Lecture 14:
3D, camera geometry, calibration

COS 429: Computer Vision

Slides adapted from: Szymon Rusinkiewicz, Jia Deng, Steve Seitz, David Fouhey
Our goal: Recovery of 3D structure

J. Vermeer, *Music Lesson*, 1662

Application: Single-view modeling

A. Criminisi, I. Reid, and A. Zisserman, Single View Metrology, IJCV 2000
2.5-D: estimating depth from single image

Inherent ambiguity

Source: S. Lazebnik
Pictorial Cues – Shading

[Figure from Prados & Faugeras 2006]
Pictorial Cues – Texture

Pictorial Cues – Perspective effects

Image credit: S. Seitz
Monitor: probably not 12 feet wide.

Desk surface: probably flat.

Source: D. Fouhey
Resolving Single-view Ambiguity

• Stereo: given 2 calibrated cameras in different views and correspondences, can solve for X

Original diagram credit: S. Lazebnik
Multi-view stereo

Many slides adapted from S. Seitz
Multi-view stereo or 3D photography

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape.
Multi-view stereo or 3D photography

• “Images of the same object or scene”
  • Arbitrary number of images (from two to thousands)
  • Arbitrary camera positions (camera network or video sequence)
  • Calibration may be initially unknown

• “Representation of 3D shape”
  • Depth maps
  • Meshes
  • Point clouds
  • Patch clouds
  • Volumetric models
  • Layered models
  • ...
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Single-view geometry
Our goal: Recovery of 3D structure

Source: S. Lazebnik
Review: Pinhole camera model

Source: S. Lazebnik
Review: Pinhole camera model

\[
(X, Y, Z) \mapsto (\frac{fX}{Z}, \frac{fY}{Z})
\]

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\mapsto
\begin{pmatrix}
fX \\
fY
\end{pmatrix}
= \begin{bmatrix}
f & 0 \\
f & 0 \\
1 & 0
\end{bmatrix}
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]

\[x = PX\]

Source: S. Lazebnik
Change #1: Principal point offset

Source: S. Lazebnik
Change #1: Principal point offset

We want the principal point to map to \((p_x, p_y)\) instead of \((0,0)\)

\[
(X, Y, Z) \mapsto (f X / Z + p_x, f Y / Z + p_y)
\]

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix} \mapsto 
\begin{pmatrix}
f X + Z p_x \\
f Y + Z p_y \\
Z
\end{pmatrix} = 
\begin{bmatrix}
f & p_x & 0 \\
f & p_y & 0 \\
1 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix}
\]

Source: S. Lazebnik
Change #1: Principal point offset

principal point: \((p_x, p_y)\)

\[
\begin{pmatrix}
 f X + Z p_x \\
 f Y + Z p_y \\
 Z
\end{pmatrix} =
\begin{bmatrix}
 f & p_x \\
 f & p_y \\
 1 & 1
\end{bmatrix}
\begin{bmatrix}
 1 & 0 \\
 1 & 0 \\
 1 & 1
\end{bmatrix}
\begin{pmatrix}
 X \\
 Y \\
 Z
\end{pmatrix}
\]

\[
K = \begin{bmatrix}
 f & p_x \\
 f & p_y \\
 1 & 1
\end{bmatrix}
\]

calibration matrix

\[
P = K[I \mid 0]
\]

Source: S. Lazebnik
Change #2: Pixel coordinates

Pixel size: \( \frac{1}{m_x} \times \frac{1}{m_y} \)

\[
K = \begin{bmatrix}
  m_x & m_y \\
  1 & 1
\end{bmatrix} \begin{bmatrix}
  f & p_x \\
  f & p_y
\end{bmatrix} = \begin{bmatrix}
  \alpha_x & \beta_x \\
  \alpha_y & \beta_y \\
  m & 1
\end{bmatrix}
\]

pixels/m

Source: S. Lazebnik
Change #3: Camera rotation and translation

- Conversion from world to camera coordinate system (in non-homogeneous coordinates):

\[ \tilde{X}_{\text{cam}} = R (\tilde{X} - \tilde{C}) \]

