

## Matrix form of KernelCAD Frames

Matrices can be considered as arrays of vectors

Suppose point

$$p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix}$$

Is the position of a frame and

$$v^j = \begin{bmatrix} v_0^j \\ v_1^j \\ v_2^j \end{bmatrix}$$

Is the vector of its j-th axis. The array of axes  $(v^0, v^1, v^2)$  as a matrix is:

$$A = \begin{bmatrix} v_0^0 & v_0^1 & v_0^2 \\ v_1^0 & v_1^1 & v_1^2 \\ v_2^0 & v_2^1 & v_2^2 \end{bmatrix}$$

The transform of a point  $x$  (The result of  $f.ToGlobal(x)$ ), we discussed previously, is

$$y = A * x + p$$

Where  $*$  is the multiplication of column  $x$  by the matrix  $A$ . In coordinates it is:

$$y_i = \sum_{j=0}^2 v_i^j x_j + p_i$$

In terms of vectors  $y$  is a linear combination of axes with coefficients  $x_j$  and offset  $p$ :

$$y = \sum_{j=0}^2 v^j x_j + p$$

There is a neat way to represent this as a single linear operation. It involves embedding 3D into projective space ([https://en.wikipedia.org/wiki/Projective\\_space](https://en.wikipedia.org/wiki/Projective_space)). Very sketchy, projective space consists of 4D directions, or non-zero 4D vectors where vectors different by length only are considered the same. In coordinates elements of projective space are written down as vectors with keeping in mind that only its direction matters

The location  $p$  above and generally any point as a projective direction is mapped to:

$$\hat{p} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ 1 \end{bmatrix}$$

The axis  $v^j$  (or any vector), is mapped to:

$$v^j = \begin{bmatrix} v_1^j \\ v_1^j \\ v_1^j \\ 0 \end{bmatrix}$$

If we construct a matrix  $(v^0, v^1, v^2, p)$  out of axes and the position we will get

$$\hat{A} = \begin{bmatrix} v_0^0 & v_0^1 & v_0^2 & p_0 \\ v_1^0 & v_1^1 & v_1^2 & p_1 \\ v_2^0 & v_2^1 & v_2^2 & p_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And for a point

$$\hat{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Its transform  $\hat{y}$  is simply:

$$\hat{y} = \hat{A} * \hat{x}$$

This makes everything uniform: Vectors and points become the same thing and translation can be written down as a linear operation