Stereo Wrap Up
and
Structure from Motion

Introduction to Computer Vision

CSE 152

Lecture 8

Announcements

- HW1 solution posted to Piazza
- HW2 posted, due Friday 2/14
- Presidents day (No Class), Mon 2/17
- Midterm, Wed 2/19

Stereo Vision Outline

- Offline:

B Calibrate cameras & determine epipolar geometry

- Online

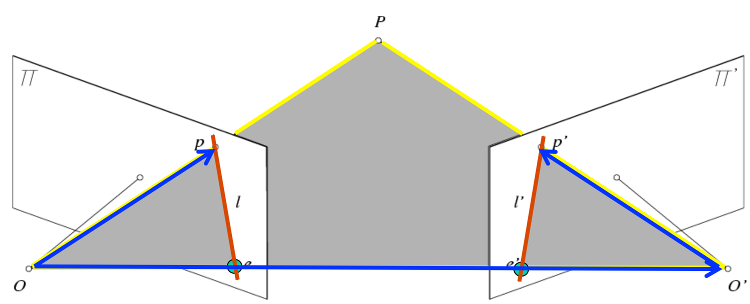
1. Acquire stereo images

C 2. Rectify images to convenient epipolar geometry

D 3. Establish correspondence

A **4. Estimate depth**

Epipolar Constraint: **Calibrated** Case



The vectors $\overrightarrow{Op}$, $\overrightarrow{OO'}$ and $\overrightarrow{O'p'}$ are coplanar

$$\overrightarrow{Op}\cdot[\overrightarrow{OO'}\times\overrightarrow{O'p'}]=0$$

$$ {}^1\mathbf{p}\cdot\left[{}^1\mathbf{t}_2\times({}^1_2\mathbf{R}\,{}^2\mathbf{p}')\right]=0$$

Essential Matrix
(Longuet-Higgins, 1981)

$${}^1\mathbf{p}^T\mathbf{E}\,{}^2\mathbf{p}'=0 \text{ with } \mathbf{E}=[({}^1\mathbf{t}_2)_\times]^1_2\mathbf{R}$$

skew

How do we use the Essential Matrix E?

Given a pixel coordinates **q** in image 1, what is
epipolar line equation in image 2?

$$ {}^1\mathbf{p}^T \mathbf{E}\, {}^2\mathbf{p}' = 0 \text{ with } \mathbf{E} = [({}^1\mathbf{t}_2)_\times]\, {}^1_2\mathbf{R} $$

- Given a point in Image 1 with homogenous pixel coordinates **q**, convert it to a direction as ${}^1\mathbf{p} = (\mathbf{K}_1^{-1})\,\mathbf{q}$
- Let $\mathbf{a} = {}^1\mathbf{p}^T\mathbf{E}$ where **a** is 1 X 3 vector.
- Then apply the epipolar constraint, we have $\mathbf{a}\, {}^2\mathbf{p}' = 0$
- And converting direction to image coordinates with ${}^2\mathbf{p}' = \mathbf{K}_2^{-1}\,\mathbf{q}'$, we have that

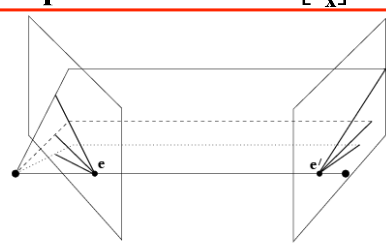
$$ (\mathbf{a}\mathbf{K}_2^{-1})\,\mathbf{q}' = \mathbf{0} $$

is the equation of the epipolar line in Image 2.

- Likewise, given a point **q'** in Image 2, we can obtain a line equation in Image 1.

Computing the epipoles given E

$$ \boxed{\mathbf{p}^T\mathbf{E}\mathbf{p}' = 0 \quad \text{with} \quad \mathbf{E} = [\mathbf{t}_x]\mathbf{R}} $$

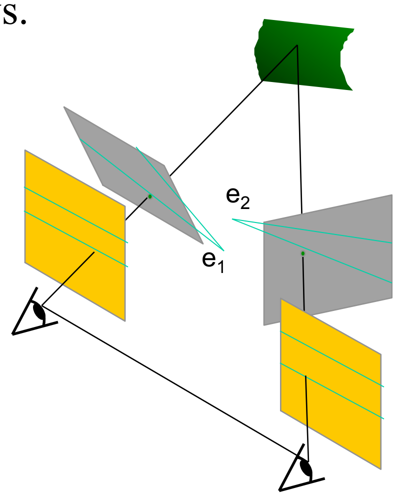


- The epipole **e'** in the right image is the Eigenvector of **E** corresponding to the zero eigenvalue.
- The epipole **e** in the left image is the Eigenvector of $\mathbf{E}^T$ corresponding to the zero eigenvalue.

