CS 430 – Computer Graphics
Fall 2017
Assignment 7 – 3D Depth Cuing

In this assignment you will demonstrate your understanding of 3D graphics as it pertains to depth cuing a filled polygon mesh specified via an SMF file.

From the previous assignment you should be able to:

1. Read an SMF file containing specifications for a polygonal 3D mesh.
2. Store 3D mesh data in homogenous coordinates
3. Apply the graphics pipeline for drawing 3D specified polygons. In particular:
    a. Project points to a canonical view.
    b. Trivially test polygons against a canonical view volume.
    c. Project points to an orthogonal canonical view.
    d. Go from an orthogonal canonical view to 2D view plane/window
    e. Transform view plane to viewport
    f. Draw viewport into software framebuffer.

In addition, in this assignment you should be able to:

1. Extend your XPM implementation to add RGB color with 20 shades of pure red, pure green, and pure blue and to have a background color of black ("#000000")
2. Allow for specification of the near and far clipping planes in VRC coordinates.
3. Support for drawing of up to three models.
4. Scan-fill polygons projected from 3D to 2D
5. Interpolate depth information during scan filling.
6. Use depth information to provide depth cuing.
7. Use depth information to do z-buffering

Make sure you give yourself adequate time. The programming components in particular can be quite time consuming.

As a reminder you may use the programming language of your choice, though I recommend C/C++ and also make sure that your program can run on the Drexel tux cluster to insure its system independence.

Submission Guidelines

1. Assignments must be submitted via Bd Learn
2. Submit a single compressed file (zip, tar, etc..) containing:
    a. A PDF file with your solutions to the theory questions.
    b. A README text file (**not** Word or PDF) that explains
        i. Features of your program
        ii. Language and OS used
        iii. Compiler or interpreter used
        iv. Name of file containing main()
        v. How to compile/link your program
    c. Your source files and any necessary makefiles, scripts files, etc… to compile and run your program.

Theory Questions:

1. Given a triangle in 3D space specified by vertices $A = [1, 1, 1]^T, B = [5, 1, 1]^T,\ C = [2, 3, -2]^T$ compute the normal of this polygon. Assume that the vertices are given in counter-clockwise order for rendering. Show your work. Make sure to normalize your final vector. (4pts)

Assignment Details

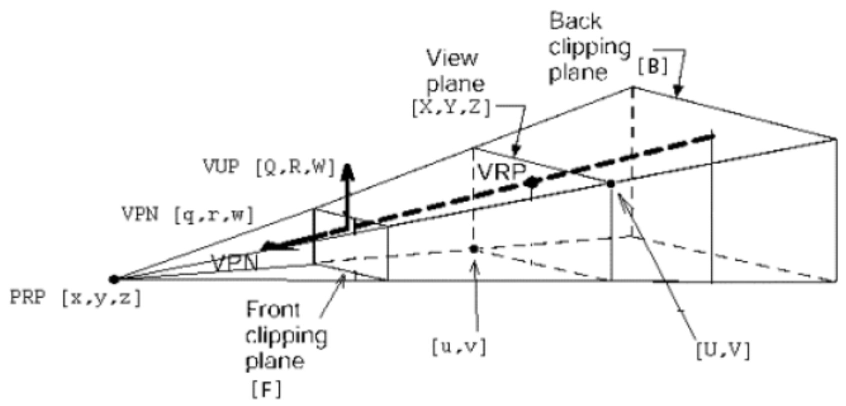
In addition to the command arguments from the prior assignment, allow for the following parameter options:

- [-F] followed by the floating point coordinate of the Front (near) plane in VRC coordinates (0.6)
- [-B] followed by the floating point coordinate of the Rear (far) plane in VRS coordinates (-0.6)
- [-f] followed by the name of the first SMF model. Its base color is red, "ff0000" (bound-sprellpsd.smf).
- [-g] followed by the name of the second SMF model. Its base color is green, "00ff00" (no file)
- [-i] followed by the name of the third SMF model. Its base color is blue, "0000ff" (no file).

