CSE 252A: Rigid Transformations

Lecturer: David Kriegman

Scribed by Owen Jow on October 11, 2018

1 Coordinate Changes

So far, we've always assumed we were projecting points in camera coordinates. However, for many practical applications, those points will actually start in some other coordinate system (e.g. a *world* coordinate system) and we'll have to transform them into camera coordinates based on the position/orientation of the camera. As a result, we must be able to perform coordinate changes.

Notation

${}^{A}P$ - point/vector P in coordinate system A

${}^{B}_{A}R$ - rotation matrix from coordinate system A to coordinate system B

1.1 Rotation Matrices

A 3D rotation matrix (non-homogeneous coordinates) contains as its rows the coordinates of the *destination axes* in the *current frame*. Let i_A, j_A, k_A be vectors corresponding to the axes of frame A, and let i_B, j_B, k_B be vectors corresponding to the axes of frame B. Then

$$
{}^{B}_{A}R = \begin{bmatrix} — & {}^{A}i_B & — \\ — & {}^{A}j_B & — \\ — & {}^{A}k_B & — \end{bmatrix} = \begin{bmatrix} {}^{A}i_A \cdot {}^{A}i_B & {}^{A}j_A \cdot {}^{A}i_B & {}^{A}k_A \cdot {}^{A}i_B \\ {}^{A}i_A \cdot {}^{A}j_B & {}^{A}j_A \cdot {}^{A}j_B & {}^{A}k_A \cdot {}^{A}j_B \\ {}^{A}i_A \cdot {}^{A}k_B & {}^{A}j_A \cdot {}^{A}k_B & {}^{A}k_A \cdot {}^{A}k_B \end{bmatrix} = \begin{bmatrix} | & | & | \\ {}^{A}i_B & {}^{A}j_B & {}^{A}k_B \\ | & | & | \end{bmatrix}^T \begin{bmatrix} | & | & | \\ {}^{A}i_A & {}^{A}j_A & {}^{A}k_A \\ | & | & | \end{bmatrix}
$$

For example, the first row is the projection of i_B onto each of A's axes.

Rotation matrices R obey the equation $R^T R = I$, which is equivalent to saying rotation matrices are orthogonal. Since the product contains 1s on the diagonal and 0s elsewhere, the row/column vectors must be unit length and orthogonal to each other. It is also equivalent to saying $R^T = R^{-1}$, and that the determinant is ± 1. If the coordinate system is right-handed, the determinant is 1.

There are three degrees of freedom in a rotation matrix, which can again be seen from the $R^T R = I$ constraint. There are nine equations in $R^T R = I$ (one for each entry), but only six of them are independent, leaving three degrees of freedom (nine numbers, six constrained, three free).

Rotation matrices also form the **special orthogonal group SO(n)**[1] under the matrix product operation. For example, SO(3) is the group of 3D rotation matrices. Note that SO(n) is not a vector space because it is not closed under addition: if you add two rotation matrices, the result is not necessarily a rotation matrix.

[1] n is the dimension of the space.

There are multiple rotation parameterizations.

- **RPY.** Rotate around *fixed* axes.
- **Euler angles.** Rotate around *moving* axes.
- **Matrix exponential.** A mapping from skew-symmetric matrices onto rotation matrices.
- **Quaternions.** A generalization of complex numbers (one real part, three complex parts).
- **...**

1.2 Rigid Transformations

Between two coordinate frames, there can be a rotation and a translation. Thus, the transformation from coordinate system A to coordinate system B can be represented as the rigid transformation

$$^{B}P = {}^{B}_{A}R\,{}^{A}P + {}^{B}O_A$$

where $^{B}O_A$ is the translation in frame B between the two frames' origins. A rigid transformation is one which preserves distances between pairs of points, and includes rotations and translations.

We can write this as the 4×4 block matrix

$$\begin{bmatrix} {}^{B}_{A}R & {}^{B}O_A \\ \mathbf{0}^T & 1 \end{bmatrix}$$

2 World coordinates to image coordinates

We can write the full mapping from 3D world coordinates to 2D image coordinates as

$$\begin{aligned} \text{2D image point} = &\ (3 \times 3 \textit{ intrinsic matrix}) \\ &* (\textit{reduction of dimension from } 4 \textit{ to } 3) \\ &* (4 \times 4 \textit{ extrinsic matrix}) \\ &* \text{3D world point} \end{aligned}$$

Note: there are potentially many intrinsic parameters beyond those we have previously discussed. In addition to the focal length f, we might have parameters for translating the origin to a corner of the image, performing a conversion of units (e.g. from mm to px), orienting the image plane relative to the camera, or setting the pixel aspect ratio (as pixels are not always square).

We estimate intrinsic and extrinsic parameters through **camera calibration**, which will typically involve taking a picture of some known geometry.