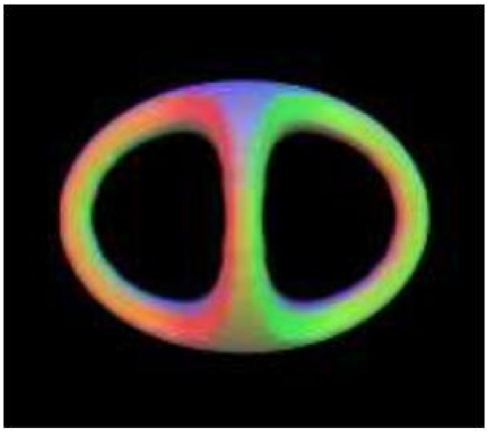
1.Camera and Ray geometry

By adding another case to the sdg file to generate a double ring topology and change the look-at point in the middle of the rendered image, we adjusted the orientation to produce the front view by setting the parameters as:

-fr 10 10 0 -at 0 0 0 -up 0 0 1

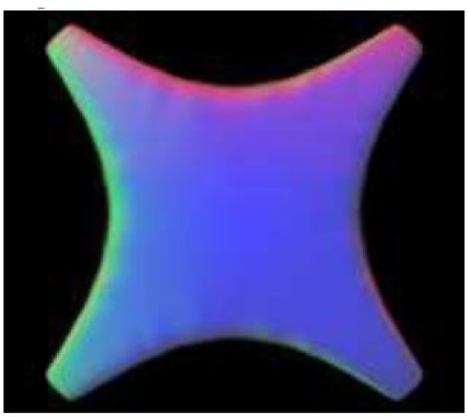
and the front view image is rendered as below:



and to produce the top view by setting the parameters as:

-fr 0 0 10 -at 0 0 0 -up 1 0 0

and the top view image is rendered as below:



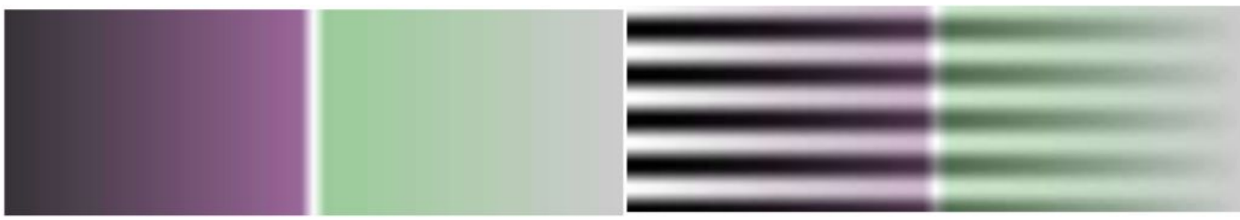
As we can see, the output images can be expected from the double ring topology specified in the sdg file as below:

```
case 11:
    ret = 0;
    ret += gauss1D(2, xx, 0) * gauss2D(0.5, yy, zz, 1);
    ret += gauss1D(2, yy, 0) * gauss2D(0.5, xx, zz, 1);
    break;
```

which is generated by the tensor product between a 1D Gaussian and a 2D Gaussian overlapping in the north and south poles.

2. Transfer functions

We generated a transfer function to show that our lookup table is working. The color changes from a dark gray to purplish to white to light green and finally back to dark gray when the position goes from 0 to 0.48 to 0.5 to 0.52 to 1. Meanwhile we also changed the opacity from 0 to 0.25 to 0.50 to 0.75 to 1 as the position goes from 0 to 0.25 to 0.50 to 0.75 to 1, and finally generated Colormap using the RGBA LUT using the thresh-map.txt and cube-atxf2.txt:



using the following two commands:

./rendr lgen -i $SCIVIS/cmap/thresh-cmap.txt $SCIVIS/cmap/cube-atxf2.txt -l 300 -mm 0 1 -o color1.nrrd -oi color1.png

./rendr lgen -i $SCIVIS/cmap/thresh-cmap.txt x -l 300 -mm 0 1 -o color2.nrrd -oi color2.png

Then we want to test the Levoy transfer function also works. So we set the opacity from 0 to 0.5 to 1.0 with isosurface functions at 0.25 to 0.5 to 0.75 respectively:

using the following commands:

```
./rendr go $CUBE -fov 14 -us 0.03 -s 0.03 -k ctmr -p rgba -b over \
-lut lut.nrrd -lit $SCIVIS/lit/rgb.txt -lev $SCIVIS/cmap/cube-levoy1.txt -o levoy.nrrd
overrgb -i levoy.nrrd -b 0 0 0 -o levoy.png
```

Where the cube-levoy1.txt is:

```
0.25 0.035 1.0
0.50 0.035 1.0
0.75 0.035 1.0
```

As we can see from the above picture, when opacity goes from 0 to 0.5 to 1, the corresponding image goes from showing nothing(total black) to showing translucent surface to showing a more opaque surface, which is aligned with our expectation.