Additional things to note:

- Your frame buffer size should be 500x500

The command line options correspond to the value in the image below.



The general program flow should be:

1. Read the SMF file(s) containing specifications for a polygonal 3D mesh(es).
2. Compute the necessary view and projection matrices based on command line options to arrive at points in canonical space.
3. Perform clipping in 3D. **Note for this assignment you only need to detect if a 3D polygon is completely within the view volume**.
4. Project the accepted 3D polygons (currently in a canonical view) to 2D default view plane, retaining depth/z information for z-buffering and depth cuing.
5. Map the vertices from the default view plane coordinates to the viewport.
6. Scan fill the mesh polygons using the depth information to do depth buffering and to provide depth cuing.

Grading Scheme:

In order to get any credit at all you must be able to generate at least read in a PS file and write out a valid XPM file

1. Theory Questions (4pts)
2. Can load and draw up to 3 models (20pts)
3. XPM file can draw three colors, one per model (20pts).
4. Scan-filling of polygons (20pts).
5. Depth Cuing (18pts)
6. Z-Buffering (18pts).

Common deductions:

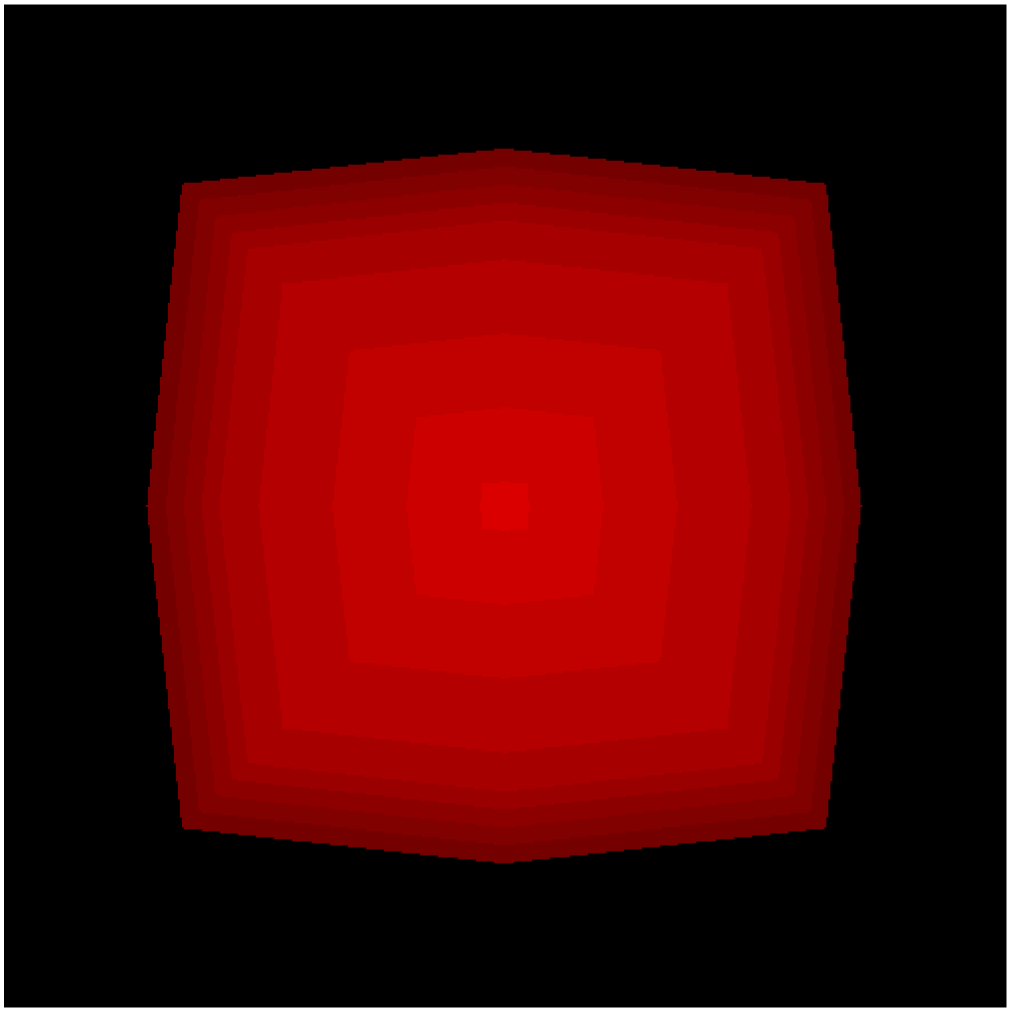
1. Nothing drawn (-100pts)
2. Implementation not done in homogenous coordinates (-40pts)
3. Missing files (including readme) (-5pts each)
4. Cannot compile/run out-of-the-box on TUX (-10pts)

