Programming Assignment 5 – 600.445/645 Fall 2016

Score Sheet (hand in with report) Also, PLEASE INDICATE WHETHER YOU ARE IN 600.445 or 600.645

<table>
<thead>
<tr>
<th>Name 1</th>
<th>Email</th>
<th>Other contact information (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name 2</th>
<th>Email</th>
<th>Other contact information (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Signature (required) I (we) have followed the rules in completing this assignment

_________________________ _______________________________

Grade Factor

<table>
<thead>
<tr>
<th>Program (40)</th>
<th>Design and overall program structure</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reusability and modularity</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Clarity of documentation and programming</td>
<td>10</td>
</tr>
<tr>
<td>Results (20)</td>
<td>Correctness and completeness</td>
<td>20</td>
</tr>
<tr>
<td>Report (40)</td>
<td>Description of formulation and algorithmic approach</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Overview of program</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Discussion of validation approach</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Discussion of results</td>
<td>10</td>
</tr>
</tbody>
</table>

TOTAL 100

NOTE: This is an optional assignment.
If you hand it in, I will use the grade to replace the lowest other programming assignment or written homework assignment with one exception:
You may not drop both HW#3 and HW#4. If these are your two lowest grades, then I will drop the lower of those two under the drop 1 homework scenario and replace the next lowest grade (other than the other of HW#3-4) with this score.
Instructions

• You may work alone or in groups of 2 people. I am opposed to groups of three or more.
  
  o Contact the TA if you are having trouble finding a partner.
  
  o In the past, a team consisting of one person who is a bit stronger in math with one who is a bit stronger

• Except in extraordinary circumstances, both members of the group will receive the same grade, but your report should indicate who did what.

• You may use standard numerical libraries and packages for least-squares problems and Cartesian coordinate transformations. One such library is that associated with Numerical Recipes in C. There is also a numerical library I wrote for the ERC.
  
  o If you do use software from our web site, remember that it is copyrighted and part of the CIS ERC infrastructure. You have permission to use it for the purposes of this class, but must secure my written permission before you (a) transfer it to any other persons or (b) use it for other purposes than this class. I am considering making these libraries “open source”, but we aren’t quite there yet.

• You must cite the source of any software that you do not write yourselves.
• You may not use the results of previous years’ programming assignments or the work of other students on this year’s assignments.

• Your programs (and the reports) should clearly indicate authorship of the code and also should cite any libraries that you use, including source.

• Do not go out and try to find a package that solves this particular problem. Numerical libraries are one thing, but using someone’s registration or calibration code is entirely another matter.

• The TA will also post some possibly useful software from the ERC on our web site. You do not have to use this software, but may find it helpful.

• Note that these rules are intended to supplement the general instructions discussed at the opening session of the class, not to replace them. If you have questions, see me or the TA.

• You should hand in a bound report containing the following:
  o A narrative report (typically about 5-8 pages long) summarizing
    ▪ The mathematical approach taken
    ▪ The algorithmic steps followed
    ▪ An overview of the structure of the computer program, sufficient to enable someone with reasonable skill (the grader) to understand your approach and follow your code.
The steps taken to verify that the program is working correctly. Typically, this would take the form of a discussion of the results using the debugging examples.

- A tabular summary of the results obtained for unknown data
- A short discussion for the results of running your program. This certainly includes the tabular summary above, but may also include a discussion of convergence if you adopt an iterative process or of difficulties if you suspect that your answer is wrong.

- A short statement of who did what.
  - A well structured and well documented program listing
  - A CD, ZIP disk, or diskette containing:
    - A directory called “PROGRAMS”, containing (at least) all source files for your program, together with a file “README.TXT”, containing the names of all the source files, together with a 1-line description of each file. Optionally, you may also include an executable program and instructions for using it.
    - Another directory, called “OUTPUT”, containing the program output files in the specified format and with the specified name (see below).
  - My recommendation is that you use a loose-leaf notebook with a pocket in back for the CD or diskette.
Abstract of the Problems

• In these problems, you will be asked to extend the ICP program of assignments 3 and 4 to perform a simple deformable registration. The input files are similar to the earlier exercises. They include

  1. A 3D surface, represented as a mesh of triangles. You will be given the coordinates of the vertices of this mesh in CT coordinates. For the purposes of this exercise, you can think of this object as being a bone. In this case, the surface will represent the “average” shape of the bone.

