Summer Internship Report

May 2013 – July 2013

# LOCATING FACIAL FEATURES

## WITH ACTIVE SHAPE MODELS

Satya Narayan Shukla

10EE35025

Department of Electrical Engineering

IIT Kharagpur

Supervised by Prof. A. Routray

# Abstract

This report focuses on locating features in frontal views of upright human faces. The report starts with the Active Shape Model of Cootes et al.  The appearance of objects in images varies due to several reasons including lighting effects, 3D pose, and other. Here objects are represented by a number of landmark points and then the shape variations of an object is modelled using principal component analysis to obtain a point distribution model.

# Acknowledgements

At the very beginning of this report, I would like to extend my sincere and heartfelt obligation towards all the people who have stood by and helped me in this endeavour. I am indebted to them for their dynamic guidance, help, cooperation & encouragement.

I take this opportunity to express my deep gratitude and regards towards my project guide Professor Aurobinda Routray for his exemplary guidance, monitoring and constant encouragement throughout the duration of this project and providing the best possible environment for me to do this work.

I also take this opportunity to express a deep sense of gratitude to my mentor Mr. S.L. Happy for their cordial support, valuable insights and guidance, which helped me in completing this task through various stages.

A thanks goes to my friends and colleagues in developing the project and people who willingly helped me out whenever I needed them.

*Satya Narayan Shukla*

*10EE35025*

# CONTENTS

# 1. Introduction

**Active shape models** (ASMs) are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a new image, developed by Tim Cootes and Chris Taylor in 1995.

The shapes are constrained by the PDM (point distribution model) Statistical Shape Model to vary only in ways seen in a training set of labelled examples.

The shape of an object is represented by a set of points (controlled by the shape model). The ASM algorithm aims to match the model to a new image. It works by alternating the following steps:

- Look in the image around each point for a better position for that point
- Update the model parameters to best match to these new found positions

To locate a better position for each point one can look for strong edges, or a match to a statistical model of what is expected at the point. The original methodology suggests using the Mahalanobis distance to detect a better position for each landmark point.

The technique has been widely used to analyze images of faces, mechanical assemblies and medical images (in 2D and 3D).

# 2. Active Shape Models

This chapter first describes shape models in general. It then gives a description of the classical Active Shape Model (ASM).

The classical ASM is that presented in Cootes and Taylor [3] Chapter 7. In brief, the classical ASM is characterized by its use of the Mahalanobis distance on one-dimensional profiles at each landmark and a linear point distribution model. Training determines the characteristics of the profile and point distribution models. What this all means will be explained shortly.

Here, a shape is just an n * 2 ordered set of points i.e. an array of (x, y) coordinates. The points are related to each other in some invariant sense. If you move a shape, it is still the same shape. If you expand or rotate it, it is still the same shape. Edges between points are not part of the shape but are often drawn to clarify the relationship or ordering between the points.

In practice it is convenient to represent a shape not as an n * 2 array of (x, y) coordinates, but as a 2n * 1 vector: first all the x- and then all the y-coordinates. This representation is used for the equations in this report.

The distance between two shapes is the sum of the distances between their corresponding points. The distance between two shapes x1 and x2 is the root mean square distance between the shape points √((x1- x2)*(x1 - x2)) after alignment .

## 2.1 Aligning Shapes

A shape can be aligned to another shape by applying a transform which yields the minimum distance between the shapes. Here, allowed transforms are scaling, rotating and linear translation. A transform that does all three is a similarity transform.

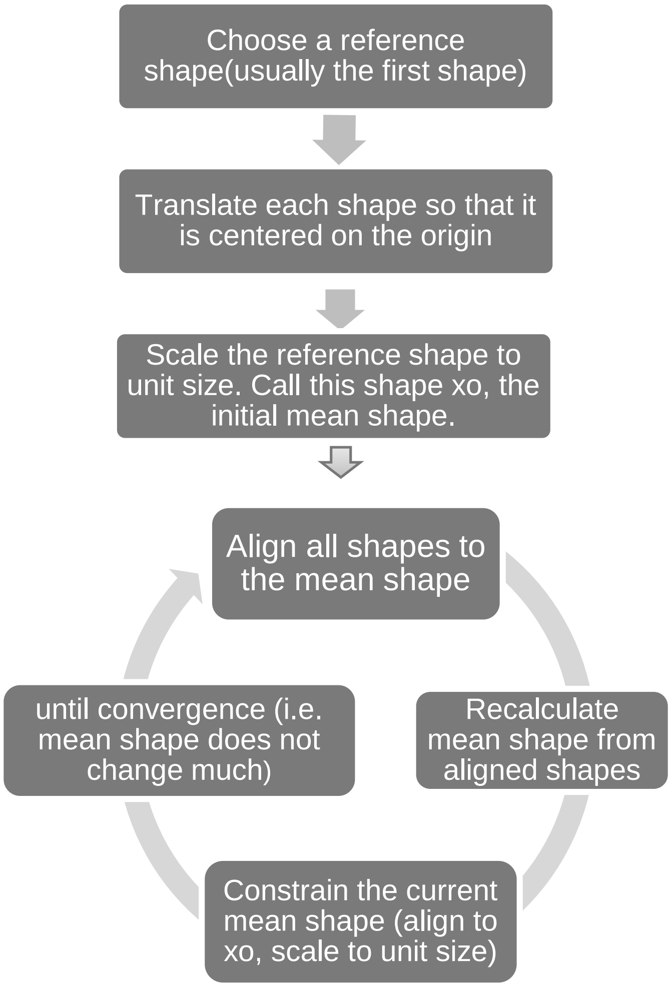
The similarity transform T which rotates the point (x, y) by $\alpha$, scales it by $s$ and translates it by $x_{translate}$ and $y_{translate}$ is –

