Two-Dimensional Discrete Fourier Transform (2D-DFT)

Definitions

- Spatial Domain (I)
  - “Normal” image space
  - Changes in pixel positions correspond to changes in the scene
  - Distances in I correspond to real distances
- Frequency Domain (F)
  - Changes in image position correspond to changes in the spatial frequency
  - This is the rate at which image intensity values are changing in the spatial domain image I

Image Processing

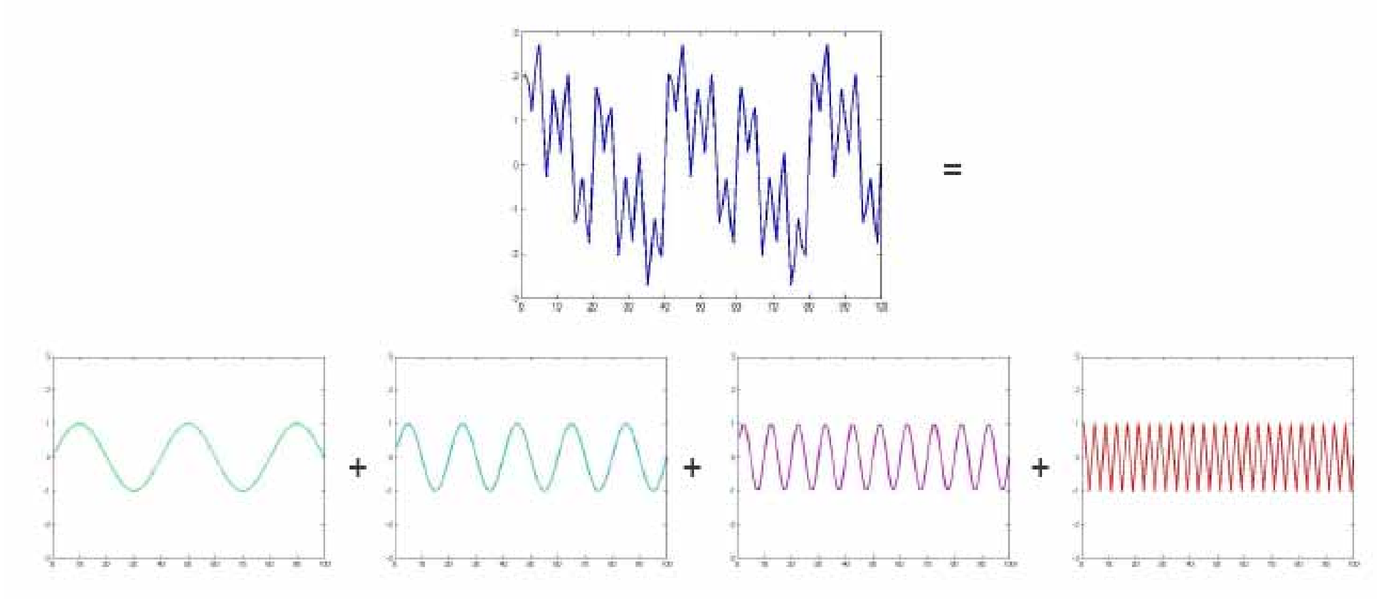
- Spatial Domain (I)
  - Directly process the input image pixel array
- Frequency Domain (F)
  - Transform the image to its frequency representation
  - Perform image processing
  - Compute inverse transform back to the spatial domain

Frequencies in an Image

- Any spatial or temporal signal has an equivalent frequency representation
- What do frequencies mean in an image
  - High frequencies correspond to pixel values that change rapidly across the image (e.g. text, texture, leaves, etc.)
  - Strong low frequency components correspond to large scale features in the image (e.g. a single, homogenous object that dominates the image)
- We will investigate Fourier transformations to obtain frequency representations of an image

The Fourier Series

- Periodic functions can be expressed as the sum of sines and/or cosines of different frequencies each multiplied by a different coefficient



Property of DFT

- Linearity:

  Time/spatial domain: $\mathbf{h} = \alpha\mathbf{f} + \beta\mathbf{g}$

  Frequency domain: $\mathbf{H} = \alpha\mathbf{F} + \beta\mathbf{G}$

- Shifting:

$$\mathrm{F}_{u \pm \frac{N}{2}} = \Im[(\text{-}1)^{x} f_{x}]$$

E.g. ***f*** = *[a b c d e f g h]* then ***F*** = *[1 2 3 4 5 6 7 8]*

f'= *[a -b c -d e -f g -h]* then ***F'*** = *[5 6 7 8 1 2 3 4]*

Property of DFT

- Conjugate symmetry:
  - For **f** is real and of length *N*, then **F** satisfies the following condition.

$$F_k = \overline{F_{N-k}}$$

  - Where $\overline{\mathrm{F}_{N-k}}$ is the complex conjugate of F_{N-k} for all *k=1,2,3..N-1*
  - Example of length 8, we have
  - This is case $F_4 = \overline{F_4}$,which F_4 must be real.

$$F_1 = \overline{F_7},\ F_2 = \overline{F_6},\ F_3 = \overline{F_5}$$

Property of DFT

- Convolution
  - Let **z** be the circular convolution of **f** and **g**.

