Programming Assignment 5 – 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	

NOTE: This is an optional assignment.

If you hand it in, I will use the grade to replace the lowest other programming assignment or written homework assignment with one exception:

You may not drop **both** HW#3 and HW#4. If these are your two lowest grades, then I will drop the lower of those two under the drop 1 homework scenario and replace the next lowest grade (other than the other of HW#3-4) with this score

CIS Programming Assignment #5

a) Overall Algorithm & Iteration Logic

Step 0: Read inputs, point clouds and meshes
Step 1: calculate mesh point cloud M, tip point cloud d
Step 2: initialize the $F_{reg} = I_4$
Step 3: Calculate $s = F_{reg} * d$
Step 4: For sample points, find closest matches to current mesh and find the closest point, $c = FindClosestPoint(d, M)$
Step 5: Find the transformation $F_t = PointCloudRegistration(s, c)$
Step 6: Calculate $F_{reg} = F_t * F_{reg}$
Step 7: Solve λ_m in the equation $Fd_k = c_k = q_{o,k} + \sum_{m=1}^{N\ modes} \lambda_m q_{m,k}$,
where $q_{m,k} = \zeta_k m_{m,s} + \xi_k m_{m,t} + \psi_k m_{m,u}$
Step 8: If change the shape parameters then update bounding boxes
Step 9: While not converge: iterate Step3~Step8; else: finish iteration and go to Step 10
Step 10: Calculate distance between s and c, output result to .txt file

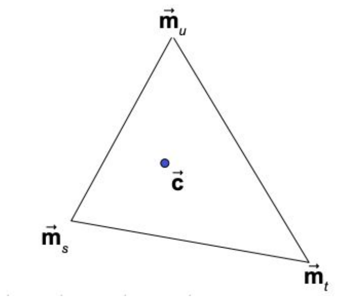
b) Methods and Theory

1. **Barycentric Coordinates of Deforming Triangle**

The closest point always can be represented as a linear combination of triangle's vertices

$$c = \zeta m_s + \xi m_t + \psi m_u$$
$$\zeta + \xi + \psi = 1$$

where m_s, m_t and m_u are the corresponding vertices



$$m_s = m_{0,s} + \sum_{m=1}^{N\ modes} \lambda_m m_{m,s}$$
$$m_t = m_{0,t} + \sum_{m=1}^{N\ modes} \lambda_m m_{m,t}$$
$$m_u = m_{0,u} + \sum_{m=1}^{N\ modes} \lambda_m m_{m,u}$$

2. **Bounding Sphere**

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 $\lambda = \gamma$

3. **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

4. **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 3: if $\lambda + \mu > 1$, Closest point = ProjectionOnSegment(c,q,r)