$$\mathrm{T}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_{translate} \\ y_{translate} \end{pmatrix} + \begin{pmatrix} s*cos\alpha & s*sin\alpha \\ -s*sin\alpha & s*cos\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \qquad (2.1)$$

A set of shapes can be aligned using an iterative algorithm as shown in Figure 2.1. The constraint is necessary to prevent the estimated mean shape from wandering around or shrinking.

Scaling and rotating shapes during alignment introduce non- linearity. These can be minimized by projecting shapes into a tangent space, but tangent spaces are not used in this project.

Input: set of unaligned shapes


Choose a reference shape(usually the first shape)
Translate each shape so that it is centered on the origin
Scale the reference shape to unit size. Call this shape xo, the initial mean shape.
Align all shapes to the mean shape
Recalculate mean shape from aligned shapes
Constrain the current mean shape (align to xo, scale to unit size)
until convergence (i.e. mean shape does not change much)

Output: set of aligned shapes, and mean shape

*Figure 2.1 Iterative algorithm for Aligning Shapes*

## 2.2 Overview of Active Shape Model

The ASM is first trained on a set of manually landmarked images. By manually landmarked I mean that somebody had to mark all the images by hand. This is done before training begins.

After training we can use the ASM to search for features on a face. An example search is shown in Figure 2.2.

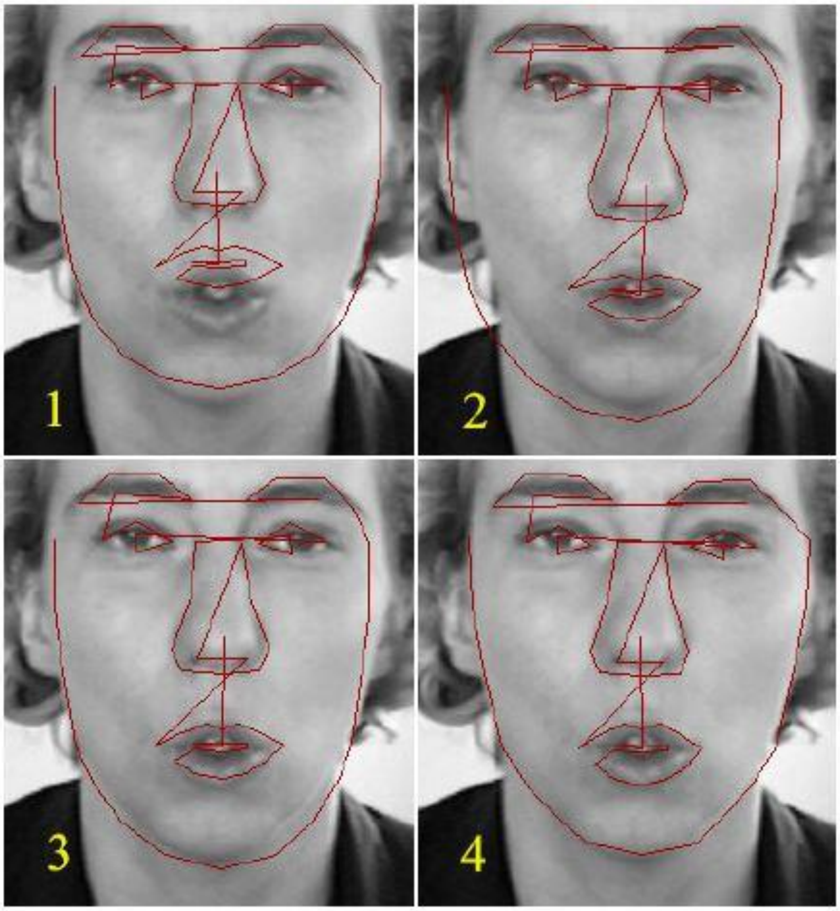


*Figure 2.2 An ASM search: Each picture shows the shape after correction by the shape model. Not all intermediate shapes are shown.*

The general idea is try to locate each landmark independently, then correct the locations if necessary by looking at how the landmarks are located with respect to each other. To do this, the ASM is constructed from two types of sub-model:

1. A **profile model** for each landmark, which describes the characteristics of the image around the landmark. The model specifies what the image is expected to “look like" around the landmark. During training, we sample the area around each landmark across all training images to build a profile model for the landmark. During search, we sample the area in the vicinity of each tentative landmark, and move the landmark to the position that best matches that landmark's model profile. This generates tentative new positions for the landmarks, called the suggested shape.

