

DRAFT: Mathematical Background for Three-Dimensional Computer Graphics

Jonathan R. Senning
Gordon College

September 2006

Contents

1	Introduction	1
2	Affine Geometry	3
2.1	Affine Space	3
2.2	Affine Combinations	6
2.3	Euclidean Space	7
3	Frames and Homogeneous Coordinates	9
3.1	Frames	9
3.2	Homogeneous Coordinates	12
4	Affine Transformations	15
4.1	Affine Transformation of Points	15
4.2	Affine Transformation of Vectors	16
4.3	Properties of Affine Transformations	17
4.4	Matrix Representation of Affine Transformations	18
5	Geometric Transformations	21
5.1	Rotation about an arbitrary axis	21
6	Shadows and Reflections	25
6.1	Shadows from a point light source	25
6.2	Shadows from a light at infinity	27
6.3	Reflections	28

Chapter 1

Introduction

Introductory Example: The Rendering Pipeline in a Raster Graphics System.

To understand why linear algebra plays such an important role in computer graphics it is necessary to understand something of the entire process from model construction to image display. The process of *rendering* or displaying three-dimensional computer graphics can be broken down into several steps. First a *model* of the three-dimensional scene must be constructed. A model is constructed of primitive shapes such as polygons which have various linear transformations applied to them to orient and position them relative to each other. Next the three-dimensional model must be *projected* or *mapped* into two-dimensions so that it can be displayed. Finally the two-dimensional data must be converted into a form that the computer can display (*rasterization*) and then be displayed. The operations that must be performed by the computer are organized into a *rendering pipeline*, shown in Figure 1.1.

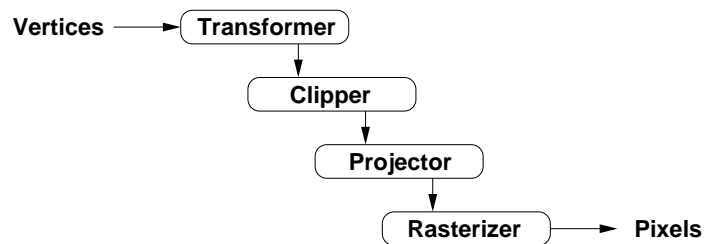


Figure 1.1: Graphics Pipeline

Models are constructed using a *world coordinate system*. This coordinate system is meaningful for the model being constructed and is completely independent of the computer graphics system being used. Models are composed of *objects* that have attributes such as color and that are oriented and positioned in the world coordinate system. These objects are usually described by a set of polygons or curved surfaces that fit together. These in turn are each described by sets of points that are either *vertices* of polygons or *control points* for curved surfaces. Depending on the location of the observer

and the point in the scene being viewed, some objects in a scene may not be entirely visible or even visible at all. Once the viewing parameters have been determined the model can be *clipped* so that only objects that will be visible are retained.

Next, the three-dimensional coordinates must be projected into two-dimensions. This requires not only the model specification but also a *viewpoint* or *direction of projection* and a *viewing* or *projection plane*.

During rasterization all visible points on the object must be shown. Fortunately, because of how graphics displays work, we don't need to draw the infinite number of points found on the visible portion of each object. Graphics displays are usually divided either physically or logically into *pixels*, short for *picture elements*. As the display contains a finite number of pixels, we need only determine what color to assign to each pixel. An area of memory in the computer called the *frame buffer* is dedicated to holding the data for each pixel and a *display controller* scans this memory and updates the display between 60 and 100 times each second.

One desirable attribute of interactive computer graphics systems is speed. Consider a video game or flight simulator where images must be generated quickly enough that our visual system sees them as continuous; usually at least 30 frames per second. Each image may in turn consist of thousands of polygons fitted together to create a scene. Even today's personal computer systems can often display millions of polygons each second. To accomplish high rates of speed the rendering steps are carried out using special computer hardware built in to the display system of the computer, with different rendering steps being carried out concurrently. The pipeline must be "primed" but once vertices are flowing into it one vertex can be transformed while another is being clipped and yet another is having a projection applied to it.

It is assumed that the reader is already familiar with basic linear algebra and vector spaces, as well as simple transformations used in computer graphics such as scaling and translation, perspective projections, and homogeneous coordinates. We will revisit each of these transformations and develop them in more detail. In addition we will also introduce other transformations and projections which are used by real-time computer graphics systems to draw reflections and shadows.

Chapter 2

Affine Geometry

Recall that vectors in \mathbb{R}^n are commonly written as

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 5 \\ 2 \\ 3 \end{bmatrix}.$$

Sometimes, in order to save space these vectors are written as $(1, 0)$ and $(5, 2, 3)$. Thus

$$\begin{bmatrix} -3 \\ 4 \end{bmatrix} \quad \text{and} \quad (-3, 4)$$

can both be interpreted as vectors in \mathbb{R}^2 .

One consequence of this approach is that it blurs the distinction between vectors and points; after all, in most other contexts $(-3, 4)$ is considered to be an ordered pair specifying the coordinates of a point in the xy -plane, not a vector. In some situations, however, the distinction between points and vectors is important as points and vectors have some mutually exclusive properties. A point has *location* but no extent while a vector in \mathbb{R}^n has both *direction* and *magnitude* but is location independent. Why is it, then, that we have been able to blur the distinction between them for so long?

This question will be answered in chapter 3 but first a new set of mathematical tools must be introduced.¹

2.1 Affine Space

An *affine geometry* is a geometry consisting of *points*, *free vectors* (or just *vectors*) and *scalars* (real numbers). Vectors will be denoted with bold lower-case Roman letters such as \mathbf{u} and \mathbf{v} and will be written as column vectors or as the transpose of a row vector. Points will be denoted with bold upper-case Roman letters (e.g. \mathbf{P} and \mathbf{Q}) and

¹The geometric framework introduced here is described more fully in the University of Washington Department of Computer Science Technical Report "Coordinate-free geometric programming" number 89-09-16 by Tony D. DeRose.

written using parentheses such as $(1, 2, 0)$. Scalars will appear as lower-case Greek letters: α, β , etc.

DEFINITION 2.1. An **affine space** A consists of the ordered pair (V, P) where V is a vector space and P is a set of points. Furthermore, vectors and points in A are related by the following two axioms:

i. *Subtraction Axiom:*

- (a) For every pair of points P and Q in P there is a unique vector $v \in V$ such that $v = P - Q$,
- (b) For every point $Q \in P$ and every vector $v \in V$ there is a unique point $P \in P$ such that $P - Q = v$.

ii. *Head-to-Tail Axiom:* Every triple of points P, Q and R satisfies

$$(P - Q) + (Q - R) = P - R.$$

It is convenient to use $A.V$ and $A.P$ to refer to the vectors and points in the affine space A . Finally, the dimension of an affine space is determined by the dimension of embedded vector space.

The first part of the subtraction axiom guarantees that there is one and only one vector in $A.V$ that “connects” every pair of points in $A.P$ while the second part ensures that there is a unique point in $A.P$ that can be “connected” with any particular point in $A.P$ using any particular vector in $A.V$. The second axiom is merely a statement of the familiar “head-to-tail” rule for the addition of vectors.

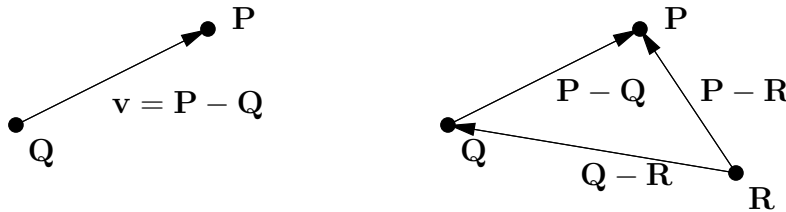


Figure 2.1: Axioms for Operations in an Affine Space

It is useful to define the operation of addition of a point and a vector. We define $v + Q$ to be the unique point P such that $P - Q = v$.

