Report: Shape Context

Rishabh Singh: 19-953-793

November 29, 2019

1 Shape Context Descriptors

The first task in any shape matching algorithm is to compute the distinguishing features over which matching needs to be done. We are calculating the shape context descriptors from template and target contours in order to find optimal match between them. In order to do so, we first sample equal number of points from the contours of template and target shapes. The *get_samples()* function randomly draws N points from the given contour. The value of N can be kept at a fixed number, say 100 or can be kept as the minimum number of samples points between the contours to be matched. The next step is to extract the descriptors which will be matched in order to find correspondences between the given shapes. We first compute the pair-wise distance between all the points sampled from the given contour. Next we calculate the value of angle between the given pairs of points. We normalise the radial distance by the mean of distances between all the pairs of points in the shape in order to increase the robustness of the computed descriptors. Based on the smallest and largest permissible values of the radial distance as provided in the framework, we create bins of histogram of radial distance. For a given point, every other point is added to a particular bin based on its distance to the point in question. All the other points are filtered out. Similarly, we define bins for angles by dividing the range *[0,2pi]* in the number of partitions as specified in the framework. The descriptor computed at a given point is a vector of size *number of radial bins times the number of angular bins*. All the other points which lie at a particular radial distance and at a particular angle is added to the corresponding bin. This step is repeated for all the points on the contour. The final shape descriptor computed for a given shape is of size *number of sample points x number of histogram bins*, where *number of histogram bins* is the product of *number of radial and angular bins.*

2 Cost Matrix

The next step is to compute the cost of matching two sets of points based on their shape descriptors. We normalise the descriptors extracted from the given shapes. We then calculate the chi-square statistics between the descriptors as the cost of matching. The detailed implementation can be found in the function *chi_cost*(). This cost matrix is minimised by the hungarian algorithm to find the set of correspondences between the sample points in the two given contours.

3 Thin Plate Spline

We utilise the point correspondences obtained in the previous step to obtain a plane transformation mapping every point from one shape to another. We learn two separate TPS models for capturing horizontal and vertical transformations respectively. Solving the linear equations as stated in the framework, we obtain w_x and w_y which are further used in transforming the points. These also help us in calculating the bending energy of the transformation. This gives us an idea of the cost of shape matching.

4 Scale Invariance of Descriptors

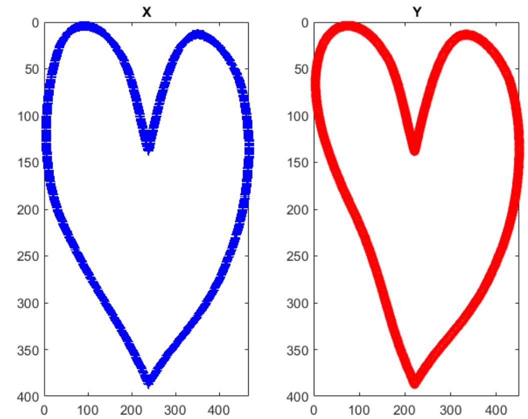
Shape context descriptors are scale invariant since we have normalised all the radial distances by the mean distance between all the point pairs in the shape.

5 Results

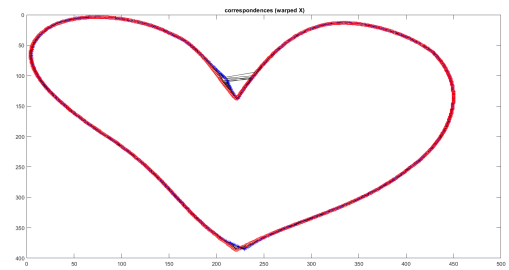
The shape matching obtained for different pairs of shapes, for each object in the dataset and for two different sampling techniques (min sampling vs fixed sampling) are shown in Fig. 1-9.