Why? See explanation in slides from lecture 7

What if stereo geometry isn't convenient?

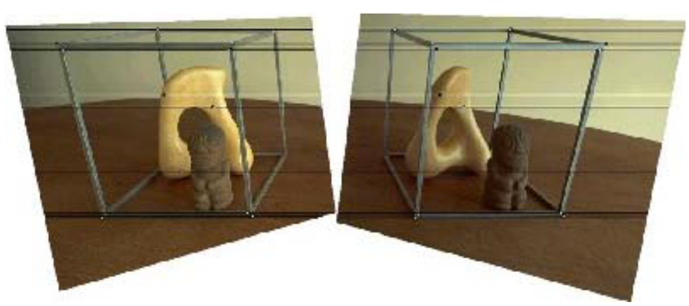
Rectificaiton: Given a pair of images, transform both images so that epipolar lines are image rows.



Rectification

Original images

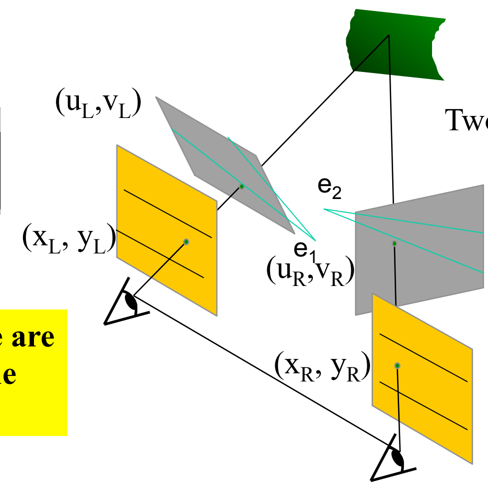
Rectified images



Rectification

Under perspective projection, the mapping from a plane to a plane is given by a linear transformation of homogeneous coordinates (called a projective transformation or homography).

$$\begin{bmatrix} x_L \\ y_L \\ w_L \end{bmatrix} = H_L \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix}$$



Two images
Two homographies
H_L, H_R

$$\begin{bmatrix} x_R \\ y_R \\ w_R \end{bmatrix} = H_R \begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix}$$

Question: Where are the epipoles in the rectified image?

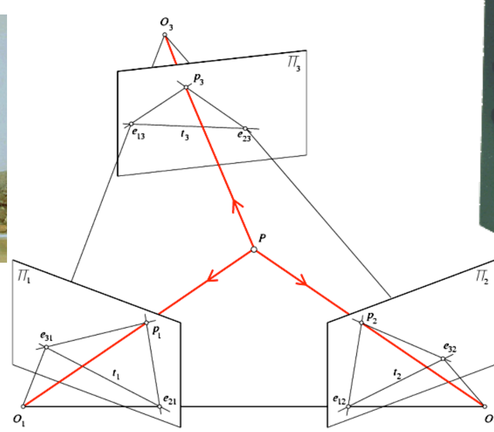
HW2

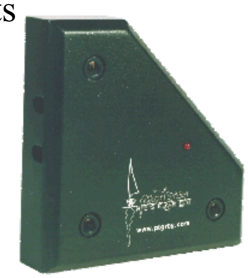
- Depth estimation equation
- Epipolar geometry
  - Code is given for computing Essential/Fundamental matrix.
  - You will
    - Draw epipolar lines
    - Find epipoles
- Rectification: You're given code for computing H_L and H_R. You need to actually rectify the images.
- Matching:
  - You will do sparse matching using feature points
  - You will improve matching using epipolar geometry
- Structure from Motion using RANSAC

Variations on Binocular Stereo

1. **Trinocular Stereopsis**
2. Multiview stereo
3. **Uncalibrated Stereo**
4. Helmholtz Reciprocity Stereopsis

Trinocular Epipolar Constraints





- Trinocular stereo can remove the binocular ambiguity.
- For each potential binocular match, the third image can be used to the match.
- There's no need to search in the third image.

Relation of calibrated camera coordinates and pixel coordinates

$$M = K\Pi {}^c_wT = \begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c_wR & {}^cO_w \\ \mathbf{0}^T & 1 \end{bmatrix}$$

K Intrinsic Parameters; Π Projection 3 x 4; ${}^c_wT$ Extrinsic Parameters

- For the measured pixel coordinates **q** in Camera 1 with intrinsic parameters K_1, the relationship between the calibrated image plane location and pixel coordinates is
  $${}^1\mathbf{p} = (K_1^{-1})\,\mathbf{q}$$
- Likewise, for Camera 2 with intrinsic parameters K_2, we have
  $${}^2\mathbf{p}' = (K_2^{-1})\,\mathbf{q}'$$

