3D Vision and Machine Perception

Prof. Kyungdon Joo

3D Vision & Robotics Lab.

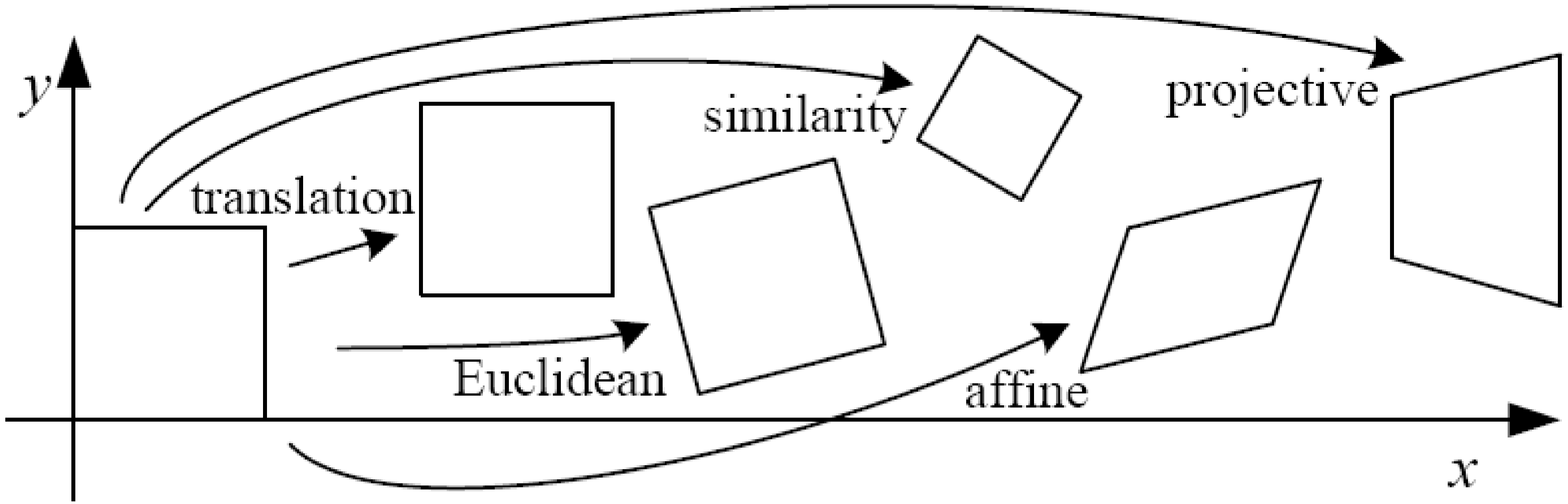
AI Graduate School (AIGS) & Computer Science and Engineering (CSE)

Contents

- Back to warping: image homographies
- Computing with homographies
- The direct linear transform (DLT)
- Random Sample Consensus (RANSAC)

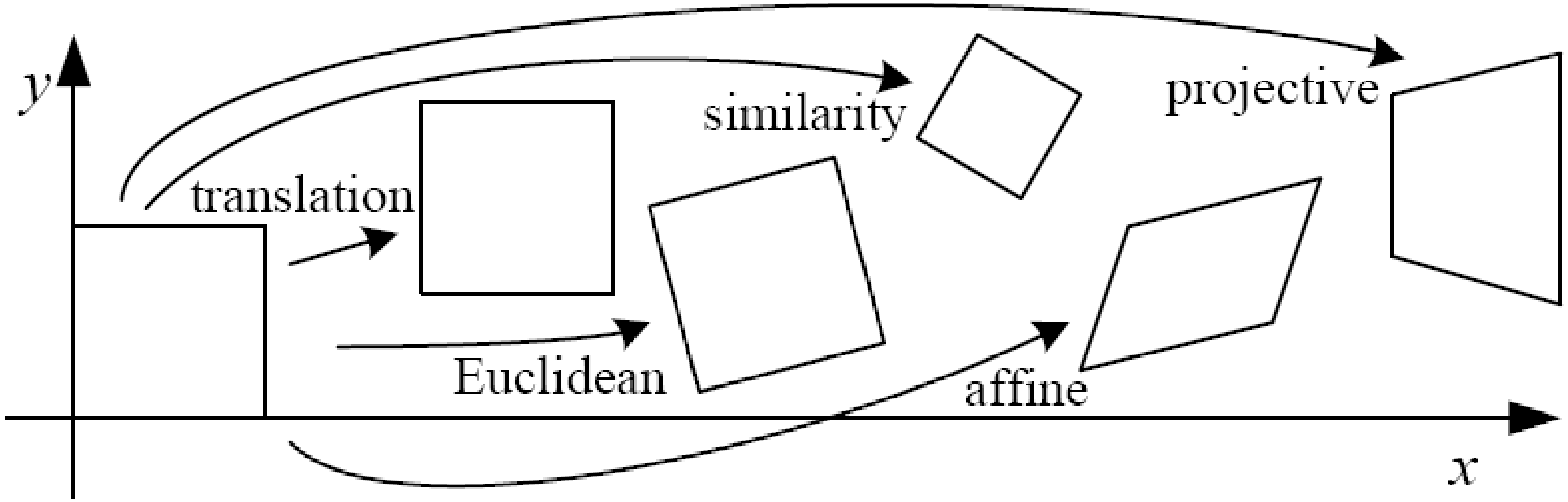
Back to warping:
image homographies

Classification of 2D transformations

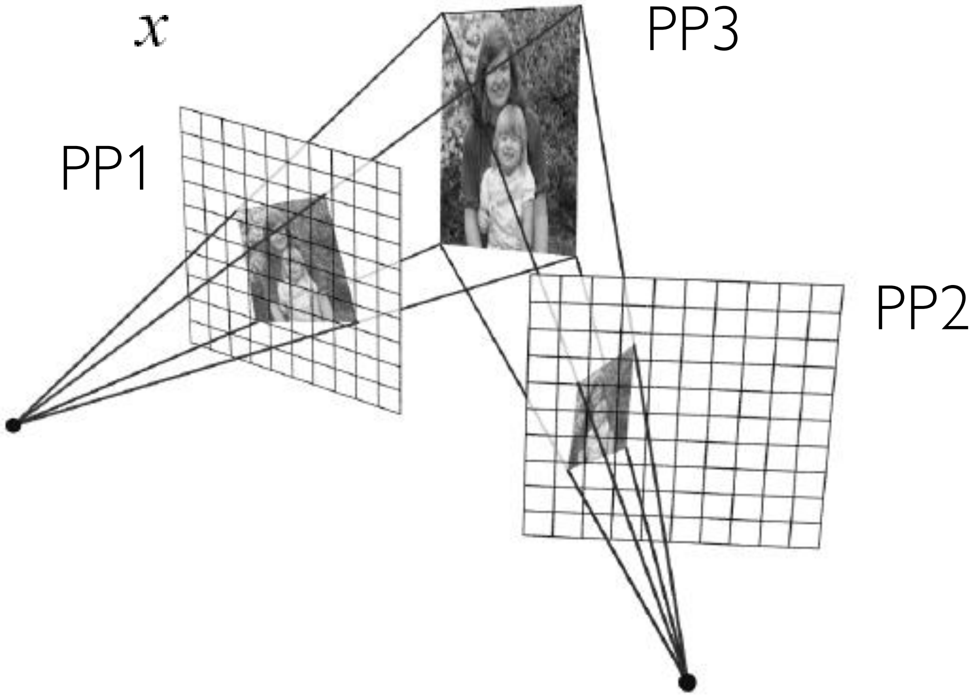


Name	Matrix	# D.O.F.
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8

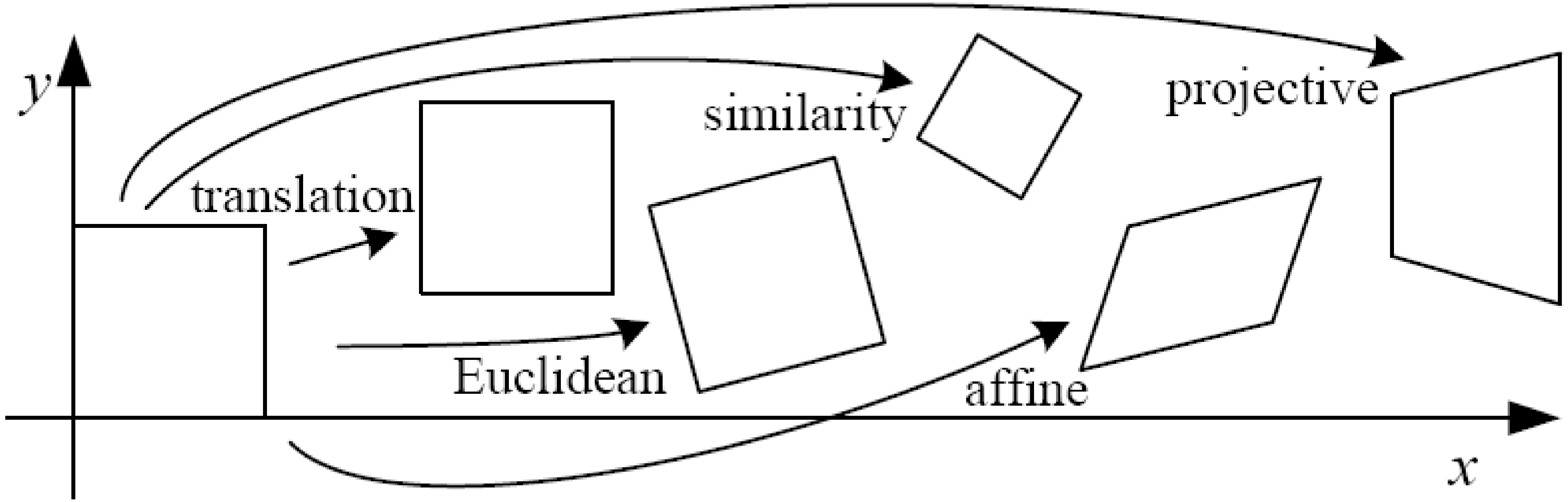
Classification of 2D transformations

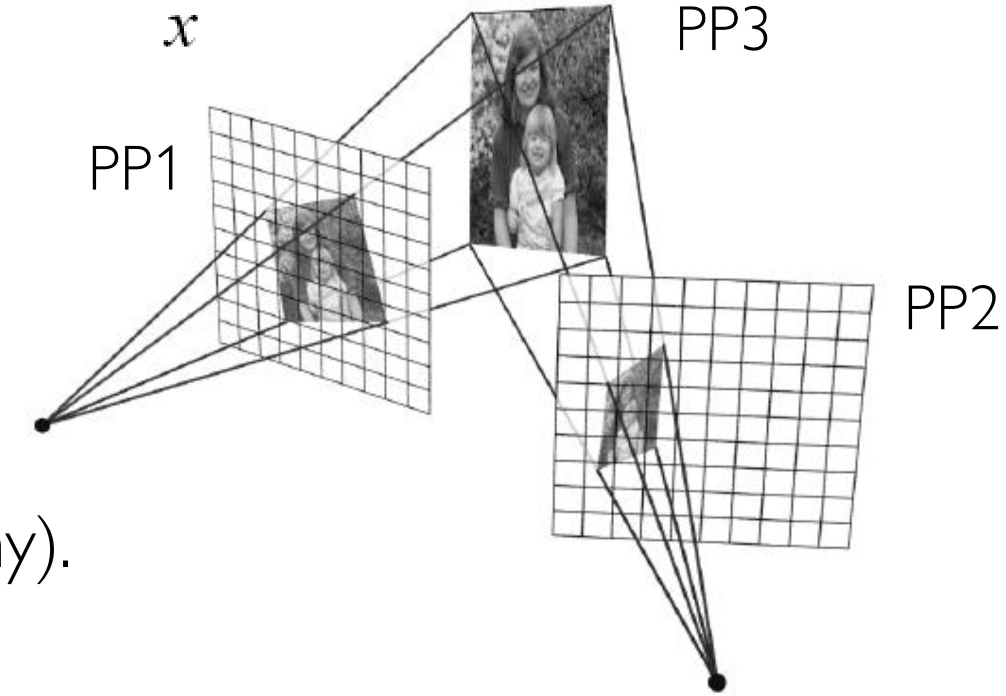


- Which kind transformation is needed to warp projective plane 1 into projective plane 2?



