TAMPERE UNIVERSITY OF TECHNOLOGY

**Zhe Sun**

# Active Shape Model with Applications in Facial Landmark Localization

Master of Science Thesis

Examiner: University Lecturer Heikki Huttunen
Examiner and topic approved by the Department of Signal Processing
on 12 November 2015

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing
**Sun, Zhe:** Active Shape Model for Shape Description
Master of Science Thesis, 42 pages
March 2016
Major: Information Technology for Health and Biology
Examiner: University Lecturer Heikki Huttunen
Keywords: Active Shape Model, Facial Landmark Localization, Stacked Model, Component Based Model

Active Shape Model (AAM) and Active Appearance Model (AAM) are the commonly used methods in facial landmark localization, in the past 20 years, there are a few extended methods, which are based on the classical ones, being published.

In this master thesis, the classical Active Shape Model (ASM) and Active Appearance Model (AAM) are well studied and summarized, especially ASM, which is also introduced in formulation. Two newly extended versions based on Active Shape Model are introduced and implemented through the thesis work. Stacked Active Shape Model (Stasm), which is much closer to the classical ASM, achieves a very good result on frontal face image landmark detection, so that it is the emphasis of this thesis. Besides we use Component based ASM as the comparison method, which is another Active Shape Model method based on component analysis. We performed these two methods for facial images from different situations: frontal and non-frontal images, single and group images. From the observation and data results, we show that Stasm still has room for improvement on facial feature localization. We explore the theoretical differences of these two extended versions and propose ideas for improvement in the later chapters.

# PREFACE

The master thesis which you are reading now is written during the period from November 2015 to April 2016. The research works have been done in the department of signal processing, Tampere University of Technology.

As a start, I would like to show my appreciation to my thesis supervisor, Hekki Huttunen. I still remember the first day when we start to discuss about the thesis topic and after that, the original task had been changed a little bit from vertebral images to face database, because of the data sources. Also during the whole work period, Heikki offers me a lot of materials, and most of them turn out to be the theoretical basement for this thesis. Besides, I would like to thank for the kindness helps from research assistant, Andrei Cramariuc, he helped me quite a lot on the compiling work about Stasm library. At the end, I express my gratitude to all of you, who helped me during the thesis work, friends all over the world, family members in China, and especially David Adcock from United States.

The six months of thesis work is a neither long nor short time. As for long, I am learning new knowledge on everyday and for short, this is going to be an end. During the half an year, I trained my independent working and studying skills, which, I am sure, it will help me a lot in the future life and work.

At the past hundreds of days' studying in TUT, I have grown up from a university freshman, who knows nothing about my major and even worries the life in Finland, to be the graduate. In the past days, I tried almost all the aspects of my major, information technology for health and biology, and finally, chose the field which is based on machine learning and pattern recognition. In the past days, I became mature not only physically, but also mentally.

As an ending, I would like to say the last 'thanks' to the Finnish winter.

20th of March 2016 in Tampere, Finland.

# CONTENTS

# TERMS AND DEFINITIONS

| | |
|---|---|
| $\lambda$ | Standard deviation |
| $\theta$ | Orientation Parameter |
| $\mathbf{b}$ | Value of shape vector |
| $\mathbf{F}$ | Error measure function |
| $n$ | Landmark points |
| $\mathbf{P}$ | Covariance matrix |
| $s$ | Scale Parameter |
| $\mathbf{X}$ | Model instance |
| $\mathbf{x}$ | Vector for example element |
| $\mathbf{Y}$ | Image points |
| AAM | Active Appearance Models |
| ASM | Active Shape Models |
| CompASM | Component based ASM |
| HOG | Histogram of Oriented Gradient |
| PCA | Principle Component Analysis |
| PDM | Point Distribution Model |
| STASM | Stacked Active Shape Models |

# 1. INTRODUCTION

Years ago, when we watched the science fiction movies, we were so surprised by the surreal products. Nowadays, with the popularity of concepts such as virtual reality, robots and artificial intelligence, as the fundamental problem, object detection attracted a lot attention. For human beings, after years of study, we get to know everything in our world and can recognize the objects naturally. But how can we let the computer do the same thing like a human being? For one example, how do we recognize a friend from a group photo. Probably, we need to locate all the faces in the image first. Then use the feature of friend's face, such as what his nose, eyes, mouth look like, to match with every single face in the image until we found the right one.

In the last 20 years, face recognition research has been rapidly expanded by all over the world, since there are quite a lot of potential applications in the field like computer vision. With the development of face detector, we can easily found the faces from images, but in order to get more information about them, facial landmark description becomes one important part for further analysis on image alignment, emotion recognition and pose estimation. However, as other recognition tasks, because faces have many variations, such as facial expression, face direction and image resolution, facial landmark description is not straightforward.

