# Lecture 5
# Geometric Transformations and Image Registration

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2014

# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image registration

# Transforming Points

- Geometric transformations modify the spatial relationship between pixels in an image
- The images can be shifted, rotated, or stretched in a variety of ways
- Geometric transformations can be used to
  - create thumbnail views
  - change digital video resolution
  - correct distortions caused by viewing geometry
  - align multiple images of the same scene

# Transforming Points

Suppose $(w, z)$ and $(x, y)$ are two spatial coordinate systems

***input space*** ***output space***

A geometric transformation $T$ that maps the input space to output space

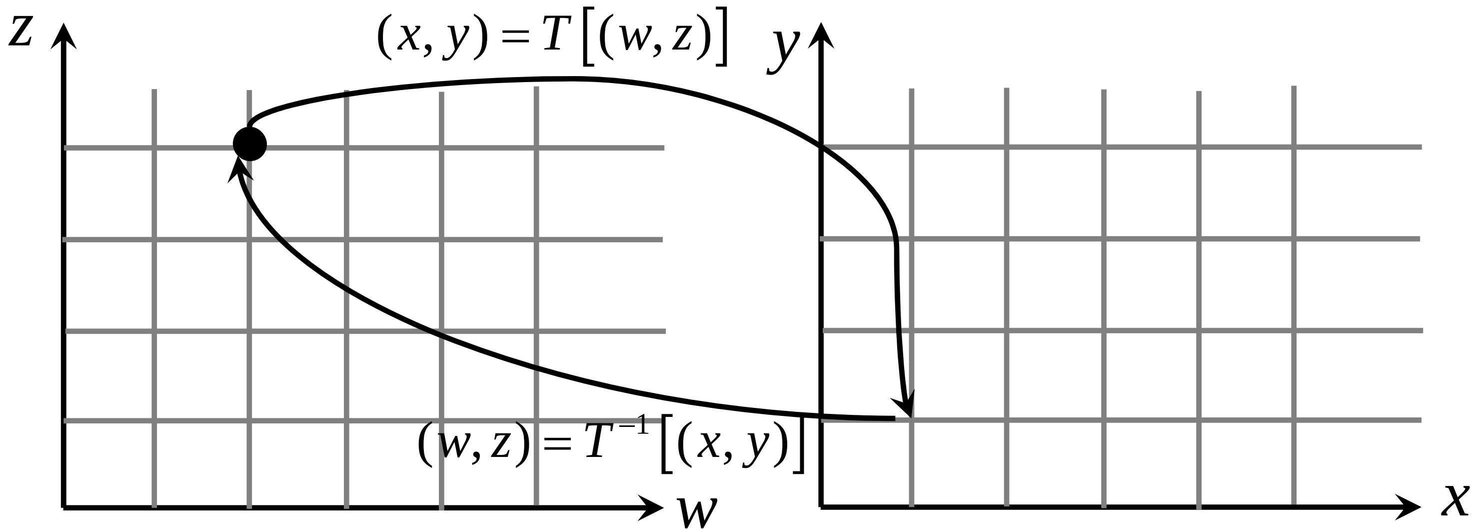
$$(x, y) = T\left[(w, z)\right]$$

$T$ is called a ***forward transformation*** or ***forward mapping***

$$(w, z) = T^{-1}\left[(x, y)\right]$$

$T^{-1}$ is called a ***inverse transformation*** or ***inverse mapping***

# Transforming Points



# Transforming Points

An example

$$(x, y) = T\left[(w, z)\right] = (w/2, z/2)$$

$$(w, z) = T^{-1}\left[(x, y)\right] = (2x, 2y)$$

# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image Registration

# Hierarchy of Geometric Transformations

- Class I: Isometry transformation

If only rotation and translation are considered

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

In homogeneous coordinates

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_1 \\ \sin\theta & \cos\theta & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(More concise!)

