Computer Vision II: Multiple View Geometry (IN2228)

Chapter 03 Image Formation
(Part 1 Perspective Projection)

Dr. Haoang Li

03 May 2023  12:00-13:30
Announcement before Class

Today, we will have the exercise session about Mathematical Background

- Time: from 16:00 to 18:00
- Room: 102, Hörsaal 2, "Interims I" (5620.01.102)
Explanations before Class

- Clarification of labels in semantic segmentation

Images presented in our previous class

Different tasks
Explanations before Class

- Clarification of labels in semantic segmentation

The prediction results depend on the type of the ground truth.

An example

Input RGB image

Ground truth

Prediction

Neural network (a “fitter”) for label prediction

For more knowledge, please attend the course “Computer Vision III: Detection, Segmentation and Tracking” provided by Dr. Nikita Araslanov
Explanations before Class

➢ Clarification of General Pipeline

Model selection -> Data fitting (parameter estimation)

- Model selection based on the type of data
  - $O(3)$
  - $SE(3)$

- Unknown-but-sought rotation $R$ and translation $t$

- Parameter estimation by minimizing the distances between correspondences

- Point correspondences
Explanations before Class

- Clarification of General Pipeline

Model selection -> Data fitting (parameter estimation)

We aim to find the optimal rotation and translation (parameters) to minimize the sum of correspondence distances (objective function).
Today’s Outline

- Recap on Digital Images
- Pinhole Camera
- Perspective Projection
Recap on Digital Images

- Pixel Intensity and RGB channels

Pixel Intensity with 8 bits ranges between [0,255]

Pixel intensity

RGB color space
... but there are also many other color spaces... (e.g., YUV)

RGB Channels

R

G

B
Recap on Digital Images

- From Light Signal to Electrical Signal

- Basic configuration

- **Aperture** controls the area over which light can enter your camera
- **Shutter speed** controls the duration of the exposure

Exposure can be explained as the amount of light collected by a camera center.
Recap on Digital Images

- From Light Signal to Electrical Signal

- Response function

The camera response function maps the log-exposure value (scene radiance) to the intensity levels in the input images.
Pinhole Camera

- Human Eye

- The human eye is a camera
  - **Pupil** corresponds to the “aperture” whose size is controlled by the iris
  - Photoreceptor cells in the **retina** correspond to the “film”
Pinhole Camera

- Converging Lens

✓ All rays parallel to the optical axis converge at the focal point

A thin converging lens focuses light onto the film

Camera and converging lens

Focal length: $f$
Pinhole Camera

- Pinhole Camera Model

- The relationship between the image and object

\[
- \frac{x}{X} = \frac{f}{Z} \Rightarrow x = -f \frac{X}{Z}
\]

Image coordinates

3D coordinates

Similar triangle
Pinhole Camera

- Perspective Effects

- Far away objects appear smaller, with size inversely proportional to distance.
Pinhole Camera

- Perspective Effects

- Intersection of parallel lines in 2D
  - Parallel lines intersect at a “vanishing point” in the image
  - Vanishing points can fall both inside or outside the image
  - The connection between two horizontal vanishing points is the horizon
Pinhole Camera

- Perspective Effects

- Vanishing directions
  - A vanishing direction is defined by the connection between a vanishing point and camera center.
  - Vanishing direction is parallel to a 3D dominant direction.
  - Vanishing direction in 3D correspond to vanishing line in 2D.
Pinhole Camera

“Front” Image Plane

For convenience, the image plane is usually represented in front of the lens, such that the image preserves the same orientation (i.e. not flipped)

Flipped image in the pinhole camera model
Illustration of virtual (upright) image plane
Pinhole Camera

- “Front” Image Plane

Illustration of image planes behind or in front of lens

Application to structure from motion (non-flipped images)
Pinhole Camera

Field of View (FOV)

FOV is the **angular portion** of 3D scene seen by the camera
Pinhole Camera

- Field of View (FOV)

- FOV is inversely proportional to the focal length

Relationship between FOV and focal length

Short focal length & large FOV

Long focal length & small FOV
Pinhole Camera

- Field of View (FOV)

- Mathematical relation between field of view $\theta$, image width $W$, and focal length $f$:

  $$\tan \frac{\theta}{2} = \frac{W}{2f} \rightarrow f = \frac{W}{2} \left[ \tan \frac{\theta}{2} \right]^{-1}$$

- We can also define the FOV angle by image height.

