Coordinate Transformation & Image Formation

Ziheng

2017/09/28

Computer Vision I
Recitation – 3

Coordinate Transformation

- Rigid-body motion
  - Rotation
  - Translation
  - Homogenous Representation

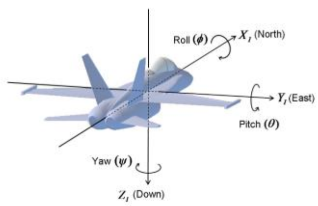
Transform coordinate in world frame to cam frame

How to parameterize rigid-body motion?

Coordinate Transformation

Computer Vision I
Recitation – 3

- Rotation – Euler angles (e.g. IMU)



$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

$$\mathbf{R}_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix}$$

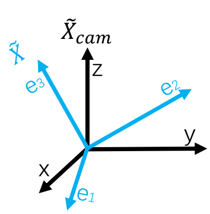
$$\mathbf{R}_y(\theta_y) = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix}$$

$$\mathbf{R}_z(\theta_z) = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Coordinate Transformation

Computer Vision I
Recitation – 3

- Rotation – Basis Transformation (e.g. Compass, Encoder)



$$R = [\boldsymbol{e}_{WC} \quad \boldsymbol{e}_{WC} \quad \boldsymbol{e}_{WC}]$$
$$= [\boldsymbol{e}_{CW} \quad \boldsymbol{e}_{CW} \quad \boldsymbol{e}_{CW}]^T$$

Where $\mathbf{e}_{wc}$ $\mathbf{e}_{wc}$ $\mathbf{e}_{wc}$ is basis of world frame, **represented in cam coordinate**

Coordinate Transformation

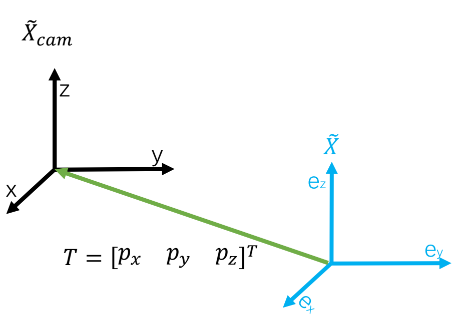
- Rotation group $SO(n)$ – DoF: $\frac{n(n-1)}{2}$

  The set of $n \times n$ orthogonal matrices forms a group O(n), known as the orthogonal group. The subgroup SO(n) consisting of orthogonal matrices with determinant **+1** is called the special orthogonal group, and each of its elements is a **special orthogonal matrix**. As a linear transformation, **every special orthogonal matrix acts as a rotation**.

Coordinate Transformation

Computer Vision I
Recitation – 3

- Translation



$$[x \quad y \quad z]^T \Longrightarrow [x \quad y \quad z]^T + T$$

Where $p_x \quad p_y \quad p_z$ is translation **along cam coordinate**

Altogether

Computer Vision I
Recitation – 3

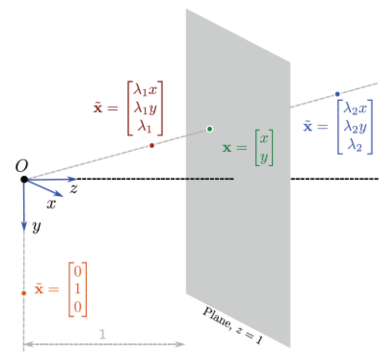
- Homogeneous Coordinates

Mapping to homogeneous coordinates

$$\begin{cases} 2\mathrm{D}: & (x,y)^T \rightarrow (x,y,1)^T \\ 3\mathrm{D}: & (x,y,z)^T \rightarrow (x,y,z,1)^T \end{cases}$$

