UNIVERSITY OF SOUTH CAROLINA

CSCE 590 INTRODUCTION TO IMAGE PROCESSING

Single Image Operations

Ioannis Rekleitis

Single pixel operations

- Determined by
  - Transformation function T
  - Input intensity value
- Not depend on other pixels and position

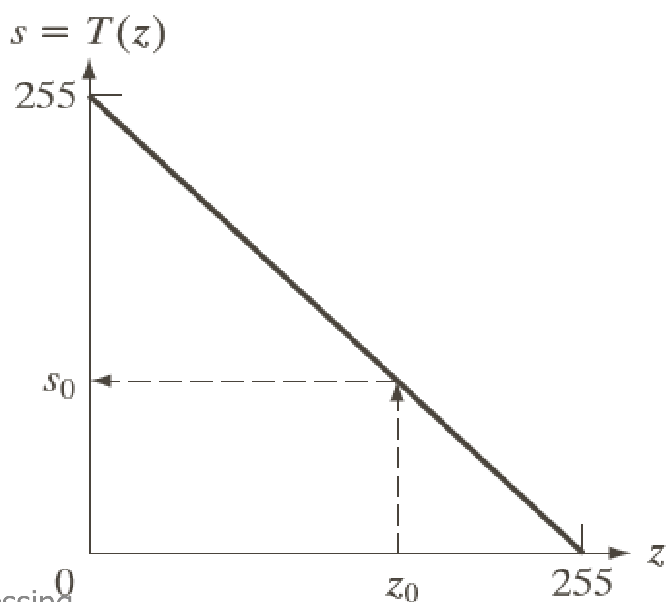


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

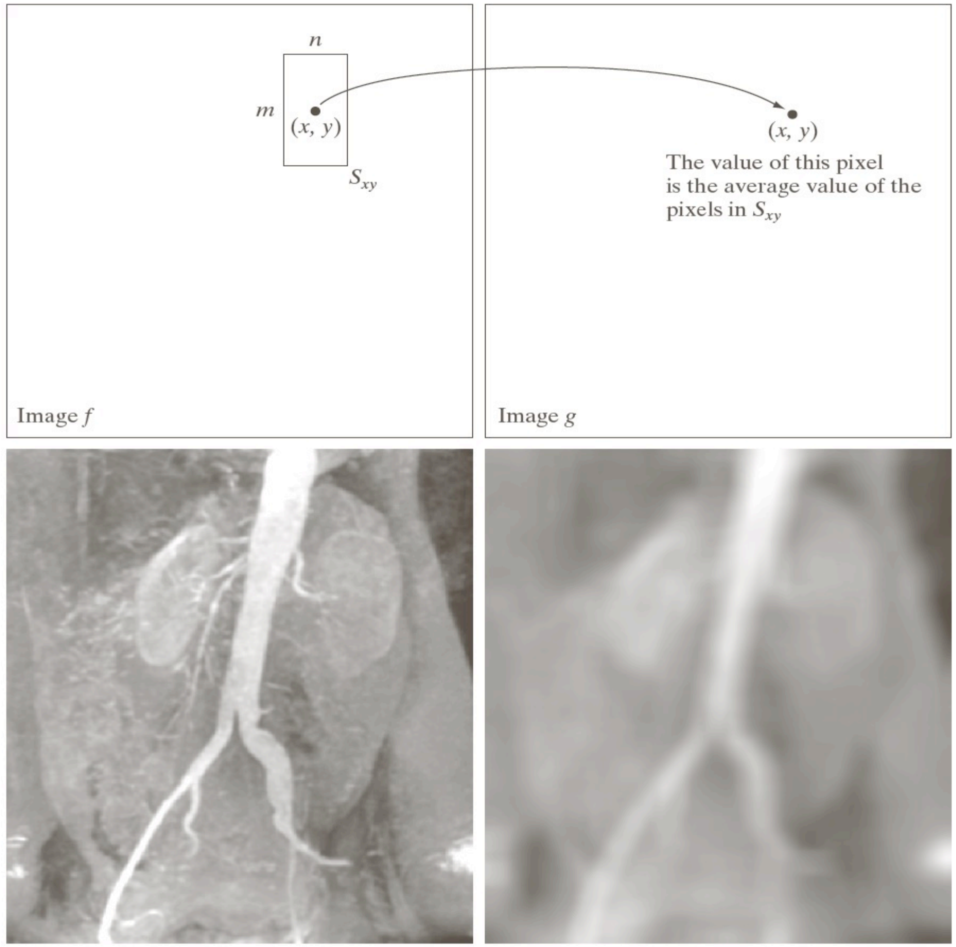
Neighborhood Operations

Image smoothing

$$g(x, y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r,c)$$

Other examples:

- Interpolation
- Image filtering



a b
c d

FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790×686 pixels.