The Fundamental Matrix

- The epipolar constraint is given by: ${}^1\mathbf{p}^T\mathbf{E}\,{}^2\mathbf{p}' = 0$ with $\mathbf{E} = [({}^1\mathbf{t}_2)_\times]\,{}^1_2\mathbf{R}$ where ${}^1\mathbf{p}$ and ${}^2\mathbf{p}'$ are calibrated coordinates in the two images.
- The relationship between the calibrated coordinates (${}^1\mathbf{p}, {}^2\mathbf{p}'$) and image coordinates (**q, q'**) in homogeneous coordinates can be expressed as ${}^1\mathbf{p} = (\mathbf{K}_1^{-1})\,\mathbf{q}$ and ${}^2\mathbf{p}' = (\mathbf{K}_2^{-1})\,\mathbf{q}'$
- Therefore, we can express the epipolar constraint as:

$$\begin{aligned} {}^1\mathbf{p}^T\mathbf{E}\,{}^2\mathbf{p}' = 0 &= \\ (K_1^{-1}\mathbf{q})^T\mathbf{E}(K_2^{-1}\mathbf{q}') &= \\ \mathbf{q}^T((K_1^{-1})^T\mathbf{E}K_2^{-1})\mathbf{q}' &= \\ \mathbf{q}^T\mathbf{F}\mathbf{q}' &= 0 \end{aligned}$$

Note: $(AB)^T=B^TA^T$

- $\mathbf{F} = (\mathbf{K}_1^{-1})^T\mathbf{E}\mathbf{K}_2^{-1}$ is a 3x3 matrix called the Fundamental Matrix.
- We can solve for **F** directly from pixel coordinates with 8 matches across a pair of images with the 8 point algorithm.
- The cameras do not need to be calibrated. $\mathbf{K}_1$ or $\mathbf{K}_2$ are not needed.

Epipolar Constraint for Calibrated and Uncalibrated Cameras

- Calibrated cameras: ${}^{1}\mathbf{p}^{T}\mathbf{E}\,{}^{2}\mathbf{p}' = 0$
  - E is the essential matrix
  - ${}^{1}\mathbf{p} = (\mathbf{K}_1^{-1})\,\mathbf{q}$, ${}^{2}\mathbf{p}' = (\mathbf{K}_2^{-1})\,\mathbf{q}'$
    where $\mathbf{K_1}, \mathbf{K_2}$ are the intrinsic parameters of the two cameras and are determined by calibration.

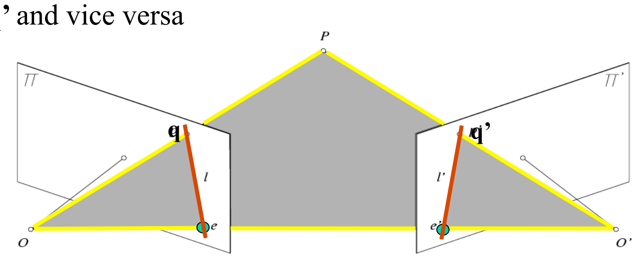
- Uncalibrated cameras: $\mathbf{q}^{T}\mathbf{F}\mathbf{q}' = 0$
  - **F** is the Fundamental matrix
  - **q** and **q'** are pixel coordinates

- Note similarity of equations.
- The fundamental matrix can be estimated directly from pixel coordinates using the 8 point algorithm.

Epipolar constraint for Uncalibrated Cameras

$$\boxed{\mathbf{q}^{T}\mathbf{F}\mathbf{q}' = \mathbf{0}}$$

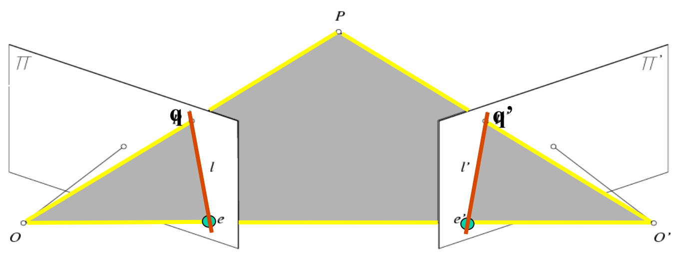
1. The epipolar constraint is homogenous in **q, q'** and **F**
2. It is bilinear in **q** and **q'**. E.g., for a given value of **q**, it is linear in **q'** and vice versa



3. Given pixel coordinates **q'** in Π', the equation of the epipolar line l in Π is $\mathbf{a}^{T}\mathbf{q} = 0$ where $\mathbf{a}=\mathbf{F}\mathbf{q}'$
4. Given pixel coordinates **q** in Π, the equation of the epipolar line l' in Π' is $\mathbf{b}^{T}\mathbf{q}' = 0$ where $\mathbf{b}=\mathbf{F}^{T}\mathbf{q}$

