16-811 Assignment 1

Emma Benjaminson

12 September 2019

1 Problem 1

Please see code contained in folder code/q1.m. The implementation was tested on the following matrices. The results are listed after the test matrix:

Test matrix:

$$A_1 = \begin{pmatrix} 10 & 9 & 2 \\ 5 & 3 & 1 \\ 2 & 2 & 2 \end{pmatrix} \tag{1}$$

Code output for matrix A_1 (taken from Problem 2):

$$L_{A_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.2 & -0.133 & 1 \end{pmatrix} \tag{2}$$

$$D_{A_1} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & -1.5 & 0 \\ 0 & 0 & 1.6 \end{pmatrix} \tag{3}$$

$$U_{A_1} = \begin{pmatrix} 1 & 9 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$A_1' = \begin{pmatrix} 10 & 9 & 2 \\ 0 & -1.5 & 0 \\ 0 & 0 & 1.6 \end{pmatrix} \tag{5}$$

Test matrix:

$$A_3 = \begin{pmatrix} 10 & 6 & 4 \\ 5 & 3 & 2 \\ 1 & 1 & 0 \end{pmatrix} \tag{6}$$

Code output for matrix A_3 (taken from Problem 2):

$$L_{A_3} = \begin{pmatrix} 1 & 0 & 0 \\ 0.1 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix} \tag{7}$$

$$D_{A_3} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{8}$$

$$U_{A_3} = \begin{pmatrix} 1 & 6 & 4 \\ 0 & 1 & -0.4 \\ 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$A'_3 = \begin{pmatrix} 10 & 6 & 4 \\ 0 & 0.4 & -0.4 \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

2 Problem 2

I calculated the $PA = LDU$ decompositions by hand as shown below. I used the built-in MATLAB function SVD() to calculate the singlular value decomposition matrices. I have provided the output of the code here. The code itself is contained in folder code/q2.m.

2.1 Matrix A_1

2.1.1 LDU Decomposition for A_1

$$A_1 = \begin{pmatrix} 10 & 9 & 2 \\ 5 & 3 & 1 \\ 2 & 2 & 2 \end{pmatrix} \tag{11}$$

L	A'	Operation
$\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 9 & 2 \\ 0 & -1.5 & 0 \\ 0 & 0.2 & 1.6 \end{pmatrix}$	row 2' = row 2 - 0.5*row 1 row 3' = row 3 - 0.2*row 1
$\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.2 & -0.13 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 9 & 2 \\ 0 & -1.5 & 0 \\ 0 & 0 & 1.6 \end{pmatrix}$	row 3" = row 3' - (-0.13)*row 2'

Answer:

$$L_{A_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.2 & -0.133 & 1 \end{pmatrix} \tag{12}$$

$$D_{A_1} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & -1.5 & 0 \\ 0 & 0 & 1.6 \end{pmatrix} \tag{13}$$

$$U_{A_1} = \begin{pmatrix} 1 & 9 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

$$A_1' = \begin{pmatrix} 10 & 9 & 2 \\ 0 & -1.5 & 0 \\ 0 & 0 & 1.6 \end{pmatrix} \tag{15}$$

2.1.2 SVD Decomposition for A_1

$$U_{A_1} = \begin{pmatrix} -0.8991 & 0.1788 & 0.3997 \\ -0.3861 & 0.1066 & -0.9163 \\ -0.2064 & -0.9781 & -0.0268 \end{pmatrix} \tag{16}$$

$$S_{A_1} = \begin{pmatrix} 15.1186 & 0 & 0 \\ 0 & 1.5362 & 0 \\ 0 & 0 & 1.0334 \end{pmatrix} \tag{17}$$

$$V_{A_1} = \begin{pmatrix} -0.7497 & 0.2376 & -0.6177 \\ -0.6391 & -0.0176 & 0.7689 \\ -0.1718 & -0.9712 & -0.1651 \end{pmatrix} \tag{18}$$

2.2 Matrix A_2

2.2.1 LDU Decomposition for A_2

$$A_2 = \begin{pmatrix} 16 & 16 & 0 & 0 \\ 4 & 0 & -2 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{19}$$

L	A'	Operation
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.25 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 16 & 0 & 0 \\ 0 & -4 & -2 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	row 2' = row 2 - 0.25*row 1
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.25 & 1 & 0 & 0 & 0 \\ 0 & -0.25 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 16 & 0 & 0 \\ 0 & -4 & -2 & 0 \\ 0 & 0 & -1.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	row 3' = row 3 - (-0.25)*row 2'
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.25 & 1 & 0 & 0 & 0 \\ 0 & -0.25 & 1 & 0 & 0 \\ 0 & 0 & -0.667 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 16 & 0 & 0 \\ 0 & -4 & -2 & 0 \\ 0 & 0 & -1.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	NOTE: swapped rows 4 and 5, see permutation matrix P below. row 4' = row 4 - (-0.667)*row 3'
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.25 & 1 & 0 & 0 & 0 \\ 0 & -0.25 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 16 & 0 & 0 \\ 0 & -4 & -2 & 0 \\ 0 & 0 & -1.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	row 5' = row 5 - row 4'

