

Graphics Transformations

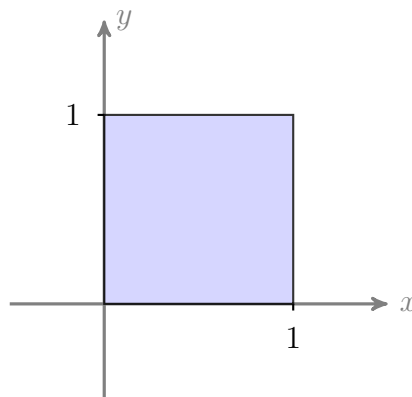
Computer graphics are images displayed or animated on a computer screen. Computer graphics are used all over the web, in video games, and computer aided design (CAD) systems. At its core, computer graphics are the manipulation of points and shapes. Most of the shapes are approximated by straight lines or polygons. In this lecture and the next we'll look at how you can formulate the manipulation of basic shapes as a set of linear transformations. We'll start with graphics in 2D and then extend the tools we develop to 3D.

Drawing a two-dimensional polygon on a screen requires knowledge about the location of the vertices of the polygon, as well as a convention to decide which points to connect with lines. We'll adopt the convention that points are given as column vectors, points in the same polygon are stored as the columns of a matrix, and lines are connected in the order that the points appear in the matrix.

Example 1: The unit square in the first quadrant has vertices $(0, 0)$, $(1, 0)$, $(1, 1)$, and $(0, 1)$. To connect the points in the given order we store them as a matrix

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Connecting the vertices in the order they appear results in the shape



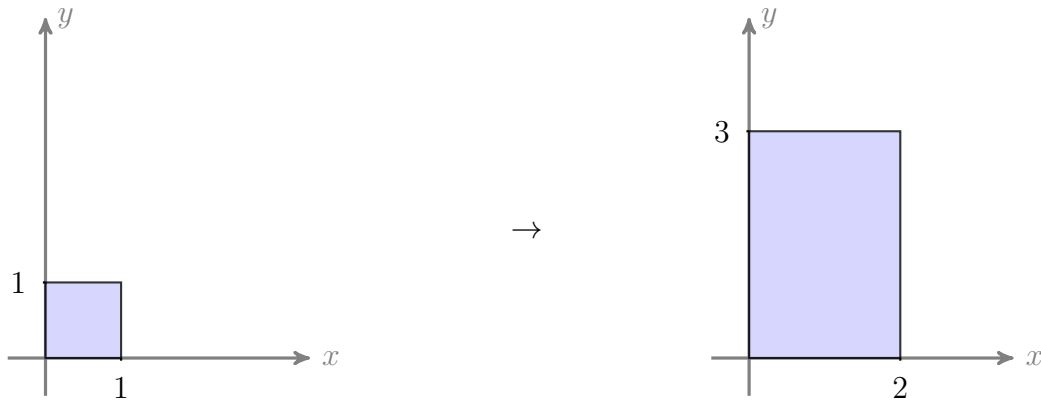
Our goal now is to manipulate this shape through linear transformations to make it look like something else. The main geometric manipulations that we will be concerned with are

- translation
- scaling
- rotation
- shearing

For now we will table the discussion of translation because it will throw a wrench into our whole operation. The rest of the transformations can be done simply using 2×2 matrices operating on length-2 vectors that represent points.

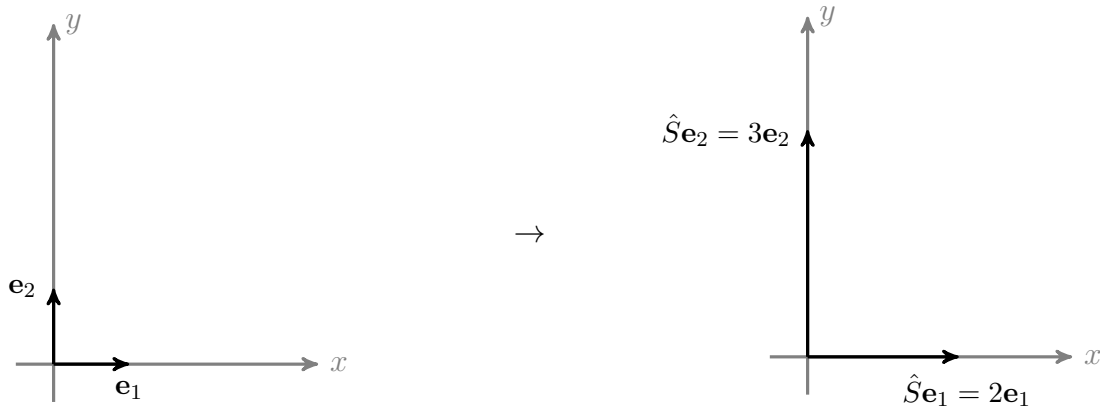
Scaling

Consider the case when we want to scale the length of each of the side of the unit square until it looks like the following rectangle.