Classification of 2D transformations



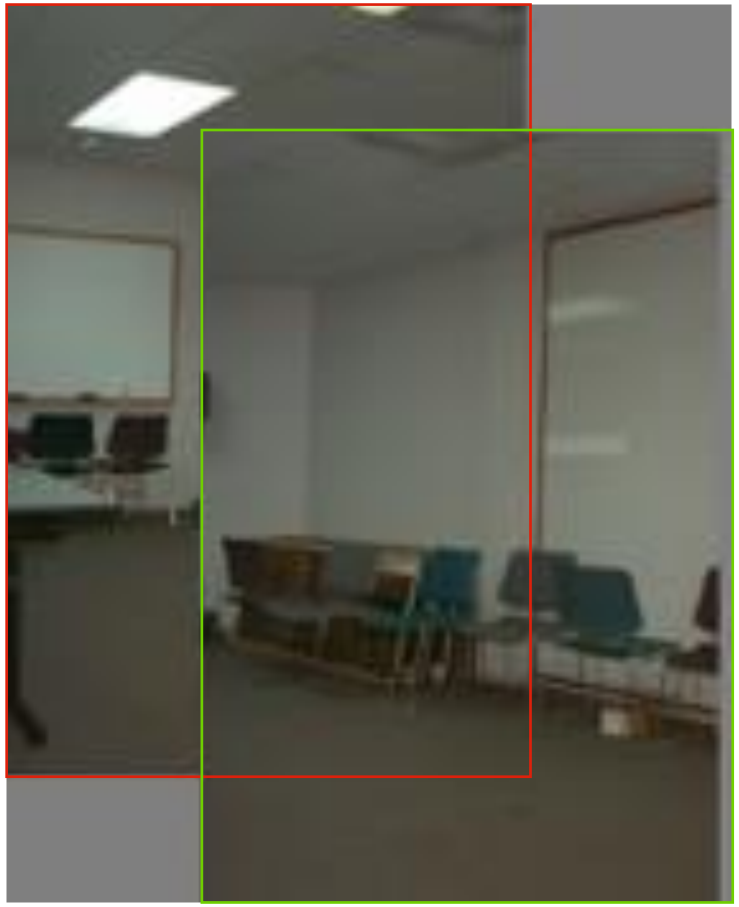


- Which kind transformation is needed to warp projective plane 1 into projective plane 2?

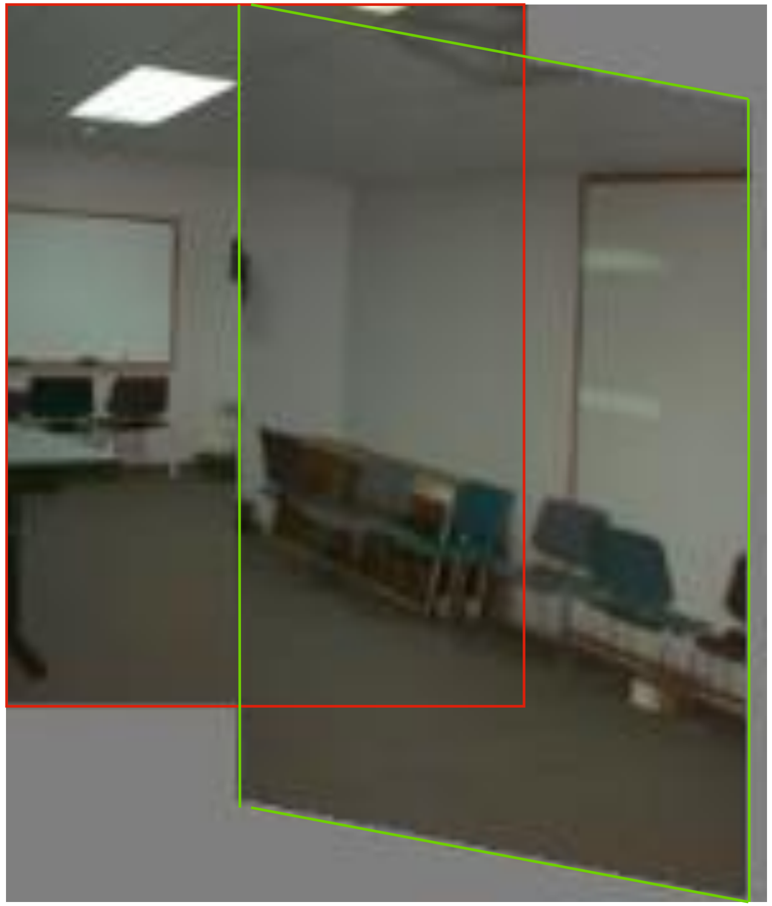
- A projective transformation (a.k.a. a homography).

Warping with different transformations

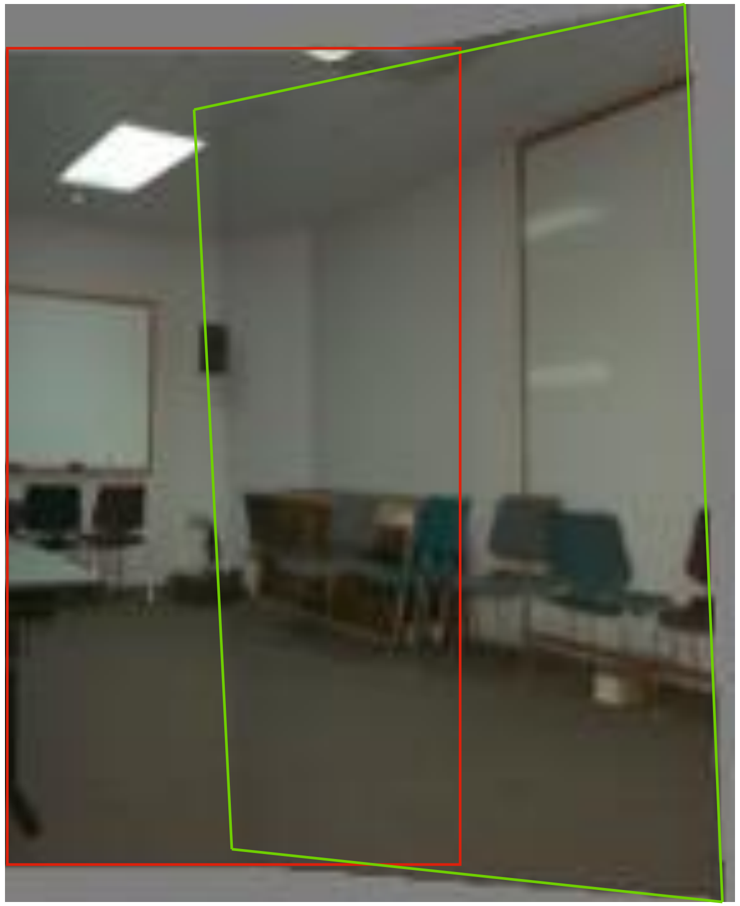
translation



affine



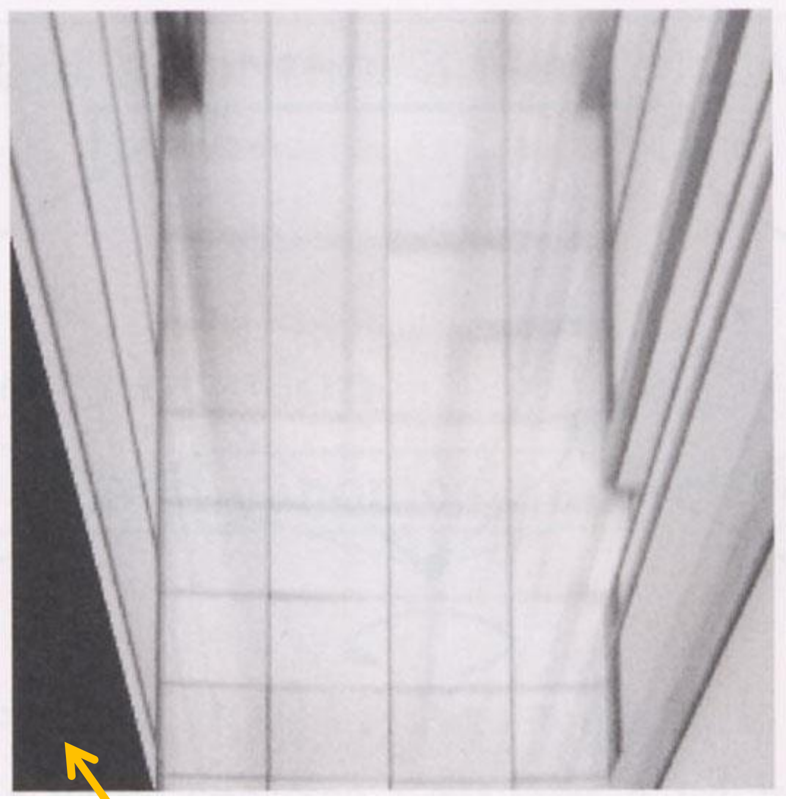
projective (homography)



View warping

original view

synthetic top view



synthetic side view

what are these black areas near the boundaries?

Virtual camera rotations

original view

synthetic rotations

Image rectification

two
original
images

rectified and stitched

Street art

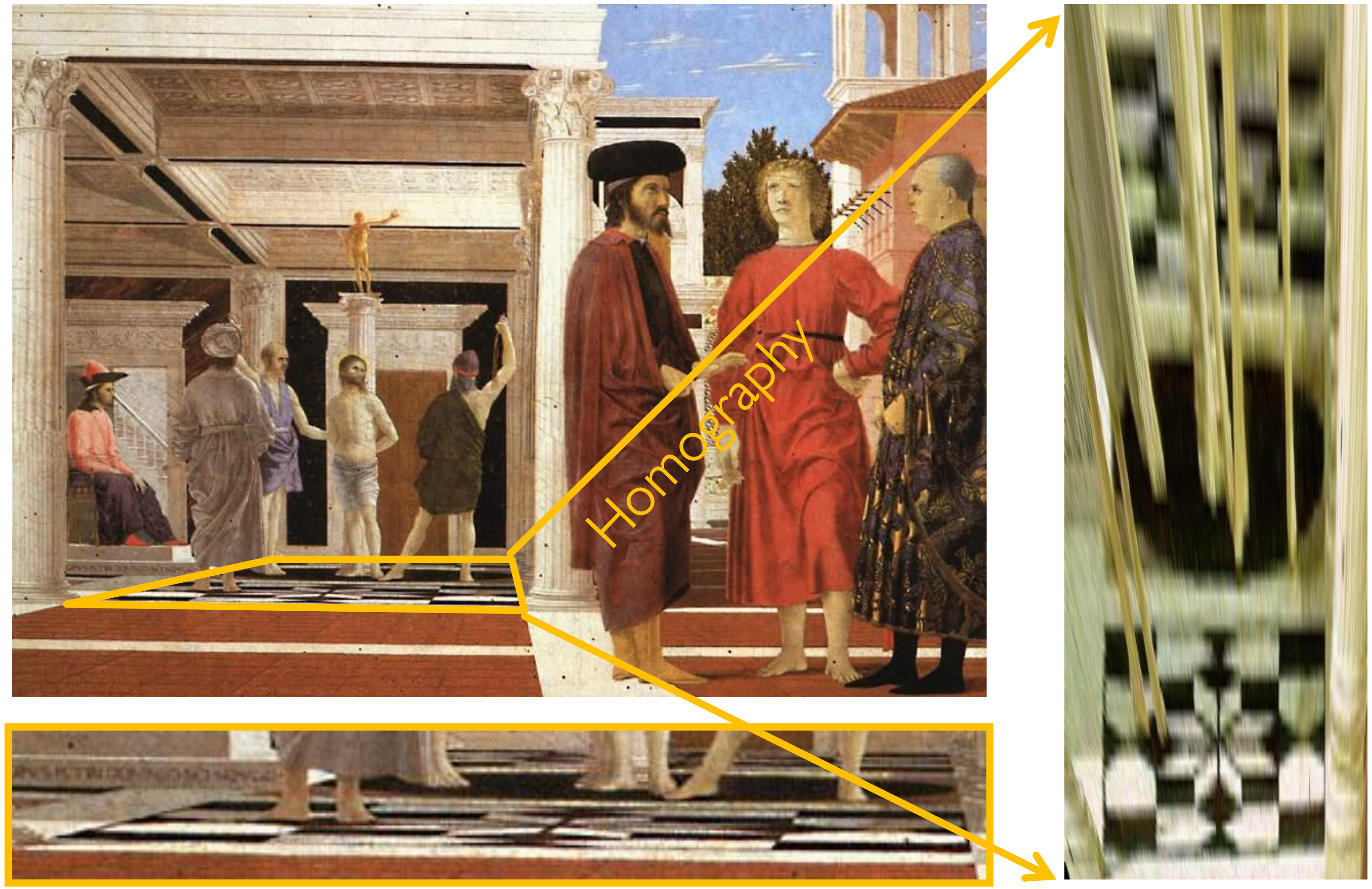
Understanding geometric patterns

- What is the pattern on the floor?