- \( \tilde{X}_{\text{cam}} \): coords. of point in camera frame
- \( \tilde{X} \): coords. of a point in world frame
- \( \tilde{C} \): coords. of camera center in world frame

Source: S. Lazebnik
Camera projection matrix

\[ \mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X} \]

\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda \\
\lambda
\end{bmatrix} = 
\begin{bmatrix}
* & * & * & * \\
* & * & * & * \\
* & * & * & * \\
* & * & * & *
\end{bmatrix} 
\begin{bmatrix} X \\
Y \\
Z \\
1
\end{bmatrix}
\]

\[ \mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \]
Camera parameters

\[ P = K [ R \ t ] \]

- **Intrinsic parameters**
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

\[
K = \begin{bmatrix}
  m_x & f & p_x \\
  m_y & f & p_y \\
  1 & 1 & 1
\end{bmatrix}
= \begin{bmatrix}
  \alpha_x & \beta_x \\
  \alpha_y & \beta_y \\
  1 & 1
\end{bmatrix}
\]

Source: S. Lazebnik
Camera parameters

\[ P = K[R \ t] \]

- **Intrinsic parameters**
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

- **Extrinsic parameters**
  - Rotation and translation relative to world coordinate system

**How many parameters here?**

Source: S. Lazebnik
Camera calibration basics
Camera calibration

\[
x = K \begin{bmatrix} R & t \end{bmatrix} X
\]

\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda
\end{bmatrix} = \begin{bmatrix}
* & * & * & * & * \\
* & * & * & * & * \\
* & * & * & * & *
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]
Camera Calibration

• Determining values for camera parameters
• Necessary for any algorithm that requires 3D ↔ 2D mapping
• Method used depends on:
  – What data is available
  – Intrinsics only vs. extrinsics only vs. both
  – Form of camera model
Camera Calibration

- General idea: place “calibration object” with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image
General camera model

- Projection matrix
- Don’t care about “z” after transformation

\[
\begin{pmatrix}
  a & b & c & d \\
  e & f & g & h \\
  i & j & k & l \\
\end{pmatrix}
\begin{pmatrix}
  x \\
  y \\
  z \\
  1 \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  ax + by + cz + d \\
  ix + jy + kz + l \\
  ex + fy + gz + h \\
  ix + jy + kz + l \\
\end{pmatrix}
\]

- Scale ambiguity $\rightarrow$ 11 free parameters
  - 6 extrinsic, 5 intrinsic
Camera Calibration – linear system

• Given:
  – 3D ↔ 2D correspondences
  – General perspective camera model

• Write equations:

\[
\frac{ax_1 + by_1 + cz_1 + d}{ix_1 + jy_1 + kz_1 + l} = u_1
\]

\[
\frac{ex_1 + fy_1 + gz_1 + h}{ix_1 + jy_1 + kz_1 + l} = v_1
\]

\[\vdots\]
Camera Calibration – linear system

\[
\begin{pmatrix}
x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & -u_1 \\
0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & -v_1 \\
x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2z_2 & -u_2 \\
0 & 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_2x_2 & -v_2y_2 & -v_2z_2 & -v_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c \\
\vdots \\
l
\end{pmatrix}
= \begin{pmatrix}
\vec{0}
\end{pmatrix}
\]

- Overconstrained (more equations than unknowns)
- Underconstrained (rank deficient matrix – any multiple of a solution, including 0, is also a solution)
Camera Calibration – linear system

- Standard linear least squares methods for $Ax=0$ will give the solution $x=0$
- Instead, look for a solution with $|x|=1$
- That is, minimize $|Ax|^2$ subject to $|x|^2=1$
• Minimize $|Ax|^2$ subject to $|x|^2 = 1$

