Programming Assignments 1 & 2 (circle one)
601.455 and 601/655 (circle one) Fall 2018

Score Sheet (hand in with report)

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	

CIS Programming Assignment #2

a) Methods and Algorithms in Our Programs

1. Distortion correction function[1]

Step 1: normalize measured data (scale to box)

$$u = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Step 2: obtain F matrix

$$F_{ijk}(u_x, u_y, u_z) = B_{5,i}(u_x) B_{5,j}(u_y) B_{5,k}(u_z)$$
$$B_{N,k}(u, v) = (N, k) u^{N-k} v^k$$
$$v = 1 - u$$

Step 3: calculate correction function matrix

$$[F]\ [C] = [P]$$
$$[C] = find\ least\ square(\ [F]^{-1} [P]\ )$$

Step 4: dewarp new measured data using the correction function

$$P_{dewarped} = [F(scale\ to\ box(P_{measure}))]\ [C_{dewarp-matrix}]$$

2. Point cloud to point cloud registration[2]

Step 1: compute the center of the two point clouds

$$\overline{a} = \frac{1}{N} \sum_{i=1}^{N} \vec{a_i} \qquad \tilde{a}_i = \vec{a_i} - \overline{a}$$

$$\overline{b} = \frac{1}{N} \sum_{i=1}^{N} \vec{b_i} \qquad \tilde{b}_i = \vec{b_i} - \overline{b}$$

Step 2: use direct techniques to solve R that minimizes, the details are shown in **5**

$$\sum_i (R\tilde{a}_i - \tilde{b}_i)^2$$

Step 3: calculate p base on the solved R

$$p = \overline{b} - R\overline{a}$$

Step 4: obtain the rigid body transformation

$$F = [R, p]$$

3. Pivot calibration

The known is a set of transformations from the frames of the tracker device to the different probe frames. The unknown is the coordinates of the dimple relative to the tracker device and those of the tip relative to probe frames, which are constant.

Given

$$F_i = \begin{bmatrix} R_i & P_i \\ 0 & 1 \end{bmatrix}$$

Calculate

$$p_{tip}$$
$$p_{pivot}$$

The formulation:

$$F_i p_{tip} = p_{pivot} = R_i p_{tip} + P_i$$

Thus,

$$R_i p_{tip} - p_{pivot} = -P_i$$

Stack all the equations together, we can form:

$$\begin{bmatrix} \cdots & \cdots \\ R_i & -I \\ \cdots & \cdots \end{bmatrix} \begin{bmatrix} p_{tip} \\ p_{pivot} \end{bmatrix} = \begin{bmatrix} \cdots \\ -P_i \\ \cdots \end{bmatrix}$$

The steps taken to solve this:

Step1: Initial guess by Moore–Penrose inverse [3]:

Let A = $\begin{bmatrix} \cdots & \cdots \\ R_i & -I \\ \cdots & \cdots \end{bmatrix}$ and b = $\begin{bmatrix} \cdots \\ -P_i \\ \cdots \end{bmatrix}$, and x = $\begin{bmatrix} p_{tip} \\ p_{pivot} \end{bmatrix}$, then $\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$.

Step2. Use least square method to solve dx = $\left(A^T A\right)^{-1} A^T (b - Ax)$

Step3. Calculate x = x+dx and evaluate dx.

If dx <1e-5, then return x as the result, else go to Step 2.

b) Description of Functions

functions	input variable	output variable
__init__.py	/	/
# Initialize the environment		
invtrans.py	***F*** rigid transform matrix(4*4)	***inv(F)*** inverse transform of F matrix(4*4)
# Inverse of rigid body transformation, from ***F*** to ***inv(F)***		
point_cloud_registration.py	***c1 c2*** two 3D point clouds(n*3)	***F*** rigid transform matrix(4*4)
# Point cloud registration, compute the rigid transform matrix with two input point clouds		
pivot_calibration.py	***NF*** n rigid transformations matrix(4*4*n)	***P*** two translations in one array(6*1)
# Pivot calibration, compute the two translations using n input transformations matrix		
import_calibrate_data.py	***a_string*** a string relates to the debug file such as 'a','b','c',etc	***C_exp, C_origin*** ground truth and measured point cloud matrix(3*3375)
# Collect in ***calbody*** and ***calreading*** data files, return ground truth and measured point cloud data for distortion correction		
distortion_correction.py	***C_exp, C_origin, C_warp***, point cloud data, txt file, ***order*** a value for polynomial order	***C_dewarp*** dewarded point cloud matrix(3*3375)
# Read in calbody and calreading data file, warped data, and polynomial order, calculate the distortion correction polynomial using the ***calbody*** and ***calreading*** files, using the obtained coefficient to dewarp the warped data		

test_distortion_correction.py	***a_string*** a string relates to the debug file such as 'a','b','c',etc	***P_EM_dimple*** array(3*1)
# Read in calbody and calreading data file, warped data, and polynomial order, calculate the ***P_EM_dimple*** matrix for debugging with the output1 data file		
Assigment2.py	/	***'NAME-OUR-OUTPUT1.TXT'*** Output file that give the ***P_em_dimple*** ***'NAME-OUR-OUTPUT2.TXT'*** Output file that give tip location with respect to the CT image
# Automatically run all steps in the suggested procedure, mainly output the tip location with respect to the CT image in the file ***'NAME-OUR-OUTPUT2.TXT'***		

c) Results of Functions

We first output our P_EM_dimple results as an intermediate step for debugging and compare them with output1 files. The residual between our results and the provided debug results is also calculated.