Epipolar constraint for Uncalibrated Cameras

$\boxed{\mathbf{q}^T\mathbf{F}\mathbf{q'} = 0}$



5. The eigenvector of **F** corresponding to the zero eigenvalue is the epipole **e'** in homogeneous coordinates.
6. The eigenvector of $\mathbf{F}^T$ corresponding to the zero eigenvalue is the epipole **e** in homogeneous coordinates.
7. **F** is singular (determinant is zero & can't be inverted)
8. **F** can be estimated from 8 corresponding points using the 8 point algorithm

But wait: How do I use this in the homework

3. Given pixel coordinates **q'** in Π' expressed in homogeneous coordinates, the equation of the epipolar line l in Π is $\mathbf{a}^T\mathbf{q} = 0$ where $\mathbf{a}=\mathbf{F}\mathbf{q'}$
   - **q'** is in homogenous coordaintes, and we can write **q'**=[u', v', 1] where [u', v'] are the measured inhomogeneous pixel coordinates.
   - **q** are homogeneous coordinates and can be written as **q**=[u,v,1] where [u, '] are the inhomogeneous pixel coordinates in Π.

5. The eigenvector of **F** corresponding to the zero eigenvalue is the epipole **e'** in homogeneous coordinates.
   - When computing **F** using 8 point algorithm, there is no guarantee that determinant of **F** will be zero and have zero Eigenvalue. Instead, use Eigenvector corresponding to smallest Eigenvalue.
   - The Eigenvectors of **F** will be in homogeneous coordinates
   - Convert to Euclidean coordinates

Uncalibrated stereo

- Epipolar geometry can be determined without calibration
- Images can be rectified so epipolar lines are rows of the rectified image.
- Matching can proceed in the same way
- But you need calibration to estimate depth.
  - If you arbitrarily make up intrinsic and extrinsic parameters and estimate depth. The estimated 3D point locations ${}^{e}\mathbf{p}$ and true locations ${}^{t}\mathbf{p}$ in homogeneous coordinates will differ by a linear transformation

    ${}^{e}\mathbf{p} = \mathbf{A}\,{}^{t}\mathbf{p}$

HW2

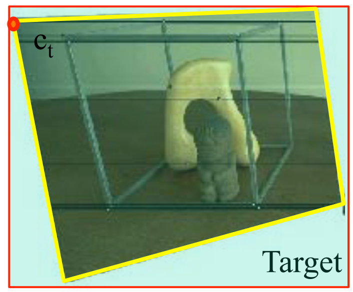
- Depth estimation equation
- Epipolar geometry
  - Code is given for computing Essential/Fundamental matrix.
  - You will
    - Draw epipolar lines
    - Find epipoles
- Rectification: You're given code for computing H_L and H_R. You need to actually rectify the images.
- Matching:
  - You will do sparse matching using feature points
  - You will improve matching using epipolar geometry
- Structure from Motion using RANSAC

How is warping done? Forward Method

- Input: Source image: I and
  Rectification matrix H
- For each corner c_s of Source image in homogenous coordinates, compute $c_t = Hc_s$
- Compute smallest and largest x and y of c_t's. Determine bounding box on target image. Create target image T with size of bounding box.
- For each pixel with coordinate p_s (homogenous) in the Source image, compute location in the Target image as $p_t = Hp_s$. Copy $I(p_s)$ to $T(p_t)$

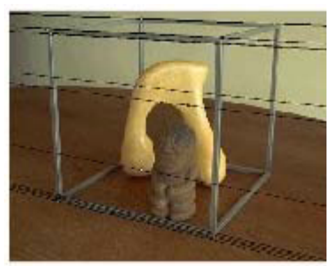
c_s

Source

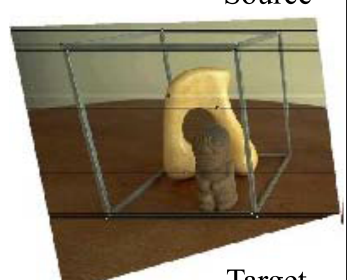


Problem with Forward Method

- There's no guarantee that every pixel in Target Image will be written to.
- If Target Image is larger than Source or Target is highly stretched, there may be missing points that appear as speckles or lines.



