

# CS5760: Computer Vision

## Image alignment



<http://www.wired.com/gadgetlab/2010/07/camera-software-lets-you-see-into-the-past/>

# Reading

- Szeliski (1<sup>st</sup> edition): Chapter 6.1

# Announcements

- Project 2 due this Friday, March 12 by 7pm
  - Please get started now if you haven't already!
  - Report due next Monday, March 15 by 7pm on CMSX
- Take-home midterm will be released next Monday, March 15, due Friday, March 19
  - Open book, open note (but no Google)
  - To be done on your own
- No class this Wednesday, March 10 (Wellness Day)

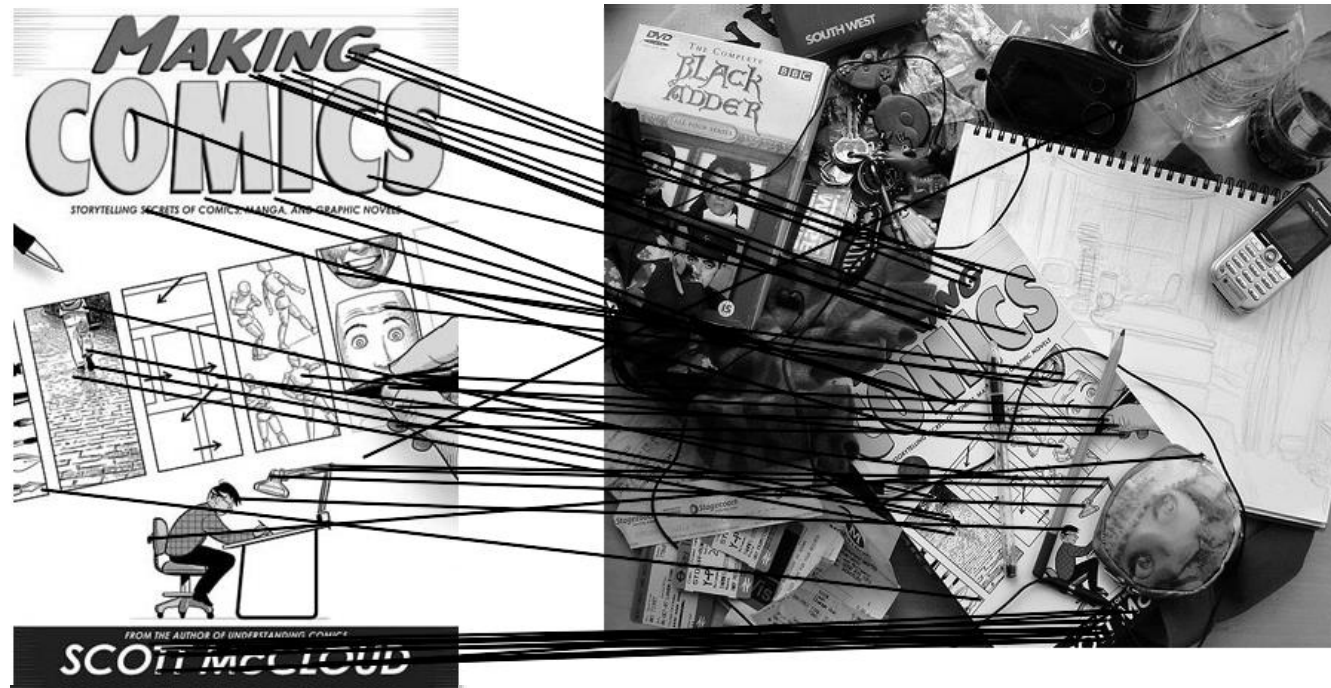
# Today in Computer Vision



<https://www.theverge.com/2021/3/5/22314980/tom-cruise-deepfake-tiktok-videos-ai-impersonator-chris-ume-miles-fisher>

