2D Geometric Transformations

CS-477 Computer Vision

Dr. Mohsin Kamal
Associate Professor
dr.mohsinkamal@seecs.edu.pk

School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Pakistan

Helpful material

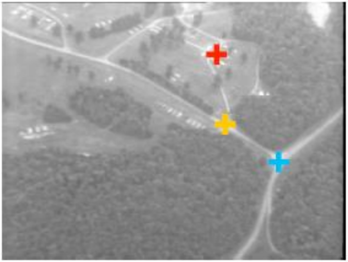
- https://www.algorithm-archive.org/contents/affine_transformations/affine_transformations.html
- https://www.youtube.com/watch?v=E3Phj6J287o

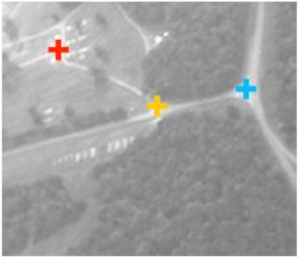
1. Recovering best affine transformation
2. 2D affine warping
3. Image interpolation

- Given two images with unknown transformation between them...

- Compute the values for $[a_1, \cdots, a_6]$

- Input: we are given some correspondences
- Output: Compute $a_1 - a_6$ which relate the images

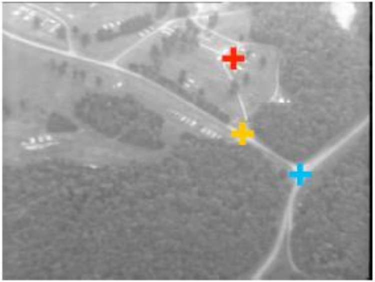


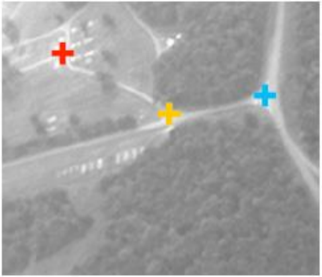


- This is an optimization problem. Find the 'best' set of parameters, given the input data

Parameter Optimization: Least Squared Error Solutions

- Input: Set of correspondences
  - Image 1: (x_i, y_i)
  - Image 2: (x_i', y_i')





Least Squared Error Solutions

- Find the solution (i.e. set of parameters $a_1, \cdots, a_6$) such that the sum of the square of error in each corresponding point is as minimum as possible
- No other set of parameters exists that may have a lower error (in the squared error sense)

Least Squared Error Solutions

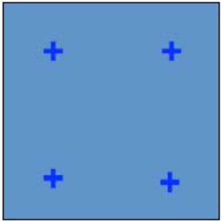


Image 1

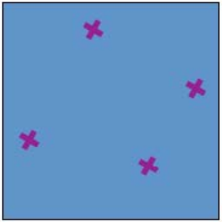
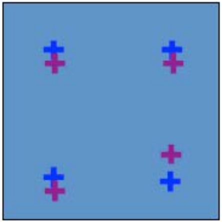


Image 2

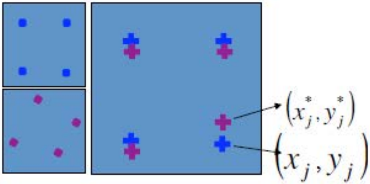


Overlap of points after recovering the transformation

We can try to find the set of parameters in which the error is minimum

Least Squared Error Solutions

$$\begin{bmatrix} x^*_j \\ y^*_j \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \\ 1 \end{bmatrix}$$



$$\boxed{E(a_1, a_2, a_3, a_4, a_5, a_6) = \sum_{j=1}^{n} (x^*_j - x_j)^2 + (y^*_j - y_j)^2}$$

$$E(\mathbf{a}) = \sum_{j=1}^{n} \left((a_1 x'_j + a_2 y'_j + a_3 - x_j)^2 + (a_4 x'_j + a_5 y'_j + a_6 - y_j)^2 \right)$$

Least Squared Error Solutions

$$E(\mathbf{a}) = \sum_{j=1}^{n}\left((a_1x_j + a_2y_j + a_3 - x_j')^2 + (a_4x_j + a_5y_j + a_6 - y_j')^2\right)$$

- Minimize E w.r.t. **a**
- Compute $\partial E / \partial a_i$, put equal to zero, solve simultaneously

$$\begin{bmatrix} \sum_j x_j^2 & \sum_j x_jy_j & \sum_j x_j & 0 & 0 & 0 \\ \sum_j x_jy_j & \sum_j y_j^2 & \sum_j y_j & 0 & 0 & 0 \\ \sum_j x_j & \sum_j y_j & \sum_j 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum_j x_j^2 & \sum_j x_jy_j & \sum_j x_j \\ 0 & 0 & 0 & \sum_j x_jy_j & \sum_j y_j^2 & \sum_j y_j \\ 0 & 0 & 0 & \sum_j x_j & \sum_j y_j & \sum_j 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum_j x_jx_j' \\ \sum_j y_jx_j' \\ \sum_j x_j' \\ \sum_j x_jy_j' \\ \sum_j y_jy_j' \\ \sum_j y_j' \end{bmatrix}$$