Source



Target

How is warping done? Backward method

- Input: Source image: I and rectification matrix H
- For each corner c_s of Source image in homogenous coordinates, compute $c_t = Hc_s$
- Compute smallest and largest x and y of c_t's, determine bounding box on target image, create target image T with size of bounding box.
- For each pixel coordinate p_t (homogenous) in the Target, compute location in the Source as $p_s = H^{-1}p_t$.
  - If p_s is within source image, copy $I(p_s)$ to $T(p_t)$

Source

Target

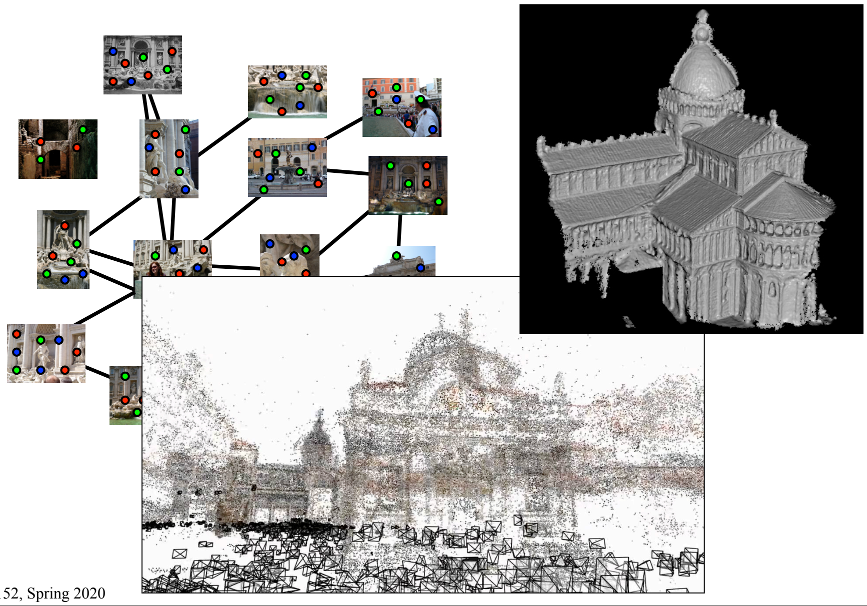
Structure from Motion

Also called

Visual SLAM or VSLAM

(Simultaneous Localization and Mapping)

Estimate 3D structure from images



Web page

- http://www.cs.cornell.edu/projects/bigsfm/

Structure-from-Motion (SFM)

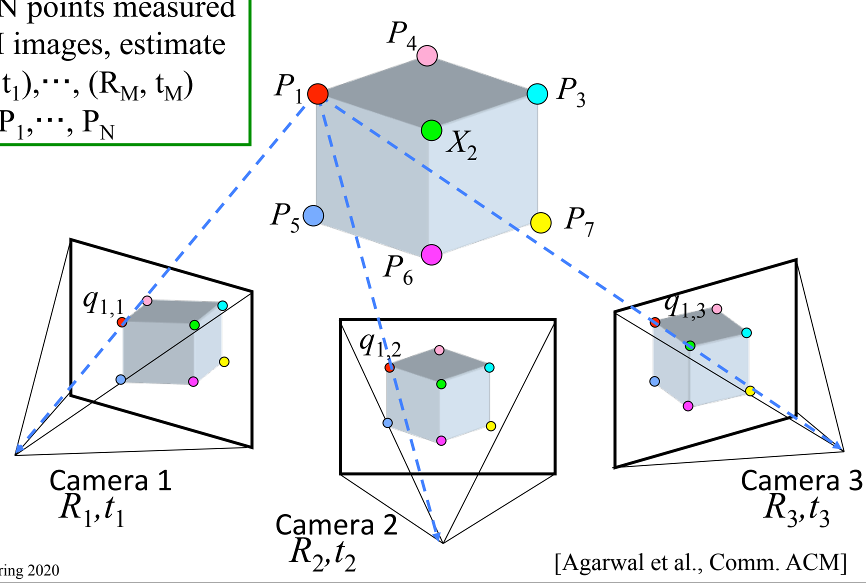
Given two or more images or video without any information on camera position/motion as input, estimate camera motion and 3-D structure of a scene.

Two Approaches

1. **Discrete motion (wide baseline)**
2. Continuous (Infinitesimal) motion usually from video

