Digital Image Processing

COSC 6380/4393

Midterm Review

Mar 28th, 2023

Image Formation

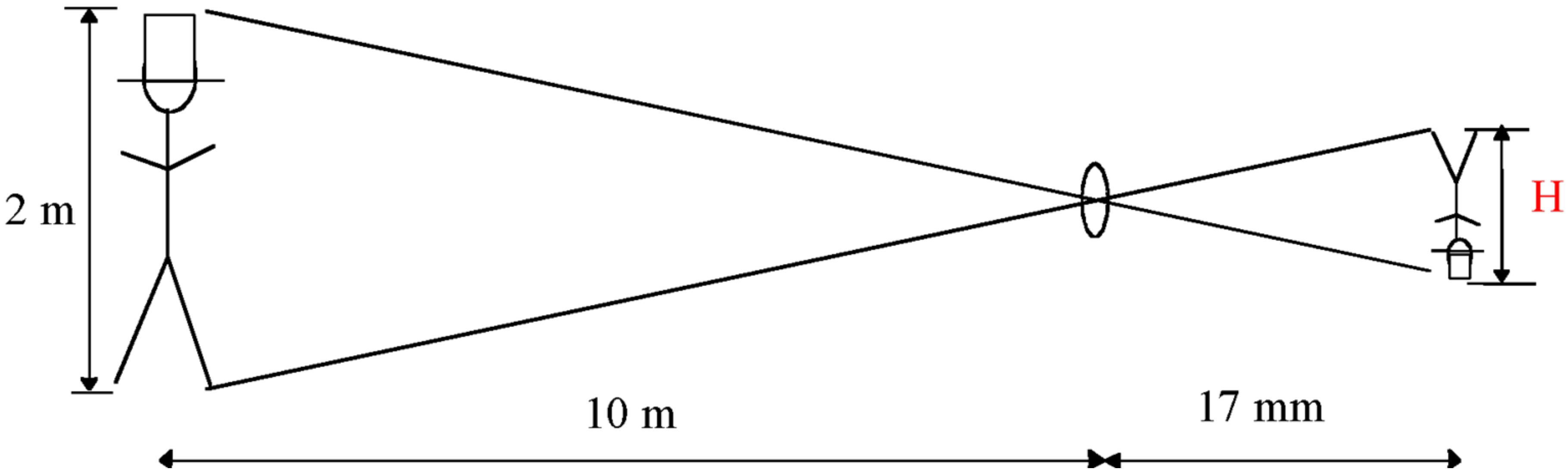
Image Formation: (Projection)

1. A person is standing M meters in front of a camera
   - h is the height of the person
   - X is the focal length of the camera
   - H is the height of the projection of the person on the imaging plane. What is H? (Show steps)

2. A sphere/square is placed at a distance of M meters (m) in front of a camera
   - a/v is the area/volume of the object
   - X is the focal length of the camera
   - A/V is the area of the projection of the object on the imaging plane. What is A/V? (Show steps)

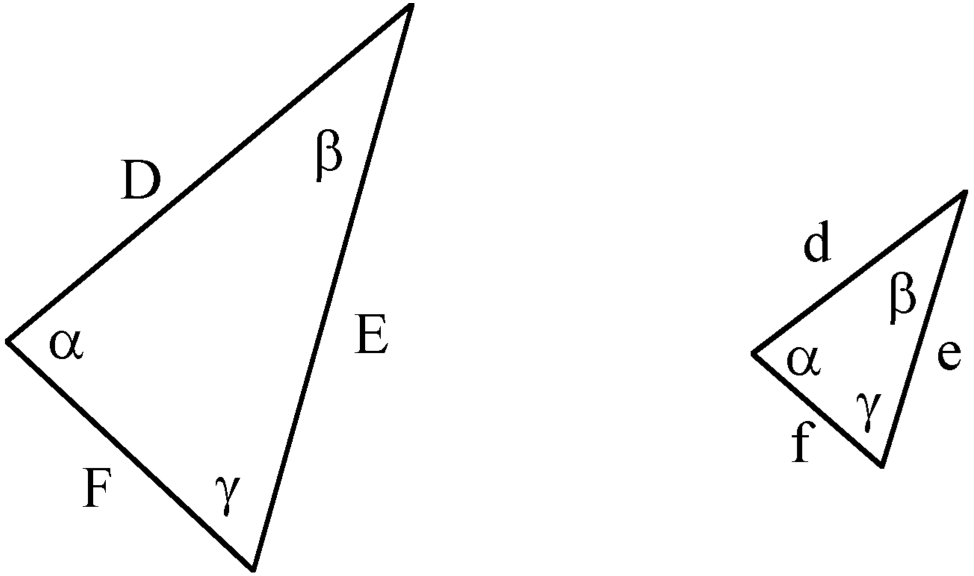
Example: Solution

- There is a man standing 10 meters (m) in front of you
- He is 2 m tall
- The focal length of your eye is about 17 mm
- Question: What is the height H of his image on your retina?



SIMILAR TRIANGLES

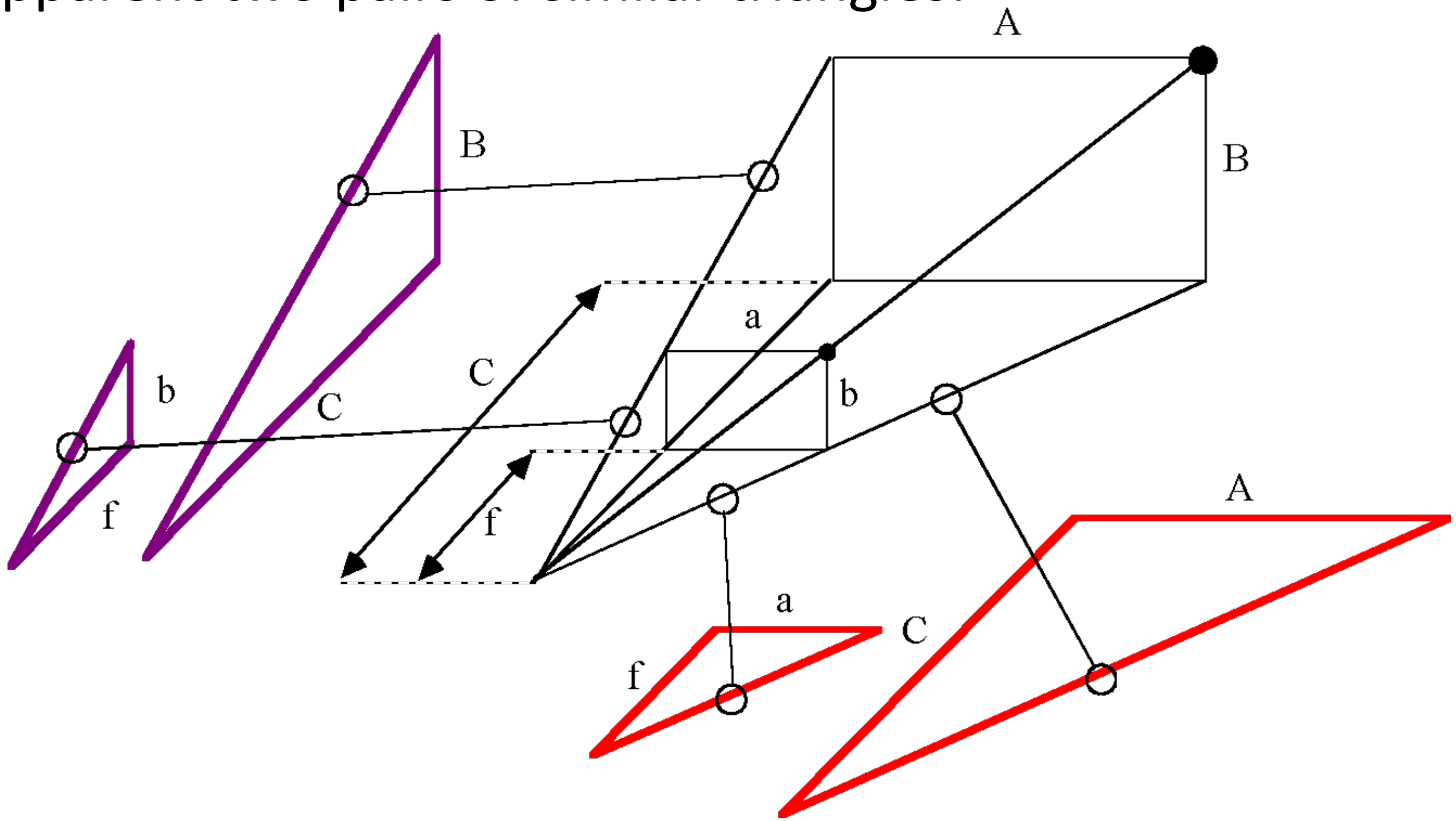
- Similar Triangles Theorem - Similar triangles have their side lengths in the same proportions.



$$\frac{D}{E} = \frac{d}{e} \qquad \frac{E}{F} = \frac{e}{f} \qquad \frac{F}{D} = \frac{f}{d}$$

SOLVING PERSPECTIVE PROJECTION

- Using similar triangles we can solve for the relationship between 3-D coordinates in space and 2-D image coordinates
- Redraw the imaging geometry once more, this time making apparent two pairs of similar triangles:



SOLVING PERSPECTIVE PROJECTION

- By the Similar Triangles Theorem, we conclude that

$$\frac{a}{f} = \frac{A}{C} \quad \text{and} \quad \frac{b}{f} = \frac{B}{C}$$

OR

$$(a, b) = \frac{f}{C} \cdot (A, B) = (fA/C, fB/C)$$

PERSPECTIVE PROJECTION EQUATION

- Thus the following relationship holds between 3-D space coordinates (X, Y, Z) and 2-D image coordinates (x, y) :

$$(x, y) = \frac{f}{Z} \cdot (X, Y)$$

where f = focal length.

