Calibration and homographies

Final perspective projection

Camera extrinsics: where your camera is relative to the world. Changes if you move the camera

$$\vec{\mathbf{x}}_{img} \equiv K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \vec{\mathbf{x}}_w$$

Camera intrinsics: how your camera handles pixel. Changes if you change your camera

$$\vec{\mathbf{x}}_{img} \equiv P \vec{\mathbf{x}}_w$$

Final perspective projection

$$\vec{\mathbf{x}}_{img} \equiv \underbrace{K \begin{bmatrix} R & \mathbf{t} \end{bmatrix}}_{\text{Camera parameters}} \vec{\mathbf{x}}_w$$

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

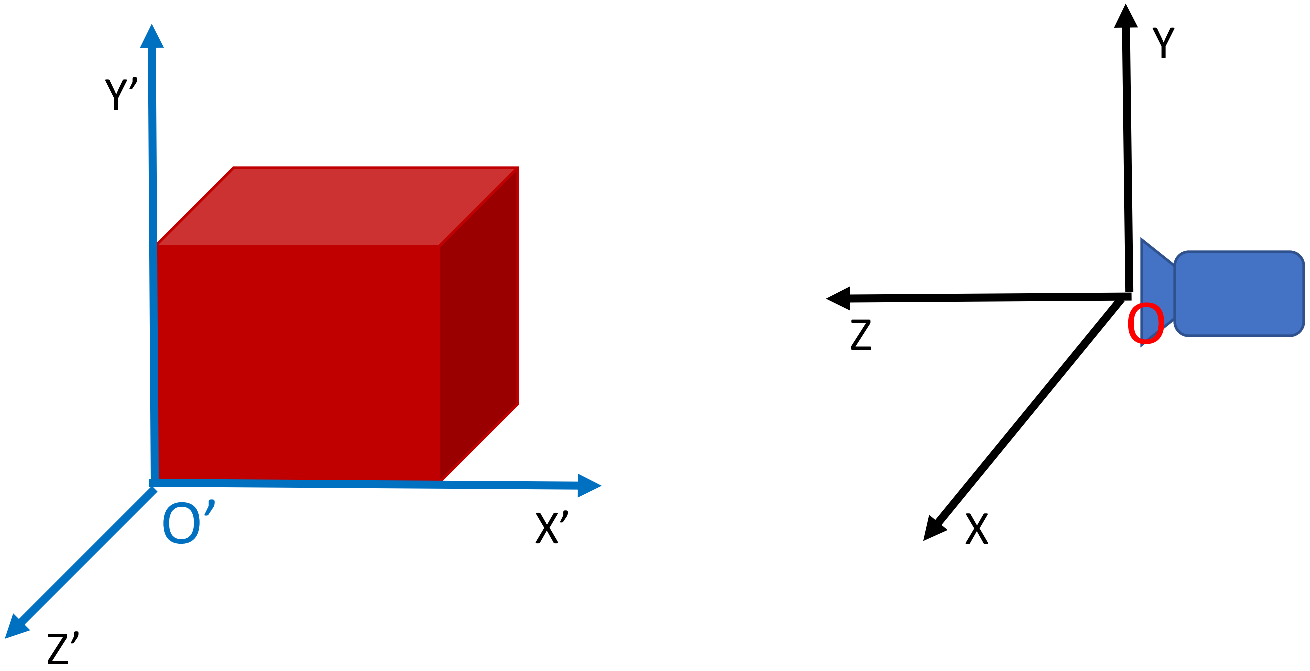
Camera calibration

- Goal: find the parameters of the camera

$$\vec{\mathbf{x}}_{img} \equiv K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \vec{\mathbf{x}}_w$$

- Why?
  - Tells you where the camera is relative to the world/particular objects
  - Equivalently, tells you where objects are relative to the camera
  - Can allow you to ”render” new objects into the scene

Camera calibration



Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Need to estimate P
- How many parameters does P have?
  - Size of P : 3 x 4
  - But: $\lambda P\vec{\mathbf{x}}_w \equiv P\vec{\mathbf{x}}_w$
  - P can only be known *upto a scale*
  - *3*4 - 1 = 11 parameters*

Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Need to convert equivalence into equality.

Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

Note: λ is unknown

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

$$\lambda x = P_{11}X + P_{12}Y + P_{13}Z + P_{14}$$

$$\lambda y = P_{21}X + P_{22}Y + P_{23}Z + P_{24}$$

$$\lambda = P_{31}X + P_{32}Y + P_{33}Z + P_{34}$$

Camera calibration

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14}$$
$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = P_{21}X + P_{22}Y + P_{23}Z + P_{24}$$

- 2 equations!
- Are the equations linear in the parameters?
- How many equations do we need?

Camera calibration

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14}$$

$$XxP_{31} + YxP_{32} + ZxP_{33} + xP_{34} - XP_{11} - YP_{12} - ZP_{13} - P_{14} = 0$$

- In matrix vector form: Ap = 0
- 6 points give 12 equations, 12 variables to solve for
- But can only solve upto scale

Camera calibration

- In matrix vector form: Ap = 0
- We want non-trivial solutions
- If p is a solution, αp is a solution too
- Let's just search for a solution with unit norm

$$A\mathbf{p} = 0$$

s.t

$$\|\mathbf{p}\| = 1$$

Camera calibration

Direct Linear Transformation

- In matrix vector form: Ap = 0
- But there may be noise in the inputs
- Least squares solution:

$$\min_{\mathbf{p}} \|A\mathbf{p}\|^2 \qquad \equiv \min_{\mathbf{p}} \mathbf{p}^T A^T A\mathbf{p}$$

s.t $\qquad\qquad$ s.t

$$\|\mathbf{p}\| = 1 \qquad\qquad \|\mathbf{p}\| = 1$$

- Eigenvector of ATA with smallest eigenvalue! (also right singular vector pf A with smallest singular value)

Camera calibration through non-linear minimization

- Problem: $||A\mathbf{p}||^2$ does not capture meaningful metric of error
  - Depends on units, origin of coordinates etc
- Really, want to measure **reprojection error**
  - If **Q** is projected to **q**, but we think it should be projected to **q**', reprojection error = $||\mathbf{q} - \mathbf{q}'||^2$ (distance in Euclidean coordinates)

Reprojection error

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\Rightarrow x = \frac{P_{11}X + P_{12}Y + P_{13}Z + P_{14}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}}$$

$$y = \frac{P_{21}X + P_{22}Y + P_{23}Z + P_{24}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}}$$

$$\boxed{\begin{aligned} E(P) =& \| x - \frac{P_{11}X + P_{12}Y + P_{13}Z + P_{14}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} \|^2 \\ &+ \| y - \frac{P_{21}X + P_{22}Y + P_{23}Z + P_{24}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} \|^2 \end{aligned}}$$

Reprojection error

Camera calibration through non-linear minimization

- Problem: $||A\mathbf{p}||^2$ does not capture meaningful metric of error
  - Depends on units, origin of coordinates etc
- Really, want to measure **reprojection error**
  - If **Q** is projected to **q**, but we think it should be projected to **q**', reprojection error = $||\mathbf{q} - \mathbf{q}'||^2$ (distance in Euclidean coordinates)

$$\min_P E(P)$$

$$\text{s.t}$$

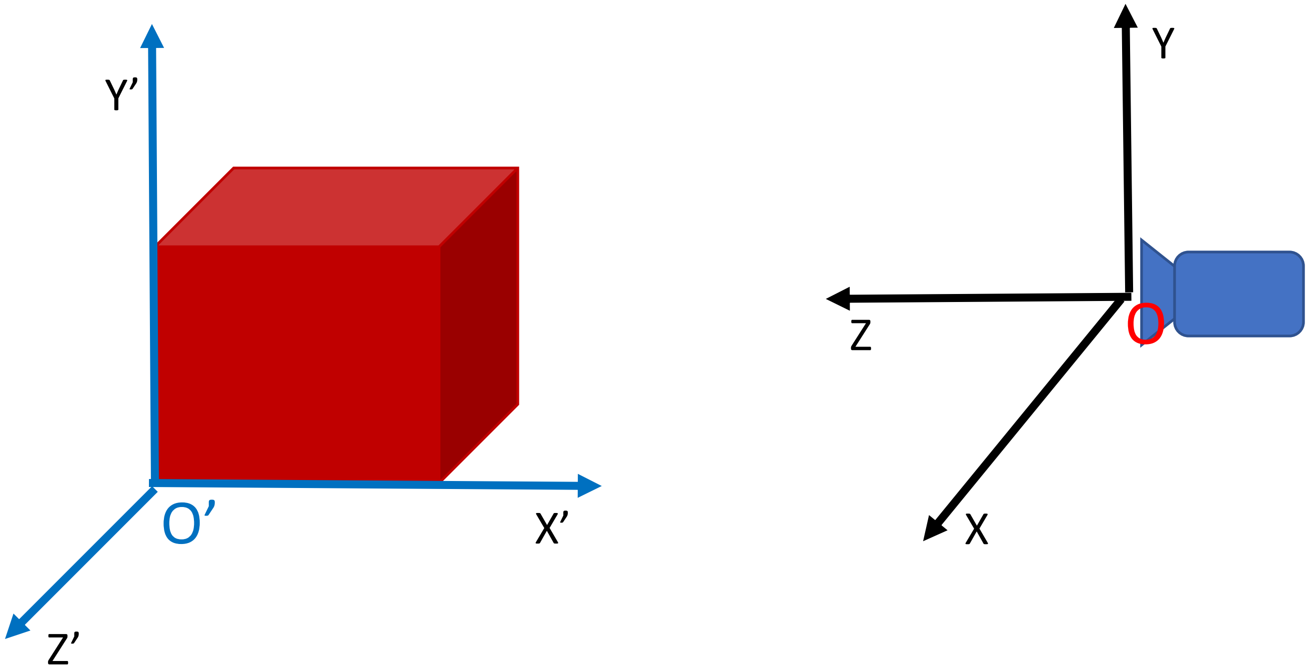
$$\|\mathbf{p}\| = 1$$

  - No closed-form solution, but off-the-shelf iterative optimization

Camera calibration

- We need 6 world points for which we know image locations
- Would any 6 points work?
  - What if all 6 points are the same?
- Need at least 6 non-coplanar points!

