Digital Image Processing

COSC 6380/4393

Lecture – 24

Nov. 9th, 2023

Pranav Mantini

Slides from Dr. Shishir K Shah, and Frank Liu

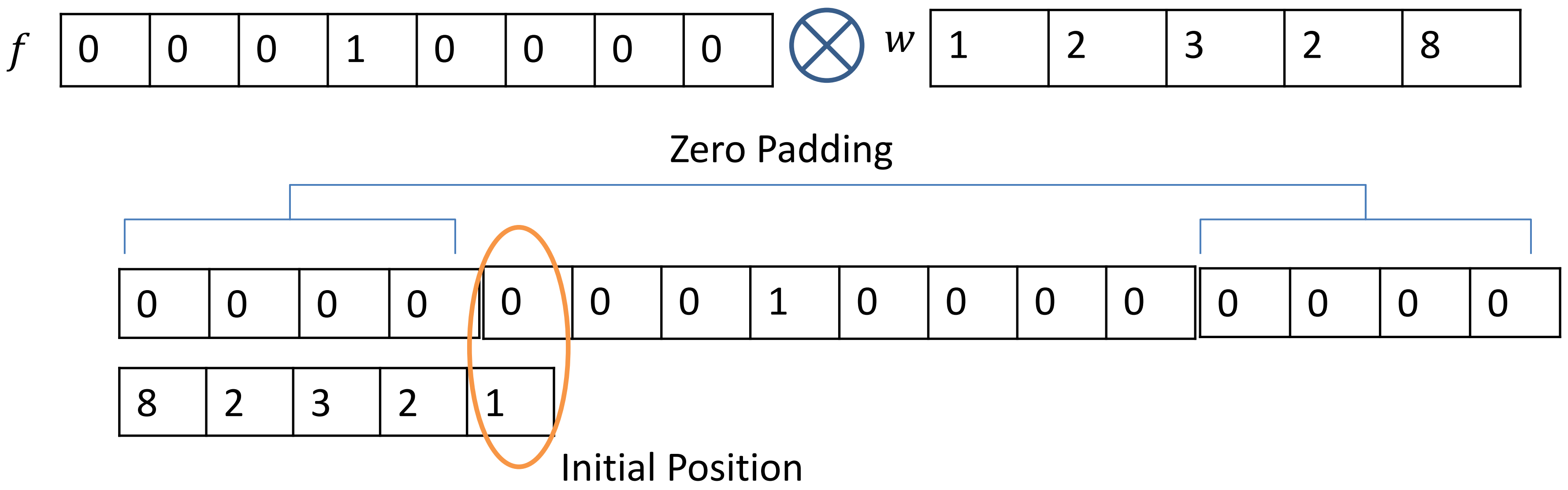
Linear Image Filtering (Review)

Digital image → Linear operation using a filter (H) → Output/ Filtered image

Review

f $\begin{array}{|c|c|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array}$ $\otimes$ w $\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 2 & 8 \\ \hline \end{array}$

Review



f
0 0 0 1 0 0 0 0
$\otimes$
w
1 2 3 2 8
Zero Padding
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
8 2 3 2 1
Initial Position