Spatial domain: $$\mathbf{z} = \mathbf{f} * \mathbf{g}$$

$$z_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n g_{k-n}$$

Frequency domain: $$Z_u = F_u \cdot G_u$$

The Discrete Fourier Transform

- Since we are dealing with images, we will be more interested in the discrete Fourier Transform (DFT)
- For a function f(x), x=0,1,…,M-1 we have

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}, \quad u = 0,\dots,M-1$$

$$f(x) = \sum_{x=0}^{M-1} F(u)e^{j2\pi ux/M}, \quad x = 0,\dots,M-1$$

The Discrete Fourier Transform

- Recall Euler's Formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

from which we obtain

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\left(\cos\frac{2\pi ux}{M} - j\sin\frac{2\pi ux}{M}\right)$$

for u = 0,…,M-1

- Each term is composed of ALL values of f(x)
- The values of u are the frequency domain
- Each F(u) is a frequency component of the transform

The 2D Discrete Fourier Transform

- Since our images are nothing more than 2D discrete functions, we are interested in the 2D DFT

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

for u=0,…,M-1 and v=0,…,N-1 and the iDFT is defined as

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

2D DFT and Inverse DFT (IDFT)

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

$f(x, y)$ ⇄ $F(u, v)$

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

M, *N*: image size

x, *y*: image pixel position

u, *v*: spatial frequency

often used short notation:

The Meaning of DFT and Spatial Frequencies

- **Important Concept**

Any signal can be represented as a **linear combination** of a set of **basic components**

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

- **Fourier components**: sinusoidal patterns
- **Fourier coefficients**: weighting factors assigned to the Fourier components

- **Spatial frequency:** The frequency of Fourier component
- **Not to confused with electromagnetic frequencies** (e.g., the frequencies associated with light colors)

Spatial Domain

- In the spatial domain, a single point corresponds to the integration of all contributing frequencies at that position

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

- Position is known well, but contributed frequency is “unlimited”

Frequency Domain

- In the frequency domain, a single point corresponds to the strength of a single frequency

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

- Each point is influenced by the intensity of ALL points in space
- Frequency is known well, but spatial contributions are “unlimited”

DFT (Continued)

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

- At u=v=0, the FDT reduces to

$$F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

- This is nothing more than the average grayscale level of the image
- This is often referred to as the DC Component (0 frequency)

DFT (Continued)

- The FT has the following translation property

$$f(x,y)e^{-j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$$

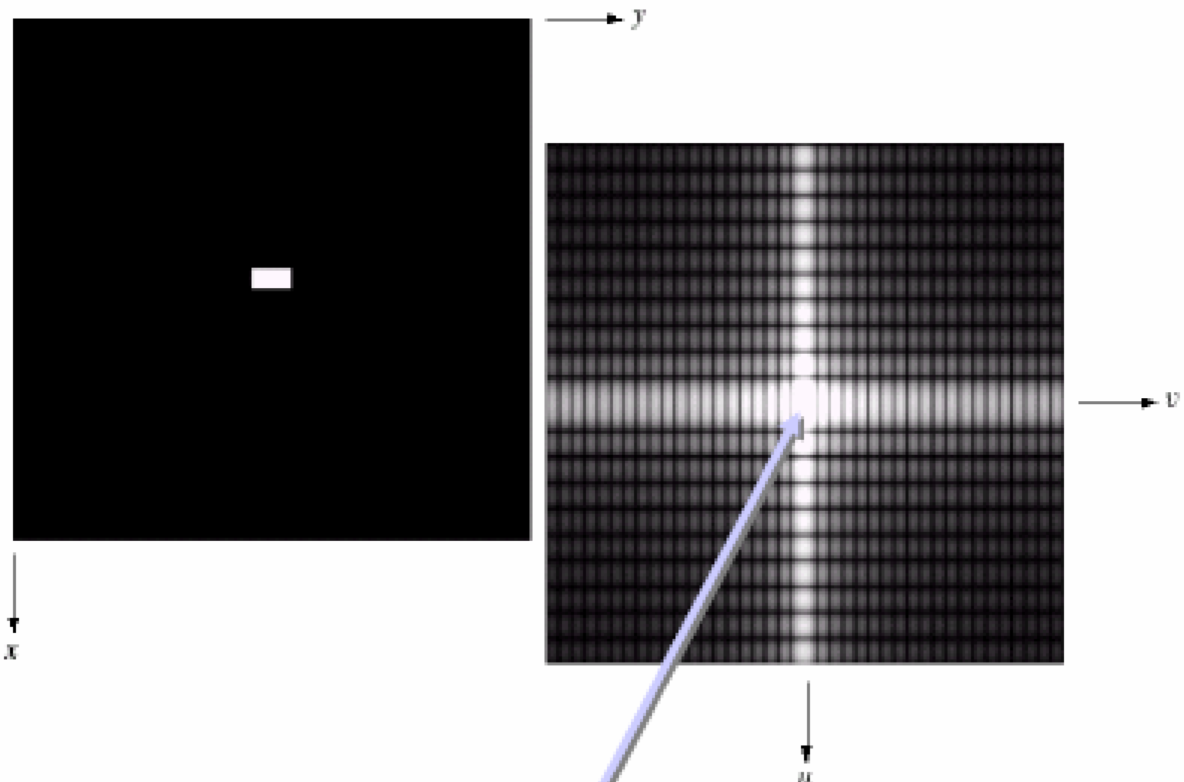
which for u0=M/2 and v0=N/2 we see that

$$e^{-j2\pi(ux/M+vy/N)} = e^{-j\pi(x+y)} = (-1)^{x+y}$$

$$\Rightarrow F(f(x,y)(-1)^{x+y}) = F(u-M/2, v-N/2)$$

- In image processing, it is common to multiply the input image by (-1)x+y prior to computing F(u,v)
- This has the effect of centering the transform since F(0,0) is now located at u=M/2, v=N/2