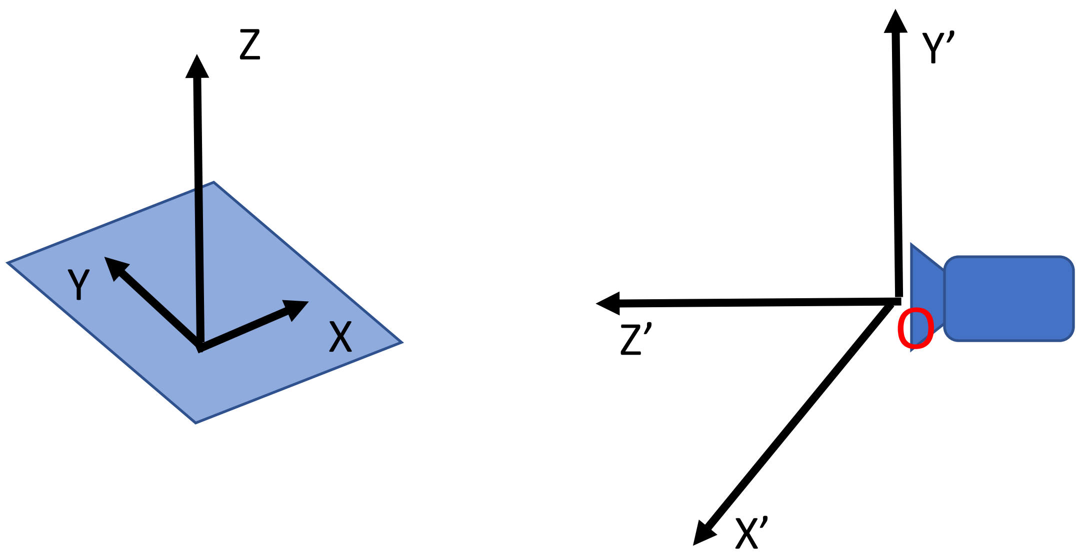
Camera calibration



What if object of interest is plane?

- Not that uncommon….

What if object of interest is plane?



- Let's choose world coordinate system so that plane is X-Y plane

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$\equiv \begin{bmatrix} P_{11} & P_{12} & P_{14} \\ P_{21} & P_{22} & P_{24} \\ P_{31} & P_{32} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

What if object of interest is a plane?

- Imagine that plane is equipped with two axes.
- Points on the plane are represented by *two* euclidean coordinates
- …Or 3 homogenous coordinates

3D object

$$\vec{\mathbf{x}}_{img} \equiv P\vec{\mathbf{x}}_w$$

3 x 4

2D object (plane)

$$\vec{\mathbf{x}}_{img} \equiv H\vec{\mathbf{x}}_w$$

3 x 3

What if object of interest is a plane?

Homography

$$\vec{\mathbf{x}}_{img} \equiv H\vec{\mathbf{x}}_w$$

3 x 3

- Homography maps points on the plane to pixels in the image

Fitting homographies

- How many parameters does a homography have?
- Given a single point on the plane and corresponding image location, what does that tell us?

$$\vec{\mathbf{x}}_{img} \equiv H\vec{\mathbf{x}}_w$$

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$

Fitting homographies

- How many parameters does a homography have?
- Given a single point on the plane and corresponding image location, what does that tell us?

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$

- Convince yourself that this gives 2 linear equations!

Fitting homographies

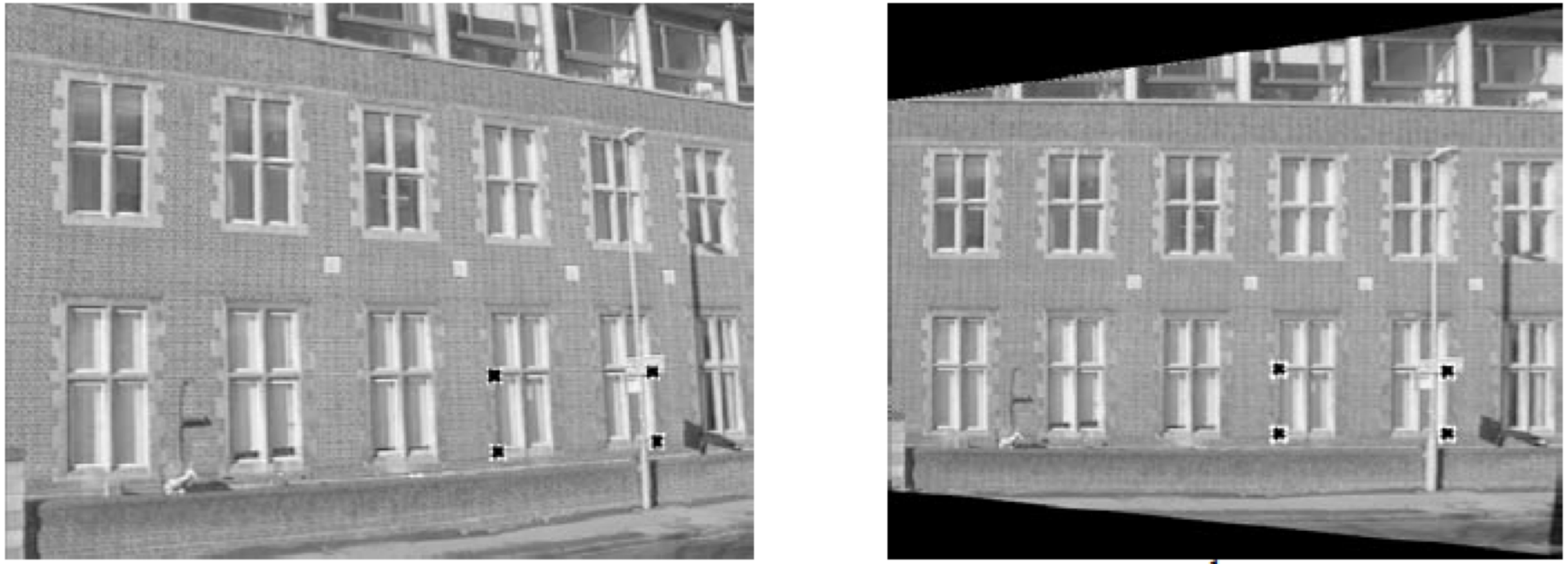
- Homography has 9 parameters
- But can't determine scale factor, so only 8: 4 points!

$$A\mathbf{h} = 0 \text{ s.t } \|\mathbf{h}\| = 1$$

- Or because we will have noise:

$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

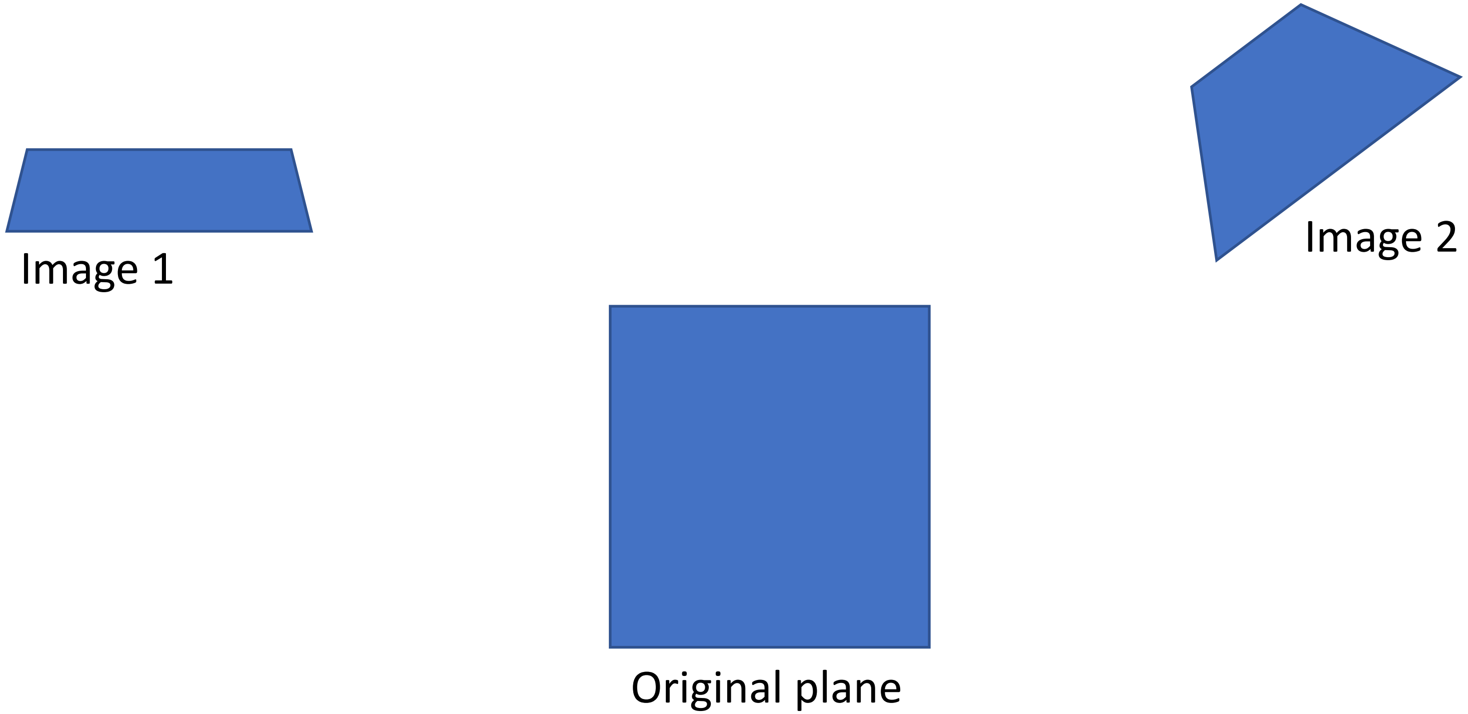
Fitting homographies



a b

Homographies for image alignment

- A general mapping from one plane to another!
- Can also be used to align one photo of a plane to another photo of the same plane



Homographies for image alignment

- Can also be used to align one photo of a plane to another photo of the same plane

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

Image Alignment Algorithm

Given images A and B

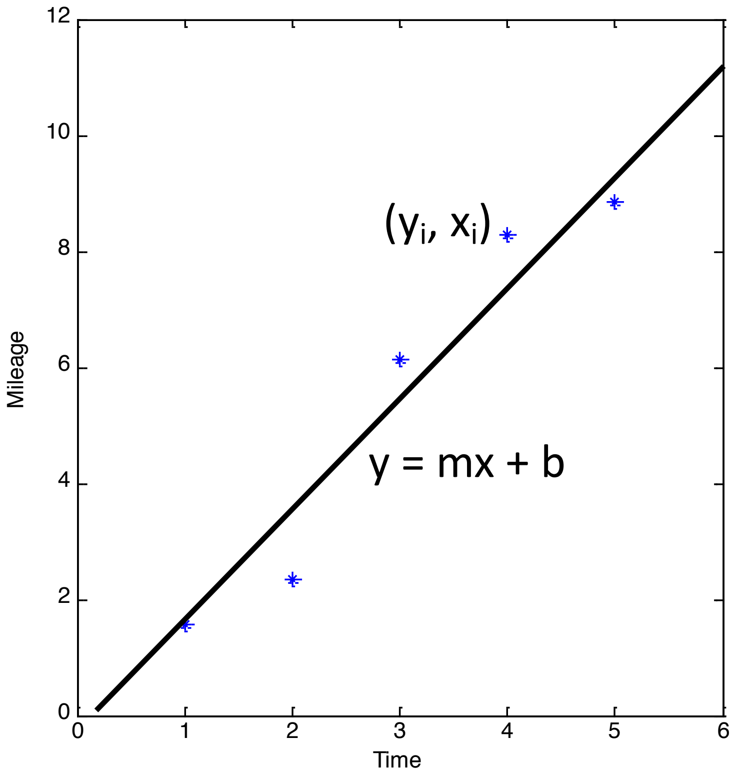
1. Compute image features for A and B
2. Match features between A and B
3. Compute homography between A and B

What could go wrong?

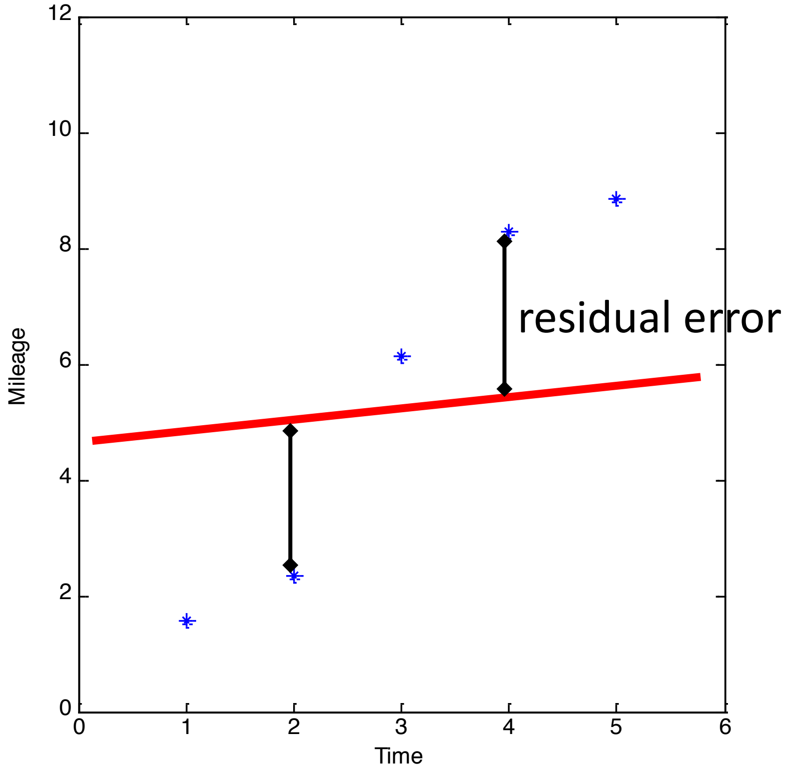
Fitting in general

- Fitting: find the parameters of a model that best fit the data
- Other examples:
  - least squares linear regression