So what is landmark and what is facial landmark? After the definition of landmark points have been given in 1991 by Bookstein [1], a lot of research has been done on how to find a flexible model. Generally speaking, landmarks are the points on the object which represent the significant features. In a human face for instance, the points on nose, eyes and mouth are the facial landmarks.

Based on a lot of previous works, Professor T.F Cootes and his colleagues found "the only realistic approach is to 'learn' specific patterns of variability from a representative training set of the structures to be modeled."[2] In 1995, they published the method, 'Active shape models (ASM)'. In ASM, a profile model, which represents the local statistical features around each landmark, improves the accuracy on find the 'best' candidate for each target landmark. Besides, a shape model constrains the global shape. When give a new face to the model, ASM fits the mean shape to the initialized landmarks, then calculates each landmarks independently according to the profile template matching, at the end, matches the global shape model to the

adjusted point set. This method allows the user to obtain a best shape, orientation, scale and position of the object in images, by just offering an initial rough guess. At that time, the first few results of Active Shape Models are based on resister model, heart model, hand model and worm model, where a better result shows that this method has a potential for more complicated shapes, such as faces.

In 1998, the Cootes team introduced Active Appearance Model (AAM) [23]. The aiming of AAM is to express a image in terms of a set of model parameters and provides a natural interface to applications of face recognition. In AAM, besides the shape and profile model which was introduced in Active Shape Model, a series of new parameters which formed a texture model has been used to improve the accuracy of facial landmark detection. And in 2001, an extended version of AAM has been published. Since then, ASM and AAM have been mentioned so many times in the following years for the field of object detection.

Although ASM and AAM have contributed a lot to the field of object detection, there are still have shortages on real world problem. ASM works to find the best contour of the object based on the initial guess, but the error can not be fixed, if the initial guess is error located. Due to biased algorithm, many of the individual points might be not correct, so that the global shape model becomes wrong. On the other hand, AAM requires high quality of illumination. Huge error results can be caused from the difference between training and test sets. Meanwhile, AAM Model training is a time consuming procedure.

In order to solve the existing potential disadvantages and to keep the original ASM and AAM growing, in the following years, research based on face aligning and face landmarks initialization have been done. A few extended version of Active Shape Model are created, such like Stacked Active Shape Model (Stasm) [21], Component-based Active Shape Model (CompASM) [16] and Texture-constrained Active Shape Models [22].Among these extensions, Stasm features with using 2D-profile model to capture the local information for each landmark, while Comp-based ASM implements component decomposition and uses PCA to model the relevant positions between components.

The purpose of this thesis is to study the shape description via using Active Shape model. In order to explore the principle of ASM, we choose Stacked Active Shape Model as the primary method and as a comparison, Component-based Active Shape Model is included.

The content of this thesis is structured as follows. Charpter 2 gives the theoretical background of the two key words of the topic, landmark localization and Active Shape Model. A basic idea will be introduced, together with terms which are commonly used in describing. Chapter 3 presents implementation of ASM in detail, especially the training and searching stage of ASM. Two extended versions of

ASM are introduced here, together with their features. Models trained by Stasm and Comp-base ASM are presented in Chapter 4 and in order to make the measurement, the theory of me17 measure, which is used to measure the accuracy of the model via calculating the distance between the results and true position, is introduced. Chapter 5 contains the discussion based on the topic of Active Shape Models and some ideas for further improving Stasm. In the last chapter 6, we make a ending with concluding the meaning of facial landmark localization for further use.

# 2. THEORETICAL BACKGROUND

## 2.1 Landmark Localization

Landmark localization is a critical aspect in most of the computer vision applications, such as object recognition, image reconstruction and movement detection.

In old English, the word "landmark" was used to describe an "object set up to mark the boundaries of a kingdom, estate, etc.""general sense of "conspicuous object in a landscape" is from 1560s. Modern figurative sense of "event, etc., considered a high point in history" is from 1859.

In medical image processing field, landmark comes from the significant point in images of biological and medical specimens, which is used to examine and measure shape changes. Bookstein calls the representative points "landmark points" and describes them in terms of their usefulness. [1]

The characteristic of the landmark points, which are not only represent their own locations but also have the "same" locations in every other form of the study and in the average of all the forms of a data set, makes them become the most effective way to analyze the forms of whole biological organs or organisms in modern biological and biomedical investigations.

In our purposes, there are three different types of landmarks[2]:

1. points marking parts of the object with particular application-dependent significance, such as the center of an eye in the model of a face or sharp corners of a boundary;

2. Points marking application-independent things, such as the highest point on an object in a particular orientation, or curvature extrema;

3. other points which can be interpolated from points of type 1 and 2; for instance, points marked at equal distances a round a boundary between two type 1 landmarks.

For easily understanding, here, we give one example. Figure 2.1 is a 32 points model of the boundary of a resistor. As we can see, points 0, 3, 5, 10, 12, 15 and so on are marked on the location, where the easily identified features are, so they are the landmark type 1. While the other points are equally marked along the boundary between type 1 landmarks, they are type 3 landmarks.