THEOREM 2.1. Let the vectors u and v and the points P, Q and R be in the same affine space. Then

- a. $P - P = 0$,
- b. $R - Q = -(Q - R)$,

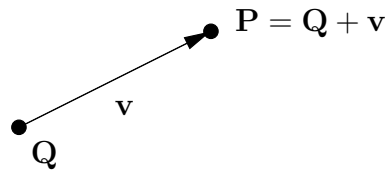


Figure 2.2: Sum of a point and a vector

$$\text{c. } \mathbf{v} + (\mathbf{Q} - \mathbf{R}) = (\mathbf{Q} + \mathbf{v}) - \mathbf{R},$$

$$\text{d. } \mathbf{Q} - (\mathbf{R} + \mathbf{v}) = (\mathbf{Q} - \mathbf{R}) - \mathbf{v},$$

$$\text{e. } \mathbf{P} = \mathbf{Q} + (\mathbf{P} - \mathbf{Q}),$$

$$\text{f. } (\mathbf{Q} + \mathbf{u}) - (\mathbf{R} + \mathbf{v}) = (\mathbf{Q} - \mathbf{R}) + (\mathbf{u} - \mathbf{v}).$$

Proof. Replace \mathbf{Q} with \mathbf{P} in the head-to-tail axiom to get

$$(\mathbf{P} - \mathbf{P}) + (\mathbf{P} - \mathbf{R}) = (\mathbf{P} - \mathbf{R}).$$

Using the rules of vector addition and subtraction we see that $\mathbf{P} - \mathbf{P} = \mathbf{0}$, proving (a). To see (b) replace \mathbf{P} with \mathbf{R} in the head-to-tail axiom to obtain

$$(\mathbf{R} - \mathbf{Q}) + (\mathbf{Q} - \mathbf{R}) = \mathbf{0}$$

which, after subtracting $(\mathbf{R} - \mathbf{Q})$ from both sides, yields the desired result. The proofs of (c), (d) and (e) are left as exercises (See exercises 6–8). Showing (f) requires several steps:

$$\begin{aligned} (\mathbf{Q} + \mathbf{u}) - (\mathbf{R} + \mathbf{v}) &= [(\mathbf{Q} + \mathbf{u}) - \mathbf{R}] + [\mathbf{R} - (\mathbf{R} + \mathbf{v})] && \text{— by head-to-tail axiom} \\ &= [(\mathbf{Q} + \mathbf{u}) - \mathbf{R}] + [(\mathbf{R} - \mathbf{R}) - \mathbf{v}] && \text{— by part (d)} \\ &= [(\mathbf{Q} + \mathbf{u}) - \mathbf{R}] - \mathbf{v} && \text{— by part (a)} \\ &= [(\mathbf{Q} - \mathbf{R}) + \mathbf{u}] - \mathbf{v} && \text{— by part (c)} \\ &= (\mathbf{Q} - \mathbf{R}) + (\mathbf{u} - \mathbf{v}) && \text{— by associative property} \end{aligned}$$

This completes the proof. \square

The operations on the elements of an affine space are summarized in the following table. Note that absent from this list are the operations of the addition of points and the product of a point and a scalar.

Operation	Result
$\text{vector} - \text{vector}$	\rightarrow vector
$\text{vector} + \text{vector}$	\rightarrow vector
$\text{scalar} \cdot \text{vector}$	\rightarrow vector
$\text{point} - \text{point}$	\rightarrow vector
$\text{point} + \text{vector}$	\rightarrow point

EXAMPLE 1. Consider the affine space $A = (\mathbb{R}^2, P)$ where P is the set of points in the xy -plane. Suppose $\mathbf{P} = (4, 3)$ and $\mathbf{Q} = (1, 2)$. The operation $2(\mathbf{P} - \mathbf{Q})$ is defined and yields

$$2(\mathbf{P} - \mathbf{Q}) = 2((4, 3) - (1, 2)) = 2 \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}.$$

Note that distributive property does not apply; $2(\mathbf{P} - \mathbf{Q})$ is defined while $2\mathbf{P} - 2\mathbf{Q}$ is not.

EXAMPLE 2. Suppose A is an affine space such that $A.V$ contains the vector $\mathbf{v} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$ and $A.P$ contains the point $\mathbf{P} = (0, 5)$. Show that the point $\mathbf{Q} = (4, -1)$ is in $A.P$.

Solution Computing $\mathbf{P} - \mathbf{Q} = (0, 5) - (4, -1) = \begin{bmatrix} -4 \\ 6 \end{bmatrix} = -2\mathbf{v}$ which is in $A.V$ so \mathbf{Q} is must be in $A.P$.

2.2 Affine Combinations

In computer graphics it is often desirable to generate a point \mathbf{R} on a line passing through two specified points \mathbf{P} and \mathbf{Q} . This can be done by forming the sum of one of these points and a scalar multiple of the vector connecting the two points:

$$\mathbf{R} = \mathbf{Q} + \alpha(\mathbf{P} - \mathbf{Q}). \quad (2.1)$$

The vector $\mathbf{P} - \mathbf{Q}$ is shown on the left of Figure 2.3 and the desired point \mathbf{R} is shown on the right. Note the position of the point \mathbf{R} on the line through \mathbf{P} and \mathbf{Q} depends only on α . Furthermore, \mathbf{R} will be between \mathbf{P} and \mathbf{Q} when $0 < \alpha < 1$.

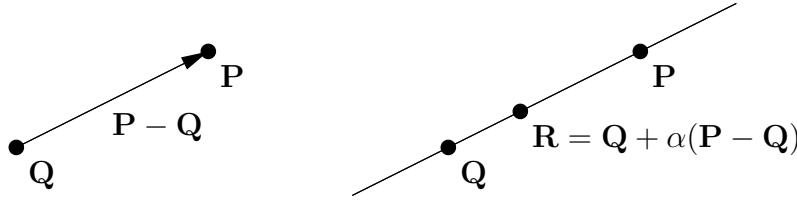


Figure 2.3: Affine Combination

If the distributive property could be applied to the right-hand-side of equation (2.1) we would obtain

$$\mathbf{R} = \mathbf{Q} + \alpha(\mathbf{P} - \mathbf{Q}) = \alpha\mathbf{P} + (1 - \alpha)\mathbf{Q}$$

suggesting that \mathbf{R} is a weighted average of the two points \mathbf{P} and \mathbf{Q} . When $\alpha = 0$ we find that $\mathbf{R} = \mathbf{Q}$ and when $\alpha = 1$ we have $\mathbf{R} = \mathbf{P}$. Note as well that if $\alpha = 1/2$ we have $\mathbf{R} = (\mathbf{P} + \mathbf{Q})/2$ which, although the notation is suspect, recalls the computation of the midpoint of a line segment. The difficulty is, of course, that the product of a scalar and

a point is not defined in general (what, after all, should the product of a scalar and a point represent?) We can deal with this difficulty by defining the *affine combination of two points*:

$$\alpha\mathbf{P} + \beta\mathbf{Q} \equiv \mathbf{Q} + \alpha(\mathbf{P} - \mathbf{Q}) \quad \text{where } \alpha + \beta = 1.$$

This same operation can be generalized for more than two points:

DEFINITION 2.2. Let A be an affine space with $A.P$ containing the points $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$ and let $\alpha_1, \alpha_2, \dots, \alpha_n$ be scalars such that $\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$. Then

$$\alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \dots + \alpha_n\mathbf{P}_n \equiv \mathbf{P}_1 + \alpha_2(\mathbf{P}_2 - \mathbf{P}_1) + \dots + \alpha_n(\mathbf{P}_n - \mathbf{P}_1). \quad (2.2)$$

It is worth noting that the values of the α_i are not constrained in any way other than that they sum to 1; negative as well as positive values are allowed. If the additional requirement that $0 \leq \alpha_i \leq 1$ is made then the combination becomes a *convex combination*.