magnified view of floor

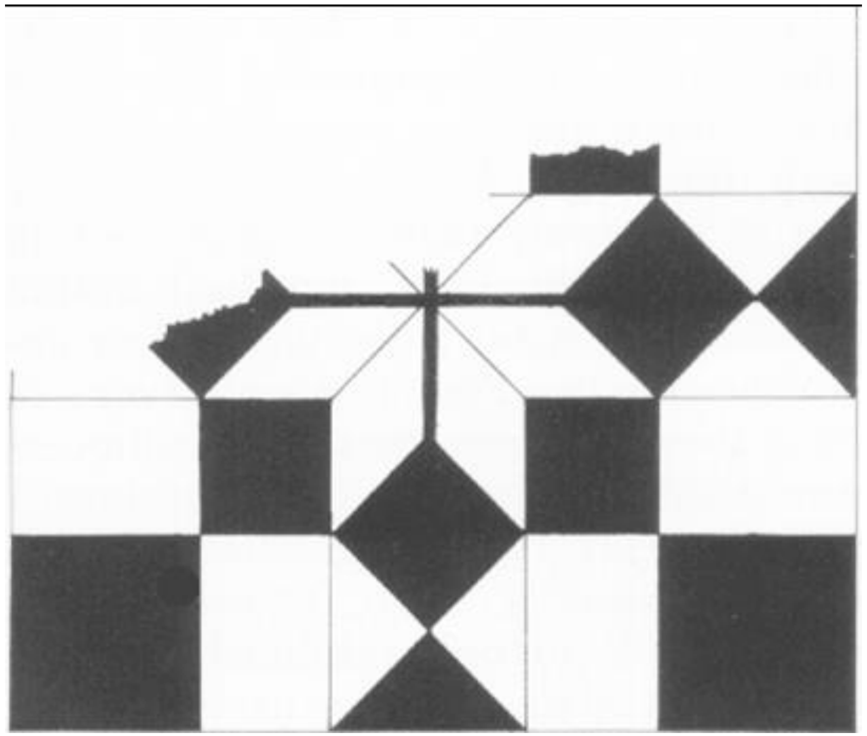
Understanding geometric patterns

- What is the pattern on the floor?



magnified view of floor

rectified view

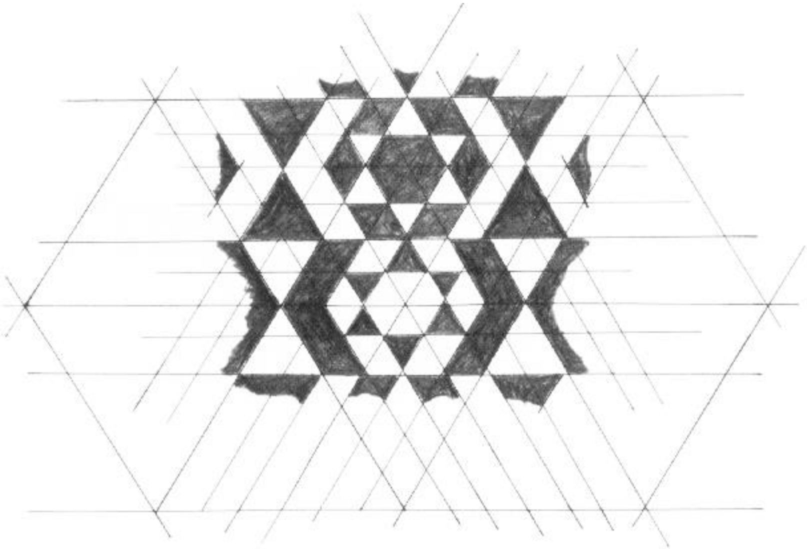


reconstruction from rectified view

Understanding geometric patterns

- Very popular in renaissance drawings (when perspective was discovered)

rectified view of floor



reconstruction

A weird drawing

- Holbein, “The Ambassadors”

A weird drawing

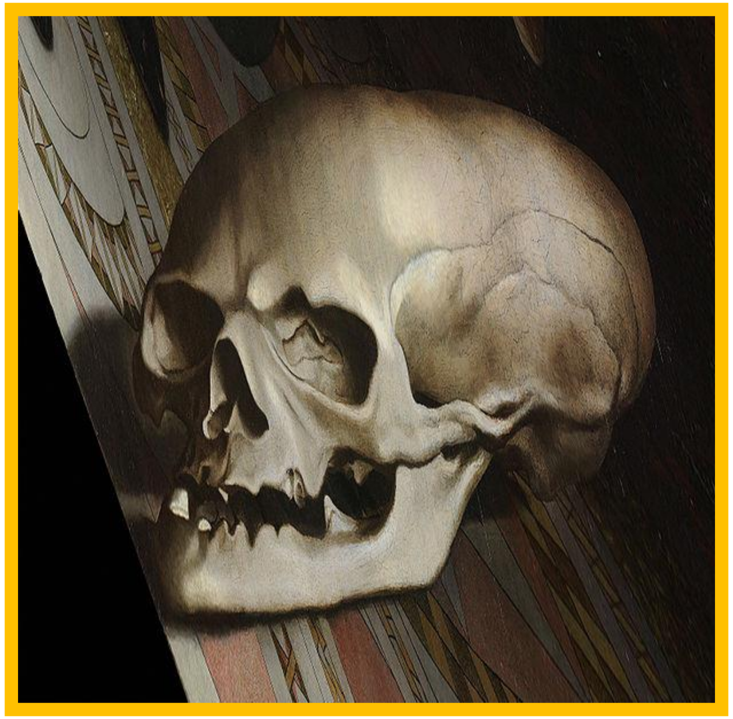
- Holbein, “The Ambassadors”

what’s this???

A weird drawing

- Holbein, "The Ambassadors"

skull under anamorphic perspective



rectified view

A weird drawing

- Holbein, “The Ambassadors”

DIY: use a polished spoon to see the skull

We can use homographies when…

1. The scene is planar

We can use homographies when…

2. The scene is very far or has small (relative) depth variation → scene is approximately planar

We can use homographies when…

3. The scene is captured under camera rotation only (no translation or pose change)

- More on why this is the case in a later lecture.

Computing with homographies

Applying a homography

1. Convert to homogeneous coordinates:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Multiply by the homography matrix (H):

$$P' = H \cdot P$$

- What is the size of the homography matrix?

3. Convert back to heterogeneous coordinates:

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \Rightarrow p' = \begin{bmatrix} x'/_{w'} \\ y'/_{w'} \end{bmatrix}$$

Applying a homography

1. Convert to homogeneous coordinates:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \;\Rightarrow\; P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Multiply by the homography matrix (H):

$$P' = H \cdot P$$

3. Convert back to heterogeneous coordinates:

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \;\Rightarrow\; p' = \begin{bmatrix} x'/_{w'} \\ y'/_{w'} \end{bmatrix}$$

- What is the size of the homography matrix?
- Answer: 3 x 3

Applying a homography

1. Convert to homogeneous coordinates:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \;\Rightarrow\; P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Multiply by the homography matrix (H):

$$P' = H \cdot P$$

3. Convert back to heterogeneous coordinates:

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \;\Rightarrow\; p' = \begin{bmatrix} x'/_{w'} \\ y'/_{w'} \end{bmatrix}$$

- What is the size of the homography matrix?
- Answer: 3 x 3
- How many degrees of freedom does the homography matrix have?

Applying a homography

1. Convert to homogeneous coordinates:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Multiply by the homography matrix (H):

$$P' = H \cdot P$$

- What is the size of the homography matrix?
- Answer: 3 x 3
- How many degrees of freedom does the homography matrix have?
- Answer: 8

3. Convert back to heterogeneous coordinates:

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \Rightarrow p' = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$$

Applying a homography

1. Convert to homogeneous coordinates:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Multiply by the homography matrix (H):

$$P' = H \cdot P$$

3. Convert back to heterogeneous coordinates:

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \Rightarrow p' = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$$

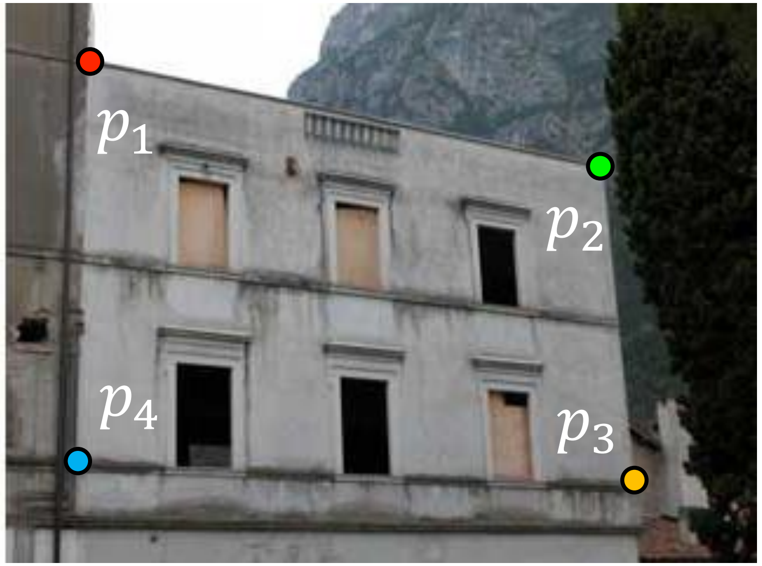
- What is the size of the homography matrix?
- Answer: 3 x 3
- How many degrees of freedom does the homography matrix have?
- Answer: 8
- How do we compute the homography matrix?

The direct linear transform (DLT)

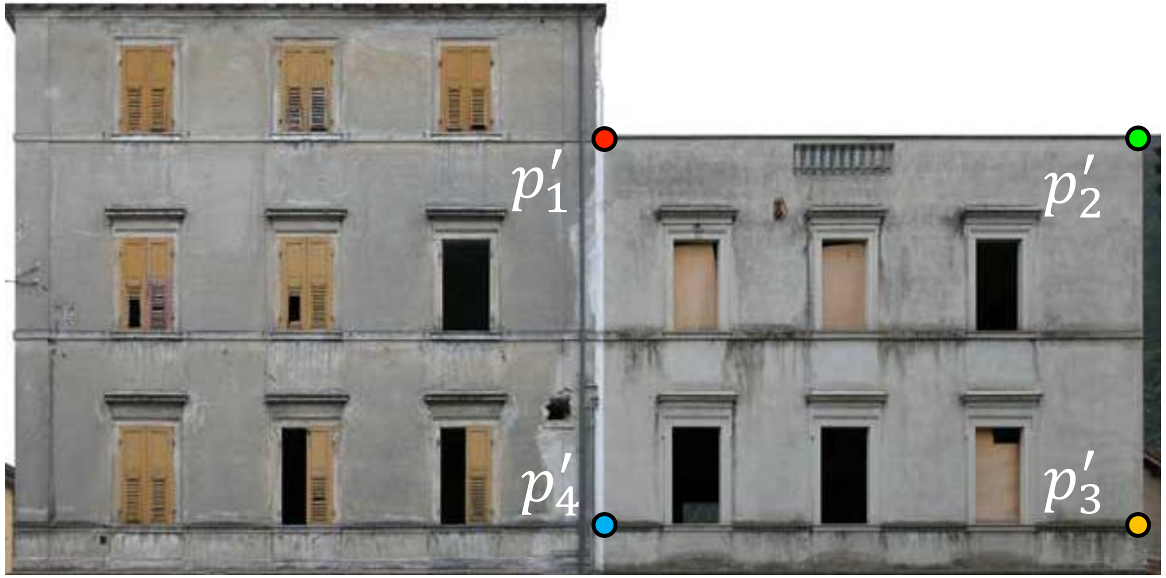
Create point correspondences

- Given a set of matched feature points $\{p_i, p_i'\}$ find the best estimate of H such that

$$P' = H \cdot P$$



original image



target image

- How many correspondences do we need?

