

Homogeneous coordinates

Overview

Geometric transformations play a key role in computer graphics and related fields. Transformations that are linear can be represented by a single matrix. On the hand, non-linear transformations such as the affine transformation, can still be partially represented by matrices. Their mathematical notation, however, can become cumbersome when expressing inverses and combinations of multiple transformations. Luckily, there is a geometric space where transformations are always linear, which means that we can always represent them as a (single) matrix. This geometry is called projective geometry and we can convert vectors and transformations from Cartesian coordinates to Projective coordinates (i.e., homogeneous coordinates) and back.

Two-dimensional transformations in homogeneous coordinates

The vector $\mathbf{x} = (x, y)^T$ can be represented in homogeneous coordinates by the vector $\tilde{\mathbf{x}} = (x, y, \lambda)^T$, with $\lambda \neq 0$, i.e.:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ \lambda \end{bmatrix}. \quad (15)$$

This is still a 2-D point but it is now represented by 3 instead of 2 coordinates. Here, all we are doing to convert from Cartesian coordinates to homogeneous coordinates is to augment the original vector by adding $\lambda \neq 0$ as an extra coordinate.

To convert homogenous coordinates back to Cartesian, we need only to divide the homogeneous coordinates as follows:

$$\begin{bmatrix} x \\ y \\ \lambda \end{bmatrix} \rightarrow \begin{bmatrix} x/\lambda \\ y/\lambda \end{bmatrix}. \quad (16)$$

The simplest and most common choice is to set $\lambda = 1$, i.e.:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (17)$$

If $\lambda = 1$, then most conversions are straightforward,

Because, we augmented the vectors, we must also augment the transformations. The following are 2-D transformations in homogeneous coordinates:

Scaling:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

Rotation:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (19)$$

And most importantly, we can now also represent translation as a matrix, which is something that we could not do in Cartesian coordinates. The 2-D translation matrix is given by:

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

The affine transformation also has its equivalent in homogeneous coordinates, i.e.:

$$A = \begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (21)$$

Because all transformations in projective geometry are linear, we can represent compositions and inverses with a concise notation. For example, the transformation:

$$\mathbf{x}' = SR\mathbf{x} + \mathbf{t}. \quad (22)$$

can be written in homogeneous coordinates as:

$$\tilde{\mathbf{x}}' = T_h S_h R_h \tilde{\mathbf{x}}, \quad (23)$$

and its inverse would be simply:

$$\begin{aligned} \tilde{\mathbf{x}} &= (T_h S_h R_h)^{-1} \tilde{\mathbf{x}}' \\ &= R_h^{-1} S_h^{-1} T_h^{-1} \tilde{\mathbf{x}}'. \end{aligned} \quad (24)$$

In Equation 23, the subscript h is being used to indicate the homogeneous version of the transformation. However, we do not usually explicitly differentiate the transformations.

In higher dimensions, homogeneous coordinates are analogous to the 2-D case, i.e., just add an extra coordinate to the original vector. In the 3-D case, the homogeneous representation of a vector is:

$$\tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (25)$$

The translation matrix is:

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (26)$$

The scaling matrix is:

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (27)$$

And a transformation $\tilde{\mathbf{x}}' = S R \tilde{\mathbf{x}}$ is:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (28)$$