Report

Monday, April 24, 2023   2:48 PM

The University of Texas at Austin
Mechanical Engineering
Cockrell School of Engineering

ME397: Algorithms for Sensor-Based Robotics (ASBR)
Prof. Farshid Alambeigi, Spring 2023

Homework/Programming Assignment #3 **Homework Due: 04/18/2023- 5:00PM**

Bhakat Mathur (BM39349)   Carl [illegible] CMS8498

bmathur@utexas.edu   cms8498@my.utexas.edu

Name/EID: **Name/EID:**
Email: **Email:**

Signature (*required*)
I/We have followed the rules in completing this Assignment.

Signature (*required*)
I/We have followed the rules in completing this Assignment.

Question	Points	Total
PA1	80	
PA2.1	20	
PA2. 2 (Bonus)	10	

Instruction:

1. Remember that this is a graded assignment. It is the equivalent of a **take-home exam**.
2. **For PA questions**, you need to write a report showing how you derived your equations, describes your approach, test functions, and discusses the results. You should show your test results for each function.

3. You are to work **alone** or **in teams of two** and are **not to discuss the problems with anyone** other than the TAs or the instructor.
4. It is open book, notes, and web. But you should cite any references you consult.
5. Unless I say otherwise in class, it is due before the start of class on the due date mentioned in the P/H Assignment.
6. **Sign and append** this score sheet as the first sheet of your assignment.
7. Remember to submit your assignment in the Canvas.

PA 1:

Goal 1

This is implemented in the file least_squares_registration.m.

The registration problem requires us to find a transform between two different point clouds (after bringing them in the same frame).

For example, we have this 2D object with optical fiducials (cirlces) on it. We know the location of the fiducials in a local coordinate frame. Let's say this frame is located at the centroid of the object.
Now, when we look at these fiducials from an optical tracking system, the tracker gives us position (x, y, z) of each fiducial in its frame of reference. Let's say we already know the correspondences between the fiducials in our CAD model and readings from the NDI optical tracker. Thus, the registration problem is to find a transform between the point cloud collected from the tracker and the one obtained from the CAD drawing. Assuming the object is rigid, this gives us its pose in optical tracker frame.

We use Arun's method [1] to solve this problem.

The method finds the centroid of both the point clouds and calculates vectors from these centroids to each fiducials. These are used to build an "H" matrix.
Taking SVD of H,

R = V * transpose(U);
p = transpose(B_centroid) - R * transpose(A_centroid);

T = [R p; 0 0 0 1] ;

If the determinant of the initial R is -1, it means that the points are coplanar and this R is actually a reflection; not a rotation. Thus, R is calculated such that, last column of the V matrix is made negative and

R = V*transpose(U);

Goal 2

This is implemented in the file pivotCalibration.m.

The goal of pivot calibration is to find the translation vector from a fiducial tracking body to the tip of a probe by pivoting it at a point (divot) and moving in a sphere to collect data.

The process starts by registering the fiducial tracking body. This gives us transformation matrix, Tk for each measurement.

For each **measurement k** , we have known $\mathbf{R}_k$ and $\mathbf{p}_k$:

$$\boxed{\vec{\mathbf{b}}_{post}} = \boxed{\mathbf{R}_k}\boxed{\vec{\mathbf{b}}_{tip}} + \boxed{\vec{\mathbf{p}}_k}$$

We can rewrite this equation as:

$$\boxed{\mathbf{R}_k}\boxed{\vec{\mathbf{b}}_{tip}} - \boxed{\vec{\mathbf{b}}_{post}} = \boxed{-\vec{\mathbf{p}}_k}$$

The Rk and pk from all the measurements are stacked to form the equation in the form:

So, we can **set up a least squares problem** in the form of **Ax=b** as follows and find the **unknowns** $\vec{\mathbf{b}}_{tip}$ and $\vec{\mathbf{b}}_{post}$:

$$\begin{bmatrix} \vdots & \vdots \\ \mathbf{R}_k & -\mathbf{I} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vec{\mathbf{b}}_{tip} \\ \vec{\mathbf{b}}_{post} \end{bmatrix} \cong \begin{bmatrix} \vdots \\ -\vec{\mathbf{p}}_k \\ \vdots \end{bmatrix}$$

This returns b_tip (translation vector from fiducial tracking body to probe tip) and b_post (translation vector from tracking system to divot).
P

Goal 3. a

This is implemented in the file solutions/problem_3_a.m on data files pa1-debug-a-calreadings.txt and pa1-debug-a-calbody.txt .

