Programming Assignments 3 and 4 – 601.455/655 Fall 2018

Score Sheet (hand in with report) Also, PLEASE INDICATE WHETHER YOU ARE IN 601.455 or 601.655

Name 1	*Tianyu Song*
Email	*tsong11@jhu.edu*
Other contact information (optional)	
Name 2	*Huixiang Li*
Email	*lhuixia1@jhu.edu*
Other contact information (optional)	
Signature (required)	I (we) have followed the rules in completing this assignment Tianyu Song Huixiang Li

Grade Factor		
Program (40)		
Design and overall program structure	20	
Reusability and modularity	10	
Clarity of documentation and programming	10	
Results (20)		
Correctness and completeness	20	
Report (40)		
Description of formulation and algorithmic approach	15	
Overview of program	10	
Discussion of validation approach	5	
Discussion of results	10	
TOTAL	100	

a) Methods and Algorithms in Our Programs

1. Bounding Sphere

To find the closest triangle to a given point in a three-dimensional space, we construct bounding spheres around each triangle, and use this to reduce the number of careful checks required while searching all the triangles. For each triangle, there are three points a, b and c . To find the bounding sphere of this triangle, we must find its center and radius.

Let edge (a,b) is the longest side of triangle

Step 1: Given three points a, b and c, compute

$$f = (a+b)/2$$

Step 2: define

$$u = a - f, \quad v = c - f, \quad d = (u \times x) \times u$$

Step 3: Calculate the sphere center q lines along the line

$$q = f + \lambda d$$

$$(\lambda d - v)^2 \le (\lambda d - u)^2 \quad \Rightarrow \quad \lambda \ge \frac{v^2 - u^2}{2d \bullet (v-u)} = \gamma$$

If $\gamma \le 0$, then $\lambda = 0$. Else $\lambda = \gamma$

2. FindClosestPoint(a,[p,q,r])

Step 1: solve the least squares problem for λ and μ

$$a - p \approx \lambda(q-p) + \mu(r-q)$$

Step 2: compute c

$$c = p + \lambda(q-p) + \mu(r-p)$$

Step 3: if $\lambda \ge 0$, $\mu \ge 0$, and $\lambda + \mu \le 1$

c lies in the triangle

c $\Rightarrow$ the closest point

Step 4: if not satisfy the condition ($\lambda \ge 0$, $\mu \ge 0$, and $\lambda + \mu \le 1$)

The closest point is on the border of the triangle

3. Find Closest Point on Triangle

Step 1: if $\lambda < 0$

Closest point = ProjectionOnSegment(c,r,p)

Step 2: if $\mu < 0$

Closest point = ProjectionOnSegment(c,p,q)

Step 1: if $\lambda + \mu > 1$

Closest point = ProjectionOnSegment(c,q,r)

4. **ProjectionOnSegment(c,p,q)**

Step 1: calculate λ

$$\lambda = \frac{(c-p)\cdot(q-p)}{(q-p)\cdot(q-p)}$$

Step 2: find λ^*

$$\lambda^* = max(0, min(\lambda, 1))$$

Step 3: find c^*

$$c^* = p + \lambda^*(q-p)$$

5. **Simple Search with Bounding Spheres**

Step 1: assume triangle i has corners[p,q,r]

Step 2: surrounding sphere i has radius ρ center q, and let bound equal to infinity

Step 3: compute for-loop

```
for i = 1 to N {
    if |q − a| − ρ ≤ bound {
        h = FindClosestPoint(a, [p, q, r]);
            if |h − a| < bound {
                c = h;
                bound = |h − a|;
            }
    }
}
```

b) Description of Functions

functions	input variable	output variable
BoundingSphere.py	**v xyz coordinates of vertices in CT coordinates(N*3 matrix), tri vertex indices of the three vertices for each triangle(M*3 matrix)**	**qq center of the sphere(M*3 matrix), rr radius of the sphere(M*1 matrix)**
Calculate the bounding sphere of the given triangle		
FindClosestPoint.py	**a, tri vertex indices of the three vertices for each triangle**	**c the closest point calculated**
Find the closest point with a given triangle		
ProjectionOnSegment.py	**c,p,q three points of the triangle**	**c_s projection on segment c***
apply the given three inputs to calculate the projection on segment c*		
ReadInput.py	**/**	**/**
Define ReadData(), ReadData1(), ReadMesh() functions		
ReadData()	**filename *SampleReadingsTest.txt in the data directory**	**NS the sum of points recorded in A,B,D frames, Nsamps number of sample frames, data point cloud data in the frames**
Read in 'PA3--SampleReadingsTest.txt' file in the data directory, return Ns, Nsamps, and data point clouds**		

