IT575 Computational Shape Modeling **Assignment - 1**

Name - Vaibhav Amit Patel
ID No. 201401222
Name - Maharshi Vyas
ID No. 201401414

Date of Submission: 19 Jan 2017

1. Show that the full Procrustes distance d_F is a metric on the shape space. Assume that $d([p],[q]) = \cos^{-1}(|\langle p_c, q_c\rangle|)$, where p_c ,q_c denotes the representation of any landmark configuration in the equivalence class $[p]$, $[q]$ respectively as a vector in $\mathbb{C}^n$, is a valid metric on the shape space.

   Answer : Let's denote the shape space to be V. Now, any metric on this shape space V is a function $d\colon V \times V \to \big[0,\infty\big]$. Where $\big[0,\infty\big]$ is the set of non-negative real numbers. d should satisfy following conditions:

   (a) $d\big(x,y\big) \geq 0$.
   (b) $d\big(x,y\big) = 0 \iff x = y$.
   (c) $d\big(x,y\big) = d\big(y,x\big)$.
   (d) $d\big(x,z\big) \leq d\big(x,y\big) + d\big(y,z\big)$.

   We will show that full Procrustes distance satisfy all of these conditions one by one.
   Assuming Kendall shape distance,

$$\theta = d([p],[q]) = \cos^{-1}(|\langle p_c, q_c\rangle|)$$

   Full Procrustes Distance, $d([p],[q]) = \sin\theta$

   (a) $d\big(x,y\big) \geq 0$.

$$d([p],[q]) = \sin(\cos^{-1}(|\langle p_c, q_c\rangle|))$$

   Range of $\cos^{-1}$ is $\big[0,\pi/2\big]$.Sine is nonegative function in first quadrant. So the metric d satisfies this condition.

   (b) $d\big(x,y\big) = 0 \iff x = y$.

$$d_F([p],[q]) = 0$$
$$sin(\cos^{-1}(|\langle p_c, q_c\rangle|)) = 0$$

   Because Sine has value 0 at 0,

$$\cos^{-1}(|\langle p_c, q_c\rangle|) = 0$$
$$e^{i\theta} = 1$$
$$\theta = 0$$

   $\implies$ $[p]$ and $[q]$ are in the same direction.

   Now, in shape space $[p]$ and $[q]$ are already made invariant to translation and scaling, the only thing left is rotation. And here, we are getting that between $[p]$ and $[q]$ angle is zero. So, there is only one conclusion to this which is that they are same.

(c) $d(x,y) = d(y,x)$.
It is the angle between the absolute value of the inner product.

$$|\langle p_c, q_c \rangle| = |\langle q_c, p_c \rangle|$$
$$\cos^{-1}(|\langle p_c, q_c \rangle|) = \cos^{-1}(|\langle q_c, p_c \rangle|)$$

Because, sine is a bijective function in first quadrant

$$\sin(\cos^{-1}(|\langle p_c, q_c \rangle|)) = \sin(\cos^{-1}(|\langle q_c, p_c \rangle|))$$

$$d_F([p],[q]) = d_F([q],[p])$$

(d) $d(x,z) \leq d(x,y) + d(y,z)$
Assuming, Kendall shape distance is a valid metric on the shape space.

$$\cos^{-1}(|\langle p_c, q_c \rangle|) \geq \cos^{-1}(|\langle p_c, r_c \rangle|) + \cos^{-1}(|\langle r_c, q_c \rangle|)$$

Assuming a θ for each distance,

$$\theta_3 \geq \theta_1 + \theta_2$$

Because sine is strictly increasing in the first quadrant,

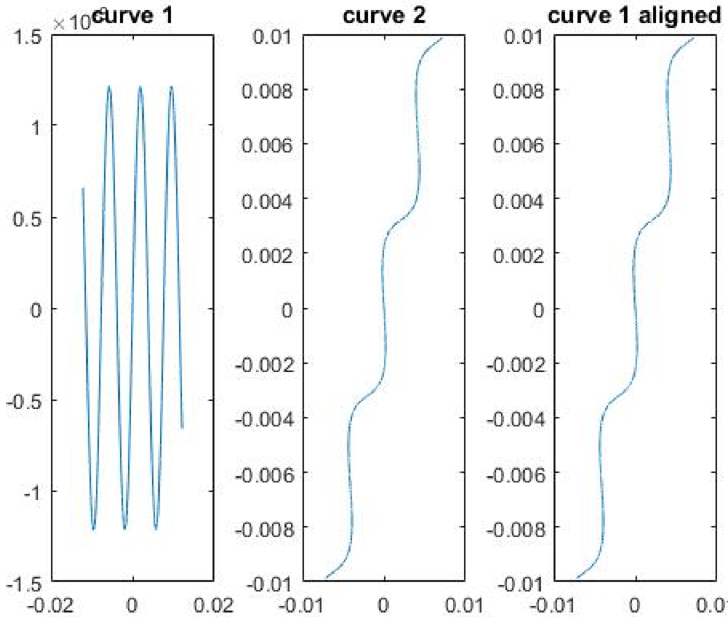
$$\sin\theta_3 \geq \sin(\theta_1 + \theta_2)$$
$$\sin\theta_3 \geq \sin\theta_1 \cos\theta_2 + \cos\theta_1 \; sin\theta_2$$

Now, $\cos\theta_2$,$\cos\theta_1 \leq 1$ because θ_1, θ_2 are in first quadrant. Hence,

$$\sin\theta_3 \geq \sin\theta_1 + \sin\theta_2$$

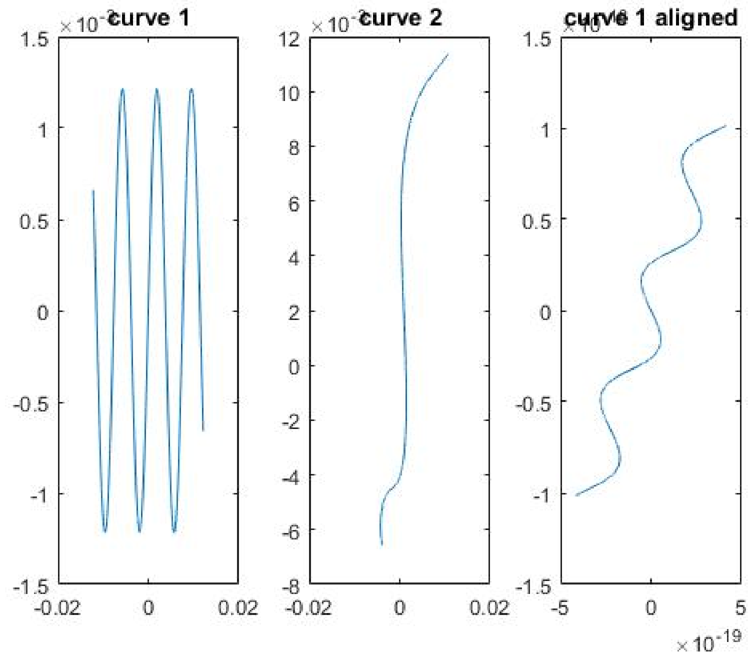
$$\sin(\cos^{-1}(|\langle p_c, q_c \rangle|)) \geq \sin(\cos^{-1}(|\langle p_c, r_c \rangle|)) + \sin(\cos^{-1}(|\langle r_c, q_c) \rangle|)$$
$$d_F([p],[q]) \geq d_F([p],[r]) + d_F([r],[q])$$

2. Write a *MATLAB* function *myProcrustesAlign.m*, to be used as $pa = myProcrustesAlign(p, q)$ that will align the set of landmarks p to landmarks in q. The inputs p, q and output pa should be $k \times 2$ matrices, where k is the number of landmarks.