Mapping back from homogeneous coordinates

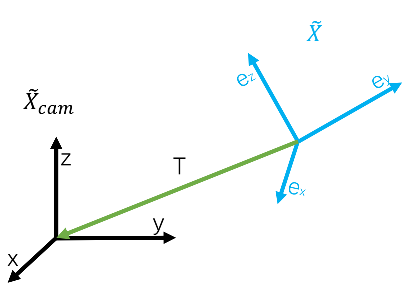
$$\begin{cases} 2\mathrm{D}: & (x,y,w)^T \rightarrow (\frac{x}{w},\frac{y}{w})^T \\ 3\mathrm{D}: & (x,y,z,w)^T \rightarrow (\frac{x}{w},\frac{y}{w},\frac{z}{w})^T \end{cases}$$



Altogether

Computer Vision I
Recitation – 3

- Rotation & Translation (Extrinsic Parameters)



$$\tilde{X} = RX + T = \begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$R = [\, e_x \quad e_y \quad e_z]^T$$

DoF: 3+3=6

Altogether

- Rigid motion group $SE(n)$– DoF: $\frac{n(n-1)}{2}+n \longrightarrow \frac{n(n+1)}{2}$

  The Euclidean group E(n) is the symmetry group including the **orientation-reversing isometries** (like reflections, glide reflections and improper rotations) . There is a subgroup of the direct isometries, i.e., isometries preserving orientation also called **rigid motions.** These include the translations, and the rotations, which together generate a **special Euclidean group**, and denoted SE(*n*).

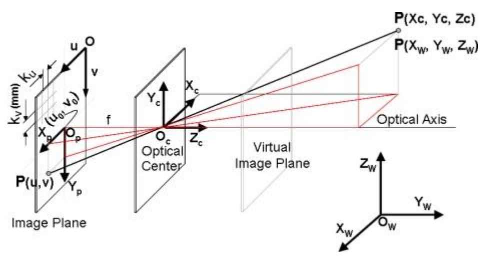
Image Formation

- Camera Models
  - Ideal Perspective Camera
  - Intrinsic Parameters
  - General Perspective Camera
  - (Thin) Lens Camera
  - Camera with Distortion

Image Formation

Computer Vision I
Recitation – 3

- Ideal Perspective Camera



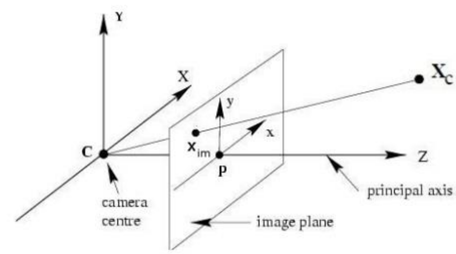
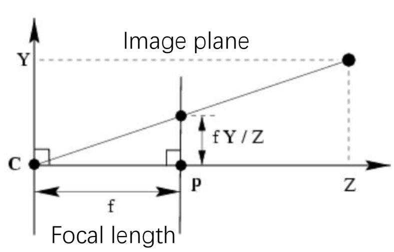


Image Formation

Computer Vision I
Recitation – 3

- Ideal Perspective Camera



By similar triangles

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} fX_0/Z_0 \\ fY_0/Z_0 \\ f \end{bmatrix}$$

Dropping third coordinate

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} fX_0/Z_0 \\ fY_0/Z_0 \end{bmatrix}$$

Image Formation

Computer Vision I
Recitation – 3

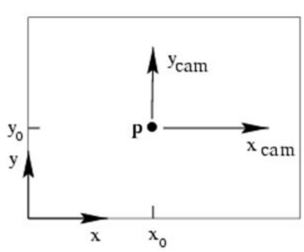
- Ideal Perspective Camera

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = K_f [\boldsymbol{I}_{3\times3}\ \boldsymbol{0}_{3\times1}] \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

Image Formation

Computer Vision I
Recitation – 3

- Intrinsic Parameters (from cam coordinate to image coordinate)



