Today

Perspective projection

Camera calibration
Quiz #1: Scaling vs Foreshortening
Recall: Full Perspective Camera Model

\[
\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} f_u r_1^t + u_0 r_3^t & f_u t_X + u_0 t_z \\ f_v r_2^t + v_0 r_3^t & f_v t_Y + v_0 t_z \\ r_3^t & t_z \end{bmatrix} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \\ 1 \end{bmatrix}
\]

\[
\lambda = Z^{(C)} = r_3^T M^{(W)} + t_z
\]
Recall Full Perspective Camera Model

\[ \lambda m^{(I)} = PM^{(W)} \]

Collinearity equation

\[ u^{(I)} - u_0 = f_u \frac{r_{11}X^{(W)} + r_{12}Y^{(W)} + r_{13}Z^{(W)} + t_X}{r_{31}X^{(W)} + r_{32}Y^{(W)} + r_{33}Z^{(W)} + t_Z} \]

\[ v^{(I)} - v_0 = f_v \frac{r_{21}X^{(W)} + r_{22}Y^{(W)} + r_{23}Z^{(W)} + t_Y}{r_{31}X^{(W)} + r_{32}Y^{(W)} + r_{33}Z^{(W)} + t_Z} \]
Weak Perspective Camera Model

Perspective projection:

\[ u^{(C)} = f \frac{X^{(C)}}{Z^{(C)}} \approx f \frac{X^{(C)}}{\tilde{Z}^{(C)}} \]

\[ v^{(C)} = f \frac{Y^{(C)}}{Z^{(C)}} \approx f \frac{Y^{(C)}}{\tilde{Z}^{(C)}} \]

\[ \lambda \begin{bmatrix} u^{(C)} \\ v^{(C)} \\ 1 \end{bmatrix} = f \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ \tilde{Z}^{(C)} \end{bmatrix} \quad \lambda = \tilde{Z}^{(C)} \]
Weak Perspective Camera Model

Spatial sampling:

\[
\begin{bmatrix}
u^{(I)} \\ v^{(I)} \\ 1
\end{bmatrix} = \begin{bmatrix}
k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
u^{(c)} \\ v^{(c)} \\ 1
\end{bmatrix}
\]

\[
\lambda \begin{bmatrix}
u^{(I)} \\ v^{(I)} \\ 1
\end{bmatrix} = f \begin{bmatrix}
k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
X^{(c)} \\ Y^{(c)} \\ Z^{(c)} \\
\end{bmatrix} = \begin{bmatrix}
f_u & 0 & u_0 f \\ 0 & f_v & v_0 f \\ 0 & 0 & f
\end{bmatrix} \begin{bmatrix}
X^{(c)} \\ Y^{(c)} \\ Z^{(c)} \\
\end{bmatrix}
\]
Weak Perspective Camera Model

Object frame to camera frame:

\[
\begin{bmatrix}
X^{(C)} \\
Y^{(C)} \\
Z^{(C)} \\
1
\end{bmatrix} = 
\begin{bmatrix}
R_{3 \times 3} & T_{3 \times 1} \\
0_{1 \times 3} & 1
\end{bmatrix}
\begin{bmatrix}
X^{(W)} \\
Y^{(W)} \\
Z^{(W)} \\
1
\end{bmatrix} =
\begin{bmatrix}
\mathbf{r}_1^t & t_x \\
\mathbf{r}_2^t & t_y \\
\mathbf{r}_3^t & t_z
\end{bmatrix}
\begin{bmatrix}
0_{1 \times 3} \\
1
\end{bmatrix}
\begin{bmatrix}
X^{(W)} \\
Y^{(W)} \\
Z^{(W)} \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
X^{(C)} \\
Y^{(C)}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{r}_1^t & t_x \\
\mathbf{r}_2^t & t_y
\end{bmatrix}
\begin{bmatrix}
X^{(W)} \\
Y^{(W)} \\
Z^{(W)} \\
1
\end{bmatrix}
\text{or}
\begin{bmatrix}
X^{(C)} \\
Y^{(C)} \\
\bar{Z}^{(C)} \\
\frac{1}{f}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{r}_1^t & t_x \\
\mathbf{r}_2^t & t_y \\
0_{1 \times 3} & \bar{Z}^{(C)} \\
\frac{1}{f} & 1
\end{bmatrix}
\begin{bmatrix}
X^{(W)} \\
Y^{(W)} \\
Z^{(W)} \\
1
\end{bmatrix}
\]