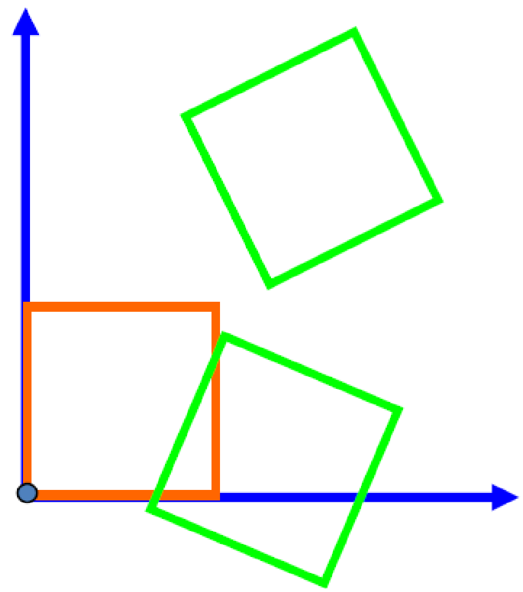
# Hierarchy of Geometric Transformations

- Class I: Isometry transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$

Properties

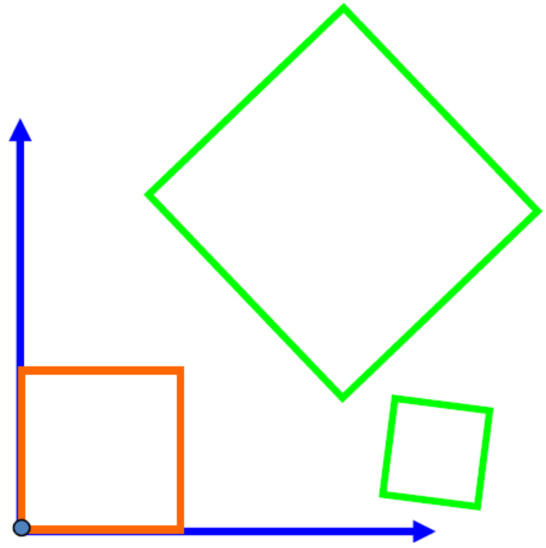
- $\mathbf{R}$ is an orthogonal matrix
- Euclidean distance is preserved
- Has three degrees of freedom; two for translation, and one for rotation



# Hierarchy of Geometric Transformations

- Class II: Similarity transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_1 \\ s\sin\theta & s\cos\theta & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$



# Hierarchy of Geometric Transformations

- Class II: Similarity transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_1 \\ s\sin\theta & s\cos\theta & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$

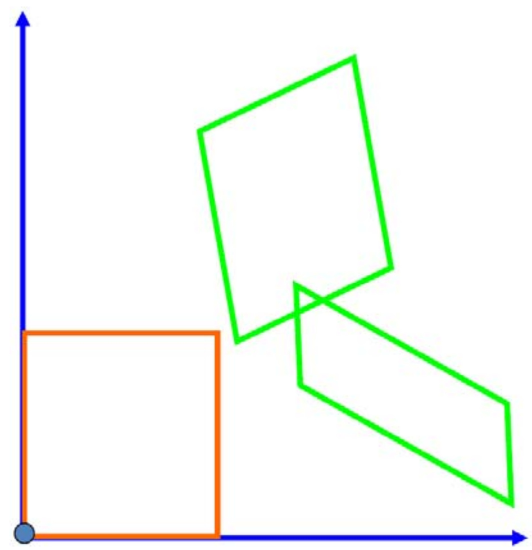
Properties

- $\mathbf{R}$ is an orthogonal matrix
- Similarity ratio (the ratio of two lengths) is preserved
- Has four degrees of freedom; two for translation, one for rotation, and one for scaling

