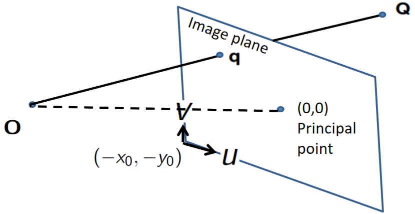
Camera Pose Estimation and RANSAC

Srikumar Ramalingam

School of Computing
University of Utah

Presentation Outline

Camera Models and Projection (Reminder)



- Let the optical center $\mathbf{O}$ be the origin of the camera.
- Let (X^m, Y^m, Z^m) be the coordinates of a 3D point $\mathbf{Q}$, relative to the world system.
- Let the 2D pixel be denoted by $\mathbf{q}(u, v, 1)^T$.

Camera Models and Projection (Reminder)

- Projection of 3D point on the image:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} K & \mathbf{0} \end{pmatrix} \begin{pmatrix} R & -R\mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

- The following 3×3 matrix is the camera matrix:

$$K = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Projection Matrix (Reminder)

- The projection matrix that map 3D points to 2D image is given by:

$$P = \begin{pmatrix} K & \mathbf{0} \end{pmatrix} \begin{pmatrix} R & -R\mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

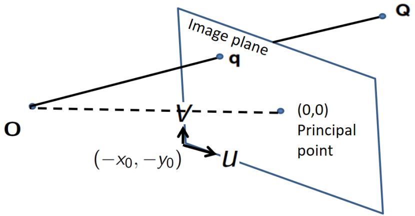
$$P = \begin{pmatrix} KR & -KR\mathbf{t} \end{pmatrix}$$

$$P = KR \begin{pmatrix} I & -\mathbf{t} \end{pmatrix}$$

What is Camera Calibration?

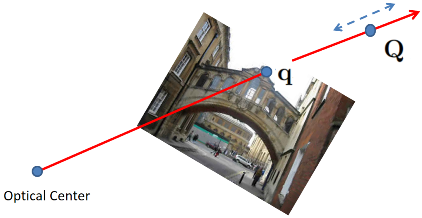
- The task refers to the problem of computing the calibration matrix K.
- In other words, we compute the focal length, principal point, and aspect ratio in the camera matrix.

Forward Projection



$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathrm{KR} \begin{pmatrix} \mathrm{I} & -\mathbf{t} \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

Backward Projection



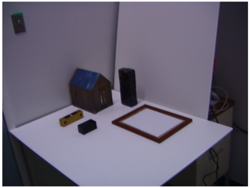
$$\mathbf{Q} \sim \mathsf{K}^{-1}\mathbf{q}$$

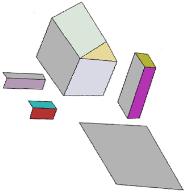
$$\mathbf{Q} \sim \mathsf{K}^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Presentation Outline

What is pose estimation?

The problem of determining the position and orientation of the camera relative to the object (or vice-versa).

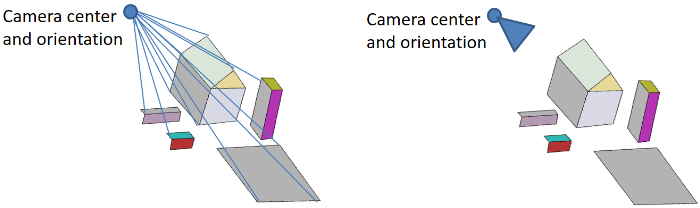




Left: Camera Image, Right: 3D model of the world

What is pose estimation?

The problem of determining the position and orientation of the camera relative to the object (or vice-versa).



We use the correspondences between 2D image pixels (and thus camera rays) and 3D object points (from the world) to compute the pose.

Pose Estimation

- We consider that the camera is calibrated, i.e. we know its calibration matrix K.
- We are given three 2D image to 3D object correspondences. Let the 3 2D points be given by:

$$\mathbf{q_1} = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \quad \mathbf{q_2} = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \quad \mathbf{q_3} = \begin{pmatrix} u_3 \\ v_3 \\ 1 \end{pmatrix}.$$

- Let the 3 3D points be given by:

$$\mathbf{Q_1^m}, \mathbf{Q_2^m}, \mathbf{Q_3^m}$$

Input and Unknowns

Given $\mathbf{q_i}, \mathbf{Q_i^m}, i = \{1, 2, 3\}$, and K in the following equation:

$$\mathbf{q_i} \sim \mathrm{K}\mathbf{R} \begin{pmatrix} \mathbf{I} & -\mathbf{t} \end{pmatrix} \mathbf{Q_i^m}, i = \{1, 2, 3\}$$

Our goal is to compute the rotation matrix R and the translation $\mathbf{t}$.

