Intro to

Computer Vision

with Prof. Kosta Derpanis

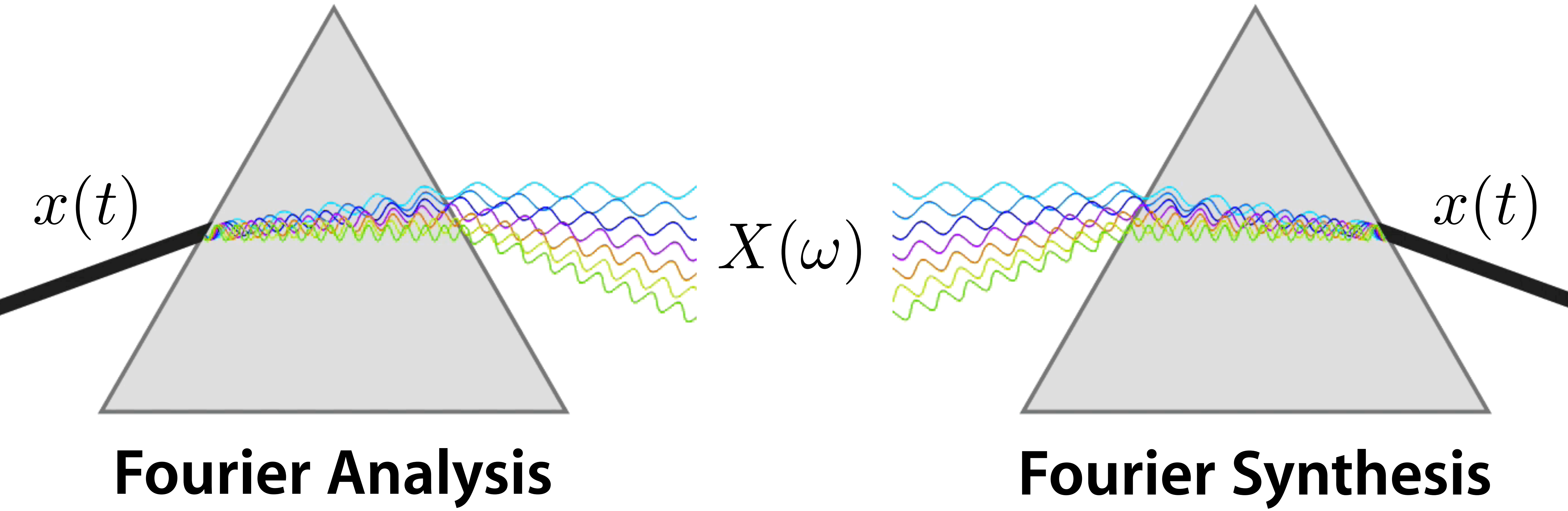
Frequency Analysis

Part 1

Thinking in terms of

FREQUENCY

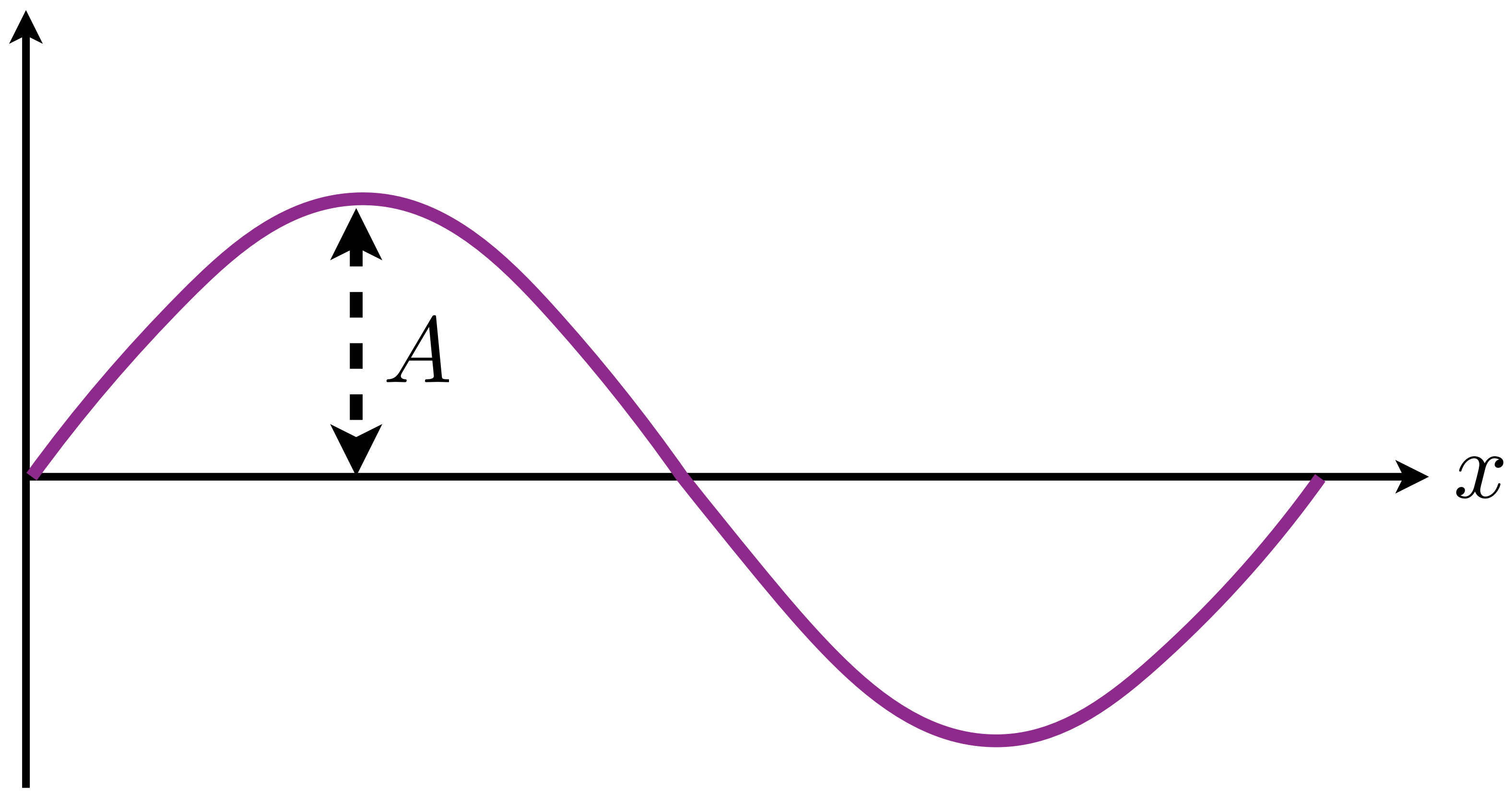
signal to frequency



ANY periodic function can be written as a weighted sum of sines and cosines of different frequencies.

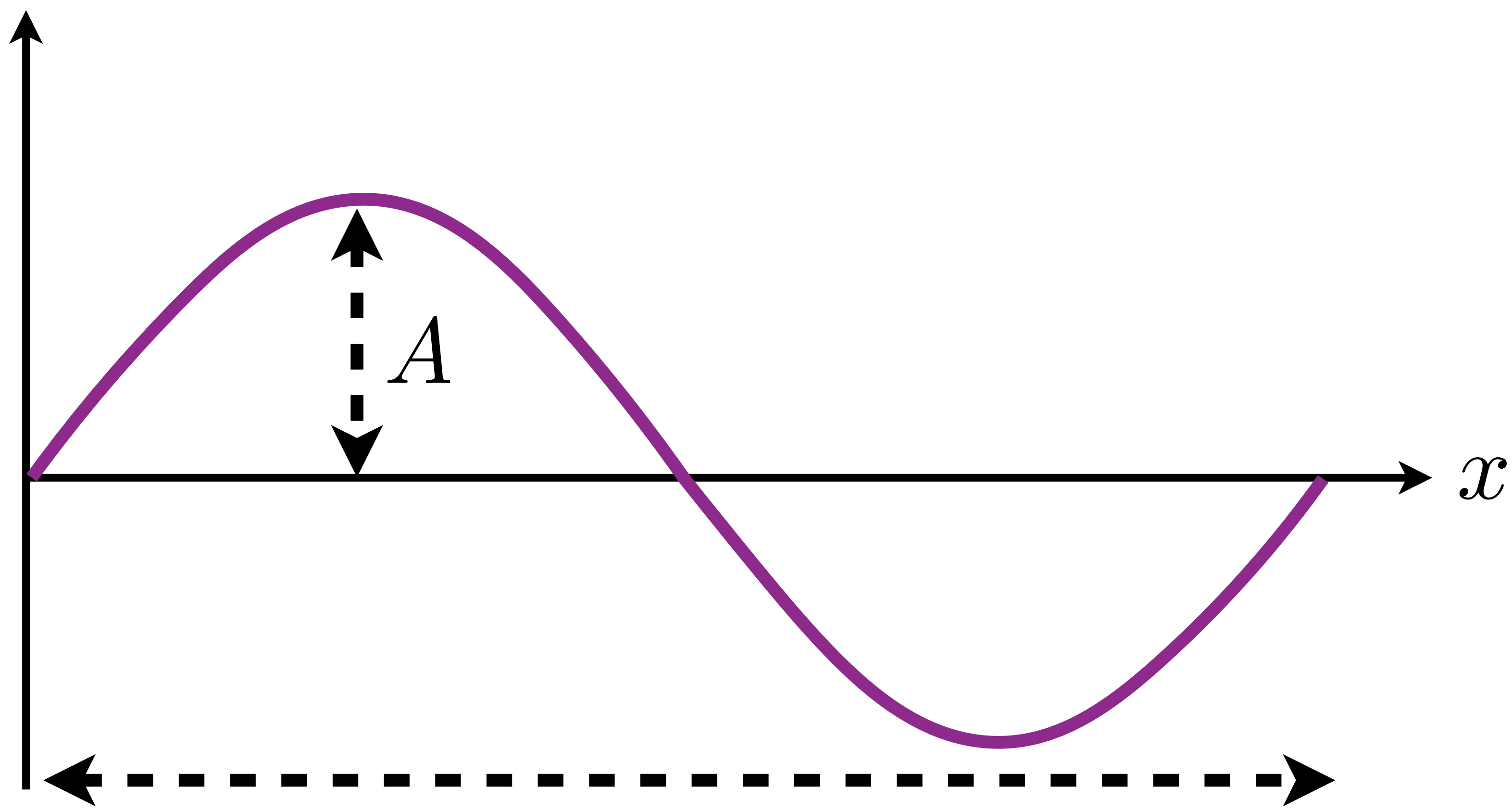
$$A \sin(\omega x + \phi)$$

amplitude



frequency

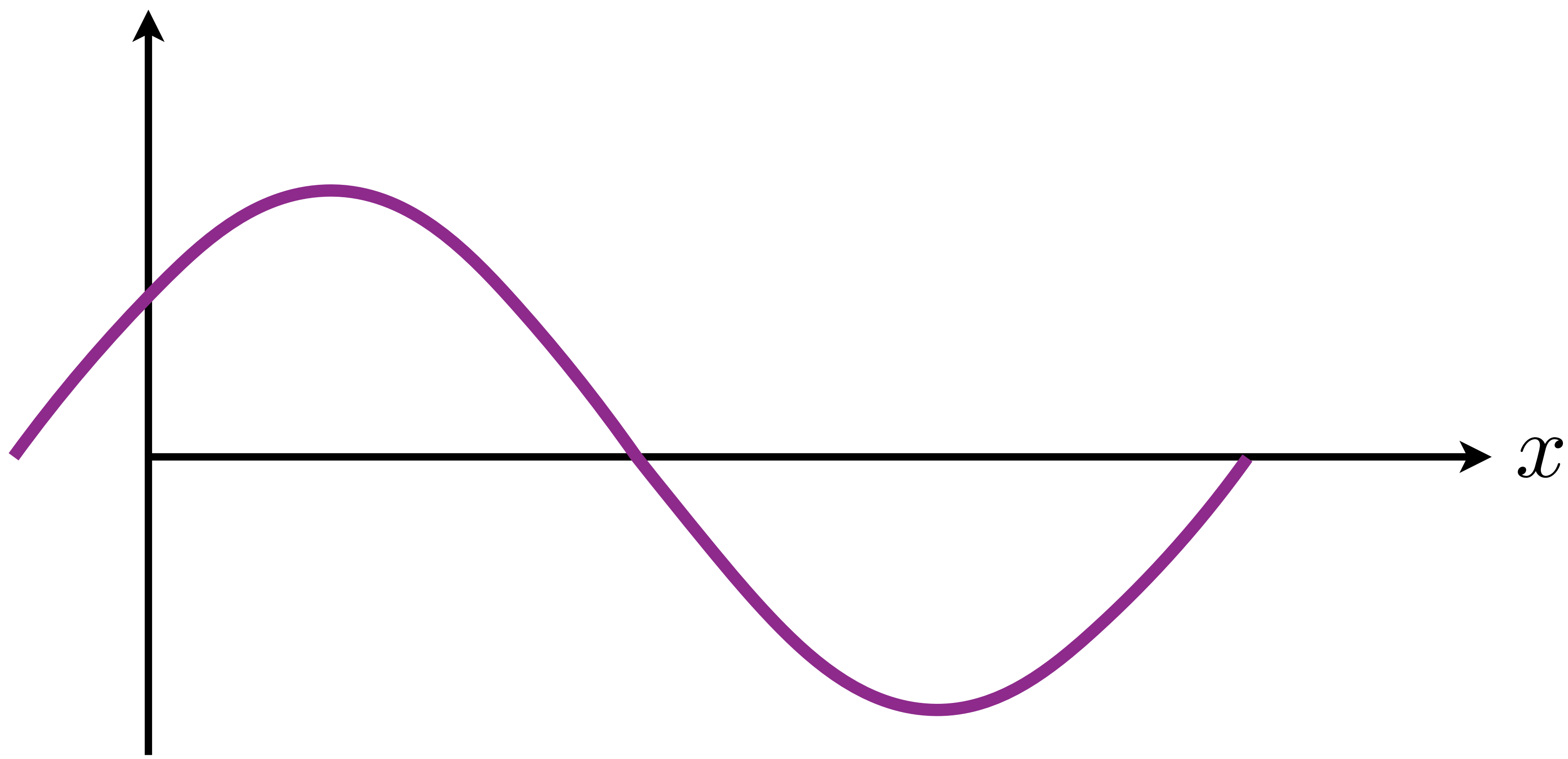
$$A \sin(\omega x + \phi)$$



$$T = \frac{2\pi}{\omega}$$

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

phase shift



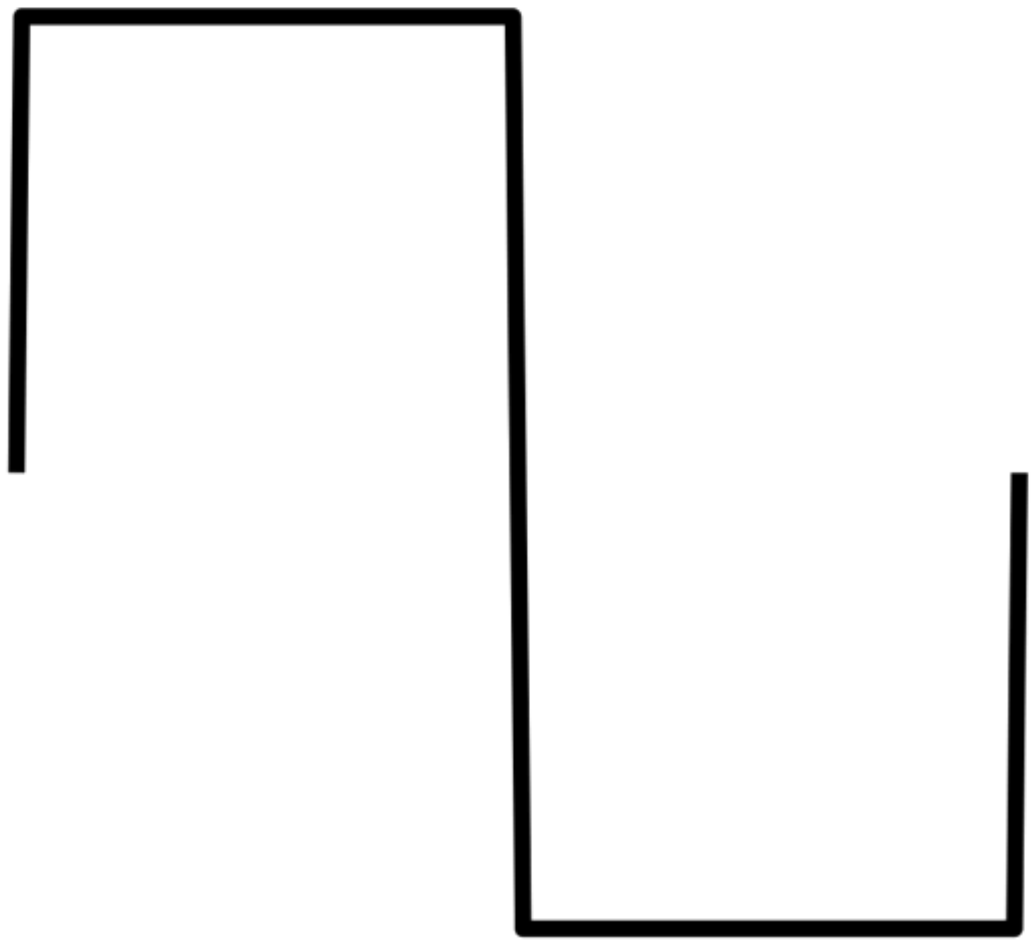
$$A\sin(\omega x + \phi)$$

$$= A[\sin(\phi)\cos(\omega x) + \cos(\phi)\sin(\omega x)]$$

$$= \underbrace{A\sin(\phi)}_{\text{constant}}\cos(\omega x) + \underbrace{A\cos(\phi)}_{\text{constant}}\sin(\omega x)$$

$$
\begin{aligned}
&A\sin(\omega x+\phi) \\
&\qquad = A[\sin(\phi)\cos(\omega x)+\cos(\phi)\sin(\omega x)] \\
&\qquad = A\sin(\phi)\cos(\omega x)+A\cos(\phi)\sin(\omega x) \\
&\qquad = \alpha\cos(\omega x)+\beta\sin(\omega x)
\end{aligned}
$$

$$A \sin(\omega x + \phi) = \alpha \cos(\omega x) + \beta \sin(\omega x)$$



$$= \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin(2\pi(2k-1)x)}{2k-1}$$

$f(x) \longrightarrow$ $\boxed{\textbf{Fourier transform}}$ $\longrightarrow F(\omega)$

For every ω, $F(\omega)$ holds the amplitude, A, and phase, ϕ, of the corresponding sinusoid function.