We have used 20000 landmarks for representing a sin curve (curve 1). After this we have scaled, rotated and translated the image (curve2). But as you can see, after the alignment curve 1 looks exactly like the curve 2.

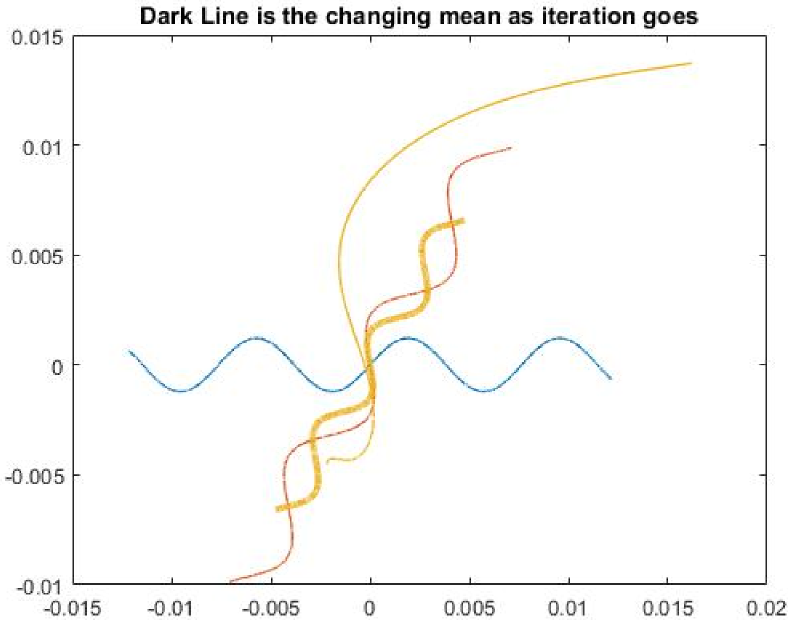
Figure 1: **Procrustes alignment for the same shapes**



We have used 20000 landmarks for representing a sin curve (curve 1). After this we have used scaling, rotation, translation and another non invariant transformation on the image (curve2). Here you can see that after the alignment curve 1 tries to look like curve 2 , but it fails do so.

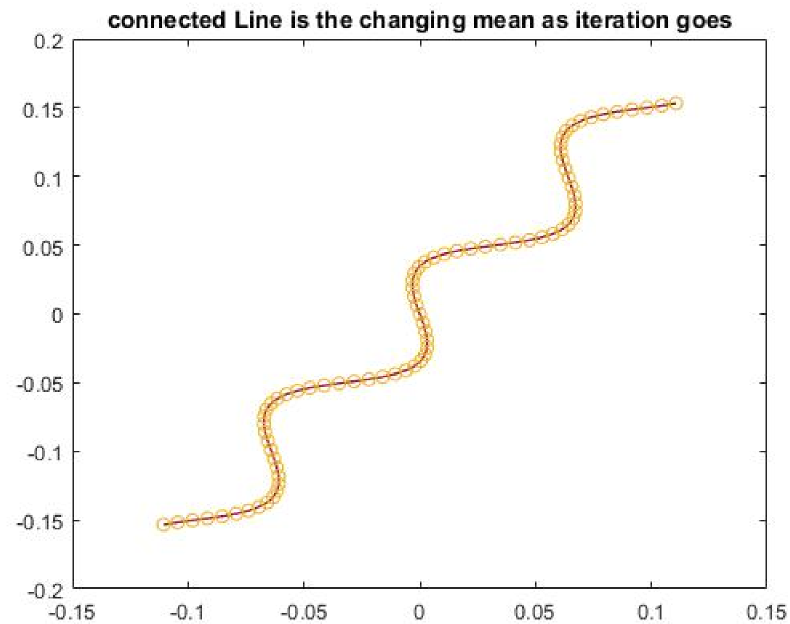
Figure 2: **Procrustes alignment for the two different shapes**

3. Write a $MATLAB$ function $myProcrustesAlign.m$, to be used as $mnshp = myProcrustesMean(p,q)$ that outputs the mean shape as a $k \times 2$ matrix, with the input p being a $k \times 2$ matrix for n objects represented as k landmarks. The x and y coordinates will be assumed to be put in alternating columns of matrix p. The mean shape should be computed using the iterative algorithm mentioned in class, wherein the iterations should stop once the norm of difference of successive mean estimates is below the threshold given as an input in the variable thr.



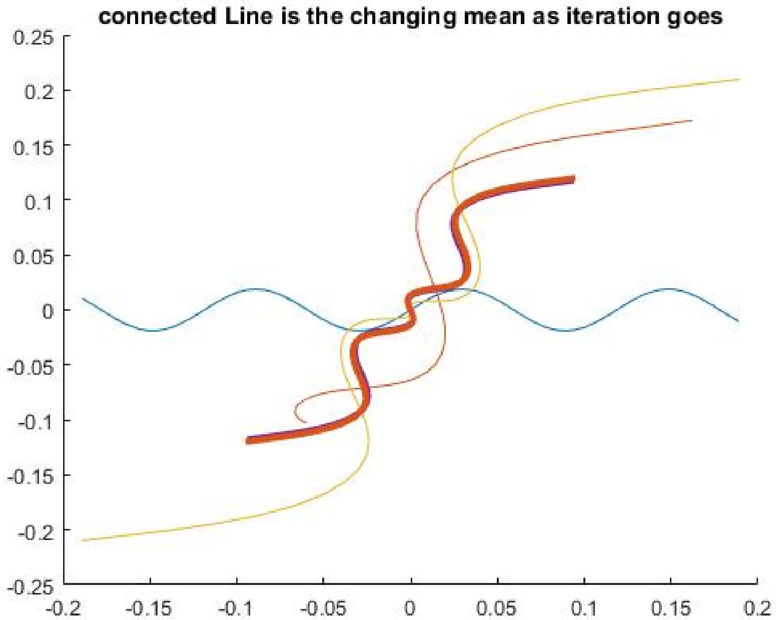
We have used 20000 landmarks for all the three shapes. The thin lines are input shapes and thick line is the mean of this shape. The three shapes were not aligned before plotting on the image. So they do not show the mean's credibility.

