ECE 5470 Computer Vision Lab6 report

Three-Dimensional Image Processing

Weiran Wang

NetID: ww463

# Section 2:

As instructed in the lab handout, I get started by running t1make script. The vgen3d command is used for generating 3D shapes and objects. The vop command combines with an or operation and we use the v3p command to convert the image to 3D polygons. Fig1 shows the outcome of running the t1make script. Then we can change the configuration of the 3d image by pressing c to bring up the option menu. Fig2 is the outcome by changing the render option to light mode.

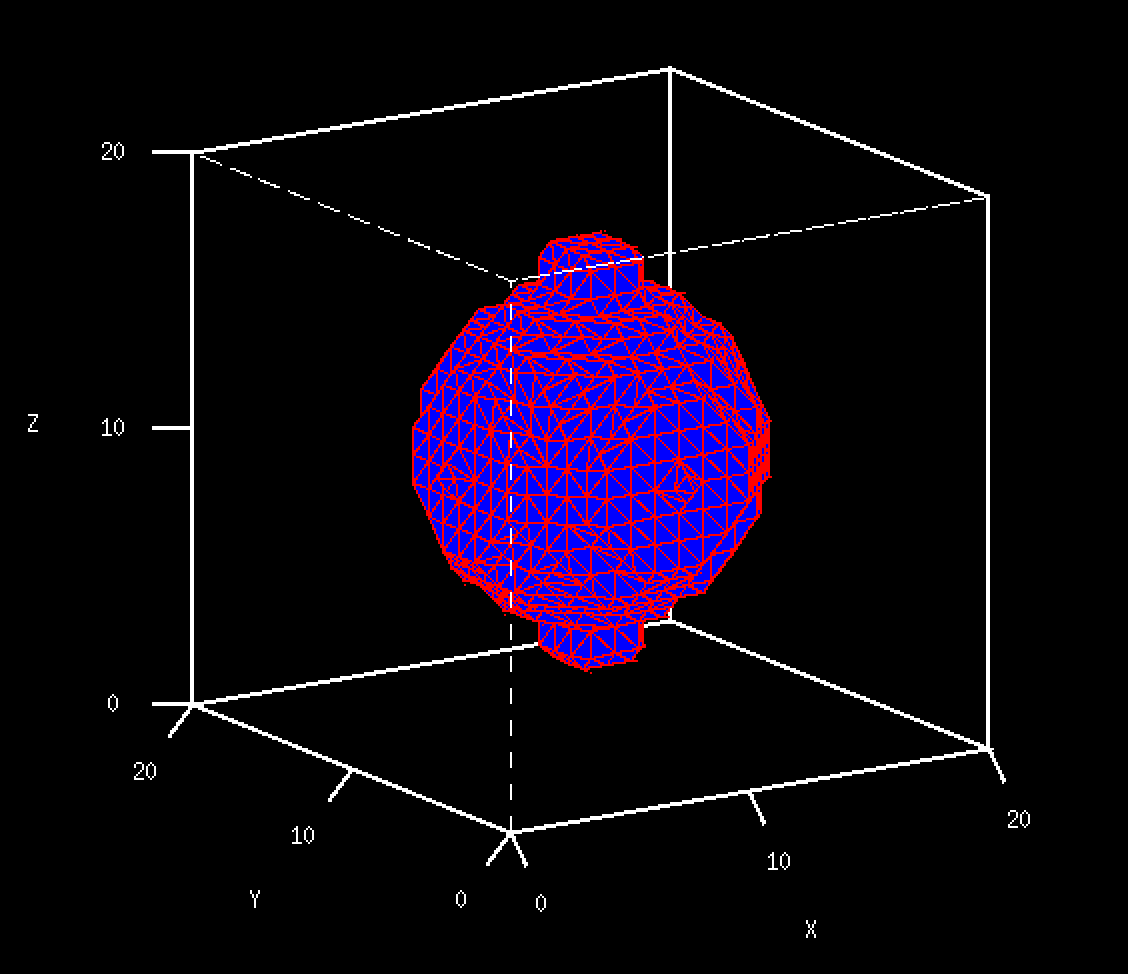
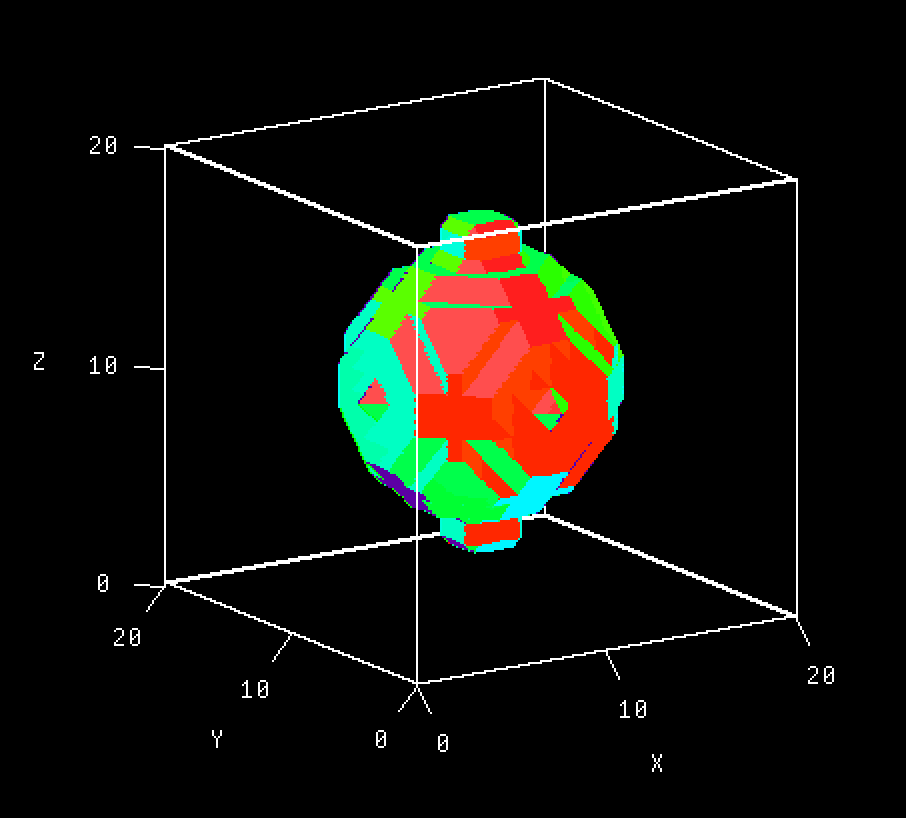
 

