Recitation 3

2D Fourier Transform

Agenda

1. Reminder
2. 2D Fourier Transform
3. Derivatives
4. Filtering

Reminder

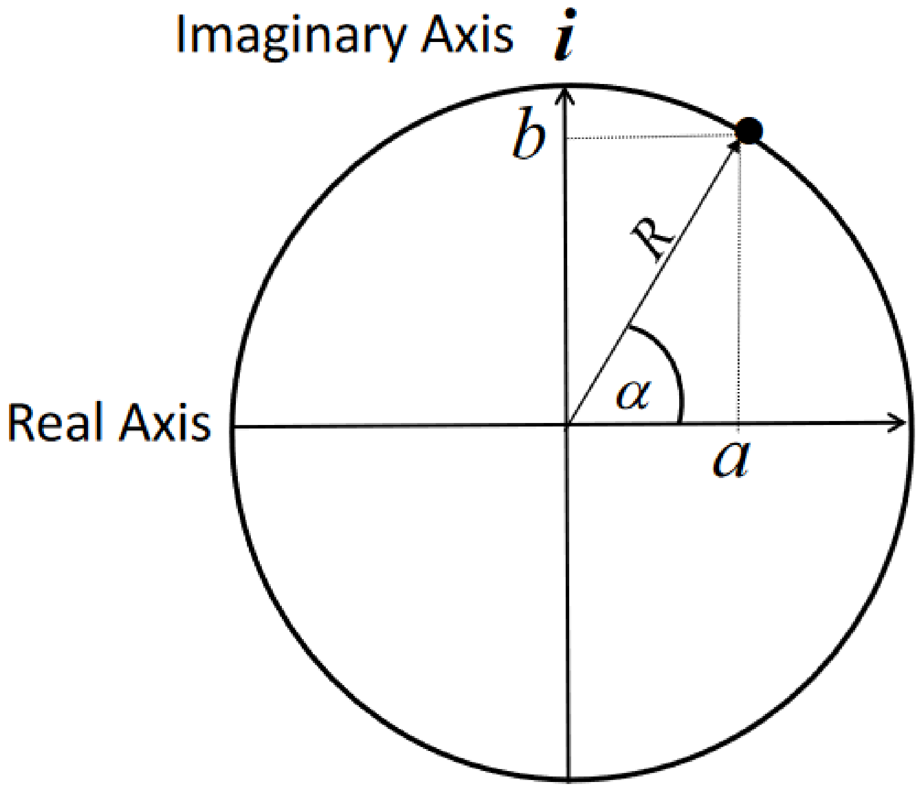
2D Fourier Transform

Derivatives

Filtering

Complex Numbers

$$i^2 = -1$$



$$c = a + bi = R \cdot e^{i\alpha}$$

$$e^{i\alpha} = \cos(\alpha) + i \cdot \sin(\alpha)$$

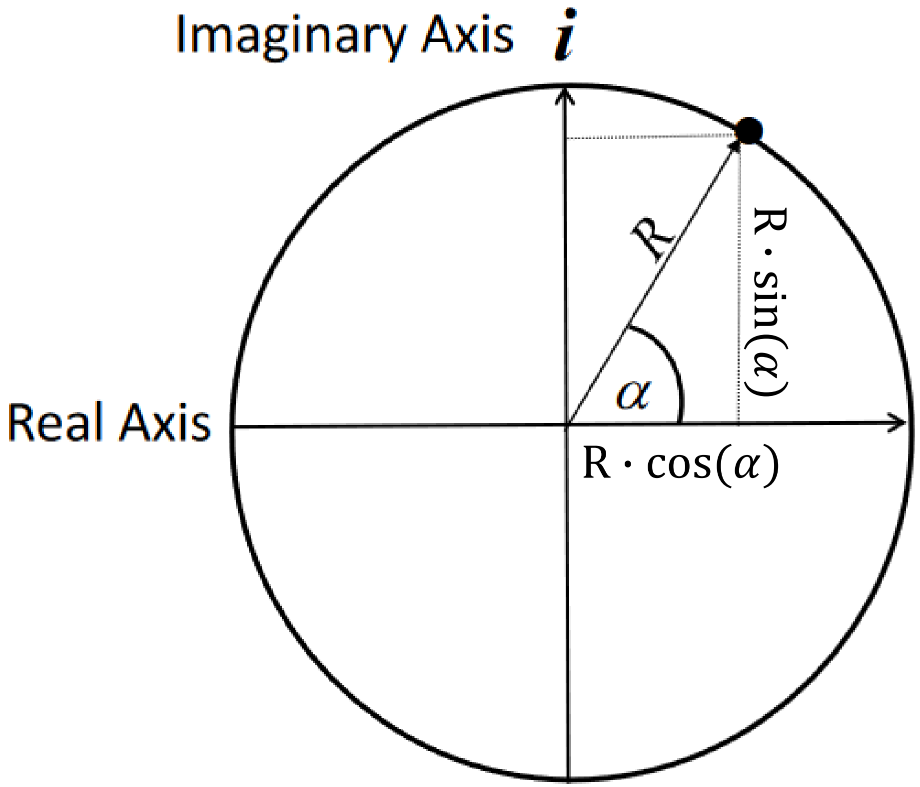
Absolute Value: $|c| = R = \sqrt{a^2 + b^2}$

Phase: $\alpha = \tan^{-1}\left(\frac{b}{a}\right)$

$$R_1 e^{i\alpha_1} \cdot R_2 e^{i\alpha_2} = R_1 R_2 e^{i(\alpha_1 + \alpha_2)}$$

Conjugate: $\bar{c} = c^* = a - bi$

Complex Numbers



$$(R, \alpha)$$

$$(R \cdot cos(\alpha), R \cdot sin(\alpha))$$

$$R \cdot \cos(\alpha) + i \cdot R \cdot sin(\alpha)$$
$$= R \cdot (\cos(\alpha) + \sin(\alpha))$$
$$= R \cdot e^{\alpha}$$

1D Fourier Transform

- Moving from the ***time*** domain to the ***frequency*** domain
- Discrete Fourier transform (DFT):

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\frac{2\pi i x \omega}{N}}$$

- Inverse Discrete Fourier transform (IDFT):

$$f(x) = \frac{1}{N} \sum_{\omega=0}^{N-1} F(\omega)\, e^{\frac{2\pi i x \omega}{N}}$$

In a Nutshell

- We want to decompose a function $f(x)$ to a set of sin and cos functions of different frequencies –

$$f(x) = \sum_{\omega} a_\omega \cos\left(\frac{2\pi\omega x}{N}\right) + b_\omega \sin\left(\frac{2\pi\omega x}{N}\right)$$

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\frac{2\pi i x\omega}{N}}$$

$$= \overset{a_\omega}{\boxed{\sum_{x=0}^{N-1} f(x)\cos\left(\frac{2\pi x\omega}{N}\right)}} - i\, \overset{b_\omega}{\boxed{\sum_{x=0}^{N-1} f(x)\sin\left(\frac{2\pi x\omega}{N}\right)}}$$

1D Fourier Transform

Any function (signal) *f* (*x*) can be decomposed into F - a set of *sin* and *cos* periodic functions of different frequencies.
f can be reconstructed from *F* without any loss of data!

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\frac{2\pi i x \omega}{N}}$$

1D DFT: Examples

$$1 \xrightarrow{Fourier} \delta(u)$$

$$\sin(ax) \xrightarrow{Fourier} \frac{1}{2}\left(\delta(u-\frac{a}{2\pi})+\delta(u+\frac{a}{2\pi})\right)$$

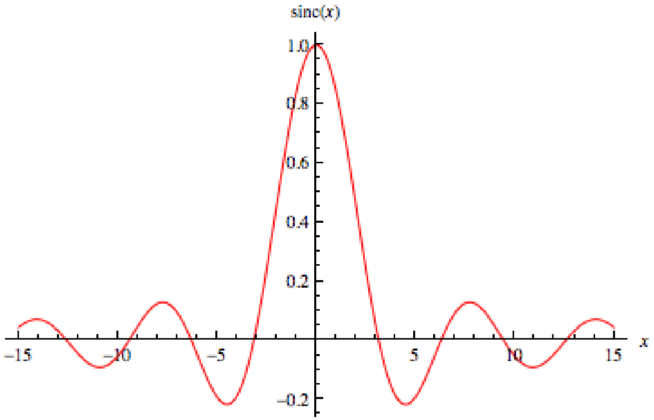
$$rect(ax) \xrightarrow{Fourier} \frac{1}{|a|} sinc(\frac{u}{a})$$

1D DFT: Examples

$$1 \xrightarrow[Fourier]{} \delta(u)$$

$$\sin(ax) \xrightarrow[Fourier]{} \frac{1}{2}\left(\delta(u-\frac{a}{2\pi})+\delta(u+\frac{a}{2\pi})\right)$$

$$rect(ax) \xrightarrow[Fourier]{}$$



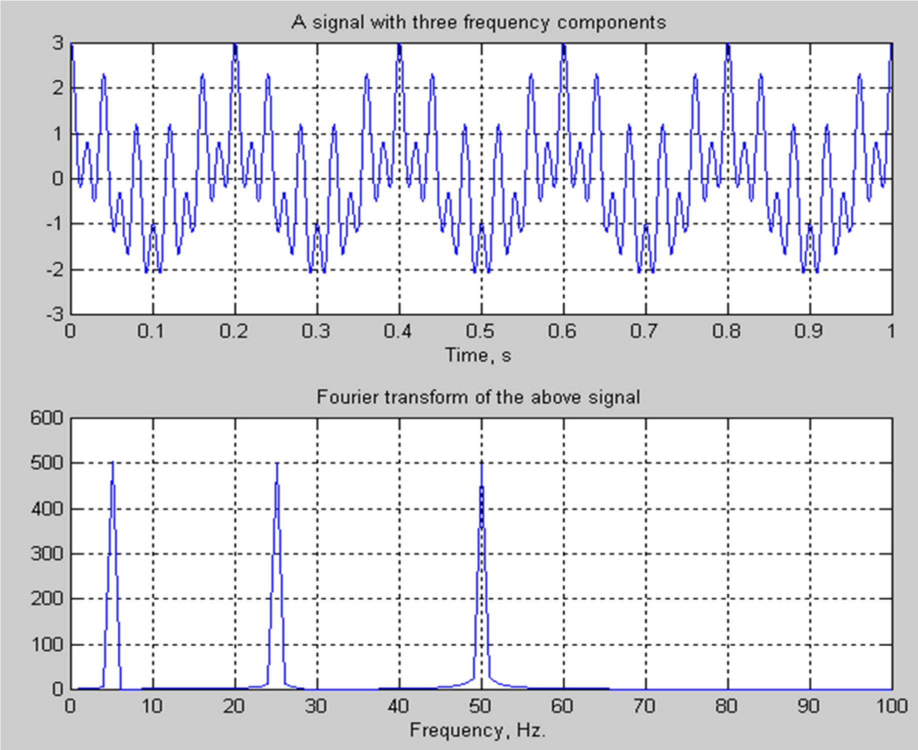
Stationary vs. Non-Stationary

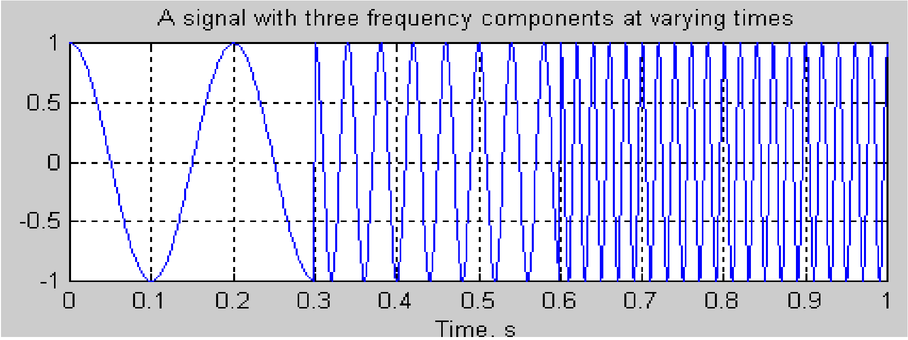
Stationary:

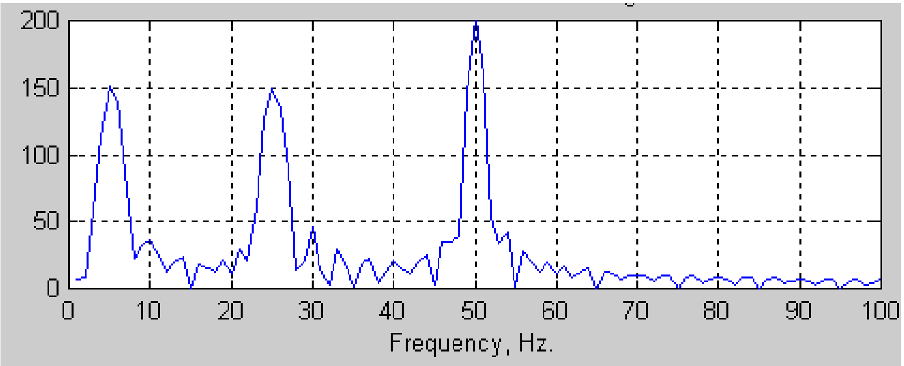
$\cos(2\pi \cdot 5t) + \cos(2\pi \cdot 25t) + \cos(2\pi \cdot 50t)$

Non-stationary

$\cos(2\pi \cdot 5t)$ ***then*** $\cos(2\pi \cdot 25t)$ ***then*** $\cos(2\pi \cdot 50t)$







STFT

DFT: $F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\frac{2\pi i x \omega}{N}}$

IDFT: $f(x) = \frac{1}{N} \sum_{\omega=0}^{N-1} F(\omega)\, e^{\frac{2\pi i x \omega}{N}}$

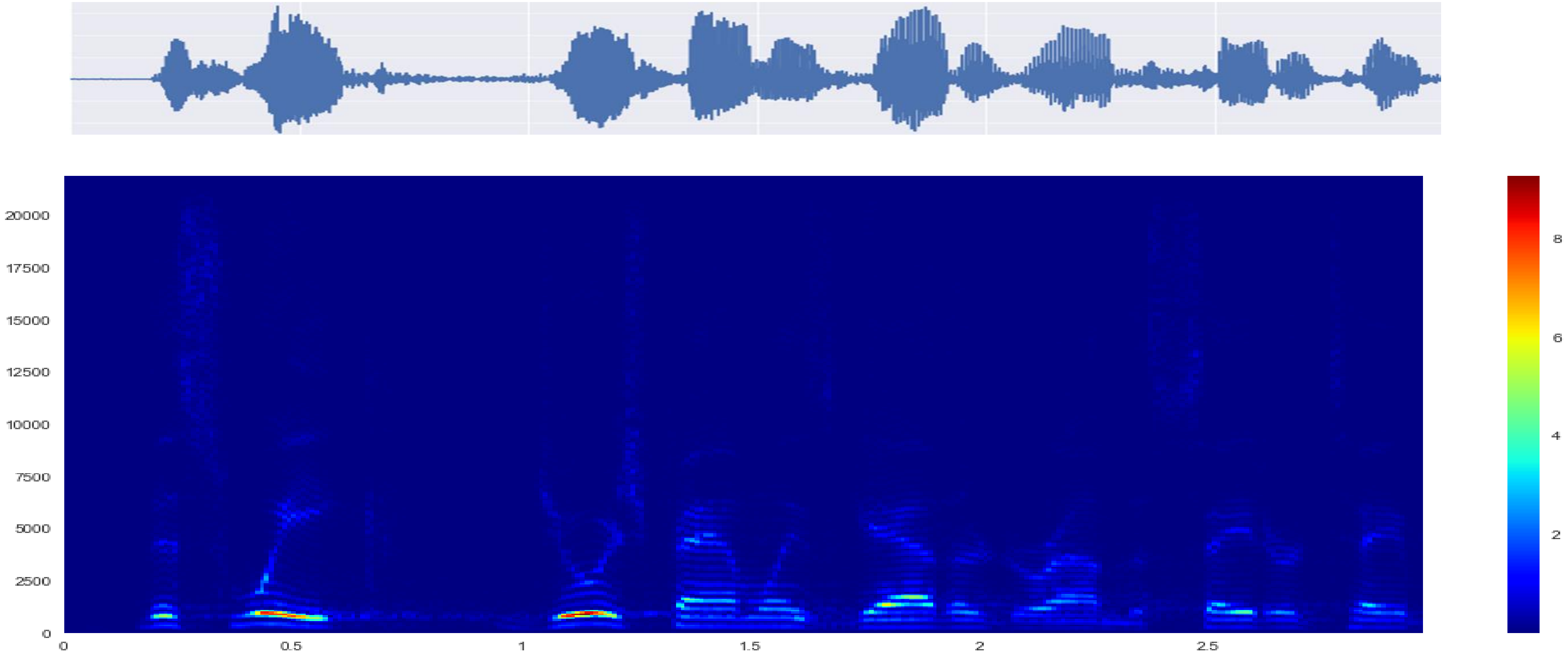
STFT for time t:

$$F(\omega, t) = \sum_{x=-\infty}^{\infty} f(x)\, W(t-x) e^{-\frac{2\pi i x \omega}{N}}$$

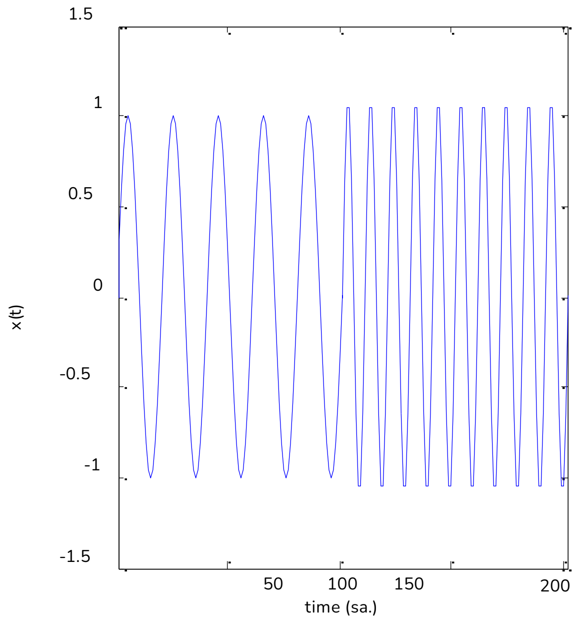
ISTFT: (K is a normalization constant)

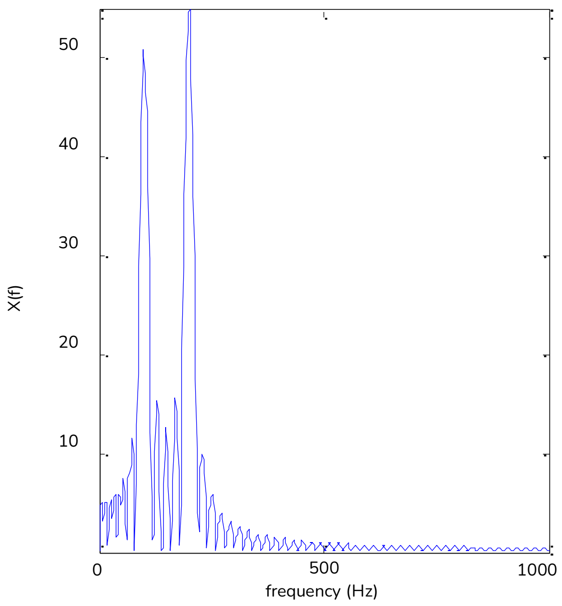
$$f(x) = K \sum_{p=-\infty}^{\infty} \sum_{u=0}^{N-1} F(u, pL) e^{\frac{2\pi i u x}{N}}$$

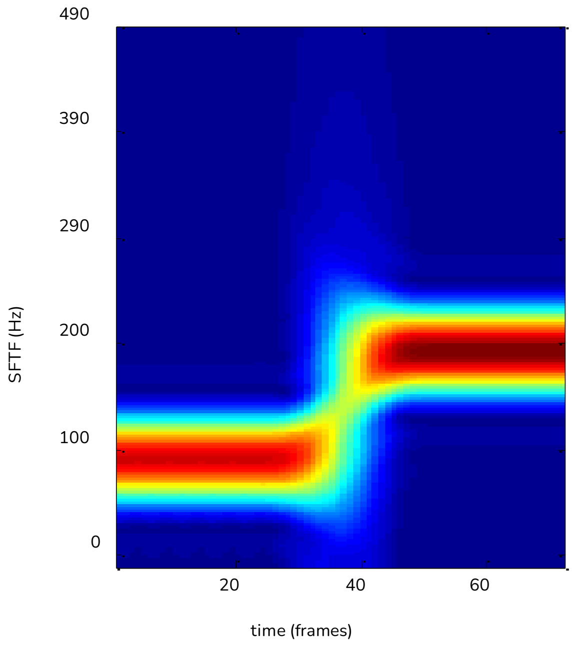
Spectrograms



Spectrograms



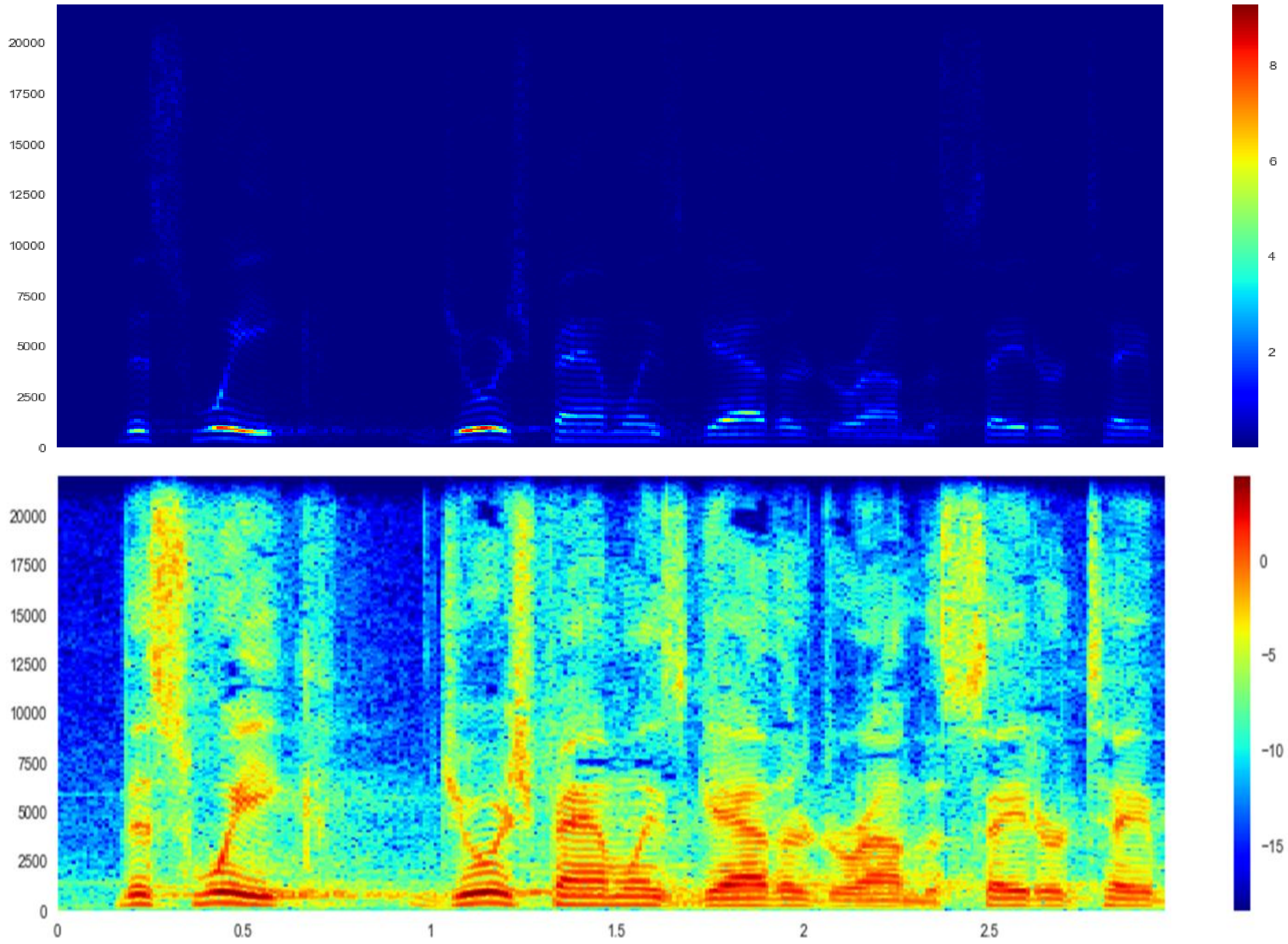




Spectrograms – Log Compression

Original

Log
compressed



Reminder

2D Fourier Transform

Derivatives

Filtering

2D Discrete Fourier Transform

2D Fourier Transform:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-\frac{2\pi i(ux+vy)}{N}}$$

