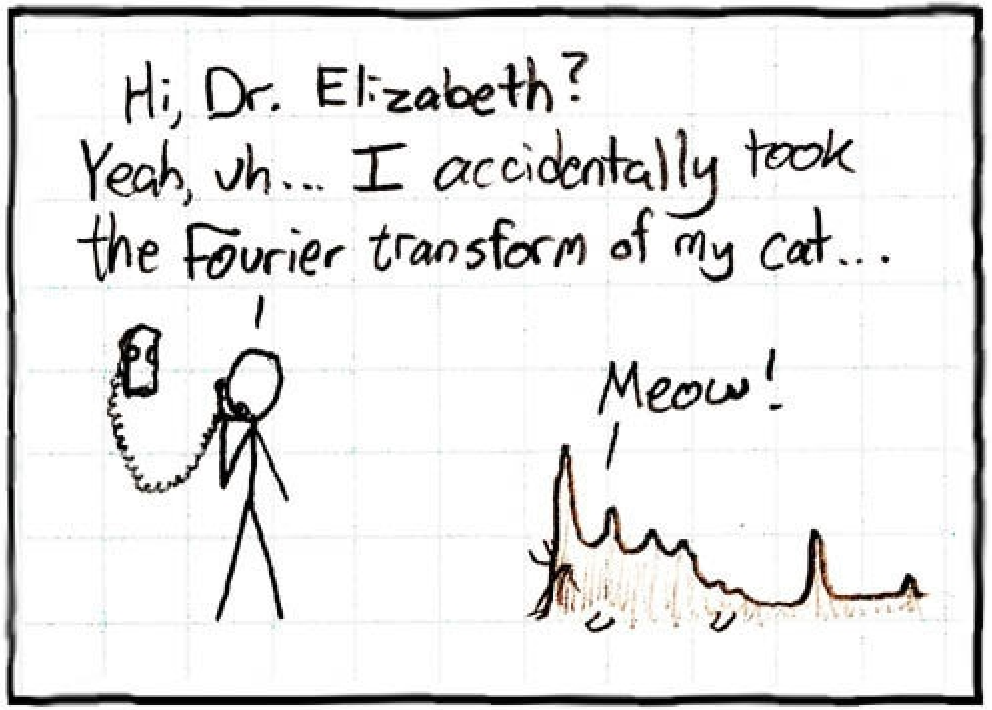
Other signals

- We can also think of all kinds of other signals the same way

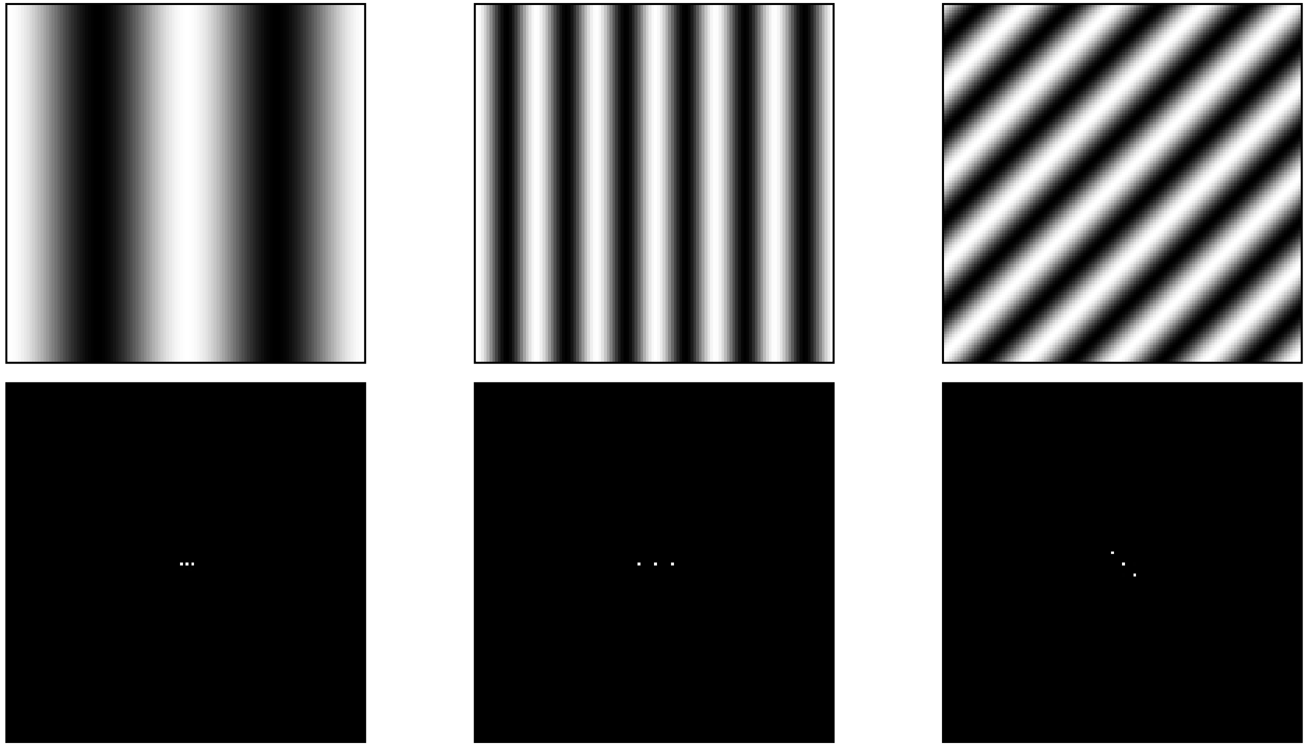


xkcd.com

Fourier Analysis in Images

Intensity Image

Fourier Image



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

2D Fourier Transform

Let $\mathbf{I}(r,c)$ be a single-band (intensity) digital image with R rows and C columns. Then, $\mathbf{I}(r,c)$ has Fourier representation

$$\mathbf{I}(r,c) = \frac{1}{RC} \sum_{u=0}^{R-1} \sum_{v=0}^{C-1} \mathbf{I}(v,u) e^{+i2\pi\left(\frac{vr}{R}+\frac{uc}{C}\right)},$$

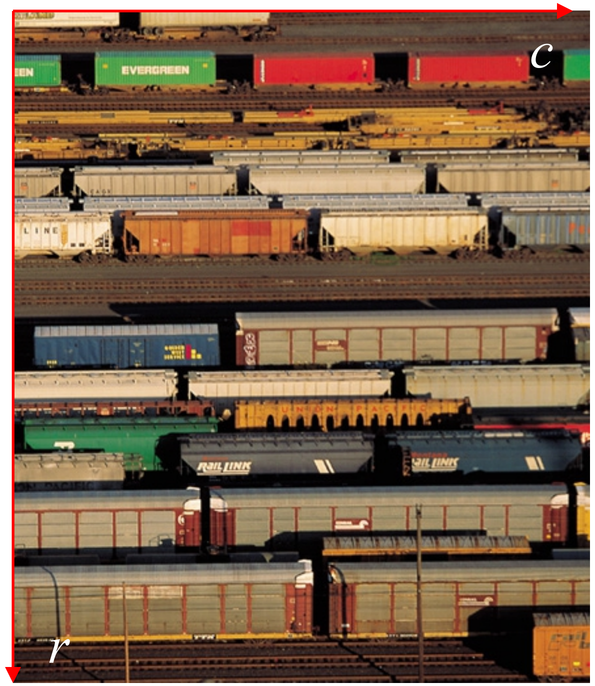
these complex exponentials are 2D sinusoids.

where

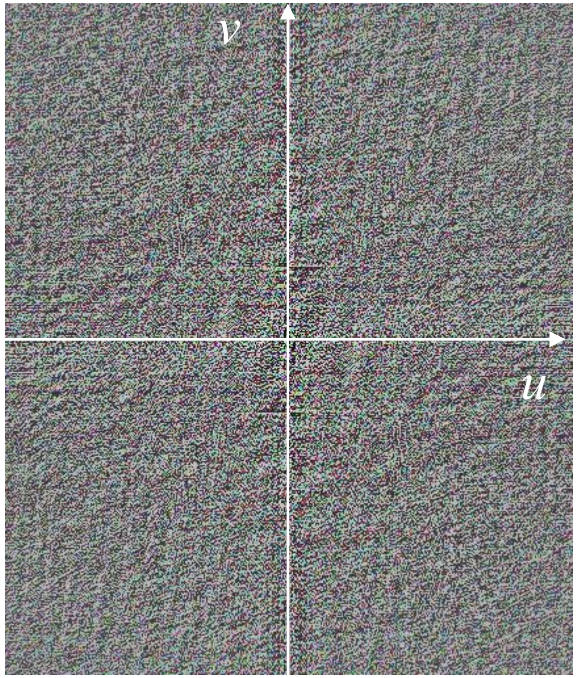
$$\mathbf{I}(v,u) = \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \mathbf{I}(r,c) e^{-i2\pi\left(\frac{vr}{R}+\frac{uc}{C}\right)}$$

are the R x C Fourier coefficients.

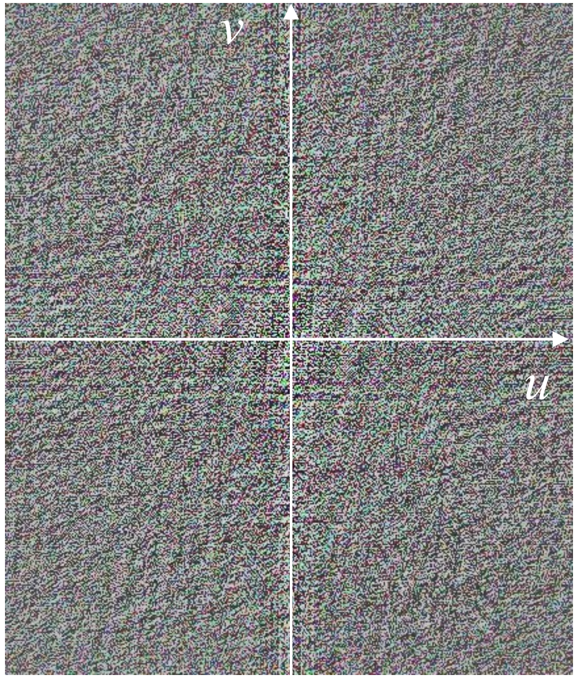
Fourier Transform of Images



$\mathbf{I}$



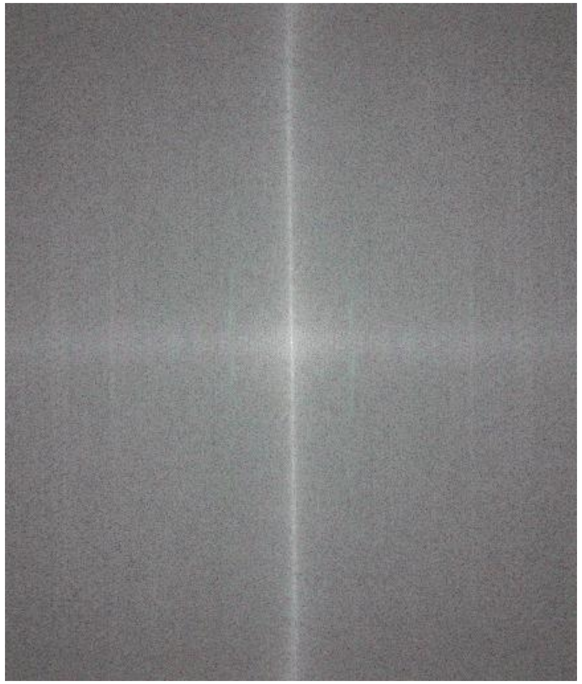
$\mathrm{Re}[\mathcal{F}\{\mathbf{I}\}]$



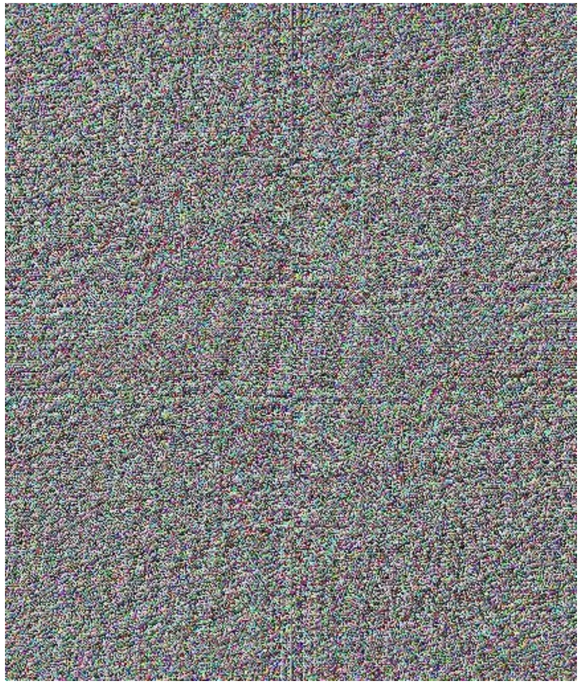
$\mathrm{Im}[\mathcal{F}\{\mathbf{I}\}]$

FT of an Image (Magnitude + Phase)

$\mathbf{I}$



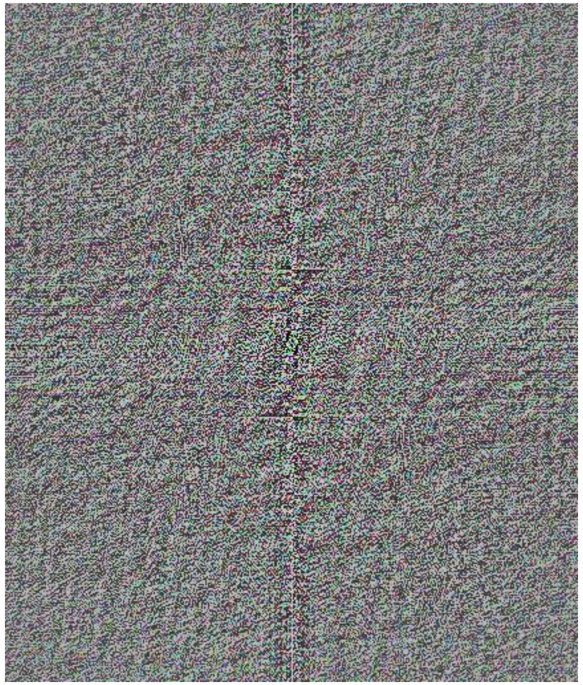
$\log\{|\mathcal{F}\{\mathbf{I}\}|^2+1\}$



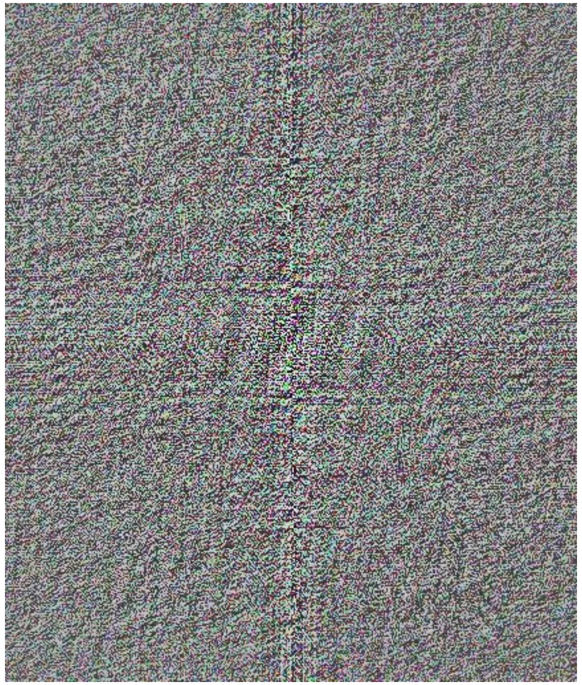
$\angle[\mathcal{F}\{\mathbf{I}\}]$

FT of an Image (Real + Imaginary)

$\mathbf{I}$



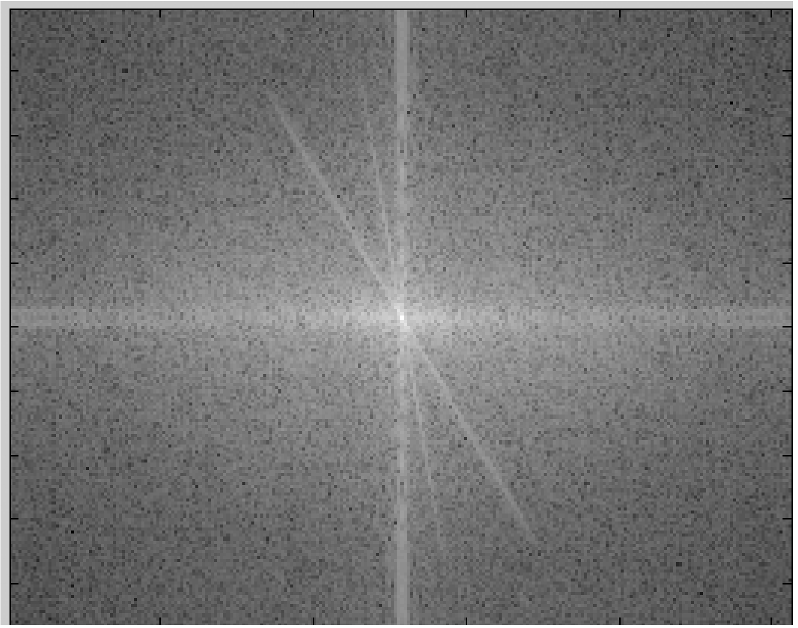
$\mathrm{Re}[\mathcal{F}\{\mathbf{I}\}]$



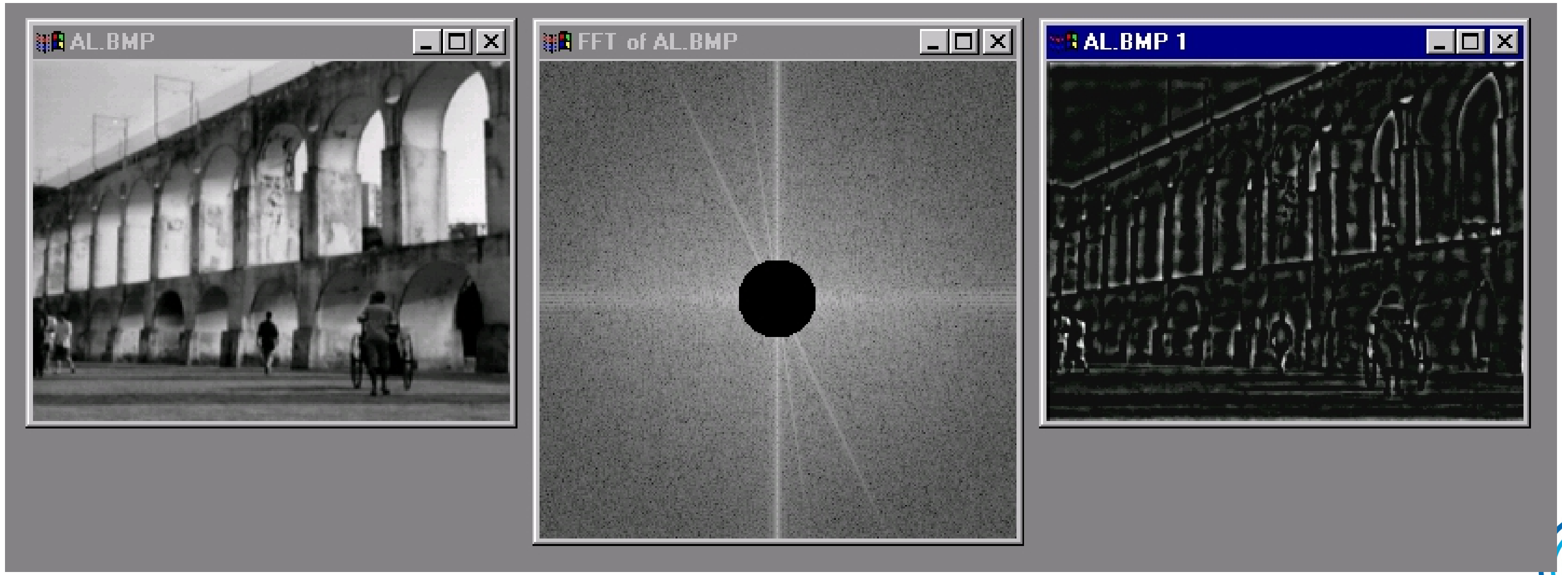
$\mathrm{Im}[\mathcal{F}\{\mathbf{I}\}]$