Least squares: linear regression



Linear regression

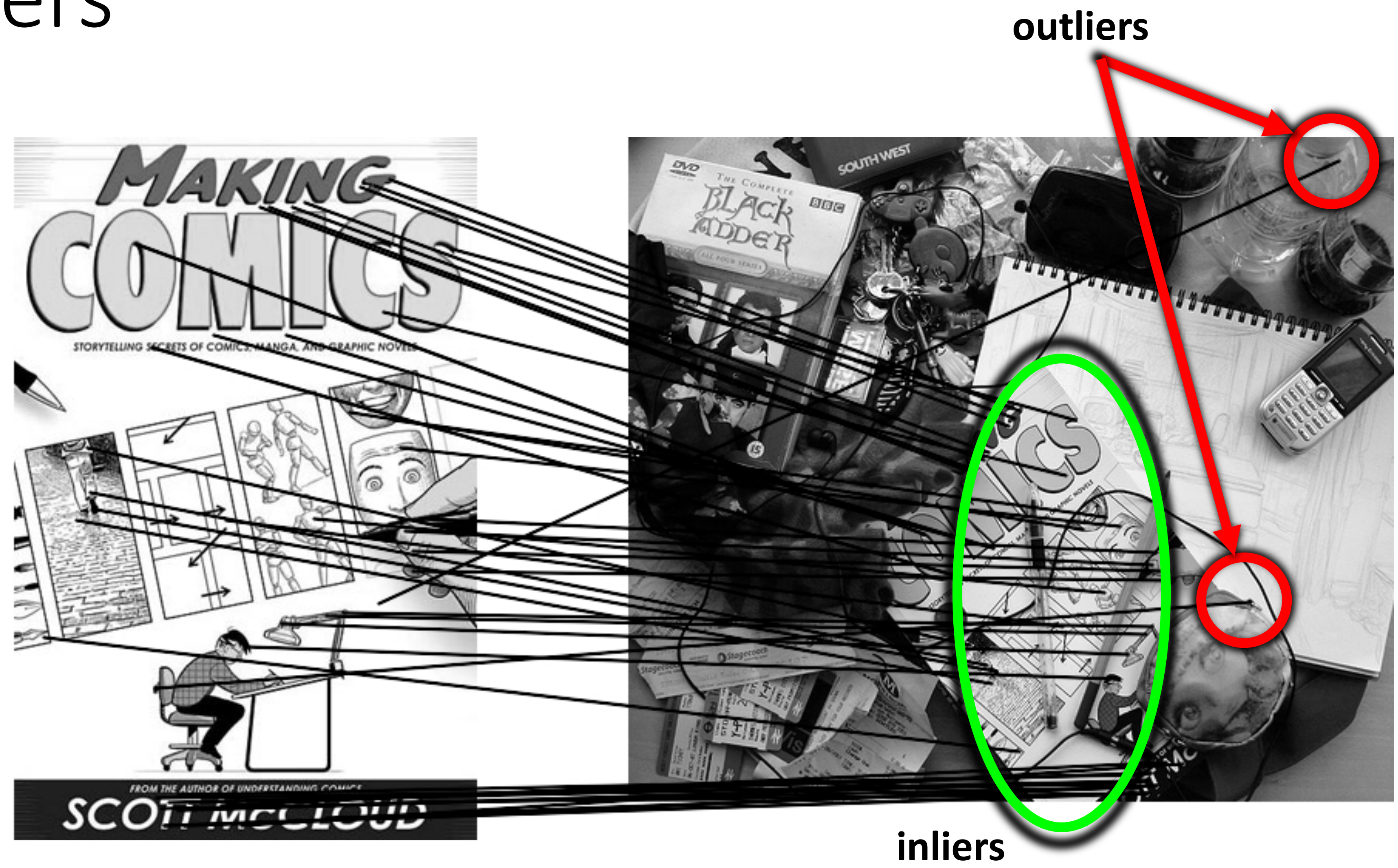


$$\mathrm{Cost}(m, b) = \sum_{i=1}^{n} |y_i - (mx_i + b)|^2$$

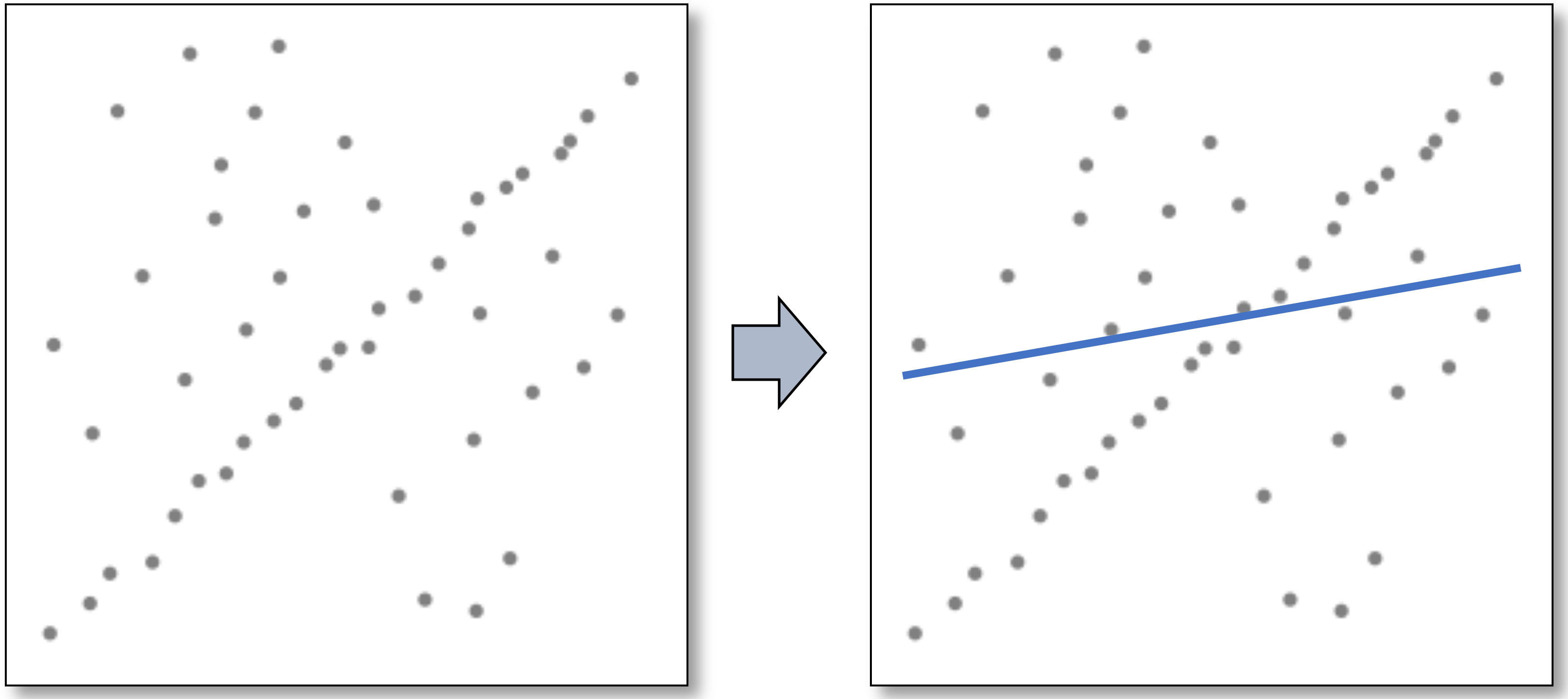
Linear regression

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

Outliers



Robustness



Problem: Fit a line to these datapoints

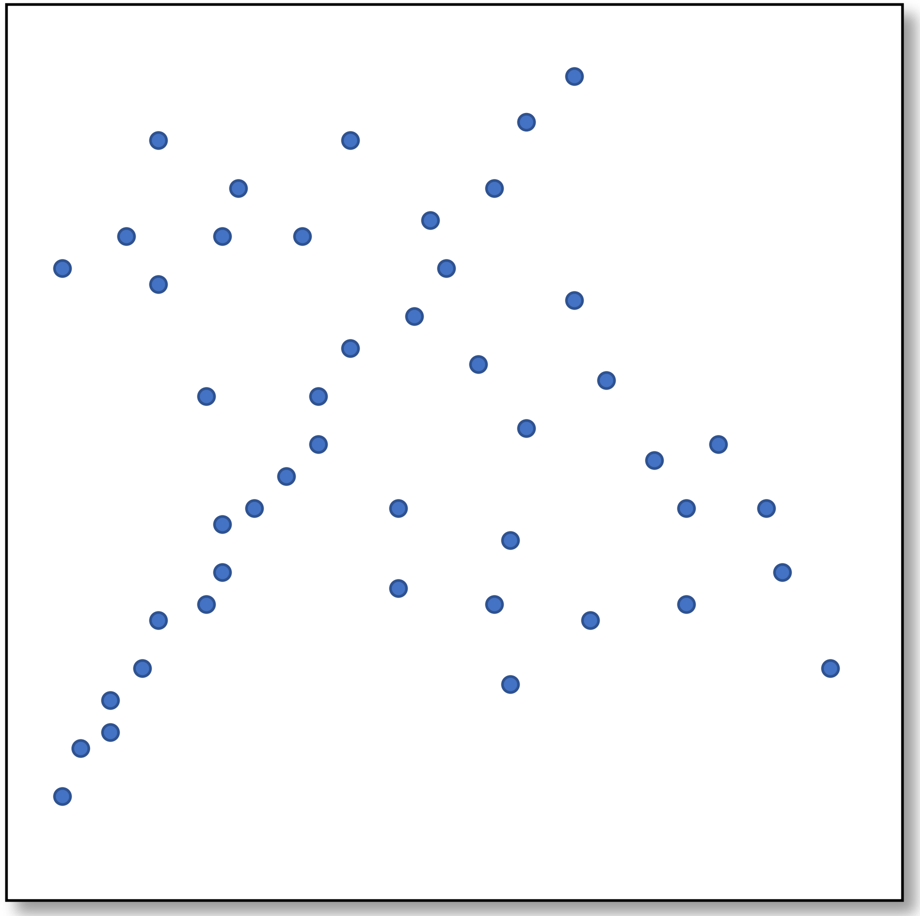
Least squares fit

Idea

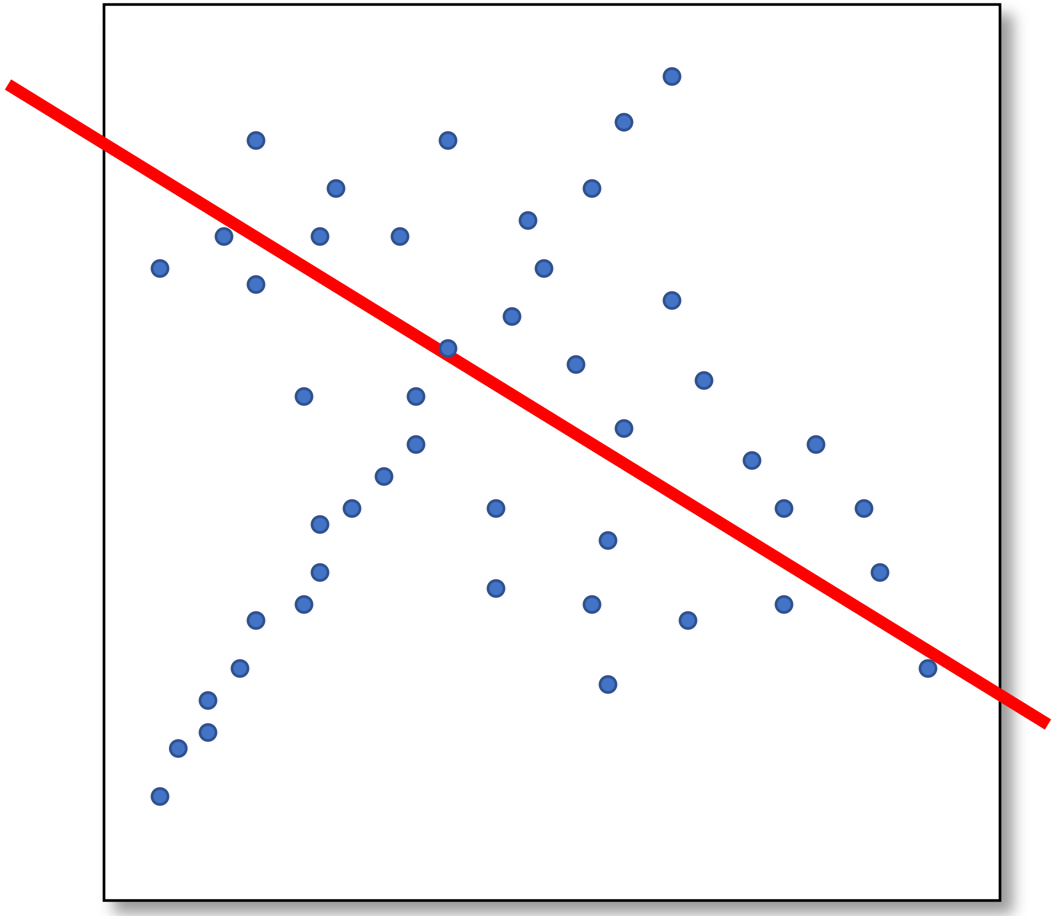
- Given a hypothesized line
- Count the number of points that “agree” with the line
  - “Agree” = within a small distance of the line
  - I.e., the **inliers** to that line

