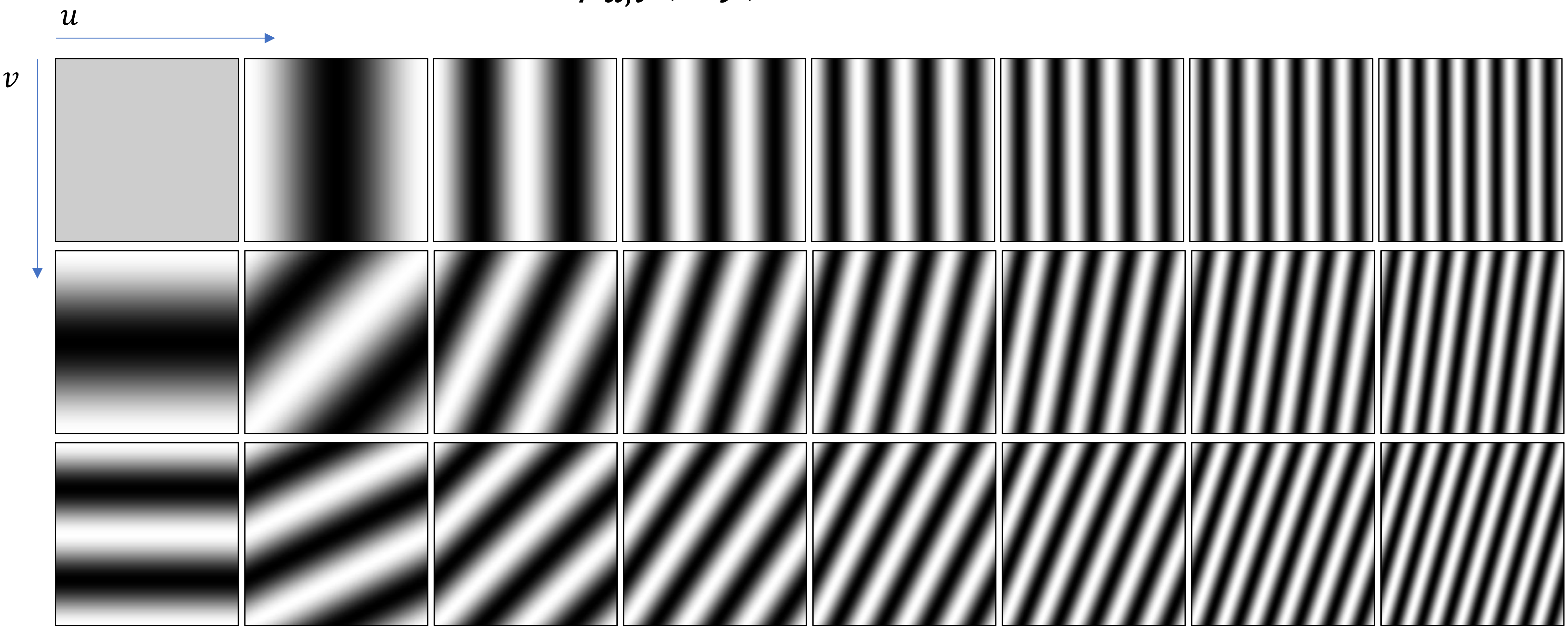
First, we need 2D basis functions:

$$\begin{aligned}\psi_{u,v}(x,y) &= e^{i2\pi ux}e^{i2\pi vy} \\ &= e^{i2\pi(ux+vy)} \\ &= \cos 2\pi(ux+vy) + i\sin 2\pi(ux+vy)\end{aligned}$$