47

Man-made Scene

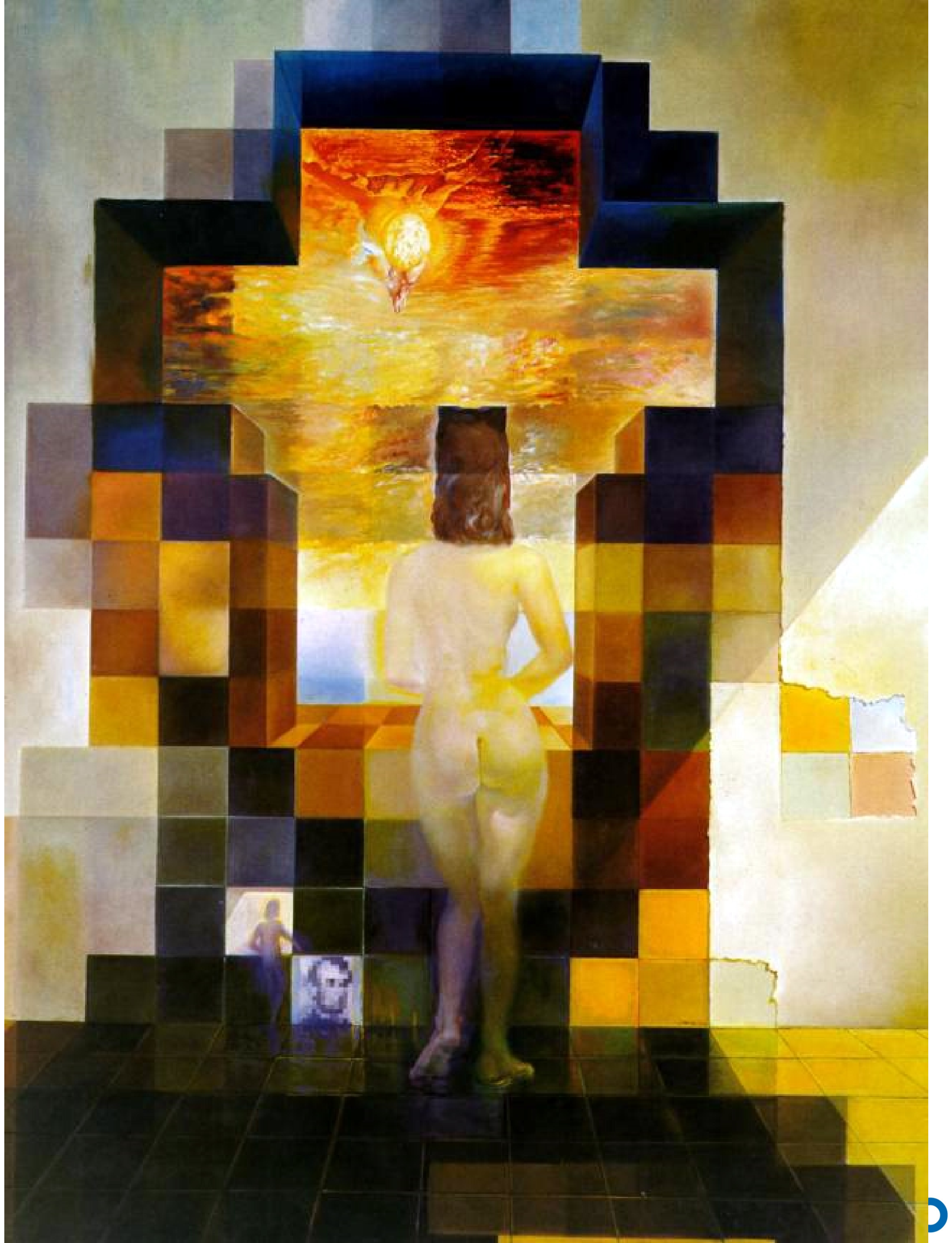


Low and High Pass filtering

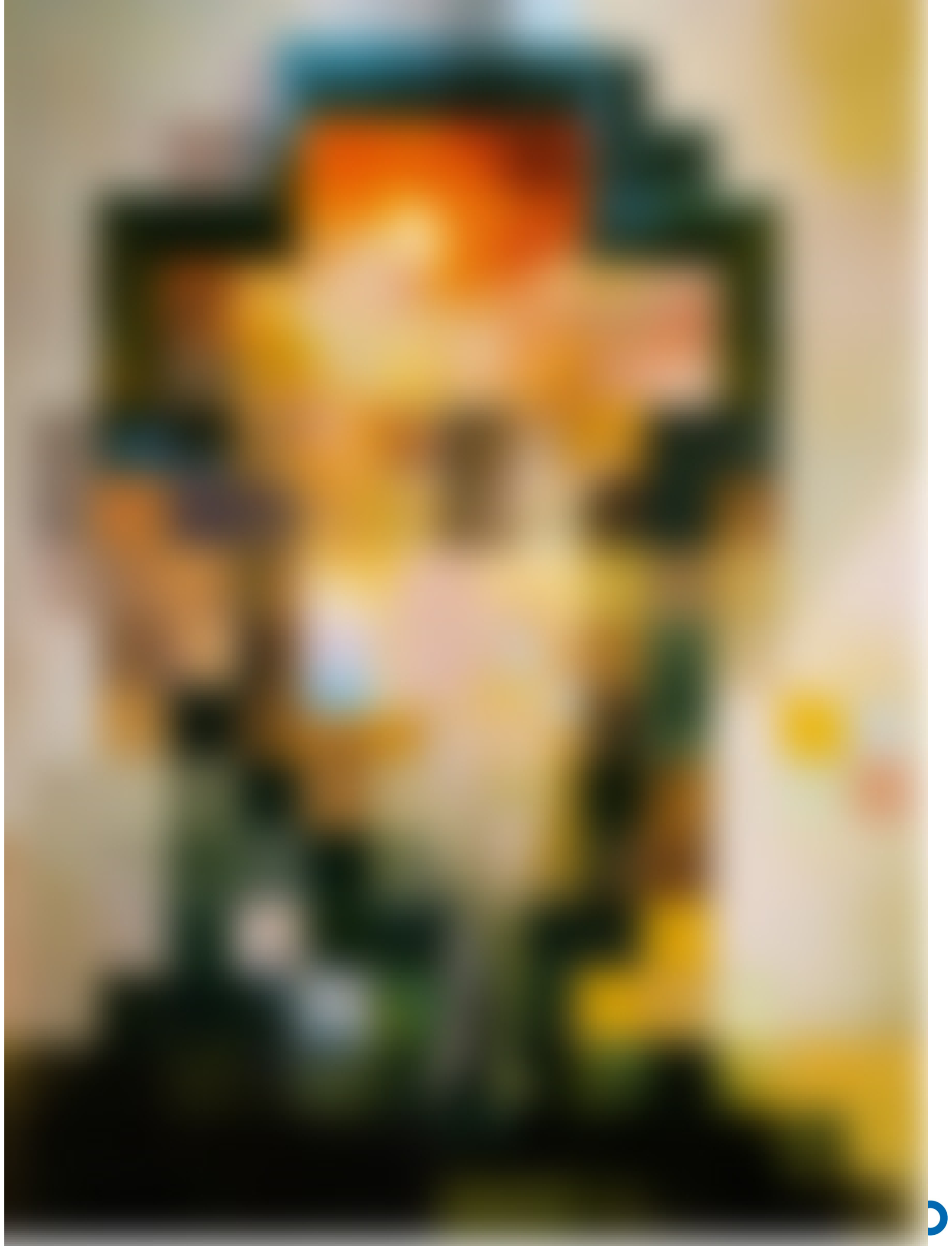


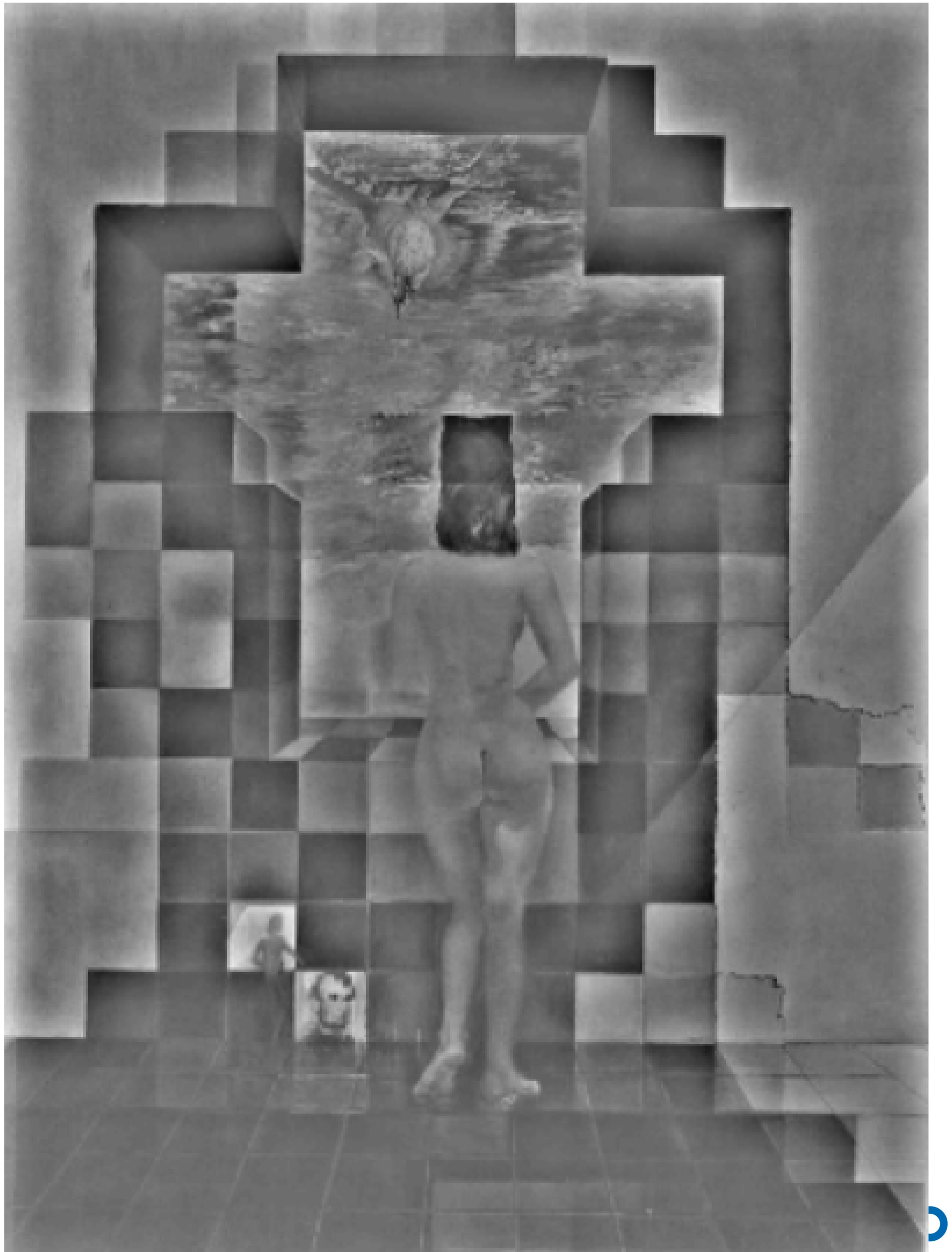
Can Change Spectrum, Then Reconstruct

Salvador Dali invented Hybrid Images?



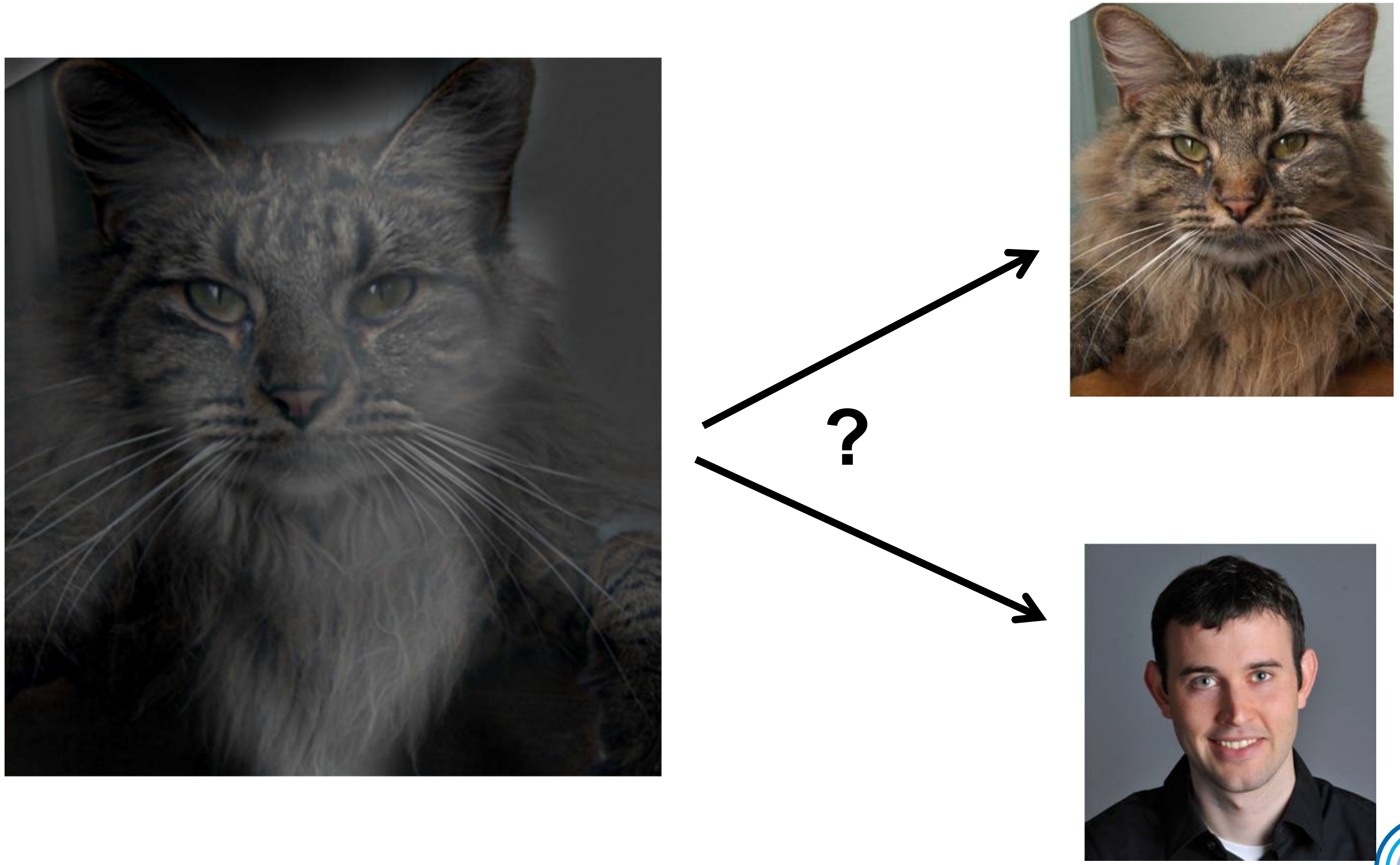
Salvador Dali
“Gala Contemplating the Mediterranean Sea, which at 20 meters becomes the portrait of Abraham Lincoln”, 1976





Visual Human Perception & Frequency

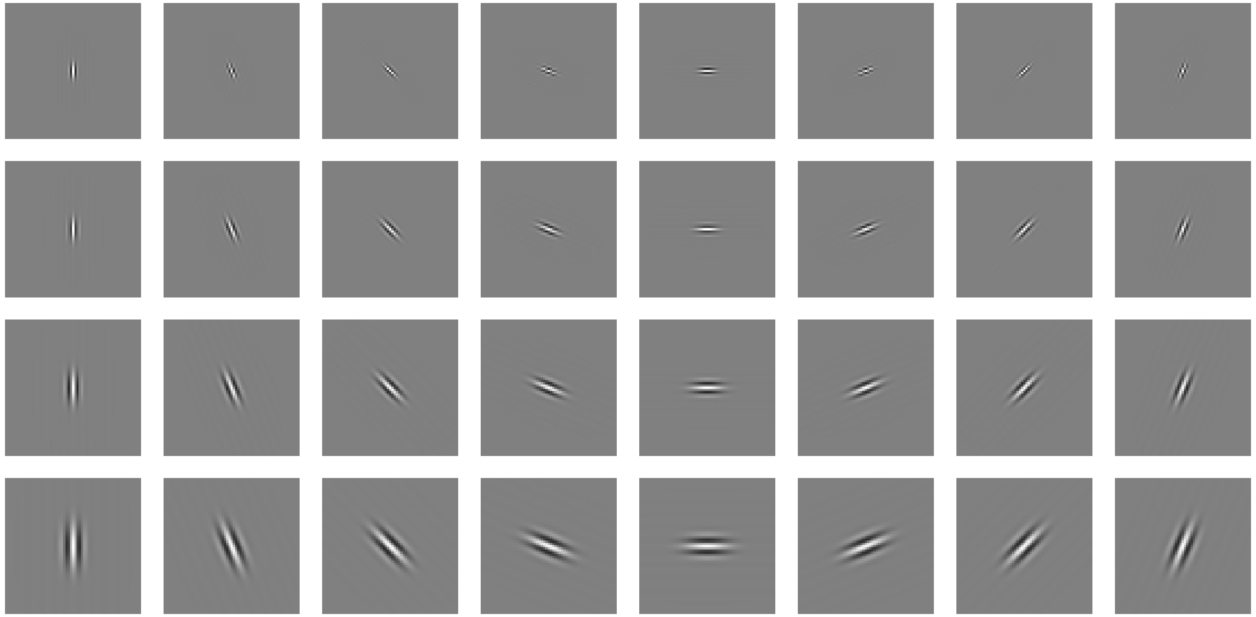
Why do we get different, distance-dependent interpretations of hybrid images?



Slide: Hoiem

Visual Human Perception & Frequency

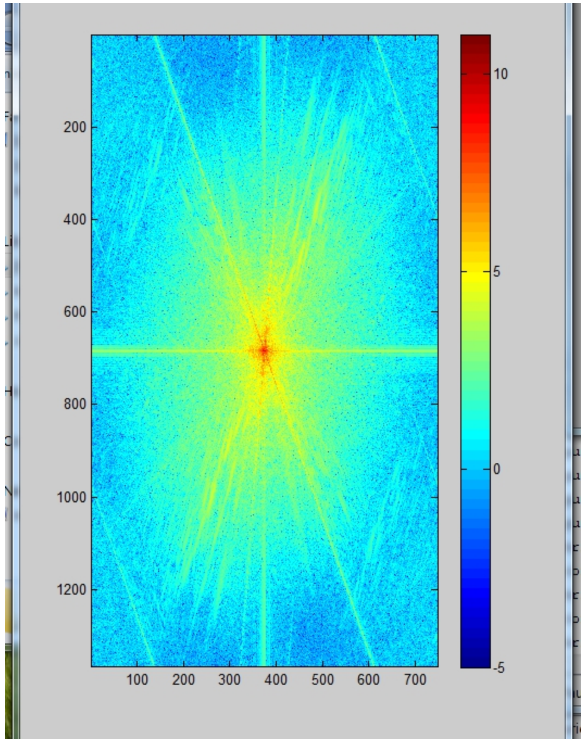
- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-high frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it



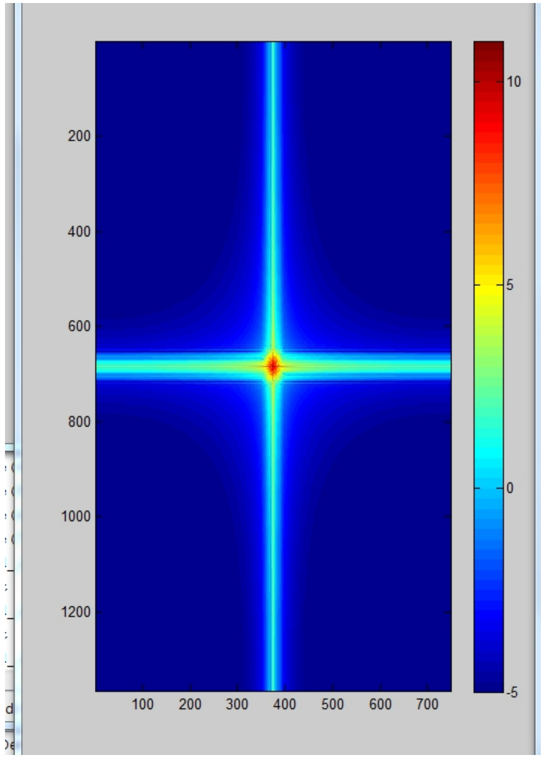
Early Visual Processing: Multi-scale edge and blob filters

Hybrid Image in FFT

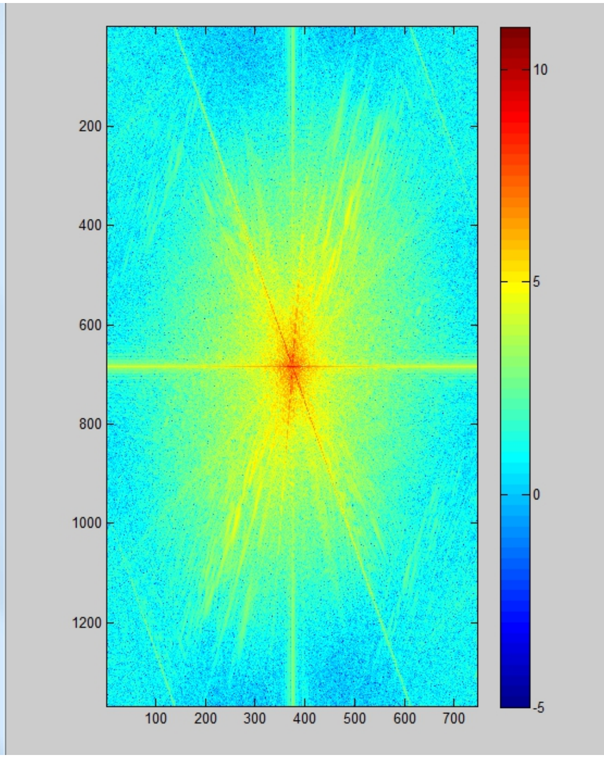
Hybrid Image



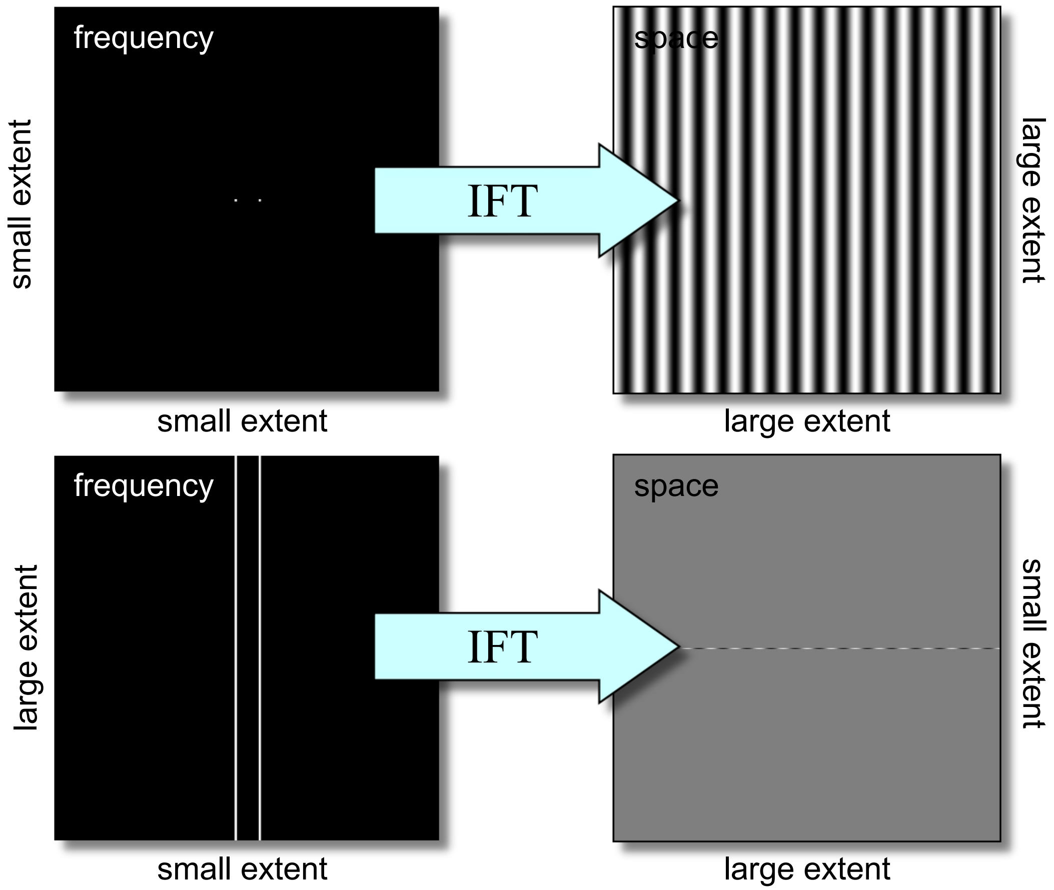
Low-passed Image



High-passed Image



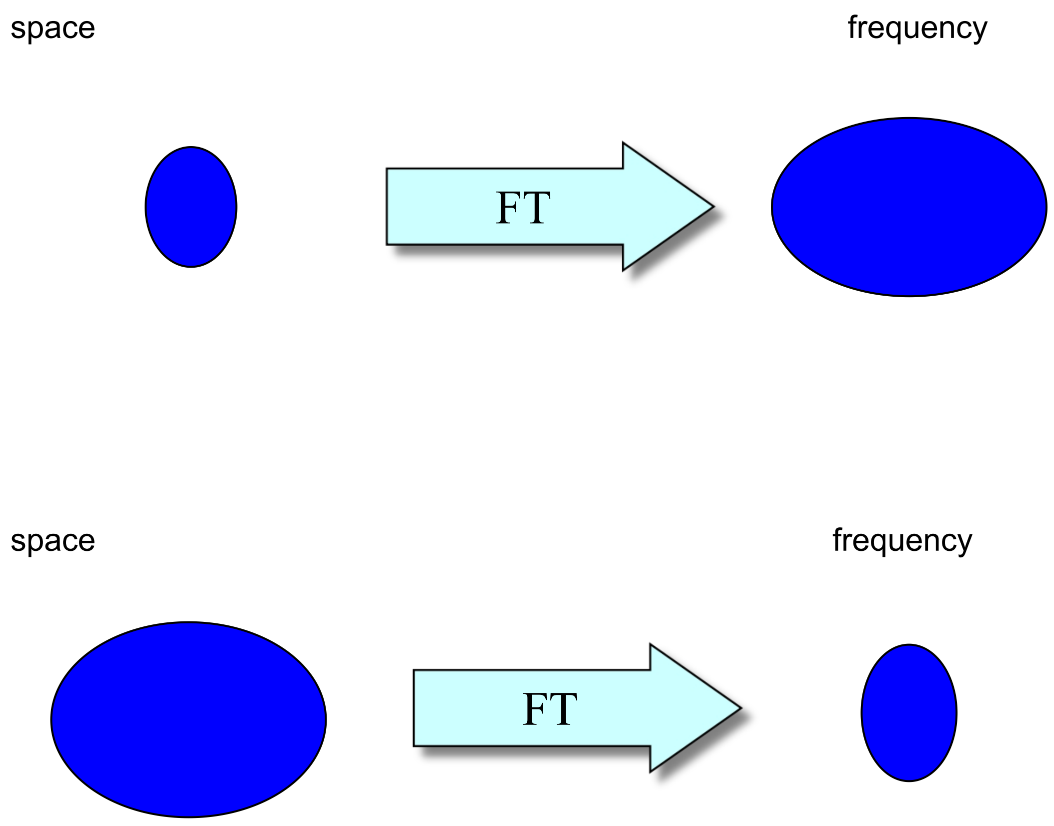
Spatial and Spectra Relations



Recall: a symmetric pair of impulses in the frequency domain becomes a sinusoid in the spatial domain.

A symmetric pair of lines in the frequency domain becomes a sinusoidal line in the spatial domain.

Spatial and Spectra Relations



If $\Delta x \, \Delta y$ is the extent of the object in space and if $\Delta u \, \Delta v$ is its extent in frequency then,

$$\Delta x \, \Delta y \cdot \Delta u \, \Delta v \geq \frac{1}{16\pi^2}$$

A small object in space has a large frequency extent and vice-versa.

Power Spectrum

For display, the log of the power spectrum is often used.

The power spectrum of a signal is the square of the magnitude of its Fourier Transform.