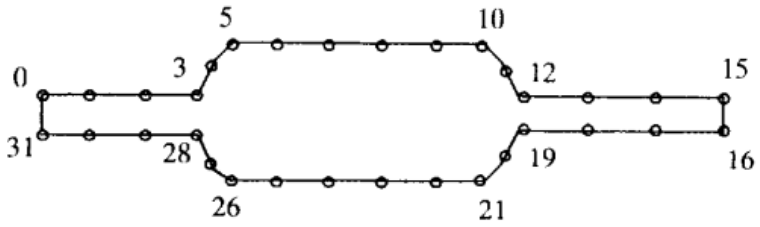



Figure 2.1: Example of 3 landmark types. [2]

## 2.2 Active Shape Model and Active Appearence Model

Models which are used in facial landmark localization have two categories, based on the types of constraint imposing: parametric methods and non-parametric method.

Active Shape Models (**ASM**) and Active Appearance Models (**AAM**) are the two most commonly used landmark localization methods which are using parametric shape constraints. Briefly, in ASM, a point distribution model represents the shape of landmark points. In AAM, the appearance is modeled by Principle Component Analysis (PCA) on the mean shape coordinates. We will introduce these two principals in the following.

### 2.2.1 Shapes and Shape Models

Before we introduce the Active Shape Models (ASM), there are a few terms that need to be explained. In this section, we will describe them in general based on our purposes.

The **shape** of an object is represented by a set of $n$ points, which may be in any dimension. Shape is usually defines the quality of a configuration of points, which is invariant under some transformation.[3]

In our case, a shape is a set of points in two dimensions. In the shape, points are related to each other, which keeps the shape in a stable condition when it is moved, rotated or scaled.

A **shape model** defines a set of shapes, which is achieved by aligning the training shapes. According to the description in Active Shape Models [2], the algorithm to align a set of $N$ shapes is showed below:

- Rotate, scale, and translate each shape to align with the first shape in the set.
- **Repeat**

  Calculate the mean shape from the aligned shapes.

  Normalize the orientation, scale and origin of the current mean to suitable defaults.

  Realign every shape with the current mean.

  **Until** the process converges.

### 2.2.2 The Active Shape Model

Active shape models were developed by Prof. T.F.Cootes and his colleagues. [2] This method can be used for image search in an iterative refinement algorithm analogous to that employed by Active Contour Modes, as known as Snakes. Figure 2.2 shows one shape model which was trained by a few images of Prof.Cootes' face. In figure 2.2, we can see there are a few shapes. Different lines shows a different situation which is generated from the training set, such as face direction, the shape of the mouth and close and open mouth.

Figure 2.2: Example of Shape Model with different variations (program to generate the image can be obtained from Prof. Cootes' website)

The **Point Distribution Model (PDM)** is a shape description technique that is used in locating new instances of shapes in images. It has been developed by Prof. Cootes and Taylor [4], which becomes a standard in computer vision for the statistical study of shape and for segmentation for medical images. This method tries to "understand" the shape, but not just building a rigid model.

Briefly, implementing of PDM method starts by aligning the training samples which have been well labeled into a common co-ordinate frame, same like what we do to obtain the shape model.The PDM approach assumes the existence of a set of examples which comprise the training set. Then from the training set, a statistical description of a shape and its variation are derived. Figure 2.3 shows one example point distribution model, where dots mark the possible positions of landmarks and the line denotes the mean shape.

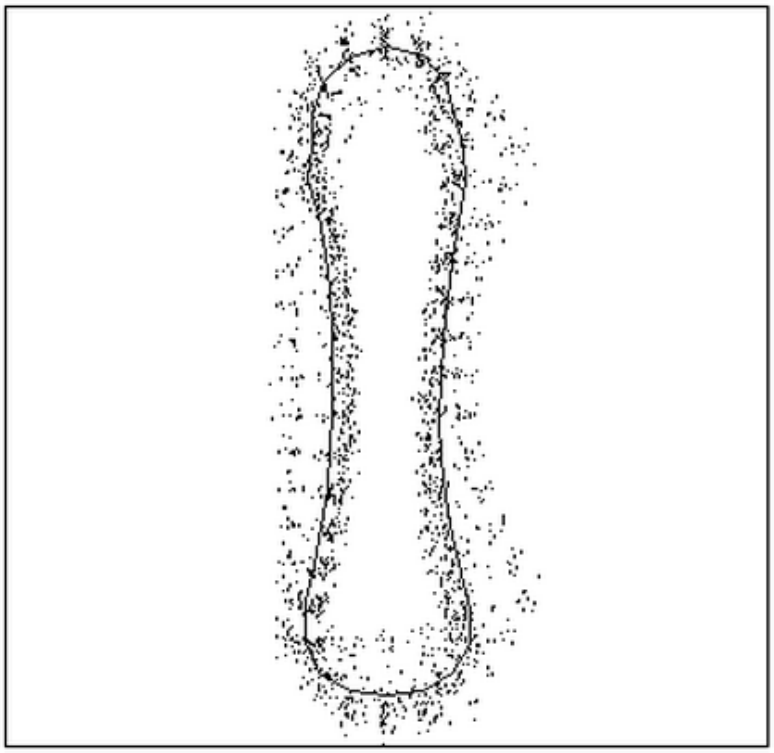


Figure 2.3: PDM of a metacarpal. Courtesy N.D.Efford, School of Computer Studies, University of Leeds.