2D Basis Functions

real

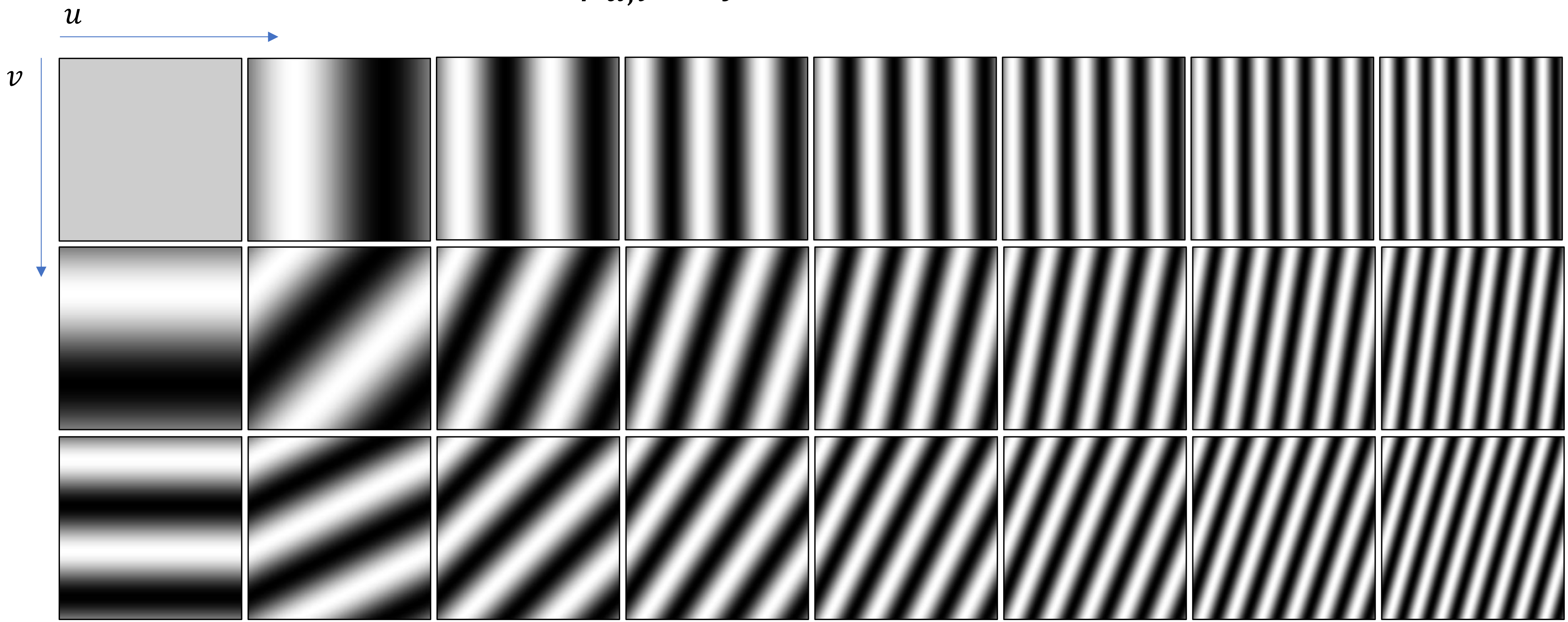
$$\psi_{u,v}(x,y) = e^{i2\pi(ux+vy)}$$



2D Basis Functions

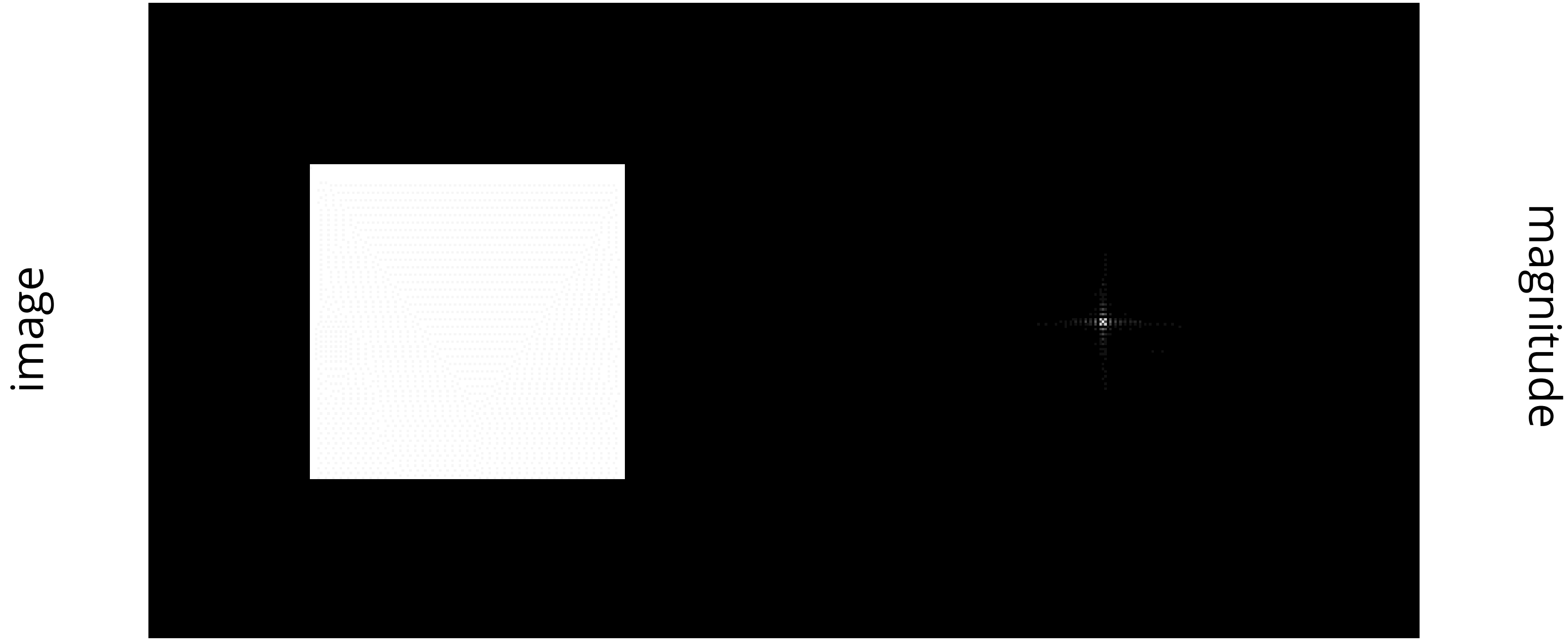
imaginary

$$\psi_{u,v}(x,y) = e^{i2\pi(ux+vy)}$$



Examples – Image Transformations

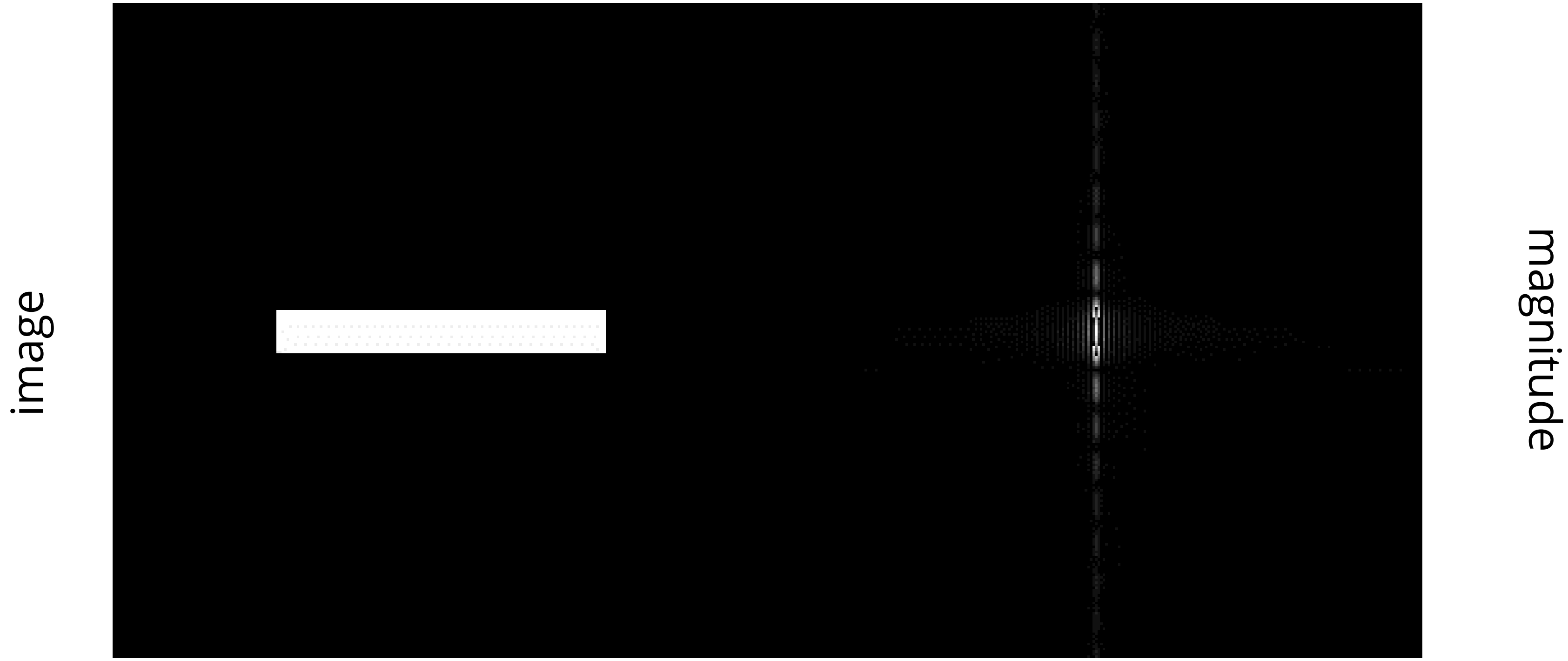
What happens when we scale the image?



