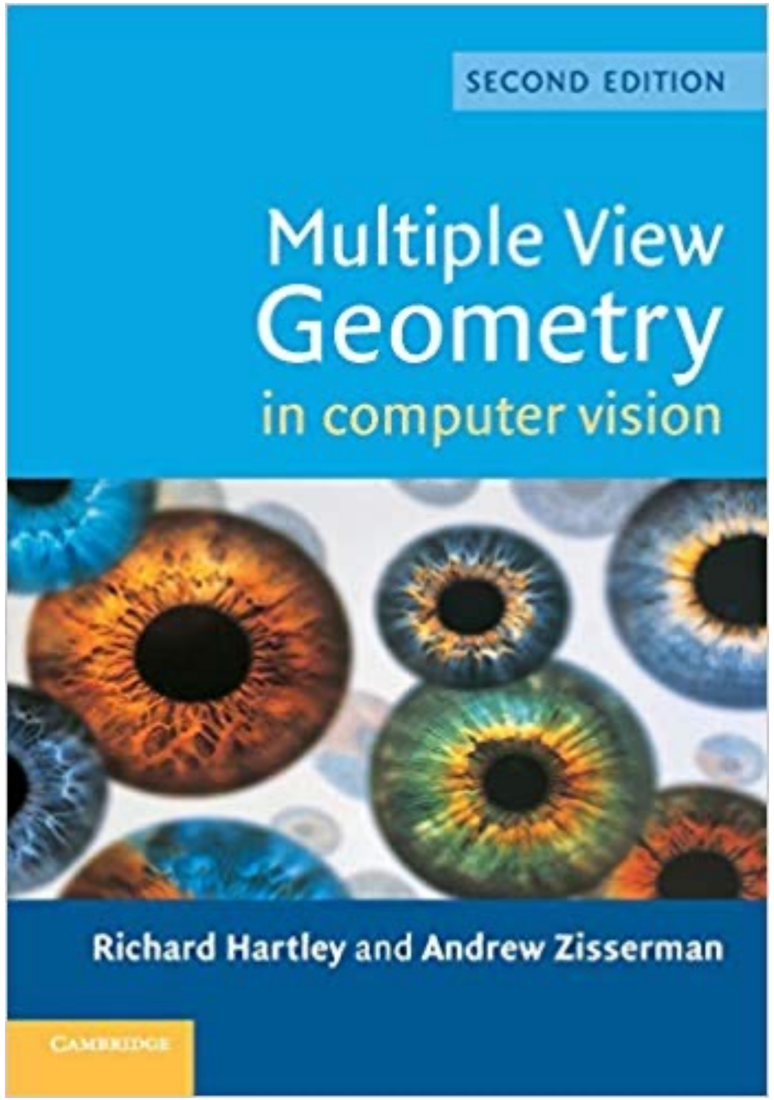
Image homographies

http://16385.courses.cs.cmu.edu/

16-385 Computer Vision
Spring 2024, Lecture 8

Textbook for geometry part of class

- Amazing resource for everything related to geometric methods in computer vision.
- Great introduction to projective geometry as well.



Overview of today’s lecture

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

Slide credits

Most of these slides were adapted from:

- Kris Kitani (15-463, Fall 2016).
- Noah Snavely (Cornell).

Motivation for image alignment: panoramas.

How do you create a panorama?

Panorama: an image of (near) 360^o field of view.

How do you create a panorama?

Panorama: an image of (near) 360^o field of view.

1. Use a very wide-angle lens.

Wide-angle lenses

Fish-eye lens: can produce (near) hemispherical field of view.

What are the pros and cons of this?

How do you create a panorama?

Panorama: an image of (near) 360^o field of view.

1. Use a very wide-angle lens.
- Pros: Everything is done optically, single capture.
- Cons: Lens is super expensive and bulky, lots of distortion (can be dealt-with in post).

Any alternative to this?

How do you create a panorama?

Panorama: an image of (near) 360^o field of view.

1. Use a very wide-angle lens.
   - Pros: Everything is done optically, single capture.
   - Cons: Lens is super expensive and bulky, lots of distortion (can be dealt-with in post).

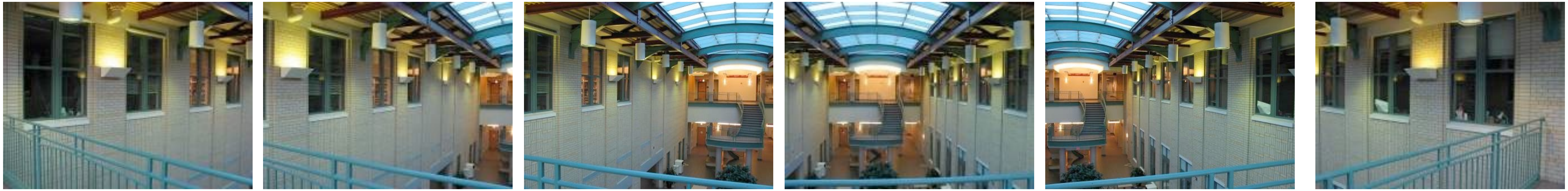
2. Capture multiple images and combine them.

Panoramas from image stitching

1. Capture multiple images from different viewpoints.

2. Stitch them together into a virtual wide-angle image.

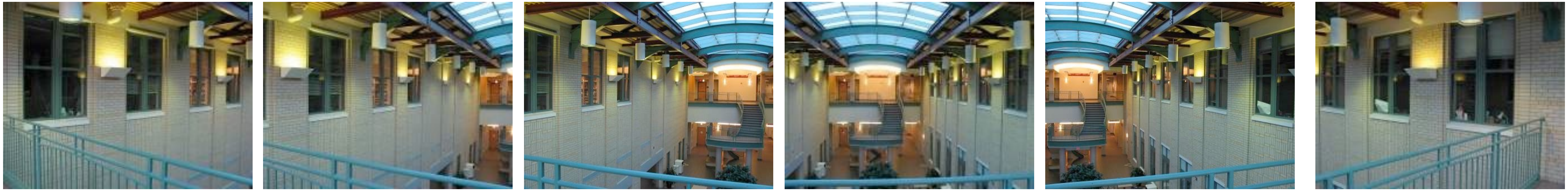
How do we stitch images from different viewpoints?



Will standard stitching work?

1. Translate one image relative to another.
2. (Optionally) find an optimal seam.

How do we stitch images from different viewpoints?



Will standard stitching work?

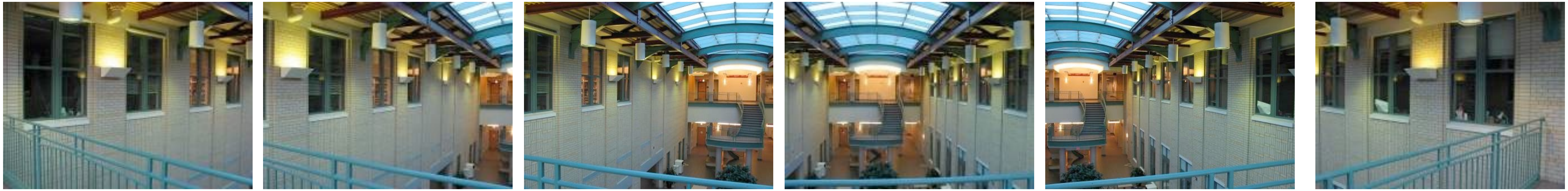
1. Translate one image relative to another.
2. (Optionally) find an optimal seam.

left on top

right on top

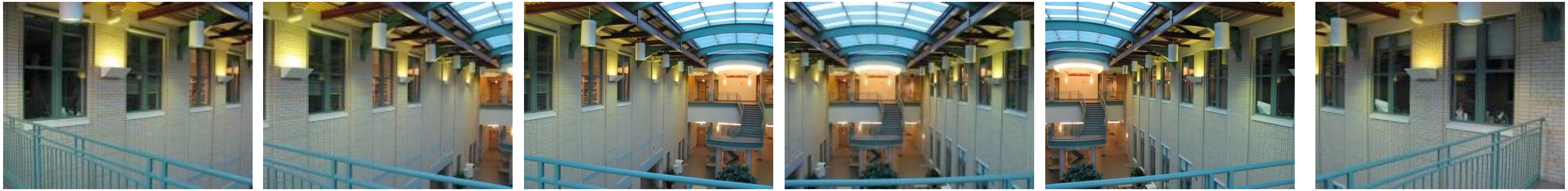
Translation-only stitching is not enough to mosaic these images.

How do we stitch images from different viewpoints?



What else can we try?

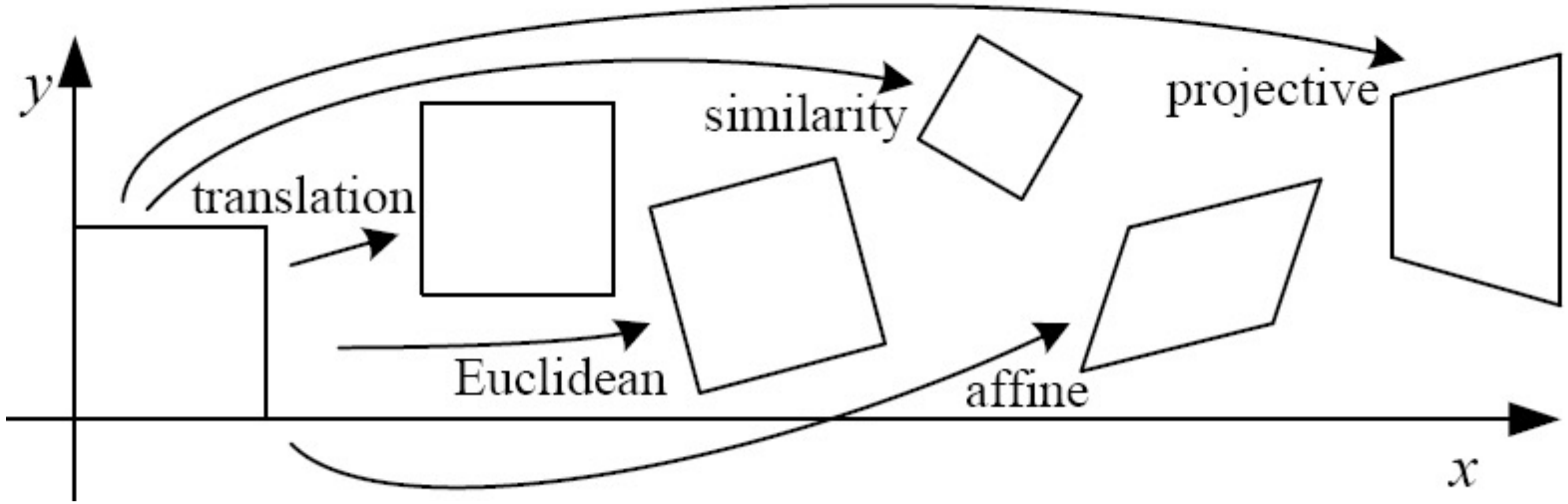
How do we stitch images from different viewpoints?



Use image homographies.