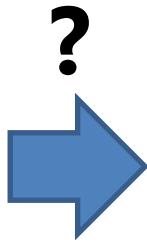
# Computing transformations

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



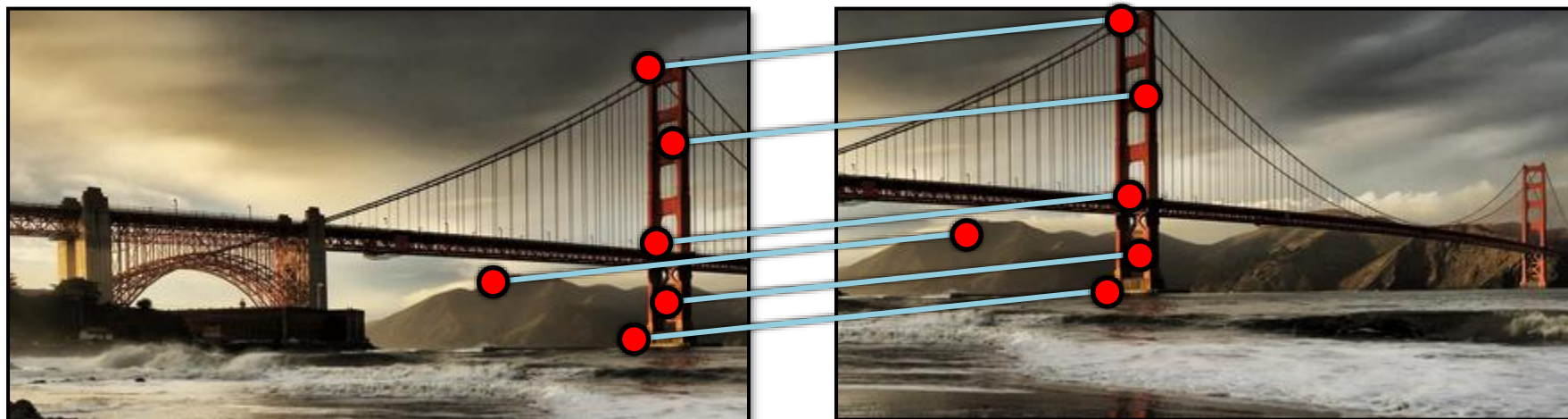
- Find transform  $T$  that best “agrees” with the matches

# Computing transformations





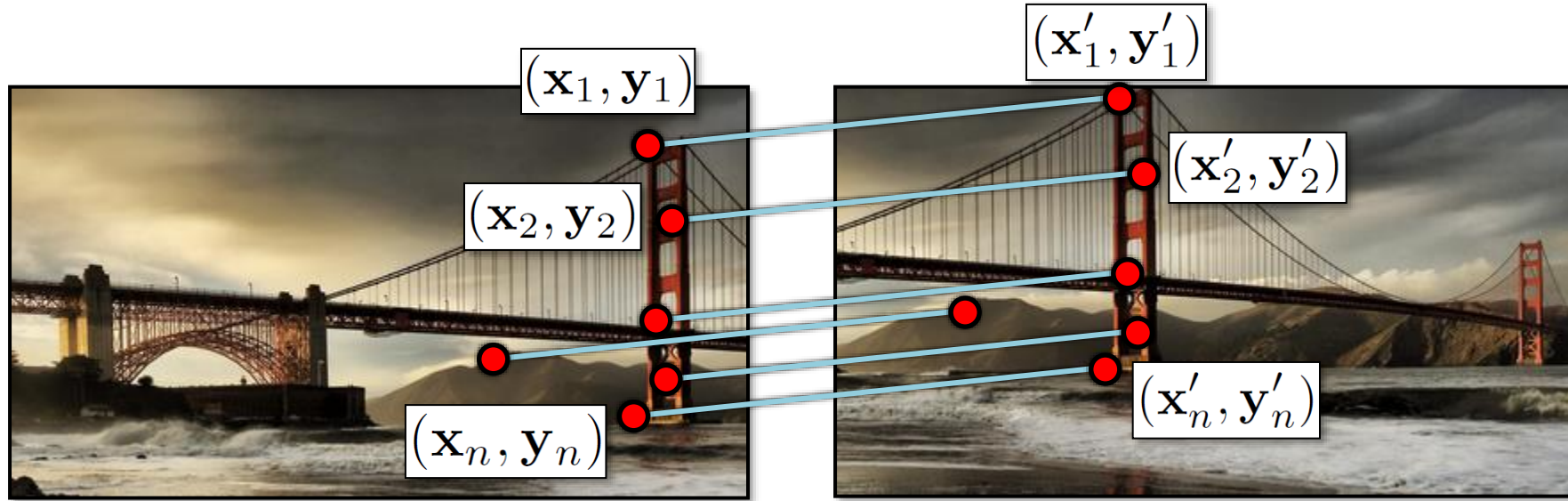
# Simple case: translations



$(x_t, y_t)$

How do we solve for  
 $(x_t, y_t)$  ?