Pairwise Distance Computation

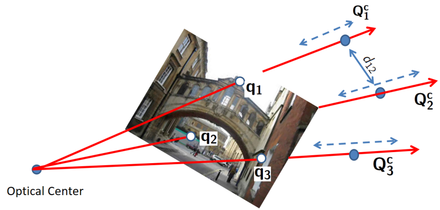
- Given the three 3D points $\mathbf{Q_i^m}$, $i = \{1, 2, 3\}$ we compute the 3 pairwise distances d_{12}, d_{23}, and d_{31} as follows:
  - $$d_{ij} = dist(\mathbf{Q_i^m}, \mathbf{Q_j^m})$$
  - $$dist(\mathbf{Q_i^m}, \mathbf{Q_j^m}) =$$
    $$\sqrt{(X_i^m - X_j^m)^2 + (Y_i^m - Y_j^m)^2 + (Z_i^m - Z_j^m)^2}$$

World frame to Camera frame

- Let the three 3D points $\mathbf{Q_i^m}, i = \{1, 2, 3\}$ be denoted by $\mathbf{Q_i^c}, i = \{1, 2, 3\}$ in the camera coordinate system.
- In other words, we have $\mathbf{Q_i^c} = \mathrm{R}\mathbf{Q_i^m} - \mathrm{R}\mathbf{t}$.
- Here $\mathbf{Q_i^m}$'s are known variables and $\mathbf{Q_i^c}$'s are unknowns.
- It is easy to observe the following since the distance between two points do not change when we transform them from one coordinate frame to another:

$$dist(\mathbf{Q_i^m}, \mathbf{Q_j^m}) = dist(\mathbf{Q_i^c}, \mathbf{Q_j^c})$$

Reformulation of Pose Estimation



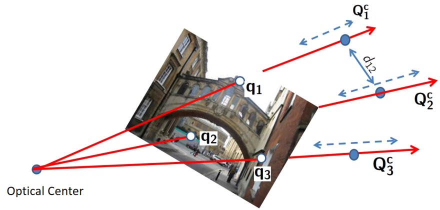
We can compute $\mathbf{Q_i^c}$ as follows:

$$\mathbf{Q_i^c} \sim K^{-1}\mathbf{q_i}$$

$$\mathbf{Q_i^c} = \lambda_i K^{-1}\mathbf{q_i}$$

Here λ_i is an unknown scalar that determines the distance of the 3D point $\mathbf{Q_i^c}$ from the optical center along the ray $\mathbf{OQ_i^c}$.

Reformulation of Pose Estimation

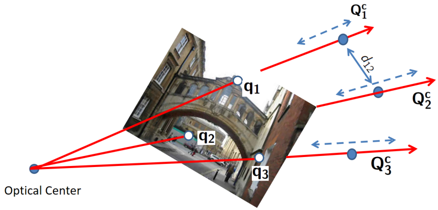


$$\mathbf{Q_i^c} = \lambda_i \mathsf{K}^{-1}\mathbf{q_i}$$

We simplify the notations, let us denote $\mathsf{K}^{-1}\mathbf{q_i}$ as follows:

$$\mathsf{K}^{-1}\mathbf{q_i} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \tag{1}$$

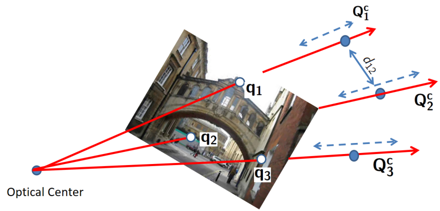
Reformulation of Pose Estimation



$$\mathbf{Q_i^c} = \lambda_i \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

The pose estimation can be seen as the computation of the unknown λ_i parameters.