Alternative approach

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Recovering Best Affine Transformation

Given three pairs of corresponding points, we get 6 equations

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \end{bmatrix}$$

$$Ax=B \qquad x=A^{-1}B$$

Recovering Best Affine Transformation

- What if we knew four corresponding points?
- We should be able to utilize the additional information

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \\ x_4 & y_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix}$$

Recovering Best Affine Transformation

- $Ax = B$
- Cannot take inverse directly
- Also, 4 correspondences may not be exactly represented by an affine transformation

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \\ x_4 & y_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix}$$

Pseudo Inverse

- For an over-constrained linear system
$$Ax = B$$
- A has more rows than columns
- Multiply by A^T on both sides
$$A^T Ax = A^T B$$
- $A^T A$ is a square matrix of as many rows as x
- We can take its inverse
$$x = (A^T A)^{-1} A^T B$$
- Pseudo-inverse gives the least squares error solution!

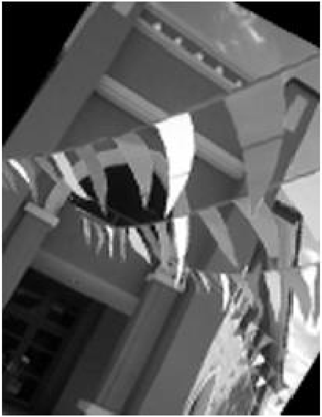
Pseudo Inverse

- In general, we may be given n correspondences
- Concatenate n correspondences in A and B
- A is $2n \times 6$ and B is $2n \times 1$
- Solve using Least Squares

$$A^T A x = A^T B$$

Image Resolution and Transformation

- Number of pixels does not change during translation, rotation, shearing and preserve the spatial resolution of image
- Number of pixels does change during scaling and perspective transformation
- Scale up corresponds to interpolating new pixels
- In scale down, some pixels are removed
- Perspective transformation can distort the image, resulting in a non-uniform distribution of pixels, which effectively changes the pixel count in different parts of the image.


Original


Transformed

Warping

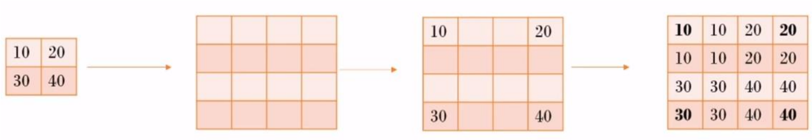
- Inputs:
  - Image X
  - Affine Transformation $A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^T$
- Output:
  - Generate X' such that $X' = AX$
- Obvious Process:
  - For each pixel in X
  - Apply transformation
  - At that location in X', put the same color as at the original location in X
- Problems?

Warping

- This will leave holes
  - Because every pixel does not map to an integer location!
- Reverse Transformation
- For each integer location in X'
- Apply inverse mapping
  - Problem?
- This will not result in answers at integer locations, in general
- Bilinearly interpolate from 4 neighbors

Nearest neighbourhood

Example 1:



Nearest neighbourhood

Example 2

10	40
20	30

10	10	10	40	40	40
10	10	10	40	40	40
10	10	10	40	40	40
20	20	20	30	30	30
20	20	20	30	30	30
20	20	20	30	30	30

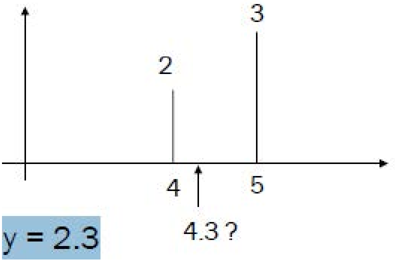
Nearest neighbourhood

Example 3

10	40
20	30

10	10	40	40	40
10	10	40	40	40
20	20	30	30	30
20	20	30	30	30
20	20	30	30	30

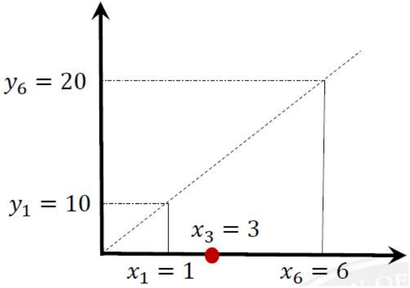
Towards bilinear interpolation



- Use the line equation i.e., $y = mx + c$
- Given: $m = 1$ and $c = -2$
- Substitute $x = 4.3$ provides $y = 2.3$

Bilinear interpolation

- At $x_3 = 3$, find $y_3 = ?$



$$\frac{y_6 - y_1}{x_6 - x_1} = \frac{y_3 - y_1}{x_3 - x_1}$$

$$y_3 = y_1 + \frac{y_6 - y_1}{x_6 - x_1}(x_3 - x_1)$$

Example 1

10	40
30	20

10	x_1	x_2	x_3	x_4	40
	x_5				
		x_6			x_7
30					20

Bilinear interpolation

Example 2