2D Inverse Fourier Transform:

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{\frac{2\pi i(ux+vy)}{N}}$$

2D Fourier Basis

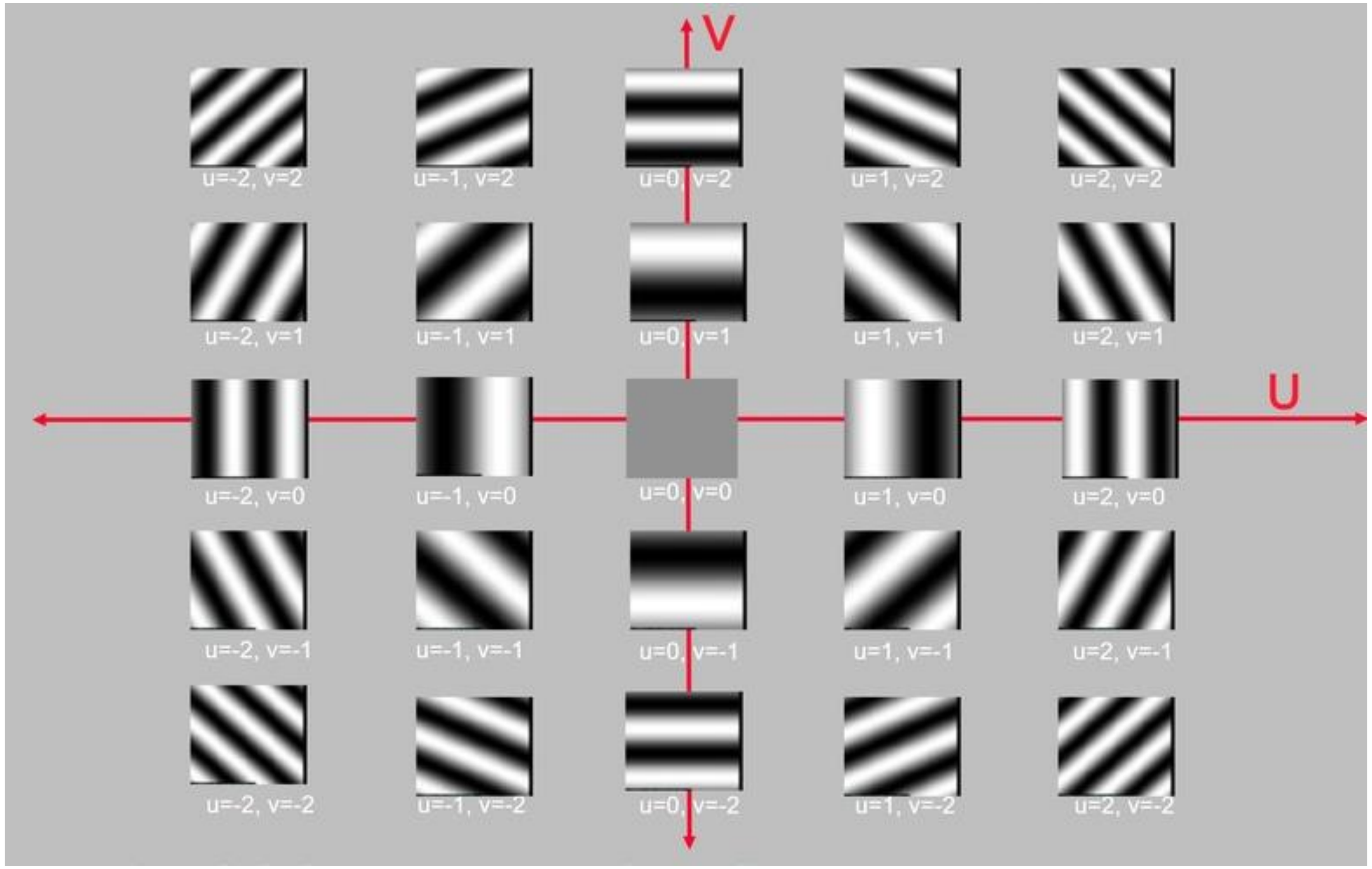
$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\,\boxed{e^{-\frac{2\pi i(ux+vy)}{N}}}$$

Basis

$$e^{-\frac{2\pi i(ux+vy)}{N}} = \cos\left(\frac{2\pi i(ux+vy)}{N}\right) - i\cdot\sin\left(\frac{2\pi i(ux+vy)}{N}\right)$$

What does the basis look like?