where ProjectionOnSegment(c,p,q) is

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 $|q - a| - \rho \le bound${
                $h = FindClosestPoint(a, [p, q, r])$;
                        if $|h - a| < bound${
                                $c = h$;
                                $bound = |h - a|$;
                        }
        }
```

}

c) 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)	a list of sphere class with properties: center, radius and object
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*		
Deform_tri_set.py	***Modes***; Lambda, an array of lambda; tri, vertex indices of the three vertices for each triangle;	Tri_set, triangle vertices
update the vertices set and triangle set by considering current deformation		
ProgrammingAssignment5.py	/	./OUTPUT/*.txt
automatically import input files in the data directory and output results for PA5 as *.txt files for each test set		

Validation.py	**/**	**./error_analysis/*.txt**
automatically import results from the ProgrammingAssignment5.py and calculate the error between the debug files		

d) Results

For the dataset ABCDEF we show the magnitude of difference between d_k and c_k and compare our output and given output. We also output the lambda values and compare them with the given results. Full result can be found in the OUTPUT folder.

Data	Mean of distance [our output]	Mean of distance [given output]	Mean of difference between our output and debug output
A	0.37589333	0	0.37589333
B	0.39381333	0	0.39381333
C	0.26827333	0	0.26827333
D	0.31146	0	0.31146
E	0.34186	0.069085	0.272775
F	0.1843425	0.0618625	0.12248

Unknown Datasets:

Data	Mean of distance [our output]	Standard deviation of our output
G	0.37152	0.30757377
H	0.36581	0.27493775
J	0.2772075	0.2102719
K	0.31427333	0.23429539

Data	lambda					
A-OUTPUT	30.0413	136.6801	47.016	-172.2492	23.533	-44.8431
A-DEBUG	32.6922	137.5444	44.071	-179.9443	24.9097	-45.6071
A-difference	2.6509	0.8643	2.945	7.6951	1.3767	0.764
B-OUTPUT	-117.78	163.02	38.20	-163.49	3.62	110.70
B-DEBUG	-124.06	159.60	35.89	-167.41	12.80	118.67
B-difference	6.2792	3.4214	2.3111	3.9183	9.1798	7.9755
C-OUTPUT	126.3327	7.3252	-71.873	78.5346	46.569	-8.7988
C-DEBUG	127.5547	8.9497	-76.417	75.5808	43.4553	-9.1477
C-difference	1.222	1.6245	4.544	2.9538	3.1137	0.3489
D-OUTPUT	114.0399	89.6852	53.4061	49.9667	-4.1371	-61.5812