3. Probing and Blending

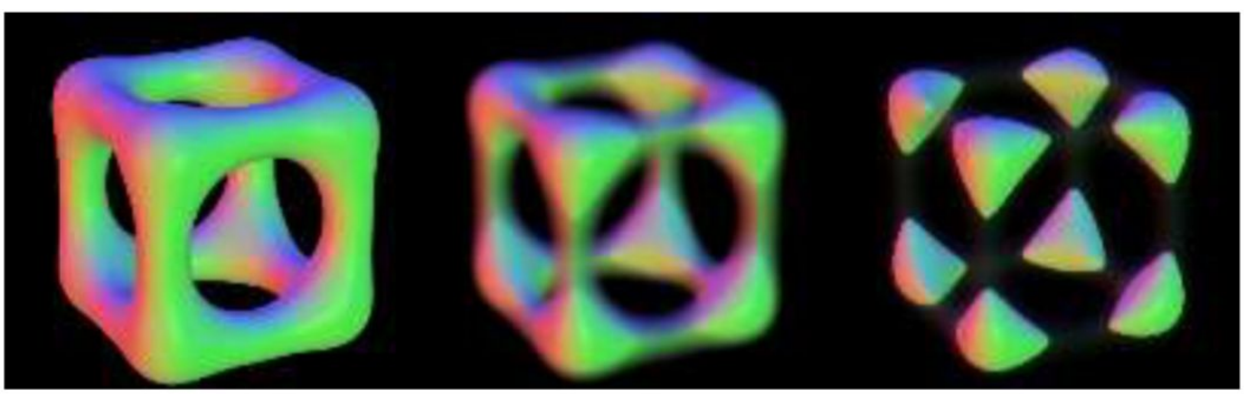
We test that all probes works correctly by checking the blending for sum, min and max. We first set the ncv = -0.3 and fcv = 0.3, and then run the three test:

using the following command lines:

for B in max sum mean; do
  ./rrendr go $CUBE -s 0.2 -k box -p in -b $B -o cube-in-$B.nrrd
done
./rrendr iq -i cube-in-max.nrrd -mm 0 1 -o cube-in-max.png ./rrendr iq -i cube-in-mean.nrrd -mm 0 1 -o cube-in-mean.png ./rrendr iq -i cube-in-sum.nrrd -mm 0 24 -o cube-in-sum.png

The three images above shows the blending for sum, mean and max respectively. The two left images shows their relative magnitude at the each pixel are the same because mean differs from sum by further being divided by the total number of samples, which is aligned with our expectation and the right most image shows the blending for max. All pixels take the maximun value of 1 and thus show the white color, which is also aligned with our expectation.

Next we also test the blendover function using three different unit sizes of 0.03, 0.3 and 3 respectively to produce the colored cubes:



by setting -us 0.03, -us 0.3 and -us 3 respectively when running the rendr go commands.

As we can see from the image above, the unit steps increase leads to the decrease in opacity, and the right most image also shows that when the unit step size is big enough, blendings would work on some rays as the big step size jumps over all pixel points along these rays, which is aligned with our expectation.

4. Lighting with Blinn-Phong and Depth-cueing

By changing the ambient term, specular term and the diffuse term, we can observe the differences among the three rendered images as shown below:

using the following commands:
./rendr bp -c 1.0 0.2 0.2 -fr 1 1 1 -ph 1 0 0 90 -lit $SCIVIS/lit/rgb.txt -o - | overrgb -i - -b 0 0 0 -o bp_only_ka.png
./rendr bp -c 1.0 0.2 0.2 -fr 1 1 1 -ph 0 1 0 90 -lit $SCIVIS/lit/rgb.txt -o - | overrgb -i - -b 0 0 0 -o bp_only_kd.png
./rendr bp -c 1.0 0.2 0.2 -fr 1 1 1 -ph 0 0 1 90 -lit $SCIVIS/lit/rgb.txt -o - | overrgb -i - -b 0 0 0 -o bp_only_ks_90.png

As we can see the left most image comes with only ambient terms shows that a uniform lighting from all direction and thus produce the same color all over within the circle. The middle image comes with only the diffuse term shows a matte surface without shininess, and only the left red part can be seen due to the position of the light source. The right most image comes with only the specular term shows light reflection only can be observed from certain directions as the three separated color blobs show in the image, which is aligned with our expectation.

Next we test the depth-cueing by changing the range of the dcn and dcf:

using the following commands:

./rendr go $CUBE -fov 14 -us 0.03 -s 0.03 -k ctmr -p rgbalit -b over \ -lut lut.nrrd -lit $SCIVIS/lit/1.txt -dcn 1 1 1 -dcf 1 1 1 -o cube-dc.nrrd
overrgb -i cube-dc.nrrd -b 0 0 0 -o no_cube-dc.png

./rendr go $CUBE -fov 14 -us 0.03 -s 0.03 -k ctmr -p rgbalit -b over \ -lut lut.nrrd -lit $SCIVIS/lit/1.txt -dcn 0 0 0 -dcf 1 1 1 -o cube-dc.nrrd
overrgb -i cube-dc.nrrd -b 0 0 0 -o low_cube-dc.png

./rendr go $CUBE -fov 14 -us 0.03 -s 0.03 -k ctmr -p rgbalit -b over \
-lut lut.nrrd -lit $SCIVIS/lit/1.txt -dcn -1 -1 -1 -dcf 2 2 2 -o cube-dc.nrrd
overrgb -i cube-dc.nrrd -b 0 0 0 -o low_cube-dc.png

to produce images from with no depth cueing(left) to certain level of depth-cueing(middle) with dcn = (0,0,0) and dcf = (1,1,1) to high level of depth-cueing(right) with dcn = (-1,-1,-1) and dcf = (2,2,2). We can see that the increase in disparity between dcn and dcf leads to a increase in contrasts between the near end part of the image and the far end part of the image, which is aligned with our expectation.