Inverse effect for the magnitude

Examples – Image Transformations

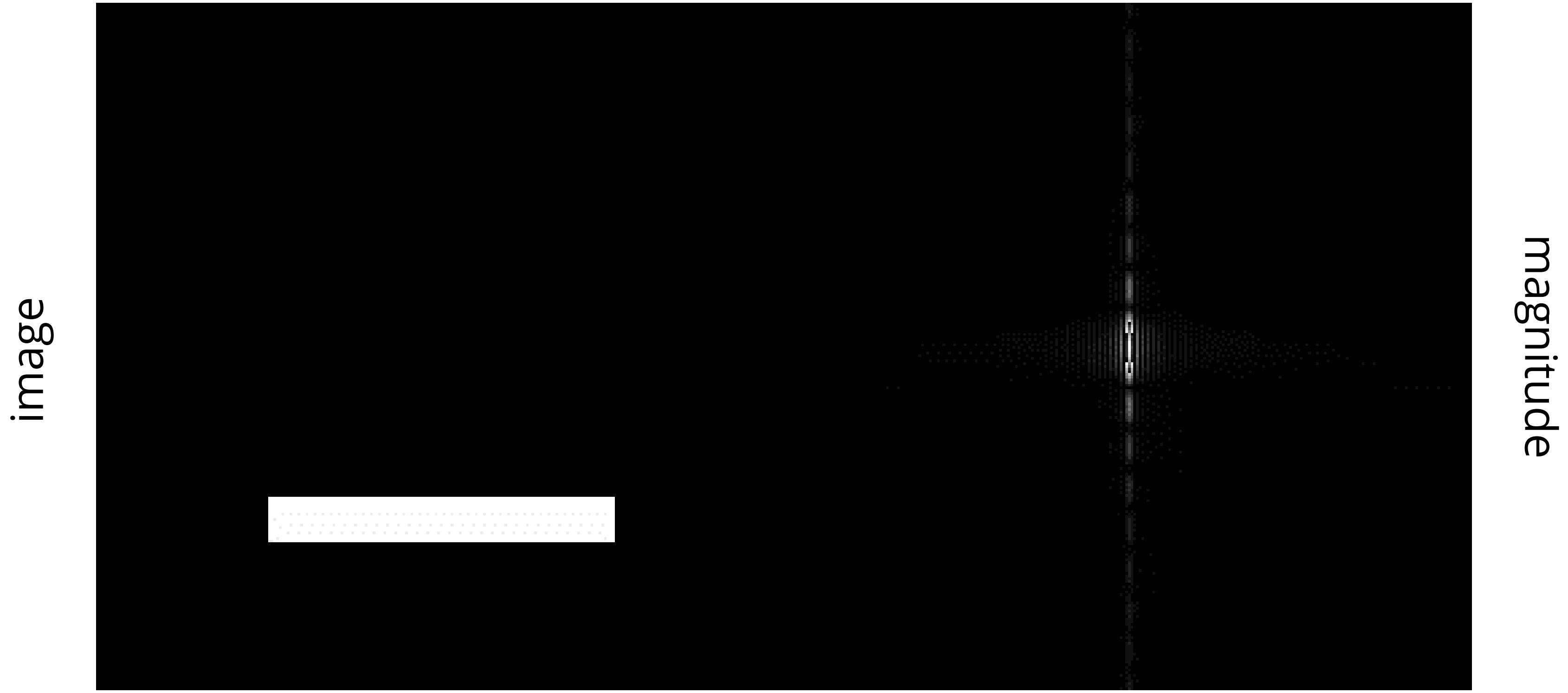
What happens when we rotate the image?



Rotates the same way.

Examples – Image Transformations

What happens when we translate the image?

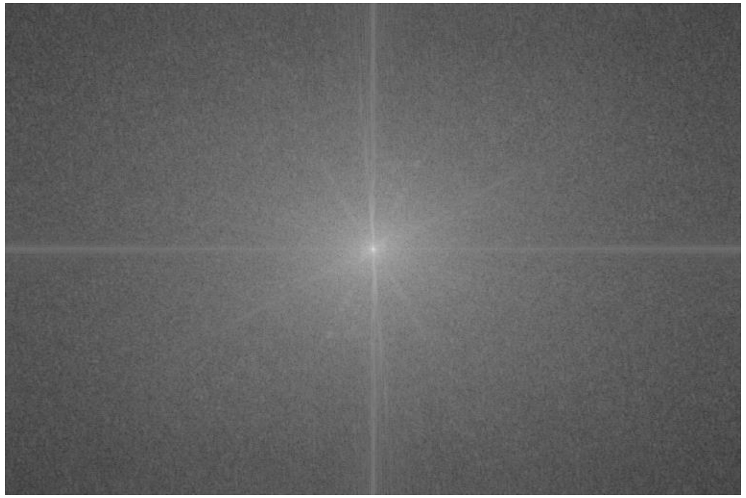


Translation does not affect the magnitude (only the phase).