6 Discussion

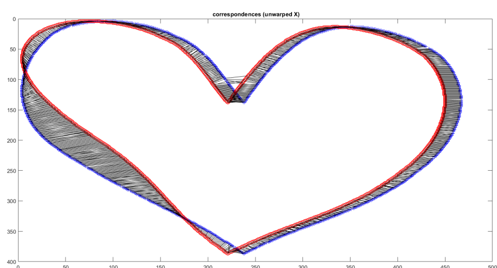
- Shape Descriptors is rich descriptor. The histogram captures the distribution over relative positions. This makes it robust, compact and a highly discriminative descriptor.
- Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as origin.
- Correspondences are obtained using bipartite matching using chi-square distance between the histograms as the cost of matching.
- Uniform bins in log-polar space are used to create the histograms so that the resultant descriptor is more sensitive to the points lying nearby than those far away.
- Shape context is translation-invariant since all the distance and angle calculation is done with respect to the points on the shape.
- Shape context is scale invariant due to normalization of radial distance.



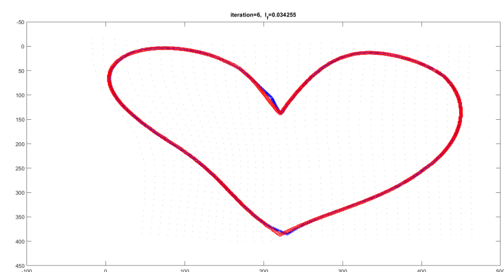
(a) Template and Target shapes



(b) matching (warped)



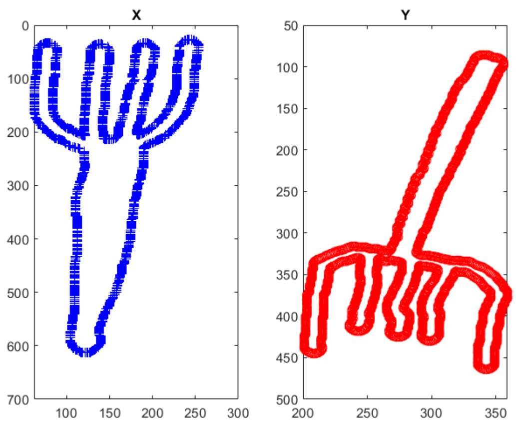
(c) matching (unwarped)



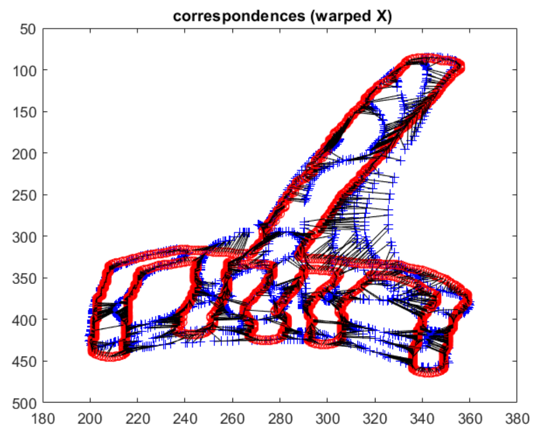
(d) Superimposed warped and target shapes

Figure 1: Shape matching between image 1 and 2 of heart templates.

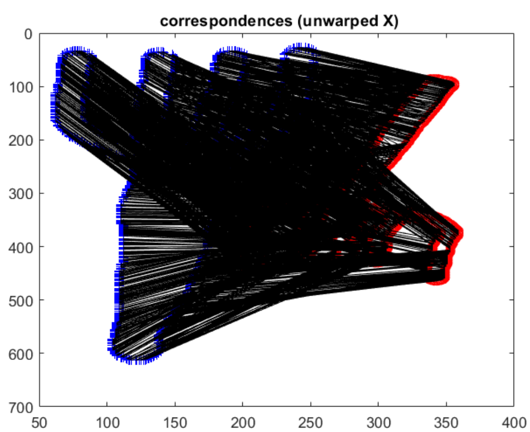
- In Fig.2, fork image 1 is matched with fork image 2, which is rotated by roughly 180 degrees. The bending energy obtained is approx. 7 units. In Fig.6, rows 1-3, fork image 1 is matched with fork images 3-5, which are roughly having the same orientation. The bending energy obtained is less than 1 in each case. This shows that the performance of shape context descriptors deteriorate when objects matched undergo rotation. Shape descriptors are quite sensitive to rotation.
- The bending energy depends on the number of sample points chosen for matching. It has lesser magnitude when 100 points were matched in both the shapes vs. when min samples between both the shapes were chosen.
- The regularisation parameter lambda in the TPS model controls the amount of smoothing and makes the transformation roust to noise in the values.
- The whole pipeline was iterated over 6 times since the initial estimate of the correspondences are usually erroneous and can interfere in the performance of the TPS model learnt.