How can $F(\omega)$ hold BOTH the amplitude and phase?

complex numbers

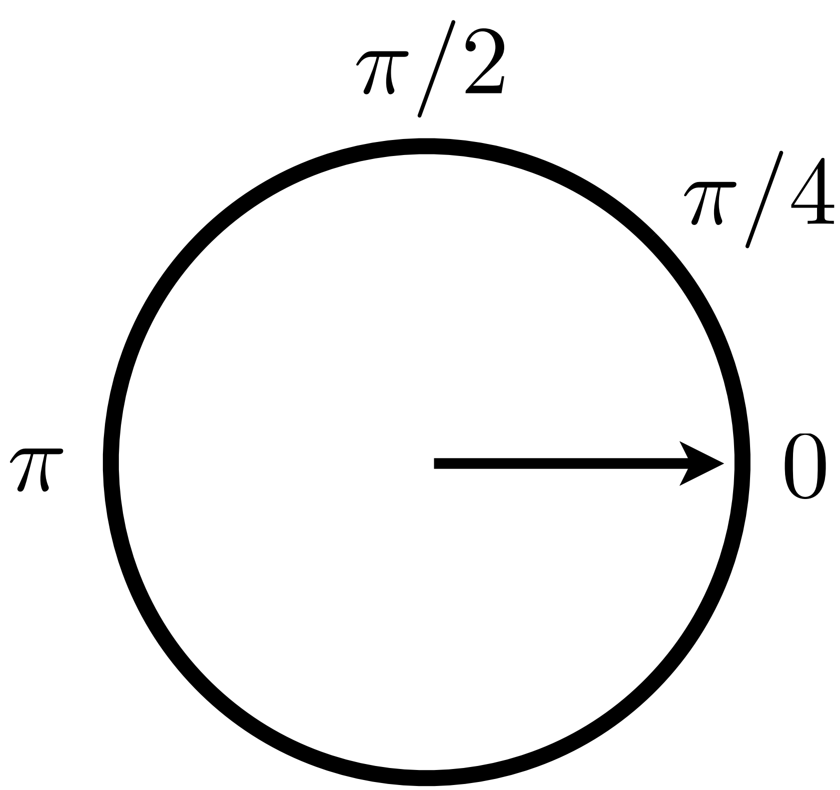
complex numbers

$$\alpha + i\beta$$

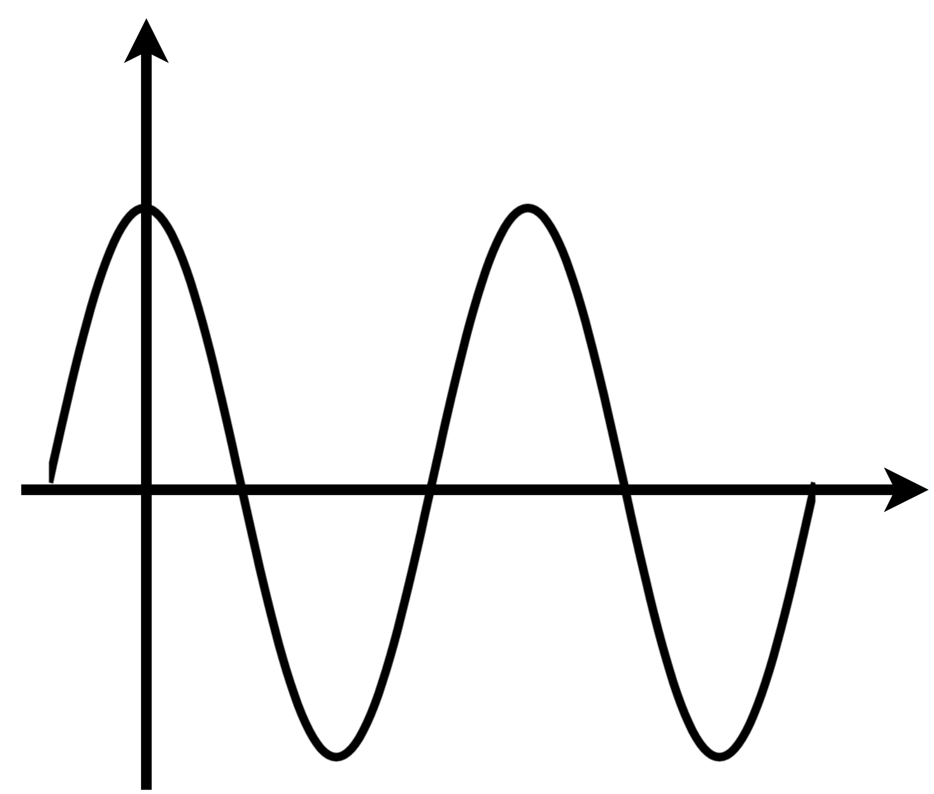
with

$$i = \sqrt{-1}$$

$$\alpha + i\beta = A(\cos(\theta) + i\sin(\theta))$$



complex plane



$$f(x) \longrightarrow \boxed{\textbf{Fourier transform}} \longrightarrow F(\omega)$$

$$F(\omega) = \mathrm{Real}(\omega) + i\,\mathrm{Imaginary}(\omega)$$

$$M(\omega) = \|F(\omega)\| = \sqrt{\mathrm{Real}(\omega)^2 + \mathrm{Imaginary}(\omega)^2}$$

Amplitude

$f(x) \longrightarrow$ **Fourier transform** $\longrightarrow F(\omega)$

$$F(\omega) = \text{Real}(\omega) + i\text{Imaginary}(\omega)$$

$$M(\omega) = \|F(\omega)\| = \sqrt{\text{Real}(\omega)^2 + \text{Imaginary}(\omega)^2}$$

$$\phi(\omega) = \tan^{-1}\left(\frac{\text{Imaginary}(\omega)}{\text{Real}(\omega)}\right)$$

Phase

Euler's Identity

$$Ae^{ik} = A(\cos(k) + i\sin(k))$$

1D Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

where

$$e^{-i2\pi\omega x} = \cos(-2\pi\omega x) + i\sin(-2\pi\omega x)$$

measures how much of each individual frequency is present in the function

1D Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

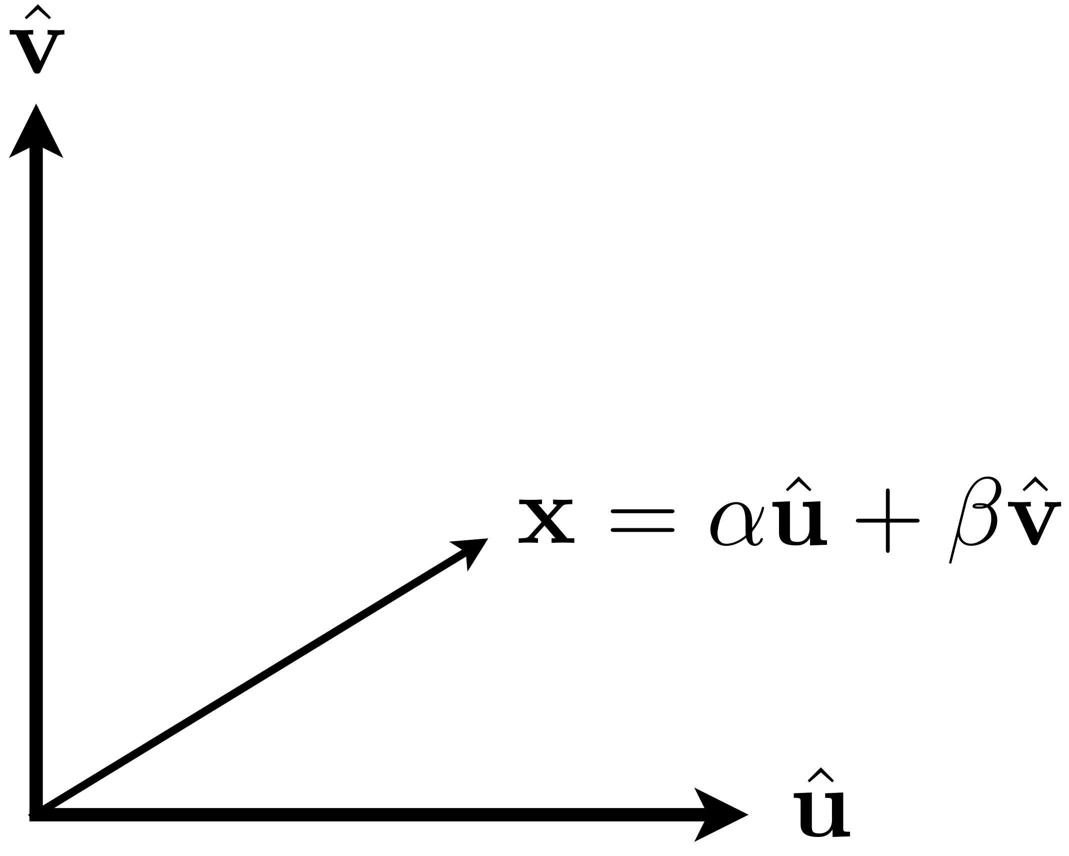
where

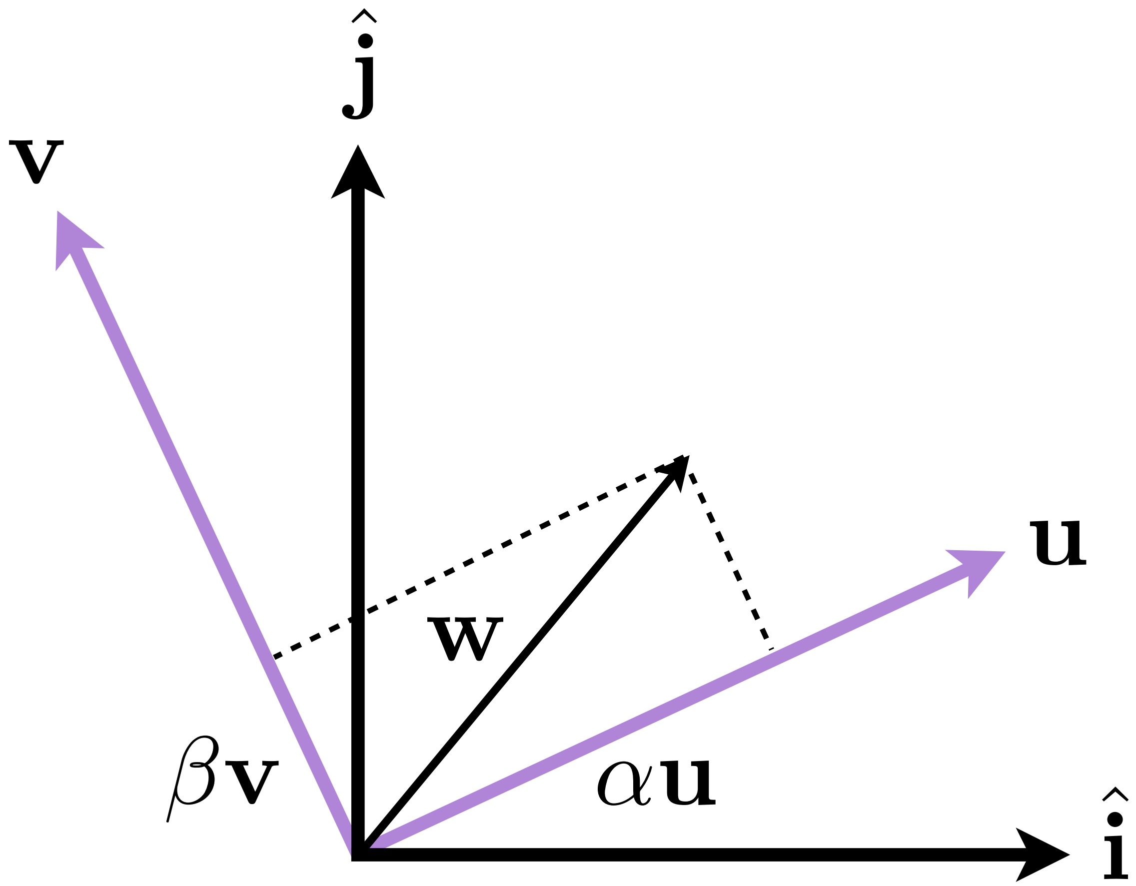
$$e^{-i2\pi\omega x} = \cos(-2\pi\omega x) + i\sin(-2\pi\omega x)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(x)\cos(-2\pi\omega x)dx + i\int_{-\infty}^{\infty} f(x)\sin(-2\pi\omega x)dx$$

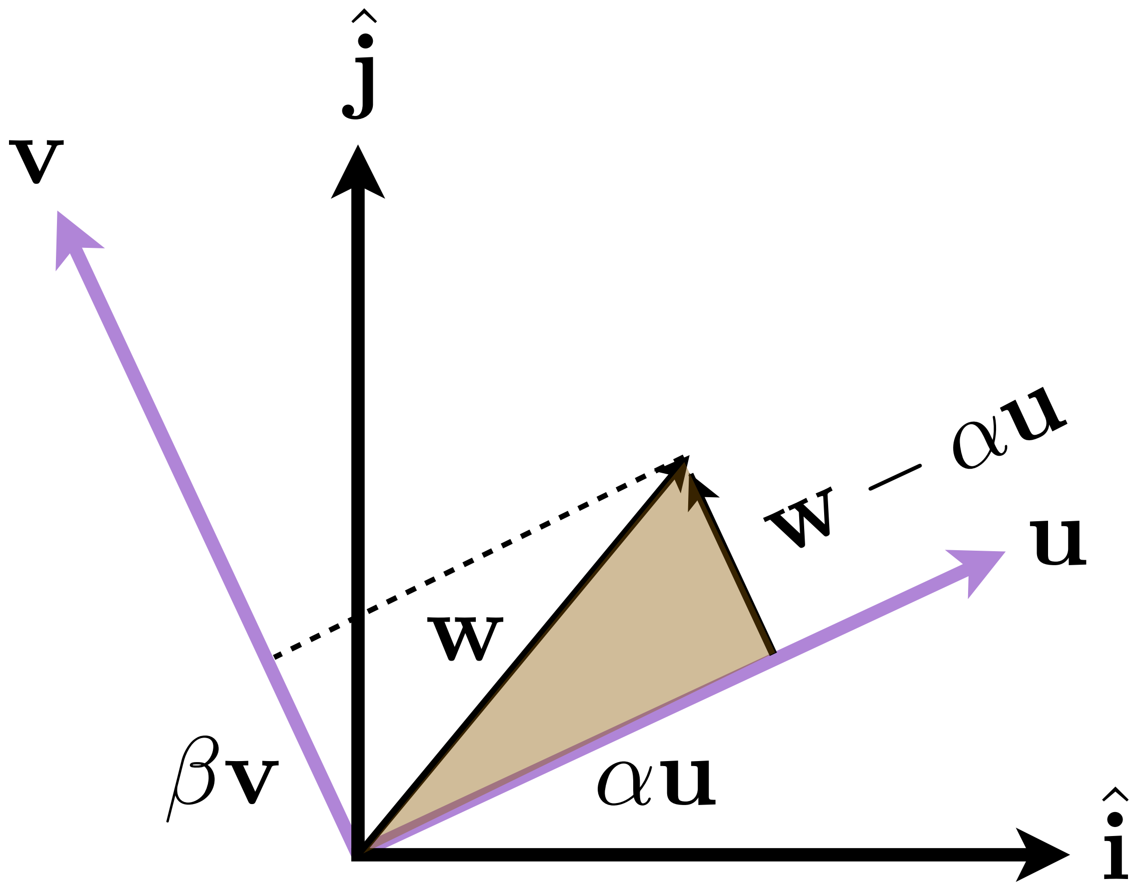
Definition: A *basis* is a set of linearly independent vectors that via linear combination can represent every vector in a given vector space.

Definition: A *basis* is a set of linearly independent vectors that via linear combination can represent every vector in a given vector space.





A vector (in 2D) can be expressed as a sum of two vectors.



Assuming the basis is orthogonal, what is α and β ?

$$\alpha\mathbf{u} \cdot (\mathbf{w} - \alpha\mathbf{u}) = 0$$
$$\alpha\mathbf{u} \cdot \mathbf{w} - \alpha\mathbf{u} \cdot \alpha\mathbf{u} = 0$$
$$\alpha\mathbf{u} \cdot \mathbf{u} = \mathbf{u} \cdot \mathbf{w}$$

Vector Projection

$$\alpha = \frac{\mathbf{u} \cdot \mathbf{w}}{\mathbf{u} \cdot \mathbf{u}} \qquad \beta = \frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{v} \cdot \mathbf{v}}$$

Images as points in vector space

Given an image $N \times N$, can treat it as a vector

$$\begin{bmatrix} x_{00} & x_{10} & \cdots & x_{(N-1)(N-1)} \end{bmatrix}^\top$$

Images as points in vector space

Given an image $N \times N$, can treat it as a vector

$$\begin{bmatrix} x_{00} & x_{10} & \cdots & x_{(N-1)(N-1)} \end{bmatrix}^\top$$

Standard basis is the vector set with a single pixel set to one

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^\top$$

Standard Image Basis

3	8
10	50

Standard Image Basis

$$\begin{bmatrix} 3 \\ 8 \\ 10 \\ 50 \end{bmatrix} = 3\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 8\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + 10\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + 50\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{u} = \sum \alpha_i \mathbf{e}_i = \sum (\mathbf{u} \cdot \mathbf{e}_i) \mathbf{e}_i$$

vector projection

1D Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

where

$$e^{-i2\pi\omega x} = \cos(-2\pi\omega x) + i\sin(-2\pi\omega x)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(x)\cos(-2\pi\omega x)dx + i\int_{-\infty}^{\infty} f(x)\sin(-2\pi\omega x)dx$$

Projection onto sinusoidal basis set

$$F(\omega) = \int_{-\infty}^{\infty} f(x)\cos(-2\pi\omega x)dx + i\int_{-\infty}^{\infty} f(x)\sin(-2\pi\omega x)dx$$

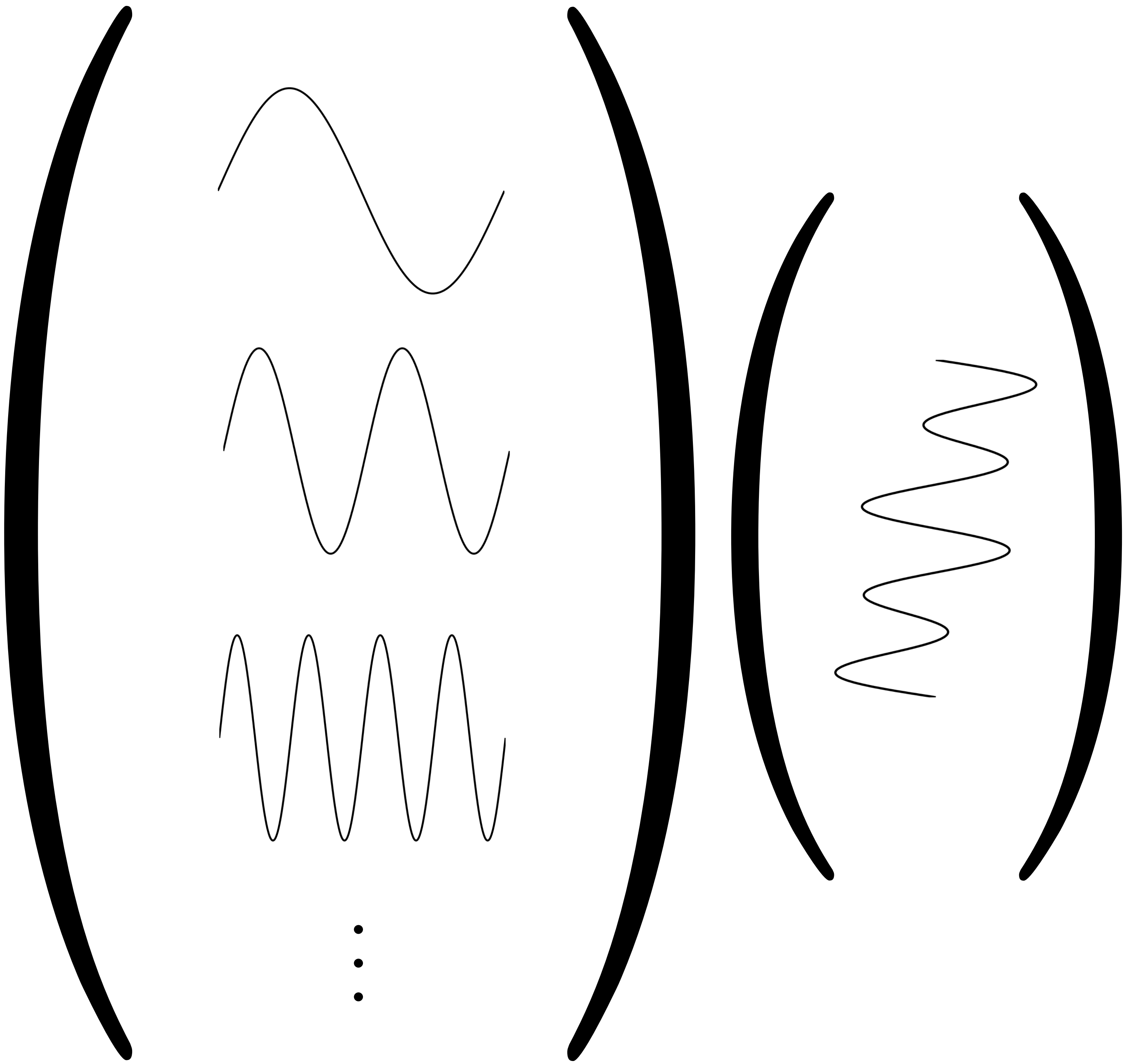
sines and cosines are orthogonal

1D Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x)\cos(-2\pi\omega x)dx + i\int_{-\infty}^{\infty} f(x)\sin(-2\pi\omega x)dx$$

sinusoids of different frequencies are orthogonal

Fourier Transform Intuition



$$= \begin{pmatrix} F(\omega_1) \\ F(\omega_2) \\ \vdots \end{pmatrix}$$

Fourier Transform Intuition

Just a change of basis

$$F(\omega) \longrightarrow \boxed{\text{inverse Fourier transform}} \longrightarrow f(x)$$

