27.09.2019

Digital Image Processing (CSE/ECE 478)

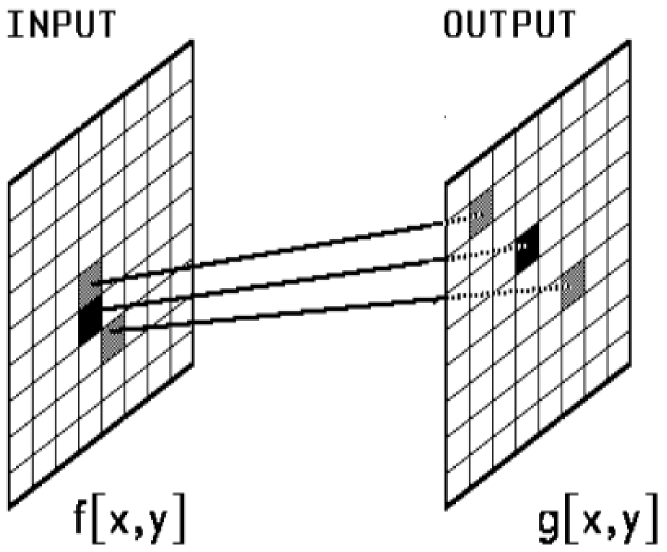
Lecture-13: Geometric Operations

Ravi Kiran

Center for Visual Information Technology (CVIT), IIIT Hyderabad

Geometric Operations

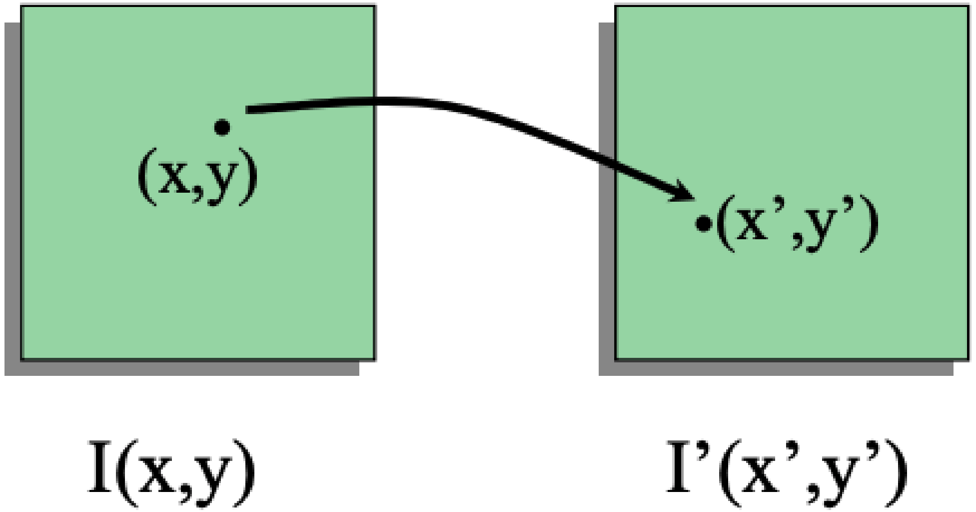
- Filters, Point Operations
  - modify color values (range) of pixels
  - domain (x,y) remains (mostly) fixed
- **Geometric transformation**
  - modify the positions of pixels
  - .. but keep their colors (mostly) unchanged



$$x \to f_x(x, y) = x'$$

$$y \to f_y(x, y) = y'$$

$$I(x, y) = I'\left(f_x(x, y), f_y(x, y)\right)$$

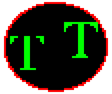


Geometric Operations

- **Scale** - change image content size

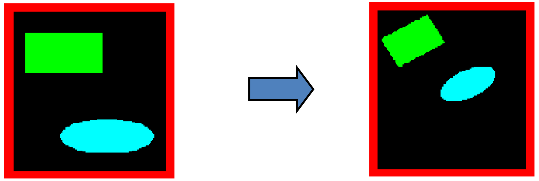
- **Rotate** - change image content orientation

- **Reflect** - flip over image contents



- **Translate** - change image content position

- **Affine Transformation**
  - general image content linear geometric transformation