2x4 matrix
3x4 matrix
**Weak Perspective Camera Model**

**Overall Weak Perspective Camera Model:**

\[
\lambda \begin{bmatrix} u^{(l)} \\ v^{(l)} \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0f \\ 0 & f_v & v_0f \\ 0 & 0 & f \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ \bar{Z}(C) \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0f \\ 0 & f_v & v_0f \\ 0 & 0 & f \end{bmatrix} \begin{bmatrix} r_1^t \\ r_2^t \\ t_X \\ t_Y \\ \bar{Z}(C) \end{bmatrix} \begin{bmatrix} 0_{1\times3} \\ f \end{bmatrix} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \end{bmatrix}
\]

**Projection matrix** \( P_{\text{weak}} \)

\[
P_{\text{weak}} = \begin{bmatrix} f_u r_1^t & f_u t_X + u_0 \bar{Z}(C) \\ f_v r_2^t & f_v t_Y + v_0 \bar{Z}(C) \\ 0_{1\times3} & \bar{Z}(C) \end{bmatrix}
\]

What is the issue?

\[
\lambda = \bar{Z}(C) \approx r_3^t \mathbf{M}^{(W)} + t_Z
\]
Perspective V.S. Weak Perspective

\[ P = \begin{bmatrix} f_u r_1^t + u_0 r_3^t & f_u t_X + u_0 t_z \\ f_v r_2^t + v_0 r_3^t & f_v t_Y + v_0 t_z \\ r_3^t & t_z \end{bmatrix} \]

\[ P_{weak} = \begin{bmatrix} f_u r_1^t & f_u t_X + u_0 \bar{Z}^{(C)} \\ f_v r_2^t & f_v t_Y + v_0 \bar{Z}^{(C)} \\ 0_{1\times3} & \bar{Z}^{(C)} \end{bmatrix} \]

What’s the major difference?
What is Camera Calibration?

Estimate the intrinsic parameters, which do not depend on camera position

1. \( f_u = f k_u \)
2. \( f_v = f k_v \)
3. \( u_0 \)
4. \( v_0 \)
5. \( \gamma = f_u \cot \theta \)

Skew of the two axes on the image plane

\[
W' = \begin{bmatrix}
  f_u & \gamma & u_0 \\
  0 & f_v & v_0 \\
  0 & 0 & 1 \\
\end{bmatrix}
\]

In practice, we assume \( \theta = 90^\circ \Rightarrow \cot \theta = 0 \)
Camera Calibration

Reliable and accurate camera calibration is a basic problem in computer vision

It has been investigated for many years

There are still many researchers working on it

Calibration using features that are easier to achieve
Camera Calibration Methods

- **Conventional camera calibration method**
  - Input: pairs of 3D object points and 2D image points
  - May work for a single view of the object

- **Camera self-calibration**
  - Input: only 2D image points
  - Require multiple images taken from different views
  - Convenient but less constrained

- **Other methods using circles, balls, lines, vanishing points, etc.**

Pattern printed on a cubic  
Planar pattern
Conventional Camera Calibration

General strategy:
- Take pictures from calibration patterns with known coordinates of 3D features (e.g., points, lines, or curves)
- Identify the corresponding image features
- Obtain the projection matrix by minimizing projection error
- Obtain intrinsic parameters from the projection matrix

Projection matrix:

\[
P = \begin{bmatrix}
    f_u r_1^t + u_0 r_3^t & f_u t_x + u_0 t_z \\
    f_v r_2^t + v_0 r_3^t & f_v t_y + v_0 t_z \\
    r_3^t & t_z
\end{bmatrix}
\]
Estimating Camera Parameters

For N pairs of 3D point-- 2D image point

\[(u_1, v_1), (X_1, Y_1, Z_1)\]
\[(u_2, v_2), (X_2, Y_2, Z_2)\]
\[(u_3, v_3), (X_3, Y_3, Z_3)\]
\[(u_N, v_N), (X_N, Y_N, Z_N)\]
Recall: Full Perspective Camera Model

\[ \lambda m^{(I)} = WM^{(C)}, \text{ and } M^{(C)} = DM^{(W)} \Rightarrow \lambda m^{(I)} = WDM^{(W)} \]

\[
\begin{bmatrix}
\lambda u^{(I)} \\
\lambda v^{(I)} \\
1
\end{bmatrix} =
\begin{bmatrix}
f_u & 0 & u_0 & 0 \\
0 & f_v & v_0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R_{3\times3} \\
T_{3\times1} \\
0_{1\times3} \\
1
\end{bmatrix}
\begin{bmatrix}
X^{(W)} \\
Y^{(W)} \\
Z^{(W)} \\
1
\end{bmatrix}
\]

