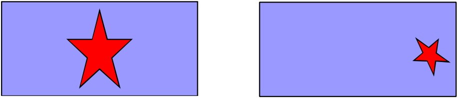
Lecture 11: Image Registration

Image Registration

- Registration is the process of aligning two images so that they share a common coordinate system.



Applications

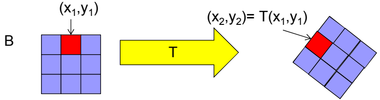
- Some applications of image registration
  - -Medical imaging
  - -Image stitching / panoramic images
  - -Video super-resolution

Image Transformations

- For two images A and B, we would *ideally* want to find a transformation T such that

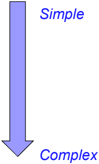
$$A = T(B).$$

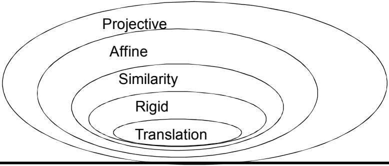
- But in practice, the aligned images would never be exactly equal. So we want A and T(B) to be equivalent in some geometric sense.
- We obtain the transformed image T(B) by mapping every pixel (x,y) to new coordinates T(x,y).



Parametric Transformations

- <u>Translation</u> - translation only
- <u>Rigid</u> - translation, rotation
- <u>Similarity</u> - translation, rotation, scaling
- <u>Affine</u> - translation, rotation, scaling, shear
- <u>Projective / Homography</u> - All of the above plus projective distortion

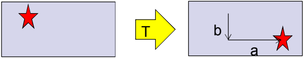




Translation

- Image has only been shifted left/right by amount a and up/down by amount b.

$$x_2 = x_1 + a$$
$$y_2 = y_1 + b$$



- We can write this in matrix form as

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}}_{\mathrm{T}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- The transformation T is a 3x3 matrix that we multiply by the homogeneous coordinates (x,y,1).

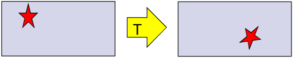
Rigid

- We can rotate a point (x,y) by an angle θ by multiplying it by the rotation matrix:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

- To include both rotation and translation

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & a \\ \sin\theta & \cos\theta & b \\ 0 & 0 & 1 \end{bmatrix}}_{\mathrm{T}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$



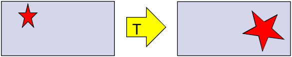
Similarity

- We can scale the (x,y) coordinates by a scaling factor S to make the objects larger or smaller.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

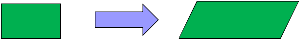
- Including rotation and translation gives

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} S\cos\theta & -S\sin\theta & a \\ S\sin\theta & S\cos\theta & b \\ 0 & 0 & 1 \end{bmatrix}}_{\text{T}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$



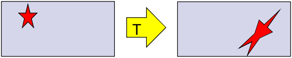
Affine

- An affine transformation allows for shear effects.



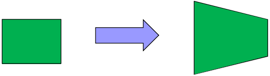
- Under an affine transformation, parallel lines remain parallel.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} A\cos\theta & B\sin\theta & a \\ C\sin\theta & D\cos\theta & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$



Projective

- The projective transformation is the most general linear transformation.
- It is also referred to as a planar homography.
- Under a projective transformation, parallel lines may not stay parallel. This represents perspective distortion caused by the camera converting a 3D world to a 2D image.



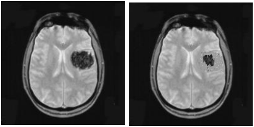
Finding the Transformation

- There are 2 general approaches to finding the transformation T.
- Intensity-based Registration □

  Align the entire images to match up the colors of as many pixels as possible.
- Feature (or land-mark)-based Registration
  - Locate features in each image (called *control points* or *interest points*), then line up the features.

Intensity-based vs. Feature-based

- The approach you use depends on the application.
- <u>Ex</u> Which approach is best for the data below?



Intensity-based Registration

- Intensity-based registration lines up the colors in two images.
- A simple measure of how the gray values line up is given by the Mean Square Error (MSE) or SSD:

$$MSE = \sum_x \sum_y [A(x,y) - B(x,y)]^2$$

- If the images A and B are identical, then MSE=0.

A

3	2
3	8

B

4	2
1	5

MSE =

Intensity-based Registration

- The Matlab command imregister will transform the image B to the reference (fixed) image A by minimizing the MSE.
- First generate the parameters for the registration procedure using imregconfig.

  ```
  [optimizer, metric] = imregconfig('monomodal');
  ```

- You need to specify which type of transformation relates the images: Translation, Rigid, Similarity, or Affine.

```
C = imregister(B, A, 'Affine', optimizer, metric);
```

Ouput image Transformed B

Image to transform

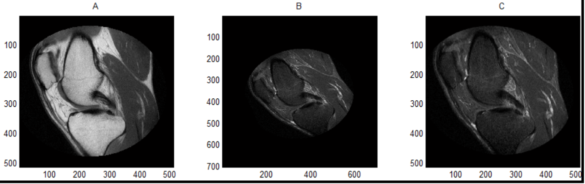
Fixed reference image

Transformation type

Intensity-based Registration

- If your gray values are different, such as in multimodal MRI, we might be able to align the images e.g., using the <u>Mutual Information</u> metric rather than MSE.
- Mutual Information is a measure of how the image histograms match up, based on Shannon Entropy.