# Simple case: translations

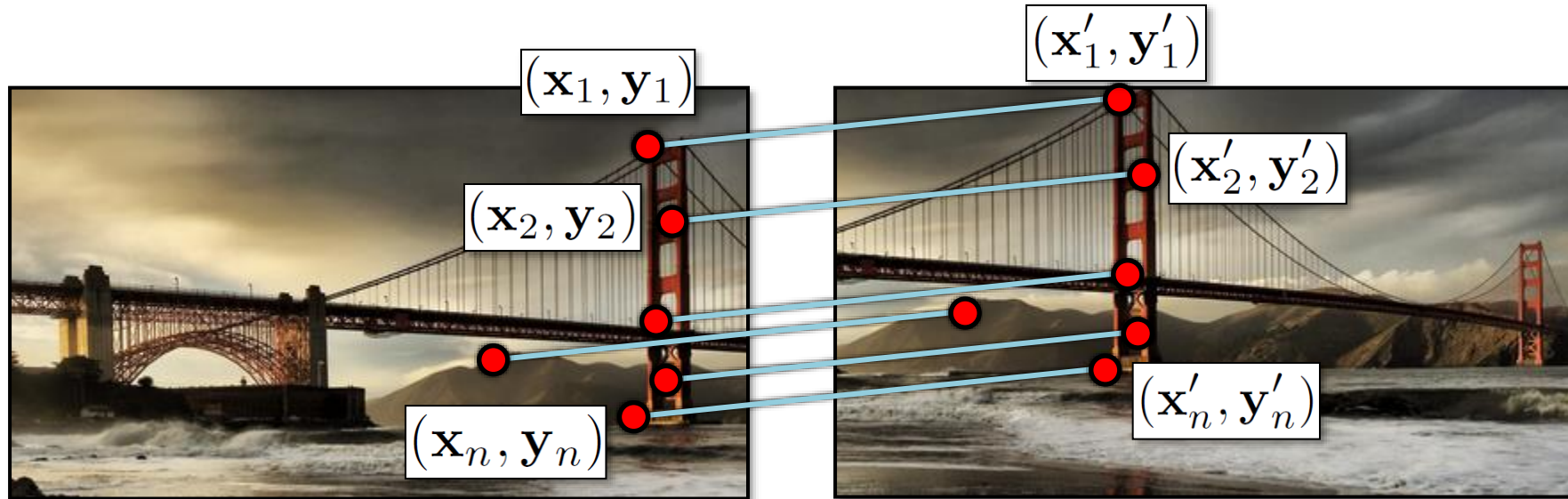


Displacement of match  $i = (\mathbf{x}'_i - \mathbf{x}_i, \mathbf{y}'_i - \mathbf{y}_i)$

