CSE 152A: Computer Vision
Manmohan Chandraker

Lecture 8: Two-View Reconstruction
Overall goals for the course

• Introduce fundamental concepts in computer vision

• Enable one or all of several such outcomes
  – Pursue higher studies in computer vision
  – Join industry to do cutting-edge work in computer vision
  – Gain appreciation of modern computer vision technologies

• Engage in discussions and interaction

• This is a great time to study computer vision!
Course details

• Class webpage:
  – http://cseweb.ucsd.edu/~mkchandraker/classes/CSE152A/Fall2022/

• TAs
  – Meng Song, Mallikarjun Swamy, Rishi Chandrasekaran, Vishal Vinod:
    {mes050, mswamy, r3chandr, vvinod}@ucsd.edu

• Tutors
  – Nick Chua, Navya Sharma, Ang Li: {nchua, n1sharma, a3li}@ucsd.edu

• Discussion section: M 3-3:50pm

• Office hours posted on course calendar

• Piazza: https://piazza.com/ucsd/fall2022/cse152a/
Self-Study Assignment

• Pick a technology area primarily driven by computer vision
  – Can pick one of these suggestions, or use anything else that you like

• **Virtual Reality**
  – Meta Quest Pro
  – Oculus Rift

• **Augmented Reality**
  – Microsoft Hololens
  – Magic Leap 2

• **Self-Driving**
  – Waymo
  – Tesla

• **Content Creation**
  – Adobe Photoshop
  – OpenAI Dall-E

• **Cloud Services**
  – Amazon Rekognition
  – Microsoft Azure Cognitive Services

• **Sports**
  – Hawk-Eye
  – Gameface.ai

• **Face Recognition**
  – Face++
  – Apple FaceID

• **Robotics**
  – Boston Dynamics
  – iRobot Roomba

• **Space Exploration**
  – James Webb Telescope
  – Mars Rover

• **Social Media**
  – Snap
  – Instagram
Self-Study Assignment

• Pick a technology area primarily driven by computer vision
  – Can pick one of these suggestions, or use anything else that you like

• Write a 1 page essay (single-spaced)
  – Can be longer (hopefully not too long)
  – Great if you include pictures (with citations)

• Example prompts (feel free to add to these or use your own):
  – How does computer vision overcome barriers or solve user needs in this technology?
  – Can you identify where knowledge of photometric or geometric image formation is used?
  – Can you identify where such knowledge does not suffice and machine learning is used?
  – Can you identify where photometric or geometric models are used along with learning?
  – What was possible in this area 10 years ago and how did computer vision advance it?
  – How do you anticipate technology in this chosen area will advance in the next 10 years?

• Due date: Nov 11, 2022
Mid-Term Logistics

• In-class, on Nov 2, during 5 – 6:15pm PST
• “Paper-and-pen” exam
  – Bring your own pen, bring scratch paper if needed
  – Calculators should not be needed, can bring one just in case
• Write clearly and legibly to ensure correct grading
  – Show intermediate steps for partial grades

• Exam will test understanding of lectures
  – Some direct questions, some need applying concepts to new situations
  – Mimics real-life practice: know concepts (details can always be looked up)

• Open notes
  – Can refer to any material on your mobile device or computer
  – Can bring any printed or hand-written material
  – Any reference books or sources are fine
  – Only restriction: no communication among students
Recap
Photometric: How light is recorded

Light emitted

Illumination (energy) source

Light reflected to camera

Scene element

Imaging system

(Internal) image plane

Light recorded by sensor
Projection matrix

$$\mathbf{q} = (x, y, z, 1)$$

(3D point in homogeneous coordinates, a vector of length 4)

$$\Pi = K \begin{bmatrix} R & -Rc \end{bmatrix}$$

(2D image in homogeneous coordinates, a vector of length 3)

$$\left( \frac{(\Pi \mathbf{q})_1}{(\Pi \mathbf{q})_3}, \frac{(\Pi \mathbf{q})_2}{(\Pi \mathbf{q})_3} \right)$$