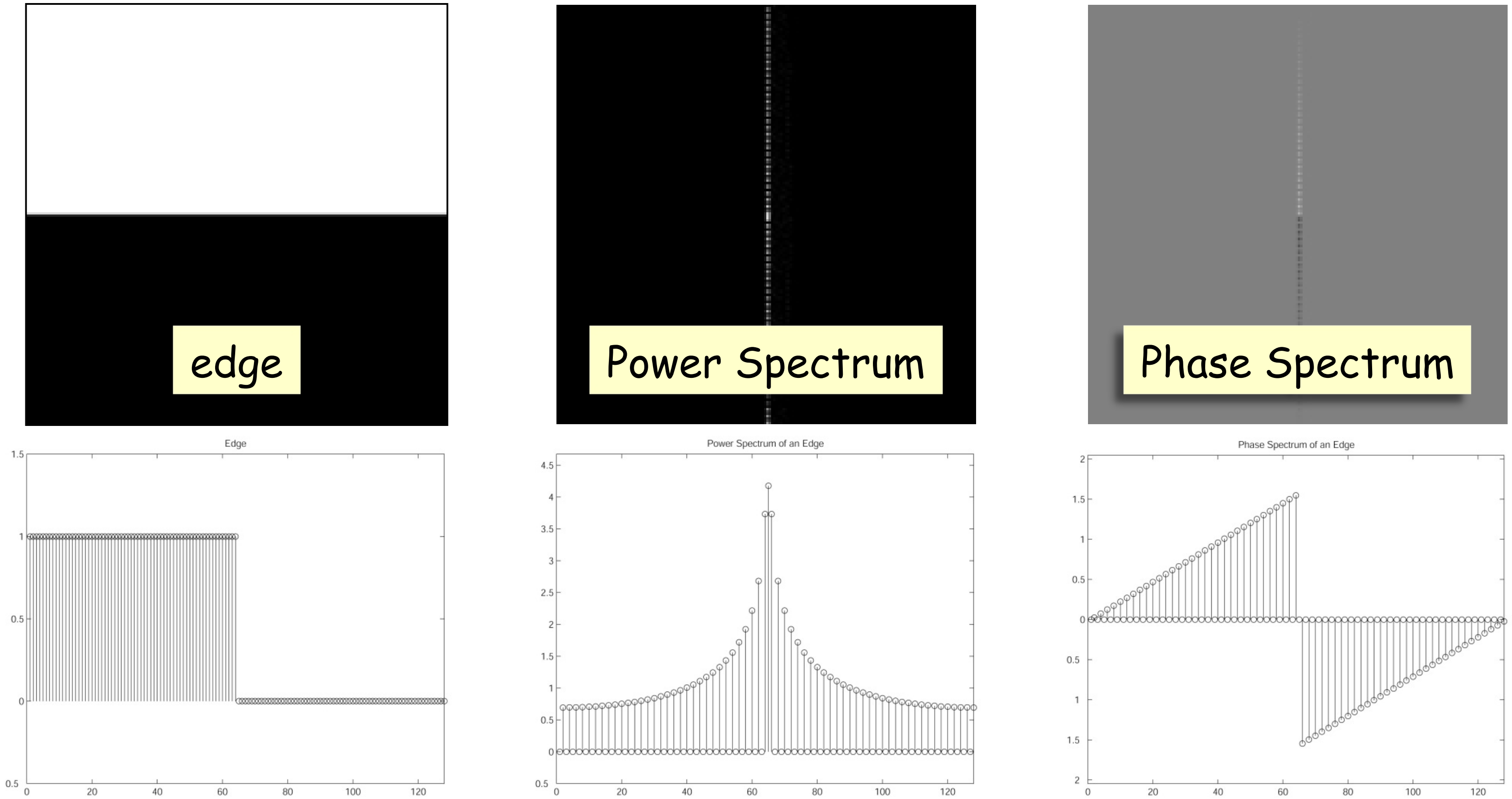
$$
\begin{aligned}
\left|\mathbf{I}(u,v)\right|^2 &= \mathbf{I}(u,v)\,\mathbf{I}^*(u,v) \\
&= \left[\operatorname{Re}\mathbf{I}(u,v) + i\operatorname{Im}\mathbf{I}(u,v)\right]\left[\operatorname{Re}\mathbf{I}(u,v) - i\operatorname{Im}\mathbf{I}(u,v)\right] \\
&= \left[\operatorname{Re}\mathbf{I}(u,v)\right]^2 + \left[\operatorname{Im}\mathbf{I}(u,v)\right]^2 .
\end{aligned}
$$

At each location (u,v) it indicates the squared intensity of the frequency component with period $\lambda = 1\big/\sqrt{u^2+v^2}$ and orientation

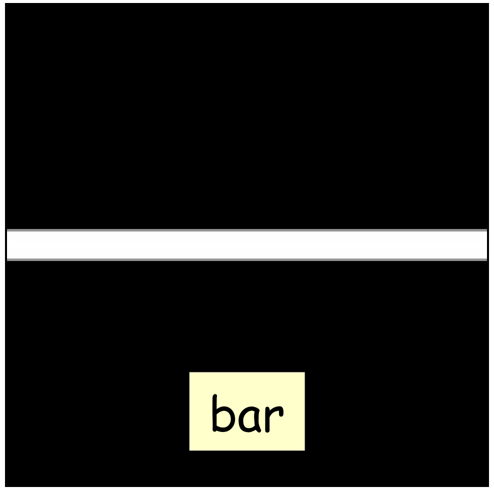
$\theta_{\mathrm{wf}} = \tan^{-1}\left(\frac{\omega_v}{\omega_u}\right) = \tan^{-1}\left(\frac{vC}{uR}\right).$

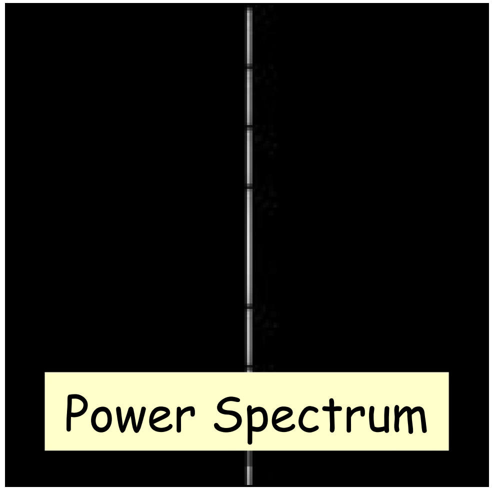
```
For display in Matlab:
PS = fftshift(2*log(abs(fft2(I))+1));
```

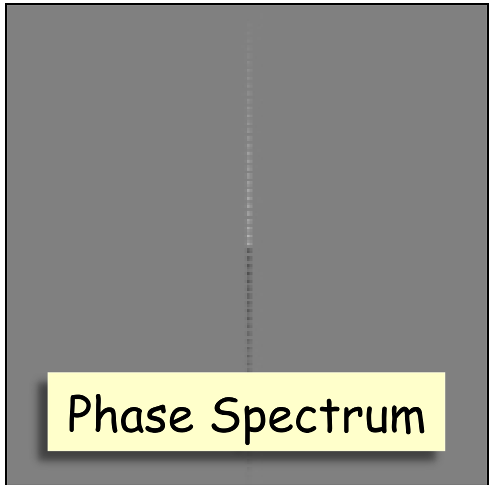
Fourier Transform of Edges

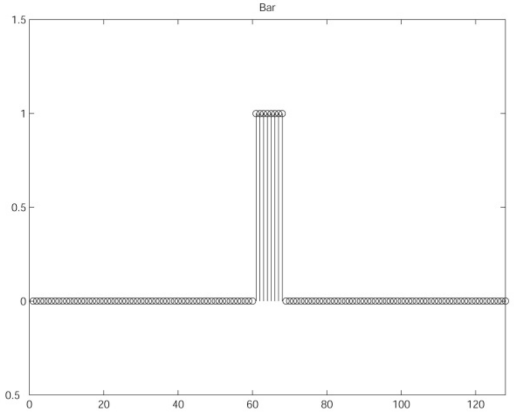


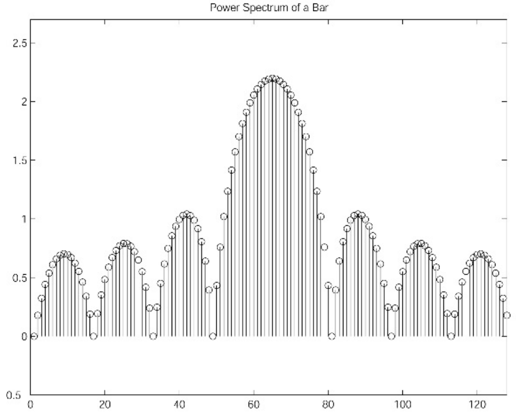
Fourier Transform Bar

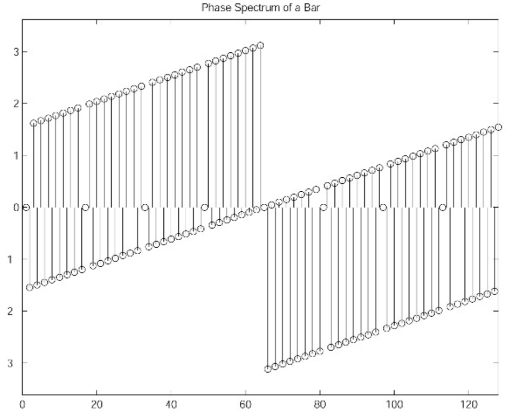












2D Fourier Transform Properties

$af(r,c)+bg(r,c) \Leftrightarrow aF(v,u)+bG(v,u)$	Linearity
$f(r-r_0,c-c_0) \Leftrightarrow e^{-j2\pi(vr_0+uc_0)}F(v,u)$	Shifting
$e^{j2\pi(rv_0+cu_0)}f(r,c) \Leftrightarrow F(v-v_0,u-u_0)$	Modulation
$f(r,c)*g(r,c) \Leftrightarrow F(v,u)G(v,u)$	Convolution
$f(r,c)g(r,c) \Leftrightarrow F(v,u)*G(v,u)$	Multiplication
$f(r,c)=f(r)f(c) \Leftrightarrow F(v,u)=F(v)F(u)$	Separability
$\sum_{r=1}^{R}\sum_{c=1}^{C}\lvert f(r,c)\rvert^2 = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\lvert F(v,u)\rvert^2 dvdu$	Parseval Thm.

Examen Partiel

- **Date examen partiel:** 31/03/2025
- **Horaire:** 13:30 - 15:20 (en salle de cours)
- **Contenu:** jusqu’a transformée de Fourier

Sampling & Discrete Signals

- Relation to Shannon Theorem
  - For the signal $x(t)$ sampled with frequency f_e

$$
\begin{aligned}
y(t) &= x(t).e(t) \\
&= x(t) \sum_{k=-\infty}^{\infty} \delta(t-kT_e) \\
&= \sum_{k=-\infty}^{\infty} x(t).\delta(t-kT_e) \\
&= \sum_{k=-\infty}^{\infty} x(kT_e).\delta(t-kT_e)
\end{aligned}
$$

  - So the FT

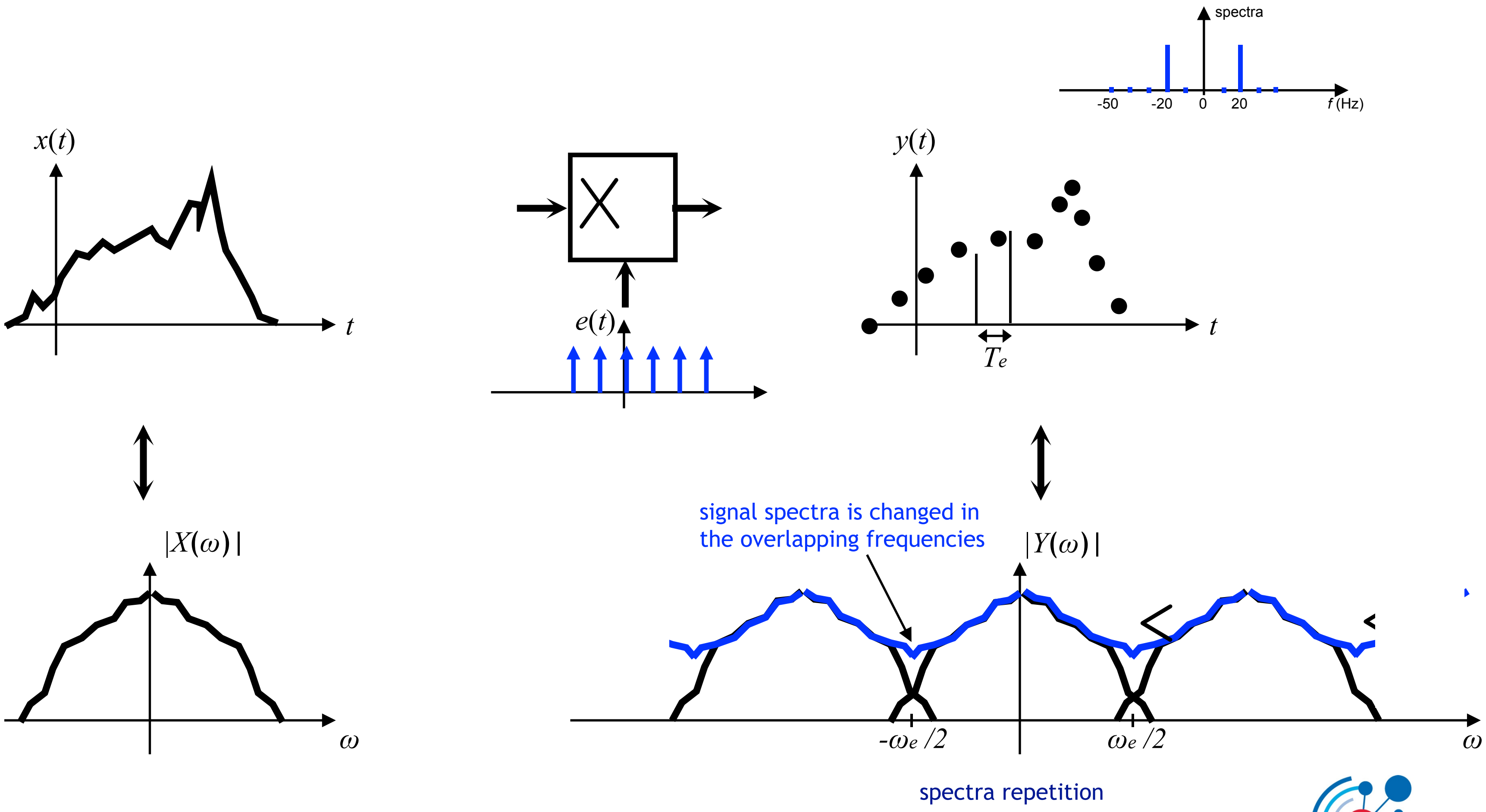
$$
\begin{aligned}
Y(\omega) &= \frac{1}{2\pi} X(\omega) * \omega_e \sum_{n=-\infty}^{\infty} \delta\left(\omega - n\omega_e\right) \\
&= \frac{1}{T_e} \sum_{n=-\infty}^{\infty} X(\omega) * \delta\left(\omega - n\omega_e\right) \\
&= \frac{1}{T_e} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_e)
\end{aligned}
$$

transformations

$$
\begin{aligned}
\sum_{k=-\infty}^{\infty} \delta(t-kT_e) &\longleftrightarrow \omega_e \sum_{n=-\infty}^{\infty} \delta\left(\omega - n\omega_e\right) \\
x(t).y(t) &\longleftrightarrow \frac{1}{2\pi} X(\omega) * Y(\omega) \\
x(t) * y(t) &\longleftrightarrow X(\omega).Y(\omega)
\end{aligned}
$$

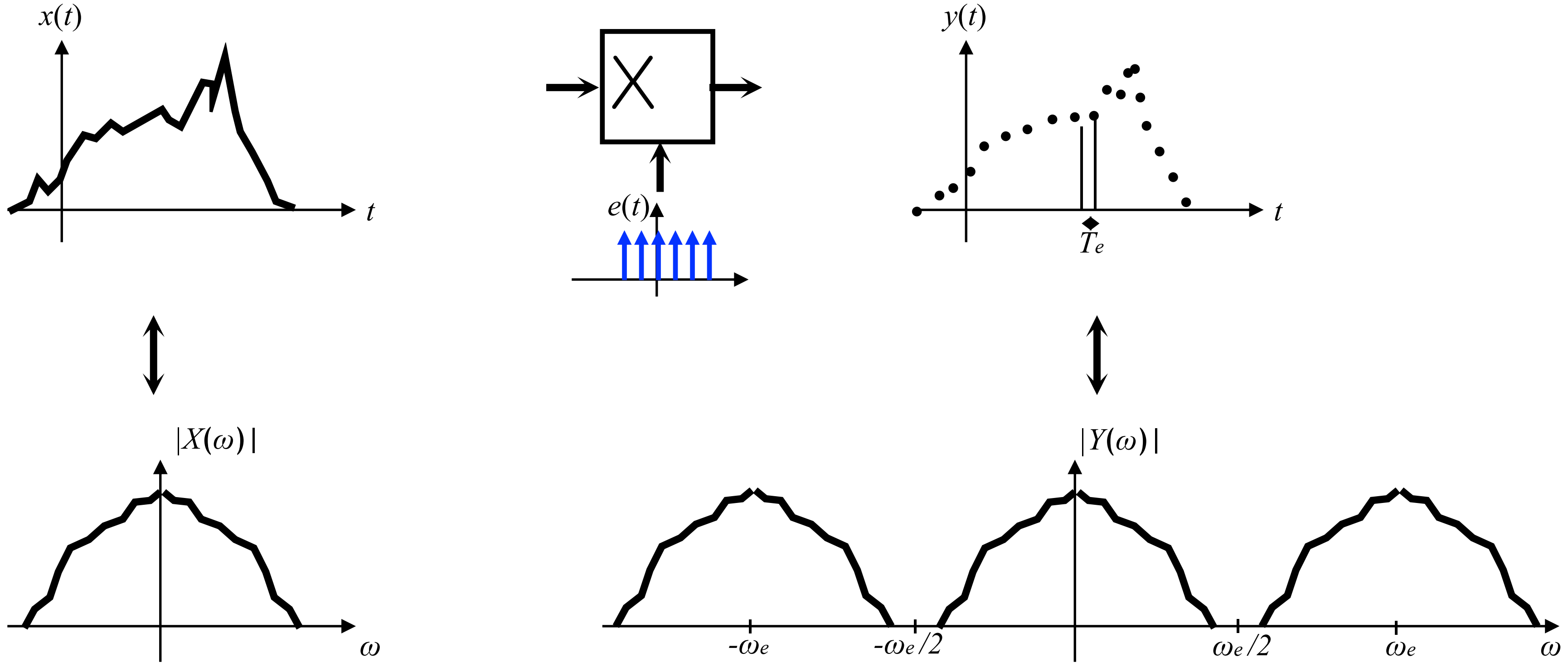
Sampling & Discrete Signals

- If frequency of sampling is smaller than twice the max frequency in the spectra



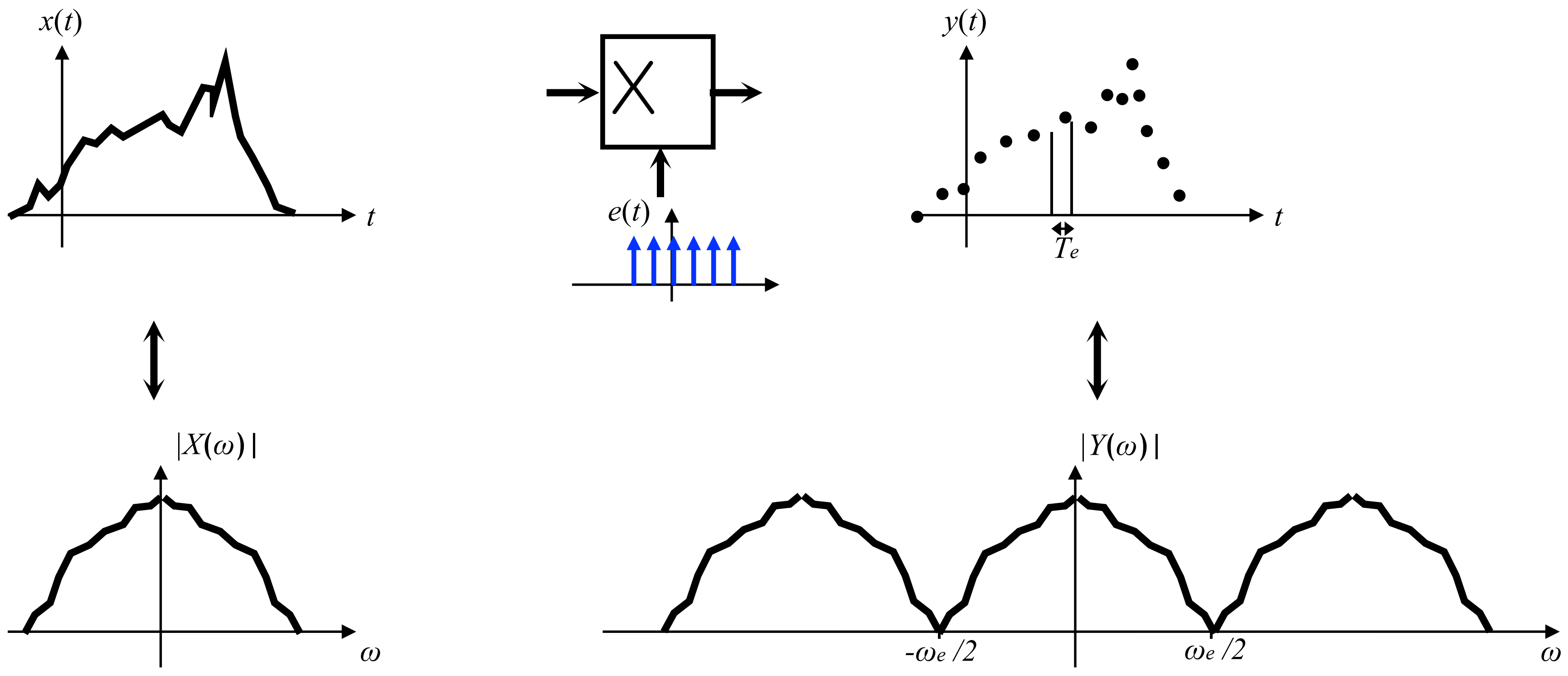
Sampling & Discrete Signals

- Sampling makes the frequency components periodic!



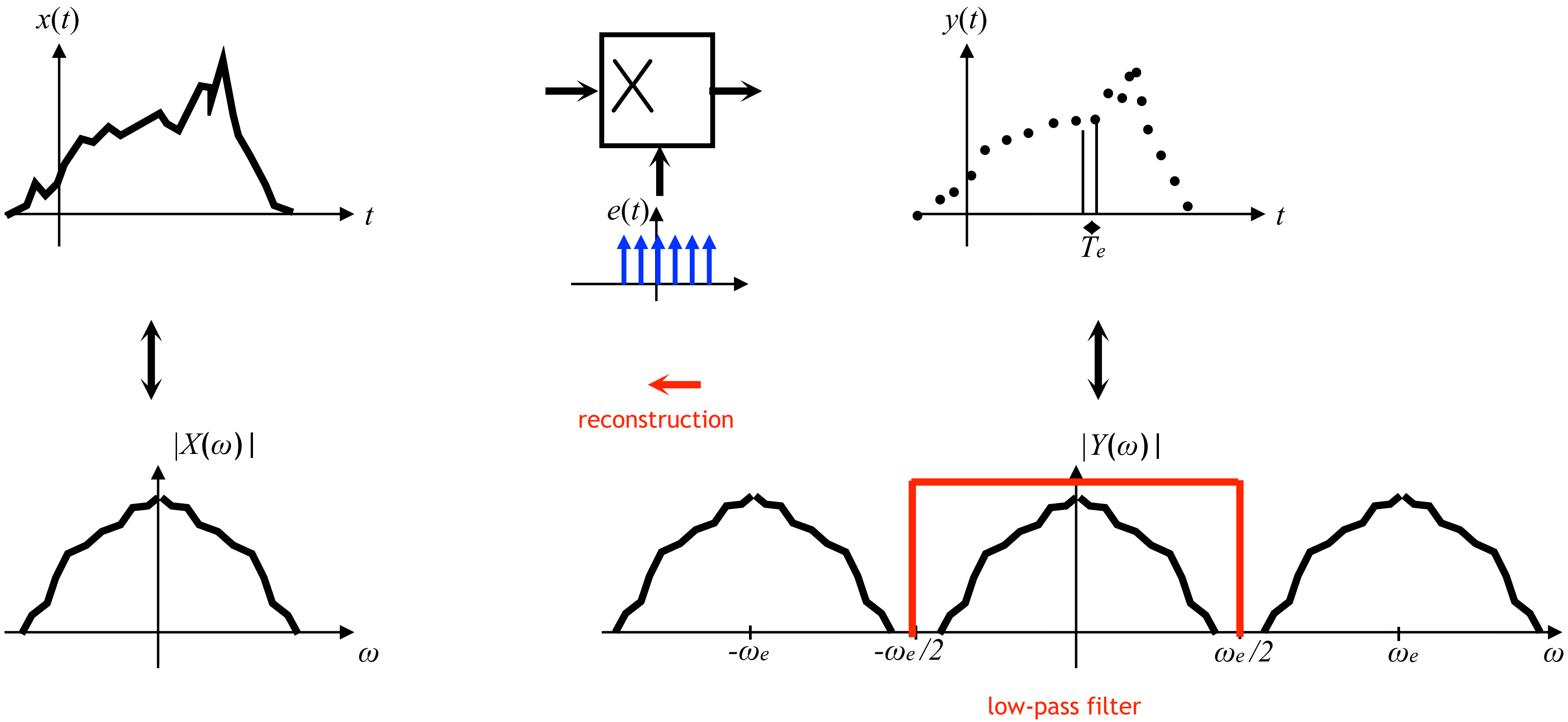
Shannon-Kotelnikov-Whittaker Condition

- No loss of information when sampling if $f_{max} < f_e/2$



spectra repetition

- Reconstruction of the signal if cropping the frequencies to $f_{max} = f_e/2$



Discrete Fourier Transform - DFT

- DFT is directly linked with the sampling of the signal

$$S(\omega) \quad = \quad \sum_{k=-\infty}^{\infty} s[k] e^{-j\omega k T_e}$$

- Problems:
  - Infinite sum
  - S is continuous but s is discrete

- Thus this definition is not adapted to discrete signals in practice...

DTF

- We limit the sum to N points ⇒ we therefore apply a window to the signal
- Starting from N points in the spatial domain, we arrive at N points in the frequency domain ⇒ the frequency step is therefore Δf = fe / N, since

  $f \in [-fe/2; fe/2]$

$$
\begin{aligned}
S(n) &= \sum_{k=0}^{N-1} s[k] e^{-j2\pi . n . \Delta f . k T_e} \\
&= \sum_{k=0}^{N-1} s[k] e^{-j2\pi . n . \frac{fe}{N} f . k T_e} \\
&= \sum_{k=0}^{N-1} s[k] e^{-j2\pi . \frac{n.k}{N}}
\end{aligned}
$$

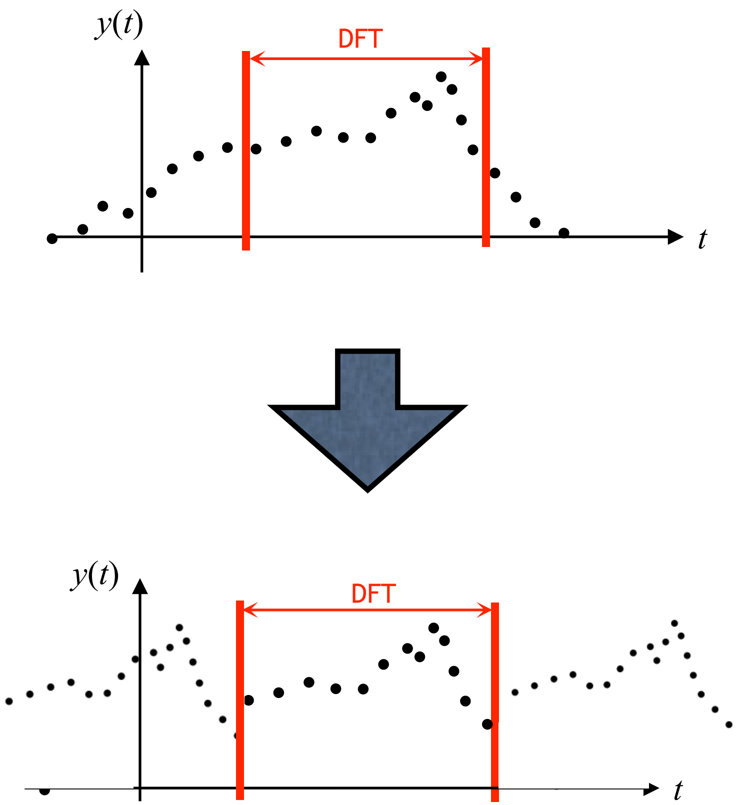
$n \in [-N/2; N/2\text{-}1]$

- DTF inverse :

$$
x[k] = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} X(n) e^{j2\pi . \frac{n.k}{N}}
$$

DFT

DFT gives a frequency content corresponding to a periodic signal!