Our goal is to find a 2×2 matrix \hat{S} that transforms each of the vectors representing the vertices in the square into the corresponding vectors representing the vertices of the rectangle.

Note that one way to accomplish this is to choose \hat{S} such that it scales a vector pointing in the x -direction to be twice its original length, and at the same time scales a vector pointing in the y -direction to be three times its original length. To do this we can choose \hat{S} so that it has the desired effect on the canonical basis vectors pointing in the x - and y -directions.



The nice thing about operating on the the canonical basis vectors is that multiplication by \hat{S} reveals the columns of \hat{S} . Mathematically, we have

$$\hat{S} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \text{and} \quad \hat{S} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Since $\hat{S}\mathbf{e}_1$ should give us the first column of \hat{S} and $\hat{S}\mathbf{e}_2$ should give us the second column of \hat{S} , we have

$$\hat{S} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

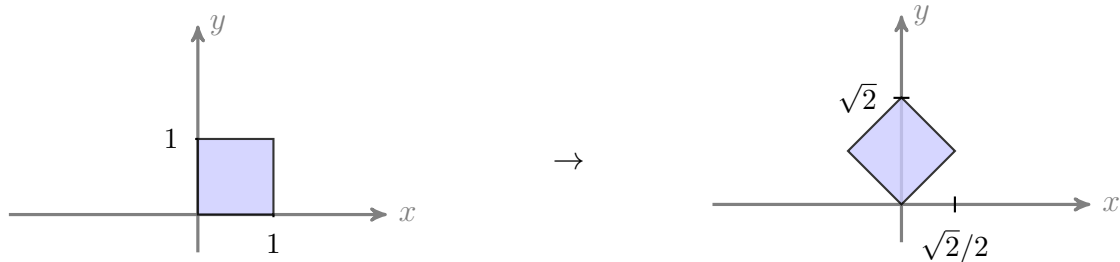
To scale all of the vertices of the unit square at the same time, we can simply multiply the vertex matrix \mathbf{X} by \hat{S} . We have

$$\hat{S}\mathbf{X} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 3 \end{bmatrix}$$

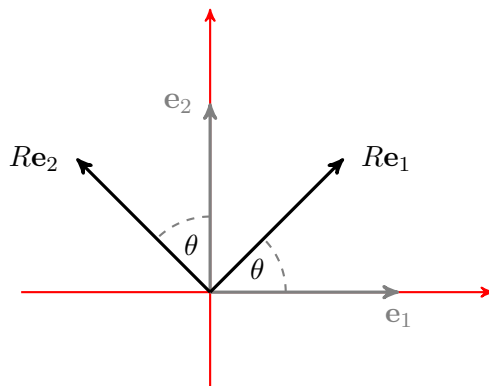
which are clearly the vertices of the desired rectangle.

Rotations

Consider now the case when we want to rotate the unit square some angle θ counterclockwise about the origin. If we choose $\theta = \frac{\pi}{4}$ then the result looks as follows:



We can again accomplish this by choosing a matrix \hat{R} that accomplishes the desired effect on the canonical basis vectors. We'll figure out the rotation matrix for a general θ . The desired effect on the basis vectors looks as follows



A little trig convince us that we want \hat{R} to have the following desired specific effect on \mathbf{e}_1 and \mathbf{e}_2 :

$$\hat{R} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad \text{and} \quad \hat{R} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}$$

