Programming Assignments 1 & 2 601.455 and 601/655 Fall 2021

Please also indicate which section(s) you are in (one of each is OK)

Score Sheet

Name 1	Shuojue Yang
Email	syang131@jh.edu
Other contact information (optional)	
Name 2	Zijian Wu
Email	zwu52@jhu.edu
Other contact information (optional)	
Signature (required)	I (we) have followed the rules in completing this assignment Shuojue Yang Zijian Wu

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	

Shuojue Yang
Zijian Wu

Programming Assignment 2

Overview

Our task of this assignment is to implement the distortion calibration for the EM tracker system and to calculate the landmarks' coordinates relative to the CT coordinates system. The complete procedure is designed according to the suggested procedure in the stem.

1. According to Programming Assignment 1 Question 4, obtain the $\vec{C}_i[k]$ and $\vec{C}_i^{(expected)}[k]$ as the coordinates with distortion and the accurate coordinates respectively.

2. Use $\vec{C}_i[k]$ and $\vec{C}_i^{(expected)}[k]$ to fit the interpolation curve as the distortion correction function $\Gamma(\cdot)$.

3. According to Programming Assignment 1 Question 5, using dewarped $\vec{G}_i[k]$ (coorected by dewarping function Γ) as the input, implementate the EM pivot calibration to get $\vec{p}_{dimple}$ and $\vec{p}_{tip}$.

4. Calculate $\vec{B}_j$ in EM tracker system corresponding to $\vec{b}_j$ by utilizing $\vec{p}_{tip}$ and $\vec{G}_i^{dewarp}[k]$, which are obtained from the previous steps.

5. Calculate F_{reg} by using 3D to 3D point cloud registration package developed in Programming Assignment 1. The input data are point set $\vec{B}_j$ and $\vec{b}_j$.

6. First use distortion correction function to dewarp $\vec{G}_i$. Then use the dewarped $\vec{G}_i$ to repeat the EM pivot calibration to get $\vec{p}_{tip}$. According to $\vec{B}_i = F_T \cdot \vec{p}_{tip}$, we can get $\vec{B}_i$. Finally, we can calculate $\vec{b}_i$ by $\vec{b}_i = F_{reg} \cdot \vec{B}_{tip}$.

Method

A. **Load data** $\{C_i[k]\}$ from EM sensor and **compute the expected data** $\{C_i^{(expected)}[k]\}$ based on the optical tracker, where k denotes the k-th frame.

   1. Method overview
      Similar to Programming Assignment I, we have mathematics conclusions as following for each frame:

$$\vec{D}_i = F_D \vec{d}_i \tag{1}$$

$$\vec{A}_i = F_A \vec{a}_i \tag{2}$$

$$\vec{C}_i^{(expected)} = F_D^{-1} F_A \vec{d}_i \tag{3}$$

2. Algorithm and implementation
We have four point sets in this task, a_i corresponding to A_i and d_i corresponding to D_i. To obtain the Cartesian Coordinate Transformations (F_D and F_A) that satisfy Eq. (1) and Eq. (2), respectively. Then we can use the point-cloud-to-point-cloud registration method described in Programming Assignment I to calculate the transformation F_D and F_A. Finally, we can calculate the expected $C_i^{(expected)}$ coordinates relative to the EM system by Eq. (3) for each frame.

B. **Point-cloud-to-point-cloud registration algorithm** to compute the transformation between two coordinates.

1. Method overview

• Registration error
To tackle the registration task, two given point sets $\{a_i\}$ and $\{b_i\}$ are fed into the algorithm to compute the transformation minimizing the registration error between these two point sets. Typically, the transformations are represented by Cartesian Coordinate Transformation group and the goal can be formulated as following:

$$\underset{F=(R,\vec{p})}{arg\ min} \sum_{i}^{N} ||R\ \vec{a_i}\ +\ \vec{p}\ -\ \vec{b_i}||^2 \tag{4}$$