# Hierarchy of Geometric Transformations

- Class III: Affine transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$



# Hierarchy of Geometric Transformations

- Class III: Affine transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$

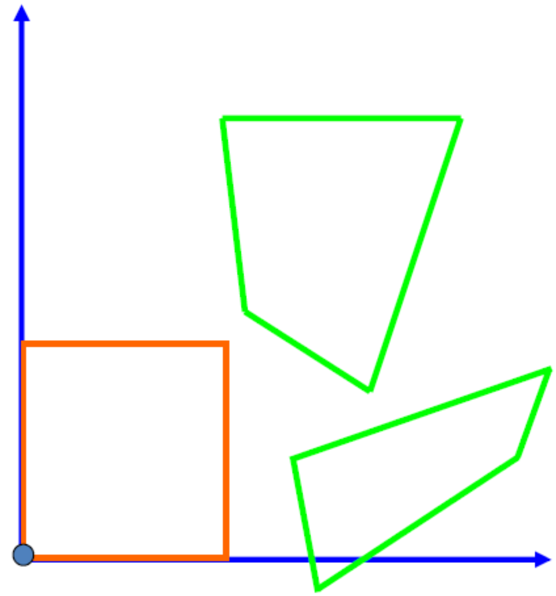
Properties

- $\mathbf{A}$ is a non-singular matrix
- Ratio of lengths of parallel line segments is preserved
- Has six degrees of freedom; two for translation, one for rotation, one for scaling, one for scaling direction, and one for scaling ratio

# Hierarchy of Geometric Transformations

- Class IV: Projective transformation

$$c\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



# Hierarchy of Geometric Transformations

- Class IV: Projective transformation

$$c\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Also referred to as homography matrix

Properties

- Cross ratio preserved
- Though it has 9 parameters, it has **8** degrees of freedom, since only the ratio is important in the homogeneous coordinates

# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image Registration

# Applying Geometric Transformations to Images

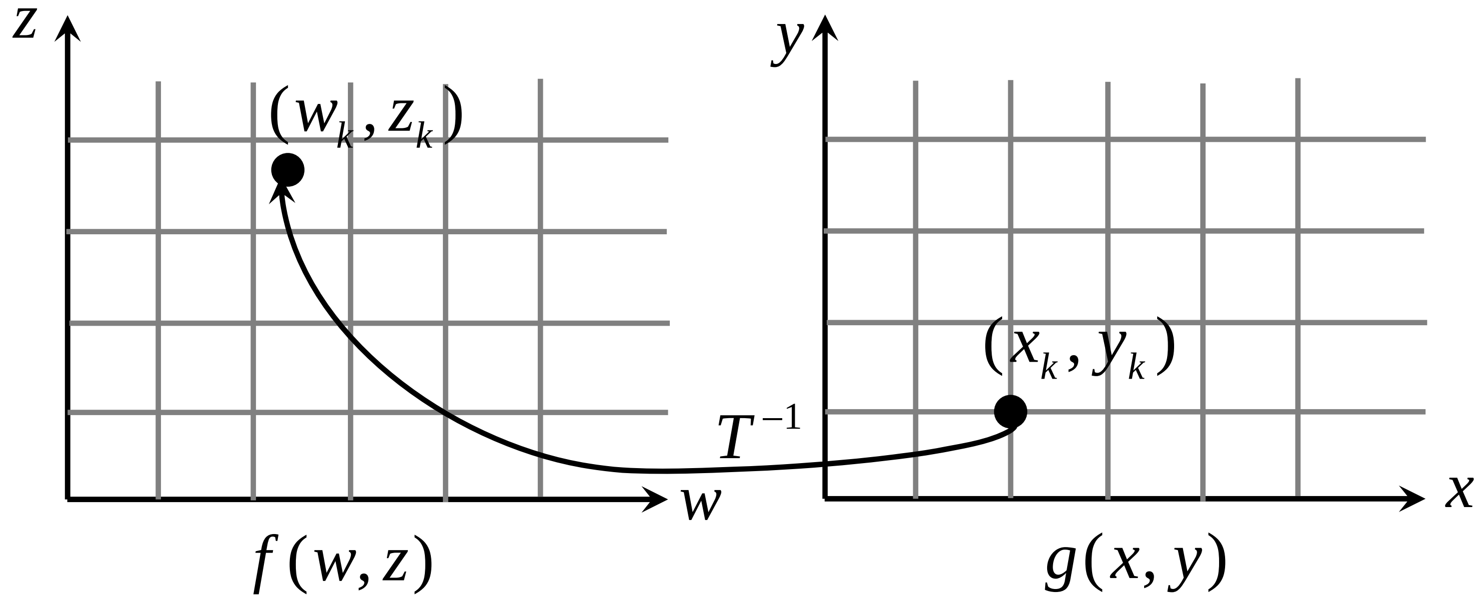
Given the image $f$, apply $T$ to $f$ to get $g$, how to get $g$?