Thus the rotation matrix that rotates things θ radians about the origin is

$$\hat{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

For the specific example with $\theta = \pi/4$ we have

$$\hat{R} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

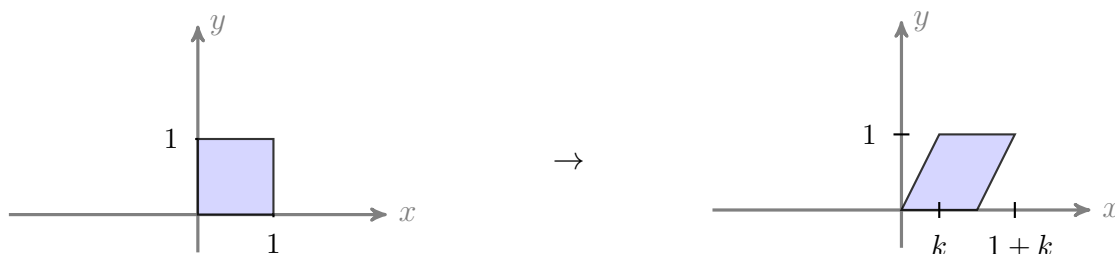
Multiplying the coordinate matrix of the unit square by the rotation matrix gives the vertices of the rotated square:

$$\hat{R}\mathbf{X} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{2}/2 & 0 & -\sqrt{2} \\ 0 & \sqrt{2}/2 & \sqrt{2} & \sqrt{2} \end{bmatrix}$$

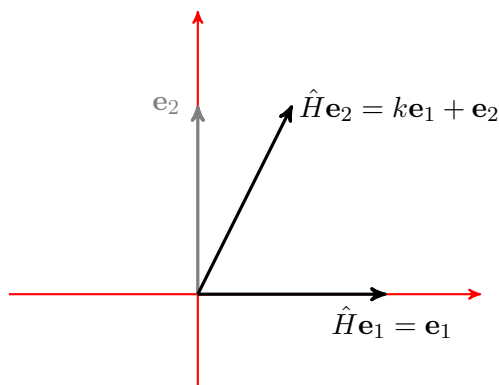
which are of course the vertices that we expect from the picture.

Shears

Shear transformations allow us to turn squares into things like parallelogram. There are shears in the horizontal direction and the vertical direction. Application of a horizontal shear moves points in a horizontal direction proportional to their distance from the x -axis. Consider the following shear transformation applied to the unit square in the first quadrant:



The shear transformation pictured here moved the vertices lying on the line $y = 1$ to the right k units. To determine the matrix \hat{H} that applies this transformation we again consider how the transformation moves the first two canonical basis vectors.



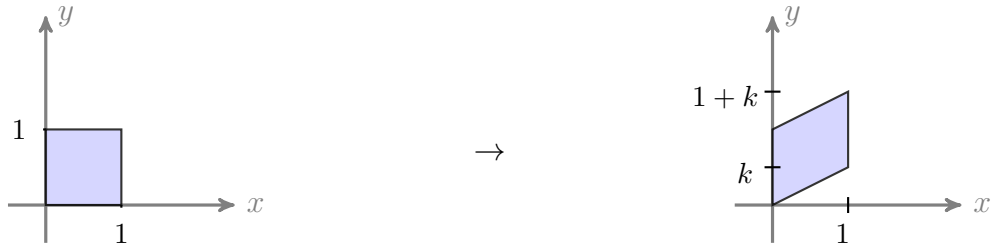
From this we see that

$$\hat{H} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \hat{H} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} k \\ 1 \end{bmatrix}$$

Thus the horizontal shear matrix is given by

$$\hat{H} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

A similar analysis shows that the vertical shearing matrix $\hat{V} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$ performs the following transformation



Translation

Translation is the transformation that moves an object a fixed distance in the x and y -directions. In the image below the unit square is translated x_0 in the x -direction and y_0 in the y -direction.



Unfortunately, we cannot find a matrix T that performs the translation operation, because translation is not a linear operation. To see this, recognize that there is no matrix that can transform the origin (i.e. the zero vector) into a nonzero vector. To overcome this we have to increase the dimension of the space that we're working in.

Instead of representing the point (x, y) by the vector $\begin{bmatrix} x \\ y \end{bmatrix}$ we represent it by $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$.

The extra 1 in the third entry will allow us to perform translations by multiplication by a 3×3 matrix. The representation of (x, y) by the length-3 vectors is called the *homogeneous coordinate system*. For our purposes, the third coordinate will always be a 1, but there are interpretations of the coordinate system where it can take on different values.