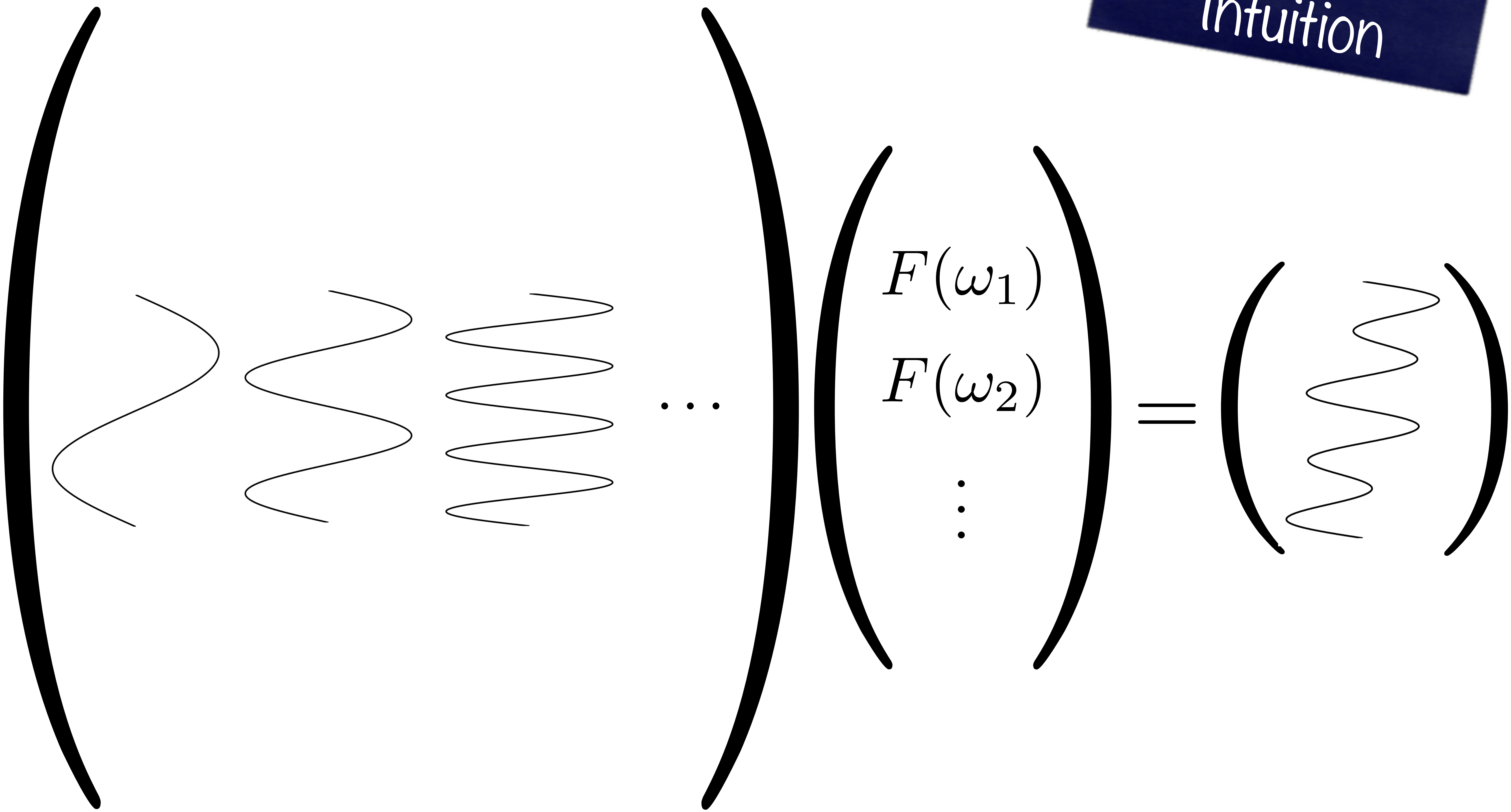
NOT an approximation

1D inverse Fourier Transform

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega x} d\omega$$

combines the contributions of each frequency to represent the signal in the "spatial" domain

Inverse Fourier Transform Intuition



forward transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

versus

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega x} d\omega$$

inverse transform

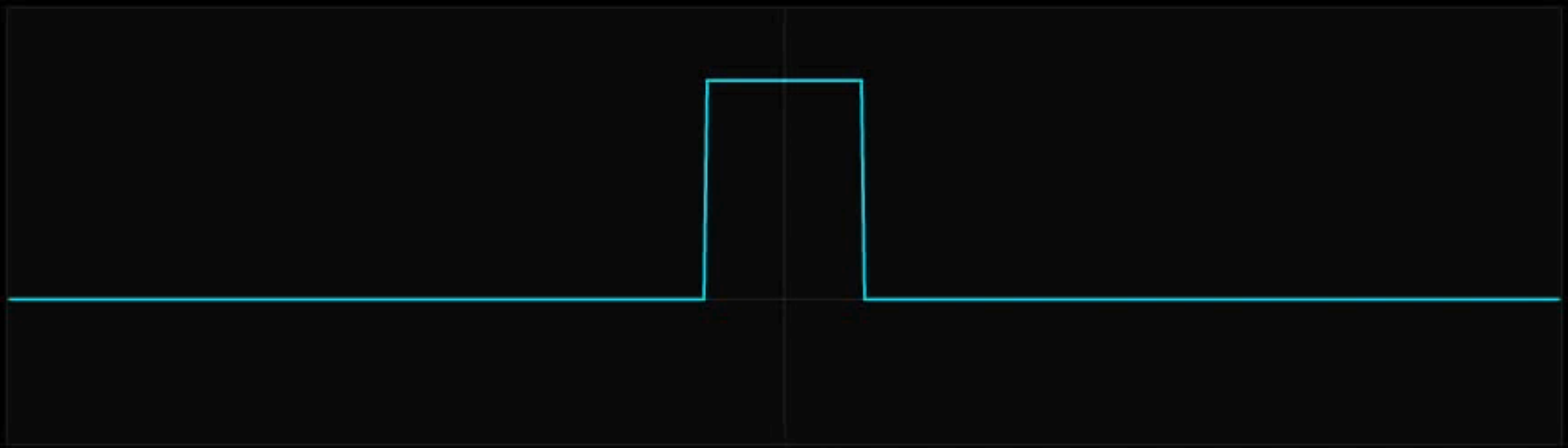
forward transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

versus

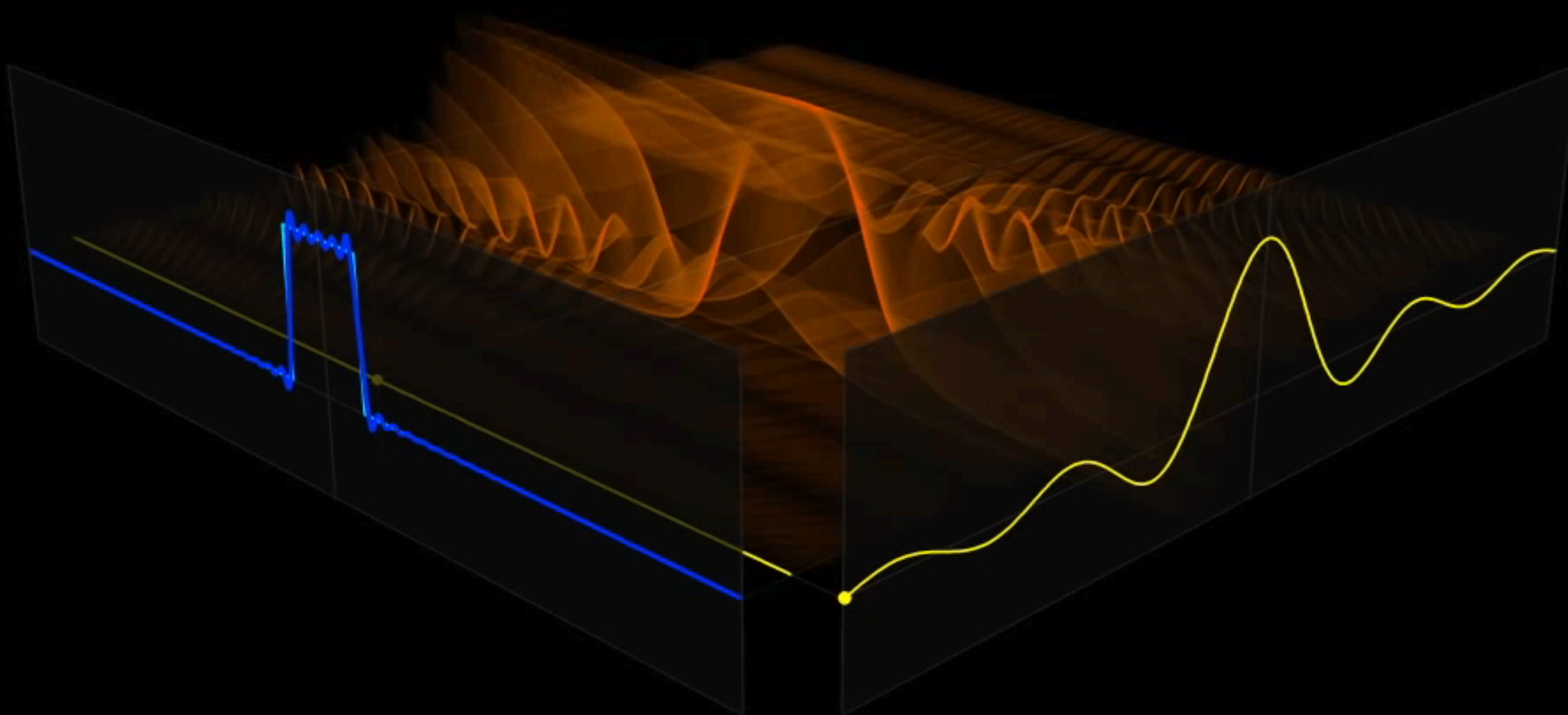
$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega x} d\omega$$

inverse transform

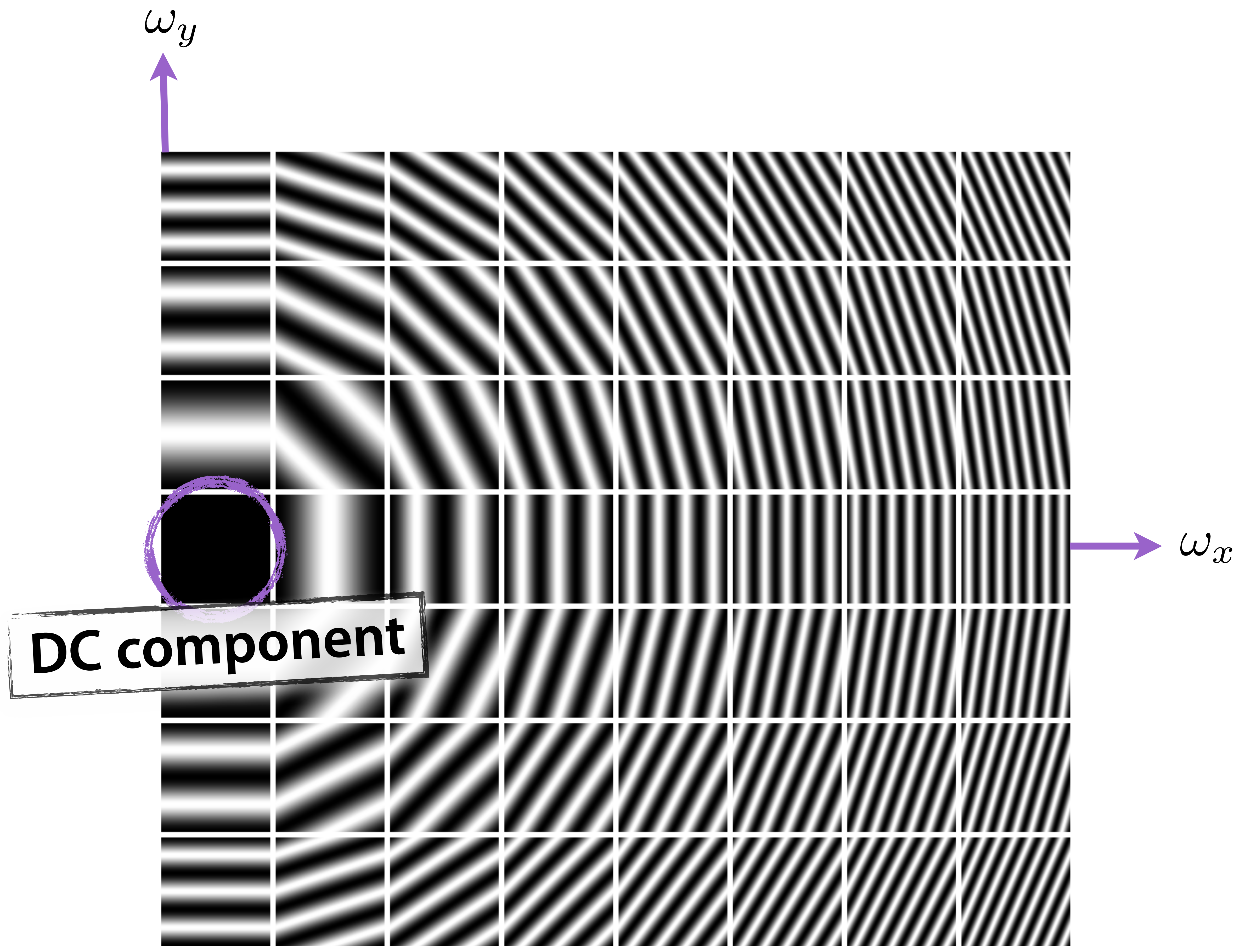


$$f(x)$$

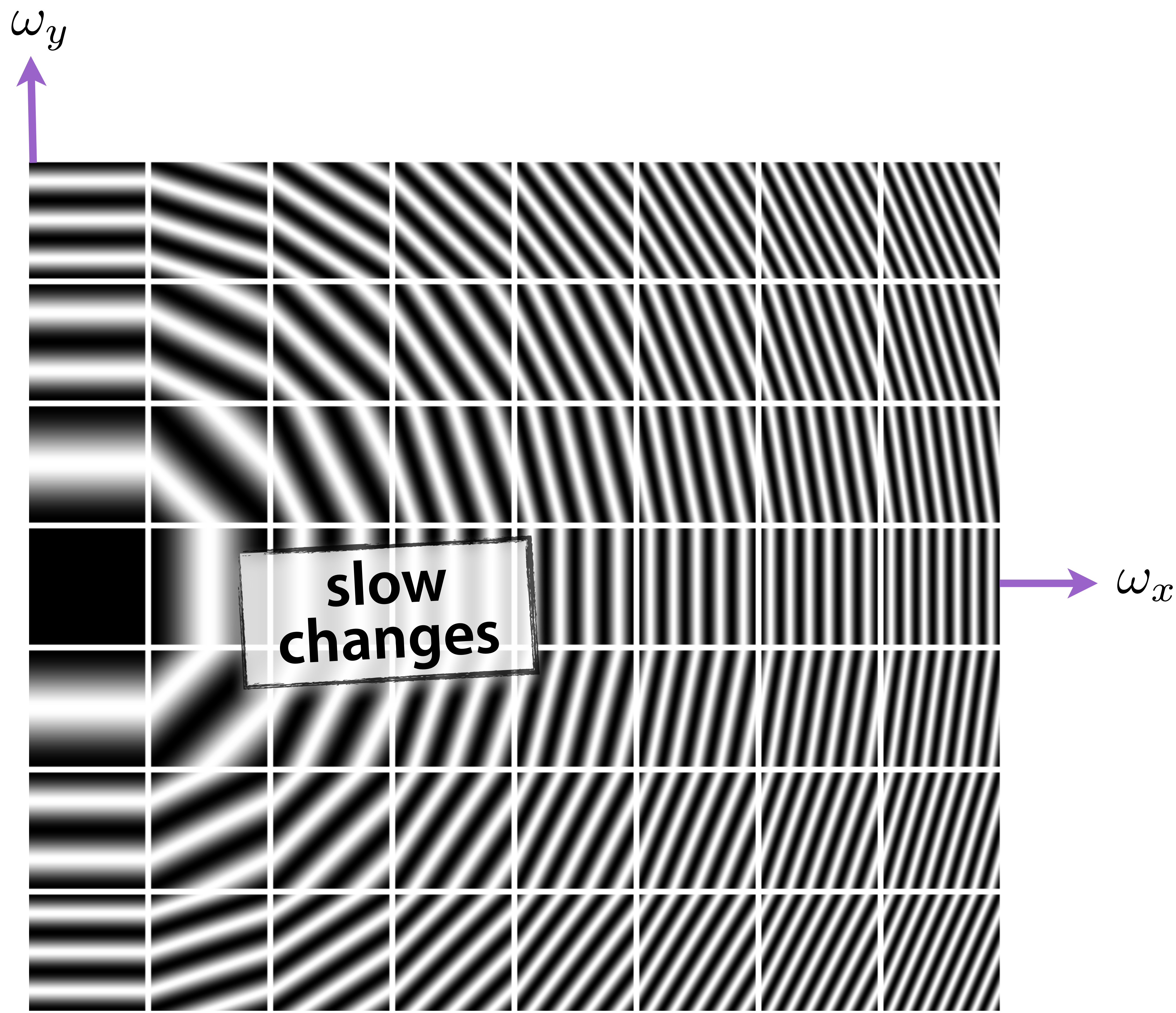
$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