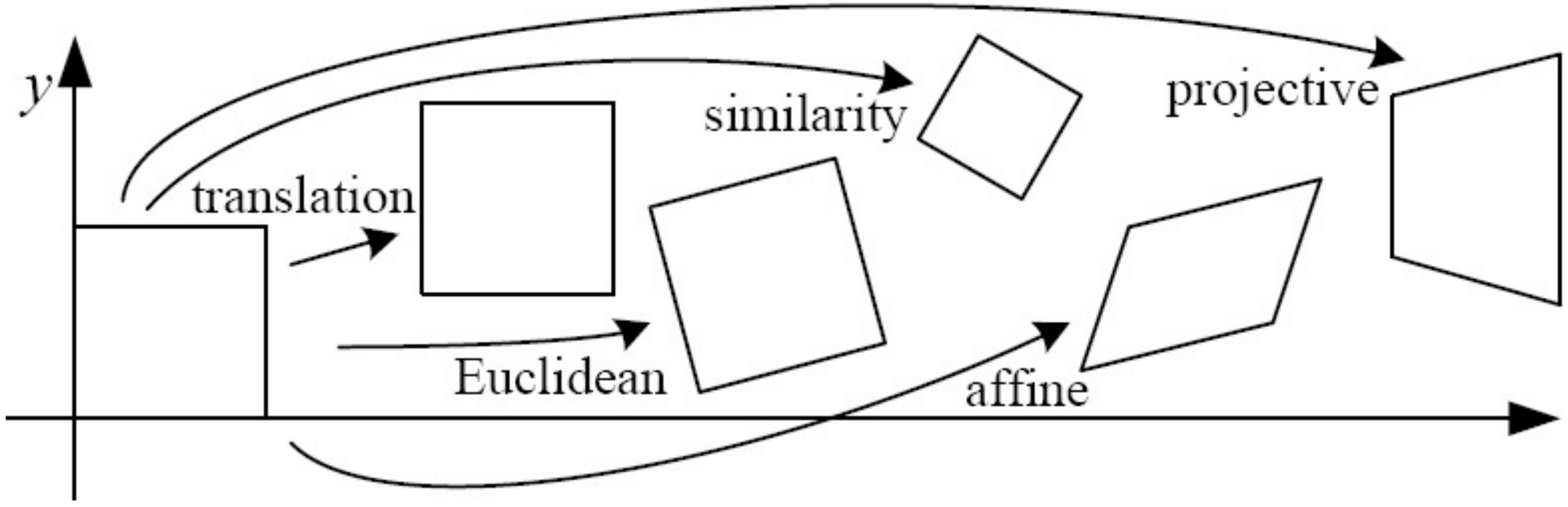
Back to warping: image homographies

Classification of 2D transformations

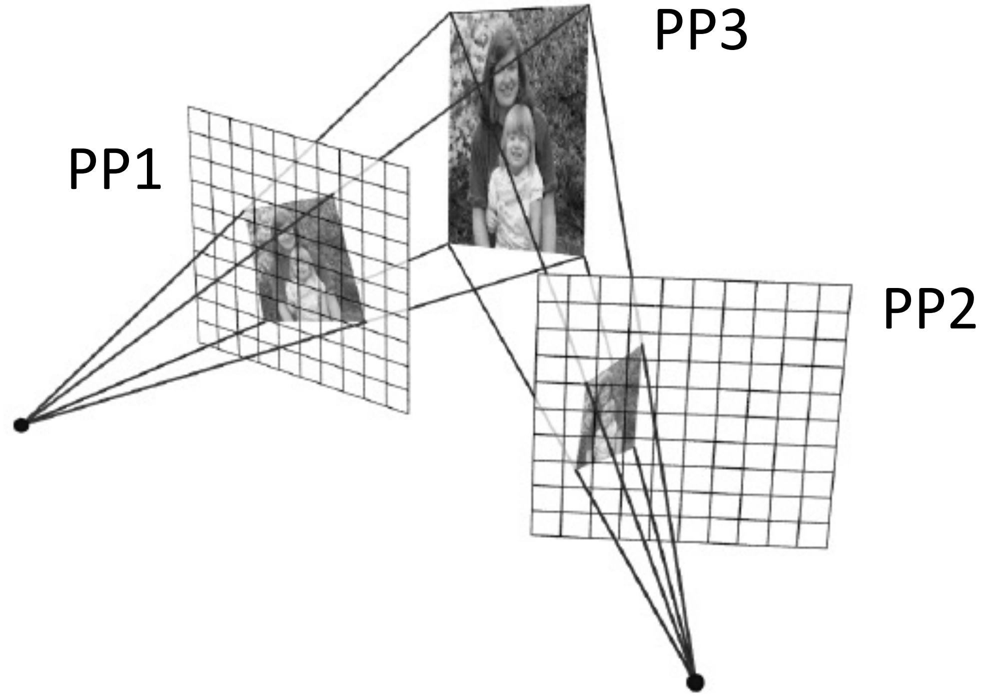


Name	Matrix	# D.O.F.
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$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

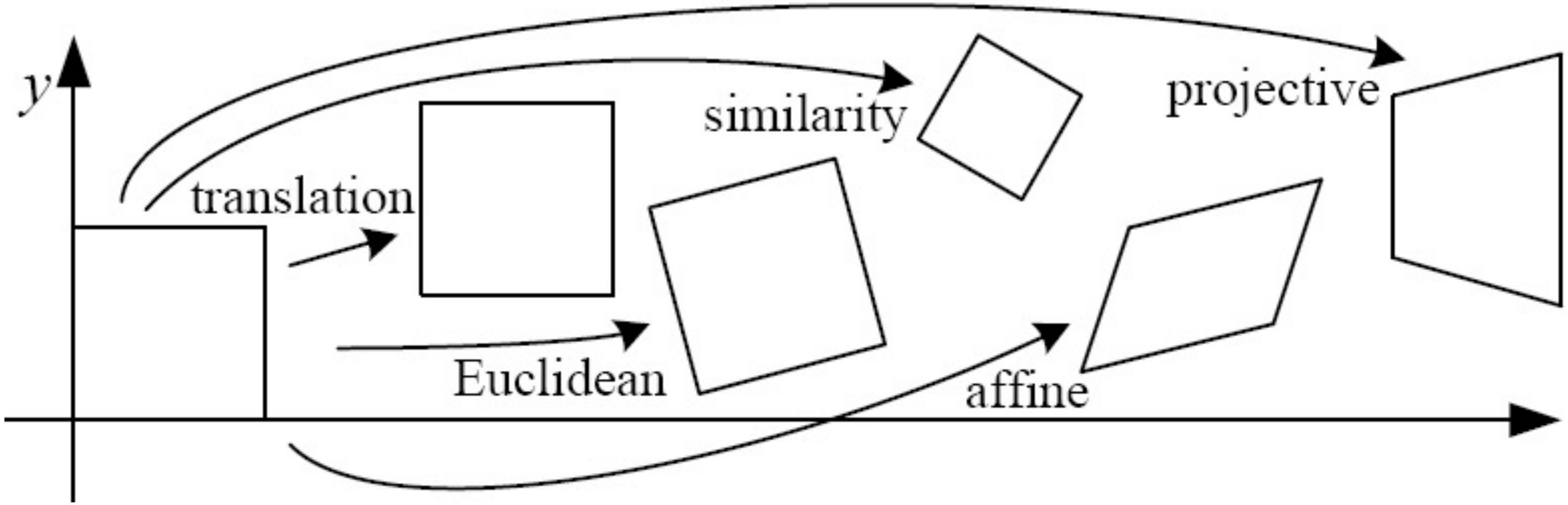
Classification of 2D transformations



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

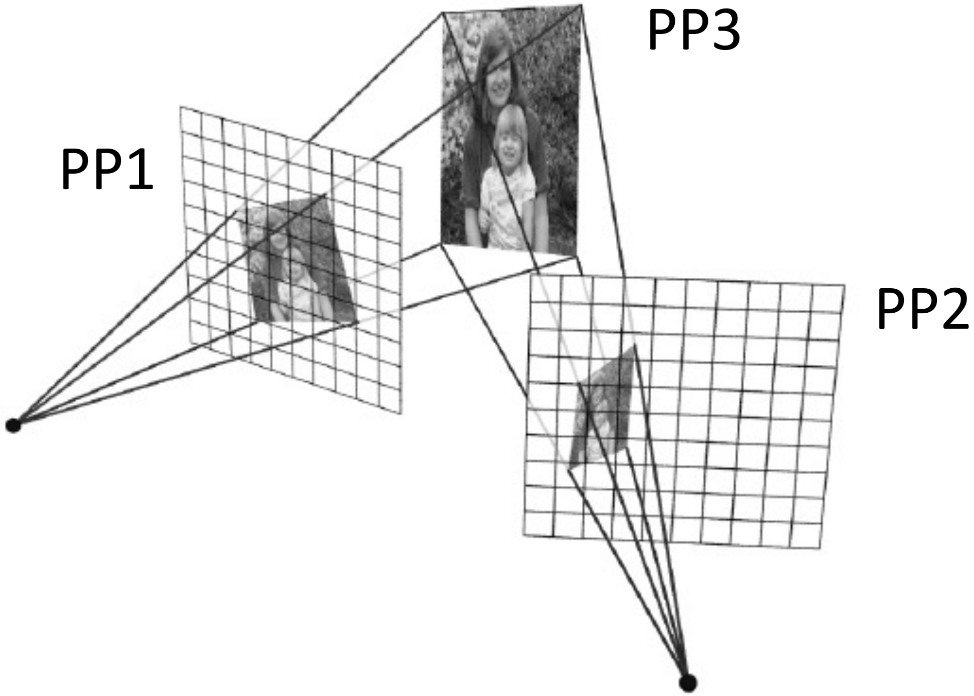


Classification of 2D transformations



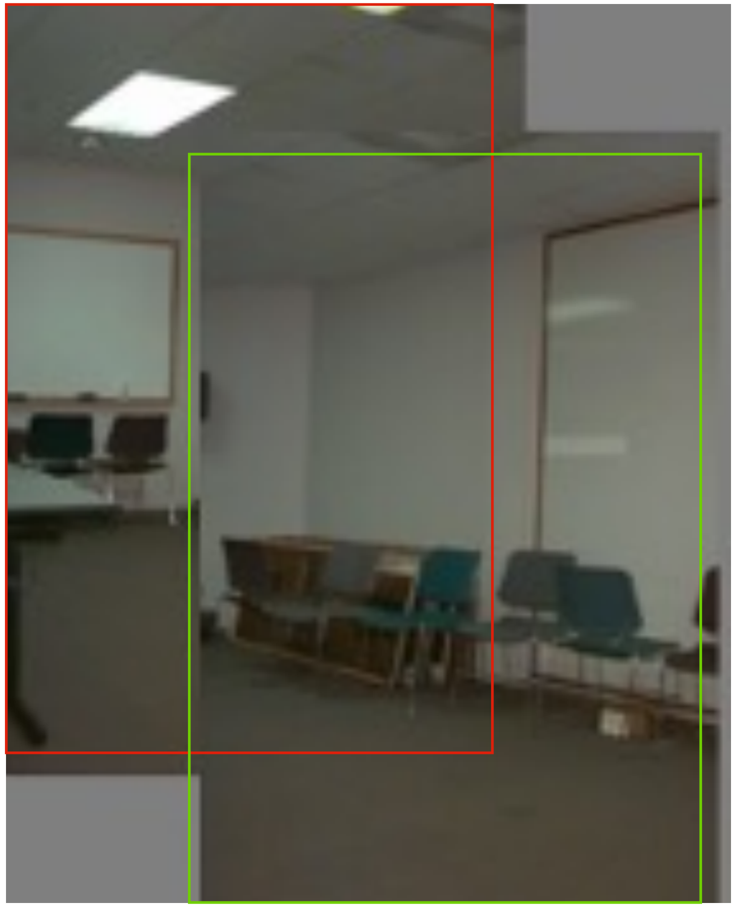
Which kind of transformation is needed to warp projective plane 1 into projective plane 2?