OK, so how do we represent the above transformation by a 3×3 matrix? Our goal is to find a matrix T such that

$$T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ 1 \end{bmatrix}$$

Interpreting multiplication by T as a linear combination of its columns, we have

$$T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x \begin{bmatrix} \\ \\ \end{bmatrix} + y \begin{bmatrix} \\ \\ \end{bmatrix} + 1 \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ 1 \end{bmatrix}$$

Filling in the entries of the column vectors to put the appropriate values of x , y , x_0 , and y_0 in the correct positions, we have

$$T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \\ 1 \end{bmatrix}$$

which gives us the translation matrix $T = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$

Now, we want to use a consistent coordinate system that allows us to perform scalings, rotations, shears, and translations. Because of the translations, we'll use the homogeneous coordinate systems for them all. This means that we need to come up with 3×3 representations of the non-translation transformations. This turns out to be fairly simple. We simply replace each transformation by a 3×3 identity and replace its upper 2×2 block with the desired 2×2 transformation matrix.

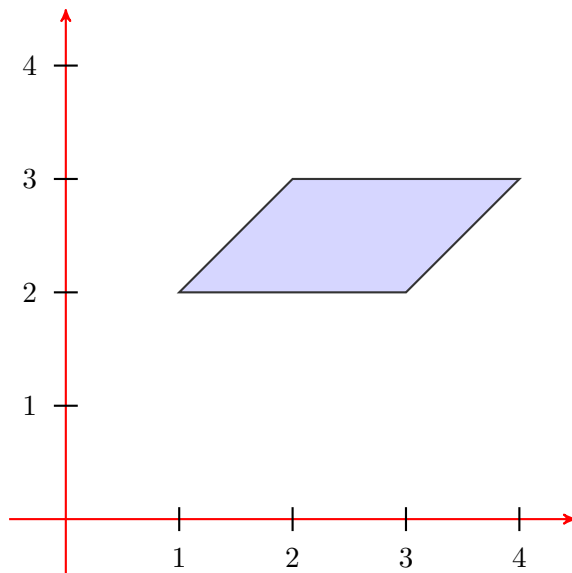
$$S = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that since the last row and column of each matrix have only a 1 in the $(3,3)$ -position, the third entry of the homogeneous coordinate vector does not affect the action of the original transformation.

Composite Transformations

Most interesting shapes are created by applying multiply simple transformations to a representative shape, like the unit square.

Example 1: Suppose that we want to transform the unit square into the following shape:



One way to accomplish this is

1. Scale the square to a rectangle two units wide and one unit tall.
2. Apply a horizontal shear with $k = 1$
3. Translate the result two units in the y -direction and one unit in the x -direction.

If we let \mathbf{X} be the coordinate matrix describing the unit square, the new shape will be given by

$$\mathbf{X}_{\text{new}} = T H S \mathbf{X}$$

where

$$S = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

We have

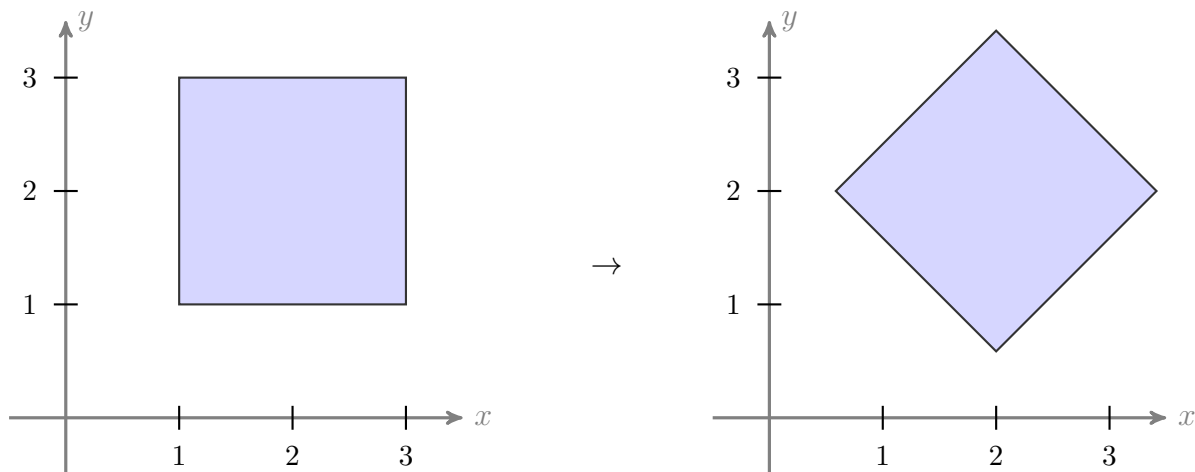
$$S\mathbf{X} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H(S\mathbf{X}) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 3 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$T(HS\mathbf{X}) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 4 & 2 \\ 2 & 2 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Notice that the first two entries of the four column-vectors of \mathbf{X} are the precise coordinates of the parallelogram in the example.

