FPT Education

FPT UNIVERSITY

Image processing
- Geometric transformations

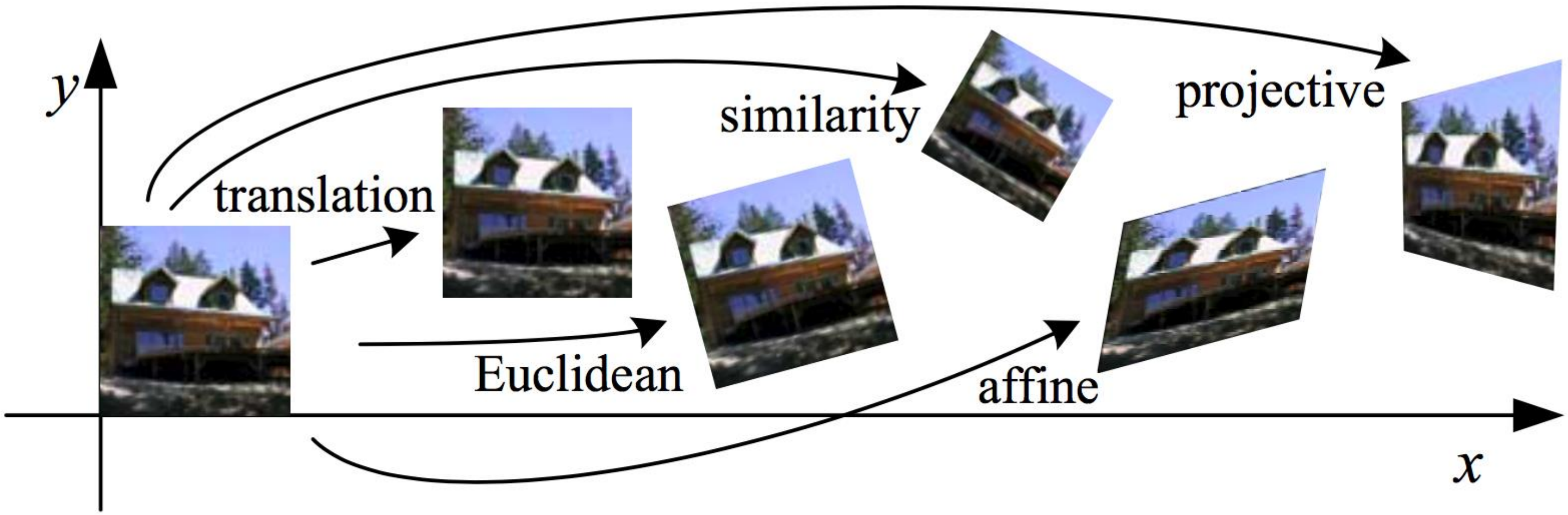
Objectives

- What is Geometric transformations in image domain?
- Why using Geometric transformations ?
- Applications in image, video

Geometric Transformations

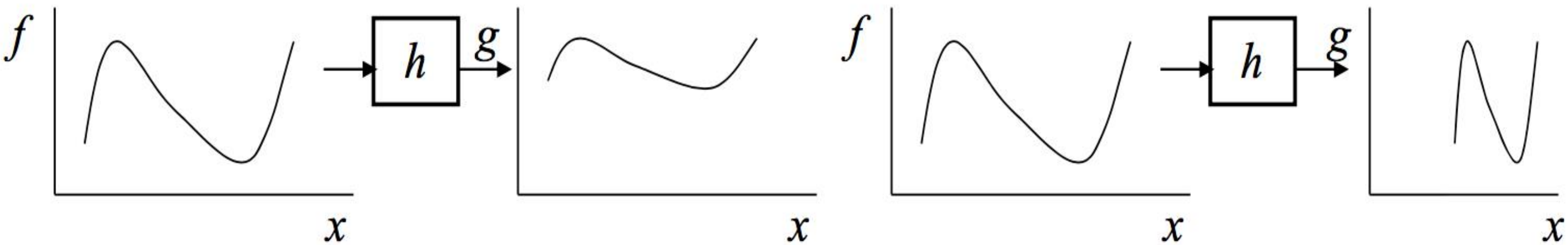
- Image transforms the range of the image

