Computer Vision Assignment Eight @ ETH Zurich

Shape Context

Ossama Ahmed 18-936-872

December 6, 2018

1 Shape Matching Overview

We match two shapes using a descriptor called the shape context descriptor. We are given sampled points from the template contour where we want to match sampled points from the target contour. The algorithm that was implemented in this part can be summarized as follows:

a) Compute shape context descriptors for the points from both sets, the template and the target contour.

b) Estimate the cost matrix between the two sets of descriptors.

c) Use the cost matrix to solve the correspondence problem between the two sets of descriptors, finding the one-to-one matching that minimizes the total cost (e.g. with the provided Hungarian algorithm).

d) Use the solution of the correspondence problem to estimate a transformation from template to target points (e.g. with Thin Plate Splines) and perform this transformation on the template points.

e) Iterate steps (a-d).

(a) **(b)**

Figure 1: We need to match the first image to the second image given the "shape context".

2 Shape Context Descriptors

A shape context descriptor is calculated for each point which is basically a bivariate histogram that describes the given point compared to the other points through assigning each other point to one of the bins. The bins are calculated according to the radial log euclidean distances and the relative angle. We limit the other points that should affect the descriptor of the point's descriptor through having a small radius and a large one.

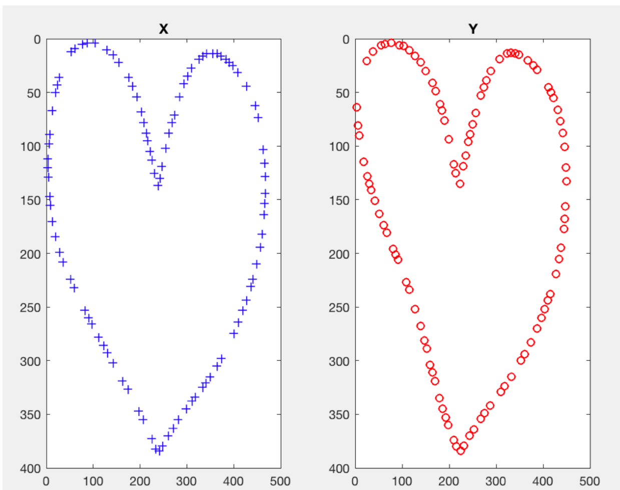

Figure 2: Sampled points from the two shapes shown above.

3 Cost Matrix

The cost matrix is calculated for a set of descriptors of two different shapes. For each two descriptors in two different shapes, the cost is caclulated according to the chi2 cost. One thing to mention is that sometimes if the the descriptors compared have no points in the same bin; it leads to a division by zero which is subsequently dealt with by setting the cost of these two descriptors as zero.

4 Hungarian Algorithm

The hungarian algorithm takes a square cost matrix of initial guesses of how two descriptors match and subsequently outputs a one-to-one correspondence between descriptors; which is the assignment between descriptors that minimizes the total cost.

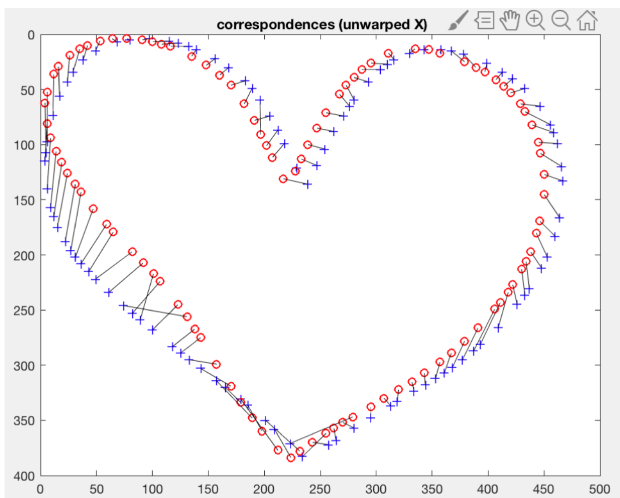

Figure 3: Correspondences from the hungarian algorithm unwarped.

5 Thin Plate Splines

Given the correspondence descriptors/points between the two shapes; we look for a tranform function $T : \mathbb{R} \rightarrow \mathbb{R}$. Which maps points from the first shape to points in the second shape. This is done through using an independent thin plate spline models for each coordinate seperatly. Applying this process iteratively improves the result. The metric of similarity chosen here is the binding energy between two shapes matching.

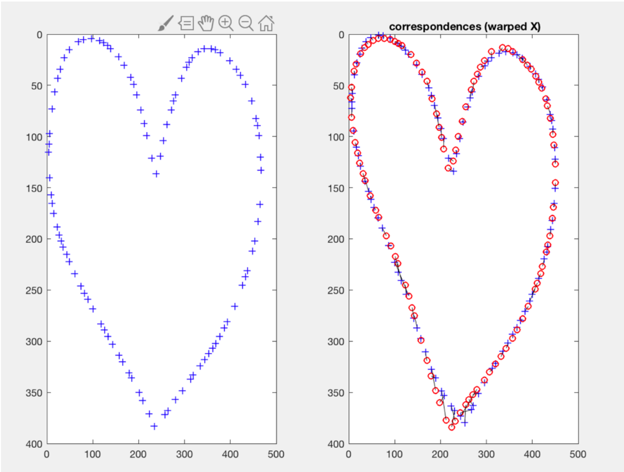

Figure 4: Correspondences from the hungarian algorithm warped.

