PA5 Program Listing

1. Classes
    - 1.1. Vector.m
    - 1.2. Rotation.m
    - 1.3. Frame.m

2. Functions
    - 2.1. registration3dTo3d.m
    - 2.2. find_frame_transformation.m
    - 2.3. find_closet_point_on_triangle.m
    - 2.4. find_closest_point_on_mesh.m
    - 2.5. verify.m
    - 2.6. get_barycentric_coord.m
    - 2.7. get_deformed_coord
    - 2.8. termination_test.m
    - 2.9. Functions for I/O
        - 2.9.1. get_num.m
        - 2.9.2. validate.m
        - 2.9.3. form_vector_set.m
        - 2.9.4. read_from_body.m
        - 2.9.5. read_from_mesh.m
        - 2.9.6. read_from_modes.m
        - 2.9.7. read_from_sample.m
        - 2.9.8. write_result.m

3. Executable program
    - 3.1. main.m

1. Classes

1.1 Vector.m - please refer to the program listing for PA1 for details (A new function is added)

Class Method

`result = dot(obj1, obj2)` computes the dot product of the two given Vector objects.

Parameter(s)

- `obj1` one of the Vector objects which the dot product is to be found
- `obj2` one of the Vector objects which the dot product is to be found

Returned value(s)t

- `result` the dot product of the two given Vector objects

1.2 Rotation.m - please refer to the program listing for PA1 for details

1.3 Frame.m - please refer to the program listing for PA1 for details

---

2. Functions

2.1 registration3dTo3d.m - please refer to the program listing for PA3 for details

2.2 find_frame_transformation.m

`frame_set = find_frame_transformation(a, b)` obtains a set of frame that transforms the set of points in a to each set of observed points b

Parameter(s)

- `a` the set of points to be transformed to the other set of points
- `b` sets of point from which a is to be transformed into

Returned value(s)

- `frame_set` the set of frame that transforms a into each set of b

2.3 find_closet_point_on_triangle.m - please refer to the program listing for PA3 for details

2.4 find_closest_point_on_mesh.m - please refer to the program listing for PA4 for details

2.5 verify.m

`bool = verify(m, v_set)` verifies that the vertices of the mesh from the input file matches with "Mode 0"

Parameter(s)

- `m` the set of points to be transformed to the other set of points
- `v_set` sets of point from which a is to be transformed into

Returned value(s)

- `bool` 1 if the vertices from the input file matches with "Mode 0", 0 otherwise

2.6 get_barycentric_coord.m

`[lambda, mu, v] = get_barycentric_coord(c, m_s, m_t, m_u)` computes the barycentric coordinates of a point given the three vertices of the triangle that the point is on.

Parameter(s)

- `c` the position of the point
- `m_s` the position of the first vertex of the triangle that the point is on
- `m_t` the position of the second vertex of the triangle that the point is on
- `m_u` the position of the third vertex of the triangle that the point is on

Returned value(s)

- `lambda` the coordinates in terms of the first vertex of the triangle
- `mu` the coordinates in terms of the second vertex of the triangle
- `v` the coordinates in terms of the third vertex of the triangle

---

2.7 get_deformed_coord.m

`[m_new, m_m] = get_deformed_coord(m, vertex, m_lams)` computes the coordinates of the given point in the deformed atlas and the corresponding vector displacements of each mode of that point

Parameter(s)

- `m` a matrix consisting of vector displacements of each vertex for each mode
- `vertex` the vertex which the coordinates in the deformed atlas are to be found
- `m_lams` mode weights

Returned value(s)

- `m_new` coordinates of the vertex in the deformed atlas
- `m_m` the vector displacements of the vertex for each mode

2.8 termination_test.m

`count = termination_test(sigma, epsilon_max, epsilon_avg, n, count)` returns the number of iterations which the conditions for termination have been satisfied.

Parameter(s)

- `sigma` the variance of the c_k from s_k from the 1st to the nth iteration
- `epsilon_max` the maximum magnitude difference between c_k from s_k from the 1st to the nth iteration
- `epsilon_avg` the average magnitude difference between c_k from s_k from the 1st

to the n^{th} iteration

- `n` current iteration number
- `count` the number of iterations which the conditions for termination have been satisfied

Returned value(s)

- `count` the number of iterations which the conditions for termination have been satisfied

---

2.9 Functions for I/O

2.9.1 get_num.m

`num = get_num(fileID)` reads a floating point number in the format of '%f, ' from a text file

Parameter(s)

- `fileID` the file ID of the file to be read from

Returned value(s)

- `num` the number that has been read

2.9.2 validate

`validate(fileID, filename)` validates a text file by comparing its name to the name given in the text file. It terminates the program by throwing an error if the text file is invalid.