$$g(\boldsymbol{x}) = h(f(\boldsymbol{x}))$$



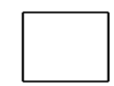
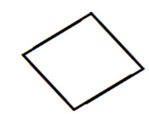
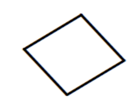
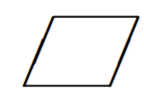
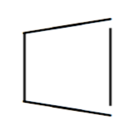
- Transform the domain

$$g(\boldsymbol{x}) = f(\boldsymbol{h}(\boldsymbol{x}))$$



Parametric transformations

- Why name is parametric transformation?
  - The behavior of the transformation is controlled by a small number of parameters

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

- We have source image f(x) and transformation formula x'= h(x). How to compute the values of the pixels in the new image g(x)?

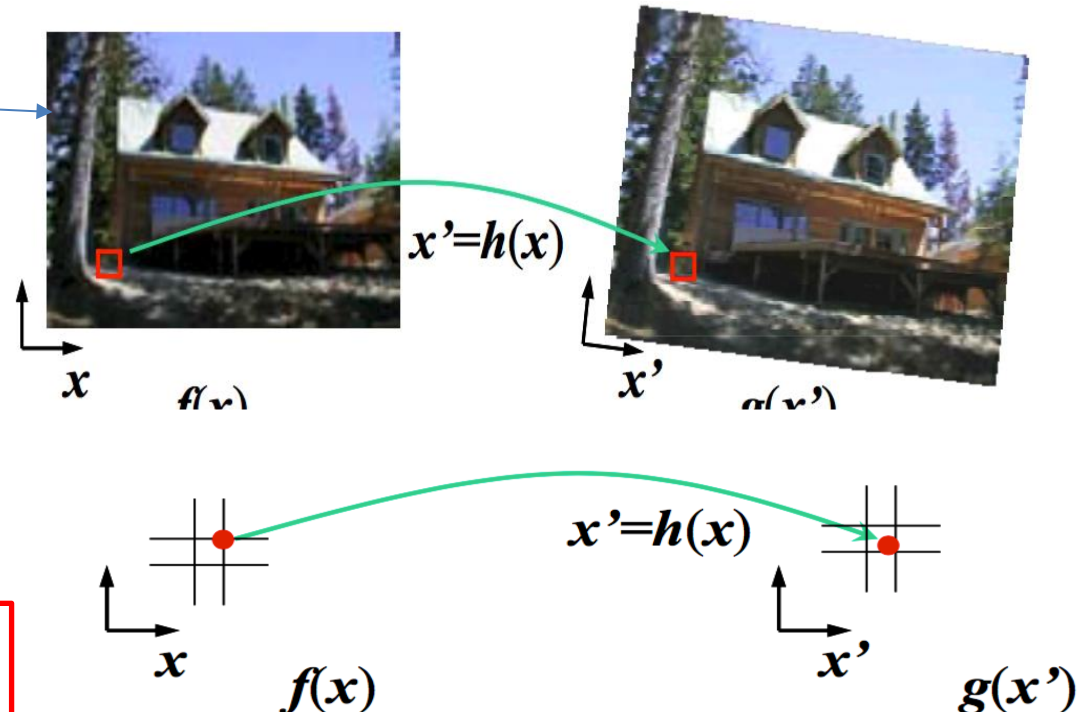
forward warping

procedure $forwardWarp(f, \boldsymbol{h}, \textbf{out } g)$:

For every pixel $\boldsymbol{x}$ in $f(\boldsymbol{x})$

1. Compute the destination location $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x}$
2. Copy the pixel $f(\boldsymbol{x})$ to $g(\boldsymbol{x}')$.

Is that good ?