2.3 Euclidean Space

Affine spaces provide a clear framework for distinguishing points and vectors, but many desirable and useful properties are missing. Most importantly, at least in the context of computer graphics, is that no *metric* exists, i.e., there is no concept of distance or the measure of an angle. This deficiency can be addressed by defining an *inner product* for an affine space. In the context of computer graphics the vector spaces embedded in affine spaces are usually one of the \mathbb{R}^n spaces so it makes sense to use the familiar *dot product* for an inner product. An affine space together with an inner product is called a *Euclidean Space*.

The *length* of a vector is computed as $\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$. Once vectors can have lengths assigned to them a vector can be *normalized* by scaling it by the reciprocal of its length, creating a *unit vector*.

Then *angle between two vectors*, θ , is defined as

$$\cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad \text{or} \quad \theta = \cos^{-1} \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}.$$

Recall that two vectors are *orthogonal* if their inner product is zero which occurs when $\theta = \pm\pi/2$, precisely the angles for which cosine is zero.

2.3 Exercises

1. Let $\mathbf{P} = (8, -2)$. Find a vector \mathbf{v} so that the point $(2, 5)$ can be written as $\mathbf{P} + \mathbf{v}$.
 2. Let A be an affine space whose vector space is spanned by the vector $[-1 \ 3]^T$ and that contains the point $(2, 5)$. Determine the affine combination that shows that the point $(15, -34)$ is in A .
 3. Let A be an affine space whose vector space is spanned by the vectors $[2 \ 0 \ 1]^T$ and $[0 \ -2 \ 5]^T$ and that contains the point $(7, -2, 5)$. Determine the affine combination that shows that the point $(17, 2, 0)$ is in A .
 4. Show that the straight line segment connecting points \mathbf{P} and \mathbf{Q} transforms into a straight line segment connecting $L(\mathbf{P})$ and $L(\mathbf{Q})$ under the linear transformation given by L .
 5. Show that by replacing α_1 with $1 - \alpha_2 - \dots - \alpha_n$ in equation (2.2) that every point \mathbf{P} in A can be written as the point \mathbf{P}_1 plus a vector.
 6. Prove part (c) in Theorem 2.1.
 7. Prove part (d) in Theorem 2.1.
 8. Prove part (e) in Theorem 2.1.
-

Chapter 3

Frames and Homogeneous Coordinates

In the last chapter it was pointed out that, at least in the context of computer graphics, using vectors in some vector space \mathbb{R}^n can lead to confusion between points and vectors. In addition, the very desirable operation of translation cannot be performed on vectors in \mathbb{R}^n without projecting these vectors into \mathbb{R}^{n+1} . In this chapter we'll not only move from the abstract concepts of affine and Euclidean spaces to a concrete coordinate system that can be used in computer graphics, but as a consequence, we'll see why it is that homogeneous coordinates provide such an elegant solution to the translation problem.

Recall that an *ordered basis* defines a coordinate system for a vector space. We shall see that a *frame* is to an affine space what a basis is to a vector space.

3.1 Frames

An n -dimensional affine space $A = (V, P)$ is specified by the vector space V and the set of points P . The n -dimensional vector space $A.V$ is completely described by providing a basis for it. From the definition of an affine space it is known that for every pair of points in $A.P$ there is a vector in $A.V$ that "connects" them. Once a particular point \mathbf{O} is selected from $A.P$ every other point in $A.P$ can be obtained by adding a vector from $A.V$ to \mathbf{O} . Therefore, supplying a basis for $A.V$ and a single point in $A.P$ is sufficient to specify the affine space A .

DEFINITION 3.1. A **frame** for the n -dimensional affine space $A = (V, P)$ consists of the set of basis vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ for $A.V$ and a point \mathbf{O} from $A.P$. The point \mathbf{O} locates the origin of the frame within A . We use the notation $\mathcal{F} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n, \mathbf{O})$ to denote a frame. Every vector \mathbf{u} in $A.V$ can be expressed as

$$\mathbf{u} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots + \alpha_n \mathbf{v}_n$$

and every point \mathbf{P} in $A.P$ can be written as

$$\mathbf{P} = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2 + \cdots + \beta_n \mathbf{v}_n + \mathbf{O}.$$

Specifying a frame for an affine space is equivalent to providing a *coordinate system* for it; once a frame has been determined any point or vector in the affine space can be described by a set of scalar values. To do this in matrix notation, however, the following definition must be made. This is often specified as a third axiom to Definition 2.1:

iii. *Coordinate Axiom:* For every point $\mathbf{P} \in A.P$

- (a) $0 \cdot \mathbf{P} \equiv \mathbf{0}$ (the zero vector),
- (b) $1 \cdot \mathbf{P} \equiv \mathbf{P}$ (the point P).

Note that in one instance the product is defined to be a vector while in the other the product is a point.

Now consider the frame $\mathcal{F} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{O})$ for an affine space $A = (\mathbb{R}^2, P)$. Any vector \mathbf{u} in $A.V$ can be written as

$$\mathbf{u} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{O}] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{bmatrix}$$

so that the column vector on the right is a *coordinate vector*. Similarly, a point \mathbf{P} in $A.P$ can be expressed as

$$\mathbf{P} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{O}] \begin{bmatrix} \beta_1 \\ \beta_2 \\ 1 \end{bmatrix}$$

The coordinate vectors of a vector \mathbf{v} and a point \mathbf{P} relative to frame \mathcal{F} are denoted $[\mathbf{v}]_{\mathcal{F}}$ and $[\mathbf{P}]_{\mathcal{F}}$ respectively. When writing coordinates we will write $(\beta_1, \beta_2, 1)_{\mathcal{F}}$ for a point and $(\alpha_1, \alpha_2, 0)_{\mathcal{F}}$ for a vector to make explicit the frame defining the coordinate system. In cases where there is just one frame being used we will sometimes drop the \mathcal{F} subscript.

EXAMPLE 1. Given the frame

$$\mathcal{F} = \left(\left(\begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \end{bmatrix}, (7, 1) \right) \right)$$

determine the point \mathbf{Q} that has the coordinates $(2, 3, 1)_{\mathcal{F}}$

Solution We use the coordinates to form a linear combination of the vectors in the frame which we then add to the frame's origin. Because we are adding a vector to a point the result will indeed be a point.

$$\mathbf{Q} = 2 \begin{bmatrix} 2 \\ 6 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ -2 \end{bmatrix} + 1 \cdot (7, 1) = \begin{bmatrix} 7 \\ 6 \end{bmatrix} + (7, 1) = (14, 7).$$

Often it is desirable to find the coordinates of a point relative to one frame given the coordinates of that point relative to another frame. This operation, called a *change of frames*, is analogous to the change of basis operation. Let $\mathcal{F} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{O})$ and $\mathcal{G} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{Q})$ be two frames for the 3-dimensional affine space A . To find $[\mathbf{P}]_{\mathcal{G}}$ given $[\mathbf{P}]_{\mathcal{F}} = [\alpha_1, \alpha_2, \alpha_3, 1]^T$ we must first write the basis vectors and point in \mathcal{F} in terms of the basis vectors and point in \mathcal{G} :

$$\begin{aligned}\mathbf{v}_1 &= a_1\mathbf{u}_1 + b_1\mathbf{u}_2 + c_1\mathbf{u}_3 \\ \mathbf{v}_2 &= a_2\mathbf{u}_1 + b_2\mathbf{u}_2 + c_2\mathbf{u}_3 \\ \mathbf{v}_3 &= a_3\mathbf{u}_1 + b_3\mathbf{u}_2 + c_3\mathbf{u}_3 \\ \mathbf{O} &= a_4\mathbf{u}_1 + b_4\mathbf{u}_2 + c_4\mathbf{u}_3 + \mathbf{Q}.\end{aligned}$$

Then

$$\begin{aligned}[\mathbf{P}]_{\mathcal{G}} &= [\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3 + \mathbf{O}]_{\mathcal{G}} \\ &= \alpha_1[\mathbf{v}_1]_{\mathcal{G}} + \alpha_2[\mathbf{v}_2]_{\mathcal{G}} + \alpha_3[\mathbf{v}_3]_{\mathcal{G}} + [\mathbf{O}]_{\mathcal{G}} \\ &= \begin{bmatrix} [\mathbf{v}_1]_{\mathcal{G}} & [\mathbf{v}_2]_{\mathcal{G}} & [\mathbf{v}_3]_{\mathcal{G}} & [\mathbf{O}]_{\mathcal{G}} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 1 \end{bmatrix}.\end{aligned}$$

This can be written more concisely as

$$[\mathbf{P}]_{\mathcal{G}} = M[\mathbf{P}]_{\mathcal{F}}$$

where the matrix

$$M = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is the *change of frame matrix*.

