Linear Transformations on 3D Objects as Matrix Multiplication

Devin Peevy, Chase Dolan, Trae Claar

March 2023

1 Abstract

This paper discusses how linear transformations are used to scale, rotate, and shear 3D objects. The authors have written some short example code in python to accompany the explorations, so that the reader may more easily visualize the transformations occurring in a three dimensional space. The code is available here at github.com/tclaar/CubeTransformations. For each transformation, there will be a brief explanation of what the transformation does, how it is implemented, and an example to go alongside it, followed by some possible applications of that specific transformation.

2 Introduction

While linear algebra is a widely useful topic, it can be hard to find simple examples of practical applications, or to explain the applications of matrix algebra in a straight-forward fashion. One fairly obvious example is that of three dimensional computer graphics. A matrix can represent a 3D object by storing each of its vertices as a single column vector. For this reason, the three simple transformations discussed in this paper can be done with linear operators on A_{3xn} matrices.

Knowing a linear operator on an A_{3xn} matrix would need to produce another matrix B_{3xn} (the shape should not lose any vertices!), it is clear for these three simple transformations that we need to find a transformation matrix T_{3x3} such that $B = T \cdot A$. The process for finding this matrix T will be explained for each simple transformation.

For the examples of this paper, the transformations will be modeled on a standard `cube` (or variations of this standard `cube`) defined by the matrix:

$$\texttt{cube} = \begin{bmatrix} -5 & -5 & -5 & -5 & 5 & 5 & 5 & 5 \\ -5 & -5 & 5 & 5 & -5 & -5 & 5 & 5 \\ -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 \end{bmatrix}$$

That is, a 10 x 10 cube centered at the origin with all faces orthogonal to the x, y, and z axes, as shown in Figure 2a.

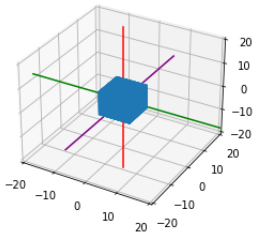

Figure 2a

3 Scaling

One basic transformation operation for 3D objects is scale, which changes the size of an object. Scale operations can involve both increasing and decreasing the size of an object. These two types of scaling are sometimes differentiated by calling them dilations and contractions, respectively. Scaling is often uniform; that is, it maintains the proportions of the object being operated on. However, it is also possible to scale an object by different factors on each axis, allowing it to be stretched, compressed, or both in the same operation.

If an object is represented in 3D space as a matrix A, then scaling can intuitively be thought of as scalar multiplication on A. However, such a transformation is necessarily uniform, since each element of A is scaled by a single factor. Additionally, other 3D transformations cannot be represented in this way, and it would be better to generalize all transformations under a single operation. To accomplish this, one can recognize that scaling is a linear transformation and can therefore be represented by a matrix. To scale objects, this matrix can be left-multiplied with the object's matrix representation, yielding a scaled product.

In 3D space, a scale matrix takes the form of a 3 by 3 diagonal matrix:

$$\mathrm{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

Elements s_x, s_y and s_z on the diagonal are the scale factors for their respective axes. For example, the scale matrix:

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

is a uniform dilation by a factor of 2. If A is the matrix representation of a 3D object, then the operation $S \cdot A$ doubles the size of the object on all axes, as shown in Figure 3a.

Now, if we have:

$$\mathrm{S} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

then SA results in an object which is shrunk in half on the X-axis, stretched

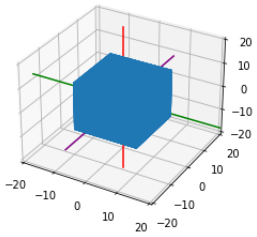

Figure 3a

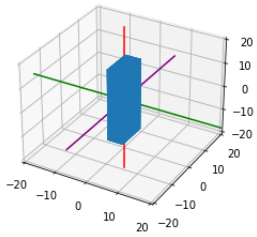

Figure 3b

by a factor of 3 on the Z-axis, and which remains the same on the Y-axis. Figure 3b visualizes this transformation.

From this example, it can be seen that a scale factor less than 1 results in a size decrease, greater than 1 an increase, and also that the first, second, and third rows correspond to the X, Y, and Z axes respectively in 3D space. Negative scale factors, in addition to scaling an object, also perform a reflection across an axis.

To demonstrate this last remark, let A be the standard `cube` translated up the x axis 15 units units and let S be the below matrix:

$$S = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}$$

The result of this transformation is shown in Figure 3c.

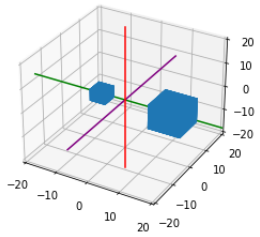

Figure 3c

There are several things worth mentioning about this result. First, reflections across the z and x-axes have occurred, but are not visible because the cube is centered on these axes. Second, note that the transformed cube is closer to the origin than the original. This is due to the properties of a scale transformation. Recall that a scale linear transformation is analogous to scalar multiplication, in which each element of a matrix is multiplied (that is, scaled). So too here: distance from the origin is realized in the matrix as a positional offset in each coordinate, or element. When the matrix is multiplied, this offset is modified - in this case, reduced.