Figure 3: **The variations in the landmarks (Invariant transformations)**



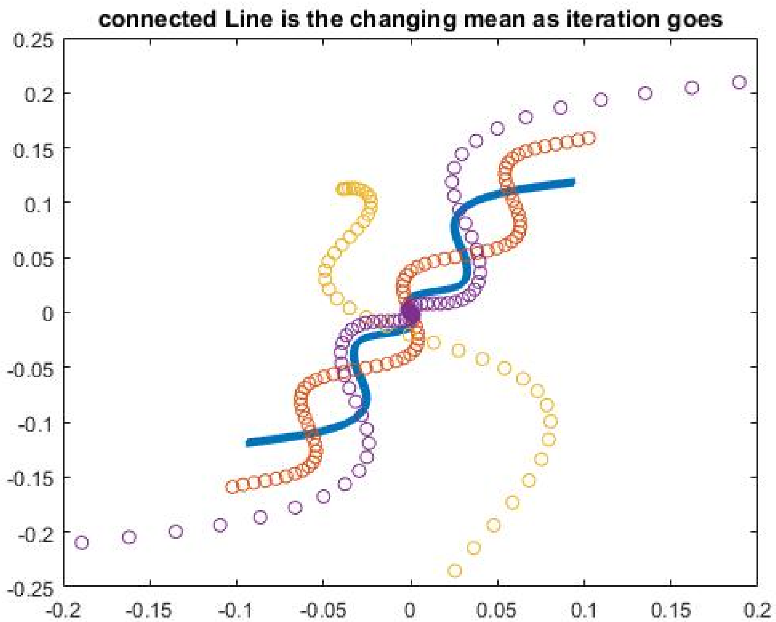
After aligning all the images to the mean shape we can see that there is no difference. We have used 80 landmarks for all the three shapes (shape invariant transformations). The thin lines are input shapes and thick line is the mean of this shape. Now, The three shapes were aligned before plotting on the image. here we can see that the mean actually shows the mean of these shapes (they are essentially the same).

Figure 4: **The variations in the landmarks (Invariant transformations)**



We have used 80 landmarks for all the three shapes. The thin lines are input shapes and thick line is the mean of this shape. The three shapes were not aligned before plotting on the image. So they do not show the mean's credibility.

Figure 5: **The variations in the landmarks (Shape variant transformations)**



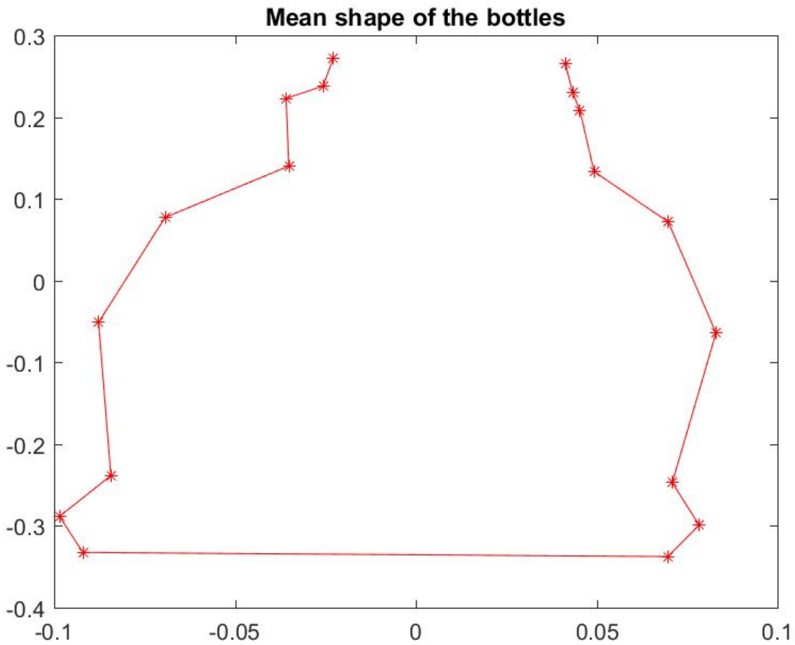
Even after aligning all the images to the mean shape the difference is too much. We have used 80 landmarks for all the three shapes (shape variant transformations). The thin lines are input shapes and thick line is the mean of this shape. Now, The three shapes were aligned before plotting on the image. The transformation are shape variant so the mean shape is not aligned to images.

Figure 6: **The variations in the landmarks (Shape variant transformations)**

4. The shape of coke bottles has gone through a few changes over the years. You will find some representative images of these bottles in the folder bottles. For marking the *landmarkpoints*, use the MATLAB function **getpointsASM.m** . Type **help getpointsASM** to see its usage and related information. Use **imread.m** (available in MATLAB) for reading image files.

   (a) Justify the number of landmarks you will use to compute shape related informa-tion for these bottles. Compute and plot the mean shape of the coke bottle using myProcrustes-Mean.m.

   Ans: We have taken a total of 18 landmark points to fit all the information in the bottles images. Landmarks should be oriented same with the edges in each image. We should consider an image with highest number of curves and edges to count number of landmarks and fit the same landmarks for all the images with corresponding segments.Thus, Shape information of each image is stored is optimize;



We have used 18 landmark points for each image. And we can see that these landmarks represents outline of a bottle very well.

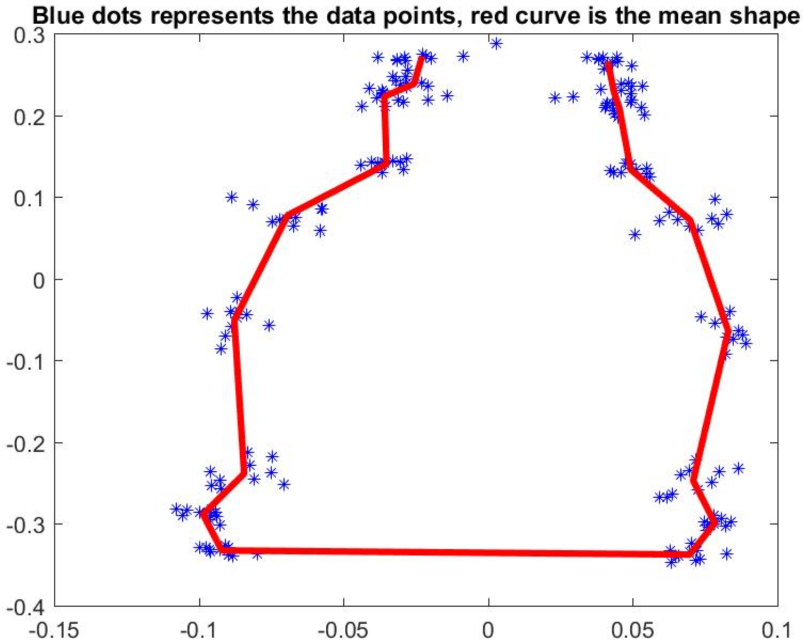
Figure 7: **Mean shape of 10 coke bottles.**

   (b) Compute the shape covariance matrix and plot the eigenvalue profile. Find out the number of eigenvectors required to summarize 95% variance of the entire dataset.

   Ans: Eigenvalue profile is given in figure 11. According to the profile, 5 eigenvectors are required to summarize 95% variance of the entire dataset. Eigen values needed for 95% variance is 5 in coke bottles.