<table>
<tr><td>ReadData1()</td><td>filename *Problem3-Body*.txt in the data directory</td><td>N number of markers, a xyz coordinates of marker LEDs in body coordinates, tip xyz coordinates of tip in body coordinates</td></tr>
<tr><td colspan="3">Read in '**Problem3-Body*.txt' file in the data directory, return N, number of markers, and xyz coordinates for a & tip</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>ReadMesh()</td><td>filename Problem3MeshFile.sur in the data directory</td><td>Nv number of vertices, v xyz coordinates of vertices in CT coordinates, Nt number of triangles, tri vertex indices of the three vertices for each triangle</td></tr>
<tr><td colspan="3">Read in '*Problem3MeshFile.sur' file in the data directory, return number of vertices and number of triangles, xyz coordinates for the vertices in CT coordinates, vertex indices of the three vertices for each triangle</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>ProgrammingAssignment3.py</td><td>/</td><td>./OUTPUT/*.txt</td></tr>
<tr><td colspan="3">automatically import input files in the data directory and output results for PA3 as *.txt files for each test set</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>ErrorAnalysis.py</td><td>/</td><td>./error_analysis/*.txt</td></tr>
<tr><td colspan="3">Automatically import results from the ProgrammingAssignment3.py and calculate the error between the debug files</td></tr>
</table>

c) Results of Functions

For the dataset ABCDEF we show the magnitude of difference between d_k and c_k and compare our output and given output. Full result can be found in the OUTPUT folder.

Output			
	Given Output	**Our Output**	**Residual**
A	0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	0.002 0.001 0.001 0.004 0.002 0.001 0.000 0.001 0.005 0.001 0.003 0.001 0.003 0.001 0.001	-0.002 -0.001 -0.001 -0.004 -0.002 -0.001 0.000 -0.001 -0.005 -0.001 -0.003 -0.001 -0.003 -0.001 -0.001
B	2.146 1.087 1.421 3.240 0.220 1.018 0.471 2.089 0.755 1.263 1.257 2.176 2.993 0.941 1.556	2.139 1.088 1.423 3.235 0.219 1.018 0.471 2.088 0.757 1.266 1.249 2.170 2.988 0.944 1.548	0.007 -0.001 -0.002 0.005 0.001 0.000 0.000 0.001 -0.002 -0.003 0.008 0.006 0.005 -0.003 0.008

C	0.950	0.951	-0.001
	1.756	1.756	0.000
	0.278	0.279	-0.001
	0.964	0.964	0.000
	0.520	0.516	0.004
	0.305	0.304	0.001
	0.836	0.834	0.002
	1.817	1.813	0.004
	0.284	0.283	0.001
	1.024	1.021	0.003
	0.833	0.829	0.004
	0.996	0.993	0.003
	0.506	0.511	-0.005
	0.382	0.380	0.002
	0.483	0.476	0.007
D	0.224	0.233	-0.009
	0.267	0.267	0.000
	3.285	3.290	-0.005
	2.090	2.097	-0.007
	0.868	0.867	0.001
	1.720	1.719	0.001
	0.607	0.609	-0.002
	3.757	3.746	0.011
	1.608	1.610	-0.002
	0.740	0.739	0.001
	0.931	0.933	-0.002
	0.238	0.240	-0.002
	2.661	2.672	-0.011
	0.635	0.634	0.001
	3.258	3.252	0.006
E	0.383	0.385	-0.002
	2.733	2.730	0.003
	1.087	1.097	-0.010
	1.112	1.110	0.002
	2.343	2.349	-0.006
	4.228	4.230	-0.002
	2.093	2.092	0.001
	0.008	0.004	0.004
	2.021	2.028	-0.007
	0.944	0.943	0.001
	1.369	1.365	0.004
	2.664	2.665	-0.001
	1.311	1.317	-0.006
	1.882	1.888	-0.006
	1.450	1.445	0.005

F	0.976 2.044 2.143 1.918 0.055 1.185 1.726 0.205 0.644 0.003 1.541 0.162 1.455 1.878 0.918	0.976 2.046 2.141 1.911 0.053 1.181 1.719 0.204 0.643 0.003 1.547 0.164 1.455 1.877 0.921	0.000 -0.002 0.002 0.007 0.002 0.004 0.007 0.001 0.001 0.000 -0.006 -0.002 0.000 0.001 -0.003
G		16.94 9.43 61.94 17.03 9.40 63.19 1.248 -45.16 -11.66 -28.20 -43.51 -10.89 -27.12 2.120 2.56 -8.65 64.76 2.55 -6.87 63.29 2.311 -17.19 2.98 -38.16 -17.78 5.84 -41.70 4.588 4.00 21.06 22.17 4.09 23.72 22.54 2.686 -6.73 -8.86 34.46 -3.79 -5.97 33.11 4.339 -7.23 -5.98 59.23 -4.08 -3.35 59.50 4.109 -40.23 -24.86 -14.71 -37.41 -22.49 -16.42 4.063 23.12 20.49 -5.67 24.78 24.27 -5.79 4.132 24.07 -14.43 -19.49 23.81 -13.40 -18.88 1.217 19.16 21.39 9.25 20.12 24.62 9.99 3.443 -6.35 -7.70 42.80 -3.71 -4.63 42.73 4.046 -20.10 -9.33 -46.50 -20.44 -8.75 -48.84 2.440 -8.70 10.74 9.76 -8.82 10.82 9.69 0.174 -0.91 -8.11 65.19 -0.06 -5.93 63.48 2.901 -3.04 2.15 64.91 -2.79 2.34 63.27 1.664 -21.44 -19.69 -46.27 -21.51 -20.02 -47.07 0.869 -37.66 0.84 -19.79 -38.76 1.85 -19.46 1.527 15.15 -8.75 25.94 14.88 -6.43 25.70 2.350 16.50 -8.87 22.97 16.18 -6.52 22.73 2.384	