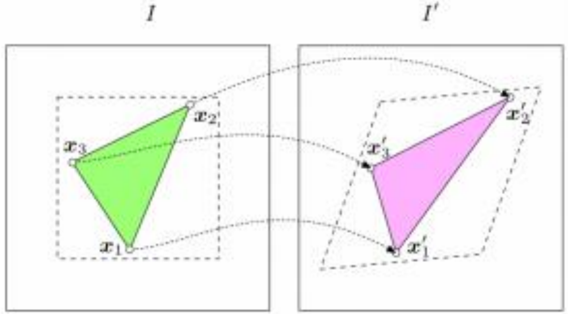


Affine (3-Point) Mapping

- Can use homogeneous coordinates to rewrite translation, rotation, scaling, etc as vector-matrix multiplication

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **Affine mapping:** Can then derive values of matrix that achieve desired transformation (or combination of transformations)



- Inverse of transform matrix is **inverse mapping**

https://web.cs.wpi.edu/~emmanuel/courses/cs545/S14/slides/lecture11.pdf

B translation

B rotation

Translation: $x(k,l) = k + 50; y(k,l) = l;$

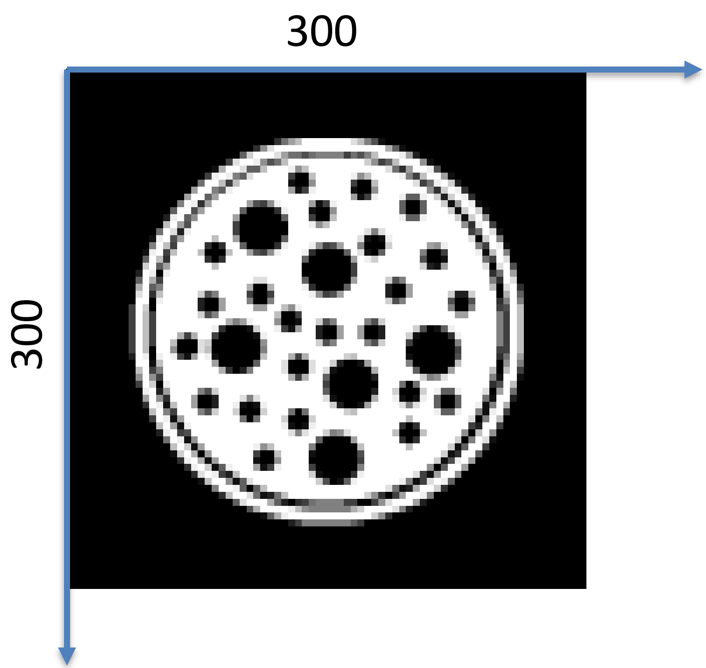
Rotation: $x(k,l) = (k - x_0)cos(\theta) + (l - y_0)sin(\theta) + x_0;$

$y(k,l) = -(k - x_0)sin(\theta) + (l - y_0)cos(\theta) + y_0;$

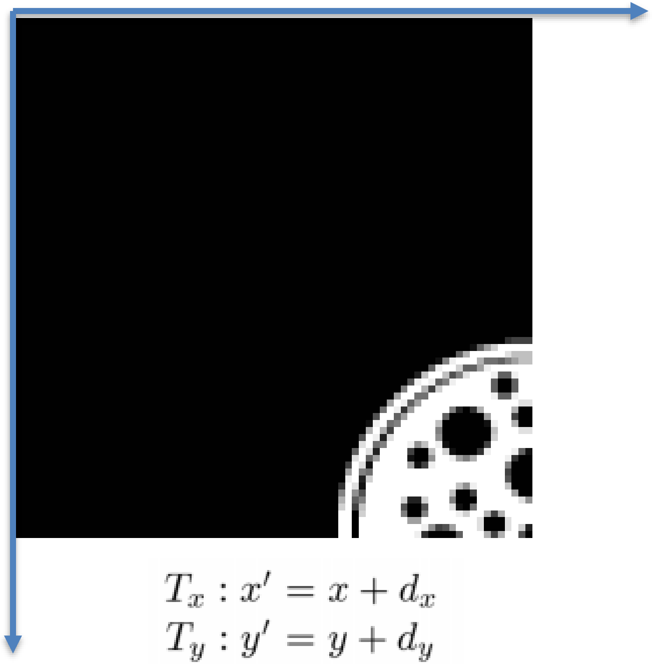
$x_0 = y_0 = 256.5$ the center of the image $\mathbf{A}$, $\theta = \pi/6$

By Onur Guleyuz

Translation



Translation by (150,150)

