Image Processing in Python

Mahdi Cherakhloo

Local Processing

- Local processing is very effective approach in image processing (Natural images are Non-stationary)
  - Local histogram equalization
  - Local and adaptive intensity transform
  - Local statistics (mean and variance)

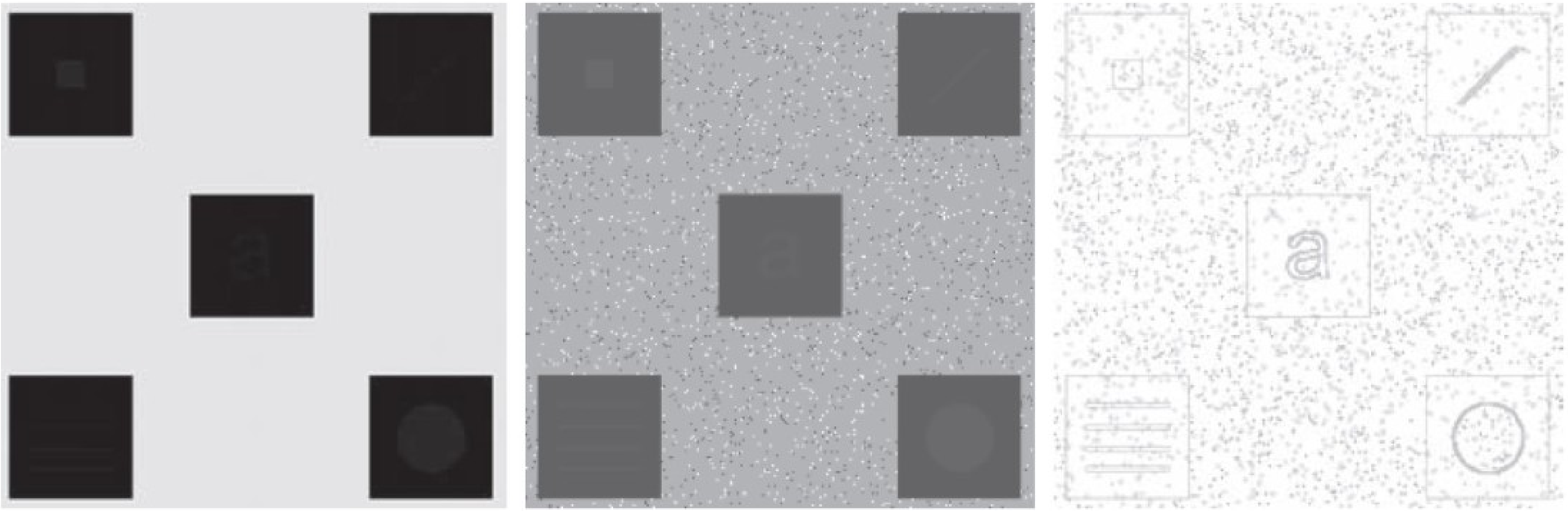


Image Enhancement

Spatial Domain Process

- For $n \times n$ window,
  - –Filtering/Mask/Kernel/Window/Template Processing

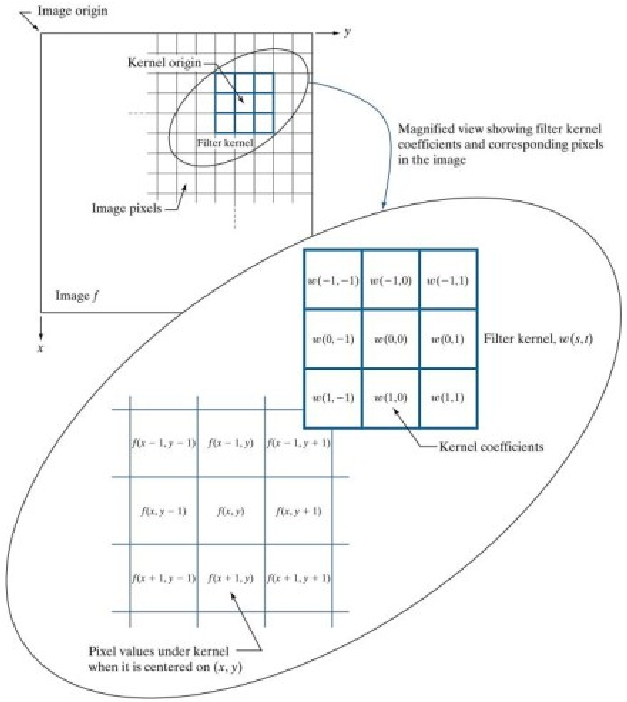


Image Enhancement

Spatial Domain Process

- Smoothing Linear Filtering (Correlation and/or Convolution)

- $g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t)$
- $g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)$

Spatial Domain Process

- Correlation/Convolution, valid (center) and same (right)

Initial position for w

1	2	3	0	0	0	0
4	5	6	0	0	0	0
7	8	9	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

Full correlation result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Rotated w

9	8	7	0	0	0	0
6	5	4	0	0	0	0
3	2	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Full convolution result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Image Enhancement

Spatial Domain Process

- Blurring Effect:
- Boxcar windows:

1×1	3×3
11×11	21×21

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

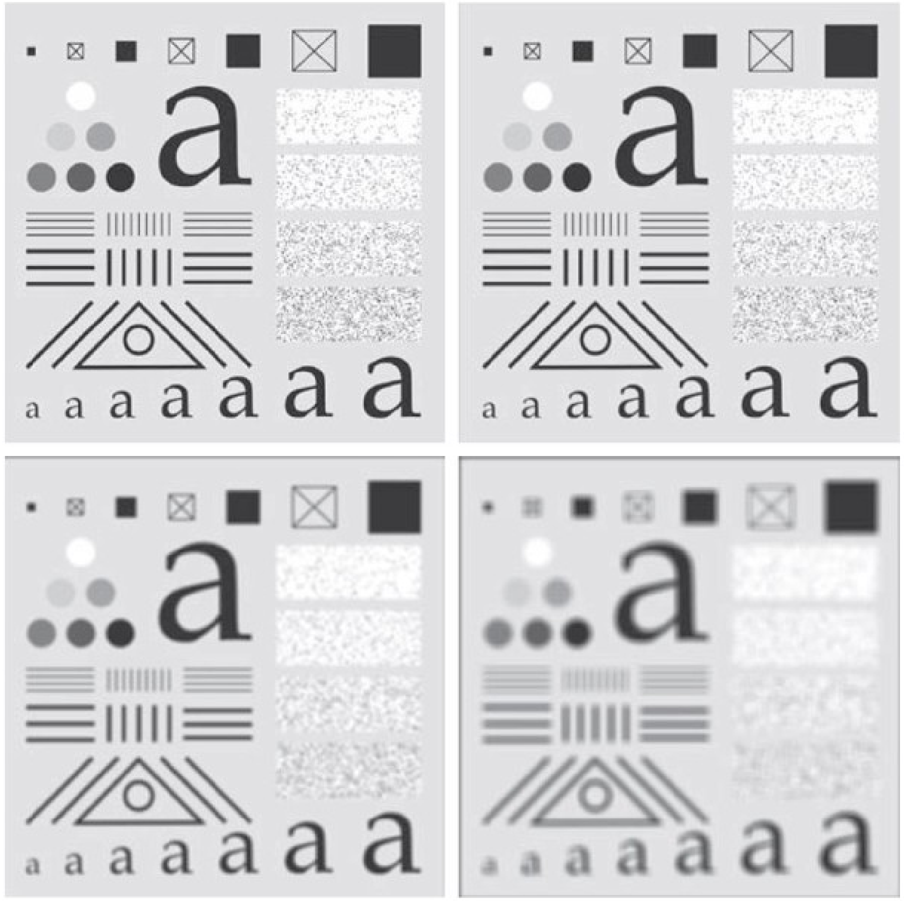


Image Enhancement

Spatial Domain Process

› Most Common Spatial Filter:

–Gaussian Kernel:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \xrightarrow{kernel\ limited\ size} G_\sigma(x,y) = Ke^{-\frac{x^2+y^2}{2\sigma^2}}$$

-Kernel/Windows size: $\approx\ ([6\sigma] \times [6\sigma])$

–K: Normalization factor $(\sum_x \sum_y G_\sigma(x,y) = 1)$

–Less blurring

Spatial Domain Process

- Order statistics filter:
  - Median (Best simple choice for salt & pepper noise)
  - $g(x,y) = \sum_{(s,t)\in S(x,y)} median\{f(s,t)\}$

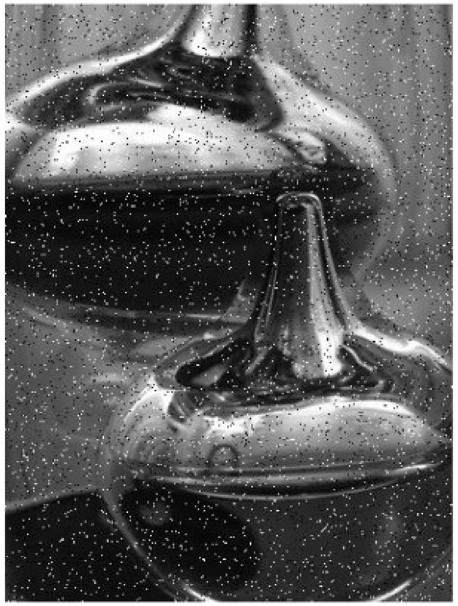


Image Enhancement

Spatial Domain Process

- Image Sharpening:
  - Highlight edges using *first* or *second* derivative:
- Laplacian of image

$$\pm\left(\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial Y^2}\right)=\pm\nabla^2 f$$

- Discrete Implementation with + sign (left) and – sign (right):

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Image Enhancement

Spatial Domain Process

- Image enhancement:

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

- Kernel formulation:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 00 \end{bmatrix} + c \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Image Enhancement

Spatial Domain Process

› Example:

a	b
c	d

(a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 3.45(a). (c) Image sharpened using Eq. (3-54) with $c = -1$. (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b).