  2. An atlas “modes” file, giving modes of variation for the model. “Mode 0” represents the average shape (i.e., the shape provided in the mesh file). If \( \vec{m}_{m,k} \) represents a 3D value associated with vertex \( k \) and mode \( m \), then the actual coordinate of vertex \( k \) will be given by

\[
\vec{m}_k = \vec{m}_{0,k} + \sum_{m=1}^{N_{modes}} \lambda_m \vec{m}_{m,k}
\]

  3. A pair of definition files for two rigid bodies, “A” and “B”. Each file gives the positions of LED markers \( \vec{A}_i \) and \( \vec{B}_i \) in body coordinates, together with the positions of two tips \( \vec{A}_{tip} \) and \( \vec{B}_{tip} \) in body coordinates. As shown in the illustration below, we assume that
the “tip” of the B rigid body is rigidly screwed into the bone in some unknown orientation. The A rigid body is used as a pointer. I.e., its tip is placed into contact with a number of points on the surface of the bone.

4. A file of “sample” readings giving the positions and $\vec{b}_{i,k}$ of the LED markers relative to an optical tracker when each sample $k$ is taken.

- You are to output the CT coordinates $\vec{c}_k$ corresponding to each sample taken, for the “deformed” atlas, together with the values for the $\lambda_m$.

- Note that this assignment is based loosely on the “active appearances” method of Cootes and Taylor and of other authors:


Figure 1: Scenario
Hints on suggested procedure

• Start with your program for Programming Assignment #4.

• Input the data and verify that the vertices of the mesh you have read in are the same as “Mode 0”.

• Perform an initial “rigid” registration using the method for Programming Assignment #4. This will produce a rigid transformation $F_{reg}^{(0)}$.

• Let $s_k(t) = F_{reg}^{(t)} \cdot d_k$ be your current estimate of transformed sample point $k$ at iteration $t$, where $F_{reg}^{(t)}$ is the current estimate of the rigid transformation, and $d_k = F_{B,k}^{-1} \cdot F_{A,k} \cdot A_{tip}$ represents the corresponding measured sample point value.

• Suppose $c_k(t)$ represents your current estimate of the closest point on the deformed surface to the transformed sample point. Then $c_k(t)$ will be on some triangle. Suppose the vertex indices of this triangle are $\{s, t, u\}$. Then the coordinates of the corresponding deformed mesh will be
\[
\mathbf{m}_s = \mathbf{m}_{0,s} + \sum_{m=1}^{N\text{modes}} \lambda_m^{(t)} \mathbf{m}_{m,s}
\]
\[
\mathbf{m}_t = \mathbf{m}_{0,t} + \sum_{m=1}^{N\text{modes}} \lambda_m^{(t)} \mathbf{m}_{m,t}
\]
\[
\mathbf{m}_u = \mathbf{m}_{0,u} + \sum_{m=1}^{N\text{modes}} \lambda_m^{(t)} \mathbf{m}_{m,u}
\]

(1)

- If we compute the barycentric coordinates of \( \mathbf{c}_k^{(t)} \) on the triangle

\[
\mathbf{c}_k^{(t)} = \zeta_k \mathbf{m}_s + \xi_k \mathbf{m}_t + \psi_k \mathbf{m}_u
\]

we can get an expression in terms of mode coordinates \( \mathbf{c}_k^{(t)} \)

\[
\mathbf{c}_k^{(t)} = \mathbf{q}_{0,k} + \sum_{m=1}^{N\text{modes}} \lambda_m^{(t)} \mathbf{q}_{m,k}
\]

(2)

where

\[
\mathbf{q}_{m,k} = \zeta_k \mathbf{m}_{m,s} + \xi_k \mathbf{m}_{m,t} + \psi_k \mathbf{m}_{m,u}
\]

and \( \Lambda^{(t)} = \{\lambda_1, \cdots, \lambda_{N\text{modes}}\}^{(t)} \) represents the current estimate of the mode weights. So
\[ \vec{s}_k^{(t)} = \mathbf{F}_{\text{reg}}^{(t)} \bullet \vec{d}_k \]

\[ \vec{c}_k^{(t)} = \vec{q}_{0,k} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t)} \vec{q}_{m,k} \]

should represent the same point. I.e.,

\[ \mathbf{F}_{\text{reg}}^{(t)} \bullet \vec{d}_k \approx \vec{q}_{0,k} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t)} \vec{q}_{m,k} \]

**Note:** In the above, remember that \( s, t, u \) also depend on \( \vec{c}_k \).

- Now, you have a few choices. One choice is to iterate the following sequence:
  - Iterate the following sequence some number of times until it seems to be stalled:
    - Keeping \( \vec{s}_k^{(t)} \) fixed, find the corresponding \( \vec{q}_{m,k} \) solve the following least squares problem for \( \Lambda^{(t+1)} = [\lambda_1^{(t+1)}, \ldots, \lambda_{N_{\text{modes}}}^{(t+1)}] \)

\[ \vec{s}_k^{(t)} \approx \vec{q}_{0,k} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t+1)} \vec{q}_{m,k} \]
- Use $\Lambda^{(t+1)}$ to update the surface mesh model to find a new estimate for the vertices of the deformed model.