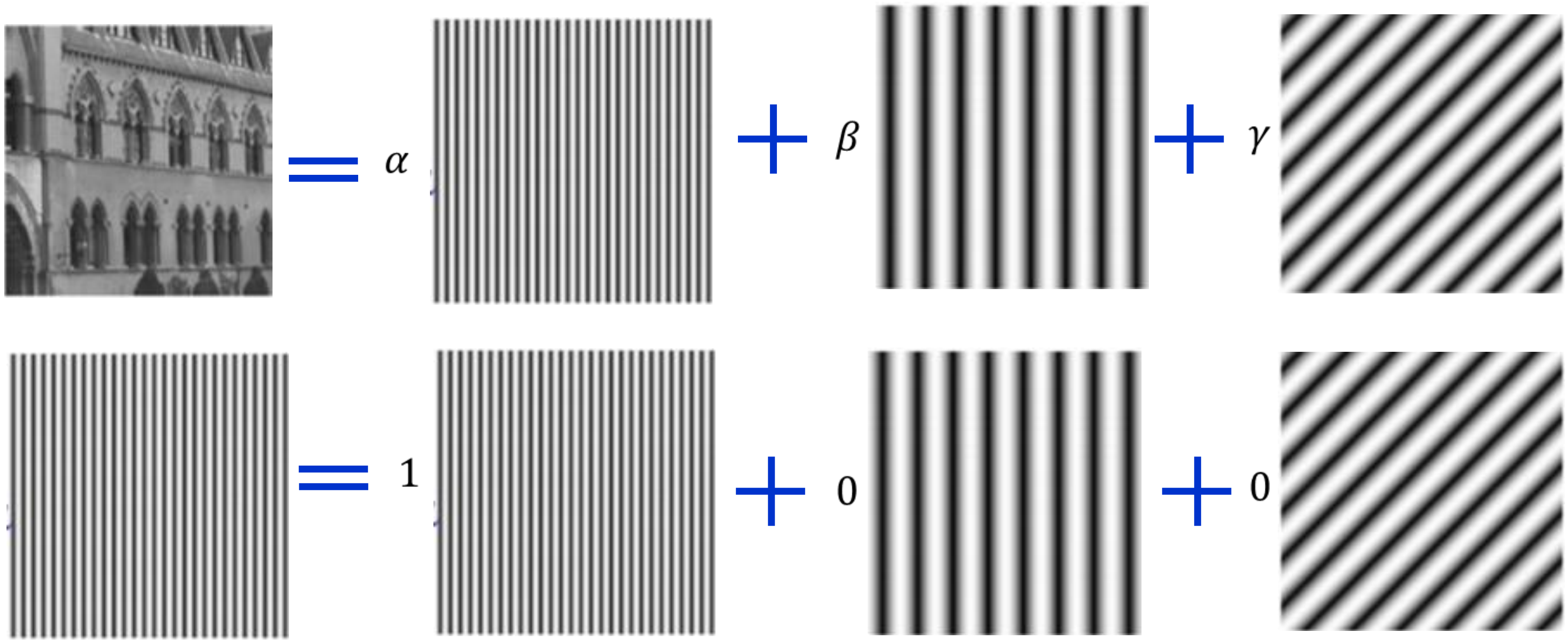
2D Fourier Basis



Desmos

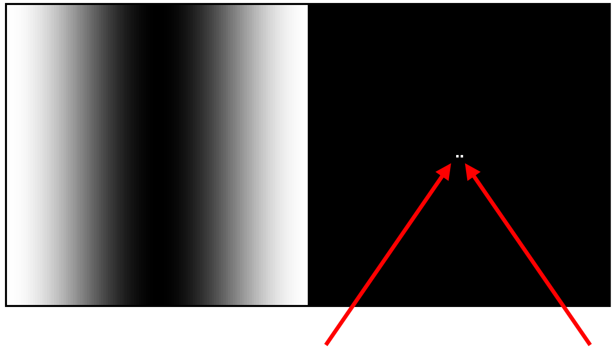
$$black = -1; white = 1; grey = 0$$

2D Discrete Fourier Transform



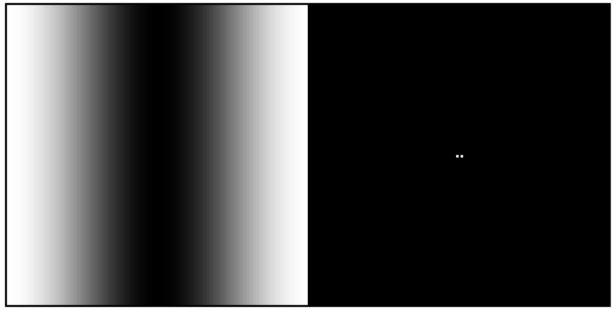
2D DFT – Simple Examples

Low frequency

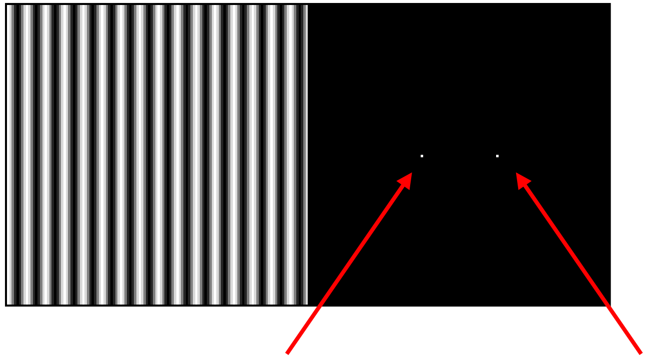


2D DFT – Simple Examples

Low frequency

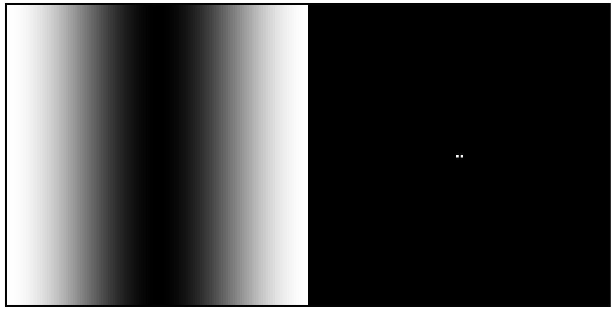


Medium frequency

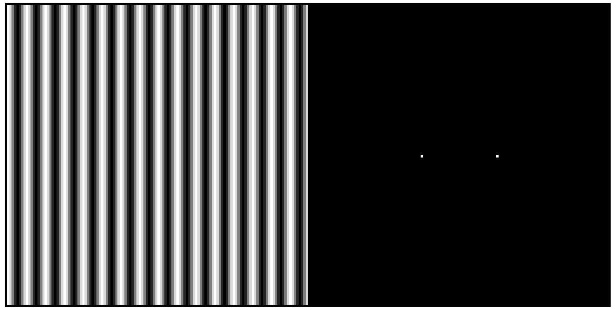


2D DFT – Simple Examples

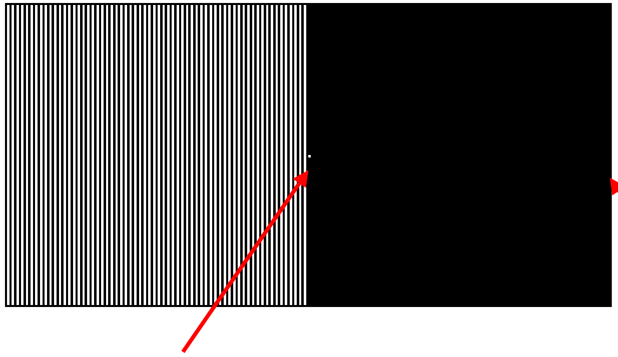
Low frequency



Medium frequency

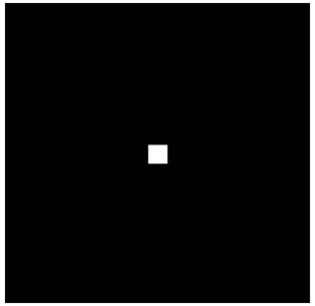


High frequency

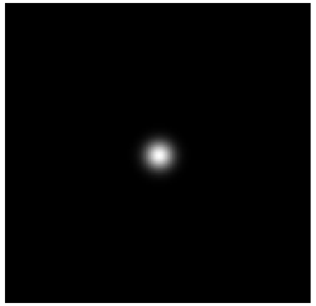


2D DFT – Simple Examples

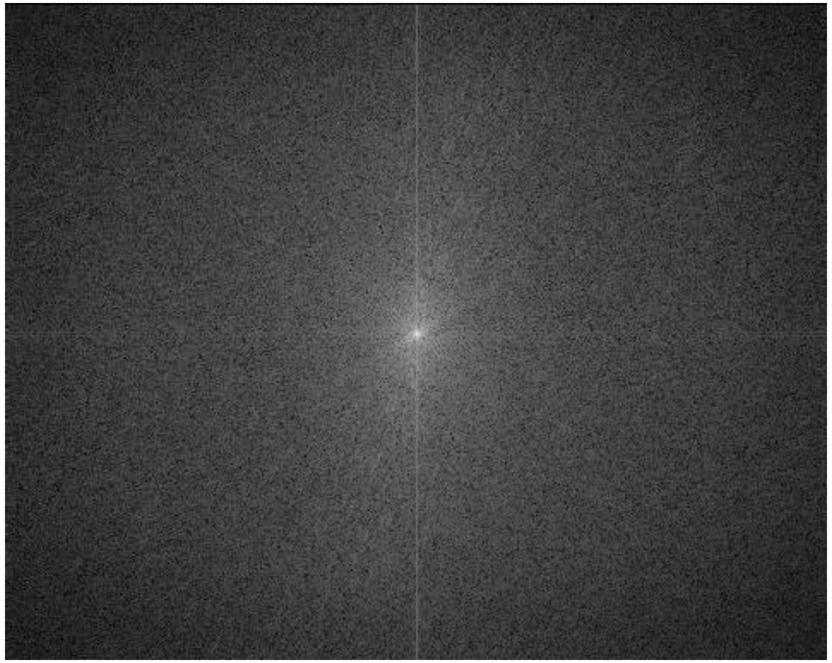
2D rect



Gaussian

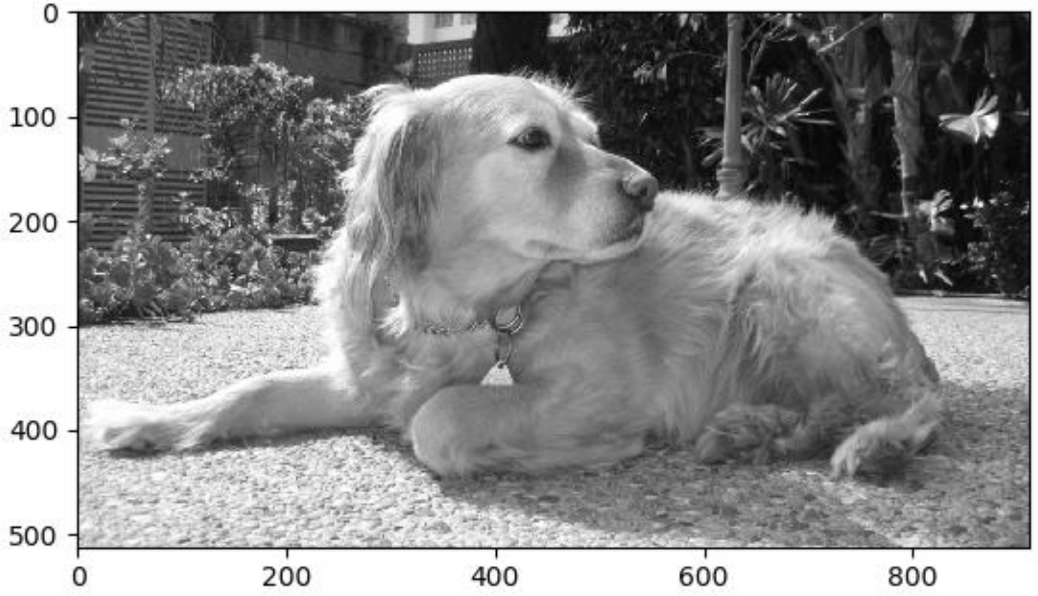


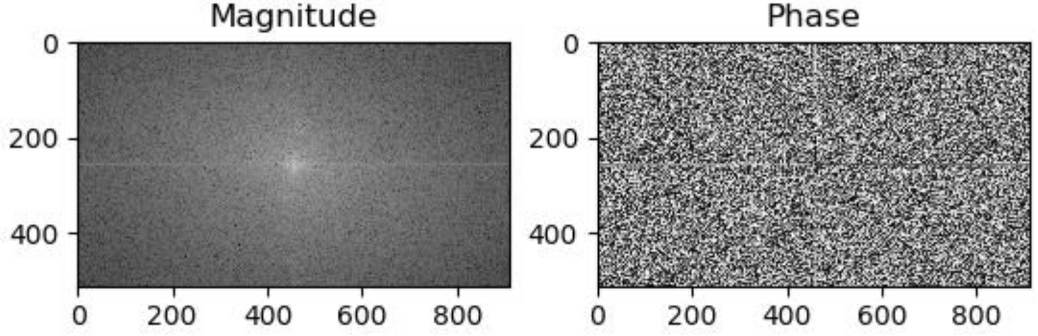
Real Example



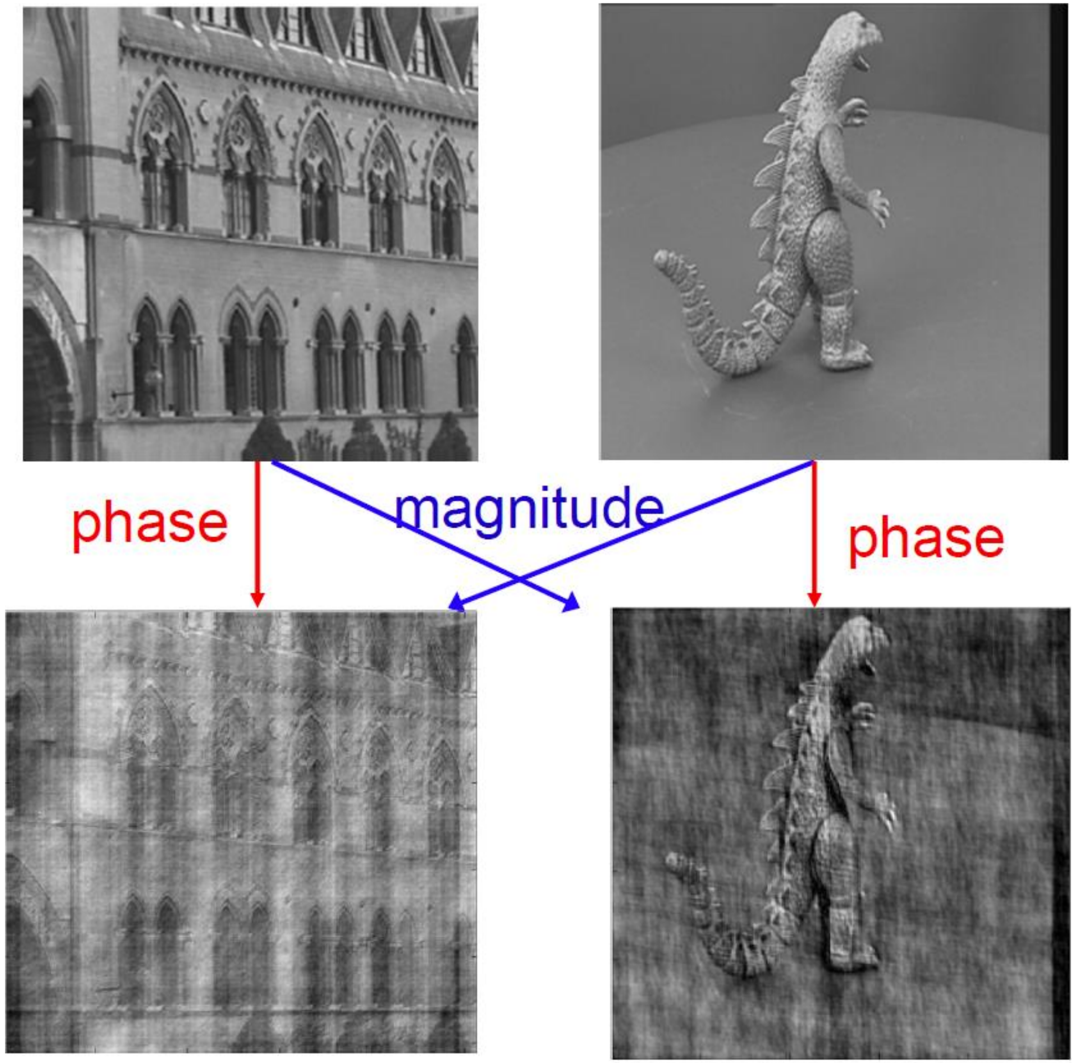
Visualization

- FT produces complex numbers
- Can Be presented as two images in two ways:
  - Real and imaginary part
  - Magnitude and phase
    - Magnitude: $\sqrt{R(u)^2 + I(u)^2}$
    - Phase: $\tan^{-1}(\frac{I(u)}{R(u)})$
- In image processing we use only the magnitude
  - Contains most relevant information


0
100
200
300
400
500
0
200
400
600
800


Magnitude
Phase
0
200
400
0
200
400
600
800

Importance of Phase



Desmos

Computing the 2D Fourier Transform

Repeat the 1D Fourier twice:

- Compute the 1D Fourier for each **row**
- On the result, compute the 1D Fourier for each **column**
- (Multiply by N , application dependent)

- The 1D Fourier transform is sufficient for computing any multi-dimensional Fourier transform.

Decomposing 2D DFT to 1D DFT

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

Decomposing 2D DFT to 1D DFT

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

$$= \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-\frac{2\pi i\cdot ux}{N}} \cdot e^{-\frac{2\pi i\cdot vy}{N}}$$

Decomposing 2D DFT to 1D DFT

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

$$=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)e^{-\frac{2\pi i\cdot ux}{N}}\cdot e^{-\frac{2\pi i\cdot vy}{N}}$$

$$=\frac{1}{N}\sum_{x=0}^{N-1}e^{-\frac{2\pi i\cdot ux}{N}}\sum_{y=0}^{N-1}f(x,y)\cdot e^{-\frac{2\pi i\cdot vy}{N}}$$

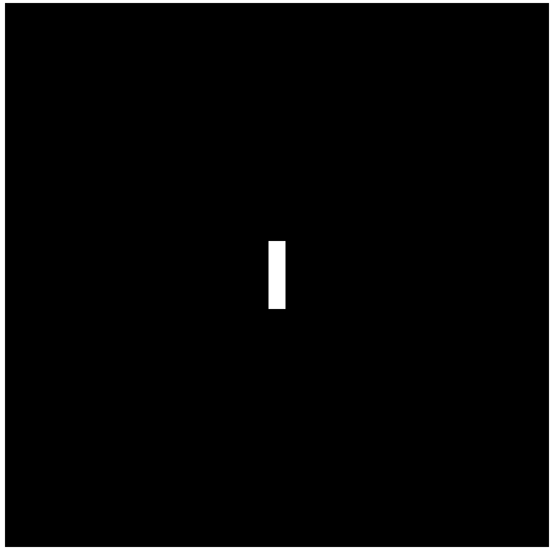
Decomposing 2D DFT to 1D DFT

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

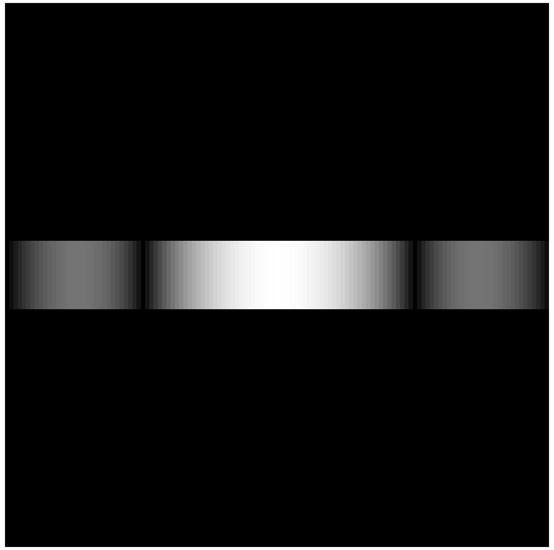
$$= \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-\frac{2\pi i\cdot ux}{N}} \cdot e^{-\frac{2\pi i\cdot vy}{N}}$$

$$= \frac{1}{N}\sum_{x=0}^{N-1} e^{-\frac{2\pi i\cdot ux}{N}} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-\frac{2\pi i\cdot vy}{N}} = \frac{1}{N}\sum_{x=0}^{N-1} e^{-\frac{2\pi i\cdot ux}{N}} \cdot F(x,v)$$