Determining the homography matrix

- Write out linear equation for each correspondence:

$$P' = H \cdot P \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Determining the homography matrix

- Write out linear equation for each correspondence:

$$P' = H \cdot P \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Expand matrix multiplication:

$$x' = \alpha(h_1 x + h_2 y + h_3)$$
$$y' = \alpha(h_4 x + h_5 y + h_6)$$
$$1 = \alpha(h_7 x + h_8 y + h_9)$$

Determining the homography matrix

- Write out linear equation for each correspondence:

$$P' = H \cdot P \quad \text{or} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Expand matrix multiplication:

$$\begin{aligned} x' &= \alpha(h_1 x + h_2 y + h_3) \\ y' &= \alpha(h_4 x + h_5 y + h_6) \\ 1 &= \alpha(h_7 x + h_8 y + h_9) \end{aligned}$$

- Divide out unknown scale factor:

$$\begin{aligned} x'(h_7 x + h_8 y + h_9) &= (h_1 x + h_2 y + h_3) \\ y'(h_7 x + h_8 y + h_9) &= (h_4 x + h_5 y + h_6) \end{aligned}$$

how do you rearrange terms to make it a linear system?

Determining the homography matrix

- How do you rearrange terms to make it a linear system?

$$x'(h_7x + h_8y + h_9) = (h_1x + h_2y + h_3)$$
$$y'(h_7x + h_8y + h_9) = (h_4x + h_5y + h_6)$$

just rearrange the terms

$$h_7xx' + h_8yx' + h_9x' - h_1x - h_2y - h_3 = 0$$
$$h_7xy' + h_8yy' + h_9y' - h_4x - h_5y - h_6 = 0$$

Determining the homography matrix

- Re-arrange terms:

$$h_7 x x' + h_8 y x' + h_9 x' - h_1 x - h_2 y - h_3 = 0$$
$$h_7 x y' + h_8 y y' + h_9 y' - h_4 x - h_5 y - h_6 = 0$$

- Re-write in matrix form:

$$\mathbf{A}_i \boldsymbol{h} = \mathbf{0}$$

$$\mathbf{A}_i = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix}$$

$$\boldsymbol{h} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{bmatrix}^\top$$

- How many equations from one point correspondence?

Determining the homography matrix

- Stack together constraints from multiple point correspondences:

$$\mathbf{A}\boldsymbol{h} = \mathbf{0}$$

$$\begin{bmatrix} \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \vdots \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Homogeneous linear least squares problem

Reminder: Determining affine transformations

- Affine transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Vectorize transformation parameters:
- Stack equations from point correspondences:
- Notation in system form:

$$\underbrace{\begin{bmatrix} x' \\ y' \\ x' \\ y' \\ \vdots \\ x' \\ y' \end{bmatrix}}_{\boldsymbol{b}} = \underbrace{\begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ & & & \vdots & & \\ x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix}}_{\boldsymbol{x}}$$

$$\boxed{Ax = b}$$

Reminder: Determining affine transformations

- Convert the system to a linear least-squares problem:

$$E_{\mathrm{LLS}} = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$$

- Expand the error:

$$E_{\mathrm{LLS}} = \boldsymbol{x}^\top(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} - 2\boldsymbol{x}^\top(\mathbf{A}^\top\boldsymbol{b}) + \|\boldsymbol{b}\|^2$$

- Minimize the error:

$$(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} = \mathbf{A}^\top\boldsymbol{b}$$

in matlab

x = A \ b

- Set derivative to 0, solve for x

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

note: you almost <u>never</u> want to compute the inverse of a matrix.

Determining the homography matrix

- Stack together constraints from multiple point correspondences:

$$\mathbf{A}\boldsymbol{h} = \mathbf{0}$$

$$\begin{bmatrix} \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \vdots \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Homogeneous linear least squares problem

- How do we solve this?

Determining the homography matrix

- Stack together constraints from multiple point correspondences:

$$\mathbf{A}\boldsymbol{h} = \mathbf{0}$$

$$\begin{bmatrix} \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \vdots \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Homogeneous linear least squares problem

- How do we solve this? Solve with SVD

Singular value decomposition

orthonormal diagonal orthonormal

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

unit norm constraint

n x m n x n n x m m x m

$$= \sum_{i=1}^{9} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

n x 1 1 x m

Singular value decomposition

- General form of total least squares
  (warning: change of notation. x is a vector of parameters!)

$$E_{\mathrm{TLS}} = \sum_i (\boldsymbol{a}_i \boldsymbol{x})^2$$

$$= \|\mathbf{A}\boldsymbol{x}\|^2 \quad \text{(matrix form)}$$

$$\|\boldsymbol{x}\|^2 = 1 \quad \text{constraint}$$

minimize $\|\mathbf{A}\boldsymbol{x}\|^2$

subject to $\|\boldsymbol{x}\|^2 = 1$

(equivalent)

(Rayleigh quotient)

minimize $\dfrac{\|\mathbf{A}\boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^2}$

- Solution is the eigenvector corresponding to smallest eigenvalue of $A^T A$

- Solution is the column of V corresponding to smallest singular value $A = U\Sigma V^T$

Solving for H using DLT

- Given $\{x_i, x_i'\}$ solve for H such that $x' = Hx$

1. For each correspondence, create 2x9 matrix A_i

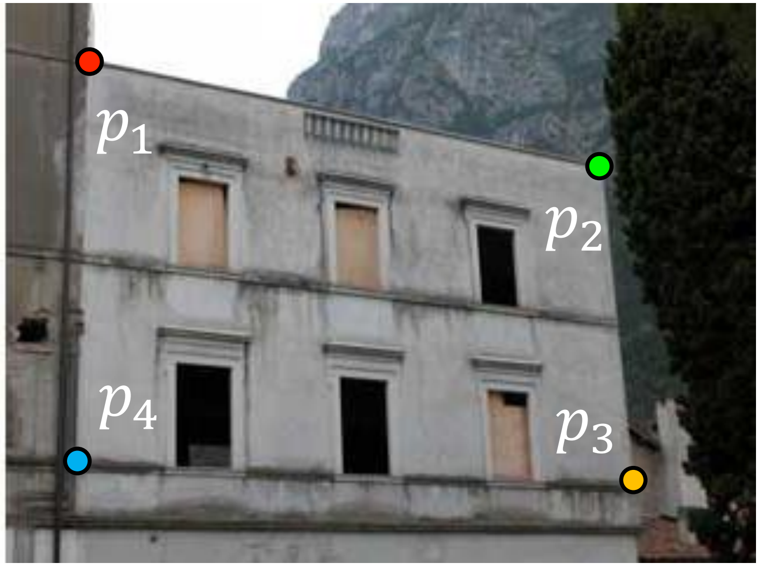
2. Concatenate into single 2n x 9 matrix A

3. Compute SVD of $A = U\Sigma V^T$

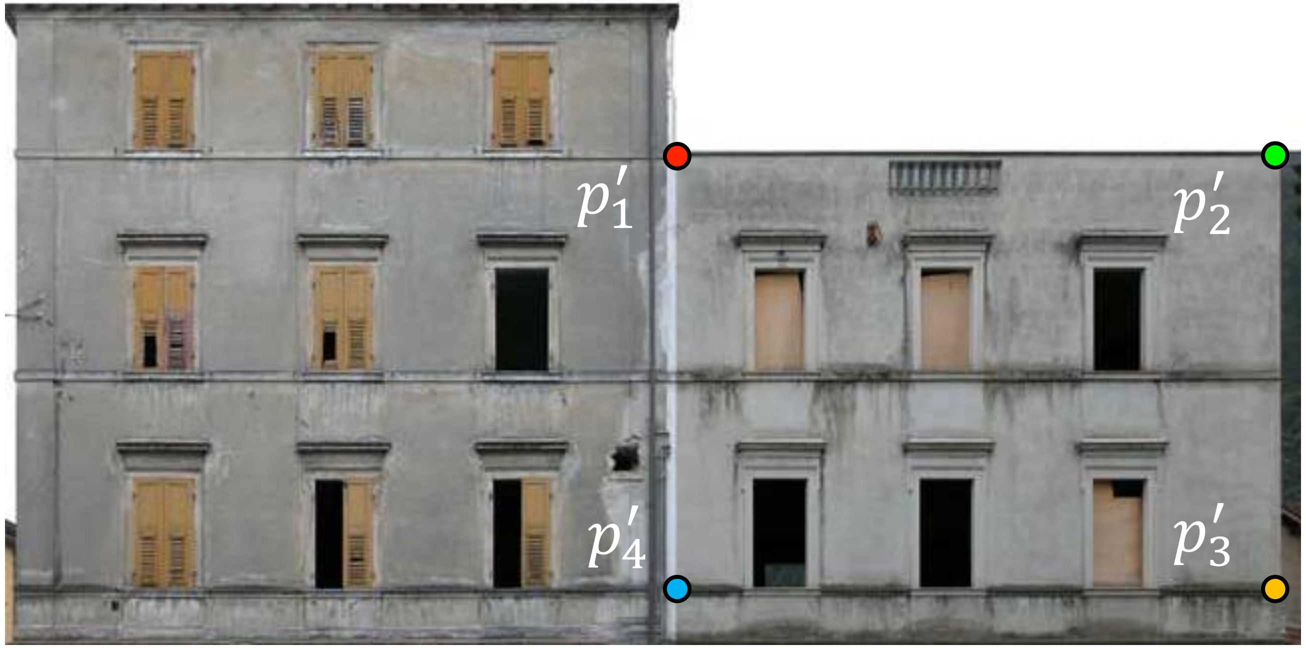
4. Store singular vector of the smallest singular value $h = \boldsymbol{v}_{\hat{i}}$

5. Reshape to get H

Create point correspondences



original image



target image

- How do we automate this step?

The image correspondence pipeline

- Feature point detection
  - Detect corners using the Harris corner detector.

- Feature point description
  - Describe features using the multi-scale oriented patch descriptor.

- Feature matching

The image correspondence pipeline

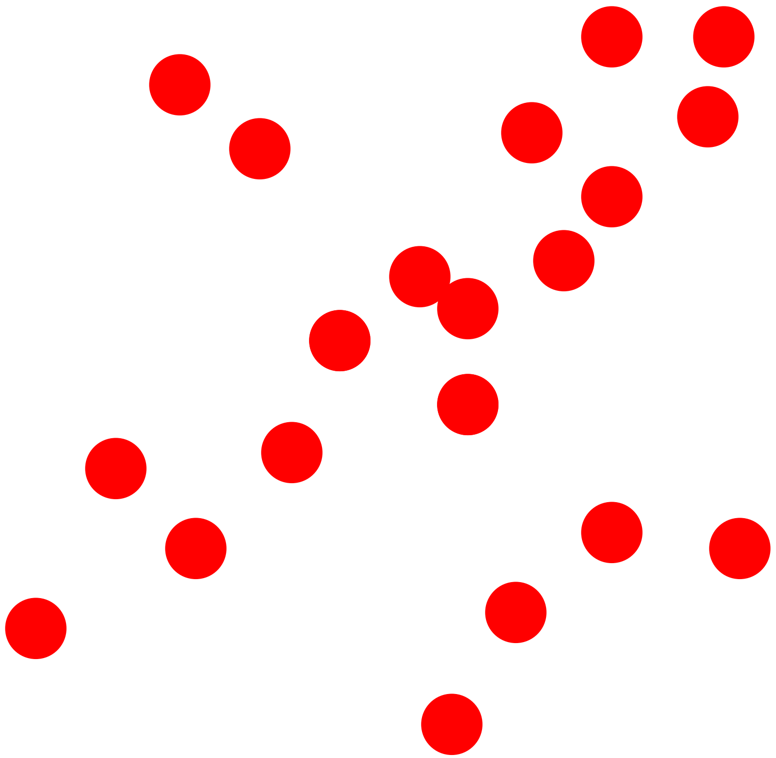
- Feature point detection
  - Detect corners using the Harris corner detector.
- Feature point description
  - Describe features using the multi-scale oriented patch descriptor.
- Feature matching


bad correspondence
good correspondence

Random Sample Consensus (RANSAC)

Random Sample Consensus (RANSAC)