EXAMPLE 2. Let \mathcal{F} and \mathcal{G} be two frames for the same affine space such that

$$\mathcal{F} = \left(\begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \end{bmatrix}, (7, 1) \right), \quad \mathcal{G} = \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, (2, -1) \right).$$

If $[\mathbf{Q}]_{\mathcal{F}} = (2, 3, 1)_{\mathcal{F}}$ then find $[\mathbf{Q}]_{\mathcal{G}}$.

Solution The basis vectors in \mathcal{F} can be written

$$\begin{aligned} \begin{bmatrix} 2 \\ 6 \end{bmatrix} &= 3 \begin{bmatrix} 1 \\ 2 \end{bmatrix} - 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0 \cdot (2, -1), \\ \begin{bmatrix} 1 \\ -2 \end{bmatrix} &= -1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0 \cdot (2, -1), \\ (7, 1) &= 1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \cdot (2, -1) \end{aligned}$$

so the change of frame matrix M is

$$M = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 2 & 4 \\ 0 & 0 & 1 \end{bmatrix}.$$

Knowing M we can compute $[\mathbf{Q}]_{\mathcal{G}}$

$$[\mathbf{Q}]_{\mathcal{G}} = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 2 & 4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 1 \end{bmatrix}.$$

Note that using the frame \mathcal{G} we still recover the same point \mathbf{Q} as in example 1:

$$\mathbf{Q} = 4 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 8 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \cdot (2, -1) = \begin{bmatrix} 12 \\ 8 \end{bmatrix} + (2, -1) = (14, 7).$$

3.2 Homogeneous Coordinates

Recall that one advantage of using homogeneous coordinates is that they allow translation to be described as a linear transformation, thereby allowing all the transformations we would like to affect to be represented by matrix multiplication.¹ This is very important as permits a composite transformation to be represented by a single matrix which in turn means that arbitrarily complex transformations are no more expensive computationally than a single simple transformation.

EXAMPLE 1. Consider the frame $\mathcal{F} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{O})$ for the affine space A where the \mathbf{e}_i vectors form the standard basis for \mathbb{R}^3

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and \mathbf{O} is the point $(0, 0, 0)$. Let

$$\mathbf{v} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

¹For example translation in \mathbb{R}^2 can be implemented as shear in the plane of \mathbb{R}^3 containing vectors $[x, y, 1]^T$.

be a vector in \mathbb{R}^3 and let $\mathbf{P} = (\beta_1, \beta_2, \beta_3)$ be a point. The coordinate vectors for \mathbf{v} and \mathbf{P} relative to the frame \mathcal{F} are

$$[\mathbf{v}]_{\mathcal{F}} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 0 \end{bmatrix} \quad \text{and} \quad [\mathbf{P}]_{\mathcal{F}} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ 1 \end{bmatrix}$$

The choice of basis vectors in example 1 was not arbitrary. When the standard basis vectors (which are orthogonal unit vectors) of \mathbb{R}^n are chosen then the coordinates of points relative to that frame will be homogeneous coordinates.

Actually the definition of homogeneous coordinates is more general than this, and although the coordinates of a point relative to a particular frame are homogeneous coordinates, it is possible to have homogeneous coordinates that are not coordinates of a point relative to a frame. The key difference is that it is possible for the last coordinate, which is 1 when we consider the coordinates of a point relative to a frame, to be some nonzero number other than one. This is made explicit by the definition of homogeneous coordinates:

DEFINITION 3.2. We define (x, y, z, w) to be a **homogeneous coordinate representation** of the point $(x/w, y/w, z/w)$ when $w \neq 0$. Thus the three-dimensional coordinate (x, y, z) is represented by a ray in four-dimensions starting at (but not including) the origin and passing through the point $(x, y, z, 1)$.

Normally transformations applied to points in homogeneous coordinates do not move the point off the $w = 1$ plane (we will, however, see one example of how this can occur, forcing us to *renormalize* by dividing the coordinates by w) so that, at least in the context of computer graphics, we are justified in equating frame coordinates and homogeneous coordinates.

We will use the convention that point notation such as (x, y, z) will be used to describe points in the usual way and points in an affine space by listing the coordinates, such as $(x, y, z, 1)$; whether or not the point is expressed in homogeneous coordinates will be clear from the context.

Chapter 2 began with the question “Why is it possible to blur the distinction between points and vectors?” The answer is that, when working in a space such as \mathbb{R}^3 , there is an assumed frame that has an origin. By taking the origin to be $(0, 0, 0)$ we find that a point has exactly the same coordinates as a vector from the origin to that point, thus allowing us to use them interchangeably. Of course, when the origin or the the basis vectors change the coordinates themselves need to change.

3.2 Exercises

1. Explain why the set of vectors $[x, y, z, 1]^T$ is not a subspace of \mathbb{R}^4 .
2. Suppose that an affine space A has two frames

$$\mathcal{F} = \left(\left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 0 \\ 2 \end{array} \right], (1, -5) \right)$$

and

$$\mathcal{G} = \left(\left[\begin{array}{c} 1 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ -1 \end{array} \right], (-3, 1) \right).$$

3. Find the change of frame matrix M and use it to compute $[\mathbf{Q}]_{\mathcal{G}}$ if $[\mathbf{Q}]_{\mathcal{F}} = (1, 3, 1)_{\mathcal{F}}$.
 3. Suppose M is the change of frame matrix that transforms coordinates relative to frame \mathcal{F} to coordinates relative to frame \mathcal{G} . Prove that M^{-1} exists.
 4. Let $\mathbf{Q} = \alpha_1 \mathbf{P}_1 + \alpha_2 \mathbf{P}_2 + \cdots + \alpha_n \mathbf{P}_n$ be an affine combination of points in an affine space with frame \mathcal{F} . Show that $[\mathbf{Q}] = \alpha_1 [\mathbf{P}_1] + \alpha_2 [\mathbf{P}_2] + \cdots + \alpha_n [\mathbf{P}_n]$.
-

Chapter 4

Affine Transformations

As discussed in Chapter 1 there are many steps involved between specifying vertices in world coordinates and actually displaying the polygon defined by those vertices. Typically geometric transformations such as rotation or scaling are applied first, often as part of the modeling process, and then projection transformations are applied to map the three-dimensional data to the two-dimensional viewing plane. A final transformation maps this data to the grid of pixels on the display device. In this chapter we seek an answer to the question “what effect do the geometric, projection and rendering transformations have on the shapes and other properties of objects?” To answer this we first define *affine transformations*, which can be applied both points and vectors.