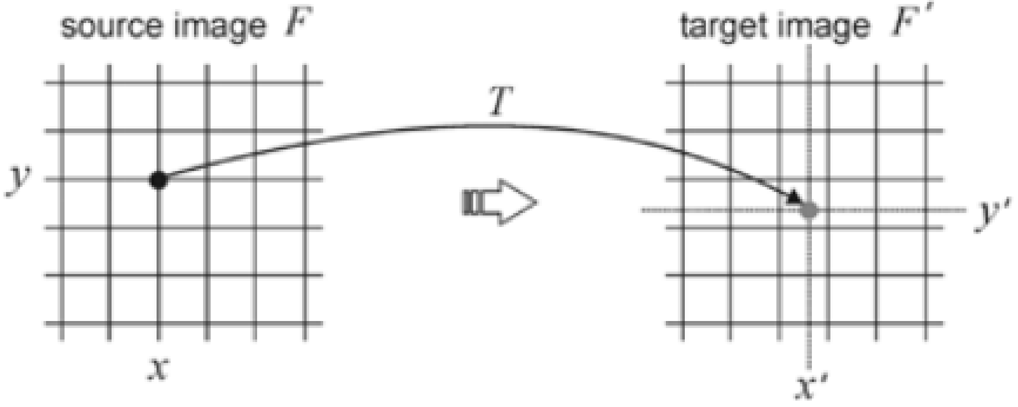


$$T_x : x' = x + d_x$$
$$T_y : y' = y + d_y$$

- Issues ?
- Translation by (30.7, 30.7)

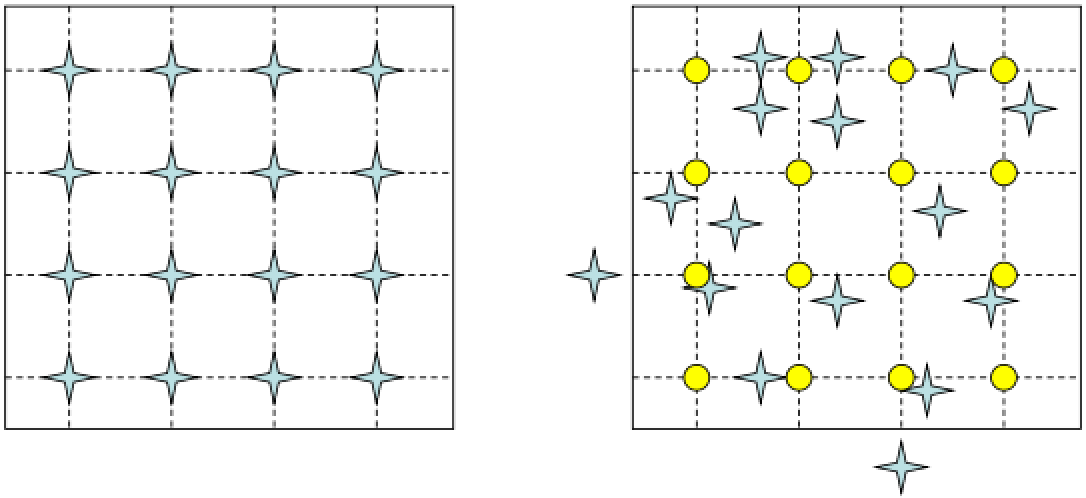
Interpolation methods

- Forward mapping



https://www.d.umn.edu/~dzhou/ME4060/ME%204060%20Chapter%207-1%20Geometria%20operation.pdf

Forward Warping



Warping points are often non-integer samples

Many integer samples “o” are not assigned Values

http://eeweb.poly.edu/~yao/EL5123/lecture12_ImageWarping.pdf

Two Issues: Mapping

- The issue of how integer-valued source coordinates are mapped onto integer-valued destination coordinates must also be addressed.
  - Forward mapping takes each pixel of the source image and copies it to a location in the destination by rounding the destination coordinates so that they are integer values.
  - Forward mapping yields generally poor results since certain pixels of the destination image may remain unfilled. Example: a source image is rotated by 45 degrees using a forward mapping strategy. Example: scaling an image to make it larger!

