MINI Project #3 Report

3D Shape Registration

Shuaiyu Liang (sl5352)

1. Introduction to submitted bag

project_3_sl5352.m	Script for project
hungarian_sl5352.m	Function to implement Hungarian algorithm
Mesh015.off	Mesh#1
Mesh054.off	Mesh#2
Mesh015.vtk	With color value mapped #1 vtk file
Mesh054.vtk	With color value mapped #1 vtk file

2. How this was coded and how to run

2.1 Project script

In this part, I firstly loaded two hks mat. And sampled them as well as sum 101 hks together as one weight of one point. Then I load mesh, computes it all points wise geodesic distance. Than mapping color value as $\frac{\sum w_i c_i}{\sum c_i}$, while represented in code as color_weight. Than write to vtk file form. The same processing method for second mesh load and geodesic distance compute. The only difference is sample the geodesic matrix, where the first mesh we sample in every 125 points and here we sampled by assignment method. And this represent how we do image registration.

2.2 Hungarian algorithm function

Inspired by the link on the project description, I choose switch case forms to handle these steps. Then things getting clearer. Step 1 subtract min value from each row. Step 2 find zero and star them element by element then cover the row and column. In this step introduced two vector to represent zero status of row and column. Introduced mask = 1 for stared zeros. Step 3 to check column value is equal to matrix entries or not. Step 4 similarly prime zeros and subtract min value from rest uncovered row and column. Introduced zeroflag and exit flag. Introduced mask = 2 for primed zeros. This step is crucial for preparing expanding rows covered for zeros. Step 5 followed by algorithm find stared zero in the primed column and get the other primed in that stared row. And then un-star all stared and star all primed for expanding rows. And step 6 was to add and to subtract.

2.3 How to run

After run the project script, you get two vtk files. Load these vtk file get the following outcome.

3. My VTK files(with color value based on point matching and geodesic distance)

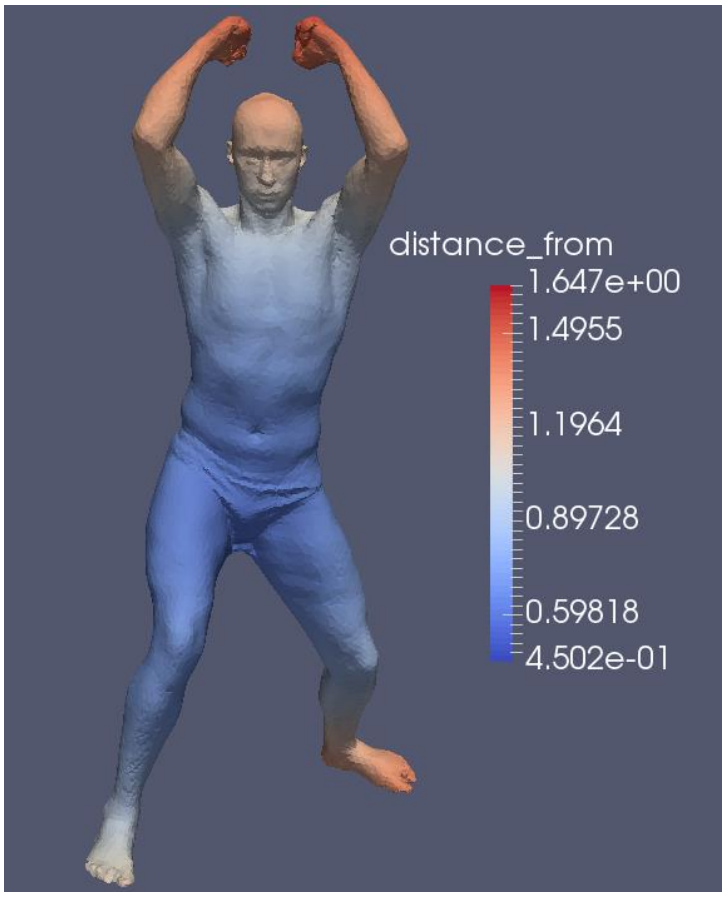

Figure 1. mesh015.vtk with heat starting from head looks like

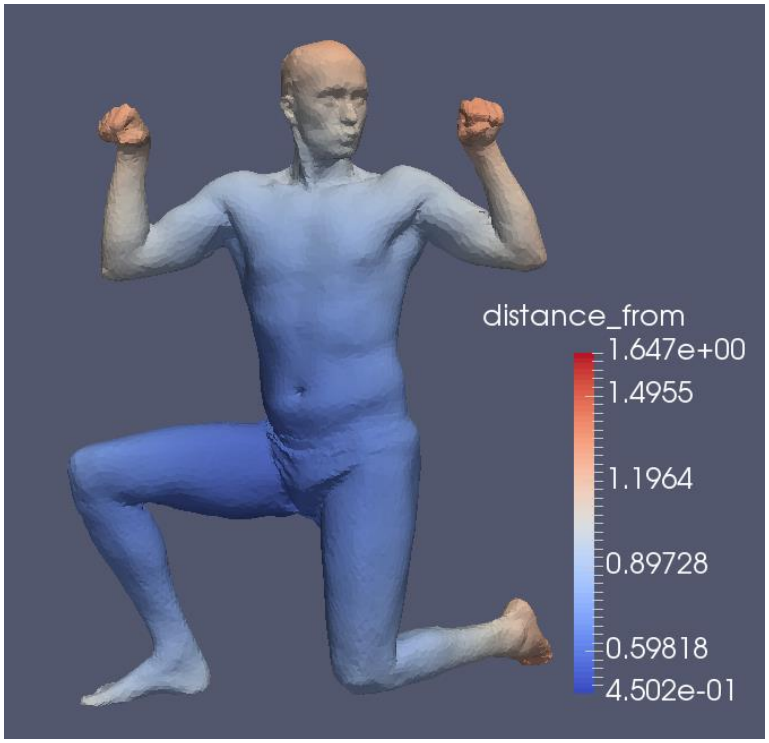

Figure 2. mesh054.vtk with heat starting from head looks like

4. Matching evaluation

We selected points on pre-awareness that is every 125 points, we pick on based purely on index. And two meshes' index is perfectly matched. However using my Hungarian, a mis-matching could be found in point # 65 to point #68 and may be some more.

I tried to implement the distortion term by extracting these 100 pair wise geodesic distance and tried to sum the distance. This is the distance from one certain point to all other settled points. If points are perfectly matching, this sum should be the same. However the value varies in a large range.