Computer Vision Project
Incisor Segmentation

Sumit Kumar Dey (R0607761)
Master of Artificial Intelligence

September, 2016

Contents

Problem Statement and Road-map

Problem Statement: Given a set of panaromic dental radiograph images, incisors have to be segmented using model-based segmentation approach - Active Shape Models.

1.1 Brief Road Map

From Training Data:

- Create template to find region of interest
- Create Model shapes (Procrustes Analysis + PCA)
- Create models of gray profiles (Covariance Matrix & Mean profiles)

On Test Data:

- Match template and initialize with mean shape
- Find the best gray profile match for each landmark
- Fit shape model to the new points obtained from the previous step

Active Shape Model

2.1 Data Reprsentation

Each landmark is a point on the image (x, y). The landmarks are formed into a vector as,

$$z^j = [x_1^j, y_1^j, x_2^j, y_2^j, x_3^j, y_3^j, ...x_n^j, y_n^j]_{2n \times 1}$$

where, n is the number of landmarks and j is the number of images.

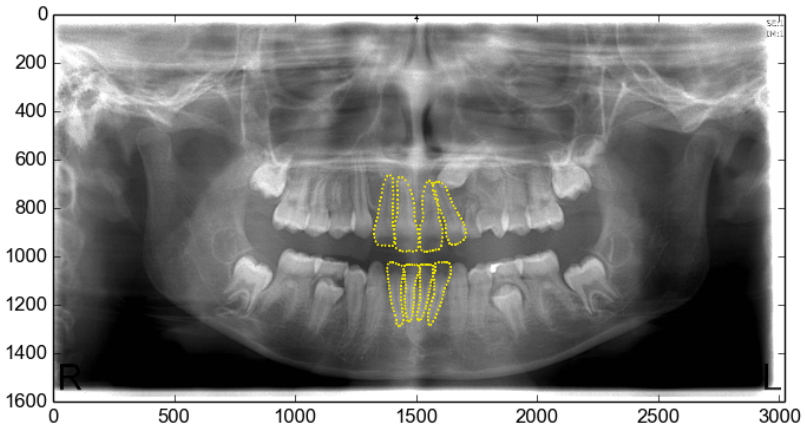
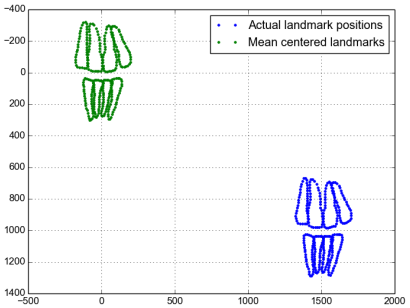


Figure 2.1: Landmarks on an Image

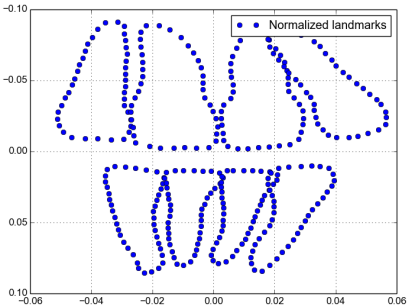
2.2 Procrustes Analysis

Steps for Procrustes Analysis

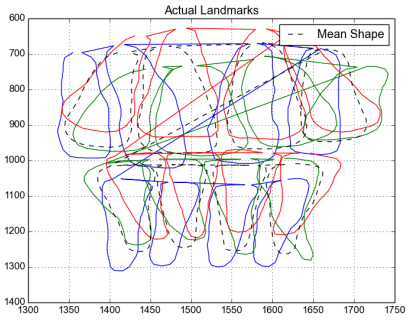
1. Center each shape at (0,0)
2. Fix any shape and scale such that $||z^j||_2 = 1$
3. Scale and rotate rest of the shapes to align this shape.
4. Iterate till Convergence



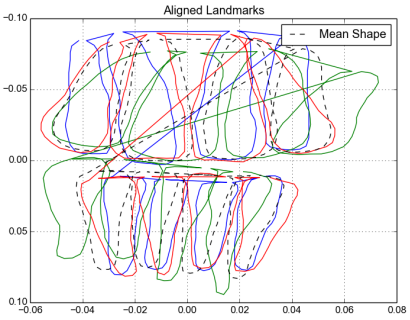
(a) Centering Shape at (0,0)



(b) Normalized Shape



(c) Landmarks before alignment (3 shapes)



(d) Landmarks after alignment (3 shapes)

Figure 2.2: Procrustes Analysis

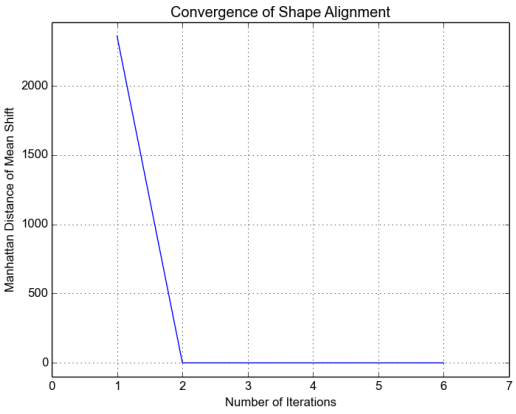
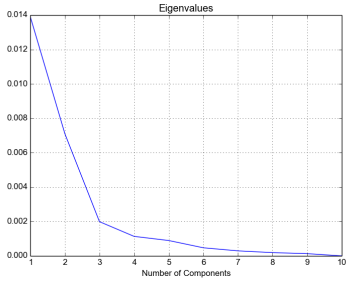


Figure 2.3: Convergence of Shape Alignemnt

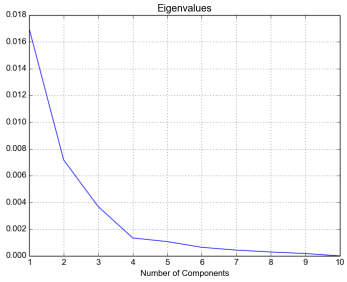
2.3 Principle Component Analysis

Steps for PCA:

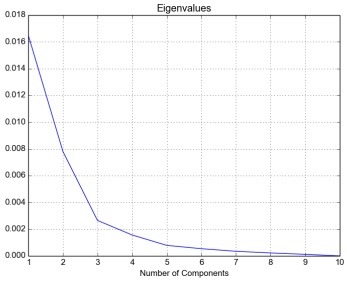
1. Compute mean of the data
2. Compute Covariance Matrix
3. Obtain Eigen Vectors and corresponding Eigen Values
4. Select k eigen vectors as per the largest eigen values



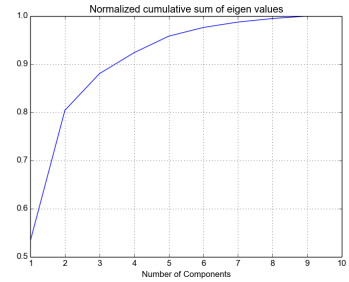
(a) Eigen values considering all 8 teeth



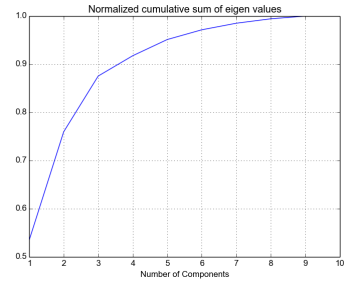
(b) Eigen values considering upper 4 teeth



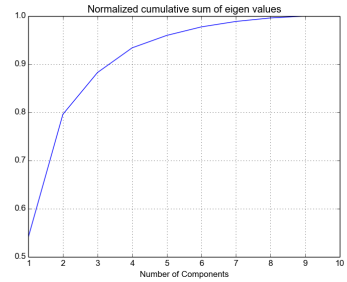
(c) Eigen values considering lower 4 teeth



(d) normalized cumulative sum of eigen values considering all 8 teeth

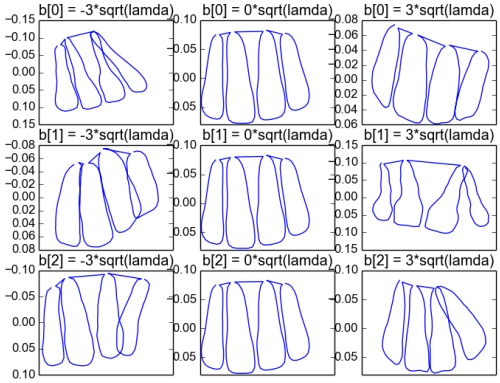


(e) normalized cumulative sum of eigen values considering upper 4 teeth

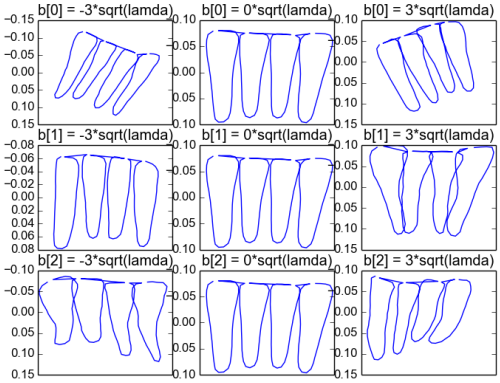


(f) normalized cumulative sum of eigen values considering lower 4 teeth

Figure 2.4: Eigen Values for different sets of teeth



(a) Shape variations in upper 4 teeth



(b) Shape variations in lower 4 teeth

Figure 2.5: Shape variations for the set of 4 teeth

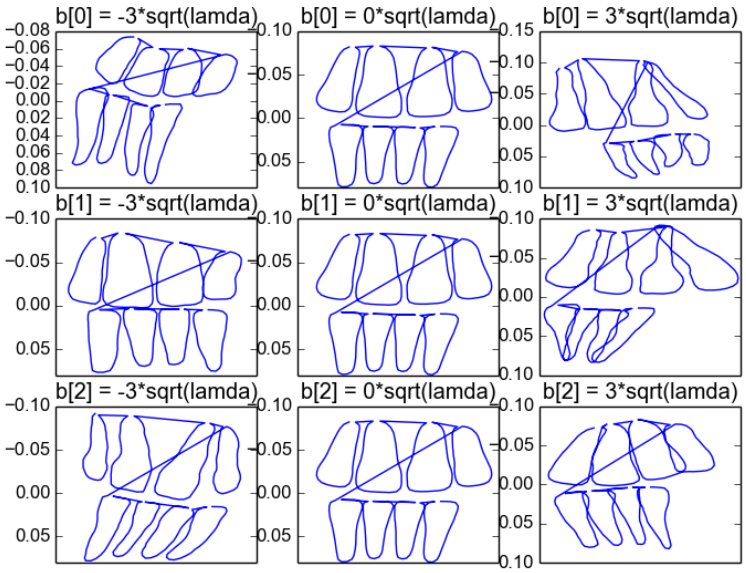
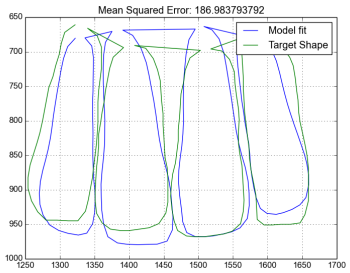


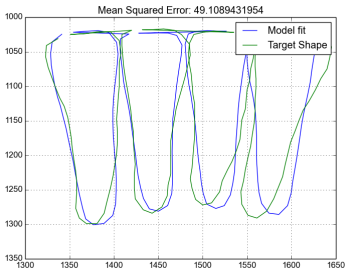
Figure 2.6: Shape variations for the set of 8 teeth

2.4 Model Fitting

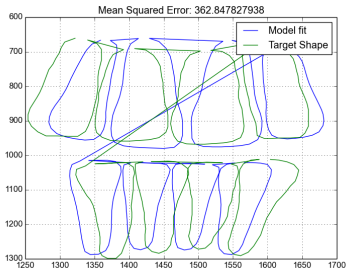
Following are the result of generating the shape model from first 10 images and fitting on the shape of 11th image. The shape is initialized withe mean shape.



