Introduction to Multiview Geometry

Haibin Ling

CIS 5543 – Computer Vision, Fall 2017

Image Formation & Camera Models

How do we see the world?

- Let’s design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

Pinhole Camera

- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the aperture

Pinhole camera model

- Pinhole model:
  - Captures pencil of rays – all rays through a single point
  - The point is called Center of Projection (focal point)
  - The image is formed on the Image Plane

Dimensionality Reduction Machine (3D to 2D)

- 3D world
- 2D image

What have we lost?
- Angles
- Distances (lengths)
**Historical context**

- **Pinhole model:** Mozi (470-390 BCE), Aristotle (384-322 BCE)
- **Principles of optics (including lenses):** Alhacen (965-1039 CE)
- **Camera obscura:** Leonardo da Vinci (1452-1519), Johann Zahn (1631-1707)
- **First photo:** Joseph Nicephore Niepce (1822)
- **Daguerreotypes** (1839)
- **Photographic film:** Eastman (1889)
- **Cinema:** Lumière Brothers, 1895
- **Color Photography:** Lumière Brothers, 1908
- **Television:** Baird, Farnsworth, Zworykin, 1920s
- **First consumer camera with CCD:** Sony Mavica (1981)
- **First fully digital camera:** Kodak DCS100 (1990)

**Projection Model of the Pinhole Camera**

- Camera coordinate system \( (x', y', z') \)
- Optical center \( O \)
- Focal length \( f \)
- Image plane \( \Pi \)

**Projection properties**

- **Many to one:** any points along same ray map to same point in image
- **Points \( \rightarrow \) points**
  - But projection of points on focal plane is undefined
- **Lines \( \rightarrow \) lines** (collinearity is preserved)
  - But line through focal point projects to a point
- **Planes \( \rightarrow \) planes** (or half-planes)
  - But plane through focal point projects to line

**Homogeneous coordinates**

- **Linear transformation?**
  - No — division by \( z \) is nonlinear

To make it "linear": add one more coordinate:

\[
\begin{pmatrix}
  x \\
  y \\
  z
\end{pmatrix} \Rightarrow
\begin{pmatrix}
  x' = x/x_u \\
  y' = y/y_u \\
  1
\end{pmatrix}
\]

From homogeneous coordinates to normal ones:

\[
\begin{pmatrix}
  x' \\
  y' \\
  z'
\end{pmatrix} \Rightarrow
\begin{pmatrix}
  x = x' z' \\
  y = y' z' \\
  z = z'
\end{pmatrix}
\]
Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates:

\[
\begin{pmatrix}
\frac{x'}{w'} = \frac{x}{w} = \begin{pmatrix} 1 & 0 & 0 & -x \end{pmatrix} \begin{pmatrix} \frac{1}{w} & 0 & 0 & 0 \\
0 & \frac{1}{w} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\end{pmatrix}
\]

- Further simplified

\[
\begin{pmatrix}
\frac{x'}{w'} = \frac{x}{w} = \begin{pmatrix} 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\end{pmatrix}
\]

More Coordinate Systems

- From camera coordinates to image coordinates
  - 3D to 2D
  - Rescale to pixel coordinates
- From world coordinates to camera coordinates
  - Rotation R, translation t

More Coordinate Systems

\[
\begin{pmatrix}
2D point (3x1) \\
\end{pmatrix} = \begin{pmatrix}
\text{Camera to pixel coord. trans. matrix (3x3)} \\
\text{Perspective projection matrix (3x4)} \\
\text{World to camera coord. trans. matrix (4x4)} \\
\end{pmatrix} \begin{pmatrix}
3D point (4x1) \\
\end{pmatrix}
\]

Camera Parameters

\[
\begin{pmatrix}
\text{Intrinsic parameters} \\
\text{Extrinsic parameters}
\end{pmatrix}
\]

Camera Calibration

- Determine the coefficients of camera models
- Decide camera parameters:
  - Intrinsic parameters: pixel size, focal length, image center, non-linear distortion (optional)
  - Extrinsic parameters (camera pose): rotation and translation
- Decide projection matrix
  - Elements in M