Answer:

$$L_{A_2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.25 & 1 & 0 & 0 & 0 \\ 0 & -0.25 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{20}$$

$$D_{A_2} = \begin{pmatrix} 16 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & -1.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{21}$$

$$U_{A_2} = \begin{pmatrix} 1 & 16 & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

$$A'_2 = \begin{pmatrix} 16 & 16 & 0 & 0 \\ 0 & -4 & -2 & 0 \\ 0 & 0 & -1.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{23}$$

$$P_{A_2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{24}$$

2.2.2 SVD Decomposition for A_2

$$U_{A_2} = \begin{pmatrix} -0.9914 & 0.125 & -0.003 & 0.0268 & -0.0265 \\ -0.1269 & -0.9728 & 0.1589 & -0.0326 & 0.106 \\ -0.0311 & -0.0136 & -0.5399 & -0.7264 & 0.4239 \\ 0 & 0.0173 & 0.3936 & -0.6638 & -0.6358 \\ 0.0005 & 0.1937 & 0.7269 & -0.1729 & 0.6358 \end{pmatrix} \tag{25}$$

$$S_{A_2} = \begin{pmatrix} 22.8186 & 0 & 0 & 0 \\ 0 & 3.4932 & 0 & 0 \\ 0 & 0 & 1.6873 & 0 \\ 0 & 0 & 0 & 1.1227 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{26}$$

$$V_{A_2} = \begin{pmatrix} -0.7174 & -0.5414 & 0.3482 & 0.2664 \\ -0.6965 & 0.5687 & -0.3485 & -0.2645 \\ 0.0125 & 0.6163 & 0.5624 & 0.5511 \\ 0 & 0.0604 & 0.6641 & -0.7452 \end{pmatrix} \tag{27}$$

2.3 Matrix A_3

2.3.1 LDU Decomposition for A_3

$$A_3 = \begin{pmatrix} 10 & 6 & 4 \\ 5 & 3 & 2 \\ 1 & 1 & 0 \end{pmatrix} \tag{28}$$

L	A'	Operation
$\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 6 & 4 \\ 0 & 0 & 0 \\ 0 & 0.4 & -0.4 \end{pmatrix}$	row 2' = row 2 - 0.5*row 1 row 3' = row 3 - 0.1*row 1 NOTE: We swap rows 2 and 3 after this step, see permutation matrix P at end.
$\begin{pmatrix} 1 & 0 & 0 \\ 0.1 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 6 & 4 \\ 0 & 0.4 & -0.4 \\ 0 & 0 & 0 \end{pmatrix}$	

Answer:

$$L_{A_3} = \begin{pmatrix} 1 & 0 & 0 \\ 0.1 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix} \tag{29}$$

$$D_{A_3} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{30}$$

$$U_{A_3} = \begin{pmatrix} 1 & 6 & 4 \\ 0 & 1 & -0.4 \\ 0 & 0 & 1 \end{pmatrix} \tag{31}$$

$$P_{A_3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{32}$$

$$A_3' = \begin{pmatrix} 10 & 6 & 4 \\ 0 & 0.4 & -0.4 \\ 0 & 0 & 0 \end{pmatrix} \tag{33}$$

2.3.2 SVD Decomposition for A_3

$$U_{A_3} = \begin{pmatrix} -0.8905 & -0.084 & -0.4472 \\ -0.4452 & -0.042 & 0.8944 \\ -0.0939 & 0.9956 & 0 \end{pmatrix} \tag{34}$$

$$S_{A_3} = \begin{pmatrix} 13.8451 & 0 & 0 \\ 0 & 0.5595 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{35}$$

$$V_{A_3} = \begin{pmatrix} -0.8107 & -0.0967 & 0.5774 \\ -0.4892 & 0.6538 & -0.5774 \\ -0.3216 & -0.7505 & -0.5774 \end{pmatrix} \tag{36}$$

3 Problem 3

Please see code contained in folder code/q3.m.