2. A **shape model** which defines the allowable relative position of the landmarks. During search, the shape model adjusts the shape suggested by the profile model to conform to a legitimate face shape. This is needed because the profile matches at each landmark are unreliable.

```
Input: image of a face

1. Generate the start shape by locating the overall position of the face
2.  repeat
3a.     for each shape point
3b.         for each offset (i.e. look in the area around the point)
3c.             Build a profile by sampling the area around the offset
3d.             Measure the fit of the profile against the Profile Model
3e.         Move the point to the offset of the best profile match
4.      Adjust the suggested shape to conform to the Shape Model
5.  until convergence (i.e. until no further improvements in fit are
possible)

Output: shape giving the (x, y) coordinates of the face landmarks
```

*Figure 2.3 ASM search algorithm for faces*

The algorithm iterates for a solution using both sub-models as shown in Figure 2.3 on the previous page. The algorithm combines the results of the weak profile classifiers to build a stronger overall classifier. It is a shape constrained feature detector: the shape model acts globally; each profile matcher acts locally.

## 2.3 The Shape Model

The job of the shape model is to convert the shape suggested by the profile models to an allowable face shape. Before building the shape model, the training shapes are aligned. Then the shape model consists of an average face and allowed distortions of the average face:

$$\hat{X} = \bar{X} + \emptyset * b \qquad (2.2)$$

where $\hat{X}$ is the generated shape vector (all the x- followed by all the y-coordinates). The hat on X reminds us that it is generated by a model, b is shape coefficient vector.

$\bar{X}$ is the mean shape: the average of the aligned training shapes $x_i$, defined as

$$\bar{X} = \frac{1}{n_{shapes}} \sum_{i=1}^{n_{shapes}} x_i \qquad (2.3)$$

$\emptyset$ is the matrix of eigenvectors of the covariance matrix $S_s$ of the training shape points.

$$S_s = \frac{1}{n_{shapes} - 1} \sum_{i=0}^{n_{shapes}} (x_i - \bar{x}).(x_i - \bar{x})^T \qquad (2.4)$$

Using a standard principal components analysis approach, the eigenvalues $\lambda_i$ of $S_s$ were ordered and a limited number of the corresponding eigenvectors were kept in $\emptyset$. The retained columns of $\emptyset$ are the eigenvectors corresponding to the largest eigenvalues of $S_s$. Important aspects of the training shapes were captured but noise was ignored.

## 2.3.1 Generating shapes from the shape model

Equation 2.2 can be used to generate different shapes by varying the vector parameter b. By keeping the elements of b within limits it is ensured that generated faces are lifelike.

Figure 2.4 shows three faces generated by setting $b_1$ to -3$\sqrt{\lambda_1}$ , 0 and +3$\sqrt{\lambda_1}$, with all other $b_i$'s fixed at 0 (where $b_1$ is the first element of b and $\lambda_1$ is the largest eigenvalue).

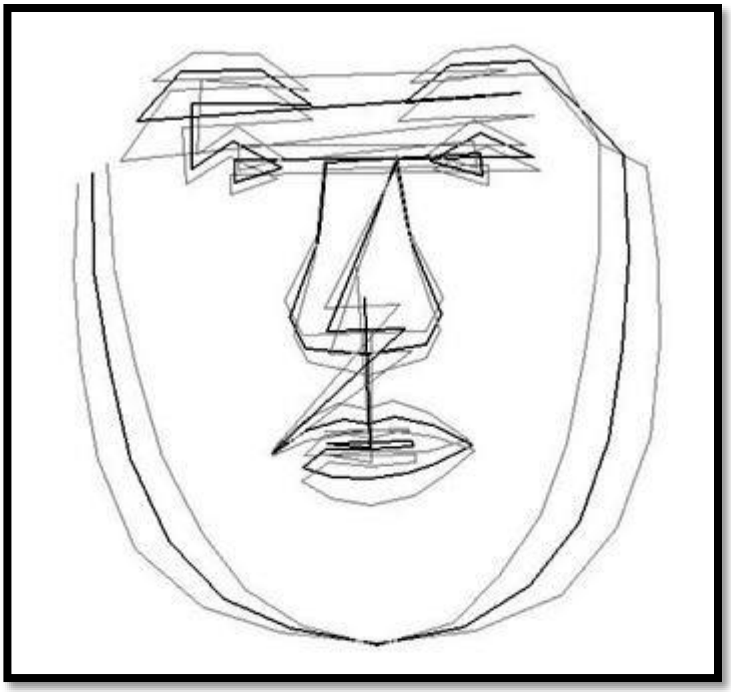
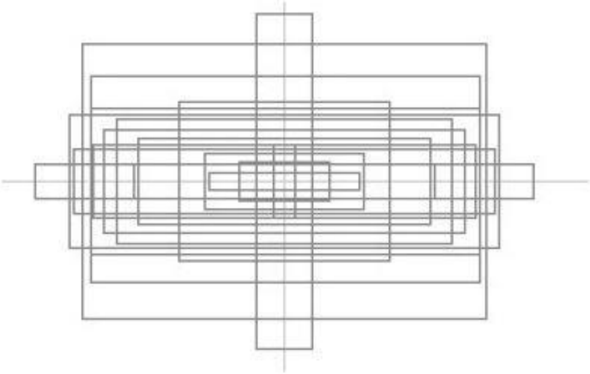