Structure from motion

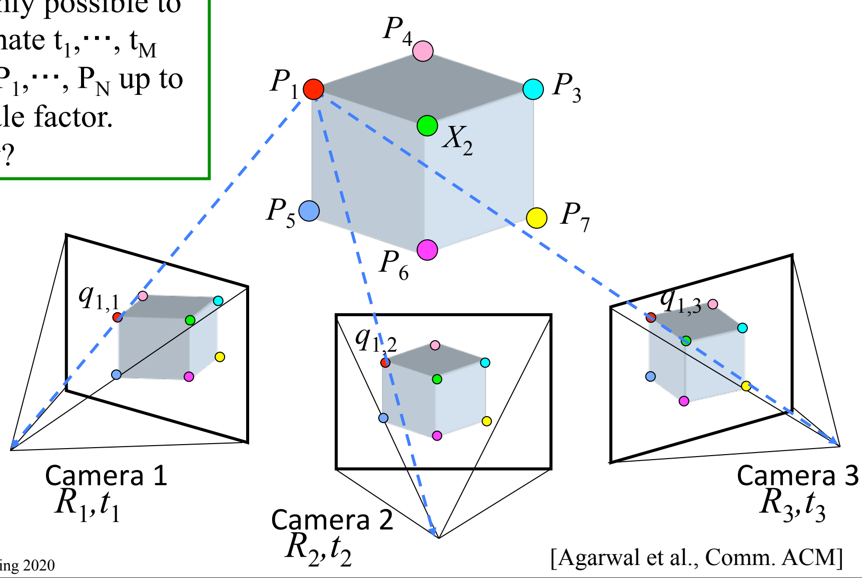
For N points measured in M images, estimate $(R_1, t_1), \cdots, (R_M, t_M)$ and $P_1, \cdots, P_N$



[Agarwal et al., Comm. ACM]

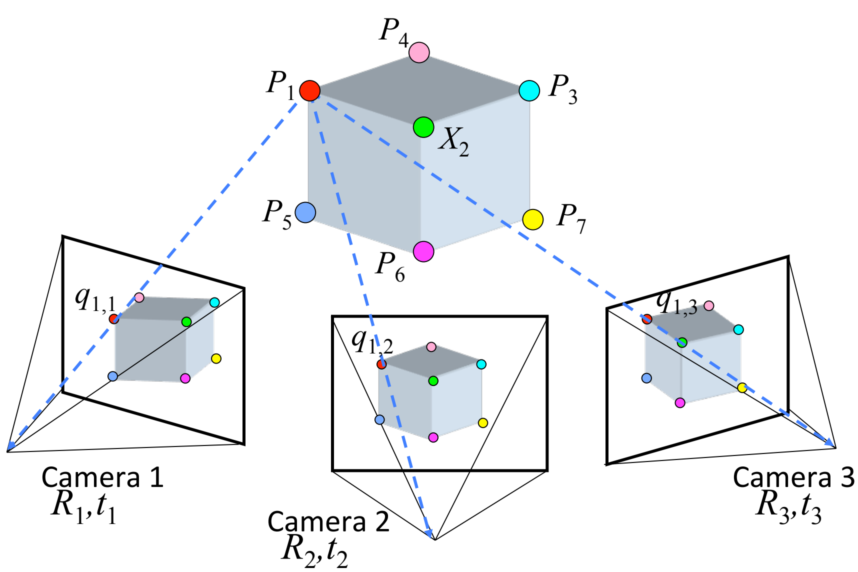
Structure from motion

Its only possible to estimate $t_1, \cdots, t_M$ and $P_1, \cdots, P_N$ up to a scale factor. Why?



[Agarwal et al., Comm. ACM]

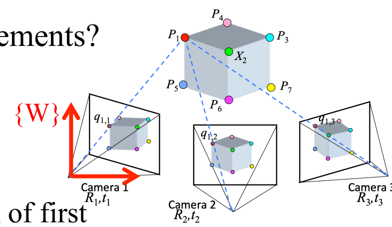
Structure from motion



How many views and how many points are needed to solve this?

M images of *N* points. How many measurements?

2*M*N



How many unknowns?

1. 3-D Structure: 3*N unknowns for points
2. Affix world coordinate system to location of first camera frame: (M-1)*6 unknowns for cameras
3. Can only recover structure and motion up to scale factor (one fewer unknown)

Total number of unknowns: (M-1)*6+3*N-1

Solution is possible when more measurements than unknowns:

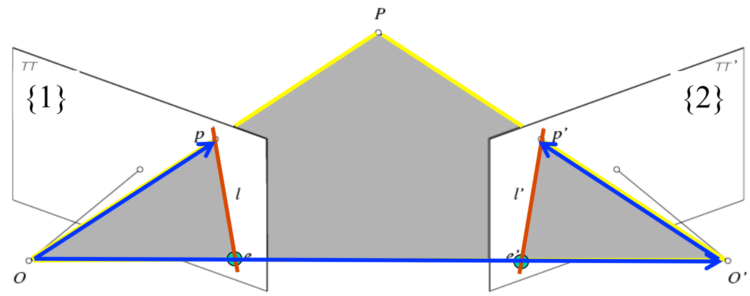
(M-1)*6+3*N-1 ≤ 2*M*N

Some values of N and M satisfying this: M=2, N = 5

M=3, N = 4

Two view structure from motion

Epipolar Constraint: Calibrated Case



The vectors $\overrightarrow{Op}$, $\overrightarrow{OO'}$ and $\overrightarrow{O'p'}$ are coplanar