Besides on 2D images, the Point Distribution model can be extended to deal with volume data. Because 3D images are not included in my thesis, such models and their use in image search can be found in other articles [5]. PDM can also be used in a classifier to estimate the mean shape for a set of given shapes. The distributions of the parameters can be estimated from the training set, allowing probabilities to be assigned. [6] This technique has been successfully used to recognise simple handwritten characters and faces. [7]

Generally speaking, the Active Shape Model algorithm is using Point Distribution Model in image search, more elaborately, PDM is used as a local optimiser.

Suppose we have a PDM of an object, and we have an estimate of the position, orientation, scale and shape parameters of an example of the object in an image. The approach we use to improve the estimate as follows: we calculate a suggested movement for every point in the model which is required to displace the point to a better position; we calculate the needs for overall position changing, in order to obtain the best displacements; any residual differences are used to deform the shape of the model object by calculating the required adjustments to the shape parameters. To do these, two types of sub-models are needed to construct the ASM:

1. a profile model for every landmark, which describes the characteristics of the image around the landmark.

2. a shape model which defines the allowable relative position of the landmarks.

We can understand the two sub-models in this way: the profile model is used for locating every landmark in the model in order to get a perfect location, but the shape model defines the relationship between two or more landmarks, so that the whole shape will not be deformed to a totally strange one after a few changes.

### 2.2.3 Active Appearance Models

Active Appearance Models are developed after Active Shape Models. Prof.Cootes and his colleagues brought the idea in 2001. [8] The AAM performs a full model of appearance, which contains both shape variation and the texture (intensity) of the region covered by the model. [9] Figure 2.4 shows an trained example of Prof.Cootes' face.

An **appearance model** can represent both the shape and texture variability seen in a training set. [9] Generally, the appearance modeling has following steps [8]:

---

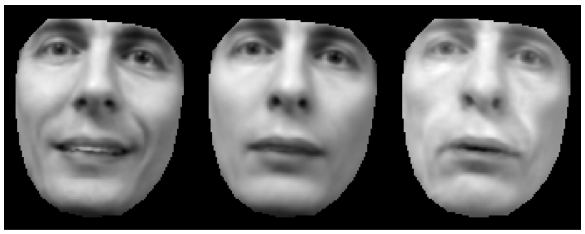
1. prepare a well annotated training set, which the corresponding points have been marked on each sample.

2. apply Procrustes analysis, which is a form of statistical shape analysis used to analyse the distribution of a set of shapes, to align the sets of points and build a statistical shape model.

3. warp each training image so the points match those of the mean shape, obtaining a "shape-free patch".

4. learn the correlations between shape and texture are learned to generate a combined appearance model.

(a) Example Shape Model



(b) Example Texture Model

(c) Example Combined Model

Figure 2.4: Example of Active Appearance Model

### 2.2.4 Difference between ASM and AAM

Based on the understanding of these two algorithms, there are three key differences between Active Shape Models and Active Appearance Models: 1. texture model producing: the texture model in ASM comes from a small region around every landmark point, while the AAM uses the appearance model from the whole region; 2. area sampling: the ASM searches around the current position, typically along profiles normal to the boundary, whereas the AAM only samples the image under the current position; 3. distance minimising: the ASM essentially seeks to minimise the distance between model points and the corresponding points in the image, whereas the AAM seeks to minimise the difference between the synthesized model image and the target image. [9]

## 2.3 Principal Component Analysis

Principal Component Analysis (PCA) was invented in 1901 [29], which has been showed to be the simplest multivariate analyses tool via using true eigenvectors.

The most common process of PCA can be done by eigenvalue decomposition of a data co-variance matrix. Here we are showing steps to perform a PCA on a set of data.[30]

---

1. Get some data.
2. Subtract the mean from each of the data dimensions.
3. Calculate the covariance matrix.
4. Calculate the eigenvectors and eigenvalues of the covariance matrix.
5. Choose components and form a feature vector.
6. Derive the new data set.

---

Nowadays, researchers need to deal with huge amount of data, which are always in high dimensions. Principal Component Analysis has one advantage on supplying the user with a lower-dimensional picture. Besides, PCA is also used as the feature selection and feature extraction method. As we showed above, after the calculations, we use t only the first few principle components, so that the dimensions of the dataset are reduced.

Back to our case, PCA is used in the profile search, where features near the target landmark need to be collected. We will introduce more in the implementation chapter.

