Filtering Operation in Frequency Domain-1

Fundamentals

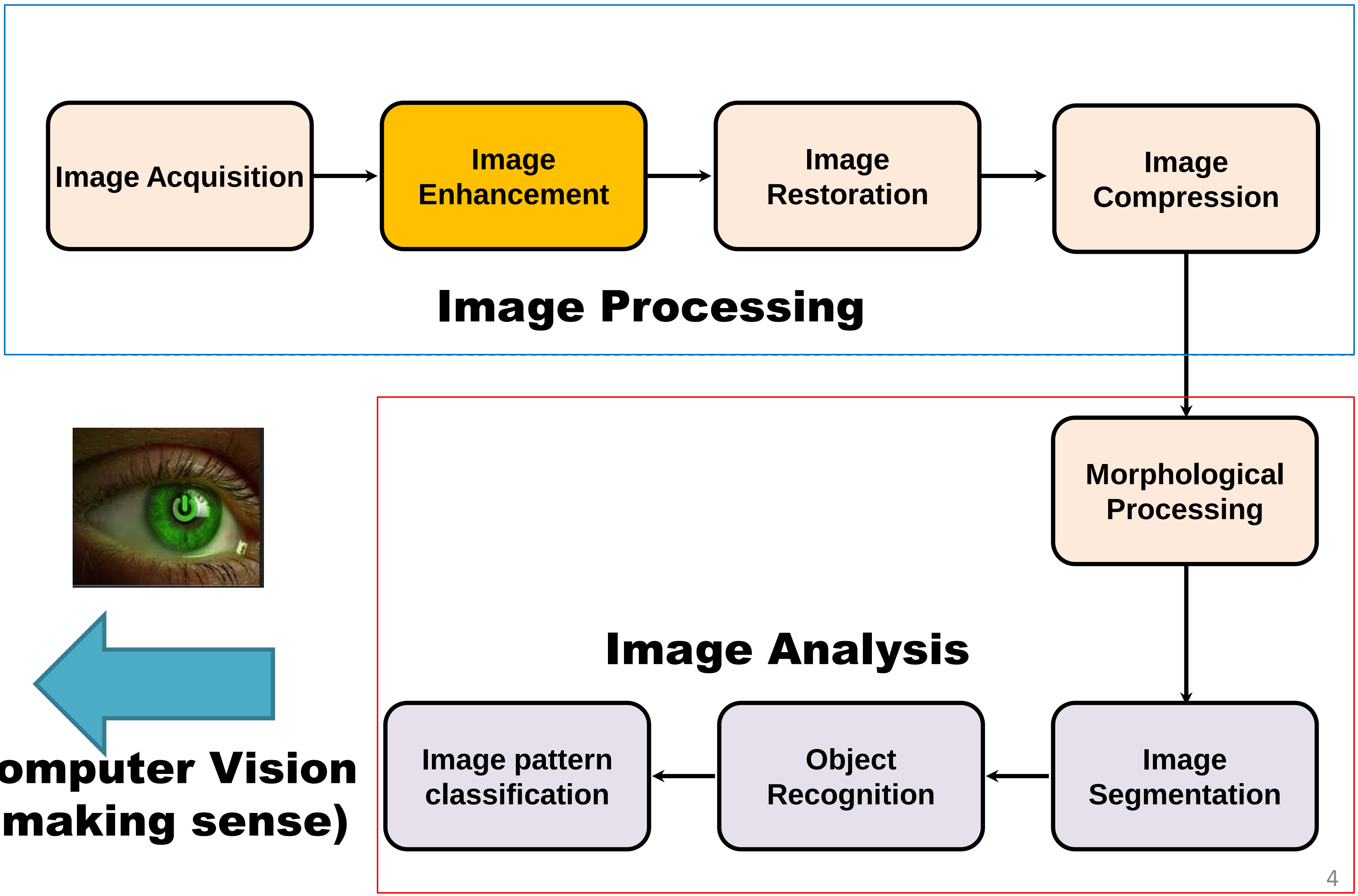
Recap

- DFT of one variable
- DFT of two variables
- How to overcome Wraparound Error?
- Properties of the 2-D DFT and IDFT

Lecture Objectives

- Filtering in Frequency Domain -Basic Observations
- Filtering in Frequency Domain – Requirements
- What About the Padding for Filters in Frequency Domain?
- Steps for Filtering in the Frequency Domain
- Correspondence Between Filtering in Spatial and Frequency Domain
- Constructing Spatial Filters from Frequency Domain Filters
- Constructing Frequency Domain Filters from Spatial Filters

Key Stages in DIP



Computer Vision (making sense)

Filtering in Frequency Domain
Basic Observations

Basic Observations

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

for **u** = 0, 1, 2,…,M–1 and **v** = 0, 1, 2,…, N-1

- **Each** term of **F(u,v)** contains **all** values of **f(x,y)**, modified by multiplying the values of the exponential terms
  - Establishing a direct one-to-one correspondence between the components of image and its transform is not possible.
  - Only general relationships can be described like, the frequency range available in image.

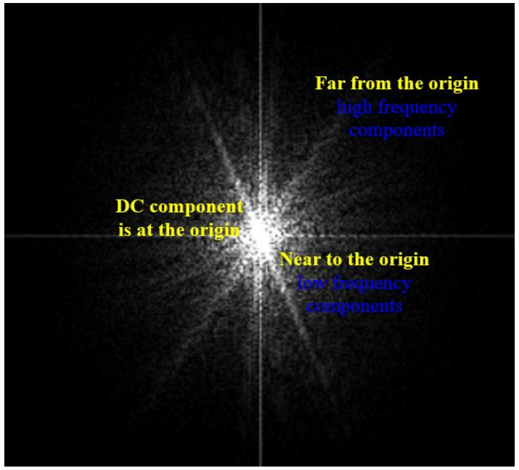
- **Frequency ≈ spatial rates of change in the image**
  - Frequencies in the Fourier transform are intuitively related with patterns of intensity variations in the image.

Frequency Values with (u,v) Coordinates

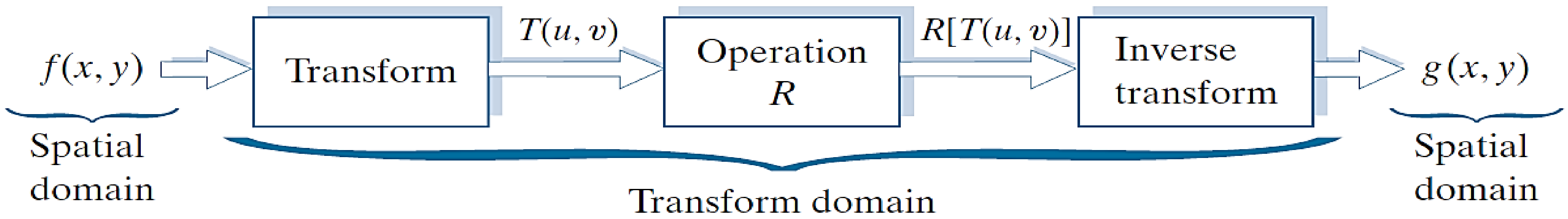
- **F(0,0)** -> slowest varying frequency value (constant, 'dc' term)
  - Proportional to the average value of the entire image pixel intensities

- Values **near the origin** correspond to the slowly varying intensity components in image --> smoother regions in the image
  - **Walls, clear sky, plain diffuse surfaces**

- Values **far from the origin** correspond to the faster varying intensity components in image --> higher intensity variation regions in the image
  - **edges**



Recall: Image Domain Transforms



$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, r(x,y,u,v)$$ Linear Transform

Forward transformation kernel

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v)\, s(x,y,u,v)$$ Inverse Linear Transform

Inverse transformation kernel

Filtering in Frequency Domain
Requirements

How to Filter in the Frequency Domain?

- Basic tool to work with: **Fourier Transform.**

- **Spatial filtering** works on the intensity values.

- **Frequency domain filtering** works on the Fourier transform values.
  1) compute the Fourier transform of the image.
  2) modify the Fourier transform of an image (apply filters).
  3) then compute the inverse transform to obtain the spatial domain representation of the processed result.