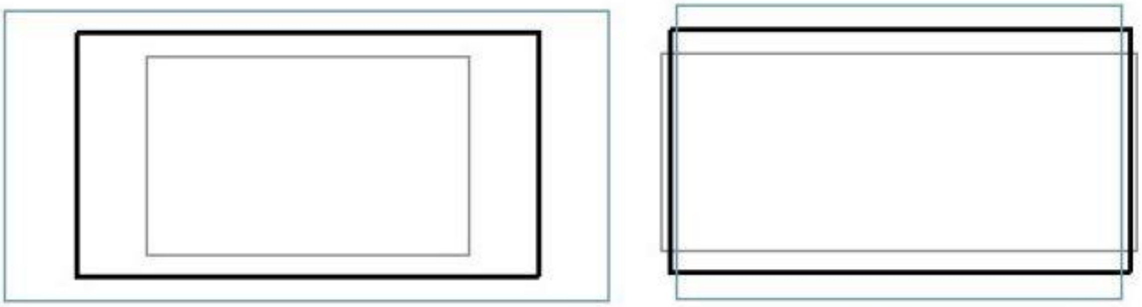
*Figure 2.4: The mean face (black) with variations of the first principal component (gray).*

## 2.3.2 Understanding the shape model

It is easier to gain an intuitive understanding of the shape model if rectangles are used instead of complicated face shapes. Figure 2.5 shows an example. To specify the four points of any of these rectangles, we need eight numbers: four x- and four y-coordinates. But the rectangles are centered and symmetrical about the origin. Thus we can actually specify a rectangle by just two parameters: its width and its height.



Some simple shapes: symmetrical rectangles



The mean shape and two variations created by adding a multiple of the first eigenvector.

the mean shape and two variations created by adding a multiple of the second eigenvector.

*Figure 2.5: A simple shape model*

Now Equations 2.2, 2.3 and 2.4 are used to generate the shape model from the given rectangles, which becomes (x- and y-coordinates in column1 and column 2 respectively)

$$\hat{x} = \begin{pmatrix} 23 & 12 \\ -23 & 12 \\ -23 & -12 \\ 23 & -12 \end{pmatrix} + b_0 \begin{pmatrix} 12 & 4 \\ -12 & 4 \\ -12 & -4 \\ 12 & -4 \end{pmatrix} + b_1 \begin{pmatrix} -4 & 12 \\ 4 & 12 \\ 4 & -12 \\ -4 & -12 \end{pmatrix} + \ldots..$$

The sorted eigenvalues of the covariance matrix $S_s$ are 3778; 444; 2; 0.1; …... There are eight eigenvalues altogether. The first two are much larger than the rest. (The remaining eigenvalues represent noise in the form of numerical errors.) Thus the process has discovered that the shapes can be parameterized by just two parameters, $b_o$ and $b_1$. The first parameter varies the contribution of the first eigenvector and, in this example, chiefly changes the size of the generated shape. The second parameter varies the contribution of the second eigenvector, which mainly adjusts the aspect ratio of the shape.

For face shapes there will be many eigenvalues and no abrupt cutoff point. The relative size of the eigenvalues tells us the proportion of variation captured by the corresponding eigenvectors. We can capture as much variation of the input shapes as we want by retaining the appropriate number of eigenvectors.

## 2.3.3 Representing a given shape by a shape model

In the reverse direction, given a suggested shape x on the image, we can calculate the parameter b that allows Equation 2.2 to best approximate x with a model shape $\hat{x}$. We seek the b and T that minimizes

$$\text{Distance}\ \left(\text{X},\ \text{T}\ (\bar{X} + \emptyset * b)\right) \qquad (2.5)$$

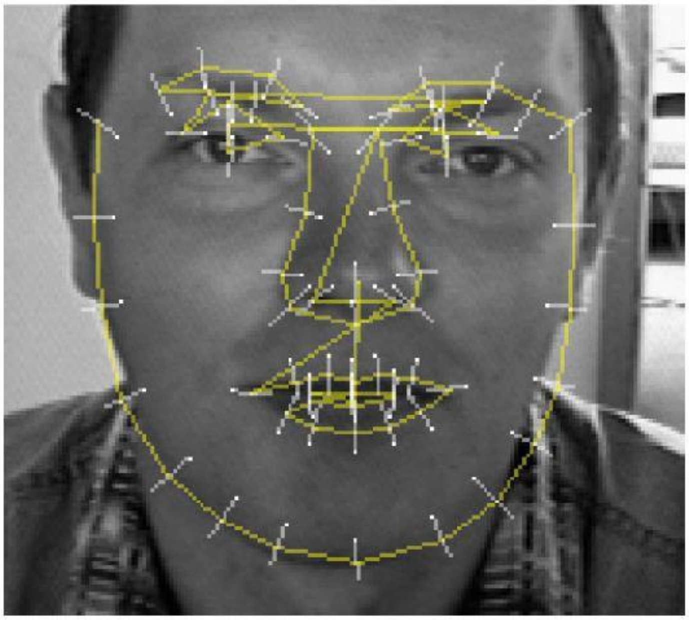
T is a similarity transform which maps the model space into the image space. The transform is needed because the face shape x could be anywhere in the image plane, but the model works of scaled upright shapes positioned on the origin. Cootes and Taylor [3] section 4.8 describes an iterative algorithm for finding b and T.