Provided Tests

As usual we will provide several tests for you and may perform addition ones when grading you. Here are some tests to help you gauge the correctness of your implementation:

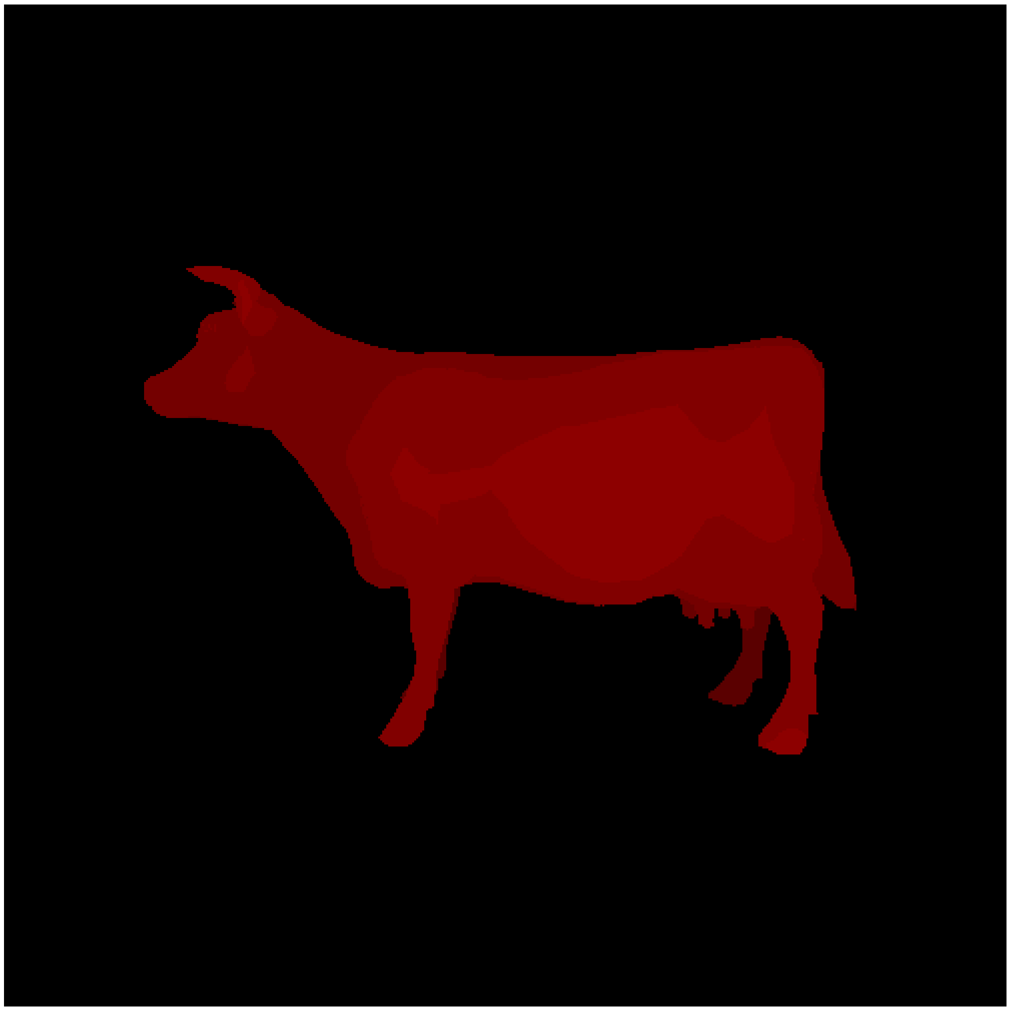
Default settings:

```
./A7 –f bound-sprellpsd.smf> out1.xpm
```