- For all possible lines, select the one with the largest number of inliers

Counting inliers

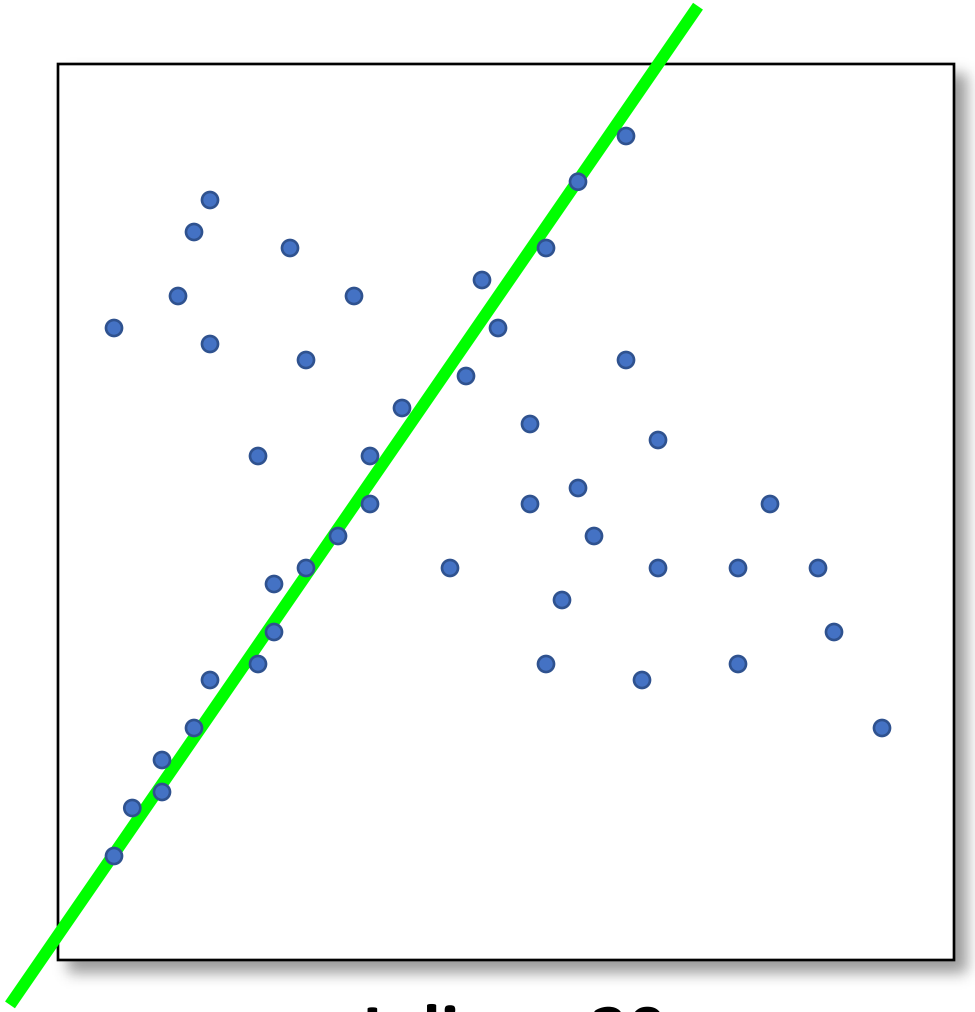


Counting inliers



Inliers: 3

Counting inliers



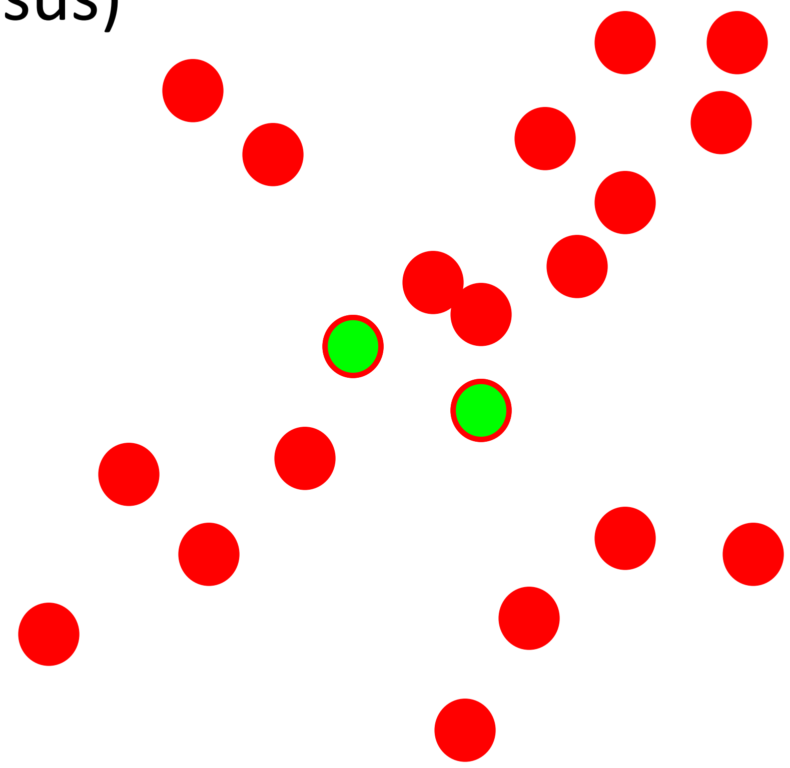
Inliers: 20

How do we find the best line?

- Unlike least-squares, no simple closed-form solution
- Hypothesize-and-test
  - Try out many lines, keep the best one
  - Which lines?

RANSAC (Random Sample Consensus)

Line fitting example



Algorithm:

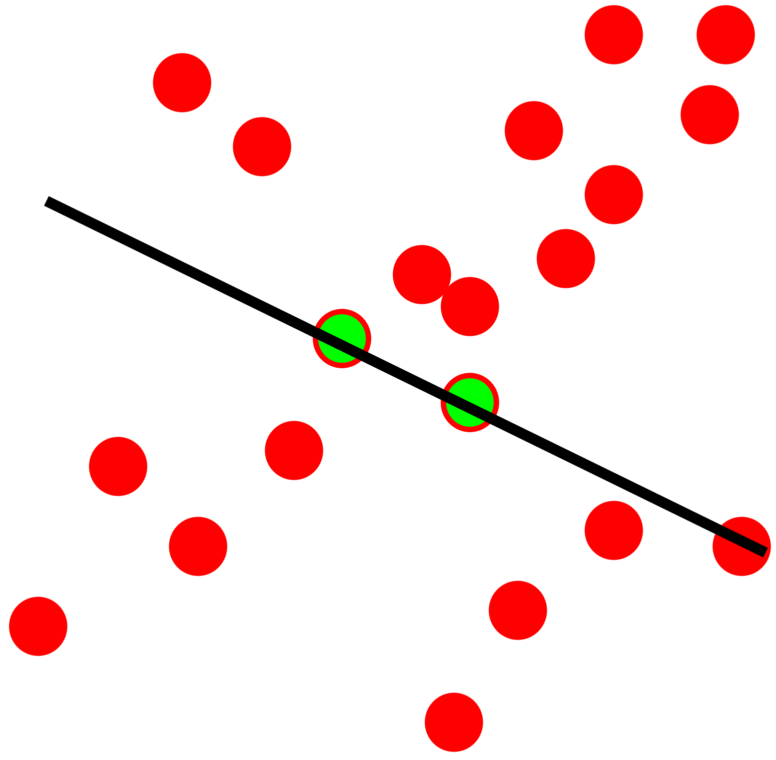
1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Illustration by Savarese