3.1 Part (a)

This matrix is invertible and has a trivial null space as shown in the code. This means that Singular Value Decomposition should return one exact solution. I calculate:

$$x = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} \tag{37}$$

3.2 Part (b)

This matrix is singular (non-invertible) and has a non-trivial null-space. I can say that the column space is spanned by the vectors:

$$v_1 = \begin{pmatrix} 1 \\ 0.5 \\ 0 \end{pmatrix} \tag{38}$$

and:

$$v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{39}$$

Knowing this, and looking at the vector b, I can see that the vector b is in the column space of the matrix so there is one unique solution in this case. I calculate x to be:

$$x = \begin{pmatrix} 0.33 \\ -1.333 \\ 1.667 \end{pmatrix} \tag{40}$$

3.3 Part (c)

This part has the same matrix as before but this time, the vector b is not in the column space of the matrix. Therefore, the best I can do is find the least squares solution for x, which is a projection of the vector b' onto the column space. This is why I find the same solution as for Part (c) - the vector b' is not in the column space of the matrix A so the closest I can get is a projection onto the column space which, by coincidence, has the same value as the vector x in Part (b).

$$x = \begin{pmatrix} 0.33 \\ -1.333 \\ 1.667 \end{pmatrix} \tag{41}$$

4 Problem 4

Please note: I had help for this problem from the TA's (thank you!) and another student, Sophie (last name unknown).

4.1 Part (a)

The matrix A removes the component of a vector, v, in the direction of the unit vector u. We can show this mathematically:

$$Av = (I - uu^T)v = v - uu^Tv \tag{42}$$

Here we can say that $u^T v$ is the projection of v onto the vector u. It results in a scalar which is the magnitude of the projection of v onto u. This value, $u^T v$, is multiplied by the vector u, and subtracted from v. As a result, we are left with the vector v with all of its components intact except for the component in the u direction.

4.2 Part (b)

The expression for finding eigenvalues is:

$$Av = \lambda v \tag{43}$$

If we know that Av results in the same vector v, without a component in the u direction, then the eigenvalues λ are equal to 1 for all rows except the row corresponding to the direction u, which is 0. Therefore, A has n-1 repeated eigenvalues equal to 1.

4.3 Part (c)

The rank of A (i.e. the column space of A) and the null space of A should sum to n. Since we already know that A negates the component of v in the direction of u and keeps all the other components, we know that the size of the null space of A is equal to 1. And therefore we also know the column space (or rank) of A is of size $n - 1$.

4.4 Part (d)

We can solve for A^2 as follows:

$$A^2 = (I - uu^T)(I - uu^T) = I - 2uu^T + (uu^T)^2 \tag{44}$$

Since u is a unit vector, we can say that uu^T is the projection of u onto itself, which is therefore a scalar of magnitude 1 in the direction of u, which I will denote as $\hat{u}$. The equation becomes:

$$A^2 = I - 2(1)\hat{u} + (1)^2\hat{u} = I - 1\hat{u} \tag{45}$$

The identity I must have the same size as A, $n \times n$, so if we subtract $1\hat{u}$, we are left with:

$$A^2 = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ . & & & & \\ . & & & & \\ . & & & & \end{pmatrix} \tag{46}$$

If we were to pass a vector v through A^2, we would find that it went through the same operation as described by matrix A. Therefore, I can conclude that:

$$A^2 = A \tag{47}$$

5 Problem 5

Please note that I referred to these references to help in solving this problem:

(1) Ho, Nghia. "Finding Optimal Rotation and Translation Between Corresponding 3D Points." `http://nghiaho.com/?page_id=671`

(2) Arun, K.S., Huang, T.S., Blostein, S.D. "Least-Squares Fitting of Two 3-D Point Sets." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 5, September 1987.

At first, I wanted to directly apply $Ax = b$ to this problem, where x was the original point cloud, consisting of a minimum of 3 points as indicated in the problem statement, and b was the same point cloud after the transformation A was applied. I could not understand why this problem could not simply be solved as $A = bx^{-1}$. I did not see where SVD was required in this solution. I think the failure in my thinking was that I was assuming that I could use geometric mechanics (similar to those described by Murray, Li and Sastry in "A Mathematical Introduction to Robotic Manipulation") to pull out the rotation and translation components from the matrix A, but I did not go far enough to try to develop a mathematical explanation for whether or not that would work (I'm still not sure of the answer). But I can guess that it will not require SVD to prove, so I looked elsewhere.

I am going to try to summarize what I have learned from the sources listed above in words here. The implementation in MATLAB can be found in code/q5.m.

The problem asks us to find matrices to describe the rotation and translation of a rigid body given two sets of point cloud data, collected before and after the transformation (rotation + translation) takes place. The problem indicates that we should use SVD to obtain a solution.

