Affine Geometry Lesson

George Francis

February 3, 2001

1 Introduction

This and the following two lessons amplify and explain in greater detail what was only summarized earlier. Moreover, the text dates back to 1989, when the documentation for the SGI graphics library first appeared. At that time, and not until OpenGL was documented, vectors were written as rows, and matrices operated on the right of vectors. As a result, matrix operations in the order they appeared in the code were applied to a vertex reading from left to right. Jim Clark, the author of `gl` and founder of Silicon Graphics, once remarked that he switched from math style (column vectors multiplied on their left) at the request of computer graphicists. It is a good exercise in linear algegra to "transpose" your thoughts periodically.

In order to understand the so-called *Geometry Pipeline* of the Silicon Graphics Iris, which is hardware and software heart of this computer, it helps to understand the basics of *affine* and *projective* geometry. This chapter provides the mathematical background for Chapter 7 of the 1989 edition of the **Guide**.

The word "geometry" has suffered linguistic abuse in the jargon of computer science which can no longer be corrected. However, as an inducement for you to avoid misusing this ancient word, let us review its traditional meaning in mathematics[1]. The writers of the **Guide** apply the identifyer of a science to the object of its study. Thus "...the clipping hardware eliminates all geometry [sic] with coordinates outside the –1.0 to 1.0 range". Compare this to "...Jupiter's astronomy is superficially similar to that of the solar system", or worse, "...students in Math428 manage their own computer science as long as it can be conveniently handled by the electrical engineering in the Renaissance Experimental Laboratory ..."

2 Point and Vector

Perhaps you still remember the difficulty you had with an apparent ambiguity in your first course on vector calculus. The number triple (x, y, z) could mean: the point in 3-space whose coordinates relative to some Cartesian

[1]Sometimes, this helps. An article by William Safire reminded the airlines that "momentarily" also means "from moment to moment". This successfully discouraged flight attendants and pilots from promising to take off or land "momentarily".

coordinate system are x, y, and z. It also means the vector with components x, y, z. After that you experienced a period of confusion[2] trying to distinguish between free and bound vectors with your own, or your professor's baroque notation.

What we really have here is two copies of the same set of number triplets, $\mathbb{R}^3$, used in two different ways. First, $\mathbb{R}^3$, is a way of naming and locating points, lines and planes in space. Secondly, it is a representation of an abstract algebraic thing called a *vector space*. Recall that vectors add, subtract, suffer scalar multipliplication, and have an additively neutral element called the zero vector. Between two vector spaces there are *linear transformations*, i.e. those point-wise mappings,

$$g : \mathrm{R}^n \to \mathrm{R}^p, \mathbf{v} \to \mathbf{v}g$$

which preserve linear combinations[3]:

$$(\sum x_i \mathbf{v}_i)g = \sum x_i(\mathbf{v}_i g)$$

A linear transformation g is *invertible* if there is an inverse h so that the composition is the identity, $gh = 1$. (The identity transformation is "obviously" a linear transformation. Why?). Among the linear transformations of a vector space with itself is the *general linear group* of invertible ones. For $n = 2, 3, 4...$ the general linear group of $\mathbb{R}^n$ is denoted by $\mathbb{GL}(\mathrm{n})$. The $n \times n$ matrices representing elements of $\mathbb{GL}(\mathrm{n})$ are precisely those with inverses[4].

Let's return to the other meaning of $\mathbb{R}^3$. To reduce a set of concepts to their bare essentials (memory being a limited commodity) mathematicians resort to the *axiomatic method*. That is, they define something abstractly even though they may be thinking primarily of one example thereof.

3 Affine Structures

An *affine structure*[5] is a set X with an associated *vector space* V, and a map

$$d : X \times X \to V : (x, a) \to d(x, a)$$

[2]If you are old enough to have studied from W. Kaplan's *Advanced Calculus* you may remember an echelon of WWII airplanes flying over page 32.

[3]To get you used to multiplying matrices on the right, as is the *Guide* convention, we shall apply function on the *right* of their arguments in this section.

[4]I.e. those with non-zero determinants. Can you prove the equivalence of these two qualities?

[5]cf p74ff of Ian Porteous, *Topological Geometry*, VanNostrand, 1969.

satisfying 3 rules.

The first rule says that X and V are point-wise the "same" set, but in disconcertingly many different ways:

(i) For every a in X the map $X \to V : x \to d(x, a)$ is bijective[6].