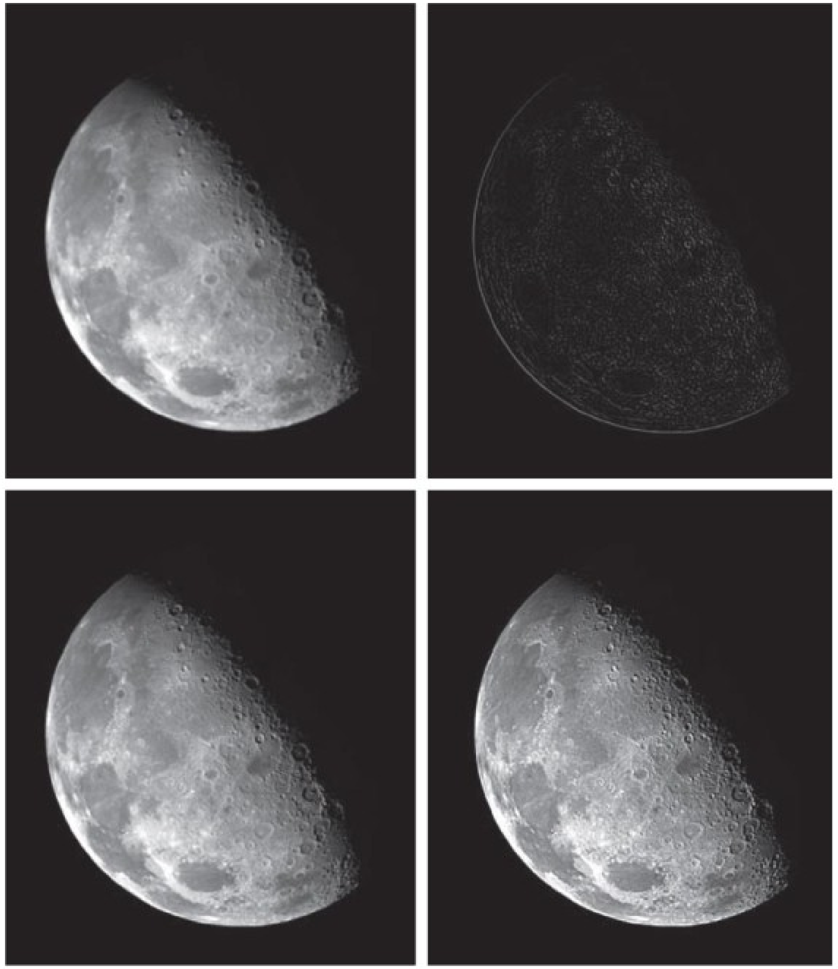


Image Enhancement

Spatial Domain Process

- Noise suppression in Laplacian processing:

$$-\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial Y^2}\right) = -\nabla^2 f$$

- Laplacian of Gaussian (LoG) of image:

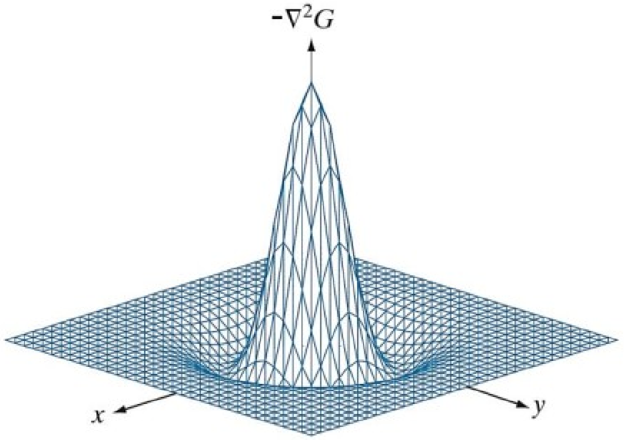
$$LoG(f) = -\nabla^2(G_\sigma * f) = (-\nabla^2 G_\sigma) * f$$

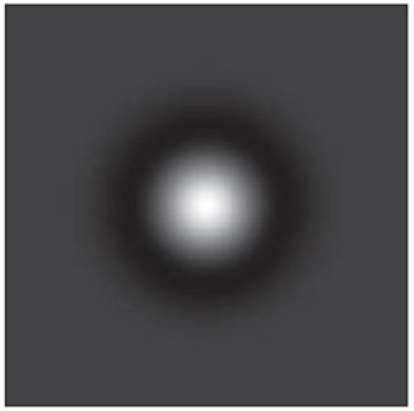
$$\nabla^2 G_\sigma = \left(\frac{2\sigma^2 - x^2 - y^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

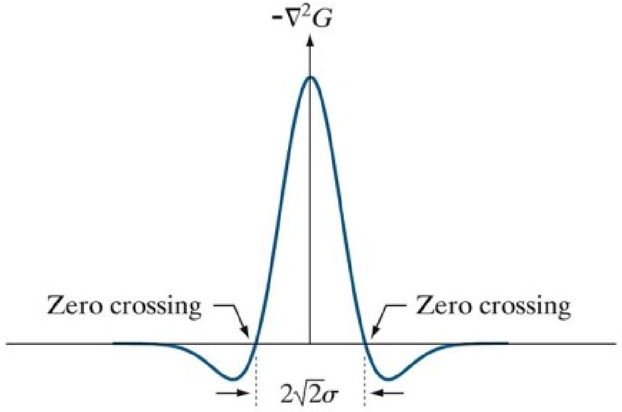
Image Enhancement

Spatial Domain Process

› *LoG Kernel*







0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Spatial Domain Process

- *LoG* approximation via *DoG* (Difference of Gaussian)

$$G_D(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

Image Enhancement

Spatial Domain Process

› Image Sharpening using image gradient:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x}, & \frac{\partial f}{\partial y} \end{bmatrix}^T \Rightarrow M(x, y) = \|\nabla f\|, \widehat{M}(x, y,) = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$

