# Geometric Operations and Image Registration

CMPUT 206: Introduction to Digital Image Processing

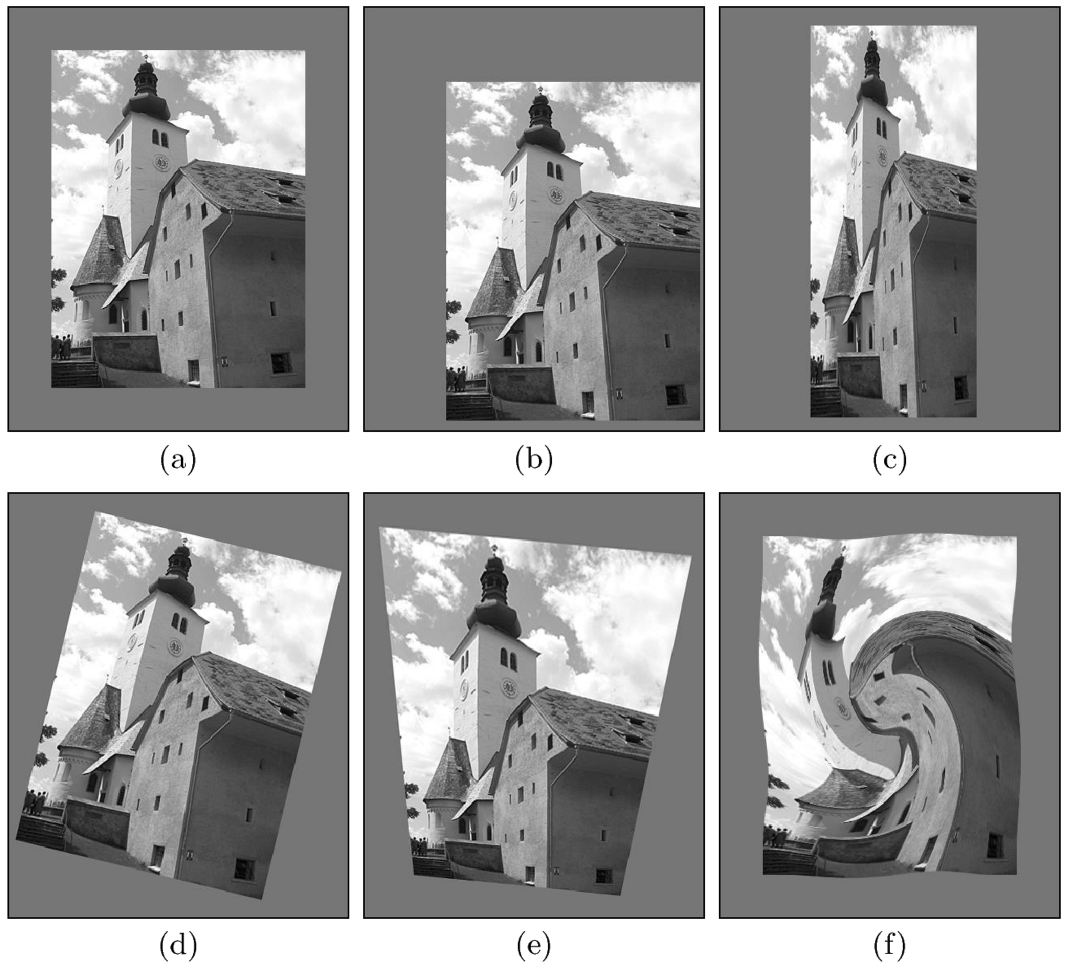
Nilanjan Ray

Source: http://www.imagingbook.com/

# Geometric Operations

- Let’s start with some examples
  - Image rotations, translations, scaling are some of the geometric operations

# Geometric Operations: Some Examples




(a) Original image, (b) translation, (c) scaling, (d) rotation,
(e) projective transformation, (f) nonlinear distortion

# Geometric Operations…

- Can you guess from these examples, in what ways geometric operations are fundamentally different from filtering and point operations?

