Geometric transformations in 2-D

1 Overview

In these notes, we introduce the topic of two-dimensional geometric transformations. We begin with the definition of the basic single-valued linear function, which we then upgrade to its two-dimensional counterpart. The 2-D transformation usually consists of two separate equations, each generating a component of the transformed vector by combining (i.e., mixing) the components of the original input vector with the parameters of the transformation. When these transformations are linear, they can be written in matrix form which allows us to take advantage of the tools of matrix algebra and linear algebra. By using matrices to represent transformations, we can perform efficient computations on large quantities of data as well as write formulations that are both elegant and concise.

2 One-dimensional linear transformations

The function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that:

$$f(x) = ax, \tag{1}$$

with $a \in \mathbb{R}$ is a simple case of a 1-D linear transformation. It maps (i.e., transforms) the value x onto a new value ax. For example, $f(x) = 3x$ maps an input value x into its triple.

3 2-D (Linear) Geometric Transformations

The 2-D extension of the function in Equation 1 transforms two-dimensional vectors $\mathbf{x} = (x, y)^\mathsf{T}$ onto two-dimensional vectors. In this case, we write $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that:

$$\begin{aligned} x' &= f_1(x, y) = a_1 x + a_2 y, \\ y' &= f_2(x, y) = a_3 x + a_4 y. \end{aligned} \tag{2}$$

In the 2-D case, the linear transformation consists of two separate equations, each resulting in one component (or coordinate) of the transformed vector. Equation 2 can be written in matrix form as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{3}$$

or, in short:

$$\mathbf{x}' = A\mathbf{x}. \tag{4}$$

Note that Equations 1 and 4 have the same overall form but the 2-D version maps vectors instead of numbers, and the transformation is represented by a 2×2 matrix:

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}. \tag{5}$$

This matrix is called the *matrix of the linear transformation*.

4 Some important 2-D geometric transformations

The matrix of some important linear geometric 2-D transformations are:

Scaling:

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}. \tag{6}$$

Rotation:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{7}$$

Shearing:

$$S_x = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad S_y = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}, \tag{8}$$

where S_x is a horizontal shear (i.e., parallel to the x-axis), and S_y is a vertical shear (i.e., parallel to the y-axis).

5 Examples

5.1 Transforming a single point

Consider the following scaling transformation matrix:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \tag{9}$$

which we can use the matrix to transform the point $\mathbf{x} = (9, 6)^\mathsf{T}$ as follows:

$$\begin{aligned} \mathbf{x}' &= A\mathbf{x} \\ \begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 9 \\ 6 \end{bmatrix} \\ &= \begin{bmatrix} (2 \times 9) + (0 \times 6) \\ (0 \times 9) + (3 \times 6) \end{bmatrix} \\ &= \begin{bmatrix} 18 \\ 18 \end{bmatrix}. \end{aligned} \tag{10}$$

5.2 Transforming a shape (or polygon) given by a set of vertices

Equation 10 transformed a single 2-D point. To transform a whole shape represented by a set of n 2-D vertices, we can transform each individual point of the shape (i.e., vertices) by repeatedly applying Equation 10. Alternatively, we can collect all vertices of the shape as columns of a matrix, and then apply the transformation just as we did before for the single-point case, i.e.:

$$X' = A\,X \tag{11}$$

$$\underbrace{\begin{bmatrix} x'_1 & x'_2 & \dots & x'_n \\ y'_1 & y'_2 & \dots & y'_n \end{bmatrix}}_{X'} = \underbrace{\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix}}_{X}. \tag{12}$$

Here, matrix X' is the transformed shape. Computationally, this approach is usually much faster than looping through all points to transform them individually. Also, using a matrix to represent the entire shape shortens the mathematical notation.

For example, given a rectangle described by a set of four 2-D points, $\mathbf{p}_i = (x_i, y_i)^\mathsf{T}$, for $i = 1, \ldots, 4$. We can store these points in a single matrix, X, as follows:

$$X = \begin{bmatrix} \mathbf{p}1 & \mathbf{p}2 & \mathbf{p}3 & \mathbf{p}4 \end{bmatrix} = \begin{bmatrix} 0 & 4 & 4 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix} \tag{13}$$

We can then scale the entire shape by transforming it with the scale transformation from Example 1 (Equation 10) as follows:

$$X' = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 4 & 4 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 8 & 8 & 0 \\ 0 & 0 & 6 & 6 \end{bmatrix}. \tag{14}$$

Figure 1 shows the original and transformed shapes.

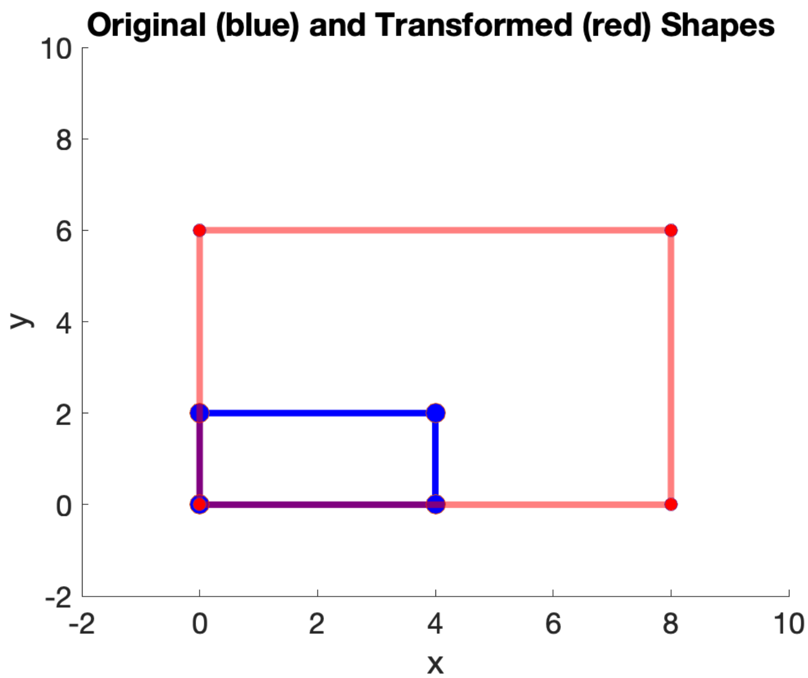

Figure 1: Scaling a rectangle.