Centered Fourier Spectrum



a b

FIGURE 4.3 (a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

Real Part, Imaginary Part, Magnitude, Phase, Spectrum

Real part: $R = \text{Real}(F)$

Imaginary part: $I = \text{Imag}(F)$

Magnitude-phase representation:

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

Magnitude (spectrum):

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

Phase (spectrum):

$$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)}\right]$$

Power Spectrum:

$$P(u, v) = |F(u, v)|^2$$

2D DFT Properties

Mean of image / DC component:

$$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Highest frequency component:

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$$

“Half-shifted” Image:

$$f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$$

Conjugate Symmetry:

$$F(u, v) = F^*(-u, -v)$$

Magnitude Symmetry:

$$|F(u, v)| = |F(-u, -v)|$$

2D DFT Properties

Spatial domain differentiation: $$\frac{\partial^n f(x,y)}{\partial x^n} \Leftrightarrow (ju)^n F(u,v)$$

Frequency domain differentiation: $$(-jx)^n f(x,y) \Leftrightarrow \frac{\partial^n F(u,v)}{\partial u^n}$$

Distribution law: $$\Im[f_1(x,y) + f_2(x,y)] = \Im[f_1(x,y)] + \Im[f_2(x,y)]$$

Laplacian: $$\nabla^2 f(x,y) \Leftrightarrow -(u^2 + v^2)F(u,v)$$

Spatial domain Periodicity: $$f(x,y) = f(x+M,y) = f(x,y+N) = f(x+M,y+N)$$

Frequency domain periodicity: $$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N)$$

Properties of Two-Dimension DFT

- Linearity (same as 1D DFT)
- DC coefficient: sum of all term in the matrix (same as 1D)
- Conjugate symmetry:

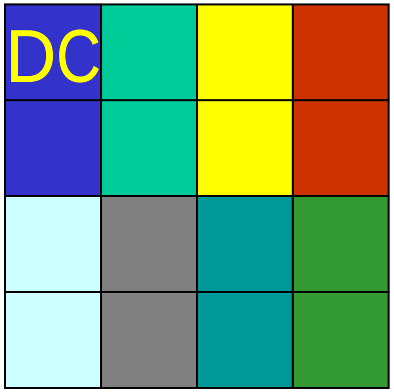
$$F_{u,v} = \overline{F}_{-u+pM,-v+qN};\ p,q \in I$$

Properties of Two-Dimension DFT

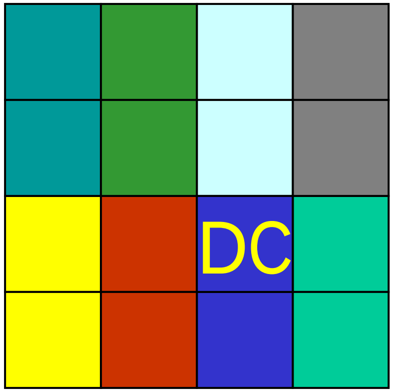
- Shifting (similar to 1D DFT)

Fourier Transform

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p



a	-b	c	-d
-e	f	-g	h
i	-j	k	-l
-m	n	-o	p



Computation of 2D-DFT

Fourier transform matrices: remember

$$\mathbf{F}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \qquad \mathbf{F}_N^* = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{1-N} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{1-N} & W_N^{2(1-N)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}$$

relationship: $\mathbf{F}_N^{-1} = \frac{1}{N}\mathbf{F}_N^*$

In particular, for $N = 4$:

$$\mathbf{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \qquad \mathbf{F}_4^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

Computation of 2D-DFT

- To compute the 1D-DFT of a 1D signal **x** (as a vector):

$$\tilde{\mathbf{x}} = \mathbf{F}_N \mathbf{x}$$

To compute the inverse 1D-DFT:

$$\mathbf{x} = \frac{1}{N} \mathbf{F}_N^* \tilde{\mathbf{x}}$$

- To compute the 2D-DFT of an image **X** (as a matrix):

$$\tilde{\mathbf{X}} = \mathbf{F}_N \mathbf{X} \mathbf{F}_N$$

To compute the inverse 2D-DFT:

$$\mathbf{X} = \frac{1}{N^2} \mathbf{F}_N^* \tilde{\mathbf{X}} \mathbf{F}_N^*$$

Computation of 2D-DFT: Example

- A 4x4 image

$$\mathbf{X} = \begin{bmatrix} 1 & 3 & 6 & 8 \\ 9 & 8 & 8 & 2 \\ 5 & 4 & 2 & 3 \\ 6 & 6 & 3 & 3 \end{bmatrix}$$

- Compute its 2D-DFT:

$$\tilde{\mathbf{X}} = \mathbf{F}_4 \mathbf{X} \mathbf{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 & 3 & 6 & 8 \\ 9 & 8 & 8 & 2 \\ 5 & 4 & 2 & 3 \\ 6 & 6 & 3 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

MATLAB function: *fft2*

$$= \begin{bmatrix} 21 & 21 & 19 & 16 \\ -4-3j & -1-2j & 4-5j & 5+j \\ -9 & -7 & -3 & 6 \\ -4+3j & -1+2j & 4+5j & 5-j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

lowest frequency component

highest frequency component

$$= \begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix}$$

建立大小爲 m × n的矩陣

- 在每一橫列結尾加上分號（;），例如：

```
>> X = [1 3 6 8; 9 8 8 2; 5 4 2 3;6 6 3 3]; % 建立 3×4 的矩陣 A
>> X      % 顯示矩陣 A 的內容
X =
    1    3    6    8
    9    8    8    2
    5    4    2    3
    6    6    3    3
```

$$\mathbf{X} = \begin{bmatrix} 1 & 3 & 6 & 8 \\ 9 & 8 & 8 & 2 \\ 5 & 4 & 2 & 3 \\ 6 & 6 & 3 & 3 \end{bmatrix}$$