This signal, made periodic by the DFT, often exhibits discontinuities.
These discontinuities generate high-frequency spectra.

Computing the Fourier Transform

$$H(\omega) = \mathcal{F}\{h(x)\} = Ae^{j\phi}$$

Continuous

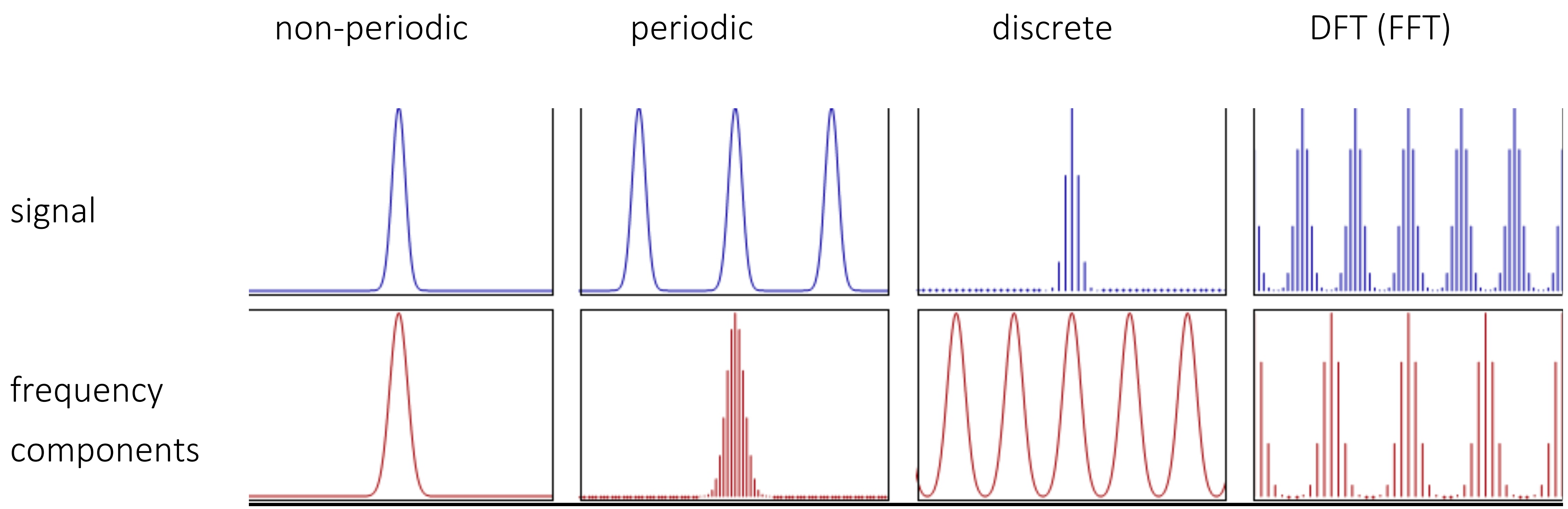
$$H(\omega) = \int_{-\infty}^{\infty} h(x) e^{-j\omega x} dx$$

Discrete

$$H(k) = \frac{1}{N} \sum_{x=0}^{N-1} h(x) e^{-j\frac{2\pi k x}{N}}$$ *k = -N/2..N/2*

Fast Fourier Transform (FFT): N.logN

Sampling & Frequency Overview



2D Discrete Fourier Transform

2D Discrete Fourier Transform

- Direct extension of the 1D counterpart

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\, e^{-j\frac{2\pi}{N}ux} e^{-j\frac{2\pi}{M}vy} \qquad \begin{array}{l} u = 1,2,\dots,N-1 \\ v = 1,2,\dots,M-1 \end{array}$$

$$f(x,y) = \frac{1}{NM}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)\, e^{j\frac{2\pi}{N}ux} e^{j\frac{2\pi}{M}vy} \qquad \begin{array}{l} x = 1,2,\dots,N-1 \\ y = 1,2,\dots,M-1 \end{array}$$

2D Discrete Fourier Transform

- Since 2D DFT in general is complex, we can express in polar form

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)}$$

where

$$|F(u,v)| = \left(R^2(u,v) + I^2(u,v)\right)^{1/2}$$

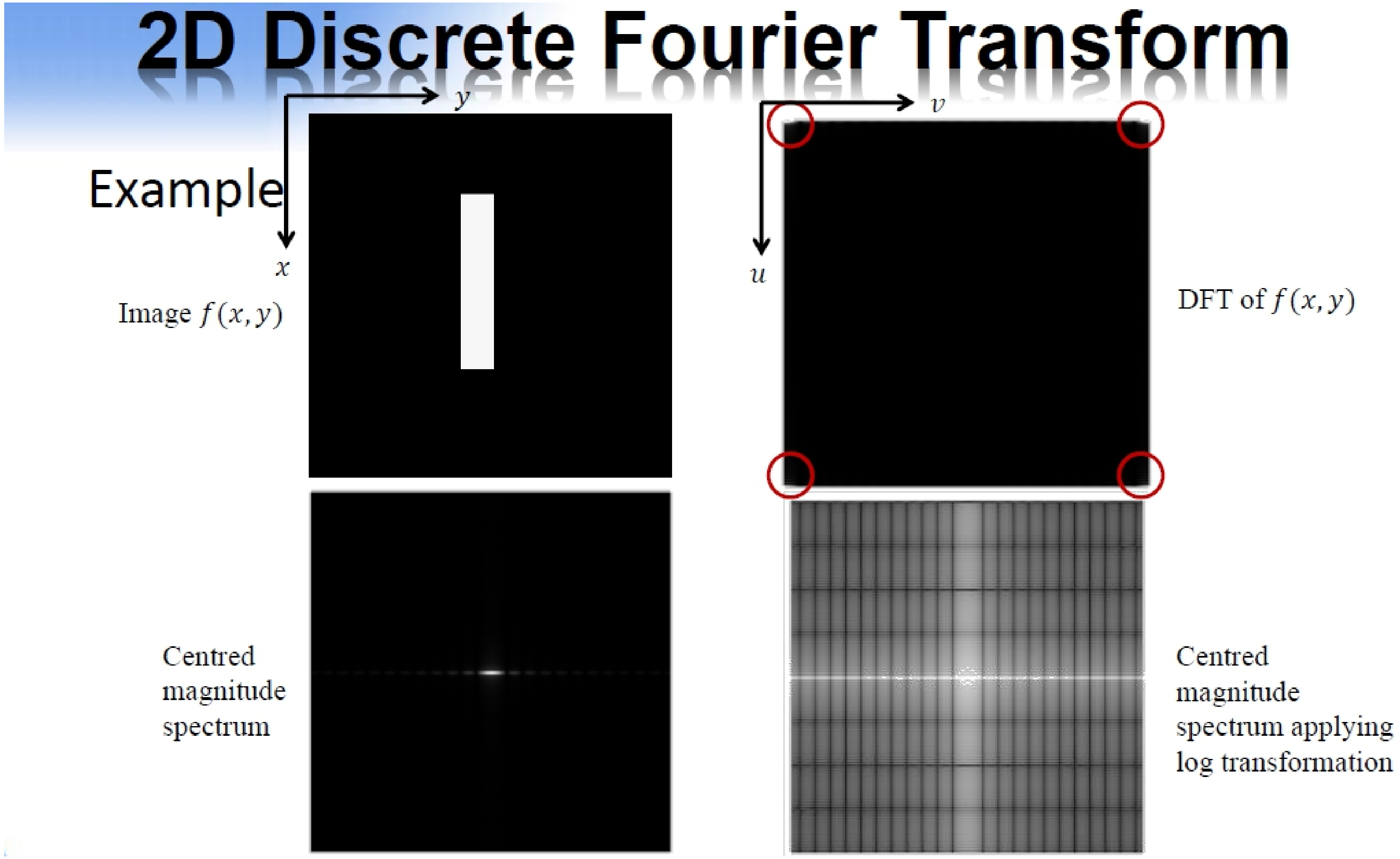
Magnitude spectrum

and

$$\phi(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$$

Phase spectrum

2D Discrete Fourier Transform



2D Discrete Fourier Transform

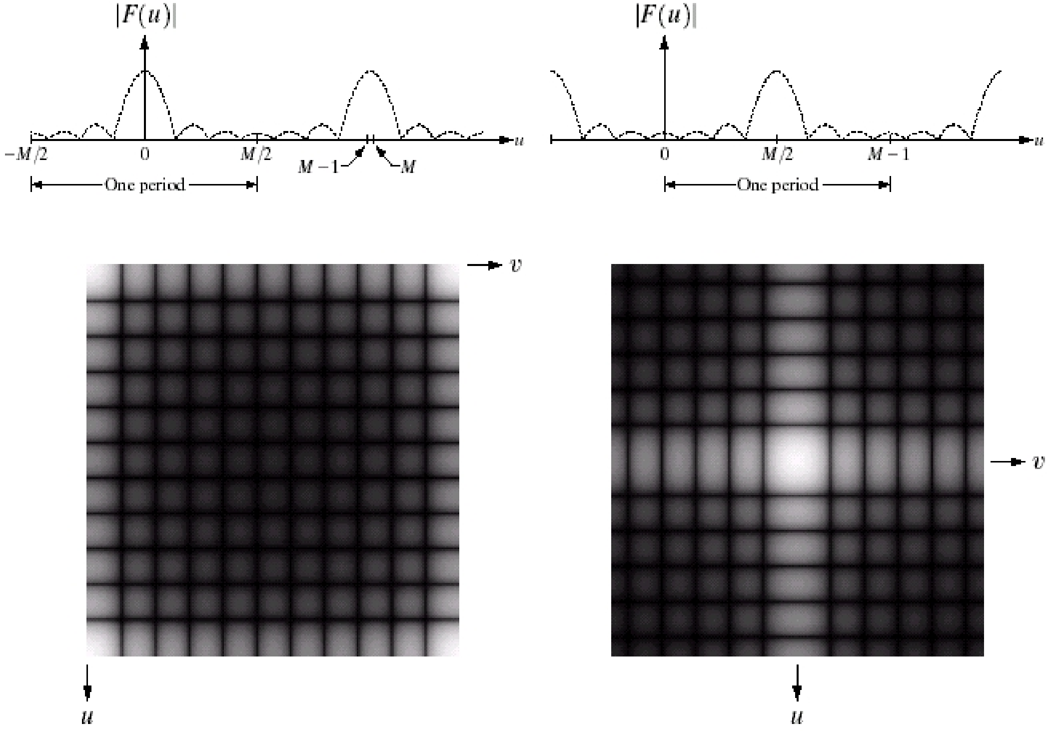
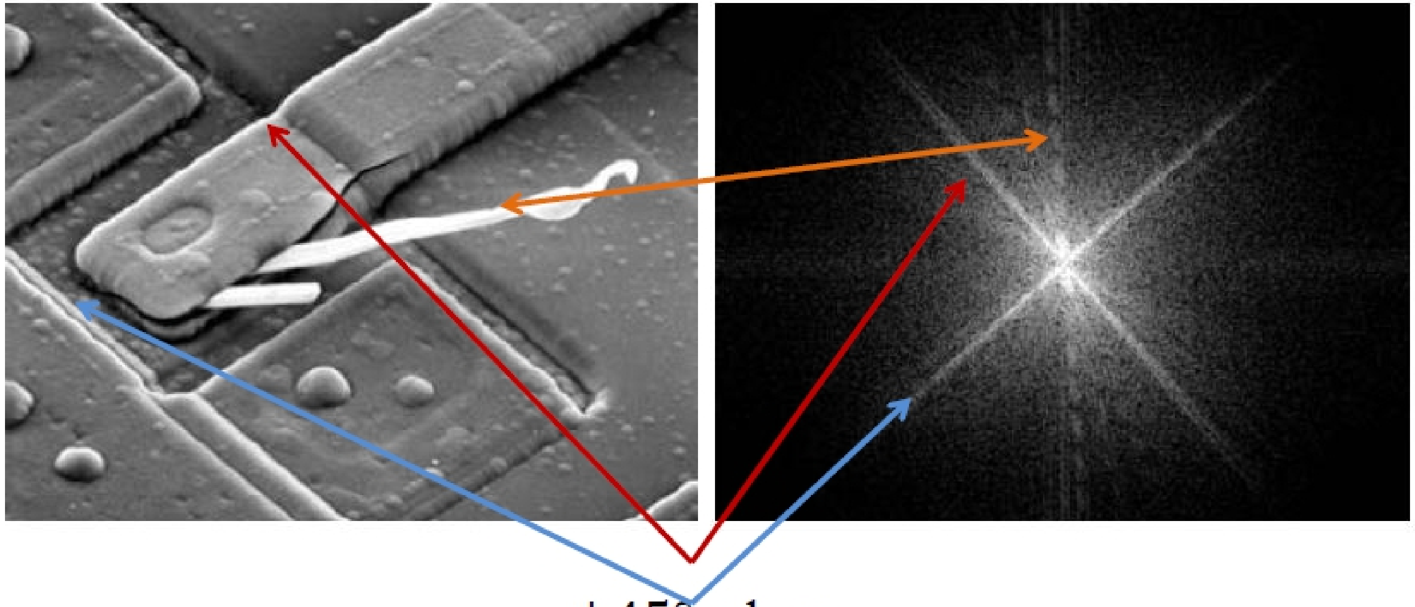


FIGURE 4.34 (a) Fourier spectrum showing back-to-back half periods in the interval $[0, M-1]$. (b) Shifted spectrum showing a full period in the same interval. (c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions. (d) Centered Fourier spectrum.

2D Discrete Fourier Transform



$\pm 45°$ edges

Spatial domain – with respect to horizontal line
Frequency domain – with respect to vertical line

2D Discrete Fourier Transform

- How to do this?

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

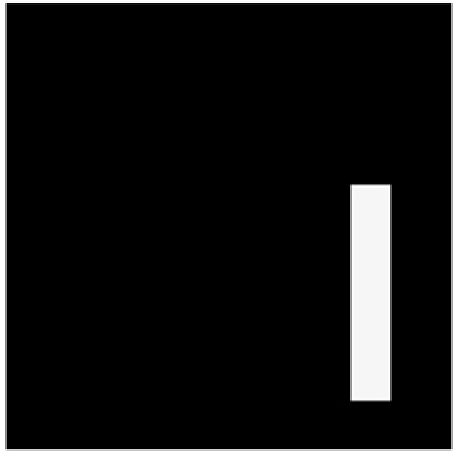
Shifting by N/2 and
M/2 in frequency
domain

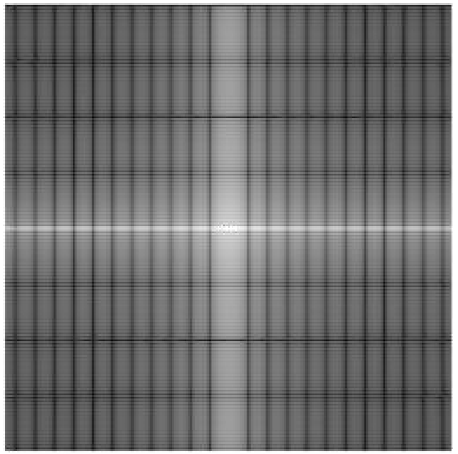
- Fortunately, Matlab has a simple command for achieving this.

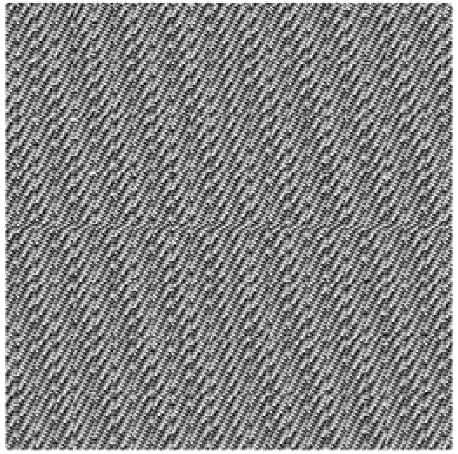
2D Discrete Fourier Transform

- Translation

$$f(x-x_0,y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi\left(\frac{x_0u}{M}+\frac{y_0v}{N}\right)}$$


Translated bar


Magnitude spectrum


Phase spectrum

2D Discrete Fourier Transform

- Rotation
- If we write $f(x, y)$ in polar form i.e.

$$x = r\cos\theta \qquad u = w\cos\varphi$$

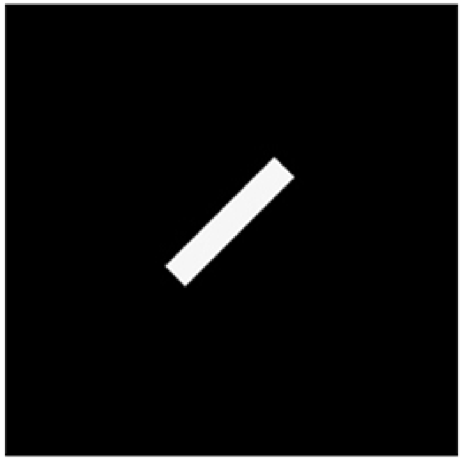
$$y = r\sin\theta \qquad v = w\sin\varphi$$

then $f(x, y)$ and $F(u, v)$ become $f(r, \theta)$ and $F(w, \varphi)$

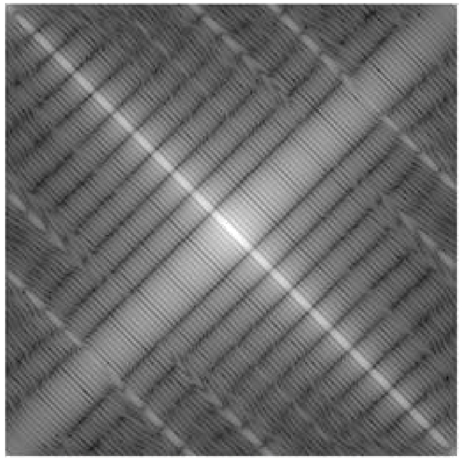
thus

$$f(r, \theta + \theta_0) \Leftrightarrow F(w, \varphi + \theta_0)$$

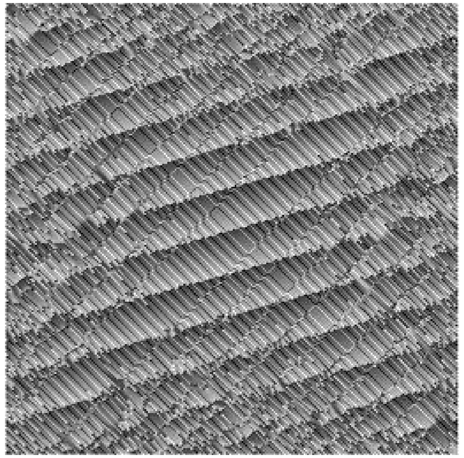
2D Discrete Fourier Transform



Rotated bar



Magnitude spectrum

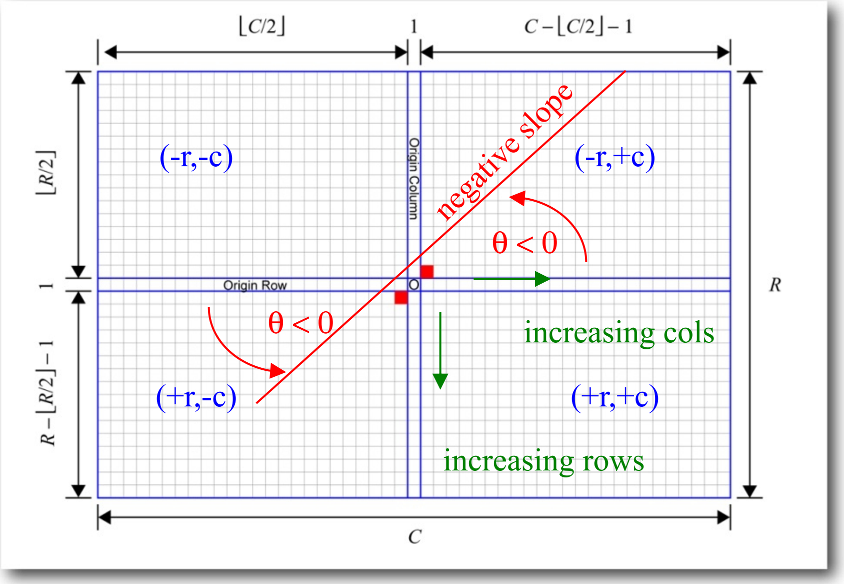


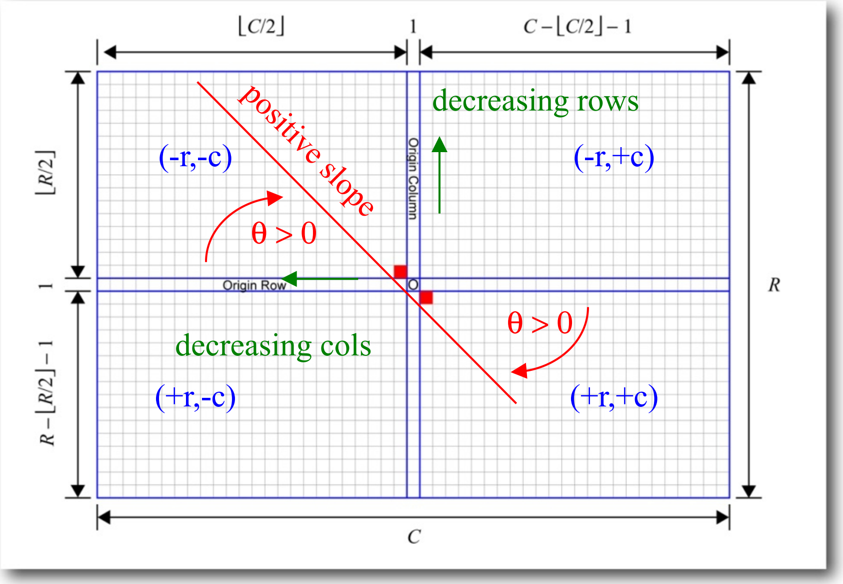
Phase spectrum

2D Discrete Fourier Transform

- Amplitude spectrum - How Much of each sinusoid component is present.
- Phase spectrum - Where each of the sinusoidal components resides within the image.

Coordinates and Directions in the Fourier Plane

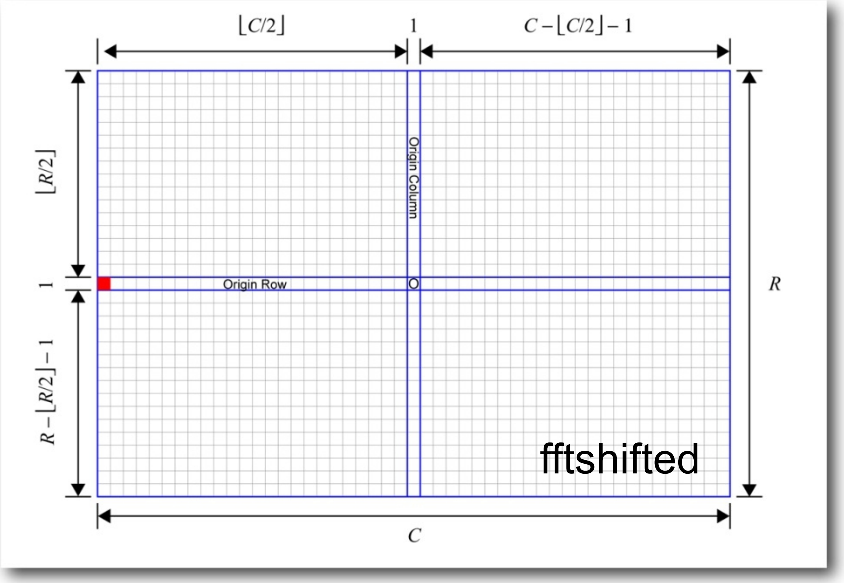


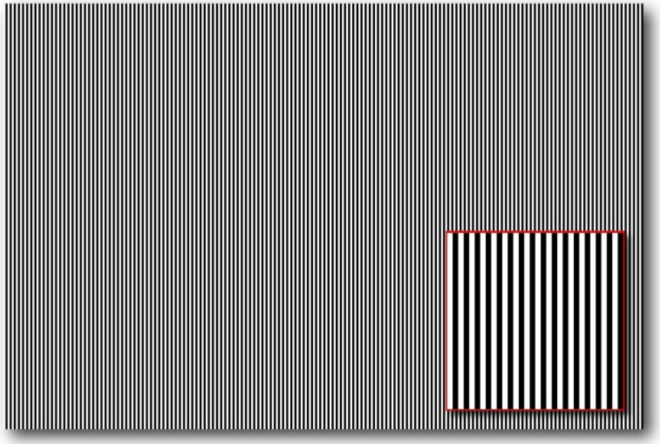


Since rows increase down and columns to the right, slopes and angles are opposite those of a right-handed coordinate system.

Inverse FFTs of Impulses

“horizontal” is the wavefront direction.