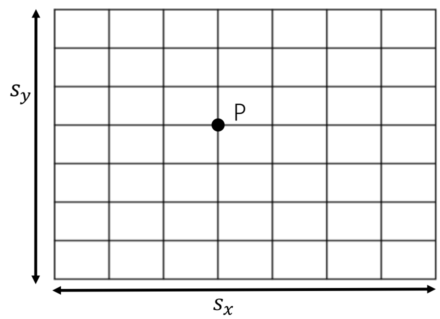
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} X_0 + p_x \\ Y_0 + p_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$P = [p_x, p_y]$ Principal-point offset in **image coordinate**

Image Formation

Computer Vision I
Recitation – 3

- Intrinsic Parameters (from cam coordinate to image coordinate)



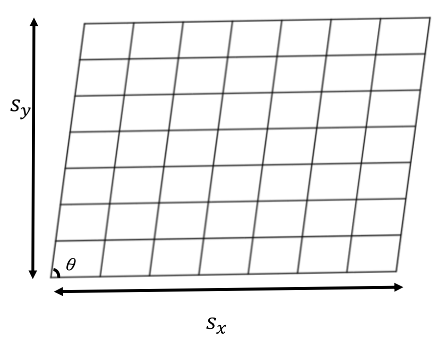
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_y & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Number of pixels along cam coordinate

Image Formation

Computer Vision I
Recitation – 3

- Intrinsic Parameters (from cam coordinate to image coordinate)



$s_\theta \sim \cot(\theta)$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & s_\theta & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Image Formation

Computer Vision I
Recitation – 3

- General Perspective Camera (from cam coordinate to image coordinate)

$$K \doteq K_s K_f = \begin{bmatrix} s_x & s_\theta & p_x \\ 0 & s_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & s_\theta & p_x \\ 0 & fs_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Calibration Matrix (Intrinsic Parameters) Standard Projection Matrix

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K\Pi_0 X = \begin{bmatrix} fs_x & s_\theta & p_x \\ 0 & fs_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} fs_x & s_\theta & p_x \\ 0 & fs_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda K^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Image Formation

Computer Vision I
Recitation – 3

- General Perspective Camera (from world coordinate to image coordinate)

$$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$ Extrinsic Parameters

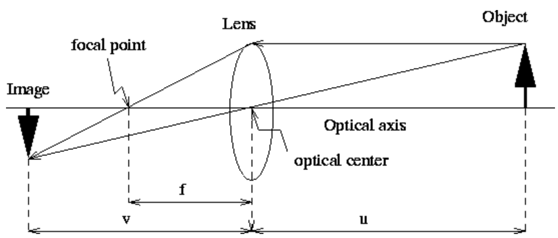
$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K\Pi_0 g X_0 = \begin{bmatrix} fs_x & s_\theta & p_x \\ 0 & fs_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} X_0 = K[R\ T]X_0$$ DoF: 5+6=11

Intrinsic Parameters Extrinsic Parameters

Image Formation

Computer Vision I
Recitation – 3

- (Thin) Lens Camera



$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

Image Formation

Computer Vision I
Recitation – 3

- (Thin) Lens Camera

Thin Lens => Paraxial Refraction Model

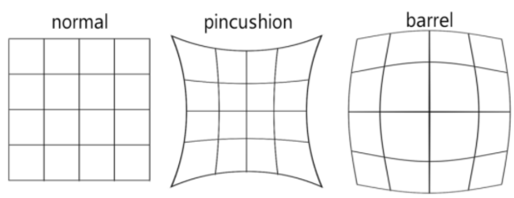
For the angle θ that incoming light rays make with the optical axis of the lens, the paraxial assumption substitutes θ for any place $\sin\theta$ is used. This approximation of θ for $\sin\theta$ holds as θ approaches 0.