Example 2: Suppose that we want to take a cube of side length 2 centered at the point $(2, 2)$ and rotate it $\pi/4$ radians counterclockwise about it's center.



Recall that the simple rotation transformation rotates points about the origin. This means that we cannot accomplish this transformation with a simple rotation. Since we want to rotate the square about the point $(2, 2)$ our strategy will be to translate the square so that the point $(2, 2)$ moves to the origin, rotate the square about the origin, and translate it back to it's original position.

This can be accomplished by the following composition of simple transformations

$$\mathbf{X}_{\text{new}} = T_2 R T_1 \mathbf{X}$$

where

$$T_1 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

The homogeneous coordinates of the original square are given by

$$\mathbf{X} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We then have

$$T_1 \mathbf{X} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$R(T_1 \mathbf{X}) = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{2} & 0 & -\sqrt{2} \\ -\sqrt{2} & 0 & \sqrt{2} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$T_2(RT_1 \mathbf{X}) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \sqrt{2} & 0 & -\sqrt{2} \\ -\sqrt{2} & 0 & \sqrt{2} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 + \sqrt{2} & 2 & 2 - \sqrt{2} \\ 2 - \sqrt{2} & 2 & 2 + \sqrt{2} & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The extension of these transformations to three dimensions is relatively straight forward. We will only consider scalings, rotations, and translations. Of these the only one that is not identical to its 2D transformation is rotation.

A point in three dimensions (x, y, z) is represented in homogeneous coordinates by a four dimensional vector with the fourth entry equal to 1:

$$(x, y, z) \mapsto \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling an object in three dimensions by c_1 in the x -direction, c_2 in the y -direction, and c_3 in the z -direction is given by the diagonal matrix

$$S = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

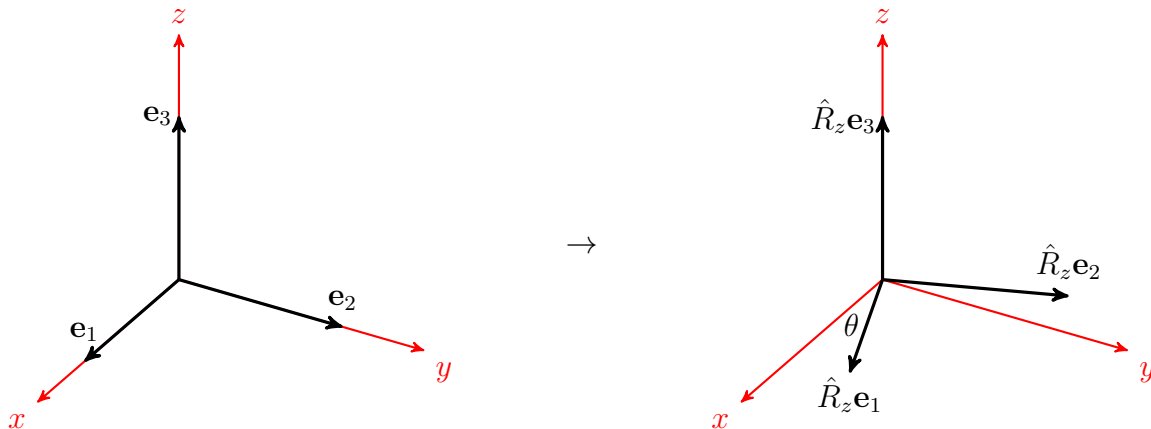
Translation of a point a fixed amount in each coordinate direction is analogous to it's 2D variant. The transformation

$$(x, y, z) \mapsto (x + x_0, y + y_0, z + z_0)$$

is given by the matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotations pose a slight complication. It no longer makes sense to talk about a rotation about the origin since this is inherently ambiguous. Instead we can build rotations based on simple rotations about each of the coordinate axes. For example, the 2D rotation of vectors in the xy -plane considered before can be interpreted as rotation about the z -axis. Rotation an angle θ in the counterclockwise direction is depicted below.