The procedure for computing the output pixel at location $(x_k, y_k)$ is

- Evaluate $(w_k, z_k) = T^{-1}[(x_k, y_k)]$
- Evaluate $f(w_k, z_k)$
- $g(x_k, y_k) = f(w_k, z_k)$

# Applying Geometric Transformations to Images

- Notes on interpolation
  - Even if $(x_k, y_k)$ are integers, in most cases $(w_k, z_k)$ are not
  - For digital images, the values of $f$ are known only at integer-valued locations
  - Using these known values to evaluate $f$ at non-integer valued locations is called as ***interpolation***



# Applying Geometric Transformations to Images

- Notes on interpolation
  - In Matlab, three commonly used interpolation schemes are built-in, including nearest neighborhood, bilinear, and bicubic
  - For most Matlab routines where interpolation is required, “bilinear” is the default

# Applying Geometric Transformations to Images

- Matlab implementation
  - “Maketform” is used to construct a geometric transformation structure
  - “imtransform” transforms the image according to the 2-D spatial transformation defined by tform

Note: in Matlab, geometric transformations are expressed as

$$\left(x' \; y' \; 1\right) = \left(x \; y \; 1\right)\mathbf{A}$$

where $\mathbf{A}$ is a 3 by 3 transformation matrix

# Applying Geometric Transformations to Images

- Matlab implementation

An example

```
im = imread('tongji.bmp');
theta = pi/6;
rotationMatrix = [cos(theta) sin(theta) 0;-sin(theta) cos(theta) 0;0 0 1];
tformRotation = maketform('affine',rotationMatrix);

rotatedIm = imtransform(im, tformRotation,'FillValues',255);
figure;
subplot(1,2,1); imshow(rotatedIm,[]);

rotatedIm = imtransform(im, tformRotation,'FillValues',0);
subplot(1,2,2); imshow(rotatedIm,[]);
```

# Applying Geometric Transformations to Images

- Matlab implementation

An example

original image


rotated images

# Applying Geometric Transformations to Images

- Matlab implementation

  Another example

original image


affine transformed images

# Applying Geometric Transformations to Images

- Output image with location specified
  - This is useful when we want to display the original image and the transformed image on the same figure

In Matlab, this is accomplished by

imshow(image, 'XData', xVector, 'YData', yVector)