Parameter(s)

- `fileID` the file ID of the file to be read from
- `filename` the name of the file (including the path)

2.9.3 form_vector_set.m

`set = form_vector_set(num, fileID)` reads all vectors in a frame from a text file and uses them to form a set

Parameter(s)

- `num` the number of vectors in a frame
- `fileID` the file ID of the file to be read from

Returned value(s)

- `set` the set of vectors formed

2.9.4 read_from_body.m

`[set, tip] = read_from_body(filename)` reads all vectors describing the rigid body from a text file

Parameter(s)

- `filename` the name of the text file to be read from (including the path)

Returned value(s)

- `set` a set of vectors describing the position of marker LEDs in body coordinates
- `tip` a set of vectors describing the position of the tip in body coordinates

---

2.9.5 read_from_mesh.m

`[v_set, triangle_v_set] = read_from_mesh(filename)` reads all data describing the surface model from a .sur file

Parameter(s)

- `filename` the name of the .sur file to be read from (including the path)

Returned value(s)

- `v_set` a set of vectors describing the position of vertices on the surface model

- `triangle_v_set` a 2D array of vertex indices of the three vertices for each triangle

---

2.9.6 read_from_modes.m

`[vertex_displacements] = read_from_sample(filename)` reads all vectors describing the sample readings from a text file

Parameter(s)

- `filename` the name of the text file to be read from (including the path)

Returned value(s)

- `vertex_displacements` sets of vectors describing the displacement of each mode

---

2.9.7 read_from_sample.m

`[A_sample, B_sample, num_modes] = read_from_sample(filename, num_a, num_b)` reads all vectors describing the sample readings and the number of modes used for this dataset from a text file

Parameter(s)

- `filename` the name of the text file to be read from (including the path)

Returned value(s)

- `A_sample` a set of vectors describing the position of marker LEDs in optical tracker coordinates in rigid body A
- `B_sample` a set of vectors describing the position of marker LEDs in optical tracker coordinates in rigid body B
- `num_modes` the number of modes needed for this data set

2.9.8 write_result.m

`write_result(filename, s_set, c_set, m_lams)` outputs the results to a text file in the specified format

Parameter(s)

- `filename` the name of the text file to be outputted to (including the path)
- `s_set` position of sample points (`a_tip`) in the coordinates of the bone
- `c_set` the points on the surface mesh that are closest to s_k
- `m_lams` the mode weights of the deformed atlas

---

3. Executable program

3.1 main.m

An executable program that reads from the following text files from the INPUT directory:

- "`Problem5-Body[Y].txt`"– body design files, where Y is A or B
- "`Problem5MeshFile.sur`"– body surface definition file
- "`Problem5Modes.txt`"- the deformed atlas mode file
- "`PA5-[X]-[ddddd]-SampleReadingsTest.txt`" – file of sample readings, where, X is a letter, and ddddd is "debug" or "unknown".

This program outputs a text file named "`pa5-[`α`]-Output.txt`" in the OUTPUT directory where α is a letter between characters "a" and "k" (except i). This file includes the mode weights determined, vectors s_k (position of sample points in the coordinates of the bone), c_k (the CT coordinates corresponding to each sample taken), and the magnitude difference between c_k and s_k. This program also saves a plot named "`pa5-[`α`].png`" in the OUTPUT directory showing the average magnitude difference between `s` and `c` after each iteration of the ICP algorithm.

Program Input

When the program is run, the following will be displayed in the command window:
"`Enter a letter from 'a' to 'k' (except i) to specify a dataset:`
"

The expected input from the command line is an alphabet from `a` to `k` which identifies the data set in the relative path "`../INPUT/`" to be read.

Program Output

The program will output a text file named "`pa5-[`α`]-Output.txt`" and a plot of ε_{avg} named "`pa5-[`α`]-Output.png`" in the relative path "`../OUTPUT/`" folder. Upon program completion, the following will be displayed in the command window: "`Results written to: ../OUTPUT/pa5-[`α`]-Output.txt`".