# GO: Formal Descriptions

- A geometric operation (GO) transforms a given image $I$ to a new image $I'$, *i.e.*, the intensity value at $(x, y)$ moves to $(x', y')$ in the new image:

$$I(x, y) \rightarrow I'(x', y')$$

- To model this process, we need a mapping function $T$ that specifies for each original 2D coordinate point $\mathbf{x} = (x, y)$, the corresponding target point $\mathbf{x}' = (x', y')$:

$$T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\boldsymbol{x}' = T(\boldsymbol{x})$$

# 2D Mapping Functions

- Simple mappings, such as translation, rotation, …
- Homogeneous coordinates
- Affine mapping
- Projective mapping
- Bilinear mapping
- Nonlinear mapping

# Simple Mappings

- Translation (shift) by a vector ($d_x$, $d_y$):

$$\begin{aligned} T_x &: x' = x + d_x \\ T_y &: y' = y + d_y \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

- Scaling along the $x$ or $y$ axis by the factor $s_x$ or $s_y$, respectively:

$$\begin{aligned} T_x &: x' = s_x \cdot x \\ T_y &: y' = s_y \cdot y \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

- Shearing along the $x$ and $y$ axis by the factor $b_x$ and $b_y$, respectively:

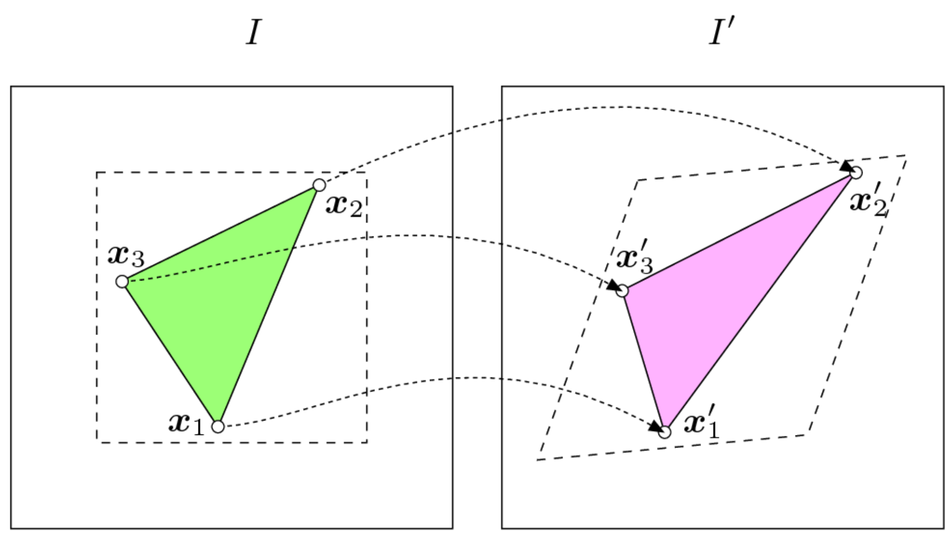
$$\begin{aligned} T_x &: x' = x + b_x \cdot y \\ T_y &: y' = y + b_y \cdot x \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

- Rotation by an angle $\alpha$ (coordinate origin is the center of rotation):

$$\begin{aligned} T_x &: x' = x \cdot \cos\alpha - y \cdot \sin\alpha \\ T_y &: y' = x \cdot \sin\alpha + y \cdot \cos\alpha \end{aligned} \qquad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

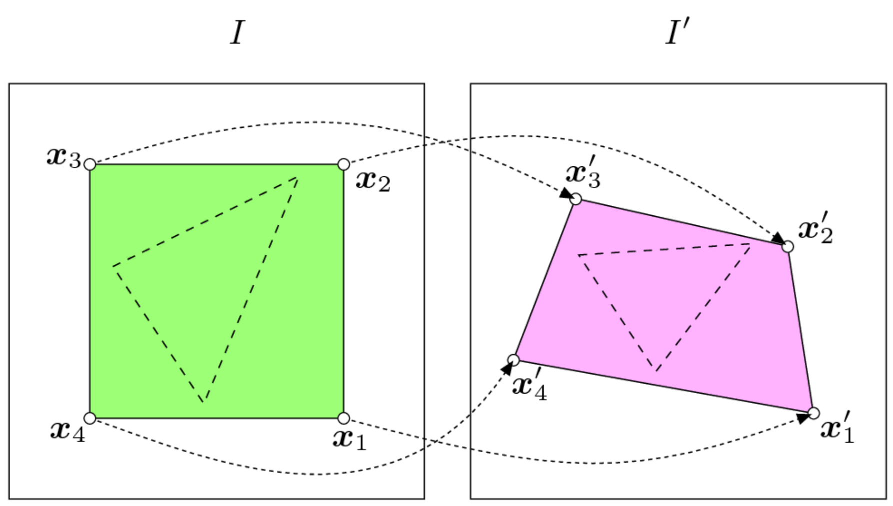
# Affine Mapping

Preserves parallel lines.

