CS 70 - Foundations Of Applied Computer Science

Carter Kruse

Reading Assignment 3

CS 70 - Chapter 5

Linear Maps

Geometric Intuition: A *linear map* (*linear mapping/transformation/function* is a function that maps vectors to vectors while ensuring the following:

- Straight lines remain straight lines.
- Origin (the zero vector) stays fixed.
- Parallel lines remain parallel.
- Length ratios are preserved.
- Is closed under composition.

Definition: A function $T : V \to W$, where V and W are vector spaces, is linear if it satisfies:

- $T(\boldsymbol{u} + \boldsymbol{v}) = T(\boldsymbol{u}) + T(\boldsymbol{v})$ (Additivity)
- $T(\alpha \boldsymbol{u}) = \alpha T(\boldsymbol{u})$ (Homogeneity)

for all vectors $\boldsymbol{u}, \boldsymbol{v} \in V$ and scalar α.

Superposition: $T(\alpha \boldsymbol{u} + \beta \boldsymbol{v}) = \alpha T(\boldsymbol{u}) + \beta T(\boldsymbol{v})$

Definition: A map $T : \mathbb{R}^m \to \mathbb{R}^n$ is linear if it can be expressed as

$$T(\boldsymbol{x}) = T(x_1, \ldots, x_m) = \sum_{i=1}^{m} x_i \boldsymbol{a}_i$$

In other words, it uses the m elements of the input vector $\boldsymbol{x}$ to perform a linear combination of a fixed set of n-vectors $\{\boldsymbol{a}_i\}$.

Theorem: Every linear map $T : \mathbb{R}^m \to \mathbb{R}^n$ can be expressed as a matrix-vector multiplication $A_{n \times m} \boldsymbol{x}$ by setting the i-th column of A to the i-th m-dimensional standard basis vector $\boldsymbol{e}_i$ transformed by T, i.e. $\boldsymbol{a}_i = T(\boldsymbol{e}_i)$.

$$T(\boldsymbol{x}) = \begin{bmatrix} | & & | \\ \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_m \\ | & & | \end{bmatrix} \boldsymbol{x} = \sum_{i=1}^{m} x_i \boldsymbol{a}_i$$

Geometric Transformations

The machinery of linear algebra allows us to perform many of the operations needed to represent shapes on a computer, arrange and manipulate them, and view them on screen.

Scaling: The most basic geometric transform is to scale an object along the coordinate axes. We can accomplish this by placing the desired x and y scale factors along the corresponding entries of a diagonal matrix.

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

Shear: A shear allows us to push things askew either horizontally or vertically (or both).

$$H = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix}$$

Rotation: A 2D rotation matrix rotates a shape counter-clockwise by an angle θ about the origin (again, the origin always stays fixed with a linear transformation).

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

Composition: It is common to want to apply more than one transformation to a shape. For instance, we may want to first scale a shape, and then rotate it. If we denote these two mappings as functions $S : \mathbb{R}^2 \to \mathbb{R}^2$ and $R : \mathbb{R}^2 \to \mathbb{R}^2$, then we can chain them together by passing a vector through the transformations in sequence.

This function composition is often denoted using "circle notation" as $(R \circ S)(p) = R(S(p))$, and it is valid as long as the output of the first transform matches the expected input of the second.

Order Matters! It is important to note that all three of these notations mean we first transform by S and then pass the result into R. To understand the effect on a vector $\boldsymbol{p}$ it is easier to read expressions like $RS\boldsymbol{p}$ from right to left: p gets transformed by S, and then the result of that gets transformed by R.

3D Linear Transformations: All the concepts from 2D generalize quite naturally to 3D just by adding another row and column to our transformation matrices.

Affine Maps: Affine maps or affine transformations combine linear maps with a translation. Such transformations inherit many of the properties of linear maps, but are enhanced to allow translation.

One way to represent an affine map is to carry around a translation vector along with a linear transformation matrix:

$$T(\boldsymbol{p}) = M\boldsymbol{p} + \boldsymbol{t}$$

where M takes care of scaling, shearing, rotation, etc, and $\boldsymbol{t}$ takes care of translation.

Homogeneous Coordinates: We can instead us a clever mathematical trick to allow us to express linear maps and translations using only matrix-vector multiplication. The idea is simple: we will append an extra coefficient of 1 (called a "homogeneous" coordinate) to our vectors.

We can then express affine transformations as a single matrix-vector multiplication:

$$T(\boldsymbol{p}) = M\boldsymbol{p} + \boldsymbol{t} = \begin{bmatrix} M & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \begin{bmatrix} M\boldsymbol{p} + \boldsymbol{t} \\ 1 \end{bmatrix}$$

The linear transformation M occupies the top-left block of the matrix, the translation t occupies the top-right block. The additional row of the matrix contains all zeros except a 1 in the bottom right entry, which ensures that this additional "homogeneous" coordinate always remains 1.