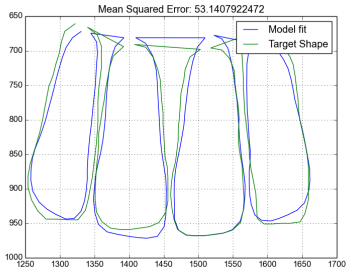
(a) Before fitting - Upper 4 teeth



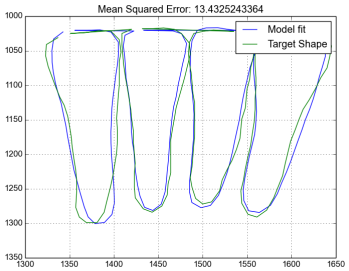
(b) Before fitting - Lower 4 teeth



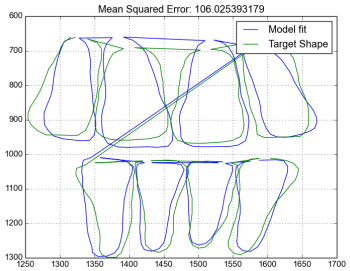
(c) Before fitting - All 8 teeth



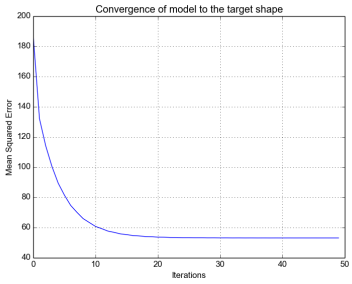
(d) After fitting - Upper 4 teeth



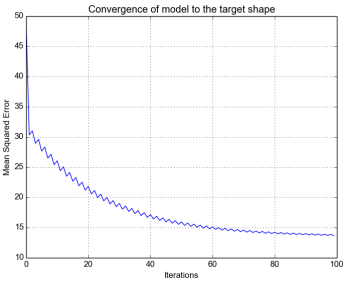
(e) After fitting - Lower 4 teeth



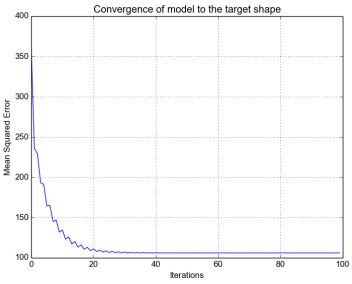
(f) After fitting - All 8 teeth



(g) Convergence of fitting - Upper 4 teeth



(h) Convergence of fitting - Lower 4 teeth

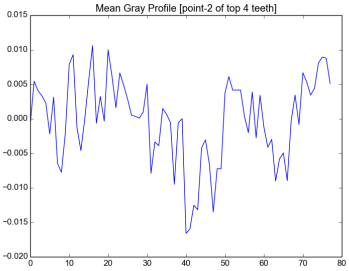


(i) Convergence of fitting - All 8 teeth

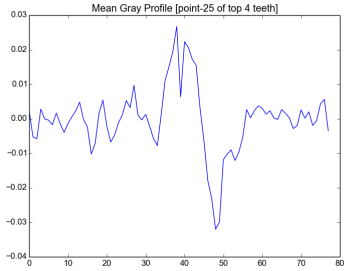
Figure 2.7: Model shape fitting to a target shape

2.5 Gray Profile Match

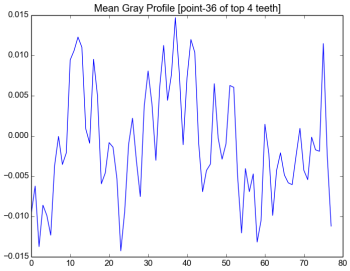
The gray model (profile mean & covariance matrix) is trained on first 10 images with profile length $2k+1$, $k = 40$. The test images are 11, 12 and 13 with search profile length of $2k+1$, $k = 80$ with the ground truth landmark in the middle of the profile. The minimum of mahalanobis distance for each test profile is marked with a black dot in the graph.



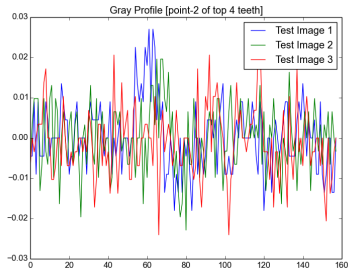
(a) Mean gray profile of landmark-2 of upper 4 teeth



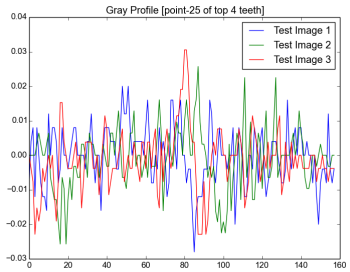
(b) Mean gray profile of landmark-25 of upper 4 teeth



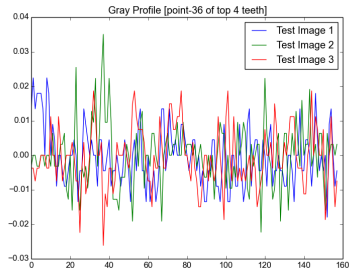
(c) Mean gray profile of landmark-36 of upper 4 teeth



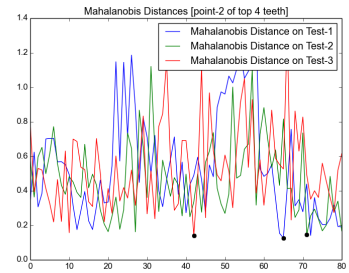
(d) Test images' gray profile of landmark-2 of upper 4 teeth



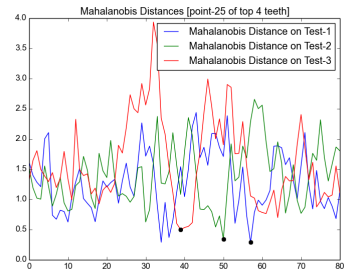
(e) Test images' gray profile of landmark-25 of upper 4 teeth



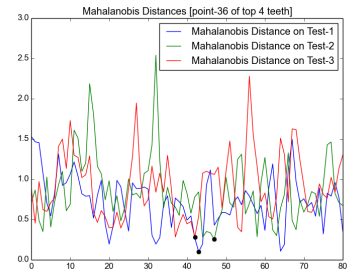
(f) Test images' gray profile of landmark-36 of upper 4 teeth



(g) Mahalanobis Distance of test gray profiles from the mean profile at landmark-2