- A=fft2(X)

$$=\begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix}$$

- A =
-

77.0000	2.0000 - 5.0000i	3.0000	2.0000 + 5.0000i
4.0000 - 9.0000i	-11.0000 + 8.0000i	-4.0000 - 7.0000i	-5.0000 - 4.0000i
-13.0000	-6.0000 +13.0000i	-11.0000	-6.0000 -13.0000i
4.0000 + 9.0000i	-5.0000 + 4.0000i	-4.0000 + 7.0000i	-11.0000 - 8.0000i

Matlab

- 共軛複數 x* = a- bi
- 複數大小r = $\sqrt{a^2 + b^2}$
- 複數向量的夾角 θ = tan-1 (*b*/*a*)
- 複數實部*a* = *r cos* θ ,
- 複數虛部*b* = *r sin* θ ,
- 複數指數表示法 *x*=r e*i*

上述各函數對應MATLAB的複數指令爲

- x*=conj(x)
- r=abs(x),
- θ =angle(x),
- a=real(x),
- b=imag(x),
- x=r*exp(i*angle(x))

Computation of 2D-DFT: Example

$$\tilde{\mathbf{X}} = \begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix}$$

Real part:

$$\tilde{\mathbf{X}}_{real} = \begin{bmatrix} 77 & 2 & 3 & 2 \\ 4 & -11 & -4 & -5 \\ -13 & -6 & -11 & -6 \\ 4 & -5 & -4 & -11 \end{bmatrix}$$

Imaginary part:

$$\tilde{\mathbf{X}}_{imag} = \begin{bmatrix} 0 & -5 & 0 & 5 \\ -9 & 8 & -7 & -4 \\ 0 & 13 & 0 & -13 \\ 9 & 4 & 7 & -8 \end{bmatrix}$$

Magnitude:

$$\tilde{\mathbf{X}}_{magnitude} = \begin{bmatrix} 77 & 5.39 & 3 & 5.39 \\ 9.85 & 13.60 & 8.06 & 6.4 \\ 13 & 14.32 & 11 & 14.32 \\ 9.85 & 6.40 & 8.06 & 13.60 \end{bmatrix}$$

Phase:

$$\tilde{\mathbf{X}}_{phase} = \begin{bmatrix} 0 & -1.19 & 0 & 1.19 \\ -1.15 & 2.51 & -2.09 & -2.47 \\ 3.14 & 2.00 & 3.14 & -2.00 \\ 1.15 & 2.47 & 2.09 & -2.51 \end{bmatrix}$$

Real part:

- real(A)
- ans =

77	2	3	2
4	-11	-4	-5
-13	-6	-11	-6
4	-5	-4	-11

Real part:

$$\tilde{\mathbf{x}}_{real} = \begin{bmatrix} 77 & 2 & 3 & 2 \\ 4 & -11 & -4 & -5 \\ -13 & -6 & -11 & -6 \\ 4 & -5 & -4 & -11 \end{bmatrix}$$

Imaginary part:

- imag(A)
- ans =

0	-5	0	5
-9	8	-7	-4
0	13	0	-13
9	4	7	-8

Imaginary part:

$$\tilde{\mathbf{X}}_{imag} = \begin{bmatrix} 0 & -5 & 0 & 5 \\ -9 & 8 & -7 & -4 \\ 0 & 13 & 0 & -13 \\ 9 & 4 & 7 & -8 \end{bmatrix}$$

Magnitude:

Magnitude:

$$\tilde{\mathbf{X}}_{magnitude} = \begin{bmatrix} 77 & 5.39 & 3 & 5.39 \\ 9.85 & 13.60 & 8.06 & 6.4 \\ 13 & 14.32 & 11 & 14.32 \\ 9.85 & 6.40 & 8.06 & 13.60 \end{bmatrix}$$

- abs(A)

ans =

```
  77.0000    5.3852    3.0000    5.3852
   9.8489   13.6015    8.0623    6.4031
  13.0000   14.3178   11.0000   14.3178
   9.8489    6.4031    8.0623   13.6015
```

Phase:

- angle(A)
- ans =

0	-1.1903	0	1.1903
-1.1526	2.5128	-2.0899	-2.4669
3.1416	2.0032	3.1416	-2.0032
1.1526	2.4669	2.0899	-2.5128

Phase:

$$\tilde{\mathbf{X}}_{phase} = \begin{bmatrix} 0 & -1.19 & 0 & 1.19 \\ -1.15 & 2.51 & -2.09 & -2.47 \\ 3.14 & 2.00 & 3.14 & -2.00 \\ 1.15 & 2.47 & 2.09 & -2.51 \end{bmatrix}$$

- B=[77 2-5*i 3 2+5*i; 4-9*i -11+8*i -4-7*i -5-4*i; -13 -6+13*i -11 -6-13*i; 4+9*i -5+4*i -4+7*i -11-8*i]
- B =

77.0000	2.0000 - 5.0000i	3.0000	2.0000 + 5.0000i
4.0000 - 9.0000i	-11.0000 + 8.0000i	-4.0000 - 7.0000i	-5.0000 - 4.0000i
-13.0000	-6.0000 +13.0000i	-11.0000	-6.0000 -13.0000i
4.0000 + 9.0000i	-5.0000 + 4.0000i	-4.0000 + 7.0000i	-11.0000 - 8.0000i

$$\tilde{\mathbf{X}} = \begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix}$$

