A LINEAR ALGEBRAIC INTRODUCTION TO 3D COMPUTER GRAPHICS

JOSHUA R. DAVIS

1. Introduction

the basic steps:

describe some objects in $\mathbb{R}^3$

use rigid-body transformations (rotations and translations) to move them around to their desired locations

also use rigid-body transformations to place a virtual camera in the scene

the camera defines a projection of the 3D scene onto a 2D plane; then we can draw the projected scene using ordinary 2D graphics (which we will take as given).

If we do this 30 times per second, each time adjusting the objects and camera to indicate movement in the scene, then we have an animated film. Video games typically strive for higher frame rates, say 60 frames per second, to improve responsiveness.

2. Describing Objects in 3D

only need to model surfaces

a surface can be approximated by triangles joined together

we will be transforming these triangles in various ways

to transform a triangle, just need to transform its vertices

the sides of the triangle are line segments; they are still line segments after being transformed

linear algebra is “linear” because it preserves lines in this way

3. Rigid-Body Transformations

The rigid-body transformations of $\mathbb{R}^n$ are the transformations (functions) $f : \mathbb{R}^n \to \mathbb{R}^n$ that do not “distort” the space in any way. To be slightly more precise, they consist of only rotations and translations, and compositions of these.

SO(2): rotations in 2D

SO(3): rotations in 3D

move up to projective coordinates to handle translation

translation matrix...

in general, these rigid-body transformations do not commute with each other

4. Projection

orthogonal projection: isometric (preserves length relationships), so useful in drafting, but not realistic

perspective projection? need to understand projective geometry a little more now let the fourth coordinate be anything, not just 1; two columns are considered equal iff they are parallel; the column $[0,0,0,0]^T$ is not allowed.

then can write perspective as...

columns with a 0 in fourth coordinate are "at infinity"; these correspond to vanishing points in Renaissance drawing; they represent directions in space

5. Lighting

we want to place lights in our scene to make things more realistic

lighting is subtle, but we make simplification that there are two basic kinds of reflection of light off surfaces:

diffuse: in every direction, proportional to dot product between unit vector toward light and unit vector toward camera

specular: mirror reflection; shiny spot

to do these we need to know the unit normal vector at every point on the surface; in practice this normal information is part of the geometric model, along with the triangles

when we transform these vectors with 4×4 matrices, we put a 0 in their fourth entry, since they should be unchanged by translations; this makes them transform correctly (they represent directions, not really points)

6. Other Techniques

normals can also be used for backface culling; don't bother to draw triangles pointing away from camera; detect these by dotting normals with vector to camera

once you have lighting, you also want shadows; to cast a shadow onto the ground, project from light source onto ground plane

clipping: intersect triangles with boundary planes of viewing volume; just amounts to solving some linear equations

also cut off front and back of viewing volume; this is why rain/fog/smoke is used so much in films

scene graphs

7. Problems

In these problems, the matrices are all 4×4, acting on four-dimensional vectors with a 1 in the last entry, as discussed in class. For any vector $\langle a, b, c\rangle \in \mathbb{R}^3$, let $T_{\langle a,b,c\rangle}$ be the 4×4 matrix that represents translation by $\langle a, b, c\rangle$. For any angle α, let X_α be the 4×4 matrix that represents rotation about the x-axis through an angle of α (counterclockwise, following the right-hand rule). Similarly, let Y_β and Z_γ represent rotations about the y- and z-axes.

1. Write down the matrix $X_{-\pi/6}$ (simplifying $\cos -\pi/6$ and $\sin -\pi/6$, of course). Use it to rotate the standard basis vectors $\langle 1, 0, 0\rangle$, $\langle 0, 1, 0\rangle$, $\langle 0, 0, 1\rangle$. Draw a picture with all of these vectors (before and after rotation) labeled.

2. Do translations commute with each other? To answer this, write down two translation matrices $T_{\langle a,b,c\rangle}$ and $T_{\langle d,e,f\rangle}$, compute $T_{\langle a,b,c\rangle}T_{\langle d,e,f\rangle}$ and $T_{\langle d,e,f\rangle}T_{\langle a,b,c\rangle}$, and compare.

3. Do rotations about different axes (say, the x- and y-axes) commute with each other? To answer this, compute $X_\alpha Y_\beta$ and $Y_\beta X_\alpha$ and compare.

4. Do rotations commute with translations? Check this for the rotation X_α and the translation $T_{\langle a,b,c\rangle}$ as above.

5. In my 3D graphics program, I want to rotate the scene by $-\pi/6$ about the x-axis, then translate the scene by $\langle 5, -1, 2\rangle$, and then project the scene orthogonally onto the xy-plane. Compute the matrix that represents this sequence of transformations. (Make sure to get the order right; the first matrix to act on a vector should be the rotation.)