1 Overview

In this assignment you will learn about 3-D viewing and clipping. You'll implement both a 3-D viewing package and a 3-D clipping package and use these packages to display an animation of a rotating cube. You'll need to be familiar with 3D matrix transformations (Chapter 5 in FvDFH) and with orthographic and perspective 3D viewing (parts of Chapter 6 in FvDFH).

2 Implementation

Your program will be demonstrated with a simple animation: a cube rotating about its main diagonal. Allow the user to choose *orthographic* or *perspective* projections, and to select the number of degrees of rotation per frame. The skeleton program we provide shows how to use SRGP for achieving smooth animation (of a spinning triangle) by double buffering. You may start with our code, or build a more ergonomic interface using SUIT.

For the orthographic projection, you don't need to do any clipping and you can assume that the viewer is looking straight on. However, you may be happier with the animation — and it won't be hard — if you first rotate the cube -30 deg about the x axis and 30 deg about the y axis.

For the perspective projection, you need to let the user set a viewing position and clipping planes. However, you do not need to give the user all the degrees of freedom that are discussed in the book. In particular, you do not need to support oblique screen views, so no shearing is needed. Also, you can define the location of the cube yourself, so you can clip in 3D rather than in homogeneous coordinates. The 3D clipping is a straightforward extension of the 2D clipping algorithm discussed the first week of class. Be sure that a user of your program can select perspective views that adequately demonstrate the effects of clipping to all six sides of the viewing pyramid.

The cube should be represented on the screen by a line-drawing of its twelve edges. In addition, draw an additional edge along the cube's main diagonal, since this edge will remain invariant under the correct rotation. Try to come up with an efficient data structure for storing the cube's set of vertices and edges, so that a minimum number of operations will be required to rotate the cube. Of course, your implementation should concatenate the rotation matrices before any transformations take place, so that each vertex may be rotated with a single matrix multiply.

There is no test data for this program.

3 Suggestion

This assignment is not very long, but it can be tricky. Get started right away (but keep in mind that we won't be talking about the mathematics underlying perspective transformations until next Tuesday.) There are basically four parts to the assignment: the driver, the orthographic projection, the basic perspective projection, and the perspective projection with clipping. I suggest that you implement the four parts in this order.

4 What to Hand In

Turn in the usual.

5 Bells & Whistles

This is for those of you who wish to explore material beyond the normal assignment. No extra credit will be given. Be sure you have completed all the normal parts of the homework before attempting bells and whistles.

Implement a more general system that can perform arbitrary affine transformations on a more general class of line-drawn objects. Your program should read a script containing a sequence of commands of the form

```
Translate dx, dy, dz
Scale sx, sy, sz
Rotate {A,B,C}{X,Y,Z}, angle
  % rotation axis is the line from {A,B,C} to {X,Y,Z}
```

and use the script to create an animation. The program should also be capable of reading a description of an object (or set of objects) from a file. These bells & whistles should not be difficult to implement: you already have the tools at your disposal for performing all the transformations; you just need to write a more sophisticated driver.

If you are feeling really ambitious, you may want to try implementing a 4-D viewing and clipping package and display a rotating hypercube in addition to the ordinary cube.

On the other hand, if you are feeling unambitious (but want to do something with your copious free time), try rotating line drawings other than a cube. Block letters of your name, for instance.

Finally, if spinning block letter don't excite you, try eliminating the hidden lines. Another easy hack with some sex appeal is to eliminate the hidden faces by painting each face with a unique opaque color. The crux of the matter is to figure out the order in which to paint the faces.