Review

f

0	0	0	1	0	0	0	0

$\otimes$ w

1	2	3	2	8

Cropped Convolution result

0	1	2	3	2	8	0	0

Review

f

0	0	0	1	0	0	0	0

$\otimes$ w

1	2	3	2	8

Zero Padding

0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	2	3	2	1											

Initial Position

$$f(t) \otimes w(t) = \sum_{\tau=-2}^{2} w(\tau) f(t-\tau)$$

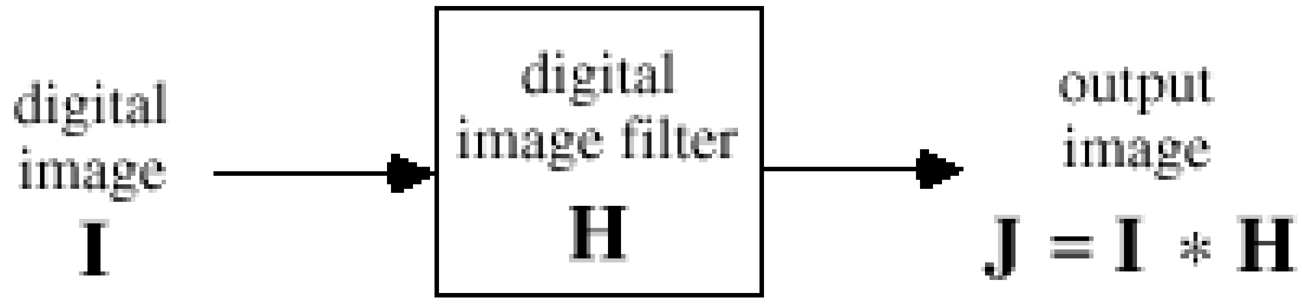
Spatial Convolution Operator

The convolution of a filter $w(x,y)$ of size $m \times n$ with an image $f(x,y)$, denoted as $w(x,y) \otimes f(x,y)$

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

Linear Systems And Linear Image Filtering (Review)

- A process that accepts a signal or image **I** as input and transforms it by an act of linear convolution is a type of **linear system**
- **Example**



Some Specific Goals

- **smoothing** - remove noise from bit errors, transmission, etc
- **deblurring** - increase **sharpness** of blurred images
- **sharpening** - emphasize significant features, such as **edges**
- **combinations** of these

Review: Two Smoothing Averaging Filter Masks

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

Review: Laplace Operator

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1

0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

a	b
c	d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Demo Filters

Review: Sharpening Spatial Filters: Laplace Operator

Image sharpening in the way of using the Laplacian:

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

where,

$f(x, y)$ is input image,

$g(x, y)$ is sharpenend images,

$c =$ -1 if $\nabla^2 f(x, y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.

Discrete Fourier Transform

Spatial Domain (x) $\longrightarrow$ Frequency Domain (u)

Fourier Transform

$$\boldsymbol{F(u) = \int_{-\infty}^{\infty} f(x) e^{-\sqrt{-1}ux} dx}$$

Discrete Fourier Transform

$$\boxed{F(u) = \sum_{x=-\infty}^{\infty} f(x) e^{-\sqrt{-\mathbf{1}}ux}}$$

$$e^{-\sqrt{-\mathbf{1}}x} = cosx - \sqrt{-\mathbf{1}}sinx$$

Frequency Domain (u) $\longrightarrow$ Spatial Domain (x)

Inverse Fourier Transform

$$\boldsymbol{f(x) = \int_{-\infty}^{\infty} F(u) e^{\sqrt{-1}ux} du}$$

Inverse Discrete Fourier Transform

$$\boxed{f(x) = \sum_{u=-\infty}^{\infty} F(u) e^{\sqrt{-\mathbf{1}}ux}}$$

$$e^{\sqrt{-\mathbf{1}}x} = cosx + \sqrt{-\mathbf{1}}sinx$$

Convolution Theorem

- Let f be an image and h a filtering window
- Lets us consider the convolution

$$f(t) \otimes h(t)$$

Convolution Theorem

- Let f and h be two function
- Lets us consider the convolution

$$f(t) \otimes h(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)$$

From the definition of convolution

$\otimes$

Convolution Theorem

- Let f and h be two function
- Lets us consider the convolution

$$f(t) \otimes h(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)$$

$$F[f(t) \otimes h(t)] = \sum_{t=-\infty}^{\infty} [\sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)]e^{-\sqrt{-1}\mu t}$$

Computing the Fourier transform of the convolution

$$F(u) = \sum_{x=-\infty}^{\infty} f(x)e^{-\sqrt{-1}ux}$$

Convolution Theorem

- Let f and h be two function
- Lets us consider the convolution

$$f(t) \otimes h(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)h\,(t-\tau)$$

$$F[f(t) \otimes h(t)] = \sum_{t=-\infty}^{\infty} [\sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)]e^{-\sqrt{-1}\mu t}$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[\sum_{t=-\infty}^{\infty} h(t-\tau)\, e^{-\sqrt{-1}\mu t}]$$

Convolution Theorem

- Let f and h be two function
- Lets us consider the convolution

$$f(t) \otimes h(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)$$

$$F[f(t) \otimes h(t)] = \sum_{t=-\infty}^{\infty} [\sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)]e^{-\sqrt{-1}\mu t}$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[\sum_{t=-\infty}^{\infty} h(t-\tau)\, e^{-\sqrt{-1}\mu t}]$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[\sum_{t=-\infty}^{\infty} h(t-\tau)\, e^{-\sqrt{-1}\mu(t-\tau)}]\, e^{-\sqrt{-1}\mu\tau}$$