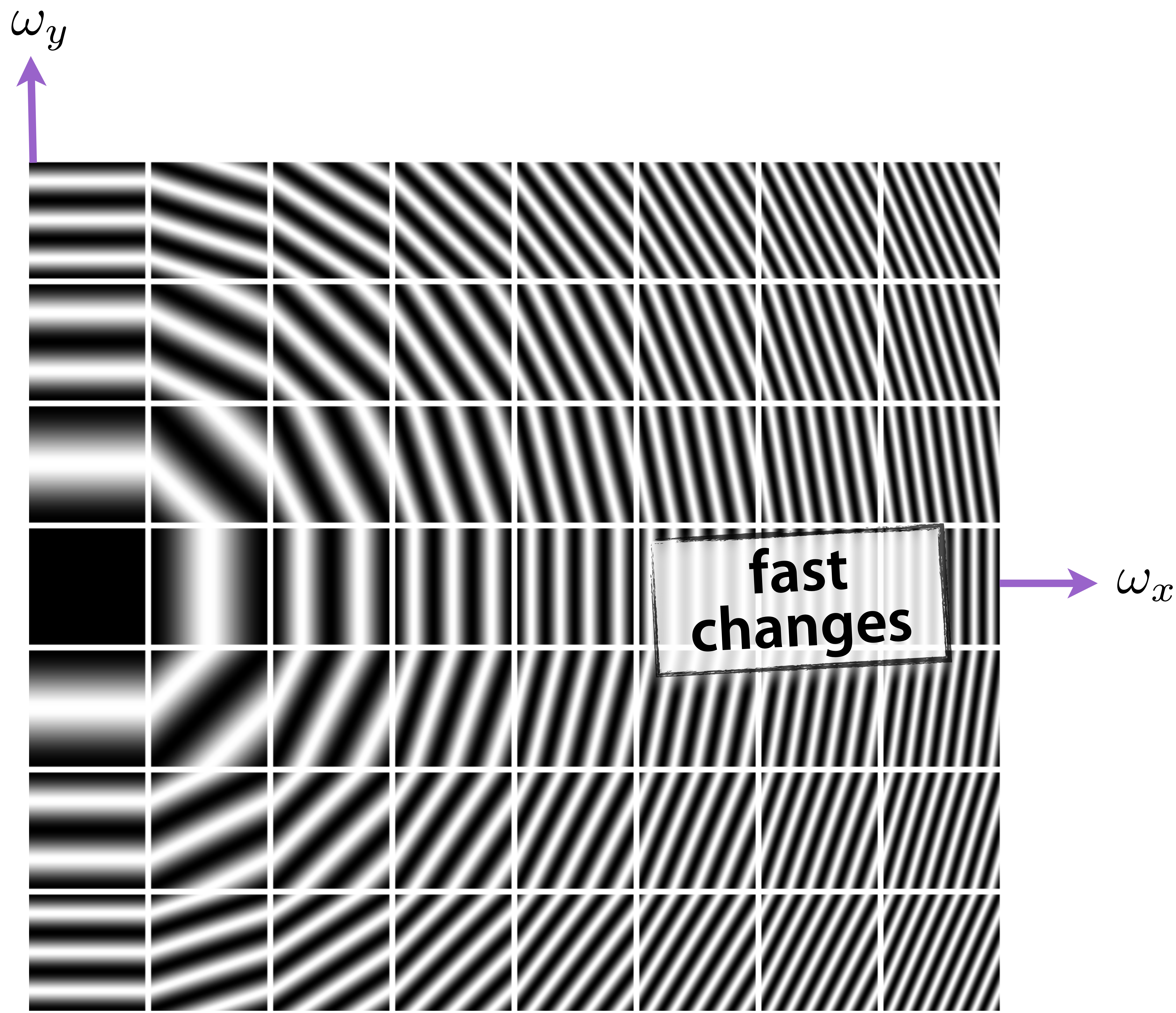


What about **2D?**

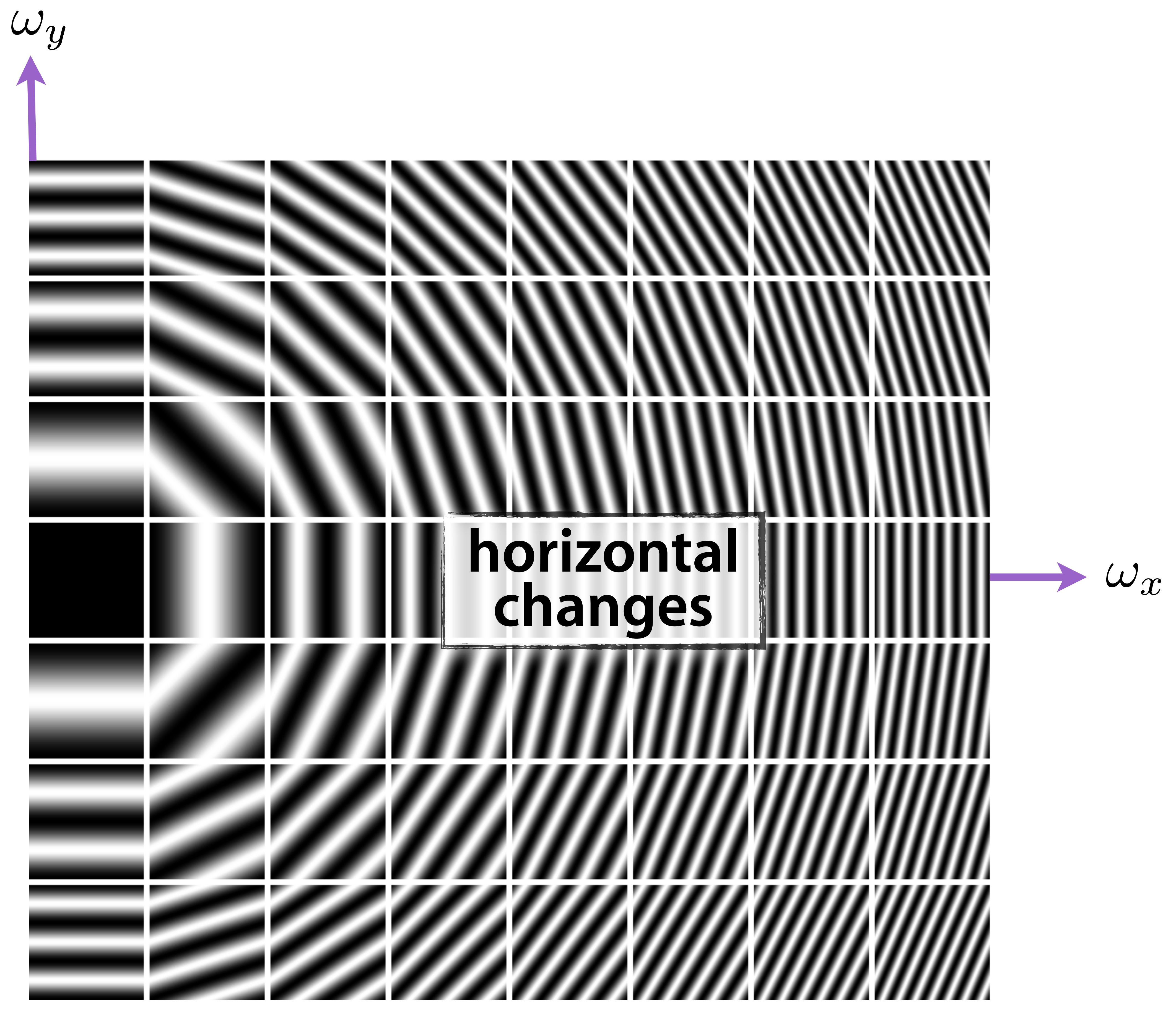


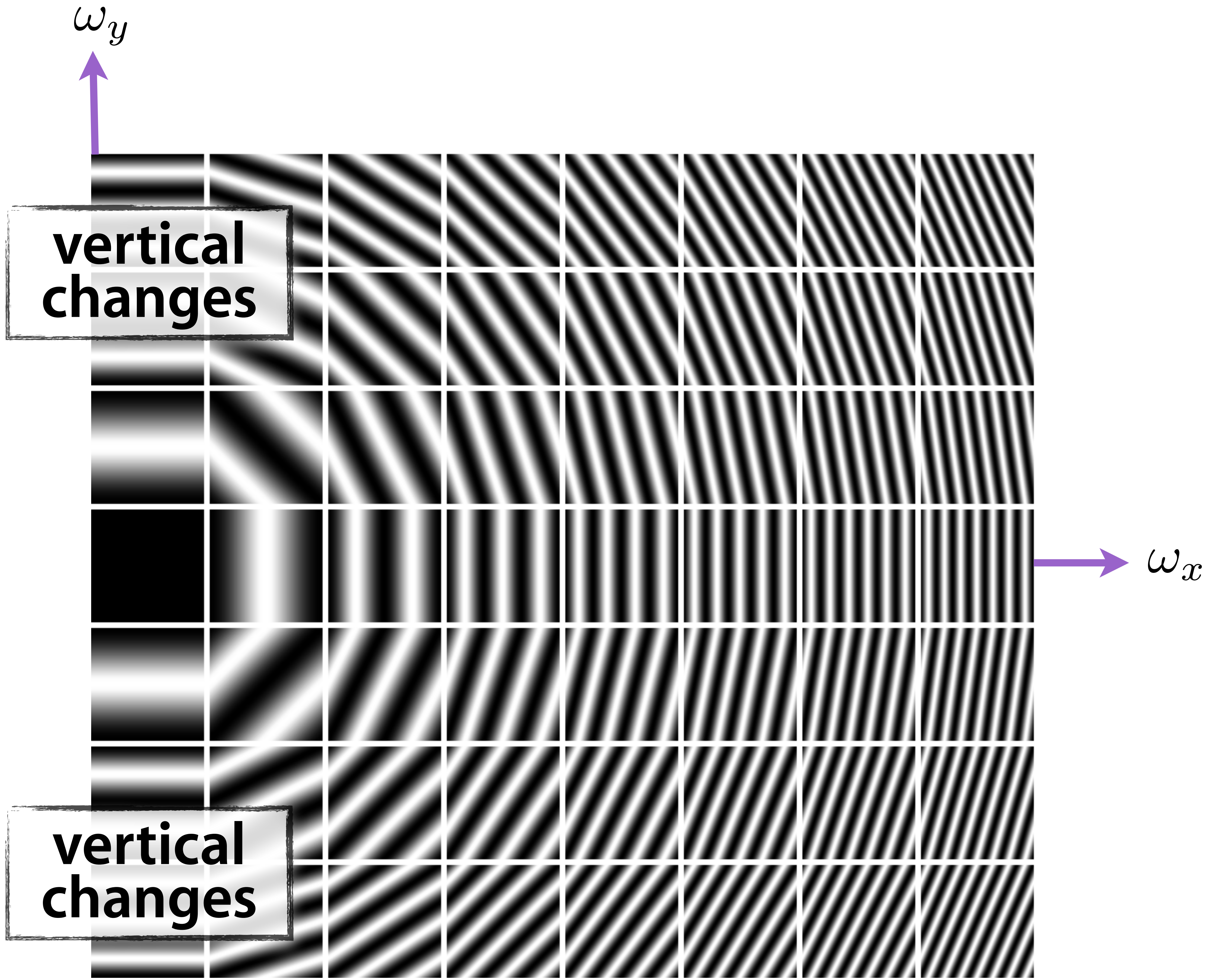
ω_y
ω_x
DC component


ω_y
ω_x
slow changes

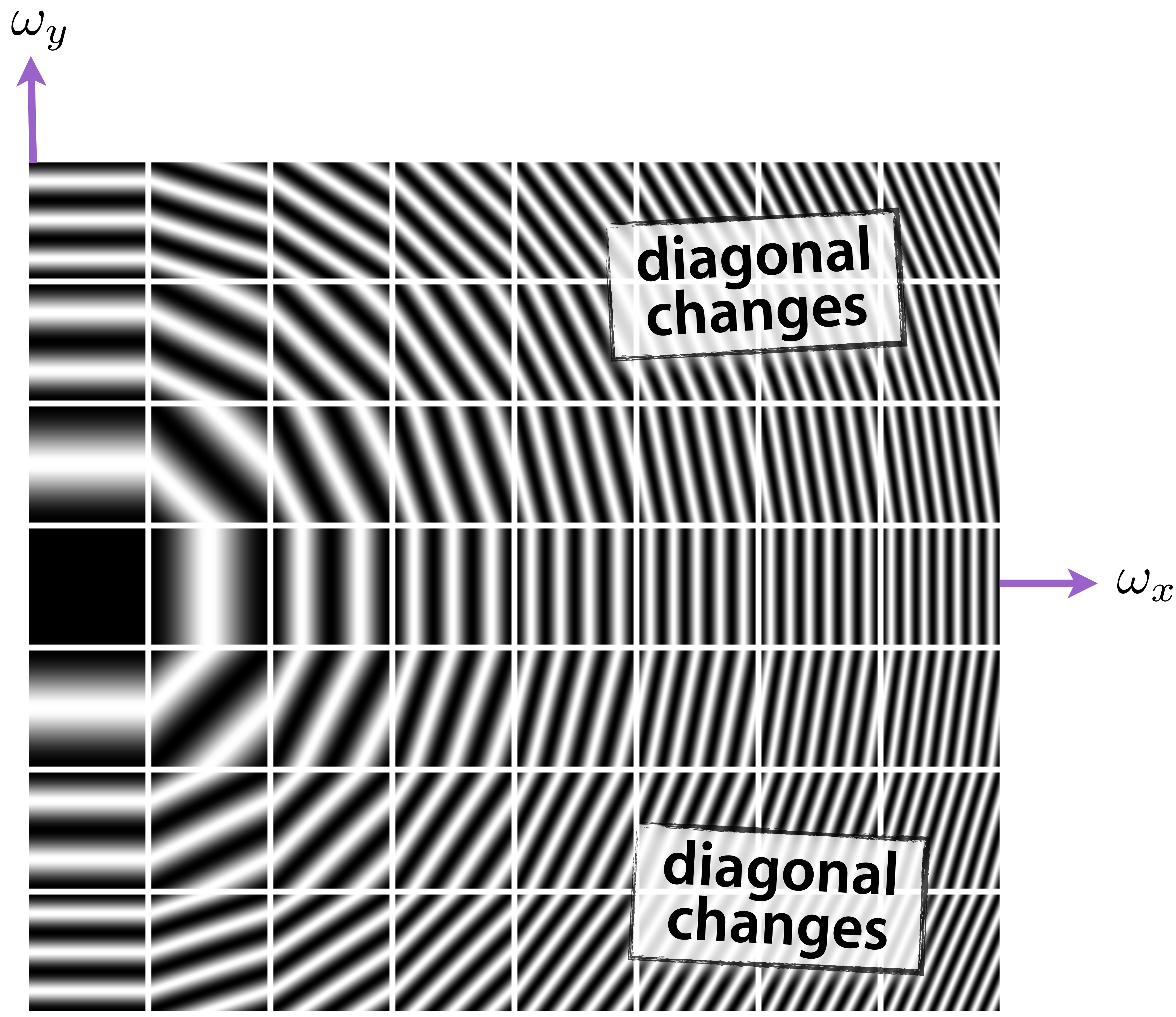


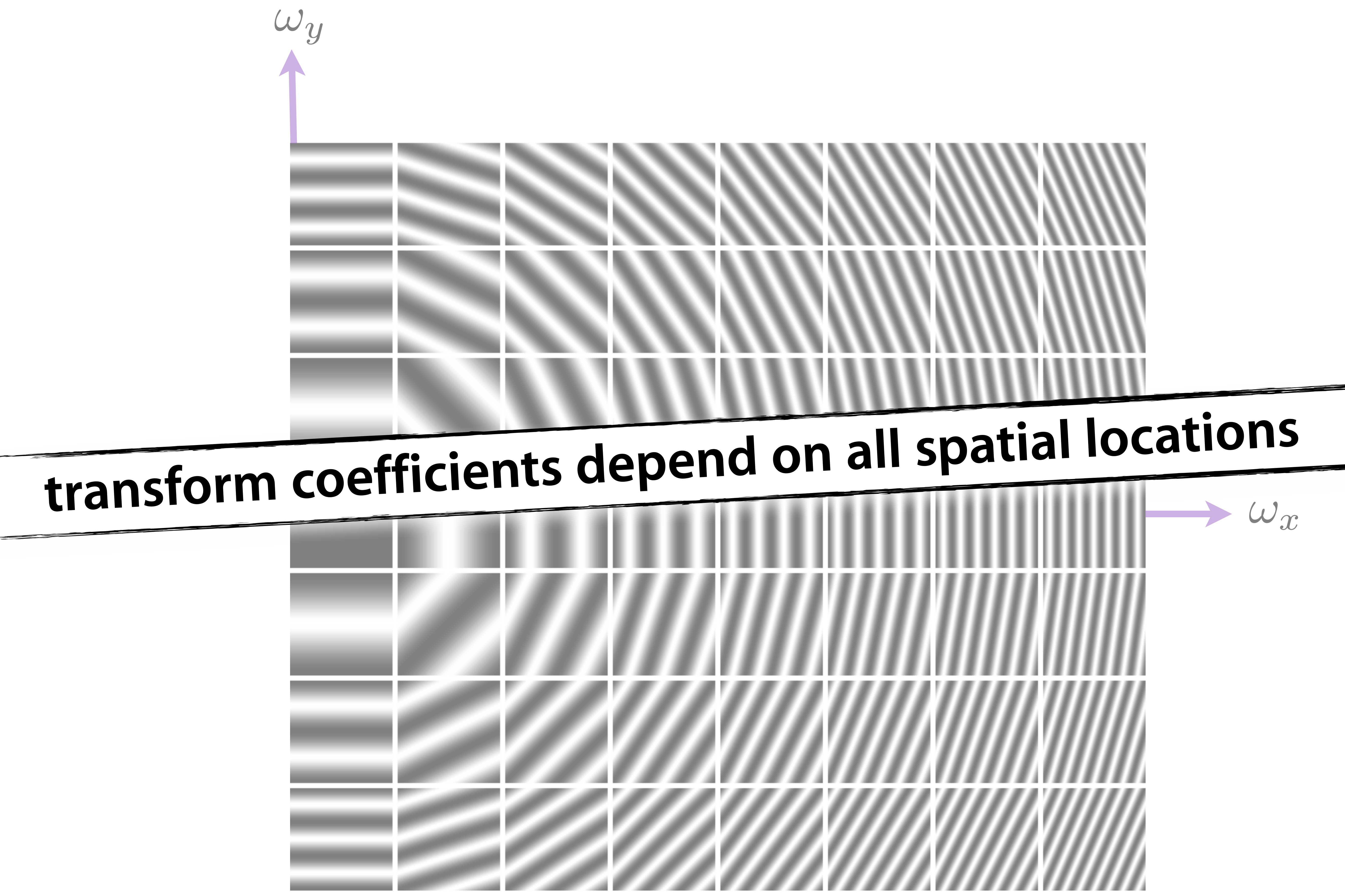
ω_y
ω_x
fast
changes


ω_y
horizontal changes
ω_x



ω_y
vertical changes
vertical changes
ω_x


ω_y
ω_x
diagonal changes
diagonal changes


ω_y
ω_x
transform coefficients depend on all spatial locations

2D Fourier Transform

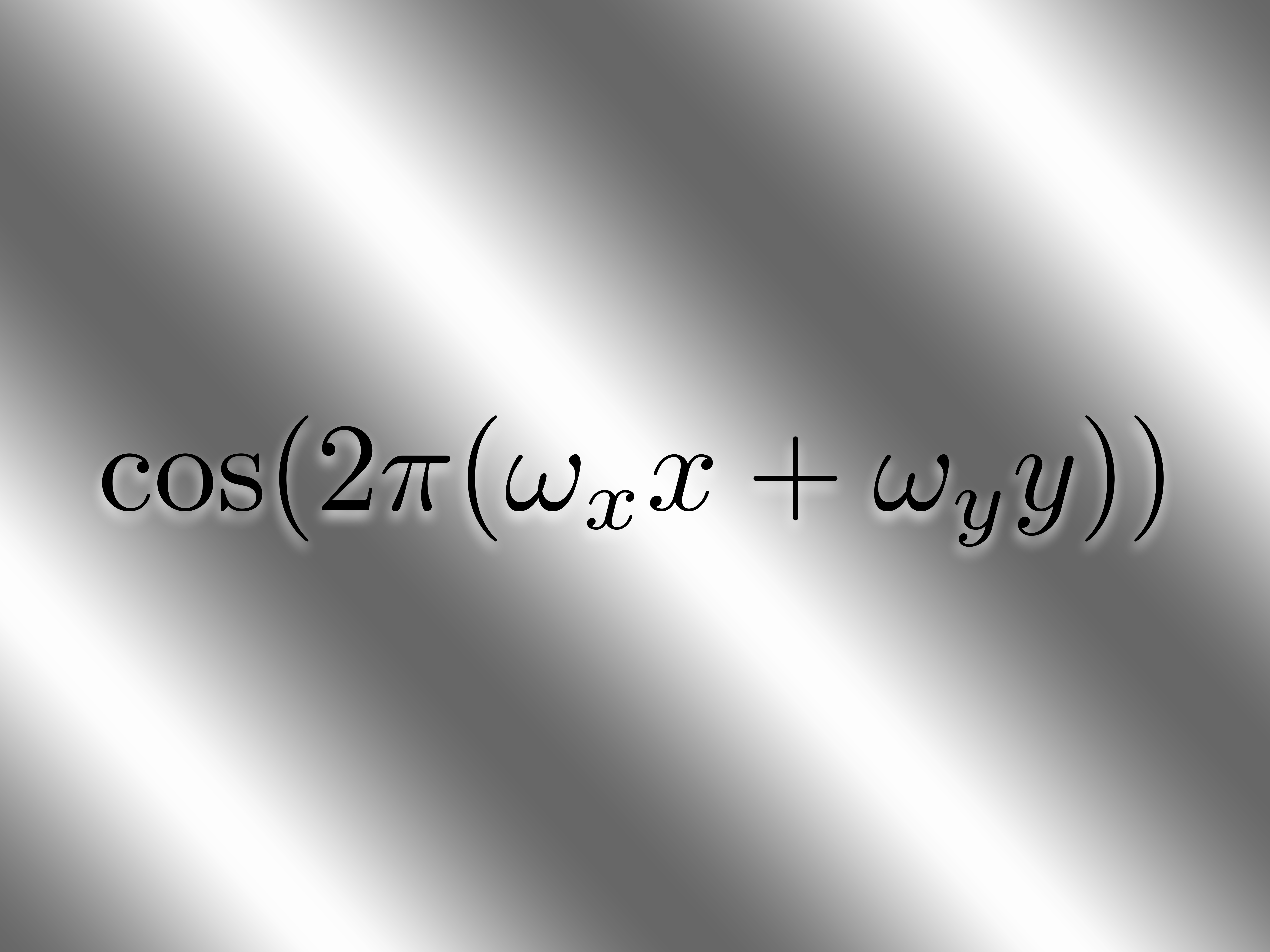
$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i(\omega_x x + \omega_y y)} dx dy$$

Euler's Identity

$$e^{2\pi(\omega_x x+\omega_y y)}$$

Euler's Identity

$$\cos(2\pi(\omega_x x + \omega_y y)) + i\sin(2\pi(\omega_x x + \omega_y y))$$



$$\cos(2\pi(\omega_x x + \omega_y y))$$

2D Fourier Transform

$$F(\omega_x, \omega_y) = \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} f(x, y) e^{-2\pi i(\omega_x x + \omega_y y)} dx dy$$

2D inverse Fourier Transform

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_x, \omega_y) e^{2\pi i(\omega_x x + \omega_y y)} d\omega_x d\omega_y$$

DFT

Discrete Fourier Transform

Euler's Identity

$$Ae^{ik} = A(\cos(k) + i\sin(k))$$

Discrete Fourier Transform (DFT)

$$F[u,v] = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x,y] e^{-2\pi i \left(x \frac{u}{M} + y \frac{v}{N}\right)}$$