4.1 Affine Transformation of Points

DEFINITION 4.1. Let A and B be two affine spaces. The map $F : A.P \rightarrow B.P$ is an **affine transformation** if it preserves affine combinations, i.e.

$$F(\alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \cdots + \alpha_n\mathbf{P}_n) = \alpha_1F(\mathbf{P}_1) + \alpha_2F(\mathbf{P}_2) + \cdots + \alpha_nF(\mathbf{P}_n).$$

EXAMPLE 1. Let $A = (V, P)$. Show that $T(\mathbf{P}) = \mathbf{P} + \mathbf{t}$ is an affine transformation for any vector $\mathbf{t} \in A.V$.

Solution Let $\mathbf{P}_1, \dots, \mathbf{P}_n$ be points in $A.P$ such that $P = \alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \cdots + \alpha_n\mathbf{P}_n$ for some set of scalars α_i where $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$. Then

$$\begin{aligned} T(\alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \cdots + \alpha_n\mathbf{P}_n) &= \alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \cdots + \alpha_n\mathbf{P}_n + \mathbf{t} \\ &= \alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \cdots + \alpha_n\mathbf{P}_n + (\alpha_1 + \alpha_2 + \cdots + \alpha_n)\mathbf{t} \\ &= \alpha_1(\mathbf{P}_1 + \mathbf{t}) + \alpha_2(\mathbf{P}_2 + \mathbf{t}) + \cdots + \alpha_n(\mathbf{P}_n + \mathbf{t}) \\ &= \alpha_1T(\mathbf{P}_1) + \alpha_2T(\mathbf{P}_2) + \cdots + \alpha_nT(\mathbf{P}_n). \end{aligned}$$

This affine transformation is, of course, translation.

EXAMPLE 2. Suppose \mathbf{P} and \mathbf{Q} are two points in an affine space A . We know that the set of points on the line segment between them are described by $(1 - \alpha)\mathbf{P} + \alpha\mathbf{Q}$ for $0 \leq \alpha \leq 1$. If F is an affine transformation then

$$F((1 - \alpha)\mathbf{P} + \alpha\mathbf{Q}) = (1 - \alpha)F(\mathbf{P}) + \alpha F(\mathbf{Q})$$

which, as α moves from 0 to 1, describes the set of points on a straight line from $F(\mathbf{P})$ to $F(\mathbf{Q})$. Thus we see that an affine transformation preserves straight lines.

The consequences of the conclusion of this example are rather important, especially in the context of computer graphics. Since affine transformations preserve straight lines, only the vertices of polygons need be transformed in the rendering pipeline; the straight line connecting the transformed points can be generated by affine combination after the transformation.

4.2 Affine Transformation of Vectors

The affine transformation of points has been defined, but what about transforming vectors? It may seem that transforming points is sufficient since geometric primitives like polygons and curved surfaces are specified by sets of points, but there is at least one important case where transformations of vectors are required, that of *surface normals*.

The process of *shading* involves adjusting the color assigned to each point on the surface of an object according to a *lighting model*. The lighting model describes the complex interaction between the position and color of each light in a scene with the color, texture and position of each visible point on the surface of each object in the scene. The computations involved require the angle of incidence between a light ray from each source and the surface normal each each point on the surface. A *surface normal at a point* is a vector that is perpendicular to a plane tangent to the surface at that point.

In modern computer graphics application programming interfaces (APIs) the programmer often specifies a surface normal for each vertex. When transformations such as rotation or translation are applied to an object, it is important that the normals are transformed as well. Figure 4.1 shows a polygon with a set associated surface normals both before and after a rotation operation R which is, as will be seen shortly, an affine transformation.

Fortunately it is easy to extend the definition of affine transformations to include the transformation of vectors.

DEFINITION 4.2. Let A and B be two affine spaces and let \mathbf{v} be any vector in $A.V$. Choose any two points \mathbf{P} and \mathbf{Q} from $A.P$ such that $\mathbf{v} = \mathbf{P} - \mathbf{Q}$. The **affine transformation** $F : A.V \rightarrow B.V$ is

$$F(\mathbf{v}) = F(\mathbf{P} - \mathbf{Q}) \equiv F(\mathbf{P}) - F(\mathbf{Q}).$$

Notice that according to the definition any pair of points whose difference is the vector \mathbf{v} can be used for \mathbf{P} and \mathbf{Q} . In particular if $\mathbf{P}' - \mathbf{Q}' = \mathbf{v}$ then it must be the case that $F(\mathbf{P}) - F(\mathbf{Q}) = F(\mathbf{P}') - F(\mathbf{Q}')$; see exercise 1.

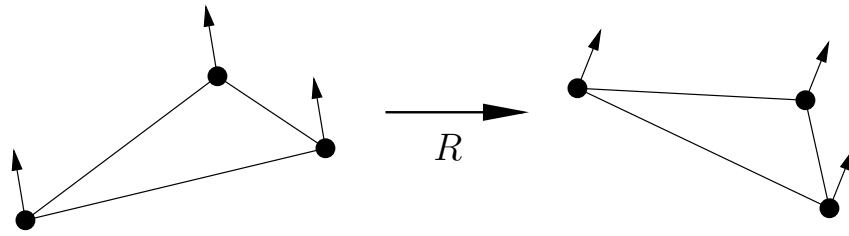


Figure 4.1: Rotation of polygon and surface normals

EXAMPLE 1. Show that if F is an affine transformation then

$$F(\mathbf{P} + \alpha\mathbf{v}) = F(\mathbf{P}) + \alpha F(\mathbf{v}).$$

Solution Recall from the definition of affine combinations that

$$(1 - \alpha)\mathbf{P} + \alpha\mathbf{Q} = \mathbf{P} + \alpha(\mathbf{Q} - \mathbf{P})$$

and let \mathbf{Q} be a point such that $\mathbf{v} = \mathbf{Q} - \mathbf{P}$. Then

$$\begin{aligned} F(\mathbf{P} + \alpha\mathbf{v}) &= F(\mathbf{P} + \alpha(\mathbf{Q} - \mathbf{P})) \\ &= F((1 - \alpha)\mathbf{P} + \alpha\mathbf{Q}) \\ &= (1 - \alpha)F(\mathbf{P}) + \alpha F(\mathbf{Q}) \\ &= F(\mathbf{P}) + \alpha(F(\mathbf{Q}) - F(\mathbf{P})) \\ &= F(\mathbf{P}) + \alpha F(\mathbf{Q} - \mathbf{P}) \\ &= F(\mathbf{P}) + \alpha F(\mathbf{v}). \end{aligned}$$

4.3 Properties of Affine Transformations

The properties of affine transformations on points and vectors are summarized in the following theorem.

THEOREM 4.1. Let \mathbf{P} and \mathbf{Q} be points and \mathbf{u} and \mathbf{v} be vectors in an affine space A . Let $F : A \rightarrow B$ be an affine transformation from A to another affine space B . Then for all scalars α and β

- $F(\alpha\mathbf{P} + \beta\mathbf{Q}) = \alpha F(\mathbf{P}) + \beta F(\mathbf{Q})$,
- $F(\mathbf{v}) = F(\mathbf{P} - \mathbf{Q}) = F(\mathbf{P}) - F(\mathbf{Q})$ when $\mathbf{v} = \mathbf{P} - \mathbf{Q}$,
- $F(\mathbf{P} + \alpha\mathbf{v}) = F(\mathbf{P}) + \alpha F(\mathbf{v})$,

d. $F(\mathbf{u} + \mathbf{v}) = F(\mathbf{u}) + F(\mathbf{v}),$

e. $F(\alpha\mathbf{v}) = \alpha F(\mathbf{v}).$

Proof. The first two properties are the definition of an affine transformation of a point and a vector. The proof of part (c) was done in example 1.