Parametric transformations

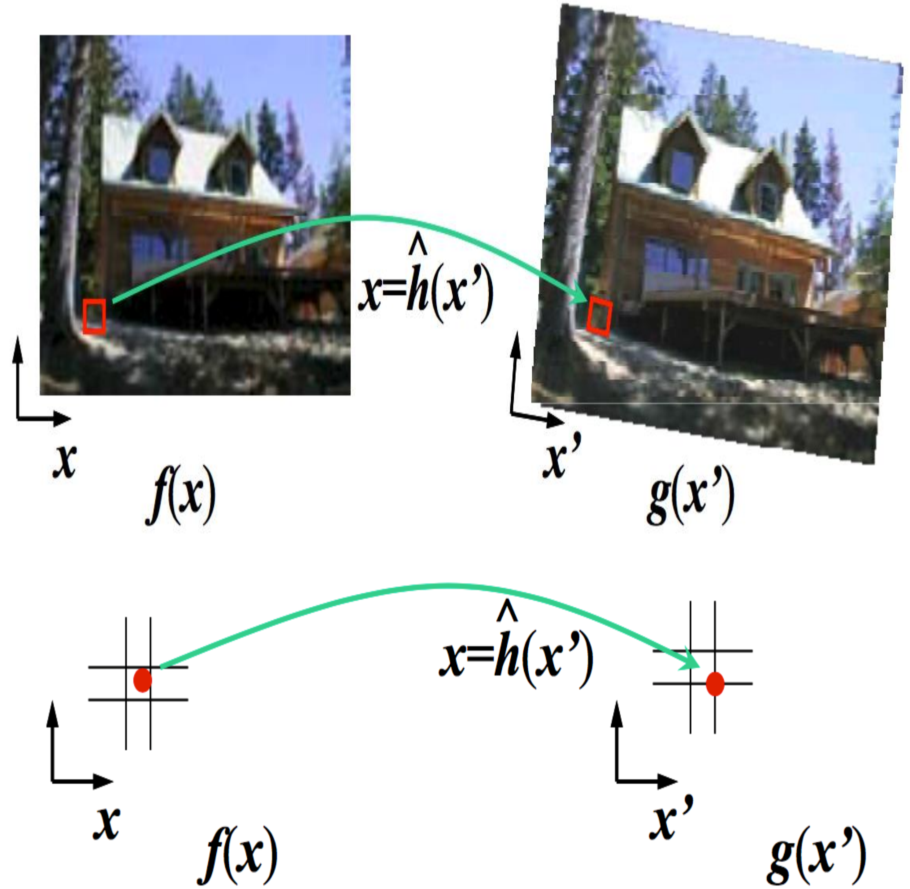
- Forward warping:
  - The process of copying a pixel $f(x)$ to a location x' in g is not well defined when x' has a non-integer value
  - forward warping is the appearance of cracks and holes, especially when magnifying an image.
- Any idea for this problems?
  - Each pixel in the destination image $g(x')$ is sampled from the original image $f(x)$
- How does this differ from ?
  - since $\hat{h}(x')$ is (presumably) defined for all pixels in $g(x')$ → we no longer have holes.
  - Resampling an image at non-integer locations is a well-studied problem
  - High-quality filters that control aliasing can be used.

- Inverse warping:

procedure *inverseWarp*$(f, h,$ **out** $g)$:

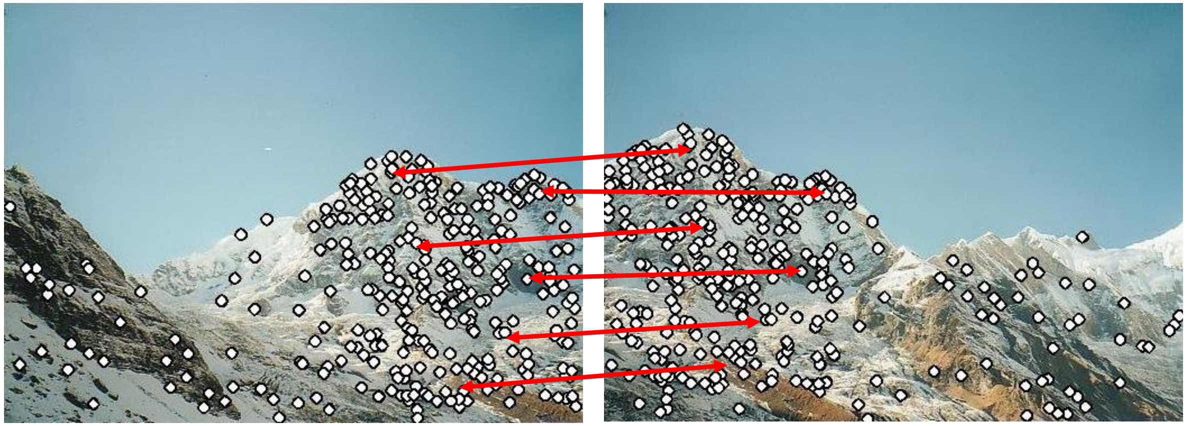
For every pixel x' in $g(x')$

1. Compute the source location $x = \hat{h}(x')$
2. Resample $f(x)$ at location x and copy to $g(x')$



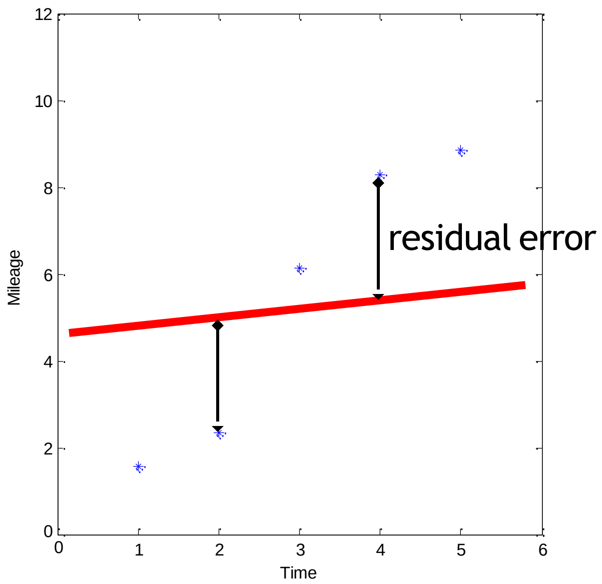
Computing transformations

- Given a set of matches between images A and B
  - How can we compute the transform T from A to B?



Find transform T that best "agrees" with the matches

Linear regression



$$b = \frac{\sum y_i - m \sum x_i}{n}$$

$$m = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$\mathrm{Cost}(m, b) = \sum_{i=1}^{n} |y_i - (mx_i + b)|^2$$

Calculate partial derivatives w.r.t. m and b and set them to 0.

Linear regression with matrix operations

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Least squares

$$\mathbf{At} = \mathbf{b}$$

- Find **t** that minimizes

$$||\mathbf{At} - \mathbf{b}||^2$$

- To solve, form the *normal equations*

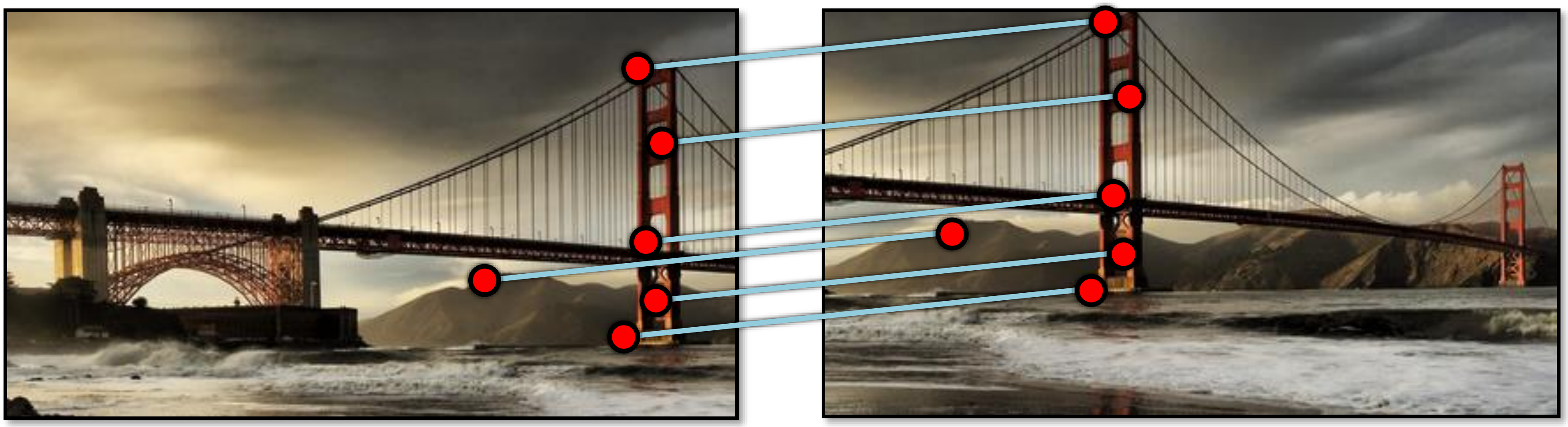
$$\mathbf{A}^{\mathrm{T}}\mathbf{At} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{t} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