Image plane
Perspective Projection

- Recap on Homogeneous Coordinates

✔ For ease of computation/representation
  
  - **3D Point**
    
    | Homogeneous | Cartesian |
    |-------------|----------|
    | (1,2,3)     | (1 2 3)  |
    | (2,4,6)     | (2 4 6)  |
    | (4,8,12)    | (4 8 12) |
    | ...         | ...      |
    | (1α,2α,3α)  | (1α 2α 3α)|

- **3D Line (Plucker Coordinates)**
  
  v: direction of 3D line (typically a unit vector)
  n: normal of projection plane

  \[
  n = Q \times v \\
  ||n|| = d \times ||v||
  \]
Perspective Projection

- Basic Knowledge
  - $C$: optical center, i.e., center of the lens, i.e., center of projection
  - $X_c, Y_c, Z_c$: axes of the camera frame
  - $Z_c$: optical axis (principal axis)
  - $O$: principal point, i.e., intersection of optical axis and image plane

Note: principal point is not exactly the image center (will be introduced later)
Perspective Projection

Perspective Projection vs. Parallel Projection

✓ Perspective Projection
  • Size varies inversely with distance – looks realistic
  • Parallel lines do not (in general) remain parallel

✓ Parallel Projection
  • Good for exact measurements
  • Parallel lines remain parallel
  • Less realistic looking
Perspective Projection

From Camera Frame to Image Coordinates

A 3D point $P_c = [X_c, Y_c, Z_c]^T$ in the camera frame is projected to $p = (x, y)$ onto the image plane.

Based on similar triangles

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$

$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$
Perspective Projection

- From Image Coordinates to Pixel Coordinates

- Let \( O = (u_0, v_0) \) be the pixel coordinates of the camera optical center
- Let \( k_u, k_v \) be the pixel conversion factors (conversion between mm and pixels)

Given image Coordinates \((x, y)\), we compute the Pixel Coordinates \((u, v)\) as

\[
\begin{align*}
    u &= u_0 + k_u x \\
    v &= v_0 + k_v y
\end{align*}
\]

\[
\begin{align*}
    u &= u_0 + \frac{k_u f X_c}{Z_c} \\
    v &= v_0 + \frac{k_v f Y_c}{Z_c}
\end{align*}
\]
Perspective Projection

- From Image Coordinates to Pixel Coordinates

\[
\begin{align*}
    u &= u_0 + k_u x \\
    v &= v_0 + k_v y
\end{align*}
\]

Expressed by mm

\[
\begin{align*}
    u &= u_0 + \frac{k_u X_C}{Z_C} \\
    v &= v_0 + \frac{k_v Y_C}{Z_C}
\end{align*}
\]

Focal lengths (expressed in pixels)

\[
\begin{align*}
    u &= u_0 + \frac{\alpha_u X_C}{Z_C} \\
    v &= v_0 + \frac{\alpha_v Y_C}{Z_C}
\end{align*}
\]
Perspective Projection

- **Intrinsic/Calibration Matrix**

Homogeneous coordinates

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Matrix form of perspective projection

- Focal length $a_u, a_v$
- Principal points $u_0, v_0$

Not equal due to conversion factor
Perspective Projection

- Intrinsic/Calibration Matrix

✓ In the past it was common to assume a skew factor in the pixel manufacturing process.
✓ However, the camera manufacturing process today is so good that we can safely assume skew factor = 0 and $\alpha u = \alpha v$ (i.e., square pixels).
Perspective Projection

- An Example of Intrinsic Parameters

Most widely-used SLAM datasets, e.g., TUM RGBD dataset provide intrinsic parameters calibrated beforehand (calibration will be introduced later).