'XData' and 'YData' can be obtained by imtransform

# Applying Geometric Transformations to Images

- Output image with location specified

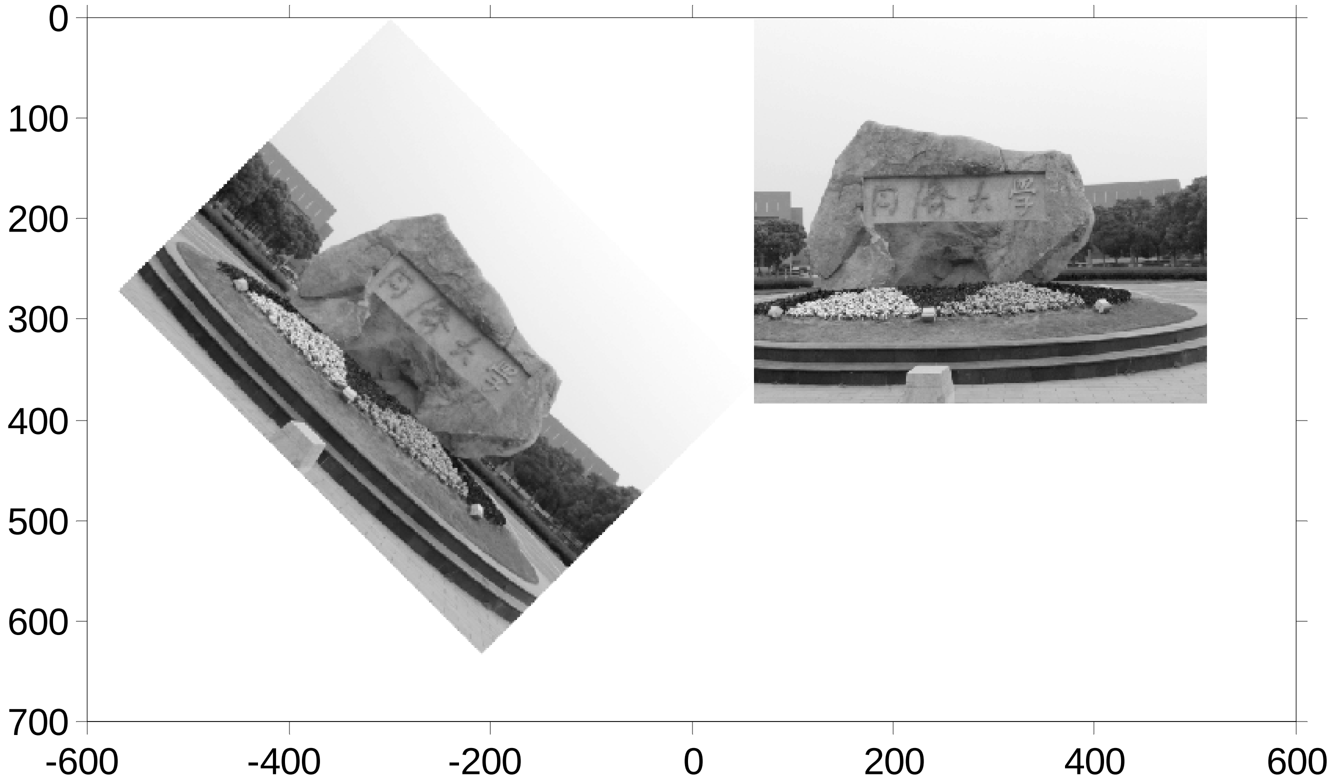
An example

```
im = imread('tongji.bmp');

theta = pi/4;
affineMatrix = [cos(theta) sin(theta) 0;-sin(theta) cos(theta) 0;-300 0 1];

tformAffine = maketform('affine',affineMatrix);

[affineIm, XData, YData] = imtransform(im, tformAffine,'FillValues',255);
figure; imshow(im,[]);
hold on
imshow(affineIm,[],'XData',XData,'YData',YData);
axis auto
axis on
```

# Applying Geometric Transformations to Images

- Output image with location specified

An example




Display the original image and the transformed image in the same coordinate system