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



# Another view

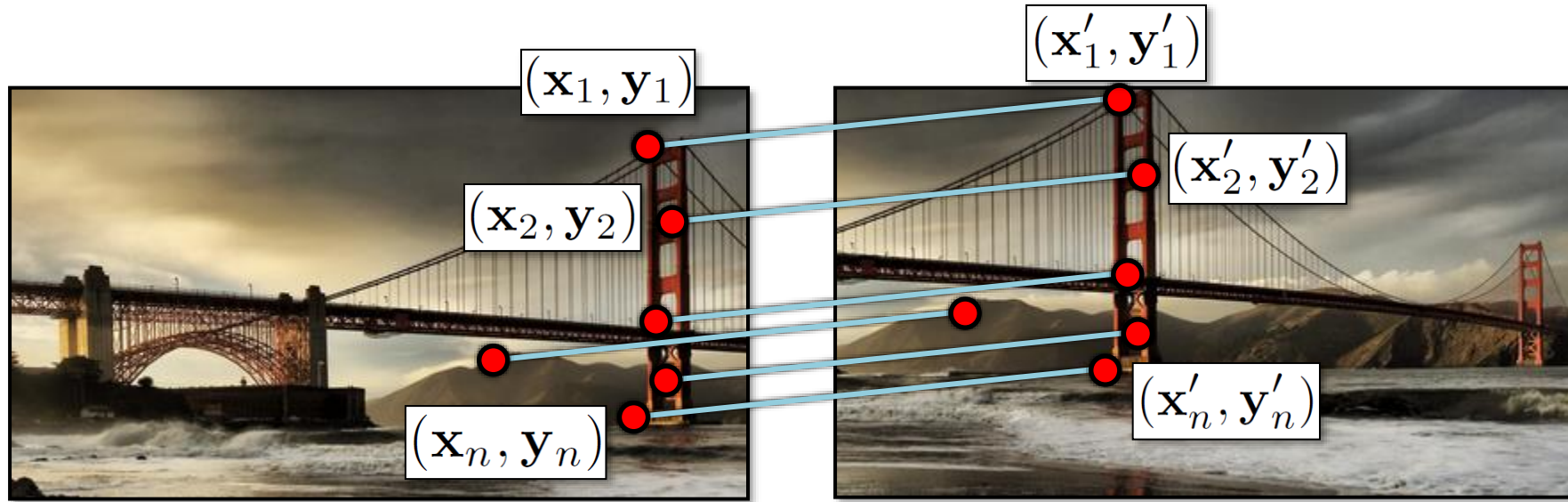


$$\mathbf{x}_i + \mathbf{x}_t = \mathbf{x}'_i$$

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

- System of linear equations
  - What are the knowns? Unknowns?
  - How many unknowns? How many equations (per match)?

# Another view



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

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

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

# Least squares formulation

- For each point

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

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

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

- we define the *residuals* as

$$r_{\mathbf{x}_i}(\mathbf{x}_t) = (\mathbf{x}_i + \mathbf{x}_t) - \mathbf{x}'_i$$

$$r_{\mathbf{y}_i}(\mathbf{y}_t) = (\mathbf{y}_i + \mathbf{y}_t) - \mathbf{y}'_i$$