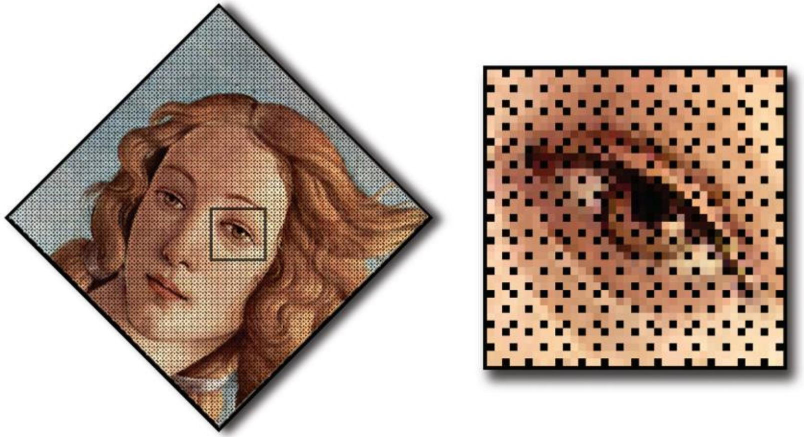


Figure 7.3. Rotation by forward mapping.

Two Issues

- Two issues:
  - Dimensionality: The destination image may not be large enough to contain all of the processed samples
  - Transformed locations are not integers: How can we place a source sample at a non-integer location in the destination?

(a) Source image. (b) Rotation. (c) Expanded view of rotated image.

Figure 7.2. Destination dimensionality under rotation.

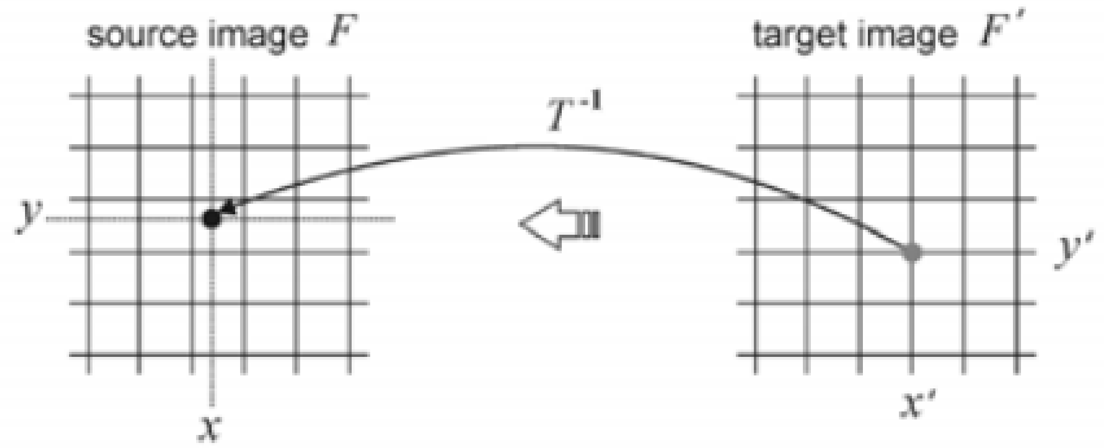
Backward mapping

- Backward mapping solves the gap problem caused by forward mapping.
  - An empty destination image is created and each location in the destination is mapped backwards onto the source.
  - The source location may not be integer-valued coordinates; hence a sample value is obtained via interpolation.
- Let A be an affine transform matrix and let v be a location in the destination image such that v = $[x,y,1]^T$

$$\begin{aligned} v' &= Av, \\ A^{-1}v' &= A^{-1}Av, \\ A^{-1}v' &= v. \end{aligned}$$