• Transferring to least-squares problem
To minimize the registration error between the two point sets, we transfer Eq. (4) into a least-squares problem. Specifically, $\{a_i\}$ and $\{b_i\}$ subtract their midpoints' coordinates and generate $\{\widetilde{a}_i\}$ and $\left\{\widetilde{b}_i\right\}$, respectively. Thus, the rotation and translation parts are decoupled and the minimization problem defined by Eq. (4) in is reduced in two subsequent problems:

$$\widehat{R}\ =\ \underset{R}{argmin} \sum_{i}^{N} ||R\ \widetilde{a}_i\ \ -\ \widetilde{b}_i\ ||^2 \tag{5}$$

$$\widehat{\vec{p}}\ =\ \widehat{R}\ \overline{a}\ -\ \overline{b} \tag{6}$$

where $\overline{a}$ and $\overline{b}$ denote the midpoint of $\{a_i\}$ and $\{a_i\}$, respectively. We use least-squares method to compute the $\widehat{R}$ and then further obtain the $\widehat{\vec{p}}$ based on the estimated rotation. In this assignment, we used quarternion based method [1] to directly solve Eq. (8)

C. **Dewarping calibration & repeat EM pivot calibration**

We develop a Dewarping Calibration utility library to rectify(dewarp) the distortion error in the EM sensor data. Then use this dewarped data to repeat EM pivot calibration according to the same method as Programming Assignment 1.

1. Method overview

• Dewarping function $\Gamma(\vec{u}; N)$
After obtaining scatter data pairs $\{C_i[k]\}$ and $\{C_i^{(expected)}[k]\}$, we intend to fit the dewarping function with Bernstein polynomials in 3D setting, which is shown as following:

$$\begin{aligned}\Gamma(\vec{u}; N) &= \sum_i^N \sum_j^N \sum_k^N c_{ijk} \cdot F_{ijk}(\vec{u}; N) \\ &= \sum_i^N \sum_j^N \sum_k^N c_{ijk} \cdot B_{i,N}(u_x) \cdot B_{j,N}(u_y) \cdot B_{k,N}(u_z) \end{aligned} \tag{7}$$

where Γ denote the dewarping function capable of rectifying the distortion error, N denote the highest order of polynomials, $\vec{u} = (u_x, u_y, u_z)$ is the normalized position of input (namely $||\vec{u}|| = 1$), $B_{i,N}(u) = \binom{N}{i} u^i (1-u)^{N-i}$ denotes the basic function constructing polynomials and $\{c_{ijk}\}$ is the coefficient set we intend to fit.

We normalize the input within $[0, 1]$ to maintain numerical stability. Specifically, we normalize the input by choosing the upper and lower limits from the bounding box of all the sensor data $\{C_i[k]\}$, as shown in Alg. **??**. With the dewarping function Γ, we can rectify new sensor data (without ground-truth) by interpolating and predicting their corresponding 'expected' positions.

• Transferring to least-squares problem
To fit the coefficients of $\Gamma(\vec{u}; N)$, we reformulate Eq. (7) as following:

$$\begin{bmatrix} & \vdots & \\ F_{000}(\vec{u}_s) & \cdots & F_{555}(\vec{u}_s) \\ & \vdots & \end{bmatrix} \cdot \begin{bmatrix} C_{000}^x & C_{000}^y & C_{000}^z \\ \vdots & \vdots & \vdots \\ C_{555}^x & C_{555}^y & C_{555}^z \end{bmatrix} \approx \begin{bmatrix} & \vdots & \\ p_s^x & p_s^y & p_s^z \\ & \vdots & \end{bmatrix} \tag{8}$$

Then we can exploit least-squares method to solve the coefficient matrix.

- Repeat EM pivot calibration using dewarped data

The EM pivot calibaration procesure is the same as the method in Programming Assignment 1. The main method is shown below.

We choose the first frame as the start frame and then establish a local coordinate system attached on the tool. We define the dewarped local and EM system coordinates at the k-th frame as $g_i^{dewarped}[0]$ and $G_i^{dewarped}[0]$, respectively. Marker coordinates relative to the tool coordinate system are defined as following:

$$g_i^{dewarped}[0] \; = \; G_i^{dewarped}[0] \; - \; \frac{1}{N_G}\sum_i^{N_G} G_i^{dewarped}[0] \tag{9}$$

where $g_i^{dewarped}[0]$ and $G_i^{dewarped}[0]$ respectively denote coordinates relative to the local and EM coordinate system in the first frame. The local coordinate system is attached on the tool, which means that the markers' local coordinates are fixed in each frame:

$$g_i^{dewarped}[k] \; = \; g_i^{dewarped}[0], \qquad k \; \in \; \{0,\; 1,\; \ldots\; ,\; N_G\;\} \tag{10}$$