## 2.4 Cross Validation

Suppose we have a model with a few parameters, and a dataset, which we used for training. The training process is aiming to optimize the parameters to fit the training dataset as well as possible. But one situation that we need to take into consideration is that the series of parameters turn out to be not fit the data outside of the training set. This is the so called overfitting.

Cross validation is one method to evaluate the performance of the trained model, in order to prevent overfitting, especially when the training set is small or the number of parameters in the model is large. Basically, we can divide cross validation into two groups, exhaustive cross validation, for one example, leave-one-out cross validation, and non-exhaustive cross validation, like 2-fold cross validation. Besides, another goal of cross validation is comparing the performances of different models.

# 3. IMPLEMENTATION

In machine learning and pattern recognition field, almost all the solutions can be divided into a few stages, So do facial landmark localization problems. In this chapter, we are going to explain the implementation stages based on our experiment, which is modified for our purpose and shown in Figure 3.1.

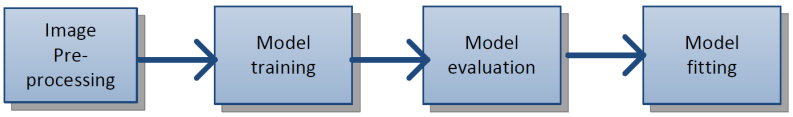



Figure 3.1: Processing

**Image pre-processing** refers to using image processing techniques, such as image segmentation, image alignment, before dealing the raw images with the system, in order to improve the quality. In our case, although we are using some of the open source data sets, some of the image are not well aligned, which makes them difficult to be used directly by the following steps.

**Model training and Model evaluation** are the core steps of our processes. Based on the method of Active Shape Model, in the training stage, a lot of parameters are involved into our model building, so that each landmarks can be matched to the correct locations. After the training stage, in order to test the accuracy and avoid the over fitting, models need to be evaluated before using in new test. The popular method is cross validation.

**Model fitting** means fitting the well trained and tested model to a new image. Based on the algorithm, this is also called 'searching stage'. In this step, a new image or new data set will be introduced into the system, via using the model on it, we can get the data for presenting the model performance.

**Results presenting** is the last stage of our work. At this stage, we use the data that we collect from the previous steps to prove the advantage and disadvantage of the method we choice. Via comparing with other research results, we try to figure out the reason and further improvement.

In order to performing this experiment, there are a few open source data sets, in which have been well annotated, can be used. After the original method of Active

Shape Model was published in 1995 [2], a few extended research have been done in the past decades around this topic, meanwhile, some efficient approaches have been published based on them. Stacked active shape model, which is known as 'Stasm', is one of the most popular methods and our experiment are mainly performed on it. As a comparison, the classical ASM and one new method, Component-based ASM will be mentioned, too.

## 3.1 Image Data Set

In our experiment, we choose to use a few public face databases. MUCT (Milborrow/University of Cape Town) [10], which is also the original training database of 'Stasm' program [10]. Another image data set in this thesis is Helen Facial Feature Data set [16], which is used data set of the method, Comp-based Active Shape Model.

### 3.1.1 MUCT

Figure 3.2: Example images from MUCT database [10]

MUCT (Milborrow/ University of Cape Town) database consists 3755 face images, with approximately equal numbers on male and female images, and all images are taken from people with different occupations such as students, teachers, employees of the university and so on. Besides, subjects were not asked to show any

particular facial expression, in other words, all images were taken with a neutral expression or a smile. Figure 3.2 shows some example images from MUCT database.

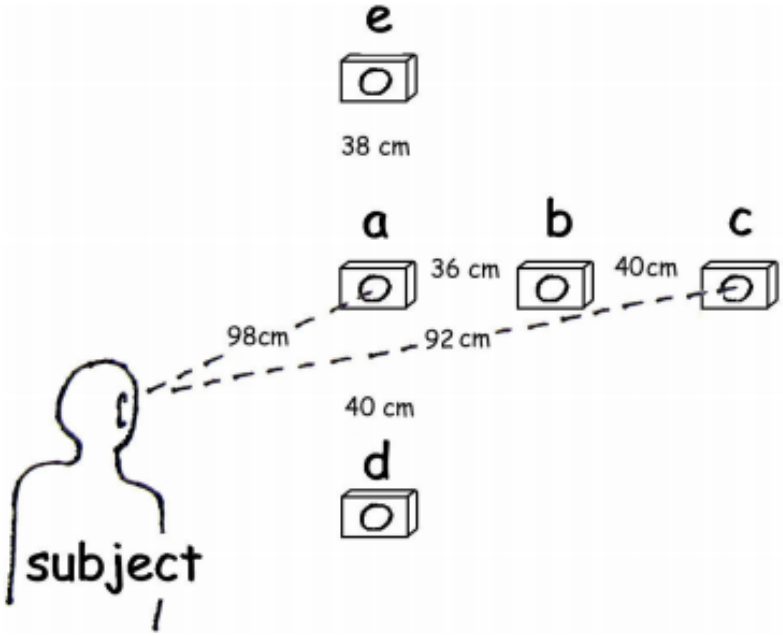



Figure 3.3: The five cameras and their positions to the subject's face [10]

In MUCT database, every subject has five images which are from different angles taken at the same time. The positions of 5 cameras is shown as in Figure 3.3 and Figure 3.4 is one set of examples which are taken from all five cameras.

