CS 553 Scientific Visualization

Project #6:

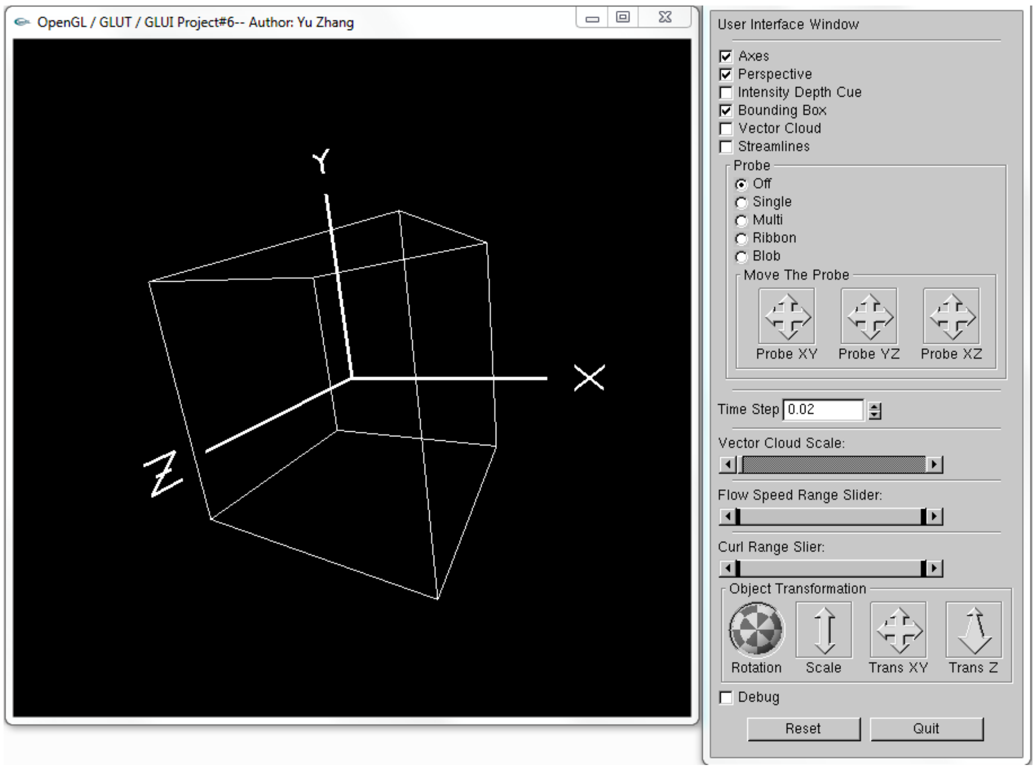
Vector Field Visualization

Yu Zhang
Zhangy6@onid.oregonstate.edu
Master Student in Computer Science
School of Electrical Engineering and Computer Science
Oregon State University

05/08/2015

1. Images and relative comments of my project

a) Interface of my project


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: [✓ Off] Single Multi Ribbon Blob

Time Step 0.02

Vector Cloud Scale:

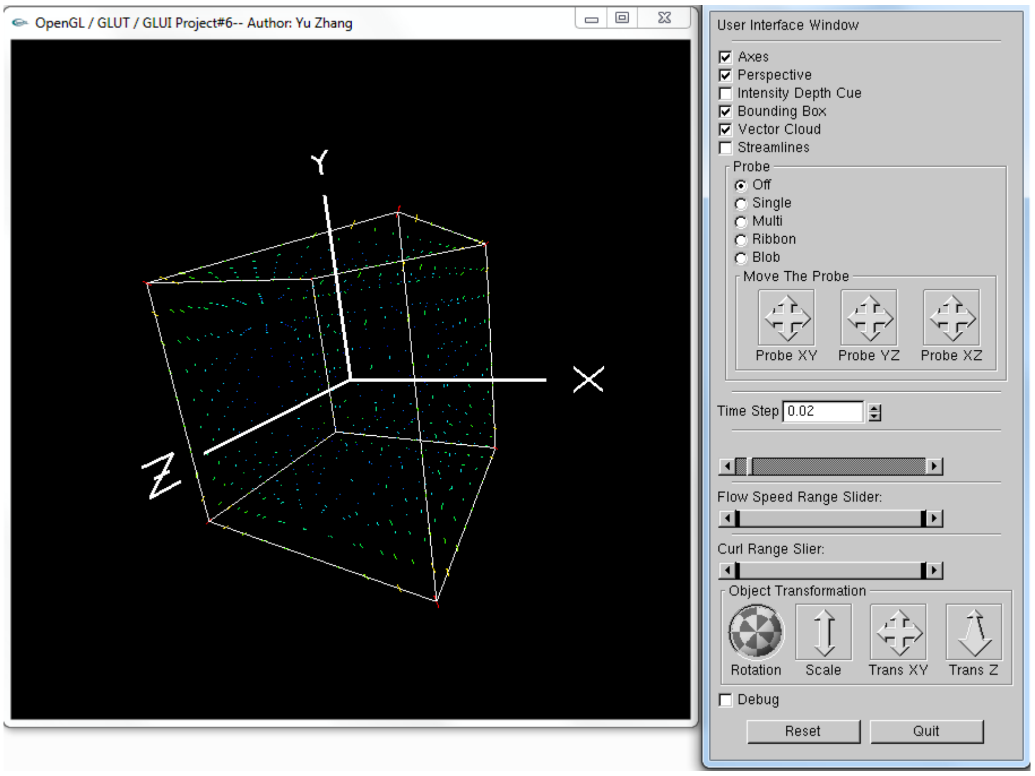
Flow Speed Range Slider:

Curl Range Slier:

- [ ] Debug

In this project, the codes are changed a lot, we need to modify the User Interface Window and the model we need to use in this project --- a cube.
For drawing the cube, I use GL_LINE_STRIP in the function InitLists() and then glCallList() in display to show the cube.

b) Vector Cloud


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [✓] Vector Cloud
- [ ] Streamlines

Probe: [✓ Off] Single Multi Ribbon Blob

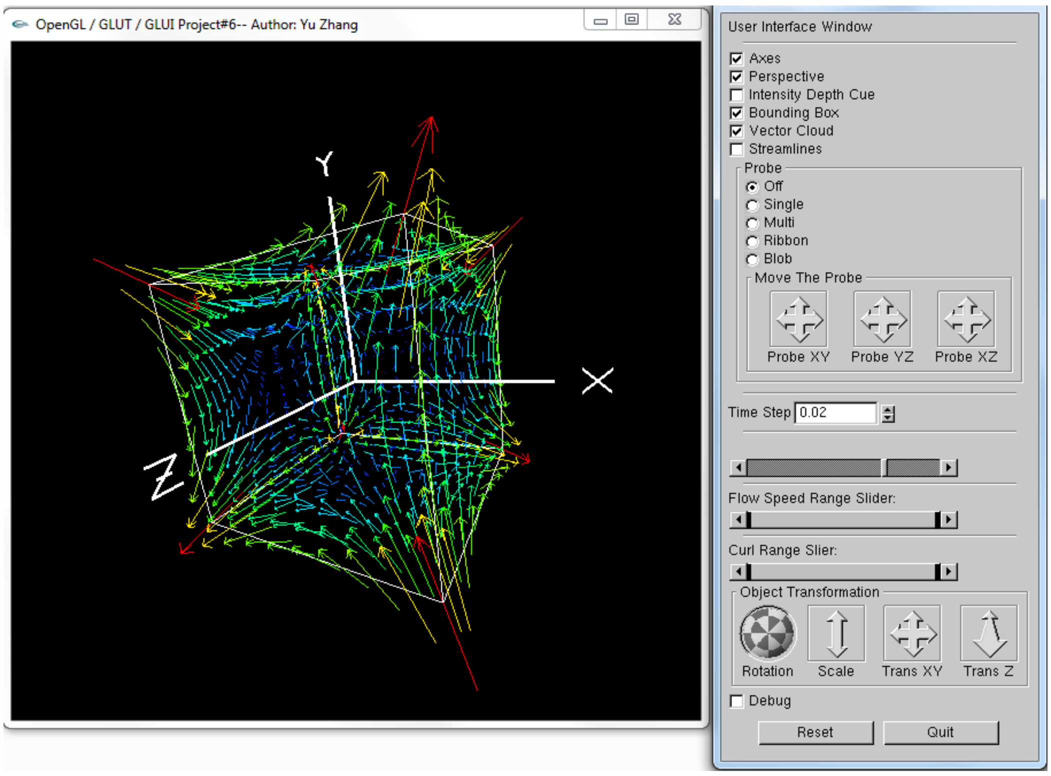
Time Step 0.02

Flow Speed Range Slider:

Curl Range Slier:

- [ ] Debug

In this project, Vector Cloud is calculated by each point's position, and it represents the velocity of each position. As we know the velocity contains magnitude and direction, so the function Arrow() can help us to draw arrows by using different color and different color. The following graph shows longer arrows and you can clearly see each point's velocity.



- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [✓] Vector Cloud
- [ ] Streamlines

Probe: [✓ Off] Single Multi Ribbon Blob

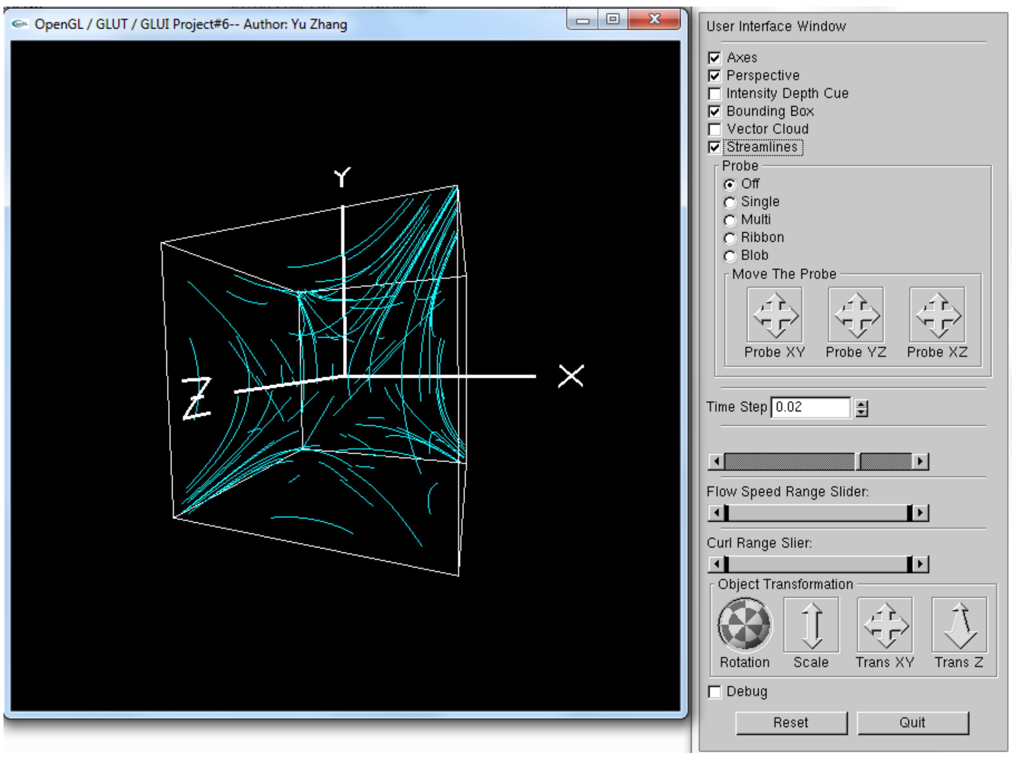
Time Step 0.02

Flow Speed Range Slider:

Curl Range Slier:

- [ ] Debug

c) Streamlines



- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [✓] Streamlines

Probe: [✓ Off] Single Multi Ribbon Blob

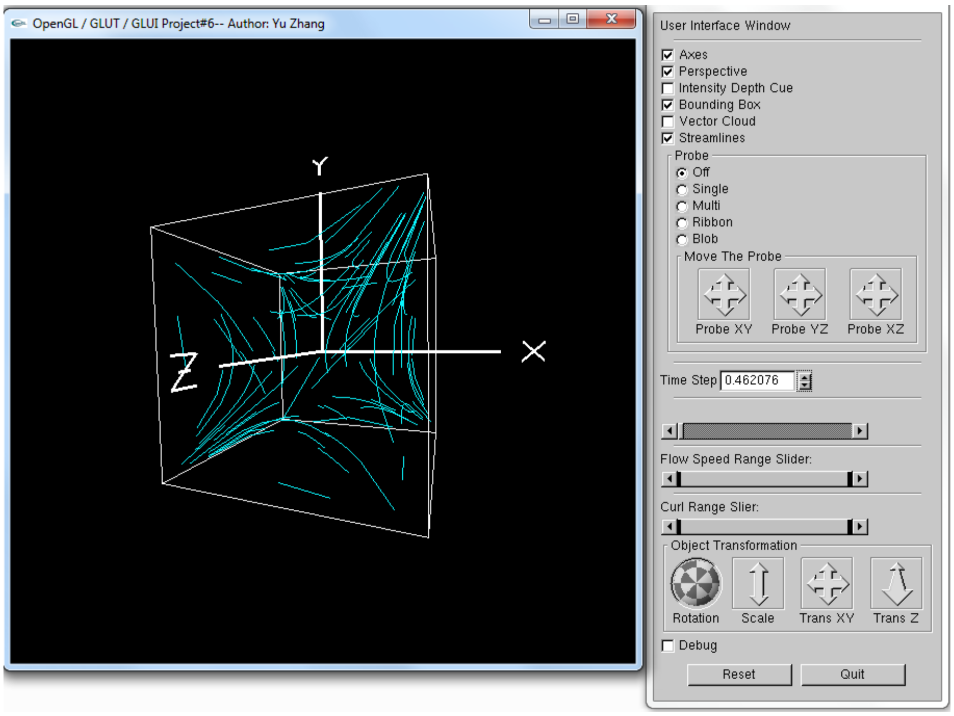
Time Step 0.02

Flow Speed Range Slider:

Curl Range Slier:

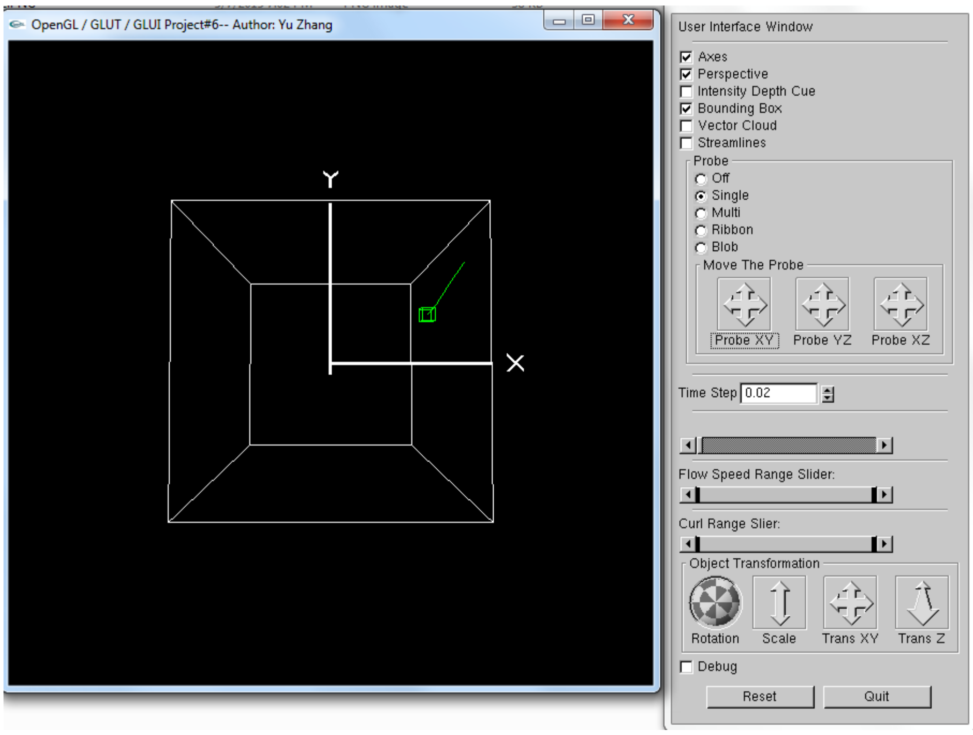
- [ ] Debug

In this part, streamline means we need to advect one point and get it's velocity change track at successive *time steps*, and then draw a line between each position. This part is introduced in slides so the implement is not that hard. The following graph shows streamlines by different time steps.