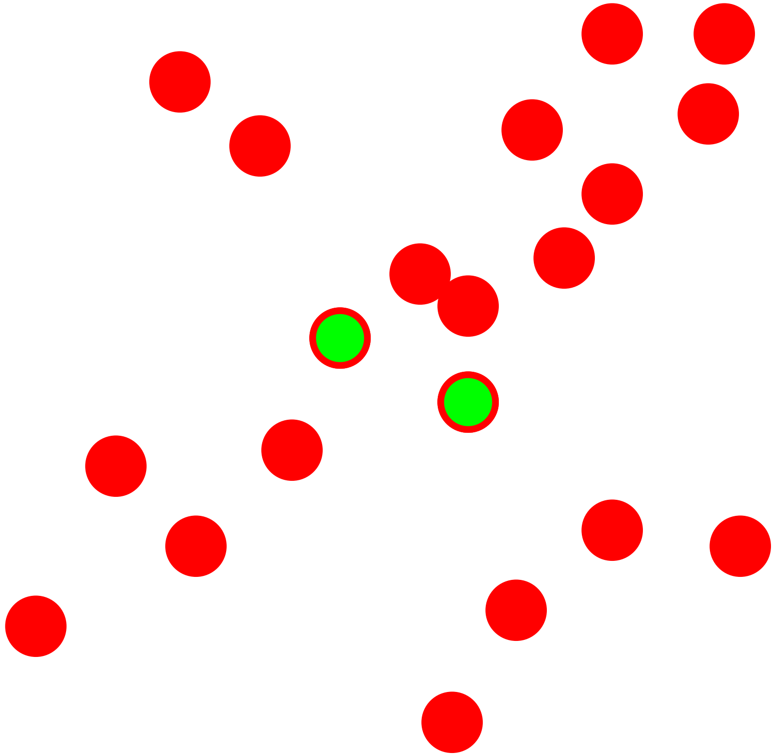
fitting lines
(with outliers)



- **Algorithm:**
  - Sample (randomly) the number of points required to fit the model
  - Solve for model parameters using samples
  - Score by the fraction of inliers within a preset threshold of the model
  - Repeat 1-3 until the best model is found with high confidence

Random Sample Consensus (RANSAC)

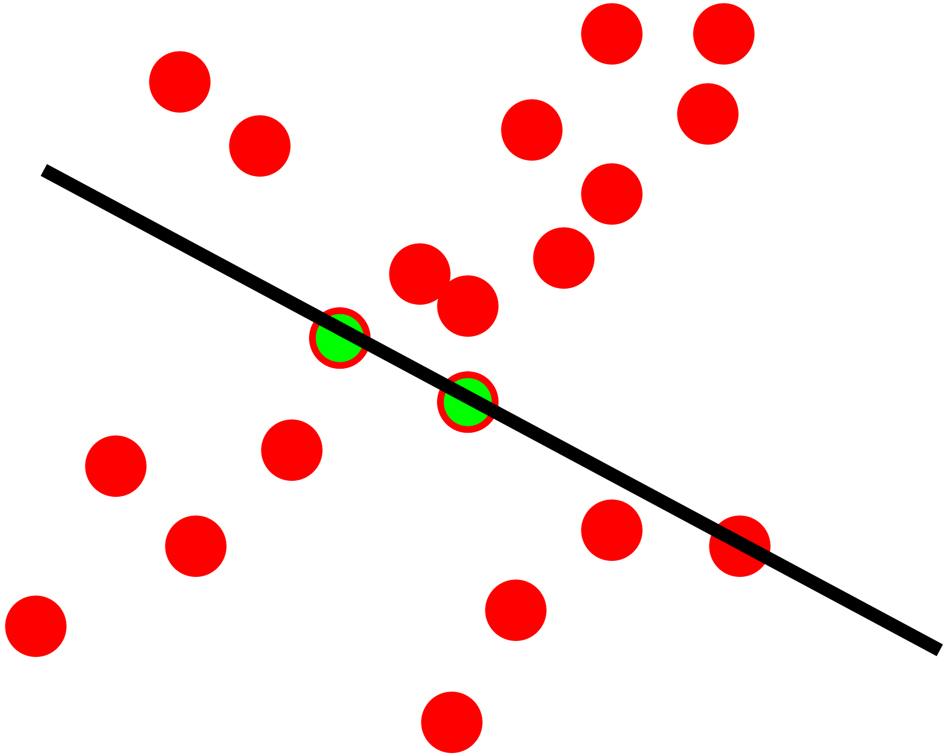
fitting lines
(with outliers)



- Algorithm:
  - **Sample (randomly) the number of points required to fit the model**
  - Solve for model parameters using samples
  - Score by the fraction of inliers within a preset threshold of the model
  - Repeat 1-3 until the best model is found with high confidence

Random Sample Consensus (RANSAC)

fitting lines
(with outliers)

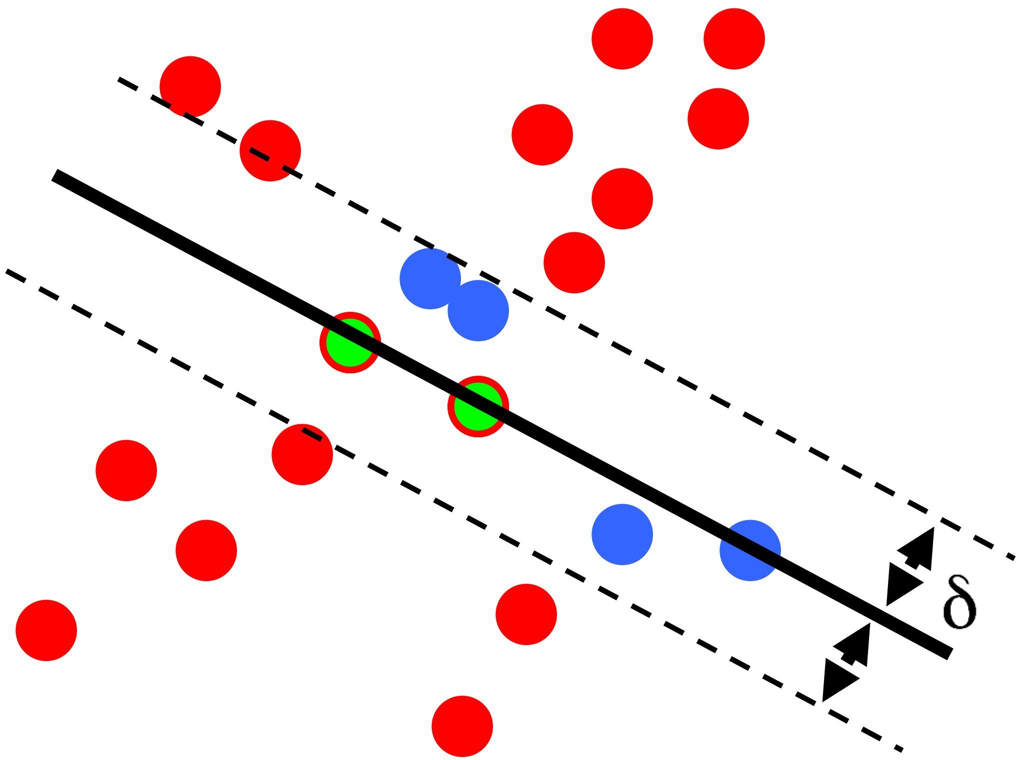


- **Algorithm:**
  - Sample (randomly) the number of points required to fit the model
  - **Solve for model parameters using samples**
  - Score by the fraction of inliers within a preset threshold of the model
  - Repeat 1-3 until the best model is found with high confidence

Random Sample Consensus (RANSAC)

fitting lines
(with outliers)

$$N_I = 6$$

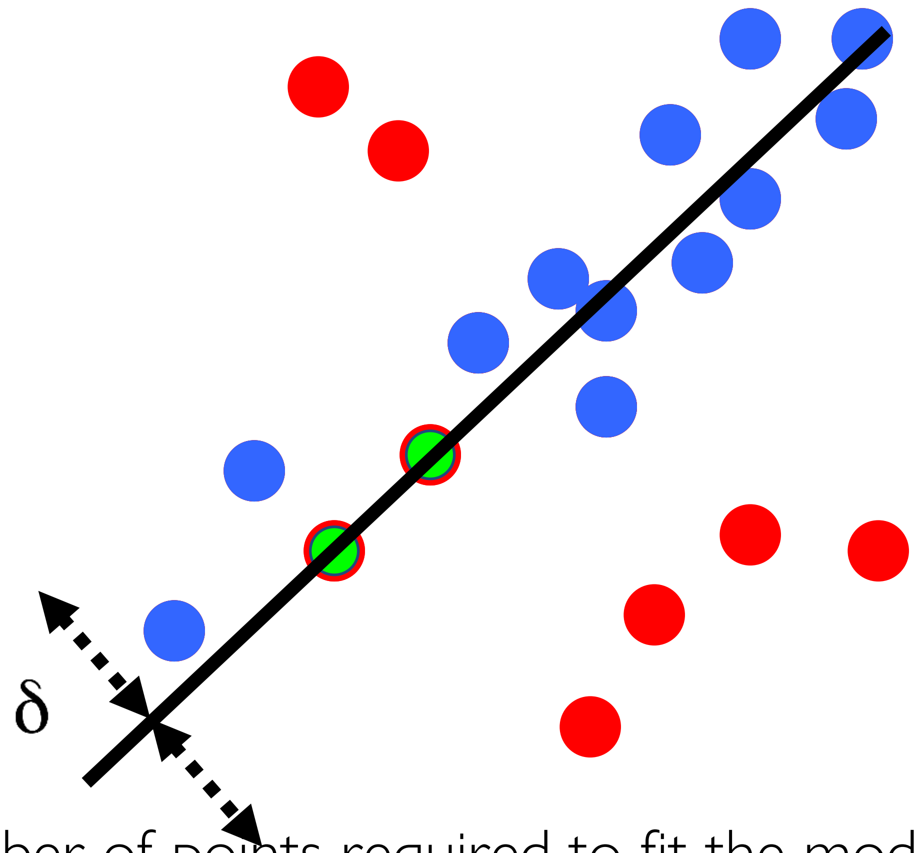


- **Algorithm:**
  - Sample (randomly) the number of points required to fit the model
  - Solve for model parameters using samples
  - Score by the fraction of inliers within a preset threshold of the model
  - Repeat 1-3 until the best model is found with high confidence

Random Sample Consensus (RANSAC)

fitting lines
(with outliers)

$$N_I = 14$$



- Algorithm:
  - Sample (randomly) the number of points required to fit the model
  - Solve for model parameters using samples
  - Score by the fraction of inliers within a preset threshold of the model
  - **Repeat 1-3 until the best model is found with high confidence**

How to choose parameters?