Landmark is another issue that we need for our experiment. The Stasm was trained with 76 landmarks on each sample. In our case, in order to compare with other results, we are using 68 landmarks for the database. The landmark number and its location have been well defined beforehand (Figure 3.5 and Figure 3.6).

### 3.1.2 Helen Facial Feature Data set

Helen Facial Feature Data set which contains 2330 images is constructed by images from Flickr. The aiming of this database is offering high resolution examples for facial feature localization. The image in the data set have detected by a face detector and, in order to simplify the further use, some images have been cropped from the original version. In the end, all the images are manually annotated with 196 landmarks.

Helen Facial Feature Data set is more challenging and diverse than other data sets. Figure 3.7 shows some example images from Helen Facial Data set. As we can see the face in the image have more facial expressions, which creates the difficulties on landmark locating.

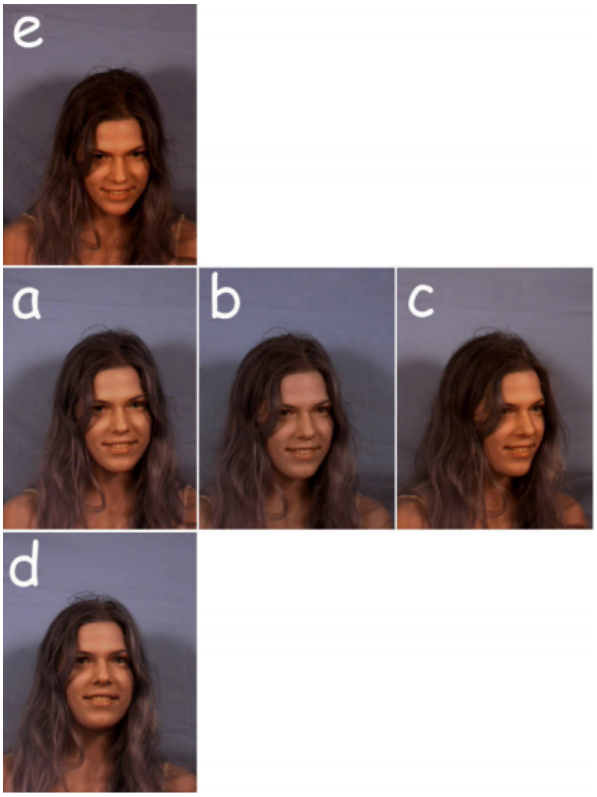



Figure 3.4: Five image views of one subject

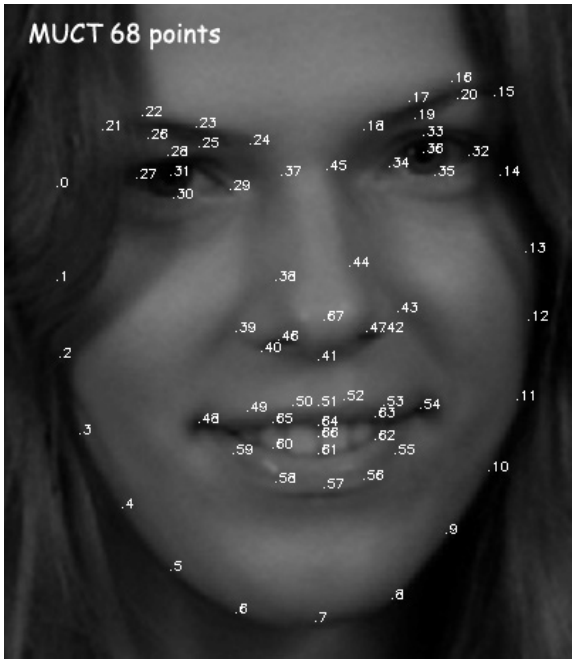



Figure 3.5: Image with 68 landmarks

During the whole process, a few other image data sets are involved, such as BioID [15], and XM2VTS [25]. Because they are not the mainly used data set, We are not going to introduce them one by one here, instead of some information in Table 3.1.

```
//   par pre next weight bits
    { 14,  1, 32,  1., AT_Beard|AT_Glasses      }, // 00 LTemple
    { 13, -1, -1,  1., AT_Beard|AT_Glasses      }, // 01 LJaw1
    { 12, -1, -1,  1., AT_Beard                 }, // 02 LJaw2
    { 11, -1, -1,  1., AT_Beard                 }, // 03 LJaw3
    { 10, -1, -1,  1., AT_Beard                 }, // 04 LJaw4
    { 9,  -1, -1,  1., AT_Beard                 }, // 05 LJaw5
    { 8,  -1, -1,  1., AT_Beard                 }, // 06 LJaw6
    { -1, -1, -1,  1., AT_Beard                 }, // 07 CTipOfChin
    { 6,  -1, -1,  1., AT_Beard                 }, // 08 RJaw6
    { 5,  -1, -1,  1., AT_Beard                 }, // 09 RJaw5
    { 4,  -1, -1,  1., AT_Beard                 }, // 10 RJaw4
    { 3,  -1, -1,  1., AT_Beard                 }, // 11 RJaw3
    { 2,  -1, -1,  1., AT_Beard                 }, // 12 RJaw2
    { 1,  -1, -1,  1., AT_Beard|AT_Glasses      }, // 13 RJaw1
    { 0,  13, 27,  1., AT_Beard|AT_Glasses      }, // 14 RTemple
    { 21, 14, 16,  1., AT_Glasses|AT_Hat        }, // 15 REyebrowOuter
    { 22, 14, 17,  1., AT_Glasses|AT_Hat        }, // 16 REyebrowTopOuter
    { 23, 16, 18,  1., AT_Glasses|AT_Hat        }, // 17 REyebrowTopInner
    { 24, 17, 29,  1., AT_Glasses|AT_Hat        }, // 18 REyebrowInner
    { 25, 18, 20,  1., AT_Glasses|AT_Hat        }, // 19 Point19
    { 26, 27, 32,  1., AT_Glasses|AT_Hat        }, // 20 Point20
    { 15,  0, 22,  1., AT_Glasses|AT_Hat        }, // 21 LEyebrowOuter
    { 16,  0, 23,  1., AT_Glasses|AT_Hat        }, // 22 LEyebrowTopOuter