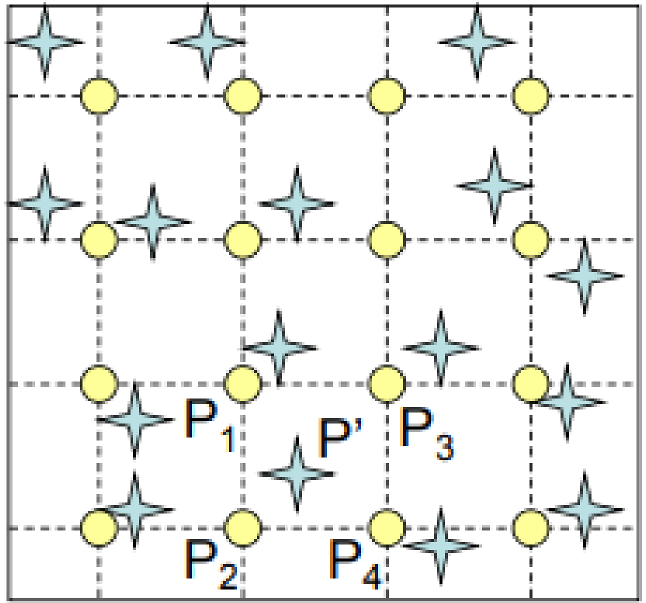
Interpolation methods

- Backward mapping

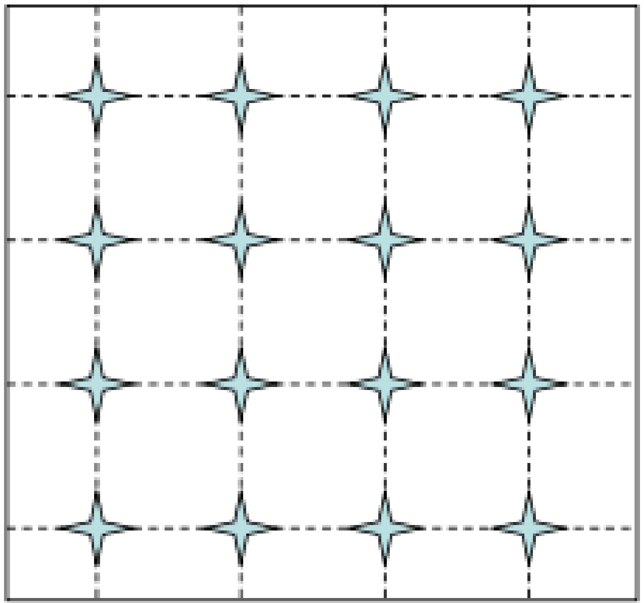


https://www.d.umn.edu/~dzhou/ME4060/ME%204060%20Chapter%207-1%20Geometria%20operation.pdf

