NYU | TANDON SCHOOL OF ENGINEERING

3D Computer Vision: Techniques & Applications

Project 3: 3D Shape Registration

Project Deadline: 11:55 PM, Oct. 30, 2015

No late submission will be accepted

In this project, you are going to match pairs of corresponding points among given models (*mesh015.off* and *mesh054.off*) by implementing Hungarian algorithm. Detailed project instructions are given as following:

1. Since there're totally 12500 points in each model (HKS features of both models are given in *mesh_hks.mat, each row corresponding a HKS feature for a point*), to make it simpler, please select one point from every 125 points. For example, point 1, point 126 and point 251 etc. Therefore, 100 points will be selected.

   Example for selected 100 points:

3D Shape mesh015

3D Shape mesh054

2. Find the 100 corresponding point pairs by implementing Hungarian Algorithm. The pseudo code of Hungarian algorithm is given as following:

Step 0: Create an n*m matrix called the cost matrix and let k=min(n,m). You could calculate the distance between features of different points as their cost.

Step 1: For each row of the matrix, find the smallest element and subtract it from every element in its row. Go to Step 2.

Step 2: Find a zero Z in the resulting matrix. If there is no starred zero in its row or column, star Z. Repeat for each element in the matrix. Go to Step 3.

Step 3: Cover each column containing a starred zero. If K columns are covered, the starred zeros describe a complete set of unique assignments. In this case, Go to DONE, otherwise, Go to Step 4.

Step 4: Find a non-covered zero and prime it. If there is no starred zero in the row containing this primed zero, Go to Step 5. Otherwise, cover this row and uncover the column containing the starred zero. Continue in this manner until there are no uncovered zeros left. Save the smallest uncovered value and Go to Step 6.

Step 5: Construct a series of alternating primed and starred zeros as follows. Let Z0 represent the uncovered primed zero found in Step 4. Let Z1 denote the starred zero in the column of Z0 (if any). Let Z2 denote the primed zero in the row of Z1. Continue until the series terminates at a primed zero that has no starred zero in its column. Unstar each starred zero of the series, star each primed zero of the series, erase all primes and uncover every line in the matrix. Return to Step 3.

Step 6: Add the value found in Step 4 to every element of each covered row, and subtract it from every element of each uncovered column. Return to Step 4 without altering any stars, primes, or covered lines.

DONE: Assignment pairs are indicated by the positions of the starred zeros in the cost matrix. If C(i,j) is a starred zero, then the element associated with row i is assigned to the element associated with column j.

Note: You can search online source code to help your implementation.

Reference:

[1] Lecture slides: Lecture_005.ppt
[2] http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html

Generate color map on the surface of 3D models based on the correspondence index values and store them as .vtk files. By mapping the correspondence index values on the 3D surface, the correspondence pair of points will have the same color. Note, since we only sample 100 points on each surface for correspondence assignment, the color value of remain points (vertices) will be assigned using the color value of their nearest sample point (one point out of those 100 sample points used for correspondence assignment above). You can find the nearest sample point for a vertex based on its geodesic distances to all 100 sample points. You might use matlab function *graphallshortestpaths(G)* to calculate geodesic distance. For outputting .vtk file, please refer lecture slides *Lectrue_003.ppt.* Below are two examples for output of models with correspondence map:

3D shape mesh015

3D shape mesh054

3. 50 bonus points will be given for those who implement assignment problem with distortion term (Please refer to Lecture_005.ppt for the design of distortion term, and you can refer to any online source code to help your implementation)

Note: You may discuss the general concepts in this project with other students, but you must finish your program on your own. NO SHARING OF CODE OR REPORT IS ALLOWED. Violation of this policy can result in grade penalty.

What to submit

Please submit a .zip file containing (1) a working project, (2) generated .vtk files for both models and (3) a report for the detailed description of the project (how the project was coded, how to run your project and screenshots of the colorful output). Before submit your project, please make sure to test your program on all the given models.