Group project 1 – CS340/MATH321 – Geometrical modelling and numerical analysis

Musabbir Abdul Majeed

Fall 2019

Instructions

This group project will contribute 10% towards the final marks. The deadline to submit the project is September 27, 2019 18:30 hours. The project must be submitted via **Github**. You should include both the tex and pdf files, and any code files you may have. Please note that this is a **group** project. Groups for this project are mentioned in the "group.txt" file.

0.1 Github Classroom instructions

Please use this link to join the GitHub Classroom

- One of the persons in the group should open that link and select the email address from the list
- On the next page you will be asked to create your team. Your team name must by named 'xxxxx', where 'xxxxx' is the id (digits) of the person creating that the team
- Then the remaining members of the group must use the Github link and select their email address from the page to join the team which was created earlier.

0.2 Marking scheme

The project will be marked out of 100.

- 10% marks will be for the presentation.
- 20% for following good coding practises
- 70% for the correctness and understanding of the topic. Both questions carry equal marks
- Each student will grade the remaining students in their group as a ratio highlighting how much effort others have put in. For example, if A, B, C, and D are in one group, then A will grade B, C, and D. If A thinks that B, C, and D have put in equal effort then A should write " B:C:D = 1:1:1". On the other hand, if A thinks B's contribution to the project is twice as much as C and D, then A should write " B:C:D = 2:1:1". If a group has three students A, B, and C, then A will grade the efforts of B and C. For example A thinks that C has put in a bit more effort than

B. In that case A may write "B:C = 1:1.1". All students in a group will **individually** submit the contribution of others in a group via LMS.

A good project would contain, but not limited to, the following:

- Document typed in LaTeX. The basic template for LaTeXand the associated makefile is available to download from the course website on LMS. You are welcome to use your own LaTeXtemplate
- A document free of typing errors
- Using figures/diagrams/set diagram, where possible, to explain your answers. As the cliché goes – a picture is worth a thousand words
- Concise and thorough answers. A long report doesn't necessarily mean a good report
- List of references
- Figures should be properly labelled
- Please make sure you read your submission from the reader's perspective
- Useful comments in the code.
- Code free of magic numbers
- Good unit tests in test_meshlib.py
- No dangling print statements
- Keeping the repository neat and tidy. For example, there should be just one code/tex file i.e. we don't want to see XXXv1.tex, XXXv2.tex, XXXv2Final.tex.
- Regular code commits

0.3 Late submission policy

Late submissions, unless approved beforehand, will be penalised according to the following.

# hours past the deadline	Percentage penalty
< 1 hour	5%
1-2 hours	15%
2-3 hours	30%
3-4 hours	45%
>4 hours	0% (not accepted)

0.4 Plagiarism and collusion

There is a zero tolerance policy towards plagiarism and/or collusion. If a student(s) is found to have plagiarised and/or colluded, or the work submitted is not their own, (s)he will be given an **F in this course**. Furthermore, they will be reported to the academic code of conduct committee which would affect your academic standing in the university. If you are unsure whether you are plagiarising, please ask.

Please note that even if you understand everything, copying someone else's work is still plagiarism.

***

In the event that something is not clear from the question, you are strongly encouraged to use the discussion forum on LMS. **Individual enquiries to instructor and Basem will not be entertained**

Questions

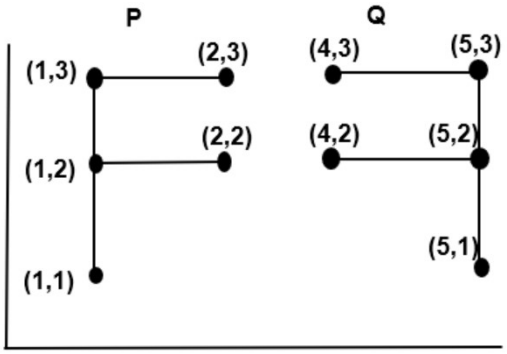

Figure 1: denoting set of points of P and Q

1. Let $\vec{p_i}$ and $\vec{q_i}$ denote the set of points in $\mathbb{R}^2$ (i.e. $\mathbf{P} = \{\vec{p_i}\} = \{(1,1),(1,2),(2,2),(1,3),(2,3)\}$ and $\mathbf{Q} = \{\vec{q_i}\} = \{(5,1),(5,2),(4,2),(5,3),(4,3)\} \in \mathbb{R}^2$). The optimal transformation that would be required to align the set of points $\mathbf{P}$ onto $\mathbf{Q}$ would be a rotation of 180° and a translation of 6 units in the x direction. In matrix notation the information above can be represented as

$$\mathbf{R} = \begin{bmatrix} \cos 180 & -\sin 180 \\ \sin 180 & \cos 180 \end{bmatrix}$$
$$\vec{t} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Now consider two sets of n points $\vec{p_i}$ and $\vec{q_i}$ in $\mathbb{R}^d$. You are required to compute the optimal rigid body transformation such that we minimize $\|\cdot\|_{L_2}$. The rigid body transformation can be found by minimising the following cost function:

$$J(\mathbf{R}, \vec{t}) = \arg\min \sum_{i=1}^{n} \omega_i \|(\mathbf{R}\vec{p_i} + \vec{t}) - \vec{q_i}\|^2$$

where ω_i are the weights and can be arbitrarily chosen.