Image Resampling & Interpolation

Need to resample the image when

- **Rescaling**
- **Geometrical transformation**
- **The output image coordinates are not discrete**

Interpolation methods:

- **Nearest neighbor**
  - **Fast and simple**
  - **Loss of sharpness**
  - **Artifacts (checkerboard)**
- **Bilinear**
- **Bicubic**
  - **Images are sharpest**
  - **Fine details are preserved**
  - **Slow**

Slides courtesy of Prof. Yan Tong

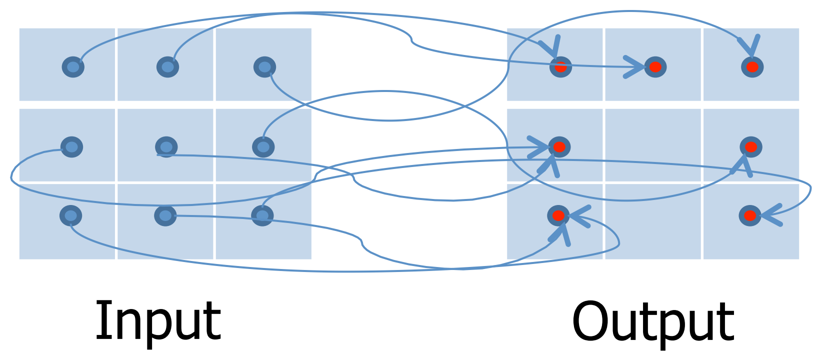
Image Resampling & Interpolation

a b c
d e f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

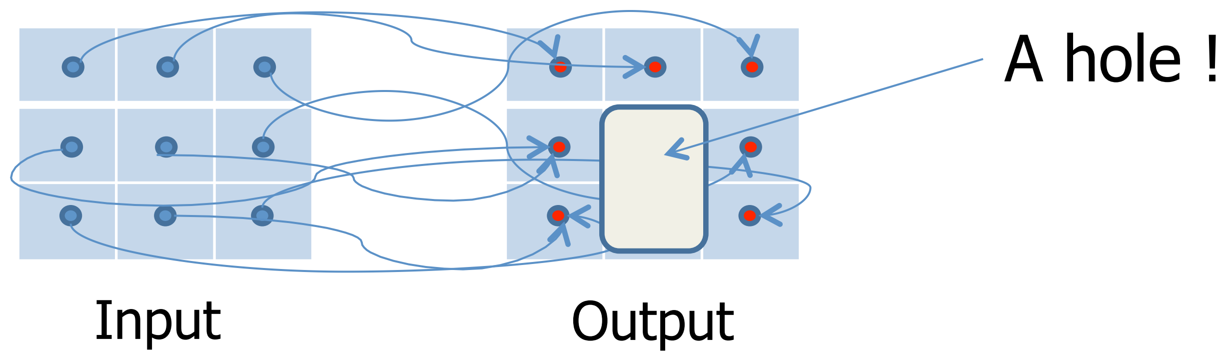
Image Resampling & Interpolation

- Forward mapping



Issues on Image Resampling & Interpolation

- Missing points in forward mapping



- Solution: perform a backward mapping

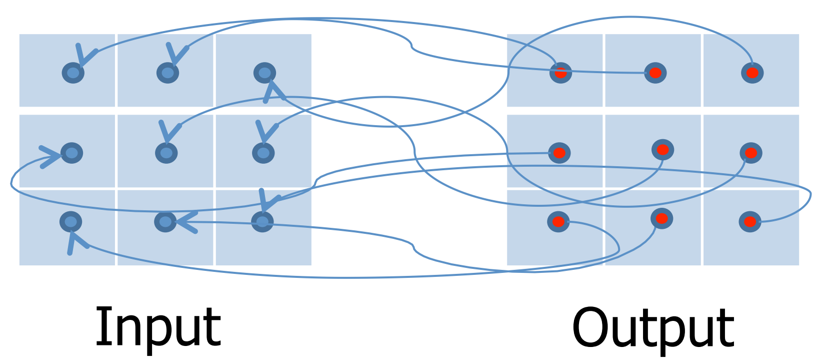
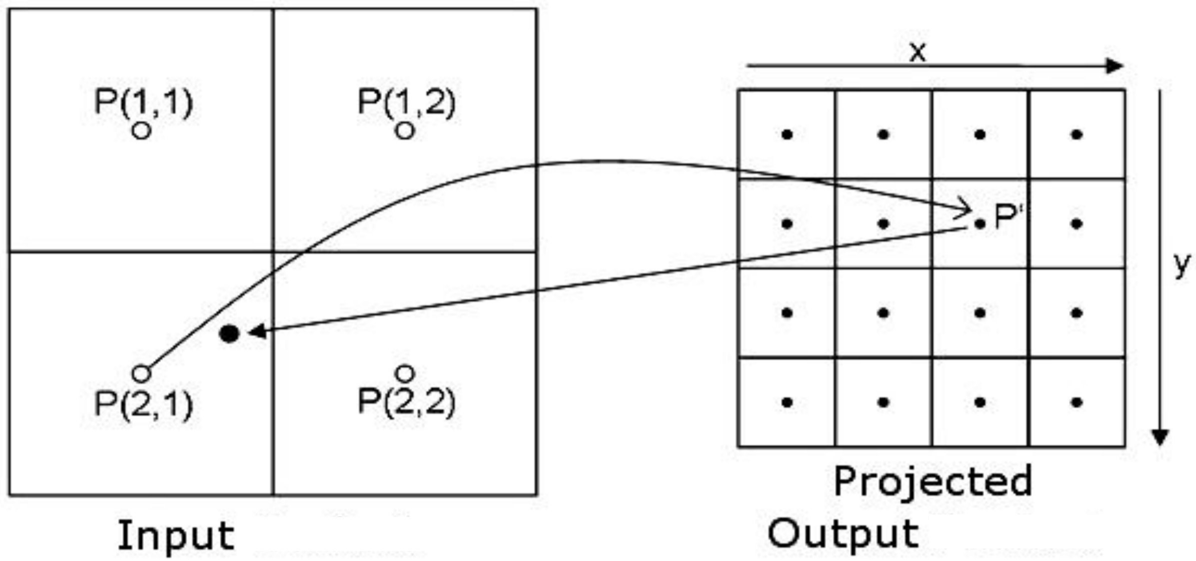