(Usual Cartesian coordinates, a vector of length 2)
Detection of features

• Examine a small window over an image

The wiggly arrows indicate graphically a directional response in the detector as it moves in the three areas shown.
Edge Detection with Image Gradients

- Gradient represents direction of most rapid change in intensity
  \[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, & 0 \end{bmatrix} \]
  \[ \nabla f = \begin{bmatrix} 0, & \frac{\partial f}{\partial y} \end{bmatrix} \]

- The gradient encodes edge strength and edge direction as
  \[
  \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \quad \theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)
  \]

- Can efficiently compute gradient using convolutions
  \[
  K_x = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad K_y = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}
  \]

- Sobel operator is often used in practice
  \[
  K_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad K_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}
  \]
Harris Corner Detector

First, consider the second moment matrix for a simpler case:

\[
C = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2 
\end{bmatrix} = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}
\]

This means dominant gradient directions align with x or y axis.

In the general case, since C is symmetric, it can be shown:

\[
C = Q^{-1} \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix} Q
\]

If either \( \lambda \) close to 0, then not a corner, so seek locations where both large.

CSE 152A, FA22: Manmohan Chandraker

Slide based on: David Jacobs
Simple Corner Detector: Implementation

- Run a small window over an image and compute spatial gradient matrix $C$ at every pixel
- Compute the minor eigenvalue of $C$ at every pixel to obtain the corner response “image” $R$
- Apply nonmaximal suppression to the “image” $R$
  - Divide into grid, choose maximum within each grid cell
  - Resulting image $R'$ has only one corner candidate per grid cell
  - Prevents corners from being too close to each other
- Threshold resulting image $R'$ using a global threshold $T$
  - Corners at pixels $(x, y)$ corresponding to $R'(x, y) > T$
Simple matching methods

Interest point:
• Localized position
• Informative about image content
• Repeatable under variations

Descriptor:
• Function applied on $W_1$ and $W_2$, to enable comparing them
• Simple descriptor: can just use the window $W$ around interest point

$W_1(x, y): k \times k$ pixel patch in image 1

$W_2(x, y): k \times k$ pixel patch in image 2
Simple matching methods

- **SSD (Sum of Squared Differences)**
  \[ \sum_{x,y} |W_1(x,y) - W_2(x,y)|^2 \]

- **NCC (Normalized Cross Correlation)**
  \[ \sum_{x,y} \frac{(W_1(x,y) - \bar{W}_1)(W_2(x,y) - \bar{W}_2)}{\sigma_{W_1} \sigma_{W_2}} \]
  \[ \bar{W}_i = \frac{1}{n} \sum_{x,y} W_i, \quad \sigma_{W_i} = \sqrt{\frac{1}{n} \sum_{x,y} (W_i - \bar{W}_i)^2} \]
  (Mean) \quad (Standard deviation)

- What advantages might NCC have over SSD?
Feature distance

How to define the distance between two features $f_1$, $f_2$?

- Better approach: ratio distance = $\frac{\text{SSD}(f_1, f_2)}{\text{SSD}(f_1, f_2')}$
  - $f_2$ is best SSD match to $f_1$ in $I_2$
  - $f_2'$ is 2nd best SSD match to $f_1$ in $I_2$
  - gives small values for ambiguous matches
True or false positives

The distance threshold affects performance

- **True positives** = number of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
  - Increase threshold (uncertain matches are also allowed)

- **False positives** = number of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?
  - Decrease threshold (matches discarded unless they are very certain)
Correspondence is a vital 3D cue
Depth from correspondence

Two measurements: $X_L, X_R$
Two unknowns: $X, Z$

Constants:
- Baseline: $d$
- Focal length: $f$

Disparity: $(X_L - X_R)$

\[
Z = \frac{d f}{(X_L - X_R)}
\]

\[
X = \frac{d X_L}{(X_L - X_R)}
\]