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

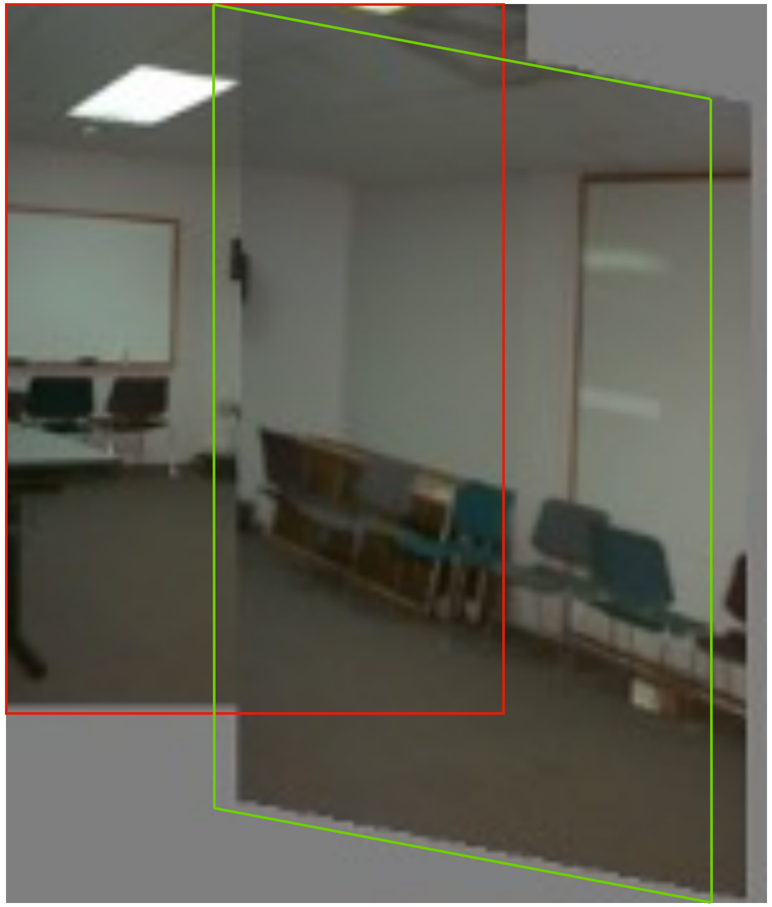


Warping with different transformations

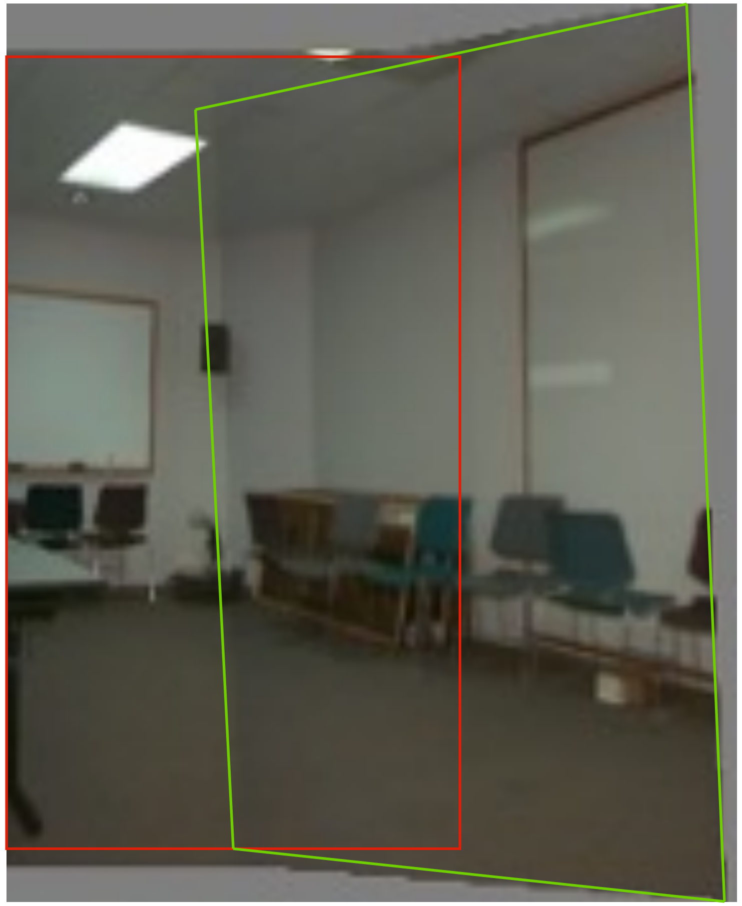
translation



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

Carpet illusion

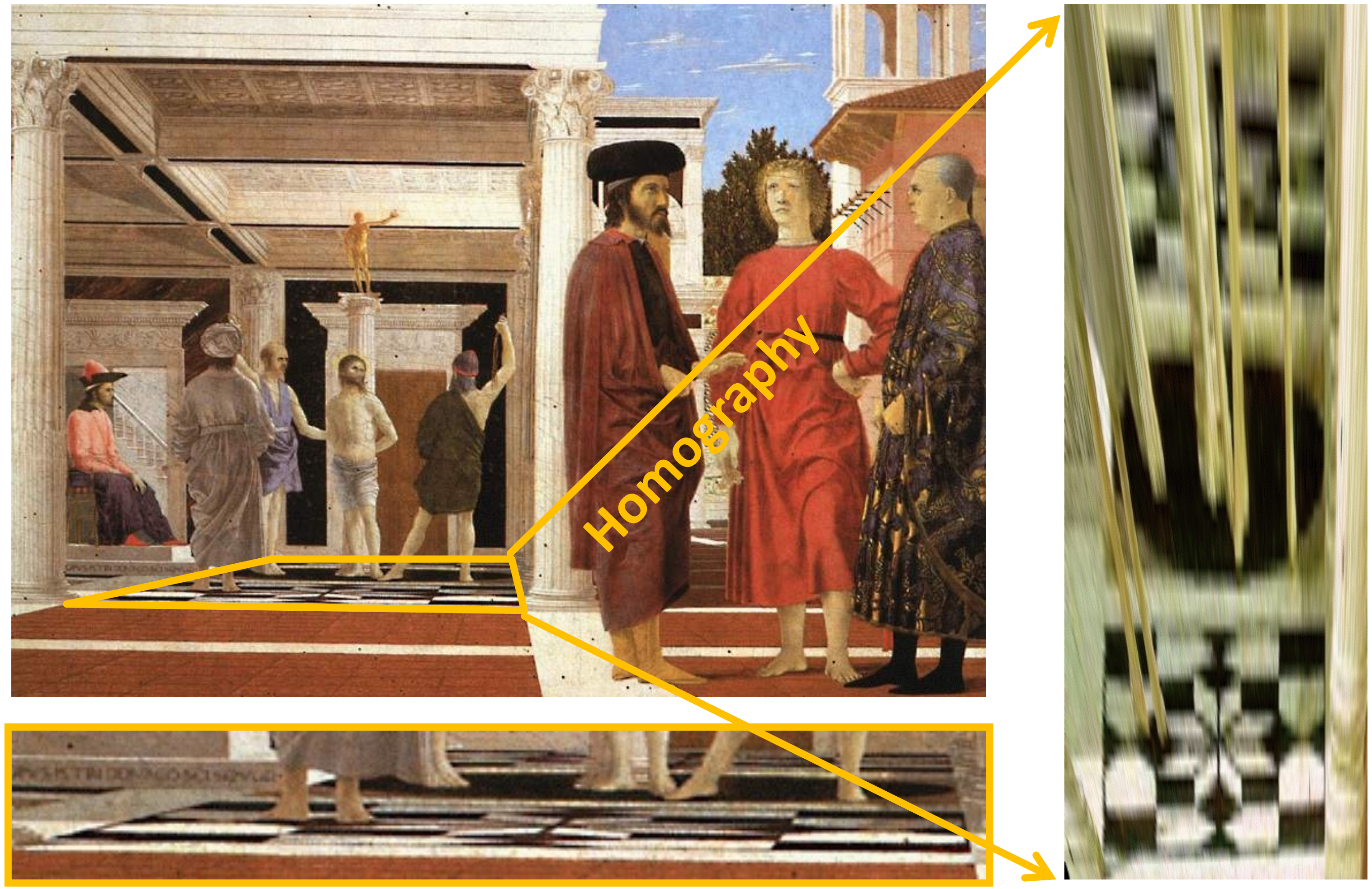
Understanding geometric patterns

What is the pattern on the floor?

magnified view of floor

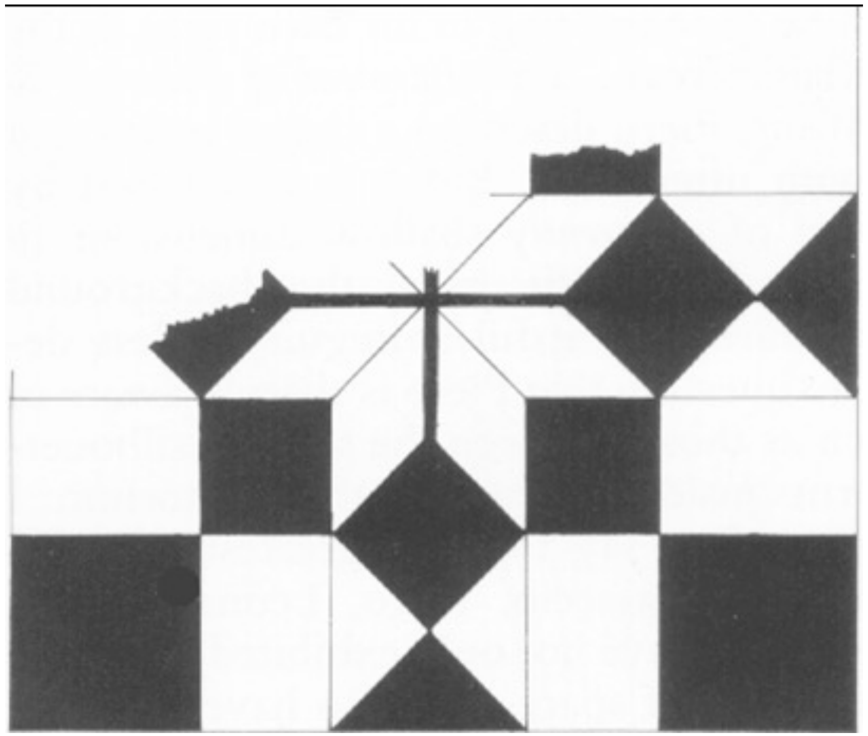
Understanding geometric patterns

What is the pattern on the floor?



magnified view of floor

rectified view

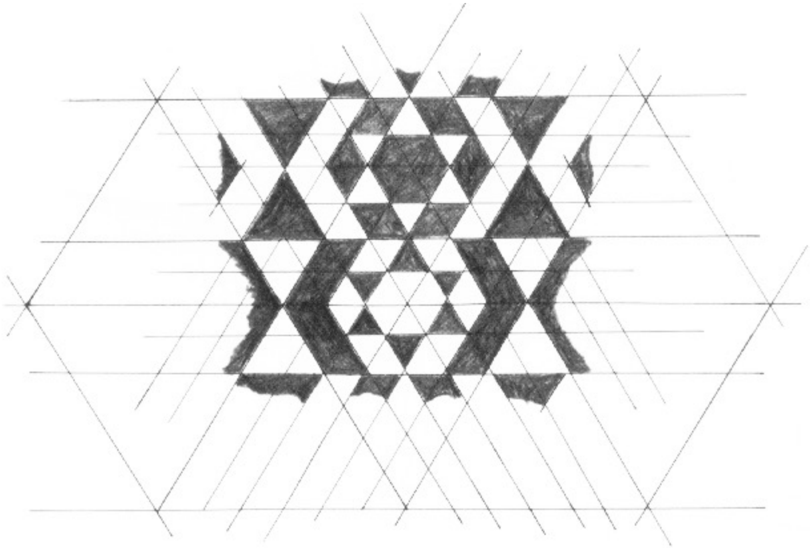


reconstruction from rectified view

Understanding geometric patterns

Very popular in renaissance drawings (when perspective was discovered)

rectified view of floor



reconstruction

A weird painting

Holbein, “The Ambassadors”

A weird painting

Holbein, “The Ambassadors”

What’s this???

A weird painting

Holbein, “The Ambassadors”

rectified view

skull under anamorphic perspective

A weird painting

Holbein, “The Ambassadors”

DIY: use a polished spoon to see the skull

Panoramas from image stitching

1. Capture multiple images from different viewpoints.

2. Stitch them together into a virtual wide-angle image.

When can we use homographies?

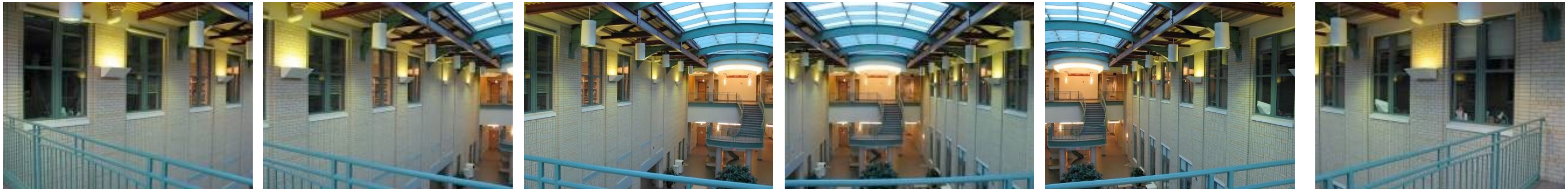
We can use homographies when…

1. … the scene is planar; or

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

We can use homographies when…

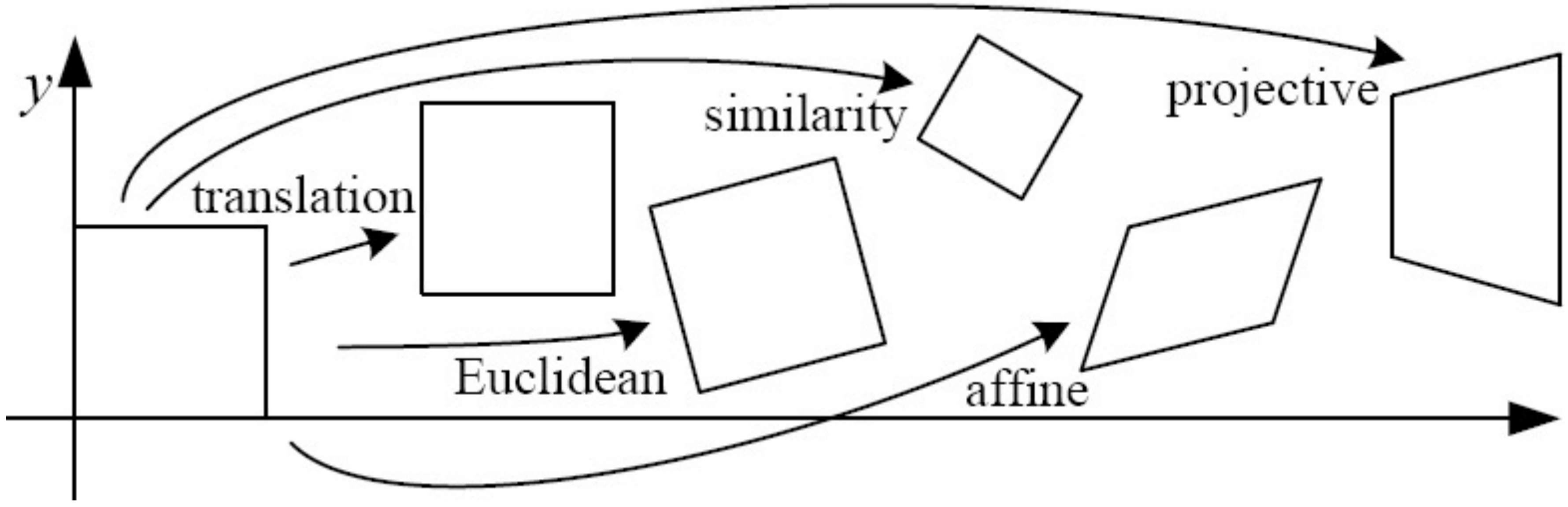
3. … the scene is captured under camera rotation only (no translation or pose change)