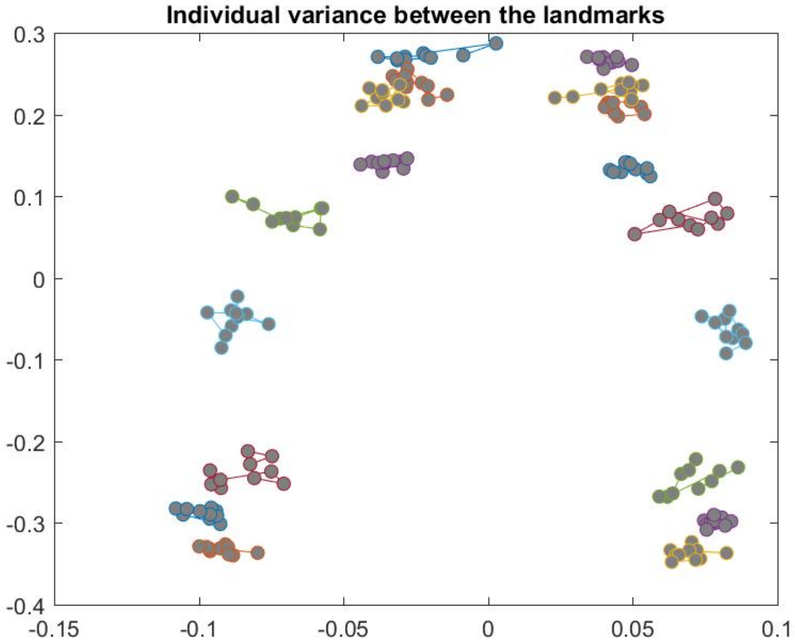
Table 1: **Eigen values for 95% variance**

Eigen value
0.0109
0.0070
0.0026
0.0019
0.0011



Here blue dots are the total 180 points ($18 landmarks \times 10 bottles$) . We can see the change between these landmark points.

Figure 8: **All the landmark points and the mean fitting these points.**



We have used 18 landmark points for each image. Here we can see the actual variation between these landmarks points as the images vary.

Figure 9: **The variations in the landmarks**

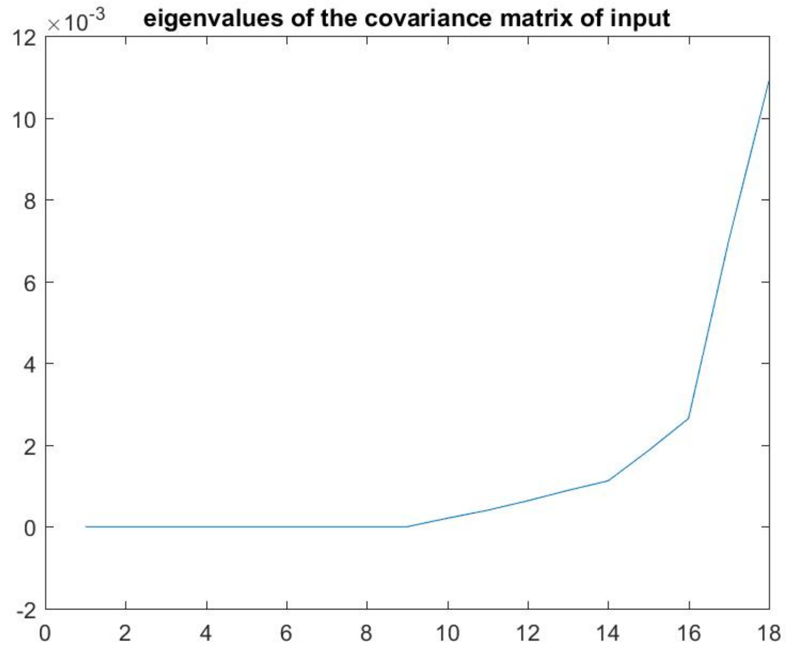


Figure 10: **The eigenvalue profile**

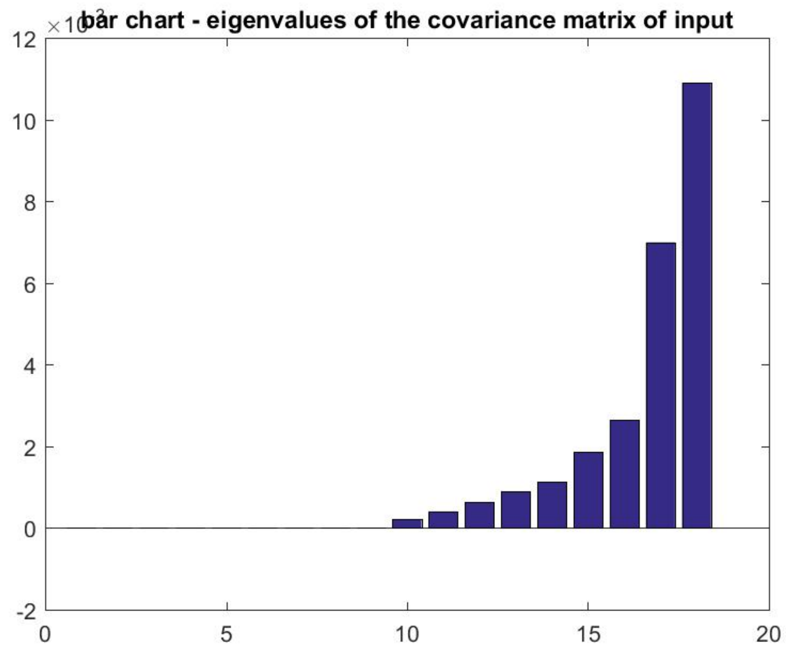
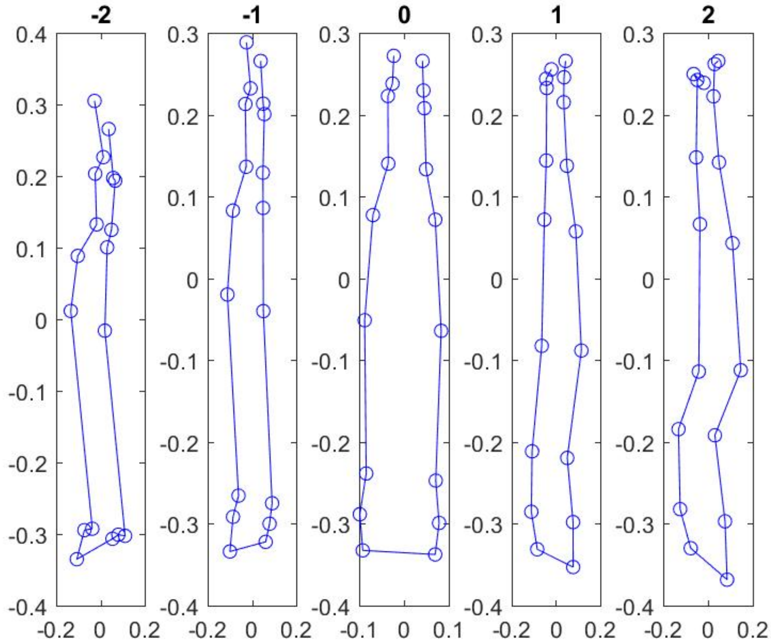