Showing part (d) is straight forward if \mathbf{P} and \mathbf{Q} are points in $A.P$ such that $\mathbf{u} = \mathbf{P} - \mathbf{Q}$ and $\mathbf{v} = \mathbf{Q} - \mathbf{R}$ and the head-to-tail axiom is applied several times.

$$\begin{aligned} F(\mathbf{u} + \mathbf{v}) &= F((\mathbf{P} - \mathbf{Q}) + (\mathbf{Q} - \mathbf{R})) \\ &= F(\mathbf{P} - \mathbf{R}) \\ &= F(\mathbf{P}) - F(\mathbf{R}) \\ &= F(\mathbf{P}) - F(\mathbf{Q}) + F(\mathbf{Q}) - F(\mathbf{R}) \\ &= F(\mathbf{P} - \mathbf{Q}) + F(\mathbf{Q} - \mathbf{R}) \\ &= F(\mathbf{u}) + F(\mathbf{v}). \end{aligned}$$

The proof of part (e) is left as an exercise; see exercise 2. □

4.4 Matrix Representation of Affine Transformations

Suppose that A and B are n -dimensional and m dimensional affine spaces respectively. Let $\mathcal{A} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n, \mathbf{O}_A)$ and $\mathcal{B} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m, \mathbf{O}_B)$ be frames for A and B . Suppose further that an affine transformation F exists such that $F : A \rightarrow B$ so that if \mathbf{P} is a point in $A.P$ then $\mathbf{Q} = F(\mathbf{P})$ is a point in $B.P$. Finally, let $[\mathbf{P}]_{\mathcal{A}} = [\alpha_1 \ \alpha_2 \ \cdot \ \cdot \ \cdot \ \alpha_n \ 1]^T$. Then

$$\begin{aligned} \mathbf{Q} &= F(\mathbf{P}) \\ &= F(\alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2 + \dots + \alpha_n\mathbf{u}_n + \mathbf{O}_A) \\ &= \alpha_1F(\mathbf{u}_1) + \alpha_2F(\mathbf{u}_2) + \dots + \alpha_nF(\mathbf{u}_n) + F(\mathbf{O}_A) \end{aligned}$$

where the last step is possible because of properties (c), (d) and (e) of Theorem 4.1 (See exercise 3). Thus

$$\begin{aligned}
 [\mathbf{Q}]_{\mathcal{B}} &= [\alpha_1 F(\mathbf{u}_1) + \alpha_2 F(\mathbf{u}_2) + \cdots + \alpha_n F(\mathbf{u}_n) + F(\mathbf{O}_{\mathcal{A}})]_{\mathcal{B}} \\
 &= \left[\begin{array}{cccccc} [F(\mathbf{u}_1)]_{\mathcal{B}} & [F(\mathbf{u}_2)]_{\mathcal{B}} & \cdots & \cdots & [F(\mathbf{u}_n)]_{\mathcal{B}} & [F(\mathbf{O}_{\mathcal{A}})]_{\mathcal{B}} \end{array} \right] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_n \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} & a_{m,n+1} \\ 0 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_n \\ 1 \end{bmatrix}
 \end{aligned}$$

since

$$[F(\mathbf{u}_1)]_{\mathcal{B}} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ \vdots \\ a_{m1} \\ 0 \end{bmatrix}, \quad [F(\mathbf{u}_2)]_{\mathcal{B}} = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ \vdots \\ a_{m2} \\ 0 \end{bmatrix}, \quad \cdots, \quad [F(\mathbf{O}_{\mathcal{A}})]_{\mathcal{B}} = \begin{bmatrix} a_{1,n+1} \\ a_{2,n+1} \\ \vdots \\ \vdots \\ a_{m,n+1} \\ 1 \end{bmatrix}.$$

The matrix

$$T = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} & a_{m,n+1} \\ 0 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

is the *standard matrix of the affine transformation*. In the common cases of two- and three-dimensional affine spaces T has the form

$$T = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad T = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

EXAMPLE 1. Let $A = (\mathbb{R}^2, P)$ be an affine space with $\mathcal{A} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{O})$ where $\mathbf{O} = (0, 0)$. Let $T : A \rightarrow A$ be defined as $T(\mathbf{P}) = \mathbf{P} + \mathbf{t}$ where $\mathbf{t} = [\alpha \ \beta]^T$. Find the 3×3 matrix T that implements this transform.

Solution Since only frame \mathcal{A} is used we have

$$[T(\mathbf{e}_1)]_{\mathcal{A}} = [\mathbf{e}_1]_{\mathcal{A}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad [T(\mathbf{e}_2)]_{\mathcal{A}} = [\mathbf{e}_2]_{\mathcal{A}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

while

$$[T(\mathbf{O})]_{\mathcal{A}} = [\mathbf{O} + \mathbf{t}]_{\mathcal{A}} = [\mathbf{O}]_{\mathcal{A}} + [\mathbf{t}]_{\mathcal{A}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}.$$

Thus the matrix T is given by

$$\begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{bmatrix}$$

which is the the standard matrix for translation in two-dimensions using homogenous coordinates.

4.4 Exercises

- Let $\mathbf{P}, \mathbf{Q}, \mathbf{P}', \mathbf{Q}'$ be points in an affine space A such that $\mathbf{P} - \mathbf{Q} = \mathbf{P}' - \mathbf{Q}'$. Show that $F(\mathbf{P}) - F(\mathbf{Q}) = F(\mathbf{P}') - F(\mathbf{Q}')$ for any affine transformation F . **This exercise needs a hint or perhaps the proof should be done in the text with one part left to an exercise.**
- Prove part (e) in Theorem 4.1.
- Prove that $F(\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n + \mathbf{P}) = \alpha_1 F(\mathbf{v}_1) + \alpha_2 F(\mathbf{v}_2) + \cdots + \alpha_n F(\mathbf{v}_n) + F(\mathbf{P})$ for scalars $\alpha_1, \dots, \alpha_n$, vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ and point \mathbf{P} .
- Determine the matrix representation of the affine transformation $S : A \rightarrow A$ if $A = (\mathbb{R}^2, P)$ and $S(\mathbf{P}) = \mathbf{Q}$ where $\mathbf{Q} = (\alpha x, \beta y)$ if $\mathbf{P} = (x, y)$. What type of transformation is this?
- Determine the matrix representation of the affine transformation $S : A \rightarrow A$ if $A = (\mathbb{R}^2, P)$ and $S(\mathbf{P}) = \mathbf{Q}$ where $\mathbf{Q} = (x+y, y)$ if $\mathbf{P} = (x, y)$. What type of transformation is this?

Chapter 5

Geometric Transformations

5.1 Rotation about an arbitrary axis

Suppose we want to rotate vertices by an angle θ about a line through the origin parallel to the **unit** vector $\mathbf{v} = [x \ y \ z]^T$. This can be accomplished by first performing one or two rotations to align \mathbf{v} with one of the coordinate axes, then rotating by an angle θ about that axis and finally undoing the first one or two rotations to reset v back to its original orientation.

In the derivation below we assume that \mathbf{v} is not along the x -axis, i.e., y and z are not both zero. If they are both zero then all we need to do is rotate about the x -axis and we are done.