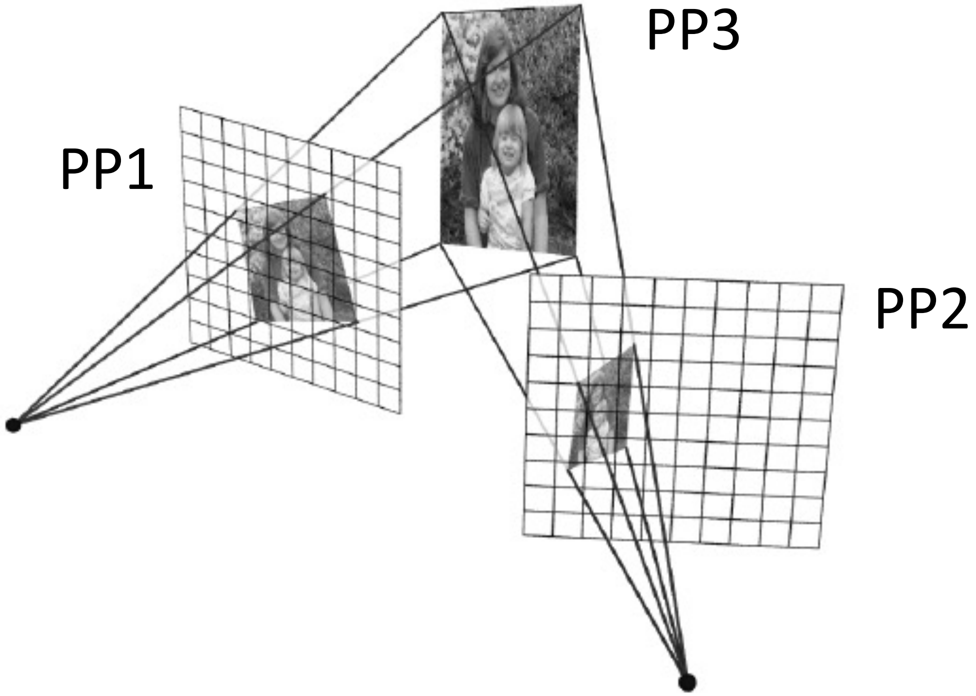


More on why this is the case in a later lecture.

Computing with homographies

Classification of 2D transformations





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

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

Applying a homography

1. **Convert to homogeneous coordinates:** $p = \begin{bmatrix} x \\ y \end{bmatrix} \;\Rightarrow\; P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

What is the size of the homography matrix? ↘

2. **Multiply by the homography matrix:** $P' = H \cdot P$

3. **Convert back to heterogeneous coordinates:** $P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \;\Rightarrow\; p' = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$

Applying a homography

1. Convert to homogeneous coordinates:

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

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

2. Multiply by the homography matrix:

$$P' = H \cdot P$$

How many degrees of freedom does the homography matrix have?

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:**

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

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

2. **Multiply by the homography matrix:**

$$P' = H \cdot P$$

How many degrees of freedom does the homography matrix have? ↗ Answer: 8

3. **Convert back to heterogeneous coordinates:**

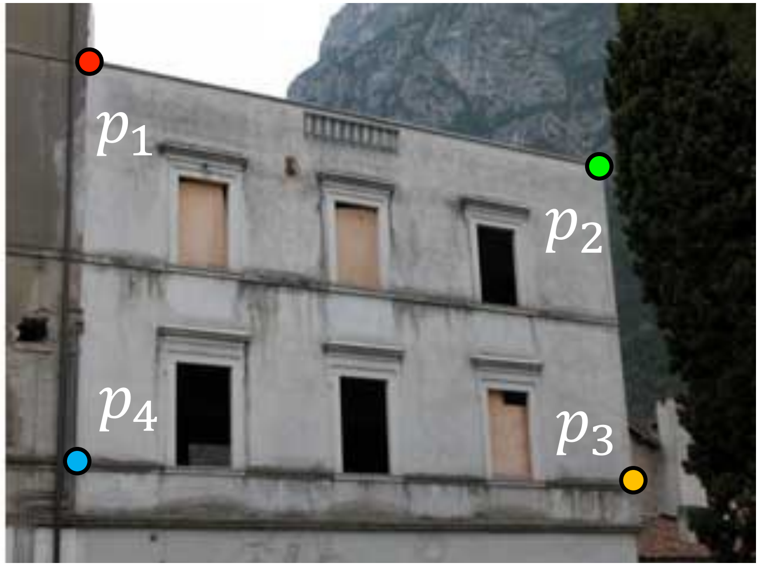
$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \quad \Rightarrow \quad p' = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$$

The direct linear transform (DLT)

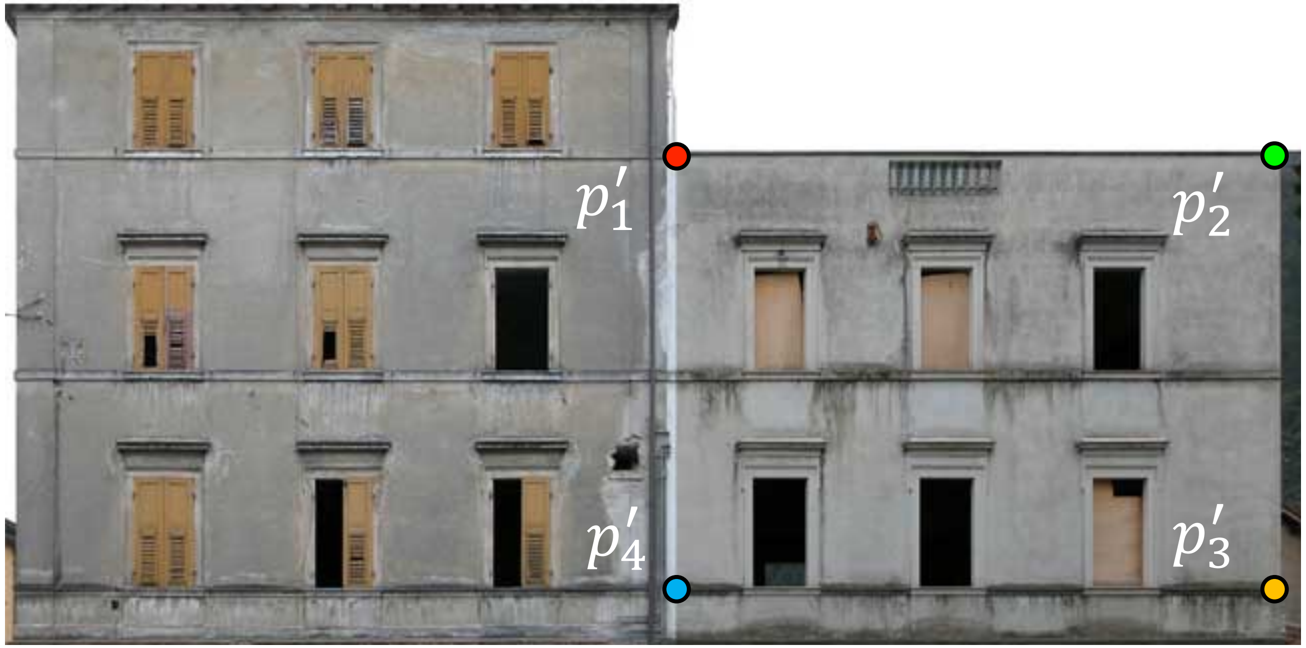
Create point correspondences

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:

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

Divide out unknown scale factor:

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

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:

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

Re-write in matrix form:

How many equations from one point correspondence?

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

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

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

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:

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

Notation in system form:

$$\boxed{Ax = b}$$

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:

Set derivative to 0 $(\mathbf{A}^\top\mathbf{A})\boldsymbol{x} = \mathbf{A}^\top\boldsymbol{b}$

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

In Python:

```
x = numpy.linalg.
    solve(A, b)
```

Note: You almost never 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

- Solve with SVD

Singular value decomposition

orthonormal diagonal orthonormal

$$\underset{n \times m}{\mathbf{A}} = \underset{n \times n}{\mathbf{U}}\,\underset{n \times m}{\mathbf{\Sigma}}\,\underset{m \times m}{\mathbf{V}}^{\mathrm{T}}$$

$$= \sum_{i=1}^{9} \sigma_i \underset{n \times 1}{u_i} \underset{1 \times m}{v_i^{\mathrm{T}}}$$

General form of total least squares

(**Warning:** change of notation. x is a vector of parameters!)

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

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

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

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

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

→

(Rayleigh quotient)

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

Solution is the eigenvector corresponding to smallest eigenvalue of

$$\mathbf{A}^\top \mathbf{A}$$

(equivalent)

Solution is the column of **V** corresponding to smallest singular value

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

Solving for H using DLT

Given $\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$ solve for H such that $\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$

1. For each correspondence, create 2x9 matrix $\mathbf{A}_i$

2. Concatenate into single 2n x 9 matrix $\mathbf{A}$

3. Compute SVD of $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$

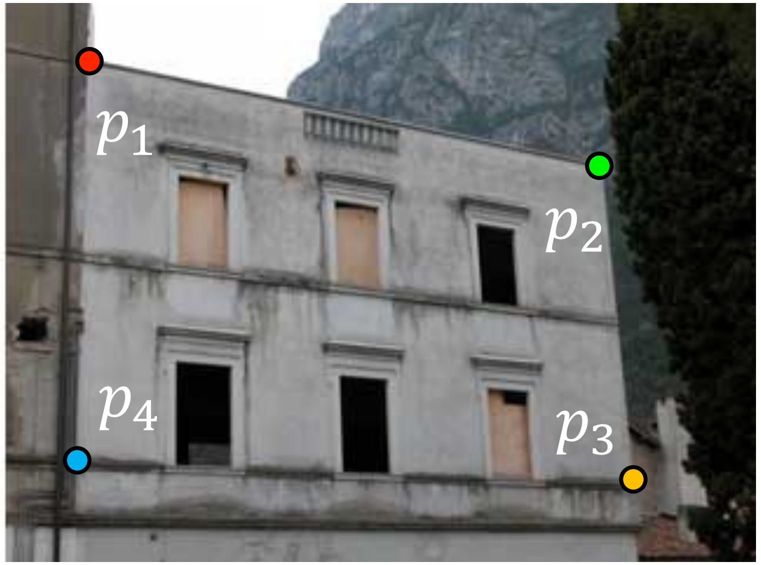
4. Store singular vector of the smallest singular value $\boldsymbol{h} = \boldsymbol{v}_{\hat{i}}$

5. Reshape to get $\mathbf{H}$

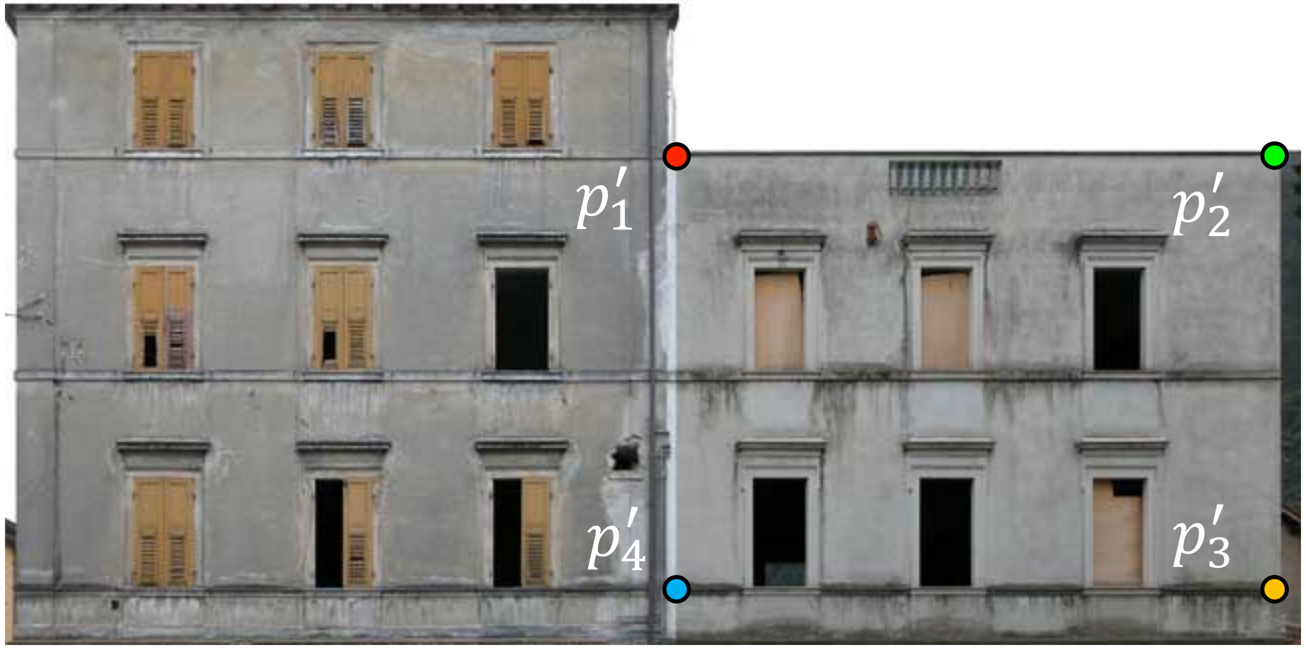
Linear least squares estimation only works when the transform function is **linear! (duh)**

Also doesn’t deal well with **outliers.**

Create point correspondences



original image



target image

How do we automate this step?