RANSAC

Line fitting example



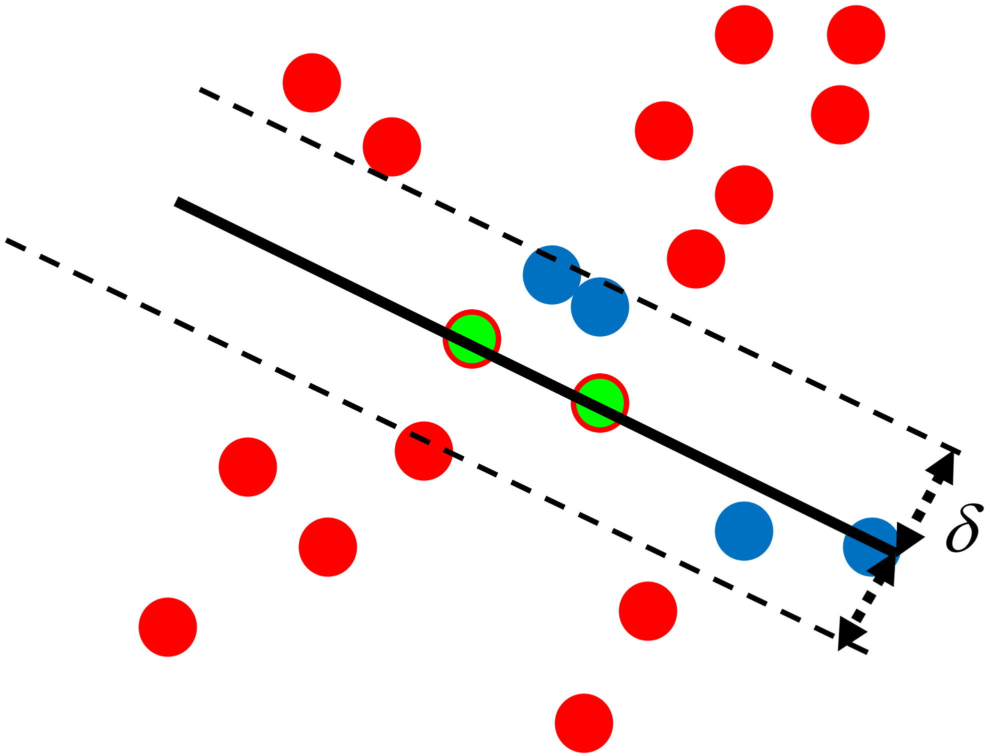
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
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

RANSAC

Line fitting example

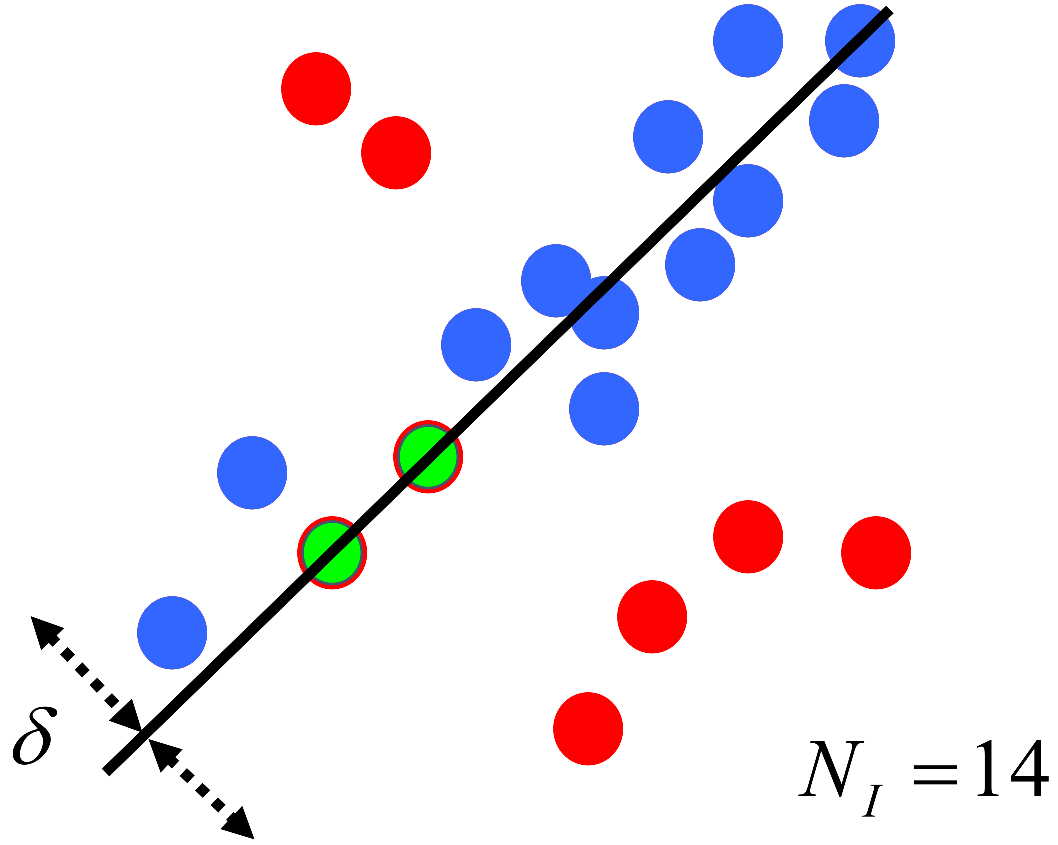


Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
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

RANSAC



Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
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

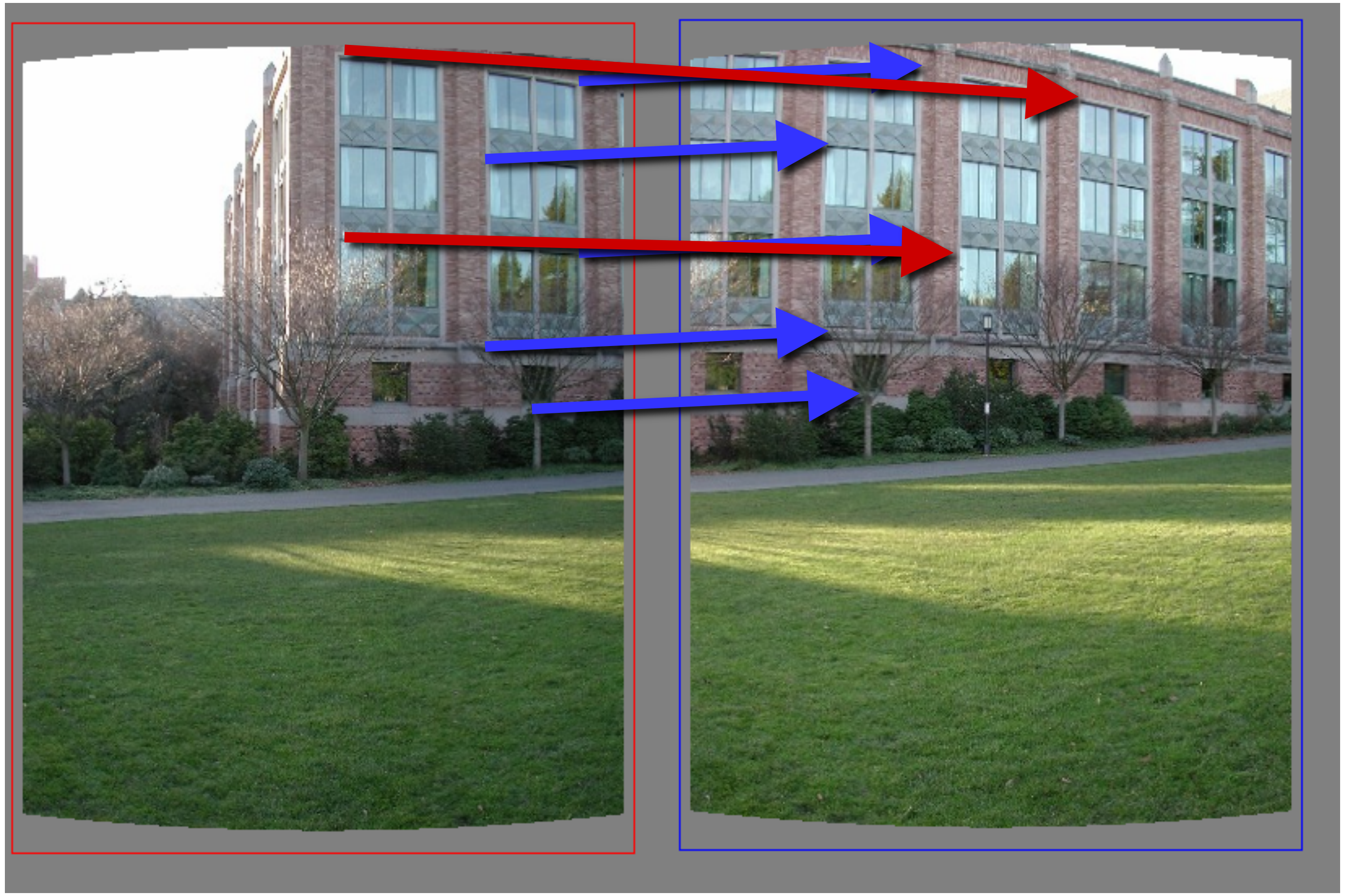
RANSAC

- Idea:
  - All the inliers will agree with each other on the translation vector; the (hopefully small) number of outliers will (hopefully) disagree with each other
    - RANSAC only has guarantees if there are < 50% outliers