Convolution Theorem

- Let f and h be two function
- Lets us consider the convolution

$$f(t) \otimes h(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)h\,(t-\tau)$$

$$F[f(t) \otimes h(t)] = \sum_{t=-\infty}^{\infty} [\sum_{\tau=-\infty}^{\infty} f(\tau)h(t-\tau)]e^{-\sqrt{-1}\mu t}$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[\sum_{t=-\infty}^{\infty} h(t-\tau)\, e^{-\sqrt{-1}\mu t}]$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[\sum_{t=-\infty}^{\infty} h(t-\tau)\, e^{-\sqrt{-1}\mu(t-\tau)}]\, e^{-\sqrt{-1}\mu\tau}$$

$$= \sum_{\tau=-\infty}^{\infty} f(\tau)[H(\mu)]\, e^{-\sqrt{-1}\mu(\tau)} = H(\mu) \sum_{\tau=-\infty}^{\infty} f(\tau)\, e^{-\sqrt{-1}\mu(\tau)}$$

$$= H(\mu)F(\mu)$$

Convolution Theorem

- Fourier transform pairs

$$f(t) \otimes h(t) \Leftrightarrow H(\mu)F(\mu)$$

$$f(t)h(t) \Leftrightarrow H(\mu) \otimes F(\mu)$$

The Basic Filtering in the Frequency Domain

- Modifying the Fourier transform of an image
- Computing the inverse transform to obtain the processed result

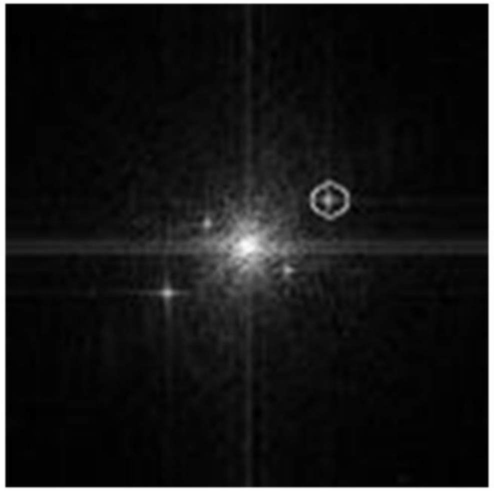
$$g(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$

$F(u,v)$ is the DFT of the input image

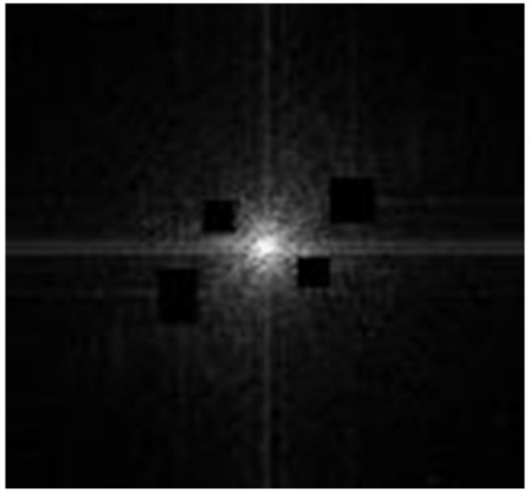
$H(u,v)$ is a filter function.