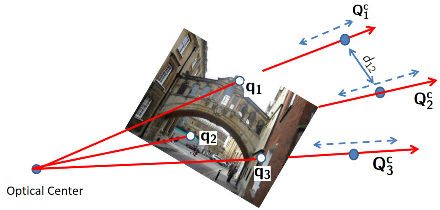
Reformulation of Pose Estimation



$$dist(\mathbf{Q_i^c}, \mathbf{Q_j^c}) = dist(\mathbf{Q_i^m}, \mathbf{Q_j^m}) = d_{ij}, \forall i, j = \{1, 2, 3\}, i \neq j$$

$$\sqrt{(\lambda_i X_i - \lambda_j X_j)^2 + (\lambda_i Y_i - \lambda_j Y_j)^2 + (\lambda_i Z_i - \lambda_j Z_j)^2} = d_{ij}$$

Reformulation of Pose Estimation



$$
\begin{aligned}
(\lambda_1 X_1 - \lambda_2 X_2)^2 + (\lambda_1 Y_1 - \lambda_2 Y_2)^2 + (\lambda_1 Z_1 - \lambda_2 Z_2)^2 &= d_{12}^2 \\
(\lambda_2 X_2 - \lambda_3 X_3)^2 + (\lambda_2 Y_3 - \lambda_3 Y_3)^2 + (\lambda_2 Z_2 - \lambda_3 Z_3)^2 &= d_{23}^2 \\
(\lambda_3 X_3 - \lambda_1 X_1)^2 + (\lambda_3 Y_3 - \lambda_1 Y_1)^2 + (\lambda_3 Z_3 - \lambda_1 Z_1)^2 &= d_{31}^2
\end{aligned}
$$

We have 3 quadratic equations and 3 unknowns.

Reformulation of Pose Estimation

$$
\begin{aligned}
(\lambda_1 X_1 - \lambda_2 X_2)^2 + (\lambda_1 Y_1 - \lambda_2 Y_2)^2 + (\lambda_1 Z_1 - \lambda_2 Z_2)^2 &= d_{12}^2 \\
(\lambda_2 X_2 - \lambda_3 X_3)^2 + (\lambda_2 Y_3 - \lambda_3 Y_3)^2 + (\lambda_2 Z_2 - \lambda_3 Z_3)^2 &= d_{23}^2 \\
(\lambda_3 X_3 - \lambda_1 X_1)^2 + (\lambda_3 Y_3 - \lambda_1 Y_1)^2 + (\lambda_3 Z_3 - \lambda_1 Z_1)^2 &= d_{31}^2
\end{aligned}
$$

- We have 3 quadratic equations and 3 unknowns.
- We can have a total of 2^3 possible solutions for the three parameters $(\lambda_1, \lambda_2, \lambda_3)$.
- Several numerical methods exist to solve the polynomial system of equations.

How to identify a unique solution?

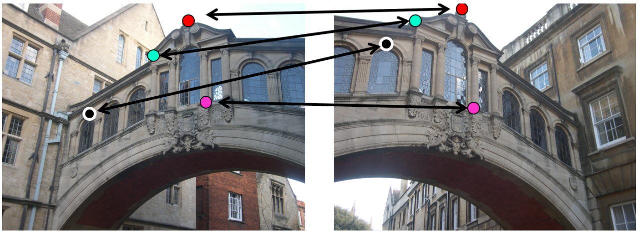
- Out of the 8 solutions, only one will be the correct solution.
- In some of the solutions, the 3D point will be behind the camera.
- Using additional point correspondence, we can identify the correct solution.

Computing the Pose

- We remind you the relation between $\mathbf{Q}_i^c$ and $\mathbf{Q}_i^m$: $\mathbf{Q}_i^c = R\mathbf{Q}_i^m - R\mathbf{t}$.
- We are given $\mathbf{Q}_i^m$ and we have computed $\mathbf{Q}_i^c$.
- From three 3D-to-3D point correspondences we can compute the transformation parameters $(R, \mathbf{t})$ using Horn's method.

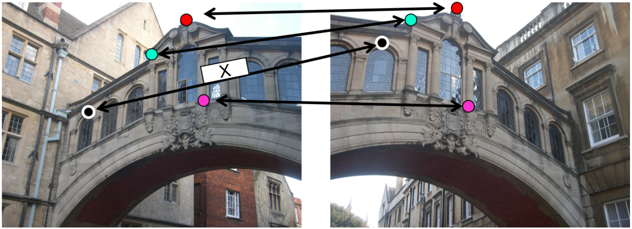
Presentation Outline

Matching Images



We match keypoints from left and right images.