Driven by this observation, the transformation registering the $g_i[0]$ to $G_i[k]$ is the $F_G[k]$, which gives the position and orientation of the pointer body at frame k with respect to tracker coordinate system. Thus by using the registration package developed in Sec.B., we obtain a sequence of rigid transformations $\{F_G[0],\; F_G[1],\; F_G[2],\; ...,\; F_G[N_{Frame}]\}$. Specifically, we can formulate the problem into computing a transformation minimizing the registration error as following:

$$F_G[k] \; = \; \underset{F=(R,\; \vec{p})}{argmin} \sum_i^{N_G} ||R \; \vec{g}_i^{\,dewarped}[0] \; + \vec{p} \; - \; \vec{G}_i^{\,dewarped}[k]||^2 \tag{11}$$

Finally, using the pivot calibration package developed in Programming Assignment 1 Sec.C. , we get the pivot vector relative to the EM tracker system.

2. Algorithm and implementation

We implemented a *DistortionCalib* class in $CalibrationDistortion.txt$ integrating both $.fit()$ and $.predict()$ methods. The former requires the sensor data and expected data (ground-truth) as input to fit the dewarping function, while the latter interpolates the 'unseen' sensor data and output the corrected values by the fitted dewarping function.

D. **Calculating** $\vec{B}_i$.

1. Method overview

In this step, we first calculate F_T, which is the rigid transformation between local system and EM tracker system. The method of F_T calculation is the same as Programming Assignment 1.

We choose the first frame as the start frame and then establish a local coordinate system attached on the tool. We define the local and EM system coordinates at the k-th frame as $g_i[k]$ and $G_i[k]$, respectively. Marker coordinates relative to the tool coordinate system are defined as following:

$$g_i[0] \; = \; G_i[0] \; - \; \frac{1}{N_G} \sum_{i}^{N_G} G_i[0] \tag{12}$$

where $g_i[0]$ and $G_i[0]$ respectively denote coordinates relative to the local and EM coordinate system in the first frame. The local coordinate system is attached on the tool, which means that the markers' local coordinates are fixed in each frame:

$$g_i[k] \; = \; g_i[0], \qquad k \; \in \; \{0, \; 1, \; \dots \; , \; N_{Frame} \; \} \tag{13}$$

Using the distortion correction function developed in Sec.B., we can obtain $\vec{G}_i^{dewarp}[k]$ corresponding to $g_i[k]$. Thereby we can use 3D to 3D point registration to compute the rigid transformation between these two point sets, which is F_T. According to Eq. (14), we can obtain the $\vec{B}_i$ coordinates.

$$\vec{B}_i \; = \; F_T \cdot \vec{p}_{tip} \tag{14}$$

In practice, both F_T and $\vec{p}_{tip}$ are returned from our EM pivot calibration program. So the input of this step are F_T and $\vec{p}_{tip}$, and the output is $\vec{B}_i$.

E. **Calculating** F_{reg}.

1. Method overview

After Sec.D., we have point sets $\{\vec{B}_i\}$ and $\{\vec{b}_i\}$ relative to EM tracker system and CT system, respectively. We use 3D to 3D point cloud registration to calculate the rigid body transformation from $\{\vec{B}_i\}$ to $\{\vec{b}_i\}$.

F. **Compute the tip location with respect to the CT image**.

1. Method overview

   To solve this section, the results in previous steps will be utilized. First, using distortion correction function developed in Sec.B., we can eliminate the error of $G_i[k]$ and get $\vec{G}_i^{dewarp}[k]$. The next step is that use the dewarped data to implement EM pivot calibration. The results of this step is $\vec{p}_{tip}$ and F_T. Then According to Eq. (14), we can compute the $\{\vec{B}_i\}$. So we have point sets $\{\vec{B}_i\}$ and $\{\vec{b}_i\}$ relative to EM tracker system and CT image system respectively. Using the 3D to 3D point cloud registration function developed in Programming Assignment 1, we can calculate the rigid body transformation F_{reg} between these two point sets. Finally, we calculate the $\vec{b}_i$ according to $\vec{b}_i = F_{reg} \cdot \vec{B}_i$.