The image correspondence pipeline

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

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

3. Feature matching

The image correspondence pipeline

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

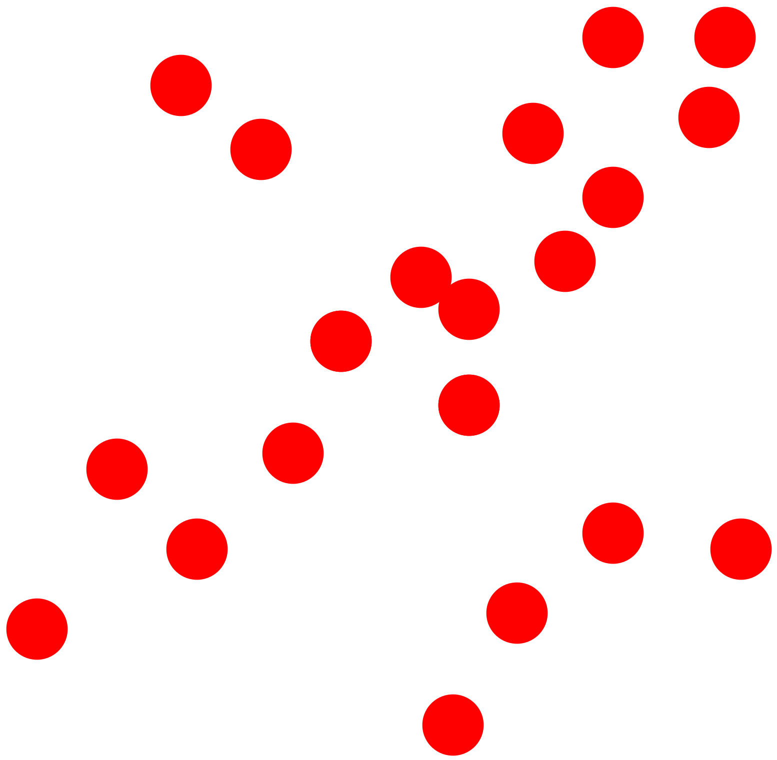
2. Feature point description
   - Describe features using the Multi-scale oriented patch descriptor.

3. Feature matching


bad correspondence
good correspondence

Random Sample Consensus (RANSAC)

Fitting lines
(with outliers)

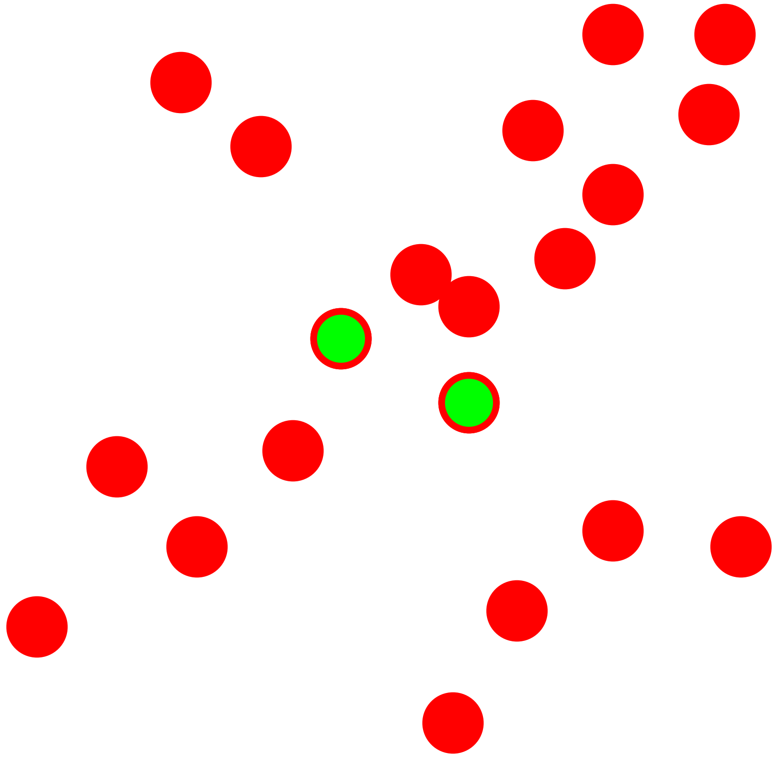


Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. 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

Fitting lines
(with outliers)

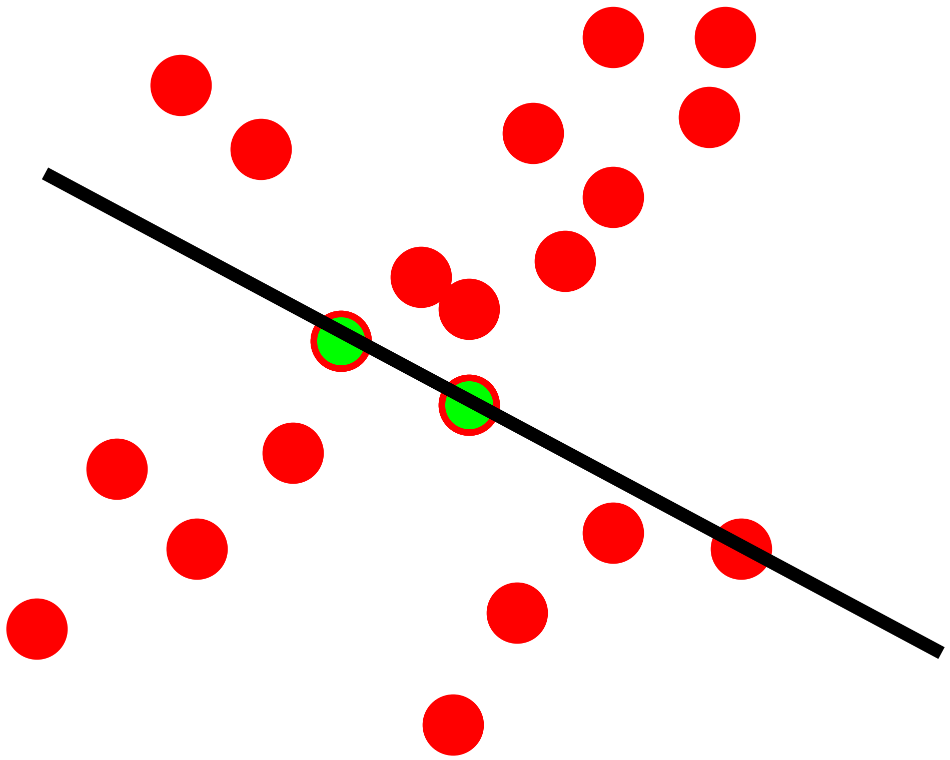


Algorithm:

1. **Sample (randomly) the number of points required to fit the model**
2. Solve for model parameters using samples
3. 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

Fitting lines
(with outliers)



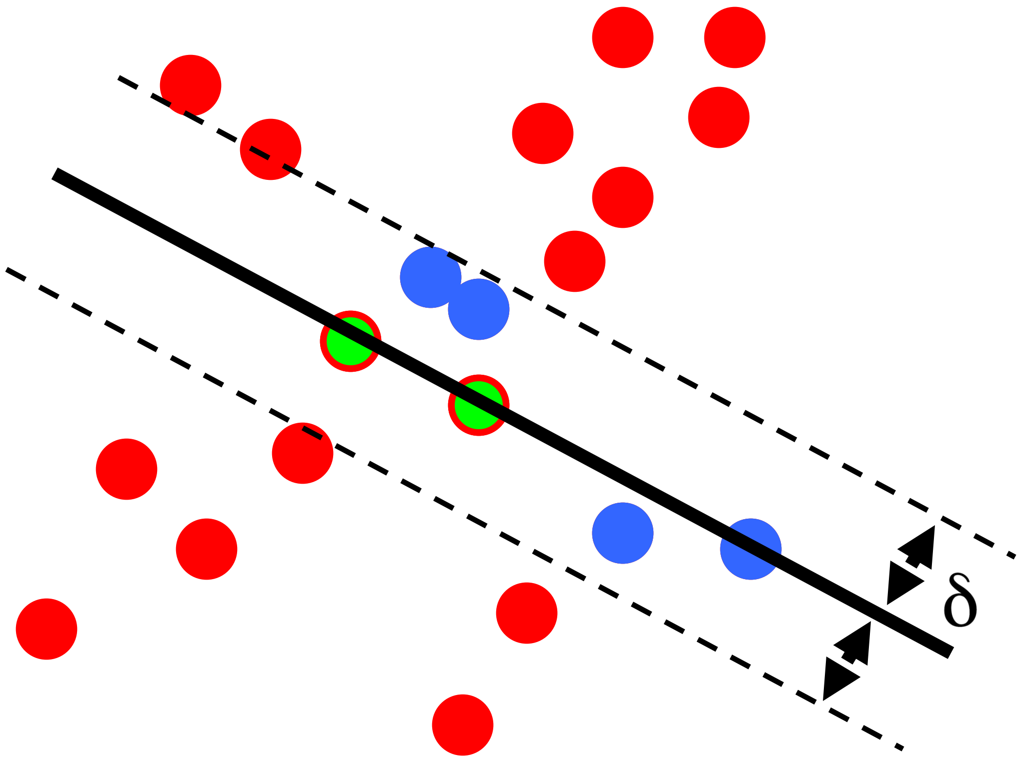
Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. **Solve for model parameters using samples**
3. 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

Fitting lines
(with outliers)

$N_I = 6$

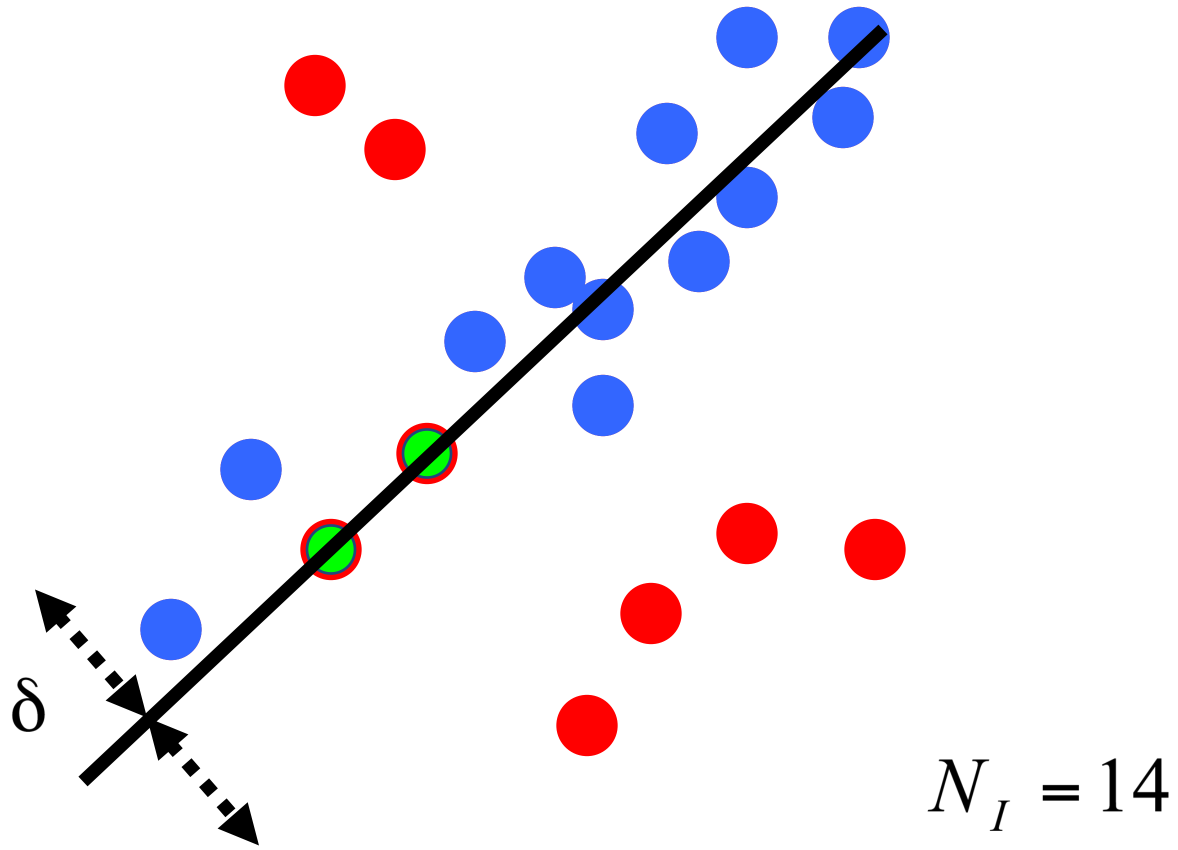


Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. **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