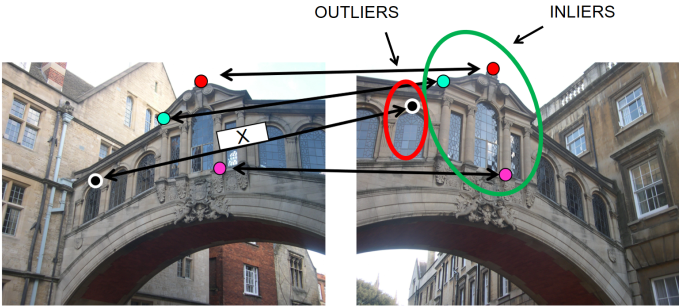
Matching Images



We match keypoints from left and right images.

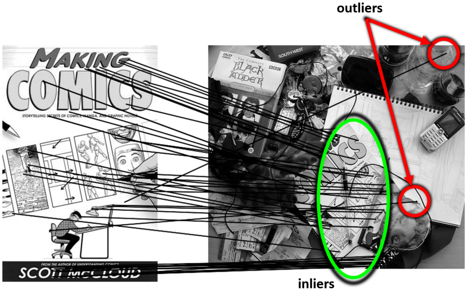
- One of the matches is incorrect!
- In a general image matching problem, we can have 100's of incorrect matches.

Outliers and Inliers



We match keypoints from left and right images.

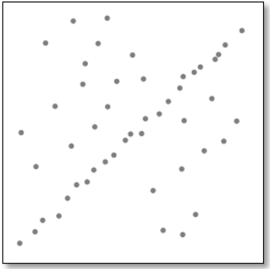
Outliers and Inliers



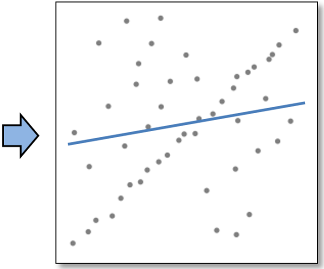
We match keypoints from left and right images.

Robustness

- Let us consider a simpler linear regression problem.



Problem: Fit a line to these datapoints



Least squares fit

- How can we fix this?

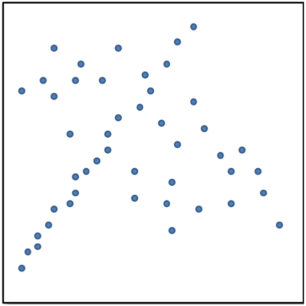
Slide: Noah Snavely

Idea

- Given a hypothesized line.
- Count the number of points that agree with the line, i.e., points within a small distance of the line.
- For all possible lines, select the one with the largest number of inliers.

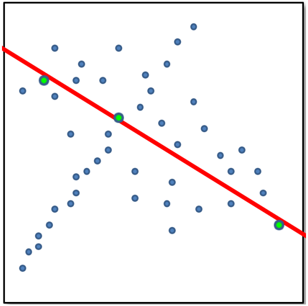
Slide: Noah Snavely

Counting Inliers



Slide: Noah Snavely

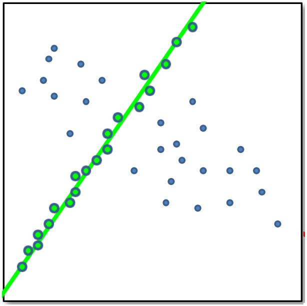
Counting Inliers



- 3 inliers

Slide: Noah Snavely

Counting Inliers



- 20 inliers!

Slide: Noah Snavely

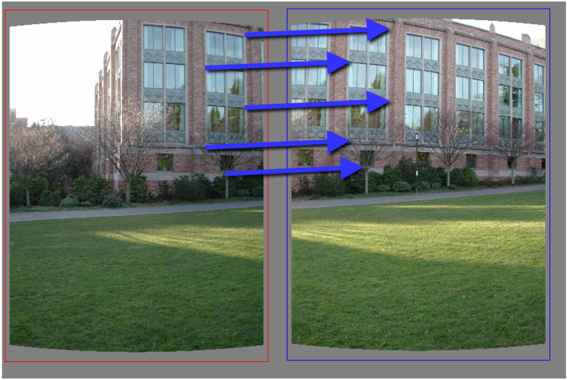
How do we find the best line?

