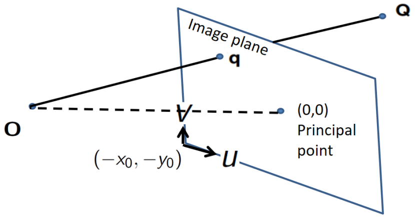
3D Reconstruction

Srikumar Ramalingam

School of Computing
University of Utah

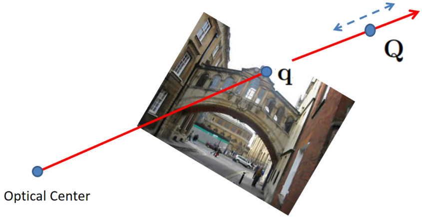
Presentation Outline

Forward Projection (Reminder)



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

Backward Projection (Reminder)

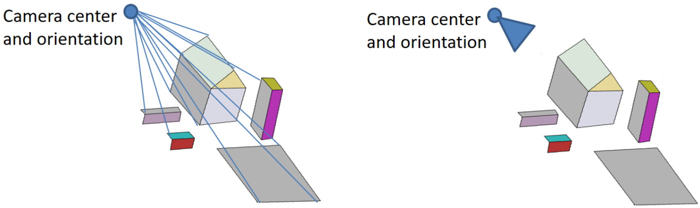


$$\mathbf{Q} \sim \mathsf{K}^{-1}\mathbf{q}$$

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

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:

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

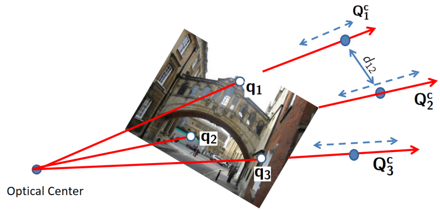
Input and Unknowns

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 $\mathbf{t}$.

Reformulation of Pose Estimation



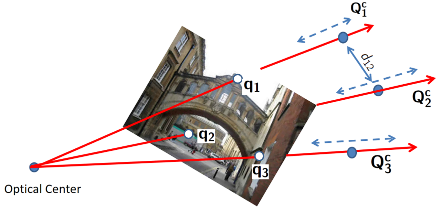
We can compute $\mathbf{Q_i^c}$ as follows:

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

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

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

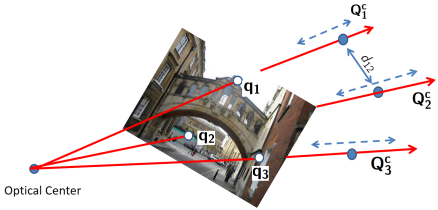


$$\mathbf{Q_i^c} = \lambda_i \mathsf{K}^{-1}\mathbf{q_i}$$

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

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

- 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.

Presentation Outline

Sample Pose Estimation Problem

Compute the solution for pose estimation when λ_1 is given.

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

- Compute λ_2 from the first equation.
- Compute λ_3 from the third equation.
- Use the second equation to remove incorrect solutions for λ_2 and λ_3.

Pose Estimation

- We consider that the camera is calibrated, i.e. we know its calibration matrix K.

$$K = \begin{pmatrix} 200 & 0 & 320 \\ 0 & 200 & 240 \\ 0 & 0 & 1 \end{pmatrix}$$

$$K^{-1} = \frac{1}{200} \begin{pmatrix} 1 & 0 & -320 \\ 0 & 1 & -240 \\ 0 & 0 & 200 \end{pmatrix}$$

- We are given three 2D image to 3D object correspondences. Let the 3 2D points be given by:

$$\mathbf{q_1} = \begin{pmatrix} 320 \\ 140 \\ 1 \end{pmatrix} \quad \mathbf{q_2} = \begin{pmatrix} 320 - 50\sqrt{3} \\ 290 \\ 1 \end{pmatrix} \quad \mathbf{q_3} = \begin{pmatrix} 320 + 50\sqrt{3} \\ 290 \\ 1 \end{pmatrix}.$$

- Let the inter-point distances be given by $\{d_{12} = 1000, d_{23} = 1000, d_{31} = 1000\}$
- Is it possible to have $\lambda_1 \neq \lambda_2$?

Pose Estimation using n correct correspondences

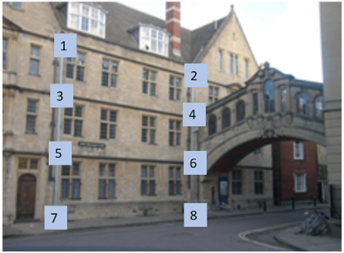
- We can compute the pose using 3 correct correspondences.
- How to compute pose using n correspondences, with outliers.
  - Use RANSAC to identify m inliers where $m \leq n$.
  - Use least squares to find the best pose using all the inliers - basic idea is to use all the forward projection equations for all the inliers and compute R and t.