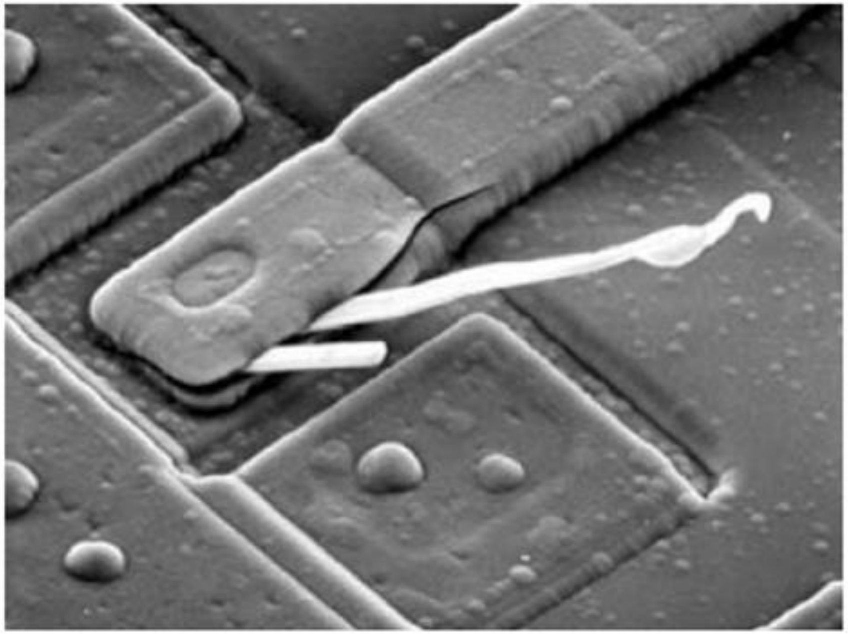
- What values do we have?
  - Fourier spectrum (magnitude)
  - Phase angle

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

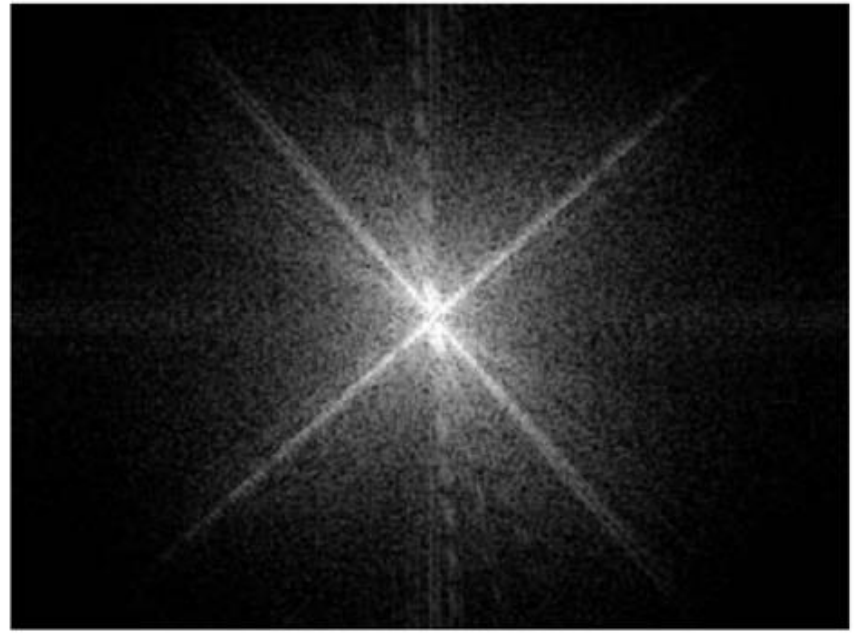
$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

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

Example



Scanning Electron Microscope image of an damaged Integrated Circuit



it's Fourier spectrum

- Strong edges in ±45 ° directions
- The "defect" has edges in other directions
  - White, oxide protrusions

- Strong edges in ±45° directions
- The "defect" is "visible" as patterns near the vertical direction

Filtering Operation in Frequency Domain

- Given (a padded) digital image, ***f* (*x*, *y*)**, of size ***P* × *Q*** pixels, the basic filtering equation in which we are interested has the form:

$$g(x,y) = \text{Real}\left\{\Im^{-1}\left[H(u,v)F(u,v)\right]\right\}$$

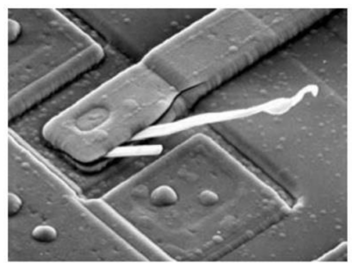
  - $\Im^{-1}$ is the IDFT
  - *F*(u,v) is the **DFT** of the input image *f*(*x*, *y*) in which *F*(0,0) is **centered** at F(u,v) by *multiplying the input image* by **(–1)$^{x+y}$** prior to computing F(u,v)
  - *H*(u,v) is a *filter function* which is symmetric about its center
  - *g*(*x*, *y*) is the *filtered* (*output*) *image*
  - Functions ***F***, ***H***, and ***g*** are arrays of size *P* ×*Q*, the *same as the padded input image*
  - The *product* ***H*(u,v)*F*(u,v)** is formed using elementwise multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Simple Frequency Domain Filter - example

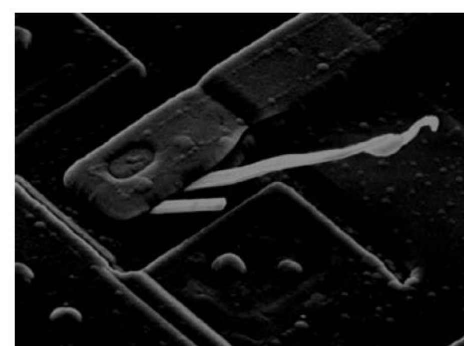
- To **mask out** the **DC term** (average intensity) - **F(0,0)**

$$H(u,v) = \begin{cases} 0 & \text{if } u = 0 \text{ and } v = 0 \\ 1 & \text{otherwise} \end{cases}$$

$\mathfrak{F}^{-1}\Big\{\mathfrak{F}\Big\{$  $\Big\} \times \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}\Big\}$

f(x,y) $\times$ $(-1)^{x+y}$

$=$



Frequency Domain Filter Nomenclature

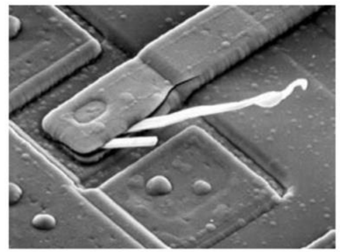
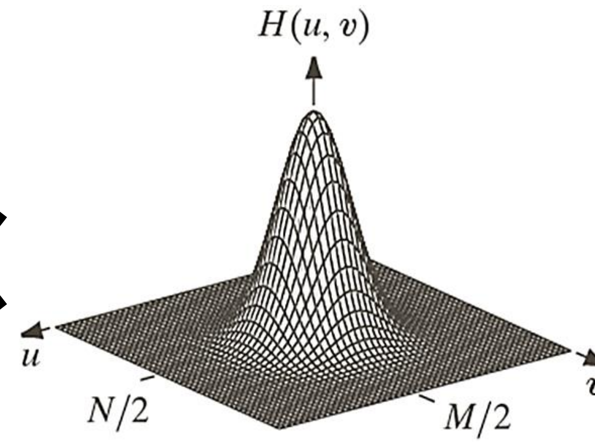
- **Low frequencies** : regions of smooth intensities

- **High frequencies** : abrupt transitions in the intensity values

- **Lowpass filter** : Pass the low frequencies, attenuate the high frequencies
  - Blur the image
  - Blur the edges

- **Highpass filter** : Pass the high frequencies, attenuate the low frequencies
  - Enhance edges, sharpen the image
  - Lower the contrast in the image

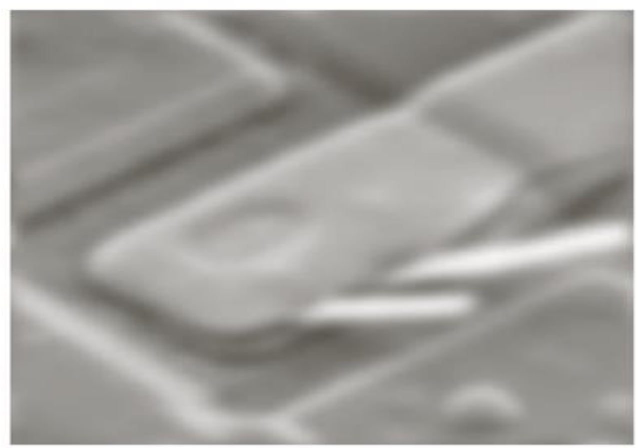
Lowpass Filter

$\mathfrak{F}^{-1}\Big\{\mathfrak{F}\Big\{$  $\Big\}$ ✖ 

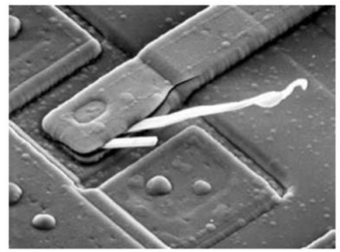
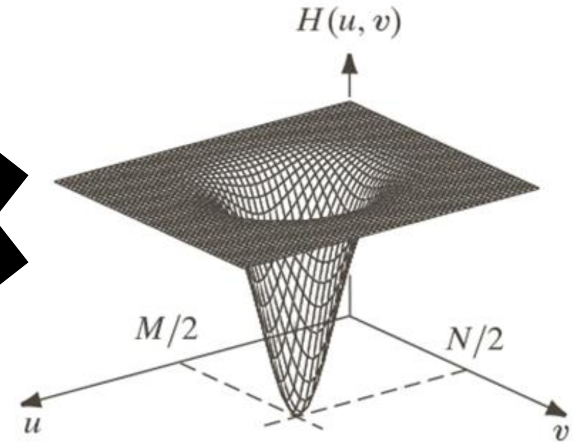
$\Big\}$