[✓] Axes
[✓] Perspective
[ ] Intensity Depth Cue
[✓] Bounding Box
[ ] Vector Cloud
[✓] Streamlines
Probe: [✓ Off] Single Multi Ribbon Blob
Time Step 0.462076
[ ] Debug

d) Probe with single streamline



[✓] Axes
[✓] Perspective
[ ] Intensity Depth Cue
[✓] Bounding Box
[ ] Vector Cloud
[ ] Streamlines
Probe: Off [✓ Single] Multi Ribbon Blob
Time Step 0.02
[ ] Debug

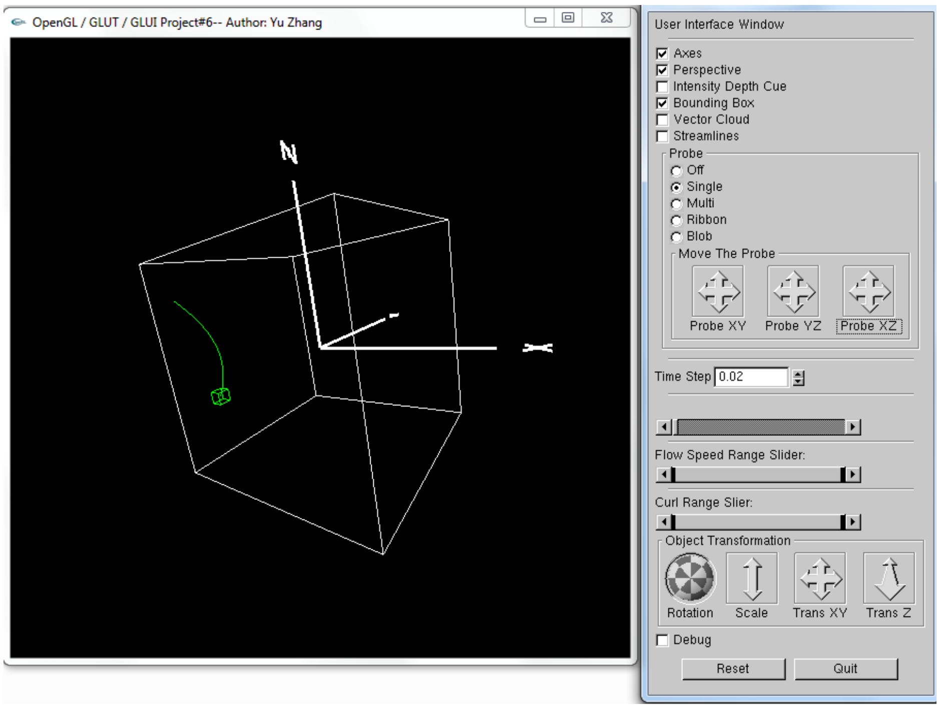
In this part, we firstly need to draw a small probe. I draw a small cube as the probe since I think it is easy to draw and what is more important is that we can use this cube to implement Blob Tracing.

First, we know that probe can be moved by mouse motions, so I use Glui->add_translation to make 3 arrow-cross for getting motions from mouse. This part may be a little bit tricky since the probe's position is represent as (x, y, z), but there're 3 arrow-cross which means one x is contained 2 parts: one from Probe XY and the other from Probe XZ, same method for y and z. For this issue, I firstly create 3 2-element arrays to record three arrow-crosses' inputting respectively. Then I use one 3-element array ProbeXYZ[3] to store the probe's position.

```
    ProbeXYZ[0] = ProbeXY[0] + ProbeXZ[0];
    ProbeXYZ[1] = ProbeXY[1] + ProbeYZ[1];
    ProbeXYZ[2] = -ProbeXZ[1] - ProbeYZ[0];
```

So we can see the little green cube. Single stream line and multiple streamlines is easy, you can just pass the position of Probe and then call Streamline() to draw the lines.

The following 2 graphs shows different positions of probe with different single streamlines.



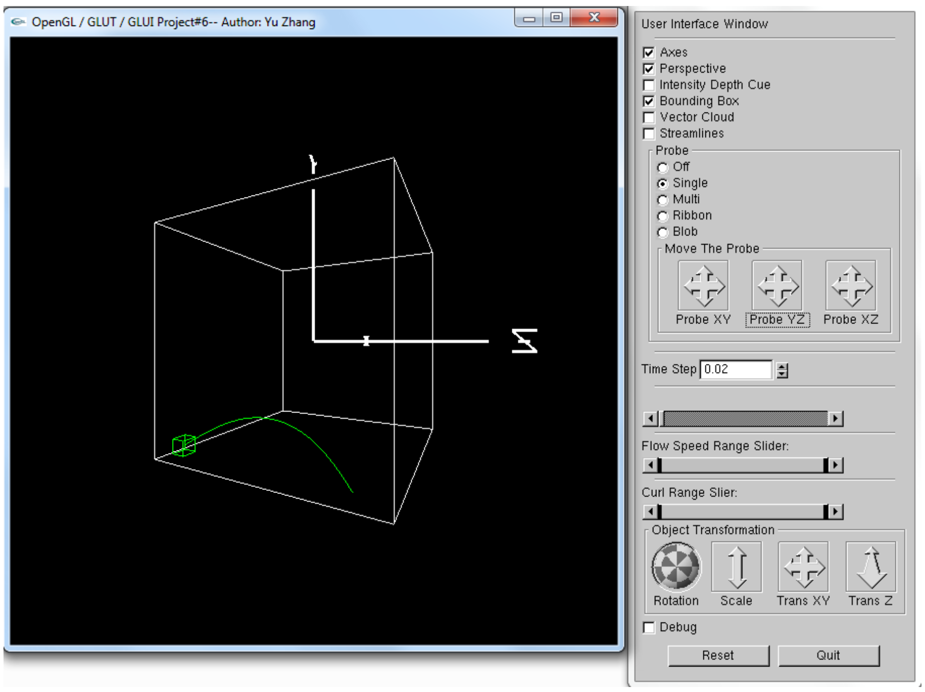
OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off [✓ Single] Multi Ribbon Blob