This is what you solved in HW 1

Check proof here

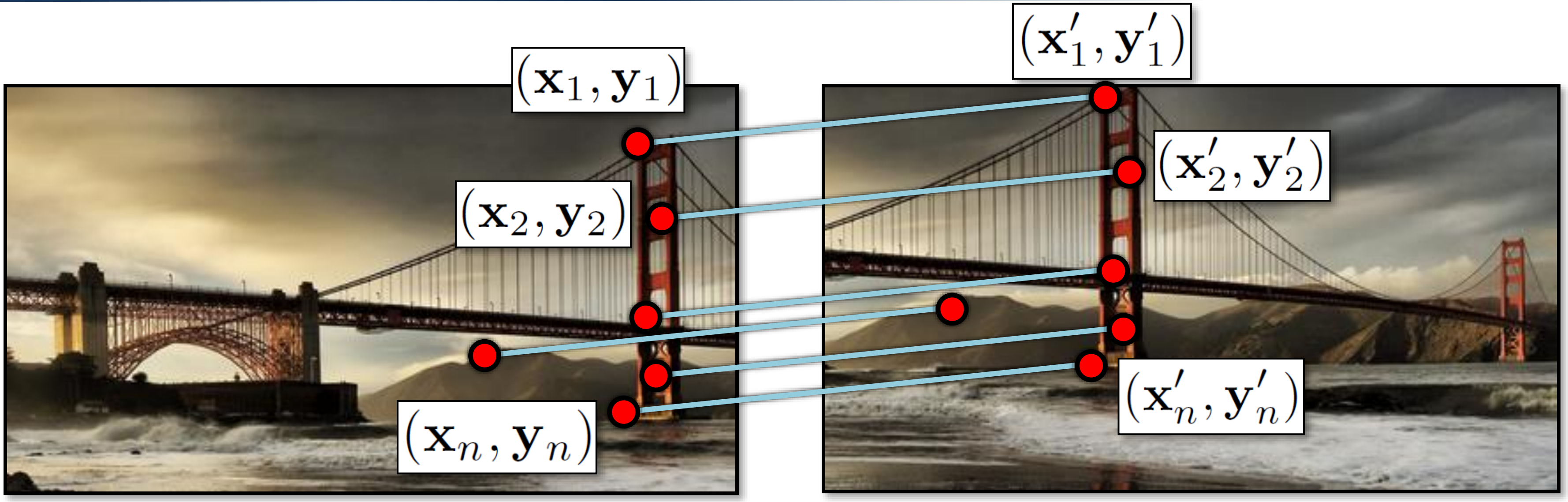
Simple case: translations



$(\mathbf{x}_t, \mathbf{y}_t)$

How do we solve for $(\mathbf{x}_t, \mathbf{y}_t)$?