where

$$u = 0, \ldots, M-1$$
$$v = 0, \ldots, N-1$$

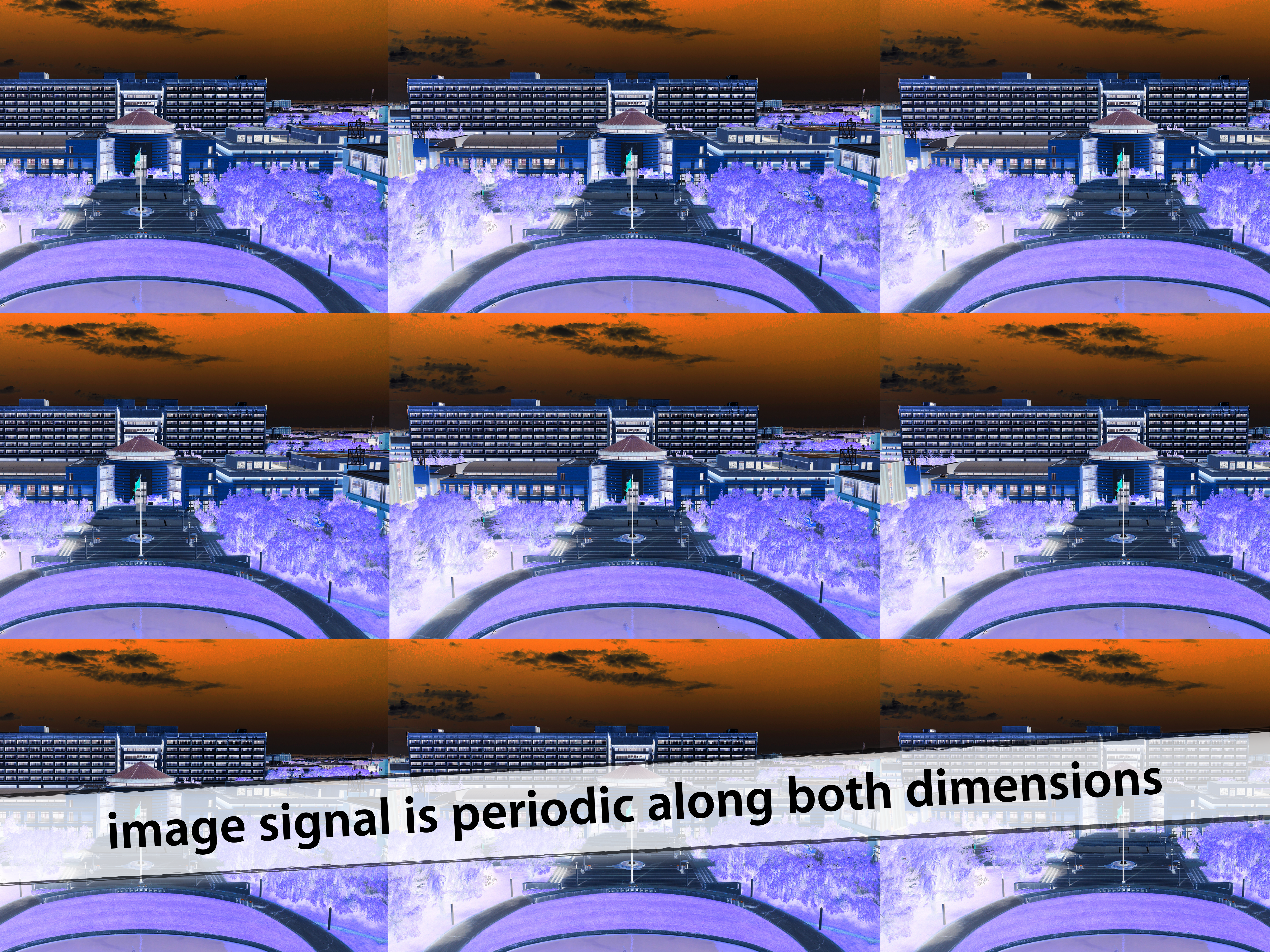
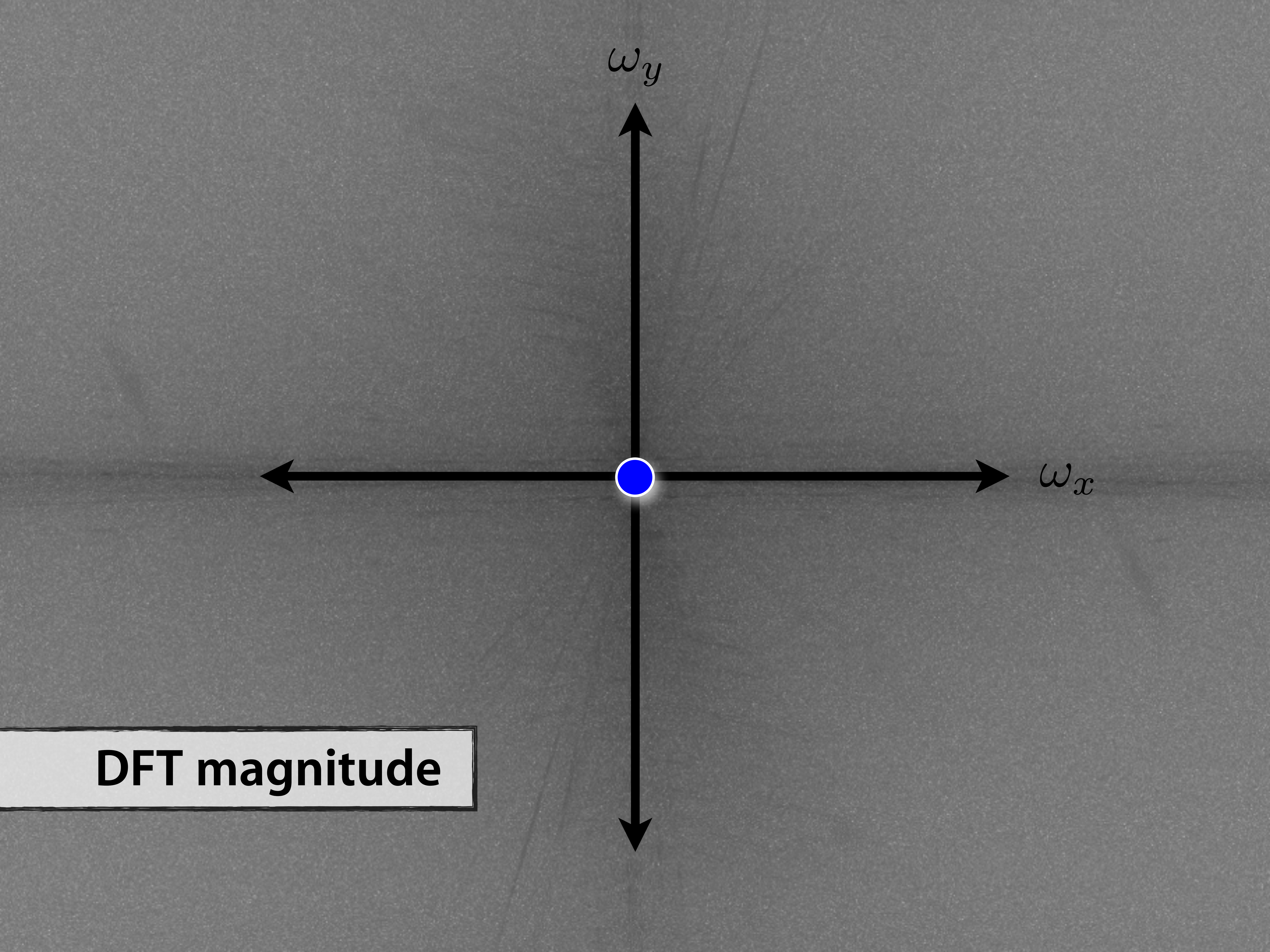
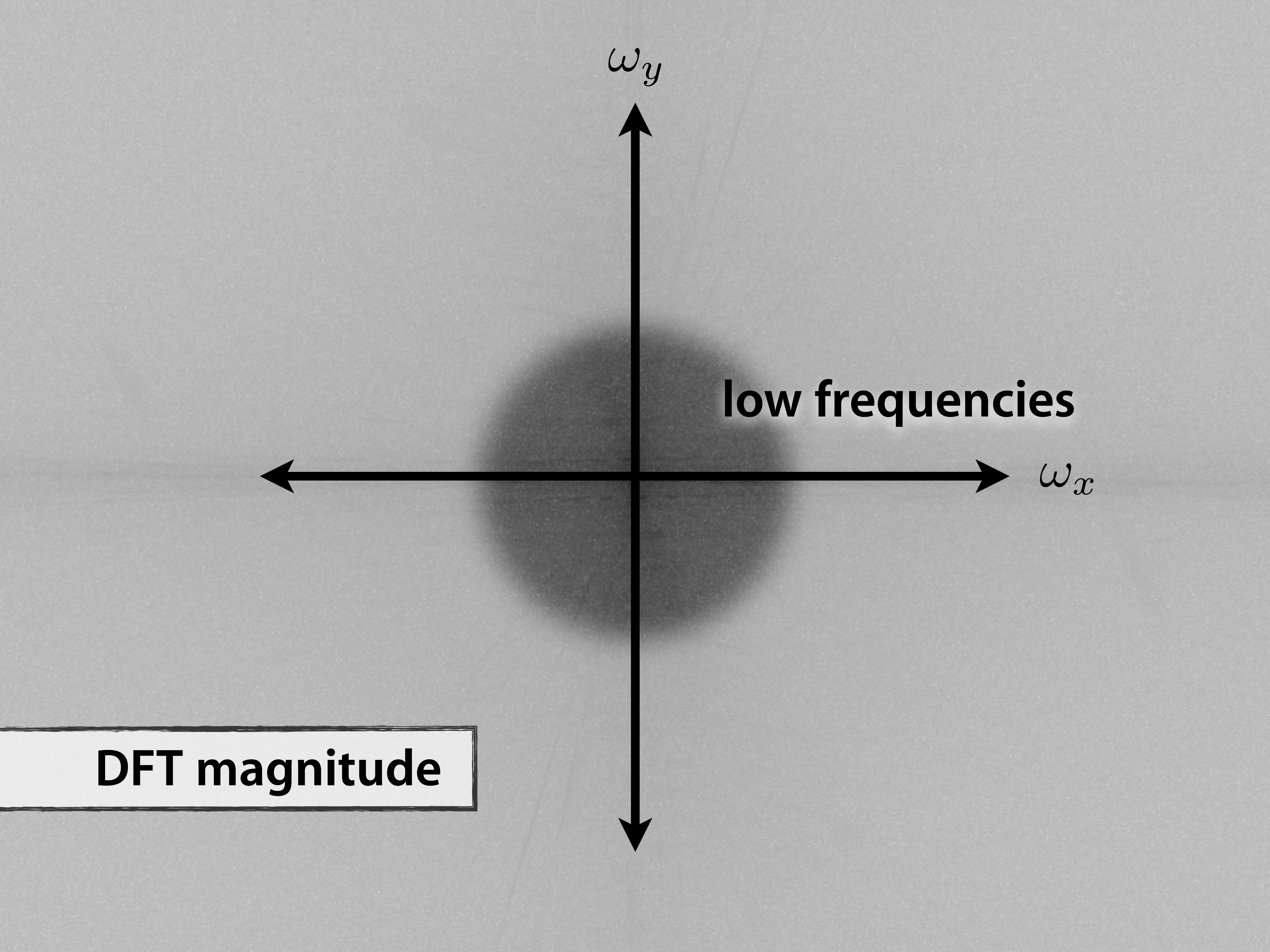


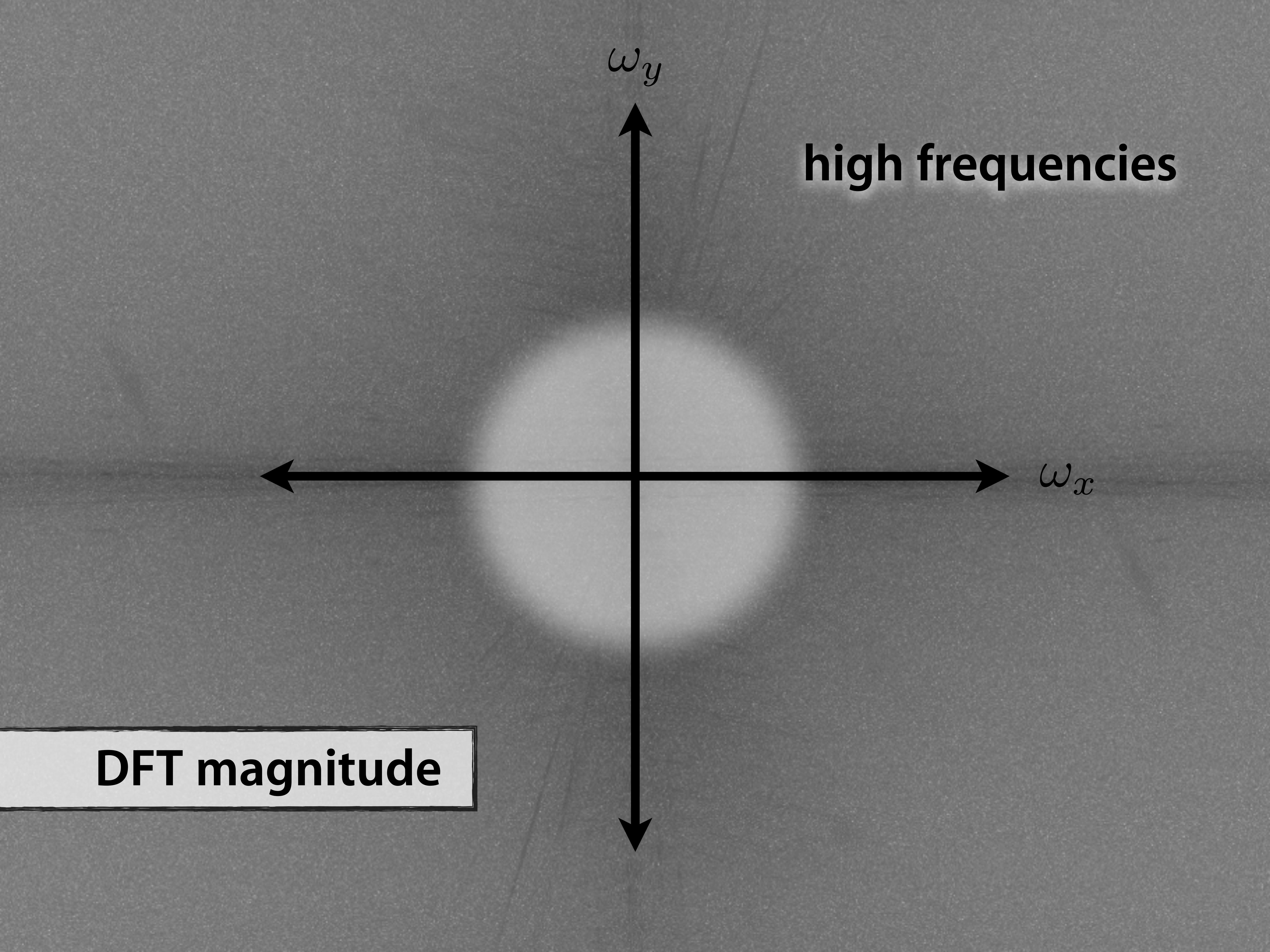
image signal is periodic along both dimensions

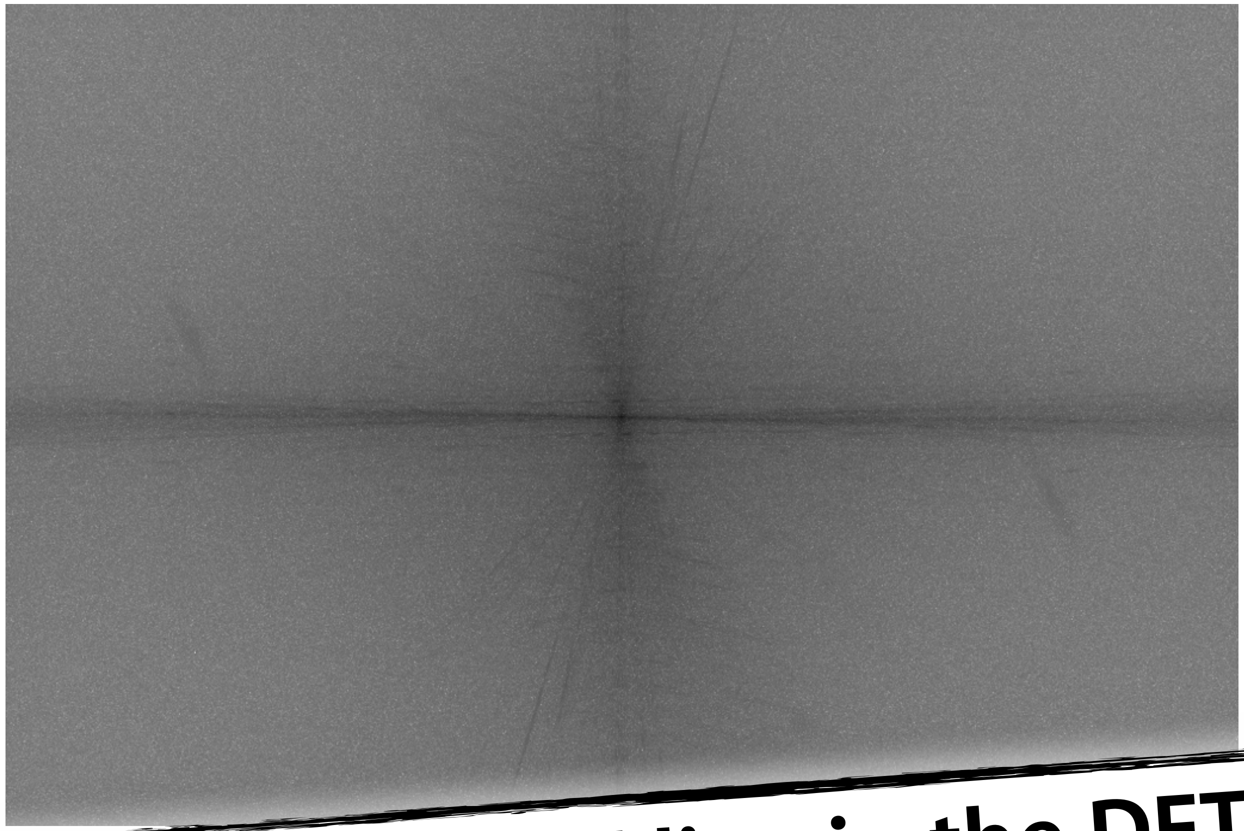


ω_y
ω_x
DFT magnitude



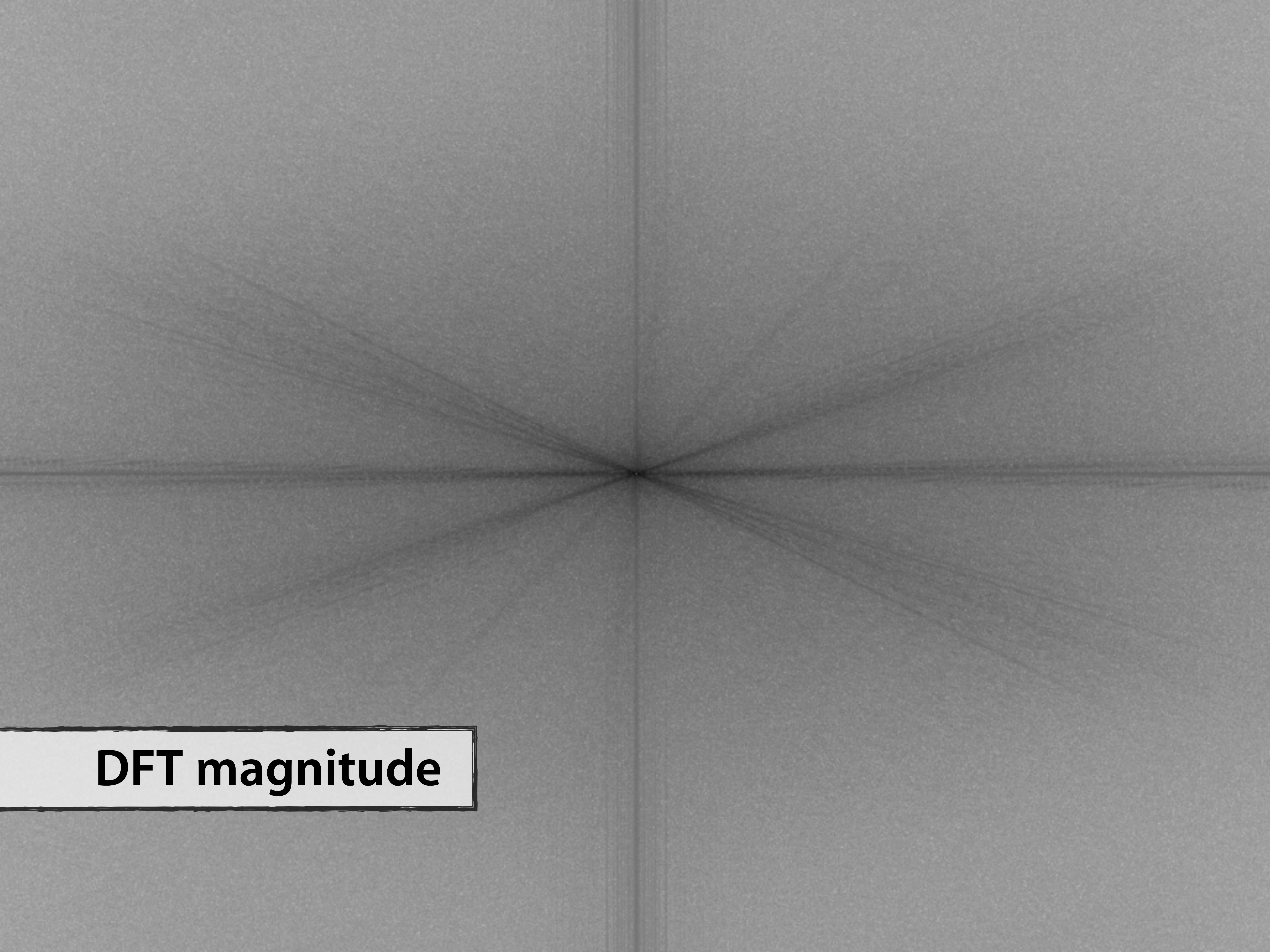
ω_y
low frequencies
ω_x
DFT magnitude

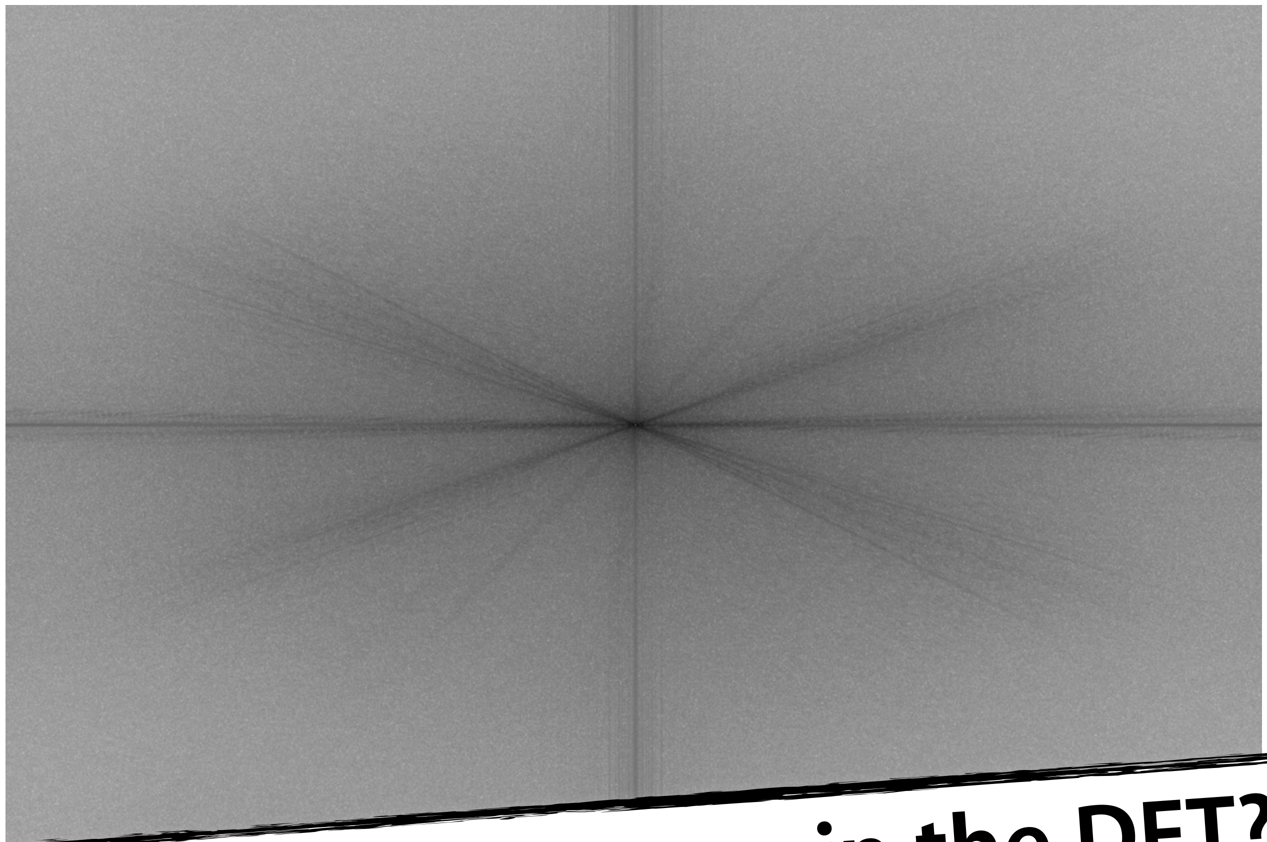

ω_y
high frequencies
ω_x
DFT magnitude



What is the source in the input of the horizontal line in the DFT?