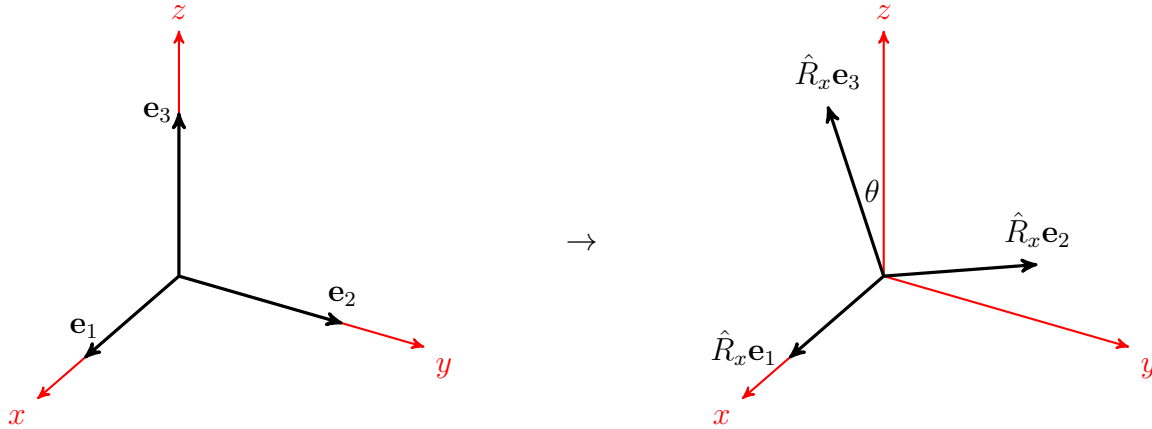
From the picture and a little trig we see that the canonical basis vectors transform as follows

$$\hat{R}_z \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} \quad \hat{R}_z \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \quad \hat{R}_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Adding the identity row and column corresponding to the homogeneous coordinate we have

$$R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about the x -axis is similar. The picture looks like



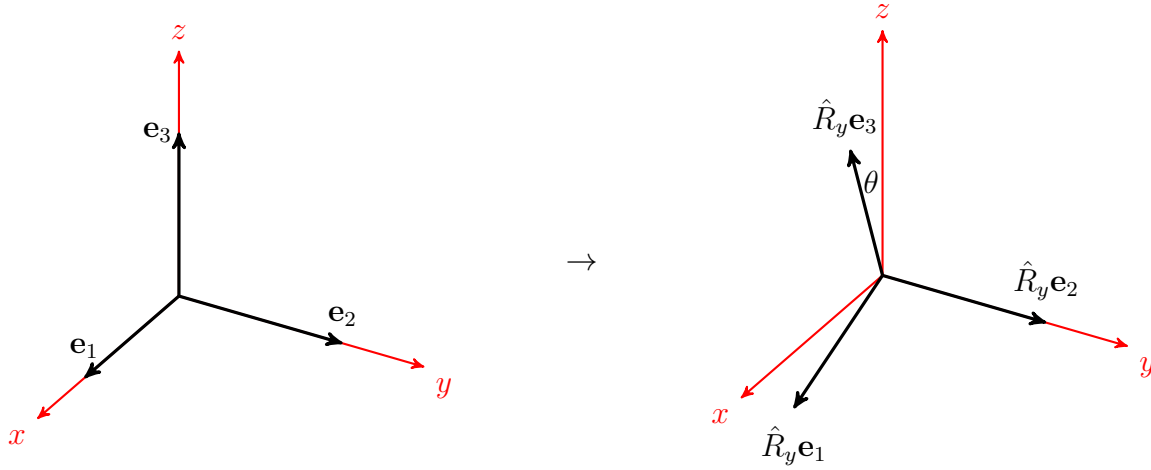
We have

$$\hat{R}_x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{R}_x \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} \quad \hat{R}_x \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin \theta \\ \cos \theta \end{bmatrix}$$

The transformation matrix in the homogeneous coordinate system is given by

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the x -axis rotation matrix looks very similar to the z -axis rotation matrix but with the roles of x and y replaced by y and z . Rotation about the y -axis is a slightly different animal. The picture looks as follows



We have

$$\hat{R}_y \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta \\ 0 \\ -\sin \theta \end{bmatrix} \quad \hat{R}_y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \hat{R}_y \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin \theta \\ 1 \\ \cos \theta \end{bmatrix}$$

The transformation matrix in homogeneous coordinates is given by

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the matrix for rotation about the y -axis is similar to those for rotation about the x - and y -axes, but the signs on the sin terms are in the reverse positions. This is due to the convention to define rotations that are counterclockwise and the relation of the pairs of axes to each other. Rotation about the z -axis moves a vector that lies on the x -axis towards the y -axis. Rotation about the x -axis moves a vector that lies on the y -axis towards the z -axis. But rotation about the y -axis moves a vector along the z -axis towards the x -axis.