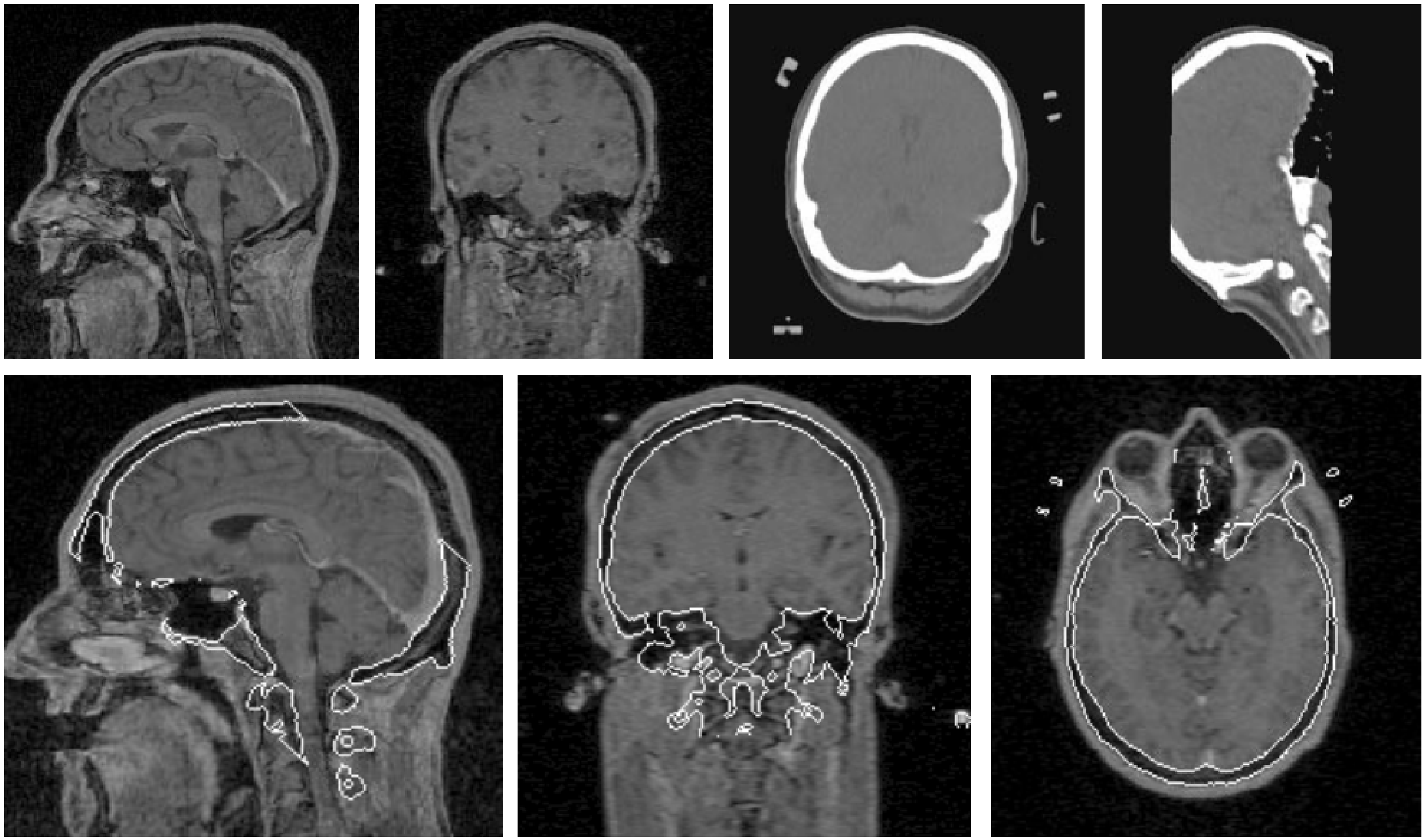
# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image Registration
    - Background
    - A manual method

# Background

- One of the most important applications of geometric transformations is image registration
- Image registration seeks to align images taken in different times, or taken from different modalities
- Image registration has applications especially in
  - Medicine
  - Remote sensing
  - Entertainment

# Background—Example, CT and MRI Registration



Top row: unregistered MR (left) and CT (right) images

Bottom row: MR images in sagittal, coronal and axial planes with the outline of bone, thresholded from the registered CT scan, overlaid

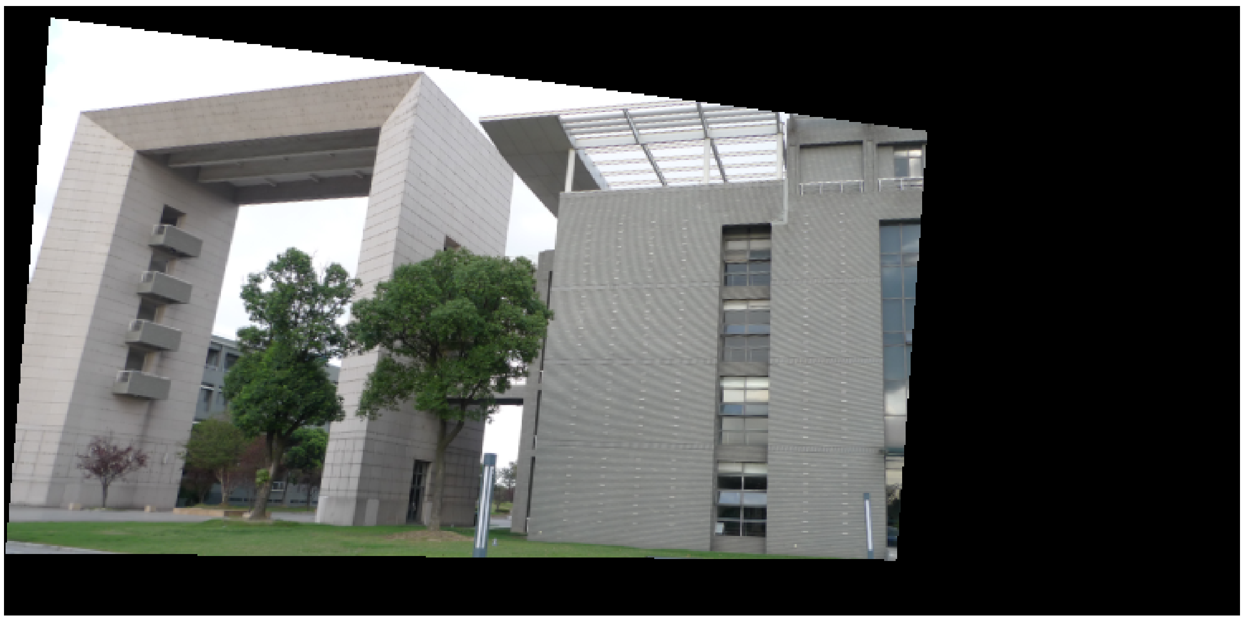
# Background—Example, Panorama Stitching