Example: Periodic Noise removal

Fourier transform

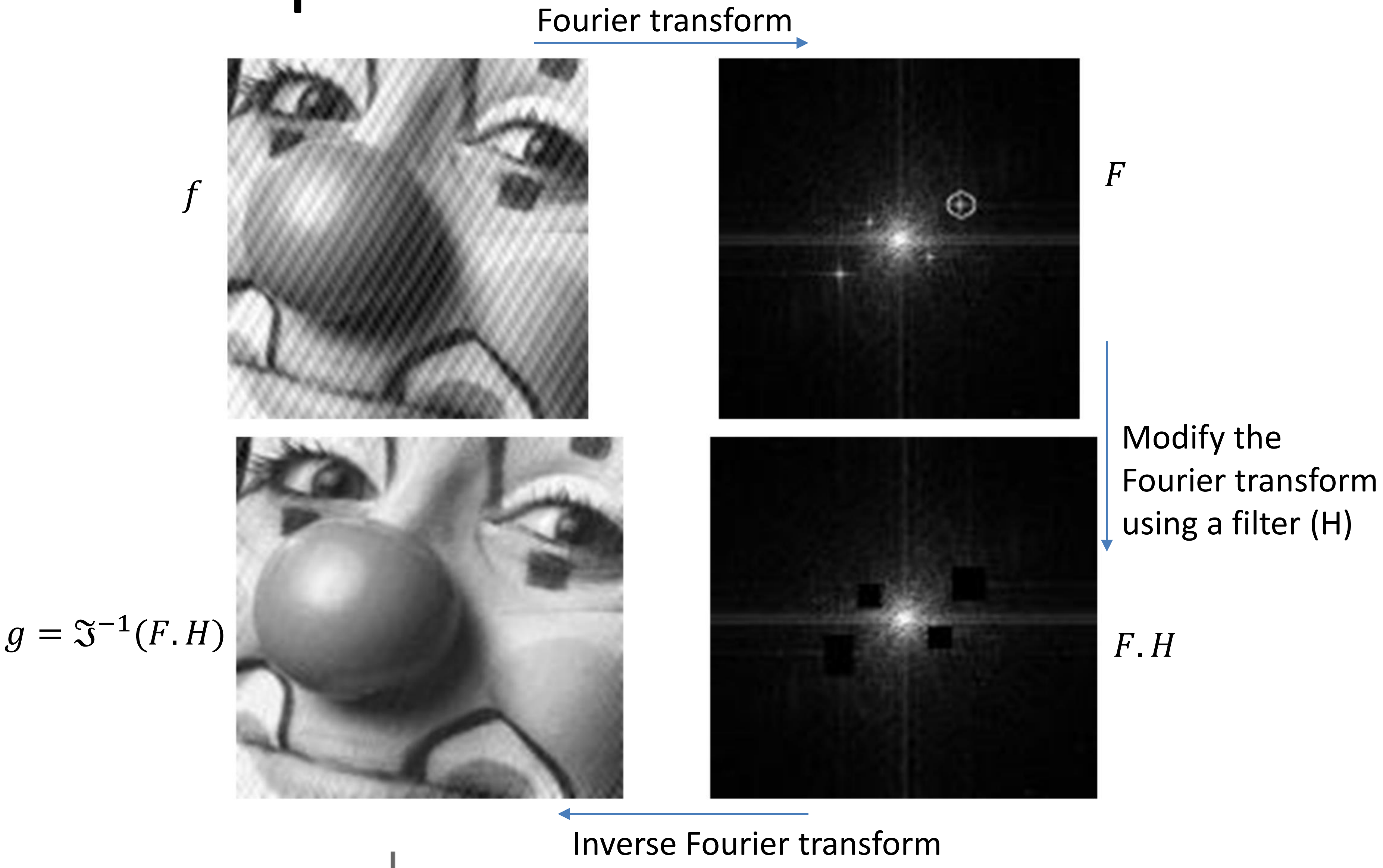


Modify the Fourier transform using a filter



Inverse Fourier transform

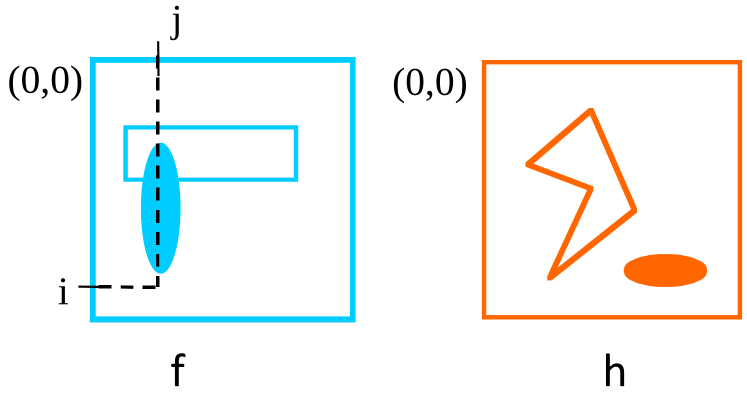
Example: Periodic Noise removal



$$
\begin{aligned}
g &= \mathfrak{I}^{-1}(F.H) \\
&\equiv \mathfrak{I}^{-1}(F) \otimes \mathfrak{I}^{-1}(H) \\
&= f \otimes h
\end{aligned}
$$