Inverse Warping

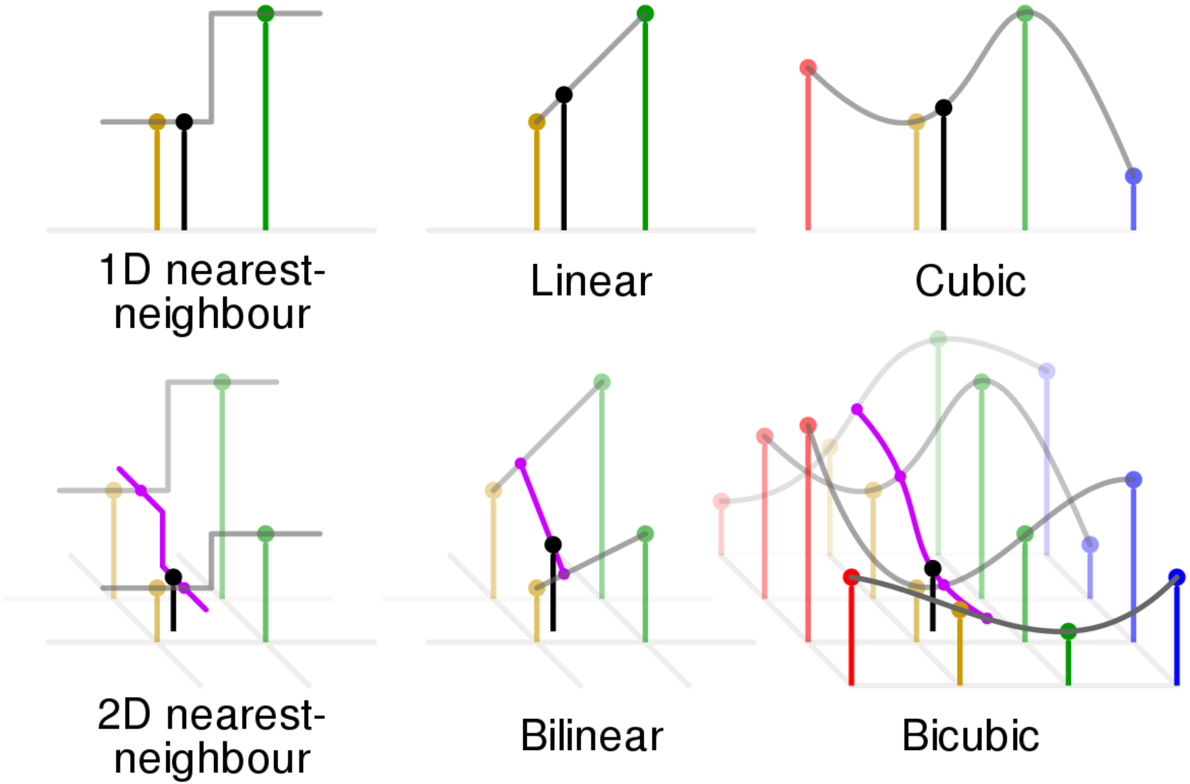


P' will be interpolated
from P_1, P_2, P_3, and P_4

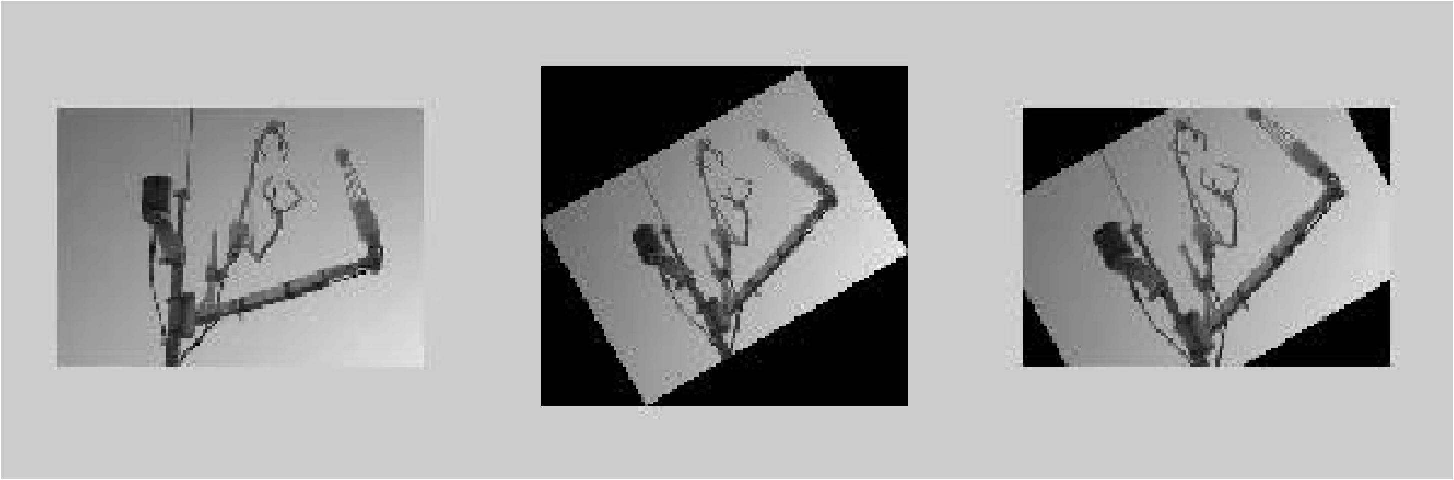


http://eeweb.poly.edu/~yao/EL5123/lecture12_ImageWarping.pdf

Interpolation Function



Example: MATLAB's `imrotate`



I

J = imrotate(I,30,`loose')

J = imrotate(I,30,`crop')

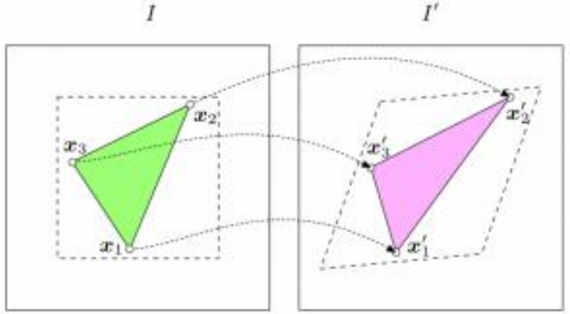
NOTE: Rotation is about center of image, MATLAB's `T = imregtform()` rotates about top-left corner of image

Affine (3-Point) Mapping

- Can use homogeneous coordinates to rewrite translation, rotation, scaling, etc as vector-matrix multiplication

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **Affine mapping:** Can then derive values of matrix that achieve desired transformation (or combination of transformations)