- If you are using a spatial data structure to speed your search, make the appropriate updates. E.g., if you are using some form of bounding box tree, update the bounds. Note that it is not always necessary to completely recompute the tree, since some loss of efficiency is ok.

- Now, find new matching points $\vec{c}_k^{(t+1)}$

- Iterate for a while, until you meet some suitable stopping condition
  - After this iteration converges, keep the model vertices fixed and use the method of PA#4 to re-estimate $F_{reg}^{(t)}$. Use the result to re-transform the sample points.
  - Iterate this sequence (mode matching, rigid body transformation, ...) until done.

- Alternatively, you can combine the two iterations by using a linearized form of Equation (4):
\[ F_{\text{reg}}^{(t+1)} \cdot \vec{d}_k = \vec{c}_k^{(t)} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t+1)} \vec{q}_{m,k} \]

\[ \Delta F^{(t+1)} \cdot F_{\text{reg}}^{(t)} \cdot \vec{d}_k = \vec{c}_k^{(t)} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t+1)} \vec{q}_{m,k} \]

\[ \Delta F^{(t+1)} \cdot \vec{s}_k^{(t)} = \vec{c}_k^{(t)} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t+1)} \vec{q}_{m,k} \]

where \( \Delta F = [\Delta R(\vec{\alpha}), \vec{\varepsilon}] \). This rearranges to

\[ \vec{s}_k^{(t)} + \vec{\alpha}^{(t+1)} \times \vec{s}_k^{(t)} + \vec{\varepsilon}^{(t+1)} \approx \vec{c}_k^{(t)} + \sum_{m=1}^{N_{\text{modes}}} \lambda_m^{(t+1)} \vec{q}_{m,k} \]

(7)

- In this case, you do the following:
  - Solve (7) to estimate \( \Lambda^{(t+1)}, \vec{\alpha}^{(t+1)}, \) and \( \vec{\varepsilon}^{(t+1)} \). Use \( \vec{\alpha}^{(t+1)} \) and \( \vec{\varepsilon}^{(t+1)} \) to produce a real value for \( \Delta F^{(t+1)} \), remembering that you need to make a proper rotation, and cannot just use a skew matrix.
  - Update \( \vec{s}_k \) and the mesh using \( \Delta F^{(t+1)} \) and \( \Lambda^{(t+1)} \). Again, remember that you may have to update bounding boxes or other spatial data structures.
• Now, find new matching points $\vec{c}_k^{(t+1)}$

• Iterate for until you meet some suitable stopping condition.

• I solved the problems both ways. You can even mix them. In solving the problem, I used the first method, followed by the second method.

• **NOTE:** Again, there are other ways to approach this problem, as well. You are free to develop an alternative algorithm so long as you explain it clearly and explain your reasoning.
Input data

You will be given a rigid body design files “ProblemX-BodyY” where X is 3 or 4 and Y is A or B. These files have the following format

<table>
<thead>
<tr>
<th>Record</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( N_{\text{markers}} ) “ProblemX-BodyY”</td>
<td>Number of marker LEDs Filename</td>
</tr>
<tr>
<td>Next ( N_{\text{markers}} ) records</td>
<td>( Y_x, Y_y, Y_z )</td>
<td>xyz coordinates of marker LEDs in body coordinates</td>
</tr>
<tr>
<td>Next record</td>
<td>( t_x, t_y, t_z )</td>
<td>xyz coordinates of tip in body coordinates</td>
</tr>
</tbody>
</table>

Each line (record) of the file will terminate with an end of line character.
The format of the body surface definition file “ProblemXMesh.sur”