Fig1 t1.vd 3D object polygon Fig2 t1.vd in light mode

The appearance of the polygon we generated is coarse, we can smooth the polygon by using v3tfilt command. Fig3 shows what it looks like after smoothing. By changing the color configuration to grey, we can better see the smoothing effect.

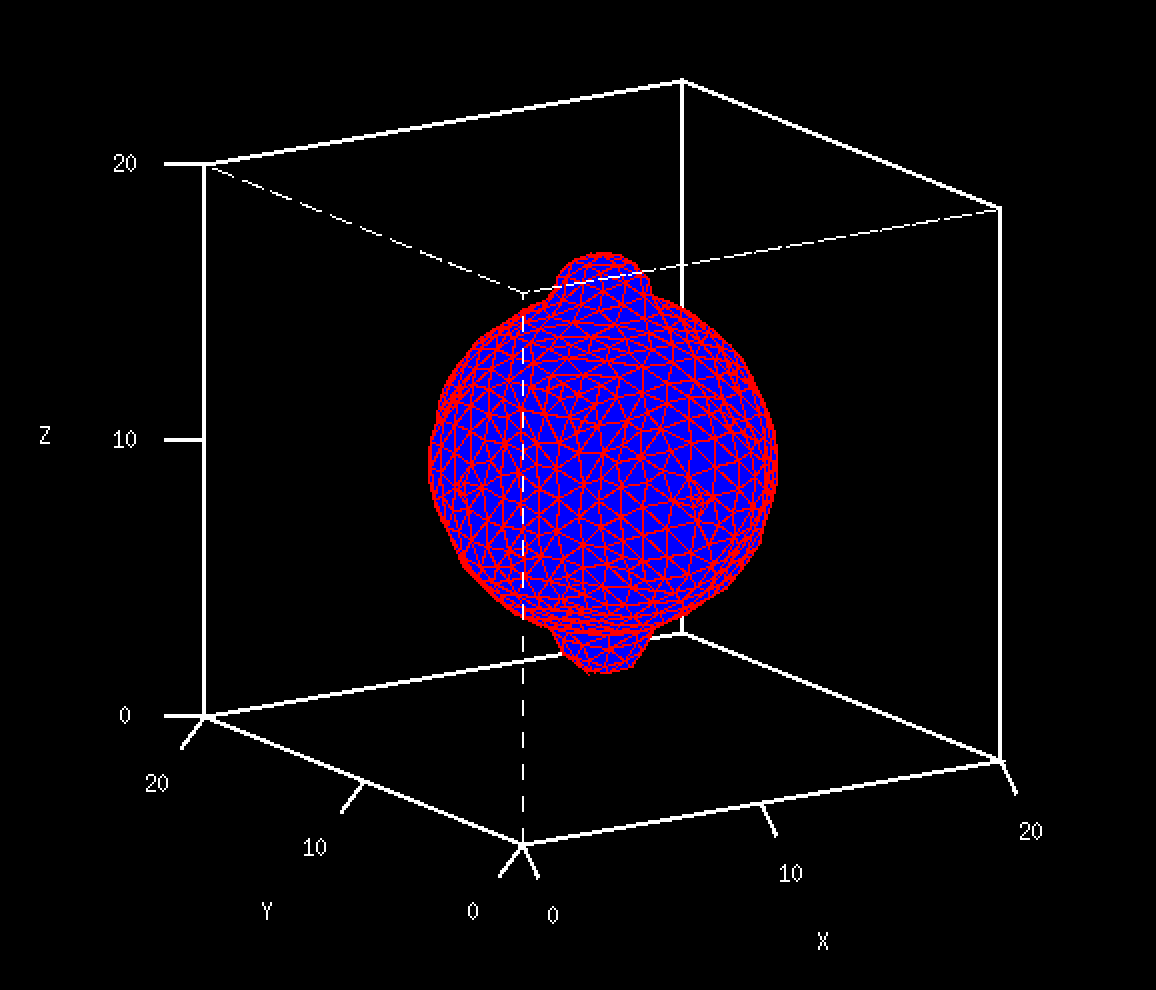


Fig3 t1f.vd 3D object polygon

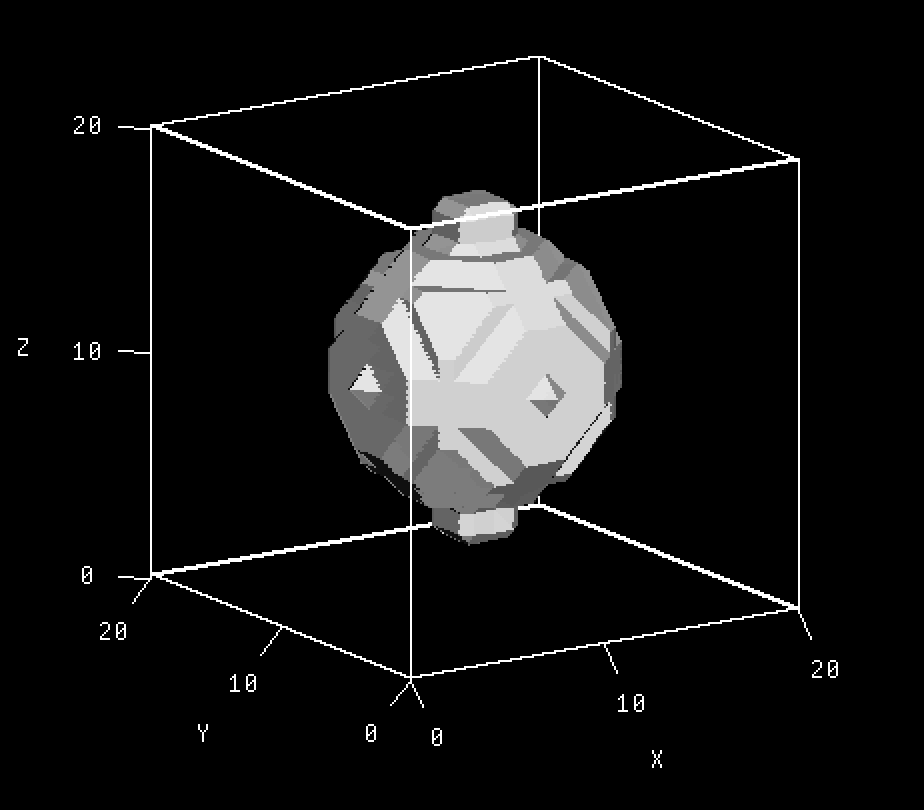
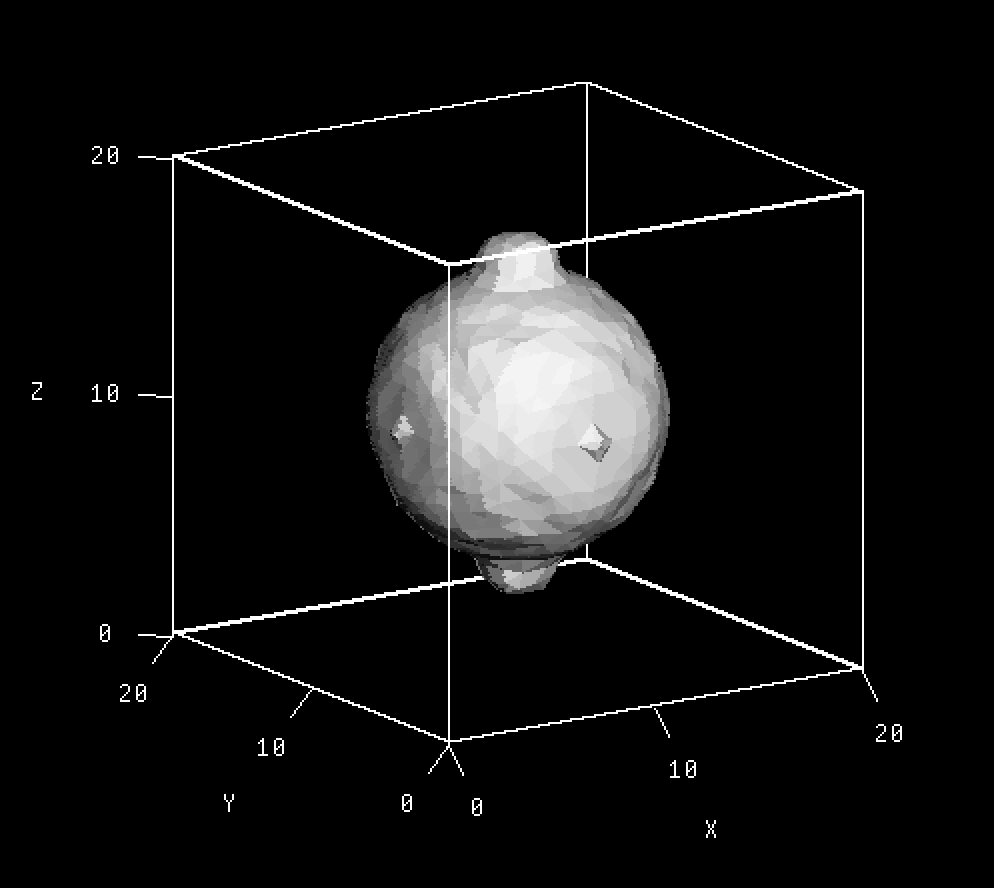
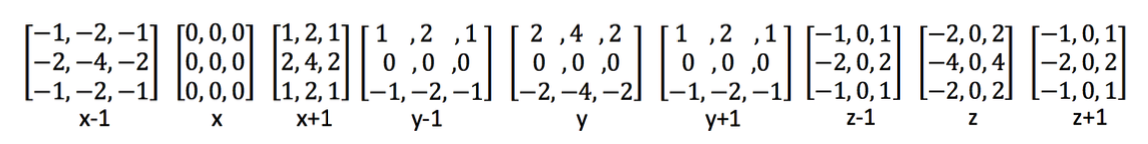
 