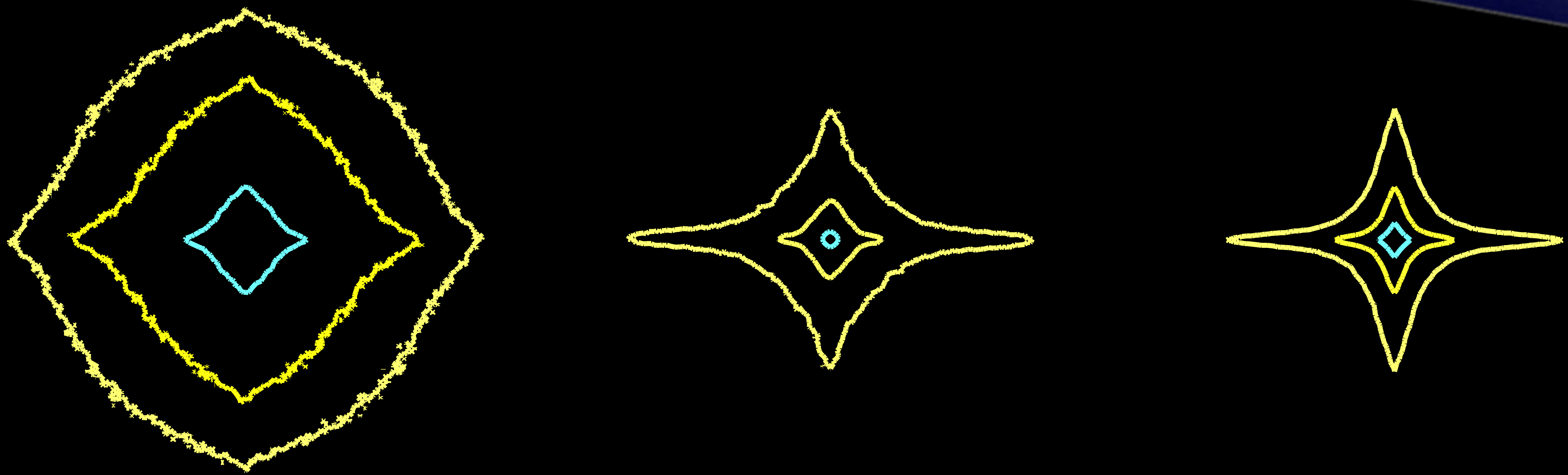

input image


DFT magnitude

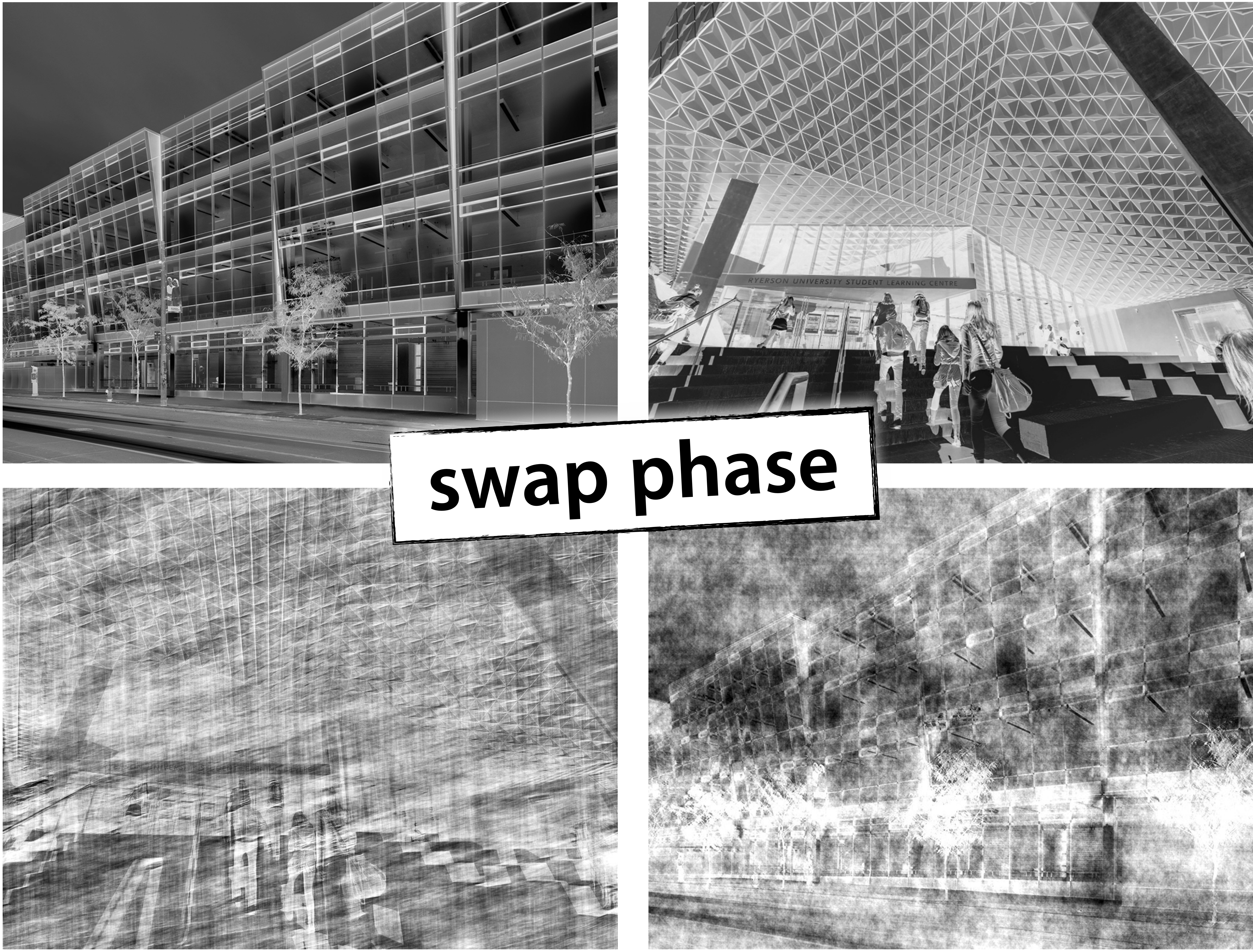


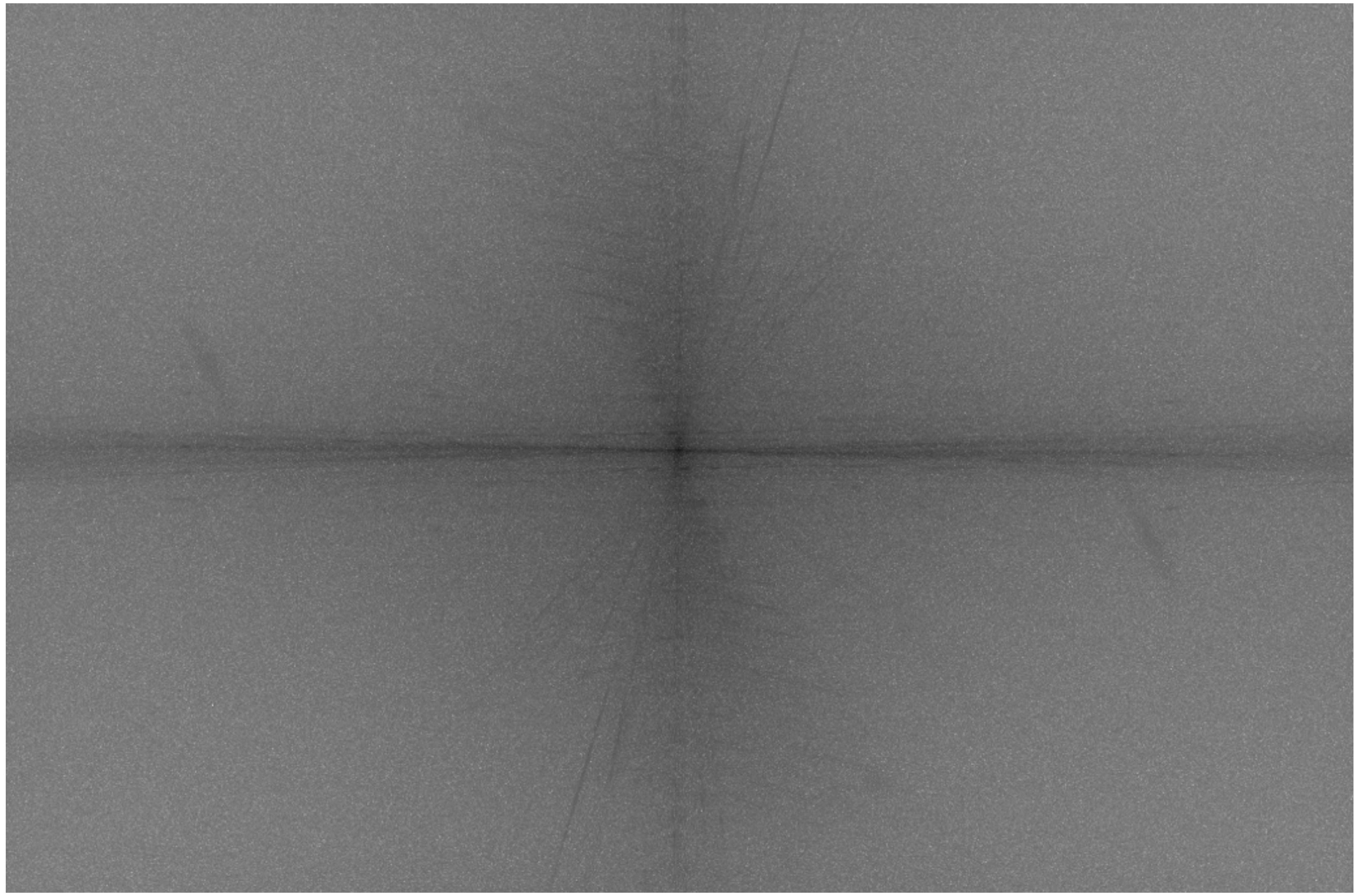
What is the source in the input of the line structures in the DFT?

average magnitude spectrum

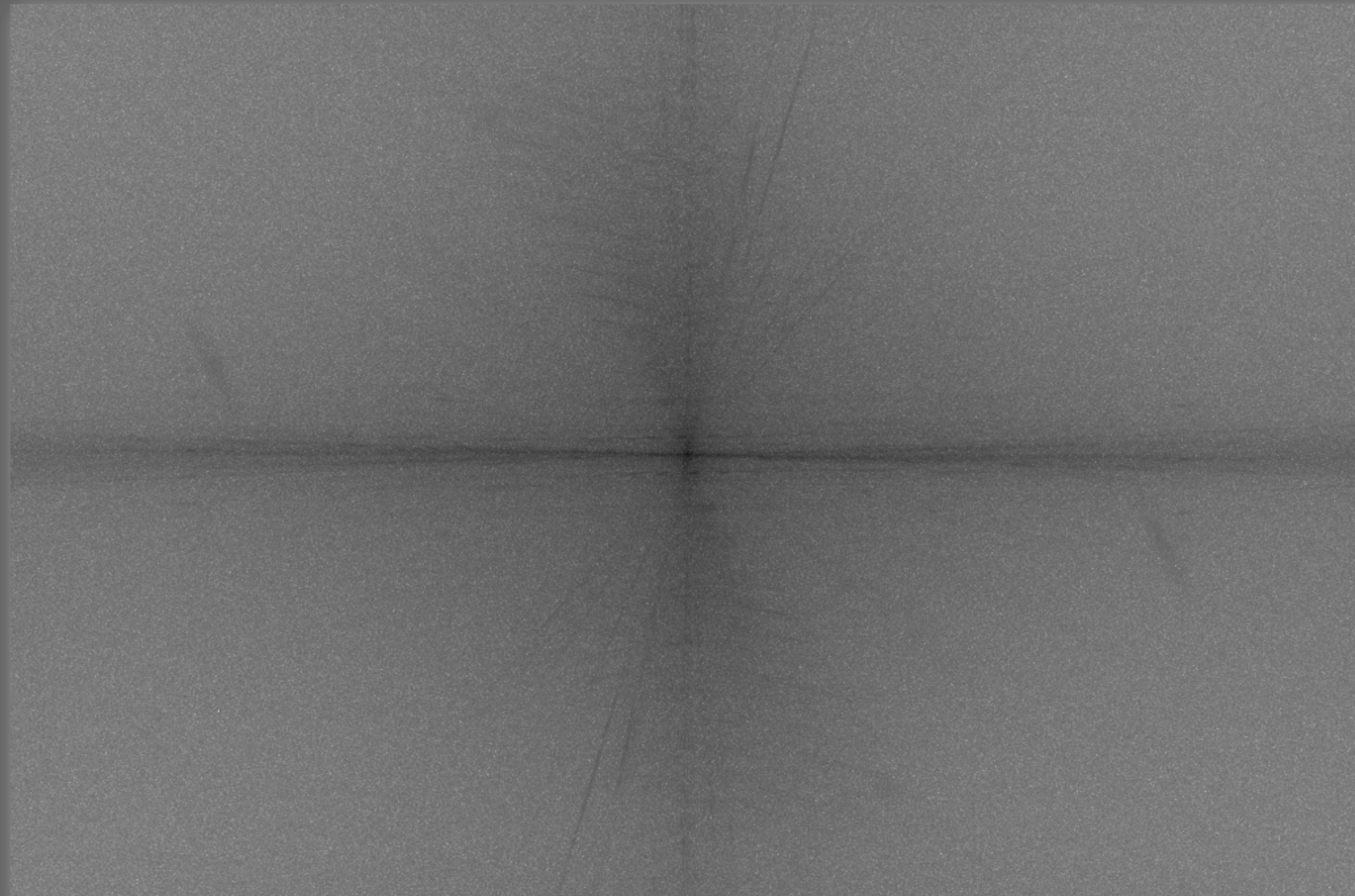


Forest **Street** **Indoors**


RYERSON UNIVERSITY STUDENT LEARNING CENTRE
swap phase

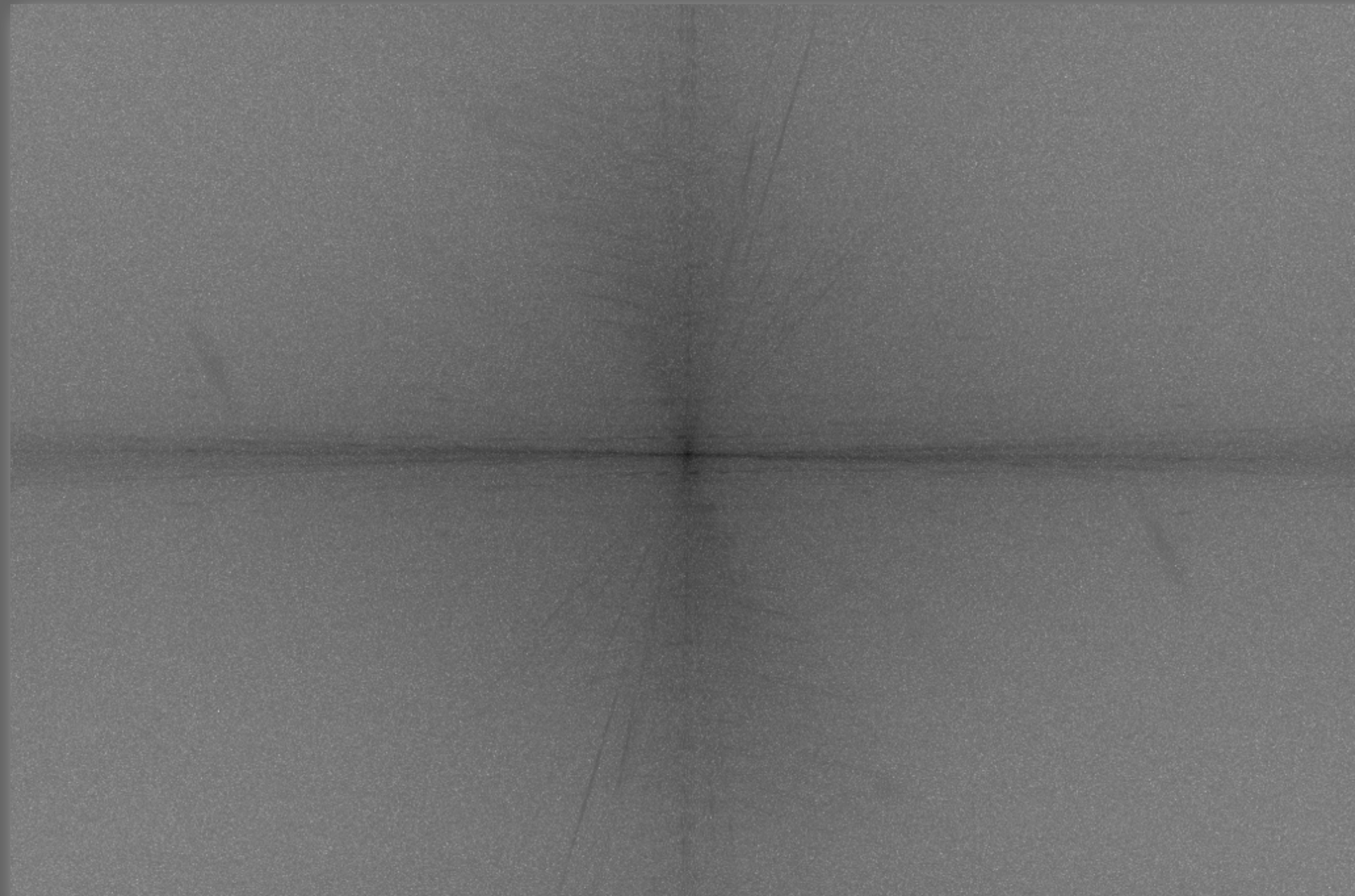


```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(1 + abs(img_fft));
>> imshow(fftshift(img_fft),[])
```