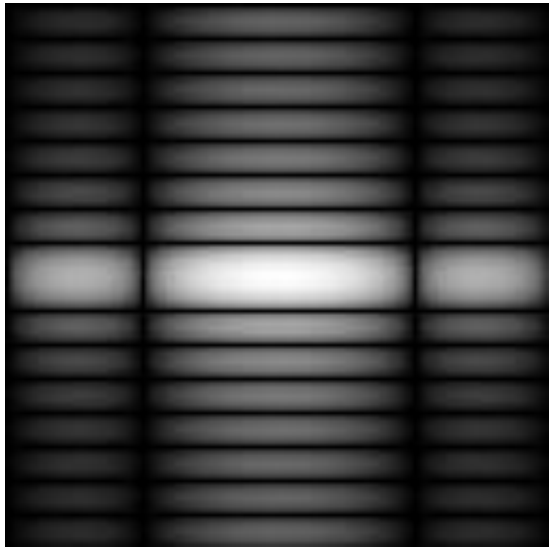
Decomposition Example



(1) Spatial Domain

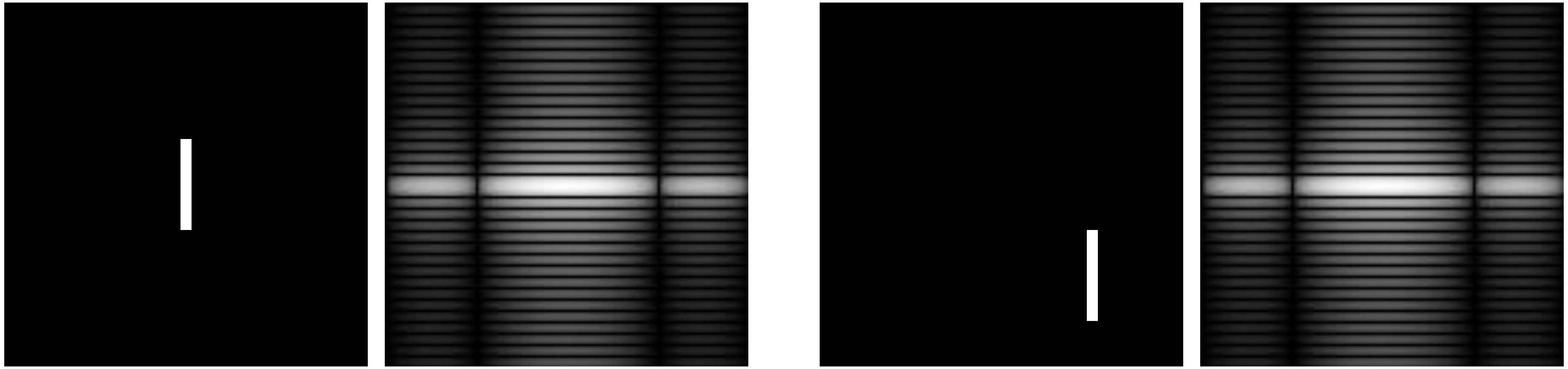


(2) After 1D-DFT on each row



(3) After 1D-DFT on each column

Image Translation



$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)\cdot e^{\frac{2\pi i(ux_0+vy_0)}{N}}$$

$$F(u-u_0, v-v_0) \Leftrightarrow f(x,y)\cdot e^{-\frac{2\pi i(u_0x+v_0y)}{N}}$$

Fourier is Non-Local!

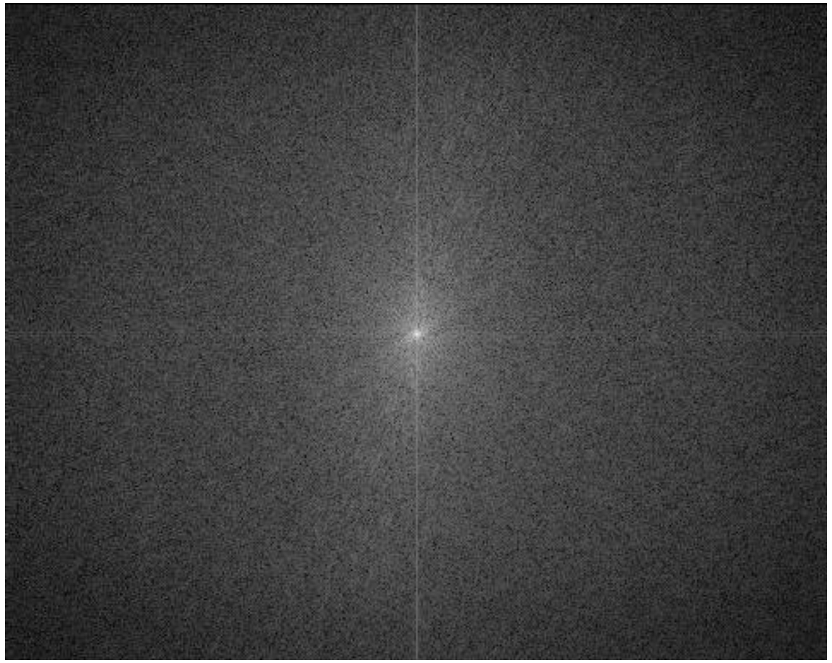
Fourier Transform supplies a **global representation** of the image in the frequency domain.

Local objects / features in the image **cannot be assigned** to specific frequencies!

In general:

- **Low frequencies** represent the coarse structure of the image (large homogenous parts like walls, sky, etc.)
- **High frequencies** represent the fine details in the image (fine texture, wrinkles, noise, etc.)

Real Example



Reminder

2D Fourier Transform

Derivatives

Filtering

Image Derivatives

Inverse DFT:

$$f(x, y) = \frac{1}{N} \sum_{u,v} F(u, v) \cdot e^{\frac{2\pi i (ux+vy)}{N}}$$

Image Derivatives

Inverse DFT:

$$f(x,y) = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Derive by x:

$$\frac{\partial f}{\partial x} = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot \boxed{\frac{2\pi i u}{N}}$$

Image Derivatives

Inverse DFT:

$$f(x,y) = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Derive by x:

$$\frac{\partial f}{\partial x} = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot \frac{2\pi i u}{N}$$

$$= \frac{2\pi i}{N} \cdot \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot u$$

Image Derivatives

Inverse DFT:

$$f(x,y) = \frac{1}{N}\sum_{u,v} F(u,v)\cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Derive by x:

$$\frac{\partial f}{\partial x} = \frac{1}{N}\sum_{u,v} F(u,v)\cdot e^{\frac{2\pi i(ux+vy)}{N}}\cdot\frac{2\pi i u}{N}$$

$$= \frac{2\pi i}{N}\cdot\frac{1}{N}\sum_{u,v} F(u,v)\cdot e^{\frac{2\pi i(ux+vy)}{N}}\cdot u$$

$$= \frac{2\pi i}{N}\cdot\frac{1}{N}\sum_{u,v} u\cdot F(u,v)\cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Image Derivatives

Inverse DFT:

$$f(x,y) = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Derive by x:

$$\frac{\partial f}{\partial x} = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot \frac{2\pi i u}{N}$$

$$= \frac{2\pi i}{N} \cdot \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot u$$

$$= \frac{2\pi i}{N} \cdot \frac{1}{N}\sum_{u,v} u \cdot F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

$\hat{F}(u,v)$

Image Derivatives

Inverse DFT:

$$f(x,y) = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}}$$

Derive by x:

$$\frac{\partial f}{\partial x} = \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot \frac{2\pi i u}{N}$$

$$= \frac{2\pi i}{N} \cdot \frac{1}{N}\sum_{u,v} F(u,v) \cdot e^{\frac{2\pi i(ux+vy)}{N}} \cdot u$$

$$= \frac{2\pi i}{N} \cdot \boxed{\frac{1}{N}\sum_{u,v} \underbrace{u \cdot F(u,v)}_{\hat{F}(u,v)} \cdot e^{\frac{2\pi i(ux+vy)}{N}}} = \text{IDFT of } \hat{F}(u,v)$$

Image derivatives by FT

To compute the ***x* derivative** of f (up to a constant):

- Compute the Fourier transform F
- Multiply each Fourier coefficient $F(u, v)$ by u
- Compute the inverse Fourier transform

To compute the ***y* derivative** of f (up to a constant):

- Compute the Fourier transform F
- Multiply each Fourier coefficient $F(u, v)$ by v ($\hat{F}(u, v)$)
- Compute the inverse Fourier transform

Multiplying by u

To compute the **x derivative** of f (up to a constant):

- Compute the Fourier transform F
- ***Multiply each Fourier coefficient*** $\boldsymbol{F(u, v)}$ ***by*** $\boldsymbol{u}$
- Compute the inverse Fourier transform

Multiplying by u

To compute the **x derivative** of f (up to a constant):

- Compute the Fourier transform F
- ***Multiply each Fourier coefficient $\boldsymbol{F(u,v)}$ by $\boldsymbol{u}$***
- Compute the inverse Fourier transform

$$(0,1,2,\ldots,{}^{N}\!/_{2},\ldots,N-1)$$

Multiplying by u

To compute the **x derivative** of f (up to a constant):

- Compute the Fourier transform F
- ***Multiply each Fourier coefficient $F(u, v)$ by u***
- Compute the inverse Fourier transform

$$\cancel{(0,1,2,\ldots,N/2,\ldots,N-1)}$$

The highest frequency is N/2

Multiplying by u

To compute the ***x* derivative** of f (up to a constant):

- Compute the Fourier transform F
- ***Multiply each Fourier coefficient*** $\boldsymbol{F(u,v)}$ ***by*** $\boldsymbol{u}$
- Compute the inverse Fourier transform

$$\cancel{(0,1,2,\ldots,N/2,\ldots,N-1)}$$

The highest frequency is N/2

(try to use Symmetric + Periodicity)

$$(0,1,\ldots,N/2-1,-N/2,\ldots,-1)$$

Multiplying by u

To compute the ***x* derivative** of f (up to a constant):

- Compute the Fourier transform F
- ***<u>Multiply each Fourier coefficient $\boldsymbol{F(u,v)}$ by $\boldsymbol{u}$</u>***
- Compute the inverse Fourier transform

$$\cancel{(0,1,2,\ldots,{}^{N}\!/_{2},\ldots,N-1)}$$

The highest frequency is N/2

$$(0,1,\ldots,{}^{N}\!/_{2}-1,-{}^{N}\!/_{2},\ldots,-1)$$

Or

$$(-{}^{N}\!/_{2},\ldots,0,\ldots,{}^{N}\!/_{2}-1)$$

In this option: should center Fourier Transform of the image (F) as well

Image derivatives

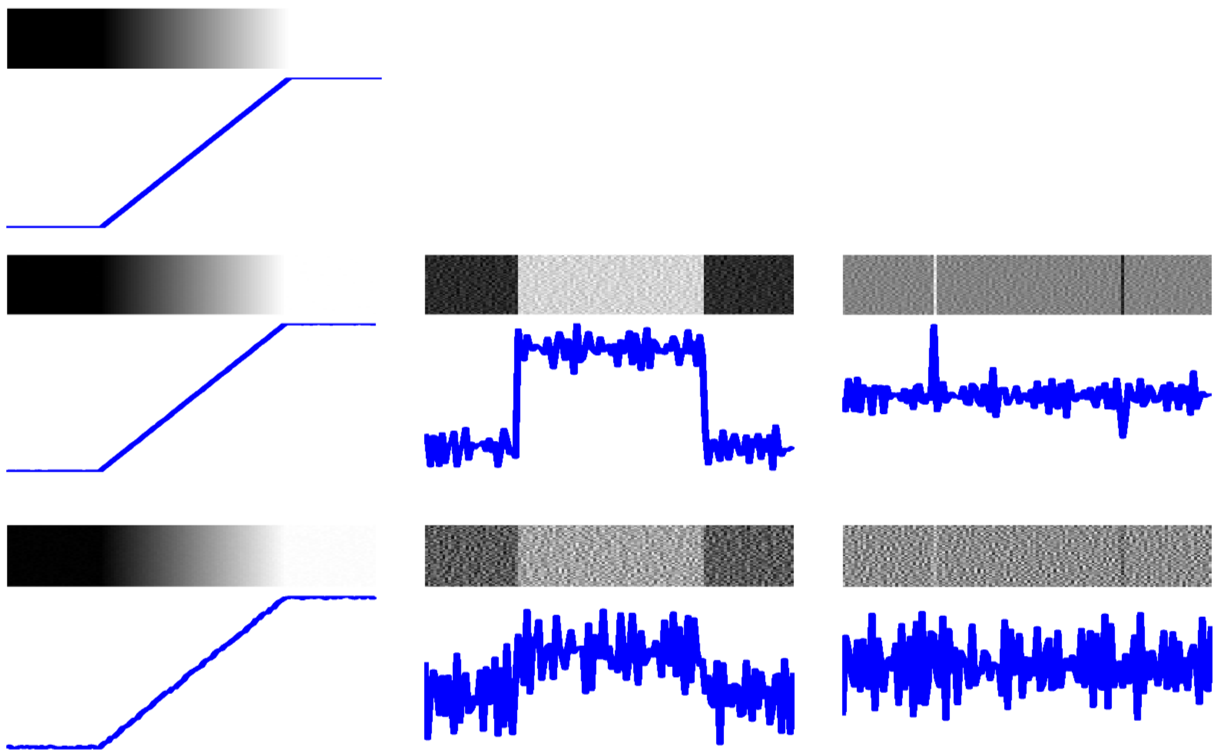
Image derivative is the inverse FT of the **weighted** frequency domain.