› Discrete implementation of g_x and g_y (*Sobel* Mask):

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Image Enhancement

Spatial Domain Process

› Image Sharpening using image gradient, $M(x, y)$:

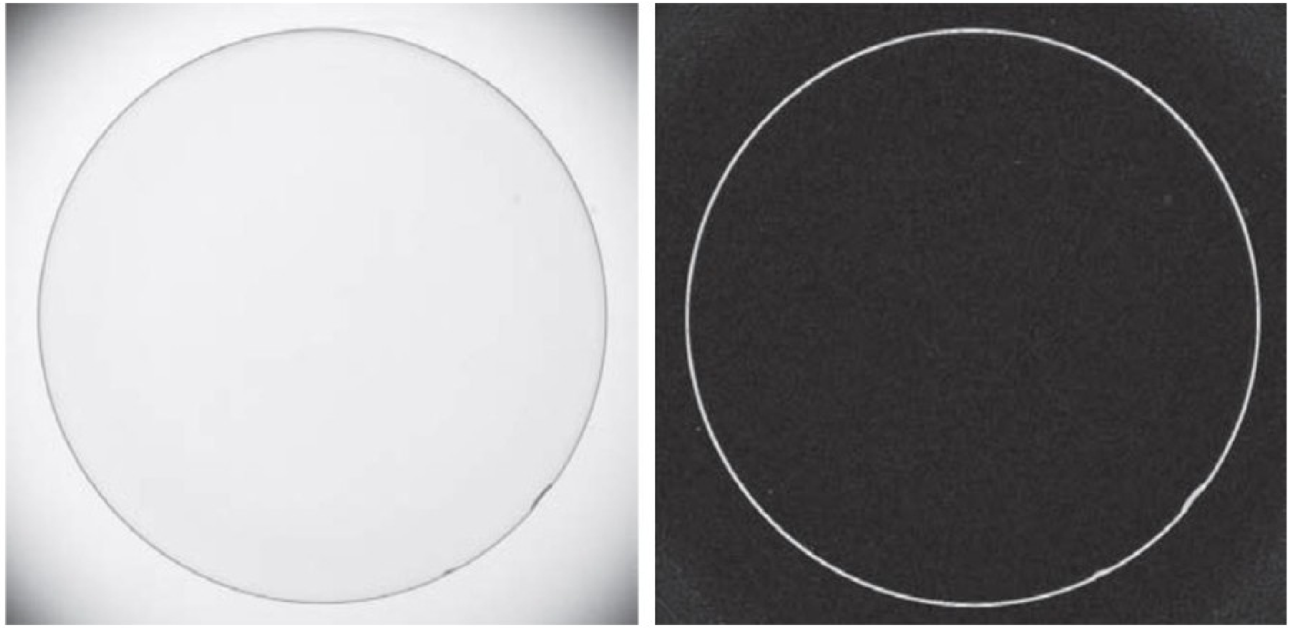


Image Enhancement

Two Dimensional Systems:

› General Definition:

$$g(x, y) = \mathcal{H}\{f(x, y)\}$$

Image Enhancement

System Properties:

- Linearity:

$$\mathcal{H}\{af_1(x,y) + bf_2(x,y)\} = a\mathcal{H}\{f_1(x,y)\} + b\mathcal{H}\{f_2(x,y)\}$$

- Spatial Invariant:

$$\mathcal{H}\{f(x-x_0, y-y_0)\} = g(x-x_0, y-y_0)$$

- Causality: We do not care about it!
- Stability: Same as before.

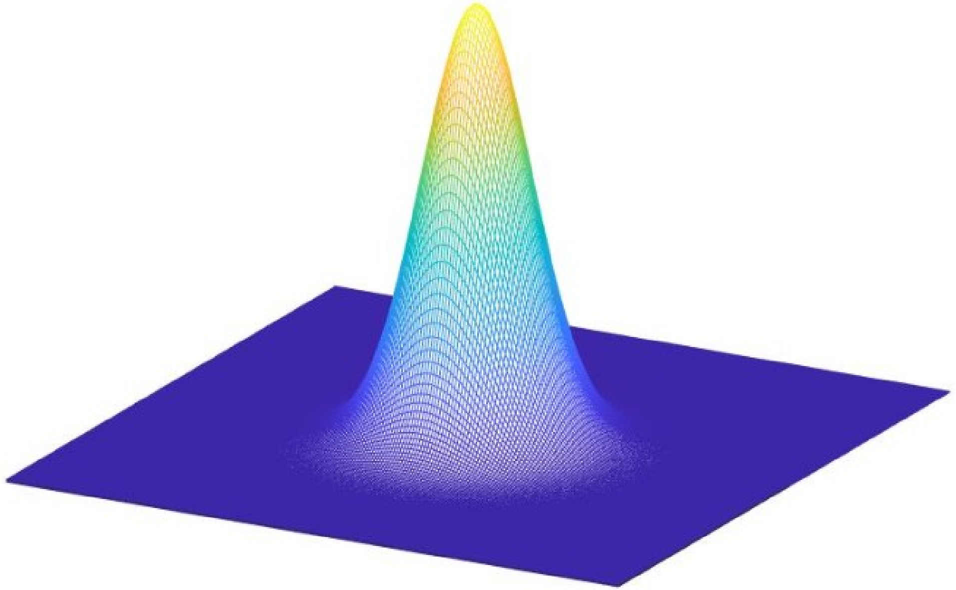
Unit Impulse (pinhole)