General Version - RANSAC (REMINDER)

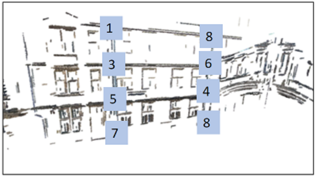
1. Randomly choose s samples
   - Typically s = minimum sample size that lets you fit a model
2. Fit a model (e.g., line) to those samples
3. Count the number of inliers that approximately fit the model
4. Repeat N times
5. Choose the model that has the largest set of inliers

Slide: Noah Snavely

Let us do RANSAC!

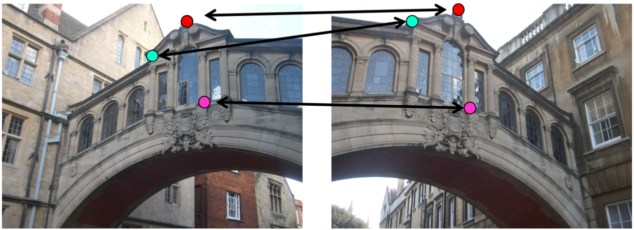


IMAGE



3D MODEL

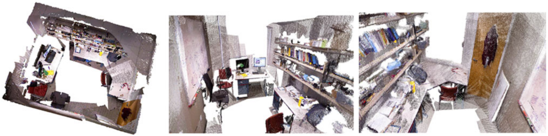
Matching Images



We match keypoints from left and right images.

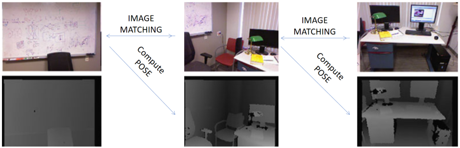
- 2D-to-2D image matching using descriptors such as SIFT.

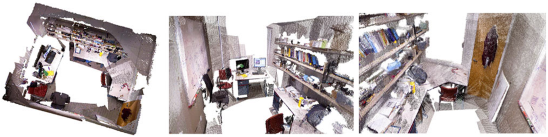
Kinect Sample Frames



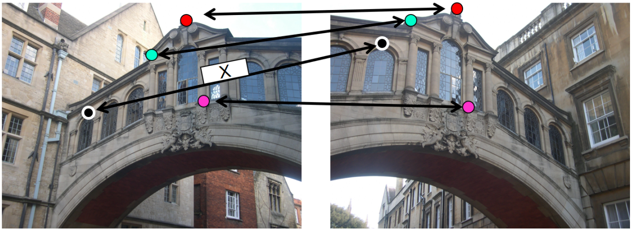
- Sequences of RGBD frames $(I_1, D_1), (I_2, D_2), (I_3, D_3), ..., (I_n, D_n)$.
- How to register Kinect depth data for reconstructing large scenes?
- We have 2D-3D pose estimators and 2D-2D image matchers.

Kinect Sample Frames





Matching Images

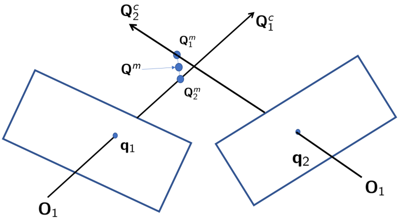


We match keypoints from left and right images.

- One of the matches is incorrect!
- In a general image matching problem with 1000s of matches, we can have 100's of incorrect matches.

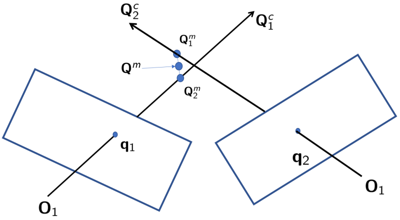
Presentation Outline

3D Reconstruction (Two view triangulation)



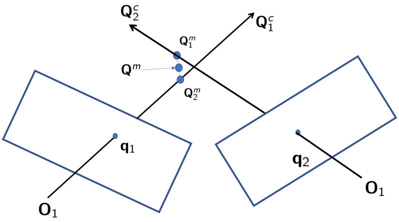
- Given: calibration matrices - (K_1, K_2).
- Given: Camera poses - $\{(R_1, \mathbf{t}_1), (R_2, \mathbf{t}_2)\}$ are known.
- Given: 2D point correspondence - $(\mathbf{q}_1, \mathbf{q}_2)$.
- Our goal is to find the associated 3D point $\mathbf{Q}^m$.

3D Reconstruction (Two view triangulation)



- Due to noise, the back-projected rays don't intersect.
- The required point is given by $\mathbf{Q}^m = \frac{\mathbf{Q}_1^m + \mathbf{Q}_2^m}{2}$.
- The 3D point on the first back-projected ray is given by: $\mathbf{q}_1 \sim K_1 R_1 (I - \mathbf{t}_1) \mathbf{Q}_1^m$.
- The 3D point on the second back-projected ray is given by: $\mathbf{q}_2 \sim K_2 R_2 (I - \mathbf{t}_2) \mathbf{Q}_2^m$.