Let's call the original point cloud $A = a_i$ and the transformed point cloud $B = b_i$. They are both a collection of i points, where $i \geq 3$. (We need at least 3 points because we are working in 3D space.) I can write the points b_i as the outcome of a rotation and translation of the points a_i like so:

$$b_i = Ra_i + T \tag{48}$$

Where R is the rotation and T is the translation. I can rearrange this equation to say that my objective is to find matrices R and T that produce the smallest possible difference between point clouds a_i and b_i. That is, I want to find matrices R and T so that they move the point cloud a_i as close as possible to point cloud b_i:

$$b_i - (Ra_i + T) = 0 \tag{49}$$

And in the real world, and with $i \geq 3$ points, I may not be able to find an exact solution, so I want to find the minimum solution, or the least squares solution, Σ^2:

$$\Sigma^2 = ||b_i - (Ra_i + T)||^2 \tag{50}$$

What I am going to try to do here is use this least-squares expression to find first the matrix R, then T, using SVD. The steps are outlined as follows:

1. Calculate the centroid of both point clouds.
2. Find the rotation that minimizes the least-squares difference between those point clouds.
3. Then find the translation that minimizes the remaining difference that wasn't accounted for by the rotation matrix.

5.1 Calculating the Centroid

This step is relatively simple. We can define the centroid as the average location of all the points, like so:

$$A_{centroid} = \frac{1}{N}\sum_{i=1}^{N} a_i \tag{51}$$

The same holds true for the points b_i.

5.2 Finding the Rotation Matrix

Notice that in Eq 50 above, we are trying to minimize both the rotation and the translation simultaneously. If we are going to solve just for the rotation first, we need to assume that the two point clouds have the same point of rotation in space (i.e. that there is no translation occurring). We can encode this mathematically by assuming that the centroids of the 2 point clouds are at the same point in space. Now we can write a vector for each point in the point cloud that describes its position with respect to the centroid as such:

$$v_{a_i} = a_i - A_{centroid} \tag{52}$$

The same is true for point cloud b_i. Now that I can describe all the points as vectors with respect to a common origin (or point of rotation), so I can apply the same logic as in Eq 50 to find the rotation matrix that rotates point cloud a_i to b_i:

$$\Sigma^2 = \sum_{i=1}^{N} ||v_{b_i} - Rv_{a_i}||^2 \tag{53}$$

This equation is simply stating that the difference between rotation applied to vectors v_{a_i} (with respect to the common origin) and the vectors v_{b_i}, representing the same body centered on the same origin after rotation, should be minimized. The closer the difference is to zero, the more accurately the matrix R has rotated v_{a_i} onto v_{b_i}.

Now here comes the trick to this whole problem. Let's expand Eq 53 out:

$$\Sigma^2 = \sum_{i=1}^{N} ||v_{b_i} - Rv_{a_i}||^2 = \sum_{i=1}^{N} (v_{b_i} - Rv_{a_i})^T (v_{b_i} - Rv_{a_i}) \tag{54}$$

$$\Sigma^2 = \sum_{i=1}^{N} \left(v_{b_i}^T v_{b_i} - v_{b_i}^T R v_{a_i} - v_{a_i}^T R^T v_{b_i} + v_{a_i}^T R^T R v_{a_i} \right) \tag{55}$$

$$\Sigma^2 = \sum_{i=1}^{N} \left(v_{b_i}^T v_{b_i} + v_{a_i}^T v_{a_i} - 2 v_{b_i}^T R v_{a_i} \right) \tag{56}$$

Trying to minimize Eq 56 is like trying to maximize the last term $\Sigma_{i=1}^N v_{b_i}^T R v_{a_i}$. I can write this as follows, rearranging to pull out R:

$$\sum_{i=1}^{N} R v_{a_i} v_{b_i}^T \tag{57}$$

I can define a matrix H as:

$$H \triangleq \sum_{i=1}^{N} v_{a_i} v_{b_i}^T \tag{58}$$

Notice that H is a diagonal matrix so when we try to maximize Eq 57, we can write it as maximizing $Trace(RH)$.

There is a lemma in (2) that states that for any positive definite matrix AA^T and any orthonormal matrix B that:

$$Trace(AA^T) \geq Trace(BAA^T) \tag{59}$$

If I perform an SVD on H, I get:

$$SVD(H) = U\Sigma V^T \tag{60}$$

Now let me define another matrix using these SVD matrices, defined as X:

$$X = VU^T \tag{61}$$

If I multiply X and H together, I get a positive definite matrix:

$$XH = VU^T U\Sigma V^T = V\Sigma V^T \tag{62}$$

Now if I apply the Lemma, I can say that the $Trace(XH)$ should be bigger than any other combination of orthonormal matrices (using B):

$$Trace(XH) \geq Trace(BXH) \tag{63}$$

Eq 63 gives me the confidence to say that X will maximize $Trace(RH)$ if $X = R$. Therefore, I have now solved for the rotation matrix using the SVD of H:

$$R = X = VU^T \tag{64}$$

5.3 Finding the Translation

Now finding the translation is easy. I simply calculate the translation as the difference between the transformed point cloud b_i and the rotated original point cloud a_i:

$$T = b_i - Ra_i \tag{65}$$

And now I have the translation and rotation matrices to take point cloud a_i to destination b_i.