After calculating b, we reduce any out-of-range elements $b_i$ to ensure that the generated model conforms to the model, yet remains close to the suggested shape.

## 2.4 The ASM Profile Model

The job of the profile model is to take an approximate face shape and produce a better suggested shape by template matching at the landmarks. Search is started with the mean face from the shape model, aligned and positioned with a global face detector. (Figure 2.6)

*Figure 2.6: The "mean face" (yellow) positioned over a search face at the start of a search. The white lines are "whiskers", along which the image intensity will be sampled to form profiles. In this example it so happens that the nose and mouth are already positioned quite accurately by the global face detector. The eyes are badly positioned.*



## 2.4.1 Forming a Profile

To form the profile vector g at a landmark, we sample image intensities along a one-dimensional whisker. The whisker is a vector at the landmark which is orthogonal to a shape edge.  The profile vector is formed as follows:

1. Set each element of the profile vector to the gray level (0...255) of the image below it.
2. Replace each profile element by the intensity gradient. This is done by replacing the profile element at each position i with the difference between it and the element at i - 1.
3. Divide each element of the resulting vector by the sum of the absolute values of all vector elements.

Using normalized gradients in this manner is intended to lessen the effect of varying image lighting and contrast.

## 2.4.2 Building the profile model during training

During training, a model is built for each landmark by creating a mean profile $\bar{g}$ and a covariance matrix $S_g$ of all training profiles (one from each image) at that landmark. The assumption is that the profiles are approximately distributed as a multivariate Gaussian, and thus can be described by their mean and covariance matrix.

If the length of the profile is 7 (as in Figures 2.7 on next page), $\bar{g}$ will have 7 elements and $S_g$ will be a 7*7 matrix.  If there are ‘n’ landmarks then there will be ‘n’ separate $\bar{g}$’s and $S_g$’s.

## 2.4.3 Searching for the best profile

During search, at each landmark we form several search profiles by sampling the image in the neighborhood of the landmark. Each search profile is centered at small positive or negative displacements along the whisker. We typically form profiles at offsets up to about ±3 pixels along the whisker. Figure 2.7 on the next page shows this process.

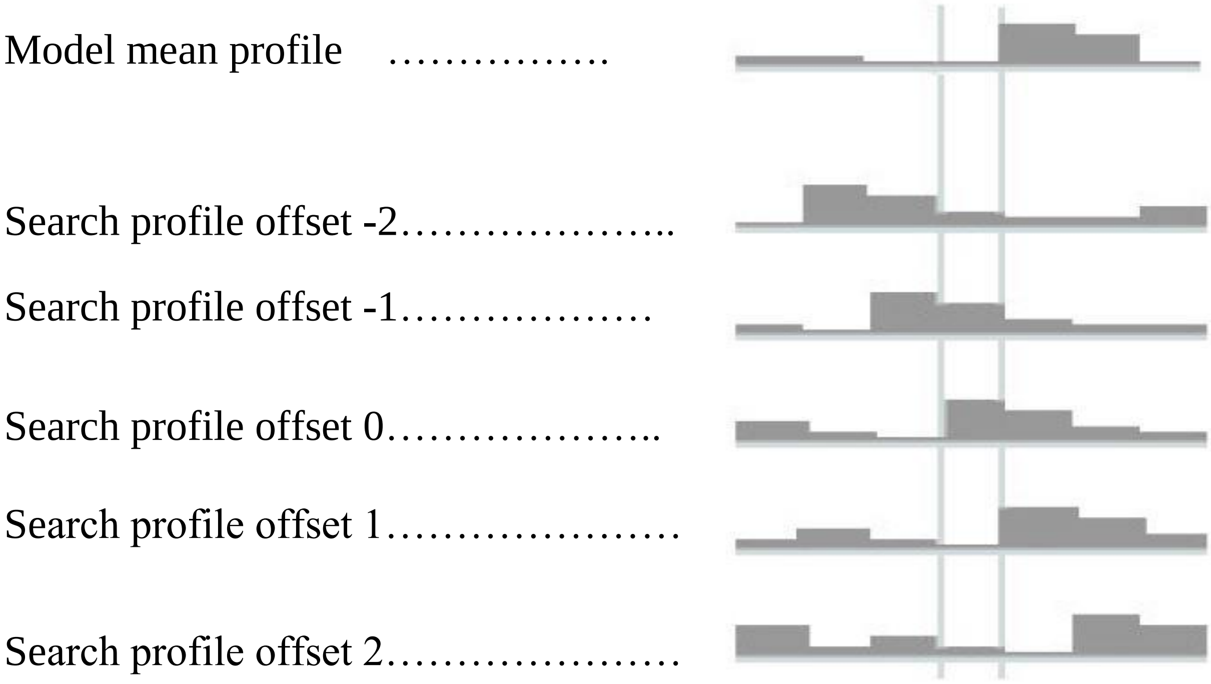

Model mean profile …………….
Search profile offset -2………………..
Search profile offset -1………………
Search profile offset 0………………..
Search profile offset 1…………………
Search profile offset 2…………………

*Figure 2.7: Searching for a profile match.*
The top profile is the model mean profile $\bar{g}$. The remaining profiles are generated during the search by sampling the image around the current position of the landmark. The best match happens to be at offset 1.
In practice, Mahalanobis distance of the normalized intensity gradient is used rather than the simple skyline match indicated here.