Time Step 0.02



OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

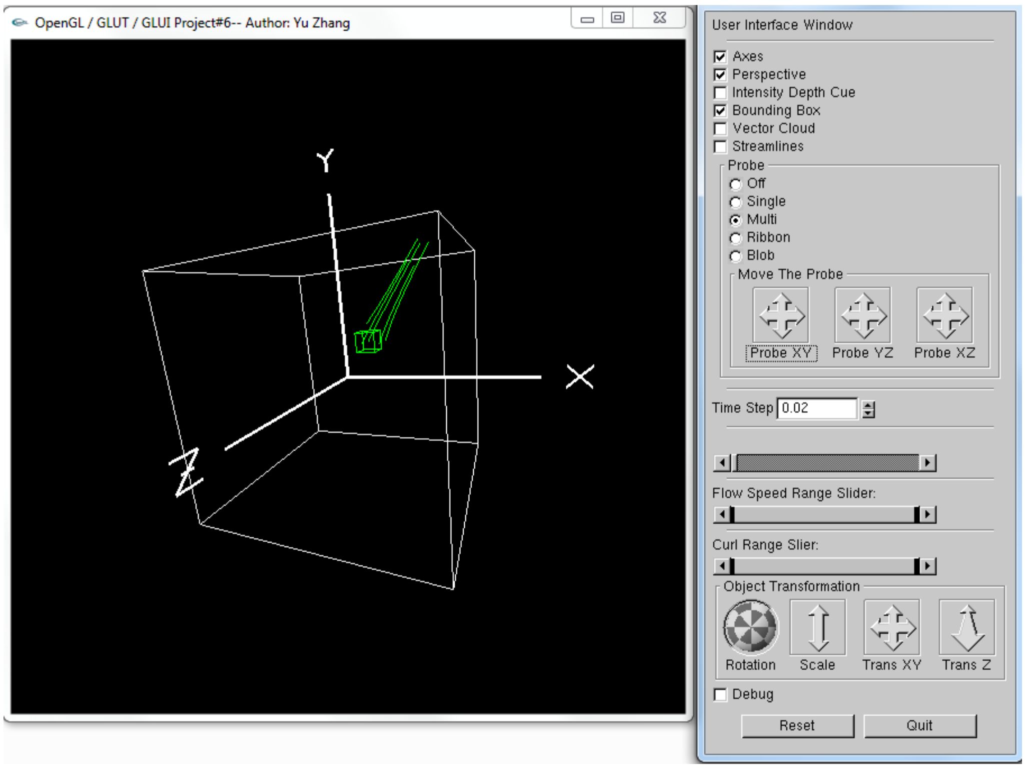
- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off [✓ Single] Multi Ribbon Blob

Time Step 0.02

e) Probe with multiple streamlines

The idea to draw multiple streamlines is almost same as single streamline. I just need to find couples of different points around the probe and draw these points' streamlines.


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

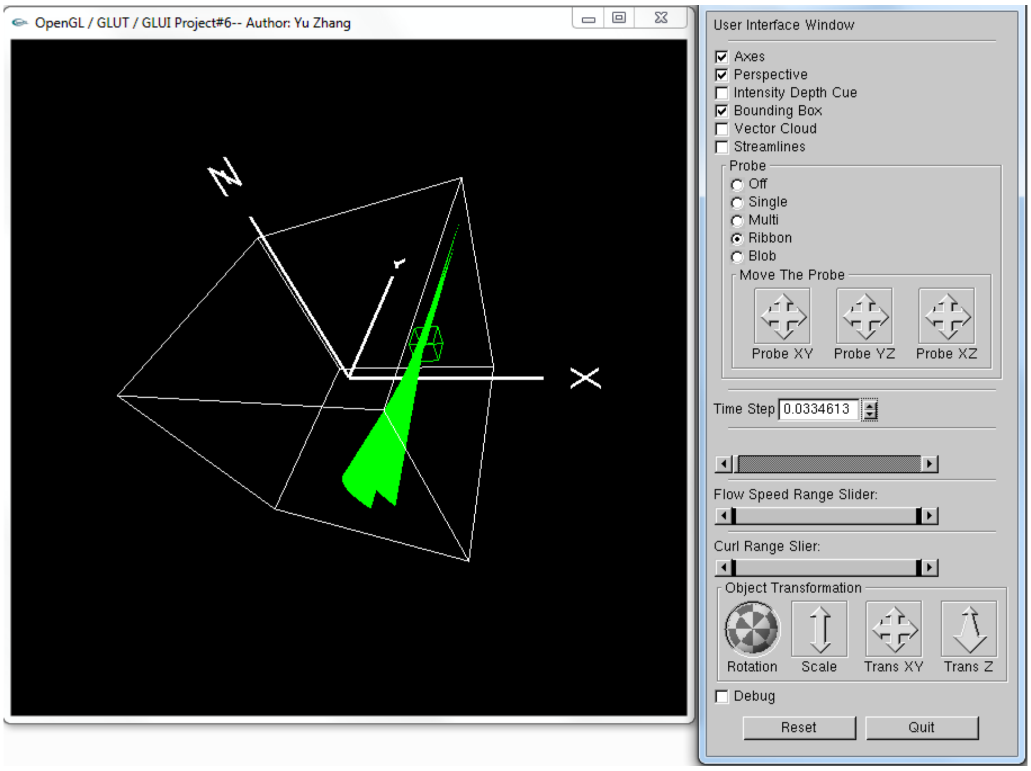
- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off, Single, [✓ Multi], Ribbon, Blob