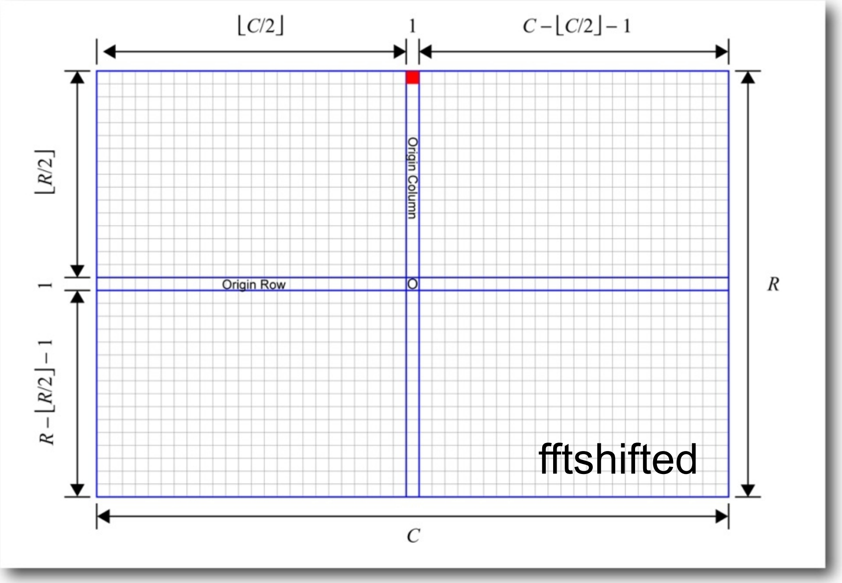


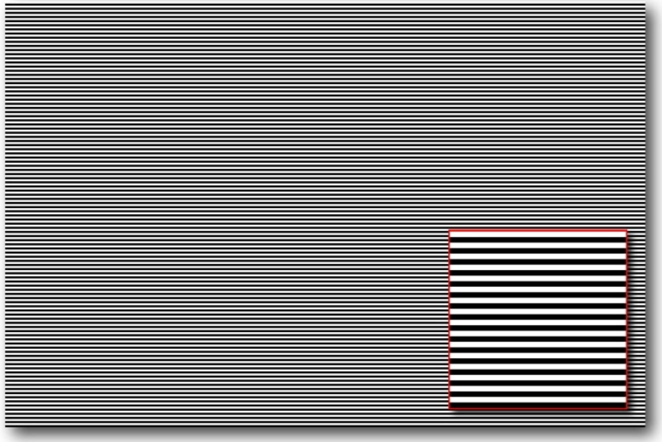


highest-possible-frequency horizontal sinusoid (C is even)

Inverse FFTs of Impulses

"vertical" is the wavefront direction.

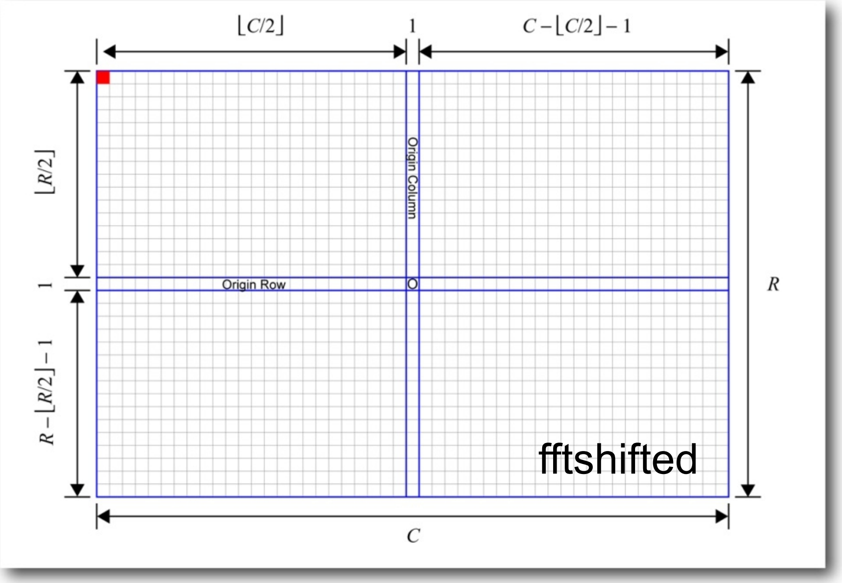


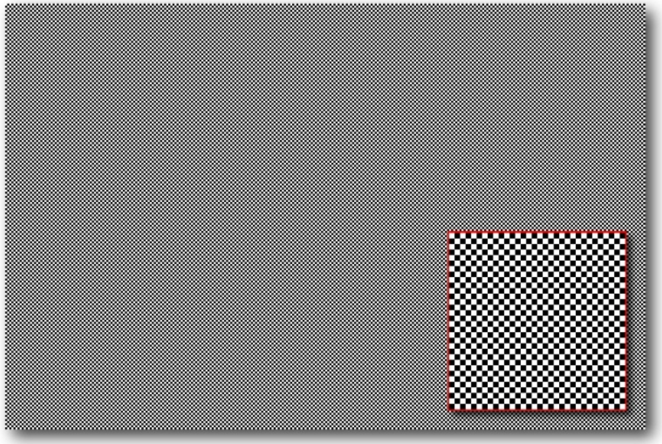


highest-possible-frequency vertical sinusoid (*R* is even)

Inverse FFTs of Impulses

a checker-board pattern.

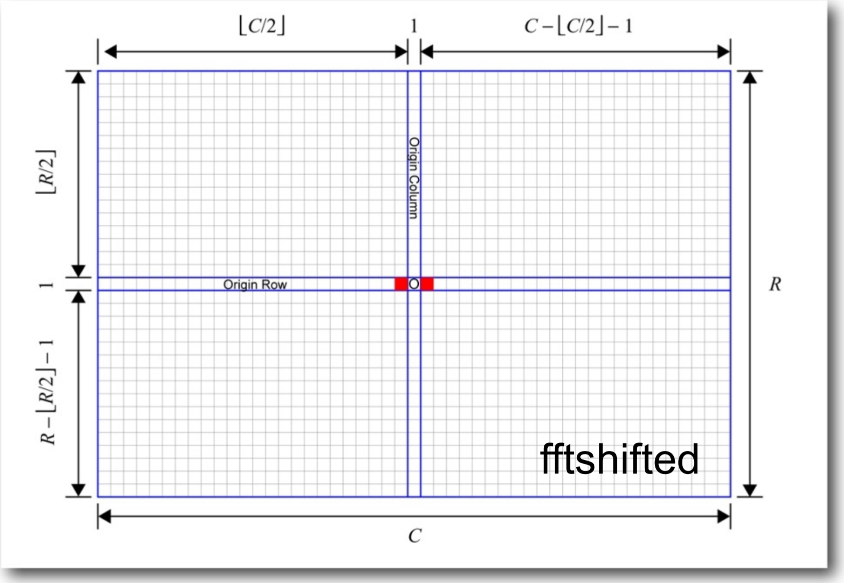


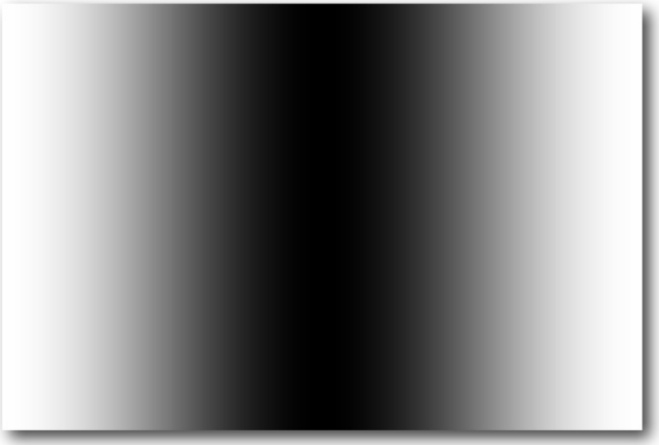


highest-possible-freq horizontal+vertical sinusoid (R & C even)

Inverse FFTs of Impulses

"horizontal" is the wavefront direction.

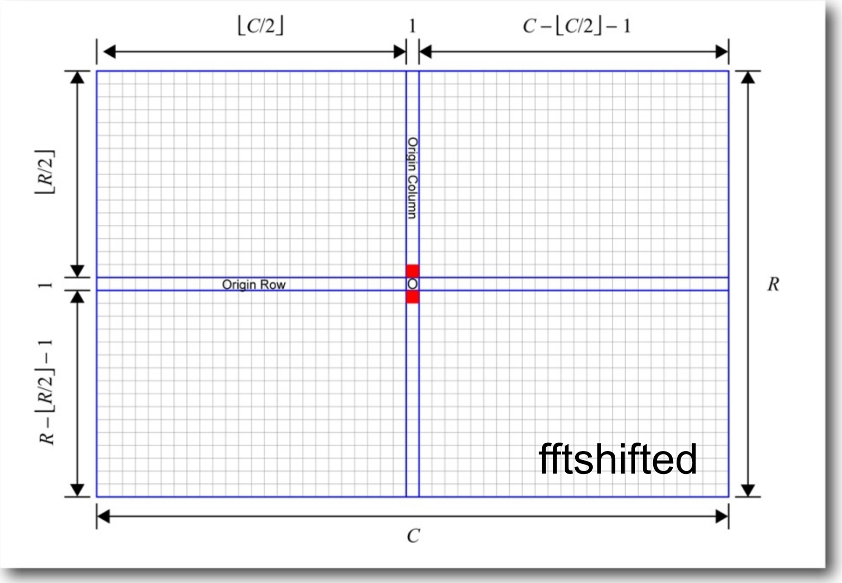


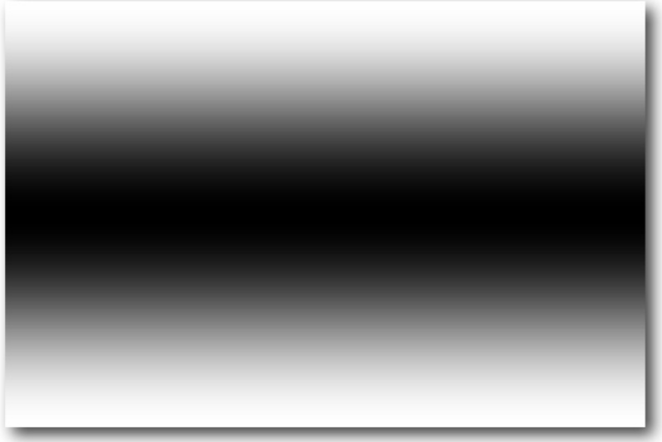


lowest-possible-frequency horizontal sinusoid

Inverse FFTs of Impulses

“vertical” is the wavefront direction.

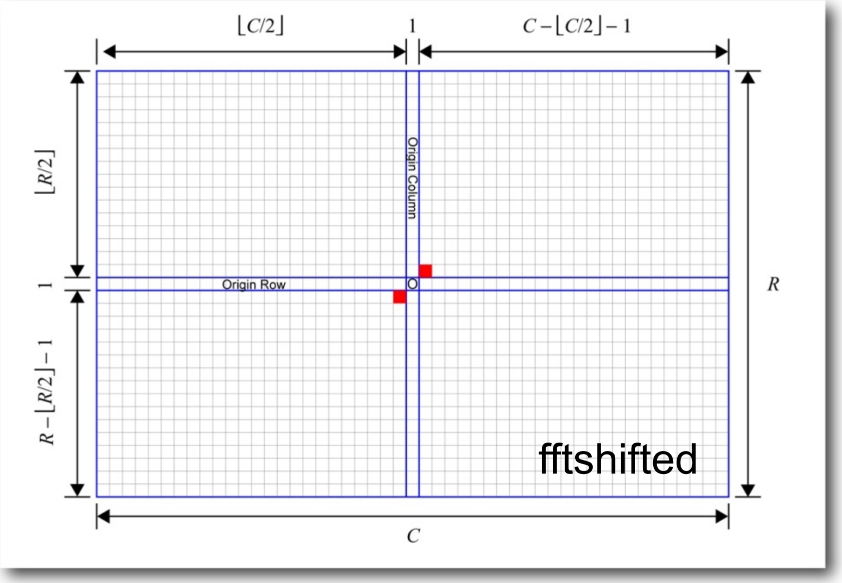


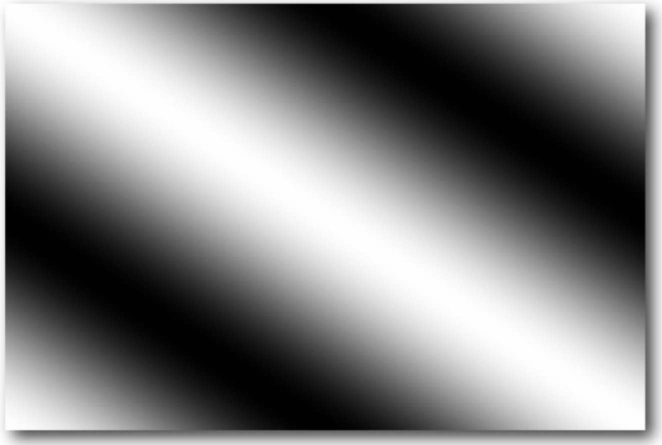


lowest-possible-frequency vertical sinusoid

Inverse FFTs of Impulses

“negative diagonal” is the wavefront direction.

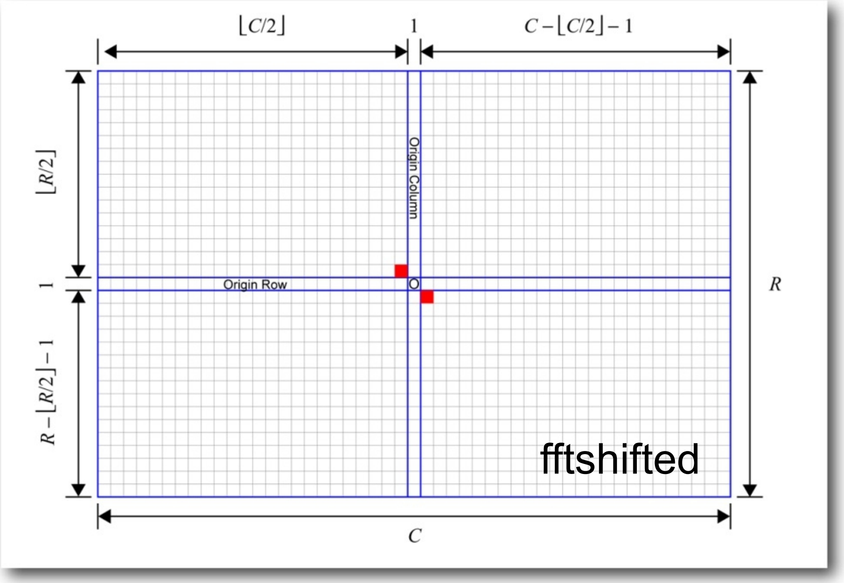


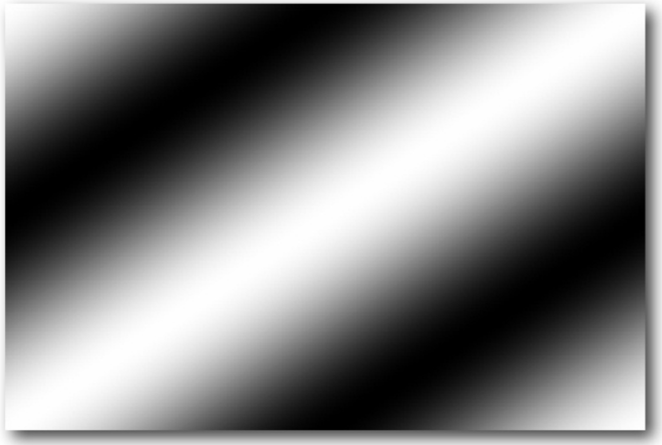


lowest-possible-frequency negative diagonal sinusoid

Inverse FFTs of Impulses

"positive diagonal" is
the wavefront direction.



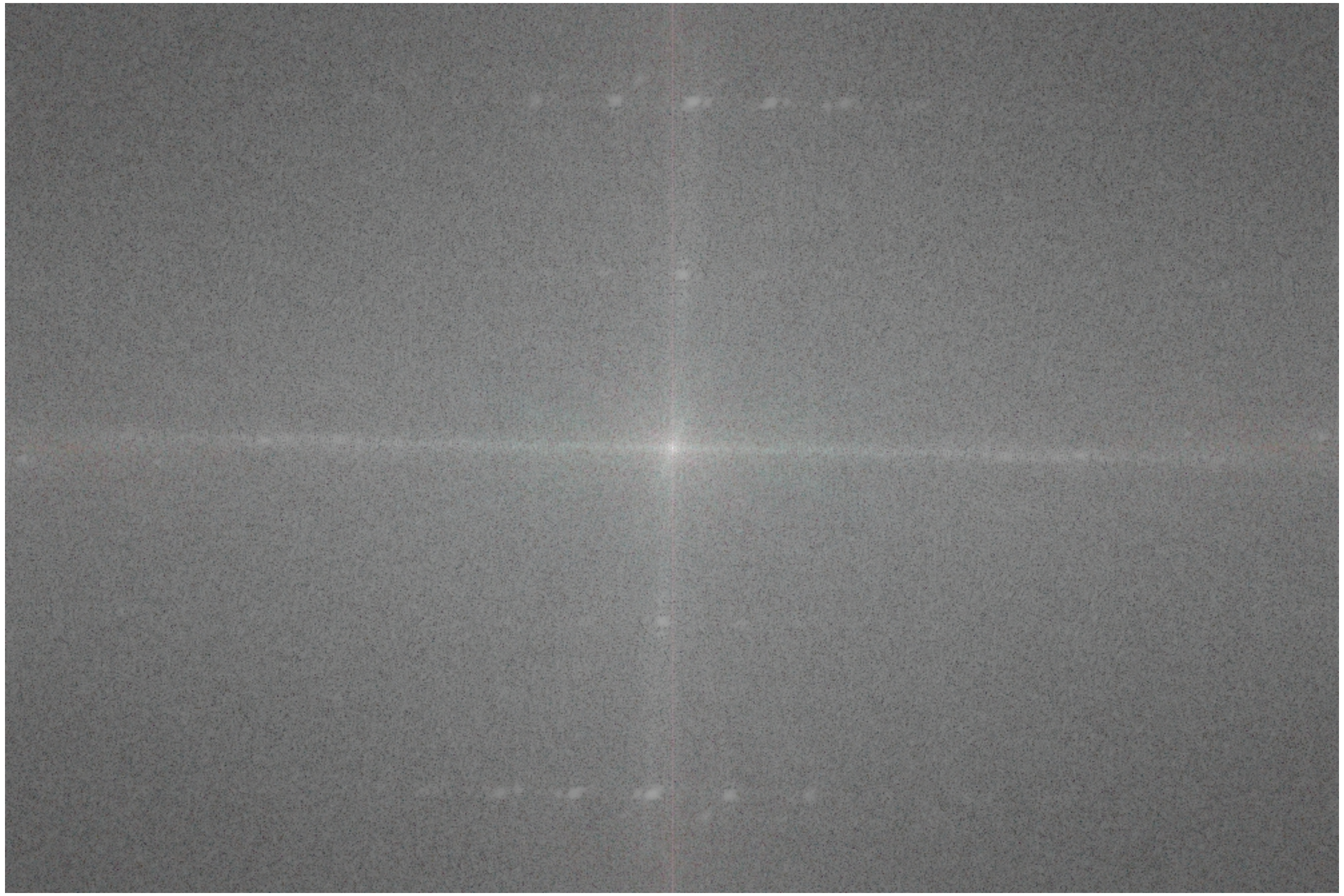


lowest-possible-frequency positive diagonal sinusoid

What's More Important Magnitude or Phase?

- Fourier magnitude

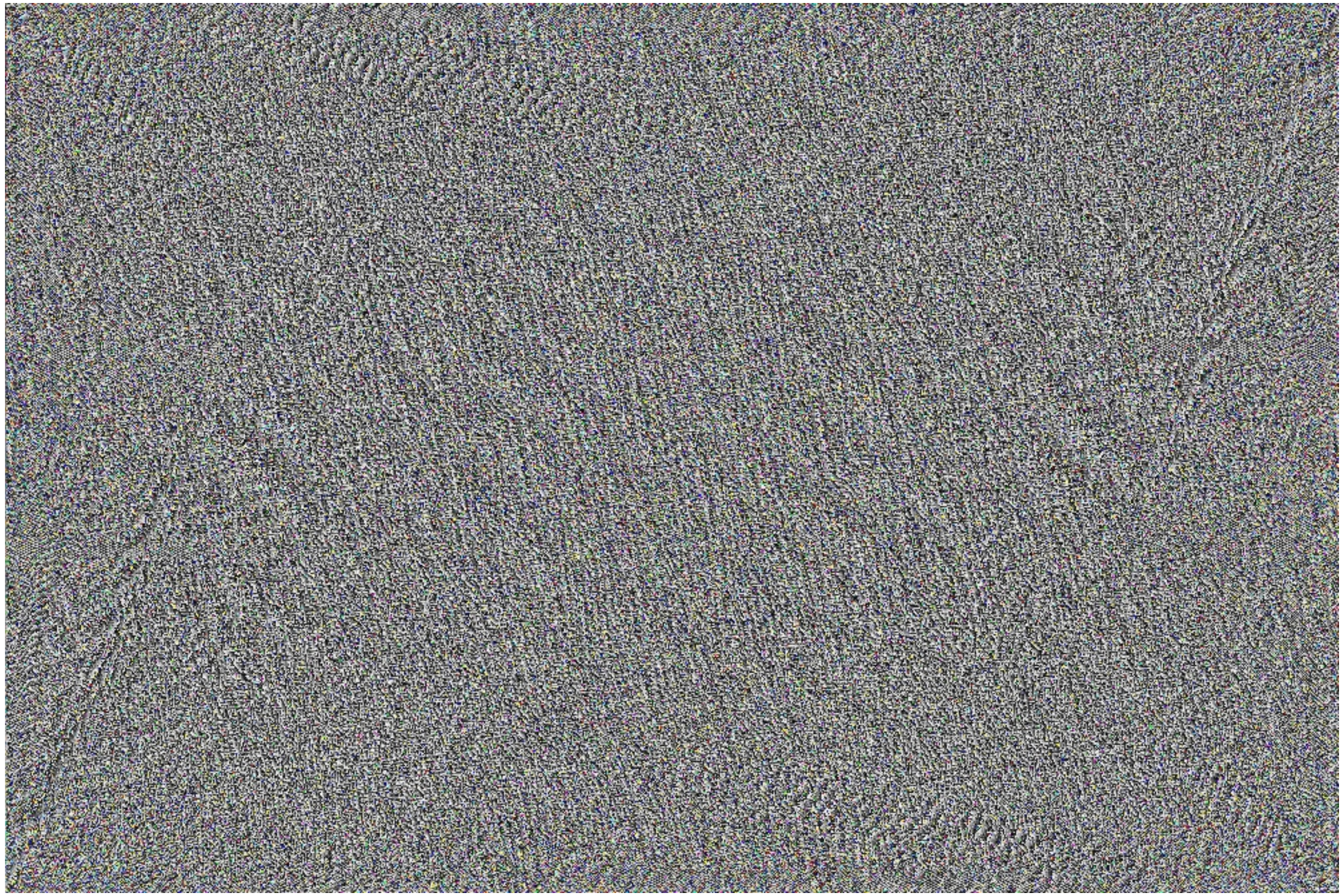
$\log \left| \mathbf{F}\{\mathbf{I}\} \right|$



What’s More Important Magnitude or Phase?

- Fourier phase

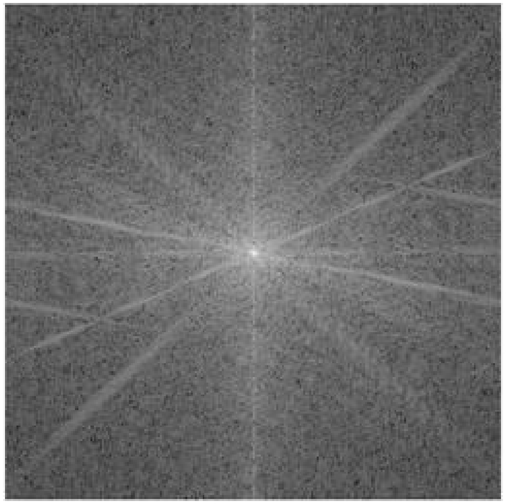
$\angle \mathbf{F}\{\mathbf{I}\}$

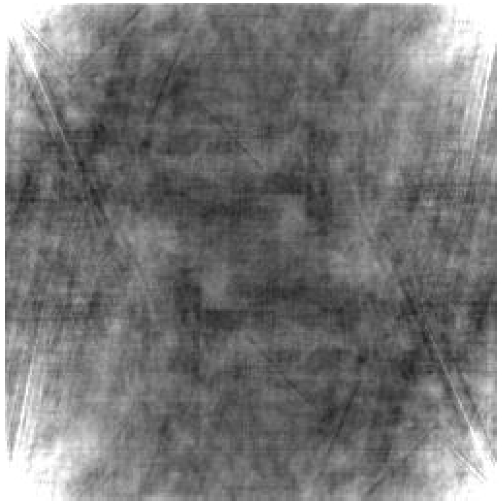


What's More Important Magnitude or Phase?

- Effect of magnitude only (setting phase = 0).


Original Image

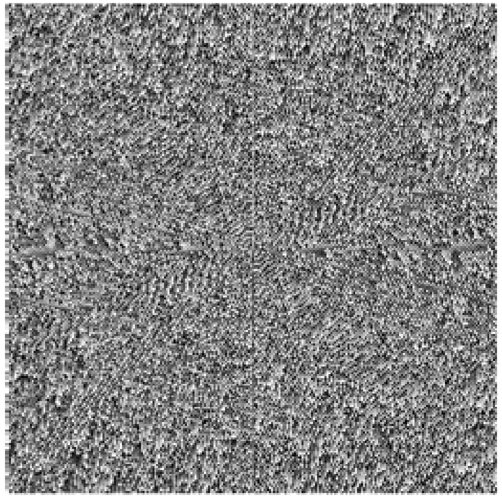

Magnitude response


Reconstruction

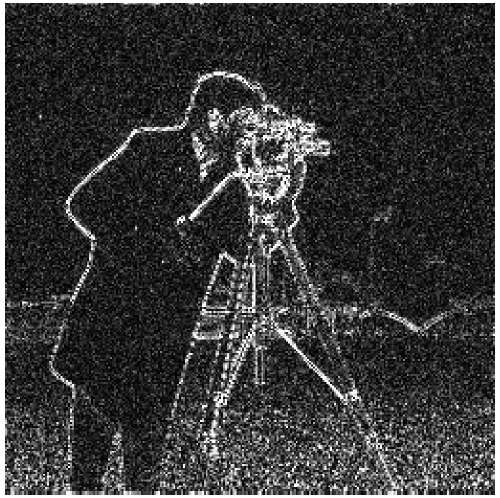
What’s More Important Magnitude or Phase?