Using similar triangles:
\[
\frac{X_L}{f} = \frac{X}{Z} \quad \frac{X_R}{f} = \frac{X - d}{Z}
\]

Depth is inversely proportional to disparity

(Adapted from Hager)
Mars Exploratory Rovers: Spirit and Opportunity, 2004

Stereo camera
Structure from Motion (SFM)
Visual SLAM
Structure from Motion
Feature detection

Several images observe a scene from different viewpoints
Feature detection

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

Match features between each pair of images
Structure from motion

Optimization problem:
minimize\[ g(R, T, X) \]

non-linear least squares

\[ \Pi_1 X_1 \sim p_{11} \]

Camera 1
\[ R_1, t_1 \]

Camera 2
\[ R_2, t_2 \]

Camera 3
\[ R_3, t_3 \]
Feature matching
Robustness

Let’s consider a simpler example... line fitting

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
  – These are 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

CSE 152A, FA22: Manmohan Chandraker
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
  – RANSAC: Random Sample Consensus

• Number of hypotheses depends on
  – Outlier ratio
  – Probability of correct answer
  – Model size
RANSAC

- General version:
  1. Randomly choose $s$ samples
     - Typically $s = \text{minimum sample size to fit a model}$
  2. Fit a model (say, line) to those samples
  3. Count the number of inliers that approximately fit the model
  4. Repeat $N$ times
  5. Choose the model with the largest set of inliers
Two-View Reconstruction

Given correspondences with potential outliers, fit a model to find maximum number of inliers, consistent with geometric image formation in the two views.
Model: Fundamental Matrix

\[ x_1 \leftrightarrow x_2 \]

\[ x_1^T F x_2 = 0 \]

- \( F \) is a 3x3 matrix of rank 2
- \( F \) has 7 degrees of freedom
Estimating $F$

• Given just the two images, can we estimate $F$?

• Yes, with enough correspondences.
Estimating F: 8-point algorithm

• The fundamental matrix $F$ is defined by

$$x'^T F x = 0$$

for any pair of matches $x$ and $x'$ in two images.

• Let $x = (u,v,1)^T$ and $x' = (u',v',1)^T$, \[ F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \]

• Each match gives a linear equation:

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$
8-point algorithm

Given \( n \) point correspondences, set up a system of equations:

\[
\begin{bmatrix}
u_1 u'_1 & v_1 u'_1 & u'_1 & u_1 v'_1 & v_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\
u_2 u'_2 & v_2 u'_2 & u'_2 & u_2 v'_2 & v_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_n u'_n & v_n u'_n & u'_n & u_n v'_n & v_n v'_n & v'_n & u_n & v_n & 1
\end{bmatrix}
\begin{bmatrix}f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33}\end{bmatrix} = 0
\]

- In reality, instead of solving \( A\mathbf{f} = 0 \), we seek \( \mathbf{f} \) to minimize \( \| A\mathbf{f} \| \).
Solving homogeneous systems

• In reality, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$.

• Singular value decomposition:

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

$\mathbf{U}$, $\mathbf{V}$ are rotation matrices

$$\Sigma = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix}$$

• Solution $\mathbf{f}$ given by the last column of $\mathbf{V}$. 

CSE 152A, FA22: Manmohan Chandraker
8-point algorithm: Problem?

• $F$ should have rank 2
• To enforce that $F$ is of rank 2, $F$ is replaced by $F'$ that minimizes $\|F^T F'\|$ subject to the rank constraint.