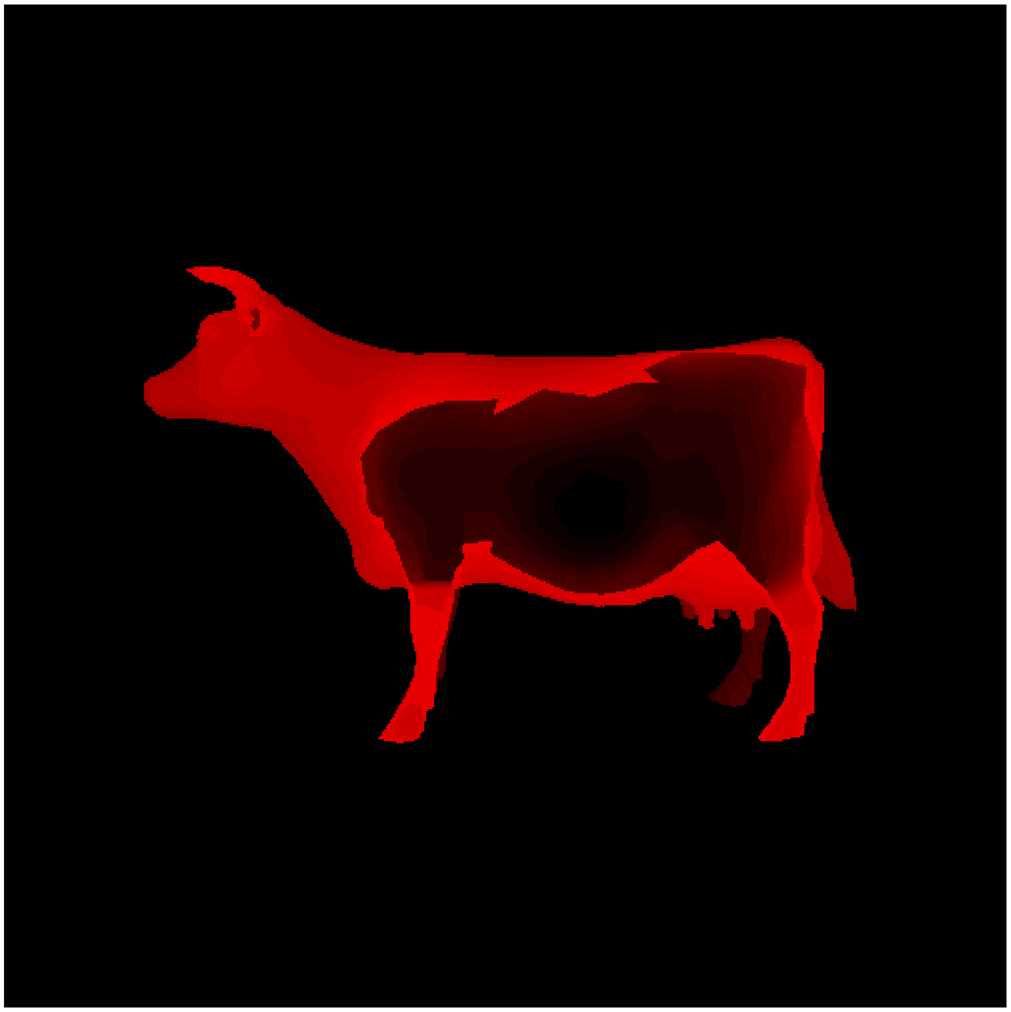
Draw the cow

```
./A7 -f bound-cow.smf  >out2.xpm
```