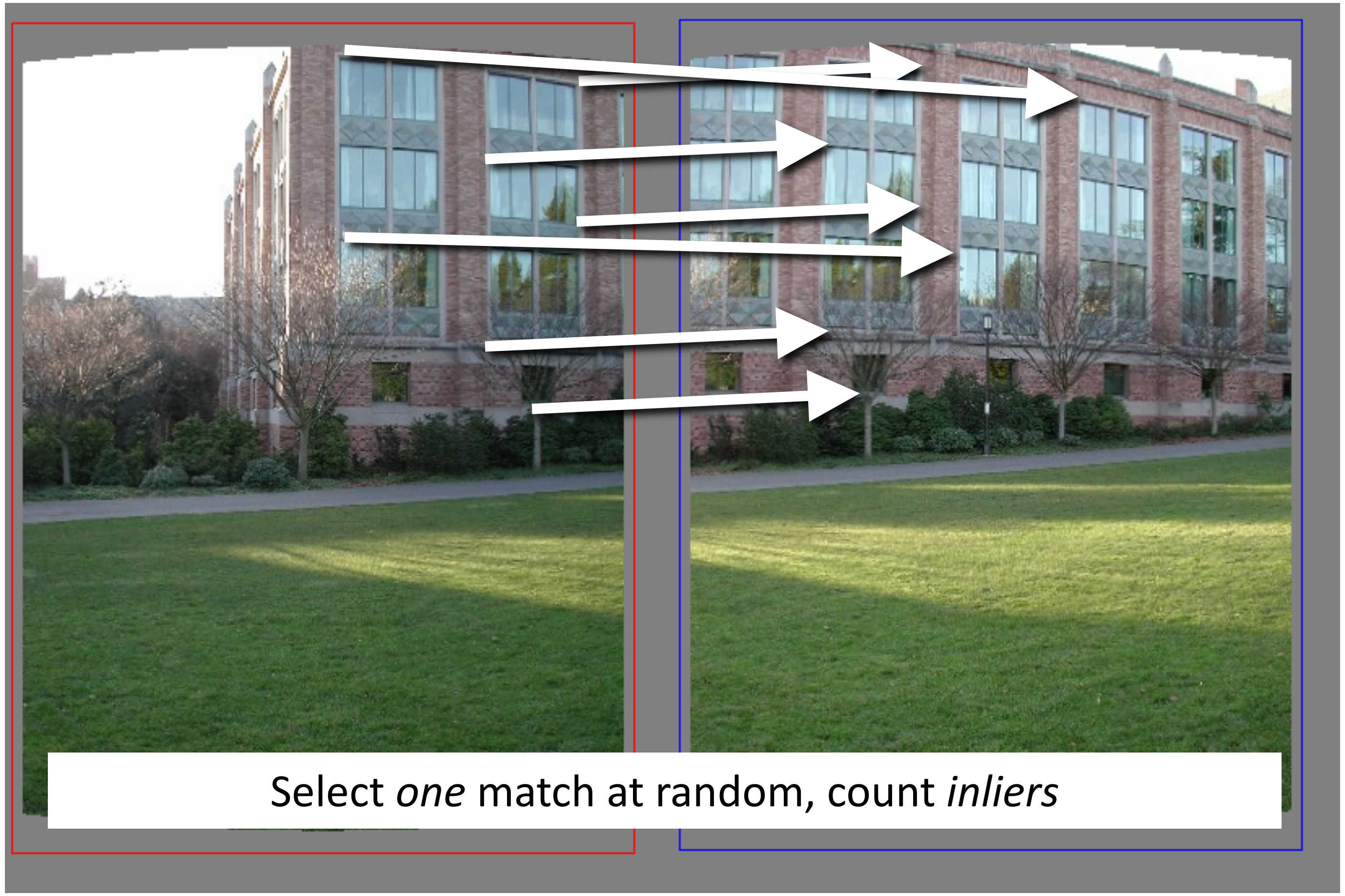
  - “All good matches are alike; every bad match is bad in its own way.”