- The ratio f/Z is the magnification factor, which varies with the range Z from the lens center to the object plane.

ANSWER

- By similar triangles,

$$\frac{2\text{ m}}{10\text{ m}} = \frac{H}{17\text{ mm}}$$

H = 3.4 mm

Alternatives: Given a circle or square of known area, parallel to the imaging plane. How do you compute area of the projected image?

Resampling

Given the location of four pixels Q_{11}, Q_{12},Q_{21}, Q_{22} and their intensity values I_{11}, I_{12}, I_{21},I_{22}. Assuming that Q_{11}, Q_{12},Q_{21}, Q_{22} are the nearest pixels to P

Estimate the intensity value of pixel located at P using bilinear interpolation?

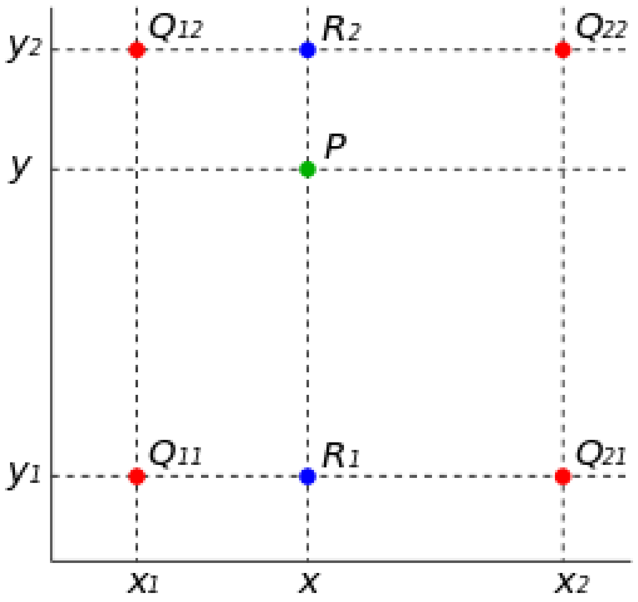
Binary Image Processing

Bi-Linear Interpolation(2D)

- $Q_{11} = (x_1, y_1)$,
$Q_{12} = (x_1, y_2)$,
$Q_{21} = (x_2, y_1)$,
and $Q_{22} = (x_2, y_2)$

$f(Q_i) \rightarrow intensity\ at\ Q_i$

Find the value at P



https://en.wikipedia.org/wiki/Bilinear_interpolation

Example: Linear Interpolation

Solve for I

$$I = \frac{I_1(x_2 - x)}{(x_2 - x_1)} + \frac{I_2(x - x_1)}{(x_2 - x_1)}$$

$$I = \frac{10(1 - 0.3)}{(1 - 0)} + \frac{15(0.3 - 0)}{(1 - 0)}$$

$$I = 7 + 4.5 = 11.5$$

Example

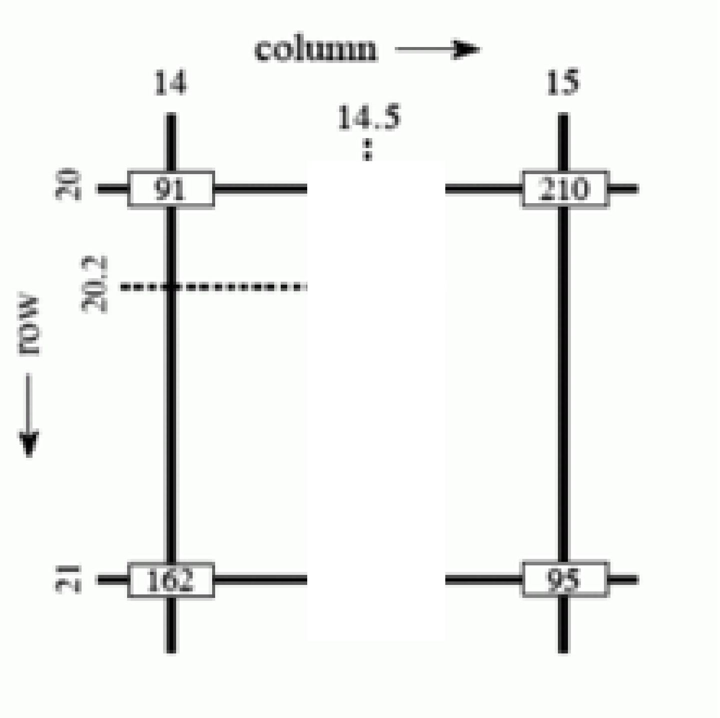
$I(21{,}14) = 162,$

$I(21{,}15) = 95,$

$I(20{,}14) = 91,$

$I(20{,}15) = 210$

$I(20.2, 14.5) = ?$

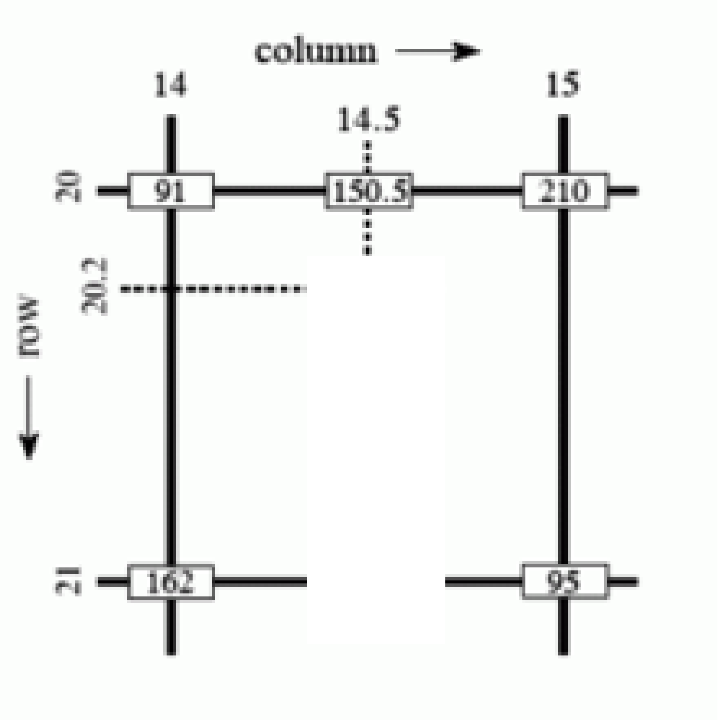


https://en.wikipedia.org/wiki/Bilinear_interpolation

Example

$$I(21,14) = 162,$$
$$I(21,15) = 95,$$
$$I(20,14) = 91,$$
$$I(20,15) = 210$$
$$I(20.2, 14.5) = ?$$

$$I_{20,14.5} = \frac{15-14.5}{15-14} \cdot 91 + \frac{14.5-14}{15-14} \cdot 210 = 150.5,$$



https://en.wikipedia.org/wiki/Bilinear_interpolation

Example

$I(21,14) = 162,$

$I(21,15) = 95,$

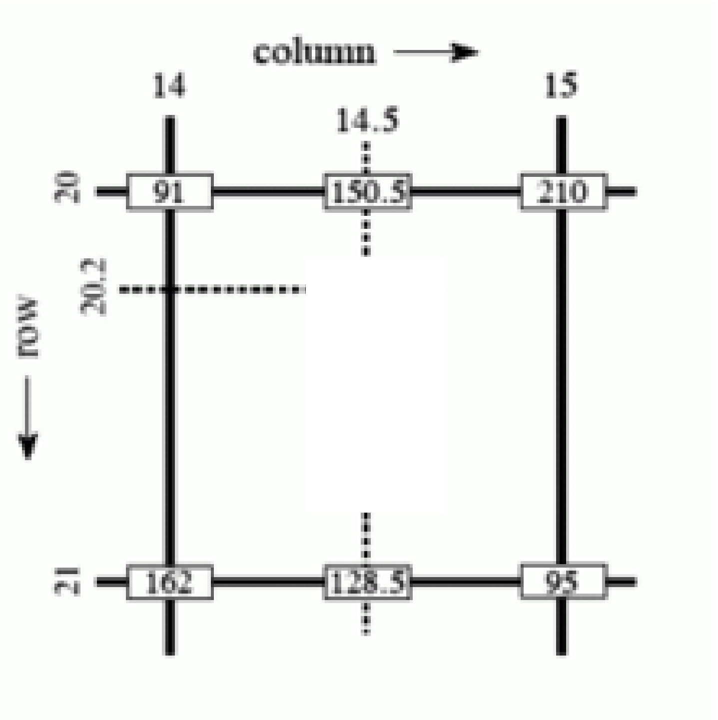
$I(20,14) = 91,$

$I(20,15) = 210$

$I(20.2, 14.5) = ?$

$$I_{20,14.5} = \frac{15-14.5}{15-14} \cdot 91 + \frac{14.5-14}{15-14} \cdot 210 = 150.5,$$

$$I_{21,14.5} = \frac{15-14.5}{15-14} \cdot 162 + \frac{14.5-14}{15-14} \cdot 95 = 128.5,$$



