BASIC CONCEPTS OF ROTATIONS AND TRANSLATIONS WITH PROCESSING

BY ANTONIO BERNARDINI

Table of contents

1 Rotations and translations

1.1 Elementary rotations around the x, y and z axes

An elementary rotation is the rotation of SR around one of the 3 coordinate axes of SR′. Let's see the corresponding rotation matrices in the 3 cases:

1. Elementary rotation around $\vec{e_z}'$ by an angle α:

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. Elementary rotation around $\vec{e_y}'$ by an angle β:

$$R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

3. Elementary rotation around $\vec{e_x}'$ by an angle γ:

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$$

1.2 Example

Now let's imagine taking a vector $\vec{v} = (\begin{smallmatrix} v_x & v_y & v_z \end{smallmatrix})^\top$ and applying any elementary rotation matrix as follows:

$$\vec{v}' = R_{\vec{r}}(\theta)\,\vec{v}$$

the result of the multiplication between the rotation matrix and the vector $\vec{v}$ is nothing more than a change of coordinates that takes us from the v_x, v_y and v_z coordinates in the vector $\vec{v}$ to the v'_x, v'_y and v'_z coordinates of the resulting vector $\vec{v}'$.

1.3 Interesting property

An interesting property of elementary rotation matricies is that the inverse is equal to the transpose, in particular:

$$R_{\vec{r}}^{-1}(\theta) = R_{\vec{r}}^{\top}(\theta) \;\; \Rightarrow \;\; R_{\vec{r}}(\theta)\, R_{\vec{r}}^{\top}(\theta) = I_n$$

More generally, there are translation matrices which are "composed" of rotation matrices plus something else that we now see.

1.4 Translations

A generic rototranslation matrix is defined as follows:

$$\begin{aligned} T(\theta, \vec{p}) &= \left(\begin{array}{ccc|c} \multicolumn{3}{c|}{R_{\vec{r}}(\theta)} & \vec{p} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\ &= \left(\begin{array}{ccc|c} & & & l_x \\ & R_{\vec{r}}(\theta) & & l_y \\ & & & l_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \end{aligned}$$

where $R_{\vec{r}}(\theta) \in \mathbb{R}^{3\times 3}$ can be any elementary rotation matrix and $\vec{p} = (\, l_x \;\; l_y \;\; l_z \,)^{\top} \in \mathbb{R}^{3\times 1}$ is a vector which, once assigned, allow us to move along the Cartesian axes by carrying out real translations.

Rototranslation matrices are very useful because as can be understood if $l_x = l_y = l_z = 0$ then we obtain an elementary rotation matrix but they do not have the same properties as rotation matrices and therefore if we have to carry out inverse rototranslations we necessarily need to calculate the inverse matrices based on the axis of rotation that we decide to use, in particular:

$$\vec{v}' = T(\theta, \vec{p})\, \vec{v} \iff \vec{v} = T^{-1}(\theta, \vec{p})\, \vec{v}'$$

similary if $\theta = 0$ then the rototranslation matrix only performs one traslation using the axes, in particular:

1. If $l_x = l_y = l_z = 0$ then:

$$\begin{aligned} T(\theta, \vec{p}) &= T(\theta, 0) \\ &= \left(\begin{array}{ccc|c} & & & 0 \\ & R_{\vec{r}}(\theta) & & 0 \\ & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \in \mathbb{R}^{4\times 4} \\ &= R_{\vec{r}}(\theta) \in \mathbb{R}^{3\times 3} \end{aligned}$$

2. If $\theta = 0$ then:

$$
\begin{aligned}
T(\theta, \vec{p}) &= T(0, \vec{p}) \\
&= \left(\begin{array}{ccc|c} & & & l_x \\ & R_{\vec{r}}(0) & & l_y \\ & & & l_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\
&= \left(\begin{array}{ccc|c} & & & l_x \\ & I & & l_y \\ & & & l_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
\end{aligned}
$$

in fact any translation matrix calculated at $\alpha = \beta = \gamma = 0$ is equal to the identity matrix, as follows:

$$
\begin{aligned}
R_z(\alpha) &= \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \;\Rightarrow\; R_z(0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3 \\
R_y(\beta) &= \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \;\Rightarrow\; R_y(\beta) = I_3 \\
R_x(\gamma) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \;\Rightarrow\; R_x(\gamma) = I_3
\end{aligned}
$$

Therefore as regards the second case, to obtain the new coordinates it will be sufficient to carry out some sums between the components of the vector $\vec{v}$ and the components of the matrix $T(0, \vec{p})$ and if you do this you will notice how in reality this matrix can be rewritten more simply as follows:

$$
T(0, \vec{p}) = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \;\Rightarrow\; T\vec{v} = \begin{pmatrix} l_x + v_x \\ l_y + v_y \\ l_z + v_z \end{pmatrix} = \begin{pmatrix} \vec{v}'_x \\ \vec{v}'_y \\ \vec{v}'_z \end{pmatrix} = \vec{v}'
$$

1.5 Inverse matrices

As already mentioned, when we are dealing with a rototranslation matrix along any Cartesian axis, the property of rotation matrices whereby the inverse is equal to the transpose does not apply and therefore now let's see a simple way to obtain the matrix $T^{-1}(\theta, \vec{p})$.

