EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2020**

Lecture 1: Rigid Body Motion and Imaging Geometry

Scribes: Andrew Fearing

1.1 Announcements

- Professor Sastry's OH canceled today (2/23/2021).
- Similar to 106A, effort, participation, and altruism (EPA) points have been introduced. Examples include: attending/asking questions in lecture/discussion, submitting discussion questions for paper presentations, and helping others on Piazza/Discord.
- Game night at 5:00pm PST on Friday, 2/2. Vote for games at `https://forms.gle/Pwq8V6vYQigodMPt5`.

1.2 Review

Last time we did an overview of what vision is about. Robotics, especially fully autonomous robotics, is a fully closed-loop system. Perception of the environment is essential to closing the loop. For robotic systems, you need to understand perception and control very well.

Last time we saw some example applications: autonomous cars, helicopters, and how vision plays an important role. Since then, mechatronics and computation has improved but vision is still far inferior to human beings. To understand vision, we need to understand the fundamental mathematical principle behind images. This will be the topic for the next few lectures.

1.3 3D Euclidean Space & Rigid-Body Motion

1.3.1 Cartesian Coordinate Frame

In order to describe a point in the physical world we specify xyz coordinates on orthogonal axes. We use a right-hand system. Once we equip the world with this coordinate system, then every point p in $\mathbb{R}^3$ gets assigned coordinates.

There is also a notion of vectors. A vector is also 3D but geometrically a different entity. A vector is specified by two points: a start q and an end p. A vector gives you the direction q to p and also a magnitude. It is sometimes helpful to think of a point as a vector from the origin. However, an important distinction between points and vectors is that vectors are fixed. p never changes unless you change the coordinate frame. When we talk about vectors, we usually refer to a free vector. If $p' - q' = p - q$, then $\boldsymbol{v}' = \boldsymbol{v}$.

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \boldsymbol{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

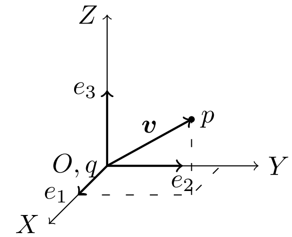

Figure 1.1: Axes and vector in $\mathbb{R}^3$.

Vectors can also be defined in terms of a linear combination of orthogonal vectors.

$$\boldsymbol{v} = \sum_{1}^{3} a_i \boldsymbol{e}_i \in \mathbb{R}^3$$

1.3.2 Vector Operations

The inner product of two vectors is a scalar.

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \ \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \in \mathbb{R}^3$$

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\intercal \boldsymbol{v} = u_1 v_1 + u_2 v_2 + u_3 v_3 \in \mathbb{R}$$

The magnitude of a vector is given by

$$\|\boldsymbol{u}\| = \sqrt{\boldsymbol{u}^\intercal \boldsymbol{u}} = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

The cosine of the angle between two vectors can be found.

$$\cos\theta = \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|\boldsymbol{u}\| \cdot \|\boldsymbol{v}\|}$$

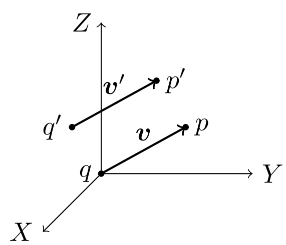

Figure 1.2: Free vector in $\mathbb{R}^3$. Note that $\boldsymbol{v} = \boldsymbol{v}'$.

There is also a notion of the cross product. The cross product of two vectors generates a vector orthogonal to those vectors, according to the right hand rule. In $\mathbb{R}^3$, the cross product can be defined in terms of the product of the skew-symmetric matrix of a vector and another vector. The length of the cross product vector is equal to the area of the parallelogram spanned by $\boldsymbol{u}$ and $\boldsymbol{v}$.

$$\boldsymbol{u} \times \boldsymbol{v} = \hat{\boldsymbol{u}} \boldsymbol{v}, \quad \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^3$$

$$\hat{\boldsymbol{u}} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

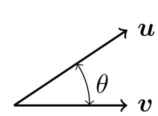

Figure 1.3: Angle of two vectors.

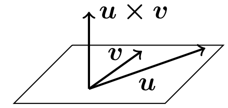

Figure 1.4: Cross product of two vectors in $\mathbb{R}^3$.

1.3.3 Rotation

Next we describe motion. We can always associate a world space. The world space has coordinates and it never moves. Then we associate the object with its own coordinate frame. To describe a rotation, all we need to know is where the object axes are after the rotation. The axes are not independent. They must always be orthogonal. We can define a rotation matrix in $\mathbb{R}^{3\times 3}$ such that

$$R = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$
$$R^\intercal R = I_{3,3}$$
$$\det(R) = +1$$

All the rotation matrices that satisfy these properties are the special orthogonal group $SO(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\intercal R = RR^\intercal = I_{3,3}, \det(R) = 1\}$. Rotation matrices have special properties. They preserve inner product (or length), and they preserve cross product (or orientation).

$$\|R\boldsymbol{v}\|_2^2 = \boldsymbol{v}^\intercal R^\intercal R\boldsymbol{v} = \|\boldsymbol{v}\|_2^2,\ \forall \boldsymbol{v} \in \mathbb{R}^3$$

$$R\boldsymbol{u} \times R\boldsymbol{v} = R(\boldsymbol{u} \times \boldsymbol{v})$$

$$\widehat{R\boldsymbol{u}} = R\hat{\boldsymbol{u}}R^\intercal,\ \forall \boldsymbol{u} \in \mathbb{R}^3,\ R \in SO(3)$$

1.3.4 Reflection

For robotics we only care about transformations that are physically realizable according to the right-hand rule. However, for vision we need symmetry. So we need the reflection matrix, which has similar properties to the rotation matrix except that its determinant is negative.

$$R = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$
$$R^\intercal R = I_{3,3}$$
$$\det(R) = -1$$

Reflection matrices are like rotation matrices, except they do not preserve orientation

$$R\boldsymbol{u} \times R\boldsymbol{v} = -R(\boldsymbol{u} \times \boldsymbol{v})$$

These reflection matrices make up the other half of orthogonal group $O(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\intercal R = RR^\intercal = I_{3,3}, \det(R) = \pm 1\}$.