https://en.wikipedia.org/wiki/Bilinear_interpolation

Example

$I(21,14) = 162,$

$I(21,15) = 95,$

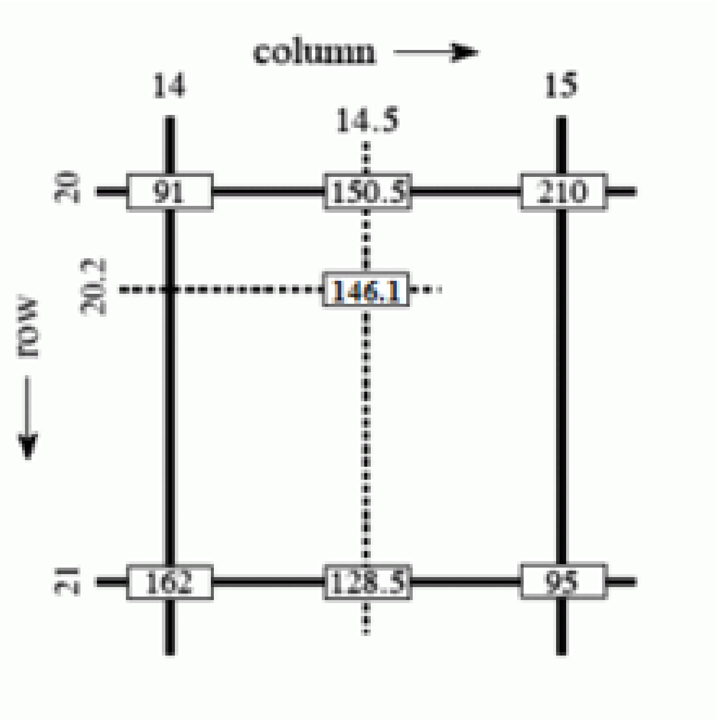
$I(20,14) = 91,$

$I(20,15) = 210$

$I(20.2, 14.5) = ?$

$$I_{20,14.5} = \frac{15-14.5}{15-14} \cdot 91 + \frac{14.5-14}{15-14} \cdot 210 = 150.5,$$

$$I_{21,14.5} = \frac{15-14.5}{15-14} \cdot 162 + \frac{14.5-14}{15-14} \cdot 95 = 128.5,$$



$$I_{20.2,14.5} = \frac{21-20.2}{21-20} \cdot 150.5 + \frac{20.2-20}{21-20} \cdot 128.5 = 146.1.$$

https://en.wikipedia.org/wiki/Bilinear_interpolation

Binary Image Logical Operations

- Given an acquired binary images I and a model binary Image M below. Generate a third binary image D representing the unmatched pixels in the acquired image compared to the model image.
- Since binary operations are quicker, you are allowed to use only binary operators (And, OR and NOT) or a combination of these on binary image M and I to accomplish this task.

0	1
1	0

M

0	1
0	1

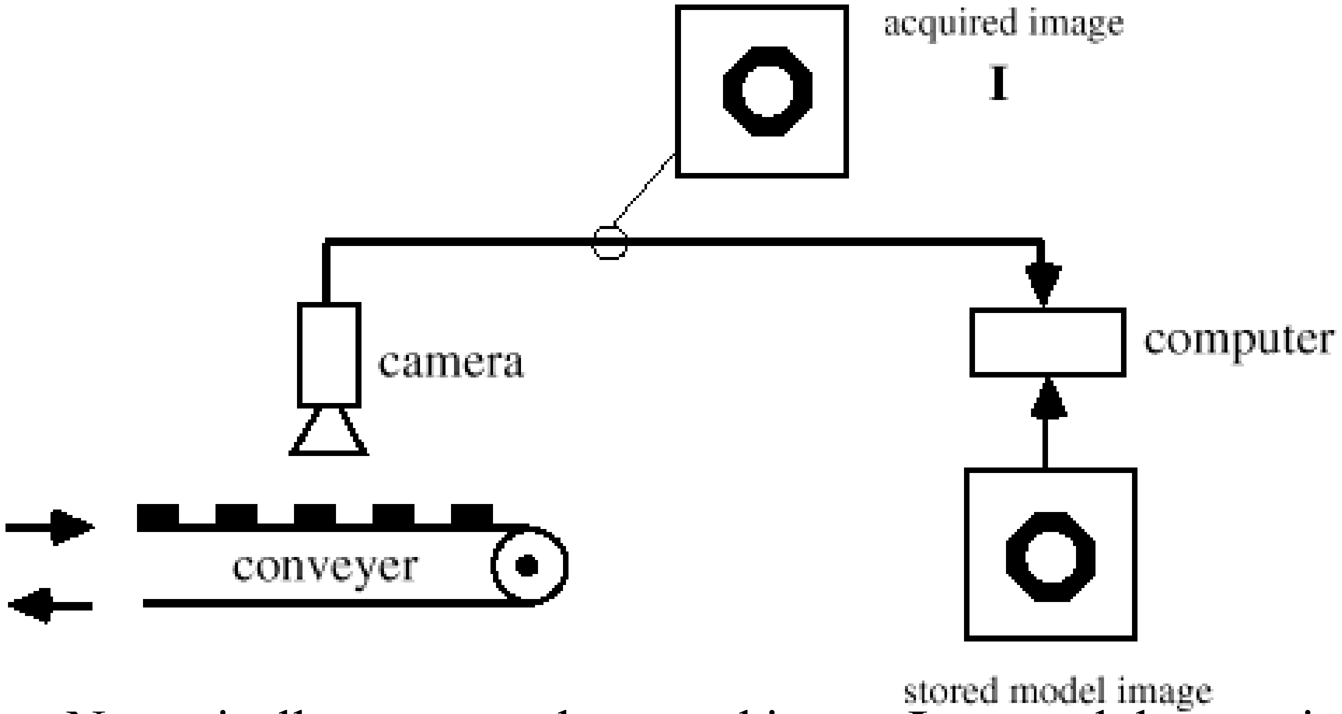
I

0	0
1	1

D

3. Image difference

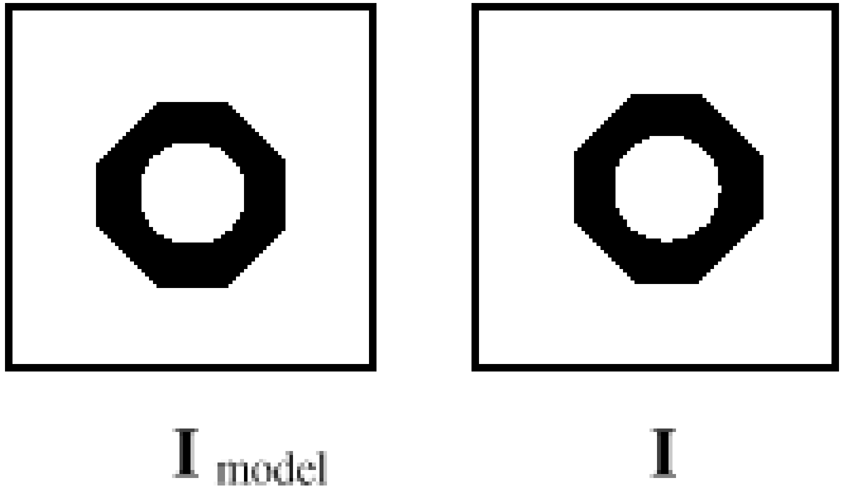
- An assembly-line image inspection system. Similar to many marketed by industry:



- **Objective**: Numerically compare the stored image $\mathbf{I}_{model}$ and the acquired image $\mathbf{I}$

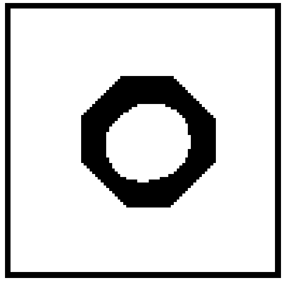
EXAMPLE

- Observe that the object in **I** has been shifted very slightly



Logical AND

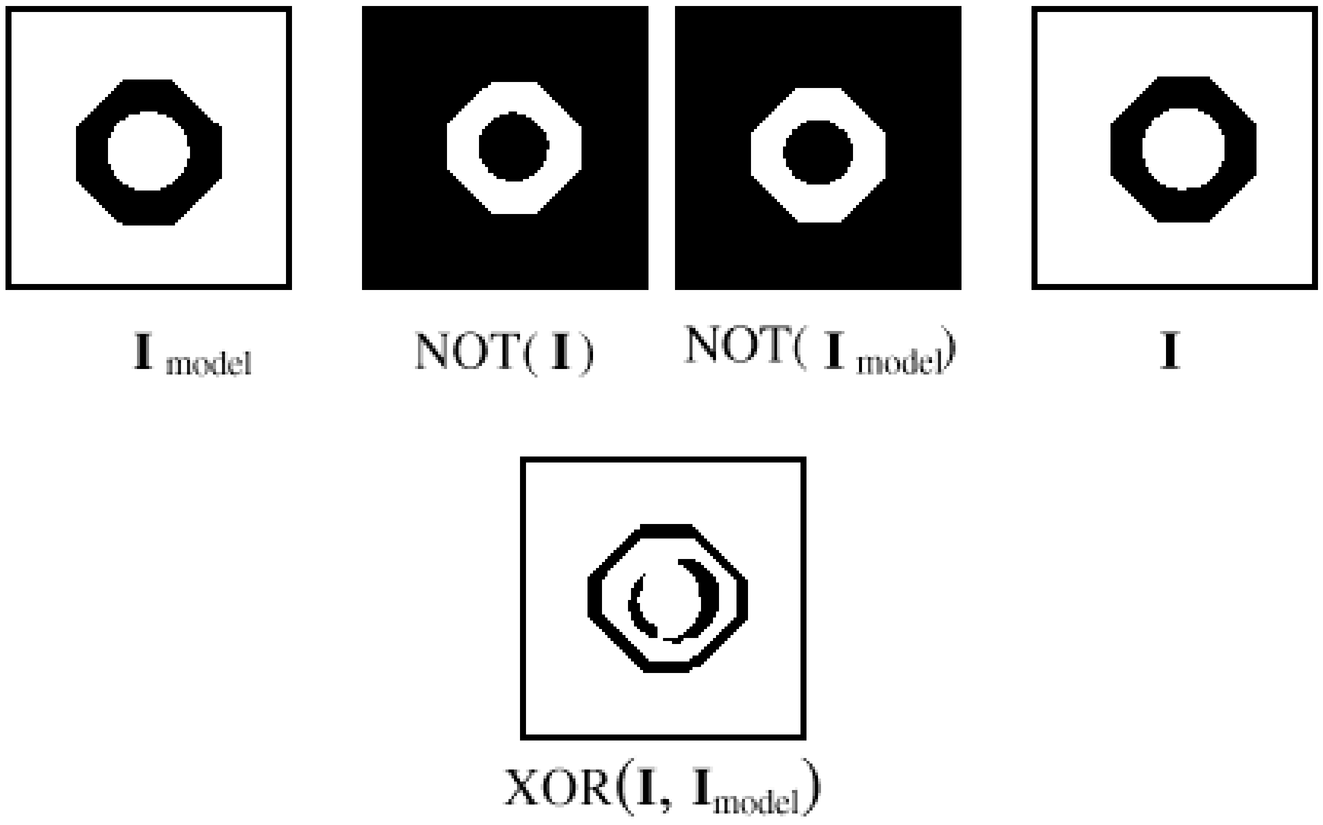
- The logical AND conveys the **overlap**