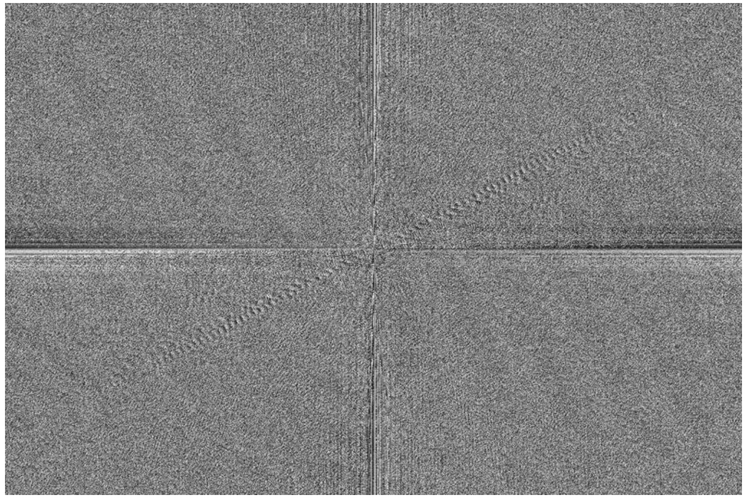
Real Images

image

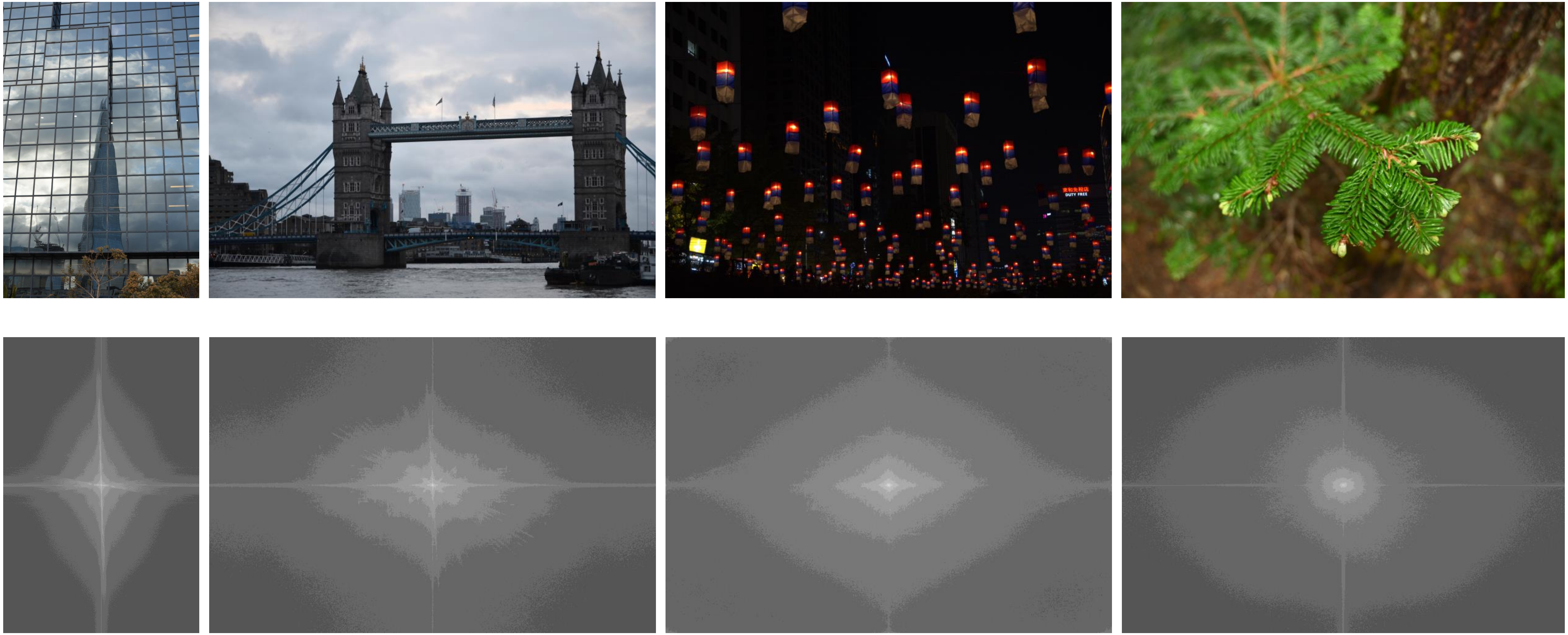
magnitude



phase



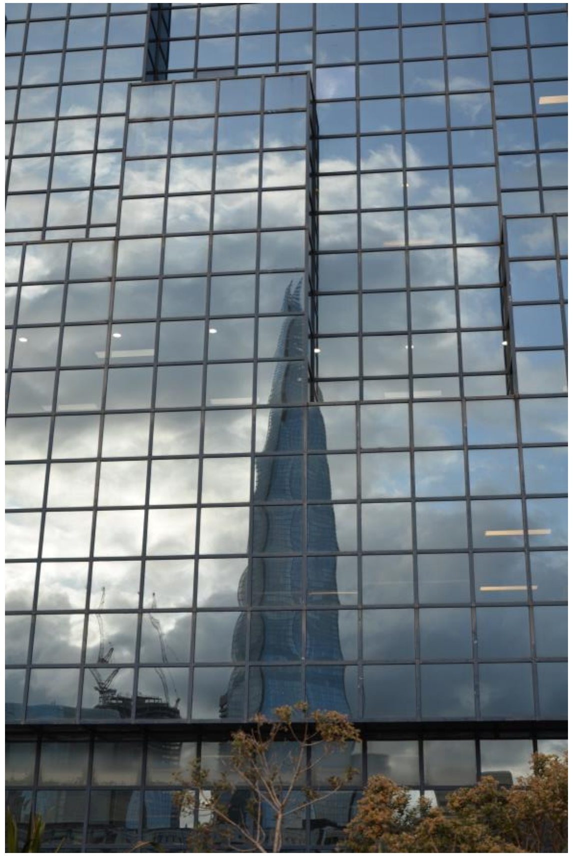
Examples



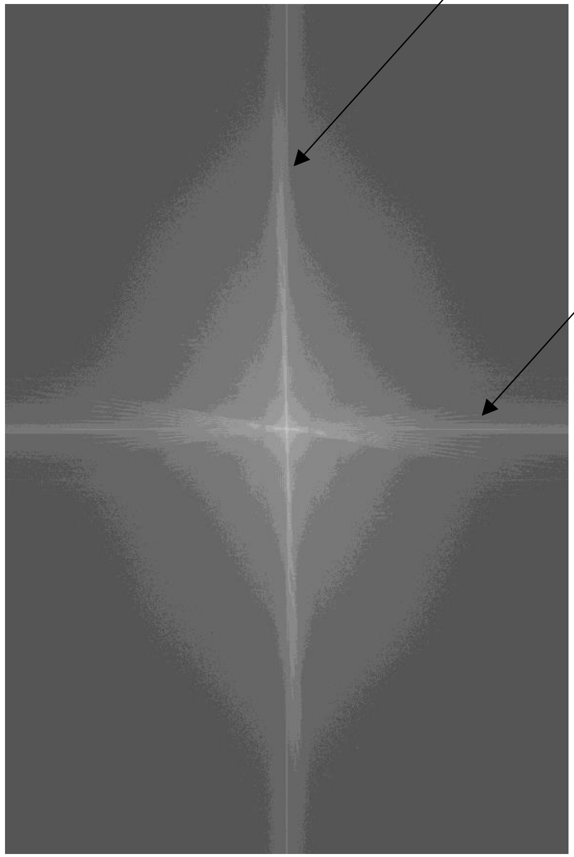
Interpreting DFTs

horizontal (!) lines

vertical (!) lines



image

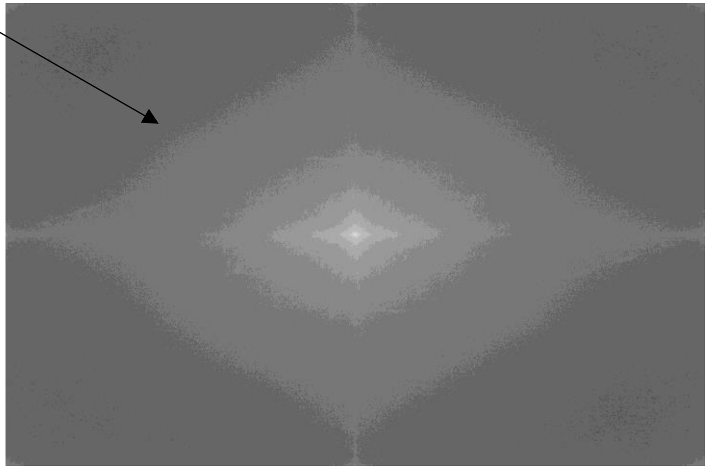


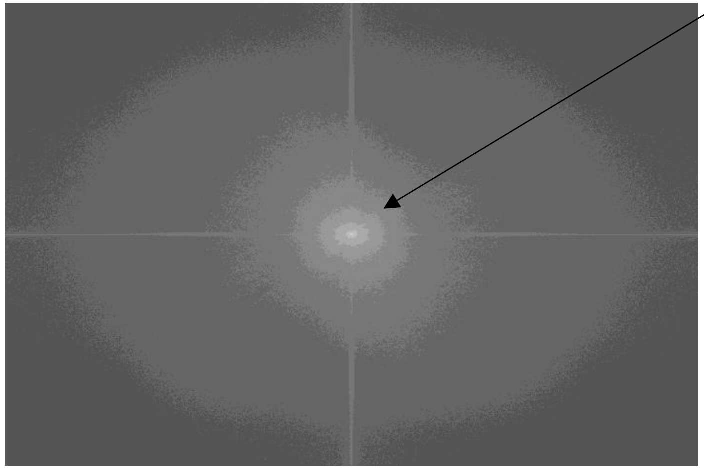
magnitude

Interpreting DFTs

Away from the centre:
high-frequency details

Close to the centre:
low frequencies





Convolution theorem

Convolution in the spatial domain translates to **multiplication** in the frequency domain (and vice versa)

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