• This is achieved by SVD. Let $F = U\Sigma V^T$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}. \text{ Let } \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

then $F' = U\Sigma' V^T$ is the solution.
8-point algorithm

% Normalization on 2D points (advanced concept, implemented for you)

% Build the constraint matrix
A = [x2(1,:)' .* x1(1,:)'   x2(1,:)' .* x1(2,:)'   x2(1,:)' ... 
   x2(2,:)' .* x1(1,:)'   x2(2,:)' .* x1(2,:)'   x2(2,:)' ... 
   x1(1,:)'   x1(2,:)'   ones(npts,1) ];

[U,D,V] = svd(A);

% Extract fundamental matrix from the column of V
% corresponding to the smallest singular value.
F = reshape(V(:,9),3,3)';

% Enforce rank 2 constraint
[U,D,V] = svd(F);
F = U * diag([D(1,1) D(2,2) 0]) * V';

% Do the reverse normalization on 2D points
RANSAC to Estimate Fundamental Matrix

• For $N$ times
  – Pick 8 points
  – Compute a solution for $\mathbf{F}$ using these 8 points
  – Count number of inliers with $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2$ close to 0

• Pick the one with the largest number of inliers
Outliers in Feature Matching

Overall plan: use the fundamental matrix as a model to remove outliers
- Points in correspondence should be consistent with some fundamental matrix
- Find the fundamental matrix with which most points are consistent (inliers)
- Remove points not consistent with the above fundamental matrix (outliers)
Fundamental Matrix for SFM
Cross-product as linear operator

**Useful fact:** Cross product with a vector $\mathbf{t}$ can be represented as multiplication with a *(skew-symmetric)* 3x3 matrix

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$$\mathbf{t} \times \mathbf{\hat{p}} = [\mathbf{t}]_\times \mathbf{\hat{p}}$$

What is the rank of $[\mathbf{t}]_\times$?
Cross-product as linear operator

**Useful fact:** Cross product with a vector \( \mathbf{t} \) can be represented as multiplication with a \((skew-symmetric)\) 3x3 matrix

\[
[t]_\times = \begin{bmatrix}
0 & -t_z & t_y \\
t_z & 0 & -t_x \\
-t_y & t_x & 0
\end{bmatrix}
\]

\( \mathbf{t} \times \tilde{\mathbf{p}} = [\mathbf{t}]_\times \tilde{\mathbf{p}} \)

**What is the rank of \([\mathbf{t}]_\times\)?**

Rank 2, since \( \mathbf{t} \) is a null vector of \([\mathbf{t}]_\times\)

\([\mathbf{t}]_\times \mathbf{t} = \mathbf{t} \times \mathbf{t} = \mathbf{0} \)

CSE 152A, FA22: Manmohan Chandraker
Two-view geometry

Corresponding point in other image is constrained to lie on a line, called the *epipolar line*.
Epipoles

Two special points: $e_1$ and $e_2$ (the *epipoles*): projection of one camera into the other.
Epipolar lines

Two special points: $e_1$ and $e_2$ (the *epipoles*): projection of one camera into the other.

All epipolar planes pass through the two camera centers.
All of the epipolar lines in an image pass through the epipole.
Essential matrix

- Assume calibrated cameras with $K_1 = K_2 = I_{3x3}$.
- Let camera 1 be $[I, 0]$ and camera 2 be $[R, t]$.
- In camera 1 coordinates, 3D point $X$ is given by $X_1 = \lambda_1 p$.
- In camera 2 coordinates, 3D point $X$ is given by $X_2 = \lambda_2 q$.
- Since camera 2 is related to camera 1 by rigid-body motion $[R, t]$

$$X_2 = RX_1 + t$$

$$\lambda_2 q = \lambda_1 R p + t$$
Essential matrix

- We have: $\lambda_2 q = \lambda_1 R p + t$
- Take cross-product with respect to $t$:
  $$\lambda_2 [t] \times q = \lambda_1 [t] \times R p$$
- Take dot-product with respect to $q$:
  $$0 = \lambda_1 q^\top [t] \times R p$$
Essential matrix

- We have: \( q^\top [t] \times R p = 0 \)
- Define:
  \[
  E = [t] \times R 
  \]
- Then, we have:
  \[
  q^\top E p = 0 
  \]