- Effect of phase only - set amplitude = constant (80)

Original Image



Phase response



Reconstruction

2D Fourier Filtering

Filtering in Frequency Domain

- Hence, in the case $h(x, y)$ is given in spatial domain we can perform filtering operation by first transform both $f(x, y)$ and $h(x, y)$ into $F(u, v)$ and $H(u, v)$ respectively. Multiply the two to obtain $G(u, v)$.

$$G(u, v) = F(u, v)H(u, v)$$

- Finally take the inverse DFT of $G(u, v)$ to obtain $g(x, y)$.
- However, because of periodicity when taking DFT we got to avoid wraparound error or aliasing.

2D Fourier Filtering

- zeropad both $f(x,y)$ and $h(x,y)$ so that their size is now $P \times Q$ where P and Q must satisfy the following

$$(P,Q) \geq \underbrace{(M,N)}_{size\ of\ f} + \underbrace{(C,D)}_{size\ of\ h} - (1,1)$$

  thus choose

$$P = M + C - 1 \text{ and } Q = N + D - 1$$

- Note: to center the DFT, both zeropaded $f(x,y)$ and $h(x,y)$ must be multiplied by $(-1)^{x+y}$. Likewise $g(x,y)$ must also be multiplied by $(-1)^{x+y}$.

2D Fourier Filtering

- *Basic steps for filtering in the frequency domain*

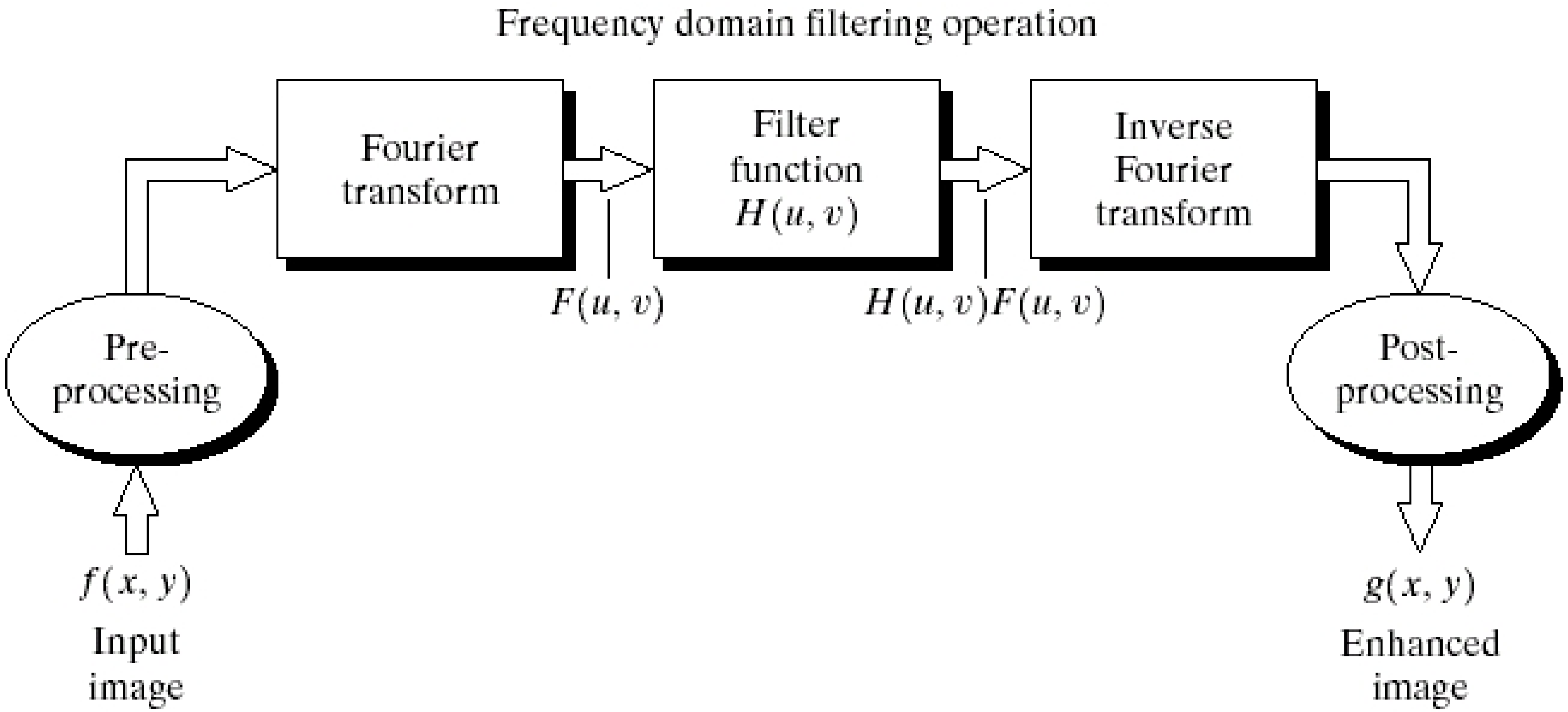


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Frequency Domain Filtering

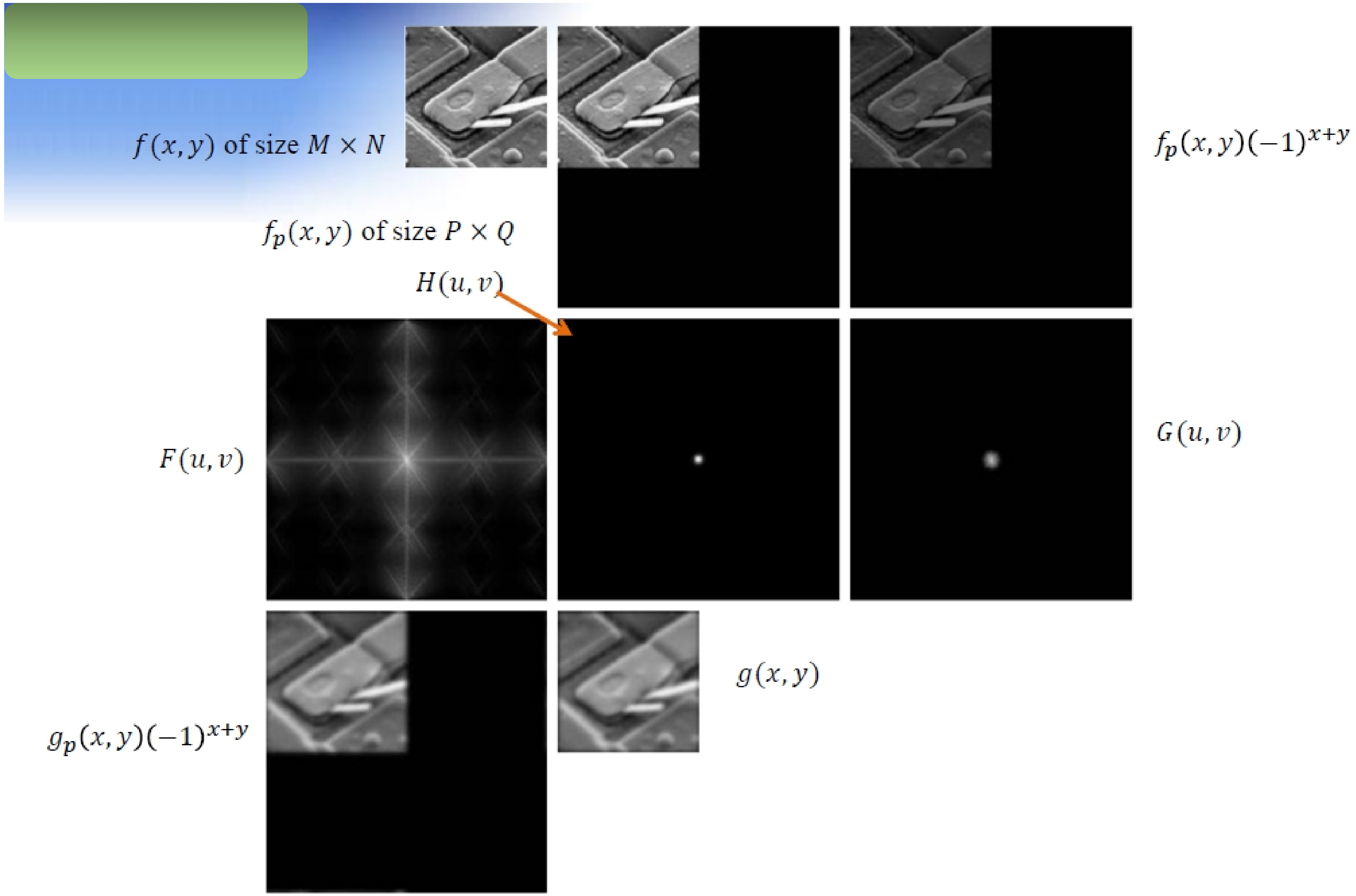
- If the filter is directly design in the frequency domain i.e. $H(u, v)$ then the following steps can be used

1. Given $f(x, y)$ of size $M \times N$, pad it with zeros to obtain $f_p(x, y)$ whose size is $P \times Q$. In this case set $P = 2M$ and $Q = 2N$.
2. To centre the transform at $\left(\frac{P}{2}, \frac{Q}{2}\right)$ perform $f_p(x, y)(-1)^{x+y}$.
3. Compute the DFT $F(u, v)$ of the image.
4. Generate the filter transfer function $H(u, v)$ of the size $P \times Q$ and centred at $\left(\frac{P}{2}, \frac{Q}{2}\right)$.

Frequency Domain Filtering

5. Perform the array multiplication to get the output i.e. $G(u, v) = F(u, v)H(u, v)$.
6. Perform the IDFT to get $g_p(x, y)$ and reverse the shift by multiplying it with $(-1)^{x+y}$. Note: $g_p(x, y)$ is of the size $P \times Q$.
7. Finally extract $M \times N$ array from the top left of $g_p(x, y)$ in order to get $g(x, y)$ – the desired output. Note: take only the real part.

2D Fourier Filtering



2D Fourier Filtering

Smoothing (Lowpass Filter)

- Retain low frequency component of the image – uniform or “constant” pixel areas
- Removes or attenuate high frequency component – details of an object such as edges and boundaries
- In the frequency domain

$$G(u,v) = F(u,v)H(u,v)$$ - array multiplication

- $F(u,v)$ is the 2-D F.T. of the image
- $H(u,v)$ is the filter transfer function

2D Fourier Filtering

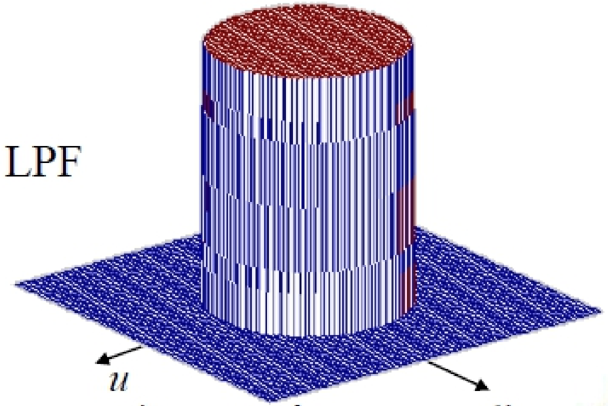
Smoothing (Lowpass Filter)

Ideal Lowpass Filter

$$H(u,v) = \begin{cases} 1 & if\ D(u,v) \leq D_0 \\ 0 & if\ D(u,v) > D_0 \end{cases}$$

- $D(u,v)$ is the filter characteristic describing the shape of the filter: circular, ellipse etc.

Circular-shape LPF



2D Fourier Filtering

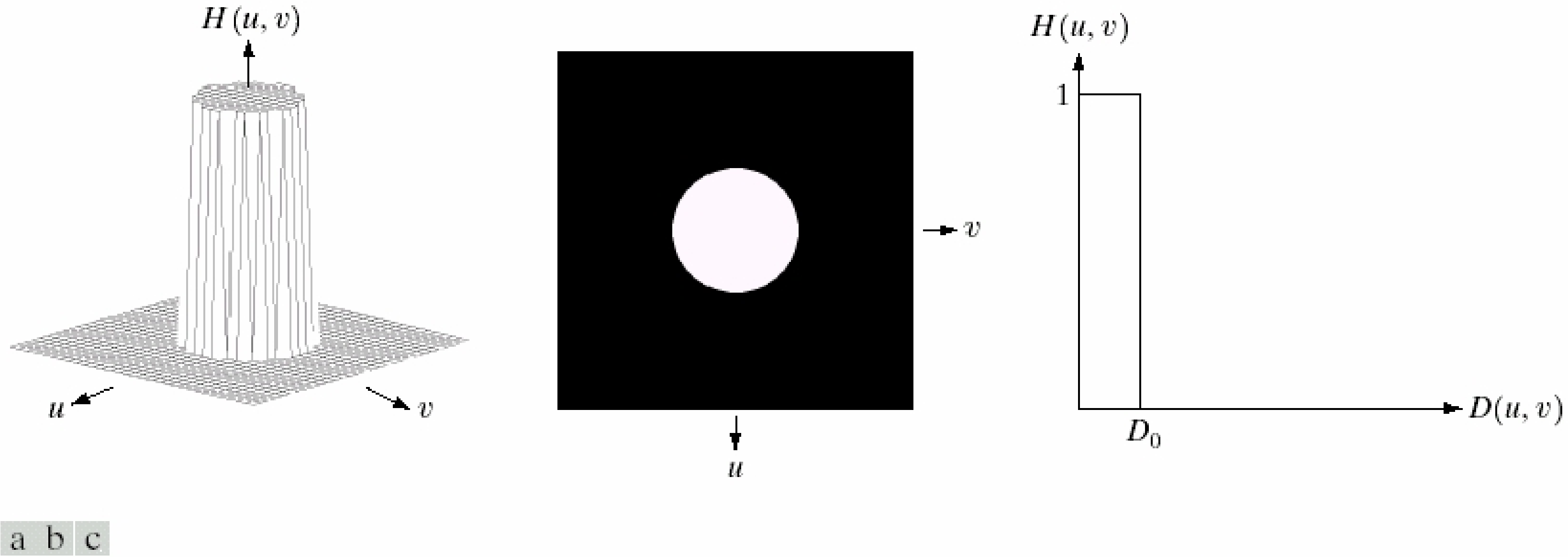


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Smoothing (Lowpass Filter)

- For circular-shaped and centred at $\left(\frac{P}{2}, \frac{Q}{2}\right)$:

$$D(u,v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

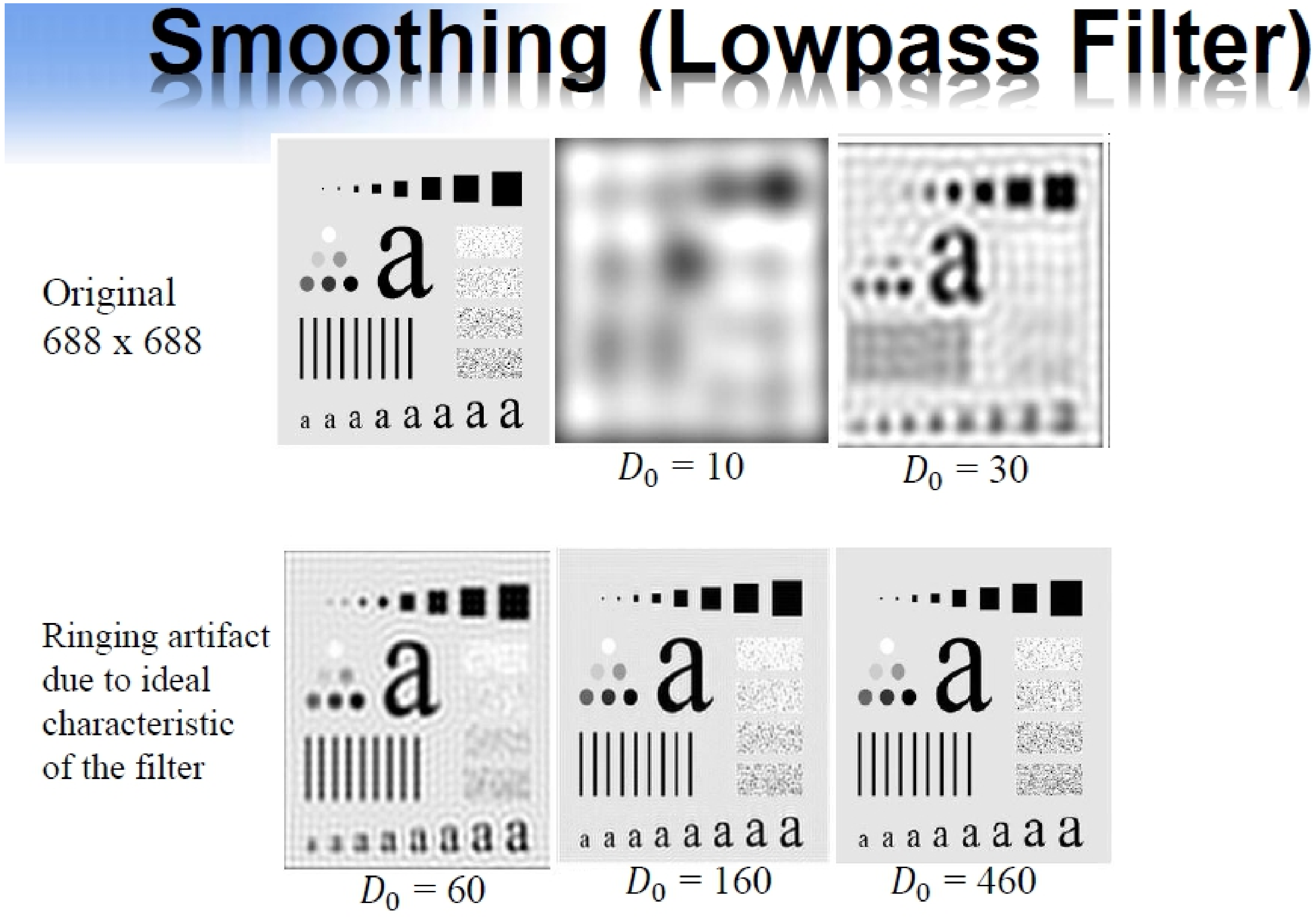
- D_0 is the filter radius (cutoff point) from the centre of the filter.
- Hence, $G(u,v) = \begin{cases} F(u,v), & H(u,v) = 1 \\ 0, & H(u,v) = 0 \end{cases}$
- Inverse transform will yield $g(x,y)$ – the desired image.

2D Fourier Filtering

Smoothing (Lowpass Filter)

- Ideal filters – sharp cutoff (transition) from passband to stopband.
- Cannot be implemented with hardware components.
- Analysis – effect on performing the filter for various D_0 values.

2D Fourier Filtering



2D Fourier Filtering

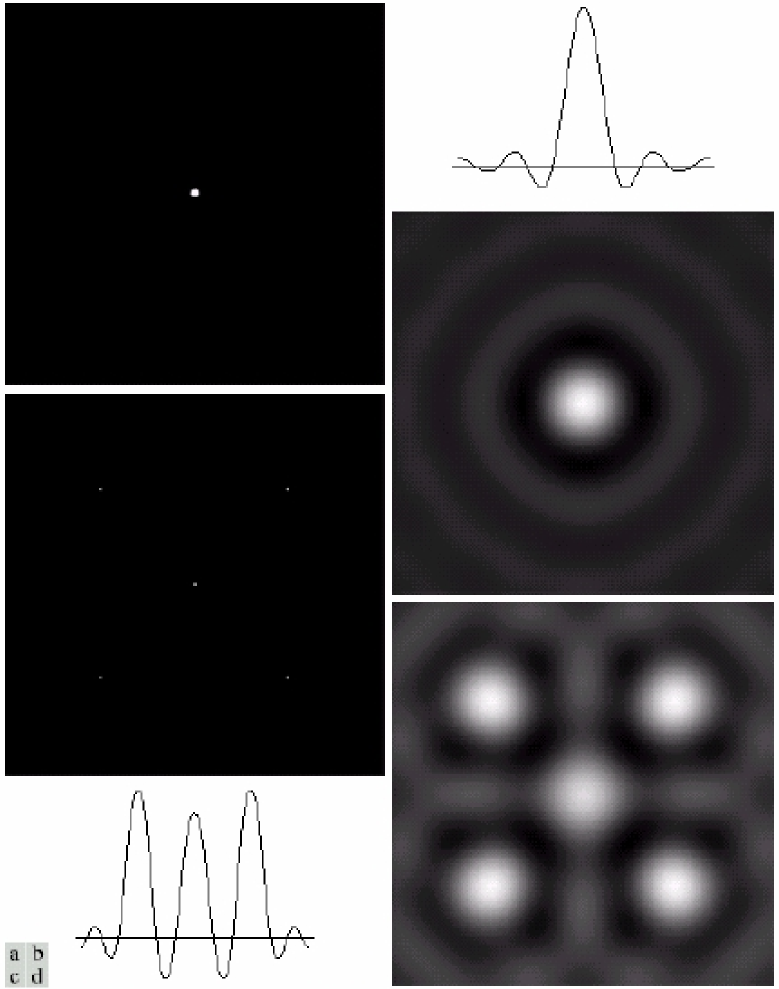


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Smoothing (Lowpass Filter)

Butterworth Lowpass Filter

- General equation

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

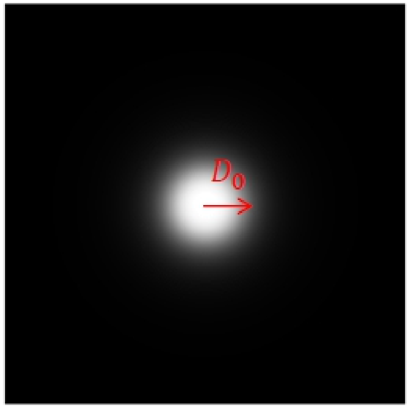
- n – order of the filter.
- D_o – cutoff frequency locus (distance from the origin)
- $D(u,v)$ – filter characteristic as before

2D Fourier Filtering

Smoothing (Lowpass Filter)

- $D(u,v) = (u^2 + v^2)^{\frac{1}{2}} \leftarrow$ circular shape
- In our case for shifted (centred) filter

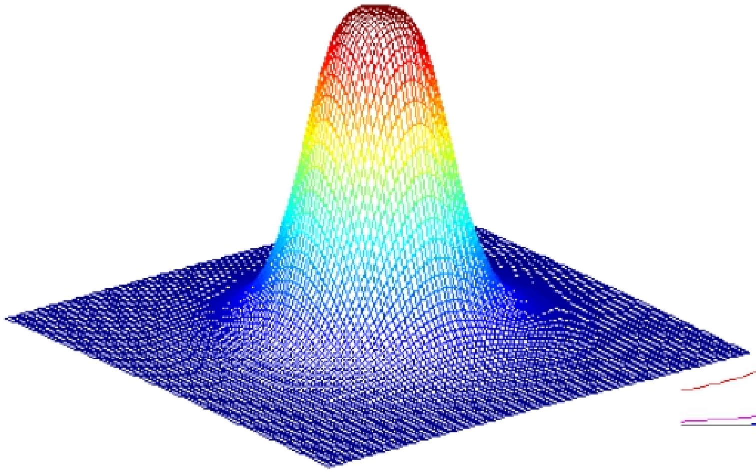
$$D(u,v) = \left(\left[u - \frac{P}{2}\right]^2 + \left[v - \frac{Q}{2}\right]^2\right)^{\frac{1}{2}}$$



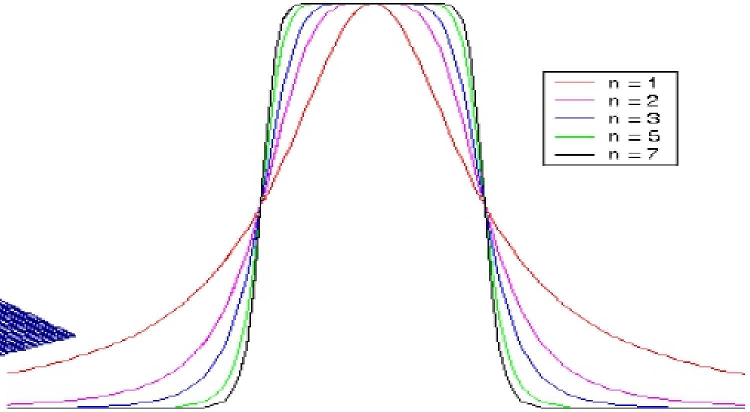
2D Fourier Filtering

Smoothing (Lowpass Filter)

Perspective and cross section profile of BLPF



Perspective Plot $n = 2$



Radial cross section

2D Fourier Filtering

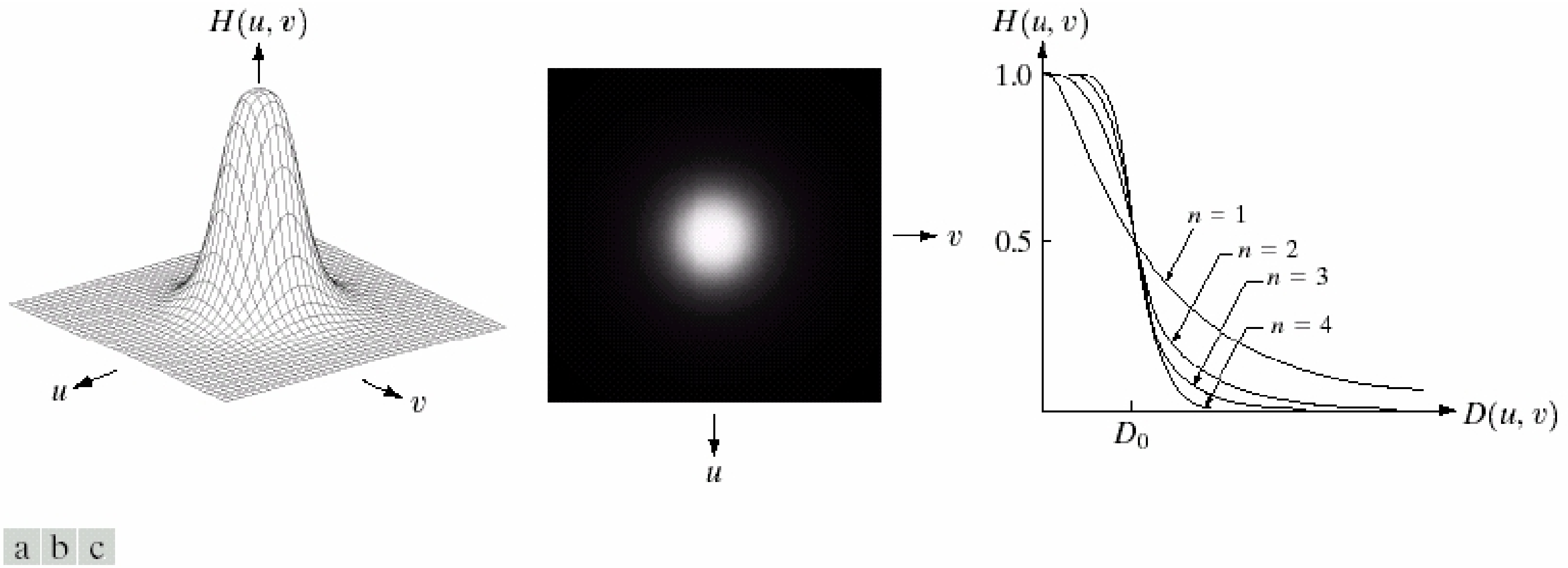


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

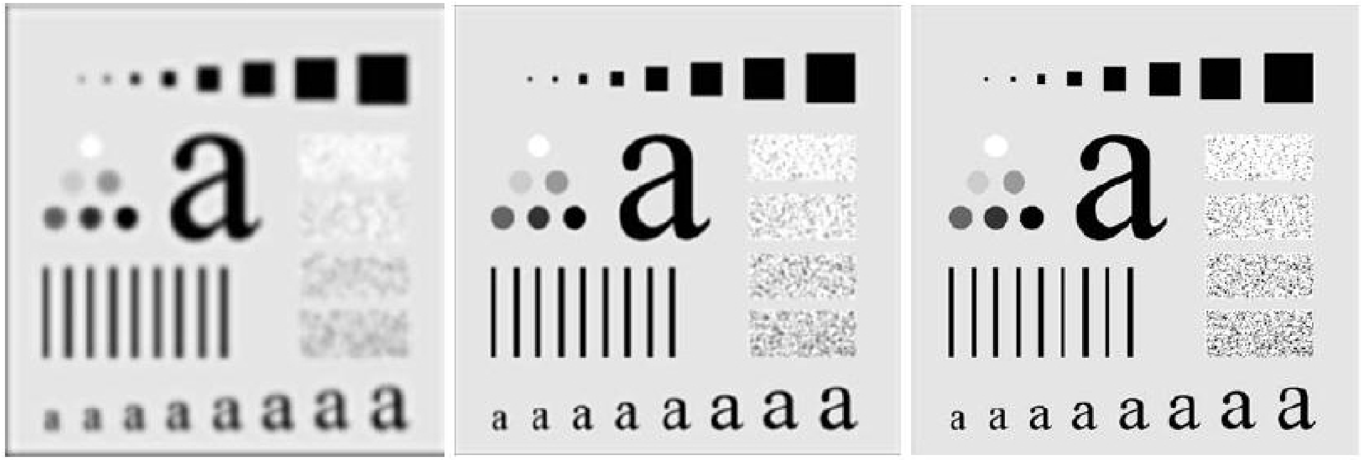
Smoothing (Lowpass Filter)

- Effect of applying different values of D_0 (5 values) with $n = 2$.