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\}\,\mathcal{F}\{g\}$$

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

$$\mathcal{F}^{-1}\{FG\} = \mathcal{F}^{-1}\{F\} * \mathcal{F}^{-1}\{G\}$$

2D convolution theorem example

Image

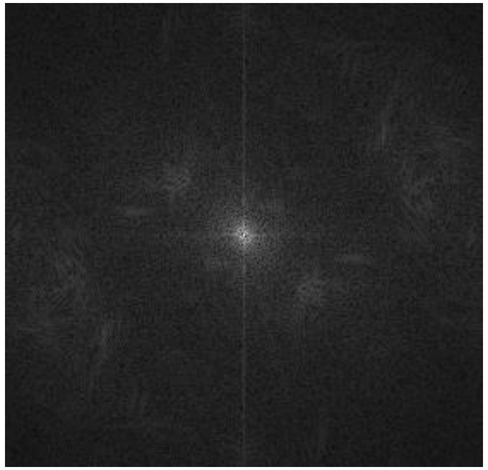
$*$

Filter

$=$

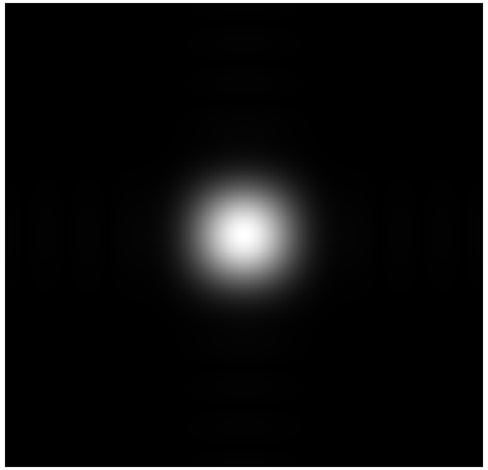
Filtered image

FT(Image)



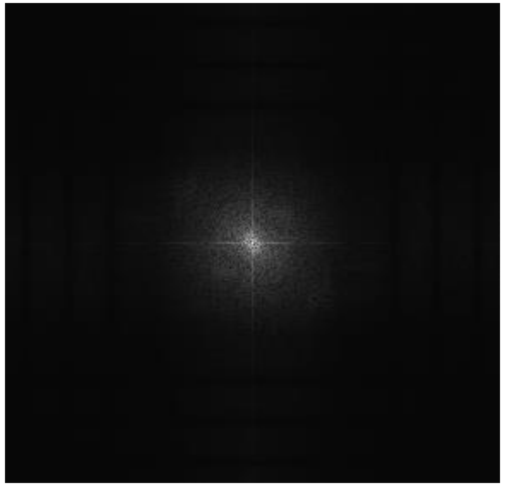
$\times$

FT(Filter)



$=$

FT(Filtered image)



Slide: A Zisserman

Convolution theorem

Suppose f and g both consist of N pixels

- What is the complexity of computing $f * g$ in the spatial domain?

$$O(N^2)$$

- And what is the complexity of computing $\mathcal{F}^{-1}\{\mathcal{F}\{f\}\mathcal{F}\{g\}\}$?