<table>
<thead>
<tr>
<th>Camera</th>
<th>fx</th>
<th>fy</th>
<th>cx</th>
<th>cy</th>
<th>d0</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ROS default)</td>
<td>525.0</td>
<td>525.0</td>
<td>319.5</td>
<td>239.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Freiburg 1 RGB</td>
<td>517.3</td>
<td>516.5</td>
<td>318.6</td>
<td>255.3</td>
<td>0.2624</td>
<td>-0.9531</td>
<td>-0.0054</td>
<td>0.0026</td>
<td>1.1633</td>
</tr>
<tr>
<td>Freiburg 2 RGB</td>
<td>520.9</td>
<td>521.0</td>
<td>325.1</td>
<td>249.7</td>
<td>0.2312</td>
<td>-0.7849</td>
<td>-0.0033</td>
<td>-0.0001</td>
<td>0.9172</td>
</tr>
<tr>
<td>Freiburg 3 RGB</td>
<td>535.4</td>
<td>539.2</td>
<td>320.1</td>
<td>247.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Image resolution: 640*480 pixels

https://cvg.cit.tum.de/data/datasets/rgbd-dataset/file_formats
Perspective Projection

➢ From World Frame to Pixel Coordinates

Coordinate systems
• Camera frame
• Image coordinates
• Pixel coordinates
• World frame

Camera parameters
• Intrinsic parameters
• Extrinsic parameters
Perspective Projection

- Projection Matrix

From the world frame to the camera frame

$$X_c = RX_w + t$$

Rigid transformation (extrinsic parameters)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Matrix form

More compact form

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Extrinsic Parameters
Perspective Projection

- Projection Matrix

- Rigid transformation

\[
\begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix} =
\begin{bmatrix}
r_{11} & r_{12} & r_{13} & t_1 \\
r_{21} & r_{22} & r_{23} & t_2 \\
r_{31} & r_{32} & r_{33} & t_3
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix} =
\begin{bmatrix}
R \\
T
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
\]

- Perspective projection (camera frame)

\[
\lambda \begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = K \begin{bmatrix}
R \\
T
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
\]
Perspective Projection

- Computer Vision vs Computer Graphics

- Pipeline
  - Transforms the view volume, i.e., the pyramidal frustum to the canonical view volume, i.e., normalized device coordinates (NDC).
  - Linearly expand the XOY plane of NDC to screen/image plane.
Perspective Projection

- Computer Vision vs Computer Graphics

✓ Step 1: Convert perspective frustum to NDC space

\[
\begin{bmatrix}
 x_{\text{clip}} \\
 y_{\text{clip}} \\
 z_{\text{clip}} \\
 w_{\text{clip}}
\end{bmatrix}
= P
\begin{bmatrix}
 x_{\text{eye}} \\
 y_{\text{eye}} \\
 z_{\text{eye}} \\
 w_{\text{eye}}
\end{bmatrix}
\]

where

\[
P = \begin{bmatrix}
 \frac{2n}{r-l} & 0 & \frac{-r+l}{r-l} & 0 \\
 0 & \frac{2n}{t-b} & \frac{-t+b}{t-b} & 0 \\
 0 & 0 & \frac{f+n}{f-n} & \frac{2fn}{f-n} \\
 0 & 0 & 1 & 0
\end{bmatrix}
\]

NDC coordinates

\[
\begin{bmatrix}
 x_{\text{ndc}} \\
 y_{\text{ndc}} \\
 z_{\text{ndc}}
\end{bmatrix}
= \begin{bmatrix}
 x_{\text{clip}} / w_{\text{clip}} \\
 y_{\text{clip}} / w_{\text{clip}} \\
 z_{\text{clip}} / w_{\text{clip}}
\end{bmatrix}
\]

XOY plane of NDC

Perspective matrix

Homogeneous coordinates of a 3D point
Perspective Projection

- Computer Vision vs Computer Graphics

- Step 2: From NDC to screen space

Clip the content outside the NDC coordinates

Convert clipped NDC coordinates to screen coordinates
Perspective Projection

- Line Projection

✓ Two-step computation method
  • Coordinates of 2D endpoints (homogeneous)
    \[ A' = KA \]
    \[ B' = KB \]