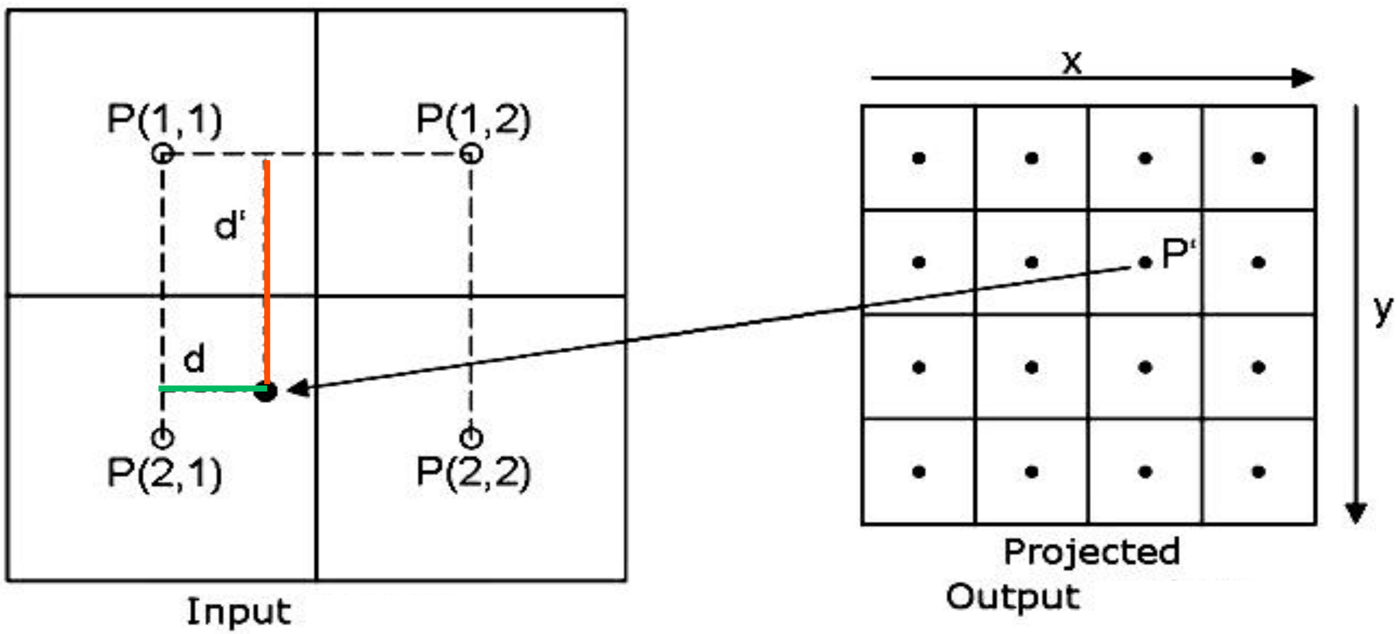
Image Interpolation – Nearest Neighbor



http://www.brockmann-consult.de/beam/doc/help/general/ResamplingMethods.html

Assign each pixel in the output image with the nearest neighbor in the input image.