$O(N \log N)$ using FFT

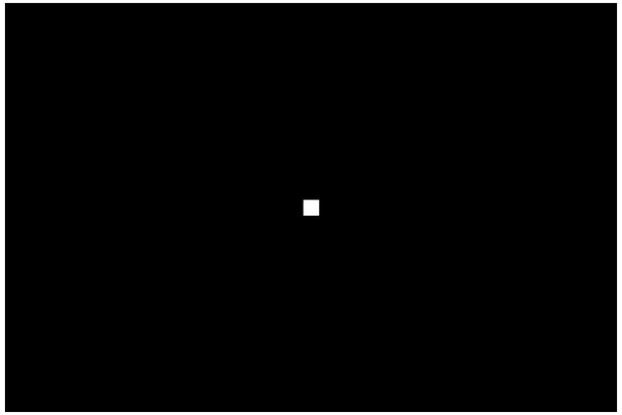
- Thus, convolution of an image with a large filter can be more efficiently done in the frequency domain.
- E.g., CUDA automatically chooses between the two options!

Box vs. Average Filter

Image

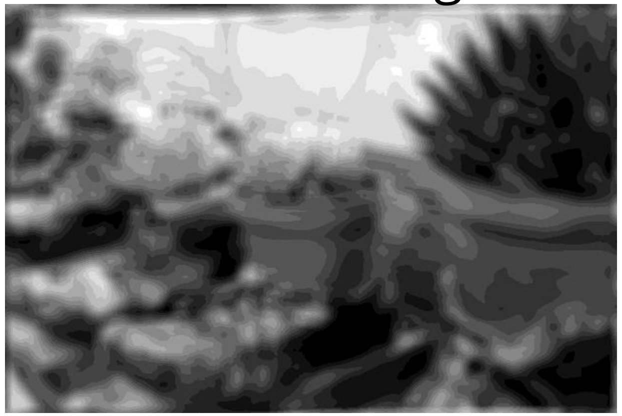
*

Filter

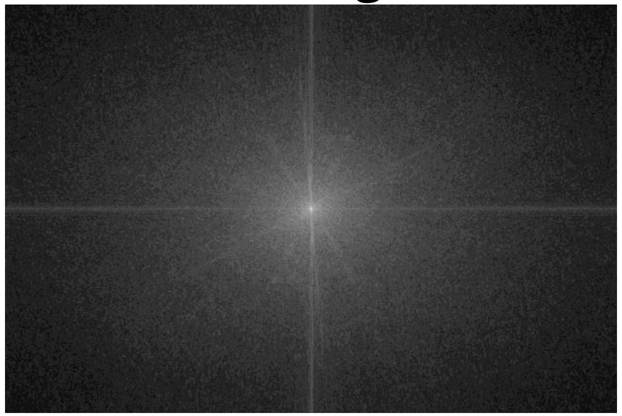


=

Filtered image

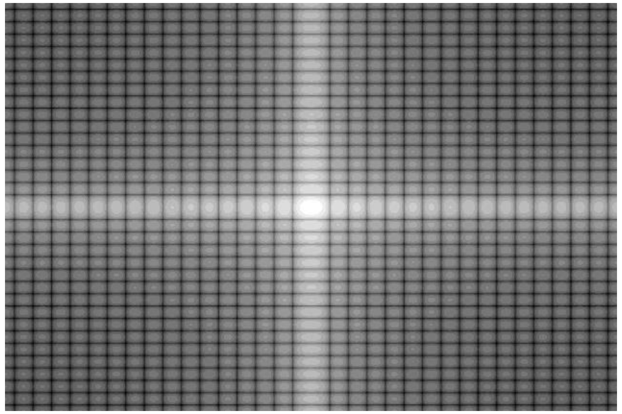


FT(Image)



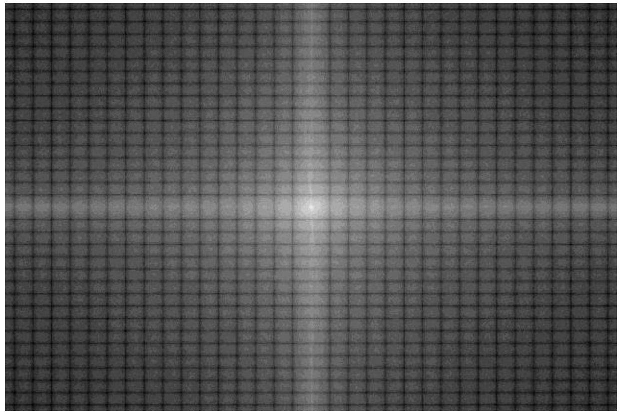
×

FT(Filter)



=

FT(Filtered image)



Box vs. Average Filter

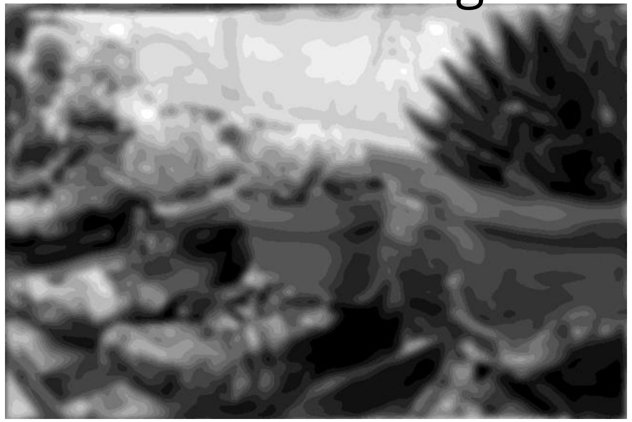
Image

$*$

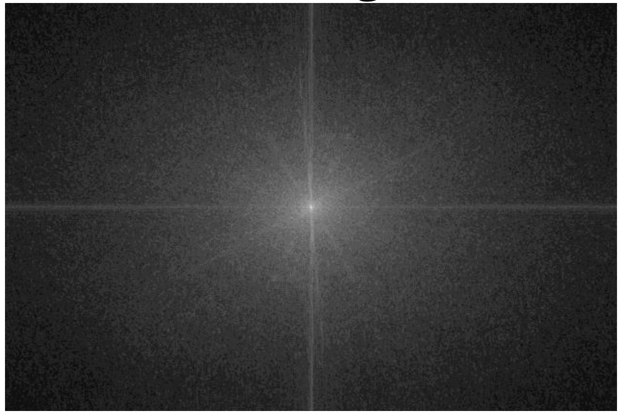
Filter

$=$

Filtered image

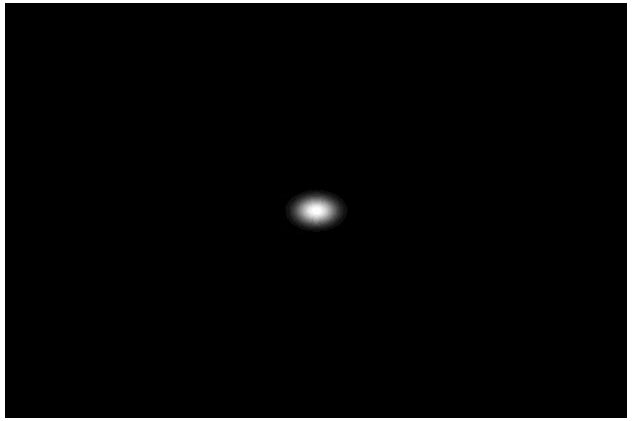


FT(Image)



