u

EL-GY 9143

3D Computer Vision: Techniques and Applications

NAMAN PATEL

N17075820

nkp269@nyu.edu

Project 3:
3D Shape Registration

Due 5th November, 2015

[5 Pages]

Objective:

To perform 3D shape registration using Hungarian Assignment Algorithm on 3D Shapes.

Introduction:

3D models represent a physical body using a collection of points in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces, etc. Being a collection of data (points and other information), 3D models can be created by hand, algorithmically (procedural modeling), or scanned.

Shape analysis is the mainly automatic analysis of geometric shapes, for example using a computer to detect similarly shaped objects in a database or parts that fit together. For a computer to automatically analyze and process geometric shapes, the objects have to be represented in a digital form.

Shape Registration is finding a meaningful correspondence between two or more shapes.

Under different contexts, the problem has also been referred to as shape correspondence, alignment, or shape matching.

A meaningful shape registration depends on the application

- Geometry-based
- Knowledge-based

Different Approached for Shape registration are:

- Alignment based Approach
    - Rigid Alignment
    - Non-Rigid Alignment
- Feature Matching based Approach
    - Local Geometry Featured
    - Spectral Shape Descriptor
    - GPS Shape Signature
    - Heat Kernel Signature
- Assignment Algorithm

Our objective is to implement a 3D Shape registration of points on an image using Assignment Algorithm.

These are the steps involved in designing the shape registration algorithm:

- Load the 3D Shape Feature datasets
- Reduce the points from 12500 to 100 by taking only points 1,126, etc into consideration.
- Find the 100 corresponding point pairs by implementing Hungarian algorithm for assignment problem to the other shape
- Evaluate geodesic distance between all the vertices
- Note down the 12500 points nearest to all the 100 corresponding points and make a heatmap(VTK) accordingly.

Hungarian Algorithm for Assignment Problem:

- Extract the corresponding 100 points of both the shapes.
- Calculate the Euclidean distance between all the corresponding shape points.
- Now this cost function is to be used for assigning.
- For each row of the matrix, find the smallest element and subtract it from every element in its row.
- For each column of the matrix, find the smallest element and subtract it from every element in its column.
- Find all the zeros in the matrix
- Add an extra column which act as a pointer to zero in the column corresponding to their respective rows. If there are more than 1 zeros in a row then the column containing the zero with the smaller index will contain the pointer to the next zero. For example for a column b is row 1 and 2 have zeros then row 1 will column b element will be -2, row 2 column b element will be 0 and the last row will be -1. Thus, the rows with negative element excluding in the last column will be unassigned rows.
- Perform an initial assignment on the matrix and find all the unassigned zeros as explained in the above example. Do this for both row and column and find all the unassigned elements.
- If an unassigned zero is found then, find the smallest element which is unassigned and subtract it from each unassigned row and then add it to each covered column.
- Perform the assignment operation again and again if there are unassigned zeros then perform the above 4 steps again.

Algorithm for Shape Registration:

- Select 100 points at the same index for both the shapes.
- Get the Geodesic Distance of each vertex to all other vertices for both the shapes.
- For the original shape, initialize a color weight matrix and give different weights to all the 100 points selected.
- For all the vertices of the shape assign each vertex the weight corresponding to the weight of the selected point nearest to it.
- Now perform the Hungarian Algorithm on the two shapes to get the assignment for each of the selected point.
- Assign the weights to the new shape corresponding to its assignment. For example is point 1 of the original shape is assigned to point 3 of the test shape then the weight corresponding to point 3 of the new shape will be equal to the weight corresponding to point 1 of the original shape.
- For all the vertices of the shape assign each vertex the weight corresponding to the weight of the selected point nearest to it.
- Generate the vtk file for both the shape using the weights of the shapes as the lookup table.

Description of Files:

Euclidean_Distance.m: Function for calculating Euclidean distance between each point of two matrices

Hungarian.m: Function for Evaluating the assignment given the cost matrix

Assignment3.m: Program which performs Hungarian algorithm and generates the vtk file for both the shapes corresponding to the weights found by Hungarian algorithm.

Euclidean_Distance.m:

Function Arguments:

Input: Matrix of points between which the Euclidean Distance has to be calculated.

Output: Euclidean Distance

Function Implementation:

- Initialize the Euclidean distance matrix to zero with size rows of 1st coordinate matrix x size of 2nd coordinate matrix
- Take the sum of square of difference of distances between each coordinate and store it in their corresponding position

Hungarian.m:

Function Arguments:

Input: Cost matrix containing the cost corresponding to each point for all the points