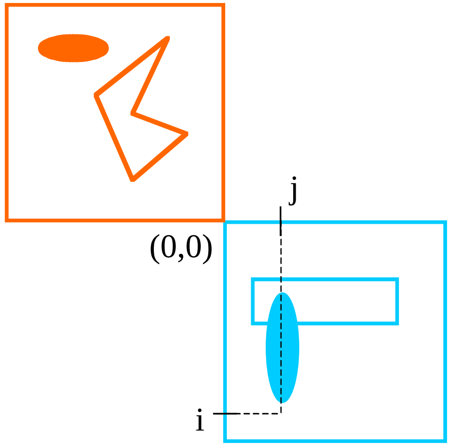
Diagrams of Convolution

- Consider the two images with image **f and h** and its contents **shaded** at each stage of processing shown:

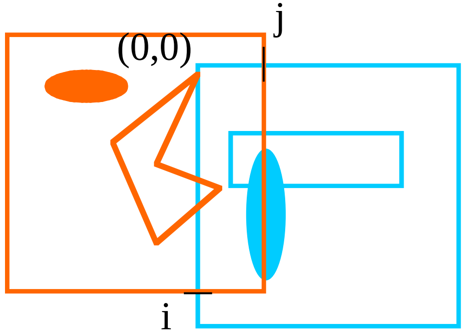


Diagrams of Convolution

- The image **h** is then reversed (reflected), along both axes.



- The reversed version of h is then shifted by the amount (i, j) along both axes:



$\mathbf{I}_2$ shifted

$$
\begin{aligned}
g &= \mathfrak{I}^{-1}(F.H) \\
&\equiv \mathfrak{I}^{-1}(F) \otimes \mathfrak{I}^{-1}(H) \\
&= f \otimes h
\end{aligned}
$$

Review: Periodic Extension of Image

- The IDFT equation $$I(i, j) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-(ui+vj)}$$
  implies the **periodic extension of the image I** as well (with period N), simply by letting the arguments (i, j) take any integer value.
- Note that for any integers n, m

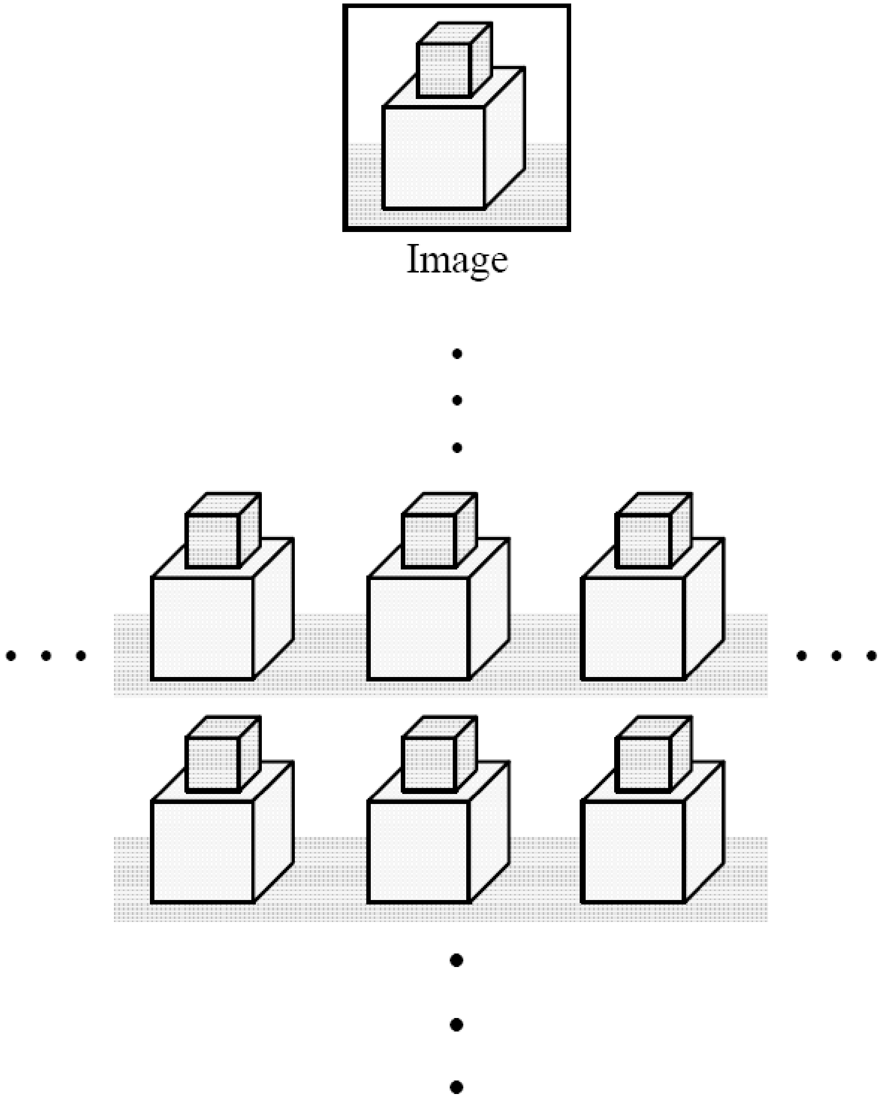
$$\begin{aligned} I(i+nN, j+mN) &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-[u(i+nN)+v(j+mN)]} \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-(ui+vj)}\; W_N^{-N(nu+mv)} \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-(ui+vj)} = I(i, j) \end{aligned}$$