The results for each frame are as follows:

```
F_D(:,:,1) =

                  1     1.66533453693773e-16     5.55111512312578e-17                        0
    2.90362741811646e-16                        1                        0     -1.4210854715202e-14
    4.27041655759008e-17    -1.11022302462516e-16                        1                    -1500
                       0                        0                        0                        1


F_D(:,:,2) =

                  1     1.66533453693773e-16     5.55111512312578e-17                        0
    2.90362741811646e-16                        1                        0     -1.4210854715202e-14
    4.27041655759008e-17    -1.11022302462516e-16                        1                    -1500
                       0                        0                        0                        1


F_D(:,:,3) =

                  1     1.66533453693773e-16     5.55111512312578e-17                        0
    2.90362741811646e-16                        1                        0     -1.4210854715202e-14
    4.27041655759008e-17    -1.11022302462516e-16                        1                    -1500
                       0                        0                        0                        1


F_D(:,:,4) =

                  1     1.66533453693773e-16     5.55111512312578e-17                        0
    2.90362741811646e-16                        1                        0     -1.4210854715202e-14
```

4.27041655759008e-17	-1.11022302462516e-16	1	-1500
0	0	0	1

F_D(:,:,5) =

1	1.66533453693773e-16	5.55111512312578e-17	0
2.90362741811646e-16	1	0	-1.4210854715202e-14
4.27041655759008e-17	-1.11022302462516e-16	1	-1500
0	0	0	1

F_D(:,:,6) =

1	1.66533453693773e-16	5.55111512312578e-17	0
2.90362741811646e-16	1	0	-1.4210854715202e-14
4.27041655759008e-17	-1.11022302462516e-16	1	-1500
0	0	0	1

F_D(:,:,7) =

1	1.66533453693773e-16	5.55111512312578e-17	0
2.90362741811646e-16	1	0	-1.4210854715202e-14
4.27041655759008e-17	-1.11022302462516e-16	1	-1500
0	0	0	1

F_D(:,:,8) =

1	1.66533453693773e-16	5.55111512312578e-17	0
2.90362741811646e-16	1	0	-1.4210854715202e-14
4.27041655759008e-17	-1.11022302462516e-16	1	-1500
0	0	0	1

Goal 3. b

This is implemented in the file solutions/problem_3_b.m on data files pa1-debug-a-calreadings.txt and pa1-debug-a-calbody.txt .

The results for each frame are as follows:

F_A(:,:,1) =

-0.333333333333334	0.666666666666667	0.666666666666667	202.63
0.666666666666667	-0.333333333333333	0.666666666666667	211.9075
0.666666666666667	0.666666666666667	-0.333333333333333	-1286.7125
0	0	0	1

F_A(:,:,2) =

-0.333333333333333	0.666666666666667	0.666666666666667	205.38
0.666666666666667	-0.333333333333333	0.666666666666667	214.64625
0.666666666666667	0.666666666666667	-0.333333333333333	-1050.625
0	0	0	1

F_A(:,:,3) =

-0.333333333333333	0.666666666666667	0.666666666666667	207.865
0.666666666666667	-0.333333333333333	0.666666666666667	449.92125
0.666666666666667	0.666666666666667	-0.333333333333333	-1288.425
0	0	0	1

F_A(:,:,4) =

-0.333333333333333	0.666666666666667	0.666666666666667	214.81
0.666666666666667	-0.333333333333333	0.666666666666667	446.9975
0.666666666666667	0.666666666666667	-0.333333333333333	-1053.635
0	0	0	1

F_A(:,:,5) =

-0.333333333333333	0.666666666666667	0.666666666666667	447.33
0.666666666666667	-0.333333333333333	0.666666666666667	212.64625

0.666666666666667	0.666666666666667	-0.333333333333333	-1287.25625
0	0	0	1

F_A(:,:,6) =

1	0	0	450.975
1.17425795290194e-16	1	5.55111512312578e-17	206.14875
-6.19146440589364e-17	-3.33066907387547e-16	1	-1043.685
0	0	0	1