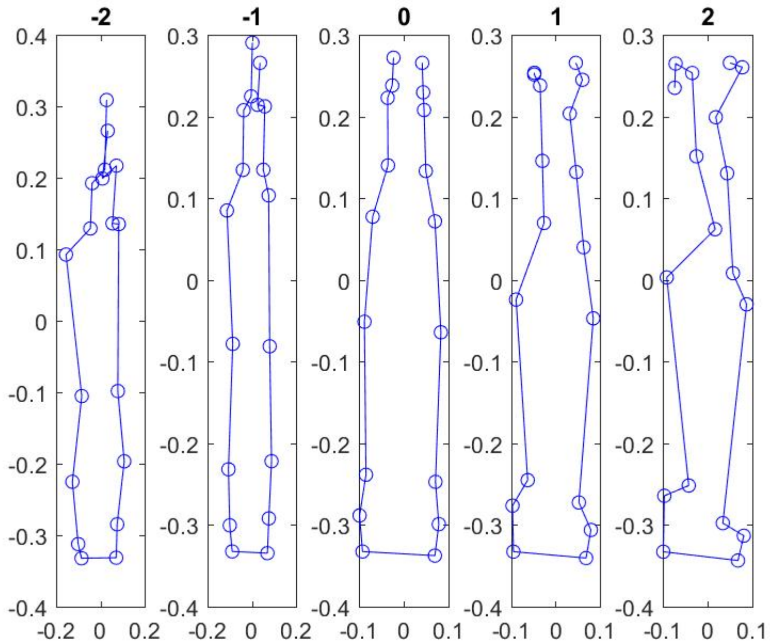
Figure 11: **The eigenvalue profile - bar chart**

(c) Let v 1 and v 2 be the first two principal components(eigen vectors). Plot variations in the mean shape along v_1 and v_2 , i.e, $\mu + k\Delta v_i$, where μ is the mean shape, $k \in [2,1,0,1,2], i = 1,2$, and Δ is an appropriate scalar.
Ans: Figure 9 shows landmark point cloud which shows how far landmarks can vary from its own set of landmark points. Figure 12 and 13 contains images with variations in the mean shape along largest and second largest eigenvector(The direction of highest variance or eigenvalue). As seen from the graph, edge between the main body and the top of the bottle(Area of the neck) has major variations. Also width of the body is affected as we change k.



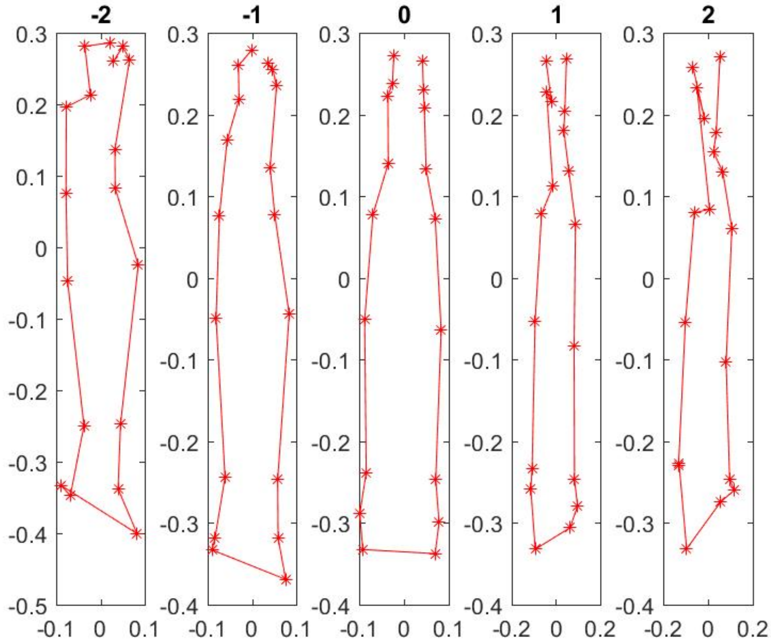
The title is the transformation constant k. We have transformed the mean shape using this formula: $\mu + k\Delta v_1$.

Figure 12: **Variation along Largest "eigenvector" (eigenvalue)**



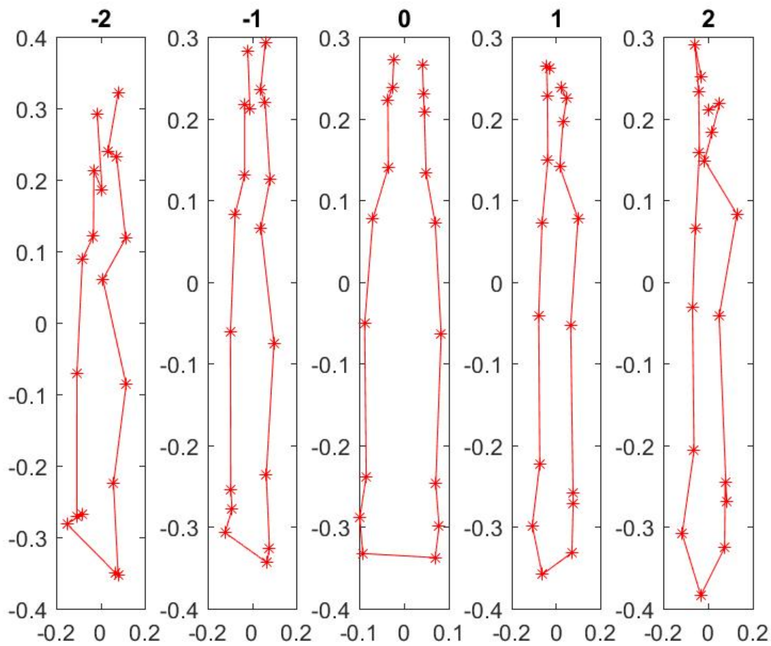
The title is the transformation constant k. We have transformed the mean shape using this formula: $\mu + k\Delta v_2$.

Figure 13: **Variation along second Largest "eigenvector" (eigenvalue)**



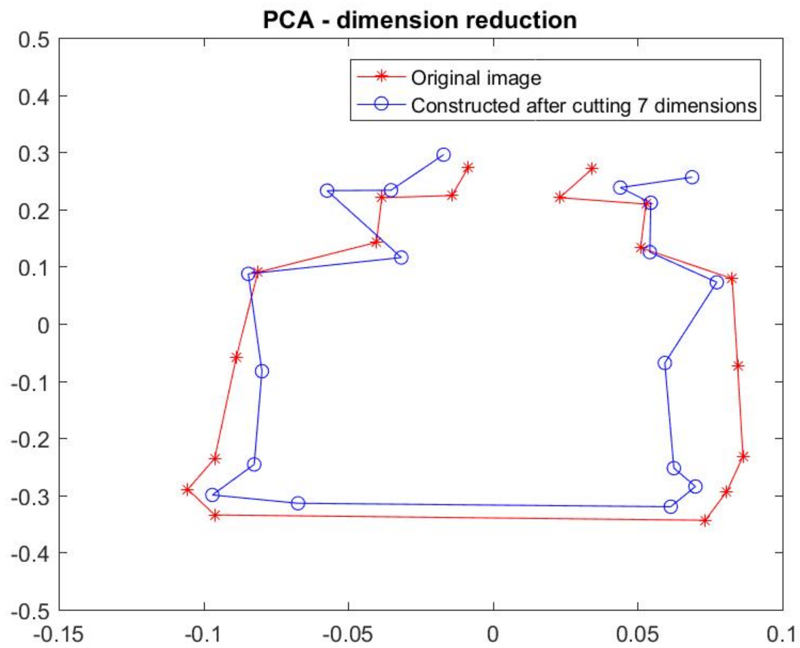
Here, we first converted the mean shape in the different basis (Same as in PCA) and then applied the transformation. After this we reverted back it to the original basis. We have transformed the mean shape using this formula: $\mu + k\Delta v_1$.