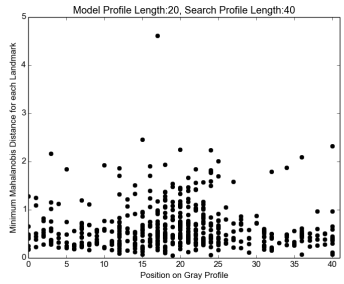
(h) Mahalanobis Distance of test gray profiles from the mean profile at landmark-25



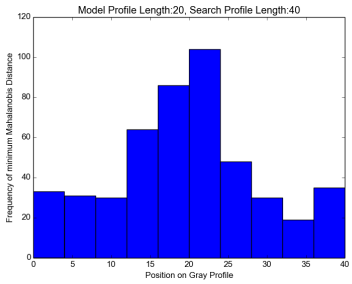
(i) Mahalanobis Distance of test gray profiles from the mean profile at landmark-36

Figure 2.8: Gray profiles and Mahalanobis distances

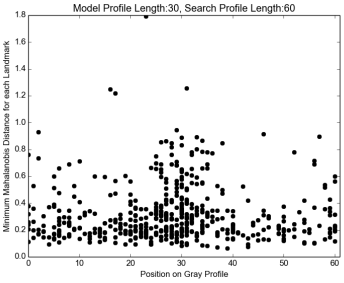
As mahalanobis distances do not seem to be very reliable, the distribution of the same is plotted to see the overall performance over different profile lengths with ground truth landmark in the middle. As ground truth is in the middle, hence, ideally all the minimums of mahalanobis distance should lie in the middle of the profile.



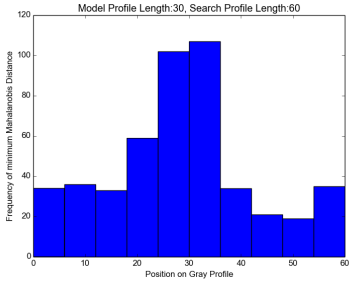
(a) Distribution of mimimum of Mahalanobis distance, Model profile length (k=20), Search profile length (k=40)



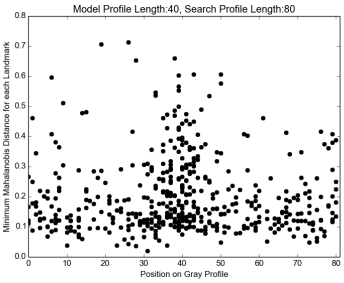
(b) Histogram of mimimum of Mahalanobis distance, Model profile length (k=20), Search profile length (k=40)



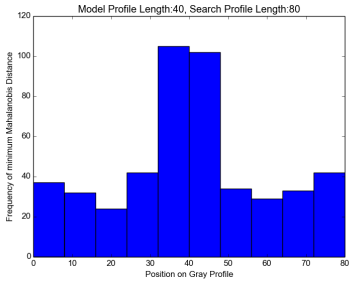
(c) Distribution of mimimum of Mahalanobis distance, Model profile length (k=30), Search profile length (k=60)



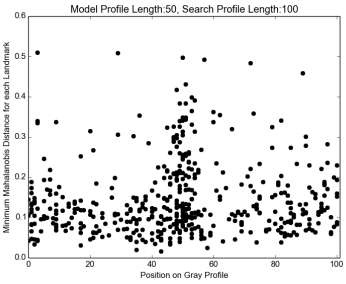
(d) Histogram of mimimum of Mahalanobis distance, Model profile length (k=30), Search profile length (k=60)



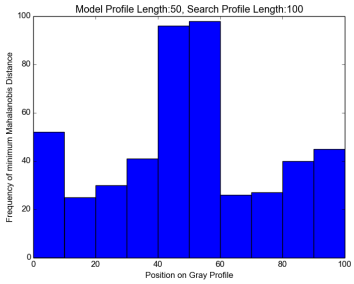
(e) Distribution of mimimum of Mahalanobis distance, Model profile length (k=40), Search profile length (k=80)



(f) Histogram of mimimum of Mahalanobis distance, Model profile length (k=40), Search profile length (k=80)



(g) Distribution of mimimum of Mahalanobis distance, Model profile length (k=50), Search profile length (k=100)



(h) Histogram of mimimum of Mahalanobis distance, Model profile length (k=50), Search profile length (k=100)

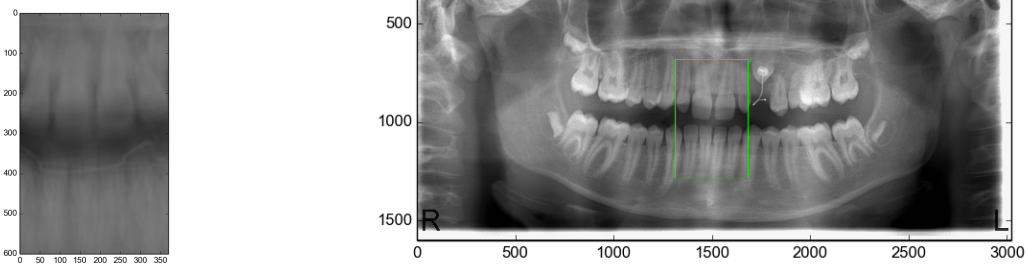
Figure 2.9: Distribution of minimum of mahalanobis distance for all landmarks in upper 4 teeth

Results

The results are based on leaved one out. One good result and one not so good result is presented.