f(x,y) ✖ $(-1)^{x+y}$

=

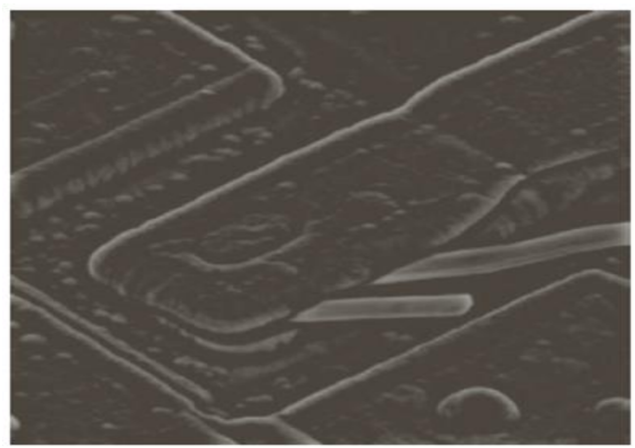


Highpass Filter

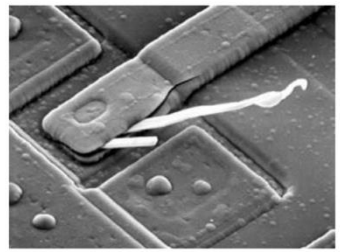
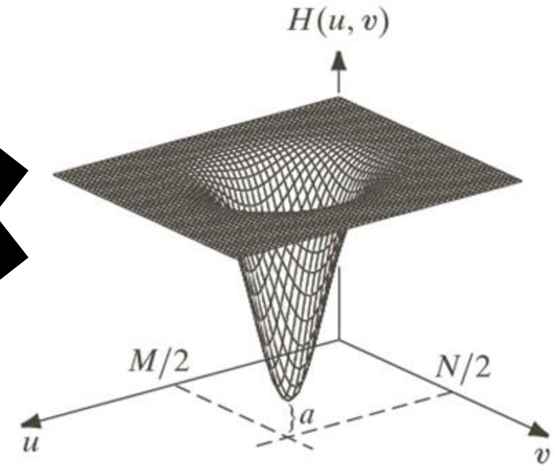
$\mathfrak{F}^{-1}\Big\{\mathfrak{F}\Big\{$  $\Big\}$ $\times$ 

$\Big\}$

f(x,y) $\times$ $(-1)^{x+y}$

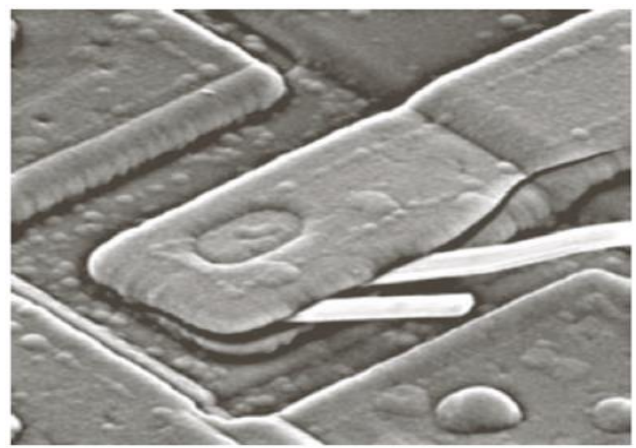
$=$ 

Offset Highpass Filter

$\mathfrak{F}^{-1}\Big\{\mathfrak{F}\Big\{$  $\Big\} \times$ 

$\Big\}$

f(x,y) $\times$ $(-1)^{x+y}$

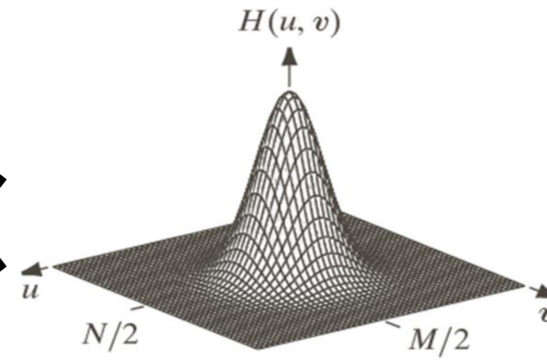
$=$ 

Adding a small constant to the highpass filter does not affect sharpening appreciably, but it does ***prevent*** elimination of the **dc term** and thus ***preserves tonality***.

Filtering with Padding the input image

$\mathfrak{F}^{-1}\Big\{\mathfrak{F}\big\{$

$\big\}\times$



$\Big\}$

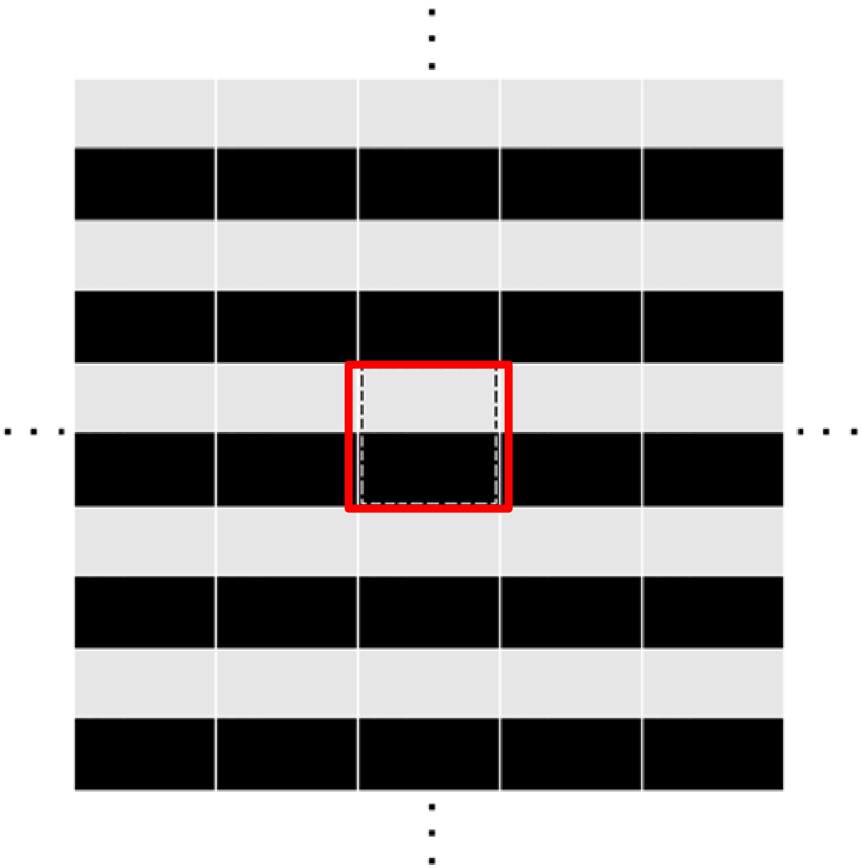
f(x,y) × $(-1)^{x+y}$

$=$

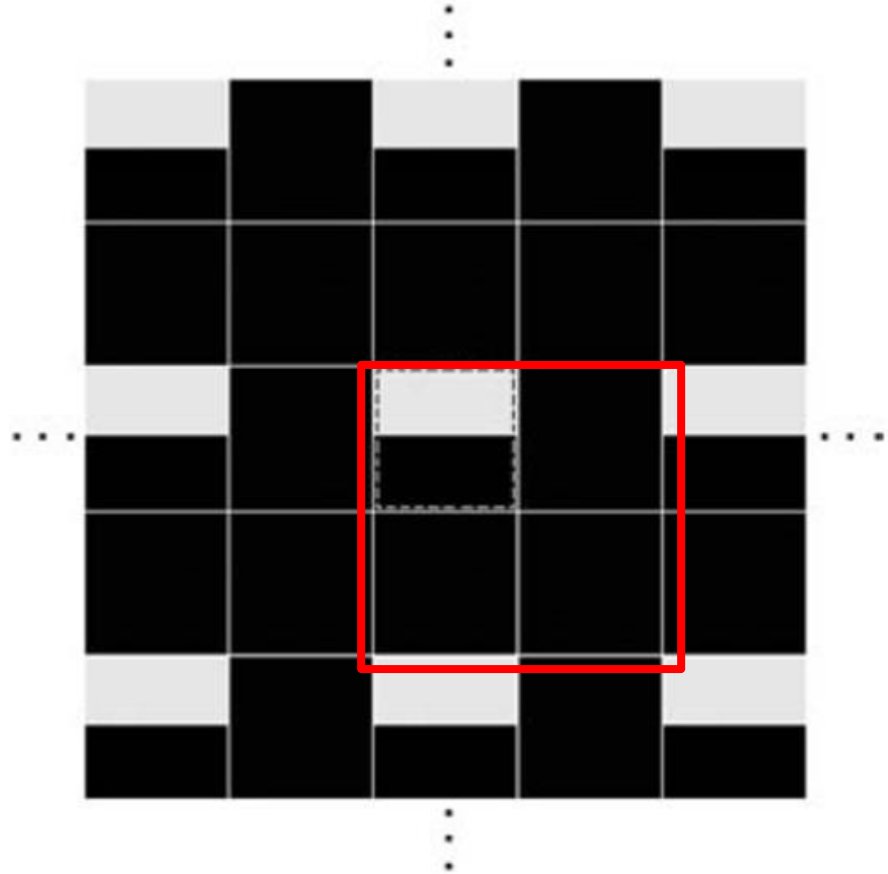
No Padding

Padding

Effect of Padding on Input Image



No Padding



Padding

What About the Padding for Filters in Frequency Domain?

