Geometry of a single camera

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

- Recovery of structure from one image is inherently ambiguous
Single-view ambiguity
Single-view ambiguity

Rashad Alakbarov shadow sculptures
Anamorphic perspective

H. Holbein The Younger, *The Ambassadors*, 1533

https://en.wikipedia.org/wiki/Anamorphosis
Ames Room

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

- When certain assumptions hold, we can recover structure from a single view
- In general, we need multi-view geometry

- But first, we need to understand the geometry of a single camera...
Review: Pinhole camera model

- **Normalized (camera) coordinate system**: camera center is at the origin, the *principal axis* is the z-axis, x and y axes of the image plane are parallel to x and y axes of the world.

- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system.
Review: Pinhole camera model

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

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

\[\mathbf{x} = \mathbf{P}\mathbf{X}\]
Principal point

- **Principal point \( (p) \):** point where principal axis intersects the image plane
- Normalized coordinate system: origin of the image is at the principal point
- Image coordinate system: origin is in the corner
Principal point offset

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

\[
(X, Y, Z) \mapsto (f \frac{X}{Z} + p_x, f \frac{Y}{Z} + p_y)
\]

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

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

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

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

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

$m_x$ pixels per meter in horizontal direction, $m_y$ pixels per meter in vertical direction.

$$K = \begin{bmatrix} m_x & m_y & 1 \\ \end{bmatrix} \begin{bmatrix} f & P_x \\ f & P_y \\ 1 & 1 \\ \end{bmatrix} = \begin{bmatrix} \alpha_x & \beta_x \\ \alpha_y & \beta_y \\ 1 & 1 \\ \end{bmatrix}$$

Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$
Camera rotation and translation

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

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

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

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

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

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

\[ X = K[I \mid 0]X_{\text{cam}} = K[R \mid -R\tilde{C}]X \]

\[ P = K[R \mid t], \quad t = -R\tilde{C} \]
Camera parameters

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

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

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

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

- **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

  \[ \text{What is the projection of the camera center?} \]

\[ PC = K \begin{bmatrix} R & -R\tilde{C} \end{bmatrix} \begin{bmatrix} \tilde{C} \\ 1 \end{bmatrix} = 0 \]

The camera center is the *null space* of the projection matrix!
Camera calibration

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

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

Source: D. Hoiem
Camera calibration

- Given $n$ points with known 3D coordinates $X_i$ and known image projections $x_i$, estimate the camera parameters $P$. 

![Camera calibration diagram](image)
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
Camera calibration: Linear method

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

- \( P \) has 11 degrees of freedom
- One 2D/3D correspondence gives us two linearly independent equations
  - 6 correspondences needed for a minimal solution
- Homogeneous least squares: find \( p \) minimizing \( \| Ap \|^2 \)
  - Solution given by eigenvector of \( A^TA \) with smallest eigenvalue
Camera calibration: Linear method

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

• Note: for coplanar points that satisfy \( \Pi^T X = 0 \), we will get degenerate solutions \((\Pi, 0, 0), (0, \Pi, 0), \text{ or } (0, 0, \Pi)\)
Camera calibration: Linear vs. nonlinear

• Linear calibration is easy to formulate and solve, but it doesn’t directly tell us the camera parameters

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

vs.

\[
x = K[R \hspace{1em} t]X
\]

• In practice, non-linear methods are preferred
  • Write down objective function in terms of intrinsic and extrinsic parameters
  • Define error as sum of squared distances between measured 2D points and estimated projections of 3D points
  • Minimize error using Newton’s method or other non-linear optimization
  • Can model radial distortion and impose constraints such as known focal length and orthogonality
A taste of multi-view geometry: Triangulation

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

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

\[ \text{O}_1 \quad \text{x}_1 \quad \text{O}_2 \quad \text{x}_2 \quad \text{X?} \]
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: Nonlinear approach

Find $X$ that minimizes

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

\[
\mathbf{a} \times \mathbf{b} = \begin{bmatrix}
0 & -a_z & a_y \\
-1 & 0 & a_x \\
-1 & -a_y & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{b}_1 \\
\mathbf{b}_2 \\
\mathbf{b}_3
\end{bmatrix} = [\mathbf{a}_\times] \mathbf{b}
\]
Triangulation: Linear approach

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

Two independent equations each in terms of three unknown entries of \( X \)
Structure from motion

- Given 2D point correspondences between multiple images, compute the camera parameters and the 3D points

Camera 1: \( R_1, t_1 \)

Camera 2: \( R_2, t_2 \)

Camera 3: \( R_3, t_3 \)
• **Structure:** Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point.
• **Motion:** Given a set of *known* 3D points seen by a camera, compute the camera parameters
Useful reference

Multiple View Geometry in Computer Vision

Richard Hartley and Andrew Zisserman