3.1 A Good Result

3.1 Initialization with template matching



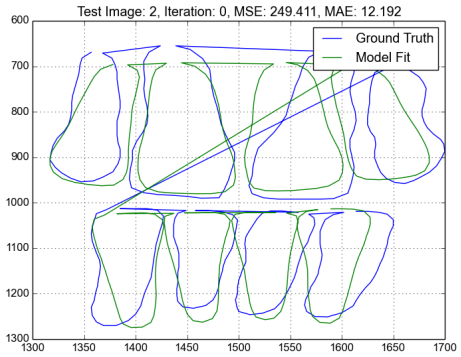
(a) The template using training images

(b) Template match on test image - 2

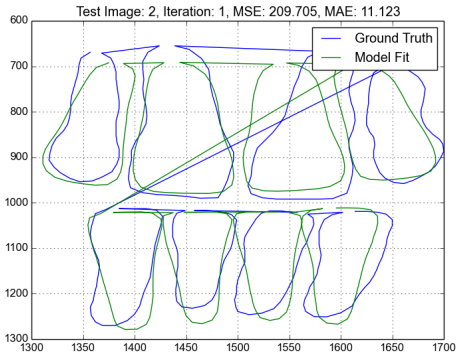
Figure 3.1: Template match on test image - 2

3.1 Iterative Fitting

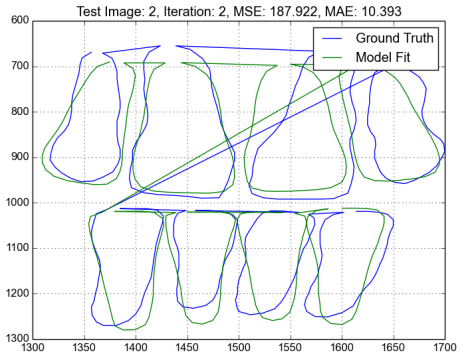
The mean shape is placed in the center of template match. Then gray profile match is run to find new points to fit. Then model is iteratively fit over these new points.



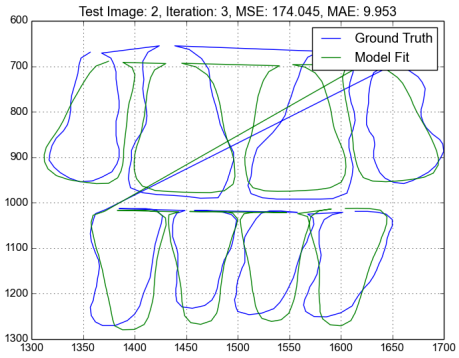
(a) Iteration 0



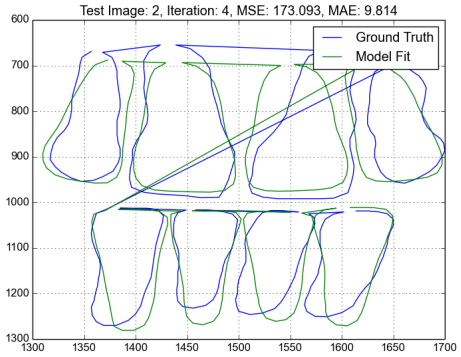
(b) Iteration 1



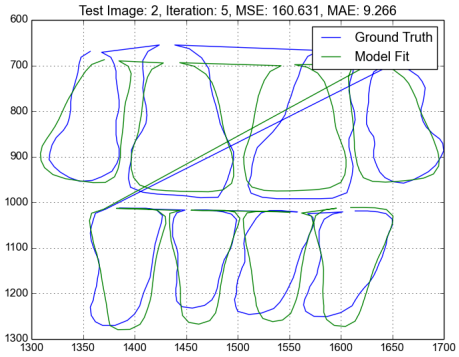
(c) Iteration 2



(d) Iteration 3

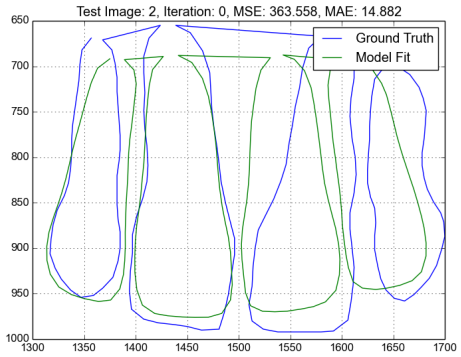


(e) Iteration 4

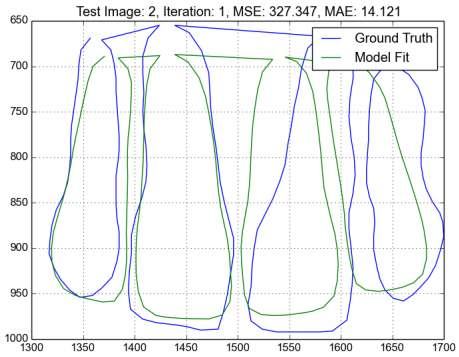


(f) Iteration 5

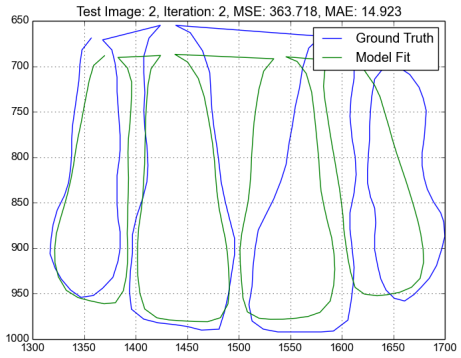
Figure 3.2: Iterative Fitting on all 8 teeth