Simple Idea (Not preferred)

Steps :

1. Construct the filter in the frequency domain having same size as the unpadded input image

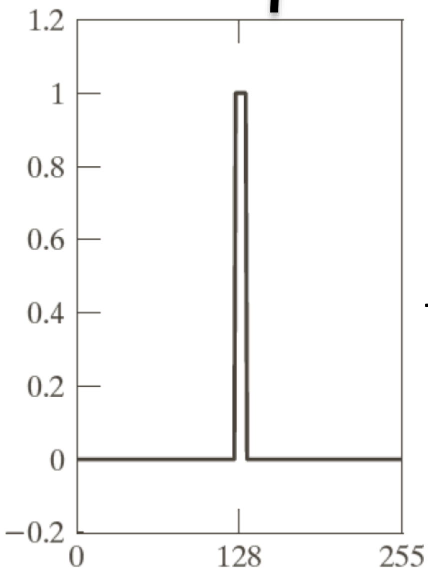
2. Perform IDFT of this filter.
   - Frequency domain -> spatial domain

3. Do the Zero padding in the spatial domain.

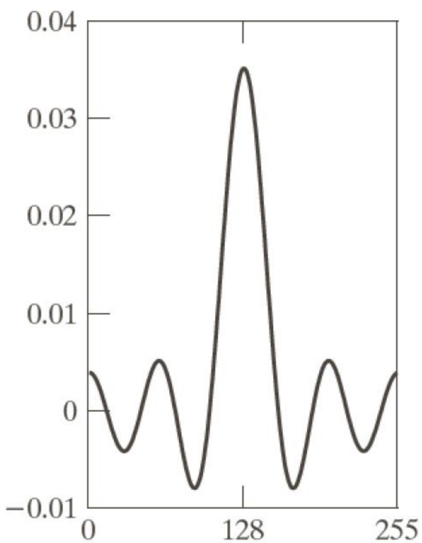
4. Perform DFT to obtain the frequency domain filter.
   - Spatial domain -> frequency domain

Simple Idea (Not preferred)

Filter structure differs

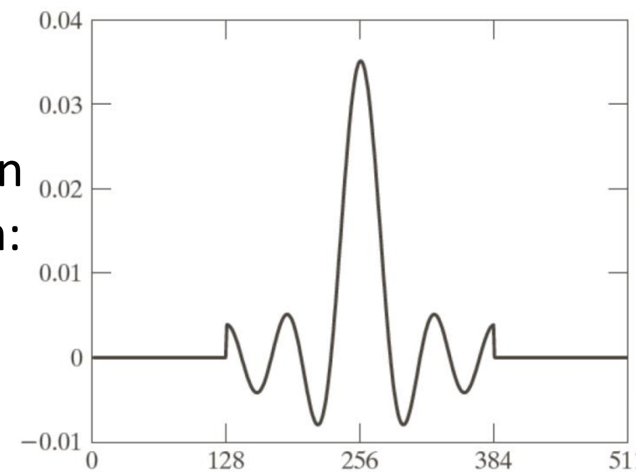


Frequency domain filter: H(u,v)

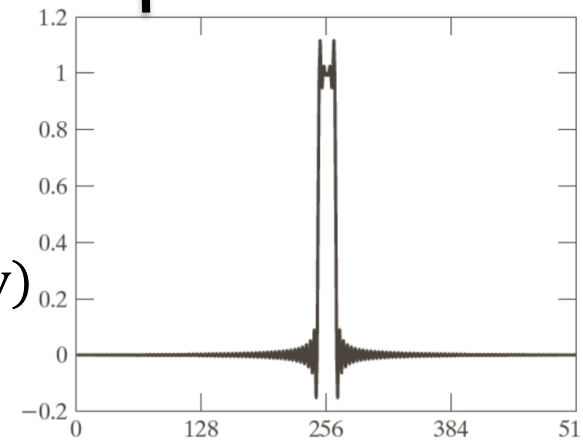


IDFT in spatial domain: h(x,y)

Zero padding in spatial domain: h’(x,y)



Resulting filter in frequency domain: H’(u,v)



Will produce **ringing effect** in the final result

Simple Idea (Not preferred) – Solution ??

Steps :

1. Pad the input image (in spatial domain of course…).

2. Construct the filter in frequency domain having same size as the padded input image.

- This approach will result in wraparound error **because no padding is used for the filter transfer function**.
  - this error is mitigated significantly by the separation provided by padding the input image
  - it is preferable to ringing
  - Results in wraparound error, but smoother images

Effect of Frequency Filtering on the Phase Angle

$$F(u,v)=R(u,v)+jI(u,v)$$

- Substitute it in $g(x,y)=F^{-1}[H(u,v)F(u,v)]$

$$g(x,y)=F^{-1}[H(u,v)R(u,v)+jH(u,v)I(u,v)]$$

- Phase angle is given by:

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

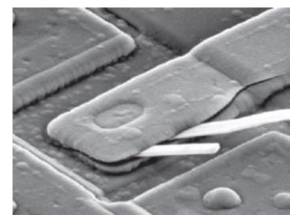
- $\Phi_G(u,v)=\Phi_F(u,v)$ since $H(u,v)$ cancel out

- The filters that have **no effect on the phase angle**, are appropriately called as ***zero-phase-shift filters***

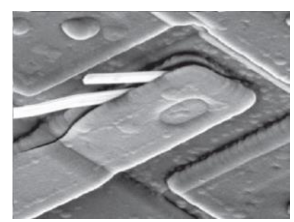
Effects of Phase Angle Change

- $F(u,v)=|F(u,v)|e^{j\phi(u,v)}$ - Original image
- $F_1(u,v)=|F(u,v)|e^{j[-1\phi(u,v)]}$ – Phase angle × (-1)
- $F_2(u,v)=|F(u,v)|e^{j[0.25\phi(u,v)]}$ – Phase angle × 0.25

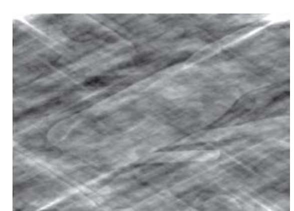
$$\mathfrak{F}^{-1}\left\{F(u,v)\right\} =$$



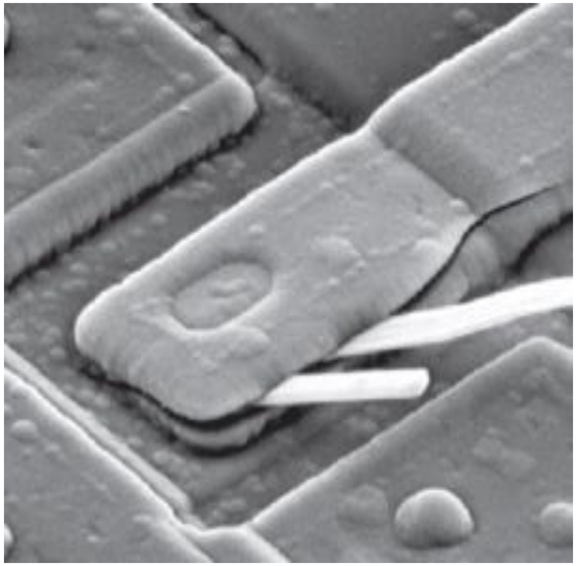
$$\mathfrak{F}^{-1}\left\{F_1(u,v)\right\} =$$



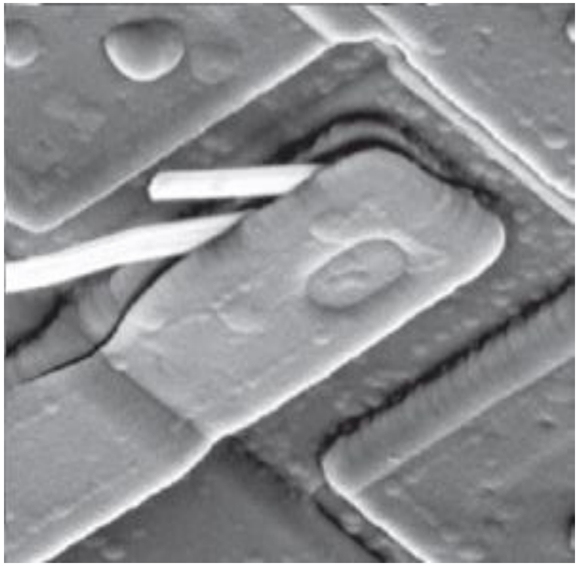
$$\mathfrak{F}^{-1}\left\{F_2(u,v)\right\} =$$