• $|Ax|^2 = (Ax)^T(Ax) = (x^T A^T)(Ax) = x^T(A^T A)x$

• Expand $x$ in terms of eigenvectors of $A^T A$:
  
  $x = \mu_1 e_1 + \mu_2 e_2 + \ldots$
  
  $x^T(A^T A)x = \lambda_1 \mu_1^2 + \lambda_2 \mu_2^2 + \ldots$

  $|x|^2 = \mu_1^2 + \mu_2^2 + \ldots$
Camera Calibration – linear system

• To minimize

\[ \lambda_1 \mu_1^2 + \lambda_2 \mu_2^2 + \ldots \]

subject to

\[ \mu_1^2 + \mu_2^2 + \ldots = 1 \]

set \( \mu_{\text{min}} = 1 \) and all other \( \mu_i = 0 \)

• Thus, least squares solution is eigenvector of \( A^T A \) corresponding to minimum (nonzero) eigenvalue
Camera calibration: Linear method

• Advantages: easy to formulate and solve

• Disadvantages
  – Doesn’t directly tell you camera parameters
  – Doesn’t model radial distortion
  – Can’t impose constraints, such as known focal length and orthogonality

• Non-linear methods are preferred
  – Define error as squared distance between projected points and measured points
  – Minimize error using Newton’s method or other non-linear optimization
Camera calibration without known coordinates

• What if world coordinates of reference 3D points are not known?
• We can use scene features such as vanishing points

Slide from Efros, Photo from Criminisi
Recall: Vanishing points

- All lines having the same direction share the same vanishing point
Computing vanishing points

- \( \mathbf{X}_\infty \) is a point at infinity, \( \mathbf{v} \) is its projection: \( \mathbf{v} = \mathbf{P} \mathbf{X}_\infty \)
- The vanishing point depends only on line direction
- All lines having direction \( \mathbf{D} \) intersect at \( \mathbf{X}_\infty \)
Calibration from vanishing points

- Consider a scene with three orthogonal vanishing directions:

- Note: $v_1$, $v_2$ are finite vanishing points and $v_3$ is an infinite vanishing point.
Calibration from vanishing points

- Consider a scene with three orthogonal vanishing directions:

- We can align the world coordinate system with these directions.
Calibration from vanishing points

\[ P = \begin{bmatrix}
* & * & * & * \\
* & * & * & * \\
* & * & * & *
\end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix} \]

- \( p_1 = P(1,0,0,0)^T \) – the vanishing point in the x direction
- Similarly, \( p_2 \) and \( p_3 \) are the vanishing points in the y and z directions
- \( p_4 = P(0,0,0,1)^T \) – projection of the origin of the world coordinate system
- Problem: we can only know the four columns up to independent scale factors, additional constraints needed to solve for them
Calibration from vanishing points

• Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

\[
e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]

\[
\lambda_i v_i = K[R | t] \begin{bmatrix} e_i \\ 0 \end{bmatrix} = KRe_i
\]

\[
e_i = \lambda_i R^T K^{-1} v_i, \quad e_i^T e_j = 0
\]

\[
v_i^T K^{-T} RR^T K^{-1} v_j = v_i^T K^{-T} K^{-1} v_j = 0
\]

• Each pair of vanishing points gives us a constraint on the focal length and principal point (assuming zero skew and unit aspect ratio).
Calibration from vanishing points

Cannot recover focal length, principal point is the third vanishing point

Can solve for focal length, principal point
Rotation from vanishing points

\[ \lambda_i v_i = K[R \mid t] \begin{bmatrix} e_i \\ 0 \end{bmatrix} = KRe_i \]

\[ \lambda_i K^{-1} v_1 = Re_1 = [r_1 \quad r_2 \quad r_3] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = r_1 \]

\[ \lambda_i K^{-1} v_i = r_i. \]

Thus,

Get \( \lambda_i \) by using the constraint \( ||r_i||^2 = 1 \).
Calibration from vanishing points: Summary