The next rules let you know that d is "really" the *difference* of two points. Porteous identifies two properties as axiomatic:

(ii) For every a in X we have that $d(a, a) = 0$.

(iii) For any three a, b, c in X we have $d(a, b) + d(b, c) = d(a, c)$.

Now you know what lies behind the two meanings of $\mathbb{R}^3$. For the time being we keep the symbol X for the affine space underlying the vector space $\mathbb{R}^3$. Elements of $\mathbb{GL}(\mathrm{n})$ because we think of them as nonsingular $n \times n$ matrices. To understand the group $A(n)$ of affine transformations on X, consider the equation in unknowns x, y and given by

$$d(y, a) = xQ$$

where a in X and Q in $\mathbb{GL}(\mathrm{n})$ are given. By rule (i) there is a unique solution y for every x. We might write the transformation determined by Q and a as $[Q, a]$ and write $y = x[Q, a]$. A pure *translation* would be $[I, a]$ where I is the identity in $\mathbb{GL}(3)$. Once a particular coordinate system is chosen for X we can write $d(y, a) = y - a$ and the solution to the equation $y - a = xQ$ can be written[7] $y = x[Q, a] = xQ + a$. The affine tranformation $[Q, 0]$ which has no translation component can be identified with the linear transformation Q.

The composition of two affine tranformations goes as follows:

$$[Q, a][P, b] = [QP, aP + b].$$

In particular, it matters in which order you combine the translation with the linear transformation $[Q, a] = [Q, 0][I, a]$ and not $[I, a][Q, 0]$, which is $[Q, Qa]$.

[6] 1:1 and onto.

[7] The square brackets should make you think of matrices.

4 Lie Groups

In $\mathbb{GL}$(n) there are some important subgroups. They are familiar examples of Lie groups[8]. A *subgroup* H of G is a subset of G closed under the operation of G, and containing the inverses of its elements. Together, these two properties of H imply (prove it!) that H is a group in its own right with the inherited operation. It will be easier to continue this exposition if we identify $\mathbb{GL}$(n) with its represenation as nonsingular $n \times n$ matrices.

Under multiplication, the nonzero real numbers, $\mathbb{R}^* = \mathbb{R}^1 - \{0\}$, form a group. The general linear group $\mathbb{GL}$(n) contains a copy of $\mathbb{R}^{*n}$, namely the diagonal matrices with nonzero entries. Rescaling the coordinate axes of a Cartesian coordinate system is represented by multiplication by a diagonal matrix in $\mathbb{GL}$(n).

The rotations form another, even more important, subgroup, $\mathbb{SO}$(n), called the *special orthogonal* group. The corresponding matrices are "orthonormal". For $n = 2$, $\mathbb{SO}(2)$ is pretty simple. There is only one kind of rotation about the origin, and it is completely specified by an angle. An angle, in turn, may be represented by a point on the unit circle, $\mathbb{S}^1$. Since the plane $\mathbb{R}^2$ also supports the complex numbers, you may think of $\mathbb{S}^1$ a the *unit complex numbers*,

$$\mathrm{S}^1 = \{\mathrm{e}^{it} | t \,\mathrm{real}\},$$

where $i^2 = -1$, with angle addition corresponding to complex multiplication:

$$\mathrm{e}^{it}\mathrm{e}^{is} = \mathrm{e}^{i(t+s)}.$$

That the rotations in 3-space form a subgroup is not geometrically obvious. Given two axes of rotation and an angular magnitude for each, can you tell right off that the composition of these two rotations is equivalent to a single rotation? What is the axis of this third rotation? It is, however, true though tedious to prove that the product of two orthonormal 3x3 matrices is again orthonormal. But if you want to know the resultant axis of rotation you have to know how to extract that from an orthonormal matrix?