  • Coordinates of 2D line (homogeneous)
    \[ l = A' \times B' \]

Intrinsic matrix

Vectors of \( l \) and \( p \) are orthogonal

\[ \ell = (l_1, l_2, l_3) \]

\[ \ell_1 \bar{x} + \ell_2 \bar{y} + \ell_3 \bar{z} = 0 \]

\[ \bar{p} = (\bar{x}, \bar{y}, \bar{z}) \]

such that \[ \ell^T \bar{p} = 0 \]

Projectors plane
Perspective Projection

- Line Projection

✓ One-step computation method
  • Coordinates of line (homogeneous)

\[ l = \mathcal{K} n \]

3D vector

\[ \mathcal{K} = \begin{bmatrix} f_y & 0 & 0 & 0 \\ 0 & f_x & 0 & 0 \\ -f_y x_0 & -f_x y_0 & f_x f_y & 0 \end{bmatrix} \]

Intrinsic matrix for line projection

\[ K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \]
Perspective Projection

Relationship between Points, Lines, and Planes

Some important conclusions

- Homogeneous coordinates of a plane
- A point P in homogeneous coordinates \((X, Y, Z, 1)\) lies on a plane

A four-dimensional vector \((A, B, C, D)\) represents the normal of a plane. The dot product of the normal \((\bar{n})\) and the projection \(\vec{P_0P}\) onto the normal is zero:

\[
\bar{n} \cdot \vec{P_0P} = 0
\]

The equation of the plane in homogeneous coordinates is:

\[
\begin{align*}
A(x-x_0) + B(y-y_0) + C(z-z_0) &= 0 \\
Ax + By + Cz &= D
\end{align*}
\]

The dot product of the normal and the point P gives the equation:

\[
P^T \bar{n} = 0
\]
Perspective Projection

- Relationship between Points, Lines, and Planes

✓ Some important conclusions

- Projection plane computed by image line

  \[ \pi_L = P^T l_L \in \mathbb{R}^4 \]

  Projection matrix (3*4)

- Intersection between a 3D line and a 3D plane

  \[ D = L \pi \]

  Homogeneous coordinates

  \[ L = \begin{bmatrix} [n]_\times & v \\ -v^T & 0 \end{bmatrix} \]

  Plucker matrix

  \[ L = (n^T, v^T)^T \]

  Plucker coordinates
Perspective Projection

- Normalized Image

A virtual image plane with **focal length equal to 1 unit** and **origin of the pixel coordinates at the principal point**.
Perspective Projection

- Normalized Image

- Computation of normalized coordinates

\[
\begin{bmatrix}
\bar{u} \\
\bar{v} \\
1
\end{bmatrix} = K^{-1}\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & -u_0 \\
\alpha & 1 & \frac{v_0}{\alpha} \\
0 & \alpha & -\frac{v_0}{\alpha} \\
1 & 0 & 1
\end{bmatrix}\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
u - u_0 \\
v - v_0 \\
1
\end{bmatrix}
\]
Perspective Projection

➢ Normalized Image

Multiply both terms of the perspective projection equation in camera frame coordinates by $K^{-1}$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \Rightarrow \lambda K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \Rightarrow \lambda K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Basic projection

Normalized image coordinates

$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/\alpha & 0 & -u_0/\alpha \\ 0 & 1/\alpha & -v_0/\alpha \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u-u_0/\alpha \\ v-v_0/\alpha \\ 1 \end{bmatrix}$$

Geometric meaning?

Collinearity between 3D vectors in camera frame
Perspective Projection

Geometric constraints of points

- Parallelism of ray directions

\[ \mathbf{d}_i^X \propto \mathbf{d}_i^X \Rightarrow \mathbf{K}^{-1} \mathbf{x}_i \propto [\mathbf{R} \quad \mathbf{t}] \mathbf{X}_i \]

3D vector 3D vector

“\( \propto \)” represents equality regardless of scale, i.e., two vectors are parallel, which leads to the cross product of 0.