Consider the situation diagrammed in Figure 5.1. We begin by rotating an amount

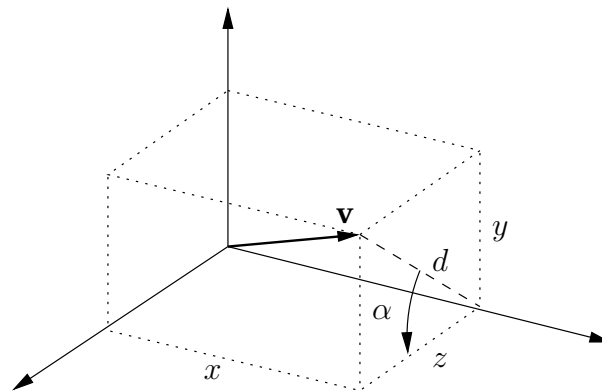


Figure 5.1: Original configuration

α about the x -axis to bring \mathbf{v} into the xz -plane. Call this new vector \mathbf{v}_{xy} . Notice that a right triangle exists with α as one of its angles and a hypotenuse of length

$d = \sqrt{y^2 + z^2}$. From this we deduce

$$\sin \alpha = \frac{y}{d}, \quad \cos \alpha = \frac{z}{d}.$$

The matrix $R_x(\alpha)$ is given by

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & z/d & -y/d & 0 \\ 0 & y/d & z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now we need to rotate about the y -axis by an angle $-\beta$ to align \mathbf{v}_{xy} with the z -axis. The angle is negative due to the clockwise rotation; see Figure 5.2. Notice that d is now

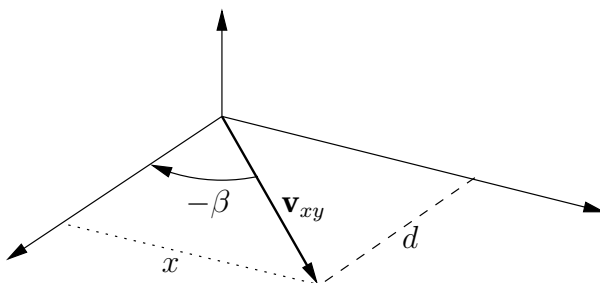


Figure 5.2: After the first rotation

the length of the side adjacent to the angle $-\beta$ and x is the length of the opposite side. The length of the hypotenuse is 1 since \mathbf{v} and hence \mathbf{v}_{xy} are both unit vectors. Thus we have

$$\sin(-\beta) = x, \quad \cos(-\beta) = d.$$

Using the fact that sine is odd and cosine is even, we have

$$\sin \beta = -x, \quad \cos \beta = d$$

and our rotation matrix $R_y(-\beta)$ is given

$$R_y(-\beta) = \begin{bmatrix} d & 0 & -x & 0 \\ 0 & 1 & 0 & 0 \\ x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrix $R_y(-\beta) R_x(\alpha)$ will align the rotation axis with the z -axis. Now we need to rotate the desired angle θ and then undo the rotations about the y and x axes. To rotate about the z -axis we need

$$R_z(\theta) = \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with $c = \cos \theta$ and $s = \sin \theta$. To undo the rotations about the other axes we will need the two matrices

$$R_x(-\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & z/d & y/d & 0 \\ 0 & -y/d & z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix} d & 0 & x & 0 \\ 0 & 1 & 0 & 0 \\ -x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Multiplying all of these together to find \mathbf{R} gives

$$R(\theta, x, y, z) = R_x(-\alpha) R_y(\beta) R_z(\theta) R_y(-\beta) R_x(\alpha)$$

which yields

$$R(\theta, x, y, z) = \begin{bmatrix} c(y^2 + z^2) + x^2 & (1-c)xy - sz & (1-c)xz + sy & 0 \\ (1-c)xy + sz & c(x^2 + z^2) + y^2 & (1-c)yz - sx & 0 \\ (1-c)xz - sy & (1-c)yz + sx & c(x^2 + y^2) + z^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

where $x^2 + y^2 + z^2 = 1$. Once this matrix is computed, rotations about the axis specified by $\mathbf{v} = [x \ y \ z]^T$ are no more computationally intensive than rotations about one of the coordinate axes.

5.1 Exercises

1. Determine the rotation matrix that implements rotation about the vector $\mathbf{v} = [1 \ 2 \ 0]^T$.
 2. The manual page for the OpenGL function `glRotatef()` shows a transformation matrix that agrees with that given in equation 5.1 except for the first three diagonal entries, which it lists as $x^2(1-c) + c$, $y^2(1-c) + c$, and $z^2(1-c) + c$. Show that these are equivalent to the first three diagonal entries shown in equation 5.1.
-

Chapter 6

Shadows and Reflections

The illumination models used by many interactive three-dimensional graphics systems make many approximations. One consequence of this is that normally shadows and reflections are not automatically drawn. Both shadows and reflections, however, provide important visual queues about the relative depths and surface characteristics of objects and the realism of a scene is greatly enhanced by including these visual effects. Since they are not automatically drawn, to include them we will have to draw them ourselves. The shadow for a particular object can be displayed by displaying the shadow of each polygon that composes the object. Reflections are created by drawing an object a second time but after applying transformations that locates the object at it's apparent position and orientation.

6.1 Shadows from a point light source

Consider the problem of creating a shadow cast by a single polygon onto another polygon. We assume that there is a single light source located at \mathbf{L} and that the polygon on which the shadow is cast is in the *shadow plane* described by $ax + by + cz + d = 0$.¹ The vertices of the shadow polygon are found by projecting rays from \mathbf{L} through the vertices of the polygon and computing where they intersect the shadow plane. In Figure 6.1 \mathbf{S} is one of the polygon's vertices and \mathbf{X} is the projected vertex on the shadow plane.

If the frame $\mathcal{F} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{O})$ is used then the coordinate vectors for \mathbf{L} , \mathbf{S} and \mathbf{X} are given by

$$[\mathbf{L}] = \begin{bmatrix} l_x \\ l_y \\ l_z \\ 1 \end{bmatrix}, \quad [\mathbf{S}] = \begin{bmatrix} s_x \\ s_y \\ s_z \\ 1 \end{bmatrix}, \quad \text{and} \quad [\mathbf{X}] = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

¹Recall that the vector $[a \ b \ c]^T$ is a normal vector for this plane; suggesting that given a polygon one can find the equation of the plane it defines by using the coordinates of one of the polygon's normal vectors.

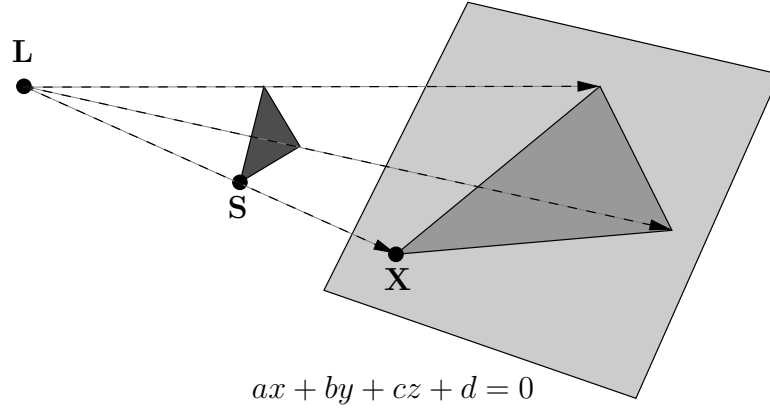


Figure 6.1: Vertex \mathbf{S} casts shadow from light at \mathbf{L} at point \mathbf{X} on plane $ax + by + cz + d = 0$

It is convenient to define \mathbf{n} as

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

which allows the equation $ax + by + cz + d = 0$ to be expressed as

$$\mathbf{n}^T[\mathbf{X}] = 0$$

when \mathbf{X} is in the shadow plane. The point \mathbf{X} also must satisfy

$$\mathbf{X} = \mathbf{S} + \alpha(\mathbf{L} - \mathbf{S}) \quad (6.1)$$

for some scalar α because \mathbf{X} is on the ray passing through both \mathbf{L} and \mathbf{S} . Combining these last two equations gives

$$\mathbf{n}^T[\mathbf{S}] = -\alpha[\mathbf{L} - \mathbf{S}]$$

or

$$\alpha = -\frac{\mathbf{n}^T[\mathbf{S}]}{\mathbf{n}^T[\mathbf{L}] - \mathbf{n}^T[\mathbf{S}]}$$