$\mathbf{I}_{model} \wedge \mathbf{I}$

- A measurement of the **displacement** is given by:
- XOR($\mathbf{I}$, $\mathbf{I}_{model}$) = OR{ AND[$\mathbf{I}_{model}$, NOT($\mathbf{I}$)], AND[NOT($\mathbf{I}_{model}$), $\mathbf{I}$]}

DISPLACEMENT



4. Morphological Operations

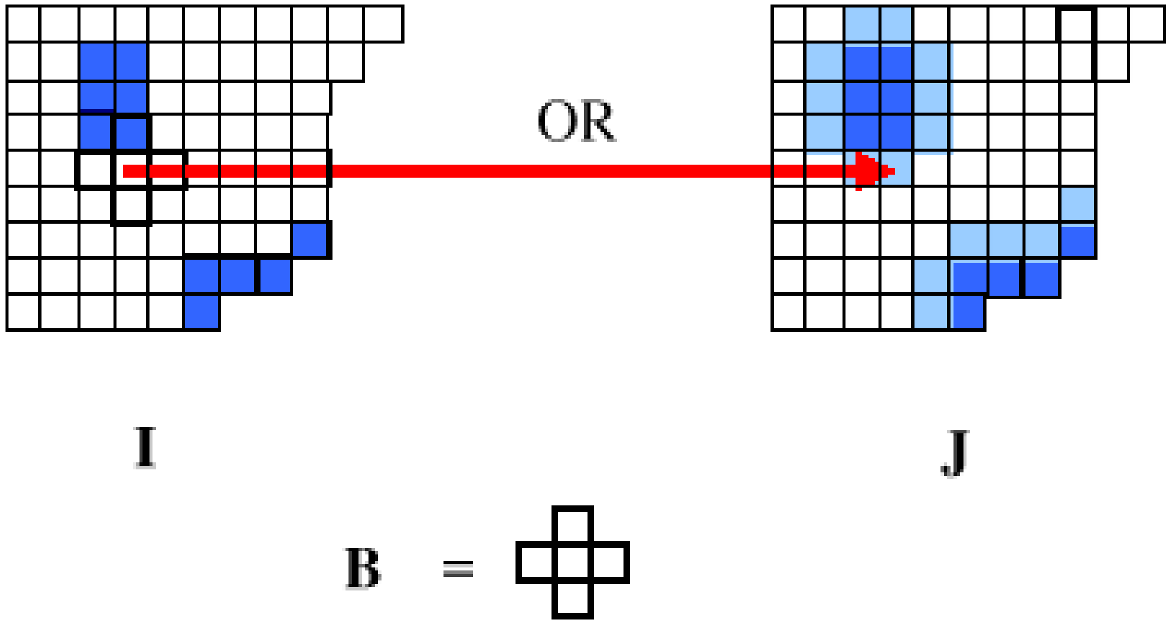
1. Perform a morphological operation using a window of your choice to remove the small object represented by 1 at pixel (2,4) in the image below.

0	0	0	0	0
1	1	0	1	0
1	1	0	0	0
0	1	0	0	0
0	0	0	0	0

Operations to cover holes, remove peninsula, etc.?

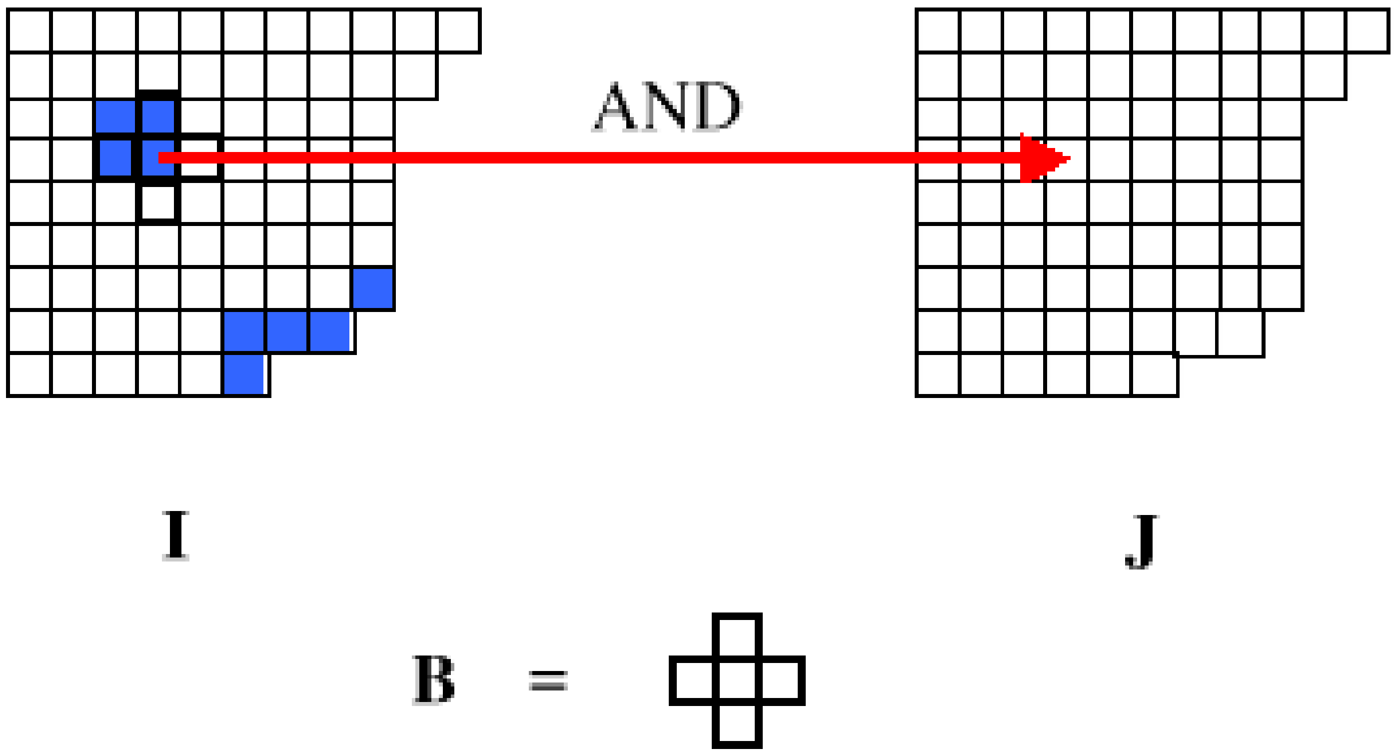
DILATION

- So-called because this operation **increases** the size of BLACK objects in a binary image
- Local Computation: **J** = DILATE(**I**, **B**)



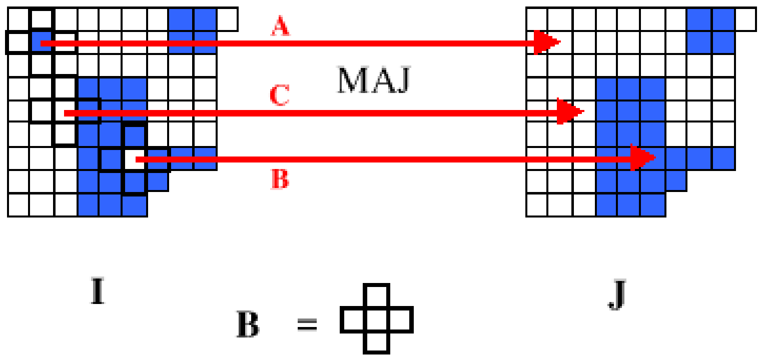
EROSION

- So-called because this operation **decreases** the size of BLACK objects in a binary image
- Local Computation: **J** = EROD(**I**, **B**)



MEDIAN

- Actually **majority**. A special case of the gray-level **median filter**
- Possesses qualitative attributes of both dilation and erosion, but does not generally change the **size** of objects or background
- Local Computation: **J** = MEDIAN(**I**, **B**)



- The median removed the small **object** A and the small **hole** B, but did not change the boundary (**size**) of the larger region C