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0 \Rightarrow {}^{1}\mathbf{p} \cdot \left[{}^{1}\mathbf{t}_2 \times ({}^{1}_{2}\mathbf{R}\, {}^{2}\mathbf{p}') \right] = 0$$

$$\Downarrow$$

Essential Matrix
(Longuet-Higgins, 1981)

$${}^{1}\mathbf{p}^T \mathbf{E}\, {}^{2}\mathbf{p}' = 0 \text{ with } \mathbf{E} = [({}^{1}\mathbf{t}_2)_\times]\, {}^{1}_{2}\mathbf{R}$$

skew

The Eight-Point Algorithm (Longuet-Higgins, 1981)

Input: 8 corresponding points in two images ${}^{1}p_i = K_1^{-1} q_i$, ${}^{2}p_i' = K_2^{-1} q_i'$

$${}^{1}\mathbf{p}^T \mathbf{E}\, {}^{2}\mathbf{p}' = 0 \text{ with } \mathbf{E} = [({}^{1}\mathbf{t}_2)_\times]\, {}^{1}_{2}\mathbf{R}$$

$$[u, v, 1] \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} uu' & uv' & u & vu' & vv' & v & u' & v' & 1 \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{31} \\ E_{32} \\ E_{33} \end{bmatrix} = 0$$

- Set E_{33} to 1
- Use 8 points (u_i, v_i), i=1..8

$$\begin{bmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{31} \\ E_{32} \end{bmatrix} = - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Solve for E_{11} to E_{32}
  These are elements of the Essential Matrix
- Then solve for **R, t**
  $\mathbf{E} = [({}^{1}\mathbf{t}_2)_\times]\, {}^{1}_{2}\mathbf{R}$

Sketch of Two View SFM Algorithm

Input: Two images

1. Detect feature points in each image
2. Using intrinsic parameters from calibration, compute

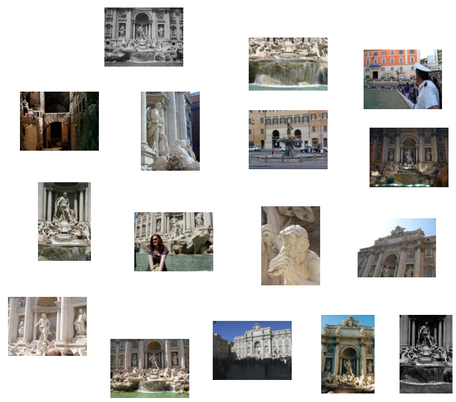
$$p_{i,j} = K_j^{-1} q_{i,j}$$

1. Find 8 matching feature points (easier said than done)
2. Compute the Essential Matrix E using 8-point Algorithm
3. Compute R and T (recall that E=SR where S is skew symmetric matrix)
4. Perform stereo matching using recovered epipolar geometry expressed via E.
5. Reconstruct 3-D geometry of corresponding points.

N-view structure from motion

Feature detection

Several images observe a scene from different viewpoints



CSE152, Spring 2020

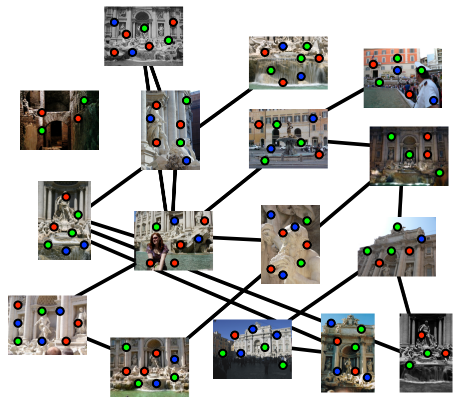
Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004]