- Number of samples N
  - Choose N so that, with probability p, at least one random sample is free from outliers (e.g. p=0.99) (outlier ratio: e)

- Number of sampled points s
  - Minimum number needed to fit the model

- Distance threshold δ
  - Choose δ so that a good point with noise is likely (e.g., prob=0.95) within threshold

$$N = \frac{\log(1-p)}{\log\left(1-(1-e)^s\right)}$$

	proportion of outliers *e*						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

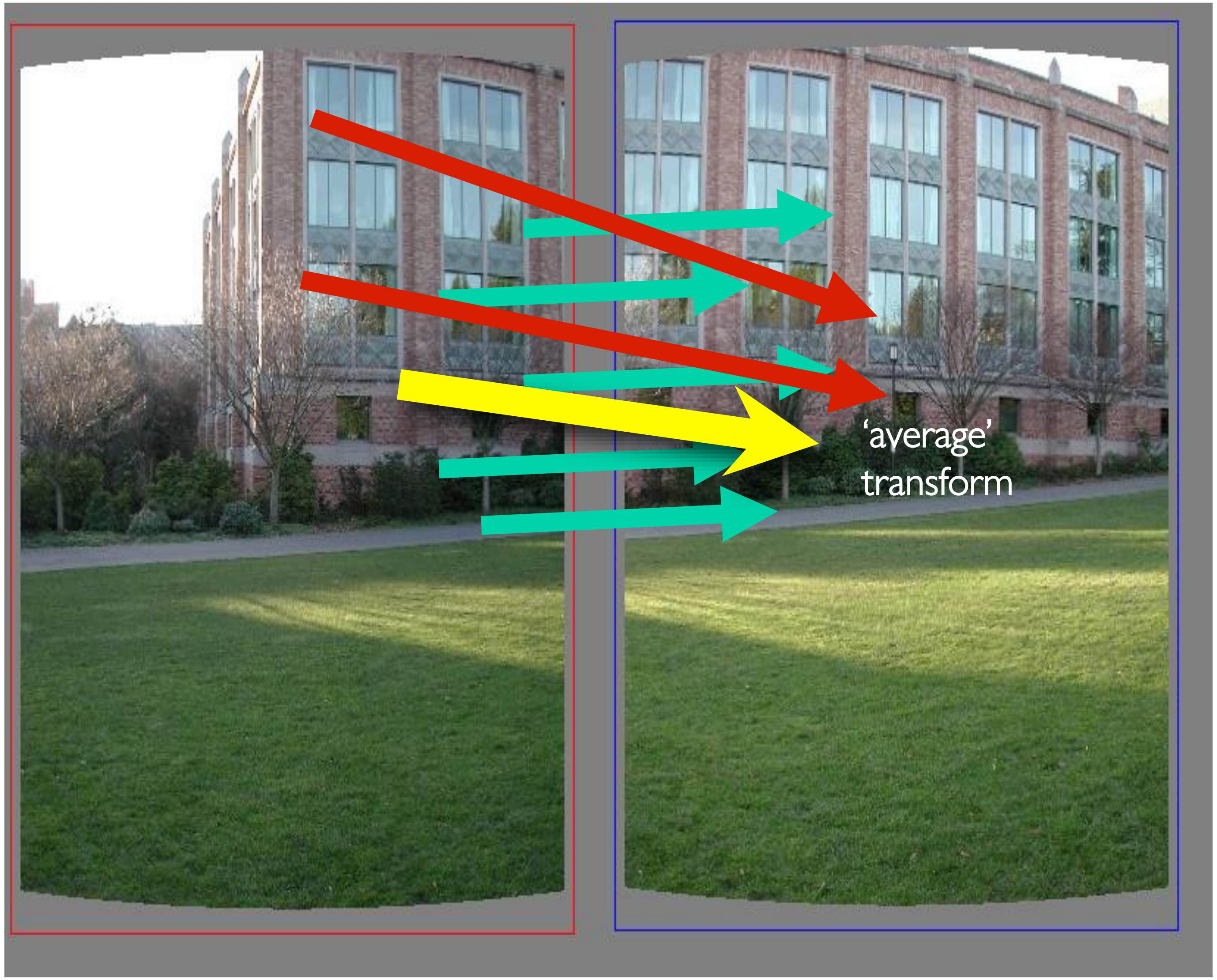
Given two images…

find matching features (e.g., SIFT) and a translation transform

Matched points will usually contain bad correspondences

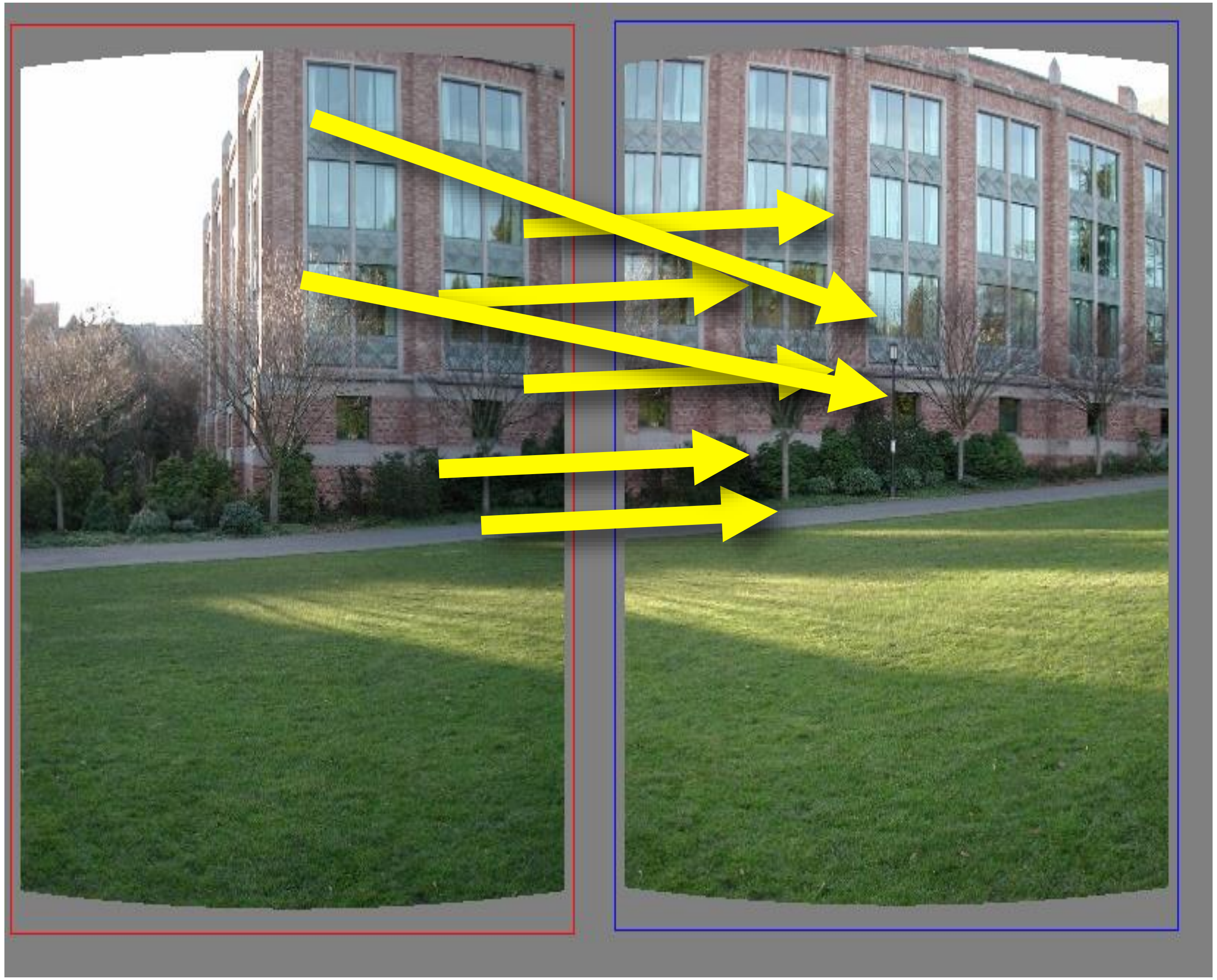
how should we estimate the transform?

LLS will find the 'average' transform

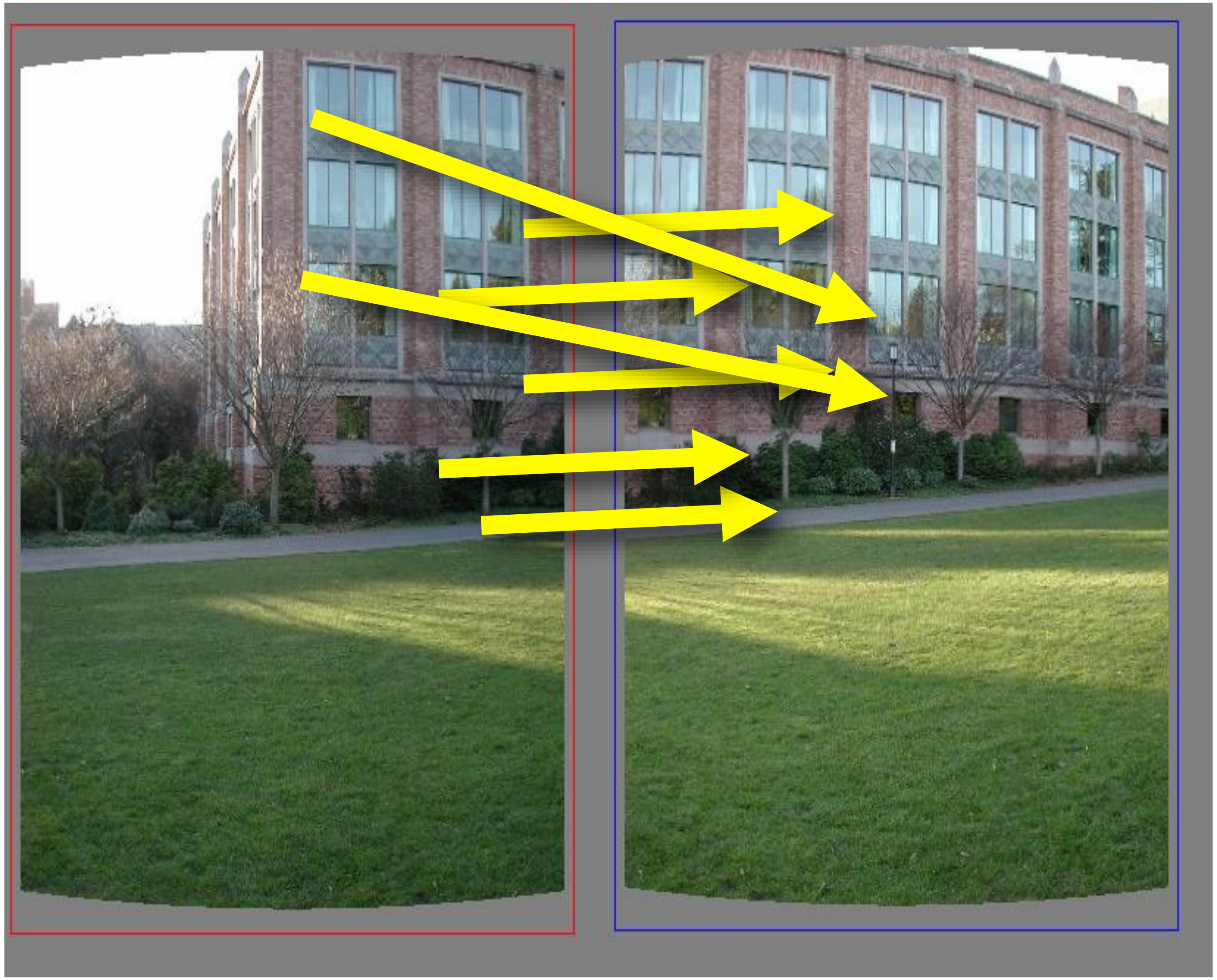


solution is corrupted by bad correspondences

Use RANSAC

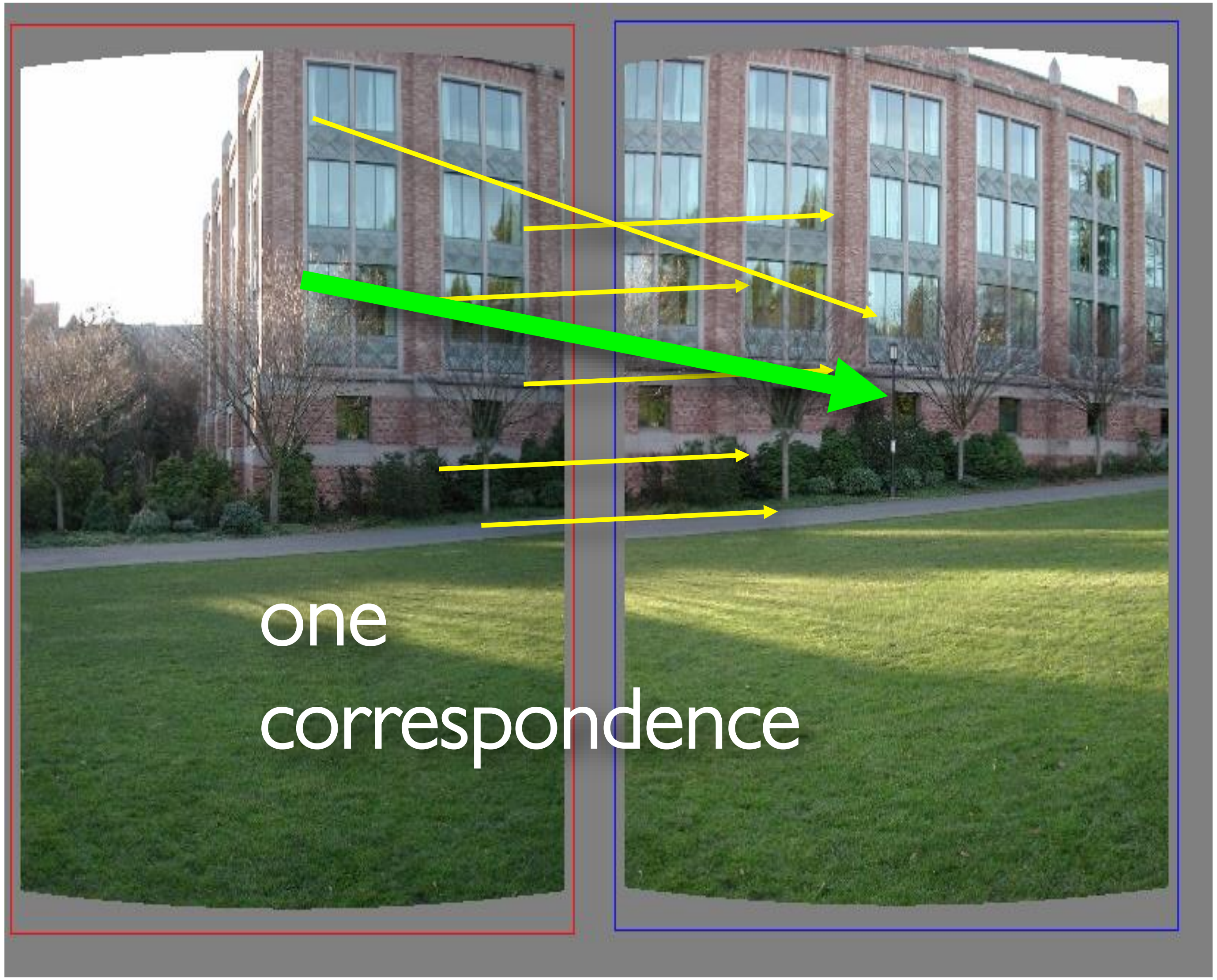


how many correspondences to compute translation transform?