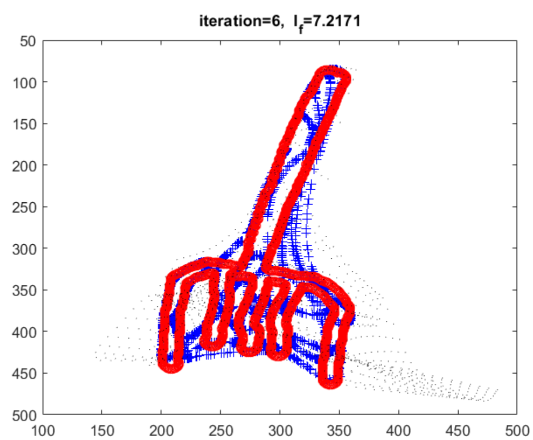
(a) Template and Target shapes



(b) matching (warped)

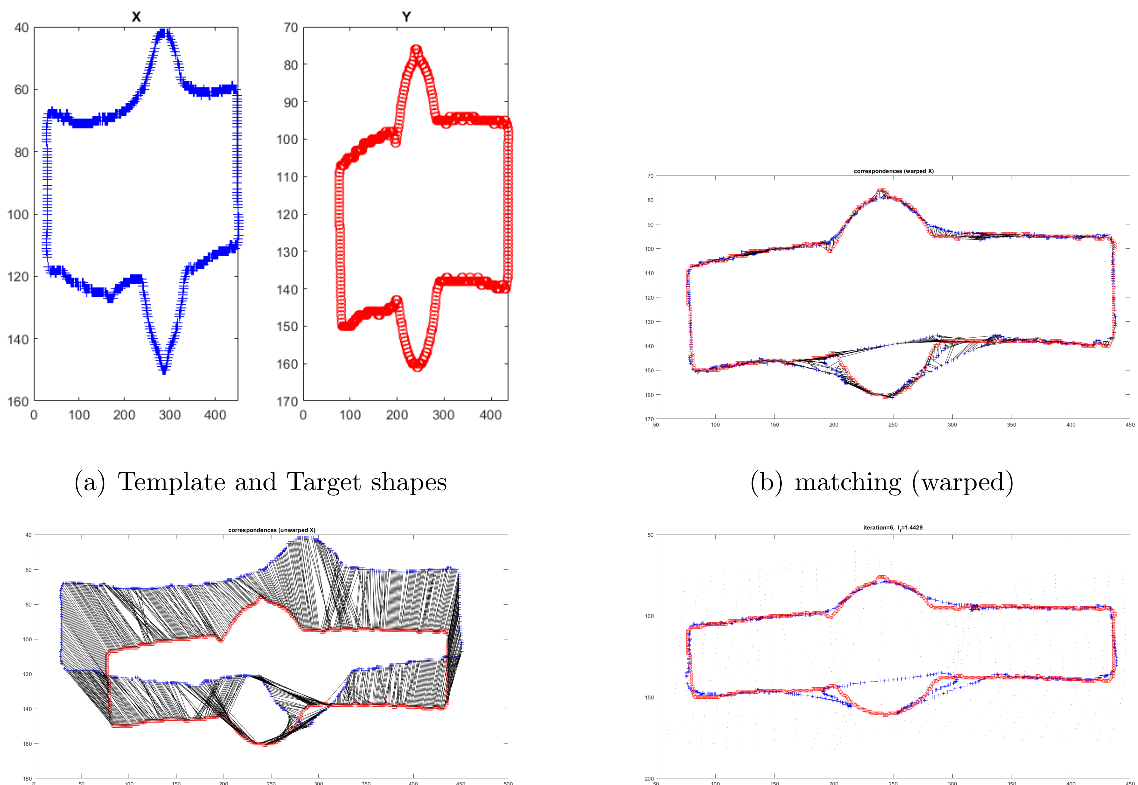


(c) matching (unwarped)



(d) Superimposed warped and target shapes

Figure 2: Shape matching between image 1 and 2 of fork templates.