Scale is an exceedingly useful transformation in 3D graphics as there are a large number of reasons one may want to change the size of an object. Some of

these include changing the size of accessories to fit a character or randomizing the proportions of props to increase variety in a scene. In live time, objects at a distance can be scaled to create a sense that they are growing closer or further away. Clearly, these examples are only a small subset of the applications of scale. It can be easily seen that this transformation is essential in 3D graphics and modeling.

4 Rotation

Rotation along any axis centered at the origin is possible by doing a linear operator on the matrix. The formula for calculating the transformation matrix R around a vector $\underline{v} = [\alpha, \beta, \gamma]^T$ is:

$$R = \begin{bmatrix} \cos\theta + \alpha^2(1-\cos\theta) & \alpha\beta(1-\cos\theta) - \gamma\sin\theta & \alpha\gamma(1-\cos\theta) + \beta\sin\theta \\ \beta\alpha(1-\cos\theta) + \gamma\sin\theta & \cos\theta + \beta^2(1-\cos\theta) & \beta\gamma(1-\cos\theta) - \alpha\sin\theta \\ \gamma\alpha(1-\cos\theta) - \beta\sin\theta & \gamma\beta(1-\cos\theta) + \alpha\sin\theta & \cos\theta + \gamma^2(1-\cos\theta) \end{bmatrix} \text{[1]}$$

It is important to note that the rotation vector needs a magnitude of one in order to simply rotate the object. If the magnitude is greater or smaller than one, then it will stretch or shrink (respectively) the shape (as it also rotates it along the axis). That means if you want to rotate an object around a vector without scaling it, you'll want to plug a vector that looks like $||\underline{v}||^{-1} \cdot \underline{v}$ into the above formula in order to get the transformation matrix.

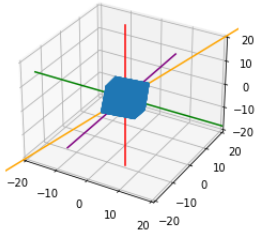

Figure 4a

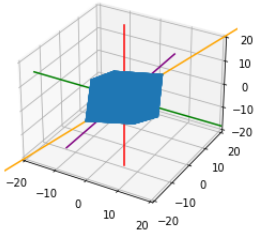

Figure 4b

Both Figure 4a and 4b rotate about the axis shared by the vector $\underline{v} = [1, 1, 1]^T$. The difference is that 4a correctly plugs in the unit vector formed by $\frac{\underline{v}}{\sqrt{3}}$. 4b just plugs in $\underline{v}$, and since $||\underline{v}|| = \sqrt{3} > 1$, the shape grows.

The amount that the object is scaled depends on θ, as well as the magnitude of the rotation vector $||\underline{v}||$. If $\theta = 0$, nothing will happen to the cube at all, even if $||\underline{v}|| \neq 1$. The greater $|\theta|$ and $||\underline{v}||$ are, the more severe the scale will be. If the object is offset from the center, the distance from the origin will change as well, similar to the reflection done when scaling times a negative (as mentioned in section 3).

Since `cube` is centered at the origin, the rotations so far have appeared as spinning in place. But moving the object away from the origin and then rotating it will cause it to move in an orbit around the axis. Let us shift `cube` up the x axis 15 units and call it `orbit_cube`.

$$\texttt{orbit\_cube} = \begin{bmatrix} 10 & 10 & 10 & 10 & 20 & 20 & 20 & 20 \\ -5 & -5 & 5 & 5 & -5 & -5 & 5 & 5 \\ -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 \end{bmatrix}$$

Now, let us show `orbit_cube` rotated around the z axis 5 times at $\theta = \frac{\pi}{3}$ radians each time. The cube orbits around $\underline{e_3}$ and maintains constant distance and scale, as in Figure 4c. But if instead of $\underline{e_3}$ we plug in $[0, 0, 0.8]^T$, the cubes are pulled closer to the origin and shrunk, as in Figure 4d.

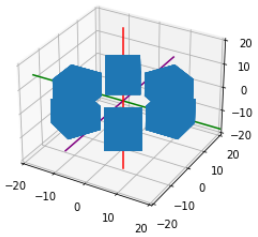

Figure 4c

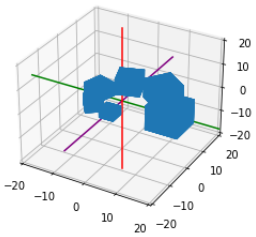

Figure 4d

The applications of rotation are plentiful in three dimensional graphics. It can be used to adjust every single object on the screen when the player changes direction. It could be used to simulate clock hands, or hypnotic patterns, or an ax being thrown through the air - the possibilities are endless. It would be critical for anybody working with 3D graphics to understand the linear algebra behind rotation.