F_A(:,:,7) =

-0.333333333333334	0.666666666666667	0.666666666666667	449.455
0.666666666666667	-0.333333333333333	0.666666666666667	451.29
0.666666666666667	0.666666666666667	-0.333333333333333	-1288.245
0	0	0	1

F_A(:,:,8) =

-0.333333333333334	0.666666666666667	0.666666666666667	450.925
0.666666666666667	-0.333333333333333	0.666666666666667	447.9375
0.666666666666667	0.666666666666667	-0.333333333333333	-1049.53
0	0	0	1

\>>

Goal 3. c

This is implemented in the file solutions/problem_3_c.m on data files pa1-debug-a-calreadings.txt and pa1-debug-a-calbody.txt .
The results are in outputs/output_3_c.txt

Lines 2 and 3 on the output file are set to [0,0,0] as pivot calibration was not performed as part of this goal.

Goal 4

This is implemented in the file solutions/problem_3_c.m on data files pa1-debug-b-empivot.txt and pa1-debug-b-optpivot.txt .

The results are:

b_tip =

```
-257.167857142857
-213.44
-337.44880952381
```

b_post =

```
-21.7378571428571
0
-39.4288095238096
```

PA 2:

Method used for hand eye calibration described on week 12 lecture 1 slideshow. Essentially we are creating a linear fit of the dataset using regression to find the rotational and translational component of X. The Ax=xB problem is solved using the quaternion method described in week 12 lecture 1.

Results for X using full set of clean data in dataset A

```
X =

   -0.2136    0.9769         0    0.0760
   -0.9769   -0.2136         0   -0.0482
         0         0    1.0000    0.0085
         0         0         0    1.0000
```

Results for X using full set of noisy data:

```
X =

   -0.2137    0.9769         0    0.0758
   -0.9769   -0.2137         0   -0.0484
         0         0    1.0000    0.0083
         0         0         0    1.0000
```

Notice how both the clean data and noisy data give a very similar solution.

Results for X using the first half the noise free data set

```
X =

   -0.1312    0.9695   -0.2072    0.4941
   -0.3872   -0.2425   -0.8896    0.4483
   -0.9126   -0.0365    0.4072    0.2884
         0         0         0    1.0000
```

Results for X using every other data point in the noise free data set (1,3,5,7,9)

```
X =

    0.0523    0.0244   -0.9983    0.4992
   -0.8080    0.5885   -0.0280   -0.0647
    0.5869    0.8081    0.0505   -0.3127
         0         0         0    1.0000
```

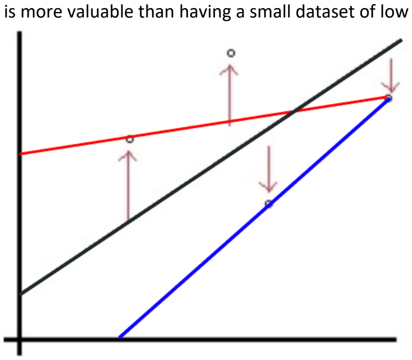
Results for X using only the first and last data point of the data set with no noise:

```
X =

                   1                   0                   0   -0.0903309439891085
                   0                  -1                   0                     0
                   0                   0                  -1    0.0308021105264186
                   0                   0                   0                     1
```

Discussion on why using a half set of noise free data gives such a wildly different solution:

It's linear regression, so in order to find the an accurate linear fit of the system, having more data points is more valuable than having a small dataset of low noise data. Below is an example:



The blue line is a fitting of the last 2 points, and the red line is a fitting of the first and last point. Even though both of these lines fall precisely along the data points that are being used to fit the line, both of these fits are wrong. The most accurate linear representation of this system of 4 points is the middle black line which is a fit of all 4 points. This example shows that if when using linear regression, more data points is more important then quiet (not noisy) data.

References

- Arun, K.S. & Huang, T.S. & Blostein, Steven. (1987). Least-squares fitting of two 3-D point sets. IEEE T Pattern Anal. Pattern Analysis and Machine Intelligence, IEEE Transactions on. PAMI-9. 698 - 700. 10.1109/TPAMI.1987.4767965.
- https://www.youtube.com/watch?v=EokL7E6o1AE