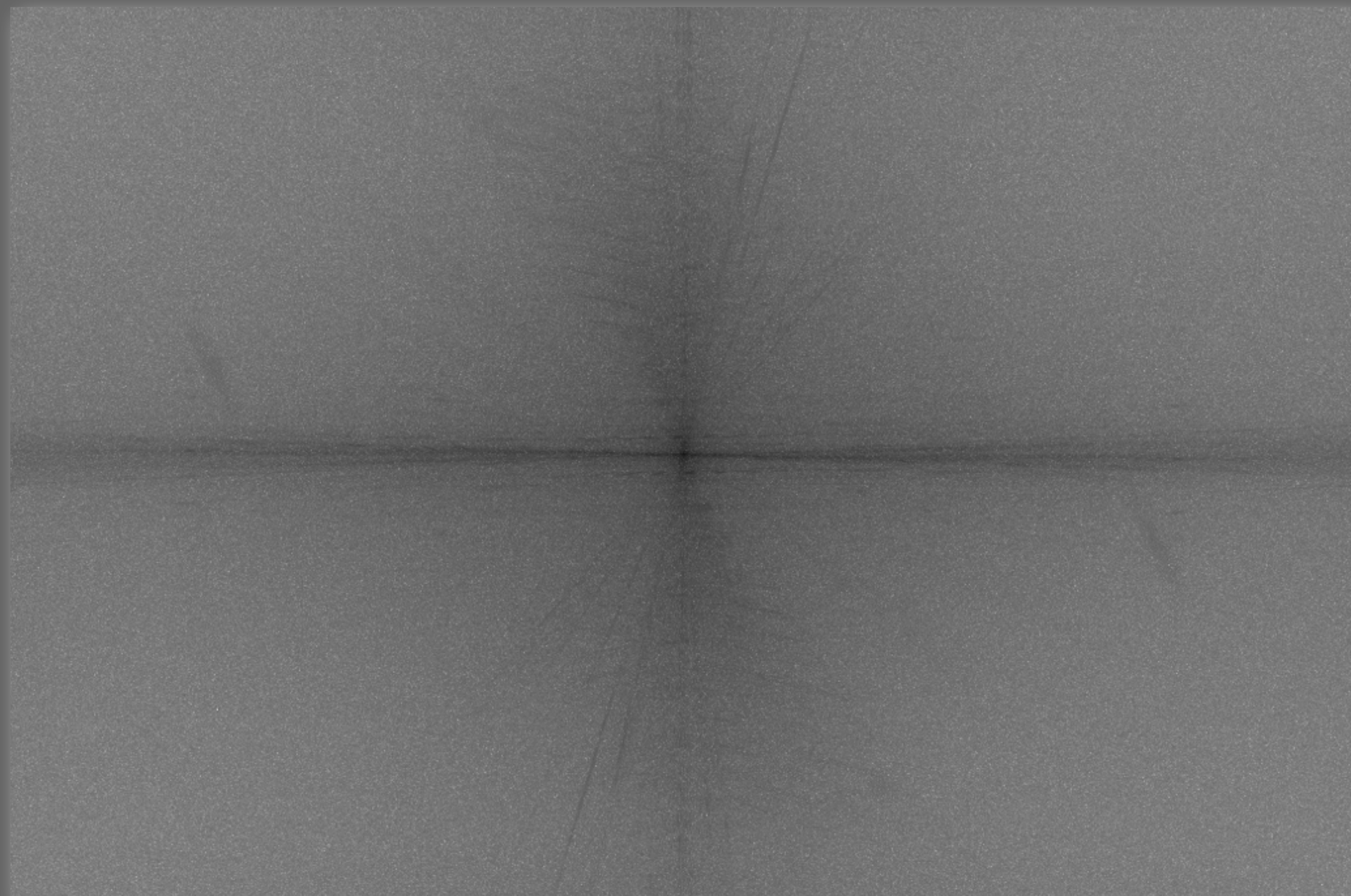
```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> imshow(fftshift(img_fft),[])
```

two-dimensional discrete Fourier transform

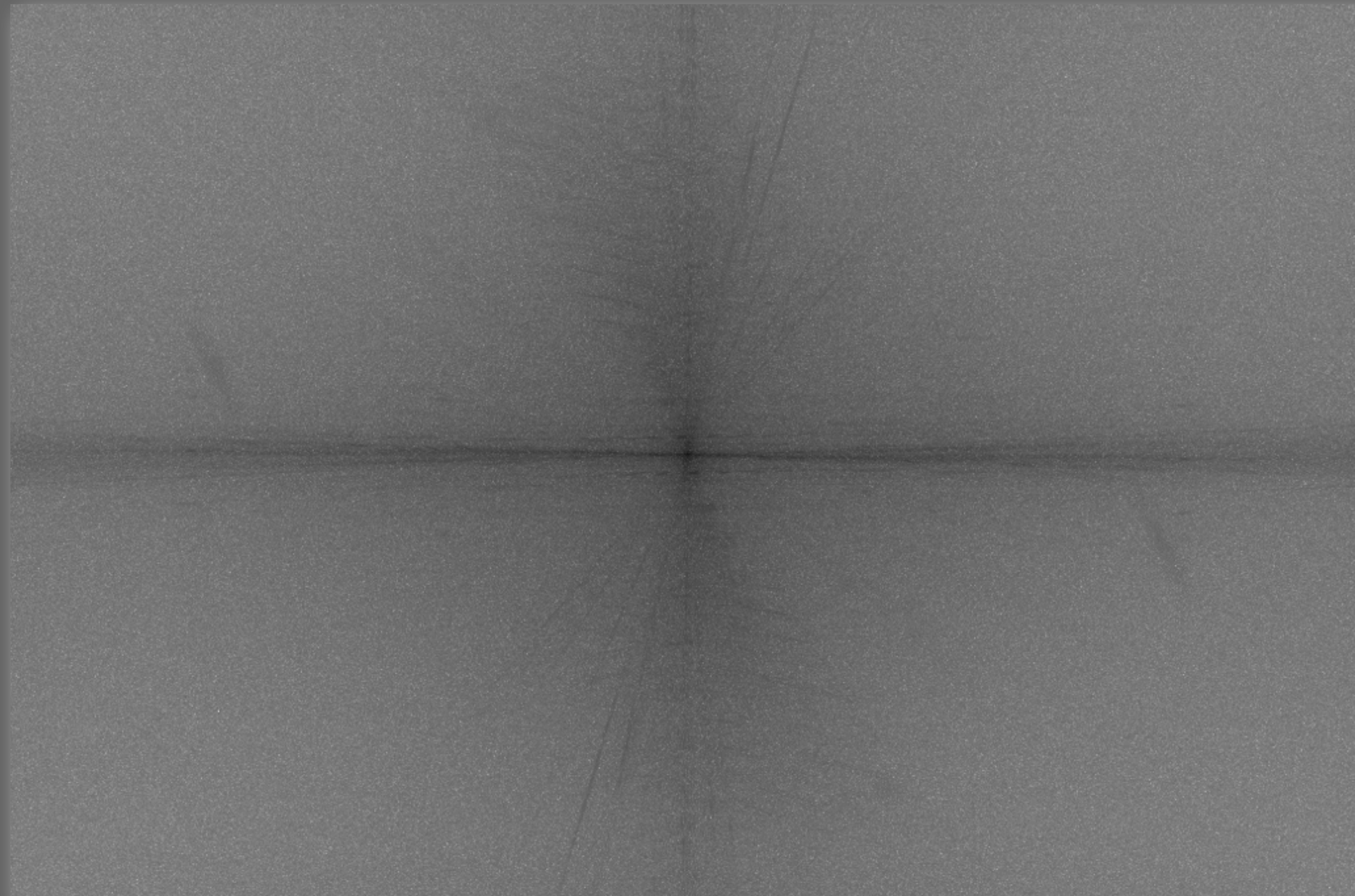


```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
```

remove DC component to improve visualization

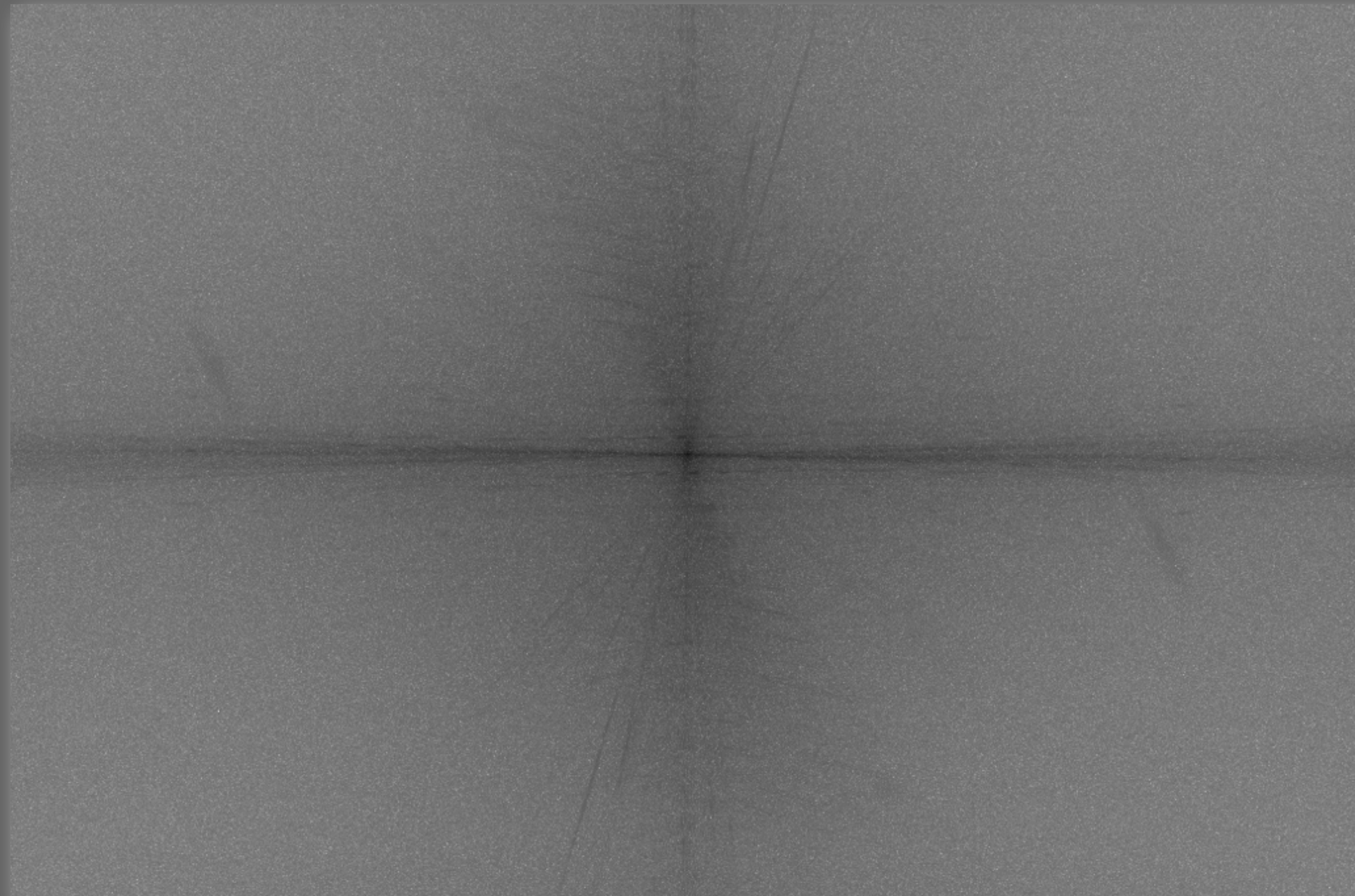


```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(1 + abs(img_fft));
>> imshow(fftshift(img_fft),[])
```



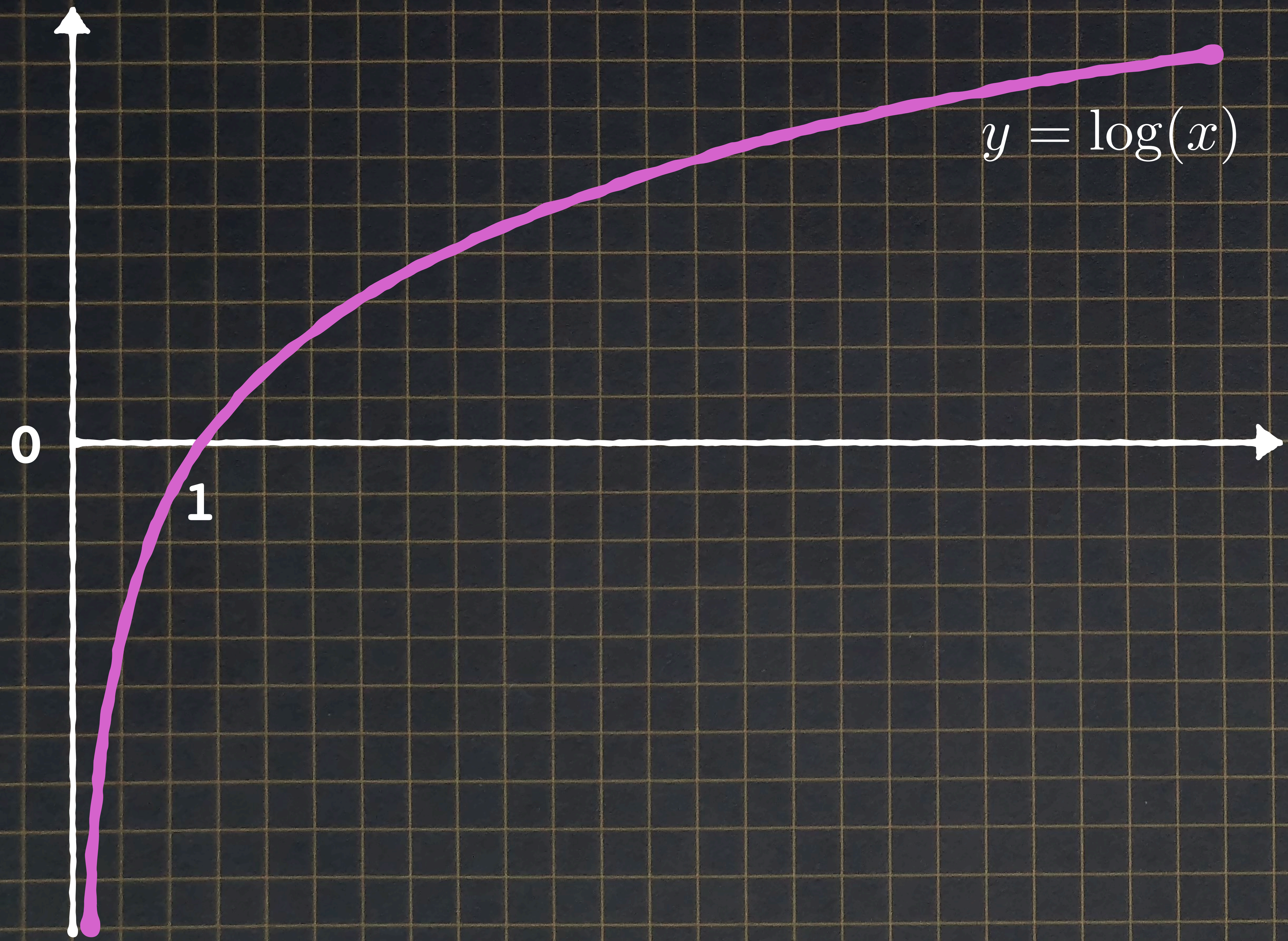
```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(1 + abs(img_fft));
>> imshow(fftshift(img_fft),[])
```

magnitude spectrum

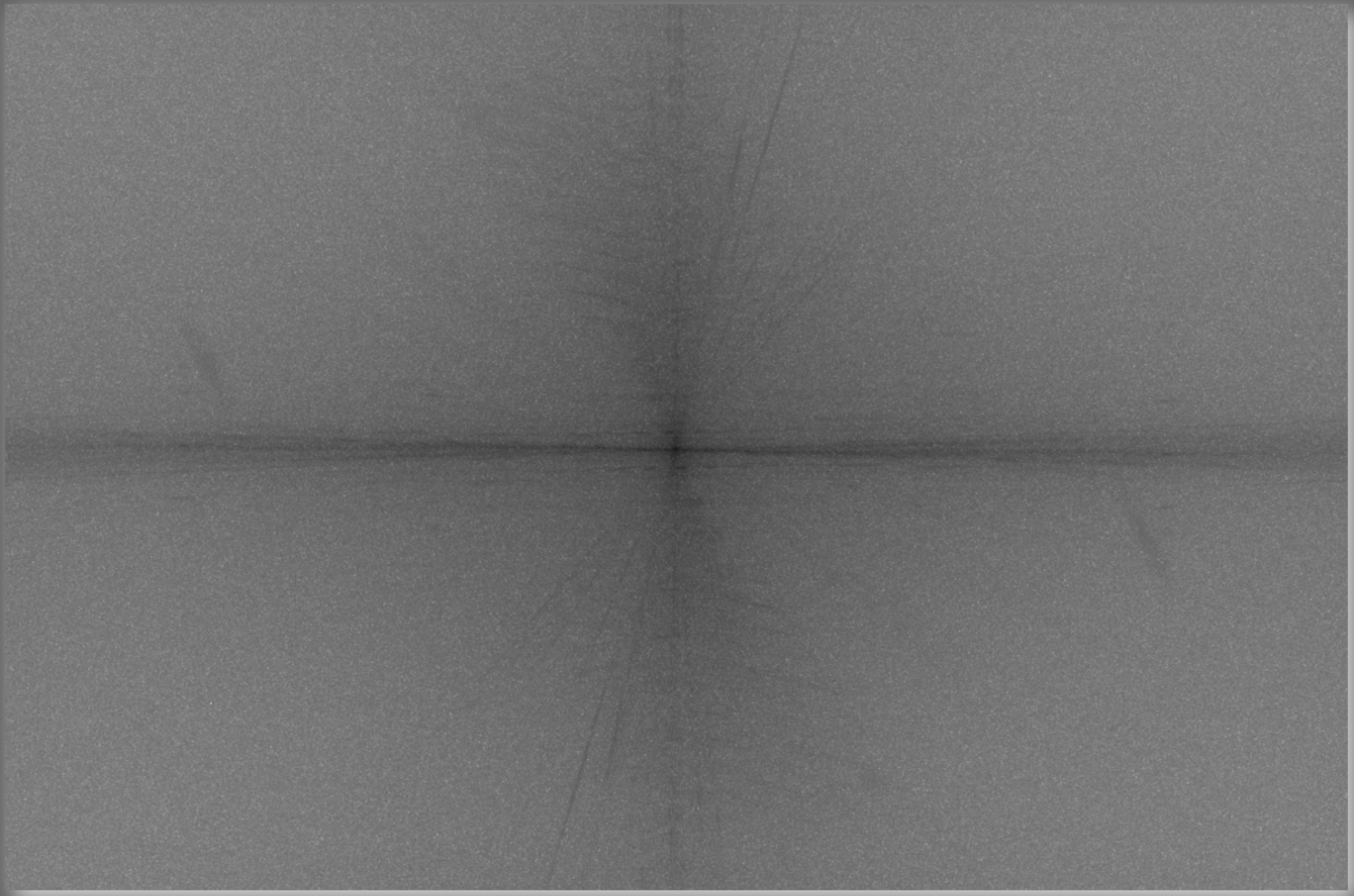


```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(1 + abs(img_fft));
>> imshow(fftshift(img_fft),[])
```