(a) Template and Target shapes

(b) matching (warped)

(c) matching (unwarped)

(d) Superimposed warped and target shapes

Figure 3: Shape matching between image 1 and 2 of watch templates.

Figure 4: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is the minimum of the samples obtained from each image.

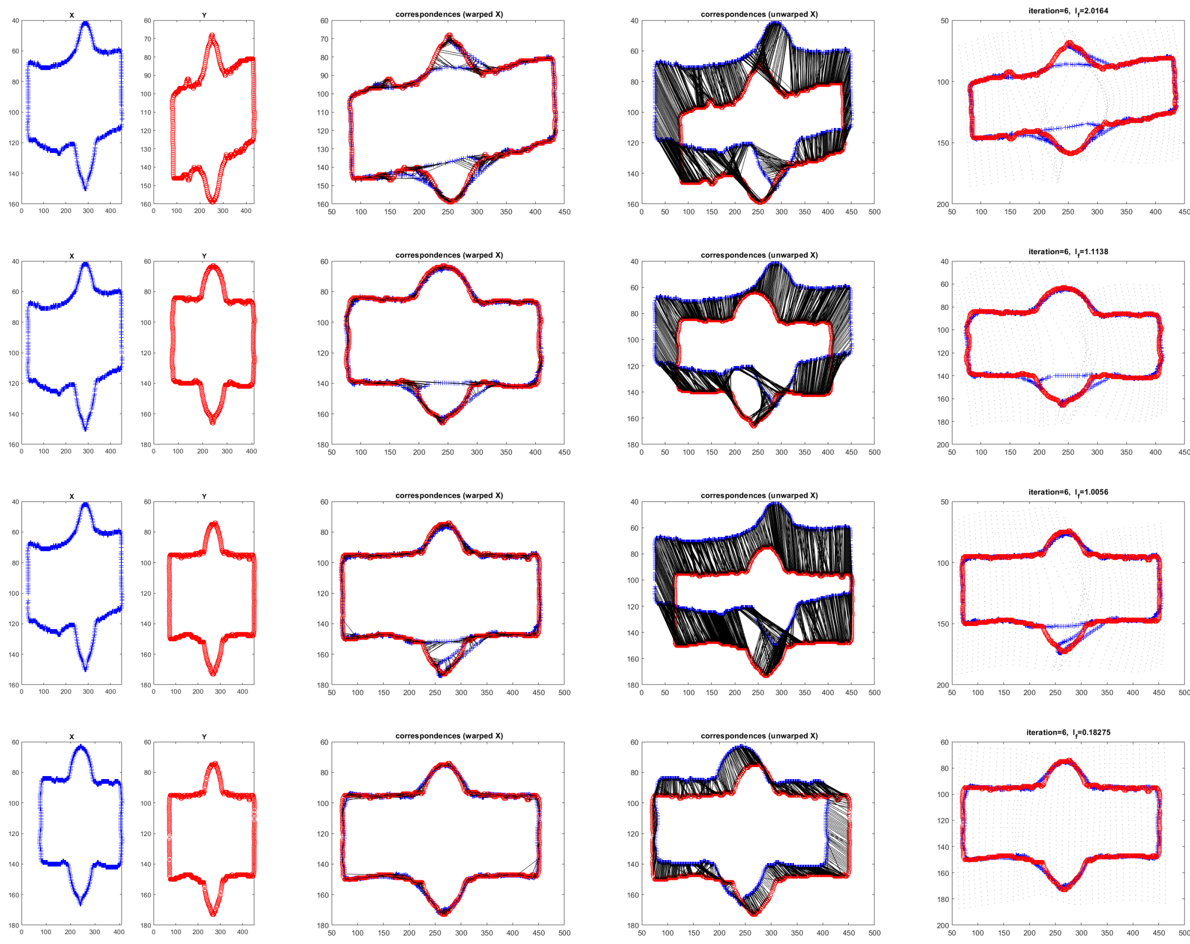


Figure 5: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is the minimum of the samples obtained from each image.