Projection matrix:

\[
P_{3\times4} = [W' \ 0_{3\times1}]
\begin{bmatrix}
R_{3\times3} & T_{3\times1} \\
0_{1\times3} & 1
\end{bmatrix}
= [W'R \ W'T]
\]

\[
\begin{bmatrix}
p_1^t \\
p_2^t \\
p_3^t \\
p_{14} \\
p_{24} \\
p_{34}
\end{bmatrix}
\]

where \( W' =
\begin{bmatrix}
f_u & 0 & u_0 \\
0 & f_v & v_0 \\
0 & 0 & 1
\end{bmatrix} \) and \( p_i =
\begin{bmatrix}
p_{i1} \\
p_{i2} \\
p_{i3}
\end{bmatrix} \)
Recall: Full Perspective Camera Model

\[
P = \begin{bmatrix}
    f_urt^t + u_0r_3^t & f_ut_X + u_0t_z \\
    f_vr_2^t + v_0r_3^t & f_vt_Y + v_0t_z \\
    r_3^t & t_z
\end{bmatrix} = \begin{bmatrix}
    p_1^t & p_{14} \\
    p_2^t & p_{24} \\
    p_3^t & p_{34}
\end{bmatrix}
\]

\[\Rightarrow p_3^t = r_3^t \quad \text{and} \quad p_{34} = t_z\]
Camera Calibration from Projection Matrix $P$: A Linear Method

$$
\lambda \begin{bmatrix} u(I) \\ v(I) \\ 1 \end{bmatrix} = \begin{bmatrix} p_1^t \\ p_2^t \\ p_3^t \end{bmatrix} \begin{bmatrix} X(W) \\ Y(W) \\ Z(W) \end{bmatrix}
$$

$$
\lambda = p_3^t M'(W) + p_{34}
$$

$$
M'(W) = \begin{bmatrix} X(W) & Y(W) & Z(W) \end{bmatrix}^t
$$

$$
(p_3^t M'(W) + p_{34}) u(I) = p_1^t M'(W) + p_{14}
$$

$$
(p_3^t M'(W) + p_{34}) v(I) = p_2^t M'(W) + p_{24}
$$

$$
p_1^t M'(W) + p_{14} - p_3^t M'(W) u(I) - p_{34} u(I) = 0
$$

$$
p_2^t M'(W) + p_{24} - p_3^t M'(W) v(I) - p_{34} v(I) = 0
$$
Camera Calibration from Projection Matrix \( P \): A Linear Method

For every pair of 3D-2D points, we have a pair of linear equations:

\[
\begin{align*}
\mathbf{p}_1^t \mathbf{M}'(W) + p_{14} - \mathbf{p}_3^t \mathbf{M}'(W) u(I) - p_{34} u(I) &= 0 \\
\mathbf{p}_2^t \mathbf{M}'(W) + p_{24} - \mathbf{p}_3^t \mathbf{M}'(W) v(I) - p_{34} v(I) &= 0
\end{align*}
\]

We can setup a linear equation system for \( N \) corresponding pairs with 2\( N \) equations:

- **Known**: 2D and 3D coordinates
- **Unknown**: 4x3 parameters of \( P \)

\[
\mathbf{A} \mathbf{v} = 0
\]

\( \mathbf{A} \) is a \( 2N \times 12 \) matrix containing the known

\( \mathbf{v} \) is a \( 12 \times 1 \) vector containing the unknown

\[
\mathbf{v} = (\mathbf{p}_1^t \quad p_{14} \quad \mathbf{p}_2^t \quad p_{24} \quad \mathbf{p}_3^t \quad p_{34})^t
\]
Camera Calibration from Projection Matrix P: A Linear Method

\[
A = \begin{bmatrix}
(M'_1(W))^t & 1 & 0_{1\times3} & 0 & -(M'_1(W))^t u_1^{(I)} & -u_1^{(I)} \\
0_{1\times3} & 0 & (M'_1(W))^t & 1 & -(M'_1(W))^t v_1^{(I)} & -v_1^{(I)} \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
(M'_N(W))^t & 1 & 0_{1\times3} & 0 & -(M'_N(W))^t u_N^{(I)} & -u_N^{(I)} \\
0_{1\times3} & 0 & (M'_N(W))^t & 1 & -(M'_N(W))^t v_N^{(I)} & -v_N^{(I)}
\end{bmatrix}
\]

How many pairs of points we need at least? 6

What is the requirement of the 3D points?
Cannot be coplanar
The projection matrix can be estimated as a solution to

\[ Av = 0 \]

The solution lies in the null space of \( A \) – the set of all vectors \( x \) that satisfies \( Ax = 0 \).

\[
\text{rank}(A) + \text{nullity}(A) = \min(\text{row}(A), \text{col}(A))
\]

- **Case 1**: If \( \text{rank}(A) = 11 \), the solution to \( v \) is unique up to a scaling factor. Estimate \( v \) by performing SVD on \( A \).