Fitting lines
(with outliers)



Algorithm:

1. Sample (randomly) the number of points required to fit the model
2. Solve for model parameters using samples
3. 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

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

Number of samples N required

	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

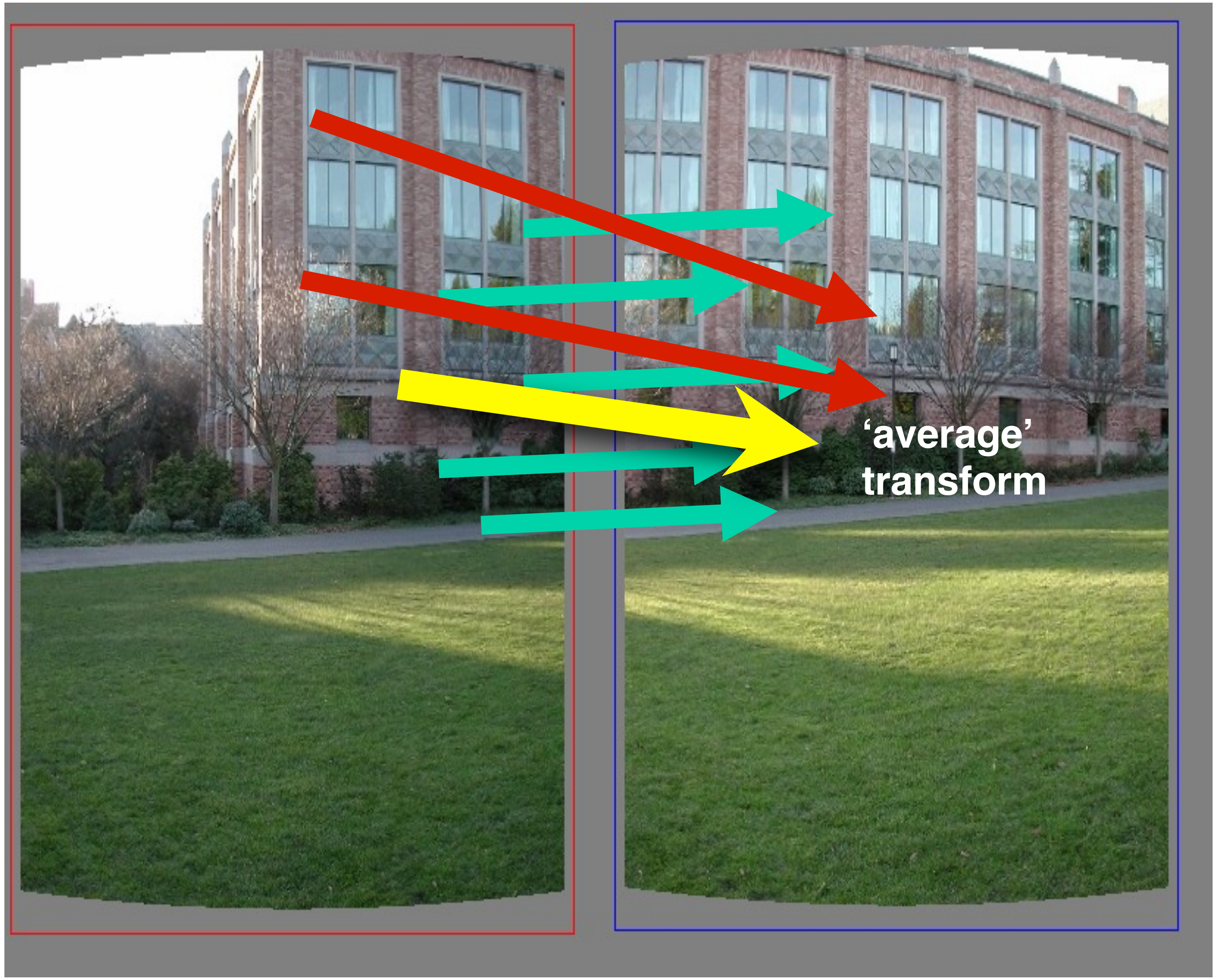
Given two images…

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

Matched points will usually contain bad correspondences

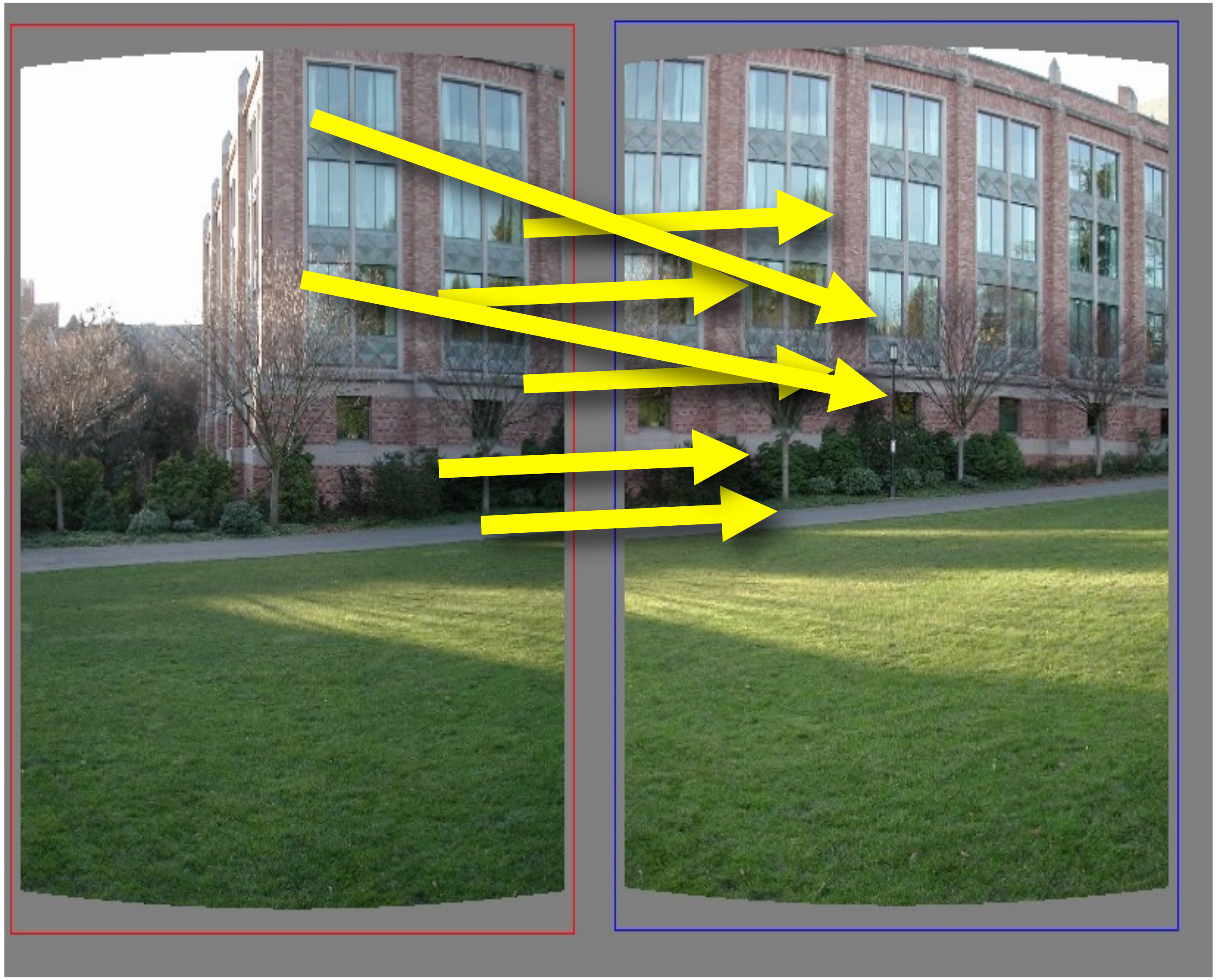
how should we estimate the transform?