High frequencies affect the image derivative more than low frequencies.

Noise has more high frequency than normal image.

$$\frac{\partial f(x,y)}{\partial x} = \frac{2\pi i}{N} \cdot \Phi^{-1}\left(u \cdot \Phi\left(f(x,y)\right)\right)$$

Effect of Noise on Derivatives



Reminder

2D Fourier Transform

Derivatives

Filtering

Filtering in the frequency domain: General Scheme

Input image $f(x,y)$

$\downarrow$

Fourier Transform: $F(u,v)$

$\downarrow$

Filter function: $H(u,v) \cdot F(u,v)$

$\downarrow$

Inverse Fourier Transform

$\downarrow$

Output image $g(x,y)$

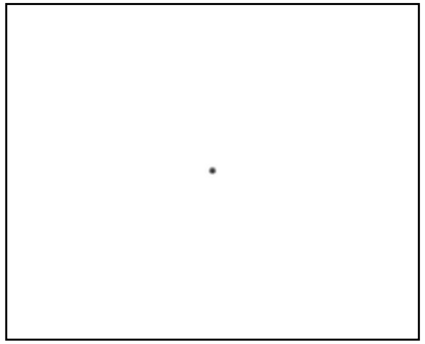
Example

Reminder: $F(0,0) = \bar{f}$

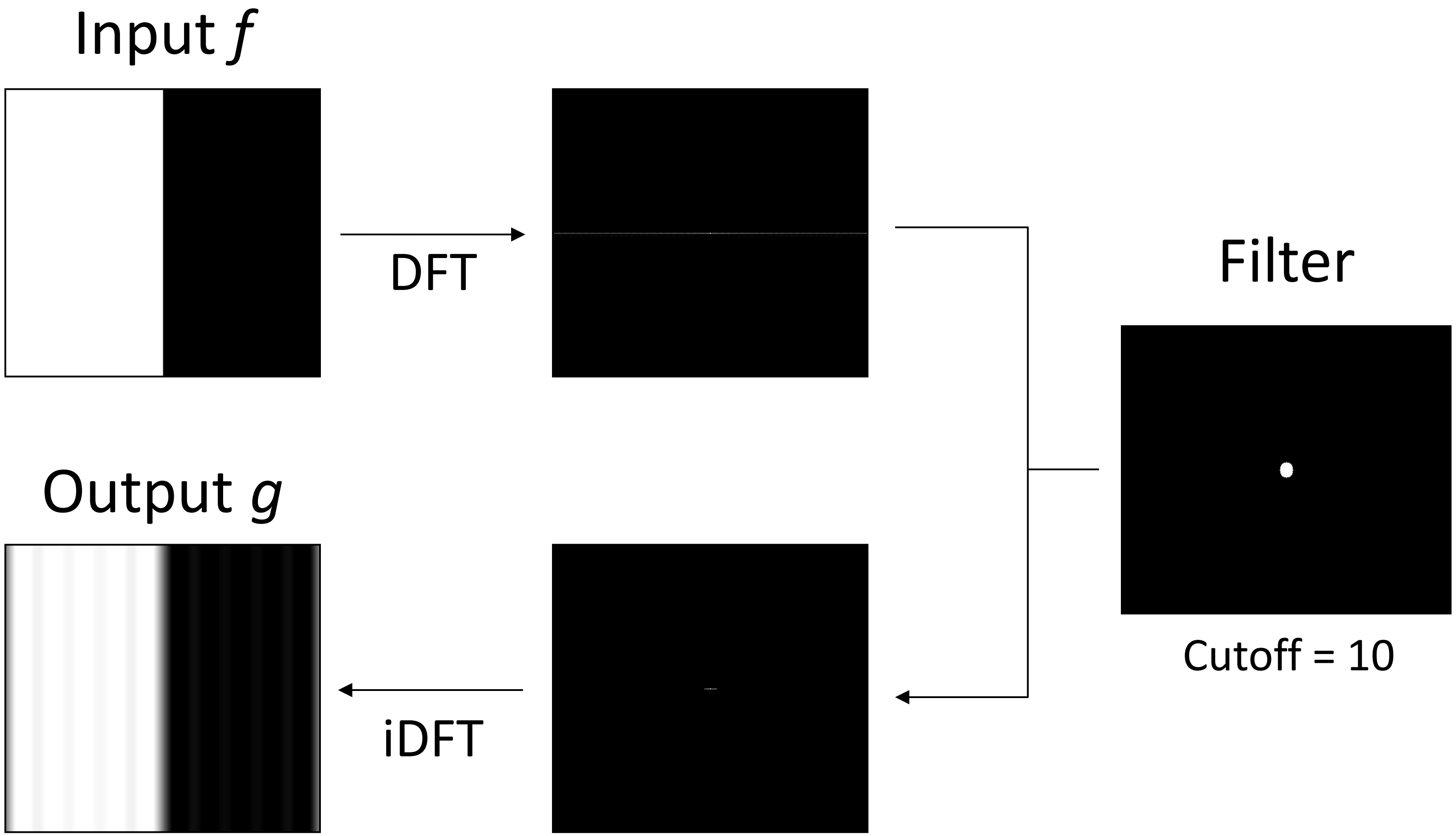
$$F(0,0) = 0$$

$\longrightarrow$

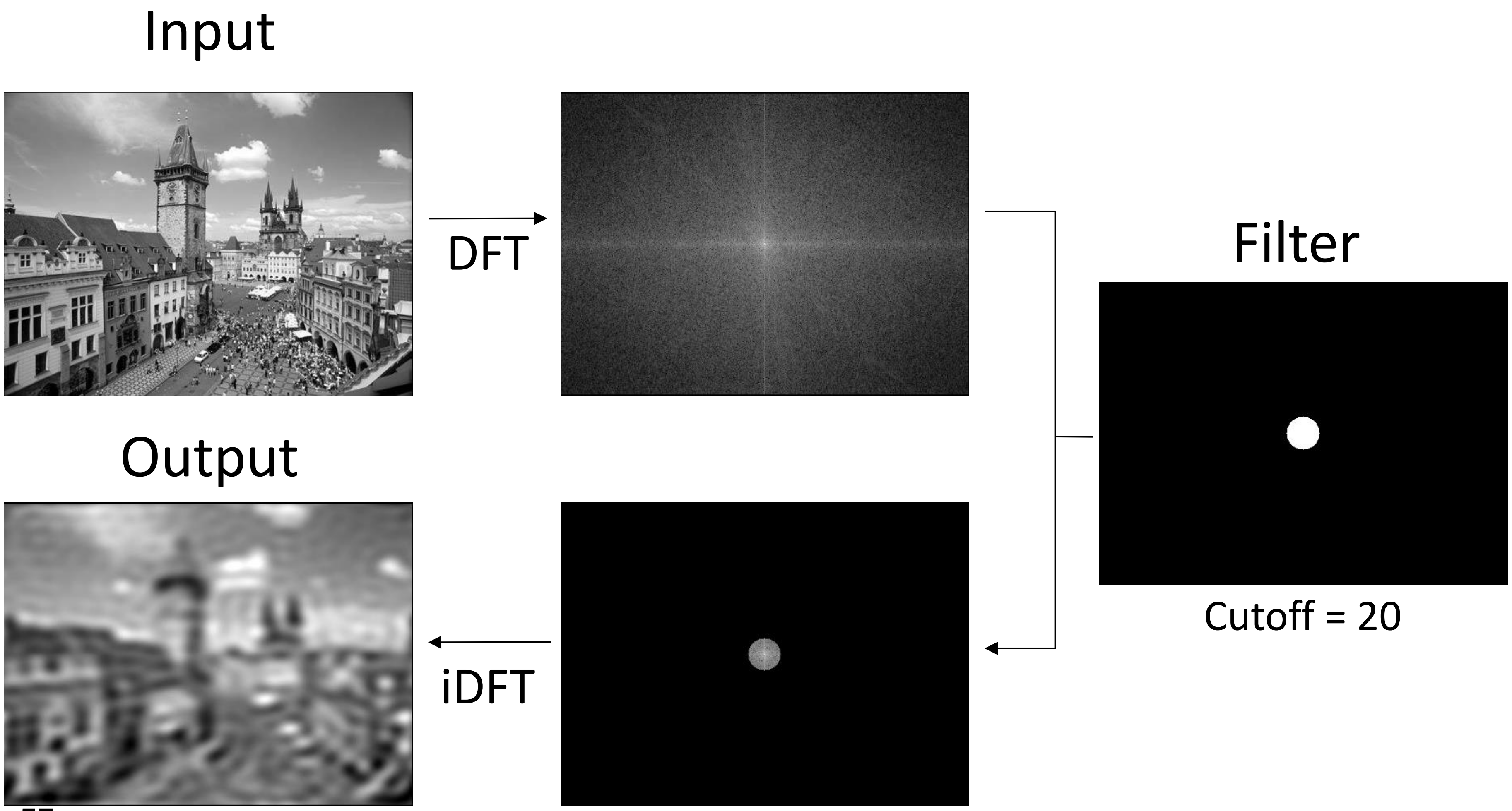
Multiply the Fourier transform by the filter:



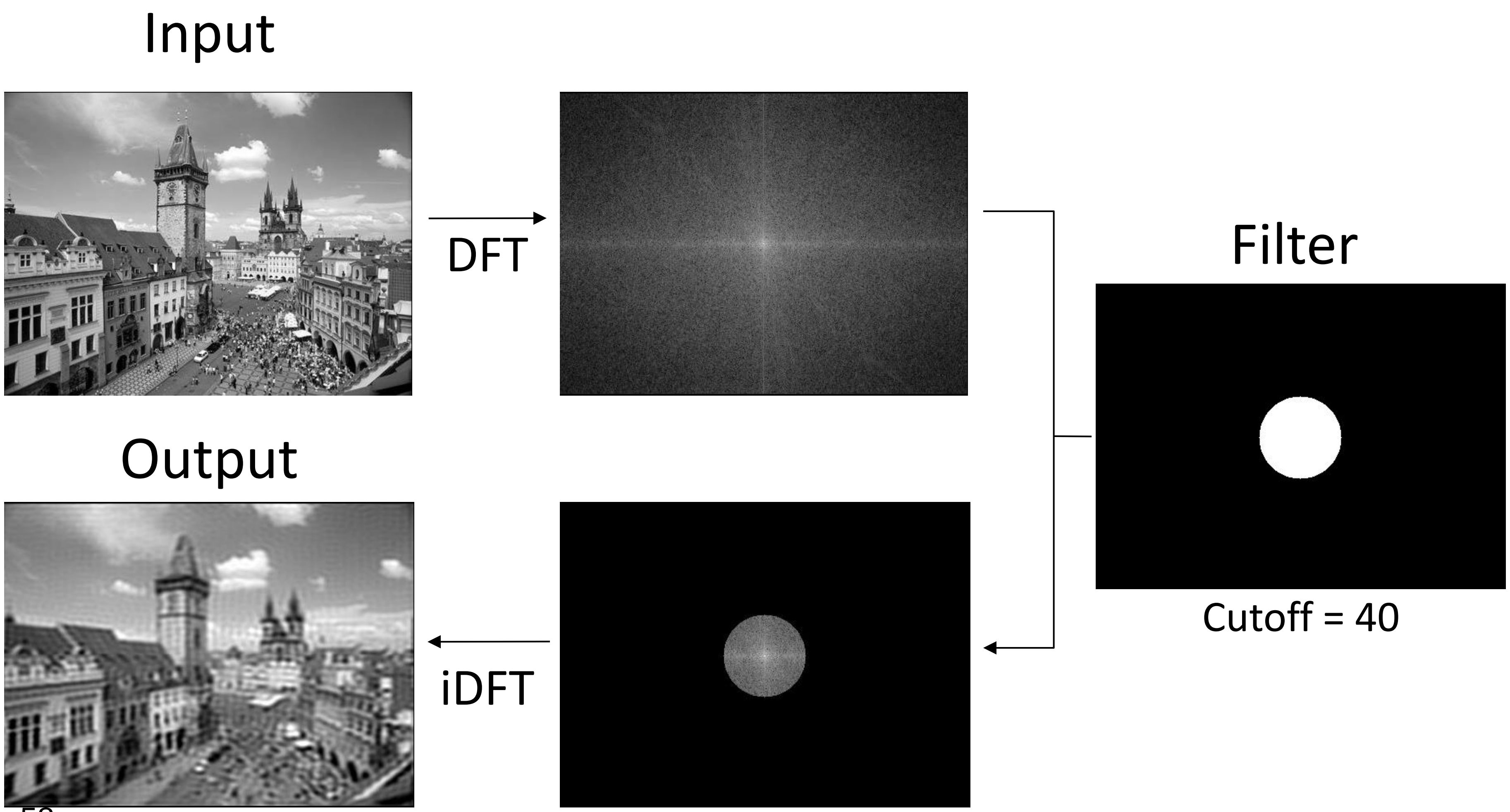
Ideal Low-pass Filters



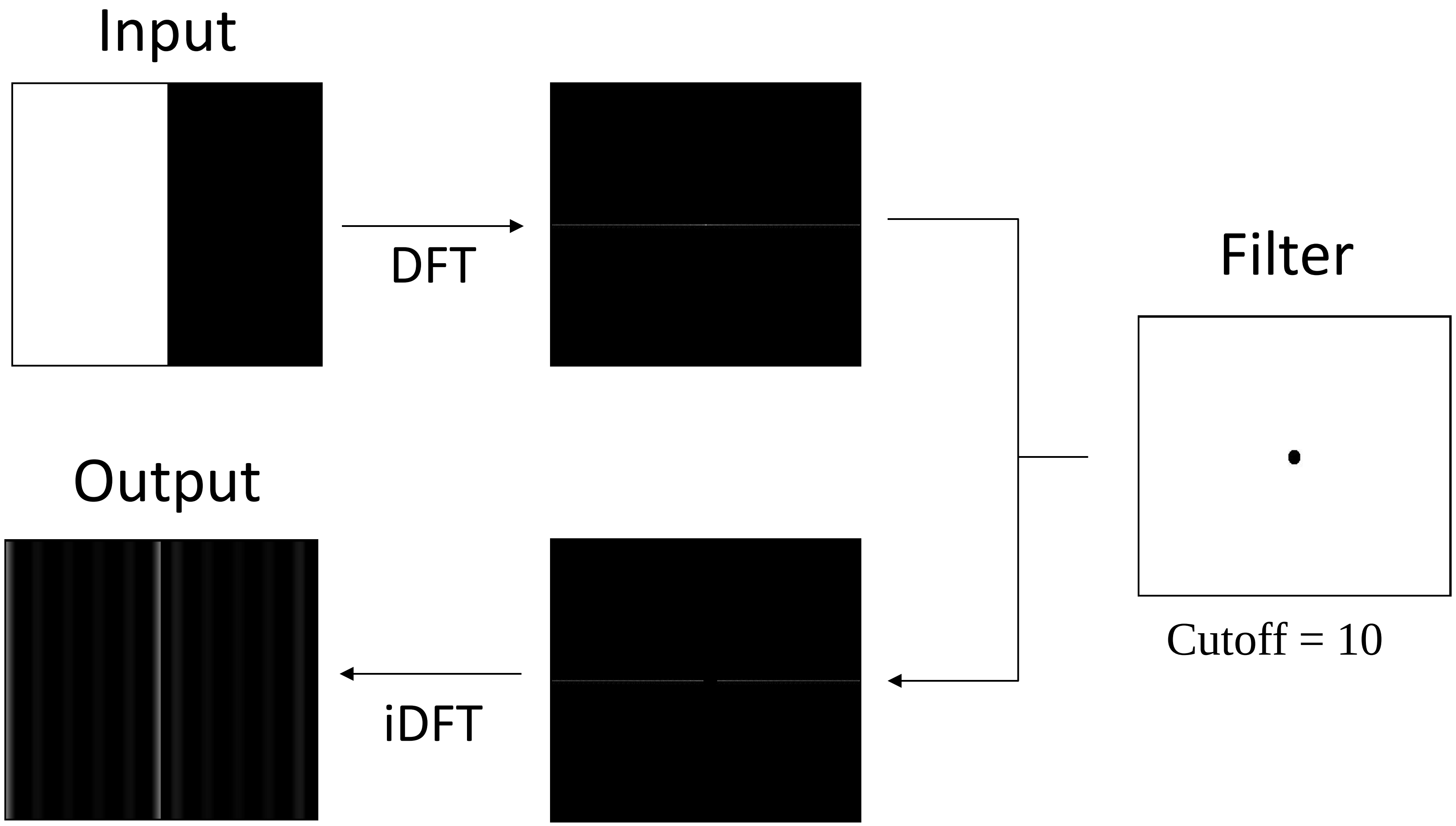
Ideal Low-pass Filters



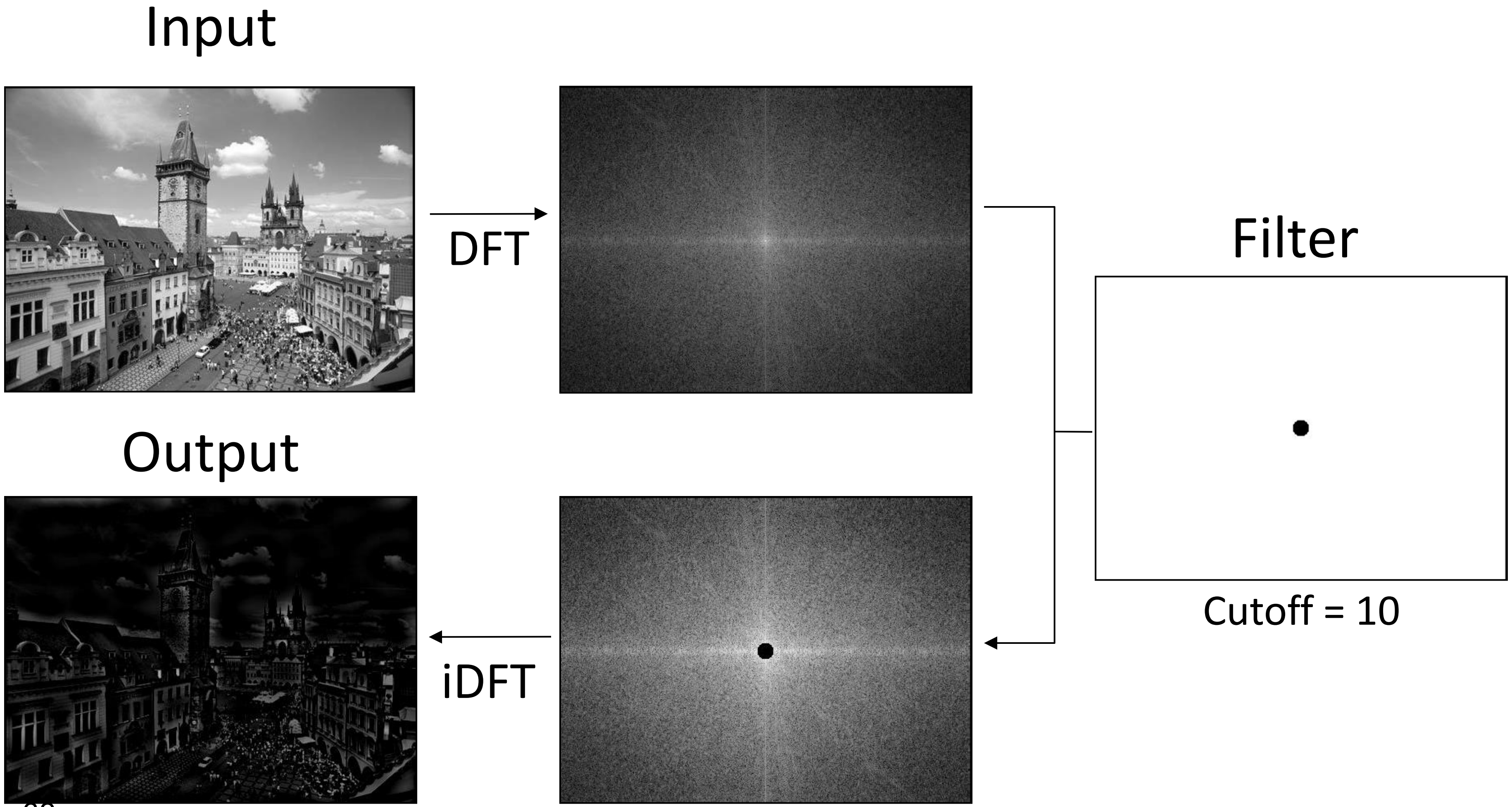
Cutoff = 40



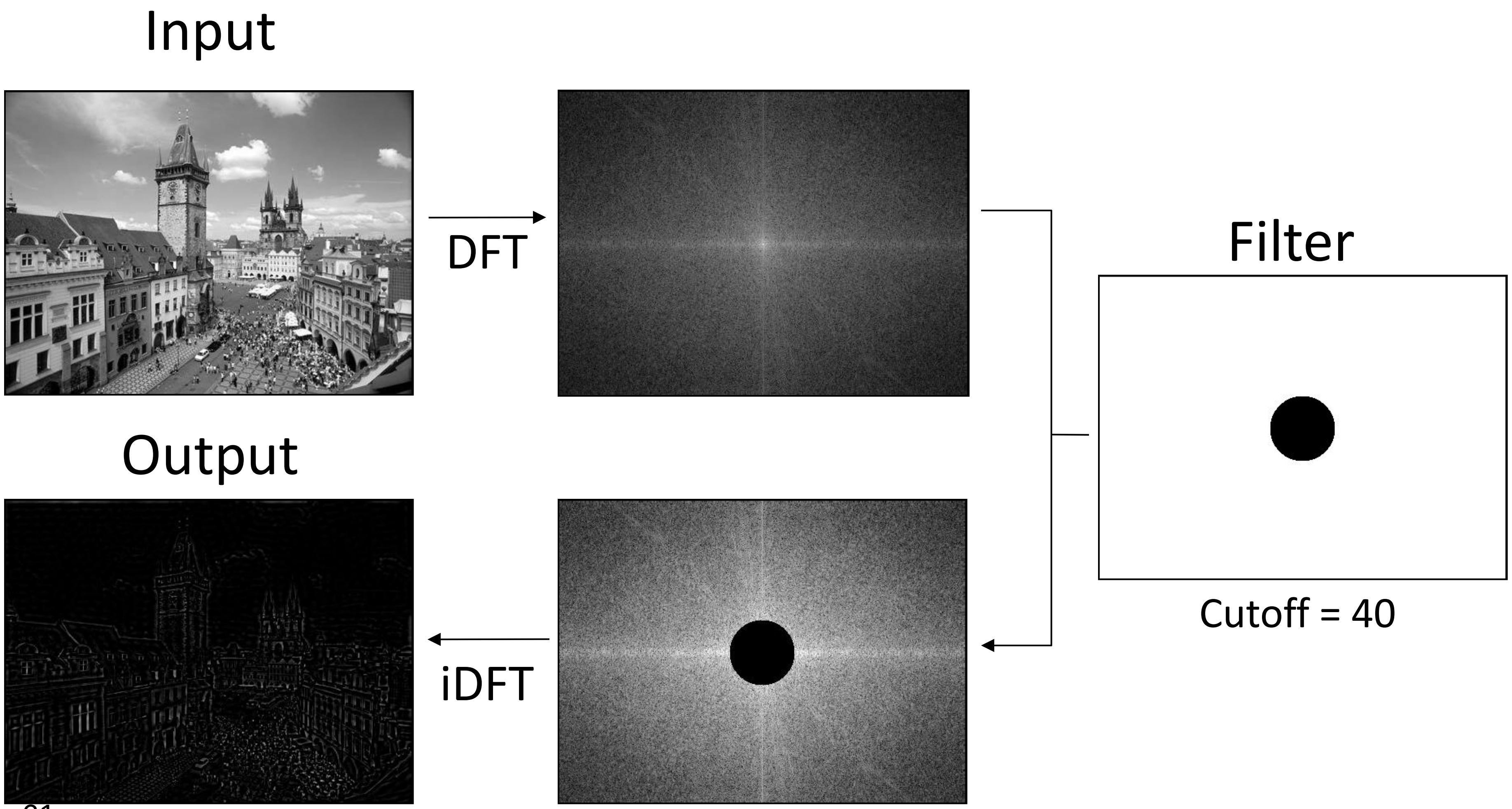
Ideal High-pass Filters