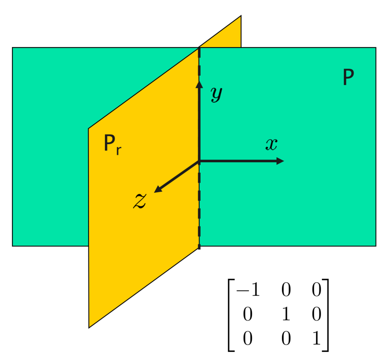

Figure 1.5: Reflection, from MASKS 2004.

1.3.5 Rotation and Translation

In the real world we have both rotation and translation. We already know how to do translation from vectors: $T = O' - O$. We just need the center of the object frame and the center of the world frame. Then find the relative orientation to get the relative position $\boldsymbol{X}_c = R\boldsymbol{X}_w + T$. Velocites are related by $\dot{\boldsymbol{X}}_c = \hat{\omega}\boldsymbol{X}_c + v$

Because $X_c = RX_w + T$ is affine, we use homogeneous coordinates. We augment the points and vectors from $\mathbb{R}^3$ to $\mathbb{R}^4$. Points are augmented with a 1 and vectors are augmented with a 0.

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \to \boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \in \mathbb{R}^4, \qquad \boldsymbol{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \to \boldsymbol{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ 0 \end{bmatrix} \in \mathbb{R}^4$$

Then homogeneous coordinates and velocities can be related in the form $\boldsymbol{b} = A\boldsymbol{x}$.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{Z}_c \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

1.4 Image Formation

1.4.1 Pinhole Camera Model

The problem of pinhole projection has been a fascination since ancient times. Mathematically, light travels in straight lines from a 3D object and is then projected on a plane. From trigonometry, the image on the 2D plane is amplified by the focal length

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \to \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

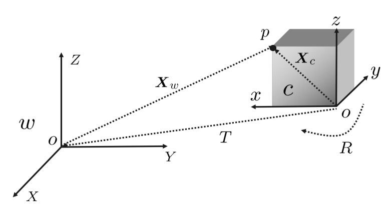

Figure 1.6: Rotation and translation, from MASKS 2004.

So we can take the 2D coordinates and make them into homogeneous coordinates.

$$\boldsymbol{x} \to \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}, \boldsymbol{X} \to \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

So in homogeneous coordinates,

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\boldsymbol{X}}$$

Then we can decompose the matrix into

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K_f \Pi_0$$

where $\Pi_0 \in \mathbb{R}^{3\times 4}$ is often referred to as the *standard (or "canonical") projection matrix.*

1.4.2 Pixel Coordinates

Then we can get to pixel coordinates. Because Z is often unknown, we substitute in an arbitrary positive number *lambda* to get $\lambda \boldsymbol{x} = K_f \Pi_0 \boldsymbol{X}$ so that

$$\lambda \boldsymbol{x}' = \underbrace{K_s K_f}_{K} \Pi_0 \boldsymbol{X}$$

where $K = K_s K_f$ is the calibration matrix. We also define a projection matrix $\Pi = \left[K, 0\right] \in \mathbb{R}^{3\times 4}$. So our camera model is finally

$$\lambda \boldsymbol{x}' = K \Pi_0 \boldsymbol{X} = \Pi \boldsymbol{X}$$

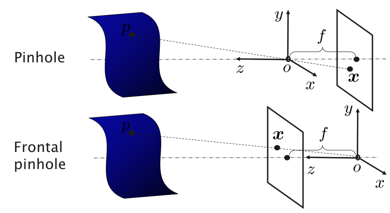

Figure 1.7: Pinhole camera models, from MASKS 2004.

1.4.3 Calibration Matrix and Camera Model

We want a nice camera where we can perform a linear transformation to correct for distortion. But what we get is often radial distortion, especially with a cheap camera. We have to correct this back to linear. Camera calibration is essential for robotics.

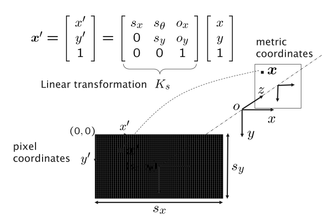

Figure 1.8: Pixel coordinates, from MASKS 2004.

1.4.4 Image of a Point

An image is given by a 3D projection onto an image plane. We only know the 2D coordinates on the image plane. We do not know the camera frame or the points in 3D. We might not even know the calibration matrix K. So we don't automatically know how to reconstruct the 3D world. Next lecture we'll talk about feature recognition and correspondence.

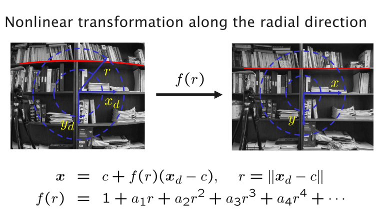

Figure 1.9: Radial distortion, from MASKS 2004.