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

Translations

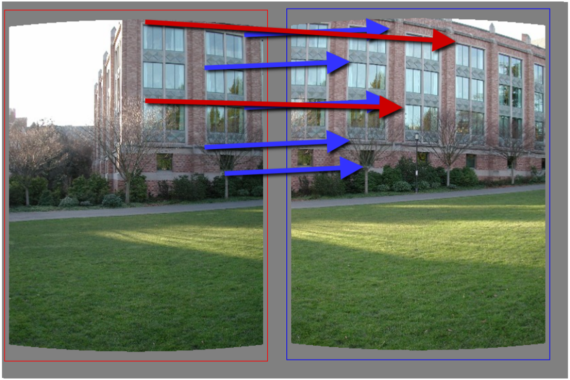
Slide: Noah Snavely

RANdom SAmple Consensus



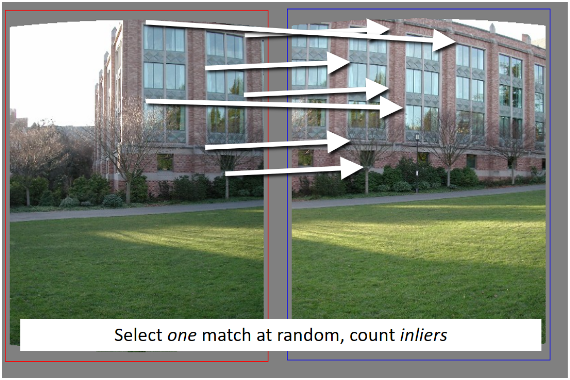
Slide: Noah Snavely

RANdom SAmple Consensus



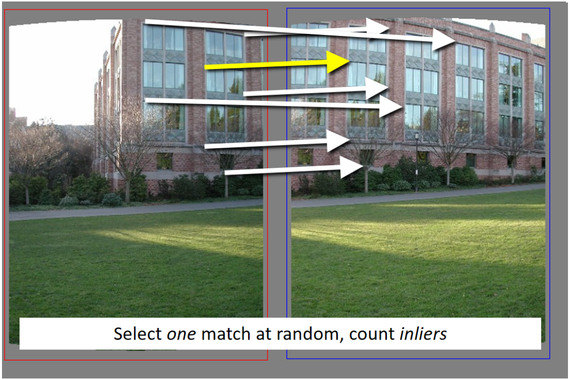
Slide: Noah Snavely

RANdom SAmple Consensus



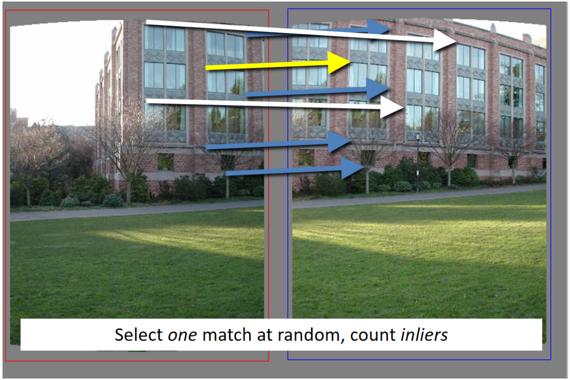
Slide: Noah Snavely

RANdom SAmple Consensus



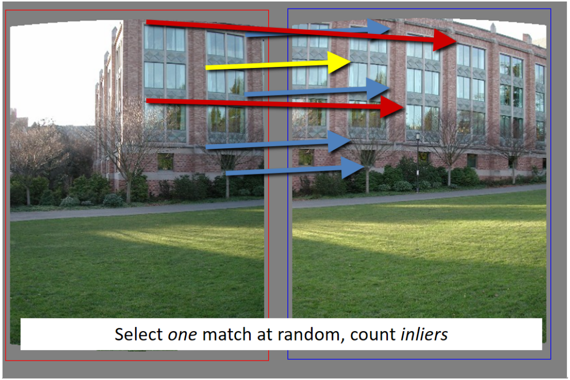
Slide: Noah Snavely

RANdom SAmple Consensus



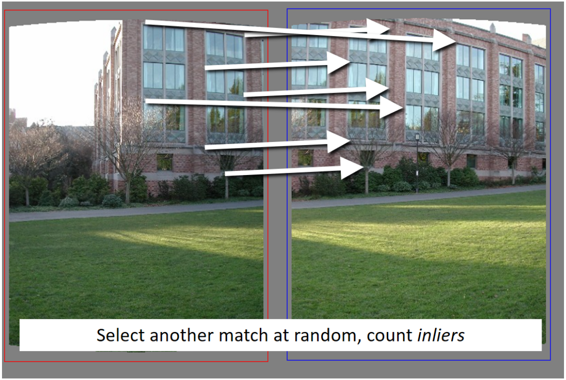
Slide: Noah Snavely

RANdom SAmple Consensus



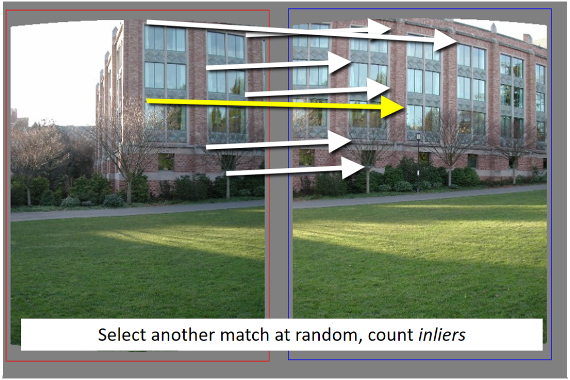
Slide: Noah Snavely

RANdom SAmple Consensus



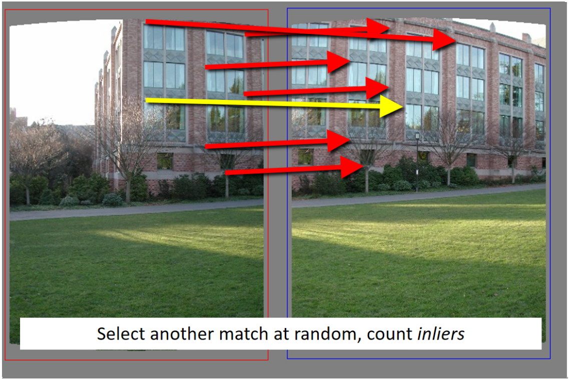
Slide: Noah Snavely

RANdom SAmple Consensus



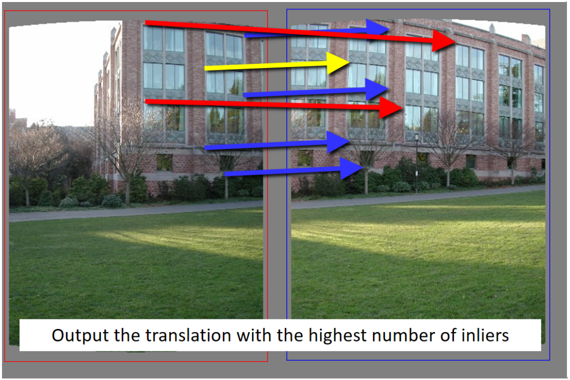
Slide: Noah Snavely

RANdom SAmple Consensus



Slide: Noah Snavely

RANdom SAmple Consensus



Slide: Noah Snavely

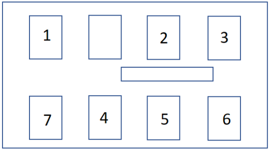
RANSAC

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 $\leq 50\%$ outliers
- All good matches are alike; every bad match is bad in its own way - Alyosha Efros, CMU

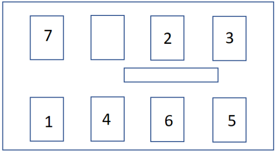
Slide: Noah Snavely

Pose Estimation