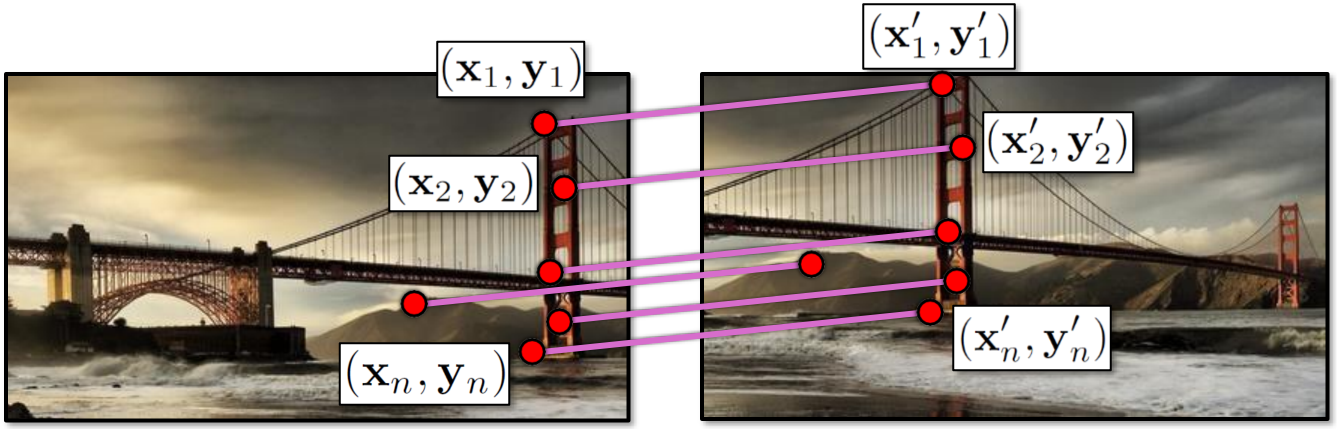
Simple case: translations



Displacement of match *i* = $(\mathbf{x}'_i - \mathbf{x}_i, \mathbf{y}'_i - \mathbf{y}_i)$

$$(\mathbf{x}_t, \mathbf{y}_t) = \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}'_i - \mathbf{x}_i, \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}'_i - \mathbf{y}_i \right)$$

Another view



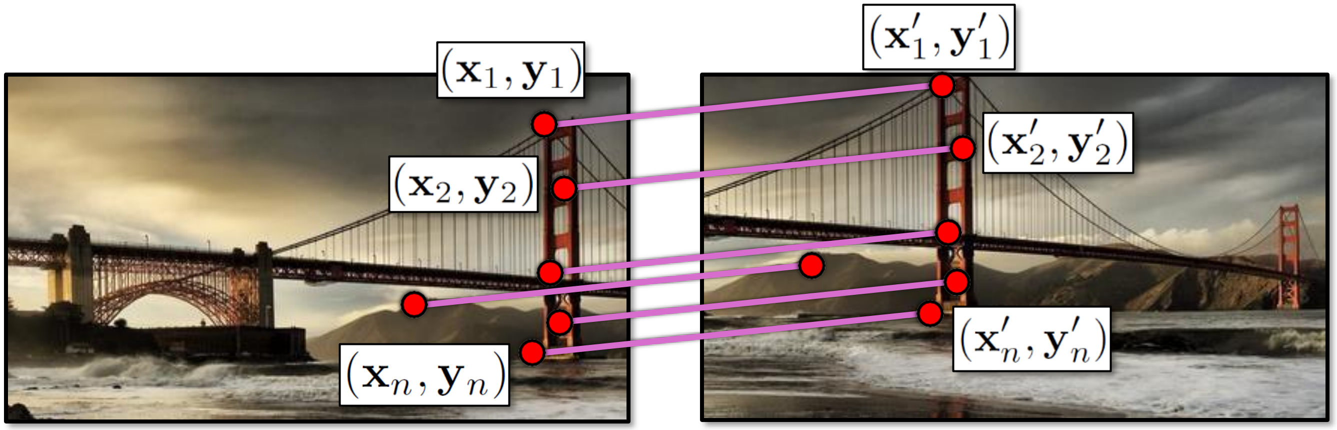
$$\mathbf{x}_i + \mathbf{x_t} = \mathbf{x}'_i$$

$$\mathbf{y}_i + \mathbf{y_t} = \mathbf{y}'_i$$

- System of linear equations
  - What are the knowns? Unknowns?
  - How many unknowns? How many matches do we need?