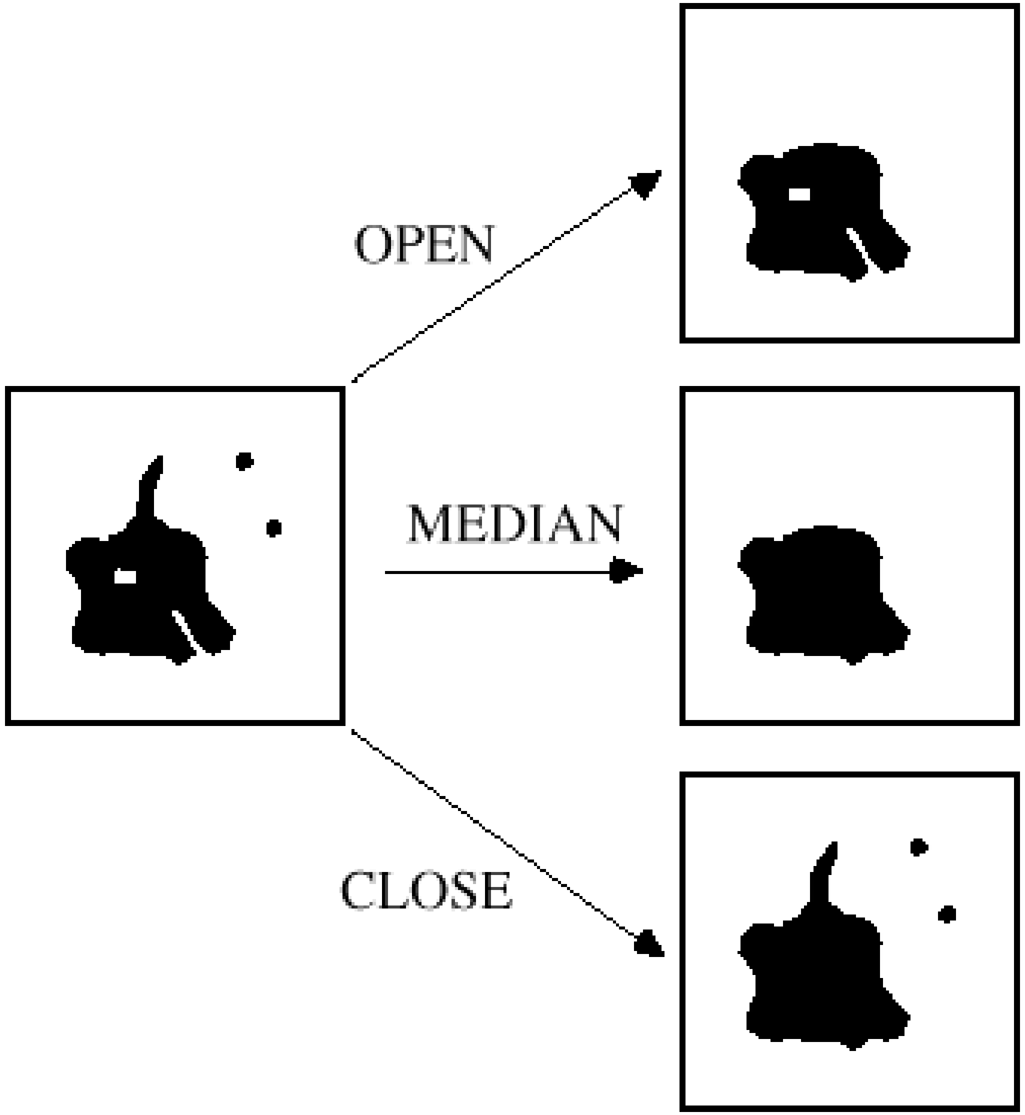
OPENing and CLOSing

- We can define **new** morphological operations by performing the basic ones in sequence
- Given an image **I** and window **B**, define

  OPEN(**I**, **B**) = DILATE [ERODE(**I**, **B**), **B**]

  CLOSE(**I**, **B**) = ERODE [DILATE(**I**, **B**), **B**]
- In other words,
- OPEN = erosion (by **B**) followed by dilation (by **B**)
- CLOSE = dilation (by **B**) followed by erosion (by **B**)

EXAMPLES



4. Morphological Operations

1. Perform a morphological operation using a window of your choice to remove the small object represented by 1 at pixel (2,4) in the image below.

0	0	0	0	0
1	1	0	1	0
1	1	0	0	0
0	1	0	0	0
0	0	0	0	0

1. Operations to cover holes, remove peninsula?

Solution

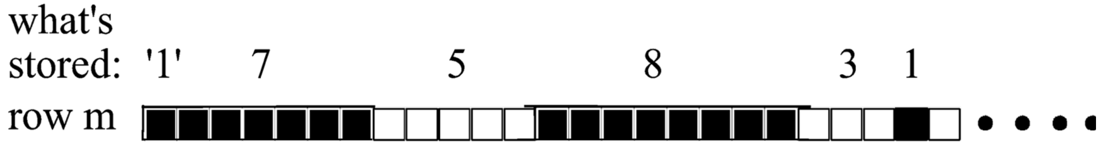
- A possible solution can be using the median operation with a filter (3,1)

0	0	0	0	0
1	1	0	1	0
1	1	0	0	0
0	1	0	0	0
0	0	0	0	0

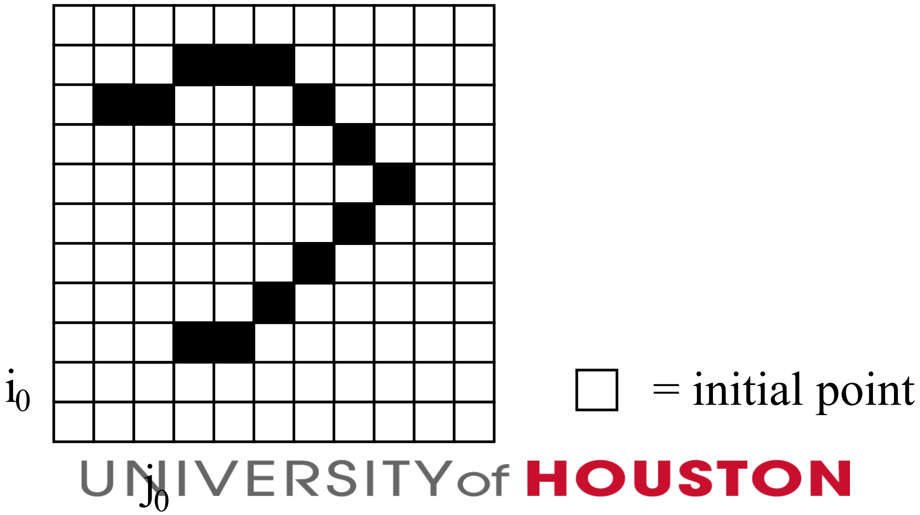
0	0	0	0	0
1	1	0	0	0
1	1	0	0	0
0	1	0	0	0
0	0	0	0	0

5. Compression

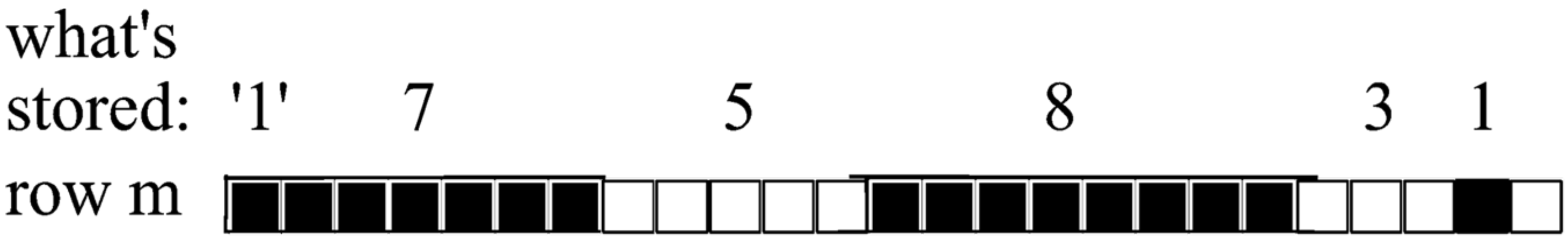
1. Perform compression to represent sequence of pixels in the binary image below



1. Perform compression to represent the contour in the binary image using chain codes



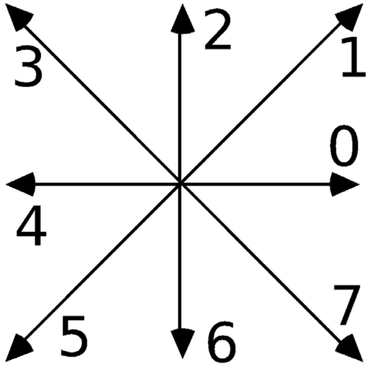
Run Length Encoding



Code: 175831

CHAIN CODE

- We use the following 8-neighbor direction codes:

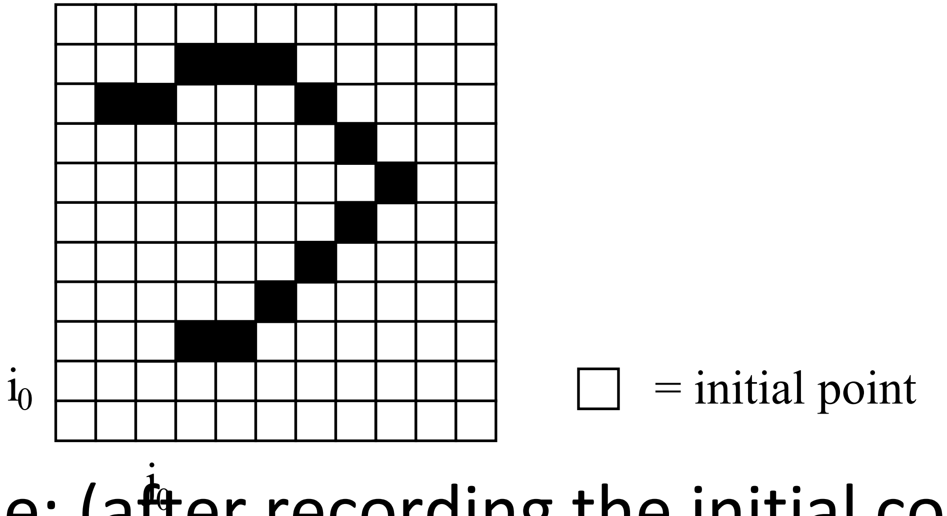


- Since the numbers 0, 1, 2, 3, 4, 5, 6, 7 can be coded by their 3-bit binary equivalents:

  000, 001, 010, 011, 100, 101, 110, 111

  the location of each point on the contour **after** the initial point can be coded by 3 bits.