Figure 14: **Variation along Largest "eigenvector" (eigenvalue) in the PCA basis**



Here, we first converted the mean shape in the different basis (Same as in PCA) and then applied the transformation. After this we reverted back it to the original basis. We have transformed the mean shape using this formula: $\mu + k\Delta v_2$.

Figure 15: **Variation along second Largest "eigen vector" (eigenvalue) in the PCA basis**

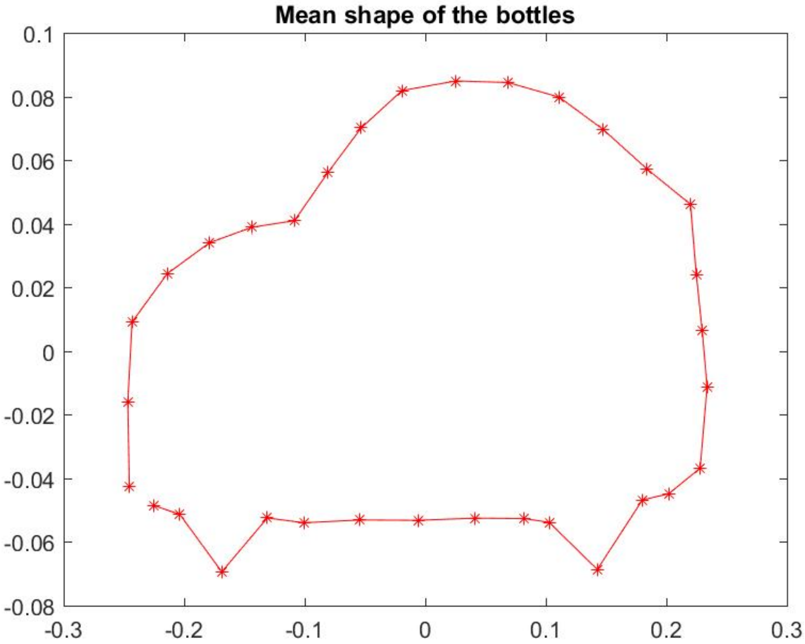


reconstruction of the images

Figure 16: **PCA dimensionality reduction**

5. Repeat the above exercise for car shape using images provided in the folder cars. If you think that only the input changes from bottle landmarks to car landmarks, you are missing something.

   (a) Justify the number of landmarks you will use to compute shape related informa-tion for these cars. Compute and plot the mean shape of the car using myProcrustesMean.m. **Ans:** We have taken a total of 33 landmark points to fit all the information in the bottles images. Landmarks should be oriented same with the edges in each image.We have taken different parts of a car with a set of landmarks. We should consider an image with highest number of curves and edges to count number of landmarks and fit the same landmarks for all the images with corresponding segments.Thus, Shape information of each image is stored and used.



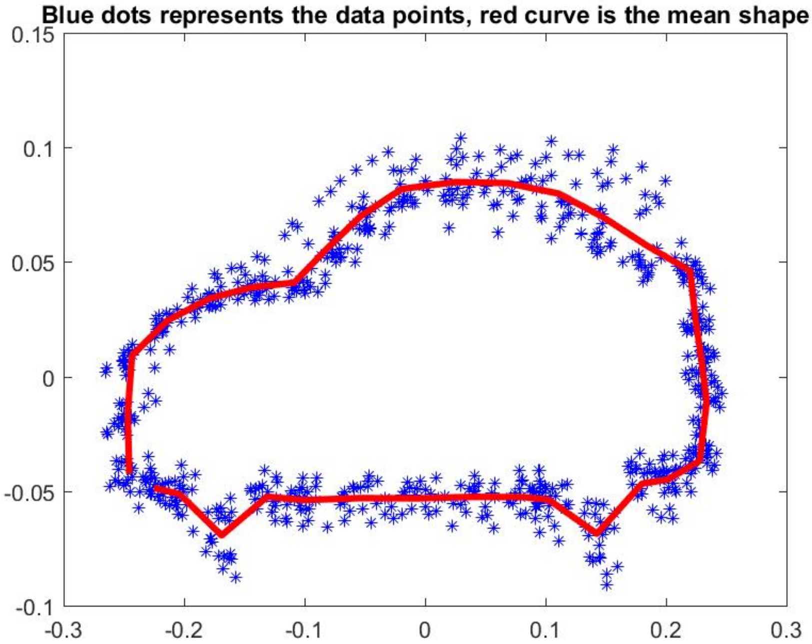
We have used 33 landmark points for each image. And we can see that these landmarks represents outline of a car very well.

Figure 17: **Mean shape of 20 cars.**

(b) Compute the shape covariance matrix and plot the eigenvalue profile. Find out the number of eigenvectors required to summarize 95% variance of the entire dataset.
Ans: Eigenvalue profile is given in figure 21. According to the profile, 6 eigenvectors are required to summarize 95% variance of the entire dataset. Eigen values needed for 95% variance is 6 in cars.

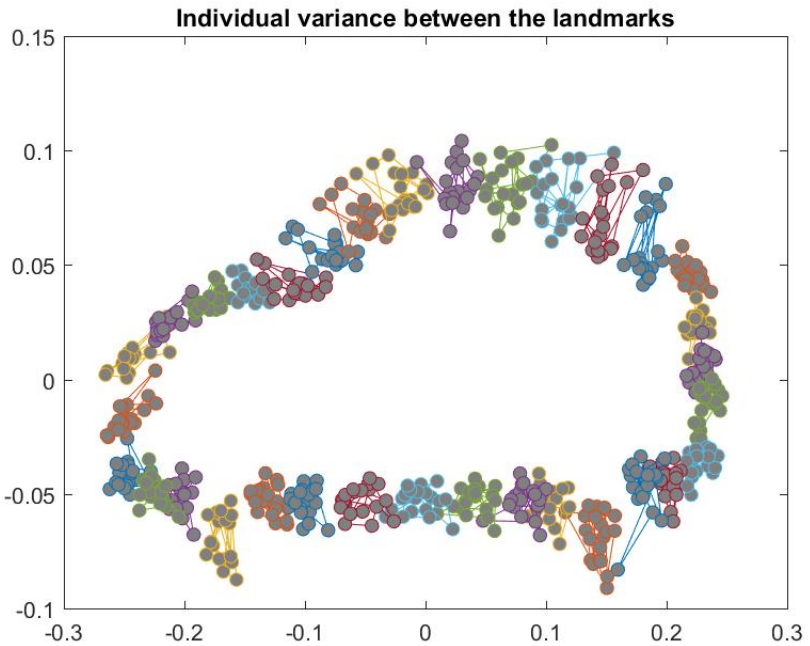
Table 2: **Eigen values for** 95% **variance**

Eigen value
0.0634
0.0175
0.0119
0.0061
0.0048
0.0031



Here blue dots are the total 660 points ($33 landmarks \times 20 cars$) . We can see the change between these landmark points.