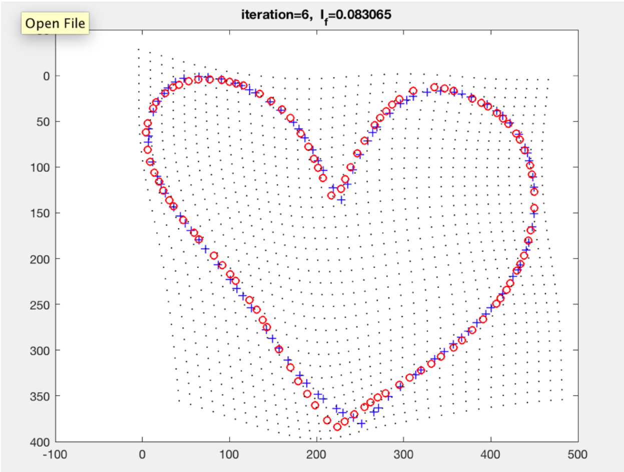

Figure 5: Resulting Energy after 6 iterations

6 Shape Matching

After training, we get a $15 * 15$ cost matrix for the provided dataset.
Note: we sample 100 samples randomly in each shape withoput replacement.
Note 2: the cost matrix is not symmetric given the way we calculate the energy cost between two shapes which is not a symmetric function.

7 Nearest Nerighbor Classifier

Classification is done through running a nearest neighbor classifier by taking all costs and sorting them and then taking the highest frequency of classes in the first k samples. As shown in the graph below, the larger the k, the lower the accuracy of our classifier which was expected since we only consider local neighborhood for the classification.

8 Is the shape context descriptor scale-invariant?

In the sc compute function when calculating the bivariate histogram, the lengths of the radii is normalized using the average distance between all points. Therefore, scaling the shape by a factor k doesn't affect the shape descriptors, making the shape descriptor more robust through normalization.

9 Comparison of sampling methods

The sampling method that I implemented has high variance between runs, however one thing was noticed using the provided sampling method is the low variance (given limited experimentation). Another thing to note that it lead to higher accuracy when k was low but lower accuracy when k was high. (see figure 6). The provided sampling method gives a more complete representation of the shape, making sure to sample probable corresponding points from the shape; which leads to the high classification accuracy.

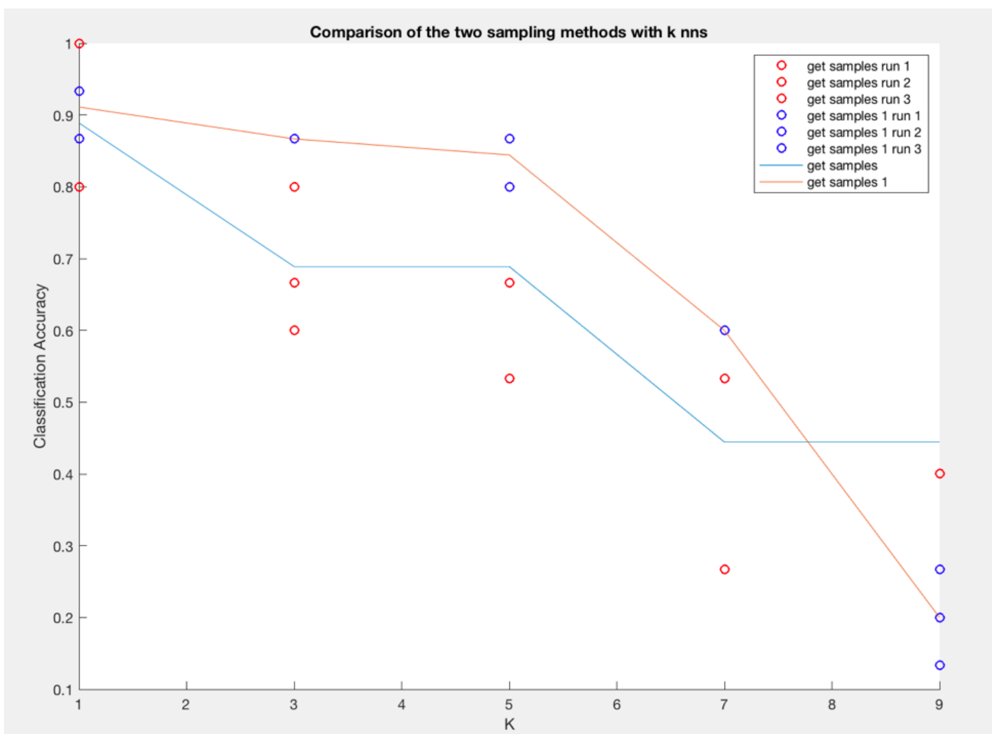

Figure 6: Classification comparison between different sampling methods.