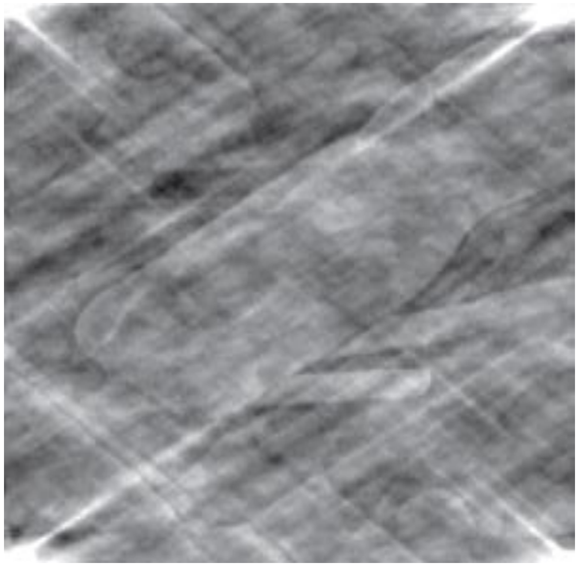
Effects of Phase Angle Change



Original Image



Phase angle × (-1)



Phase angle × (0.25)

These two results illustrate the advantage of using <u>frequency-domain filters that do not alter the phase angle</u>.

Steps for Filtering in the Frequency Domain

Important !!!

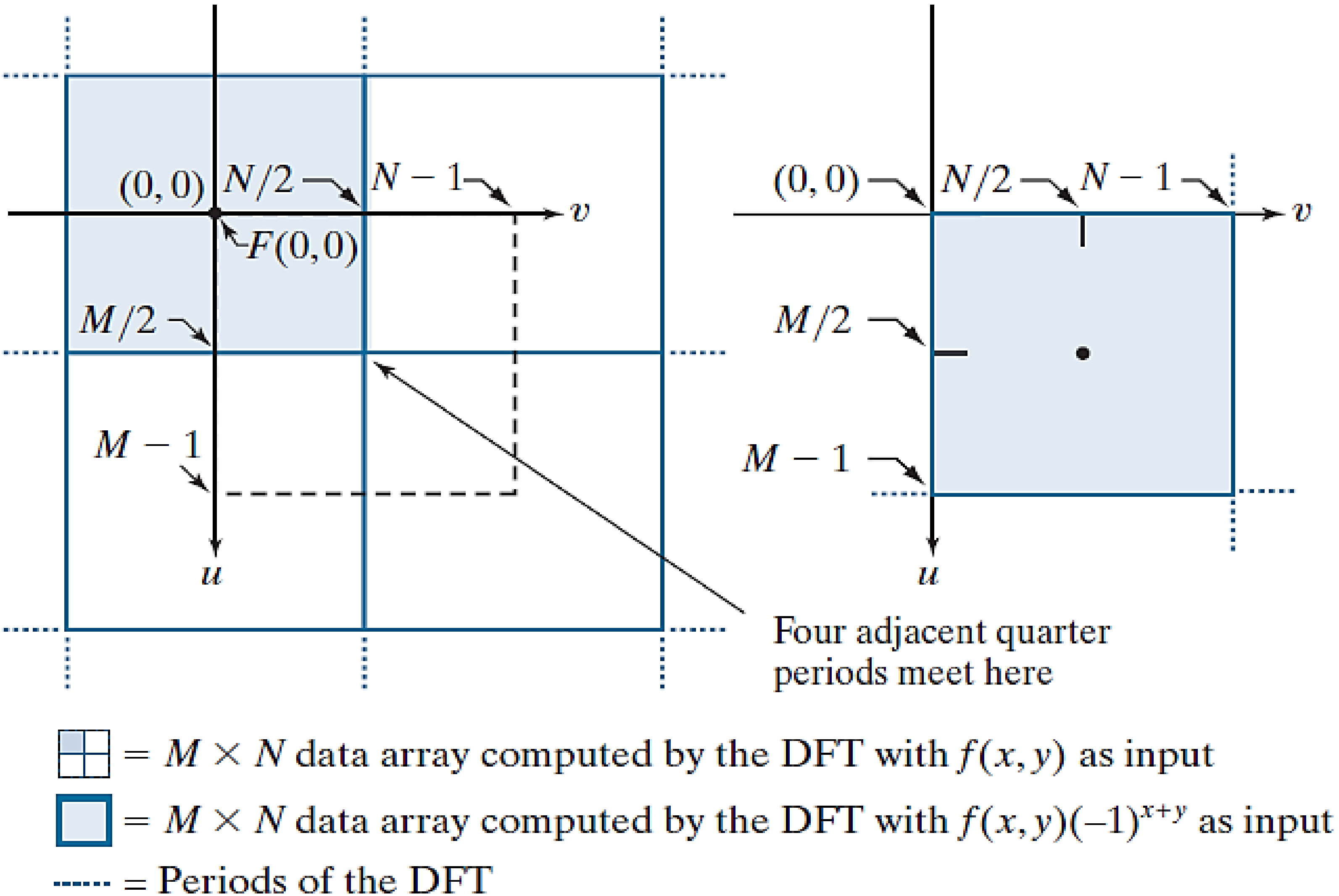
Steps for Filtering in the Frequency Domain

1. Given an input image $f(x,y)$ of size $M \times N$, obtain the padding parameters P and Q (typically, $P = 2M$ and $Q = 2N$).

2. Form a zero padded image $f_p(x,y)$ of size $P \times Q$ using zero-, mirror-, or replicate padding to the image $f(x,y)$.

3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its transform.

3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its transform



Steps for Filtering in the Frequency Domain

4. Compute the DFT, $F(u,v)$ of the image $f_p(x,y)$ from step-3.

5. Construct a real, symmetric filter $H(u,v)$ of size $P \times Q$ with center at the location $(P/2, Q/2)$.

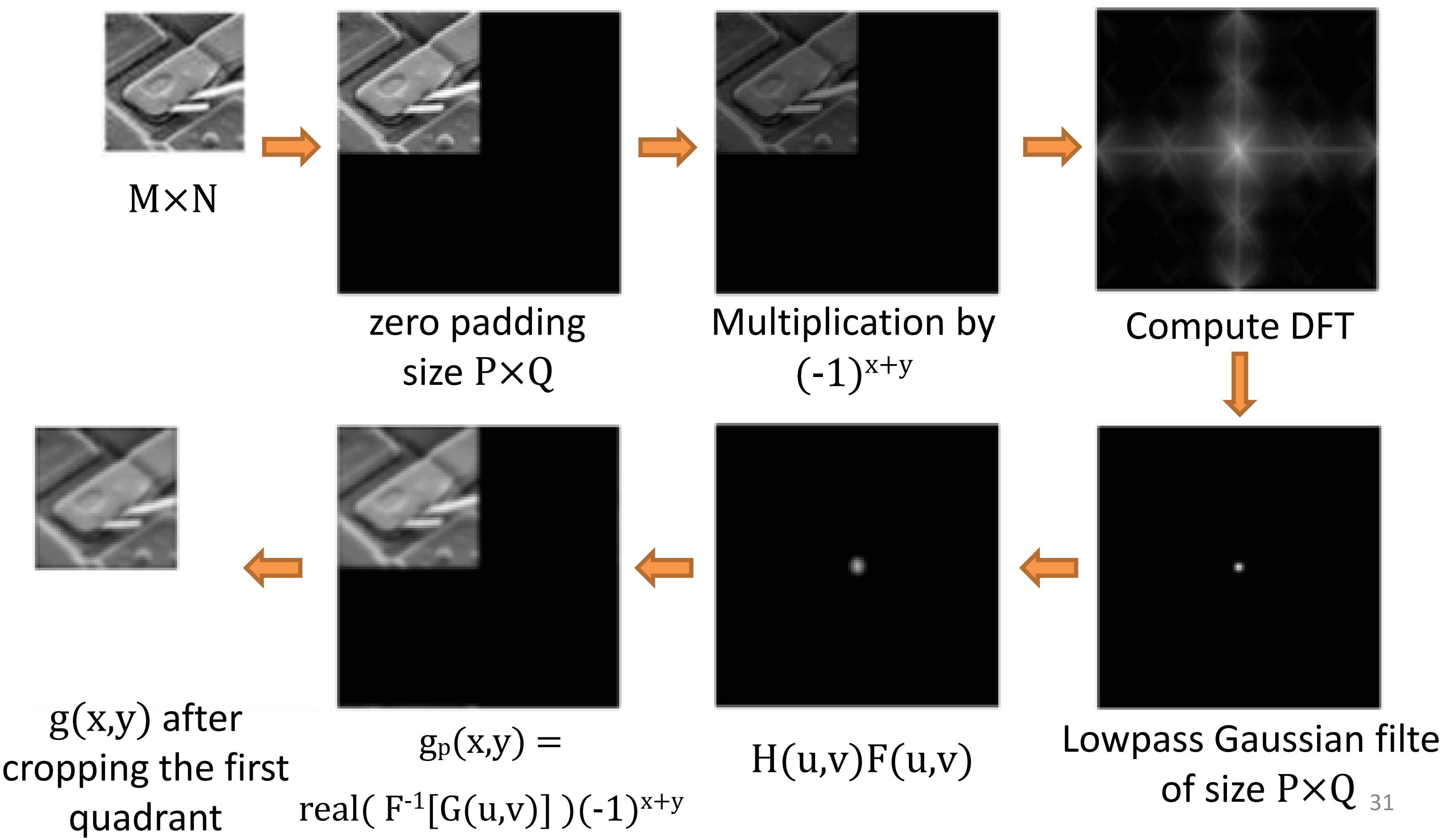
6. Form the product $G(u,v) = H(u,v) \cdot F(u,v)$ using the elementwise multiplication operation for $u = 0, 1, 2, \ldots, M-1$ and $v = 0, 1, 2, \ldots, N-1$.