since

$$W_N^{-N(nu+mv)} = e^{-\sqrt{-1}\,\frac{2\pi}{N}\cdot N(nu+mv)} = e^{-2\pi\sqrt{-1}\,(nu+mv)} = 1^{(nu+mv)} = 1$$

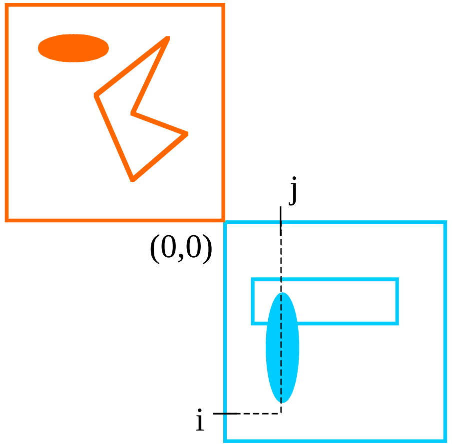
- In a sense, the DFT **implies** that the image **I** is already periodic.
- This will be extremely important when we consider **convolution**

Review: Periodic Extension of Image

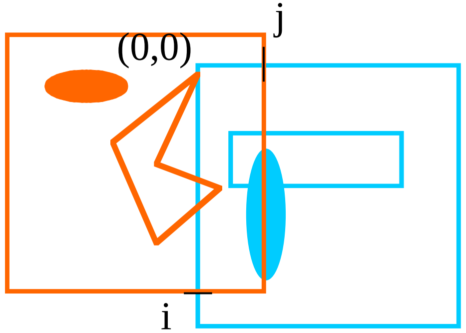


Diagrams of Convolution

- The image h is then reversed (reflected), along both axes. This requires that it be defined for negative coordinates, i.e., the periodic extension is used.



- The reversed version of **h** is then shifted by the amount (i, j) along both axes:



$\mathbf{I}_2$ shifted

Diagrams of Convolution

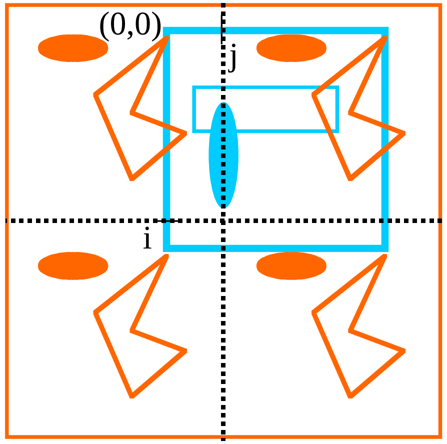
- The sum extends over $0 \le m \le M-1$, $0 \le n \le N-1$ (in blue) so some of the arguments of $h(i-m, j-n)$ fall outside the range $0, \ldots, N-1$. What is computed is the summation of the product of

$$[f(m, n) ; 0 \le m, n \le N-1]$$

and the periodic extension of

$$[h(i-m, j-n)]$$

as shown:



Wraparound Convolution

- Wraparound convolution is a consequence of the **periodic DFT**.
- **Wraparound convolution** is an **artifact** of digital processing.