Results and discussion

Table 1: Test results for different debug cases. L2 norm is adopted to measure the error

case	dewarp		without dewarp		$\vec{C}_i^{expected}$
	pivot_error	nav_error	pivot_error	nav_error	
a	0.005911069	0.007506893	0.003811838	0.006790483	0.004791565
b	0.43506735	0.231559227	0.17754853	0.044381405	0.47659153
c	0.041791033	0.018583462	2.528303894	0.760963294	0.4516165
d	0.004537099	0.007884271	0.004320853	0.007956534	0.011181919
e	0.67647187	0.399857139	11.77350029	4.528161939	1.4981236
f	0.236272339	0.257372503	2.707689392	2.471776849	1.7688808

Table 2: The corresponding positions of the probe tip in CT coordinates in unknown cases.

	case g	case h	case i	case j
frame 0	[111.83,75.80,148.57]	[116.59,63.60,146.86]	[63.72,91.55,109.05]	[143.08,113.92,59.87]
frame 1	[85.57,143.10,75.26]	[42.09,164.83,72.86]	[106.41,165.70,42.28]	[96.95,34.05,114.62]
frame 2	[76.57,47.66,92.45]	[66.43,107.56,61.55]	[73.82,145.79,122.50]	[133.89,41.20,109.10]
frame 3	[65.88,80.80,26.40]	[41.37,140.76,26.81]	[50.08,87.74,88.83]	[126.29,46.91,117.81]

1. Accuracy evaluation method

In this assignment, we adopt L2-Norm to evaluate the implemented programs. Since the final outputs of these modules are position vectors in 3D Euclidean space, it is reasonable and intuitive that Euclidean Distance between predictions and ground-truths can serve as error and evaluate the performance. Driven by this, we compute the average L2-Norm for all the output position vectors in each case.

2. The effectiveness of Distortion Calibration

In this section, we verify the effectiveness of the made distortion calibration on the simulated dataset and 'debug' dataset downloaded from the web page.

1) Simulated dataset
We simulate a point sets with random positions in a 3D space as the true positions serving as ground-truth. To simulate the nonlinear distortion error, we adopt the exponential function to construct the error function shown as:

$$\widetilde{p} = p + \lambda e^{\alpha x} e^{\beta y} e^{\gamma z} \tag{15}$$

where p denotes one certain dimension of (x,y,z) and $\widetilde{p}$ denotes the simulated noisy data. We first generate 800 points serving as the 'fitting data' with known ground-truth to fit the dewarping function, then we generate 200 points with the addition of the simulated distortion errors. To verify the effectiveness of our implemented module, we use the fitted function to rectify the test data, which showing a great improvement. Before dewarping, the average error is 4.098 while it reduces to 0.0306 after dewarping correction.

2) Debug dataset
We also validate the effectiveness of the distortion calibration over the data given by debug files. We first investigate its improvements over *EM pivot calibration* task. As shown in the Tabel 1, it can be observed that in cases c, e, f, the average pivot calibration error decreases significantly after distortion correction. Next, the validation of the distortion calibration can be reflected in the results of *nav_error*. According to Table 1, for cases c, e, f, *nav_error* demonstrate an obvious reduction.

3. Discussion about the failure cases a, b, d

In cases a, b, c, both *pivot_error* and *nav_error* do not drop apparently, even increase. According to the Summary of Problem Data Set Creation, we know that cases a and d do not contain EM distortion and case b does. This can explain why the gap between *pivot_error* and *nav_error* in cases a and d are very small. Meanwhile, we find that case b only adds EM noise but does not add EM distortion. In the distortion correction step, the noise will be regarded as the distortion which will cause an incorrect interpolation function fit. This is the reason for large gap between *pivot_error* and *nav_error* in case b.

When we appropriately reduce the order of the interpolation polynomial, we can observed that *pivot_error* and *nav_error* in dewarp status reduce.

4. Error Analysis

   Despite that the distortion error has been mitigated to variant extents with dewarping function introduced, the residual error might be attributed to following reasons: 1) the error brought by optical trackers. According to Table 1, $\vec{C}_i^{expected}$ suffers from noises and thus may limit the effectiveness of dewarping function. 2) the noise of EM measurement make the fitting function with large model complexity (i.e. orders) prone to overfit. 3) Different from previous work adopting a local bounding box to constrain the upper and lower limits of the dewarping function based on Bernstein curve, we 'globally' choose all the sensor data C as the bounding box, which could impose a negative effect on fitting the dewarping function due to local details smoothed.

Statement of teamwork

Zijian Wu developed the EM pivot calibration, expected C computation and navigation parts, and Shuojue Yang developed the distortion calibration, navigation and F_{reg} computation parts of this assignment.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.