› Mathematical Definition:

$$\delta(x,y) = \begin{cases} 0, & (x,y) \neq (0,0) \\ \infty, & (x,y) = (0,0) \end{cases}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x,y)\,dx\,dy = 1$$

› Approximation:



Point Spread Function (PSF)

- Definition:

$$H(x,y;x_0,y_0) = \mathcal{H}\{\delta(x-x_0,y-y_0)\}$$

- Linear Shift Invariant (LSI):

$$H(x,y;x_0,y_0) = \mathcal{H}\{\delta(x-x_0,y-y_0)\} = H(x-x_0,y-y_0)$$

$$H(x,y) = \mathcal{H}\{\delta(x,y)\}$$

Convolution and Correlation

› Discrete Convolution:

$$(f \star h)(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m, y-n)$$

› Discrete Correlation:

$$(f \star h)(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f\ (m,n)h(x+m, y+n)$$

Discrete Fourier Transform (DFT):

› Forward Transform

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

› Inverse Transform:

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

DFT (Definitions)

› Useful definitions:

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x,y)$	$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u,v)$	$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
3) Spectrum	$\lvert F(u,v)\rvert=\left[R^2(u,v)+I^2(u,v)\right]^{1/2}$ $\quad R=\text{Real}(F);\ I=\text{Imag}(F)$
4) Phase angle	$\phi(u,v)=\tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
5) Polar representation	$F(u,v)=\lvert F(u,v)\rvert e^{j\phi(u,v)}$
6) Power spectrum	$P(u,v)=\lvert F(u,v)\rvert^2$
7) Average value	$\bar{f}=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)=\frac{1}{MN}F(0,0)$
8) Periodicity (k_1 and k_2 are integers)	$F(u,v)=F(u+k_1M,v)=F(u,v+k_2N)$ $=F(u+k_1,v+k_2N)$ $f(x,y)=f(x+k_1M,y)=f(x,y+k_2N)$ $=f(x+k_1M,y+k_2N)$

DFT Pairs

› Useful Pairs:

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x,y) + bf_2(x,y) \Leftrightarrow aF_1(u,v) + bF_2(u,v)$
3) Translation (general)	$f(x,y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M + vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$
5) Rotation	$f(r, \theta+\theta_0) \Leftrightarrow F(\omega, \varphi+\theta_0)$ $r = \sqrt{x^2+y^2} \quad \theta = \tan^{-1}(y/x) \quad \omega = \sqrt{u^2+v^2} \quad \varphi = \tan^{-1}(v/u)$
6) Convolution theorem†	$f \star h)(x,y) \Leftrightarrow (F \bullet H)(u,v)$ $(f \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \star H)(u,v)]$
7) Correlation theorem†	$(f \,☆\, h)(x,y) \Leftrightarrow (F^* \bullet H)(u,v)$ $(f^* \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \,☆\, H)(u,v)]$

DFT Pairs

› Useful Pairs:

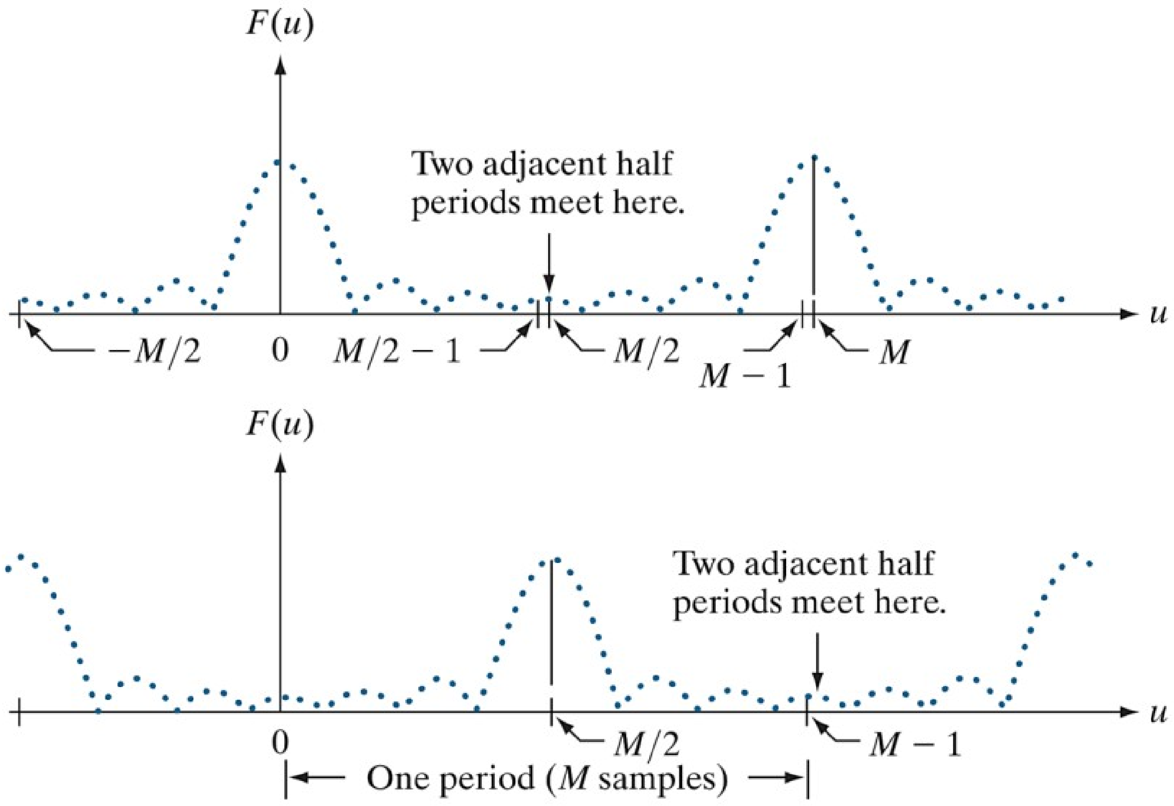
8)	Discrete unit impulse	$\delta(x,y) \Leftrightarrow 1$ $1 \Leftrightarrow MN\delta(u,v)$
9)	Rectangle	$\text{rec}[a,b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
10)	Sine	$\sin(2\pi u_0 x/M + 2\pi v_0 y/N) \Leftrightarrow \dfrac{jMN}{2}\left[\delta(u+u_0, v+v_0) - \delta(u-u_0, v-v_0)\right]$
11)	Cosine	$\cos(2\pi u_0 x/M + 2\pi v_0 y/N) \Leftrightarrow \dfrac{1}{2}\left[\delta(u+u_0, v+v_0) + \delta(u-u_0, v-v_0)\right]$

The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.

12)	Differentiation (the expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.	$\left(\dfrac{\partial}{\partial t}\right)^m \left(\dfrac{\partial}{\partial z}\right)^n f(t,z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu,\nu)$ $\dfrac{\partial^m f(t,z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu,\nu); \quad \dfrac{\partial^n f(t,z)}{\partial z^m} \Leftrightarrow (j2\pi\nu)^n F(\mu,\nu)$
13)	Gaussian	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

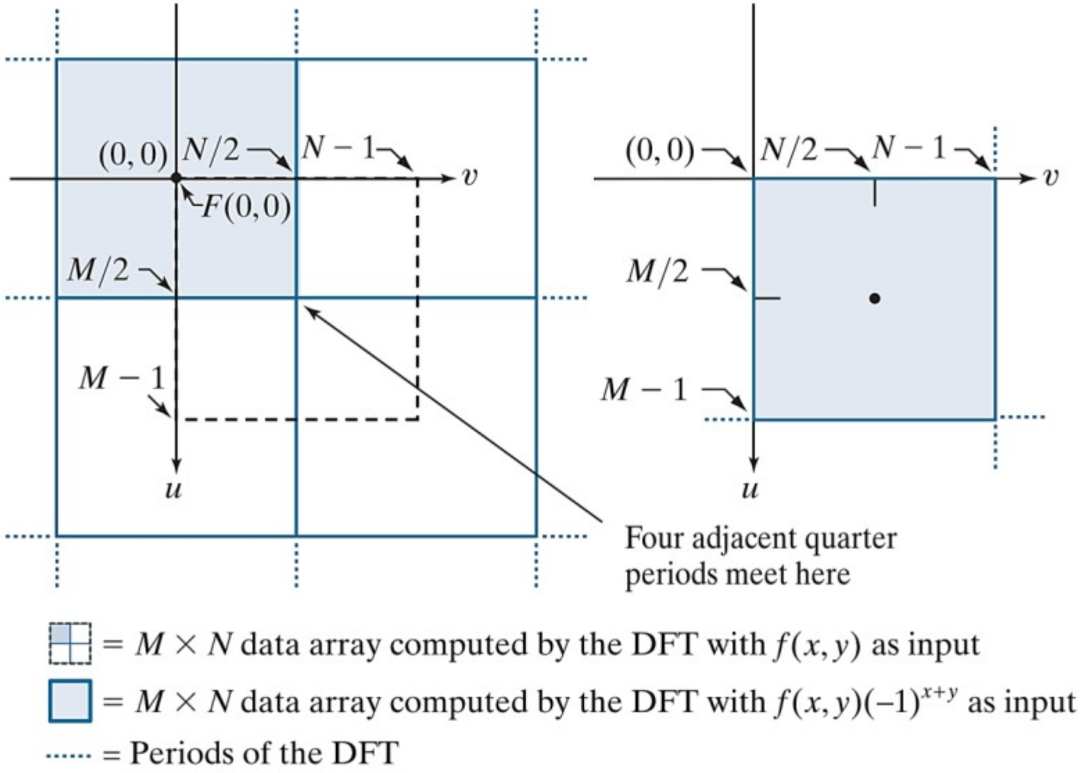
DFT Centering

› From DSP:



DFT Centering

› For DIP (*fftshift*):



DFT Centering

› Example:

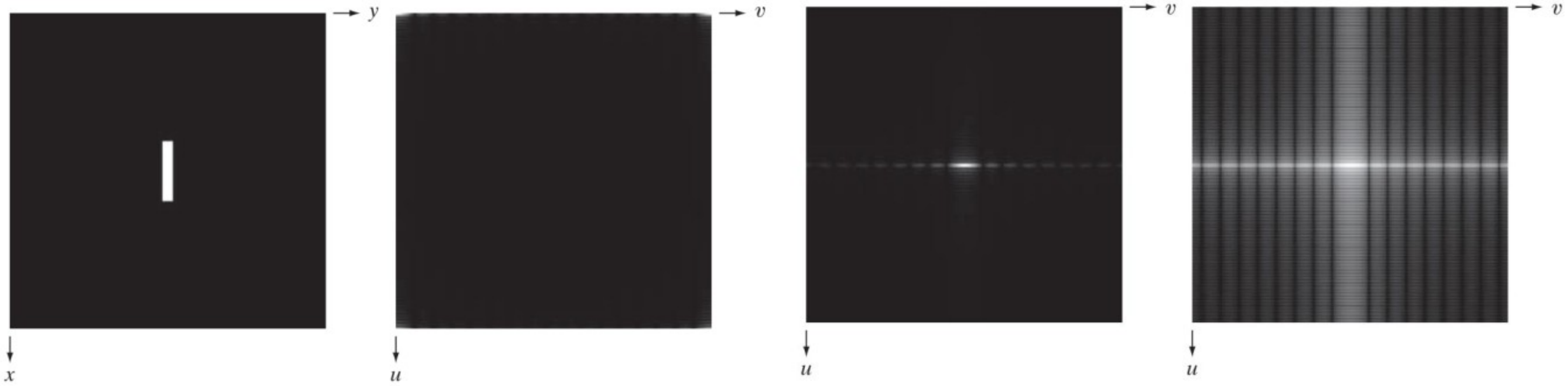
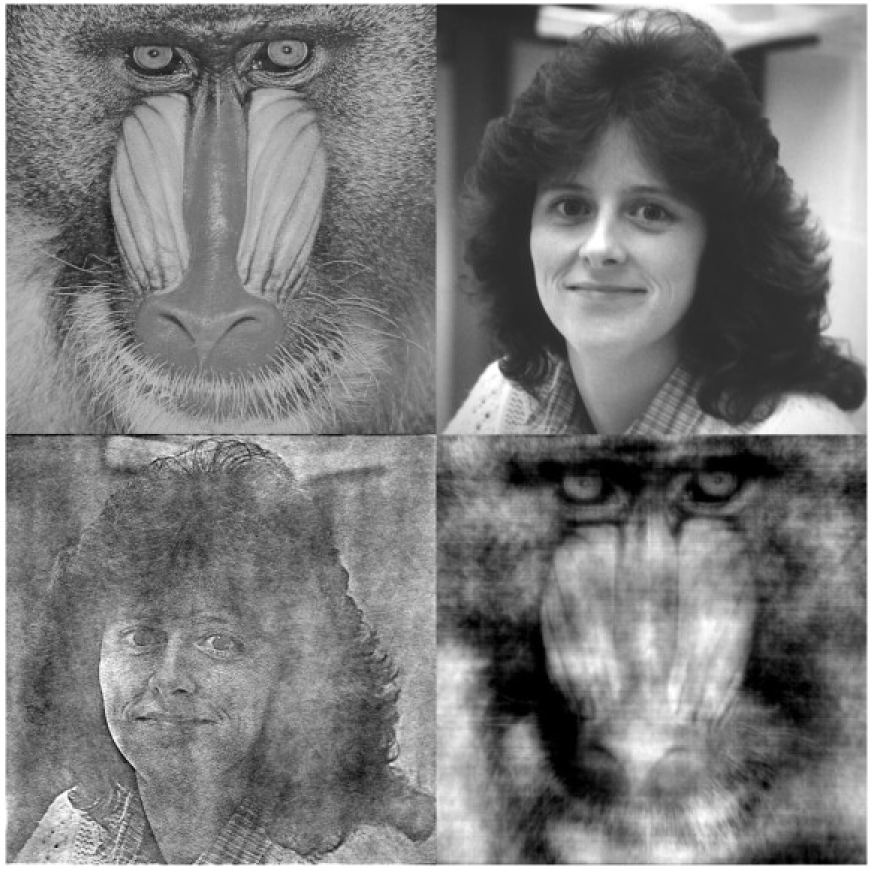