Computation of 2D-DFT: Example

- Compute the inverse 2D-DFT:

$$\mathbf{F}_4^* \tilde{\mathbf{X}} \mathbf{F}_4^* = \frac{1}{4^2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 21 & 21 & 19 & 16 \\ -4-3j & -1-2j & 4-5j & 5+j \\ -9 & -7 & -3 & 6 \\ -4+3j & -1+2j & 4+5j & 5-j \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 3 & 6 & 8 \\ 9 & 8 & 8 & 2 \\ 5 & 4 & 2 & 3 \\ 6 & 6 & 3 & 3 \end{bmatrix} = \mathbf{X}$$

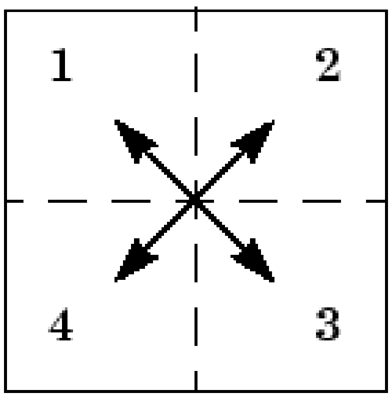
MATLAB function: *ifft2*

- ifft2(B)
- ans =

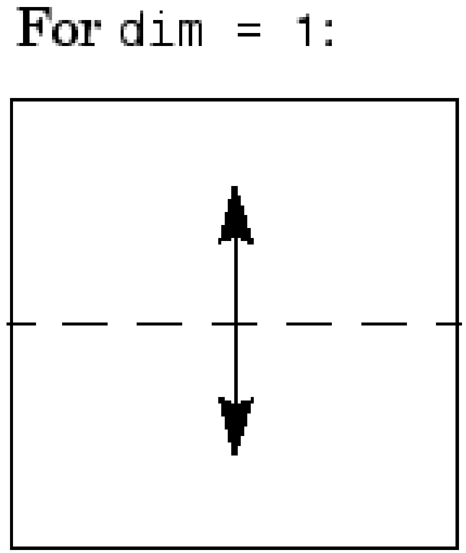
1	3	6	8
9	8	8	2
5	4	2	3
6	6	3	3

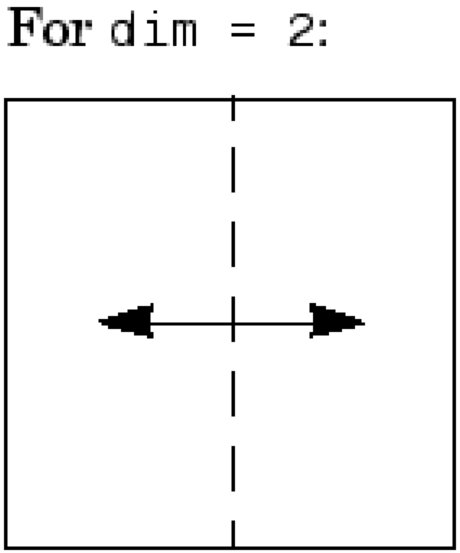
$$= \begin{bmatrix} 1 & 3 & 6 & 8 \\ 9 & 8 & 8 & 2 \\ 5 & 4 & 2 & 3 \\ 6 & 6 & 3 & 3 \end{bmatrix} = \mathbf{X}$$

- Y = fftshift(X) rearranges the outputs of fft, fft2, and fftn by moving the zero-frequency component to the center of the array.
- It is useful for visualizing a Fourier transform with the zero-frequency component in the middle of the spectrum.
- For vectors, fftshift(X) swaps the left and right halves of X.
- For matrices, fftshift(X) swaps the first quadrant with the third and the second quadrant with the fourth.



- For higher-dimensional arrays, fftshift(X) swaps "half-spaces" of X along each dimension.
- Y = fftshift(X,dim) applies the fftshift operation along the dimension dim.