LLS will find the "average" transform

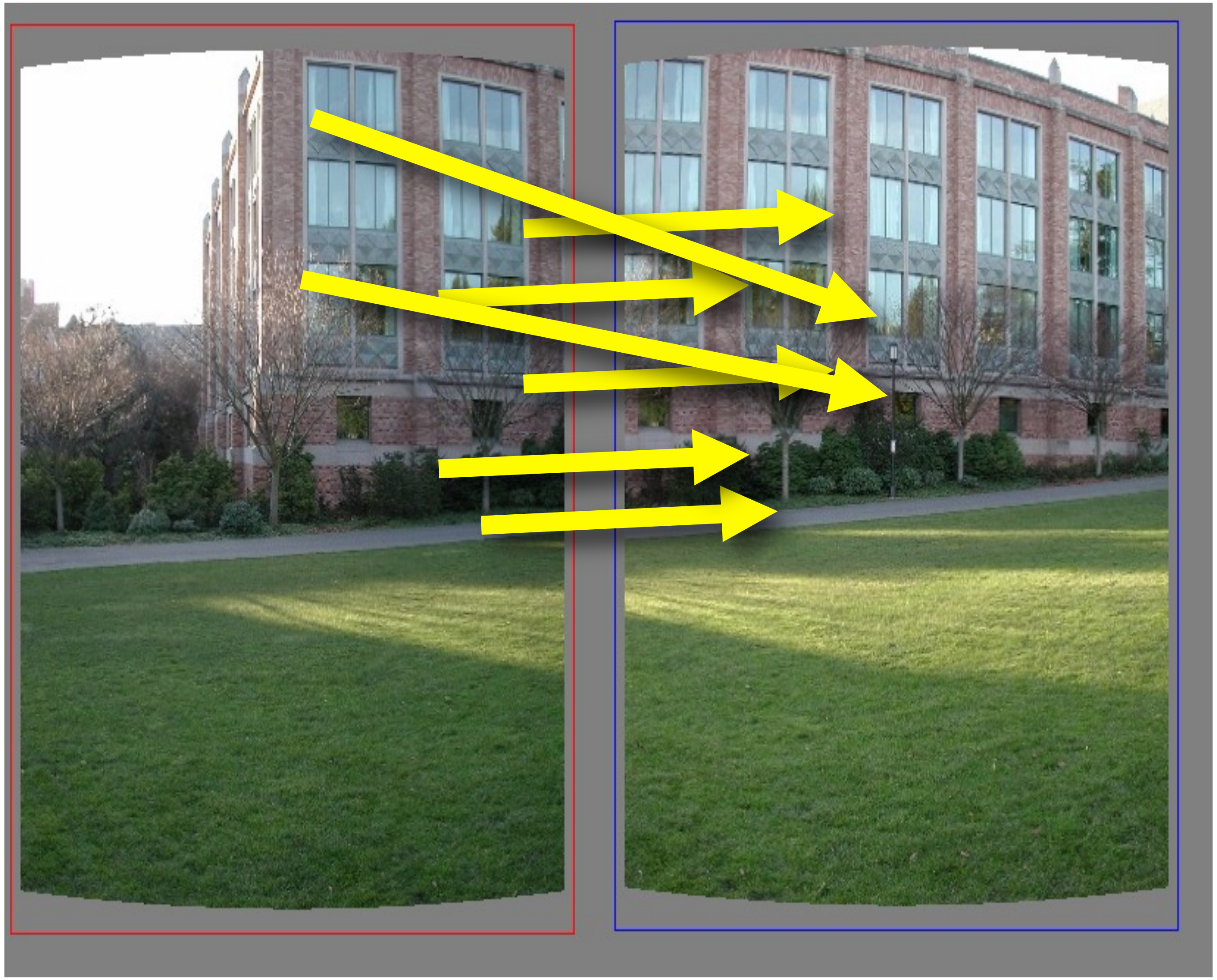


solution is corrupted by bad correspondences

Use RANSAC

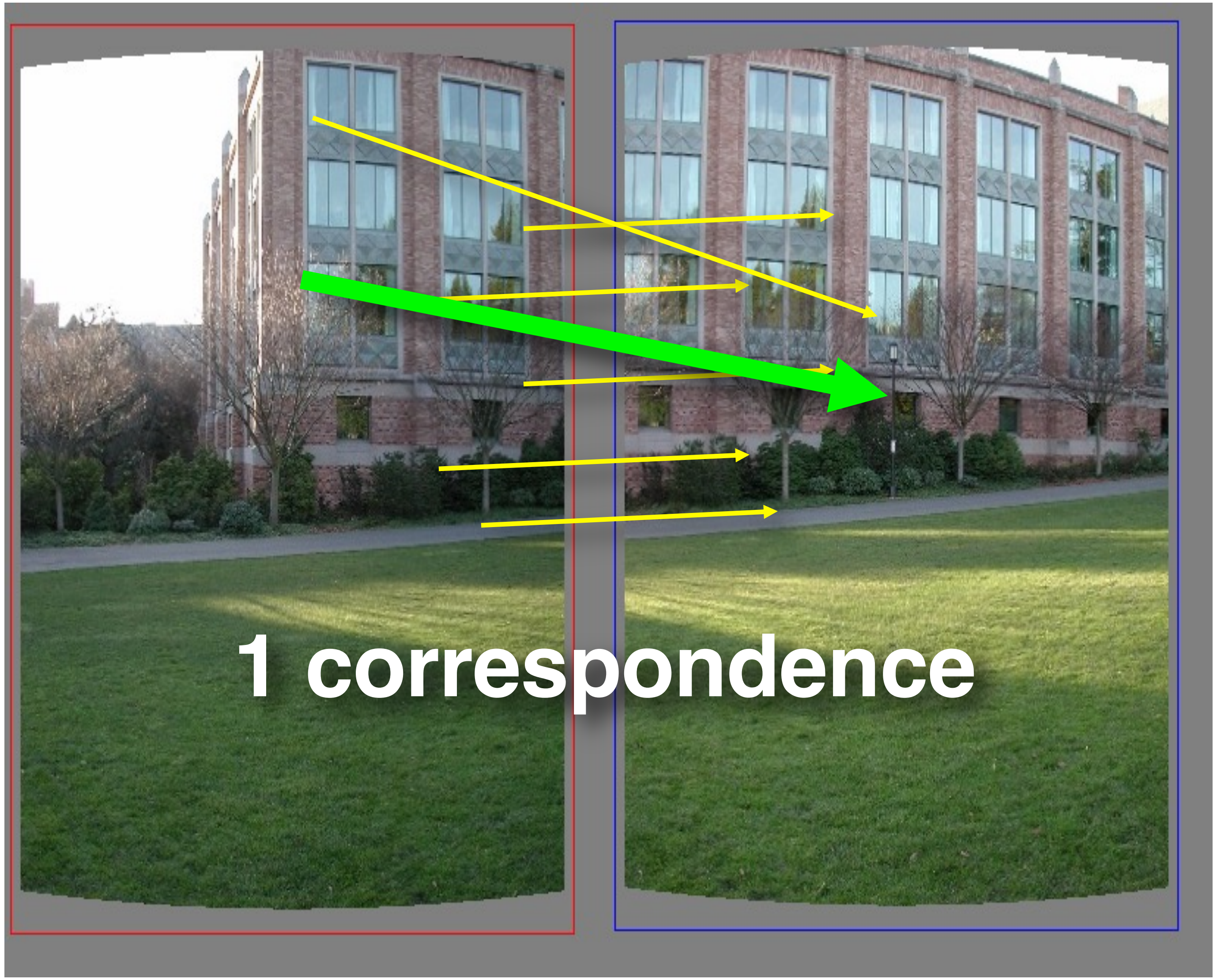


How many correspondences to compute translation transform?