3D Reconstruction (Two view triangulation)



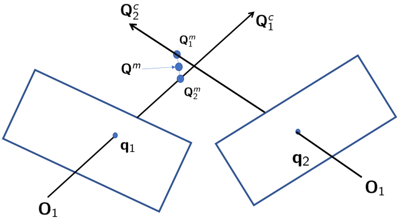
- Let us parametrize the 3D points using λ_1 and λ_2:

$$\begin{aligned}\mathbf{Q}_1^m &= \mathbf{t}_1 + \lambda R_1^T K_1^{-1} \mathbf{q}_1 \\ \mathbf{Q}_2^m &= \mathbf{t}_2 + \lambda R_2^T K_2^{-1} \mathbf{q}_2\end{aligned}$$

- We rewrite using 3×1 constant vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ for simplicity:

$$\begin{aligned}\mathbf{Q}_1^m &= \mathbf{a} + \lambda \mathbf{b} \\ \mathbf{Q}_2^m &= \mathbf{c} + \lambda \mathbf{d}\end{aligned}$$

3D Reconstruction (Two view triangulation)



- We can compute λ_1 and λ_2 as follows:

$$[\lambda_1, \lambda_2] = \arg \min_{\lambda_1, \lambda_2} dist(\mathbf{Q}_1^m, \mathbf{Q}_2^m)$$

3D Reconstruction (Two view triangulation)

$$dist(\mathbf{Q}_1^m, \mathbf{Q}_2^m) = \sqrt{\sum_{i=1}^{3}(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i)^2}$$

$$[\lambda_1, \lambda_2] = \arg\min_{\lambda_1,\lambda_2} \sqrt{\sum_{i=1}^{3}(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i)^2}$$

$$[\lambda_1, \lambda_2] = \arg\min_{\lambda_1,\lambda_2} \sum_{i=1}^{3}(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i)^2$$

$$D_{sqr} = \sum_{i=1}^{3}(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i)^2$$

3D Reconstruction (Two view triangulation)

$$D_{sqr} = \sum_{i=1}^{3} (a_i + \lambda_1 b_i - c_i - \lambda_2 d_i)^2$$

At minima:

$$\frac{\partial D_{sqr}}{\partial \lambda_1} = \sum_{i=1}^{3} 2(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i) b_i = 0$$

$$\frac{\partial D_{sqr}}{\partial \lambda_2} = \sum_{i=1}^{3} 2(a_i + \lambda_1 b_i - c_i - \lambda_2 d_i) d_i = 0$$

We have two linear equations with two variables λ_1 and λ_2. This can be solved!

Once λ's are computed then we can obtain:

$$\begin{aligned}\mathbf{Q}_1^m &= \mathbf{a} + \lambda_1 \mathbf{b} \\ \mathbf{Q}_2^m &= \mathbf{c} + \lambda_2 \mathbf{d}\end{aligned}$$

We can compute the required intersection point $\mathbf{Q}^m$ from the mid-point equation: $\mathbf{Q}^m = \frac{\mathbf{Q}_1^m + \mathbf{Q}_2^m}{2}$

Sample 3D Reconstruction

- Calibration matrices:

$$K_1 = K_2 = \begin{pmatrix} 200 & 0 & 320 \\ 0 & 200 & 240 \\ 0 & 0 & 1 \end{pmatrix}$$

- Rotation matrices: $R_1 = R_2 = I$.
- Translation matrices: $\mathbf{t}_1 = \mathbf{0}, \mathbf{t}_2 = (100, 0, 0)^T$.
- Correspondence:

$$\mathbf{q_1} = \begin{pmatrix} 520 \\ 440 \\ 1 \end{pmatrix} \mathbf{q_2} = \begin{pmatrix} 500 \\ 440 \\ 1 \end{pmatrix}$$

- Compute the 3D point $\mathbf{Q}^m$.

Simple 3D Reconstruction Pipeline

1. Given a sequence of images $\{I_1, I_2, ..., I_n\}$ with known calibration, obtain 3D reconstruction.
2. Compute correspondences for the image pair (I_1, I_2).
3. Find the motion between I_1 and I_2 using motion estimation algorithm (next class).
4. Compute partial 3D point cloud P_{3D} using the point correspondences from (I_1, I_2).
5. Initialize $k = 3$.
6. Compute correspondences for the pair (I_{k-1}, I_k) and compute the pose of I_k with respect to P_{3D}.
7. Increment P_{3D} using 3D reconstruction from (I_{k-1}, I_k).
8. $k = k + 1$
9. If $k < n$ go to Step 5.

Acknowledgments

Some presentation slides are adapted from the following materials:

- Peter Sturm, Some lecture notes on geometric computer vision (available online).