Exercise 2

Ritu Yadav   @rhrk.uni-kl.de

Shalini Bani   @rhrk.uni-kl.de

Edgar Andrés Margffoy Tuay   margffoy@rhrk.uni-kl.de

TU Kaiserslauten   Augmented Vision

3D Computer Vision   November 30, 2017

1 Properties of Rotation Matrices

1.1 Proof that a Rotation Matrix satisfies Orthogonal Matrix constraints

$U^{-1} = U^T \Rightarrow UU^T = I$

$$U^T = U^{-1}$$
$$UU^T = \cancel{UU^{-1}}^{\,I}$$
$$\Rightarrow \boxed{UU^T = I} \quad \square \tag{1}$$

$UU^T = I \Rightarrow U^{-1} = U^T$

$$UU^T = I$$
$$\cancel{U^{-1}U}^{\,I}U^T = U^{-1}$$
$$\Rightarrow \boxed{U^T = U^{-1}} \quad \square \tag{2}$$

Given that the rotation matrix U is orthogonal, it must satisfy the property $U^T = U^{-1} \Leftrightarrow UU^T = I$. As it is possible to observe on (1) and (2), each one of the hypothesis implies each of the desired conclusions, therefore it is possible to conclude that a general rotation matrix is an orthogonal matrix.

1.2 Geometrical interpretation of a Transformation Matrix determinant

As the determinant of a 3×3 matrix represents the volume of the cube spanned by the rows of the matrix, it is possible to say that given an unitary basis matrix described by sides 1 and a transformation matrix T, the determinant of the transformation $\det(T)$, should inform about changes on the volume, as well on the length of the sides. If the determinant of the matrix is ±1, then its possible to conclude that the transformation preserves volumes and line lengths, whereas if the determinant is different from one, then the line lengths and the volume changes.

For instance, given a translation matrix $(3)^1$, it is possible to conclude that this transformation preserves line lengths and volumes as it is expected, *i.e.,* (4)

$$T = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\boxed{\det(T) = -1} \tag{4}$$

Other transformations that preserve line lengths and object volumes correspond to Rotations, and therefore Rotation and Translation transforms (5), as it can be deduced on (6).

$$T = \begin{bmatrix} 0 & -1 & 0 & a \\ 1 & 0 & 0 & b \\ 0 & 0 & -1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\boxed{\det(T) = -1} \tag{6}$$

Also, it is possible to show that scaling transformation matrices (7) modify line lengths and volumes, as their determinant is different from one (8).

$$T = \begin{bmatrix} 0 & -2 & 0 & a \\ 2 & 0 & 0 & b \\ 0 & 0 & -2 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$\boxed{\det(T) = 8} \tag{8}$$

2 Transformation Chain between 3D world coordinates and 2D image coordinates

In first instance, assuming the camera position is rectified with respect to the world coordinate frame, we define the projection of 3D world point to a 2D point on the image plane, as defined on (9), f' refers to the focal distance between the lensses' image plane and the aperture pinhole of the camera.

$$\underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\textbf{3D}} \Rightarrow f' \cdot \underbrace{\begin{bmatrix} x/z \\ y/z \end{bmatrix}}_{\textbf{2D}} \tag{9}$$

---

[1]Expressed in terms of homogeneous coordinates

As it can be observed, this transformation is not linear, which implies that it cannot be done only by multiplying the 3D coordinate by a transformation matrix that projects it into 2D. However, this can projection can be expressed as a linear operation using homogeneous coordinates (10), which consists on extending the representation of the point into a line on a higher dimension, *i.e.,* Add the number 1 as third/fourth coordinate to a 2D/3D vector/point.

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{10}$$

To represent a point on Cartesian Coordinates, back from Homogeneous Coordinates, it's only necessary to divide all the other coordinates by the last one (11).

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix} \tag{11}$$

By using Homogeneous Coordinates, we can rewrite the projection expression presented on (9), and present it as a linear transformation matrix on homogeneous coordinates (12). By doing so, it is possible to apply and collapse previous linear transformations, such as rotations, scaling and translations, storing all of them on a single convenient matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f' \end{bmatrix} \rightarrow f' \cdot \begin{bmatrix} x/z \\ y/z \end{bmatrix} \tag{12}$$

It is necessary to observe that it was assumed previously that both the camera and the world share the same coordinate frame. However in the reality, these two coordinate frames are on different positions, therefore it is necessary to introduce a novel matrix R_{wc}^t that translates (T_{wc}) and rotates (R_{wc}) a point from the world's coordinate frame to the camera's one. This transformation must be done before the projection to the image plane on 2D.

$$R_{wc}^t = \begin{bmatrix} R_{wc} & T_{wc} \\ 0^T & 1 \end{bmatrix} \tag{13}$$

So far, the projection steps have been given in terms of the extrinsic camera parameters, which refer to the conversion between world coordinates points to the corresponding points on the camera reference. Also, the projection between 3D world coordinates to (x, y, z) to pixels (u, v) was given on terms of a single ideal parameter f'. However, it is necessary to understand that this step depends on a set of parameters intrinsic to the camera itself, such as the size of each pixel on the sensor matrix (14), the image sensor matrix position (coordinate origin) (15), and also possible sensor matrix rotation due to assembly defects (16). Using homogeneous coordinates, it is possible to express the intrinsic camera parameters matrix K as in (17).

$$u = \alpha \frac{x}{z} \qquad\qquad v = \beta \frac{y}{z} \tag{14}$$

$$u = \alpha \frac{x}{z} + u_0 \qquad\qquad v = \beta \frac{y}{z} + v_0 \tag{15}$$

$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0 \qquad v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0 \tag{16}$$

$$K = \begin{bmatrix} \alpha & -\alpha \cot(\theta) & u_0 & 0 \\ 0 & \beta / \sin(\theta) & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{17}$$

In summary, to transform a 3D world coordinate point $\vec{p}_w$ to a 2D pixel point on the camera image $\vec{p}$, it is essential to transform first by the extrinsic parameters, and then by the intrinsic parameters. By using homogeneous coordinates, it's possible to merge both parameters onto a single matrix M, as shown on (18).

$$\vec{p} = K R_{wc}^{t} \vec{p}_w$$
$$\Rightarrow \boxed{\vec{p} = M \vec{p}_w} \tag{18}$$

References

[1] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. ISBN: 0130851981.

[2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[3] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.