image 1

image 2

Two images, sharing some objects

# Background—Example, Panorama Stitching



Transform image 1 into the same coordinate system of image 2

# Background—Example, Panorama Stitching

Finally, stitch the transformed image 1 with image 2 to get the panorama

# Background

- The basic registration process
  - Detect features
  - Match corresponding features
  - Infer geometric transformation
  - Use the geometric transformation to align one image with the other
- Image registration can be manual or automatic depending on whether feature detection and matching is human-assisted or performed using an automatic algorithm

# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image Registration
  - Background
  - A manual method
  - Other methods

# A Manual Method

We illustrate this method by using an example, which registers the following two images

Base image

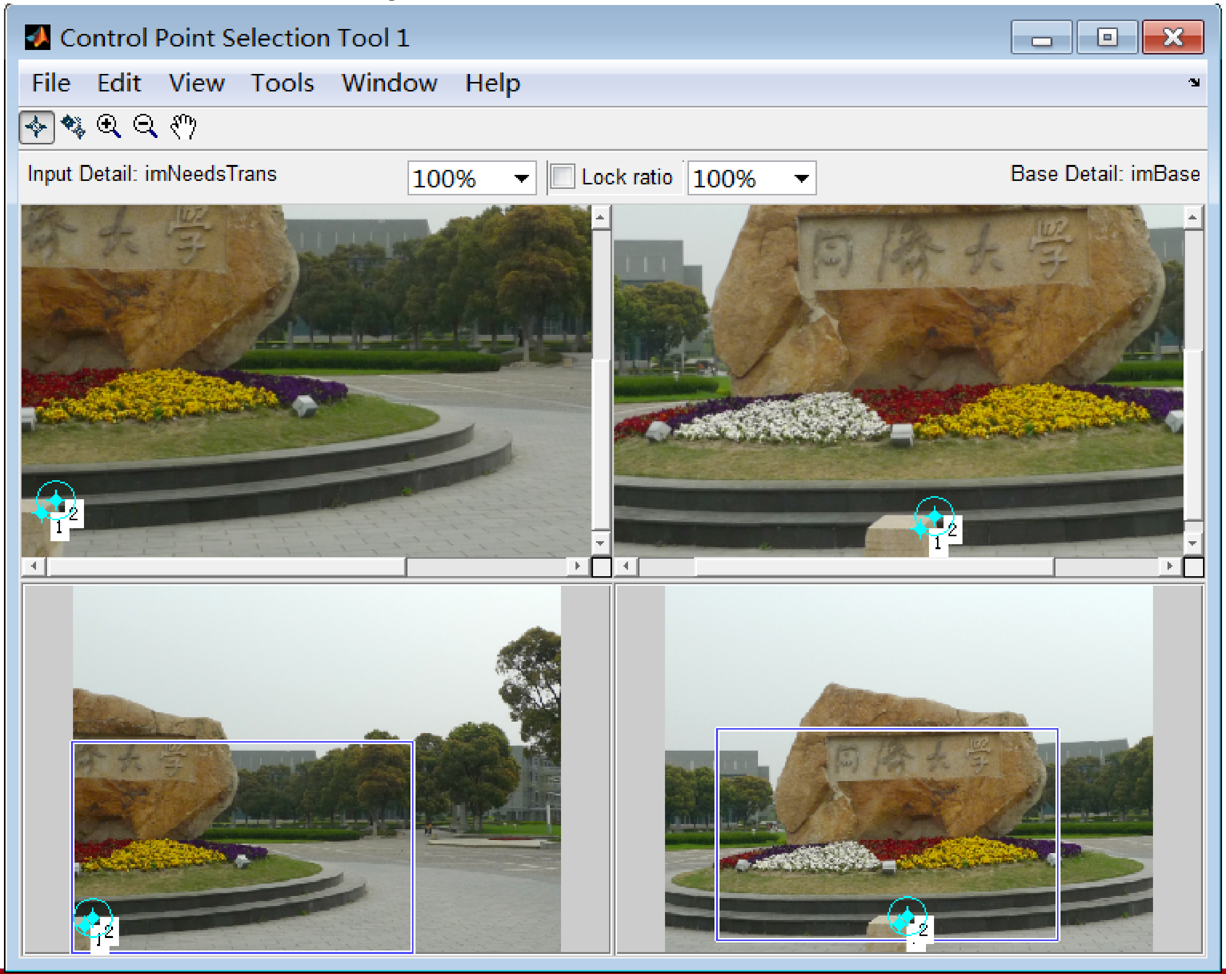
input image which needs to be registered to the base image