H		-40.56 -14.00 -12.92 -39.45 -13.81 -13.68 1.358 15.13 18.61 -29.41 14.79 19.60 -30.59 1.580 30.83 -11.03 -16.15 30.31 -10.63 -16.06 0.659 -31.10 -25.23 -10.74 -30.52 -24.41 -11.49 1.254 -1.32 -12.74 17.72 -1.28 -10.06 17.17 2.732 -14.65 -31.87 -26.39 -14.69 -31.74 -26.39 0.132 -6.27 -7.99 44.71 -3.55 -4.67 44.25 4.322 -40.60 -6.86 -12.70 -39.60 -7.13 -13.37 1.234 8.91 18.34 33.20 8.62 22.01 33.85 3.743 16.38 -13.72 3.22 16.09 -12.85 2.81 1.006 0.97 -10.62 30.93 0.80 -7.65 30.65 2.984 2.11 13.07 -3.42 1.73 13.56 -3.71 0.684 -24.63 6.39 -40.32 -24.72 6.61 -40.49 0.295 -2.86 -19.91 -35.14 -2.46 -20.19 -35.40 0.546 -31.97 6.67 -36.87 -31.71 6.20 -36.64 0.581 -2.75 -8.52 63.22 -1.36 -5.92 62.22 3.113 2.55 -3.65 63.74 2.54 -3.66 63.42 0.319 36.06 0.03 -14.40 36.98 -0.62 -14.78 1.190 24.30 -0.03 38.17 23.17 0.61 37.79 1.348 22.24 5.07 54.54 21.78 5.17 54.48 0.478	
J		33.05 -11.10 -17.87 30.97 -9.48 -17.52 2.664 34.47 2.58 -24.53 33.30 2.62 -23.71 1.428 -6.20 -29.00 -14.48 -8.67 -26.24 -16.54 4.232 20.36 19.93 23.28 21.06 21.11 23.61 1.410 -37.42 -13.59 -30.21 -40.27 -13.07 -30.98 2.997 -21.61 7.76 -18.74 -21.58 8.63 -17.53 1.494 -10.81 7.20 31.56 -9.35 8.27 31.19 1.847 -18.07 -35.12 -34.11 -18.66 -31.81 -33.34 3.444 21.53 -9.67 19.53 20.52 -6.14 19.16 3.686 17.61 18.90 38.25 18.82 19.77 38.49 1.507 -8.88 5.09 -34.63 -8.45 5.50 -34.86 0.631 4.00 6.17 -12.20 4.05 9.16 -13.71 3.351 33.11 17.48 -15.75 32.62 17.09 -15.59 0.645 -35.70 -19.30 -14.11 -36.98 -19.91 -13.37 1.597 4.03 -20.92 -8.77 2.79 -17.43 -9.35 3.751 -5.08 10.27 10.91 -7.75 11.97 9.16 3.622 -6.75 1.68 27.45 -8.06 1.47 27.72 1.355 8.32 20.78 21.58 8.16 23.87 21.99 3.117 -39.17 -8.47 -27.66 -42.30 -9.93 -29.66 3.992 -22.15 -29.16 -10.39 -22.00 -27.36 -11.93 2.369	

d) Discussion of the Results and Analysis

In conclusion, our program works well, as one can observe from the comparison between the given debug output file and ours. Most of the error residue is under 0.01, except one is 0.011, which means the calculation errors are in reasonable range. Generally speaking, our program can achieve all the goals of this assignment with good performance.

e) **Work Distribution**

Name	Work
Tianyu Song	**● Collaborated with Huixiang on implementing the boundary sphere, find closest point and project on segmentation** **● Finished ProgramAssignment3.py** **● Tested and debugged programs**
Huixiang Li	**● Collaborated with Tianyu on implementing the boundary sphere, find closest point and project on segmentation** **● Finished ErrorAnalysis.py** **● Tested and debugged programs**

References

[1] Find Closest Point from Dense Cloud (P4-P8 in lecture slide 'Finding point-pairs')