CSE152, Spring 2020

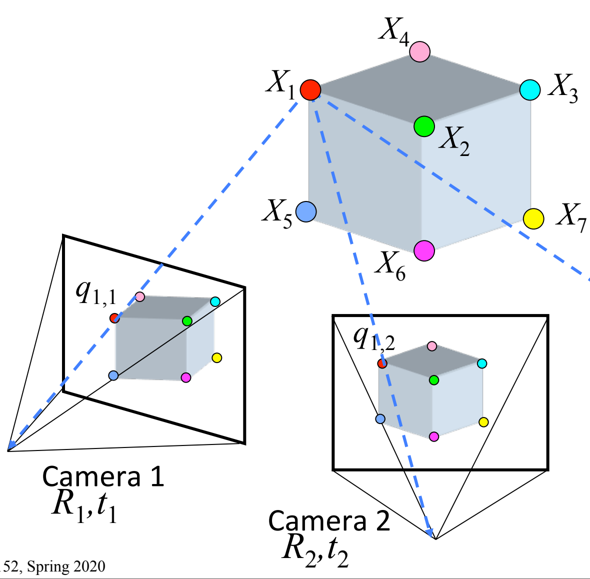
Feature matching

Match features between each pair of images



CSE152, Spring 2020

Structure from motion



minimize

$$g(\mathbf{R}, \mathbf{T}, \mathbf{X})$$

non-linear least squares

CSE152, Spring 2020

Bundle adjustment

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

indicator variable:
1: If point *i* visible in image *j*
0: Otherwise

predicted image location

observed image location

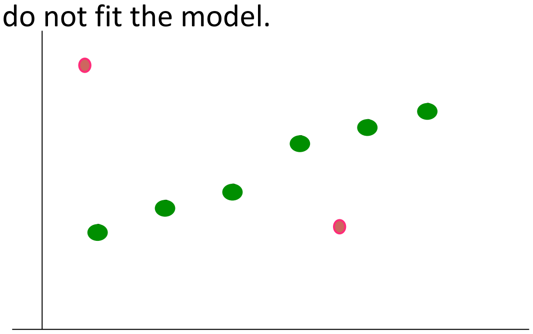
  - Optimized with non-linear least squares
  - Levenberg-Marquardt is a popular choice
- Practical challenges?
  - Initialization (Bootstrap from 2 View Solution between pairs)
  - Outliers

Inliers vs. Outliers

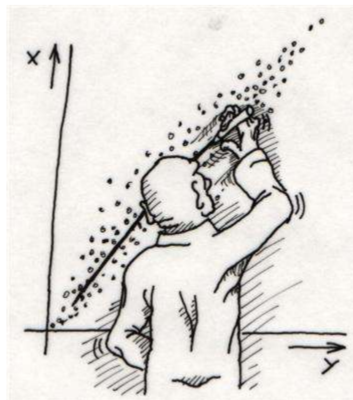
- As you saw in HW1, not every match was correct when using SIFT descriptors.
- Squared errors metrics like $\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$ highly penalize mismatches because they get squared.
- Inliers: Given a model with some assumed distribution, inliers are data points that fit the model.
- Outliers are points that do not fit the model.
- Example: line fitting

Inliers:

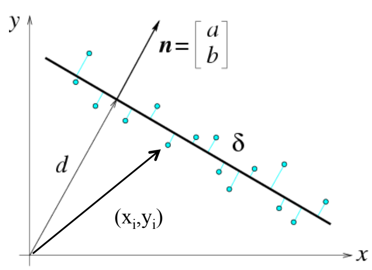
Outliers:



Line Fitting



Line Fitting



Given n points (x_i, y_i), estimate parameters of line (a,b,d)

$$ax_i + by_i - d = 0$$

subject to the constraint that

$$a^2 + b^2 = 1$$

Note: $ax_i + by_i - d$ *is distance from* (x_i, y_i) to line.

Problem: minimize

$$E(a,b,d) = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

Cost Function:
Sum of squared distances
between each point and the line

with respect to (a,b,d).

1. Minimize E with respect to d:

$$\frac{\partial E}{\partial d} = 0 \Rightarrow d = \frac{1}{n}\sum_{i=1}^{n} ax_i + by_i = a\bar{x} + b\bar{y}$$

Where $(\bar{x}, \bar{y})$ is the mean of the data points

Line fitting cont.

2. Substitute d back into E

$$E = \sum_{i=1}^{n} [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = |\mathcal{U}\boldsymbol{n}|^2 \qquad \text{where} \quad \mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

where $\mathbf{n}=(a\ b)^T$.

3. Minimize $E=|Un|^2=\boldsymbol{n}^T U^T U\boldsymbol{n}=\boldsymbol{n}^T S\boldsymbol{n}$ with respect to *a, b* subject to the constraint $\mathbf{n}^T\mathbf{n} = 1$. Note that S is given by

$$S = \mathcal{U}^T\mathcal{U} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 & \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} \\ \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} & \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 \end{pmatrix}$$

And it's a real, symmetric, positive definite

Line Fitting – Finished

4. This is a constrained optimization problem in $\boldsymbol{n}$. Solve with Lagrange multiplier

$$L(\boldsymbol{n}) = \mathbf{n}^T S\mathbf{n} - \lambda(\mathbf{n}^T\mathbf{n} - 1)$$

Take partial derivative (gradient) w.r.t. **n** and set to 0.

$$\nabla \mathbf{L} = \mathbf{2}S\mathbf{n} - \mathbf{2}\lambda\mathbf{n} = 0$$

or

$$S\mathbf{n} = \lambda\mathbf{n}$$

$\boldsymbol{n}=(a,b)$ is an Eigenvector of the symmetric matrix S (the one corresponding to the smallest Eigenvalue).

5. d is computed from Step 1.

RANdom Sample Consensus
RANSAC

Slides adapted from
Frank Dellaert and Marc Pollefeys