Linear Convolution by Zero Padding

- Performing linear convolution by wraparound convolution is a conceptually simple matter.
- It is accomplished by **padding** the two image arrays with **zero values**.
- **Generally**, both image arrays must be doubled in size:

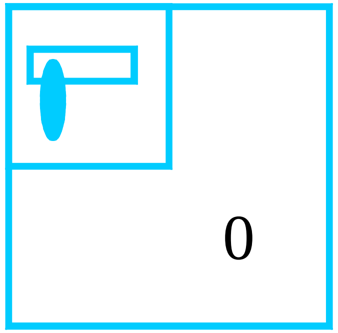


Image $\mathbf{I}_1$
(zero padded)

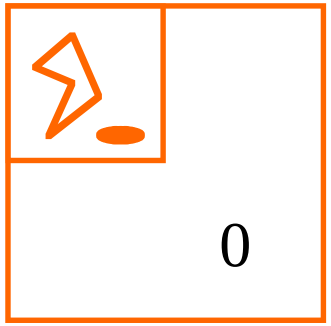
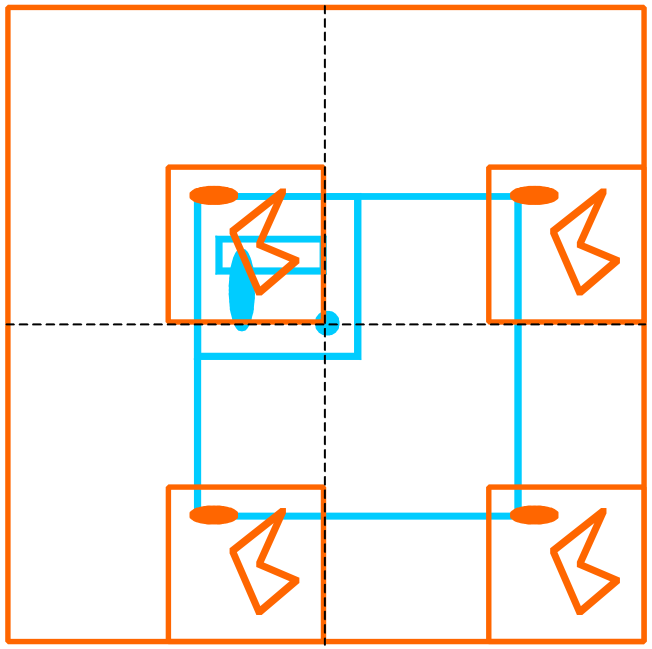


Image $\mathbf{I}_2$
(zero padded)

- At the edges, no wraparound effect will occur, since the "moving" image will be weighted by zero values only outside the domain.
- This can be seen by looking at the overlaps when computing the convolution at a point (i, j):

Linear Convolution by Zero Padding



Linear convolution by zero padding

- Remember, the summations take place only within the **blue** shaded square ($0 \leq i, j, \leq 2N-1$).
- Instead of summing over the periodic extension of the "moving image," zero values are summed with the weighted interior values.

Summary: Steps for Filtering in the Frequency Domain

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

2. Form a padded image, $f_p(x,y)$ of size PxQ by appending the necessary number of zeros to f(x,y)

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

4. Compute the DFT, F(u,v) of the image from step 3

5. Generate a real, symmetric filter function*, H(u,v), of size PxQ with center at coordinates (P/2, Q/2)

*generate from a given spatial filter, we pad the spatial filter, multiply the expadded array by $(-1)^{x+y}$, and compute the DFT of the result to obtain a centered H(u,v).

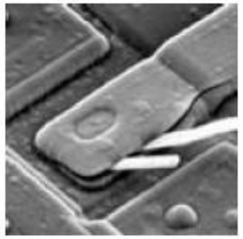
Summary:
Steps for Filtering in the Frequency Domain

6. Form the product G(u,v) = H(u,v)F(u,v) using array multiplication

7. Obtain the processed image

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

8. Obtain the final processed result, g(x,y), by extracting the MxN region from the top, left quadrant of $g_p(x,y)$



a	b	c
d	e	f
g	h	

FIGURE 4.36
(a) An $M \times N$ image, f.
(b) Padded image, f_p of size $P \times Q$.
(c) Result of multiplying f_p by $(-1)^{x+y}$.
(d) Spectrum of F_p. (e) Centered Gaussian lowpass filter, H, of size $P \times Q$.
(f) Spectrum of the product HF_p.
(g) g_p, the product of $(-1)^{x+y}$ and the real part of the IDFT of HF_p.
(h) Final result, g, obtained by cropping the first M rows and N columns of g_p.

Next

- We will design filter to perform smoothing, sharpening in frequency domain.