Figure 18: **All the landmark points and the mean fitting these points.**



We have used 33 landmark points for each image. Here we can see the actual variation between these landmarks points as the images vary.

Figure 19: **The variations in the landmarks**

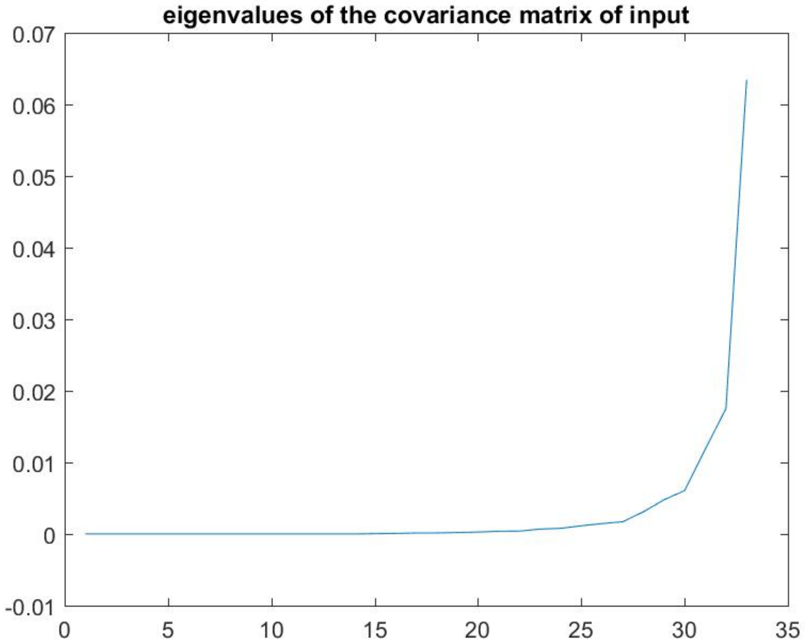


Figure 20: **The eigenvalue profile**

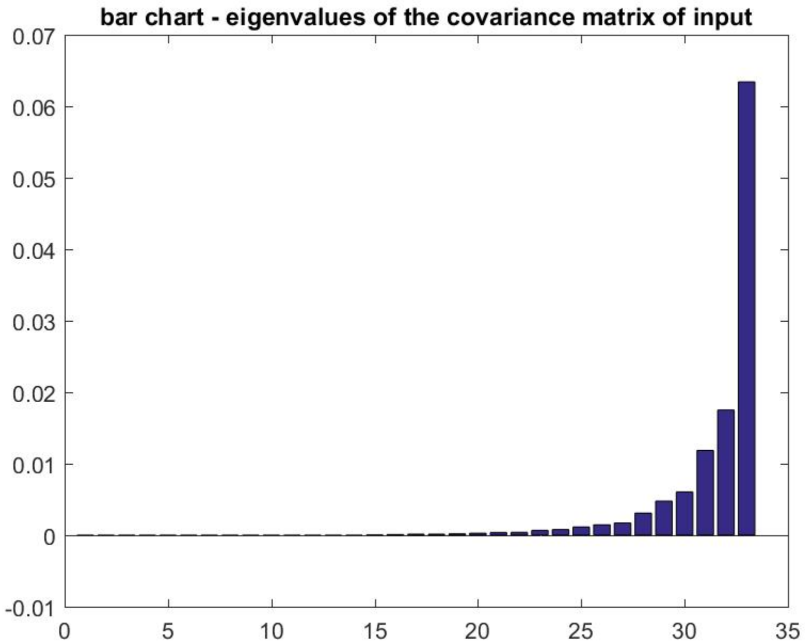
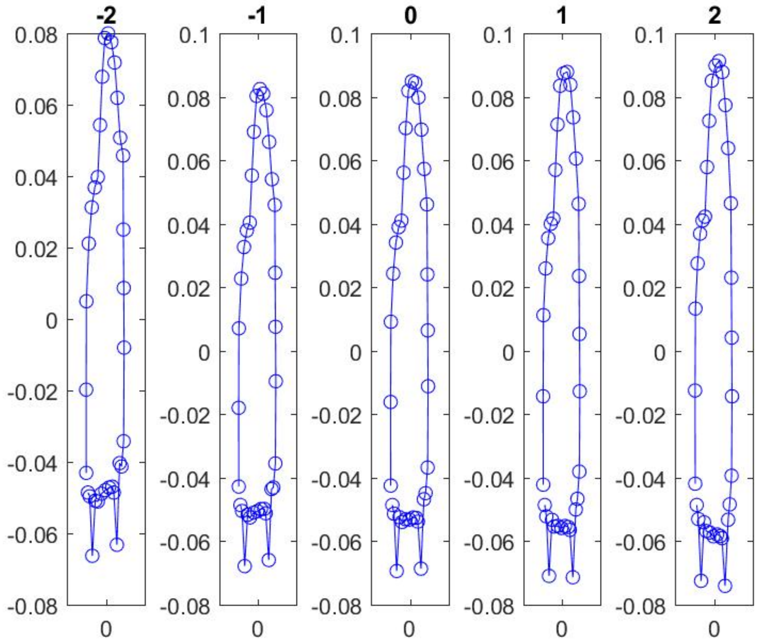


Figure 21: **The eigenvalue profile - bar chart**

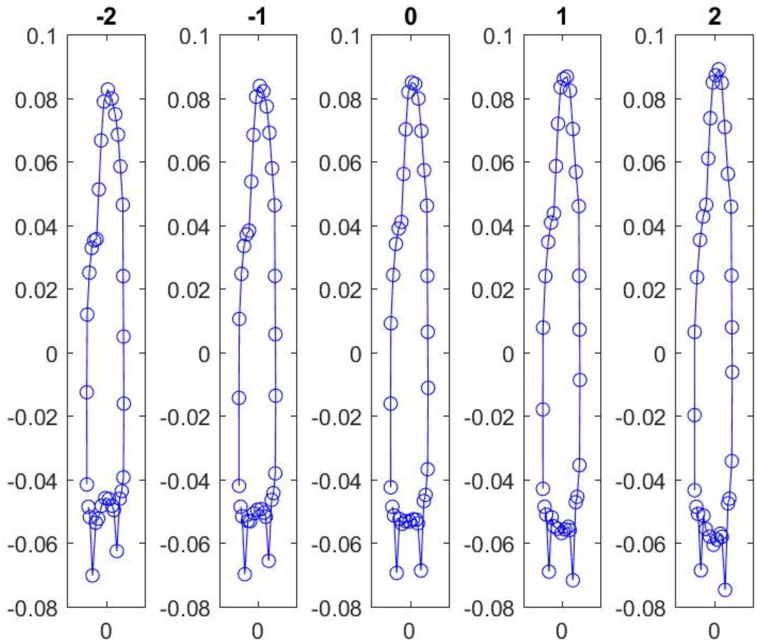
(c) Let v 1 and v 2 be the first two principal components(eigen vectors). Plot variations in the mean shape along v_1 and v_2 , i.e, $\mu + k\Delta v_i$, where μ is the mean shape, $k \in [2, 1, 0, 1, 2], i = 1, 2$, and Δ is an appropriate scalar.
Ans: Figure 22 shows landmark point cloud which shows how far landmarks can vary from its own set of landmark points. Figures 22 and 23 contains images with variations in the mean shape along largest and second largest eigenvector. As seen from the graph,