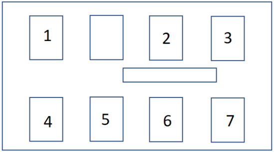
- Inliers?

Pose Estimation



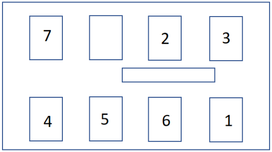
- Inliers?

Pose Estimation



- Inliers?

Pose Estimation

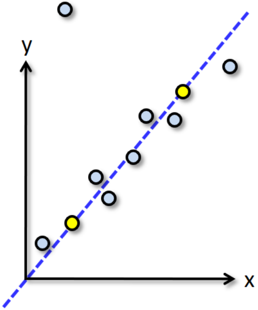


- Inliers?

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 would 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?

Sample size



- How do we generate a hypothesis?

Slide: Noah Snavely

General Version - RANSAC

1. Randomly choose s samples
   - Typically s = minimum sample size that lets you fit a model
2. Fit a model (e.g., line) to those samples
3. Count the number of inliers that approximately fit the model
4. Repeat N times
5. Choose the model that has the largest set of inliers

Slide: Noah Snavely

How many rounds?

	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

$p = 0.99$

- If we have to choose s samples each time
  - with an outlier ratio e
  - and we want the right answer with probability p

Slide: M. Pollefeys

Acknowledgments

Some presentation slides are adapted from the following materials:

- Peter Sturm, Some lecture notes on geometric computer vision (available online).
- Kristen Grauman's computer vision lecture slides
- Noah Snavely's computer vision lecture slides