7. Obtain the filtered image (of size $P \times Q$) by computing the IDFT of $G(u,v)$:

$$g_p(x,y) = \left(\text{real}\left[\Im^{-1}\{G(u,v)\}\right]\right)(-1)^{x+y}$$

8. Obtain the final image $g(x,y)$ of the same size as the input image by extracting the $M \times N$ region from the top, left quadrant of $g_p(x,y)$.

Filtering Explained with Example



Correspondence Between Filtering in Spatial and Frequency Domain

Correspondence between filtering in the Spatial and Frequency Domains

- The link between Spatial and Frequency Domains is the Convolution Theorem.

$$f \star h)(x, y) \Leftrightarrow (F \bullet H)(u, v)$$

- Filtering in frequency domain is an elementwise product of H(u,v) and F(u,v).

- **Given: H(u,v), can we find corresponding h(x,y) filter in spatial domain?**

Computing h(x,y)

- Let $f(x,y)=\delta(x,y)$, then $F(u,v)=1$
- Hence, **h(x,y)** = $\mathrm{Real}\left\{\Im^{-1}\left[H(u,v)F(u,v)\right]\right\}$ $=\Im^{-1}\{H(u,v)\}$, and it is the inverse transform of the frequency domain filter.
- So, h(x,y) is the corresponding filter in the spatial domain.
- The converse is also true: i.e., H(u,v)=F{h(x,y)}
- We conclude that: $\mathbf{h(x,y)\Leftrightarrow H(u,v)}$ form a discrete Fourier transform pair for a **filter**.

Properties of $h(x,y)$

- $h(x,y)$ is obtained by inverse discrete Fourier transform(IDFT) of the frequency domain filter with Fourier transform of an impulse function.
  - Also known as the impulse response of $H(u,v)$

- All quantities in discrete representations of $H(u,v)$ and $h(x,y)$ are **finite**.
  - Such filters are known as finite impulse response (FIR) filters

- Spatial convolution filtering is well suited (speed) for small kernels using hardware and/or firmware implementation.

- When working with general purpose computers, frequency-domain filtering using a **fast Fourier transform** (FFT) algorithm can be hundreds of times faster than using spatial convolution.

Constructing Spatial Filters from Frequency Domain Filters

Constructing Spatial Filters from Frequency Domain Filters - example

- **Goal:** use full size (P×Q) frequency domain filters as a guide to specify the spatial filters for a much smaller neighborhood.

- We shall illustrate the method with **Gaussian functions**.
  - Recall: Both the *forward* and *inverse* Fourier transform of a Gaussian function are *real* Gaussian functions.

$$H(u) = Ae^{-u^2/2\sigma^2} \qquad h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2x^2}$$

1-D Gaussian Fourier transform pair

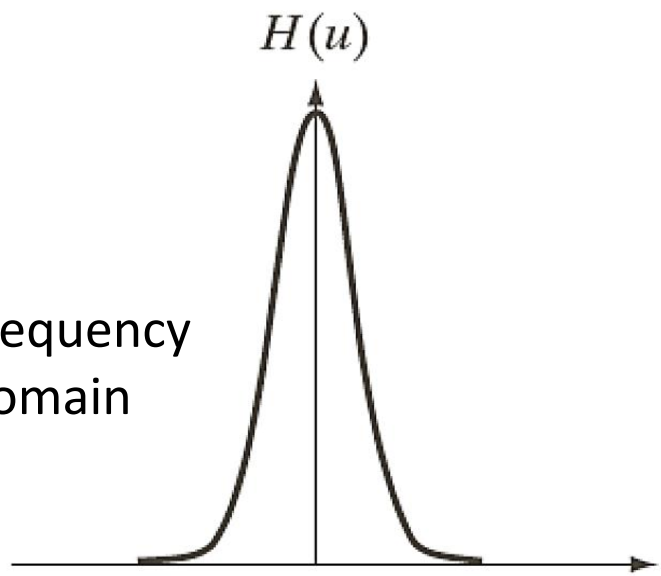
- Both $\mathrm{H(u)}$ and $\mathrm{h(x)}$ are real, and Gaussian functions
  - No complex terms (simplifies computations)

- Functions behave reciprocally. (**sigma term**)
  - When H(u) has $\sigma \to \infty$ => h(x) tends toward an **impulse**

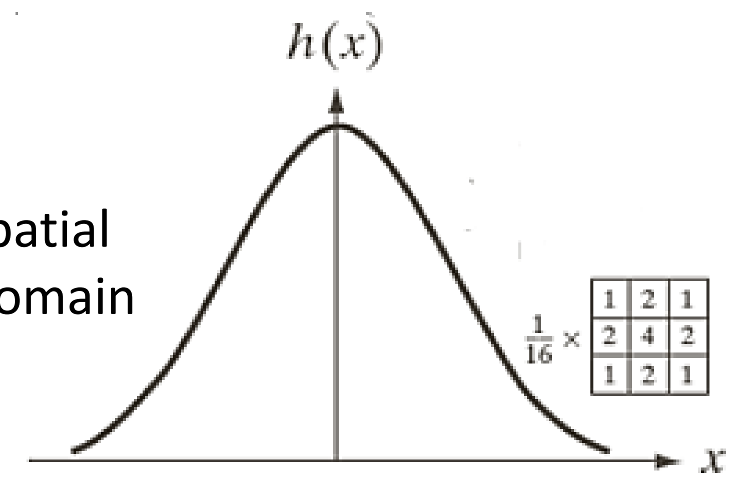
Constructing Spatial Filters from Frequency Domain Filters - example

Lowpass Filter with a Gaussian Function:

Frequency Domain



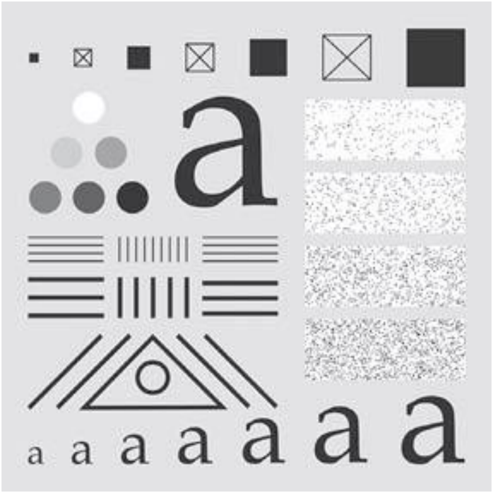
Spatial Domain



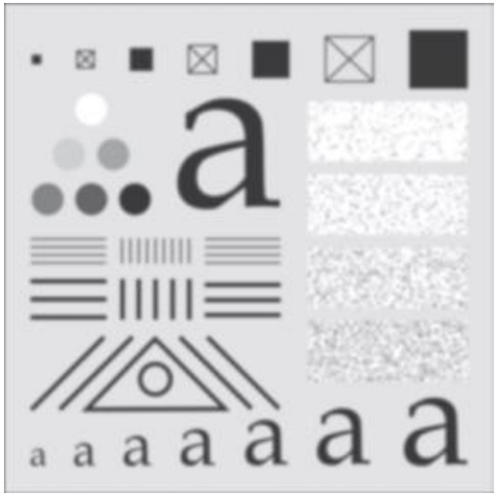
- All coefficients are positive.
- Largest value in center.
- Values decrease towards "outer" samples.