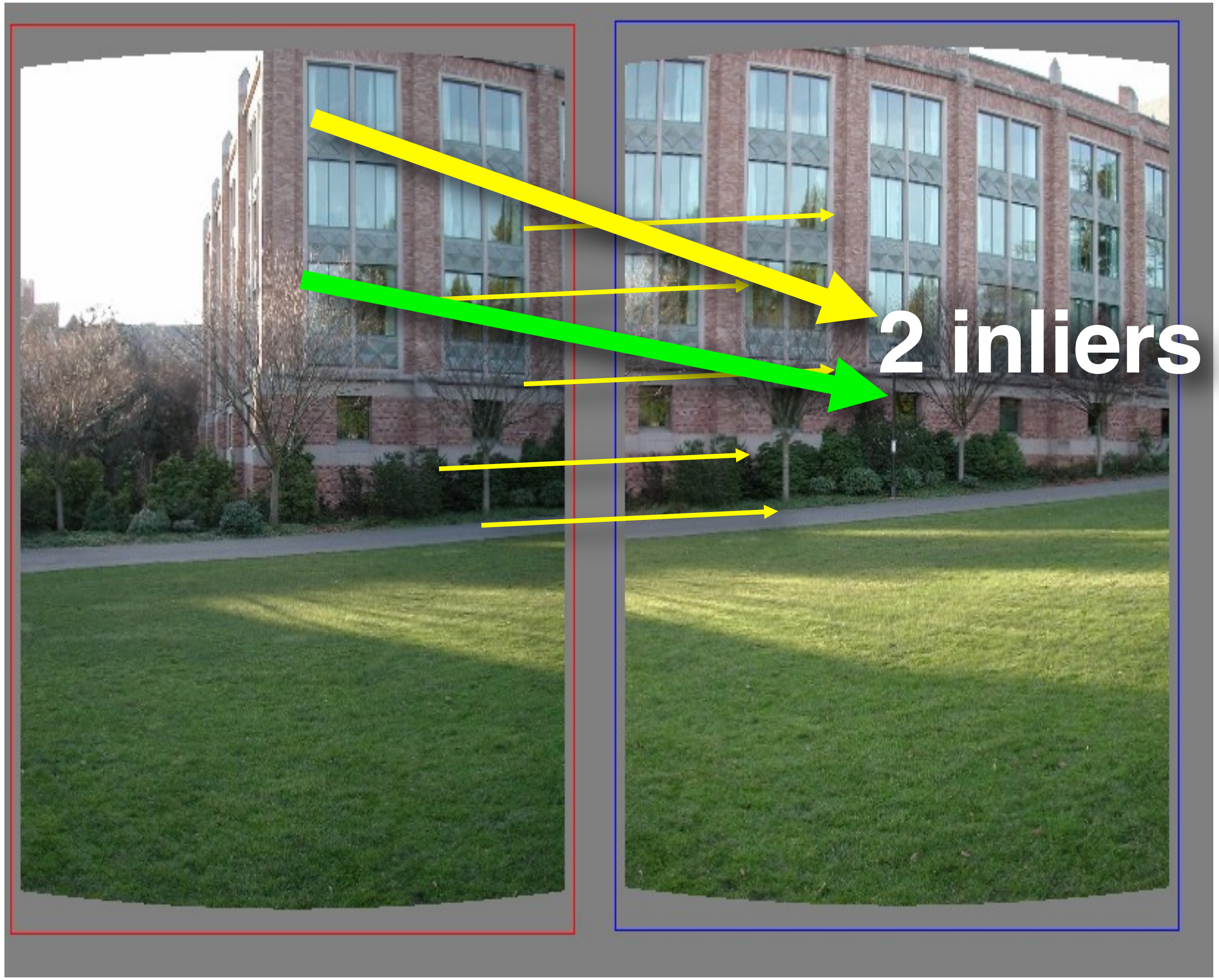


Need only **one correspondence**, to find translation model

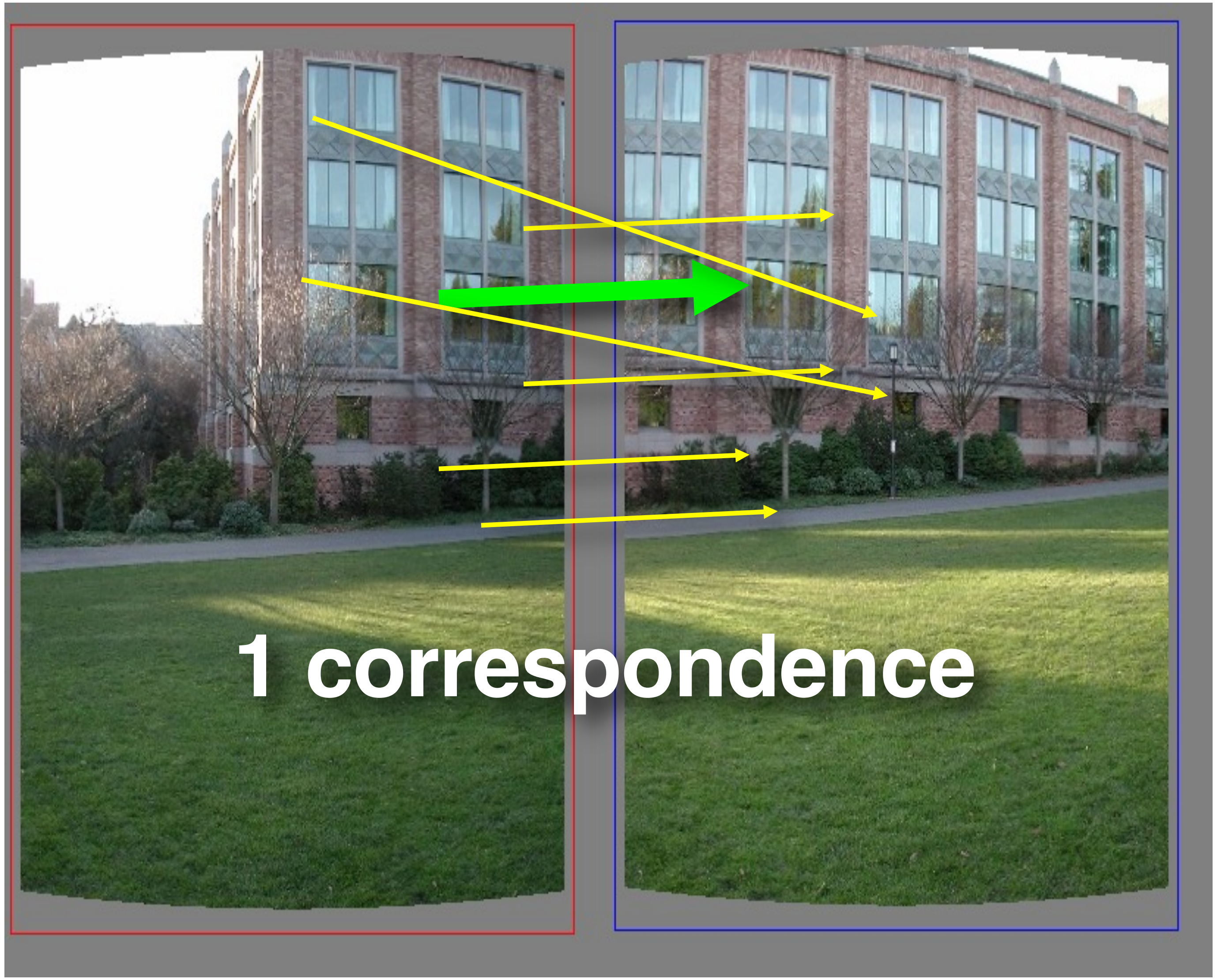
Pick one correspondence, count inliers



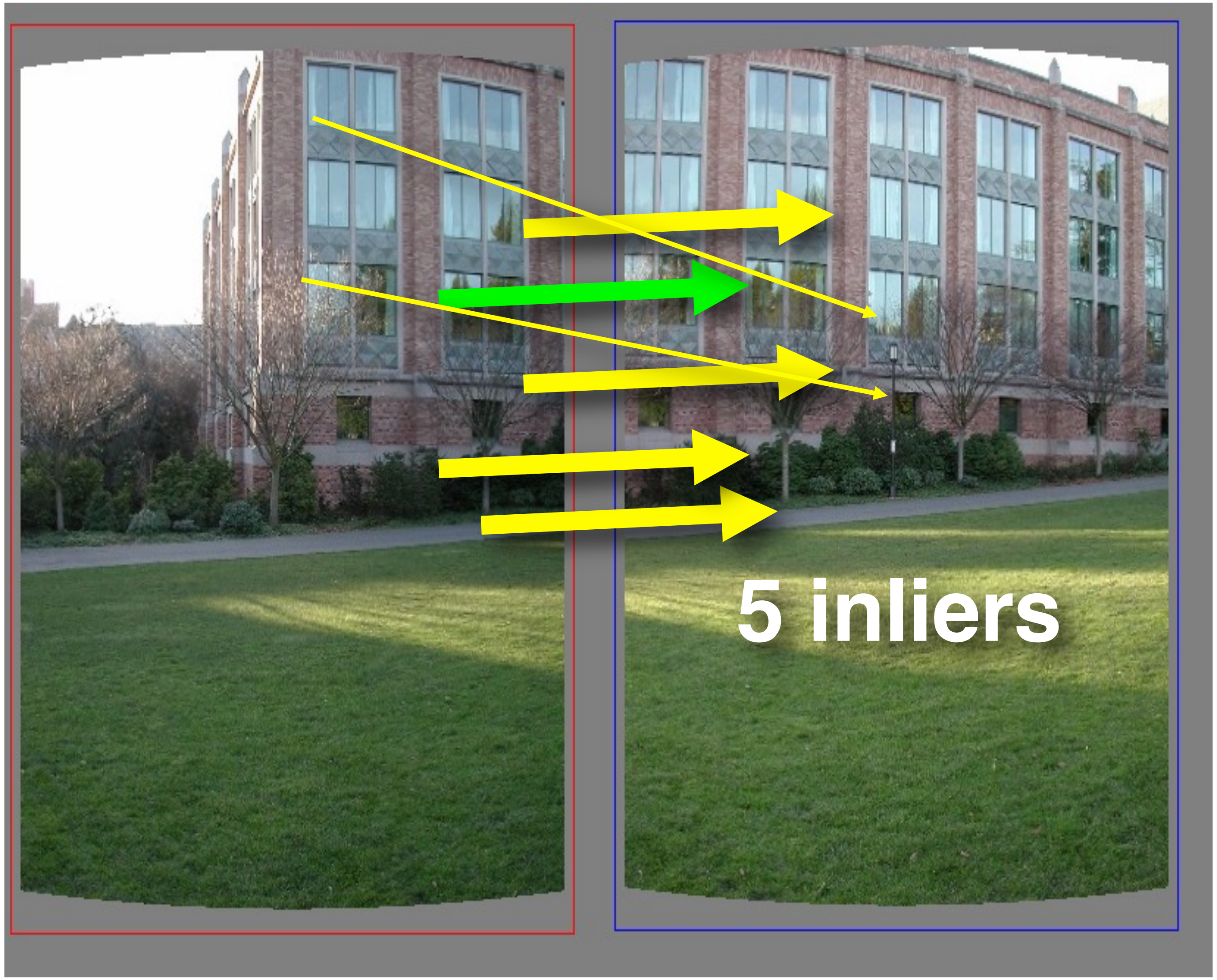
Pick one correspondence, count inliers



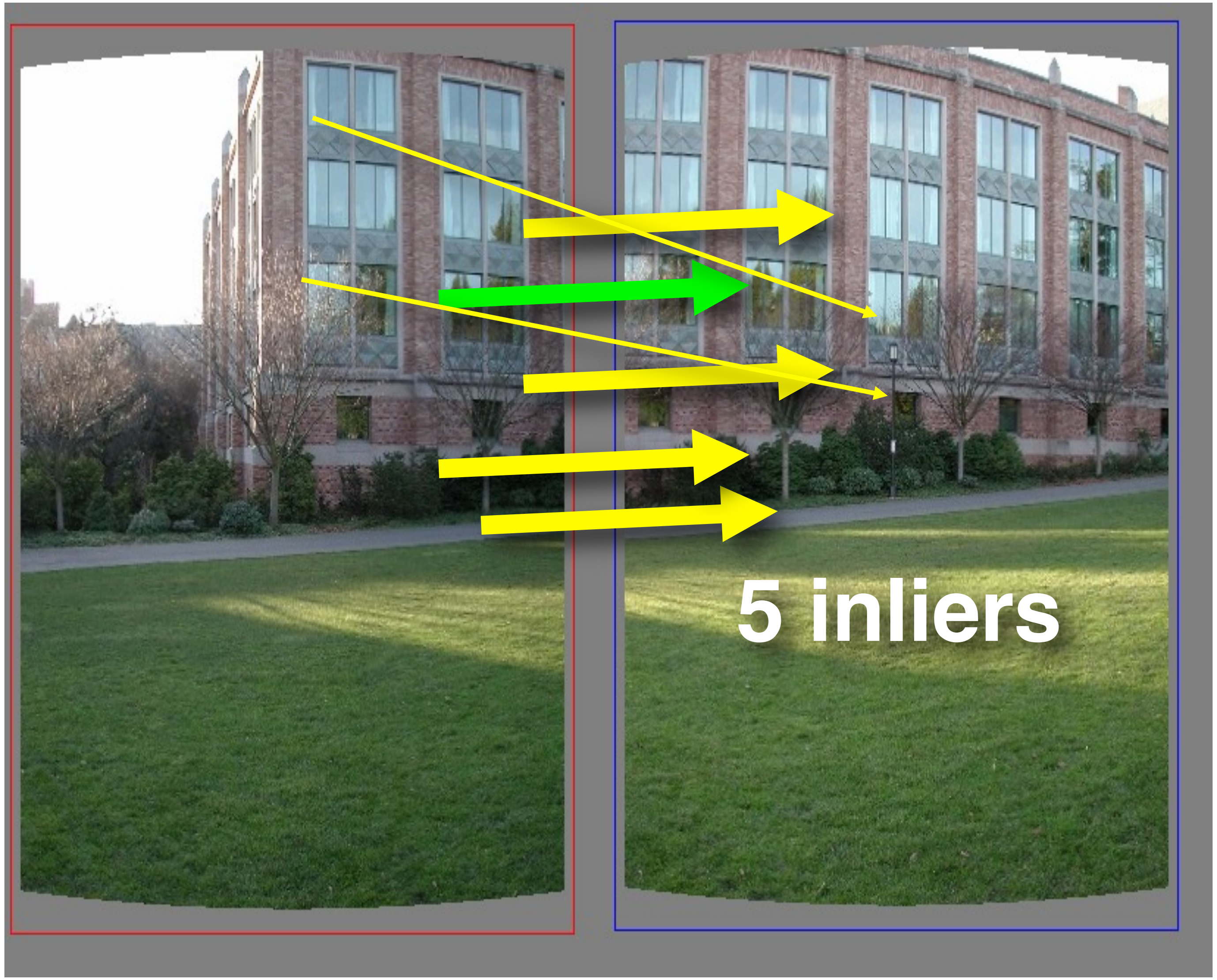
Pick one correspondence, count inliers



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

    1. Get point correspondences (randomly)

Estimating homography using RANSAC

- RANSAC loop
  1. Get four point correspondences (randomly)
  2. Compute H using

Estimating homography using RANSAC

- RANSAC loop

    1. Get four point correspondences (randomly)
    2. Compute H using DLT
    3. Count

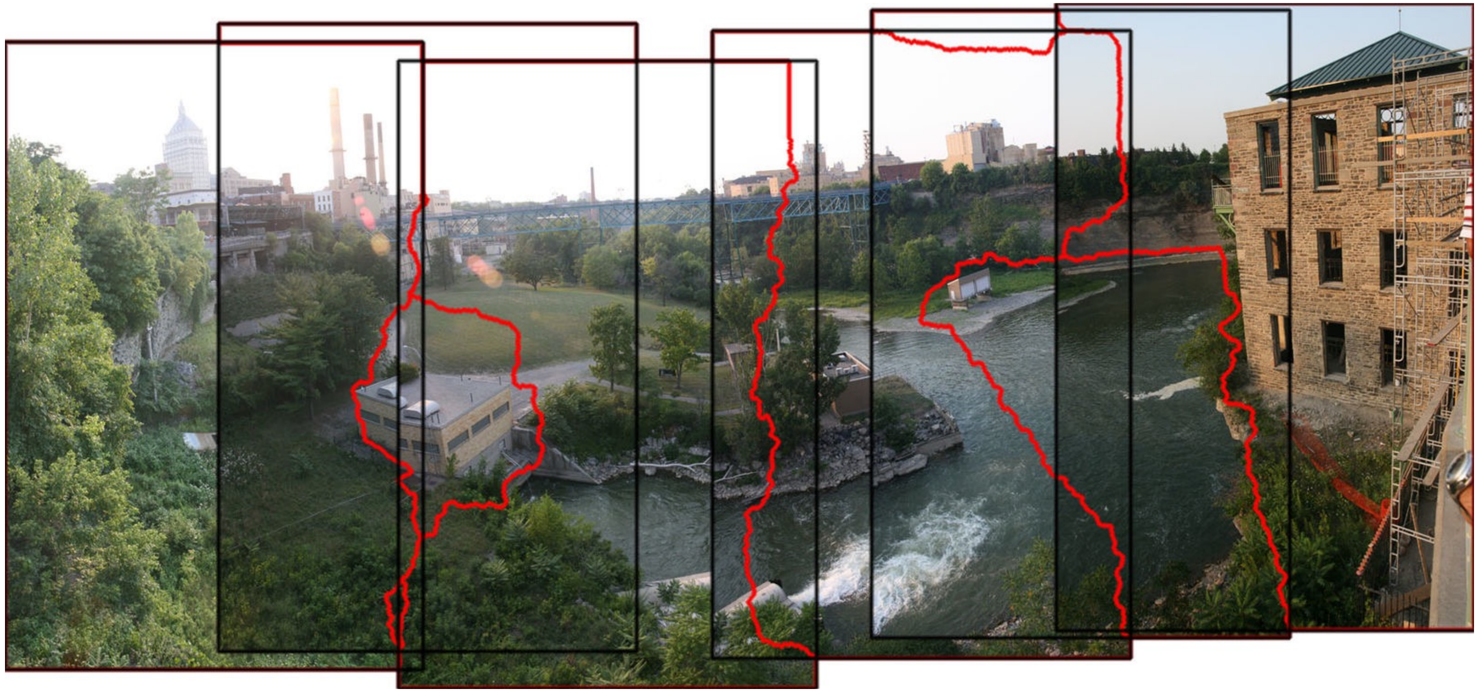
Estimating homography using RANSAC

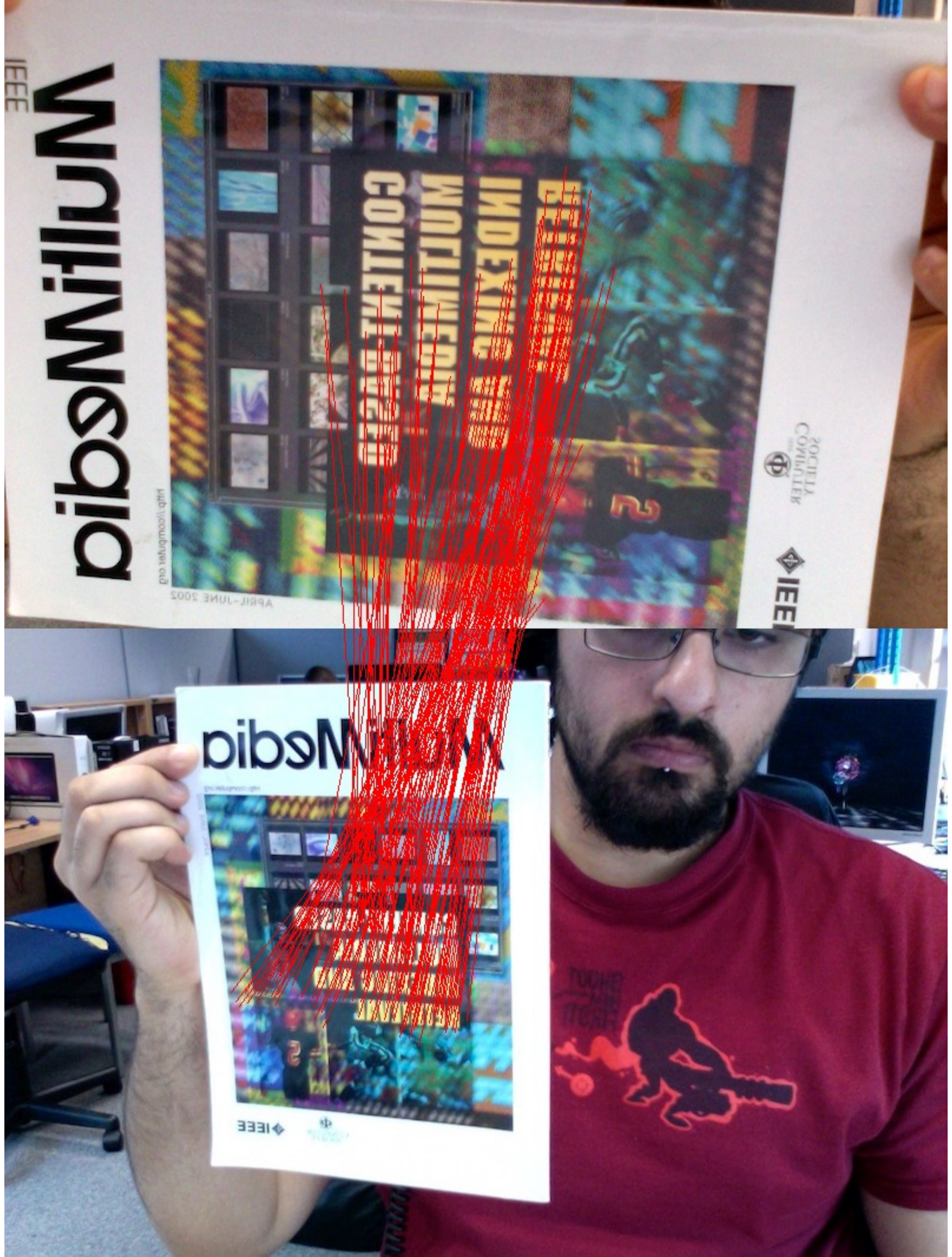
- RANSAC loop
    1. Get four point correspondences (randomly)
    2. Compute H using DLT
    3. Count inliers
    4. Keep H if

Estimating homography using RANSAC

- RANSAC loop
  1. Get four point correspondences (randomly)
  2. Compute H using DLT
  3. Count inliers
  4. Keep H if largest number of inliers
- Recompute H using all inliers

Useful for...





The image correspondence pipeline

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

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

3. Feature matching *and* homography estimation
    - Do both simultaneously using RANSAC.