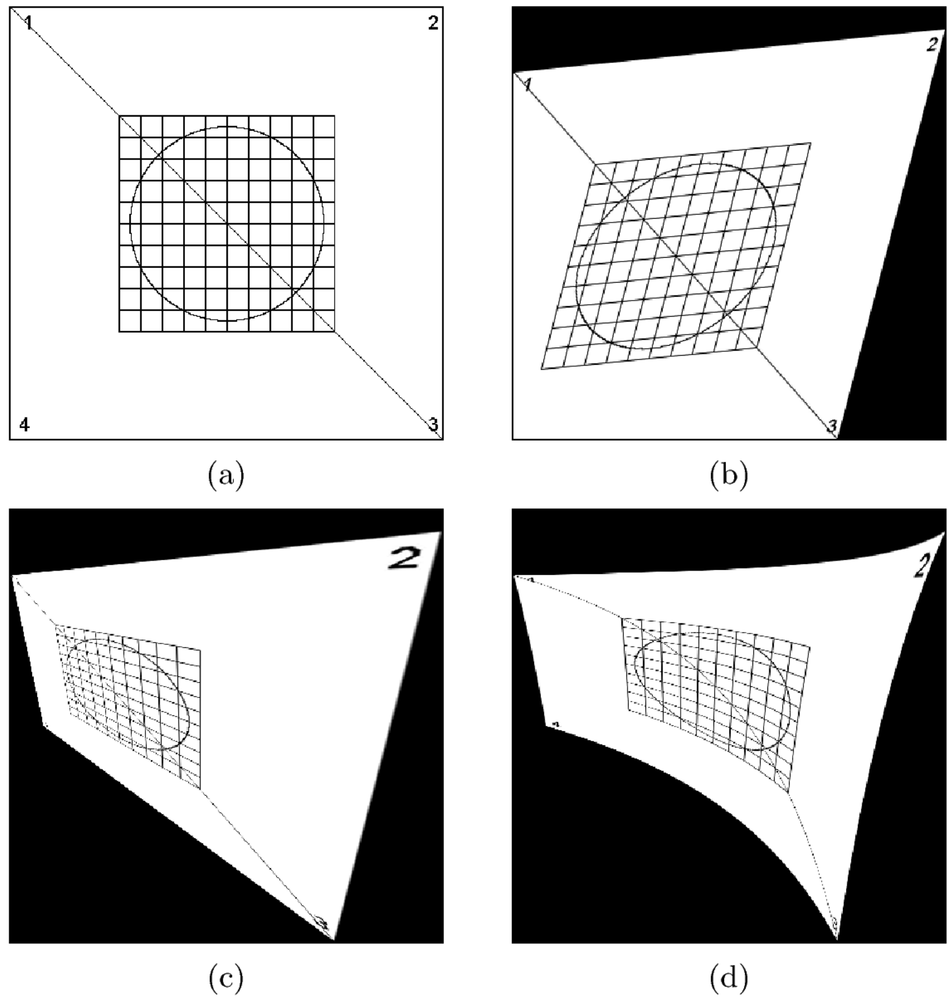


# Projective Mapping

No longer preserves parallel lines

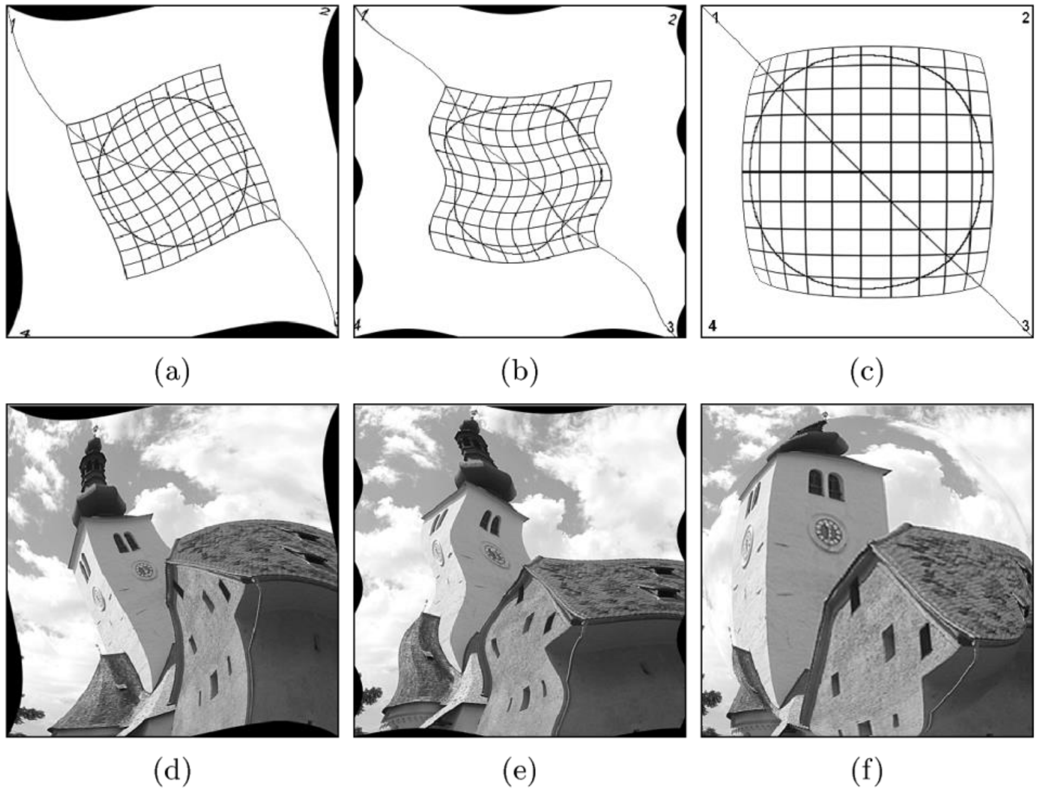


# Various GOs




(a) Original image (b) Affine Transform (c) Projective transform (d) Bilinear transform