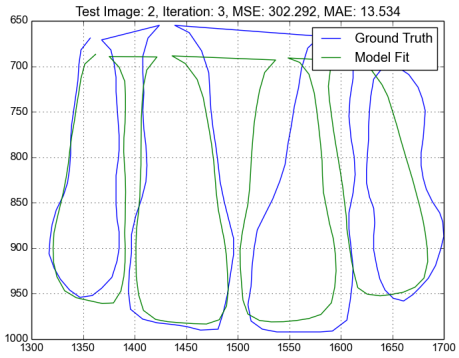
(a) Iteration 0



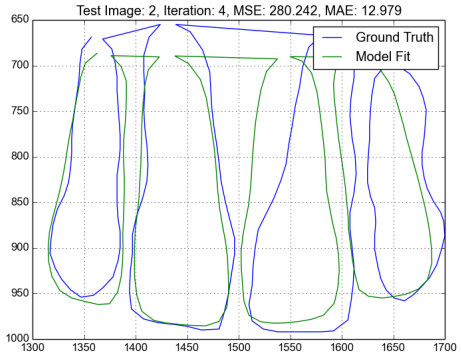
(b) Iteration 1



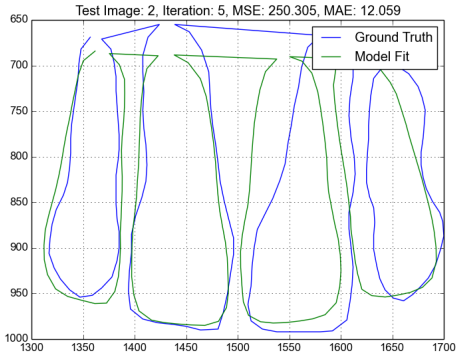
(c) Iteration 2



(d) Iteration 3

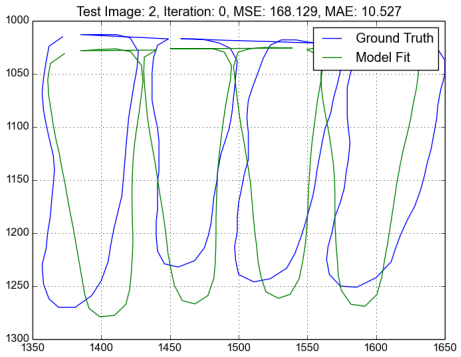


(e) Iteration 4

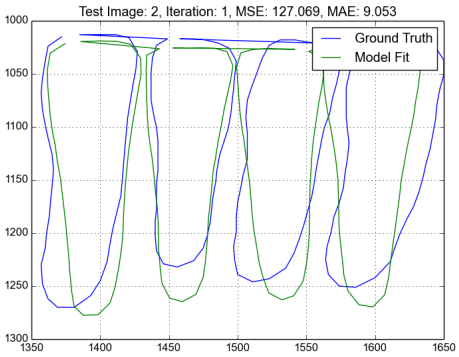


(f) Iteration 5

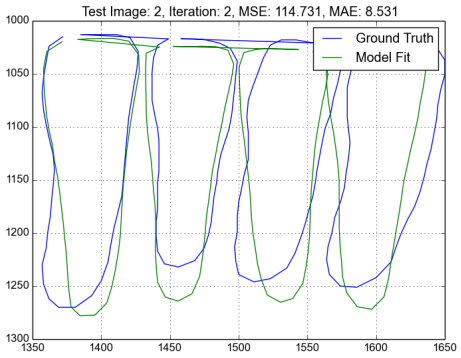
Figure 3.3: Iterative Fitting on upper 4 teeth



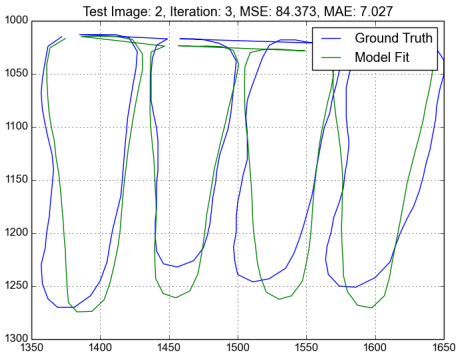
(a) Iteration 0



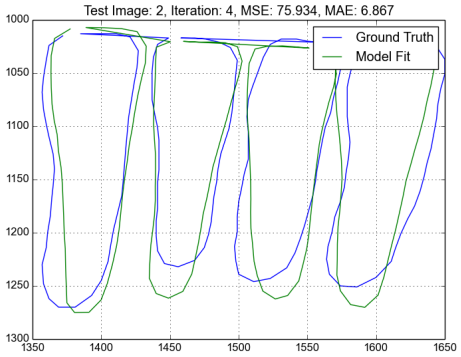
(b) Iteration 1



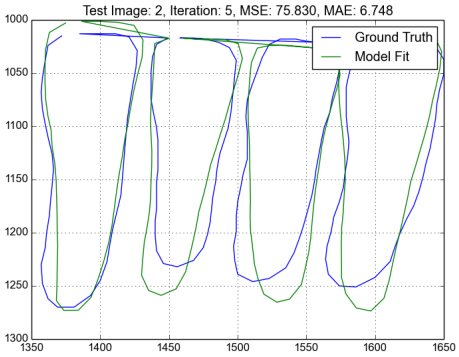
(c) Iteration 2



(d) Iteration 3



(e) Iteration 4

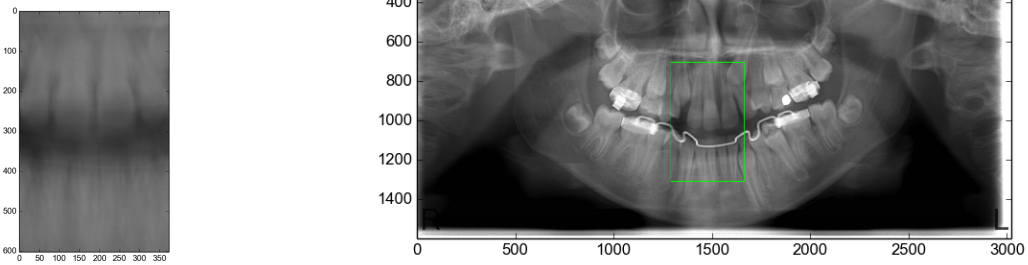


(f) Iteration 5

Figure 3.4: Iterative Fitting on lower 4 teeth

3.2 A Not So Good Result

3.2 Initialization with template matching



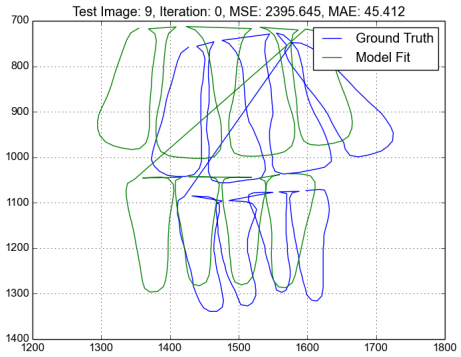
(a) The template using training images

(b) Template match on test image - 9

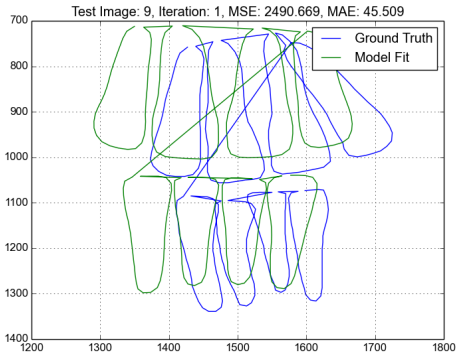
Figure 3.5: Template match on test image - 9

3.2 Iterative Fitting

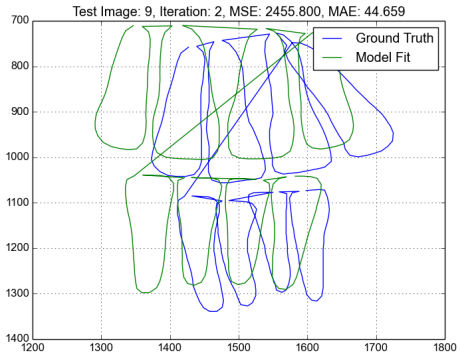
The mean shape is placed in the center of template match. Then gray profile match is run to find new points to fit. Then model is iteratively fit over these new points.