D-DEBUG	119.1634	93.2266	60.1081	49.0457	-5.8873	-62.185
D-difference	5.1235	3.5414	6.702	0.921	1.7502	0.6038
E-OUTPUT	109.7555	117.01	56.0421	96.456	26.6269	125.6033
E-DEBUG	112.0191	125.2264	59.2146	100.304	27.7944	122.2337
E-difference	2.2636	8.2164	3.1725	3.848	1.1675	3.3696
F-OUTPUT	-18.6681	28.0931	-23.5222	-44.0167	-66.6875	37.8809
F-DEBUG	-21.3094	29.6933	-26.5427	-43.4351	-68.5104	40.49
F-difference	2.6413	1.6002	3.0205	0.5816	1.8229	2.6091

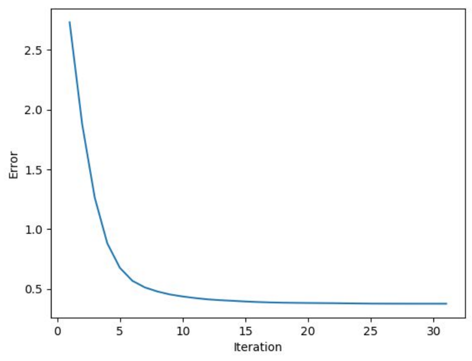
Unknown Datasets:

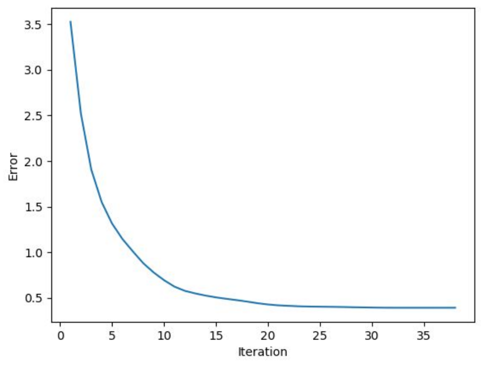
Data	lambda

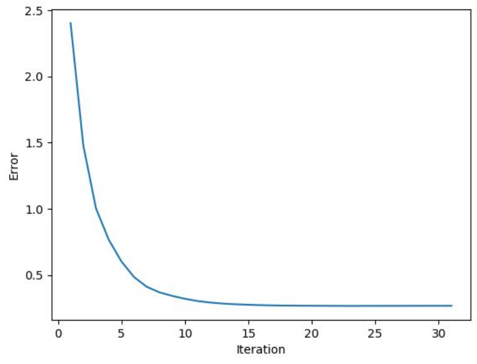
G-OUTPUT	**-77.0261**	**-136.3151**	**-95.0636**	**140.2193**	**-79.8817**	**-129.0578**
H-OUTPUT	**8.2995**	**125.5882**	**-82.6129**	**-149.5735**	**87.9976**	**-51.1081**
J-OUTPUT	**51.8358**	**-66.1709**	**3.1328**	**-141.5920**	**-49.6475**	**-26.5326**
K-OUTPUT	**-112.8076**	**83.4077**	**16.9253**	**-109.1979**	**-42.3484**	**-110.0631**

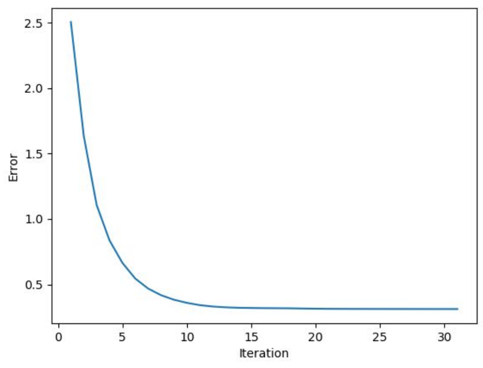
e) Discussion and Validation

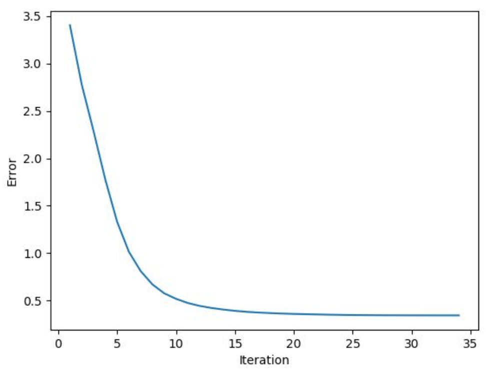
For the purpose of debugging and validation, we plotted the figure of the mean of distance vs number of iteration. From these figures, we can see that the value converges after 30 iterations. Also, the difference between our output and the debug file (as shown in the result section) reveals that both the distance and the lambda are very close to the ground truth value, which means our result are correct.

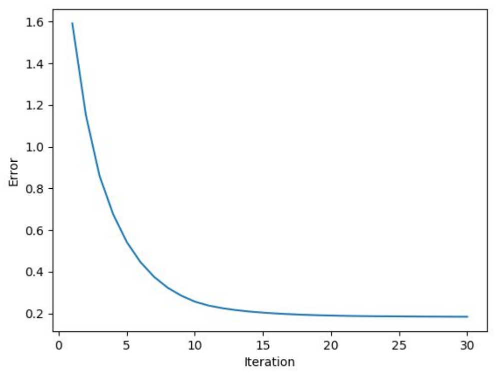


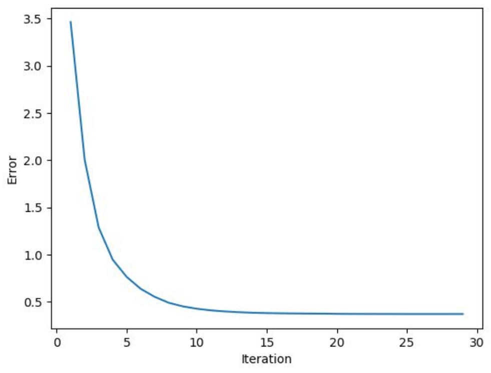


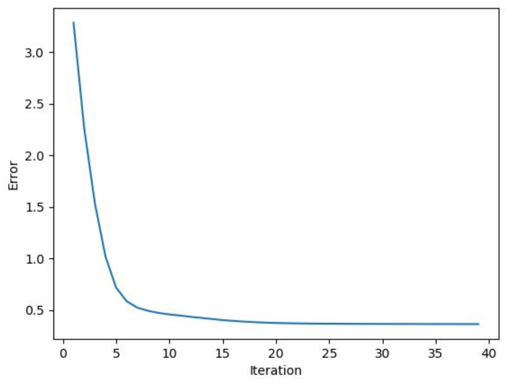


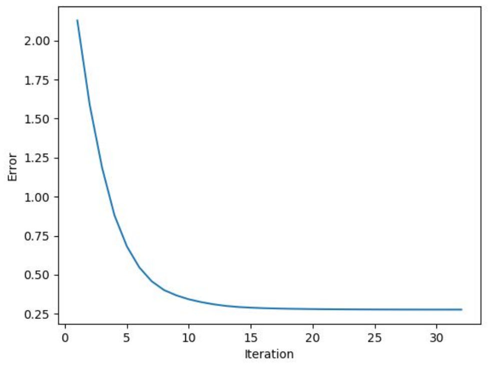


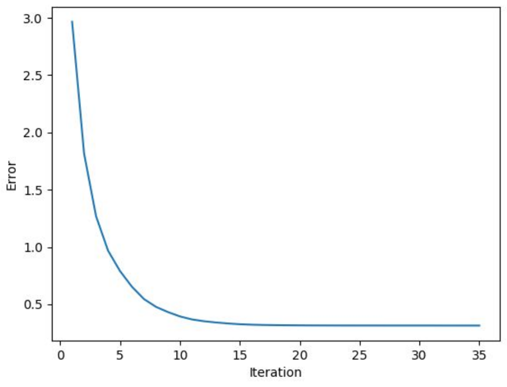












f) Work Distribution

Name	Work
Tianyu Song	● Collaborated with Huixiang on building the overall program structure and icp iteration algorithm ● Finished ProgrammingAssignment5.py and validation.py ● Tested and debugged programs

Huixiang Li	● Collaborated with Tianyu on building the overall program structure and icp iteration algorithm ● Finished ProgrammingAssignment5.py and validation.py ● Tested and debugged programs

References

[1] Barycentric Coordinates of Deforming Triangle (P1-P4 in lecture slide ‘Notes for PA5: Deformable Registration to a Statistical Shape Model’)

[2] Find Closest Point from Dense Cloud (P4-P8 in lecture slide ‘Finding point-pairs’)