Constructing Spatial Filters from Frequency Domain Filters - example

Lowpass Filter with a Gaussian Function:



Test image of size 1024 x 1024



Filtering with **Gaussian kernel** of size **21 x 21**, with $\boldsymbol{\sigma}$ **= 3.5**

Filtering with **Gaussian kernel** of size **43 x 43**, with $\boldsymbol{\sigma}$ **= 7**

Constructing Spatial Filters from Frequency Domain Filters - example

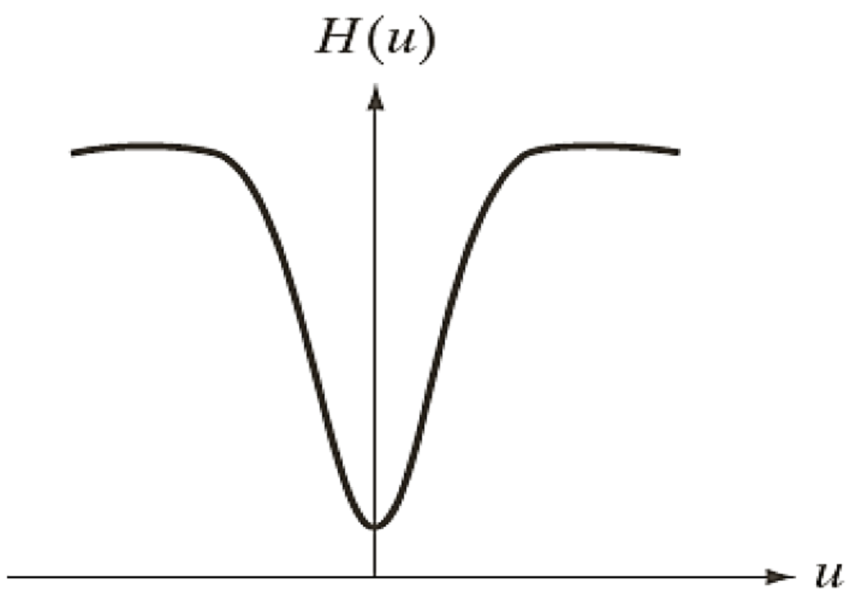
Highpass Filter with a Gaussian Function:

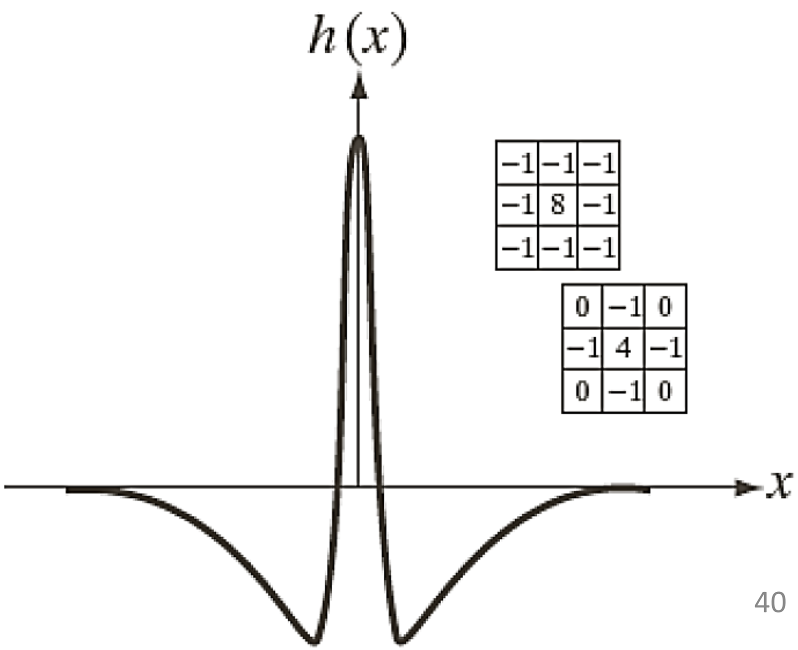
$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2}$$
$$A \geq B \quad and \quad \sigma_1 \geq \sigma_2$$

$$h(x) = \sqrt{2\pi}\sigma_1 Ae^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\sigma_2 Be^{-2\pi^2\sigma_2^2x^2}$$

Highpass Gaussian filter in **frequency domain** made of Two lowpass Gaussian functions

Corresponding Highpass Gaussian filter in **spatial domain**

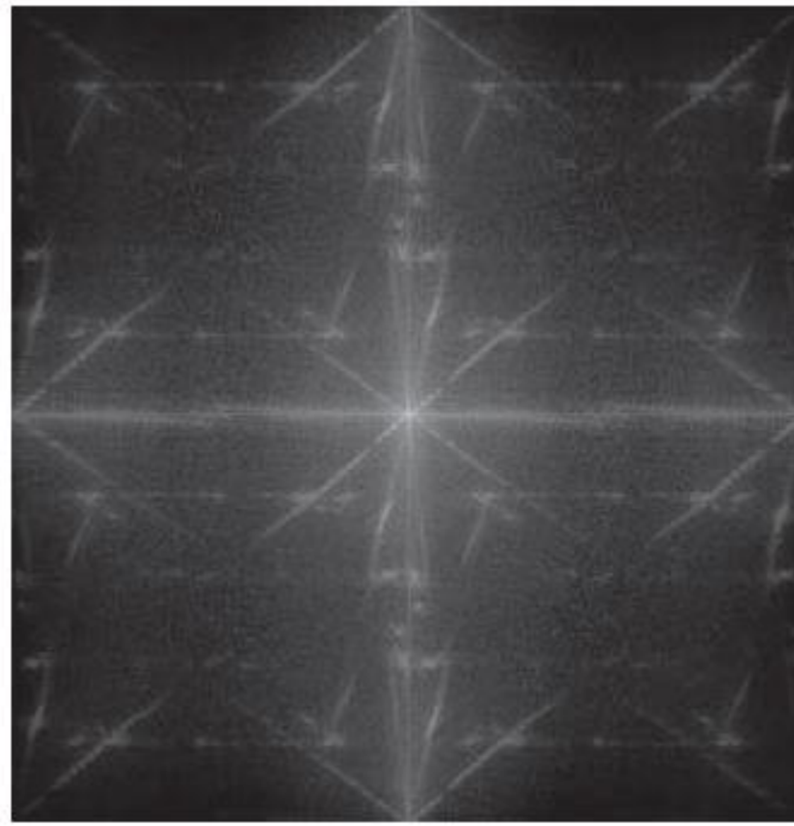




Constructing Frequency Domain Filters from Spatial Filters

Constructing Frequency Domain Filters from Spatial Filters - example

$f(x,y)$ of size 600×600



Its $F(u,v)$, centered

Constructing Frequency Domain Filters from Spatial Filters - example

- Sobel Filter h(x,y)
  - 3×3 kernel size
- Image Size f(x,y)
  - 600×600
- To avoid wraparound error, We need zero padding for both f(x,) and h(x,y) **!!!**
  - $P \geq A + B - 1$
  - $P \geq 600 + 3 - 1 = \mathbf{602}$

-1	0	1
-2	0	2
-1	0	1

Spatial Sobel filter h(x,y)

Constructing Frequency Domain Filters from Spatial Filters - example

Step-1: Sobel function exhibits ODD symmetry. However, its 1st element is not zero.

Step-2: To make it an perfect **odd function**, we have to add to it a **leading row and column of 0's**.

$$h(x,y) = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|} \hline 0 & 0 & 0 & 0 \\ \hline 0 & -1 & 0 & 1 \\ \hline 0 & -2 & 0 & 2 \\ \hline 0 & -1 & 0 & 1 \\ \hline \end{array}$$

{0,-b,0,b}

For an **odd function**, when ***M*** is an **even number**, a **1-D odd sequence** has always **zero** values for the points at locations *0* and *M/2*.

Constructing Frequency Domain Filters from Spatial Filters - example

Step-3: Place the Sobel filter from previous step into an larger array of zeros of the size 602 x 602 with both of their **centers coincide**.

Centered at index **(301,301)**

.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	**0**	**0**	**0**	**0**	.	.	.	.
.	.	.	**0**	**-1**	**0**	**1**	.	.	.	.
.	.	.	**0**	**-2**	**0**	**2**	.	.	.	.
.	.	.	**0**	**-1**	**0**	**1**	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

h(x,y) = 602 x 602 size array of zeros

Constructing Frequency Domain Filters from Spatial Filters - example

- **Very important point !!!** If odd (even) characteristic of a filter is not preserved, the filtering is not identical in spatial and frequency domains !!!.

- **Step-4:** Find the forward DFT of $h(x,y)$ resulting in the required $H(u,v)$ filter in the frequency domain.

  * Since $h(x,y)$ is **ODD** and **REAL** function, $H(u, v)$ will be **purely imaginary**.

  Recall:

  $f(x,y)$ real and odd $\Leftrightarrow$ $F(u,v)$ imaginary and odd

Constructing Frequency Domain Filters from Spatial Filters - example

The complete procedure to generate H(u,v) from h(x,y):

1. Pad $h(x,y)$ to form a 602×602 elements array: $h_p(x,y)$.

2. Multiply $h_p(x,y)$ with $(-1)^{x+y}$ to center the frequency domain filter.

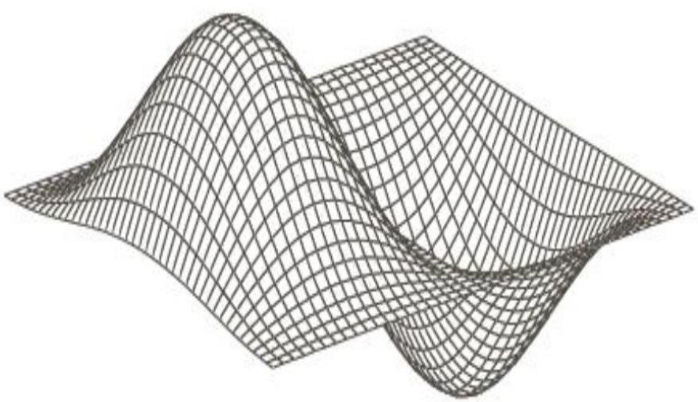
3. Compute the forward DFT of the result in step (2): $H(u,v)$.