$D_0 = 10$ $D_0 = 30$

2D Fourier Filtering

Smoothing (Lowpass Filter)



$D_0 = 60$ $D_0 = 160$ $D_0 = 460$

Ringing artifact is not visible

2D Fourier Filtering

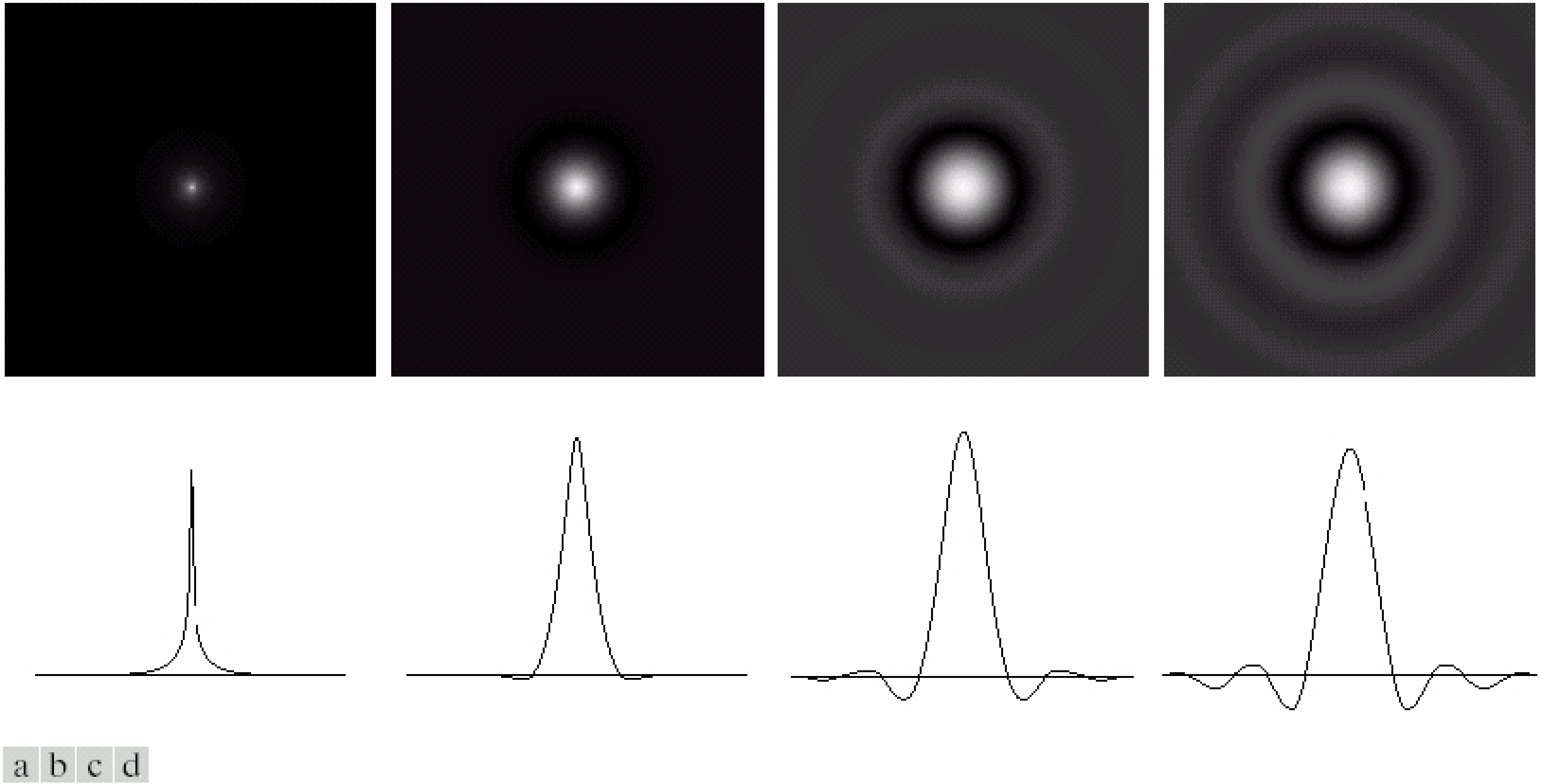


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

2D Fourier Filtering

Smoothing (Lowpass Filter)

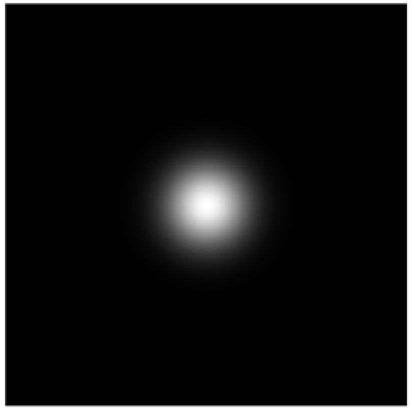
Gaussian Lowpass Filter

- General equation

$$H(u,v) = e^{-\frac{D^2(u,v)}{2D_0^2}}$$

- Less control parameter – no filter order
- Less smoothing effect
- Centred version

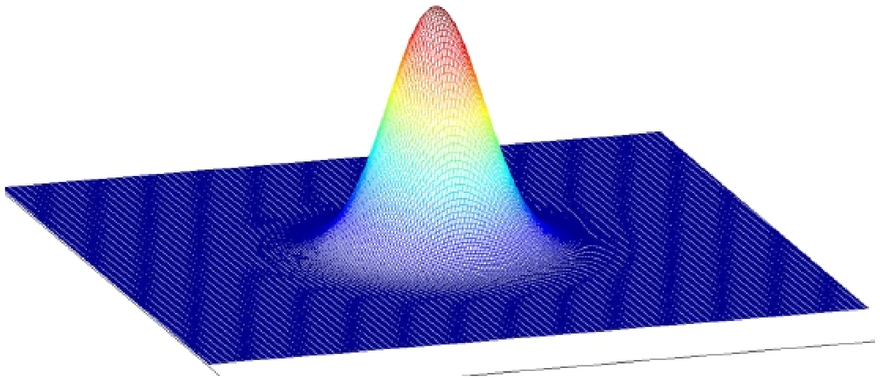
$$H(u,v) = e^{-\frac{\left(u-\frac{P}{2}\right)^2+\left(v-\frac{Q}{2}\right)^2}{2D_0^2}}$$

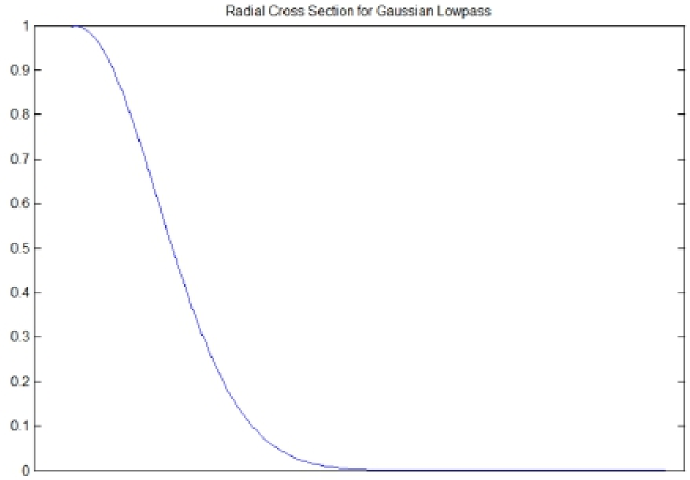


2D Fourier Filtering

Smoothing (Lowpass Filter)

- Perspective and cross section profile of GLPF





2D Fourier Filtering

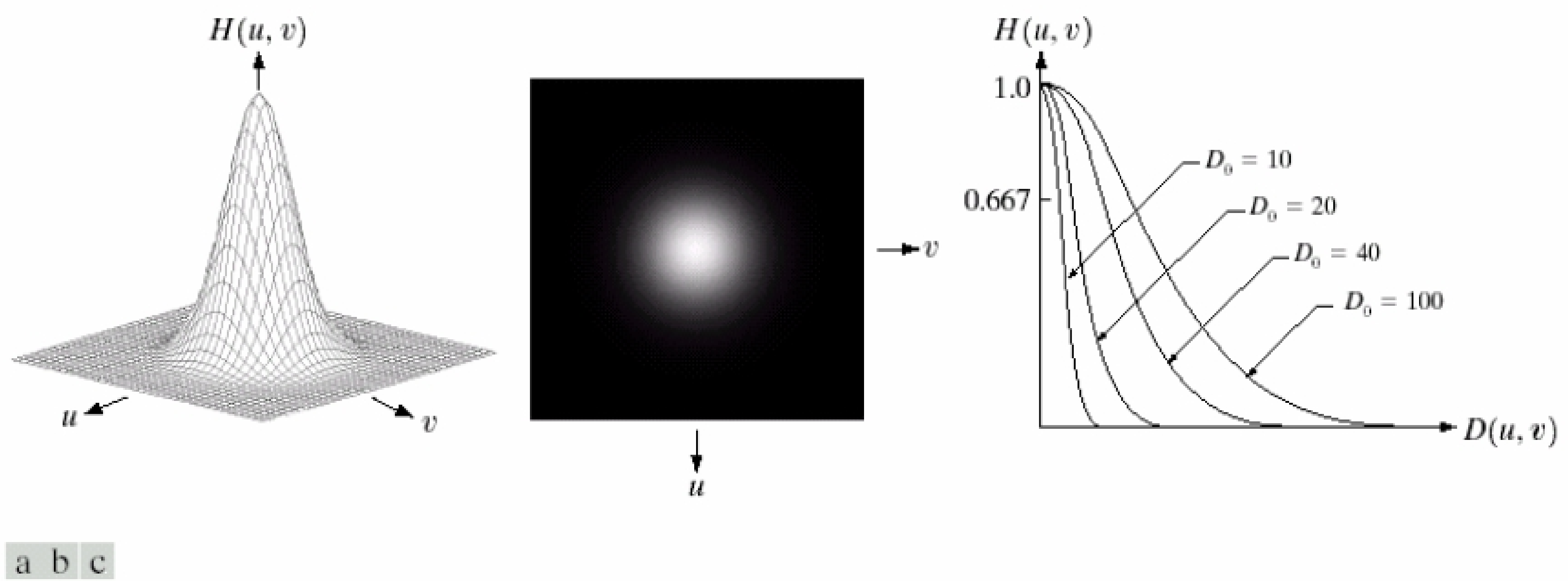


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Smoothing (Lowpass Filter)

$D_0 = 10$ $D_0 = 30$

2D Fourier Filtering

Smoothing (Lowpass Filter)

$D_0 = 60$ $D_0 = 160$ $D_0 = 460$

Ringing artifact is not visible

Image is somewhat less blurred compared to that of BLPF

2D Fourier Filtering

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

Sharpening (Highpass) Filtering

- Image sharpening can be achieved by a highpass filtering process, which attenuates the low-frequency components without disturbing high-frequency information.

- Zero-phase-shift filters: radially symmetric and completely specified by a cross section.

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

2D Fourier Filtering

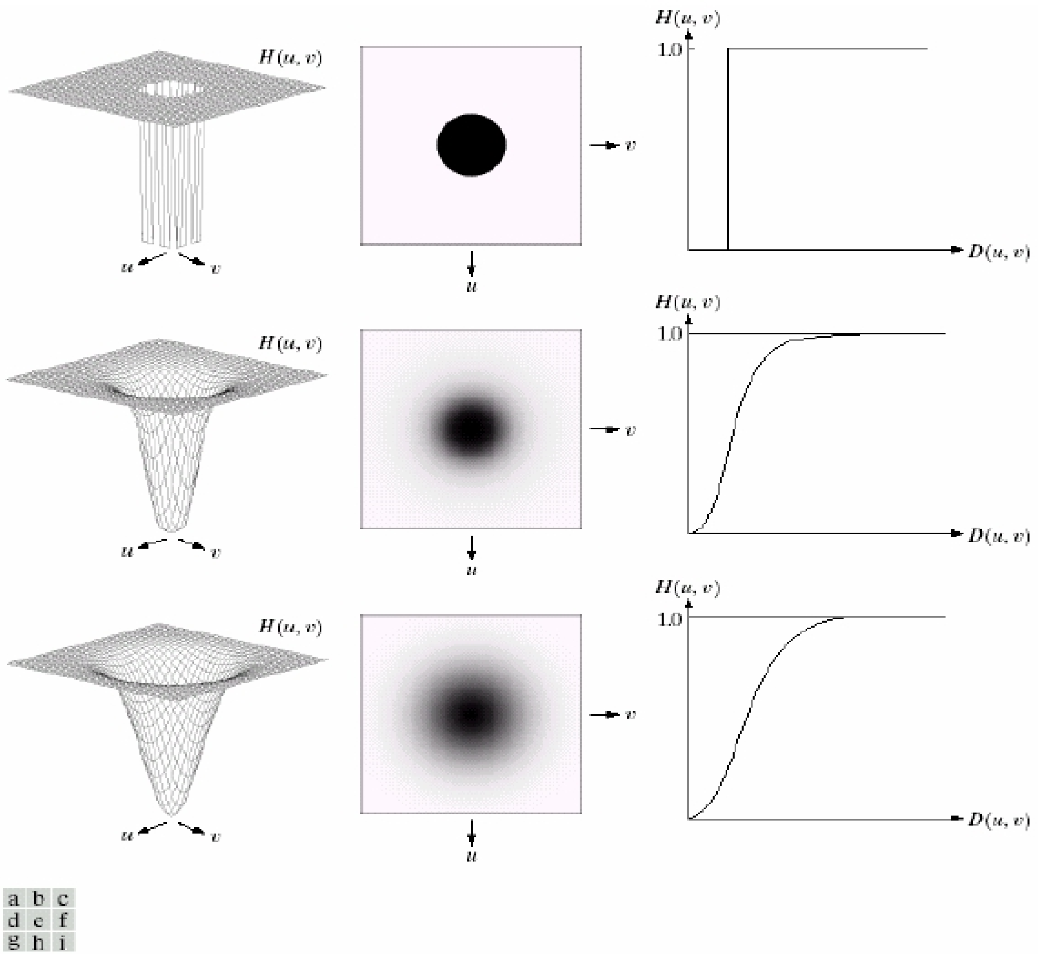


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Sharpening (Highpass Filter)

- Attenuate low-frequency components without disturbing high-frequency components.
- Retain the detail of the image information such as edges, fine texture etc.
- Highpass filter can be simply generated from a lowpass filter i.e.

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

Sharpening (Highpass Filter)

Ideal Highpass Filter

$$H(u,v) = \begin{cases} 0 & if\ D(u,v) \leq D_0 \\ 1 & if\ D(u,v) > D_0 \end{cases}$$

- $D(u,v)$ – filter characteristic
- D_0 - cutoff point

Sharpening (Highpass Filter)

Gaussian Highpass

- Mathematical expression

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$

For circular-shaped filter.

$$D(u,v) = \left(\left[u - \frac{P}{2}\right]^2 + \left[v - \frac{Q}{2}\right]^2\right)^{\frac{1}{2}}$$

2D Fourier Filtering

Sharpening (Highpass Filter)

- Effects of using different D_0 for different filters

Ideal HPF

$D_0 = 30$ $D_0 = 60$ $D_0 = 160$

2D Fourier Filtering

Sharpening (Highpass Filter)

Butterworth HPF $(n = 2)$

$D_0 = 30$ $\quad$ $D_0 = 60$ $\quad$ $D_0 = 160$

2D Fourier Filtering

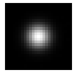
Sharpening (Highpass Filter)