How many degrees of freedom does \( E \) have?
• Relax the assumption of calibrated cameras.
• Then, \( \mathbf{p} \) and \( \mathbf{q} \) are in metric coordinates and pixel counterparts are:
  \[
  \mathbf{p}' = \mathbf{K}_1 \mathbf{p} \quad \mathbf{q}' = \mathbf{K}_2 \mathbf{q}
  \]
• Recall essential matrix constraint:
  \[
  \mathbf{q}^\top \mathbf{E} \mathbf{p} = 0
  \]
• Substituting, we have:
  \[
  (\mathbf{K}_2^{-1} \mathbf{q}')^\top \mathbf{E} (\mathbf{K}_1^{-1} \mathbf{p}') = 0
  \]
Essential matrix constraint in pixel space: \((K_2^{-1}q')^\top E(K_1^{-1}p') = 0\).

Rearranging:

\[ q'^\top (K_2^{-\top}EK_1^{-1})p' = 0 \]

Define: \( F = K_2^{-\top}EK_1^{-1} \)

Then, we have:

\[ q'^\top Fp' = 0 \]

How many degrees of freedom does \( F \) have?
• For corresponding points $x$ and $x'$, we have $x'^{T}Fx = 0$
• Define $l' = Fx$, then we have $x'^{T}l' = 0$
• Then, for point $x$, the line $Fx$ contains corresponding point $x'$
• So, $l' = Fx$ is the epipolar line in the second image
• For corresponding points \( x \) and \( x' \), we have \( x'^{\top}F x = 0 \)
• We saw that \( l' = F x \) is the epipolar line in the second image

**Epipole:** the point that lies on *all* epipolar lines \( l' \) for *any* \( x \)
• Thus, for any \( x \), we need point \( e' \), such that \( e'^{\top}F x = 0 \)
• Rewrite as \( (F^{\top}e')^{\top}x = 0 \)
• So, epipole is given by \( e' = \text{null}(F^{\top}) \)
For corresponding points \( \mathbf{x} \) and \( \mathbf{x}' \), we have \( \mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \)

- Taking transpose, it is the same as \( \mathbf{x}^T \mathbf{F}^T \mathbf{x}' = 0 \)
- Define \( \mathbf{l} = \mathbf{F}^T \mathbf{x}' \), then we have \( \mathbf{x}^T \mathbf{l} = 0 \)
- Then, for point \( \mathbf{x}' \), the line \( \mathbf{F}^T \mathbf{x}' \) contains corresponding point \( \mathbf{x} \)
- So, \( \mathbf{l} = \mathbf{F}^T \mathbf{x}' \) is the epipolar line in the first image
For corresponding points \( x \) and \( x' \), we have \( x'^T F x = 0 \).

We saw that \( l = F^T x' \) is the epipolar line in the second image.

**Epipole**: the point that lies on *all* epipolar lines \( l \) for *any* \( x' \).

Thus, for any \( x' \), we need point \( e \), such that \( x'^T F e = 0 \).

Group the elements as \( x'^T (F e) = 0 \).

So, epipole is given by \( e = \text{null}(F) \).
Properties of the fundamental matrix

- \( Fx \) is the epipolar line associated with \( x \)
- \( F^T x' \) is the epipolar line associated with \( x' \)
- Epipoles given by \( Fe = 0 \) and \( F^T e' = 0 \)
- \( F \) is rank 2.
Motion from correspondences

- Use 8-point algorithm to estimate $F$
- Get $E$ from $F$:
  \[
  F = K_2^{-\top} E K_1^{-1}
  \]
  \[
  E = K_2^\top F K_1
  \]
- Decompose $E$ into skew-symmetric and rotation matrices:
  \[
  E = [t] \times R
  \]

Can estimate rotation and translation from $E$
Given camera motion $[R \mid t]$, can find intersection of back-projected rays from inlier correspondences to estimate the 3D points.
Results (ground truth)