Image Interpolation – Bilinear

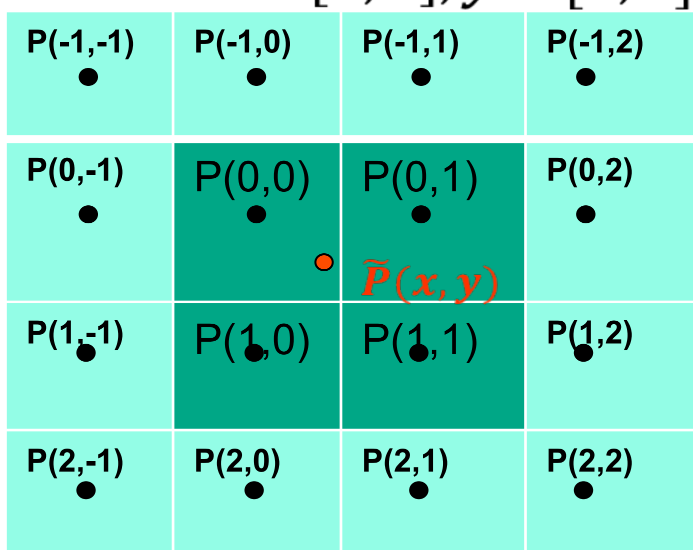


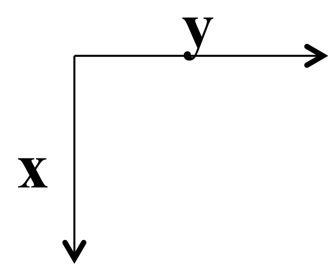
http://www.brockmann-consult.de/beam/doc/help/general/ResamplingMethods.html

$$P' = P(1,1)(1-d)(1-d') + P(1,2)d(1-d') + P(2,1) * d' * (1-d) + P(2,2)dd'$$

Image Interpolation – Bicubic

If we know the intensity values, derivatives, and cross derivatives for the four corners (0,0), (0,1), (1,0), and (1,1), we can interpolate any point (x,y) in the region $x \in [0,1], y \in [0,1]$





$$\tilde{P}(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j$$

Need to solve the 16 coefficients

Some Basic Relationships between Pixels

Neighbors of a pixel

$N_4(p)$

	(x-1,y)	
(x,y-1)	(x,y)	(x,y+1)
	(x+1,y)	

$N_D(p)$

(x-1,y-1)		(x-1,y+1)
	(x,y)	
(x-1,y+1)		(x+1,y+1)

$N_8(p)$

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	(x,y)	(x,y+1)
(x-1,y+1)	(x+1,y)	(x+1,y+1)

Slides courtesy of Prof. Yan Tong

Adjacency

- Adjacency is the relationship between two pixels p and q
- V is a set of intensity values used to define adjacency $\boldsymbol{V} \sqsubseteq \{\mathbf{0}, \mathbf{1}, \dots, \mathbf{255}\}$

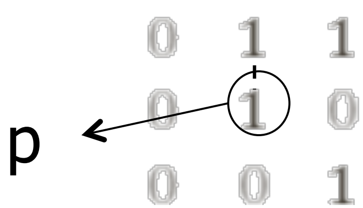
$$\boldsymbol{f}(\boldsymbol{p}) \in \boldsymbol{V} \quad \text{and} \quad \boldsymbol{f}(\boldsymbol{q}) \in \boldsymbol{V} \Rightarrow$$ Intensity constraints

- Binary image: V={1} or V={0}
- Gray level image:

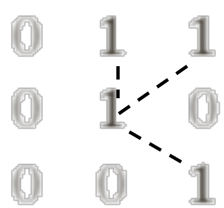
Adjecency

Three types of adjacency:

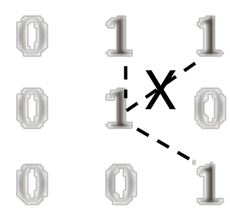
4-adjacency **8-adjacency** **m-adjacency**



$q \in N_4(p)$



$q \in N_8(p)$



$q \in N_D(p)$
$or\ q \in N_4(p)$

and $N_4(q) \cap N_4(p) = \emptyset$

13

Connectivity

- **Path from p to q: a sequence of distinct and adjacent pixels with coordinates**

Starting point p

$$\underbrace{(x_0, y_0), (x_1, y_1)}_{\textbf{adjacent}}, \ldots, (x_n, y_n)$$

ending point q

- ***Closed* path: if the starting point is the same as the ending point**
- **p and q are *connected*: if there is a path from p to q in S**
- ***Connected component*: all the pixels in S connected to p**
- ***Connected set*: S has only one connected component**

Are they connected sets?

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

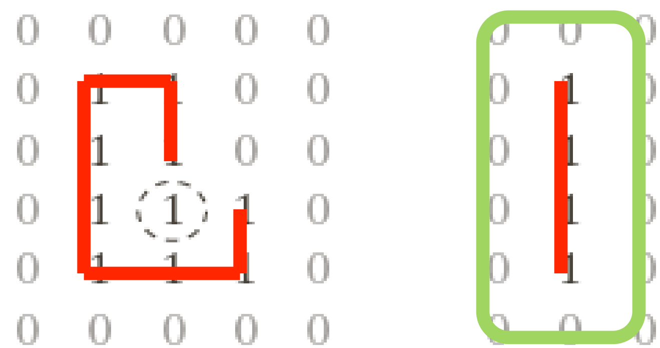
Regions

- R is a region if R is a connected set
- R_i and R_j are adjacent if $R_i \cup R_j$ is a connected set

$$\begin{array}{l} \left.\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}\right\} R_i \\ \left.\begin{matrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}\right\} R_j \end{array}$$

Boundaries

- Inner boundary (boundary) -- the set of pixels each of which has at least one background neighbor
- Outer boundary – the boundary pixels in the background