EXAMPLE



- Its chain code: (after recording the initial coordinate (i0, j0)

1, 0, 1, 1, 1, 1, 3, 3, 3, 4, 4, 5, 4

=

001, 000, 001, 001, 001, 001, 011, 011, 011, 100, 100, 101, 100

Point Operations

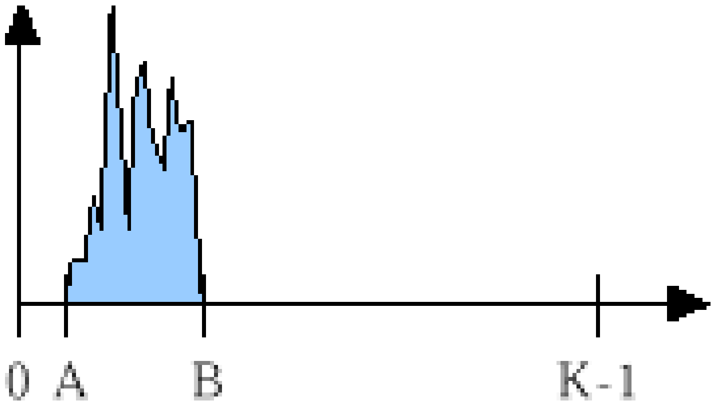
Linear Point Operations

1. Perform a full contrast stretch on the image below assuming that each dynamic range of the image is 0-15
2. Perform histogram flattening, shaping?

$$\mathbf{I} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array}$$

6. Full-Scale Contrast Stretch

- The **most common** linear point operation. Suppose **I** has a compressed histogram:



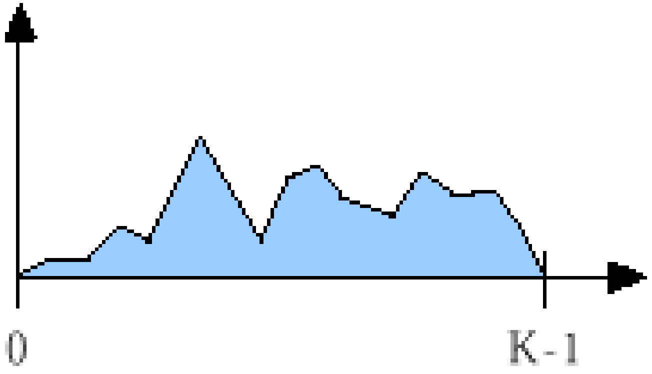
- Let A and B be the min and max gray levels in **I**
- Define

$$J(i, j) = P \cdot I(i, j) + L$$

- such that $P \cdot A + L = 0$ and $P \cdot B + L = (K-1)$

Full-Scale Contrast Stretch

- The result of solving these **2 equations in 2 unknowns** (P, L) is an image **J** with a full-range histogram:



- The solution to the above equations is

$$P = \left(\frac{K-1}{B-A}\right) \quad \text{and} \quad L = -A\left(\frac{K-1}{B-A}\right)$$

or

$$J(i, j) = \left(\frac{K-1}{B-A}\right)\left[I(i, j) - A\right]$$

$$\mathbf{I} = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array}$$

- $B = 11$
- $A = 1$
- $B - A = 10$
- $(K - 1) = 15$

- If I_c is the image with full dynamic range.
- $I_c[0,0] = INT(\frac{15}{10}(I[0,0] - 1) + 0.5)$

$$= INT\left(\frac{15}{10}(1 - 1) + 0.5\right) = 0$$

- $I_c[0,2] = INT(\frac{15}{10}(I[0,2] - 1) + 0.5)$

$$= INT\left(\frac{15}{10}(3 - 1) + 0.5\right) = 3$$

…

- $B = 11$
- $A = 1$
- $B - A = 10$
- $(K - 1) = 15$

$$\mathbf{I} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array}$$

- If I_c is the image with full dynamic range.
- $I_c[0,0] = INT(\frac{15}{10}(I[0,0] - 1) + 0.5)$

$$= INT\left(\frac{15}{10}(1 - 1) + 0.5\right) = 0$$

- $I_c[0,2] = INT(\frac{15}{10}(I[0,2] - 1) + 0.5)$

$$= INT\left(\frac{15}{10}(3 - 1) + 0.5\right) = 3$$

…

Result:

0	0	3	5
2	6	3	2
11	0	11	2
5	6	3	15

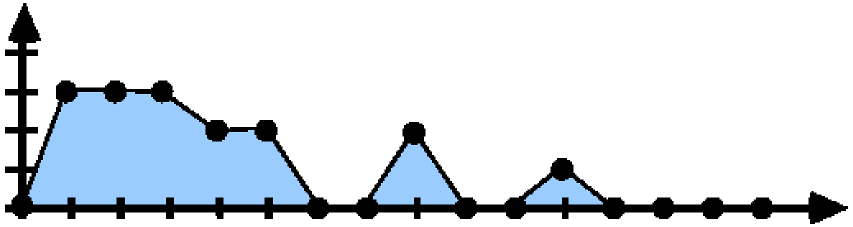
Histogram Flattening

- Given a 4 x 4 image **I** with gray-level range {0, ..., 15} (K-1 = 15):

$$\mathbf{I} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array}$$

- It's histogram is

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	0	3	3	3	2	2	0	0	2	0	0	1	0	0	0	0



Histogram Flattening

- The normalized histogram is

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
p(k)	0	$\frac{3}{16}$	$\frac{3}{16}$	$\frac{3}{16}$	$\frac{2}{16}$	$\frac{2}{16}$	0	0	$\frac{2}{16}$	0	0	$\frac{1}{16}$	0	0	0	0

- From which we can compute the intermediate image **J1** and finally the "flattened" image **J**:

$$\mathbf{J}_1 = \begin{array}{|c|c|c|c|} \hline 3/16 & 3/16 & 9/16 & 11/16 \\ \hline 6/16 & 13/16 & 9/16 & 6/16 \\ \hline 15/16 & 3/16 & 15/16 & 6/16 \\ \hline 11/16 & 13/16 & 9/16 & 16/16 \\ \hline \end{array} \qquad \mathbf{J} = \begin{array}{|c|c|c|c|} \hline 3 & 3 & 8 & 10 \\ \hline 6 & 12 & 8 & 6 \\ \hline 14 & 3 & 14 & 6 \\ \hline 10 & 12 & 8 & 15 \\ \hline \end{array}$$

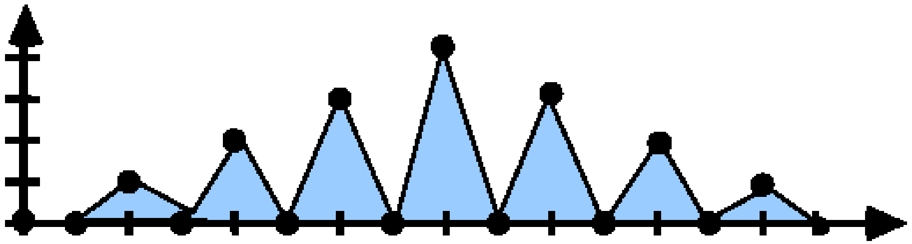
Histogram Shaping

- Consider the same image as in the last example. We had

$$\mathbf{I} = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array} \qquad \mathbf{J}_1 = \begin{array}{|c|c|c|c|}\hline 3/16 & 3/16 & 9/16 & 11/16 \\ \hline 6/16 & 13/16 & 9/16 & 6/16 \\ \hline 15/16 & 3/16 & 15/16 & 6/16 \\ \hline 11/16 & 13/16 & 9/16 & 16/16 \\ \hline \end{array}$$

- Fit this t

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$H_{\mathbf{J}}(k)$	0	0	1	0	2	0	3	0	4	0	3	0	2	0	1	0
$p_{\mathbf{J}}(k)$	0	0	$\frac{1}{16}$	0	$\frac{2}{16}$	0	$\frac{3}{16}$	0	$\frac{4}{16}$	0	$\frac{3}{16}$	0	$\frac{2}{16}$	0	$\frac{1}{16}$	0



Histogram Shaping

- Here's the cumulative (summed) probabilities associated with it:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\mathbf{P_J}(n)$	0	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{3}{16}$	$\frac{6}{16}$	$\frac{6}{16}$	$\frac{10}{16}$	$\frac{10}{16}$	$\frac{13}{16}$	$\frac{13}{16}$	$\frac{15}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$

- **Careful** visual inspection of $\mathbf{J}_1$ let's us form the new image:

$$\mathbf{J} = \begin{array}{|c|c|c|c|} \hline 4 & 4 & 8 & 10 \\ \hline 6 & 10 & 8 & 6 \\ \hline 12 & 4 & 12 & 6 \\ \hline 10 & 10 & 8 & 14 \\ \hline \end{array}$$

HISTOGRAM MATCHING

- Just a special case of histogram shaping.
- Difference: the histogram of the original image **I** is matched to that of another image **I**´ .
- Otherwise the procedure is identical, once the cumulative probabilities are computed for the model image **I**´ .
- Useful application: **Comparing** similar images of the same scene obtained under different conditions (e.g., lighting, time of day). Extends the concept of equalizing AOD described earlier.