- Inverse of transform matrix is **inverse mapping**

https://web.cs.wpi.edu/~emmanuel/courses/cs545/S14/slides/lecture11.pdf

WARP

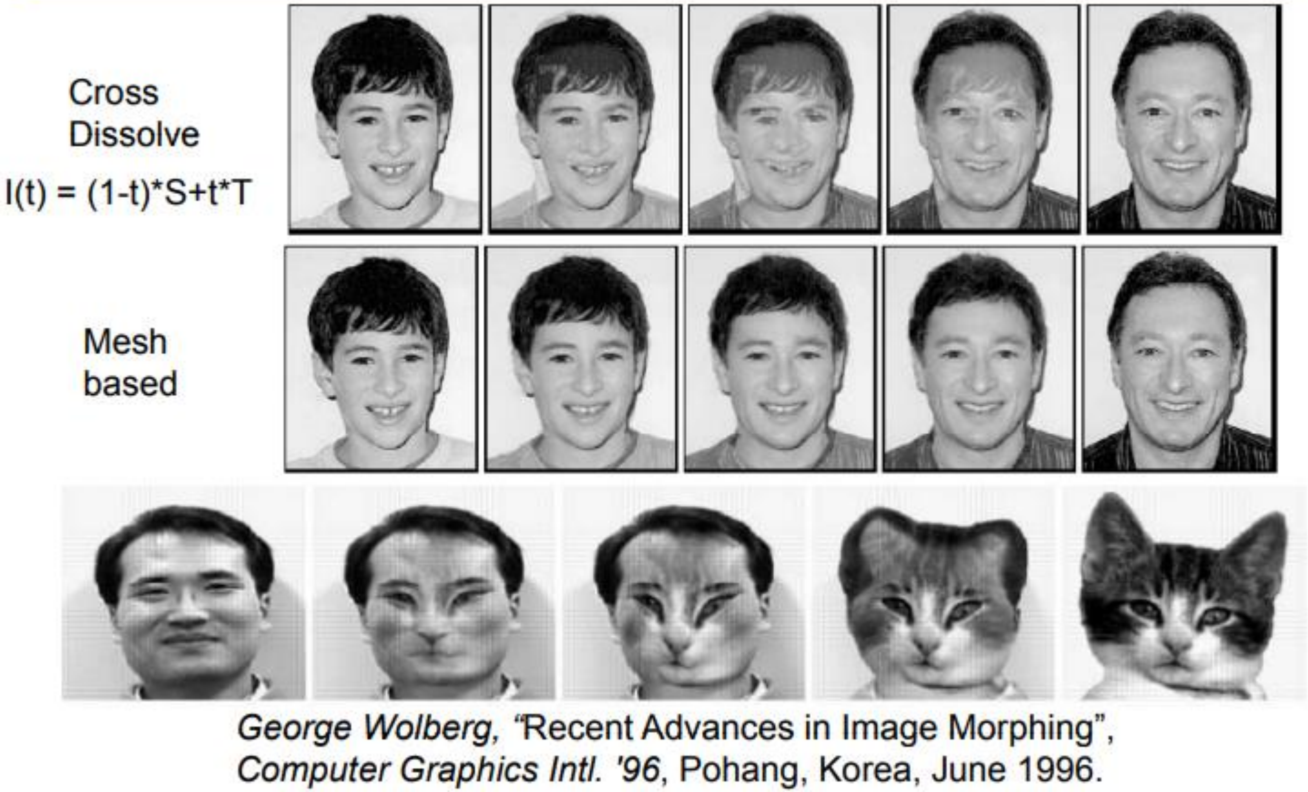
SWIRL

WARP $x(u,v) = sign(u - x_0) * (u - x_0)^2 / x_0 + x_0; y(u,v) = v$

SWIRL
$$x(u,v) = (u - x_0)\cos(\theta) + (v - y_0)\sin(\theta) + x_0;$$
$$y(u,v) = -(u - x_0)\sin(\theta) + (v - y_0)\cos(\theta) + y_0;$$
$$r = ((u - x_0)^2 + (v - y_0)^2)^{1/2}, \theta = \pi r / 512.$$

By Onur Guleyuz

Examples of Image Morphing



http://eeweb.poly.edu/~yao/EL5123/lecture12_ImageWarping.pdf

- Given: T (transformation)
- Determine: Effect of T on source image I, get target image O
- Variant:
  - Given : Source image I, Target image O
  - Determine : Transformation T