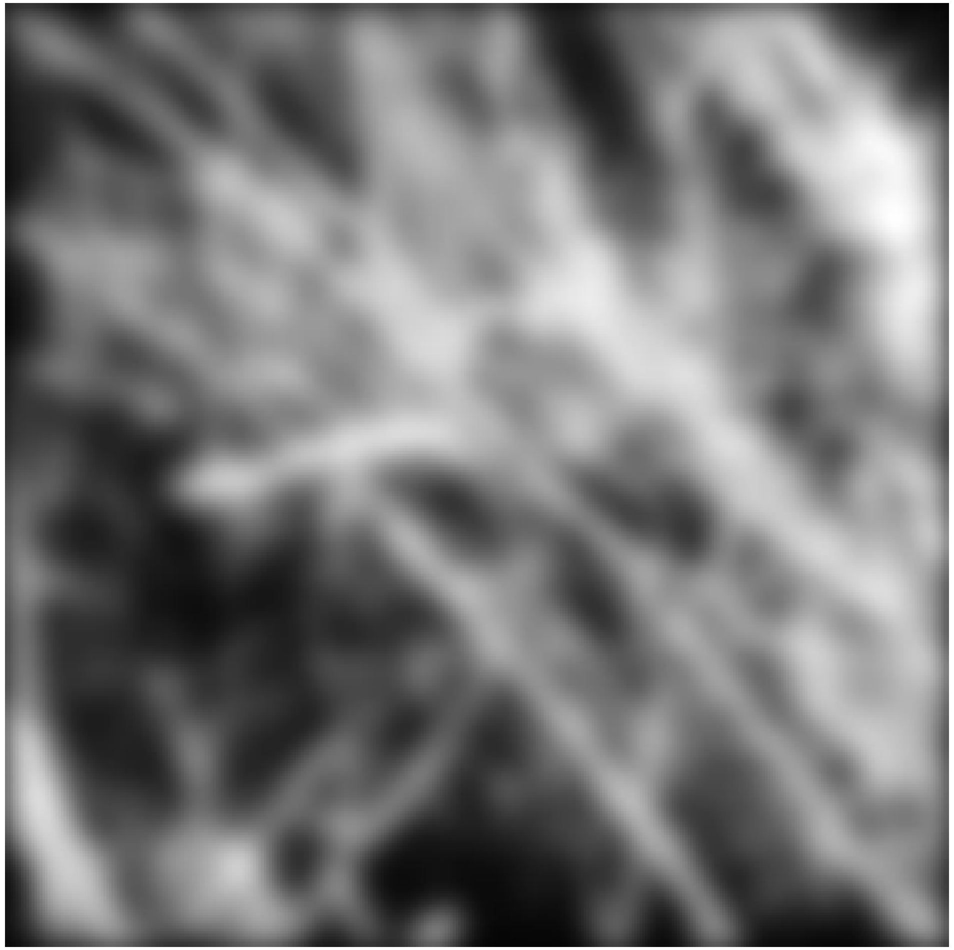
Gaussian HPF

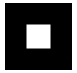
$D_0 = 30$ $D_0 = 60$ $D_0 = 160$

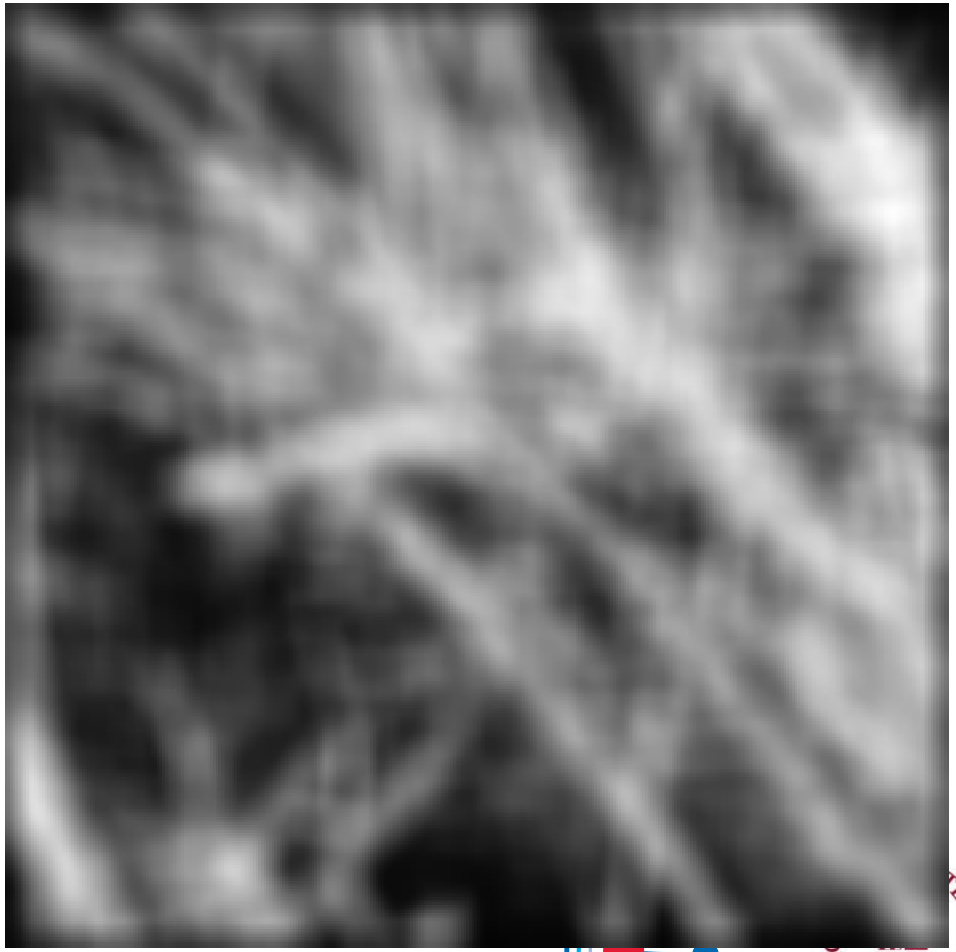
2D Fourier Filtering

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

Gaussian 

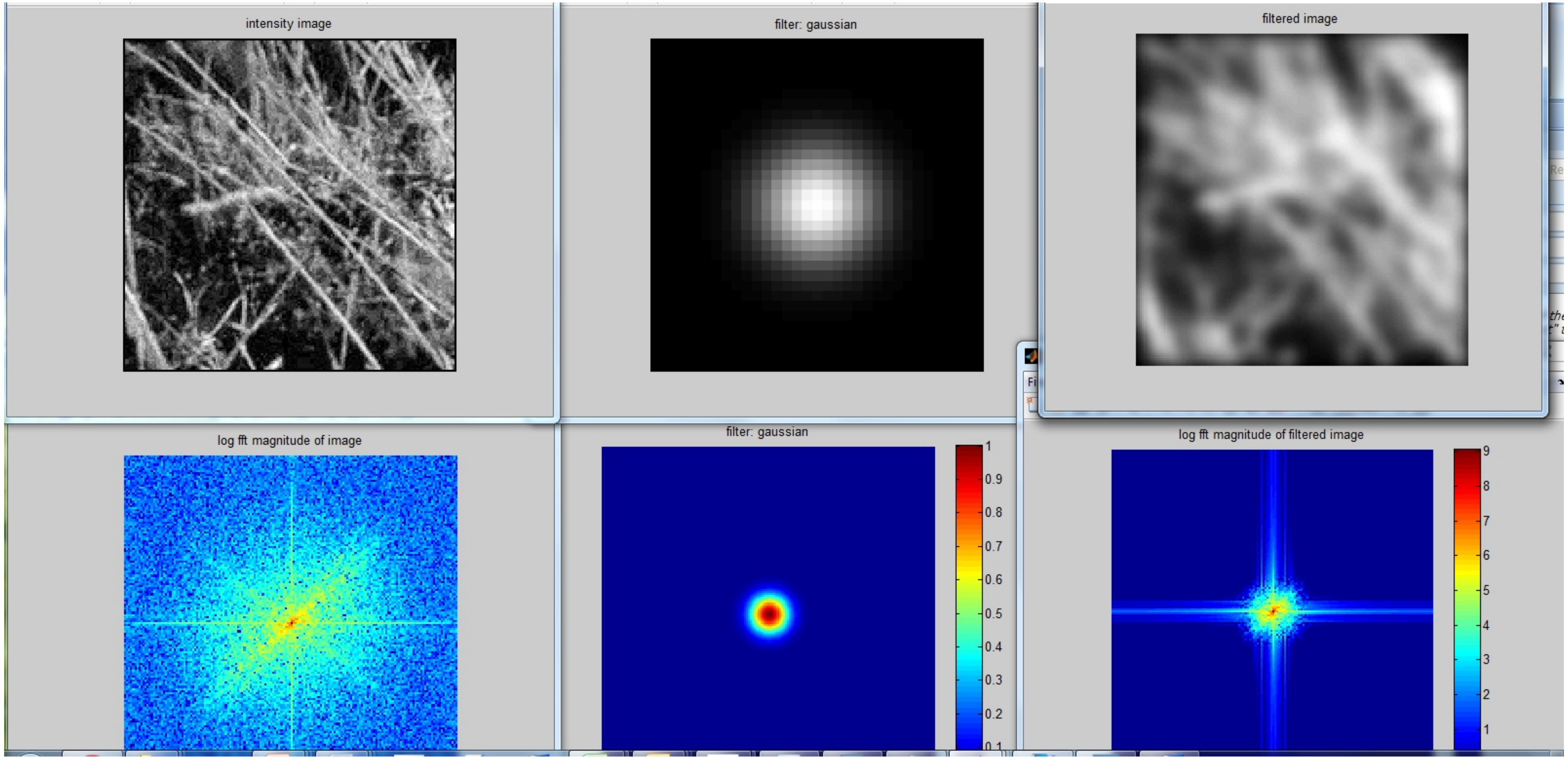


Box filter 



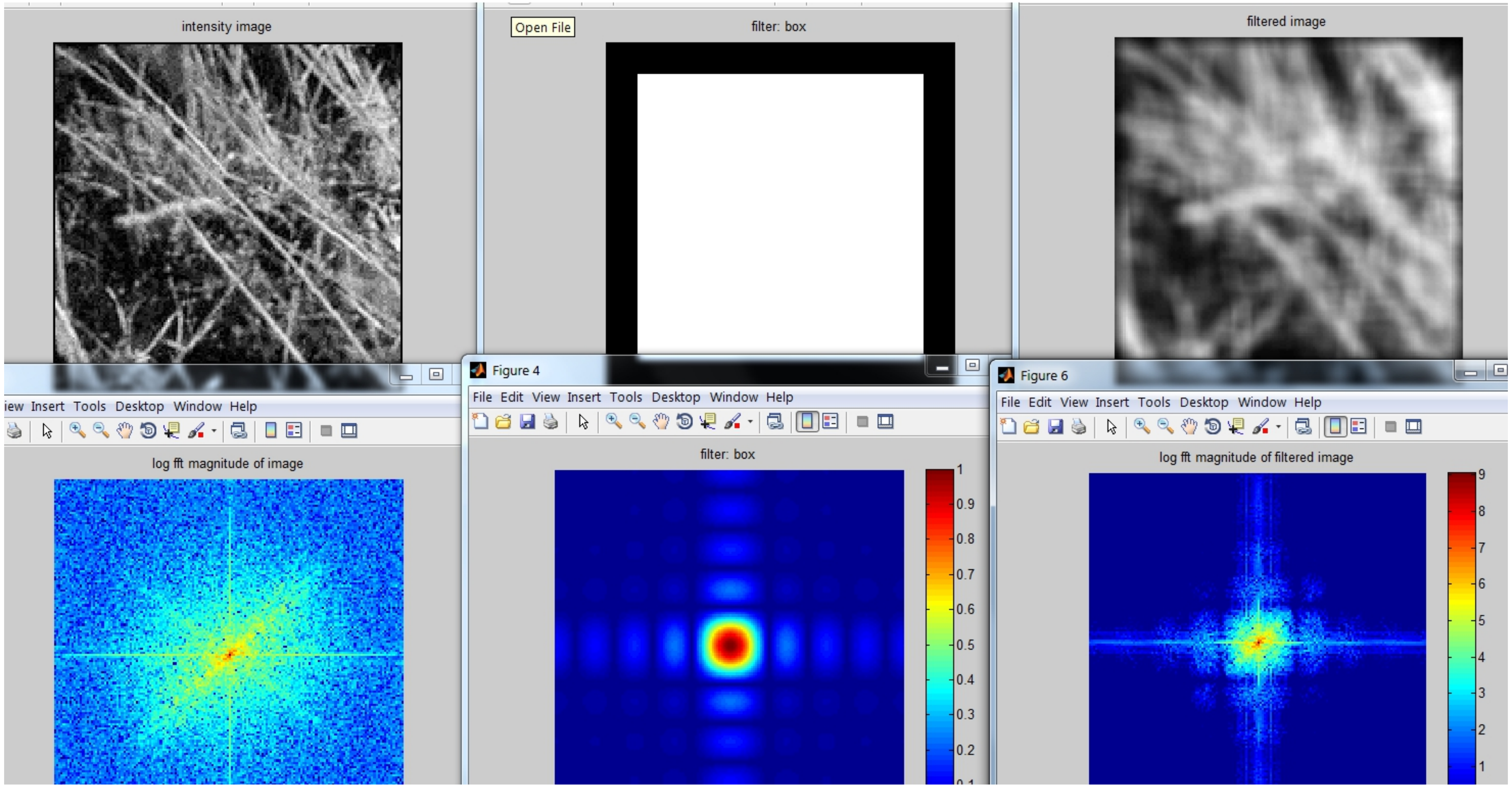
2D Fourier Filtering

Gaussian



2D Fourier Filtering

Box Filter

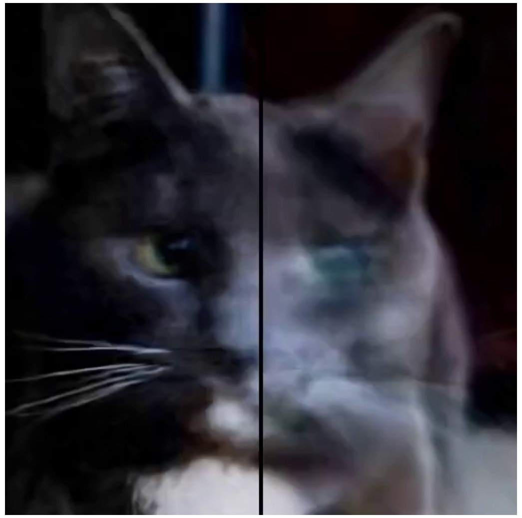


DL + Frequency

Are Fourier analysis also present today in Deep Learning?

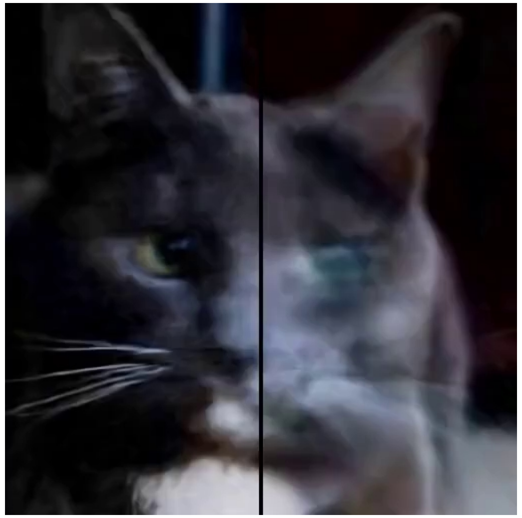
- **Representation learning & feature extraction**:

- "Fast Fourier Convolution", NeurIPS, 2020
- "Implicit Neural Representations with Periodic Activation Functions", NeurIPS, 2020



DL + Frequency

Are Fourier analysis also present today in Deep Learning?



- **Representation learning & feature extraction**:

- "**Fast Fourier Convolution**", NeurIPS, 2020

- "**Implicit Neural Representations with Periodic Activation Functions**", NeurIPS, 2020

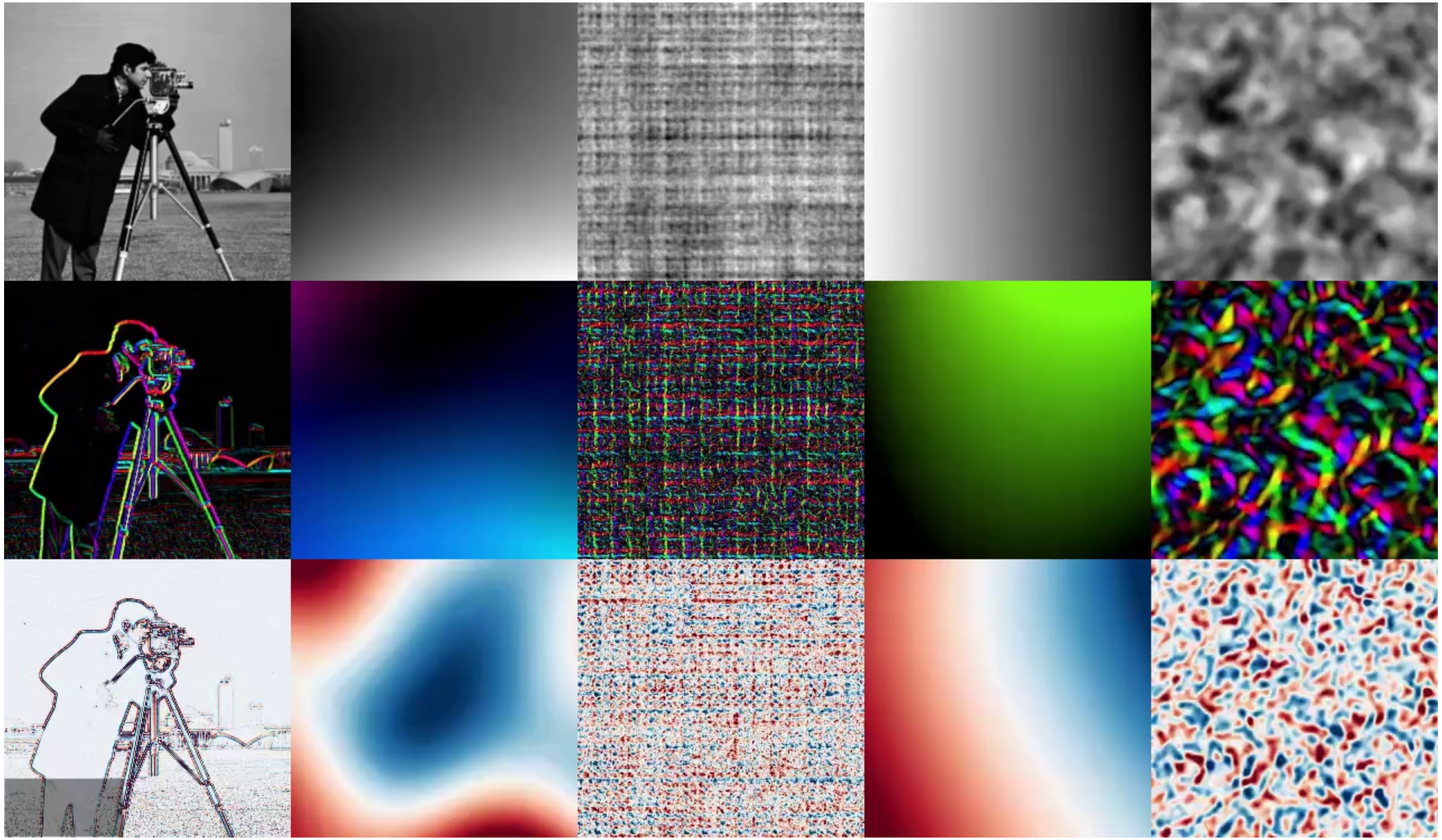
- **Positional encoding:**

- Neural rendering (NeRFs) and Fourier features: "**NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**", ECCV, 2020.

- "**Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains**", NeurIPS, 2020

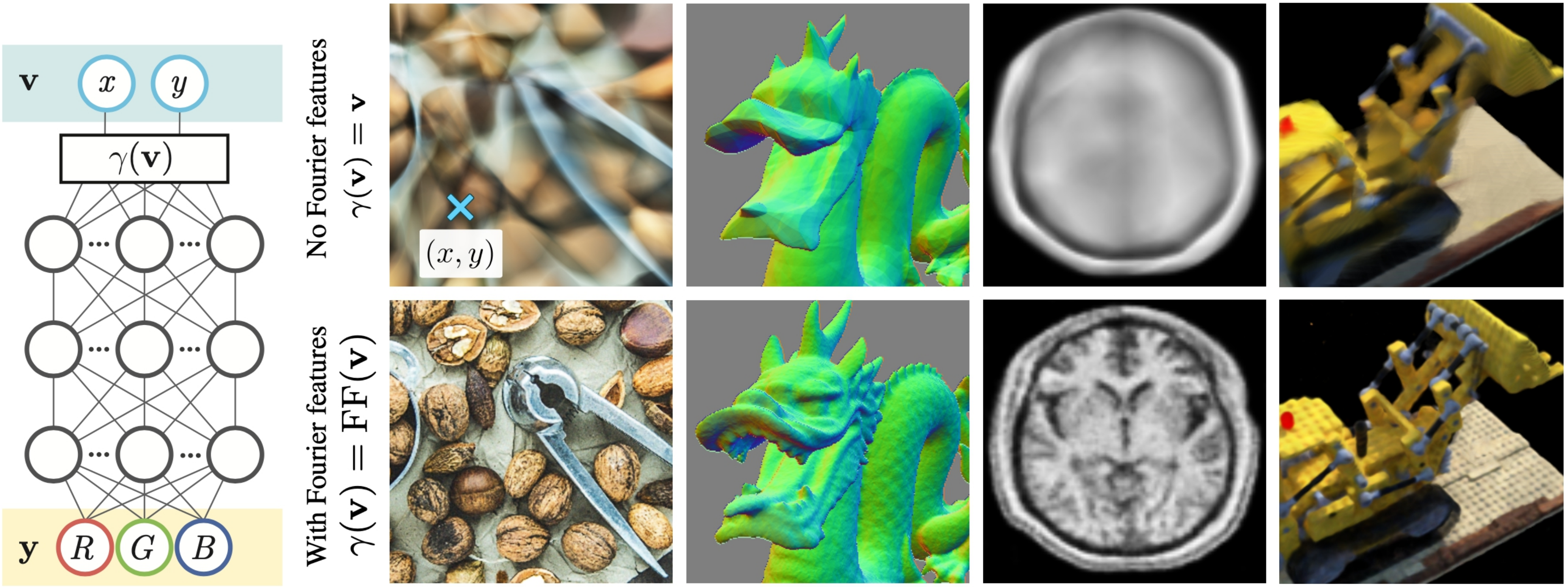
- And in most **Transformers'** architecturers...

DL + Frequency



Sitzmann et al. “**Implicit Neural Representations with Periodic Activation Functions**”, NeurIPS 2020.

DL + Frequency



(a) Coordinate-based MLP

(b) Image regression $(x,y) \rightarrow$ RGB

(c) 3D shape regression $(x,y,z) \rightarrow$ occupancy

(d) MRI reconstruction $(x,y,z) \rightarrow$ density

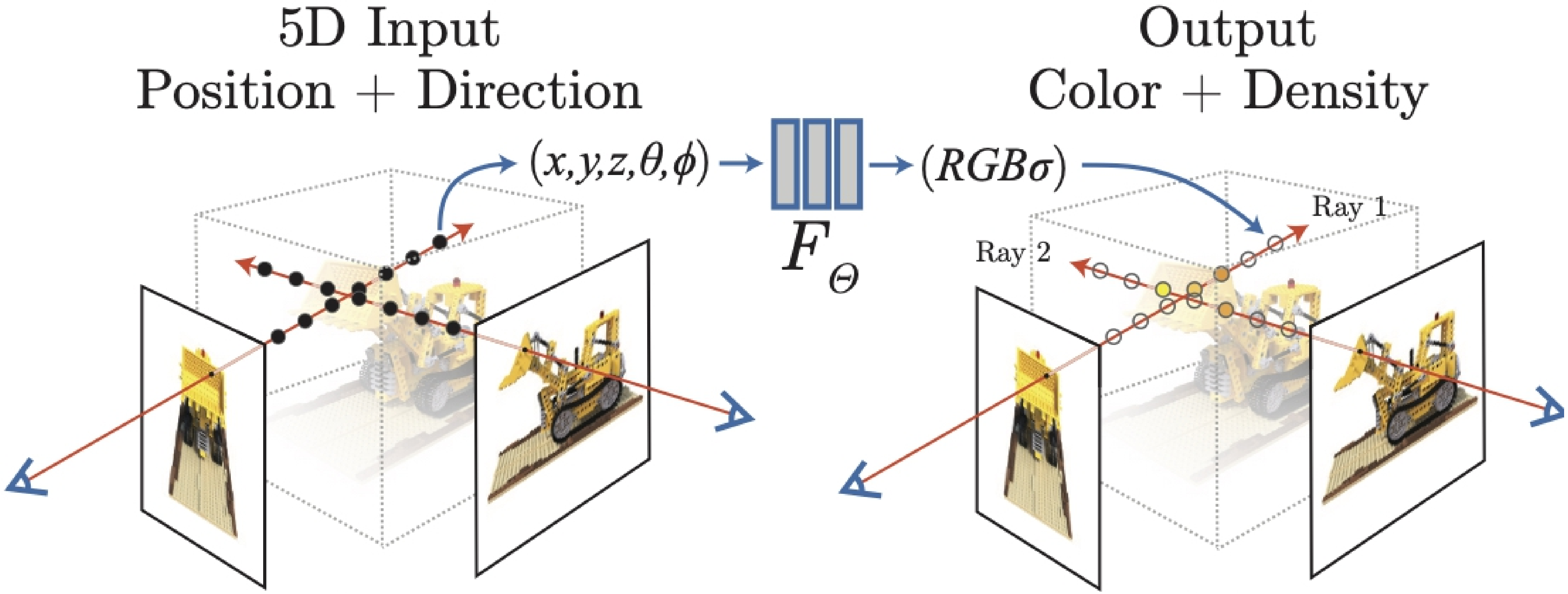
(e) Inverse rendering $(x,y,z) \rightarrow$ RGB, density

Tancik et al. "**Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains**", NeurIPS 2020.

Neural rendering (NeRF)

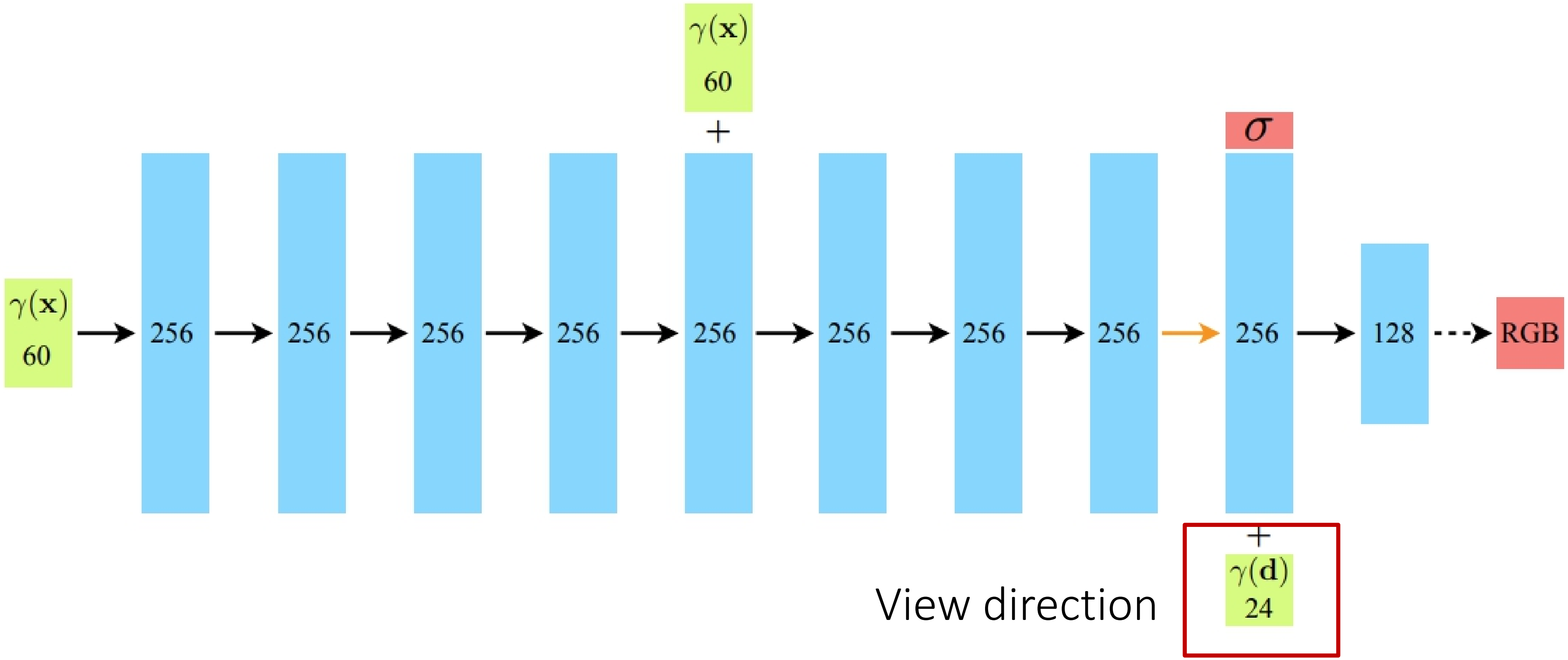
- Computer Graphics + ML

NeRF Overview



- (C, d): $\mathbb{R}^3 \times S^2 \rightarrow \mathbb{R}^4$
- $x, y, z \times \theta, \phi \mapsto r, g, b$, d

NeRF NN Architecture



Positional Encoding

Positional encoding:

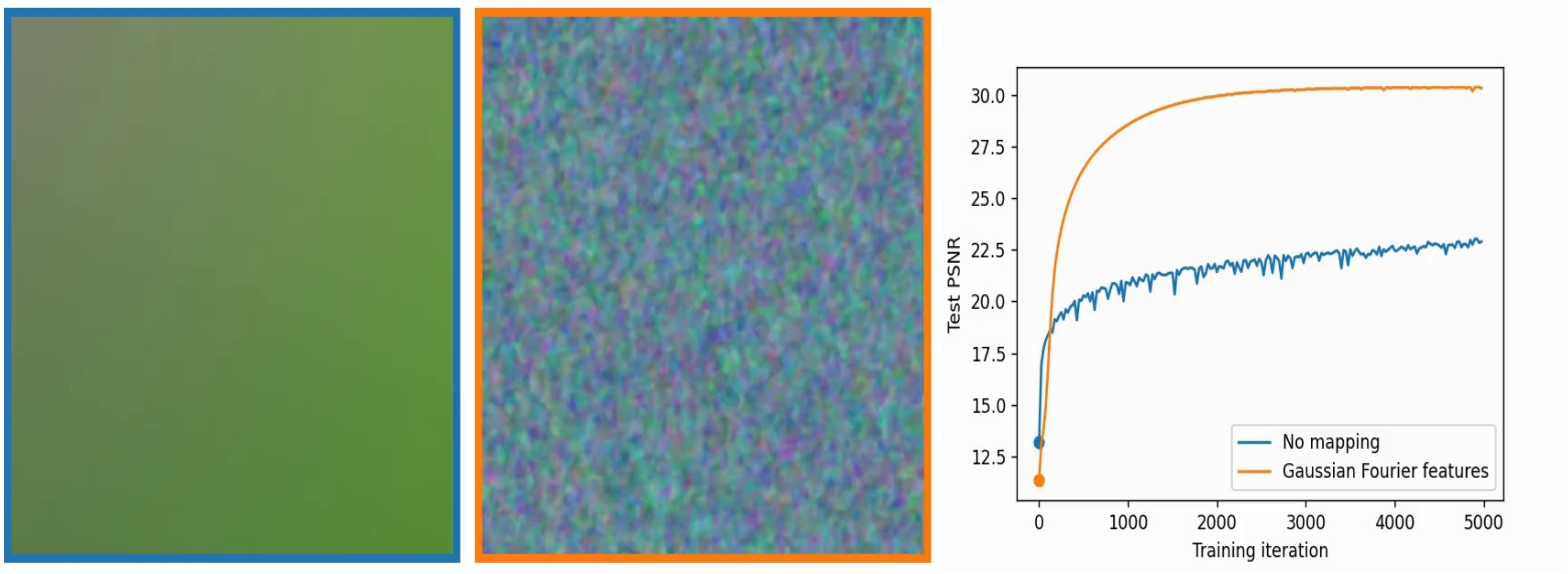
$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right)$$

GT | w/ Pos. Encoding | w/o Pos. Encoding

*Tancik et al. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains", NeurIPS'20

Positional Encoding

Positional encoding:



*Tancik et al. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains", NeurIPS 2020