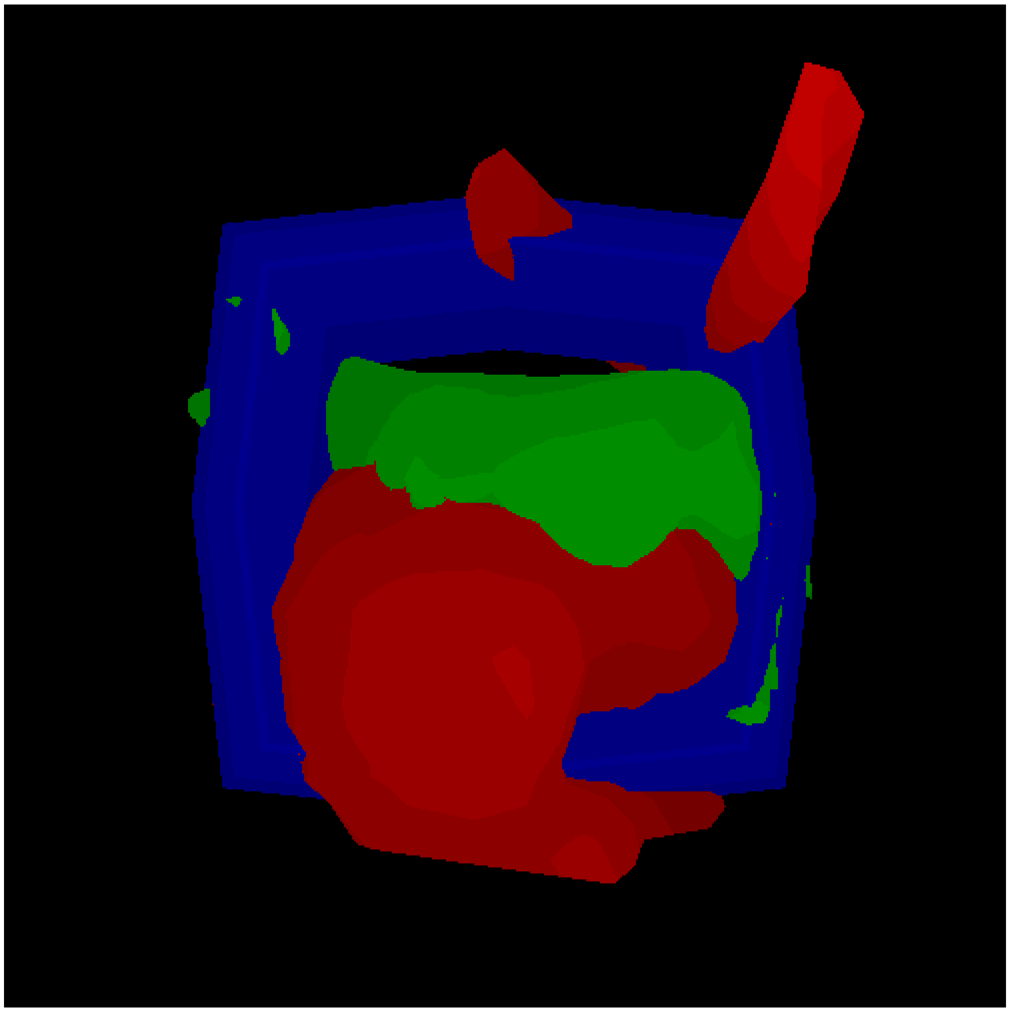
Now change where the front and back clipping planes are. The front plane is actually clipping the side of the cow! So the dark stuff you see is the other side of the cow.

```
./A7 -f bound-cow.smf -F 0.1 -B -0.18 >out3.xpm
```