– Tolstoy via Alyosha Efros

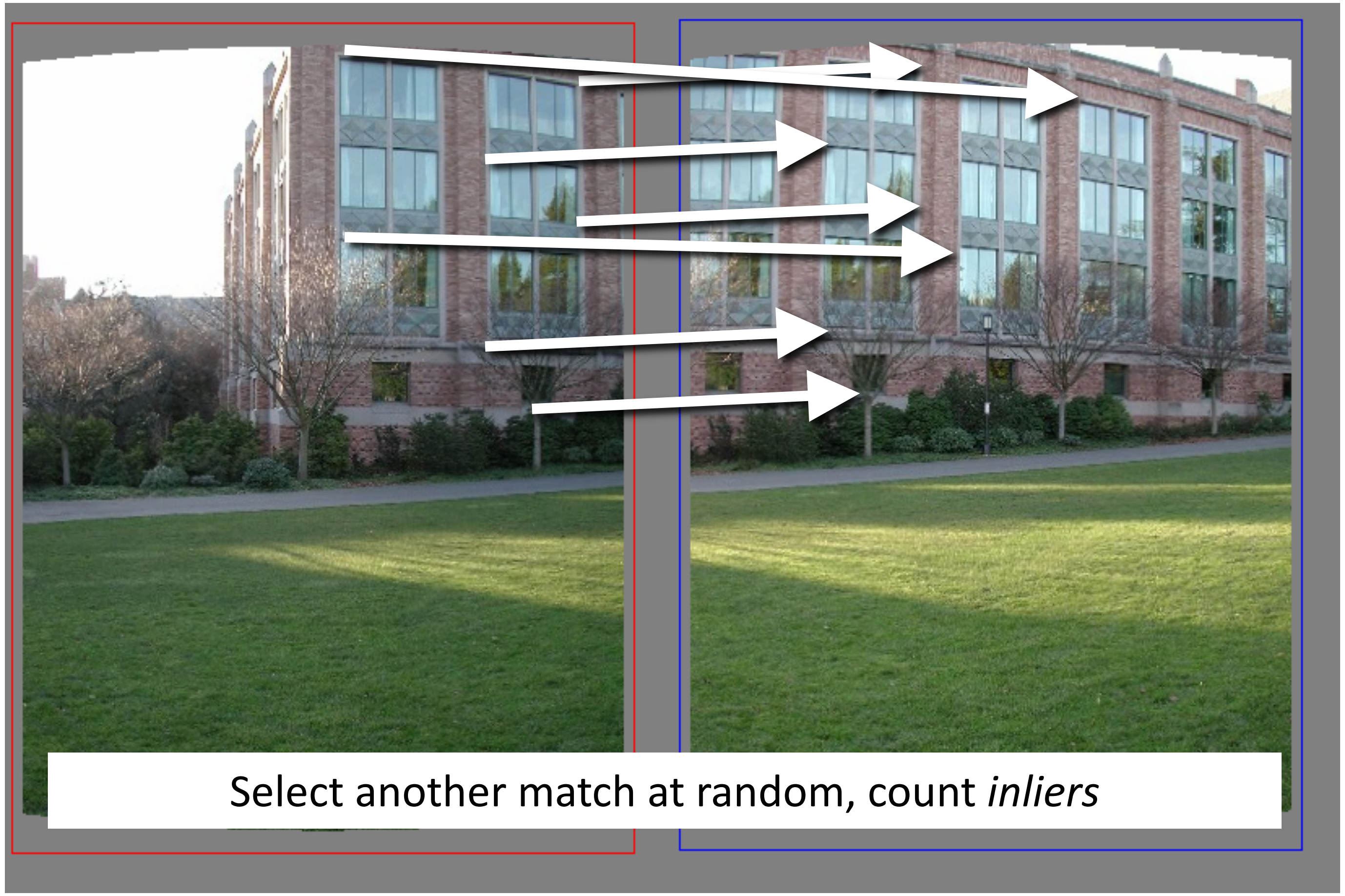
Translations



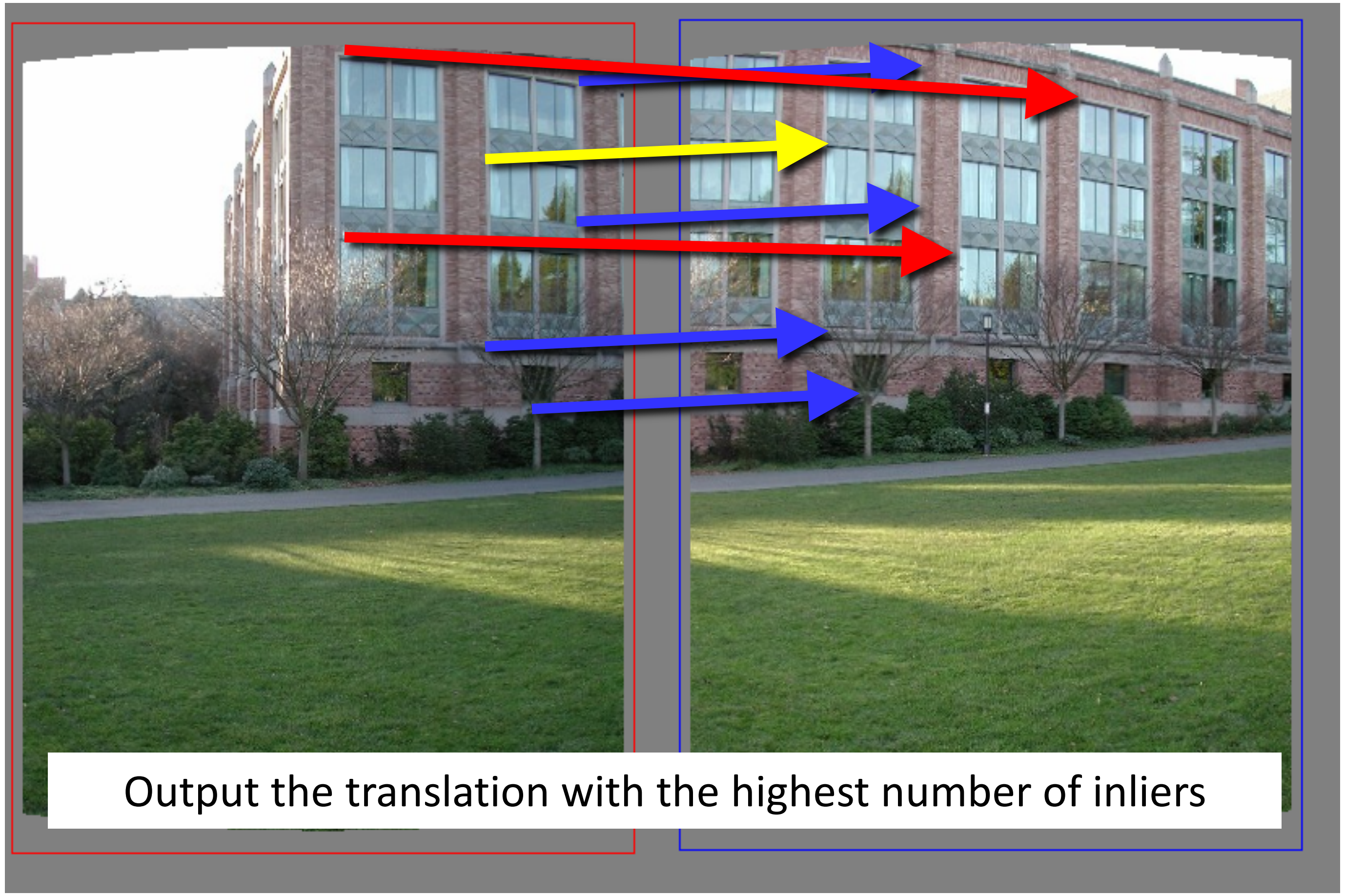
<u>RA</u>ndom <u>SA</u>mple <u>C</u>onsensus



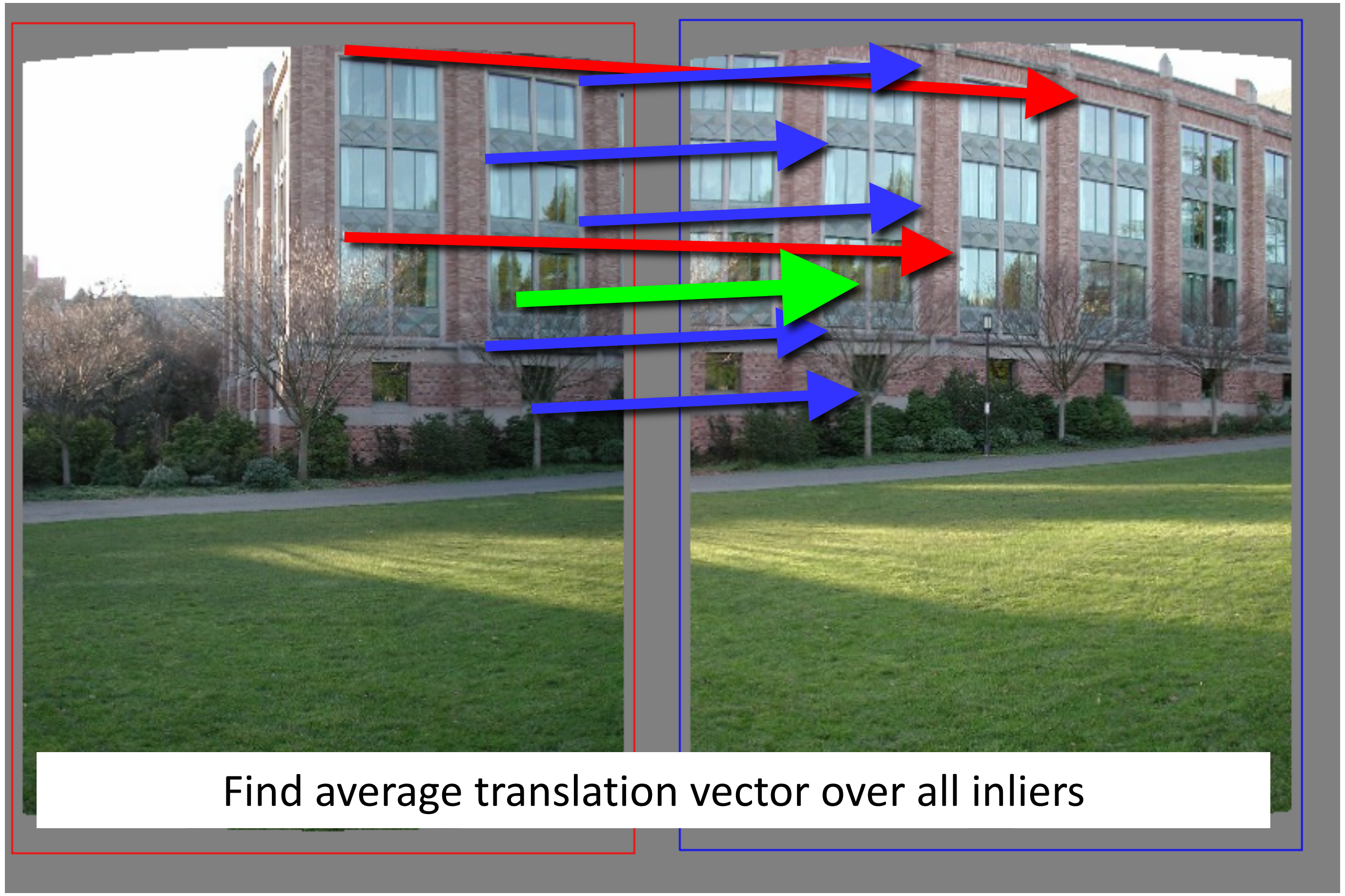
RAndom SAmple Consensus



RAndom SAmple Consensus



Final step: least squares fit



RANSAC

- **Inlier threshold** related to the amount of noise we expect in inliers
  - Often model noise as Gaussian with some standard deviation (e.g., 3 pixels)
- **Number of rounds** related to the percentage of outliers we expect, and the probability of success we'd like to guarantee
  - Suppose there are 20% outliers, and we want to find the correct answer with 99% probability
  - How many rounds do we need?

How many rounds?

- If we have to choose *k* samples each time
  - with an inlier ratio p
  - and we want the right answer with probability *P*

	proportion of inliers *p*						
k	95%	90%	80%	75%	70%	60%	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

$P = 0.99$

Source: M. Pollefeys

To ensure that the random sampling has a good chance of finding a true set of inliers, a sufficient number of trials S must be tried. Let p be the probability that any given correspondence is valid and P be the total probability of success after S trials. The likelihood in one trial that all k random samples are inliers is p^k. Therefore, the likelihood that S such trials will all fail is

$$1 - P = (1 - p^k)^S \tag{6.29}$$

and the required minimum number of trials is

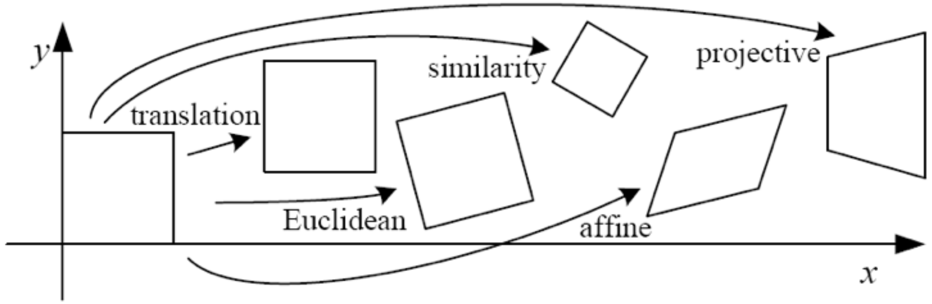
$$S = \frac{\log(1 - P)}{\log(1 - p^k)}. \tag{6.30}$$

	proportion of inliers *p*						
k	95%	90%	80%	75%	70%	60%	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

$P = 0.99$

How big is *k*?

- For alignment, depends on the motion model
  - Here, each sample is a correspondence (pair of matching points)



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Parameters to tune
  - Sometimes too many iterations are required
  - Can fail for extremely low inlier ratios

RANSAC

- An example of a “voting”-based fitting scheme
- Each hypothesis gets voted on by each data point, best hypothesis wins

- There are many other types of voting schemes
  - E.g., Hough transforms…