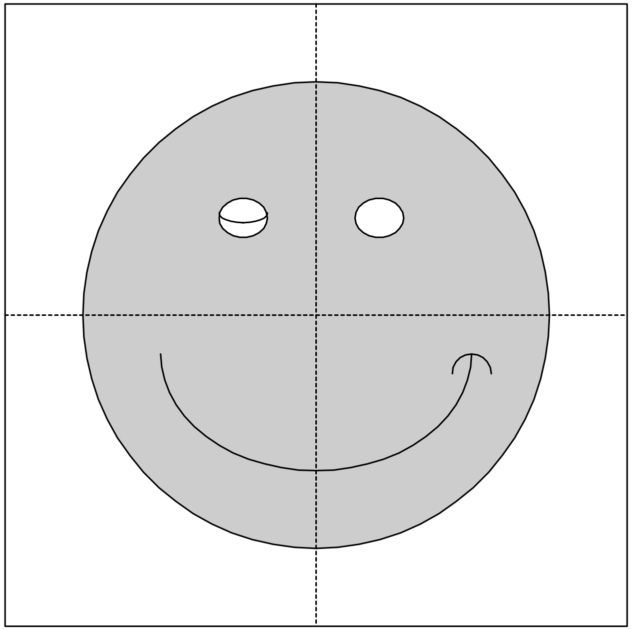


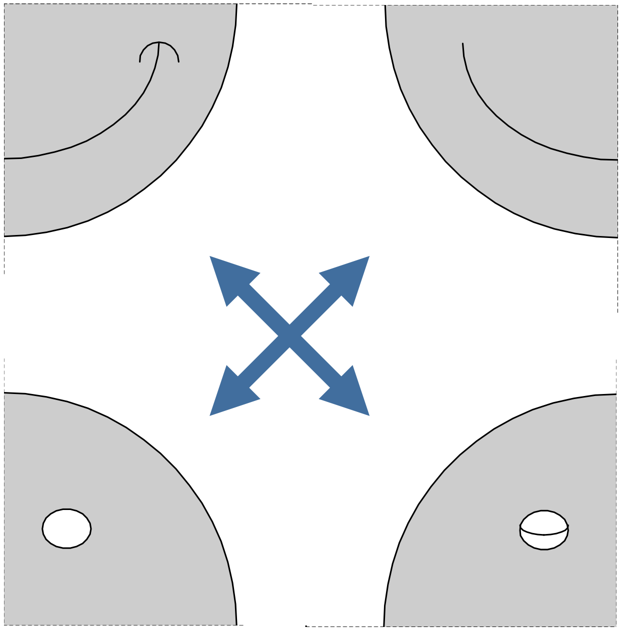


fftshift in 2D

Use fftshift for 2D functions

– >>smiley2 = fftshift(smiley);





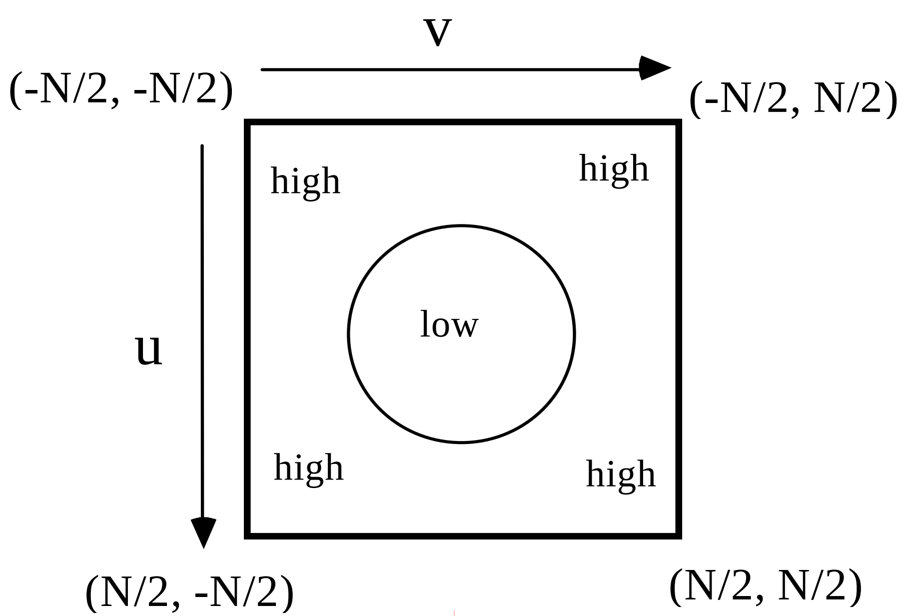
- fftshift(B)

$$\tilde{\mathbf{X}} = \begin{bmatrix} 77 & 2-5j & 3 & 2+5j \\ 4-9j & -11+8j & -4-7j & -5-4j \\ -13 & -6+13j & -11 & -6-13j \\ 4+9j & -5+4j & -4+7j & -11-8j \end{bmatrix}$$

- ans =

-11.0000	-6.0000 -13.0000i	-13.0000	-6.0000 +13.0000i
-4.0000 + 7.0000i	-11.0000 - 8.0000i	4.0000 + 9.0000i	-5.0000 + 4.0000i
3.0000	2.0000 + 5.0000i	77.0000	2.0000 - 5.0000i
-4.0000 - 7.0000i	-5.0000 - 4.0000i	4.0000 - 9.0000i	-11.0000 + 8.0000i