The distance between a search profile g and the model mean profile $\bar{g}$ is calculated using the Mahalanobis distance

$$\text{Distance} = (g - \bar{g})^T S_g^{-1} (g - \bar{g}) \qquad (2.6)$$

(This is called the squared Mahalanobis distance, since it simplifies to the square of the Euclidean distance when the covariance matrix $S_g$ is an identity matrix.) One of the search profiles will have the lowest distance. It is the position of the center of this profile (which is an offset along the whisker) that is the new suggested position of the landmark.

This process is repeated for each landmark (as shown in step 3 in Figure 2.3) before handing control back to the shape model.

## 2.5 Multi-resolution search

There is one more wrinkle. Before the search begins, we build an image pyramid, and repeat the ASM search (Figure 2.3) at each level, from coarse to fine resolution. Each image in the image pyramid is a down-scaled version of the image above it (Figure 2.8). This is a simple pyramid - more sophisticated pyramids (not used in this project) can be found [4].

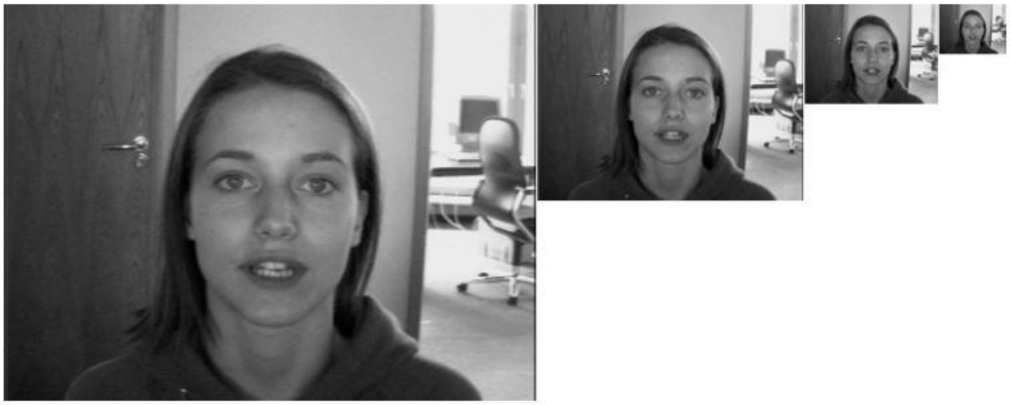


*Figure 2.8: An image pyramid.*
Each image has a quarter of the resolution of the image above it.

The start shape for the first search (which is on the coarsest image) is the shape generated from the global face detector. The start shape at subsequent levels is the best face found by the search at the level below.

We need to decide when convergence has been reached so we can move to the next level, or, at the top level, announce the search complete.

The shape model is the same across all pyramid levels (apart from scaling), but a separate profile model is needed for each level.

Using this multi-resolution approach is more efficient, more robust, and converges correctly to the correct shape from further away than searching at a single resolution.