Fig4 t1.vd 3D object polygon in grey Fig5 t1.fvd 3D object polygon in grey

After the smoothing, the object loses sharp edges and looks more rounded. The bumps on the sphere look smaller than they were. We only want to perform this kind of action when we want to get rid of the sharp appearance of the object. If the intension is to preserve the details of the object, then smoothing should not be performed.

# Section 3:

In this section, we will be programming a 3d image filtering and edge detection algorithm. I get started by running t2make script to add some gaussian noise to the image t1n. The effect of adding Gaussian noise can be seen in Fig8. To handle the gaussian noise, I first apply a median filter before doing edge detection. For each pixel in the 3d image, the median filter take the median value of the a 3\*3\*3 cube. The edge detection was achieved by applying a kernel to to cube, the kernel looks like the following.



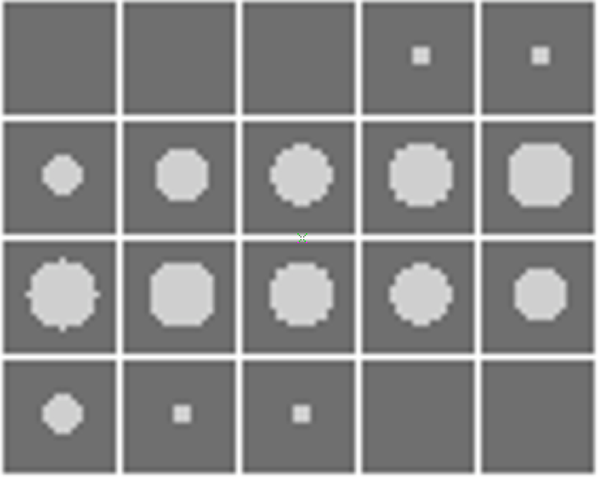
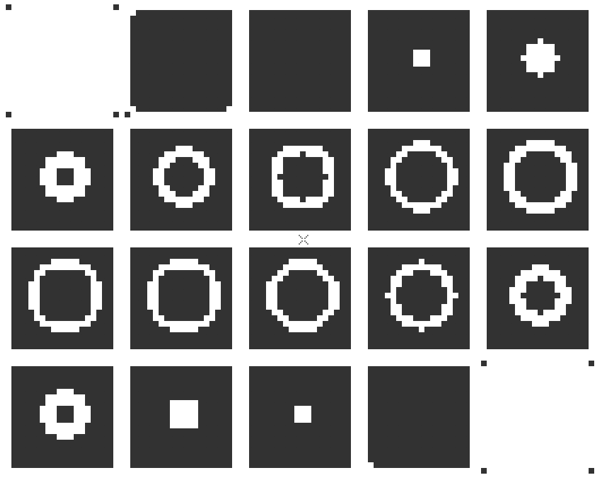
 

Fig6 t1g.vd Fig7 edge detection output of t1g

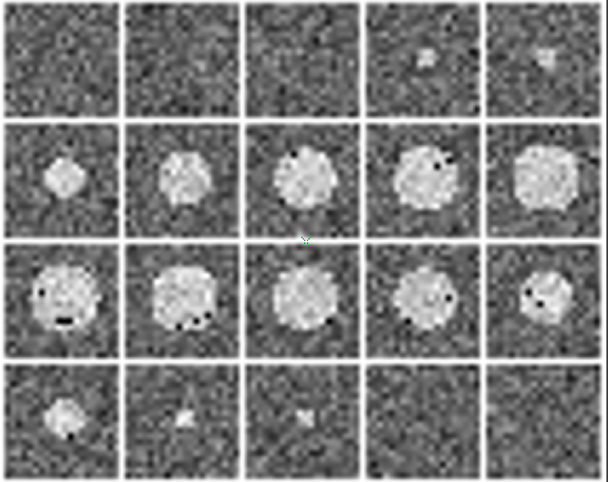
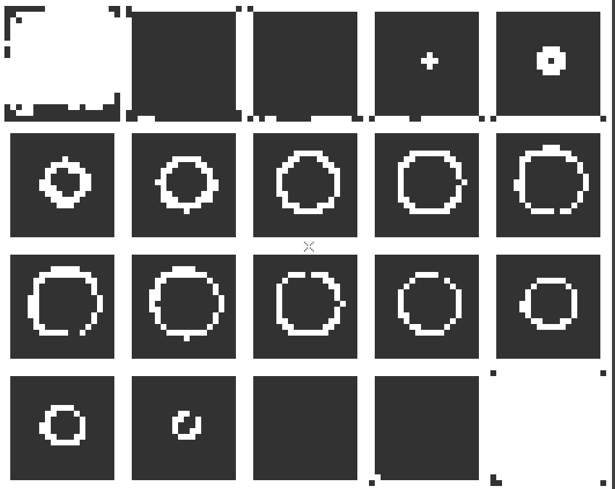
 