Discrete Fourier Transform

1. Compute the DFT of the following matrix

1	31
20	7

1. Prove that the DFT of a 2D matrix is Conjugate symmetric
    1. The magnitude of the DFT matrix is symmetric
2. Prove that the DFT of an image is periodic in nature (Periodic extension of DFT)

4. The IDFT is periodic in nature (Periodic extension of IDFT)

Example

$$I = \begin{array}{|c|c|}\hline 5 & 7 \\ \hline 8 & 3 \\ \hline \end{array}$$

$$\tilde{I}(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}$$

$$\tilde{I}(0,0) = \sum_{i=0}^{2-1}\sum_{j=0}^{2-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{2}(0*i+0*j)}$$

$$= \sum_{i=0}^{1}\sum_{j=0}^{1} I(i,j) = 21$$

$\tilde{I}(0,1) = 3.+0.j$

$\tilde{I}(1,0) = 1.+0.j$

$\tilde{I}(1,1) =$-7.+0.j

23	
	-7.+0.j

Example

$$I = \begin{array}{|c|c|}\hline 5 & 7 \\ \hline 8 & 3 \\ \hline\end{array}$$

$$\tilde{I}(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}$$

$$\tilde{I}(0,0) = \sum_{i=0}^{2-1}\sum_{j=0}^{2-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{2}(0*i+0*j)}$$

$$= \sum_{i=0}^{1}\sum_{j=0}^{1} I(i,j) = 21$$

$\tilde{I}(0,1) = 3.+0.j$

$\tilde{I}(1,0) = 1.+0.j$

$\tilde{I}(1,1) = -7.+0.j$

DFT =

23	3
1	-7

1. Compute the DFT of the following matrix

1	31
20	7

DFT =

59.+0.j	-17.+0.j
5.+0.j	-43.+0.j

2D Discrete Fourier Transform

- We will use the abbreviation

$$W_N = e^{-\sqrt{-1}\frac{2\pi}{N}} \Rightarrow \boldsymbol{W_N^{ui+vj} = e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}}$$

- Then

$$\tilde{I}(u,v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}$$

2D Discrete Fourier Transform

- We will use the abbreviation

$$W_N = e^{-\sqrt{-1}\frac{2\pi}{N}} \Rightarrow \boldsymbol{W_N^{ui+vj}} = \boldsymbol{e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}}$$

- Then

$$\tilde{I}(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, e^{-\sqrt{-1}\frac{2\pi}{N}(ui+vj)}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, \boldsymbol{W_N^{ui+vj}}$$

$$I(i,j) = \sum_{u=0}^{N-1}\sum_{u=0}^{N-1} \tilde{I}(u,v)\boldsymbol{W_N^{-(ui+vj)}}$$

Symmetry of DFT

$$\tilde{I}(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\,\boldsymbol{W_N^{ui+vj}}$$

$$\tilde{I}(N-u, N-v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, W_N^{[(N-u)i+(N-v)j]}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\, W_N^{N(i+j)}\; W_N^{-(ui+vj)}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\left[W_N^{(ui+vj)}\right]^* = \tilde{I}^*(u,v)$$

since

$$W_N^{N(i+j)} = e^{-\sqrt{-1}\,\frac{2\pi}{N}\cdot N(i+j)} = e^{-2\pi\sqrt{-1}\,(i+j)} = 1^{(i+j)} = 1 \text{ for any } i, j$$

and

$$W_N^{-(ui+vj)} = \left[W_N^{(ui+vj)}\right]^*.$$

The DFT of an image **I** is **conjugate symmetric**:

$$\tilde{I}_{real}(N-u, N-v) = \tilde{I}_{real}(u,v)\;;\; 0 \le u, v \le N-1$$
$$\tilde{I}_{imag}(N-u, N-v) = -\tilde{I}_{imag}(u,v)\;;\; 0 \le u, v \le N-1$$

Symmetry of DFT

$$\tilde{I}(N-u, N-v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j)\, W_N^{[(N-u)i+(N-v)j]}$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j)\, W_N^{N(i+j)}\, W_N^{-(ui+vj)}$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j)\, [W_N^{(ui+vj)}]^* = \tilde{I}^*(u,v)$$

since

$$W_N^{N(i+j)} = e^{-\sqrt{-1}\,\frac{2\pi}{N}\cdot N(i+j)} = e^{-2\pi\sqrt{-1}\,(i+j)} = 1^{(i+j)} = 1 \text{ for any } i, j$$

and

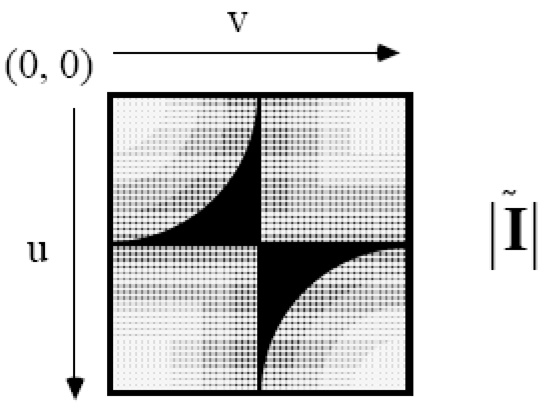
$$W_N^{-(ui+vj)} = [W_N^{(ui+vj)}]^*.$$

The DFT of an image **I** is **conjugate symmetric**:

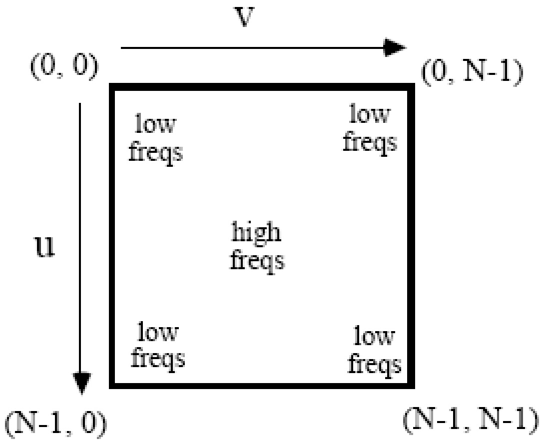
$|\tilde{I}(N-u, N-v)| = |\tilde{I}(u,v)|$ The magnitude DFT of an image **I** is **symmetric**:

Symmetry of DFT

- Depiction of the symmetry of the DFT (magnitude).



- The highest frequencies are represented near (u, v) = (N/2, N/2).



Periodicity of DFT

- We have defined the DFT matrix as **finite** in extent (N x N):

$$\tilde{\mathbf{I}} = \left[\tilde{I}(u, v)\ ;\ 0 \leq u, v \leq N\text{-}1\right]$$

- However, if the arguments are allowed to take values outside the range 0 ≤ u, v ≤ N-1, we find that the DFT is periodic in both the u- and v-directions, with **period N**:
- For any integers m, n

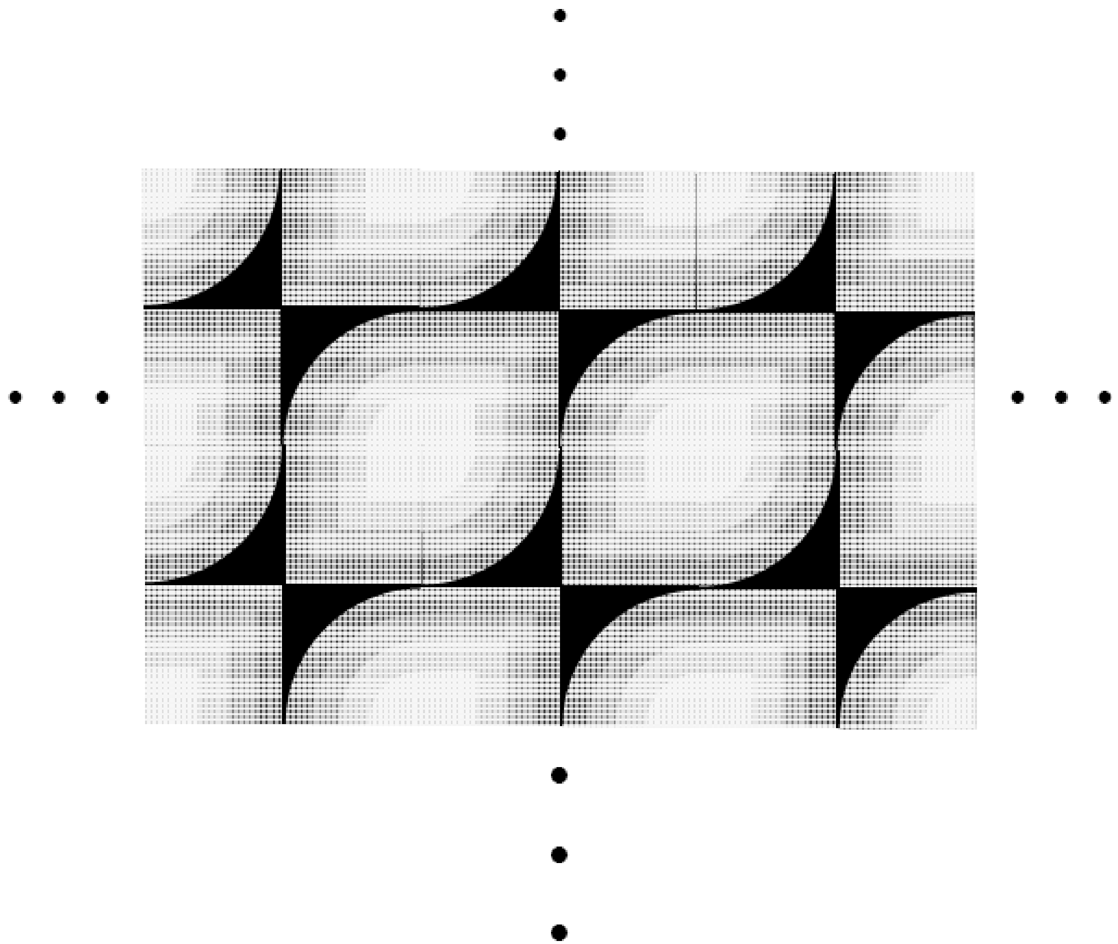
$$\begin{aligned}
\tilde{I}(u+nN, v+mN) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\ W_N^{[(u+nN)i+(v+mN)j]} \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\ W_N^{N(ni+mj)}\ W_N^{(ui+vj)} \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\ W_N^{(ui+vj)} = \tilde{I}(u, v)
\end{aligned}$$

since

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

- This is called the **periodic extension** of the DFT. It is defined for all integer frequencies u, v.