4. Set the real part of $H(u,v)$ to **0** to facilitate numerical precision or to eliminate outliers. H has to be purely imaginary.

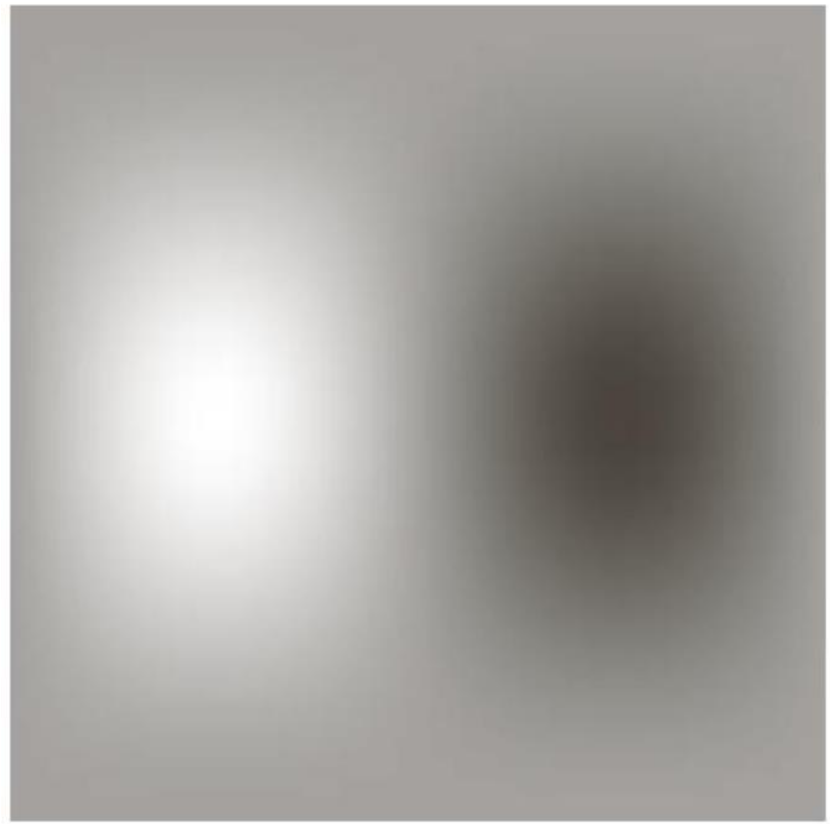
5. Finally, multiply $H(u,v)$ with $(-1)^{u+v}$ to compensate for the shift to the center of the spatial filter in step (2).

Constructing Frequency Domain Filters from Spatial Filters - example

-1	0	1
-2	0	2
-1	0	1

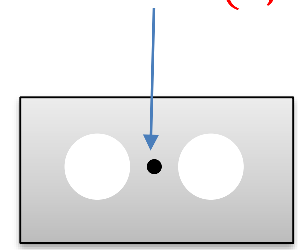


Perspective plot of the frequency domain filter

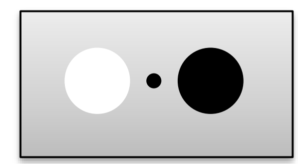


Fourier Spectrum of H(u,v)

Center of H(u,v)



Symmetry

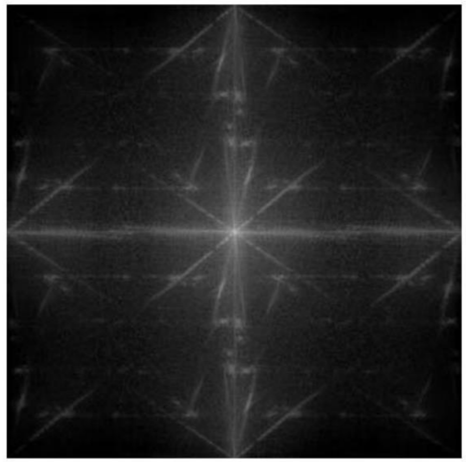


Antisymmetry

Note the **antisymmetry** in H(u,v) image about its center, a result of *H*(u,v) being odd.

Highpass filtering in Frequency Domain - example

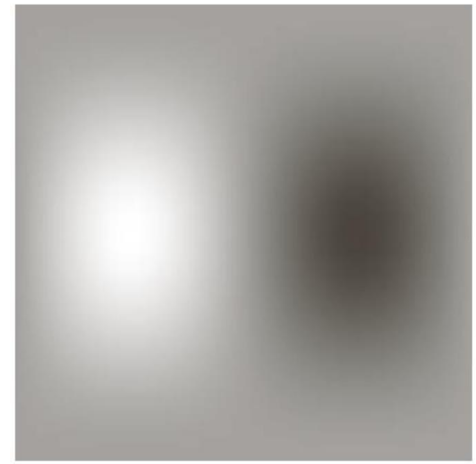
$\mathfrak{F}^{-1}\{$



F(u,v)

$\times$

multiplication



H(u,v)

$\}$

$=$

Highpass filtering in Spatial Domain - example

f(x,y)

★

convolution

-1	0	1
-2	0	2
-1	0	1

h(x,y)

=

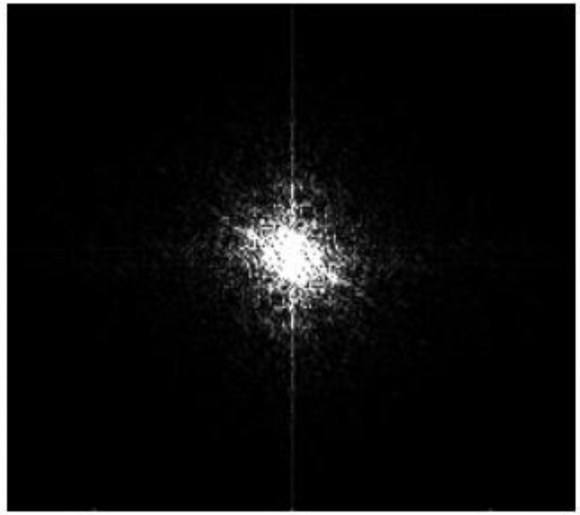
Highpass filtering in Spatial Domain Vs. Frequency domain


Frequency Domain Filter Result

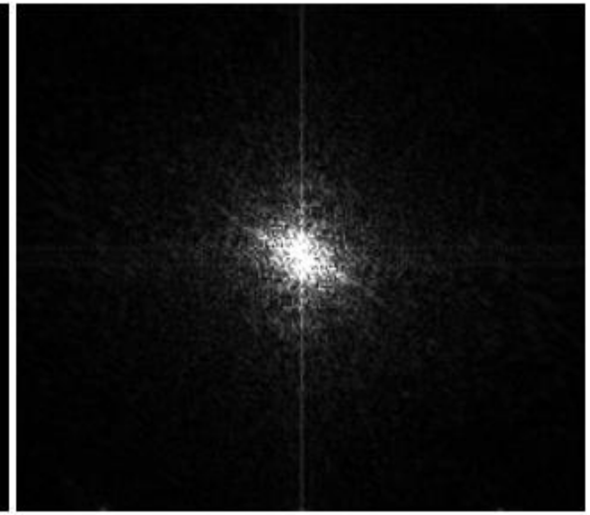

Spatial Domain Filter Result

Filtering - example

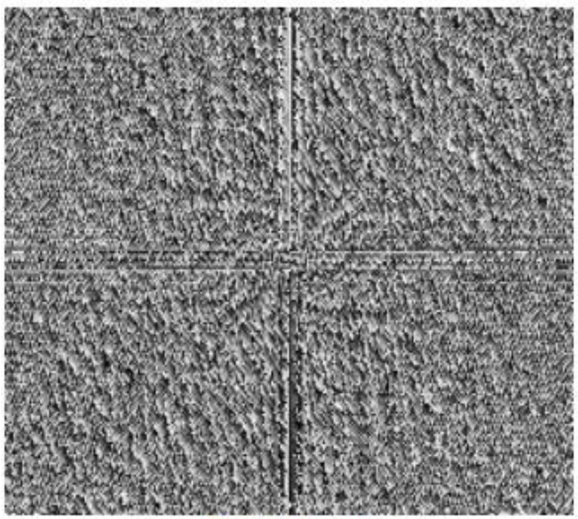
Original image



Power spectrum



Amplitude spectrum



Phase spectrum

High-pass filtering

Low-pass filtering

Next Lecture

- Image Smoothing Using Lowpass Frequency Domain Filters
- Image Sharpening Using Highpass Frequency Domain Filters
- Laplacian in the Frequency Domain
- Homomorphic Filtering
- Selective Filtering