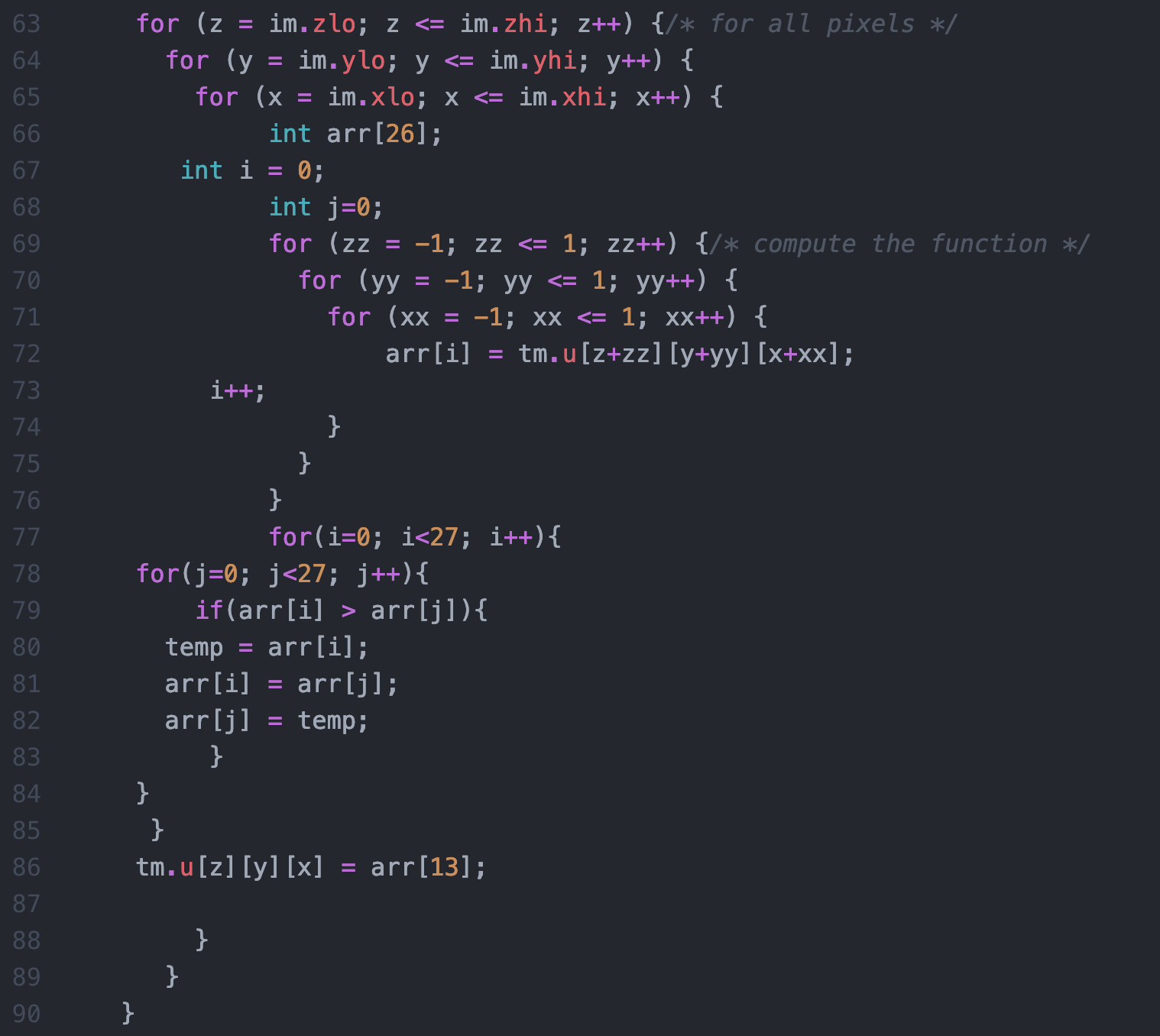
Fig8 t1n.vd Fig9 edge detection output of t1n