    { 17, 22, 24,  1., AT_Glasses|AT_Hat        }, // 23 LEyebrowTopInner
    { 18, 23, 34,  1., AT_Glasses|AT_Hat        }, // 24 LEyebrowInner
    { 19, 24, 26,  1., AT_Glasses|AT_Hat        }, // 25 Point25
    { 20, 32, 27,  1., AT_Glasses|AT_Hat        }, // 26 Point26
    { 32, 28, 30,  1., AT_Glasses|AT_Eye|AT_Hat }, // 27 LEyeOuter
    { 33, 27, 29,  1., AT_Glasses|AT_Eye|AT_Hat }, // 28 LEyeTop
    { 34,  1, 13,  1., AT_Glasses|AT_Eye|AT_Hat }, // 29 LEyeInner
    { 35, 27, 29,  1., AT_Glasses|AT_Eye|AT_Hat }, // 30 LEyeBot
    { 36, 27, 29,  1., AT_Glasses|AT_Eye|AT_Hat }, // 31 LPupil
    { 27, 33, 35,  1., AT_Glasses|AT_Eye|AT_Hat }, // 32 REyeOuter
    { 28, 32, 34,  1., AT_Glasses|AT_Eye|AT_Hat }, // 33 REyeTop
    { 29, 13,  1,  1., AT_Glasses|AT_Eye|AT_Hat }, // 34 REyeInner
    { 30, 32, 34,  1., AT_Glasses|AT_Eye|AT_Hat }, // 35 REyeBot
    { 31, 32, 34,  1., AT_Glasses|AT_Eye|AT_Hat }, // 36 RPupil
    { 45, 27, 51,  1., AT_Glasses|AT_Hat        }, // 37 LNoseTop
    { 44, -1, -1,  1., AT_Glasses|AT_Hat        }, // 38 LNoseMid
    { 43, -1, -1,  1., AT_Mustache|AT_Hat       }, // 39 LNostrilBot0
    { 42, -1, -1,  1., AT_Mustache|AT_Hat       }, // 40 LNostrilBot1
    { -1, 40, 42,  1., AT_Mustache|AT_Hat       }, // 41 CNoseBase
    { 40, -1, -1,  1., AT_Mustache|AT_Hat       }, // 42 RNoseBot1
    { 39, -1, -1,  1., AT_Mustache|AT_Hat       }, // 43 RNoseBot0
    { 38, -1, -1,  1., AT_Glasses|AT_Hat        }, // 44 MRNoseMid
    { 37, 32, 51,  1., AT_Glasses|AT_Hat        }, // 45 RNoseTop
    { 47, 31, 47,  1., AT_Mustache|AT_Hat       }, // 46 LNostril
    { 46, 36, 46,  1., AT_Mustache|AT_Hat       }, // 47 RNostril
    { 54, 51, 57,  1., AT_Mustache|AT_Hat       }, // 48 LMouthCorner
    { 53, -1, -1,  1., AT_Mustache|AT_Hat       }, // 49 Mouth49
    { 52, -1, -1,  1., AT_Mustache|AT_Hat       }, // 50 Mouth50
    { -1, -1, -1,  1., AT_Mustache|AT_Hat       }, // 51 TopOfTopLip
    { 50, -1, -1,  1., AT_Mustache|AT_Hat       }, // 52 Mouth52
    { 49, -1, -1,  1., AT_Mustache|AT_Hat       }, // 53 Mouth53
    { 48, 51, 57,  1., AT_Mustache|AT_Hat       }, // 54 RMouthCorner
    { 59, 54, 56,  1., 0|AT_Hat                 }, // 55 Mouth55
    { 58, -1, -1,  1., 0|AT_Hat                 }, // 56 Mouth56
    { -1, -1, -1,  1., 0|AT_Hat                 }, // 57 MouthBotOfBotLip
    { 56, -1, -1,  1., 0|AT_Hat                 }, // 58 Mouth58
    { 55, 48, 58,  1., 0|AT_Hat                 }, // 59 Mouth59
    { 62, 48, 61,  1., 0|AT_Hat                 }, // 60 Mouth60
    { -1, 60, 62,  1., 0|AT_Hat                 }, // 61 Mouth61
    { 60, 54, 61,  1., 0|AT_Hat                 }, // 62 Mouth62
    { 65, 48, 54,  1., 0|AT_Hat                 }, // 63 Mouth63
    { -1, 63, 65,  1., 0|AT_Hat                 }, // 64 Mouth64
    { 63, 48, 54,  1., 0|AT_Hat                 }, // 65 Mouth65
    { -1, 48, 54,  1., 0|AT_Hat                 }, // 66 MouthCenter
    { -1, 48, 54,  1., 0|AT_Hat                 }, // 67 NoseTip
```