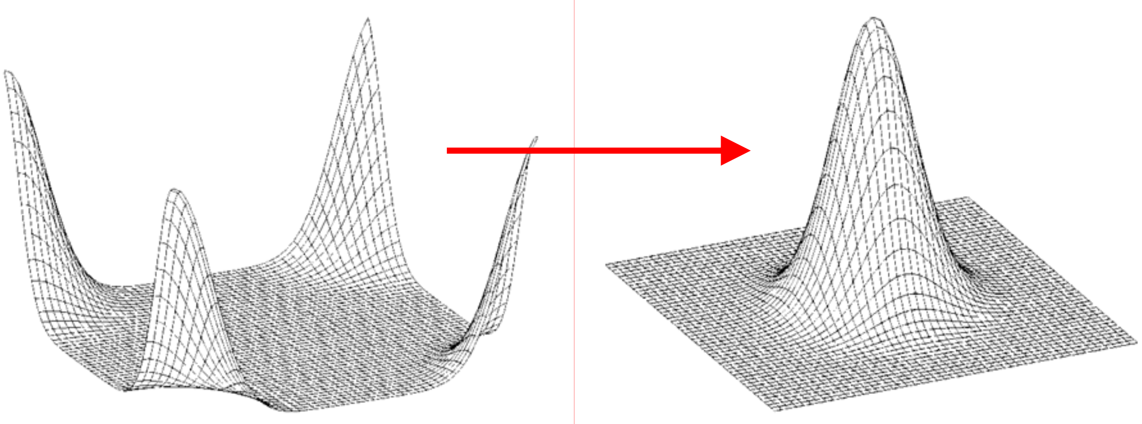
Centered Representation



MATLAB function: *fftshift*

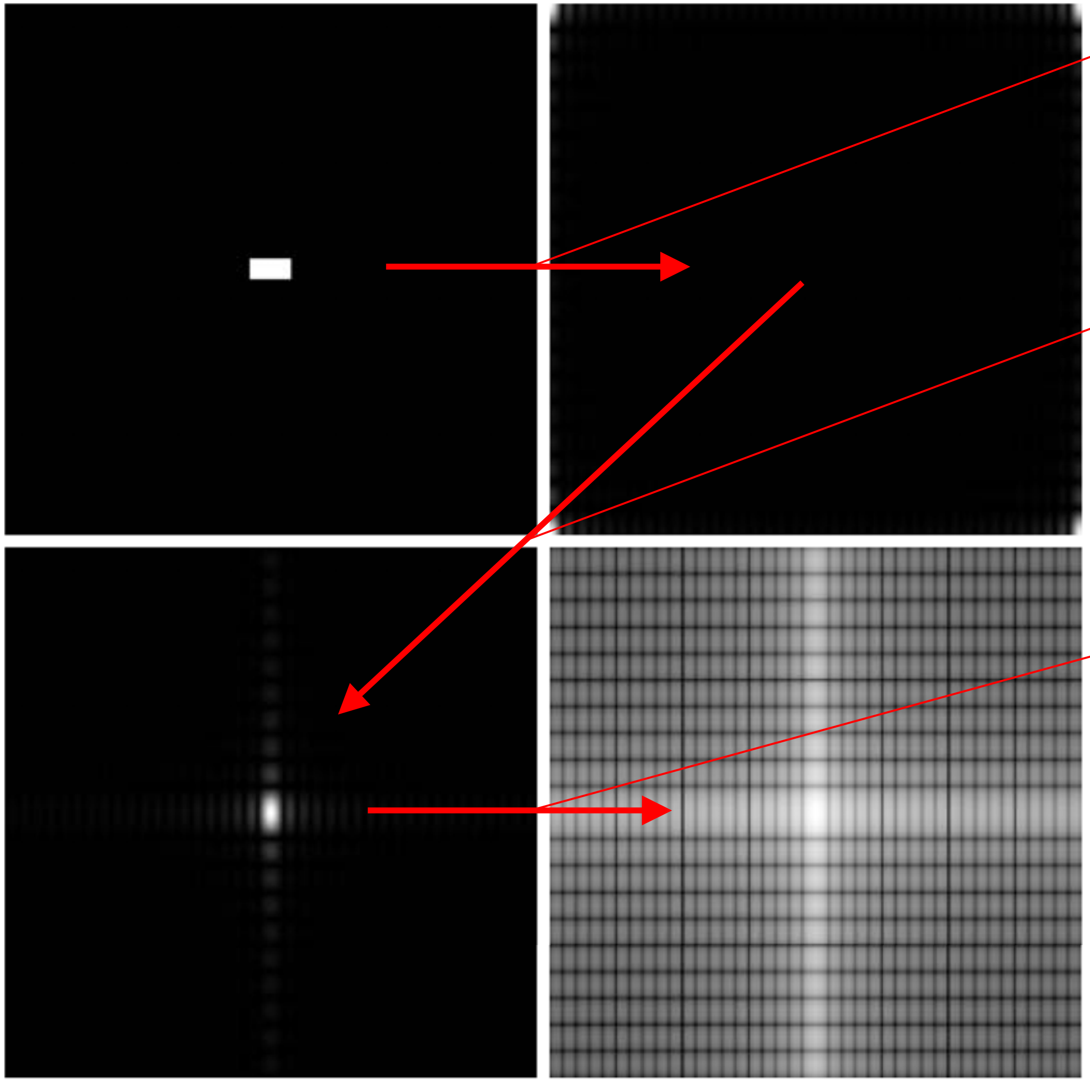
From Prof. Al Bovik

Example:



From [Gonzalez & Woods]

Log-Magnitude Visualization



From [Gonzalez & Woods]

Periodicity

- [M,N] point DFT is periodic with period [M,N]

$$f[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

$$f[m+M, n+N] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}(m+M) + \frac{l}{N}(n+N)\right)}$$

$$= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)} e^{j2\pi\left(\frac{k}{M}M + \frac{l}{N}N\right)}$$