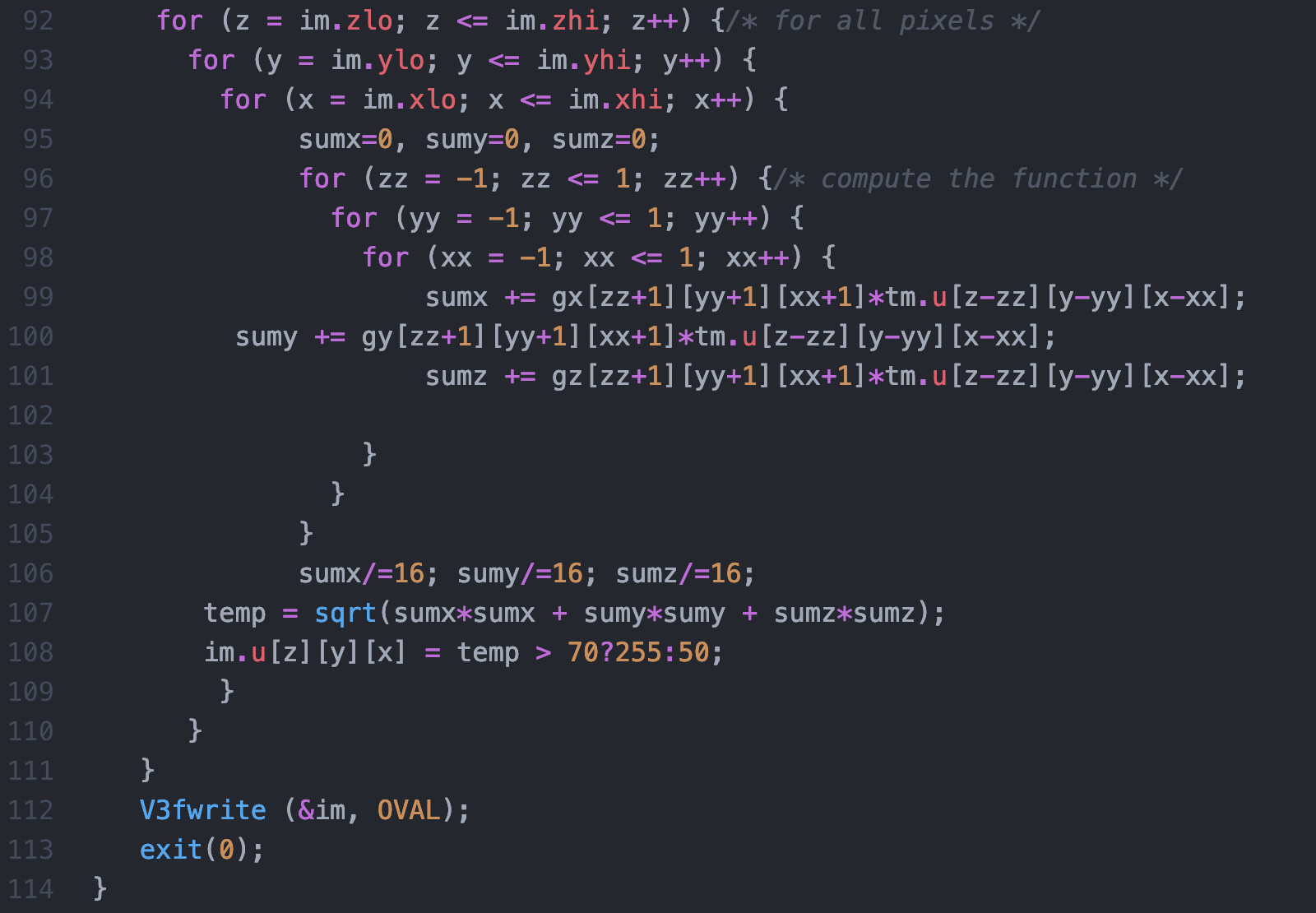
As we can see from the results above, the edge detection result of t1g.vd is pretty good. The edges of the object are nice and clear. When it comes to the image t1n.vd, which has gaussian noise, the edge detection operator still works pretty good. The median filter is very important in this case, it’s critical that we performed median filtering before the edge detection.

Program:

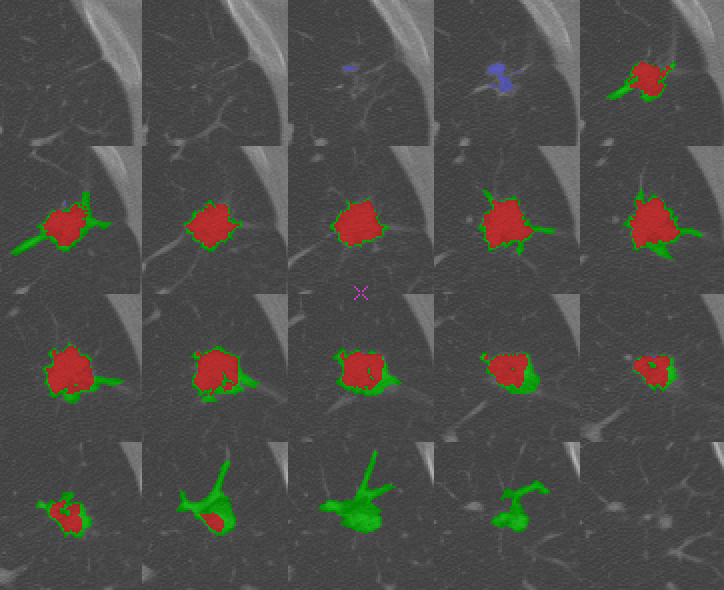




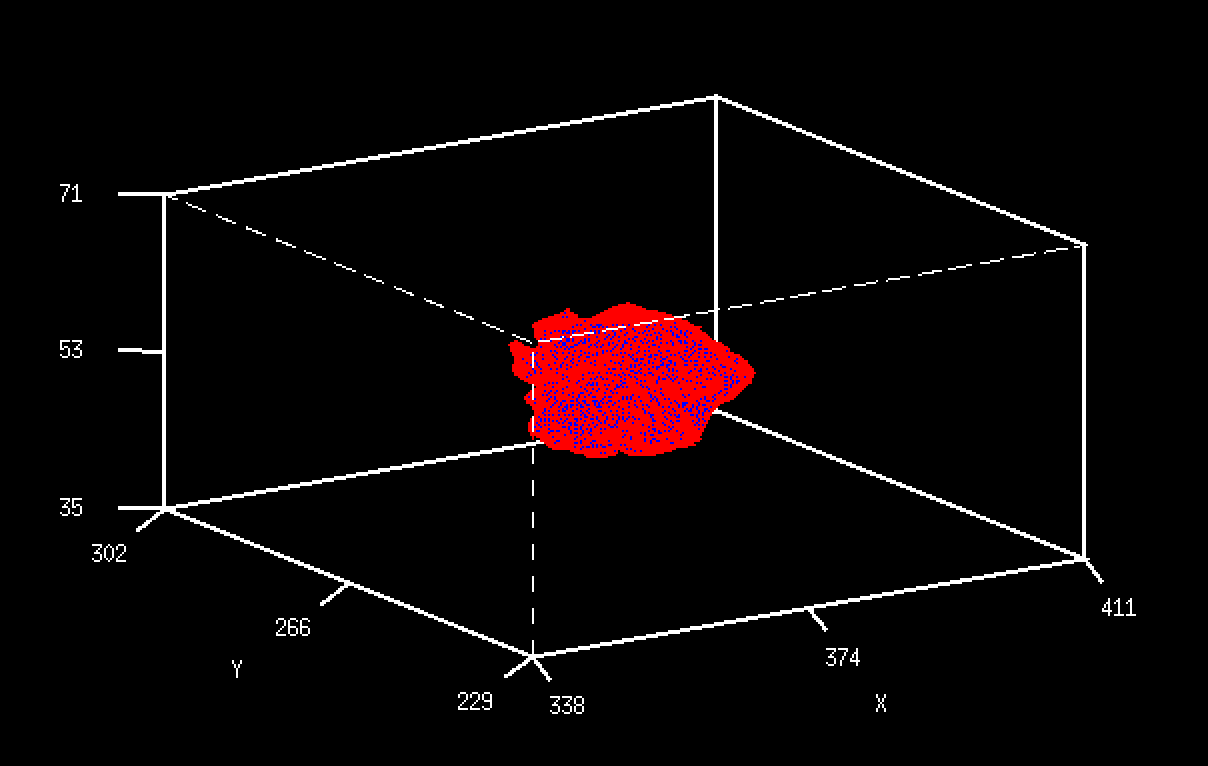




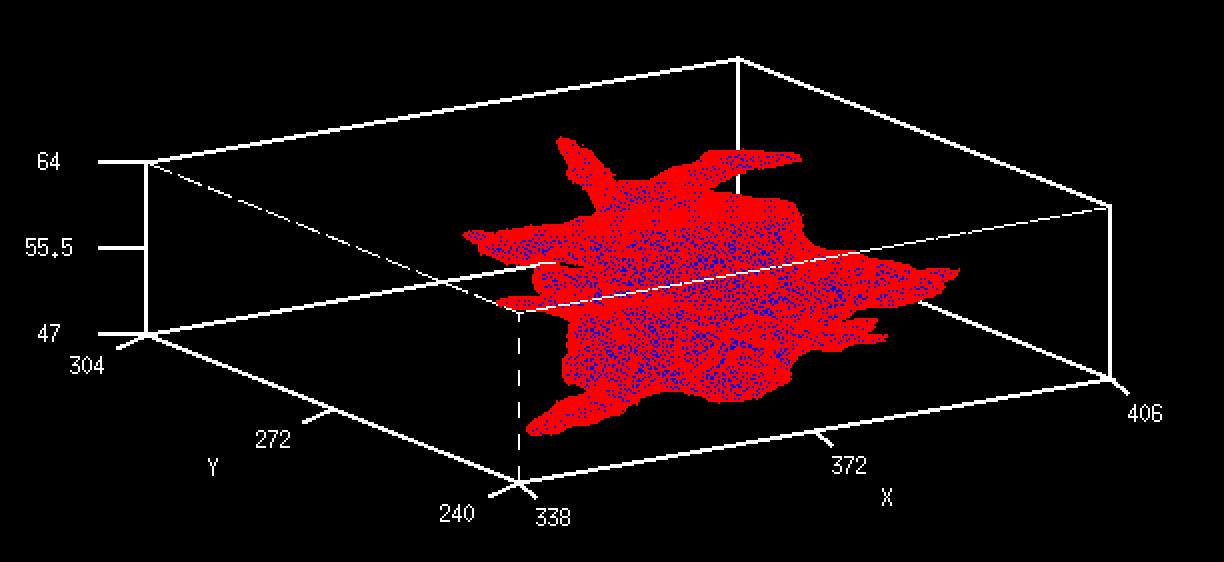
# Section 4:



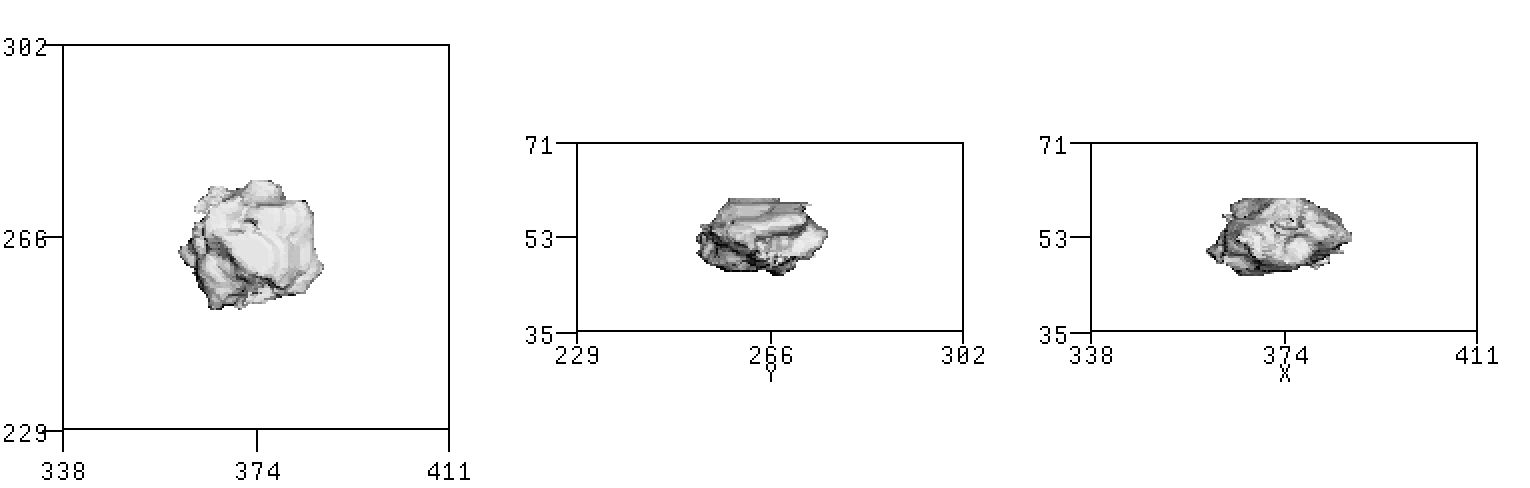
Decompare.vx



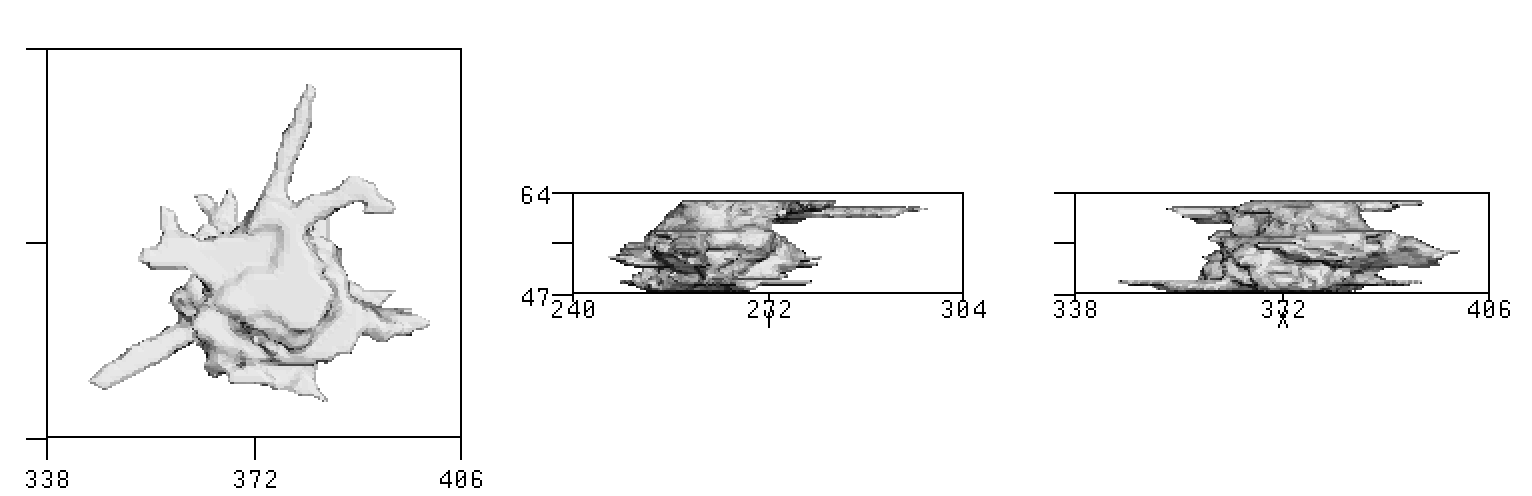
Nodseg.vd



Mregion.vd



dnodseg



dmregion