- **Case 2**: if \( \text{rank}(A) < 11 \), the solution to \( v \) is a linear combination of all null vectors of \( A \).

What about \( \text{rank}(A)=12 \)?
Singular Value Decomposition:

- Any $m \times n$ matrix can be written as the product of three matrices:
  \[ A_{m \times n} = U_{m \times m} D_{m \times n} V_{n \times n}^T \]

- Singular values $\sigma_i$ are fully determined by $A$:
  - $D$ is diagonal: $d_{ij} = 0$ if $i \neq j$; $d_{ii} = \sigma_i$ ($i = 1, 2, ..., n$)
  - $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N \geq 0$; $N = rank(A)$

- Both $U$ and $V$ are not unique:
  - Columns of each are mutual orthogonal vectors
Brief Review on SVD

- **U** contains the eigenvector of $AA^T$
  - Columns of **U** called the left-singular vectors

- **V** contains the eigenvector of $A^TA$
  - Columns of **V** called the right-singular vectors

- Both **U** and **V** are unitary matrices:
  
  $$UU^T = I_{m \times m} \text{ and } VV^T = I_{n \times n}$$
When rank($A$) = 11, $v$ can be estimated by performing SVD on $A$.

$v$ is the last column of the $V$ matrix with an unknown scalar $v = \alpha v'$

How to solve the scalar $\alpha$?

$p_3^t = r_3^t$ and $\|r_3\| = 1$  \[ \Rightarrow \|p_3\| = 1 \]

$|\alpha| = \frac{1}{\sqrt{v'_9^2 + v'_10^2 + v'_11^2}}$

The sign of $\alpha$ -- $\text{sign}(\alpha)$ is unknown!

The null vector of $A$ is solved!

Projection matrix $P$ is solved!
Discussion on the Rank of A

- **Rank(A)=11** with a proper configuration of the 3D points
  - In practice, rank(A)=12 (full rank) may happen due to the image noise and error in location estimation
- **Rank(A)<11**
  - Rank(A)=8 if the 3D points are coplanar
  - Results in infinite solutions
Issues in the Linear Method

- Numerical issue
- Outliers in observations
Numerical Stability: Data Normalization

The entries in $A$ have large difference in unit and scale

$$A = \begin{bmatrix}
\left(M'_1 (W) \right)^t & 1 & 0_{1 \times 3} & 0 & -\left(M'_1 (W) \right)^t u_1 (I) & -u_1 (I) \\
0_{1 \times 3} & 0 & \left(M'_1 (W) \right)^t & 1 & -\left(M'_1 (W) \right)^t v_1 (I) & -v_1 (I) \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0_{1 \times 3} & 0 & \left(M'_N (W) \right)^t & 1 & -\left(M'_N (W) \right)^t u_N (I) & -u_N (I) \\
\left(M'_N (W) \right)^t & 1 & 0_{1 \times 3} & 0 & -\left(M'_N (W) \right)^t v_N (I) & -v_N (I)
\end{bmatrix}$$

Unit: mm
Scale: vary

Unit: no
Scale: 0 or 1

Unit: mm*pixel
Scale: vary

Unit: pixel
Scale: the image size

Numerical instable: resulting problem when data contains significant noise

Solution: data normalization
Numerical Stability: Data Normalization

Replace $A$ by $A'$ by centering and normalizing each point, respectively.

- **Centering**: translate 2D/3D points wrt their centroid
- **Normalization**: scaling the coordinates by a scale factor
  - E.g., the scale factor can be computed as the inverse of average distance to the centroid

$$m' = H_{2D} \ m$$

$$H_{2D} = \begin{bmatrix} 1 & 0 & -\bar{m}_x \\ \frac{1}{s_{2D}} & \frac{1}{s_{2D}} & -\bar{m}_y \\ 0 & \frac{1}{s_{2D}} & \frac{1}{s_{2D}} \end{bmatrix}$$

$$\ M' = \ H_{3D} \ M$$

$$H_{3D} = \begin{bmatrix} 1 & 0 & 0 & -\bar{M}_x \\ \frac{1}{s_{3D}} & 0 & 0 & -\frac{\bar{M}_y}{s_{3D}} \\ 0 & \frac{1}{s_{3D}} & 0 & -\frac{\bar{M}_z}{s_{3D}} \\ 0 & 0 & \frac{1}{s_{3D}} & -\frac{\bar{M}_z}{s_{3D}} \end{bmatrix}$$