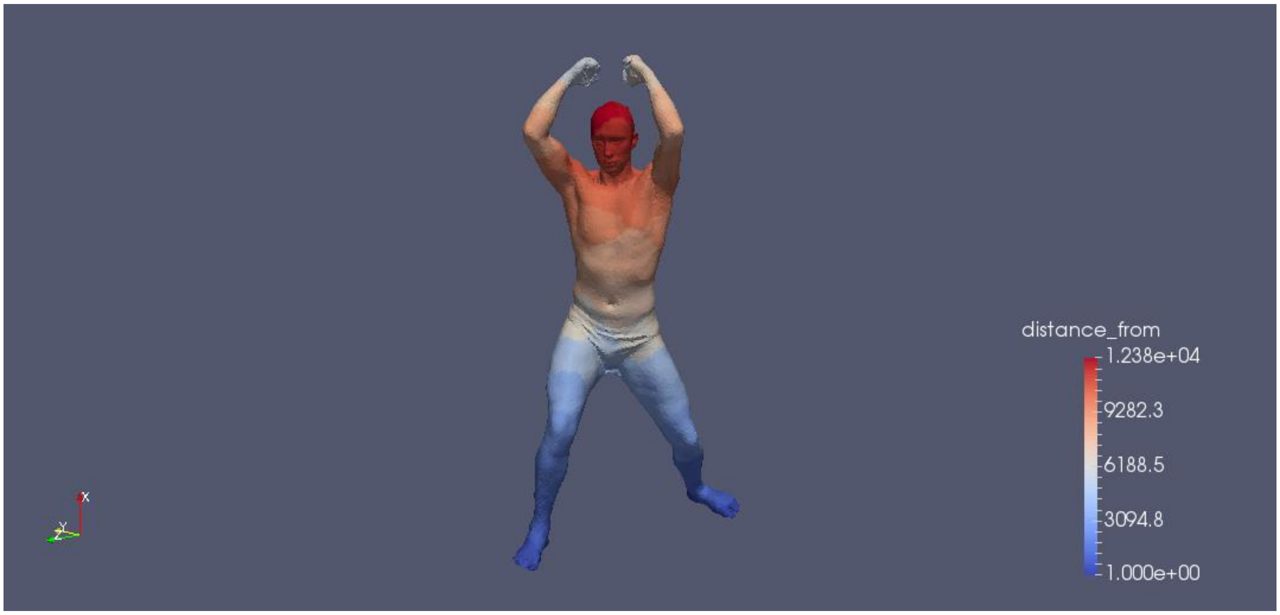
Output: Assignment for each point

Running the Program:

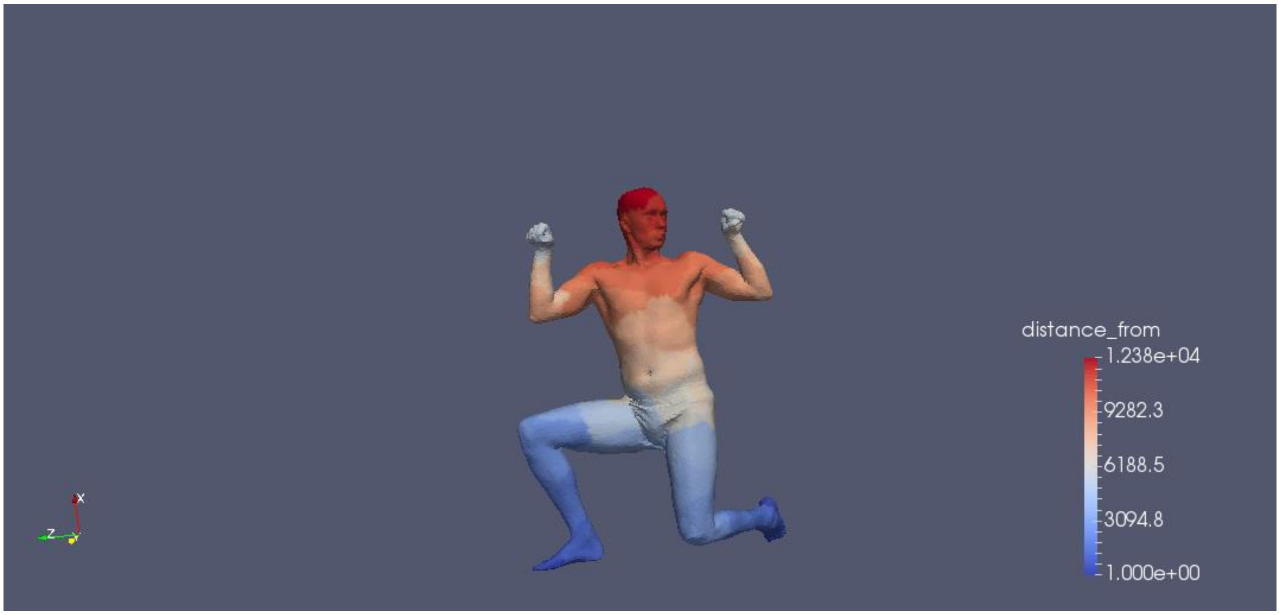
- Run **Assignment3.m** to generate .vtk files for both the shapes.
- Open Paraview to check both the shapes.

Experimental Results:

Original Shape



Test Shape:



Conclusion:

3D Shape registration was successfully performed using Hungarian Algorithm.