Single-view geometry

Odilon Redon, Cyclops, 1914
Our goal: Recovery of 3D structure

- Recovery of structure from one image is inherently ambiguous
Our goal: Recovery of 3D structure

- Recovery of structure from one image is inherently ambiguous
Our goal: Recovery of 3D structure

• Recovery of structure from one image is inherently ambiguous
Ames Room

http://en.wikipedia.org/wiki/Ames_room
Our goal: Recovery of 3D structure

- We will need *multi-view geometry*
Recall: Pinhole camera model

- **Principal axis**: line from the camera center perpendicular to the image plane
- **Normalized (camera) coordinate system**: camera center is at the origin and the principal axis is the z-axis
Recall: Pinhole camera model

\[(X, Y, Z) \mapsto (fX/Z, fY/Z)\]

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

\[x = PX\]
Principal point

- **Principal point (p):** point where principal axis intersects the image plane (origin of normalized coordinate system)
- Normalized coordinate system: origin is at the principal point
- Image coordinate system: origin is in the corner
- How to go from normalized coordinate system to image coordinate system?
Principal point offset

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

\((X, Y, Z) \mapsto \left( f \frac{X}{Z} + p_x, f \frac{Y}{Z} + p_y \right)\)

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix} \mapsto \begin{pmatrix}
fX + Zp_x \\
fY + Zp_y \\
Z
\end{pmatrix} = \begin{bmatrix}
f & p_x & 0 \\
f & p_y & 0 \\
1 & 0 & 1
\end{bmatrix} \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]
Principal point offset

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

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

\[
K = \begin{bmatrix}
 f & p_x \\
 f & p_y \\
 1 & 1 & 1 & 0
\end{bmatrix}
\text{calibration matrix}
\]

\[
P = K[I | 0]
\]
Pixel coordinates

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

\( m_x \) pixels per meter in horizontal direction,
\( m_y \) pixels per meter in vertical direction

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

pixels/m \times m \times \text{pixels}
Camera rotation and translation

- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation.

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

- coords. of point in camera frame
- coords. of a point in world frame (nonhomogeneous)
- coords. of camera center in world frame
Camera rotation and translation

In non-homogeneous coordinates:

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

\[
X_{\text{cam}} = \begin{bmatrix}
R & -R\tilde{C} \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\tilde{X} \\
1
\end{bmatrix} = \begin{bmatrix}
R & -R\tilde{C} \\
0 & 1
\end{bmatrix} X
\]

\[
x = K[I \mid 0]X_{\text{cam}} = K[R \mid -R\tilde{C}]X \quad P = K[R \mid t], \quad t = -R\tilde{C}
\]

Note: C is the null space of the camera projection matrix (PC=0)
Camera parameters

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

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

• 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
Camera calibration

\[ x = K[R \ t]X \]

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

Source: D. Hoiem
Camera calibration

- Given n points with known 3D coordinates $X_i$ and known image projections $x_i$, estimate the camera parameters.
Camera calibration: Linear method

\[ \lambda x_i = PX_i \]
\[ x_i \times PX_i = 0 \]

\[
\begin{bmatrix}
0 & -X_i^T & y_iX_i^T \\
X_i^T & 0 & -x_iX_i^T \\
-y_iX_i^T & x_iX_i^T & 0
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix} = 0
\]

Two linearly independent equations
P has 11 degrees of freedom (12 parameters, but scale is arbitrary)

One 2D/3D correspondence gives us two linearly independent equations

Homogeneous least squares

6 correspondences needed for a minimal solution
Camera calibration: Linear method

\[
\begin{bmatrix}
0^T & X_1^T & -y_1X_1^T \\
X_1^T & 0^T & -x_1X_1^T \\
\vdots & \vdots & \vdots \\
0^T & X_n^T & -y_nX_n^T \\
X_n^T & 0^T & -x_nX_n^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix} = 0 \quad Ap = 0
\]

- Note: for coplanar points that satisfy \(\Pi^T X = 0\), we will get degenerate solutions \((\Pi,0,0)\), \((0,\Pi,0)\), or \((0,0,\Pi)\)
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 difference between projected points and measured points
  - Minimize error using Newton’s method or other non-linear optimization

Source: D. Hoiem
Multi-view geometry problems

- **Structure**: Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point.
Multi-view geometry problems

- **Stereo correspondence**: Given a point in one of the images, where could its corresponding points be in the other images?
Multi-view geometry problems

- **Motion:** Given a set of corresponding points in two or more images, compute the camera parameters.
Triangulation

• Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point
Triangulation

- We want to intersect the two visual rays corresponding to $x_1$ and $x_2$, but because of noise and numerical errors, they don’t meet exactly.
Triangulation: Geometric approach

- Find shortest segment connecting the two viewing rays and let X be the midpoint of that segment
Triangulation: Linear approach

\[ \lambda_1 x_1 = P_1 X \quad x_1 \times P_1 X = 0 \quad [x_{1\times}] P_1 X = 0 \]
\[ \lambda_2 x_2 = P_2 X \quad x_2 \times P_2 X = 0 \quad [x_{2\times}] P_2 X = 0 \]

Cross product as matrix multiplication:

\[ \mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a} \times] \mathbf{b} \]
Triangulation: Linear approach

\[ \lambda_1 x_1 = P_1 X \quad x_1 \times P_1 X = 0 \quad [x_{1\times}]P_1 X = 0 \]
\[ \lambda_2 x_2 = P_2 X \quad x_2 \times P_2 X = 0 \quad [x_{2\times}]P_2 X = 0 \]

Two independent equations each in terms of three unknown entries of \( X \)
Triangulation: Nonlinear approach

Find $X$ that minimizes

$$d^2(x_1, P_1X) + d^2(x_2, P_2X)$$