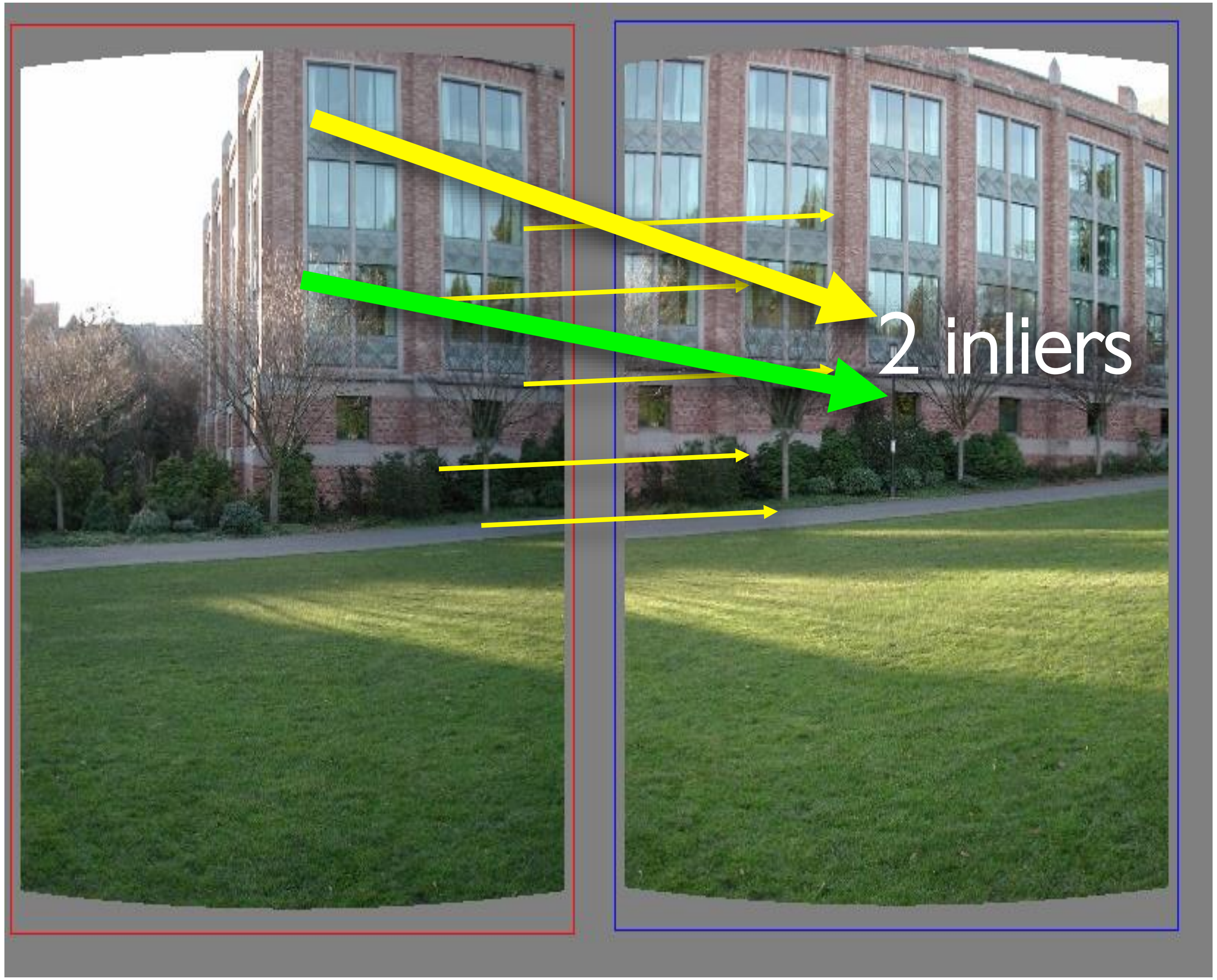


need only **one correspondence**, to find translation model

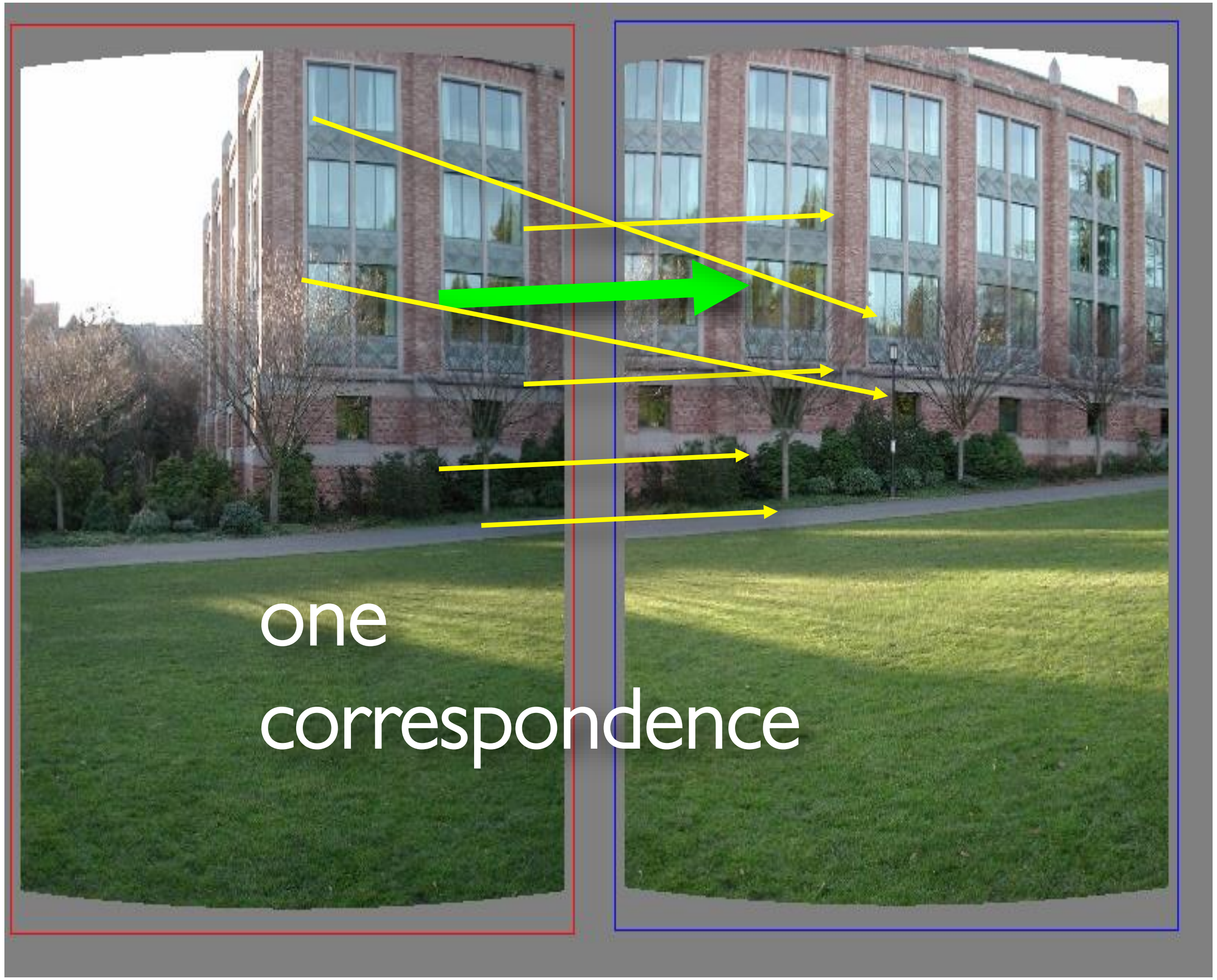
Pick one correspondence, count inliers



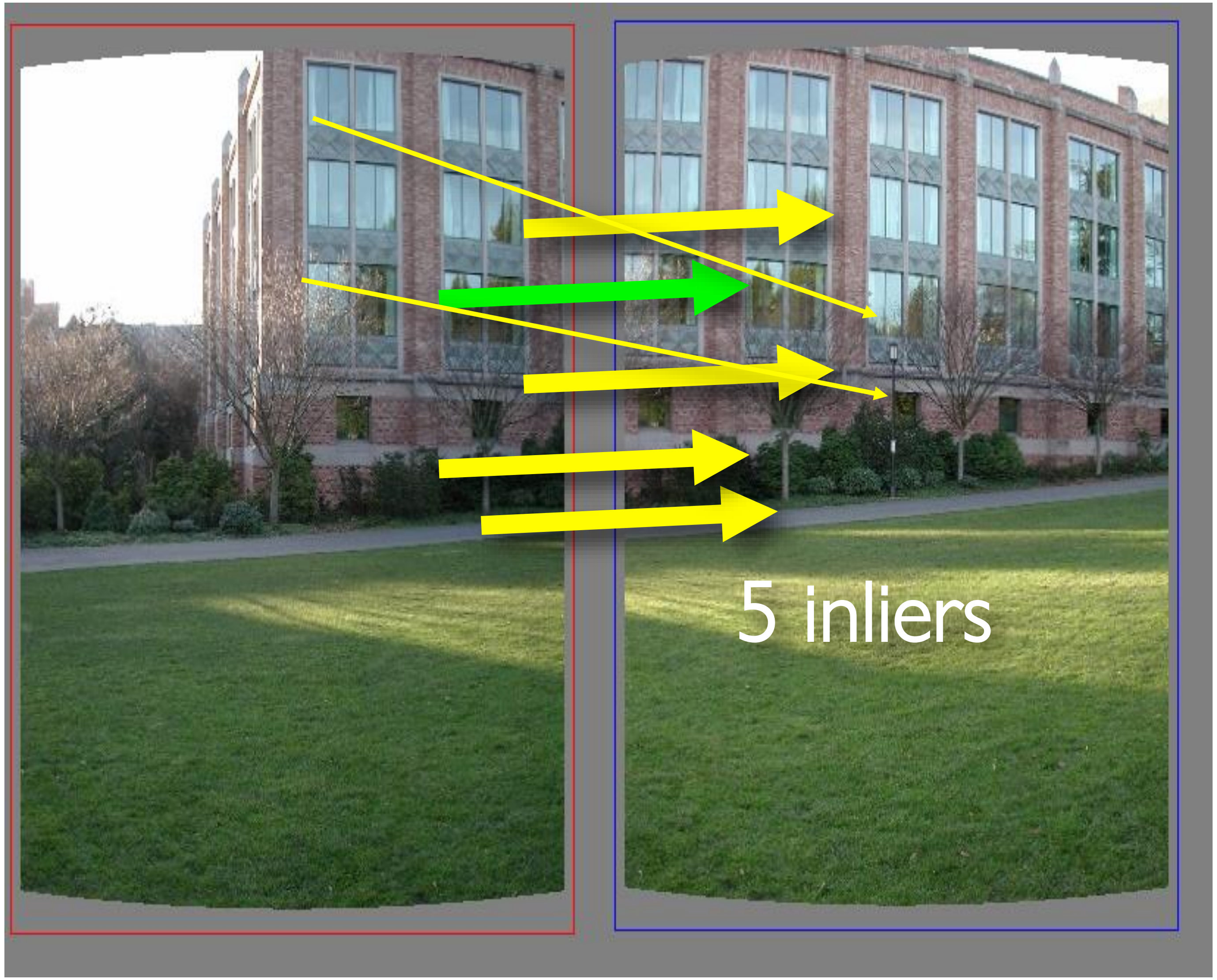
Pick one correspondence, count inliers



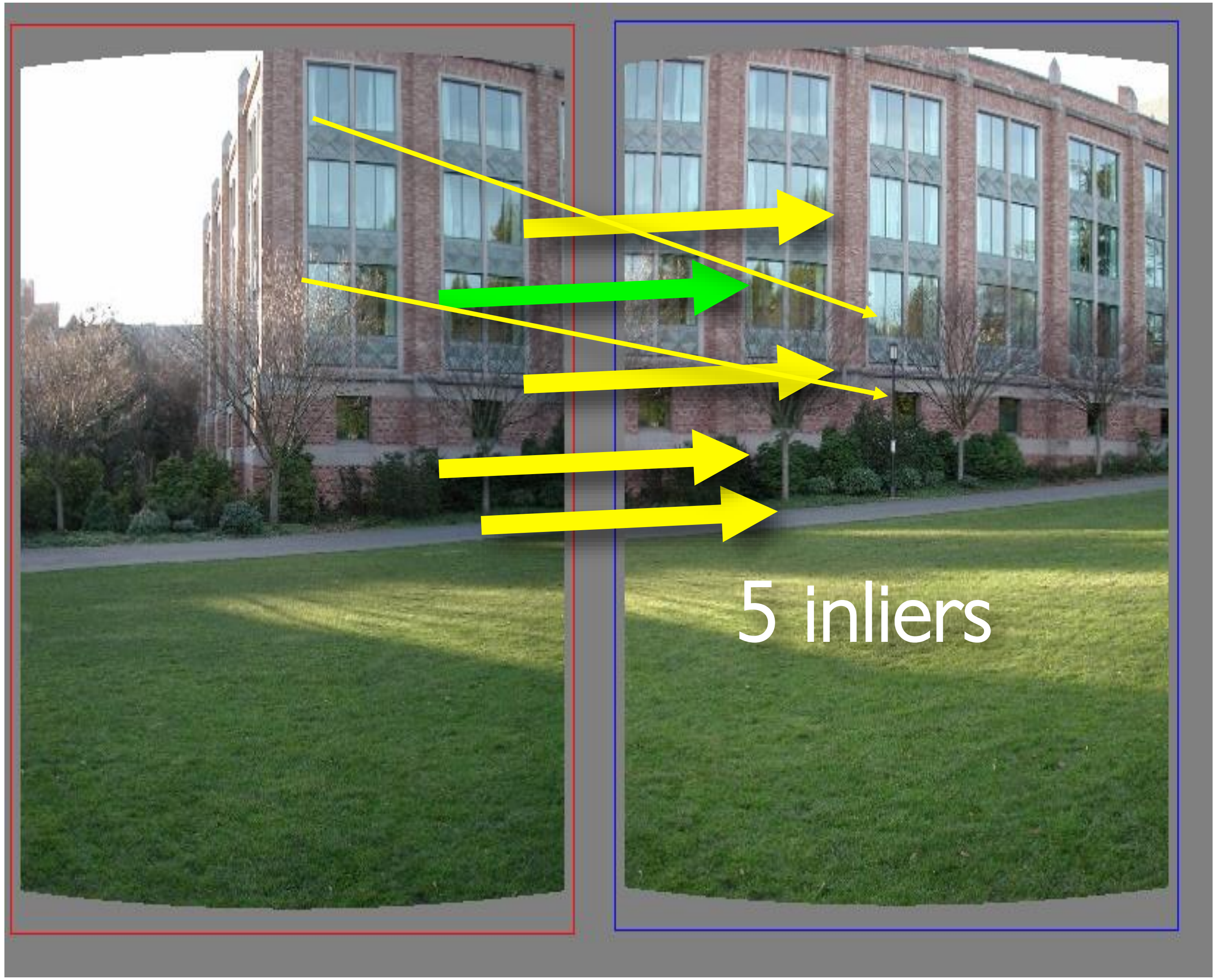
Pick one correspondence, count inliers



Pick one correspondence, count inliers



Pick one correspondence, count inliers



pick the model with the highest number of inliers!

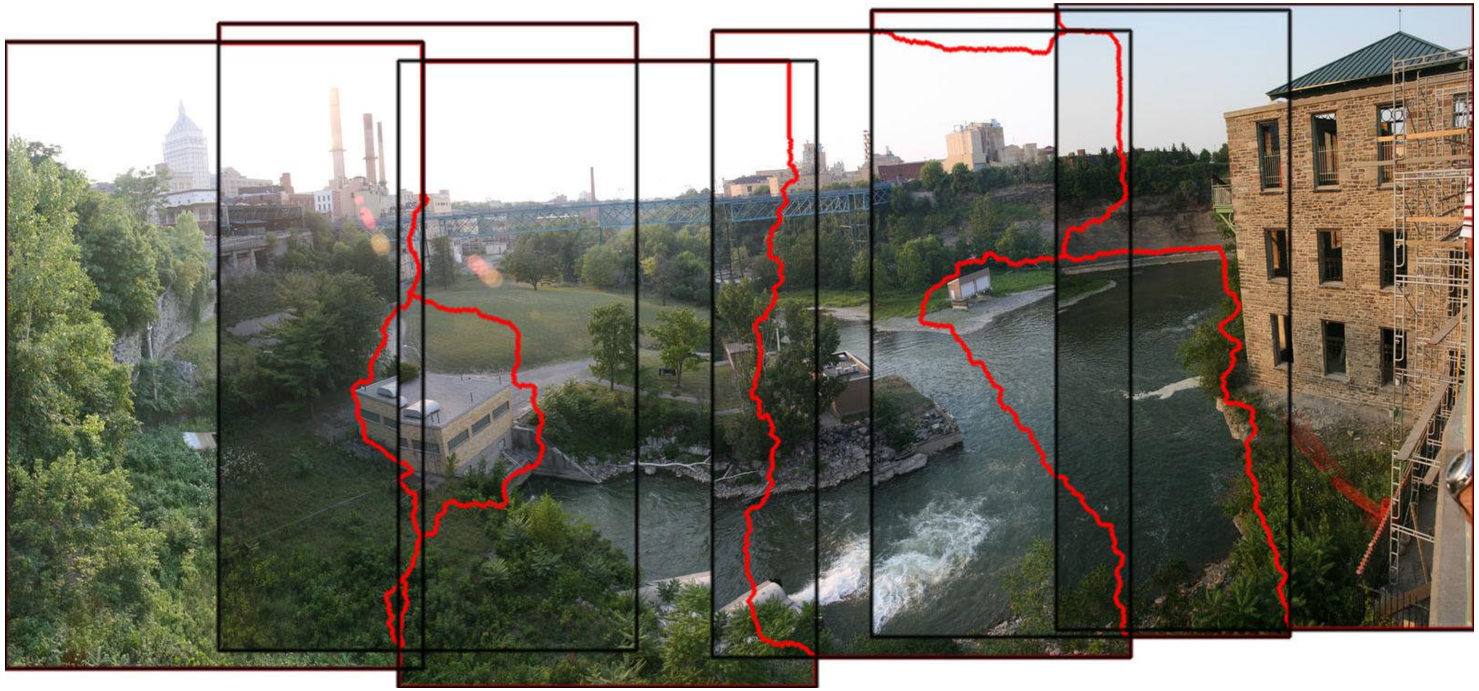
Estimating homography using RANSAC

- RANSAC loop