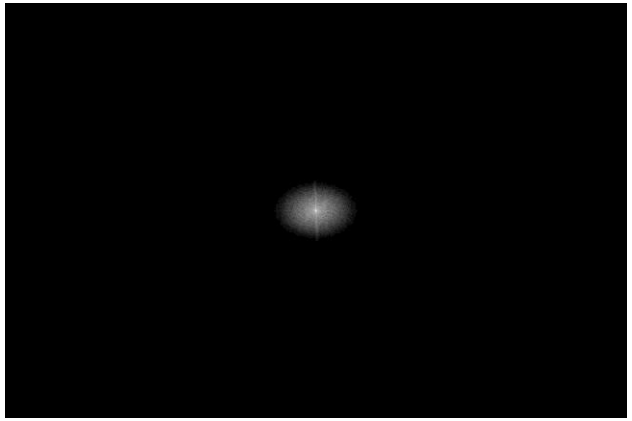
$\times$

FT(Filter)



$=$

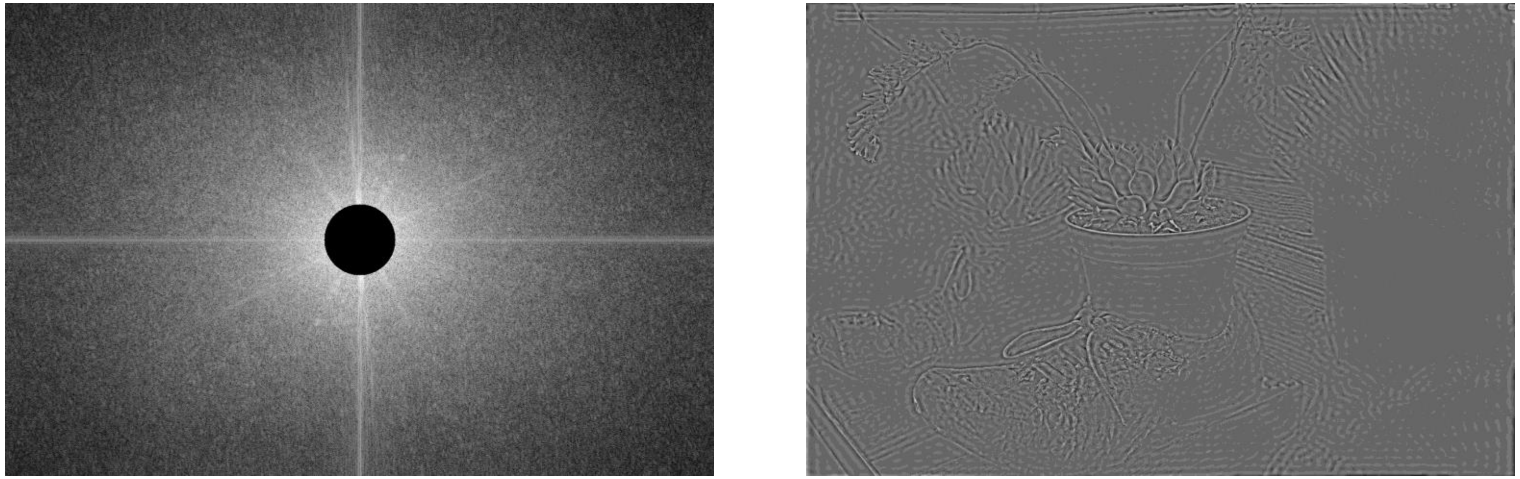
FT(Filtered image)



Low/High-Pass Filter

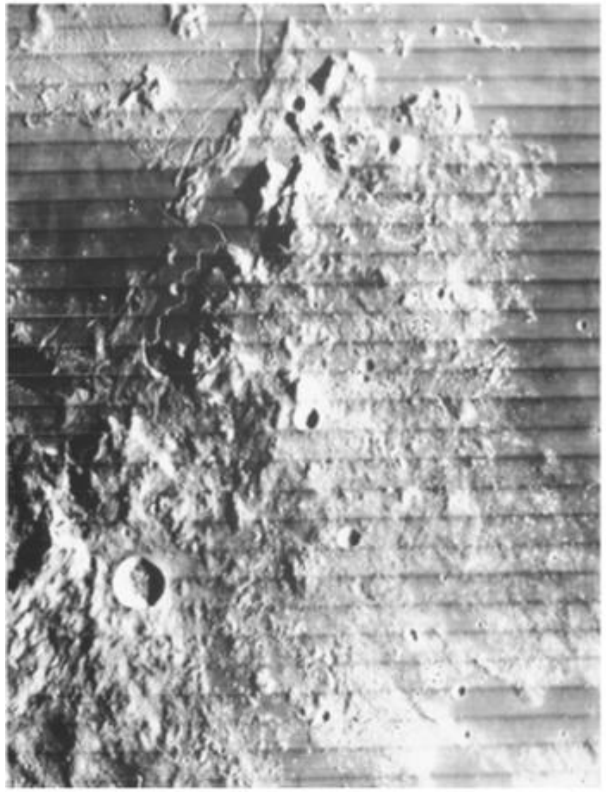
Low-pass filter: remove high frequencies