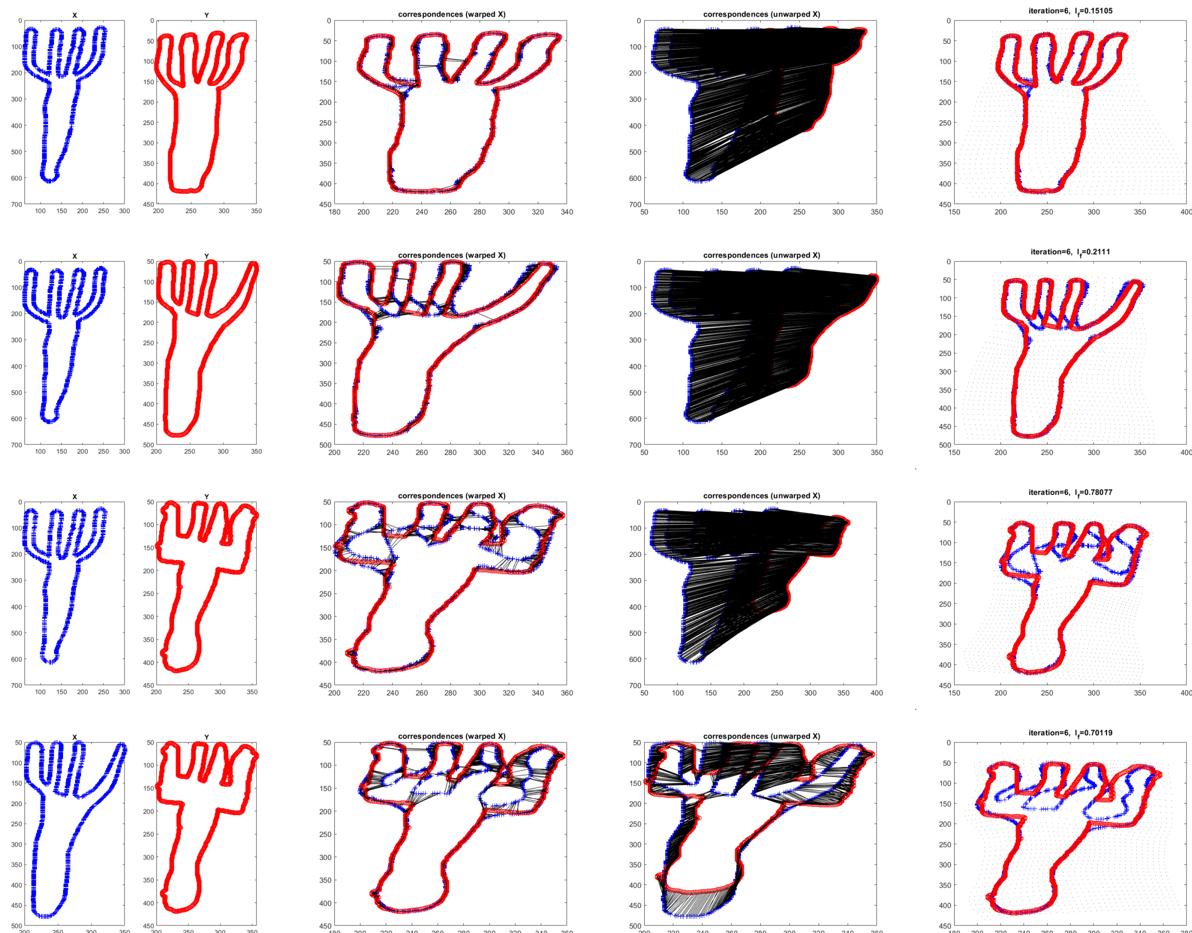


Figure 6: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is the minimum of the samples obtained from each image.