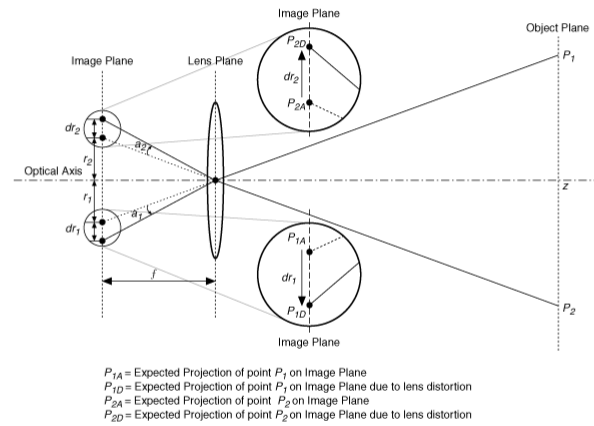
Image Formation

Computer Vision I
Recitation – 3

- Camera with Distortion

Practical Lens Camera => Radial Distortion





$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K[R\ T]X_0 \mapsto \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K \begin{bmatrix} f(r) & 0 & 0 \\ 0 & f(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} [R\ T]X_0 \text{ where } f(r) = 1 + a_1(x^2+y^2) + a_2(x^2+y^2)^2 + \cdots$$

Walk Around SIST-Building!

Computer Vision I
Recitation – 3

Coding Exercises

Computer Vision I
Recitation – 3

- Photo Competition
  - Design your own virtual camera with different intrinsic parameters
  - Choose appropriate viewpoint
  - Take photos to test your cameras
  - And hand in your favorite picture to us
- Video Advertisement Competition
  - Design scenic route as well as camera poses, and use your own virtual camera to make promotional video for SIST building

2017/10/11 23/24

Coding Exercises

Computer Vision I
Recitation – 3

- XYZ File – ASCII point cloud.

		X	Y	Z	R	G	B
5292	1491	-7.7413	-28.9407	-1.553	255	255	255
5292	1441	-7.2838	-29.0514	-1.5548	255	255	255
5292	1480	-7.6417	-28.9667	-1.5534	255	255	255
5292	1474	-7.5868	-28.9808	-1.5537	255	255	255
5292	1461	-7.4682	-29.0099	-1.5542	255	255	255
5292	1456	-7.4212	-29.0143	-1.554	255	255	255
5292	1451	-7.3757	-29.0264	-1.5542	255	255	255
5292	1453	-7.3811	-28.9716	-1.5515	255	255	255
5292	1446	-7.3239	-29.018	-1.5534	254	254	254
5292	1440	-7.2686	-29.0297	-1.5535	255	255	255
5292	1449	-7.3487	-28.9993	-1.5526	255	255	255
5293	1444	-7.2651	-28.8619	-1.554	255	255	255
5292	1435	-7.2288	-29.0669	-1.5551	255	255	255
5292	1429	-7.1762	-29.0856	-1.5557	254	254	254
5292	1437	-7.2397	-29.0336	-1.5535	255	255	255
5293	1439	-7.2164	-28.8619	-1.5536	255	255	255
5293	1434	-7.1736	-28.8838	-1.5544	255	255	255

```python
from tqdm import tqdm

# line count
num_points = sum(1 for line in open('abc.xyz'))
fp = open('abc.xyz')
keys = ['x', 'y', 'z', 'r', 'g', 'b']
# load .xyz pointcloud with progress bar
points = []
for line in tqdm(fp, total=num_points, disc='loading pointcloud'):
    # ignore empty lines
    if not line:
        continue
    row_list = line.split()
    elements = list(map(float, row_list[2:5])) + list(map(int, row_list[5:]))
    points.append(dict(zip(keys, elements)))
```

2017/10/11 24/24

Coding Exercises

- Due: 21:00(GMT+8) 28, Oct
  - Optional Exercises
  - No restrictions on languages & using of existing libraries
  - Extra bonus for excellent works
  - Exhibition for all submissions
  - Generous Extra bonus for winners

Download Point Cloud:
http://10.19.124.26:8000/d/c7a4b04f231d412db46c/

SIST_000_oct0_01.bin	Binary point cloud (downsampled)
SIST_000_oct0_01.xyz	ASCII point cloud (downsampled)
SIST_scan_000.xyz	ASCII point cloud (raw)