Distance Measures

- For pixels p, q, and z, with coordinates (x,y), (s,t) and (v,w), D is a distance function or metric if

$$(a)\ D(p,q) \geq 0 \quad D(p,q) = 0\ iff\ \ p = q$$

$$(b)\ D(p,q) = D(q,p), \quad and$$

$$(c)\ D(p,z) \leq D(p,q) + D(q,z)$$

Distance Measures

- Euclidean distance $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$
- City-block (D4) distance $D_4(p,q) = |x-s| + |y-t|$
- Chessboard (D8) distance (Chebyshev distance)

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

Distance: Sample Problem

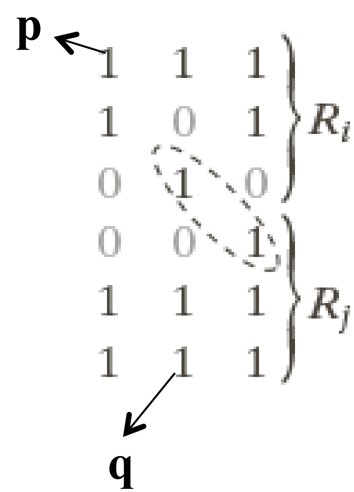
- D4 distance

6

- D8 distance

5

- Euclidean distance $\sqrt{1+5^2}$



Distance vs length of a path?

Geometric Spatial Transformations – Rubber Sheet Transformation

$$(x, y) = T\{(v, w)\}$$

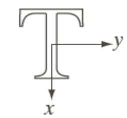
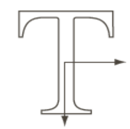
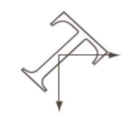
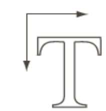
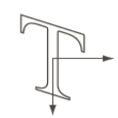
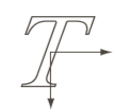
Affine transform:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{T}\begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}\begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

Inverse mapping

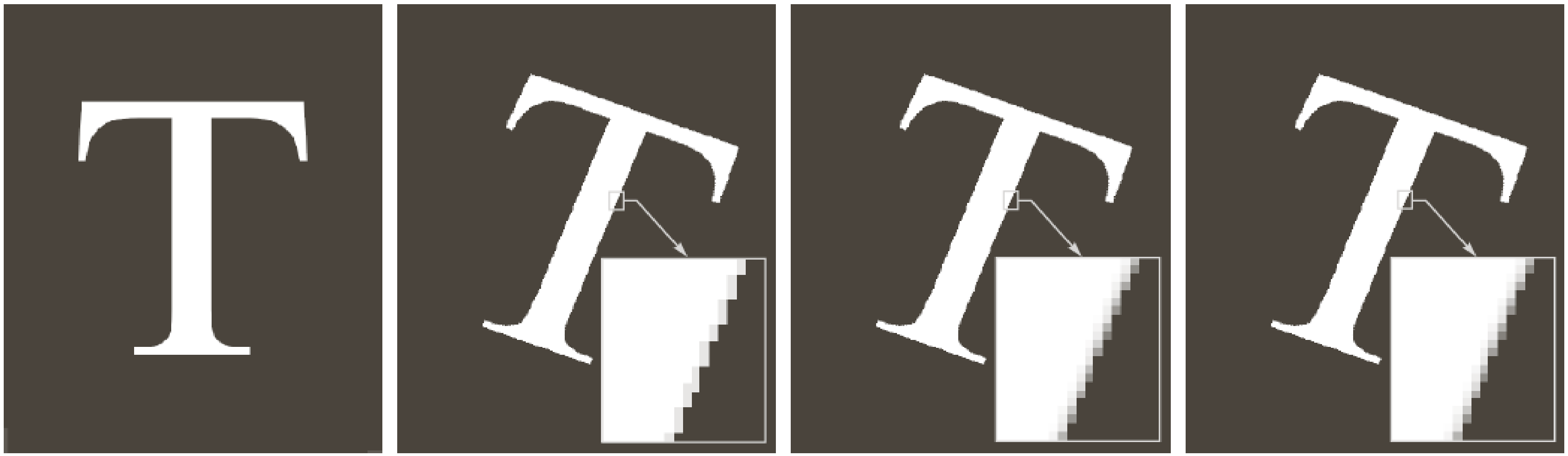
$$\begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \mathbf{T}^{-1}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Slides courtesy of Prof. Yan Tong

Geometric Spatial Transformations



Nearest neighbor Bilinear Bicubic

Note: a neighborhood operation, i.e., interpolation, is required following geometric transformation

Image Registration

Compensate the geometric change in:

- view angle
- distance
- orientation
- sensor resolution
- object motion

Four major steps:

- Feature detection
- Feature matching
- Transformation model
- Resampling

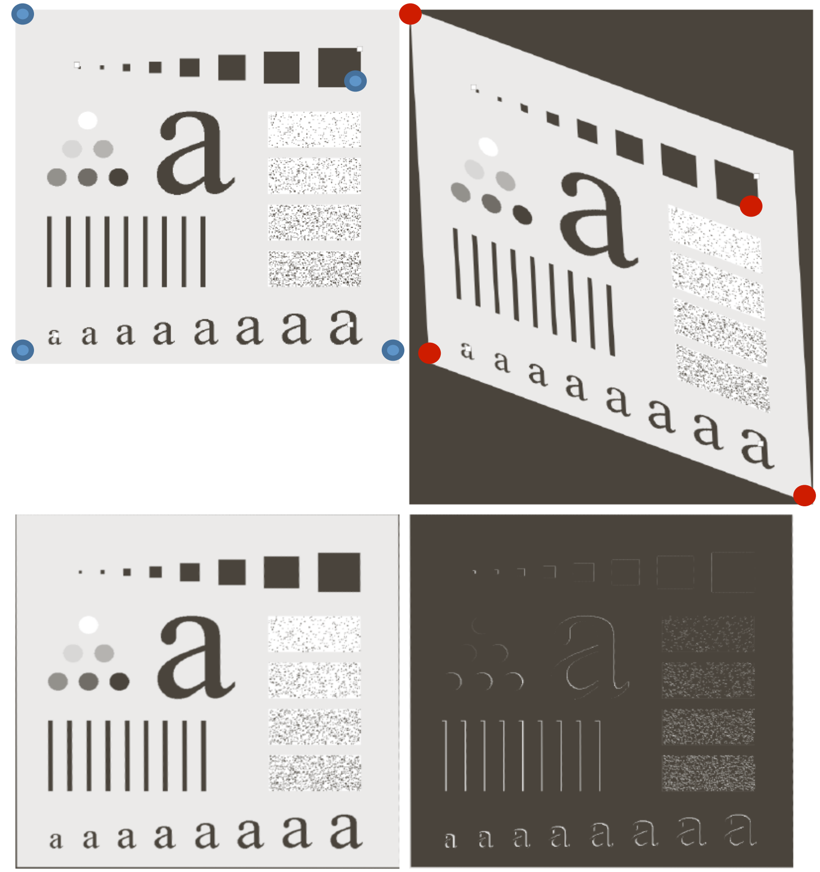
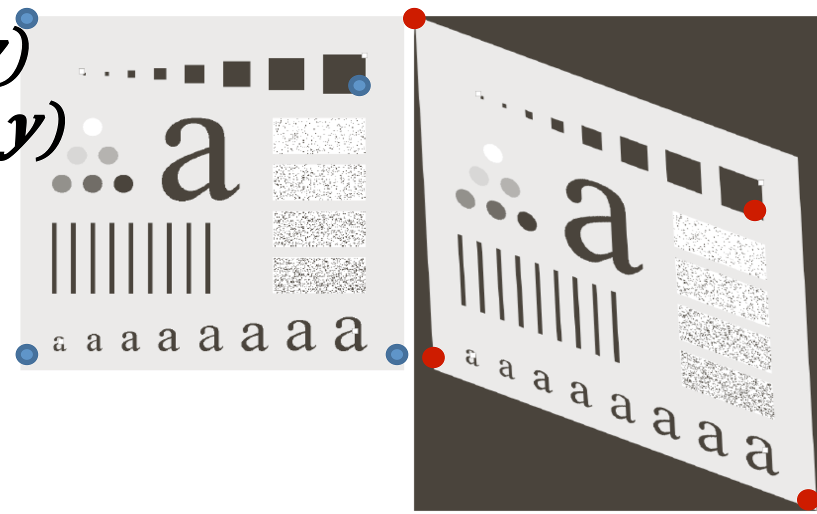


FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

Image Registration

Coordinates in the moving image *(v,w)*
Coordinates in the template image *(x,y)*