Figure 7: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is in each image is 100.

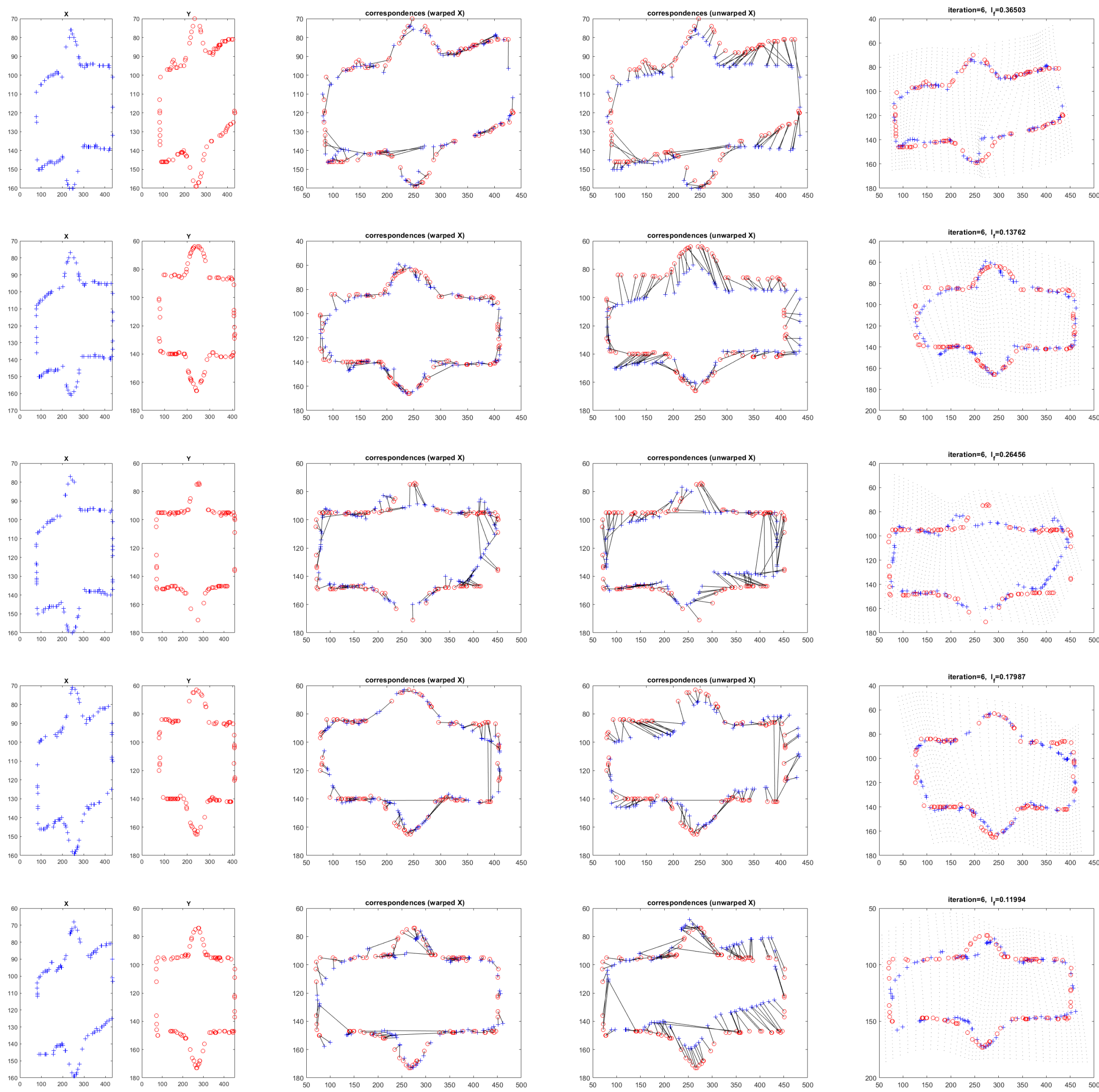


Figure 8: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is in each image is 100.