# 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
- For translations, is equal to mean (average) displacement

# Least squares formulation

- Can also write as a matrix equation

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

$$\mathbf{A}$$

$2n \times 2$

$$\mathbf{t}$$

$2 \times 1$

=

$$\mathbf{b}$$

$2n \times 1$



# Least squares

$$\mathbf{A}\mathbf{t} = \mathbf{b}$$

- Find  $\mathbf{t}$  that minimizes

$$\|\mathbf{A}\mathbf{t} - \mathbf{b}\|^2$$

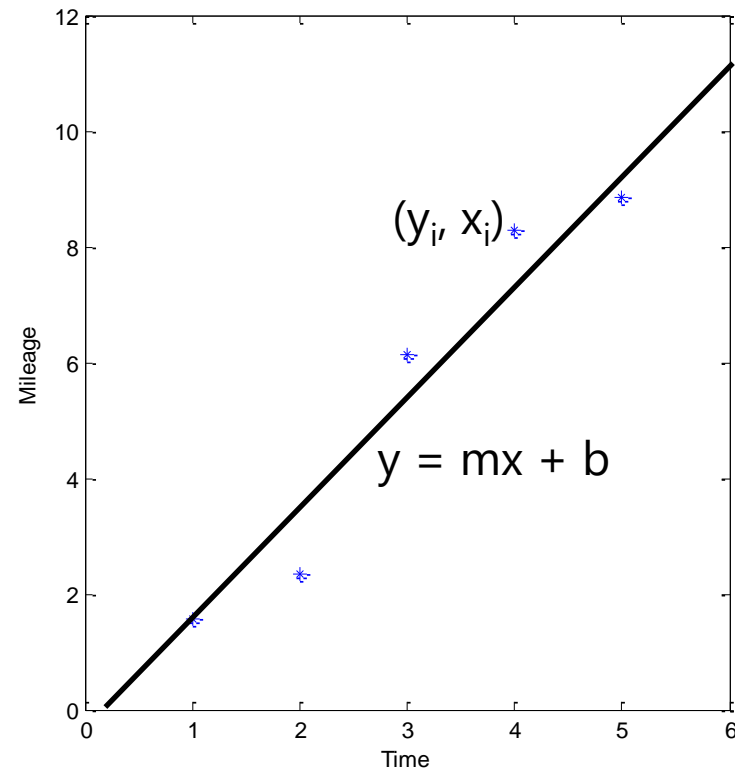
- To solve, form the *normal equations*

$$\mathbf{A}^T \mathbf{A} \mathbf{t} = \mathbf{A}^T \mathbf{b}$$

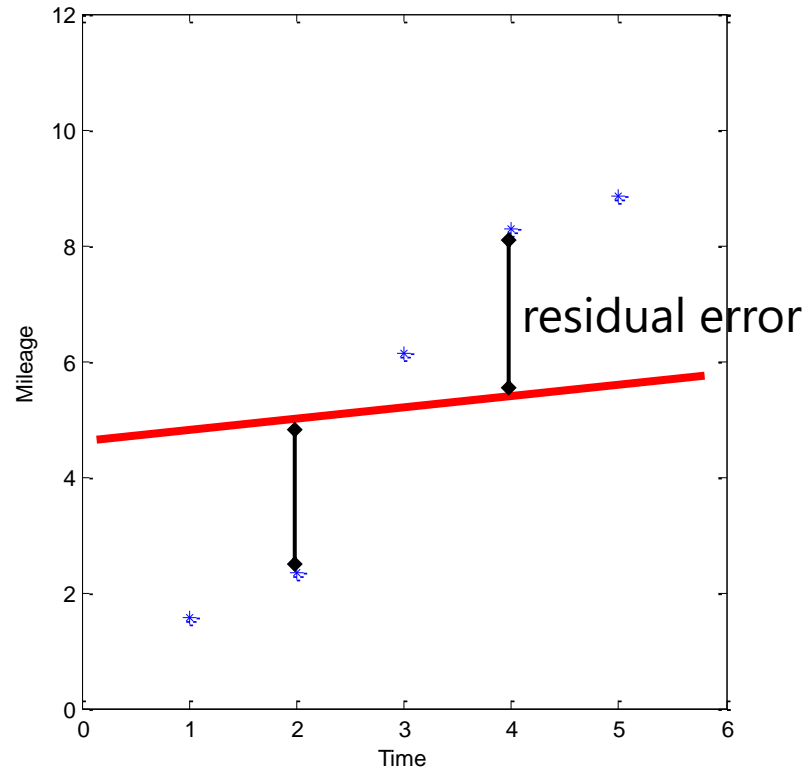
$$\mathbf{t} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

**Questions?**

# Least squares: linear regression



# Linear regression



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

# Linear regression

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



# Affine transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



- How many unknowns?
- How many equations per match?
- How many matches do we need?

# Affine transformations

- Residuals:

$$r_{x_i}(a, b, c, d, e, f) = (ax_i + by_i + c) - x'_i$$

$$r_{y_i}(a, b, c, d, e, f) = (dx_i + ey_i + f) - y'_i$$

- Cost function:

$$C(a, b, c, d, e, f) = \sum_{i=1}^n (r_{x_i}(a, b, c, d, e, f)^2 + r_{y_i}(a, b, c, d, e, f)^2)$$