Output1			
Dataset	Provided Results	Our Results	Residual
a	200.56, 197.74, 208.34	200.56, 197.74, 208.34	0.0 , -0.0 , 0.0
b	200.09, 205.27, 207.97	200.13, 205.31, 208.02	-0.04 , -0.04 , -0.05
c	196.76, 194.97, 204.59	194.51, 191.65, 205.63	2.25 , 3.32 , -1.04
d	207.29, 204.71, 193.73	207.29, 204.71, 193.73	0.0 , -0.0 , 0.0
e	207.88, 200.60, 207.76	208.39, 205.11, 206.88	-0.51 , -4.51 , 0.88

f	192.76, 206.81, 193.92	186.79, 208.73, 198.31	5.97 , -1.92 , -4.39
g	201.62, 191.88, 207.53	206.58, 195.48, 210.03	-4.96 , -3.6 , -2.5
h	/	193.42, 207.43, 210.83	/
i	/	202.62, 198.05, 207.75	/
j	/	202.14, 195.87, 189.19	/

After getting reasonable residual error, then we calculated the tip location with respect to the CT image as the final output results.

Output2			
Dataset	Provided Results	Our Results	Residual
a	104.99, 47.88, 58.45 160.11, 44.40, 62.23 42.16, 171.29, 27.05 161.56, 33.77, 44.41	105.0, 47.88, 58.45 160.12, 44.4, 62.23 42.16, 171.29, 27.05 161.56, 33.78, 44.41	-0.01 , -0.0 , -0.0 -0.01 , -0.0 , -0.0 0.0 , 0.0 , 0.0 0.0 , -0.01 , -0.0
b	114.04, 127.19, 167.22 111.60, 137.83, 48.38 111.21, 98.86, 57.58 54.17, 69.84, 116.05	114.09, 127.15, 167.22 111.63, 137.8, 48.4 111.21, 98.84, 57.62 54.16, 69.77, 116.08	-0.05 , 0.04 , -0.0 -0.03 , 0.03 , -0.02 0.0 , 0.02 , -0.04 0.01 , 0.07 , -0.03
c	42.95, 92.24, 159.77 74.37, 106.52, 154.35 161.83, 142.35, 38.59 103.69, 82.15, 80.72	42.53, 91.3, 159.96 73.11, 105.0, 154.82 161.05, 140.92, 39.53 104.21, 81.1, 80.71	0.42 , 0.94 , -0.19 1.26 , 1.52 , -0.47 0.78 , 1.43 , -0.94 -0.52 , 1.05 , 0.01
d	32.96, 111.13, 78.37 28.89, 48.56, 158.26 102.90, 75.35, 167.64 152.99, 80.68, 152.78	32.96, 111.13, 78.36 28.89, 48.56, 158.25 102.9, 75.35, 167.63 152.99, 80.68, 152.78	-0.0 , 0.0 , 0.01 0.0 , 0.0 , 0.01 0.0 , 0.0 , 0.01 -0.0 , 0.0 , -0.0
e	51.96, 99.82, 115.51 29.50, 154.24, 160.19 29.34, 32.20, 89.85 62.64, 156.26, 62.29	54.58, 100.32, 117.31 34.88, 162.11, 163.1 29.07, 30.69, 90.5 64.2, 158.68, 65.79	-2.62 , -0.5 , -1.8 -5.38 , -7.87 , -2.91 0.27 , 1.51 , -0.65 -1.56 , -2.42 , -3.5

f	150.71, 51.23, 147.32 163.84, 70.85, 128.87 46.15, 45.48, 107.40 141.03, 127.87, 157.28	143.76, 53.98, 147.0 156.92, 75.68, 126.39 45.54, 41.45, 106.32 134.95, 133.17, 160.8	6.95 , -2.75 , 0.32 6.92 , -4.83 , 2.48 0.61 , 4.03 , 1.08 6.08 , -5.3 , -3.52
g	29.52, 89.14, 27.18 117.13, 59.02, 107.64 126.33, 77.28, 46.28 94.27, 112.12, 79.16	27.59, 87.65, 24.41 118.95, 58.78, 105.35 127.49, 76.3, 42.19 92.82, 111.23, 75.27	1.93 , 1.49 , 2.77 -1.82 , 0.24 , 2.29 -1.16 , 0.98 , 4.09 1.45 , 0.89 , 3.89
h	/	73.23, 60.57, 84.3 168.11, 154.33, 92.38 103.64, 167.45, 77.86 167.1, 97.26, 103.56	/
i	/	134.19, 42.07, 159.75 156.42, 97.43, 68.71 50.99, 156.22, 86.55 75.65, 144.64, 79.28	/
j	/	90.23, 23.04, 172.09 41.1, 112.35, 24.8 48.82, 154.35, 176.3 87.23, 41.1, 75.34	/

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. All the differences of the values for a, b, c, d are within reasonable range.

For dataset e, f, g, the introduction of both EM distortion and EM noise to the data may make the distortion correction function not as accurate as that for the data without noise because the EM noise is random and non-zero mean. Plus, the introduction of OT jiggle also effects the final output results. 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 distortion correction • Finished Step 4 to 6 of assignment 2 • Tested and debugged programs
Huixiang Li	• Collaborated with Tianyu on implementing the distortion correction • Finished Step 1 to 3 of assignment 2 • Tested and debugged programs

References

[1] G. Farin, Curves and surfaces for computer-aided geometric design, a practical guide, Academic Press, Boston, 1990, chapter 10 and pp 281-284. (P20 in lecture slide ‘Interpolation Review’)

[2] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. IEEE PAMI. Vol. 9. No. 5. Sept. 1987: 698-700.

[3] E. H. Moore. On the reciprocal of the general algebraic matrix. Bulletin of the American Mathematical Society, 26(9):394–395, 1920. doi:10.1090/.