When this expression for α is substituted back into equation (6.1) we find that

$$(\mathbf{n}^T[\mathbf{L}] - \mathbf{n}^T[\mathbf{S}])[\mathbf{X}] = (\mathbf{n}^T[\mathbf{L}])[\mathbf{S}] - ([\mathbf{L}]\mathbf{n}^T)[\mathbf{S}]. \quad (6.2)$$

In arriving at this result the identity $(\mathbf{u}^T\mathbf{v})\mathbf{w} = (\mathbf{w}\mathbf{u}^T)\mathbf{v}$ for vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ was used (see exercise 5). The quantity $\mathbf{n}^T[\mathbf{L}]$ is a constant scalar while $[\mathbf{L}]\mathbf{n}^T$ is a 4×4 constant matrix. To factor $[\mathbf{S}]$ out from the right-hand-side we must introduce the identity matrix in much the same way as was done when solving the eigenvalue equation

$A\mathbf{x} = \lambda\mathbf{x}$. Setting $k = \mathbf{n}^T[\mathbf{L}]$ and factoring out $[\mathbf{S}]$ from the right-hand-side gives

$$(k - \mathbf{n}^T[\mathbf{S}])[\mathbf{X}] = (kI - [\mathbf{L}]\mathbf{n}^T)[\mathbf{S}].$$

Although it may not yet be clear, the matrix $M = kI - [\mathbf{L}]\mathbf{n}^T$ is the projection matrix we are looking for. To see this we perform the multiplication on the right-hand-side

$$\begin{aligned} (k - \mathbf{n}^T[\mathbf{S}])[\mathbf{X}] &= \begin{bmatrix} k - al_x & -bl_x & -cl_x & -dl_x \\ -al_y & k - bl_y & -cl_y & -dl_y \\ -al_z & -bl_z & k - cl_z & -dl_z \\ -a & -b & -c & k - d \end{bmatrix} \begin{bmatrix} s_x \\ s_y \\ s_z \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} ks_x - l_x\mathbf{n}^T[\mathbf{S}] \\ ks_y - l_y\mathbf{n}^T[\mathbf{S}] \\ ks_z - l_z\mathbf{n}^T[\mathbf{S}] \\ k - \mathbf{n}^T[\mathbf{S}] \end{bmatrix}. \end{aligned}$$

Once both sides are divided by the fourth element of the product $(k - \mathbf{n}^T[\mathbf{S}])$, an operation that is usually done automatically by graphics systems to put coordinate vectors into standard homogeneous form, we find that we have the coordinate vector for the point \mathbf{X} . To summarize:

1. Given a light position \mathbf{L} and a vector \mathbf{n} with the coefficients of the projection plane equation $ax + by + cz + d = 0$, the shadow projection matrix $M = \mathbf{n}^T[\mathbf{L}]I - [\mathbf{L}]\mathbf{n}^T$.
2. For each vertex \mathbf{S} on a polygon, compute $M[\mathbf{S}]$ and then “homogenize” this by dividing by the fourth component to get $[\mathbf{X}]$.

6.2 Shadows from a light at infinity

When the light source is very far away from the objects in a scene all the vectors $\mathbf{S} - \mathbf{L}$ are nearly parallel. This case often occurs, such as when the sun is modeled as a light source.

Let $\mathbf{l} = [l_x \ l_y \ l_z]^T$ be a vector specifying the direction of the light rays. Then, as before, we have

$$[\mathbf{l}] = \begin{bmatrix} l_x \\ l_y \\ l_z \\ 0 \end{bmatrix}, \quad [\mathbf{S}] = \begin{bmatrix} s_x \\ s_y \\ s_z \\ 1 \end{bmatrix}, \quad [\mathbf{X}] = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{n} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}.$$

As before we substitute $\mathbf{X} = \mathbf{S} + \alpha\mathbf{l}$ into $\mathbf{n}^T[\mathbf{S}] = 0$ and solve for α to obtain

$$\alpha = -\frac{\mathbf{n}^T[\mathbf{S}]}{\mathbf{n}^T[\mathbf{l}]}$$

which gives

$$\begin{aligned} [\mathbf{X}] &= [\mathbf{S}] - \frac{\mathbf{n}^T[\mathbf{S}]}{\mathbf{n}^T[\mathbf{l}]}[\mathbf{l}] \\ (\mathbf{n}^T[\mathbf{l}])[\mathbf{X}] &= (\mathbf{n}^T[\mathbf{l}])[\mathbf{S}] - (\mathbf{n}^T[\mathbf{S}])[\mathbf{l}] \\ &= (\mathbf{n}^T[\mathbf{l}])[\mathbf{S}] - ([\mathbf{l}]\mathbf{n}^T)[\mathbf{S}] \\ k[\mathbf{X}] &= (kI - [\mathbf{l}]\mathbf{n}^T)[\mathbf{S}] \end{aligned}$$

if $k = \mathbf{n}^T[\mathbf{l}]$. To find $[\mathbf{X}]$ we need only compute $(kI - [\mathbf{l}]\mathbf{n}^T)[\mathbf{S}]$ and divide by the fourth element of the product, which is k .

The matrix $M = kI - [\mathbf{l}]\mathbf{n}^T$ is the shadow projection matrix and has the form

$$M = \begin{bmatrix} k - al_x & -bl_x & -cl_x & -dl_x \\ -al_y & k - bl_y & -cl_y & -dl_y \\ -al_z & -bl_z & k - cl_z & -dl_z \\ 0 & 0 & 0 & k \end{bmatrix}.$$

What is useful to note is that M is identical to the shadow projection matrix for a point light source when $[\mathbf{L}]$ is replaced by $[\mathbf{l}]$, i.e., when the fourth element of the light coordinate vector is zero rather than one.

6.3 Reflections

In some ways reflections are easier to implement than shadows.

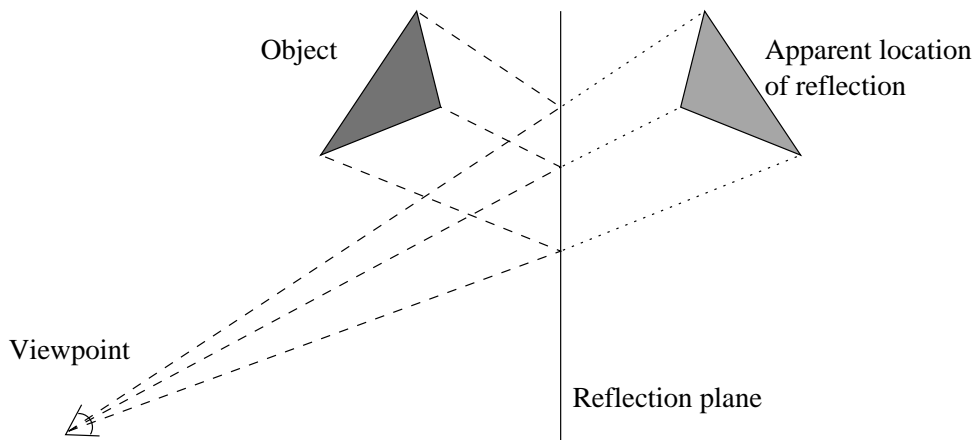


Figure 6.2: Typical reflection diagram

6.3 Exercises

1. Compute the shadow projection matrix M for a light source at $(-2, 5, 8)$ and a shadow projection plane given by $z = -10$.
 2. Compute the shadow projection matrix M for a light source at $(-2, 5, 8)$ and a shadow projection plane given by $3x + 2y - z + 5 = 0$.
 3. Find the coordinates of the point projected by the matrix M in exercise 1 from the point $(1, 3, 2)$.
 4. Find the coordinates of the point projected by the matrix M in exercise 2 from the point $(1, 3, 2)$.
 5. Show that $(\mathbf{u}^T \mathbf{v})\mathbf{w} = (\mathbf{w}\mathbf{u}^T)\mathbf{v}$ for $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^n$.
-