improves visualization

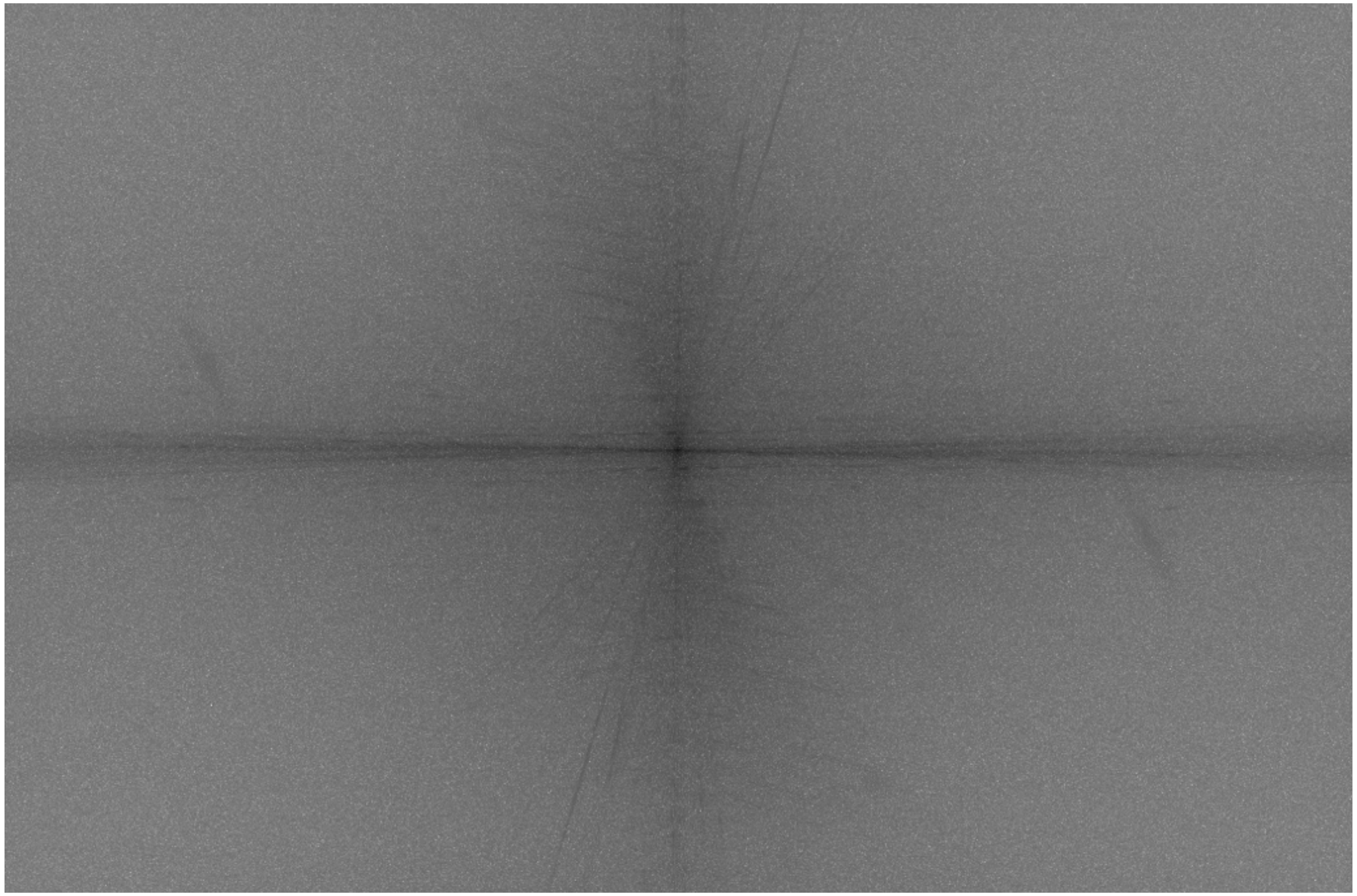


0
1
$y = \log(x)$

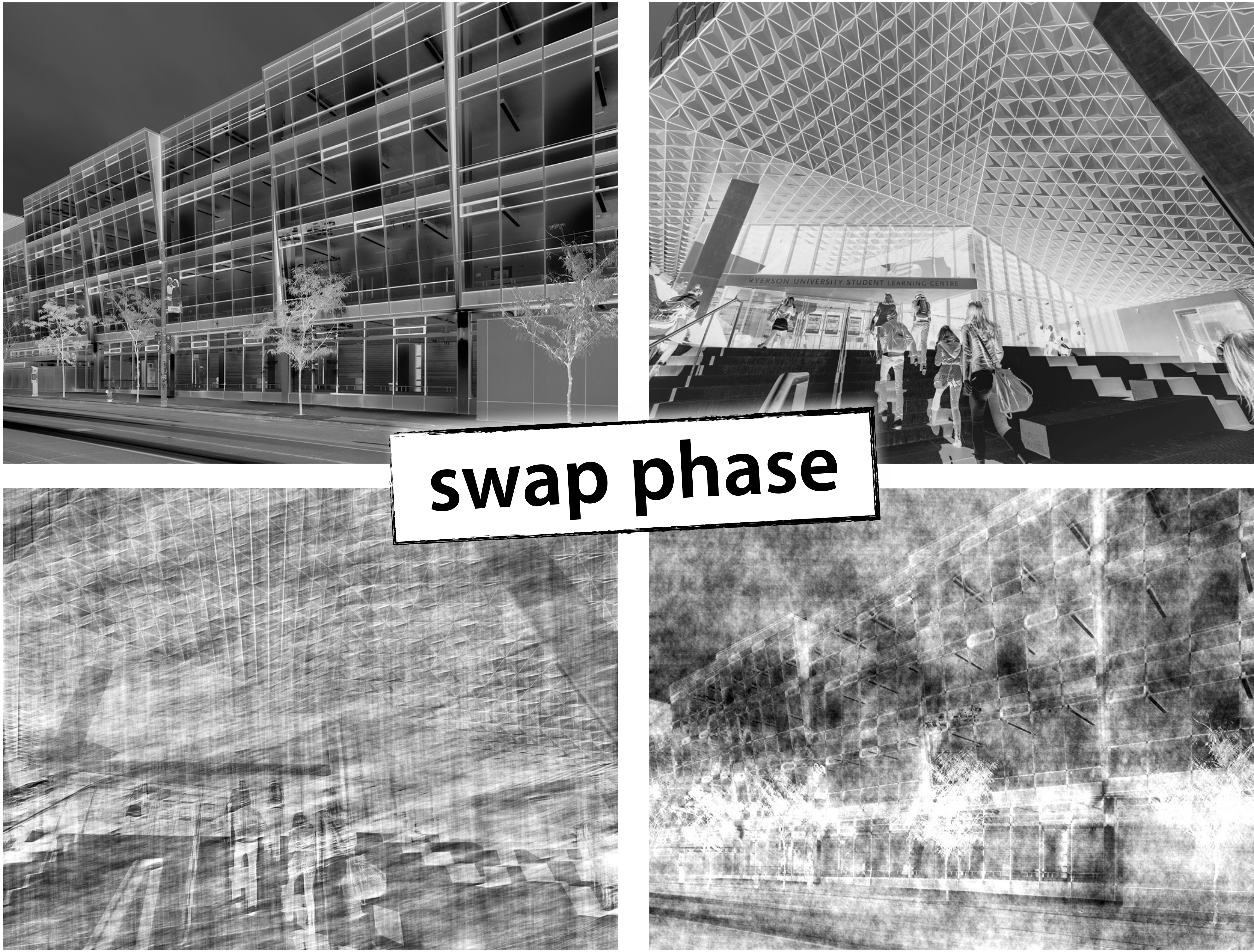


```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(abs(img_fft));
>> imshow(fftshift(img_fft),[])
```

rearrange spectrum



```
>> img = double(rgb2gray(imread('Toronto.jpg')));
>> img_fft = fft2(img);
>> img_fft(1,1) = 0;
>> img_fft = log(1 + abs(img_fft));
>> imshow(fftshift(img_fft),[])
```


swap phase
RYERSON UNIVERSITY STUDENT LEARNING CENTRE

```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
>> abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
>> abs2_phase1 = abs(i2_fft).*exp(i*angle(i1_fft));
>> i_abs1_phase2 = real(ifft2(abs1_phase2));
>> i_abs2_phase1 = real(ifft2(abs2_phase1));
```

```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
>> abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
abs(i2_fft).*exp(i*angle(i1_fft));
>> i_abs2_phase1 = real(ifft2(abs2_phase1));
```

combine magnitude of one image with phase of other

Euler's Identity

$$Ae^{ik} = A(\cos(k) + i\sin(k))$$

```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
>> abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
>> abs2_phase1 = abs(i2_fft).*exp(i*angle(i1_fft));
>> i_abs2_phase1 = real(ifft2(abs2_phase1));
```

combine magnitude of one image with phase of other

```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
>> abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
>> abs2_phase1 = ab
(abs1_phase2));
>> i_abs2_phase1 = real(ifft2(abs2_phase1));
```

compute magnitude component of first image

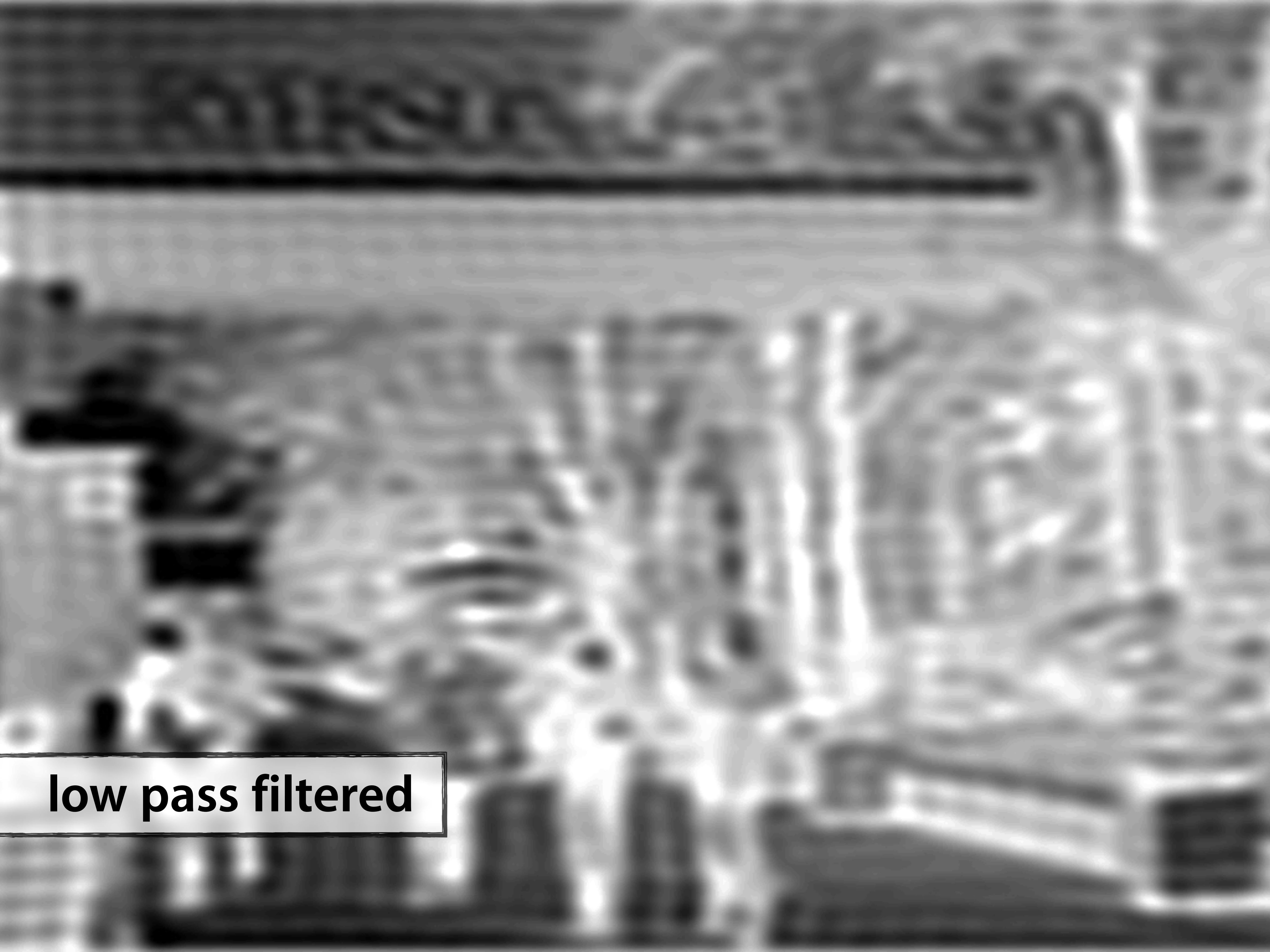
```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
>> i_abs1_phase2 = real(ifft2(abs1_p
>> i_abs2_phase1 = real(ifft2(abs2_phase1));
```

compute phase component of second image

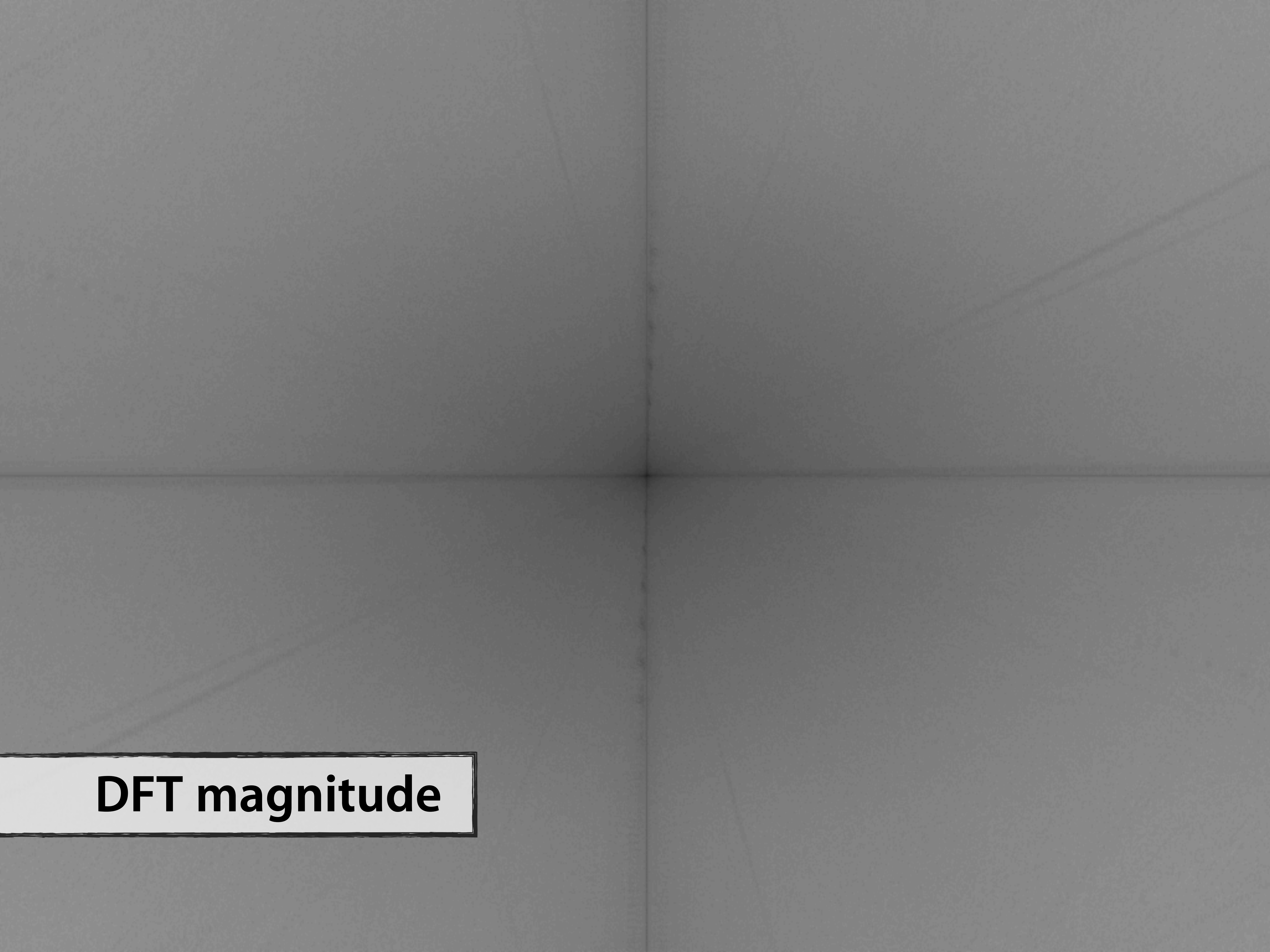
```
>> i1 = double(imread('SLC.png'));
>> i2 = double(imread('ENG.png'));
>> i1_fft = fft2(i1);
>> i2_fft = fft2(i2);
>> abs1_phase2 = abs(i1_fft).*exp(i*angle(i2_fft));
>> abs2_phase1 = abs(i2_fft).*exp(i*angle(i1_fft));
>> i_abs1_phase2 = real(ifft2(abs1_phase2));
```

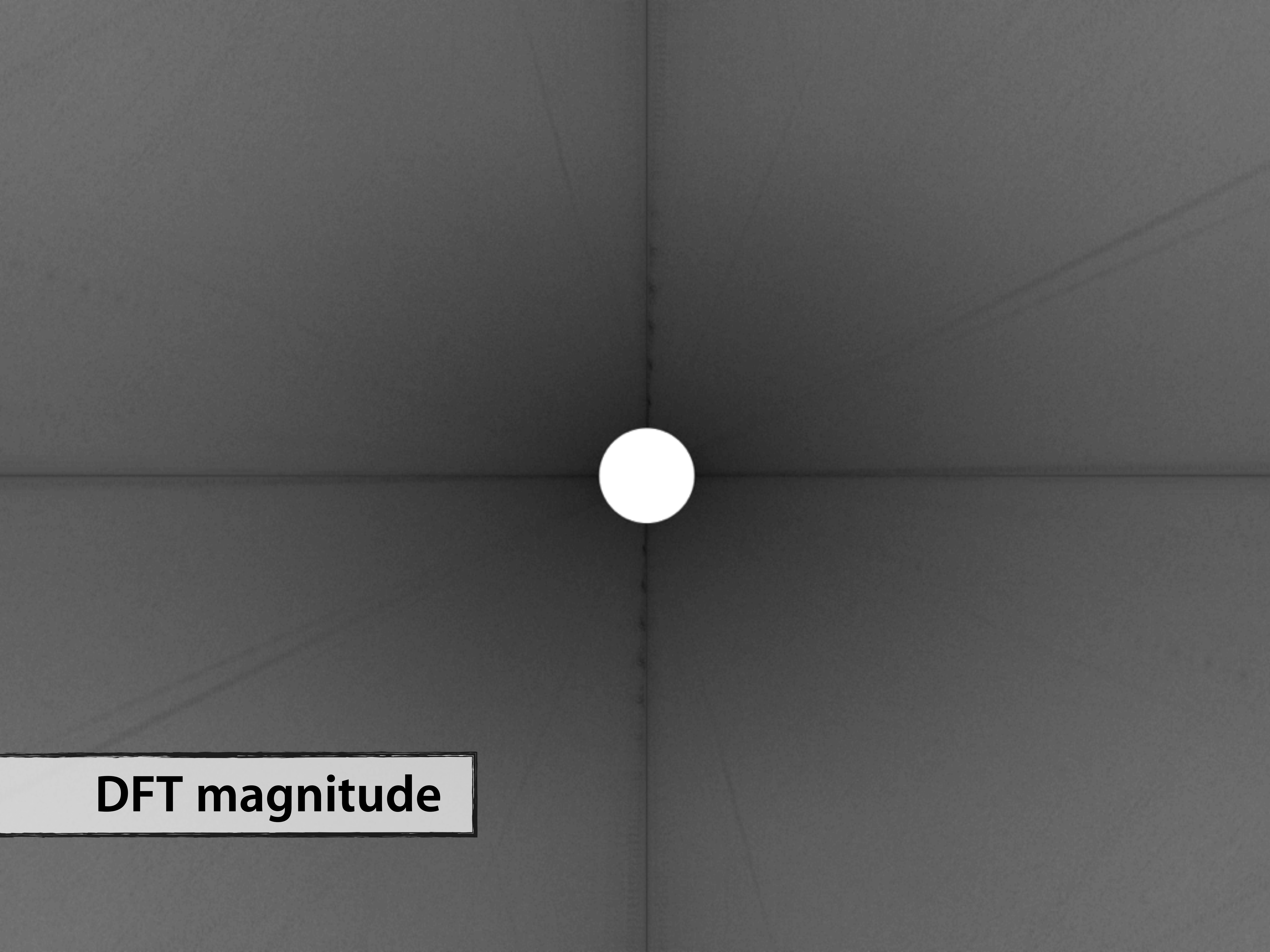
two-dimensional inverse discrete Fourier transform

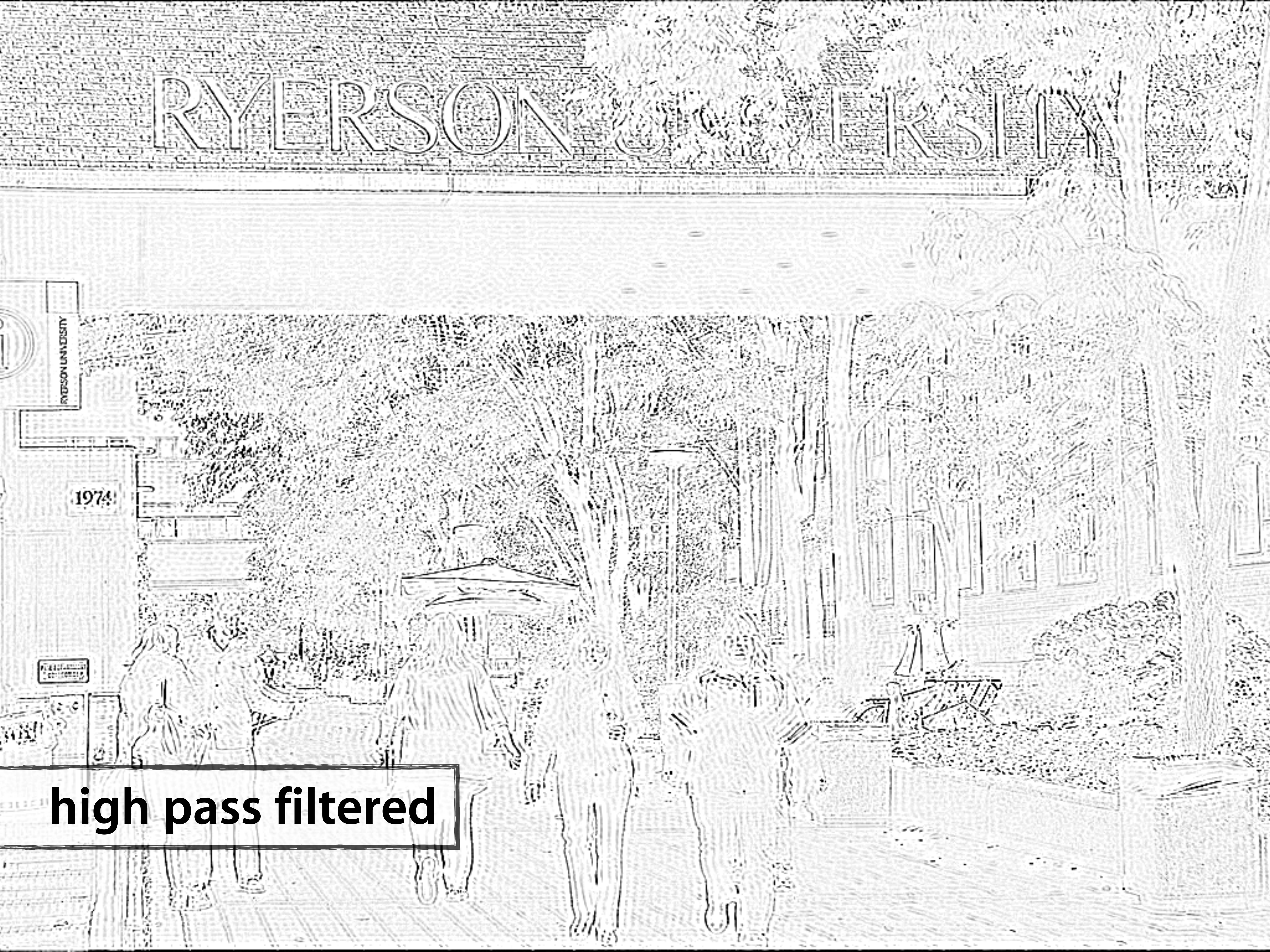
DFT magnitude


low pass filtered

input image


DFT magnitude


DFT magnitude



high pass filtered

FFT

Fast Fourier Transform

$$O(N^2)$$

time complexity of Discrete Fourier Transform

$$O(N \log N)$$

time complexity of Fast Fourier Transform

Match the spatial domain image to the Fourier magnitude image