# 3. Results

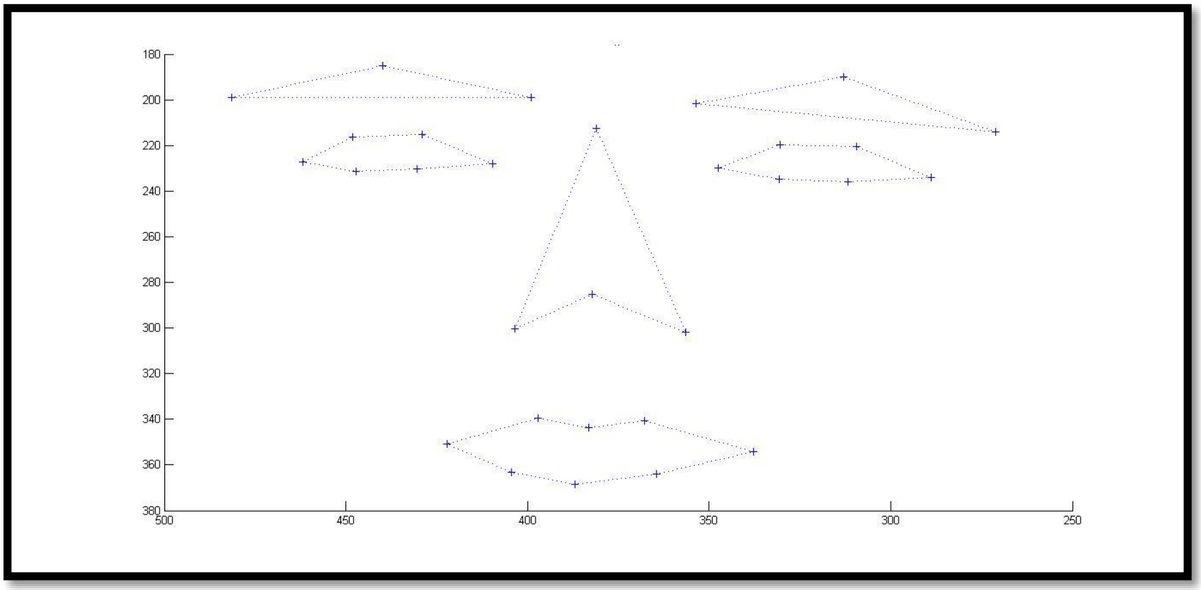


*Figure 3.1: Mean Shape obtained from training with 6 images (6 contours and 30 point per image).*