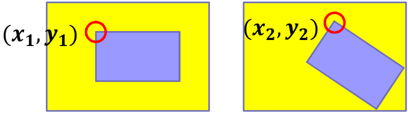
```
[optimizer,metric]=imregconfig('multimodal');
C = imregister(B, A, 'Similarity', optimizer, metric);
```



Feature-based Registration

- If we can detect corresponding control points in two images, we can align the images by a ***planar homography*** H.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$



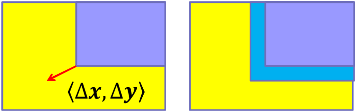
Manual Selection

- The Matlab command cpselect (Control Point Selection) brings up a GUI that displays the two images side-by-side.
- The user clicks corresponding control points in the images.

Automatic Selection

- A common feature to look for is the corners of objects.
- *(Harris-Stephens, 1988)* proposed looking at the change in pixel intensity when we shift an image slightly.

$$c(x,y) = \sum_{W} [I(x_i, y_i) - I(x_i + \triangle x, y_i + \triangle y)]^2$$



- At a corner point, we expect to see a large change.

Detecting Corners

A Taylor series approximation of the shifted image is

$$I(x_i + \triangle x, y_i + \triangle y) \approx I(x_i, y_i) + [I_x(x_i, y_i) \; I_y(x_i, y_i)] \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix}$$

$$\begin{aligned}
c(x,y) &= \sum_W [I(x_i, y_i) - I(x_i + \triangle x, y_i + \triangle y)]^2 \\
&= \sum_W \left(I(x_i, y_i) - I(x_i, y_i) - [I_x(x_i, y_i) \; I_y(x_i, y_i)] \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix} \right)^2 \\
&= \sum_W \left(-[I_x(x_i, y_i) \; I_y(x_i, y_i)] \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix} \right)^2 \\
&= \sum_W \left([I_x(x_i, y_i) \; I_y(x_i, y_i)] \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix} \right)^2 \\
&= [\triangle x \; \triangle y] \begin{bmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix} \\
&= [\triangle x \; \triangle y] C(x,y) \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix}
\end{aligned}$$

(Konstantinos, 2004)

Eigenvectors

- Consider a quadratic form $\vec{x}^T A \vec{x}$ where $\vec{x}$ is a unit vector.
- What $\vec{x}$ will maximize/minimize $\vec{x}^T A \vec{x}$?

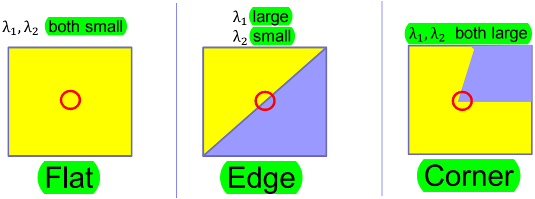
- Look at the eigenvectors!
- A 2x2 matrix A has 2 eigenvectors:

$$\vec{v}_1,\ \vec{v}_2 \text{ with } \lambda_1 > \lambda_2.$$

  - Direction of Maximum Change: $\vec{v}_1$
  - Direction of Minimum Change: $\vec{v}_2$

Eigenvalues

- For each image patch, compute the matrix C and find its eigenvalues.
- There are 3 possible cases.



Corner Points

- The corner points are the pixels where the correlation matrix C has two large eigenvalues.
- The Matlab command corner returns the N strongest corner points as a Nx2 matrix.

Image

corner points to find

```
P = corner(A, 100);
imshow(A);
hold on
plot(P(:,1), P(:,2), 'ro');
```

Computing the Transformation

- Given two sets of corresponding control points, we can find the transformation between the points using the command cp2tform *(control points to transformation)*.

T = cp2tform(P_B, P_A, 'affine');

(x,y) control points in transformed image B

(x,y) control points in fixed image A

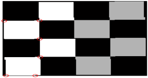
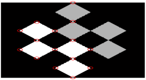
Transformation type, e.g. 'similarity', 'projective', 'polynomial'

- We can then apply this transformation T to the image B.
- The new image C will align with the fixed image A.

C = imtransform(B, T);

Control Point Matching

- But how do we know which control points correspond to the control points in the other image?
- Even the number of detected control points could be different.

6 Control Points   15 Control Points

- Random Sample Consensus (RANSAC) looks at all possible transformations generated.
- RANSAC selects the transformation that best matches the collection of points.
- RANSAC can be *very* slow.