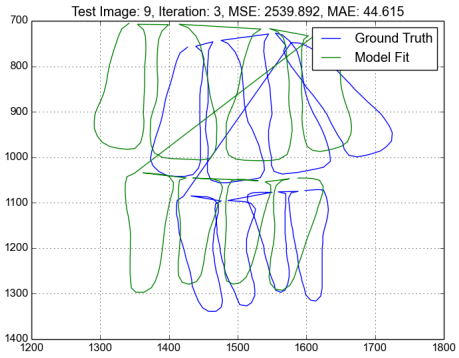
(a) Iteration 0



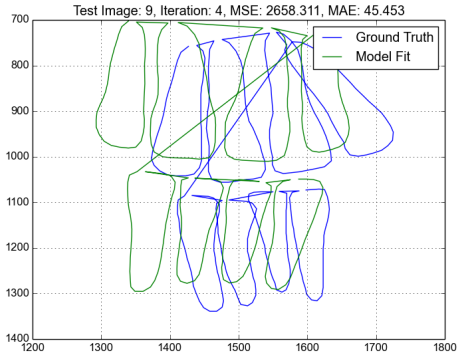
(b) Iteration 1



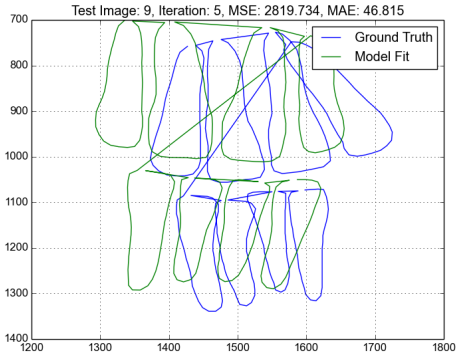
(c) Iteration 2



(d) Iteration 3

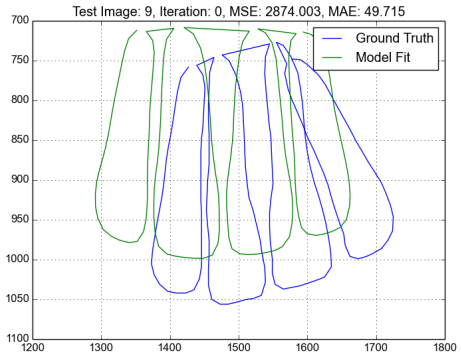


(e) Iteration 4

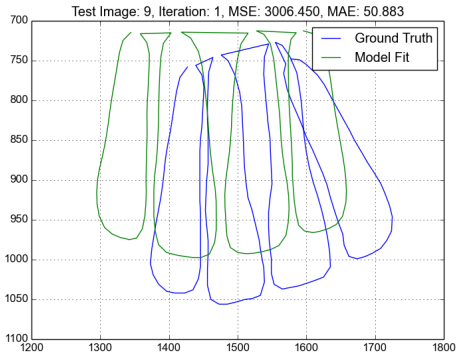


(f) Iteration 5

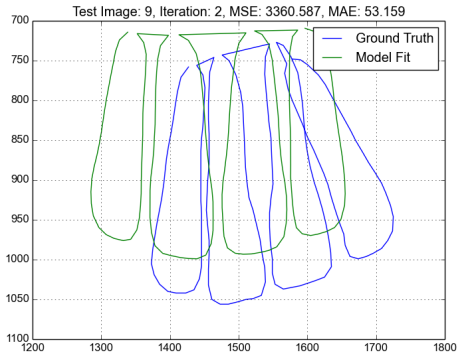
Figure 3.6: Iterative Fitting on all 8 teeth