Let's try to find a general formula starting from this simple matrix:

$$
T(\theta, \vec{p}) = \left(\begin{array}{ccc|c} & R_{\vec{r}}(\theta) & & \vec{p} \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

since $\vec{p} \in \mathbb{R}^{3\times 1}$ has no inverse then I consider a generic point $P(x, y)$ of the 3D space and I set $y = R_{\vec{r}}(\theta)\, x + \vec{p}$, where $R_{\vec{r}}(\theta)$ is our any elementary rotation matrix and $\vec{p}$ is any translation along any axis. The idea is therefore to invert this formula by making x explicit as a function of y as follows:

$$
y = R_{\vec{r}}(\theta)\, x + \vec{p} \;\Rightarrow\; x = R_{\vec{r}}^\top(\theta)\, (y - \vec{p}) = R_{\vec{r}}^\top(\theta)\, y - R_{\vec{r}}^\top(\theta)\, \vec{p}
$$

Writing this in homogeneous coordinates, the inverse matrix is:

$$
T^{-1}(\theta, \vec{p}) = \left(\begin{array}{ccc|c} & R_{\vec{r}}^\top(\theta) & & -R_{\vec{r}}^\top(\theta)\, \vec{p} \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

Fantastic! We have found a general formula to tell that $T(\theta,\vec{p})\,T^{-1}(\theta,\vec{p}) = I_n$. Now we calculate each individual translation along the 3D axes as follows:

- If $R_{\vec{r}}(\theta) = R_z(\theta)$ and $\vec{p} = (\begin{smallmatrix} 0 & 0 & l_z \end{smallmatrix})^\top$ then:

$$
\begin{aligned}
T_z(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & l_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\
-R_z^\top(\theta)\,\vec{p} &= -\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ l_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -l_z \end{pmatrix} \\
T_z^{-1}(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & -l_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
\end{aligned}
$$

- If $R_{\vec{r}}(\theta) = R_y(\theta)$ and $\vec{p} = (\begin{smallmatrix} 0 & l_y & 0 \end{smallmatrix})^\top$ then:

$$
\begin{aligned}
T_y(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & l_y \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\
-R_y^\top(\theta)\,\vec{p} &= -\begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} 0 \\ l_y \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -l_y \\ 0 \end{pmatrix} \\
T_y^{-1}(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & -l_y \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
\end{aligned}
$$

- If $R_{\vec{r}}(\theta) = R_x(\theta)$ and $\vec{p} = (\begin{smallmatrix} l_x & 0 & 0 \end{smallmatrix})^\top$ then:

$$
\begin{aligned}
T_x(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} 1 & 0 & 0 & l_x \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\
-R_x^\top(\theta)\,\vec{p} &= -\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} l_x \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -l_x \\ 0 \\ 0 \end{pmatrix} \\
T_x^{-1}(\theta,\vec{p}) &= \left(\begin{array}{ccc|c} 1 & 0 & 0 & -l_x \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
\end{aligned}
$$

With this explicit calculation every doubt should be clarified, in particular each part of the matrix has a very specific purpose as we have widely seen and this is truly impressive.

I conclude by saying that the principle of superposition of effects applies, so if for example I want to simultaneously translate both on x and y and rotate on z the rototranslation matrix will be the following:

$$T_z(\theta, \vec{p}) = \left(\begin{array}{ccc|c} \cos(\theta) & -\sin(\theta) & 0 & l_x \\ \sin(\theta) & \cos(\theta) & 0 & l_y \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

$$\begin{aligned} -R_z^\top(\theta)\,\vec{p} &= -\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} l_x \\ l_y \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} -l_y\sin(\theta) - l_x\cos(\theta) \\ l_x\sin(\theta) - l_y\cos(\theta) \\ 0 \end{pmatrix} \end{aligned}$$

$$T_z^{-1}(\theta, \vec{p}) = \left(\begin{array}{ccc|c} \cos(\theta) & \sin(\theta) & 0 & -l_y\sin(\theta) - l_x\cos(\theta) \\ -\sin(\theta) & \cos(\theta) & 0 & l_x\sin(\theta) - l_y\cos(\theta) \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

1.6 The code

The code I have reported below is just a draft compatible with Processing that implements the matrices I have explained in this pdf:

```
class Translation {
  float theta, l;

  Translation(float theta, float l) {
    this.theta = radians(theta);
    this.l = l;
  }

  float[][] x() {
    float[][] x = {
      {1, 0, 0, l},
      {0, cos(theta), -sin(theta), 0},
      {0, sin(theta), cos(theta), 0},
      {0, 0, 0, 1}
    };
    return x;
  }

  float[][] xInv() {
    float[][] xInv = {
      {1, 0, 0, -l},
      {0, cos(theta), sin(theta), 0},
      {0, -sin(theta), cos(theta), 0},
      {0, 0, 0, 1}
```

```
    };
    return xInv;
  }

  float[][] y() {
    float[][] y = {
      {cos(theta), 0, sin(theta), 0},
      {0, 1, 0, l},
      {-sin(theta), 0, cos(theta), 0},
      {0, 0, 0, 1}
    };
    return y;
  }

  float[][] yInv() {
    float[][] yInv = {
      {cos(theta), 0, -sin(theta), 0},
      {0, 1, 0, -l},
      {sin(theta), 0, cos(theta), 0},
      {0, 0, 0, 1}
    };
    return yInv;
  }

  float[][] z() {
    float[][] z = {
      {cos(theta), -sin(theta), 0, 0},
      {sin(theta), cos(theta), 0, 0},
      {0, 0, 1, l},
      {0, 0, 0, 1}
    };
    return z;
  }

  float[][] zInv() {
    float[][] zInv = {
      {cos(theta), sin(theta), 0, 0},
      {-sin(theta), cos(theta), 0, 0},
      {0, 0, 1, -l},
      {0, 0, 0, 1}
    };
    return zInv;
  }
}
```

If you have any doubts, don't hesitate to contact me.