# Affine transformations

- Matrix form

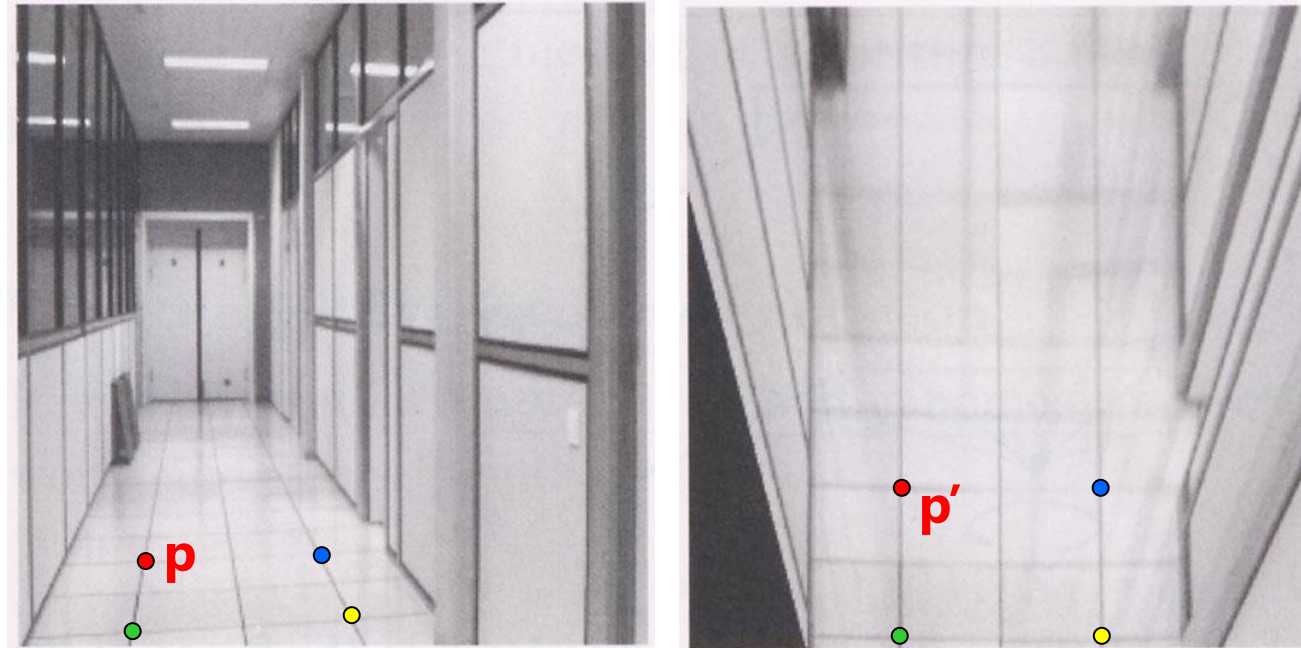
$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ & & & \vdots & & \\ x_n & y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix}$$

**A**  
 $2n \times 6$

**t**  
 $6 \times 1$

**b**  
 $2n \times 1$

# Homographies



To unwarp (rectify) an image

- solve for homography  $\mathbf{H}$  given  $\mathbf{p}$  and  $\mathbf{p}'$
- solve equations of the form:  $w\mathbf{p}' = \mathbf{H}\mathbf{p}$ 
  - linear in unknowns:  $w$  and coefficients of  $\mathbf{H}$
  - $\mathbf{H}$  is defined up to an arbitrary scale factor
  - how many matches are necessary to solve for  $\mathbf{H}$ ?

# Solving for homographies

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

**Not linear!**

$$x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}$$

$$y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}$$



# Solving for homographies

$$x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}$$

$$y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

# Solving for homographies

$$\begin{bmatrix}
 x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\
 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\
 \vdots & & & & & & & & \\
 x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\
 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n
 \end{bmatrix}
 \begin{bmatrix}
 h_{00} \\
 h_{01} \\
 h_{02} \\
 h_{10} \\
 h_{11} \\
 h_{12} \\
 h_{20} \\
 h_{21} \\
 h_{22}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 \vdots \\
 0 \\
 0
 \end{bmatrix}$$

**A**

$2n \times 9$

**h**

9

**0**

2n

Defines a least squares problem: minimize  $\|\mathbf{A}\mathbf{h} - \mathbf{0}\|^2$

- Since  $\mathbf{h}$  is only defined up to scale, solve for unit vector  $\hat{\mathbf{h}}$
- Solution:  $\hat{\mathbf{h}}$  = eigenvector of  $\mathbf{A}^T \mathbf{A}$  with smallest eigenvalue
- Works with 4 or more points

# Recap: Two Common Optimization Problems

## Problem statement

$$\text{minimize } \|\mathbf{Ax} - \mathbf{b}\|^2$$

(least squares solution to  $\mathbf{Ax} = \mathbf{b}$ )

## Solution

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b} \text{ (matlab)}$$

## Problem statement

$$\text{minimize } \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} \text{ s.t. } \mathbf{x}^T \mathbf{x} = 1$$

(non-trivial lsq solution to  $\mathbf{Ax} = 0$ )

## Solution

$$[\mathbf{v}, \lambda] = \text{eig}(\mathbf{A}^T \mathbf{A})$$

$$\lambda_1 < \lambda_{2..n} : \mathbf{x} = \mathbf{v}_1$$

# Computing transformations



**Questions?**

# Image Alignment Algorithm

Given images A and B

1. Compute image features for A and B
2. Match features between A and B
3. Compute homography between A and B using least squares on set of matches

What could go wrong?

# Outliers