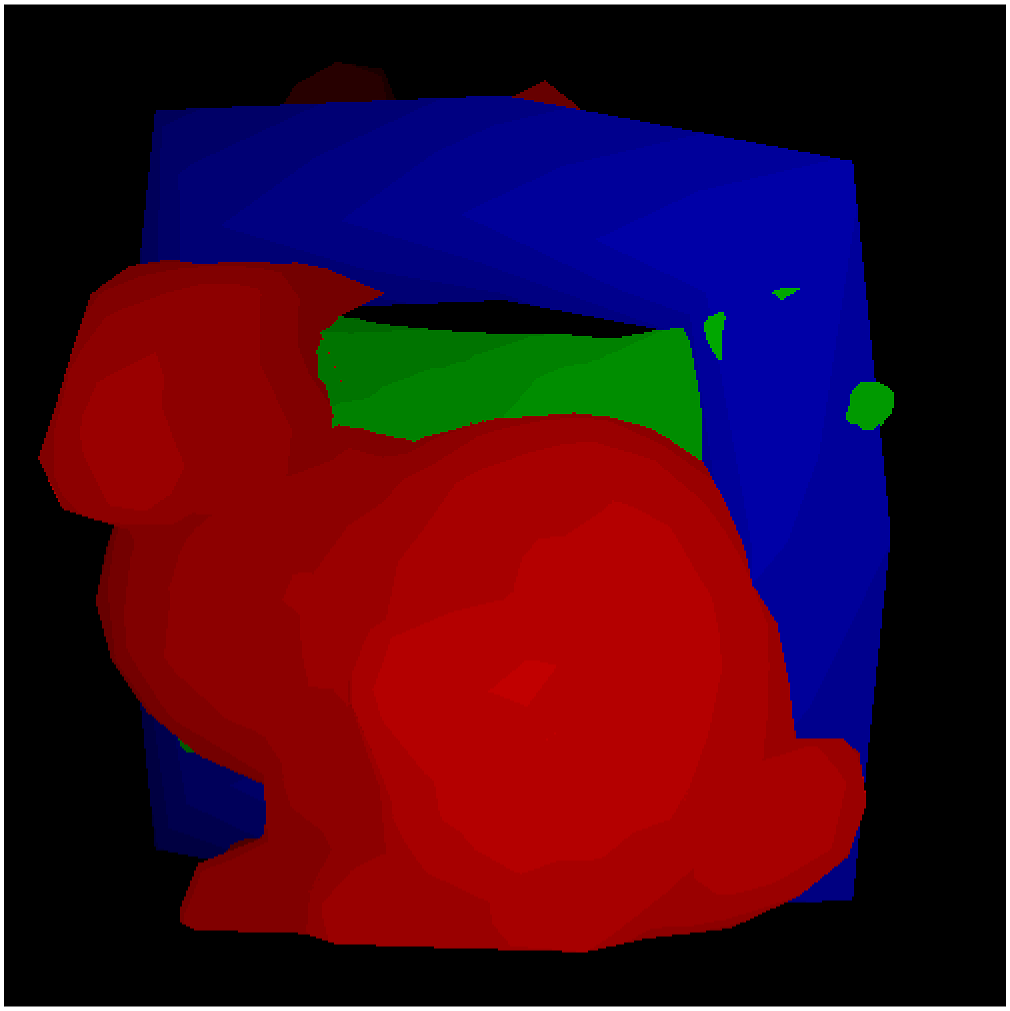
Draw all three models. Note the z-buffering.

```
./A7 -f bound-bunny_1k.smf -g bound-cow.smf -i bound-sprtrd.smf >out4.xpm
```