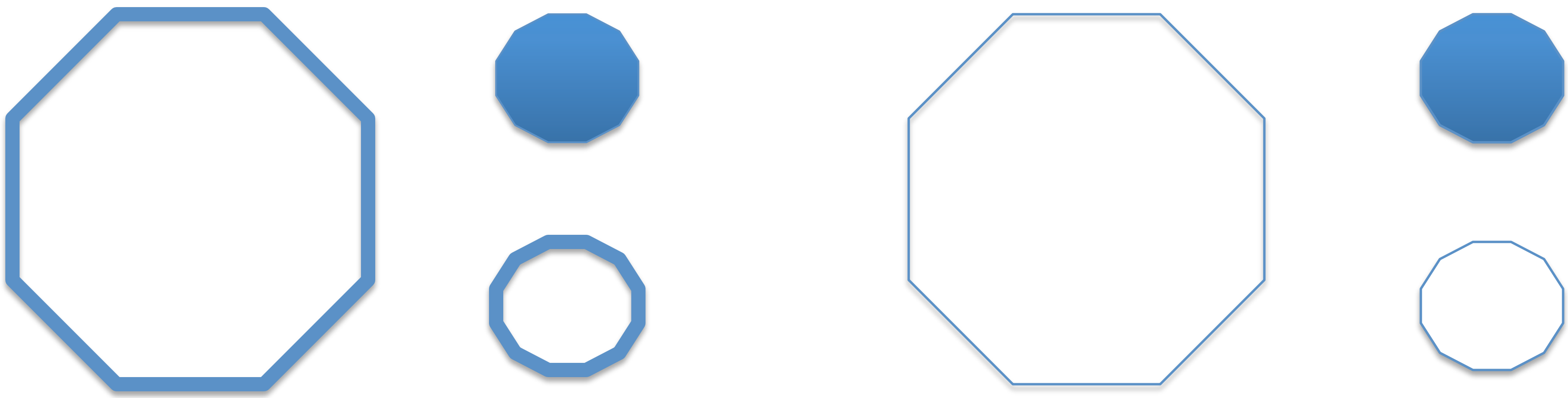
$$x = c_1 v + c_2 w + c_3 vw + c_4$$

$$y = c_5 v + c_6 w + c_7 vw + c_8$$

- **Known: coordinates of the points *(x,y)* and *(v,w)***
- **Unknown: c_1 to c_8**