  - Get four point correspondences (randomly)
  - Compute H using DLT
  - Count inliers
  - Keep H if largest number of inliers

- Recompute H using all inliers

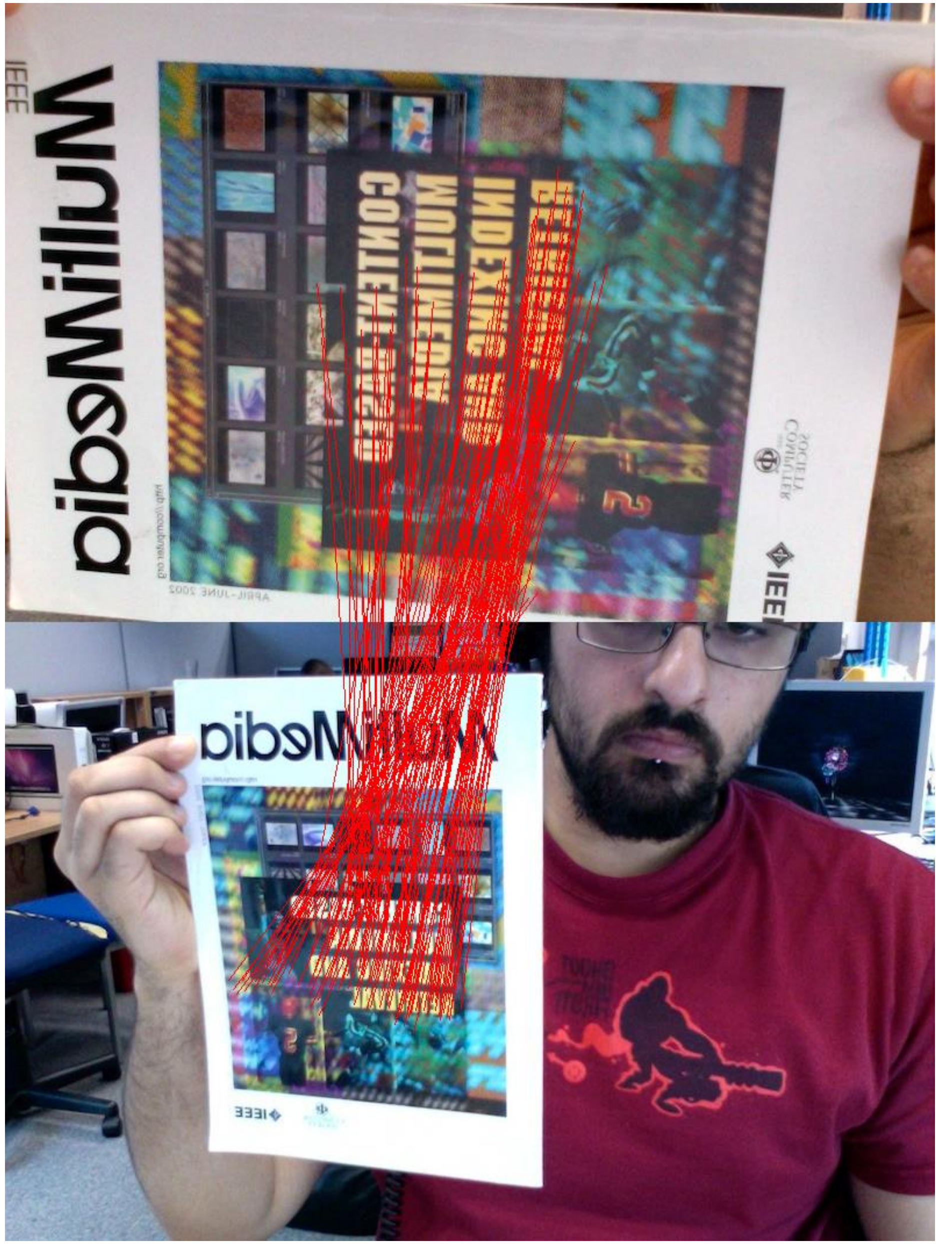
Estimating homography using RANSAC

- Useful for…



Estimating homography using RANSAC

- Useful for…



The image correspondence pipeline

- Feature point detection
  - Detect corners using the Harris corner detector.
- Feature point description
  - Describe features using the multi-scale oriented patch descriptor.
- Feature matching and homography estimation
  - Do both simultaneously using RANSAC.