Linear Systems

- Given a 3D point \( x^w \) and its image \( p = (u, v) \), after removing \( z^w \), we have:

\[
\begin{pmatrix}
x^w \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{pmatrix} \begin{pmatrix}
x^w \end{pmatrix} = \begin{pmatrix} b_1 \\
b_2 \\
b_3
\end{pmatrix}
\]

- Two linear equations are exactly a line in 3D space
- Two equations for \( m_{44} \) unknowns \( m_{44} \) is free, set to 1
- Two point pairs are sufficient for solving \( M \)
- More points could give a better solution
Linear Systems

Given a 3D point \( P \) and their images \( p_1 = (x_1,y_1)^T \), we have a linear system:

\[
\begin{bmatrix}
M^1 & 0 & 0 & -x_1M^1 & 0 & 0 & -y_1M^1 & 0 & 0 \\
0 & M^1 & 0 & 0 & -x_2 & 0 & 0 & -y_2 & 0 \\
0 & 0 & M^1 & 0 & 0 & -x_3 & 0 & 0 & -y_3 \\
M^2 & 0 & 0 & -x_1M^2 & 0 & 0 & -y_1M^2 & 0 & 0 \\
0 & M^2 & 0 & 0 & -x_2 & 0 & 0 & -y_2 & 0 \\
0 & 0 & M^2 & 0 & 0 & -x_3 & 0 & 0 & -y_3 \\
N^1 & 0 & 0 & -x_1N^1 & 0 & 0 & -y_1N^1 & 0 & 0 \\
0 & N^1 & 0 & 0 & -x_2 & 0 & 0 & -y_2 & 0 \\
0 & 0 & N^1 & 0 & 0 & -x_3 & 0 & 0 & -y_3 \\
N^2 & 0 & 0 & -x_1N^2 & 0 & 0 & -y_1N^2 & 0 & 0 \\
0 & N^2 & 0 & 0 & -x_2 & 0 & 0 & -y_2 & 0 \\
0 & 0 & N^2 & 0 & 0 & -x_3 & 0 & 0 & -y_3 \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{v}_1 \\
\mathbf{v}_2 \\
\mathbf{v}_3 \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{0} \\
\mathbf{0} \\
\mathbf{0} \\
\end{bmatrix}
\]

For \( x \geq 6 \), it can be solved by minimum least squares (MSE);
- Parameters can be inferred from the solution.
- 10 unknowns (4 intrinsic + 6 extrinsic)

Challenges in Practice

- Noise in image formation
- Noise in camera models
- Noise in feature locations
- Noise in feature matching
- ....

Perspective distortion

Radial Distortion

- Caused by imperfect lenses
- Deviations are most noticeable for rays that pass through the edge of the lens

3D Reconstruction - Points

Two review linear system

- Two cameras, two matching pairs
- Known: cameras projections \( M^1 \) and \( M^2 \)
- Images of the point: \( p_1 \) and \( p_2 \)
- Unknowns coordinates of \( P = (X,Y,Z)^T \)
- Linear systems:

\[
\begin{bmatrix}
R^1 \\
R^2 \\
\end{bmatrix}
= \begin{bmatrix}
M^1_{11} & M^1_{12} & M^1_{13} & M^1_{14} & 1 \\
M^2_{11} & M^2_{12} & M^2_{13} & M^2_{14} & 1 \\
M^1_{21} & M^2_{21} & M^1_{23} & M^2_{23} & 1 \\
M^1_{31} & M^2_{31} & M^1_{33} & M^2_{33} & 1 \\
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
R^1 \\
R^2 \\
\end{bmatrix}
= \begin{bmatrix}
M^1_{11} & M^1_{12} & M^1_{13} & M^1_{14} & 1 \\
M^2_{11} & M^2_{12} & M^2_{13} & M^2_{14} & 1 \\
M^1_{21} & M^2_{21} & M^1_{23} & M^2_{23} & 1 \\
M^1_{31} & M^2_{31} & M^1_{33} & M^2_{33} & 1 \\
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1 \\
\end{bmatrix}
\]
Intuition – Triangulation

Does not work for pure rotation!

In practice …

- How to find corresponding points $p_1$ and $p_2$?
  - Local features: e.g., Harris corners
  - Describe these points: e.g., SIFT
  - Reliable matching: e.g., RANSAC
  - Handling noise/uncertainty: e.g., multi-view