# A Manual Method

- Step 1: Manual feature selection and matching using "cpselect" (control points selection)
  - "cpselect" is a GUI tool for manually selecting and matching corresponding control points in a pair of images to be registered

# A Manual Method

- Step 1: Manual feature selection and matching using "cpselect" (control points selection)



# A Manual Method

- Step 2: Inferring transformation parameters using "cp2tform"
  - "cp2tform" can infer geometric transformation parameters from set of feature pairs

```
tform = cp2tform(input_points, base_points, transformtype)
```

The arguments input_points and base_points are both $P \times 2$ matrices containing correponding feature locations

# A Manual Method

- Step 3: Use the geometric transformation to align one image with the other
  - In Matlab, this is achieved by “imtransform”



Two images are registered

# Contents

- Transforming points
- Hierarchy of geometric transformations
- Applying geometric transformations to images
- Image Registration
  - Background
  - A manual method
  - Other methods

# Area-based Registration

- Area-based registration
  - A “template image” is shifted to cover each location in the base image
  - At each location, an area-based similarity is computed
  - The template is said to be a match at a particular position in the base image if a distinct peak in the similarity metric is found at that position

# Area-based Registration

- Area-based registration

A commonly used area-based similarity metric is the correlation coefficient

$$\gamma(x,y)=\frac{\sum_{s,t}\left[w(s,t)-\overline{w}\right]\left[f(x+s,y+t)-\overline{f_{xy}}\right]}{\sqrt{\sum_{s,t}\left[w(s,t)-\overline{w}\right]^2}\sqrt{\sum_{s,t}\left[f(x+s,y+t)-\overline{f_{xy}}\right]^2}}$$

where $w$ is the template image, $\overline{w}$ is the average value of the template, $f$ is the base image, and $\overline{f_{xy}}$ is the average value of the based image in the region where $f$ and $w$ overlap

In Matlab, such a 2D correlation coefficient can be realized by "normxcorr2"

# Area-based Registration

- Limitations of area-based registration
  - Classical area-based registration method can only deal with translation transformation between two images;
  - It will fail if rotation, scaling, or affine transformations exist between the two images

# Automatic Feature-based Registration

- Feature points (sometimes referred as key points or interest points) can be detected automatically
  - Harris corner detector
  - Extrema of LoG
- Feature points descriptors can be performed automatically
  - SIFT (scale invariant feature transform)
- Feature matching can be performed automatically

To know more, come to our another course “Computer Vision”!

Thanks for your attention