Time Step 0.02

- [ ] Debug

f) Ribbon trace


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off, Single, Multi, [✓ Ribbon], Blob

Time Step 0.0334613

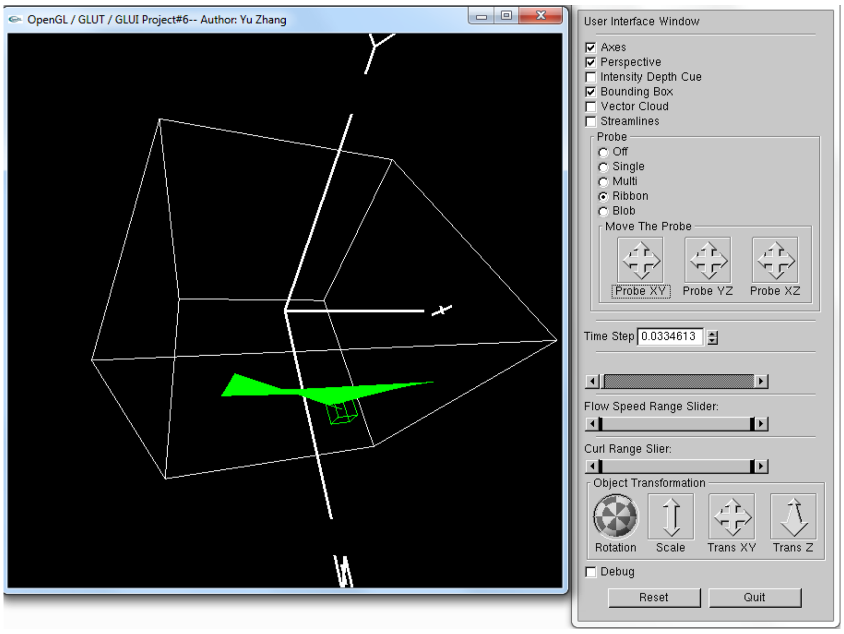
- [ ] Debug

Ribbon trace is to connect a bunch of streamlines and color them so that the effect looks like polygonal. In this part I think the most important point is to use iteration correctly to find out the points on each streamline. The key codes are shown as below:

```
glShadeModel(GL_SMOOTH);
        glBegin(GL_QUADS);
        for (int i = 0; i <= 7; i++)
        {
            p(x,y,z);
            pnext(x`,y`,z`);
            glColor3f(0, 1, 0);
            for (int j = 0; j < 50; j++)
            {
                currentX = px;
                currentY = py;
                currentZ = pz;
                currentXnext = pxnext;
                currentYnext = pynext;
                currentZnext = pznext;

                Advect(&px, &py, &pz);
                Advect(&pxnext, &pynext, &pznext);
                glVertex3f(currentXnext, currentYnext, currentZnext);
                glVertex3f(currentX, currentY, currentZ);
                glVertex3f(px, py, pz);
                glVertex3f(pxnext, pynext, pznext);
            }
        }
        glEnd();
```

The following graph shows the different points' ribbon trace:


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

[✓] Axes
[✓] Perspective
[ ] Intensity Depth Cue
[✓] Bounding Box
[ ] Vector Cloud
[ ] Streamlines

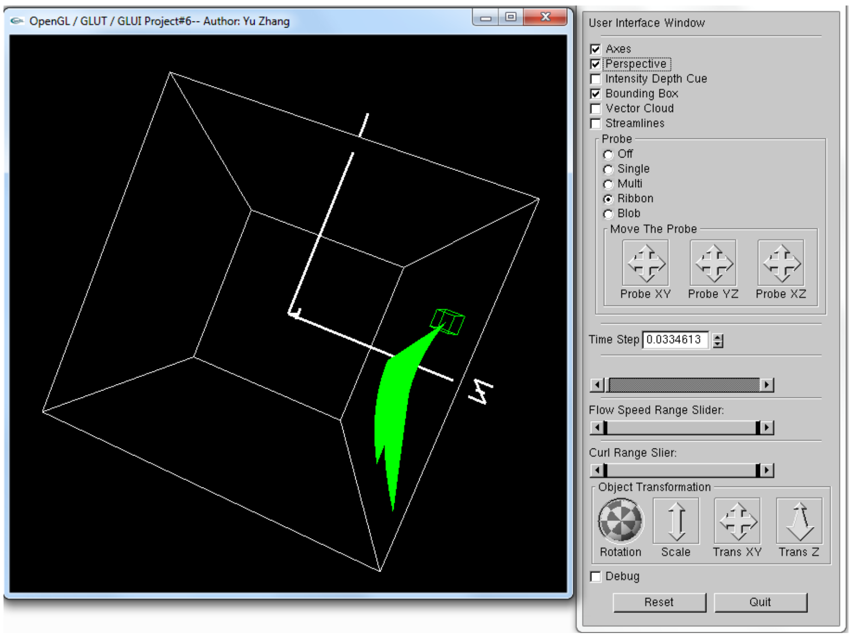
Probe: Off / Single / Multi / [✓ Ribbon] / Blob

Time Step 0.0334613

Flow Speed Range Slider:

Curl Range Slier:

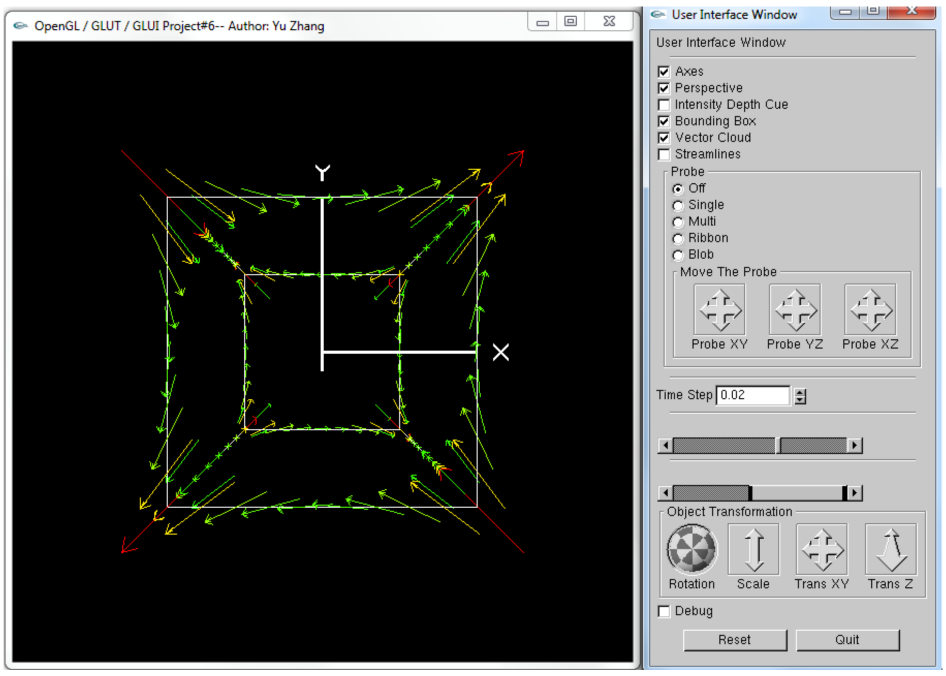
[ ] Debug



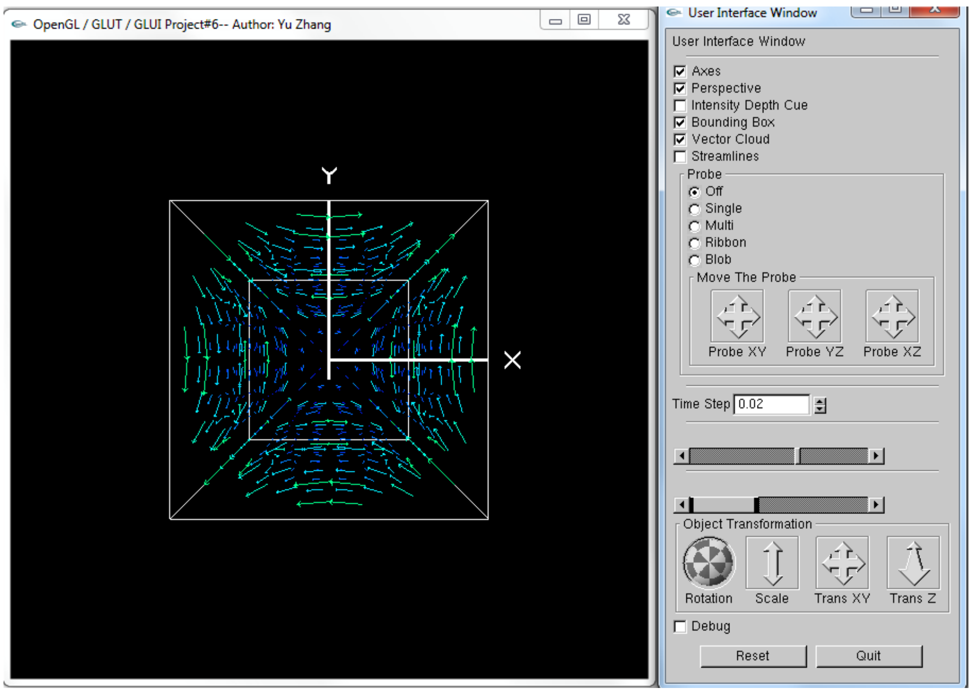
[✓] Axes
[✓] Perspective
[ ] Intensity Depth Cue
[✓] Bounding Box
[ ] Vector Cloud
[ ] Streamlines
Probe: Off / Single / Multi / [✓ Ribbon] / Blob
Time Step 0.0334613
[ ] Debug

g) Flow Speed Range Slider

In this part, the point is to use slider to filter data and show relative vector cloud. I use magnitude as the standard to filter data.



[✓] Axes
[✓] Perspective
[ ] Intensity Depth Cue
[✓] Bounding Box
[✓] Vector Cloud
[ ] Streamlines
Probe: [✓ Off] / Single / Multi / Ribbon / Blob
Time Step 0.02
[ ] Debug



User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [✓] Vector Cloud
- [ ] Streamlines

Probe: [✓ Off] Single Multi Ribbon Blob

Time Step 0.02

- [ ] Debug

h) Blob trace

As extra credit part, Blob trace is much harder for me than other parts. This part has two main points: draw the trace and animation.

For drawing the trace, we need to use the probe cube's eight vertices. I create 8*3 matrix to store each point's location. Then use iterations to call Advect() to find out all point's information. Each time I treat the eight points that found by each iteration as a cube, and I call glBegin(GL_QUADS) to draw that kind of "cube".