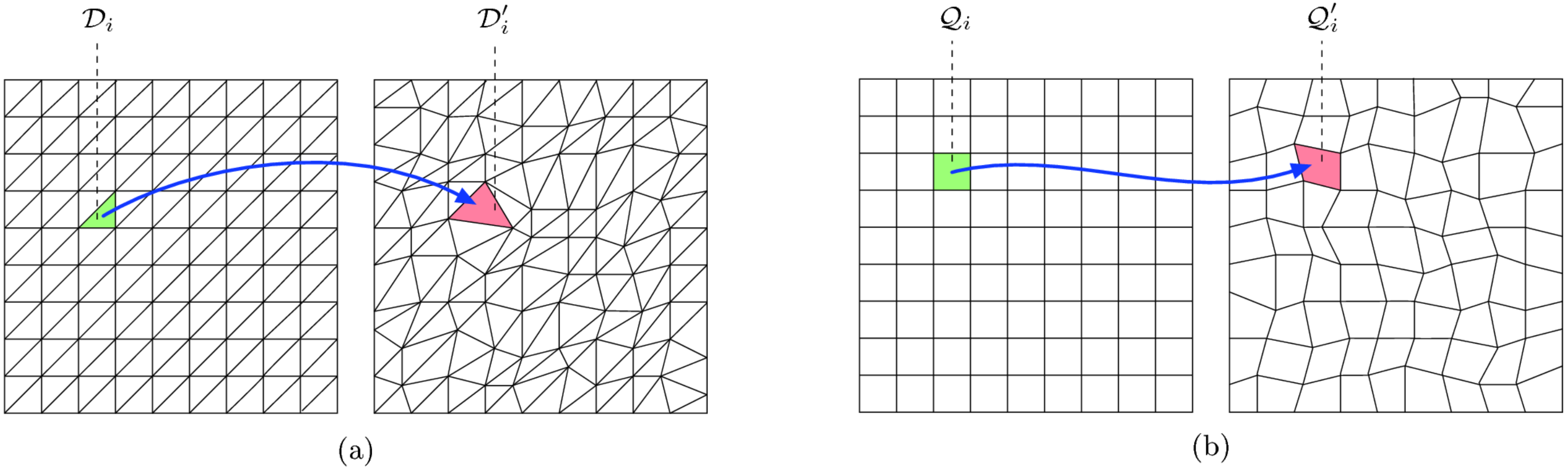
# Nonlinear GOs




(a), (d) Twirl; (b), (e) Ripple; (c), (f) Sphere transformations

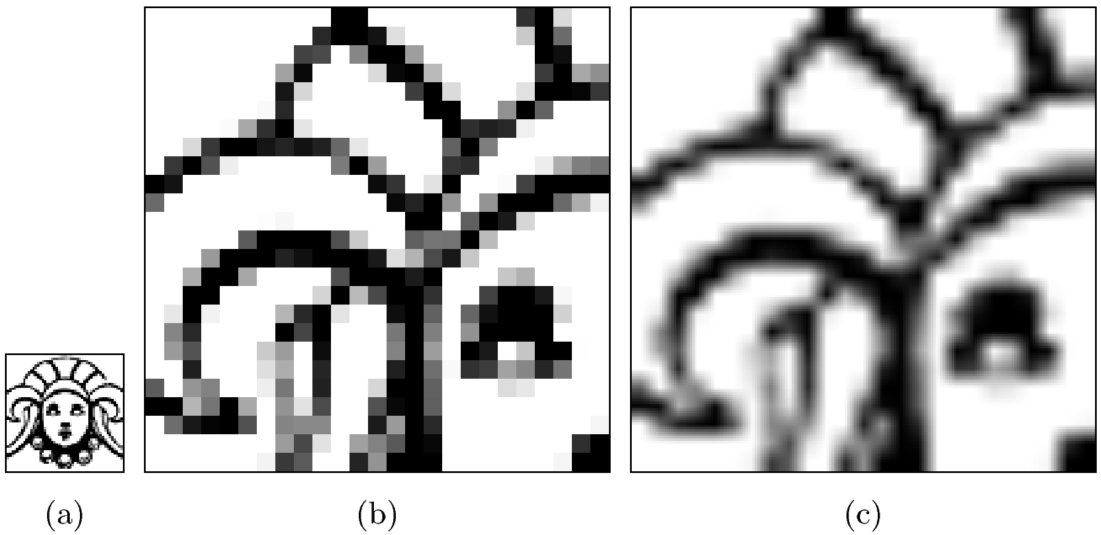
# Local Image Transformations

These mappings are "local GOs"




Examples: Mesh partitioning

# Image Enlargement



(a) Original image (b) 8x magnification by nearest neighbor method
(c) bilinear interpolation

# Image Registration

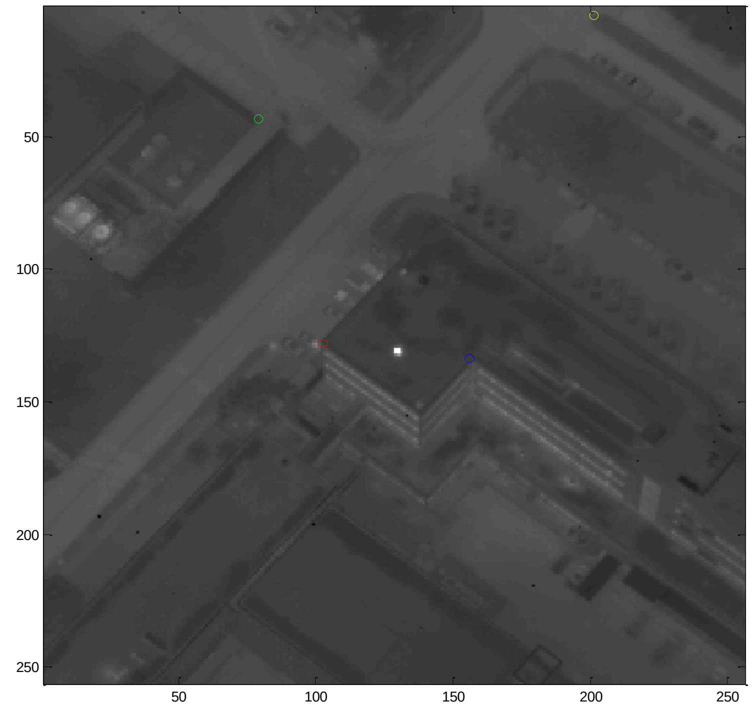
- Loosely speaking image registration brings two images acquired from the same/similar scene to the same coordinate frame.
- Image registration is based on geometric transform.
- Interactive image registration demo.

# Register Two Images by Affine Transform

- Step 1: Collect corresponding point pairs from two images.
- Step 2: Solve for six affine parameters from linear equations involving corresponding points.
- Step 3: Warp one image onto another using six affine transformation parameters.

# Corresponding Points


First image




Second image

You need at least three corresponding point pairs for affine image registration. Why?