5 Shear

In simple terms, shear mapping modifies a shape by a fixed value known as a shear factor. For a two dimensional shape, a shear in the direction of the X-axis means adding to a coordinate's x-value the product of the shear factor and the respective y-value of the coordinate pair. This is known as a horizontal shear.

For a two dimensional shape, this type of transformation can be represented by simple matrix multiplication in which the coordinate of a point on a shape is left-hand multiplied by the Identity matrix augmented by a non-zero shear factor in the i_{12} or i_{21} positions leading to the product $(x + my, y)^T$ or $(x, y + mx)^T$.

For 3D shapes, on the other hand, the shear transformation tilts or slants an object in the direction of the X,Y or Z axes (or multiple axes). The vertices of said shape are modified in the direction of the transformation's respective axis thus visually elongating the vertical edges of the cube.

Thus, the shear transformation can simulate the lateral stretching of objects relative to a basis of choice, which can be very useful in 2D and 3D modeling of various types.

For a three dimensional shape, which column the shear factor is in determines the dimension which the shear will be in. Which row the shear factor is in will

determine with respect to which variable the shape will be sheared.

The transformation matrix for a Z-axis shear with respect to Y takes a form like the following:

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

Figure 5a is the result of the shear transformation in which the original cube's matrix has been multiplied by the transformation matrix above.

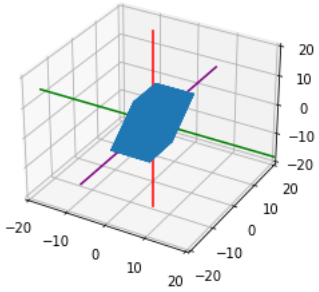

Figure 5a

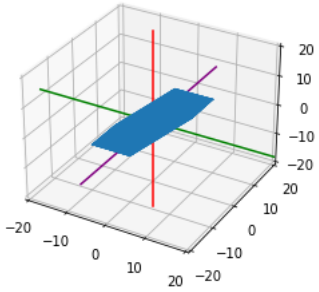

Figure 5b

While adding a non-zero value to the identity matrix at row 2, column 3 generates a Z-axis transformation with respect to Y, modifying the value at row 1, column 3 would cause a similar Z-axis shear with respect to X. The result of a Z-axis shear with respect to both X and Y can be seen in figure 5b.

Note that in both of these examples, since the transformations are exclusively in z, that the faces parallel to $z = 0$ are translated but otherwise undistorted.

For a final example, shearing in the Y-axis with respect to Z will require the following matrix:

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

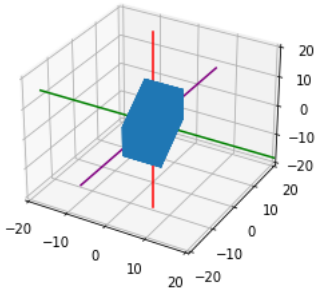

Figure 5c

The result of a shear in the Y-axis with respect to Z can be seen in Figure 5c. It can be seen that the cube's coordinate values are sheared in the direction of the positive Z-axis as the positive Y value increases. Note that the vertices of the left and right-most faces on Figure 5c are sheared in the opposite direction of one another to the same degree. This is because the shear is proportional to the magnitude of the original coordinates on either side of the respective axis of the transformation.

Some examples that come to mind for visualization purposes are simulations of the oscillations of a skyscraper in an earthquake, or rendering the stretching effect the shadow of a figure has based on the position of the light source. In the former, a building's roof-line moves laterally and further at the top than at the base because the movement is proportional to the height of the building. The shear effect is similar to simulating the shadow of a figure which can be a practical application of the shear transformation in video game design in which the shape of an object must be sheared under an altered basis. (Think of a human standing parallel to the Z-axis, while her shadow would be closer to paralleling the X or Y- axis and would be stretched considerably). The further away the projection of the shadow is from the figure, the more pronounced its shear effect becomes.

6 Conclusion

The ability to modify the characteristics of an object is very important in 3D Graphics applications. Scaling an object up or down and rotating an object are basic functionalities of many 3D modeling programs. Shear on the other hand is a bit more complex in its application, but is used frequently in civil and industrial engineering to study the effect of deformations on a structure. These tools help provide a visual representation of what matrix multiplication can accomplish, and therefore are useful explorations for students of varying focuses. The implementation of these transformations can provide a tangible introduction to graphics in Python as well, making it a good building block for engineering and technology fields.

7 Bibliography

[1] W. Moore. (2012). *Linear algebra for graphics programming* [Online]. Available: https://metalbyexample.com/linear-algebra/

[2] S. Khan. (N.D.). *3D plotting in Python using matplotlib* [Online]. Available: https://likegeeks.com/3d-plotting-in-python/amp/

[3] madhav_mohan. (2021). *Computer Graphics - 3D Shearing Transformation* [Online]. Available: https://www.geeksforgeeks.org/computer-graphics-3d-shearing-transformation/

[4] GateVidyalay.com. (2020). *3D Shearing in Computer Graphics.* [Online]. Available: https://www.gatevidyalay.com/3d-shearing-in-computer-graphics-definition-examples/