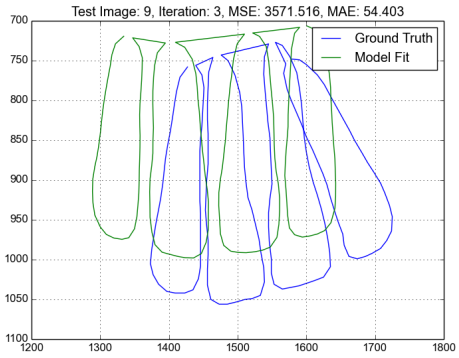
(a) Iteration 0



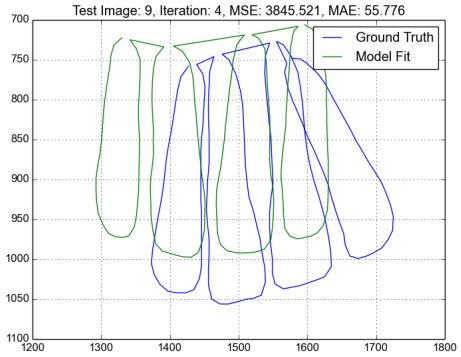
(b) Iteration 1



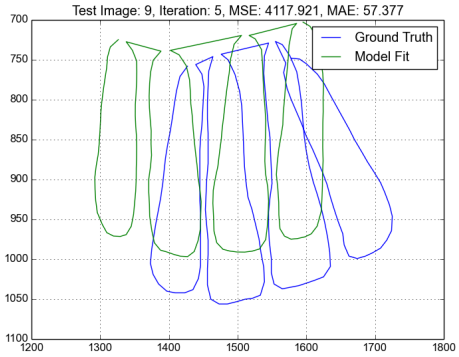
(c) Iteration 2



(d) Iteration 3

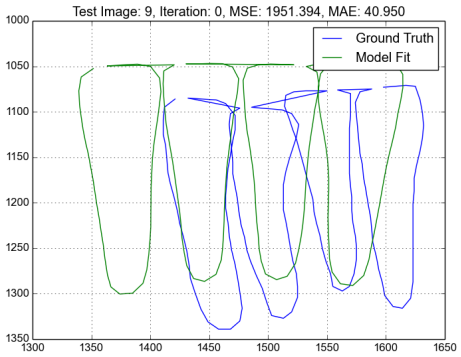


(e) Iteration 4

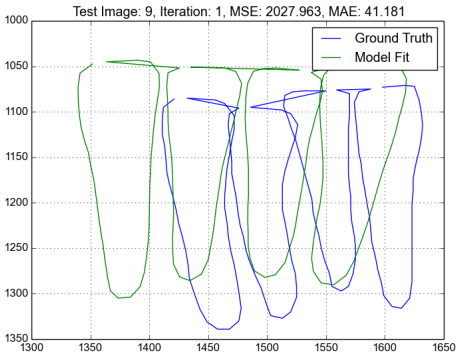


(f) Iteration 5

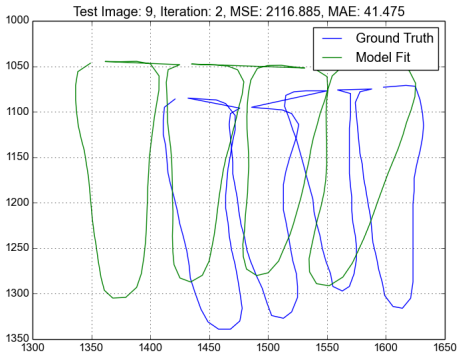
Figure 3.7: Iterative Fitting on upper 4 teeth



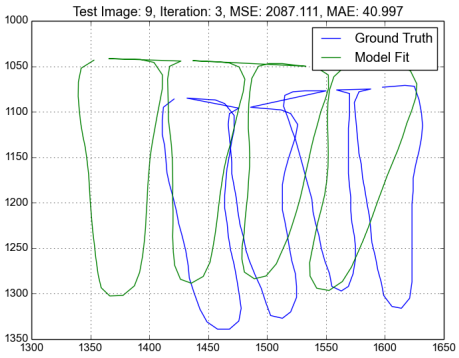
(a) Iteration 0



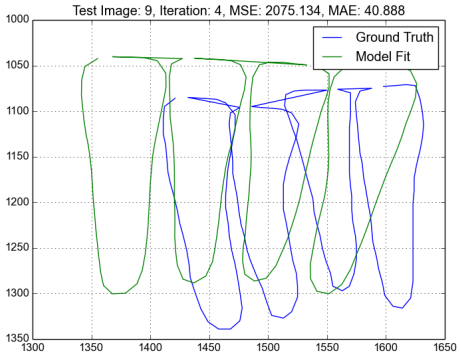
(b) Iteration 1



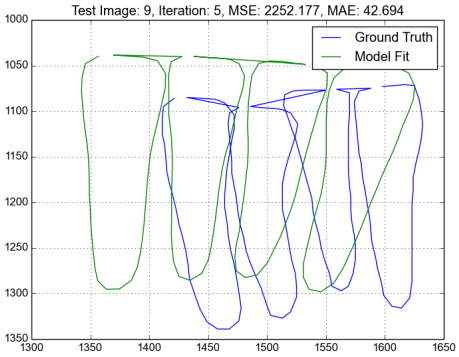
(c) Iteration 2



(d) Iteration 3



(e) Iteration 4



(f) Iteration 5

Figure 3.8: Iterative Fitting on lower 4 teeth

Conclusions

- Performance is highly sensitive to initialization (the start location of search). It works reasonably well if initial search location is good.
- Not all landmarks are equally good. Some landmarks are can be matched (gray profile match) more accurately than others.
- Choice of length of search profile is a matter of trade-off. While longer search profiles might be able to cope up better with initialization offset, it also has potential to increase the number of wrong gray profile matches.
- Convergence is not guaranteed.

Appendices

Procrustes Analysis: Aligning Shapes

The landmarks are formed into a vector as,

$$z^j = [x_1^j, y_1^j, x_2^j, y_2^j, x_3^j, y_3^j, ...x_n^j, y_n^j]_{2N\times 1}$$

where, N is the number of landmarks and j is the number of examples.

Algorithm

1. Center each shape z^j at $(0, 0)$
2. Scale the first shape such that $||z^1||_2 = 1$. (Any shape can be taken instead of the first shape)
3. Scale and rotate each shape to align to this shape.

$$a^j = \frac{z^j.z^1}{||z^j||^2}$$

$$b^j = \sum_{i=1}^{N} \frac{x_i^j.y_i^1 - x_i^1.y_i^j}{||z^j||^2}$$

$$scale^j = \sqrt{(a^j)^2 + (b^j)^2}, \quad \theta^j = \arctan(\frac{b^j}{a^j})$$

$$\begin{bmatrix} \hat{x}_i^j \\ \hat{y}_i^j \end{bmatrix} = scale^j \begin{bmatrix} \cos\theta^j & \sin\theta^j \\ -\sin\theta^j & \cos\theta^j \end{bmatrix} \begin{bmatrix} x_i^j \\ y_i^j \end{bmatrix}$$

4. If not converged then repeat.

Model Fitting

Given a target vector Y of size $2N$ to which we wish to fit our shape model. This means, we want to minimize the following function.

$$||Y - M(\mu - Pb)||^2$$

where, P is vector of modes, and b captures variance of each mode.

$$M = scale \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix}$$

where, x_t and y_t are the x and y translations.

Algorithm

1. Initialize $b = 0$, i.e. $X = \mu$
2. Generate model points $X = \mu + Pb$
3. Find $(scale, \theta, t)$. This gives $Y_{new} = M^{-1}Y$
4. Project $y = fracY_{new}Y_{new}.\mu$
5. Update Model Parameters. $b = P^T(y - \mu)$
6. Go to step 2 until convergence