   a. Show that the optimal translation is given by $\vec{t} = \vec{q} - \mathbf{R}\vec{p}$ where $\vec{q}, \vec{p}$ represent the weighted centroids of P and Q. (Hint: Assume that R is initially fixed)

   b. Using the value of $\vec{t}$ from part (a), show that the optimal rotation is given by

$$J(\mathbf{R}) = \arg\min \sum_{i=1}^{n} \omega_i \|\mathbf{R}\vec{\alpha_i} - \vec{\beta_i}\|^2$$

where $\vec{\alpha}_i's$ and $\vec{\beta}_i's$ must be defined by you.

c. Show that the optimal rotation above can be re-written as

$$\arg\max \sum_{i=1}^{n} \omega_i \vec{\beta}_i^T \mathbf{R} \vec{\alpha}_i$$

(Hint: $\mathbf{R^T R} = \mathbf{I}$)

d. By converting the expression above into matrix notation, show that the summation is equivalent to $tr(\mathbf{WB^T RA})$ (where $tr(\mathbf{C})$ denotes the trace of $\mathbf{C}$ and $\mathbf{W}, \mathbf{B}, \mathbf{A}$ are matrices containing $w_i's, \vec{\beta}_i's, \vec{\alpha}_i's$ respectively)

e. By writing $\mathbf{S} = \mathbf{AWB^T}$, compute its SVD and show that the trace from part d can be written as $tr(\sum \mathbf{V^T RU})$

f. Show that in order to maximize the trace above, $\mathbf{R} = \mathbf{VU^T}$

g. What qualifies as a rotation matrix? How would you define a rotation matrix in the context of n dimensions?

h. Implement and test your algorithm in two- and three-dimensions. State the implicit assumptions in the above formulation for computing the optimal rigid body transformation. You should test your algorithm on wide range of test cases, and state when it may break down

i. Describe the role of ω_i and how varying them will, if at all, alter the optimal transformation

2. You are given a selection of meshes in the form of a *txt* file.

- You are required to write a parser which reads the txt file and outputs an *obj* file in the current directory. You should assume that the mesh can comprise of both quadrilateral and triangle faces. Use Blender to convert one of the valid obj files to STL file format and send it to the 3d printer in the university.

- Write a function which outputs whether the face normals for a given mesh are consistently oriented. If the face normals aren't consistently oriented the function should return a list of faces whose normals are inconsistent. List any assumptions you have made in your algorithm. You should test your algorithm on the meshes provided in `testMeshes` folder, and state which txt files aren't a valid mesh

- Write a function which makes face normals consistent. The function is the previous part will tell you whether there are any inconsistent normals. List any assumptions you make about this algorithm

- Write a function that outputs the non-manifold edges of a mesh. You should test your algorithm on the meshes provided in `testMeshes` folder, and state which txt files aren't a valid mesh

- Write a function which takes as input lists of vertices and faces of a mesh and outputs a list of faces connected to a vertex. The function should return a dictionary where the key of the dictionary is the vertex id, and the value is the list of faces. Ideally, we want to ensure that the faces attached to a vertex are listed in an anticlockwise manner

- Write a function which computes the area of the mesh. You can assume that the mesh only

consists of triangles

For your convenience we have given you three valid test meshes, namely `validTriangleMesh.txt`, `validQuadrilateralMesh.txt`, and `validTriQuadMesh.txt`. The corresponding obj files can be viewed in Blender. As the names implies, `validTriangleMesh.txt`, `validQuadrilateralMesh.txt`, and `validTriQuadMesh.txt` comprises of triangles, quadrilaterals, and mix meshes respectively. You are encouraged to create smaller meshes for testing your algorithm.

To make testing easier, we have included a test driver. You should call your functions from the test driver `test_meshlib` and test the functionality of each of the above features. You are allowed to define the function signatures as you wish. Usually, you will have multiple `test_xxx` functions for a given function.

Please do not change the names of the python files provided to you. Write all your functions in meshlib.py