There are two better ways of representing $\mathbb{SO}(3)$ which should make it possible for you to develop an intuition for rotations. My favorite one is the double cover of $\mathbb{SO}(3)$ by the three dimensional sphere, $\mathbb{S}^3$, of unit quaternions. The *quaternions* are Hamilton's generalizations of the complex numbers to dimension 4. That is, $\mathbb{R}^4$ supports a multiplication, division,

[8]Recall that a group G is a set together with an *operation* $G \times G \to G$, which satisfies 3 axioms. The operation is *associative*. There is a neutral element called the *identity* of G. Every element has an *inverse* (their product is the identity).

addition and subtraction precisely the way $\mathbb{R}^2$ supports the complex number system. The quaternions inherit addition from $\mathbb{R}^4$. But their product is like nothing else. Just as the magnitude of complex number is just its distance from the origin, so is it the case for quaternions, and the multiplicative subgroup of unit quaternions occupy the unit 3-sphere $\mathbb{S}^3$ in $\mathbb{R}^4$. By a *double cover* is meant a 2:1 mapping of $\mathbb{S}^3$ to $\mathbb{SO}(3)$. Two unit quaternions represent the same rotation in $\mathbb{R}^3$ exactly if they are antipodes on $\mathbb{S}^3$. Thus $\mathbb{SO}(3)$ also has the representation of 3-dimensional *real projective space*, $\mathbb{RP}(3)$. All of this requires a good deal more explanation, which is the subject of another lecture. For now it suffices to appreciate the plausibility that the product of two unit quaternions is again a unit quaternion. Given that to each unit quaternion there is a unique axis and angle of rotation of $\mathbb{R}^3$ about that axis, it follows that the composition of two rotations is itself again a rotation.

The subroup of affine transformations whose *linear part* is a rotation is the familiar group of *sense-preserving Euclidean transformations* which you remember under the name *congruence* from high-school geometry. *Sense preserving*[9] means that we have excluded the *reflections*[10].

5 Iris Geometry

In a later chapter we introduce hyperbolic geometry for 3-space. According to Thurston, there are only 8 different geometries for our 3-space, and of those only 3 are easily represented using the Iris geometry pipeline: Euclidean, hyperbolic and spherical.

So now you can guess what *a geometry* is. *Euclidean geometry* is $\mathbb{R}^3$ together with the Euclidean group. We say two objects[11] are the *same* if there is a Euclidean transformation that takes one object to the other. If one is a different size from the other, or not all corresponding angles are equal, then they are not equivalent in a Euclidean geometry. But they could be "the same" in the *affine geometry* on $\mathbb{R}^3$. Thus we can summarize by saying that the *modeling* and *viewing* portion of the Iris geometrical machinery involves the composition[12] of affine transformations. The two viewing commands: `polarview(...)` and `lookat(...)` are just commonly used complicated affine transformations. Each time you call one of these, it is spliced between the old transformation and the objects to be transformed. In otherwords, the newly called transformation is applied first to the vertices of a polyhedron, for example, and only then are the previously constructed affine transformations applied. This feature, together with the coding convention that it is the inverse of the desired transformation that has to be coded, requires a very

[9] Also *orientation preserving*.
[10] Which turn a left hand into a right one, and vice versa.
[11] E.g. stellated dodecahedra
[12] Catenation.

clear understanding of the "geometry engine". Here is the reason for this, and a way of remembering why it is so.

The geometers at Silicon Graphics, the company that builds the Iris, do not consider an affine transformation as something that transforms the objects to be displayed. They consider the affine transformation to act on the coordinate system. There is no harm in this attitude, it comes from the realm of movie maker[13] and landscape artist. It is not adapted to the attitude of the industrial designer[14] nor to the mathematician. The two attitudes are referred to as the "alias" and "alibi" approach to coordinate changes[15]. To change coordinates means to give a point a false name. To change its position is to give it an alibi as to its whereabouts. When thinking "alibi" in an "alias"ed graphics system, just remember to call the operations in reverse order and reverse direction. To translate the object to the right, call for a left translation, for a right handed rotation of 30 degrees about the y-axis of the object, call for a `rotate(-300,'y')` command.

Before you proceed to the next chapter, you should be made aware that there are subtle difference between `gl` and `OpenGL` which are deeper than a vocabulary change and the transposition of matrix algebra. There are books on the market which explain OpenGL, and you should learn to program in OpenGL. We include these chapters primarily to make `rtica` written in `gl` accessible. We also hope that this exposition of the older, more intuitive and much less complicated `gl` will inspire OpenGL experts to restore some of the good features which had been abandoned in the interest of platform independence and generality.

[13]It is easier to move the camera than the Empire State Building.

[14]You turn a pencil-sharpener upside down, you do not make the camera-man stand on his head

[15]Cf Birkhoff and MacLane.