For animating, I use the way that professor introduced in the class. In function Animate(), I modify

GLUI_Master.set_glutIdleFunc(NULL);

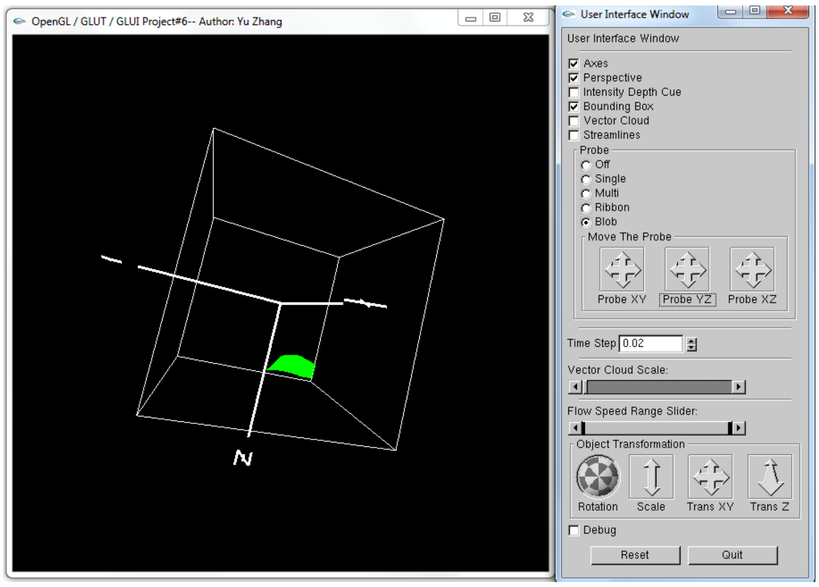
to

GLUI_Master.set_glutIdleFunc(Animate);

And in the fuction Animate(), I set *times* as times +=20; and set a global value MAX_ITERATIONS, when times equals MAX_ITERATIONS, the program will draw the "cubes" again. So the animation is made by this idea.

The part I waste most time is to think about how to make the animation, I had a mistake is that I put drawing cube part in the for loop of iterations for (int i = 0; i < times; i+=5){}, this make the effect so weird and the solution is that I just put the drawing part out of this for loop.

The following graphs are showing how Blob Trace is like. Since they are pictures, so I think it is good to look into the .exe file.


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

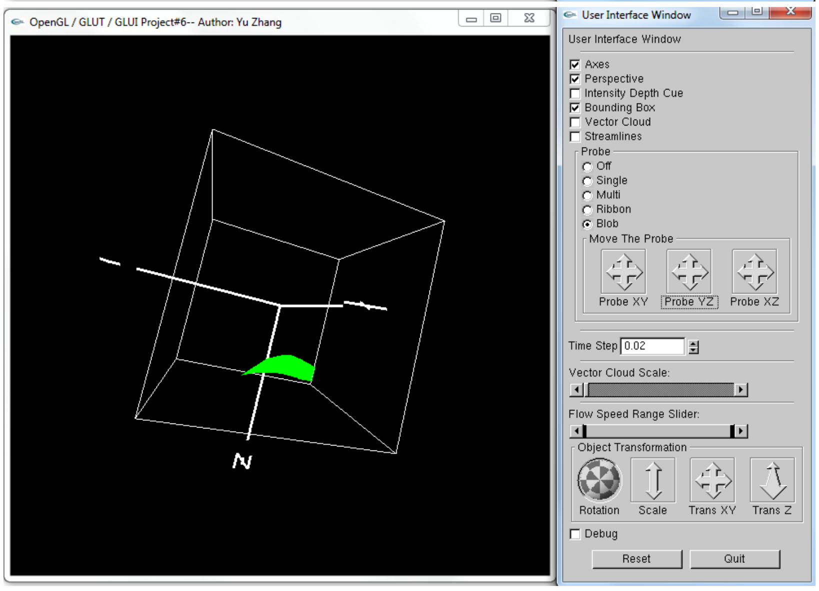
User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off / Single / Multi / Ribbon / [✓ Blob]

Time Step 0.02

- [ ] Debug


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

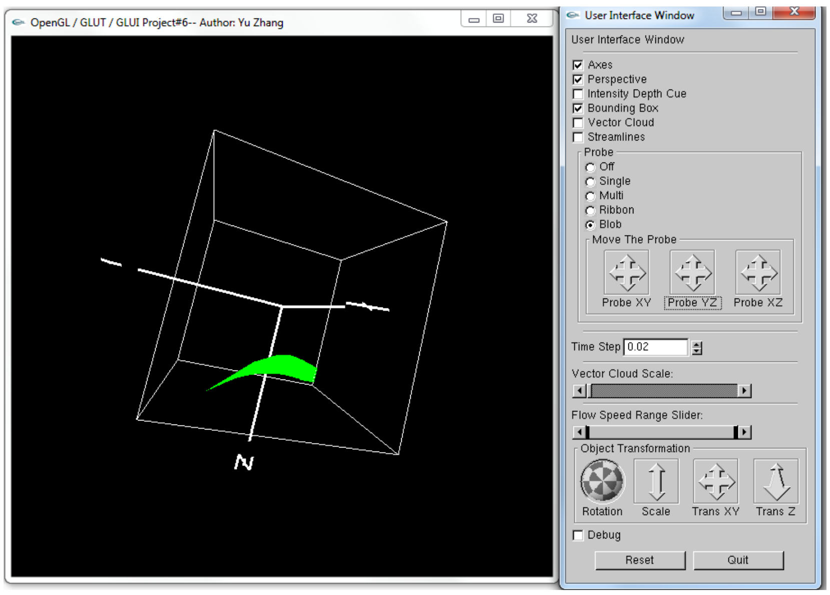
User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off / Single / Multi / Ribbon / [✓ Blob]

Time Step 0.02

- [ ] Debug


OpenGL / GLUT / GLUI Project#6-- Author: Yu Zhang

User Interface Window

- [✓] Axes
- [✓] Perspective
- [ ] Intensity Depth Cue
- [✓] Bounding Box
- [ ] Vector Cloud
- [ ] Streamlines

Probe: Off / Single / Multi / Ribbon / [✓ Blob]

Time Step 0.02

- [ ] Debug