# Solving for Affine Transform

- Let $\{(x_i, y_i)\}_{i=1}^{N}$ and $\{(\acute{x}_i, \acute{y}_i)\}_{i=1}^{N}$ be corresponding point pairs that came from the second and the first images, respectively.
- We can write following 2*N* linear equations:

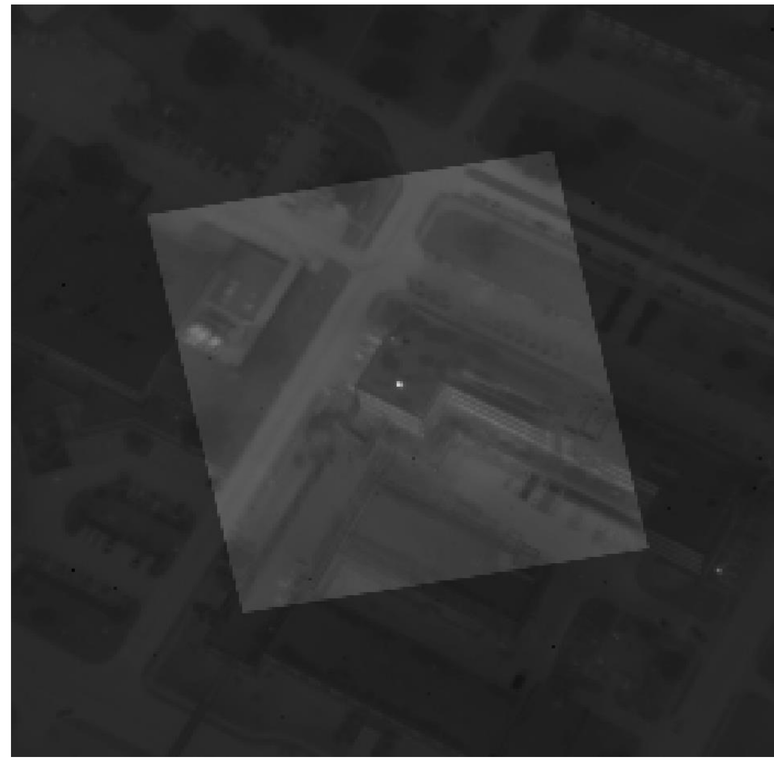
$$\acute{x}_i = ax_i + by_i + c$$

$$\acute{y}_i = dx_i + ey_i + f, \qquad i = 1, \ldots, N$$

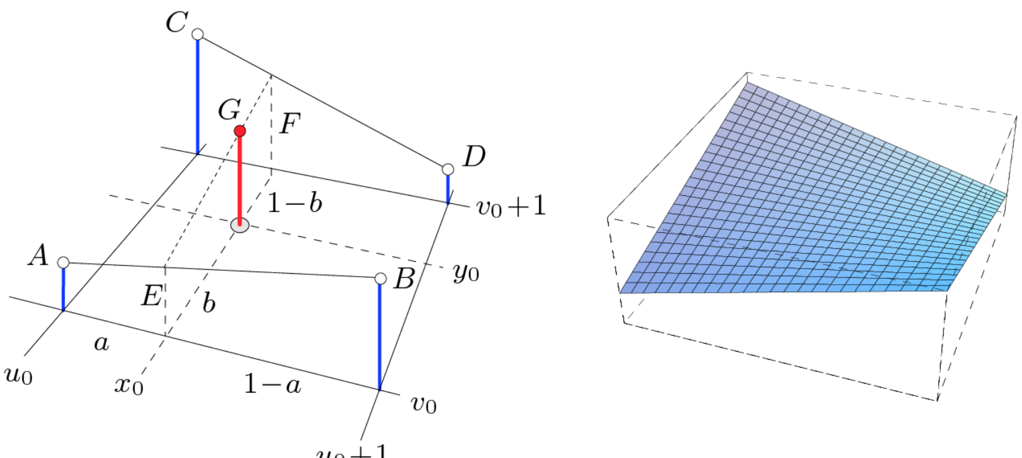
- Solve for six unknown *a*, *b*, *c*, *d*, *e* and *f*. These are six affine parameters.

# Warping

- Transform coordinates of pixel locations of the second image applying affine transform.
- Interpolate pixel values of the second image in the new coordinate system.
- Overlay two images.



# Bilinear Interpolation




(a) (b)

Image intensity values at 4 corners of a rectangle are given.

You need to estimate intensity value anywhere within the rectangle