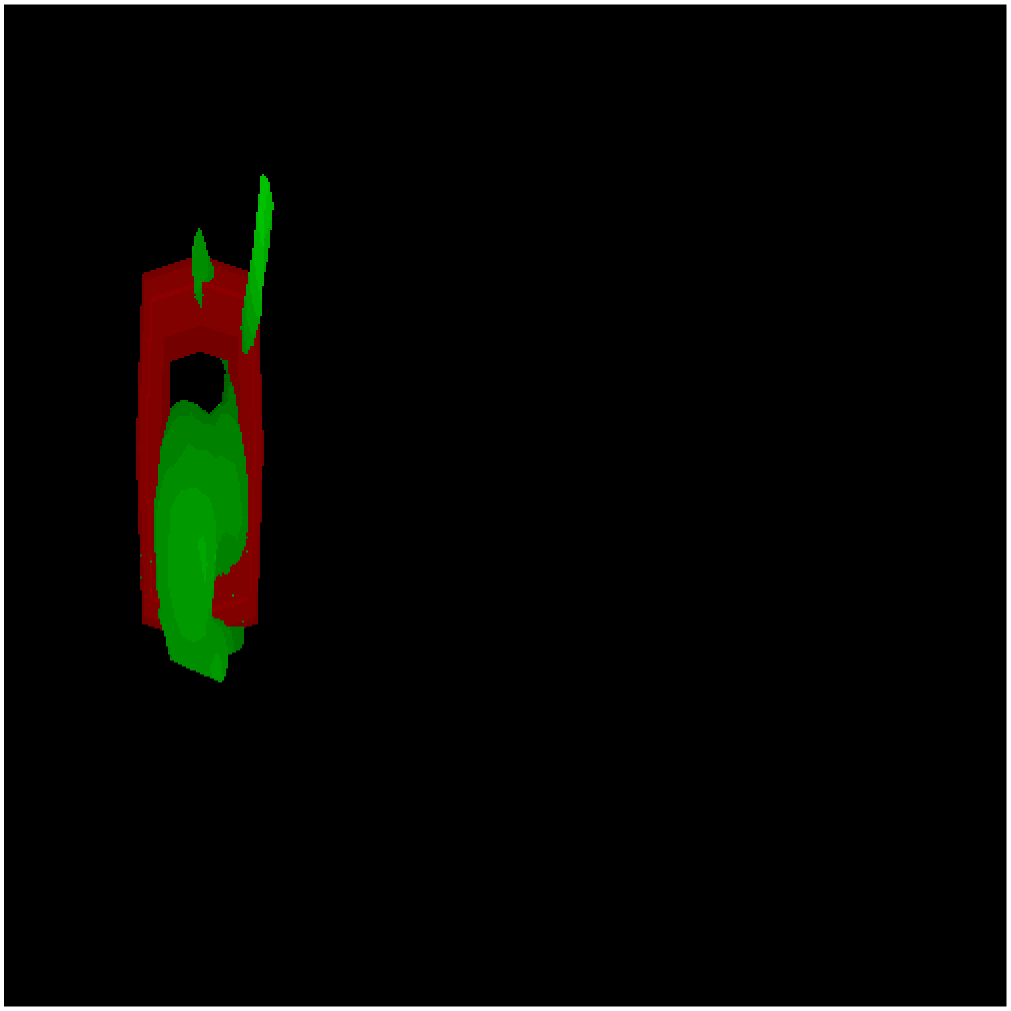
Now let's play with some other parameters! Draw using orthographic projection and change the view plane normal to be (0.4, 0.2, 1) instead of the default (0,0,-1).

```
./A7 -f bound-bunny_1k.smf -g bound-cow.smf -i bound-sprtrd.smf -q 0.4 -r 0.2 -w 1.0 -P >out5.xpm
```



Drawing just the bunny and the sprtrd to a viewport bound by (34, 104) to (149, 472)

./A7 -u -1.0 -U 0.8 -v -0.9 -V 1.0 -g bound-bunny_1k.smf -f bound-sprtrd.smf -j 34 -k 104 -o 149 -p 472 >out6.xpm



With that same viewport how about we change the View Up Vector to (-1.1, 0.9, 0.6).

```
./A7 -u -1.0 -U 0.8 -v -0.9 -V 1.0 -g bound-bunny_1k.smf -f bound-sprtrd.smf -j 34 -k 104 -o 149 -p 472 -Q
-1.1 -R 0.9 -W 0.6 >out7.xpm
```