Image	DFT (abs)	Centered DFT	Centered DFT Log

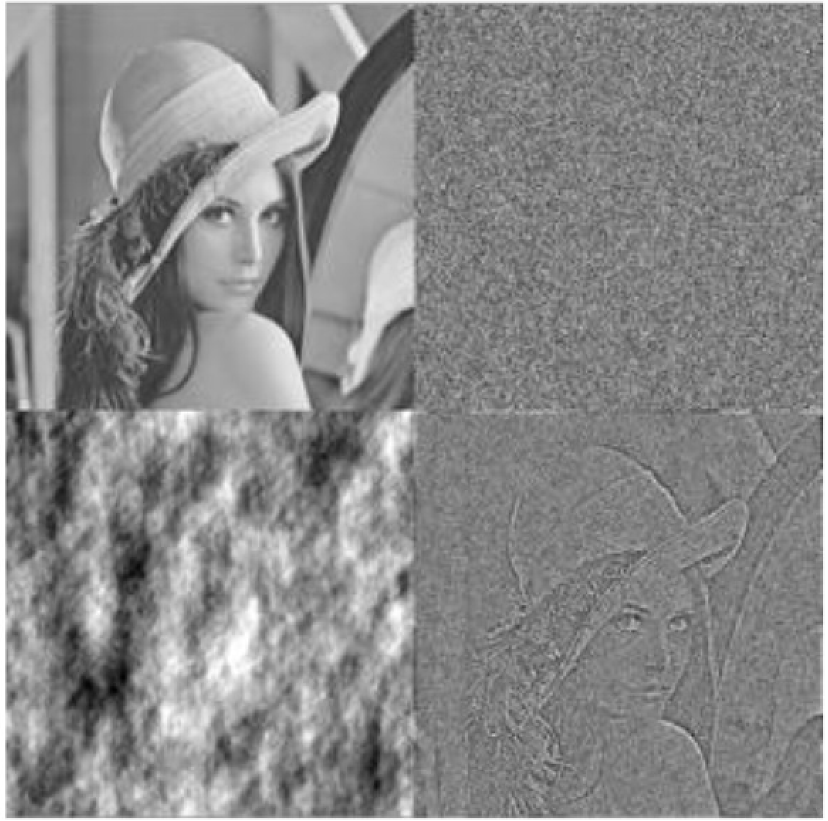
The Importance of Phase

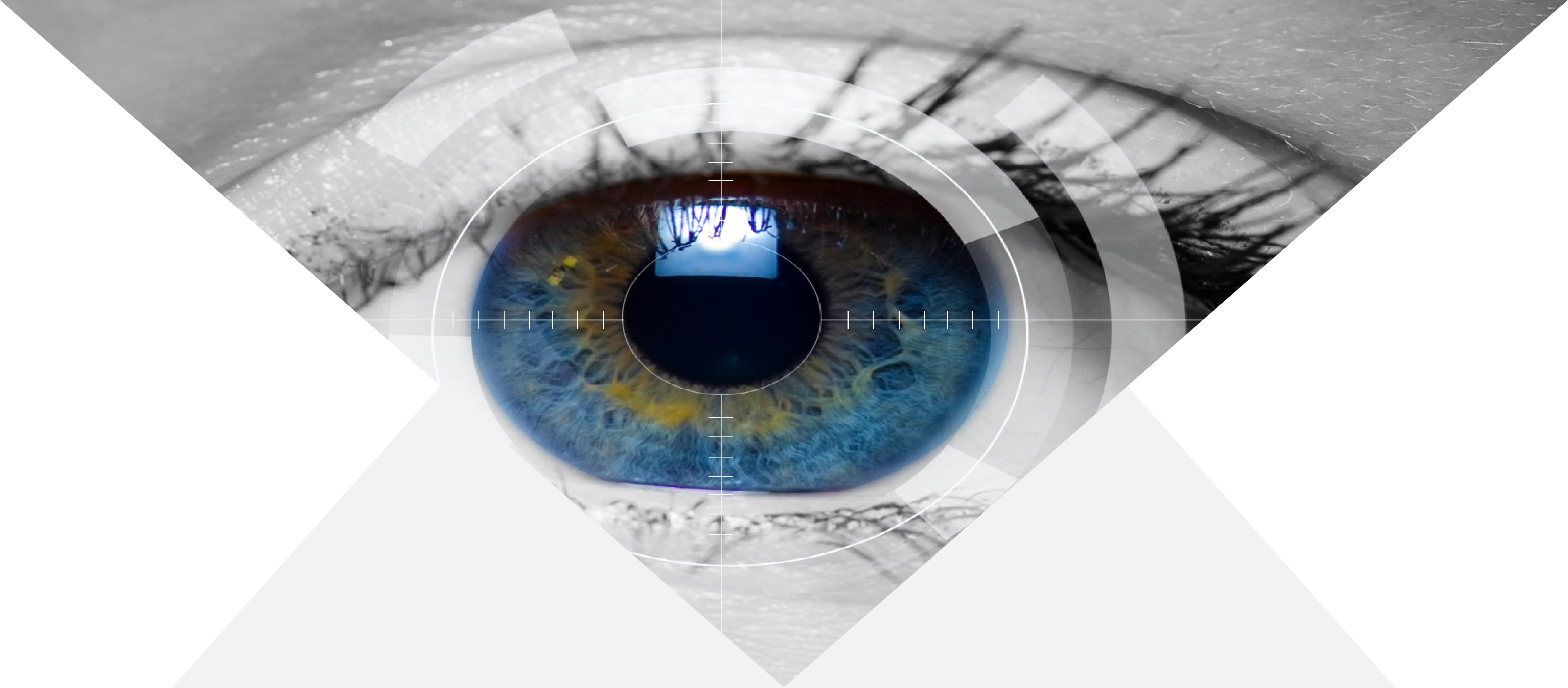
- Let swap phase and magnitude of DFT of two images:



The Importance of Phase

- Let swap phase and magnitude of DFT of two images:





Thank you!