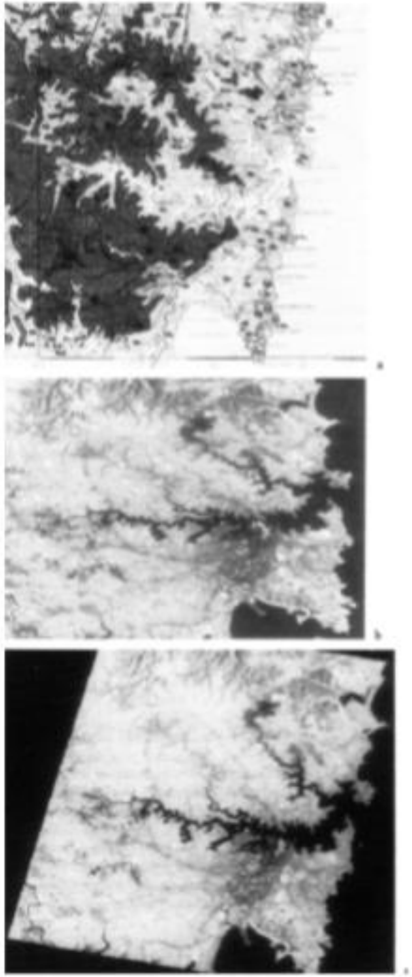


Fig. 4.10: *Image registration. (a) Map; (b) Landsat MSS image to be registered; (c) Landsat image registered to map using 2nd order polynomials (Fig. 2.16 from Richards, 1986)*

Image Registration

Image Courtesy of mPower3/Emerge

Aerial Photo Image

Image Courtesy of MassGIS

Orthophoto Image

https://in.mathworks.com/help/images/register-an-aerial-photograph-to-a-digital-orthophoto.html

Matlab Functions

- T = MAKETFORM('affine',U,X) builds a TFORM struct for a
- two-dimensional affine transformation that maps each row of U
- to the corresponding row of X. U and X are each 3-by-2 and
- define the corners of input and output triangles. The corners
- may not be collinear.
- Example
- --------
- Create an affine transformation that maps the triangle with vertices
- (0,0), (6,3), (-2,5) to the triangle with vertices (-1,-1), (0,-10),
- (4,4):
-
- u = [0 6 -2]';
- v = [0 3 5]';
- x = [-1 0 4]';
- y = [-1 -10 4]';
- tform = maketform('affine',[u v],[x y]);

- B = IMTRANSFORM(A,TFORM, INTERP) transforms the image A according to the 2-D spatial transformation defined by TFORMB; INTERP specifies the interpolation filter
- Example 1
- ---------
- Apply a horizontal shear to an intensity image.
-
- I = imread('cameraman.tif');
- tform = maketform('affine',[1 0 0; .5 1 0; 0 0 1]);
- J = imtransform(I,tform);
- figure, imshow(I), figure, imshow(J)

tform = maketform('affine',[1 0 0; .5 1 0; 0 0 1]);
In MATLAB, 'affine' transform is defined by:
[a1,b1,0;a2,b2,0;a0,b0,1]

With notation used in this lecture note

$$\mathbf{A} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

MATLAB function: cp2tform()

TFORM=CP2TFORM(INPUT_POINTS,BASE_POINTS,TRANSFORM TYPE)

- returns a TFORM structure containing a spatial transformation.
- INPUT_POINTS is an M-by-2 double matrix containing the X and Y coordinates of control points in the image you want to transform.
- BASE_POINTS is an M-by-2 double matrix containing the X and Y coordinates of control points in the base image.
- TRANSFORMTYPE can be 'nonreflective similarity', 'similarity', 'affine', 'projective', 'polynomial', 'piecewise linear' or 'lwm'.

Geometric Operations

- Some uses
  - Correct distortions introduced during imaging
  - Transformation: To create special effects (e.g. morphing)
  - Registration: Register two images taken of the same scene at different times/conditions

References

- https://in.mathworks.com/help/images/ref/fitgeotrans.html
- https://in.mathworks.com/discovery/image-registration.html

- http://eeweb.poly.edu/~yao/EL5123/lecture12_ImageWarping.pdf

References

- G&W textbook
  - 2.4.4. Image Interpolation
  - 2.6.5. Geometrical spatial transforms and image registration
  - 4.5.4. Aliasing