1

$$= f[m,n]$$

2D-DFT (Frequency) Domain Filtering

Convolution Theorem

$f(x,y)$	$h(x,y)$	$g(x,y)$
input image	impulse response (filter)	output image

$$g(x,y) = f(x,y) \otimes h(x,y)$$

DFT ↓ ↑ IDFT   DFT ↓ ↑ IDFT   DFT ↓ ↑ IDFT

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

Frequency Domain Filtering

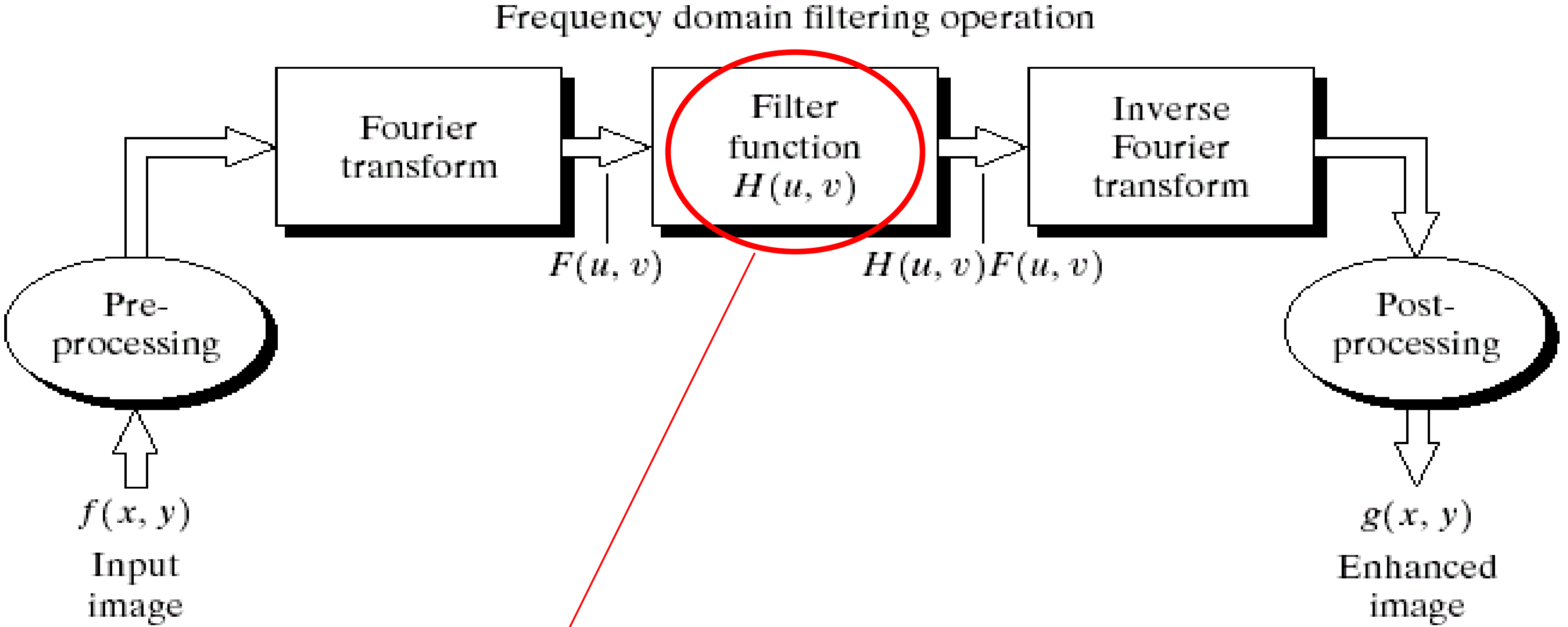


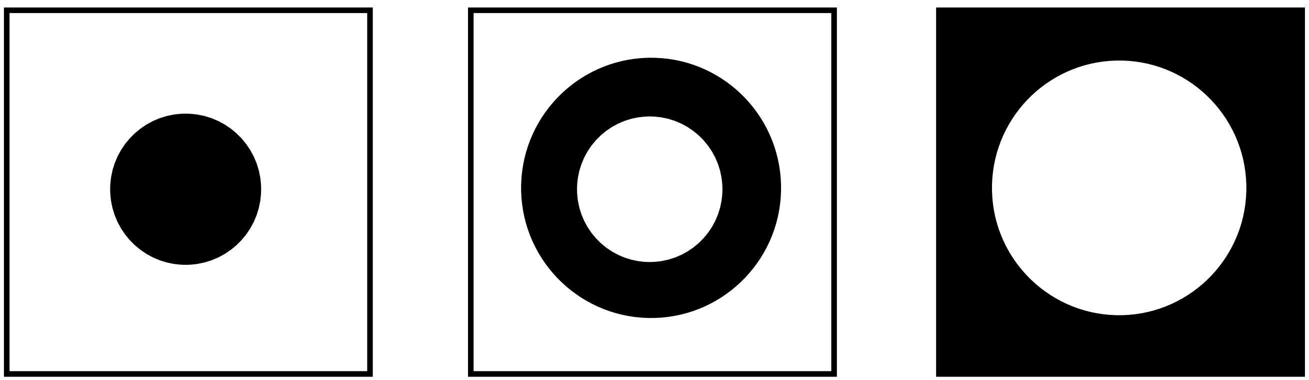
FIGURE 4.5 Basic steps for filtering in the frequency domain.

Filter design: design $H(u,v)$

From [Gonzalez & Woods]

2D-DFT Domain Filter Design

- Ideal lowpass, bandpass and highpass



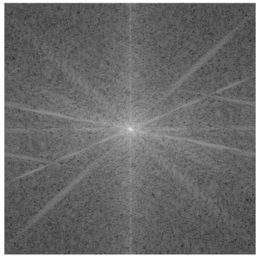
low-frequency mask | mid-frequency mask | high-frequency mask

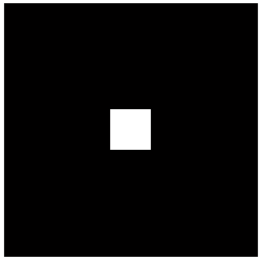
Procedure for Filtering in the Frequency Domain

1. Multiply the input image by (-1)x+y to center the transform
2. Compute the DFT F(u,v) of the resulting image
3. Multiply F(u,v) by a filter G(u,v)
4. Computer the inverse DFT transform h*(x,y)
5. Obtain the real part h(x,y) of 4
6. Multiply the result by $(-1)^{x+y}$

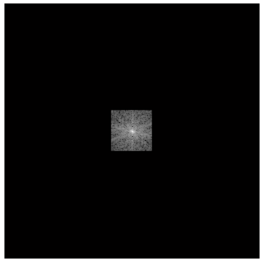
DFT-Domain Filtering

```
a = imread('cameraman.tif');
Da = fft2(a);
Da = fftshift(Da);
figure; imshow(log(abs(Da)),[]);

H = zeros(256,256);
H(128-20:128+20,128-20:128+20) = 1
figure; imshow(H,[]);

Db = Da.*H;
Db = fftshift(Db);
b = real(ifft2(Db));
figure; imshow(b,[]);
```





H



Frequency domain

Spatial domain