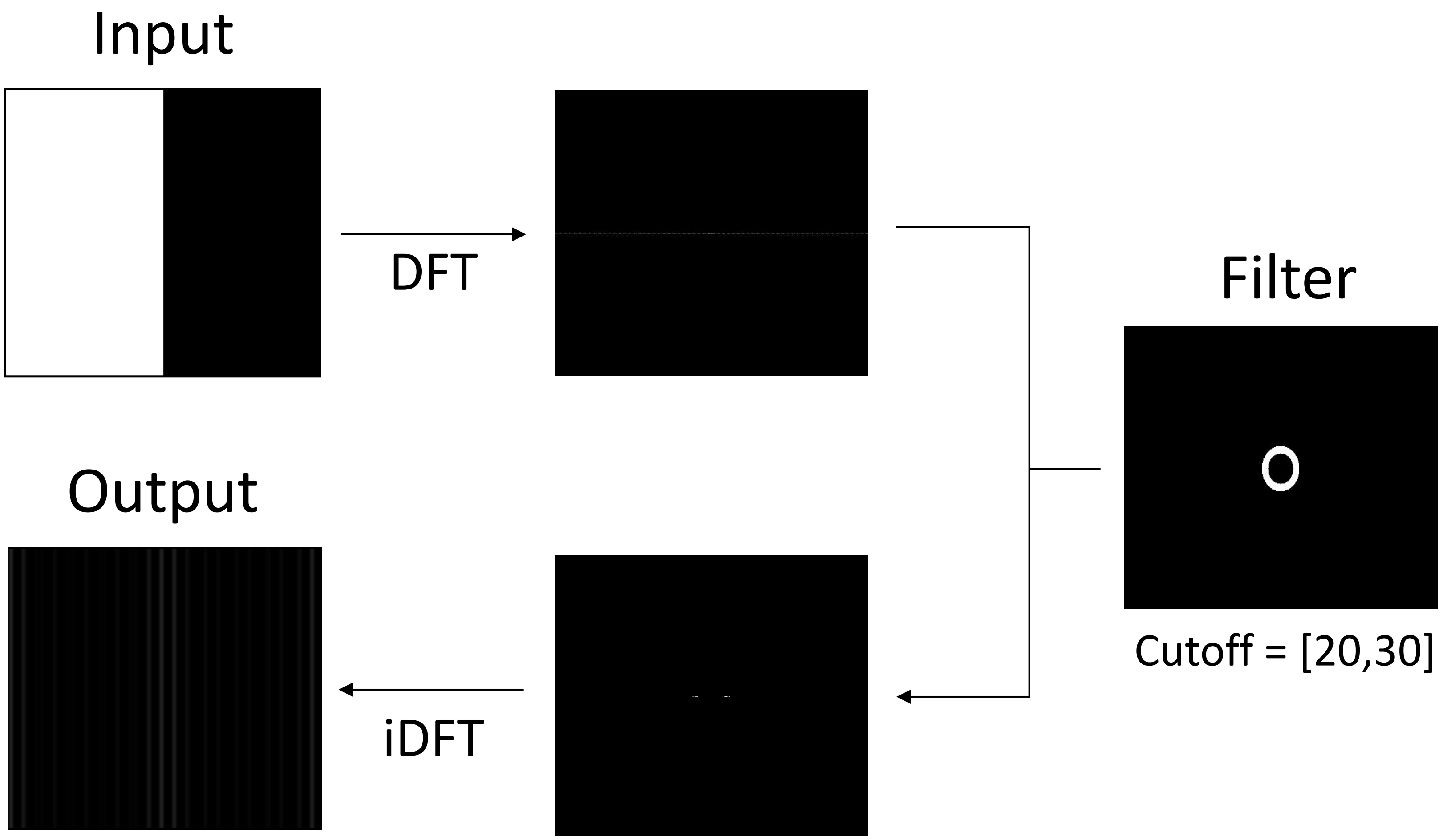
Ideal High-pass Filter



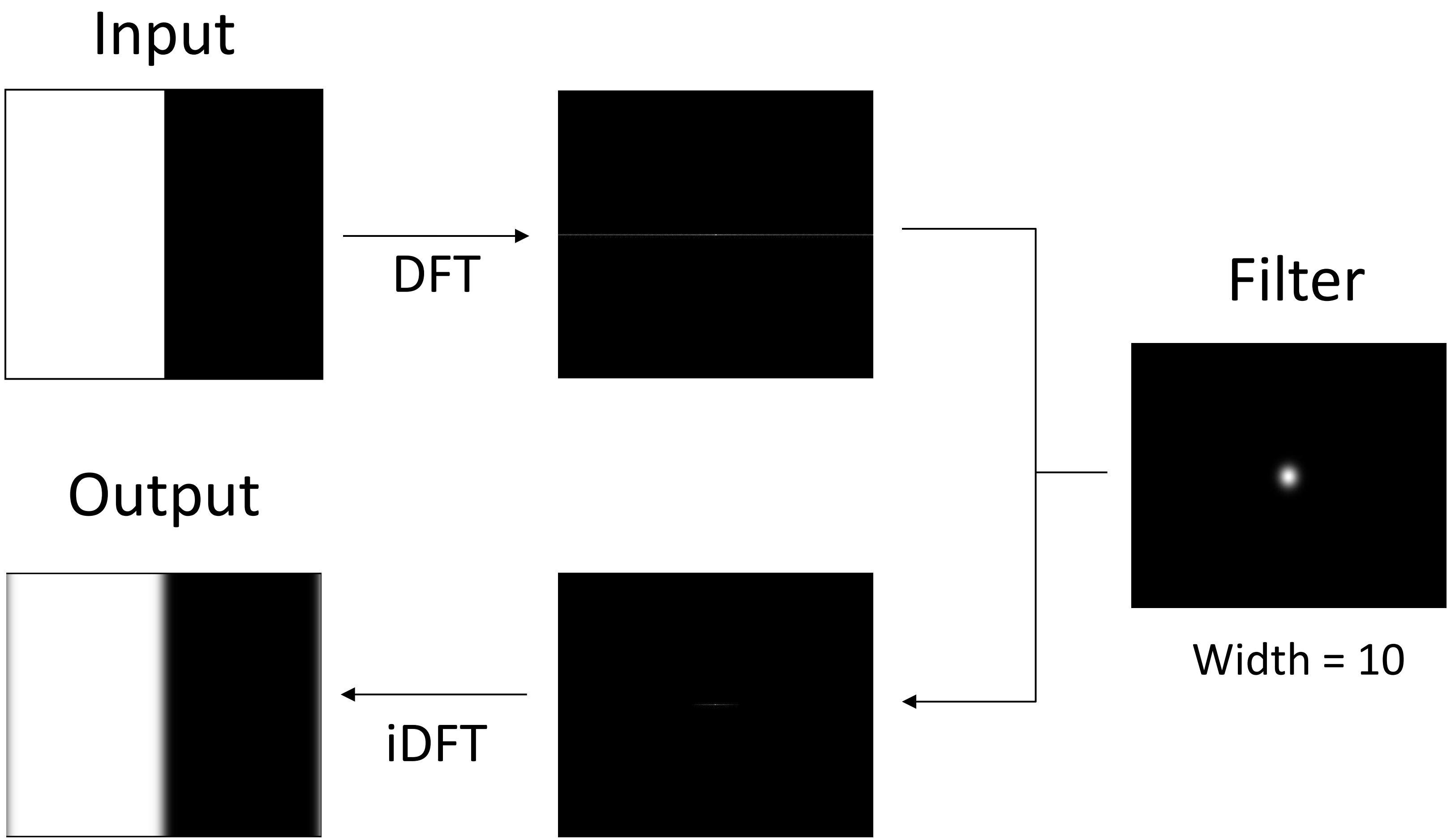
Ideal High-pass Filter



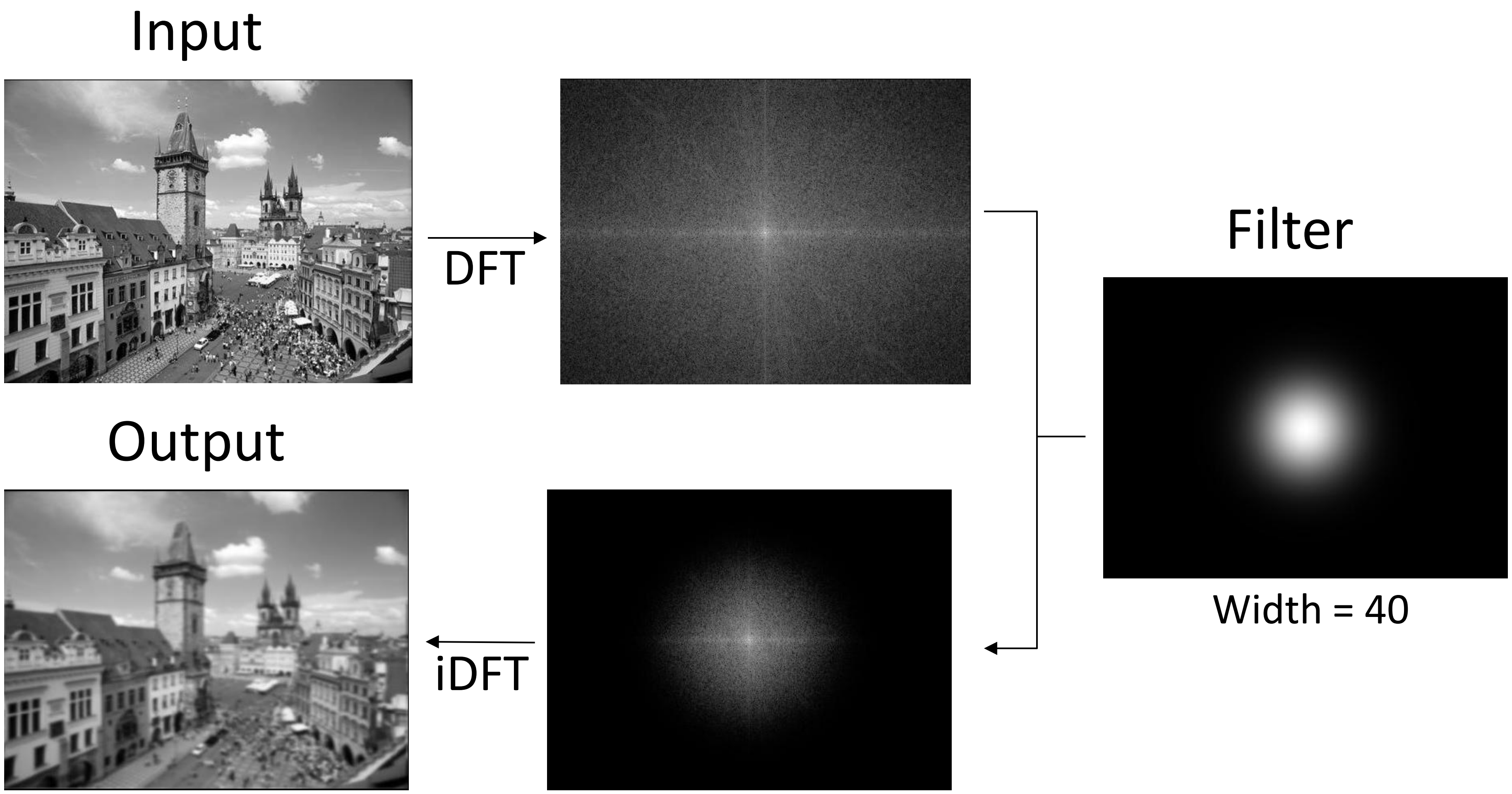
Ideal Band-pass Filter



Gaussian Filter



Gaussian Filter



Next week:
Convolution