Another view

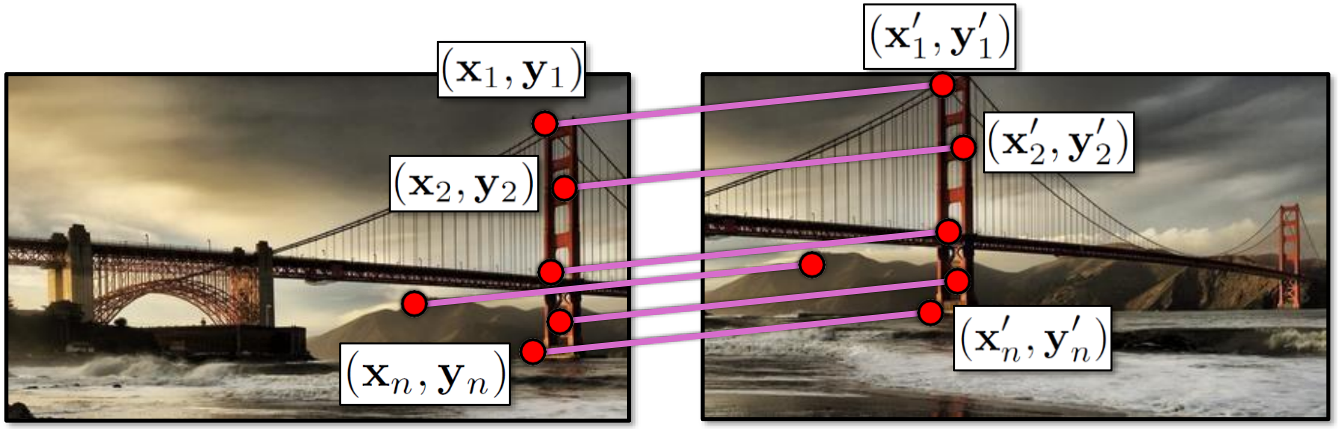
- Problem: more equations than unknowns
  - "Overdetermined" system of equations
  - We will find the least squares solution



$$\begin{aligned} \mathbf{x}_i + \mathbf{x_t} &= \mathbf{x}'_i \\ \mathbf{y}_i + \mathbf{y_t} &= \mathbf{y}'_i \end{aligned}$$

Another view

- Problem: more equations than unknowns
  - "Overdetermined" system of equations
  - We will find the least squares solution



$$\begin{aligned} \mathbf{x}_i + \mathbf{x_t} &= \mathbf{x}'_i \\ \mathbf{y}_i + \mathbf{y_t} &= \mathbf{y}'_i \end{aligned}$$

Least squares formulation

- For each point $(\mathbf{x}_i, \mathbf{y}_i)$

$$\begin{aligned}\mathbf{x}_i + \mathbf{x_t} &= \mathbf{x}_i' \\ \mathbf{y}_i + \mathbf{y_t} &= \mathbf{y}_i'\end{aligned}$$

- we define the *residuals* as

$$\begin{aligned}r_{\mathbf{x}_i}(\mathbf{x}_t) &= (\mathbf{x}_i + \mathbf{x}_t) - \mathbf{x}_i' \\ r_{\mathbf{y}_i}(\mathbf{y}_t) &= (\mathbf{y}_i + \mathbf{y}_t) - \mathbf{y}_i'\end{aligned}$$

Least squares formulation

- Goal: minimize sum of squared residuals

$$C(\mathbf{x}_t, \mathbf{y}_t) = \sum_{i=1}^{n} \left(r_{\mathbf{x}_i}(\mathbf{x}_t)^2 + r_{\mathbf{y}_i}(\mathbf{y}_t)^2 \right)$$

- "Least squares" solution
- Take partial derivative, equate to 0, and find x_t and y_t
- For translations, is equal to mean (average) displacement (practice this proof at home!
  - Mean displacement = $$\left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}'_i - \mathbf{x}_i, \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}'_i - \mathbf{y}_i \right)$$

Least squares formulation

- Can also write as a matrix equation
- For a simple problem like this, computing partial derivative and setting it to 0 and working the math out leads to a closed form solution.
- Closed form solution is significantly faster than matrix inversion for large number of matches

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ & \vdots \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x'_1 - x_1 \\ y'_1 - y_1 \\ x'_2 - x_2 \\ y'_2 - y_2 \\ \vdots \\ x'_n - x_n \\ y'_n - y_n \end{bmatrix}$$

$$\underset{2n \times 2}{\mathbf{A}} \quad \underset{2 \times 1}{\mathbf{t}} = \underset{2n \times 1}{\mathbf{b}} \qquad \mathbf{t} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Summary

- Learn about Geometric transformations in image domain.
- Learn about Computing transformations