<table>
<thead>
<tr>
<th>Record</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( N_{\text{vertices}} )</td>
<td>Number of vertices</td>
</tr>
<tr>
<td>Next ( N_{\text{vertices}} ) records</td>
<td>( V_x, V_y, V_z )</td>
<td>xyz coordinates of vertices in CT coordinates</td>
</tr>
<tr>
<td>Next record</td>
<td>( N_{\text{triangles}} )</td>
<td>Number of triangles</td>
</tr>
<tr>
<td>Next ( N_{\text{triangles}} ) records</td>
<td>( i_1, i_2, i_3, n_1, n_2, n_3 )</td>
<td>Vertex indices of the three vertices for each triangle, followed by triangle indices for the three neighbor triangles opposite to the three vertices (not needed for this problem). “-1” means not a valid neighbor.</td>
</tr>
</tbody>
</table>
The format of the atlas modes file is as follows:

<table>
<thead>
<tr>
<th>Record</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Problem5Modes.txt</td>
<td>nnnn = Number of vertices</td>
</tr>
<tr>
<td></td>
<td>Nvertices=nnnn</td>
<td>mmm = Number of modes</td>
</tr>
<tr>
<td></td>
<td>Nmodes=mmm</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mode 0 :Average Vertex Values</td>
<td>Just a comment</td>
</tr>
<tr>
<td></td>
<td>Next $N_{\text{vertices}}$ records</td>
<td>xyz coordinates of vertices in CT</td>
</tr>
<tr>
<td></td>
<td>$V_x, V_y, V_z$</td>
<td>coordinates of “mode 0” – i.e., the mean shape</td>
</tr>
<tr>
<td></td>
<td>Next record</td>
<td>Mode 1 : Vertex Displacement</td>
</tr>
<tr>
<td></td>
<td>Next $N_{\text{vertices}}$ records</td>
<td>Vertex displacements for mode 1</td>
</tr>
<tr>
<td></td>
<td>$D_x, D_y, D_z$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Next record</td>
<td>Mode 2 : Vertex Displacement</td>
</tr>
<tr>
<td>... et cetera ...</td>
<td>... et cetera ...</td>
<td>... et cetera ...</td>
</tr>
</tbody>
</table>

Note that the number of modes in the atlas file may exceed the number of modes to be used in any individual problem.
Finally, you will be given a file of sample readings “pa5-X-ddddd-
\(N_{samps}\) SampleReadings.txt”, where X is a letter and ddddd is “debug” or “unknown”. This file has the following format.

<table>
<thead>
<tr>
<th>Record</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(N_S = N_A + N_B + N_D, N_{samps}), “pa5-X-ddddd-SampleReadings.txt”, (N_{modes})</td>
<td>Number of LEDs read by the tracker in each sample frame (&quot;A&quot; markers, &quot;B&quot; markers, &quot;Dummy&quot; markers)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of sample frames</td>
</tr>
<tr>
<td></td>
<td></td>
<td>File name,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of modes you are to use from the atlas in solving this problem</td>
</tr>
<tr>
<td>Next (N_A) records</td>
<td>(x,y,z)</td>
<td>xyz coordinates of A body LED markers in tracker coordinates</td>
</tr>
<tr>
<td>Next $N_B$ records</td>
<td>$x, y, z$</td>
<td>xyzcoordinates of B body LED markers in tracker coordinates</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Next $N_D = N_S - N_A - N_B$ records</td>
<td>$x, y, z$</td>
<td>xyzcoordinates of other (unneeded) LED markers in tracker coordinates</td>
</tr>
<tr>
<td>This pattern of $N_S$ records is repeated for a total of times.</td>
<td>See above</td>
<td>Additional sets of data corresponding to each sample</td>
</tr>
</tbody>
</table>
**Output Data**

You should produce an output data file with the format.

<table>
<thead>
<tr>
<th>Record</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$N_{\text{samps}}$, “pa5-X-Output.txt”, $N_{\text{modes}}$</td>
<td>Number of sample frames, File name</td>
</tr>
<tr>
<td>1</td>
<td>$\lambda_1, ..., \lambda_{N_{\text{modes}}}$</td>
<td>Mode weights determined (give 4 decimal places)</td>
</tr>
<tr>
<td>Next $N_{\text{samps}}$ records</td>
<td>$s_x, s_y, s_z, \ c_x, c_y, c_z$, $| \vec{s}_k - \vec{c}_k |$</td>
<td>xyz coordinates of $\vec{s}_k$, xyz coordinates of $\vec{c}_k$, magnitude of difference</td>
</tr>
</tbody>
</table>

For debugging purposes, I have provided the output my program got. I have also included an “answer” file that contains data used to generate the test problems. In some cases the answer data will also have non-zero error due to various simulated noise.