Periodic Extension of DFT



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

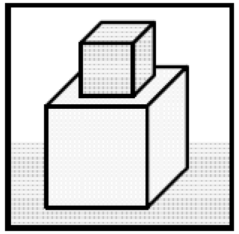
$$\begin{aligned} I(i+nN, j+mN) &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-[u(i+nN)+v(j+mN)]} \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\, W_N^{-(ui+vj)}\, W_N^{-N(nu+mv)} \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{I}(u, v)\,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

since

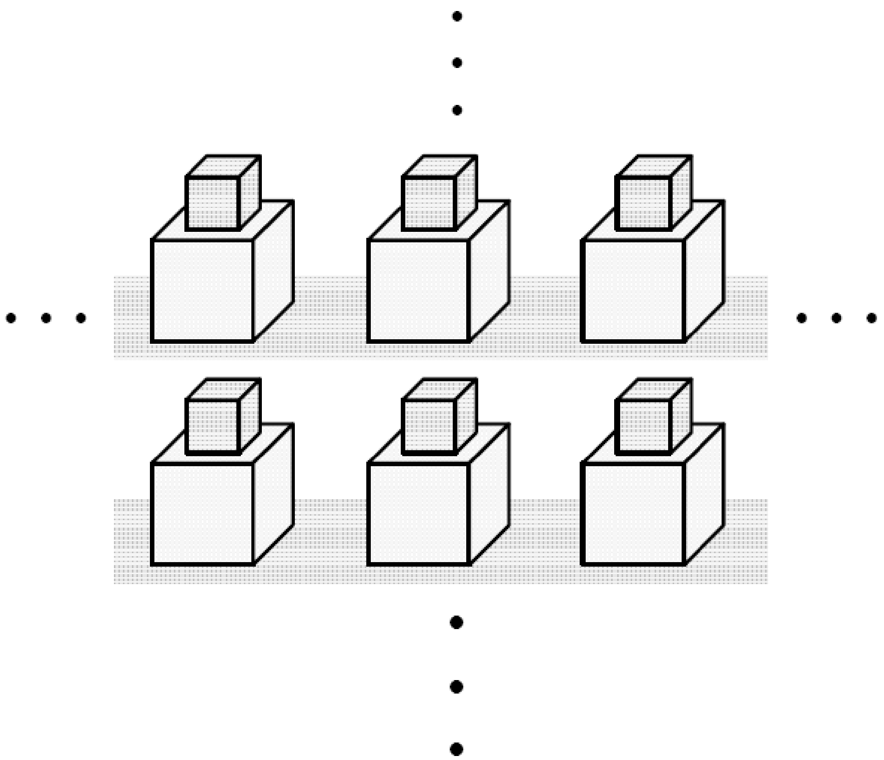
$$W_N^{-N(nu+mv)} = e^{-\sqrt{-1}\,\frac{2\pi}{N}\cdot N(nu+mv)} = e^{-2\pi\sqrt{-1}\,(nu+mv)} = 1^{(nu+mv)} = 1$$

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

Periodic Extension of Image



Image



Displaying the DFT

- Usually, the DFT is displayed with its center coordinate (u, v) = (0, 0) at the center of the image.
- This way, the lower frequency information (which usually dominates an image) is clustered together near the origin at the center of the display.

$$\tilde{I}(u - \tfrac{N}{2}, v - \tfrac{N}{2}) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\, W_N^{[(u-N/2)i+(v-N/2)j]}$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\, W_N^{(ui+vj)}\; W_N^{-N(i+j)/2}$$

$$W_N^{-N(i+j)/2} = e^{\sqrt{-1}\,\frac{2\pi}{N}\,N(i+j)/2} = e^{\sqrt{-1}\pi\,(i+j)} = (-1)^{i+j}$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)\,(-1)^{i+j}\, W_N^{(ui+vj)}$$

$$= \mathrm{DFT}\left[(-1)^{i+j} I(i, j)\right]$$

- This can be accomplished in practice by taking the DFT of the alternating image (for display purposes only!)

$$[(-1)^{i+j} I(i,j)\ ;\ 0 \le i, j \le N-1]$$

Displaying the DFT

- Usually, the DFT is displayed with its center coordinate (u, v) = (0, 0) at the center of the image.
- This way, the lower frequency information (which usually dominates an image) is clustered together near the origin at the center of the display.
- This can be accomplished in practice by taking the DFT of the alternating image (for display purposes only!)

$$[(-1)^{i+j} I(i,j) \; ; \; 0 \leq \; i, j \leq \; N-1]$$

- Observe that

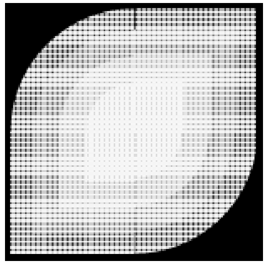
$$(-1)^{i+j} = e^{\sqrt{-1}\pi\,(i+j)} = e^{\sqrt{-1}\,\frac{2\pi}{N}\,N(i+j)/2} = W_N^{-N(i+j)/2}$$

so

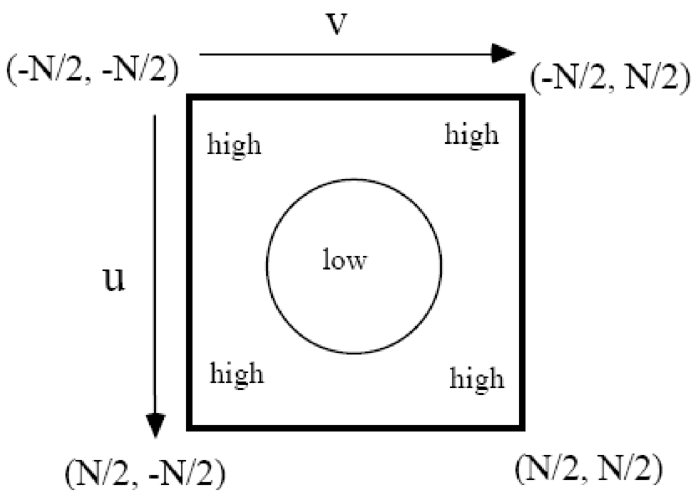
$$\begin{aligned} \mathrm{DFT}\left[(-1)^{i+j} I(i,j)\right] &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\,(-1)^{i+j}\,W_N^{(ui+vj)} \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\,W_N^{(ui+vj)}\;W_N^{-N(i+j)/2} \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\,W_N^{[(u-N/2)i+(v-N/2)j]} \\ &= \tilde{I}(u - \tfrac{N}{2}, v - \tfrac{N}{2}) \end{aligned}$$

- A simple shift of the DF

Centered DFT



Centered DFT



Spatial Filtering

Filtering

1. Apply a 3X3 spatial smoothing filter on the image I
2. Sharpen Image using a
    1. Laplacian filter (3X3 filter)
    2. Unsharp mask (3X3 filter)

$$\mathbf{I} = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 3 & 4 \\ \hline 2 & 5 & 3 & 2 \\ \hline 8 & 1 & 8 & 2 \\ \hline 4 & 5 & 3 & 11 \\ \hline \end{array}$$

Spatial Convolution Operator

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

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

Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction

Blurring is used in removal of small details and bridging of small gaps in lines or curves

Smoothing spatial filters include linear filters and nonlinear filters.

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.

Convolution

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 3 & 4 \\ 2 & 5 & 3 & 2 \\ 8 & 1 & 8 & 2 \\ 4 & 5 & 3 & 11 \end{bmatrix}$$

1. Apply a 3X3 spatial smoothing filter on the image I
   Solution: Convolution with a smoothing filter

Sum of Products

0	0	0	0	0	0
0	1	1	3	4	0
0	2	5	3	2	0
0	8	1	8	2	0
0	4	5	3	11	0
0	0	0	0	0	0

=

1			

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Rotated

Sum of Products

0	0	0	0	0	0
0	1	1	3	4	0
0	2	5	3	2	0
0	8	1	8	2	0
0	4	5	3	11	0
0	0	0	0	0	0

=

1	15/9		

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Rotated

Note: Return result without 0 padding

Sum of Products

0	0	0	0	0	0
0	1	1	3	4	0
0	2	5	3	2	0
0	8	1	8	2	0
0	4	5	3	11	0
0	0	0	0	0	0

=

1	15/9	18/9	

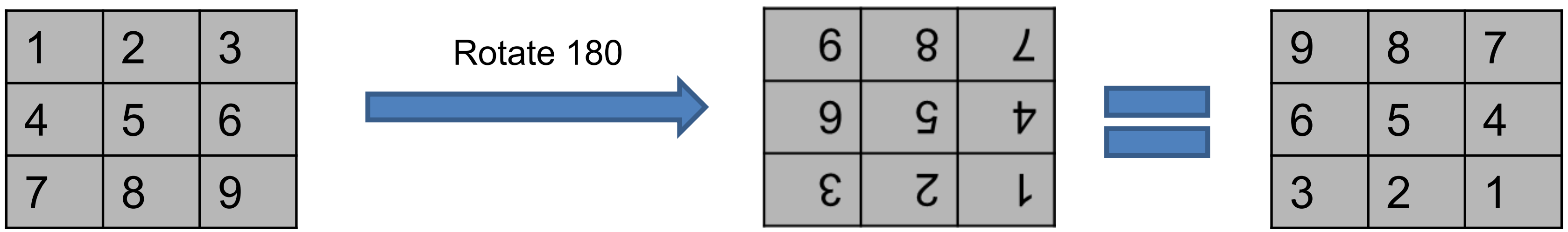
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Rotated

Note: You need to rotate the filter by 180 degrees for convolution! It's the same if the filter is symmetric (e.g., smoothing filter)

Rotate Filter Example



Laplace Operator

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

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

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.

Convolution

$$\underbrace{-1 \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 3 & 4 \\ 2 & 5 & 3 & 2 \\ 8 & 1 & 8 & 2 \\ 4 & 5 & 3 & 11 \end{bmatrix}} + \underbrace{\begin{bmatrix} 1 & 1 & 3 & 4 \\ 2 & 5 & 3 & 2 \\ 8 & 1 & 8 & 2 \\ 4 & 5 & 3 & 11 \end{bmatrix}}$$

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

Note: You need to remember the filters!

Unsharp Masking and Highboost Filtering

- Unsharp masking

  Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

  e.g., printing and publishing industry

- Steps

  1. Blur the original image
  2. Subtract the blurred image from the original
  3. Add the mask to the original

Let $\overline{f}(x, y)$ denote the blurred image, unsharp masking is

$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

when $k > 1$, the process is referred to as highboost filtering.