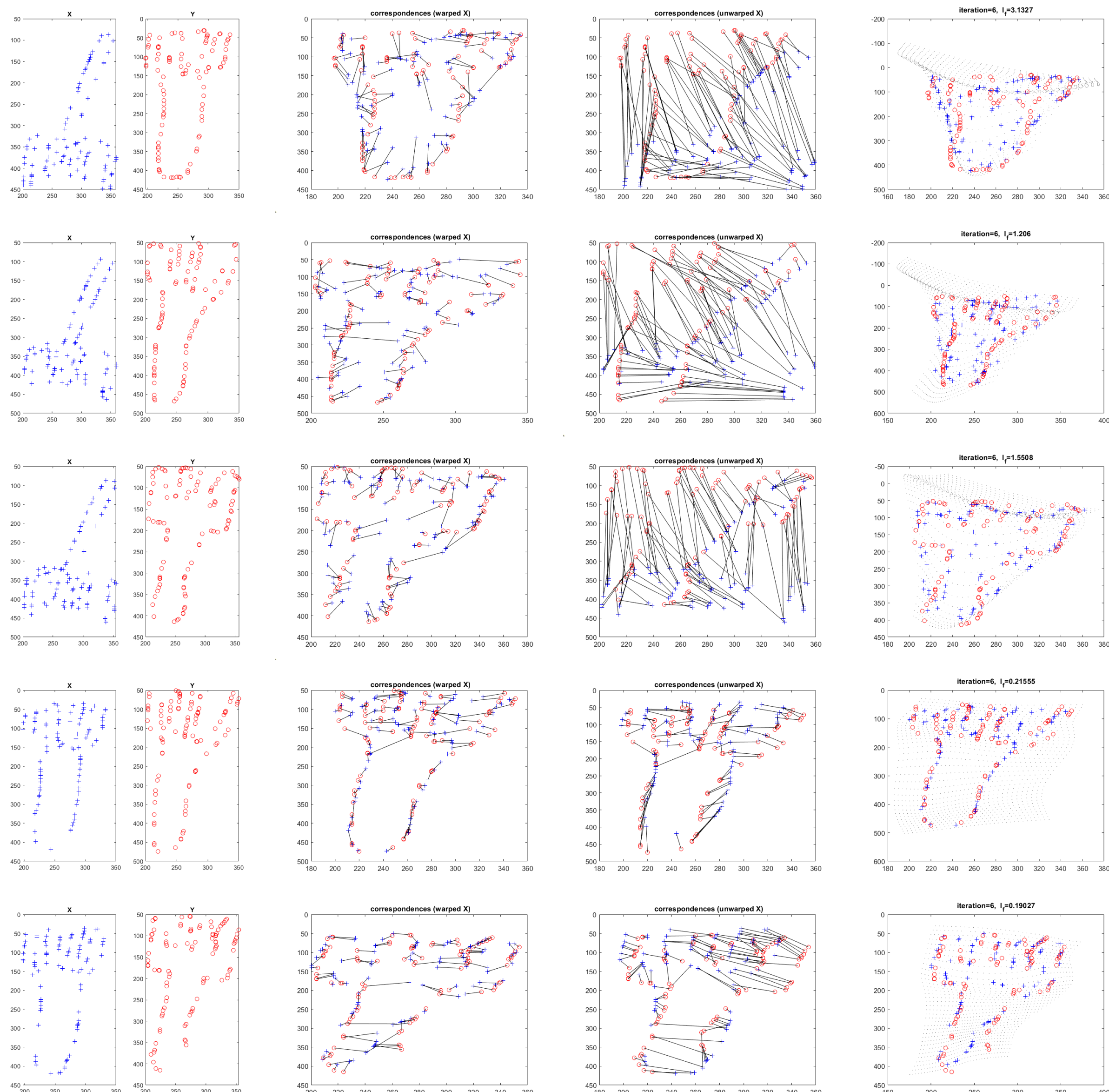


Figure 9: (from left to right per row) source and target images, matching of warped source, matching of unwarped source, TPS source and target images. Each row is a different template pair. The number of sample points is in each image is 100.