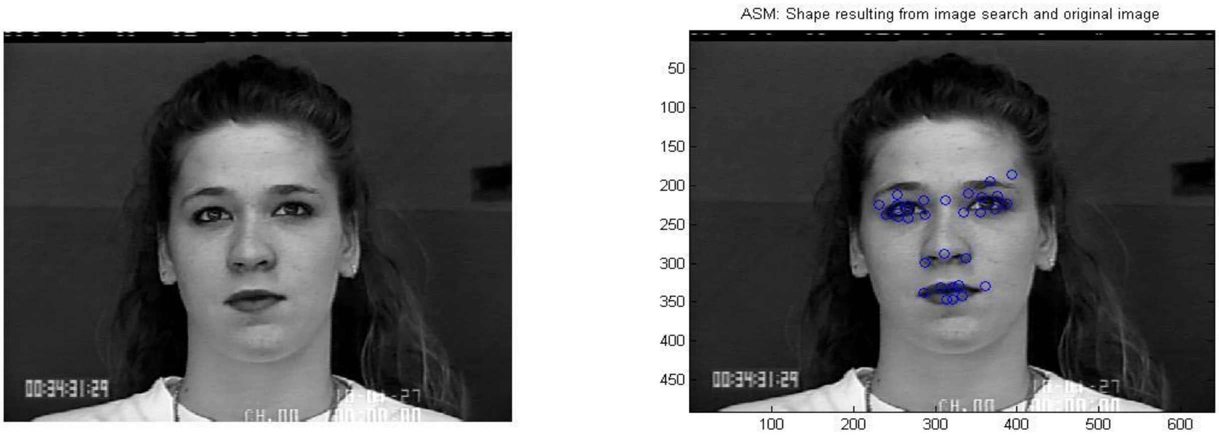


*Figure 3.2: Test Image & ASM shape resulting from image search*

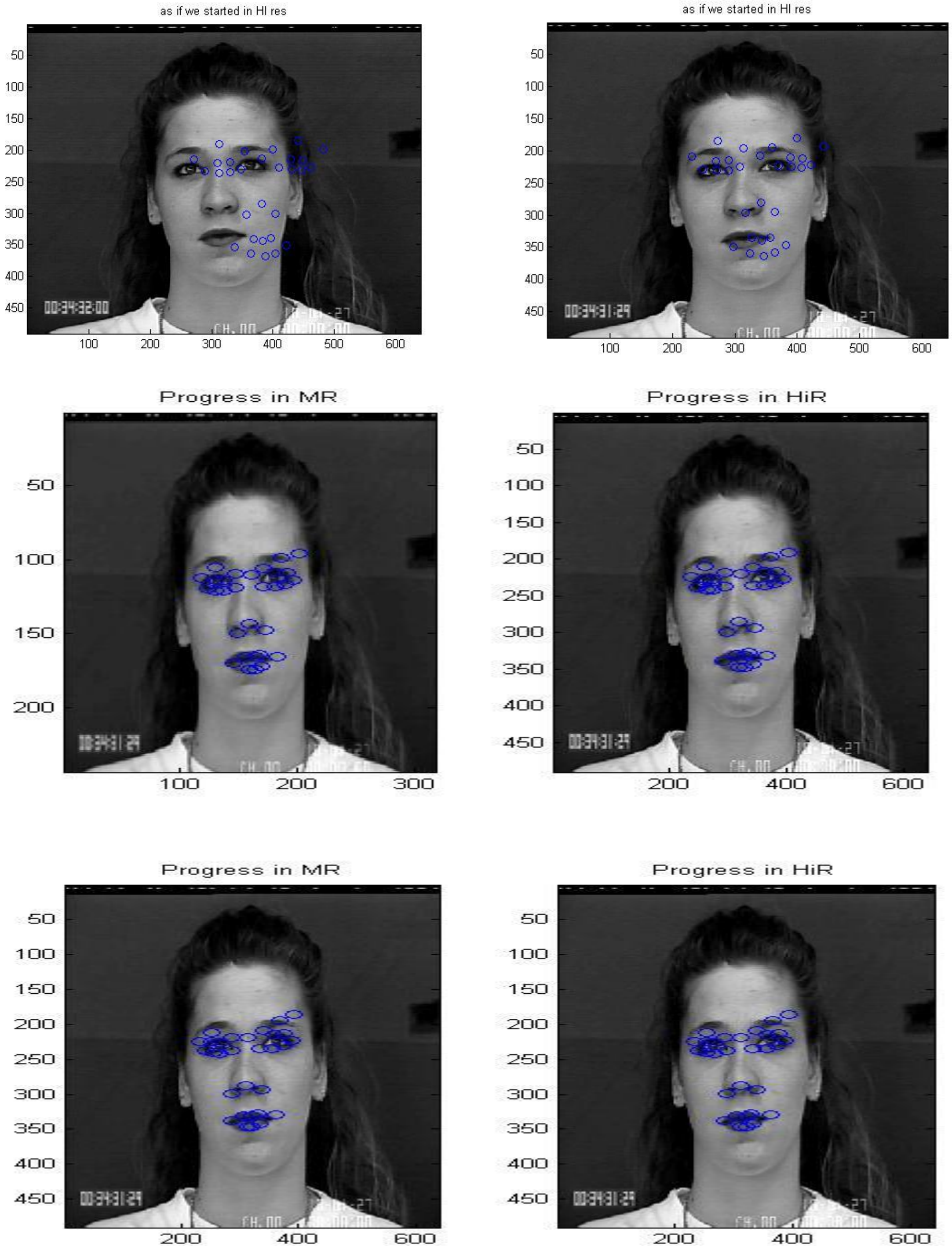


*Figure 3.3: Top Left- Mean Shape (Initial shape), Top Right- Mean shape properly initialized, All other- Intermediate Shapes*

# 4. Conclusions

The Active Shape Model is one of the simplest methods for locating facial features. By focusing on the specific application of locating features in monochrome frontal upright faces with neutral expressions. Active Shape Models can be competitive with more sophisticated methods.
On arbitrary images, computers are not yet as accurate as human landmarkers. The methods in this project are one way of getting closer to that goal.

The error in the best fit reduces with increase in number of training images and also with increase in number of landmarks.
One important point to note while labelling landmarks is to ensure that consecutive landmarks are labelled at maximum possible distance as shown in Figure A.1. This is quite helpful in reducing the number of iterations to get the best fit and further reducing the error in best fit.

Here is a list of few other possibilities for further work.
The model could be extended to include the whole head and not just the face. This could be done by building a larger shape model, or by combining two models: one for the face and one for the head.
It is usual in ASMs to reduce non-linearity by using tangent space projections [3]. Tangent spaces were not used in this project, but may produce slightly better fits.

# 5. Appendix A

## Face landmarks:

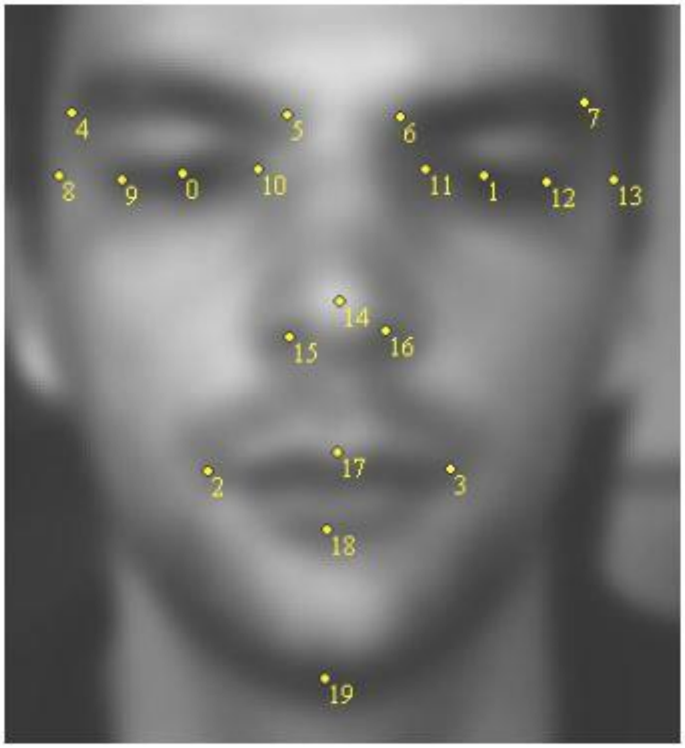


*Figure A.1*

# 5. References

1. Wikipedia http://en.wikipedia.org/wiki/Active_shape_model
2. T.F. Cootes, C.J. Taylor, D.H. Cooper and J. Graham "Active Shape Models- Their Training and Application", in Computer Vision and Image Understanding Vol.61, No. 1, January, pp.38-59, 1995
3. T. F. Cootes and C. J. Taylor. Technical Report: Statistical Models of Appearance for Computer Vision. The University of Manchester School of Medicine, 2004. www.isbe.man.ac.uk/~bim/Models/app_models.pdf.
4. R. C. Gonzalez and R. E. Woods. Digital Image Processing, 2nd Edition. Prentice Hall, 2002. www.imageprocessingplace.com.