- A 3*3 skew-symmetric matrix has the rank of 2, so each 3D-2D point correspondence provide two constraints.
Perspective Projection

Geometric constraints of lines

Parallelism of normals of projection plane

\[ n_j^L \propto n_j^L \Rightarrow K^{-1} l_j \propto [R \ [t] \times R] L_j \]

3D vector 3D vector

First 3 elements of Plucker coordinates

- Similarly, 3*3 skew-symmetric matrix has the rank of 2, so each 3D-2D line correspondence provide two constraints.

- How many points and/or lines should we use to compute 6-DOF camera pose?
Perspective Projection

- Geometric constraints of lines

- An alternative expression of line constraint

  - 3D line direction is orthogonal to the normal of projection plane. (one constraint)
  
  - The direction defined by a 3D point lying on the 3D line and the origin is orthogonal to the normal of projection plane. (one constraint)

3D-2D line correspondences $\{(L_k, l_k)\}_{k=1}^3$
Perspective Projection

- Normalized Image

- Applications to geometric constraints of lines
  - Point and Line (Ray-Point-Ray Structure)

\[
\begin{align*}
\mathbf{n}_x &= \mathbf{p} \times \mathbf{d}_x \\
\mathbf{n}_y &= \mathbf{p} \times \mathbf{d}_y \\
\mathbf{n}_x' &= \mathbf{p}' \times \mathbf{d}'_x \\
\mathbf{n}_y' &= \mathbf{p}' \times \mathbf{d}'_y
\end{align*}
\]

Two-view configuration of ray-point-ray structure

\[
\mathbf{e}_x \propto \mathbf{n}'_x \times \mathbf{Rn}_x \\
\mathbf{e}_y \propto \mathbf{n}'_y \times \mathbf{Rn}_y
\]

\[
\mathbf{e}_x^\top \mathbf{e}_y = \cos \alpha \cdot ||\mathbf{e}_x|| \cdot ||\mathbf{e}_y||
\]

Angle between two lines
Perspective Projection

- Spherical Projection

- Planar projection vs. spherical projection

Spherical projection has a larger FOV than planar projection
Perspective Projection

- Spherical Projection

- Obtaining a panorama with a 360 degree field of view

Omnicamera

Equirectangular panorama in a spherical projection.
Perspective Projection

- Spherical Projection

✓ Pipeline of spherical image generation

- Map 3D point \((X,Y,Z)\) onto sphere

\[
(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} (X, Y, Z)
\]

- Convert to spherical coordinates

\[
\begin{align*}
  r &= \sqrt{x^2 + y^2 + z^2} \\
  \theta &= \tan^{-1} \left( \frac{y}{x} \right) \\
  \phi &= \cos^{-1} \left( \frac{z}{r} \right)
\end{align*}
\]

Azimuth
\[
\theta \in [0, 2\pi)
\]

Polar angle
\[
\phi \in [0, \pi]
\]
Perspective Projection

- **Spherical Projection**

  - Pipeline of spherical image generation
    - $s$ defines size of the final image (often convenient to set $s = \text{camera focal length}$)

  $$(\tilde{x}, \tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c, \tilde{y}_c)$$  
  Displacement of origin

  Linear mapping

  ![Diagram of spherical projection](image-url)
Perspective Projection

- Spherical Projection

  ✓ Difference between latitude and polar angle

  ✓ Difference between mathematical and physical representation
Perspective Projection

- Spherical Projection

- Cube-based representation

Inscribed sphere

Cross-shaped expansion (more commonly used in computer graphics)
Perspective Projection

- Spherical Projection

- Lines and vanishing points
Perspective Projection

- Other Expressions of Sphere

- Spatial distortion due to equi-rectangular representation

- Yellow squares on both sides represent the same surface areas on the sphere.
- The area of Antarctica seems large.
Perspective Projection

Other Expressions of Sphere

- Icosahedral representation

We extrude all the vertices of icosahedron sub-faces to the unit sphere, obtaining the icosahedral spherical representation.

![Equi-angular discretization](a)

![Icosahedron](b)

![Icosahedral spherical representation](c)

Expansion of icosahedral sphere
Summary

- Recap on Image Processing
- Pinhole Camera
- Perspective Projection
Thank you for your listening!
If you have any questions, please come to me :-)