• Solve for K (focal length, principal point) using three orthogonal vanishing points
• Get rotation directly from vanishing points once calibration matrix is known

• Advantages
  – No need for calibration chart, 2D-3D correspondences
  – Could be completely automatic

• Disadvantages
  – Only applies to certain kinds of scenes
  – Inaccuracies in computation of vanishing points
  – Problems due to infinite vanishing points
Stereo and epipolar geometry
Binocular stereo

- Given a calibrated binocular stereo pair, fuse it to produce a depth image.

image 1  
image 2  

Dense depth map
Multi-Camera Geometry

• Epipolar geometry – relationship between observed positions of points in multiple cameras

• Assume:
  – 2 cameras
  – Known intrinsics and extrinsics
Epipolar Geometry

\[ p_1 \quad \text{and} \quad p_2 \]

\[ C_1 \quad \text{and} \quad C_2 \]
Epipolar Geometry

\[ P \]

\[ C_1 \]

\[ p_1 \]

\[ C_2 \]

\[ p_2 \]

\[ l_2 \]
Epipolar Geometry

Epipolar line

Epipoles
Epipolar Geometry

- Epipolar constraint: corresponding points must lie on conjugate epipolar lines
  - Search for correspondences becomes a 1-D problem
Basic stereo matching algorithm

- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match
  - Triangulate the matches to get depth information

- Simplest case: epipolar lines are corresponding scanlines
  - When does this happen?
Simplest Case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
Simplest Case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then epipolar lines fall along the horizontal scan lines of the images
What if images are not aligned?
Epipolar Geometry

- **Goal:** derive equation for $l_2$
- **Observation:** $P, C_1, C_2$ determine a plane
Epipolar Geometry

- Work in coordinate frame of $C_1$
- Normal of plane is $T \times R p_2$, where $T$ is relative translation, $R$ is relative rotation
Epipolar Geometry

- \( p_1 \) is perpendicular to this normal:

\[
p_1 \cdot (T \times R p_2) = 0
\]
Write cross product as matrix multiplication

\[ \vec{T} \times x = T^x x, \quad T^x = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix} \]
Epipolar Geometry

- $p_1 \cdot T \times R \ p_2 = 0 \implies p_1^T E p_2 = 0$
- $E$ is the essential matrix
Essential Matrix

- $E$ depends only on camera geometry
- Given $E$, can derive equation for line $l_2$
Concrete example: parallel images

- Rotation?
- Identity
- Translation?

\[
T = \begin{bmatrix}
0 & -T_z & T_y \\
T_z & 0 & -T_x \\
-T_y & T_x & 0
\end{bmatrix}
\]

\[
T = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & -t & 0 \\
0 & t & 0 & 0
\end{bmatrix}
\]
Concrete example: parallel images

The y-coordinates of corresponding points are the same!
Giving the consequence that:

\[ \text{disparity} = x - x' = \frac{B \cdot f}{z} \]

Disparity is inversely proportional to depth!
Fundamental Matrix

• Can define **fundamental matrix** $F$ analogously to essential matrix, operating on pixel coordinates instead of camera coordinates
  
  $$u_1^T F u_2 = 0$$

• Advantage: can sometimes estimate $F$ without knowing camera calibration
  
  – Given a few good correspondences, can get epipolar lines and estimate more correspondences, all without calibrating cameras
From epipolar geometry to camera calibration

- Estimating the fundamental matrix is known as “weak calibration”
- If we know the calibration matrices of the two cameras, we can estimate the essential matrix: $E = K'FK$
- The essential matrix gives us the relative rotation and translation between the cameras, or their extrinsic parameters

Source: S. Lazebnik
Structure from motion: basic idea

Camera 1 \( R_1, t_1 \)

Camera 2 \( R_2, t_2 \)

Camera 3 \( R_3, t_3 \)

Slide credit: Noah Snavely
Next time: intro to deep learning