High-pass filter: remove low frequencies

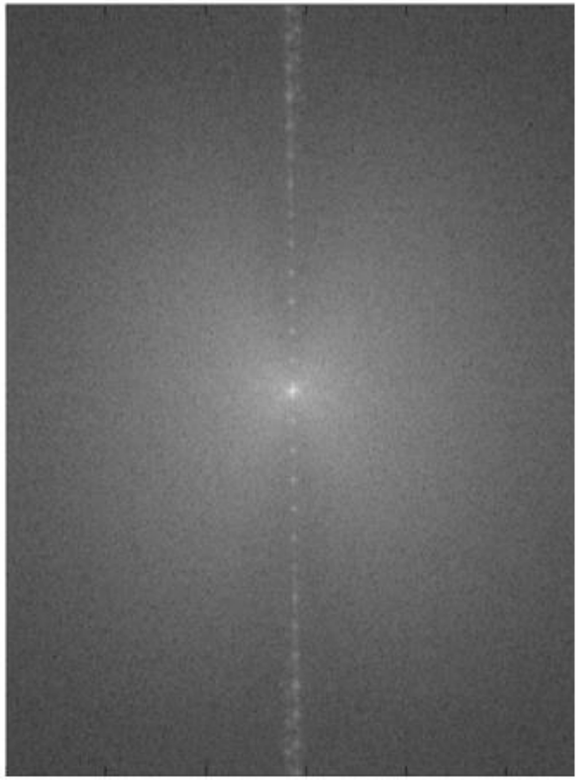


Removing periodic patterns

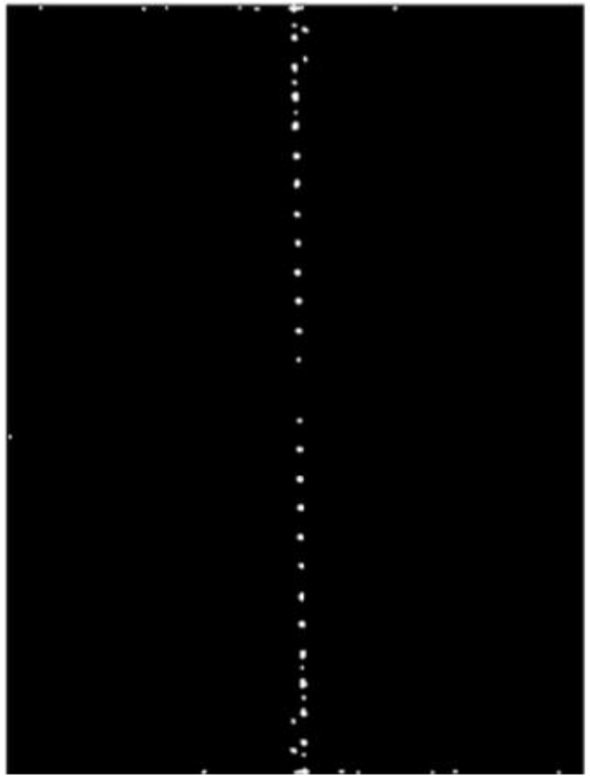
Lunar orbital image (1966)



Magnitude image



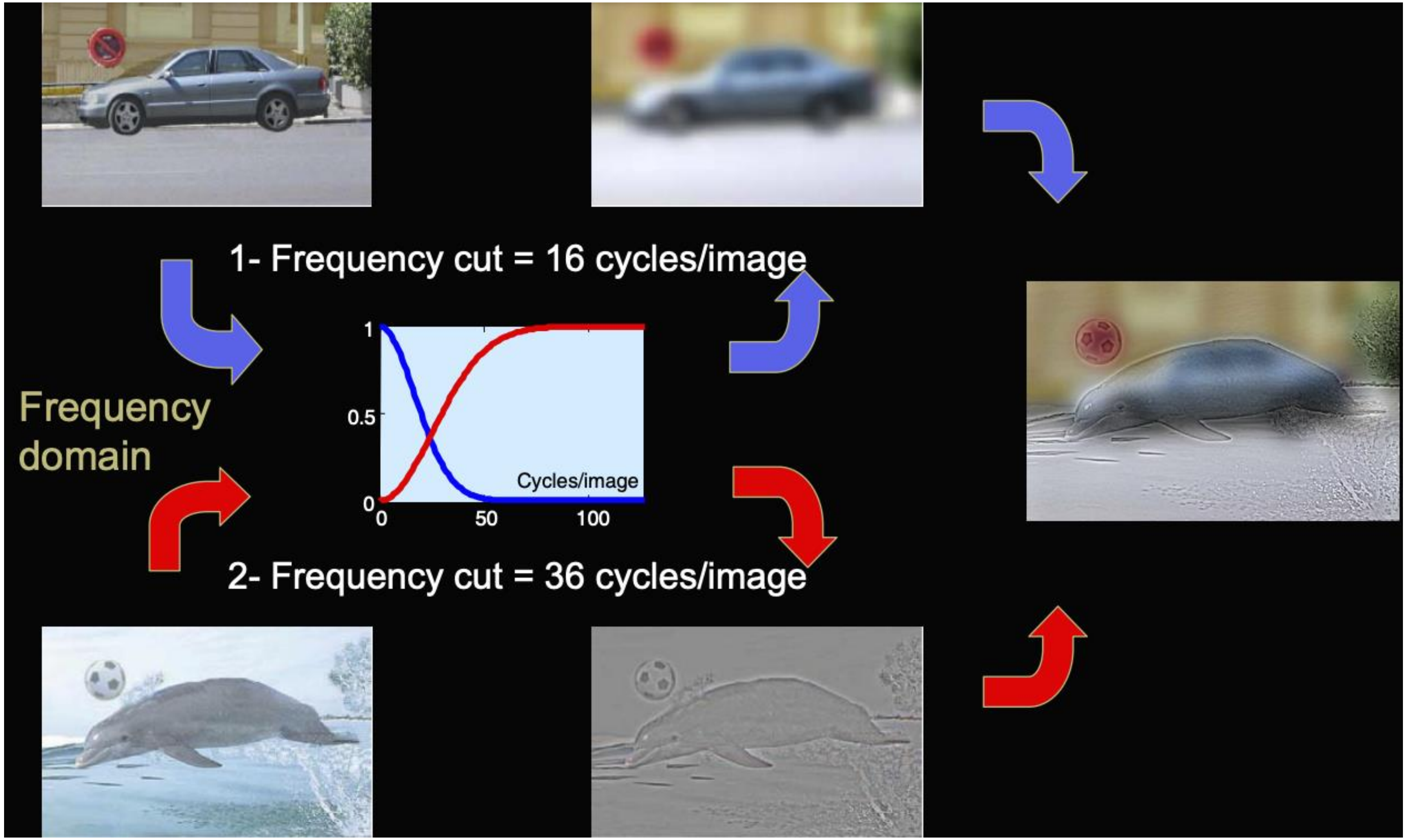
Remove peaks



Join lines removed

Source: [A. Zisserman]

Hybrid images in the frequency domain



Source