most of the variations comes from the height and length of the cars.We can confirm it with figure 14,variance of landmarks on the top are much more than rest of the landmarks. Figure 26 shows reconstructed image which shows the higher variance of SUVs which has a different shape for right hand side in the image.



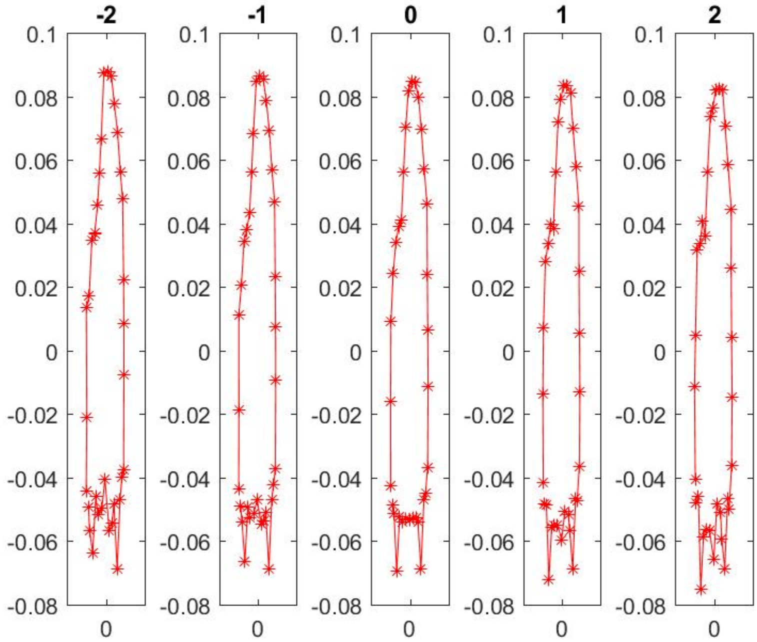
The title is the transformation constant k. We have transformed the mean shape using this formula: $\mu + k\Delta v_1$.

Figure 22: **Variation along Largest "eigenvector" (eigenvalue)**



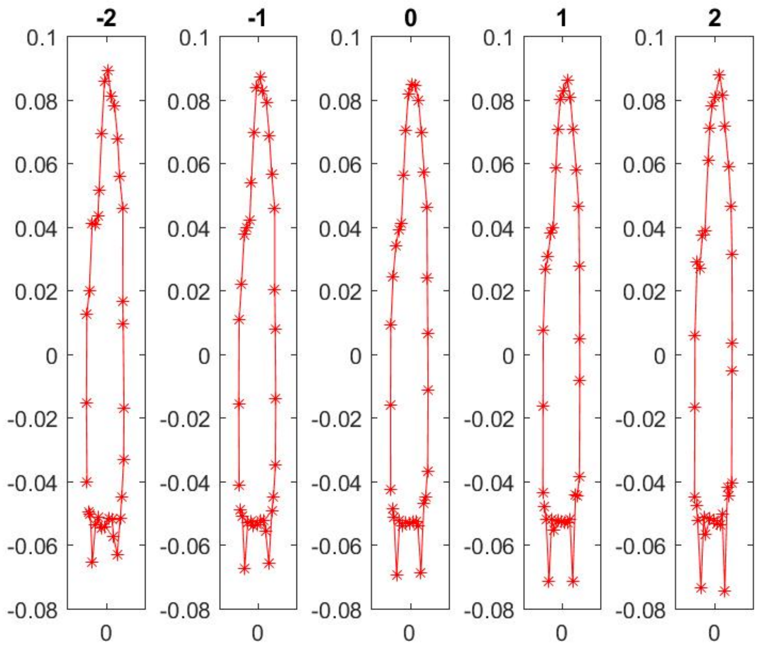
The title is the transformation constant k. We have transformed the mean shape using this formula: $\mu + k\Delta v_2$.

Figure 23: **Variation along second Largest "eigenvector" (eigenvalue)**



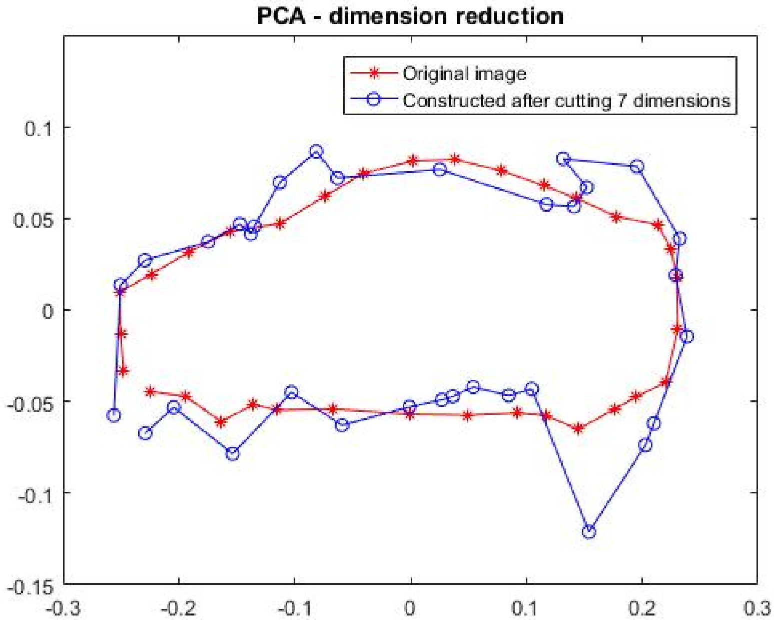
Here, we first converted the mean shape in the different basis (Same as in PCA) and then applied the transformation. After this we reverted back it to the original basis. We have transformed the mean shape using this formula: $\mu + k\Delta v_1$.

Figure 24: **Variation along Largest "eigenvector" (eigenvalue) in the PCA basis**



Here, we first converted the mean shape in the different basis (Same as in PCA) and then applied the transformation. After this we reverted back it to the original basis. We have transformed the mean shape using this formula: $\mu + k\Delta v_2$.

Figure 25: **Variation along second Largest "eigenvector" (eigenvalue) in the PCA basis**



reconstruction of the image. Figure 26 shows reconstructed image which shows the higher variance of SUVs which has a different shape for right hand side in the image.

Figure 26: **PCA dimensionality reduction**