4 tie points -> 8 equations

Dilation and Erosion

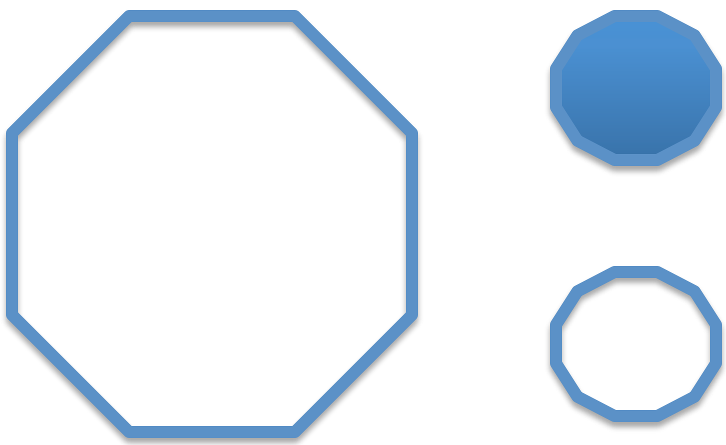


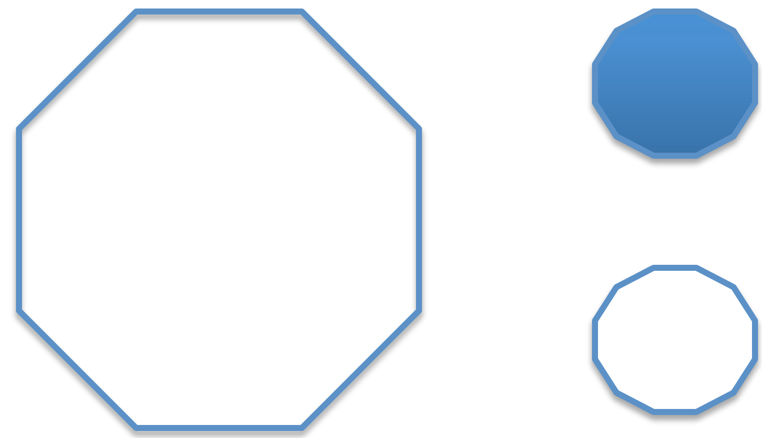
Erosion

Dilation

Usually on binary images, after thresholding and/or segmentation

Dilation and Erosion

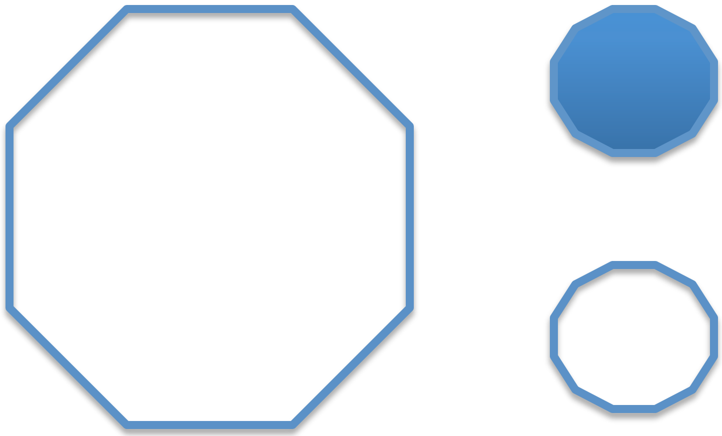


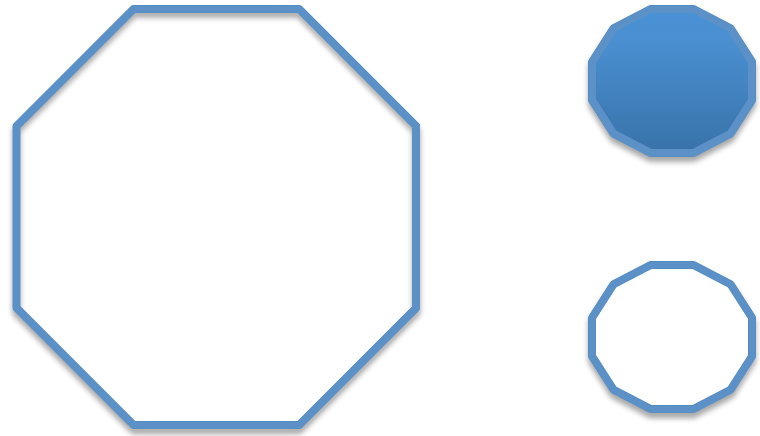


Erosion

Dilation

Dilation and Erosion

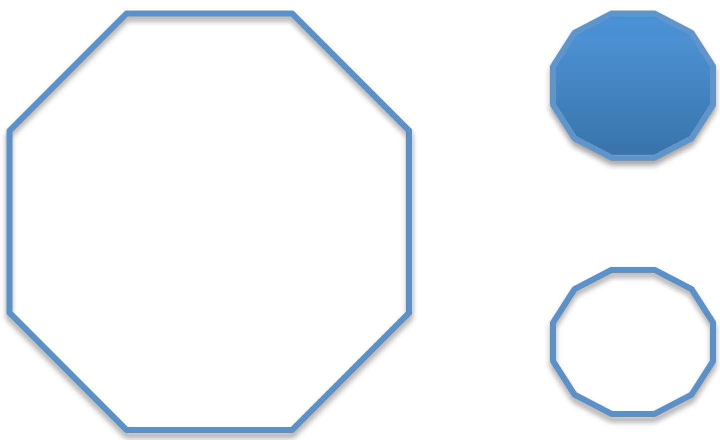


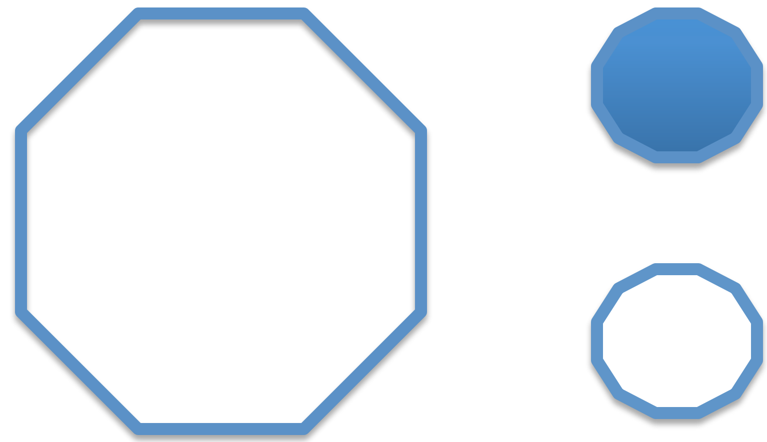


Erosion

Dilation

Dilation and Erosion

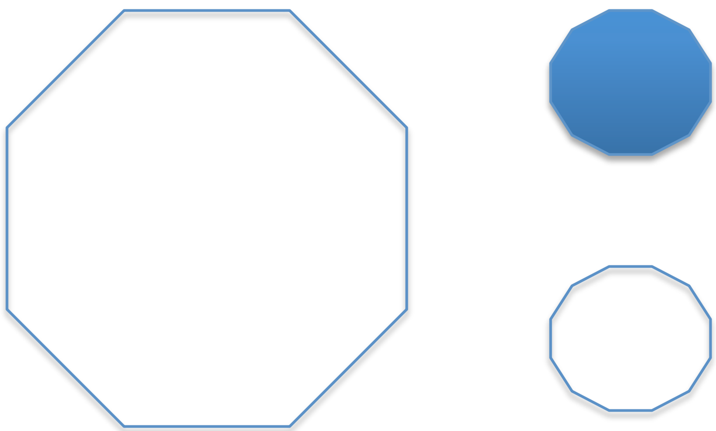


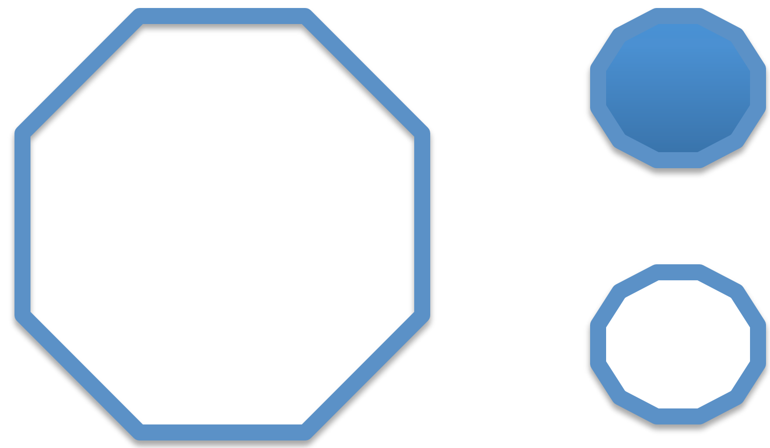


Erosion

Dilation

Dilation and Erosion





Erosion

Dilation

Questions?