Figure 3.6: Landmark number and its definition

Figure 3.7: Examples from Helen Facial Feature Dataset

Table 3.1: Basic information of BioID and XM2VTS Databases

| | BioID | XM2VTS |
|---|---|---|
| Number of Landmarks | 20 | 68 |
| Number of Images | 1521 | 2360 |
| Image Size | 384x286 | 720x576 |

## 3.2 Classic ASM Model Training in Formulation Expression

In order to locate a structure of interest, we must first build a model of it. Based on annotated images, we must decide upon a suitable set of landmarks. So suppose the landmarks along a curve are labelled $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. For a 2-D image we represent the $n$ landmark points, $\{(x_i, y_i)\}$, for a single example, as the $2n$ element vector,$\mathbf{x}$, where

$$\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^T. \tag{3.1}$$

If we have $s$ training examples, we generate $s$ such vectors $\boldsymbol{x}_j$. Before we can perform statistical analysis on these vectors, it is important that the shapes represented are in the same co-ordinate frame. The shape of an object is normally considered to be independent of the position, orientation and scale of the object. A square, when rotated, scaled and translated, remains a square.

### 3.2.1 Statistical Models of Shape

Suppose now we have $s$ sets of points $\boldsymbol{x}_i$ which have been aligned into a common co-ordinate frame. One simple iterative aligning method is as follow:

---

1. Translate each example so that its center of gravity is at the origin.
2. Choose one example as an initial estimate of the mean shape and scale so that $|\overline{x}| = 1$.
3. Record the first estimate as $\overline{x_0}$ to define the default reference frame.
4. Align all the shapes with the current estimate of the mean shape.
5. Re-estimate mean from aligned shapes.
6. Apply constraints on the current estimate of the mean by aligning it with $\overline{x_0}$ and scaling so that $|\overline{x}|$=1.
7. If the mean shape still changes significantly, return to **4**.

---

These vectors are from a distribution in the $2n$ dimensional space in which they live. If we can model this distribution, we can generate new examples, similar to those in the original training set, and we can examine new shapes to decide whether they are plausible examples.

In order to simplify the problem, we are trying to reduce the dimensionality of the data from $2n$ to a more manageable level. Here, we are using Principal Component Analysis (PCA), after that, we can approximate any of the training set, $x$, using:

$$\mathbf{x} \approx \overline{\boldsymbol{x}} + \mathbf{P}\mathbf{b}. \tag{3.2}$$

Where $\mathbf{P}$=($\boldsymbol{p}_1$ | $\boldsymbol{p}_2$ | ... | $\boldsymbol{p}_t$) contains $t$ eigenvectors of the covariance matrix and $\mathbf{b}$ is a $t$ dimensional vector of weights given by

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \overline{x}). \tag{3.3}$$

The vector $\mathbf{b}$ defines a set of parameters of a deformable model. By varying the elements of $\mathbf{b}$, we can vary the shape, $\mathbf{x}$, using Equation (3.2). The variance of the $i^{th}$ parameter,$b_i$, across the training set is given by $\lambda_i$. By applying limits of $\pm 3\sqrt{\lambda_i}$, since most of the population lies within three standard deviations of the mean [2], to the parameter $b_i$, we ensure that the shape generated is similar to those in the original training set.

We usually call the model variation corresponding to the $i^{th}$ parameter, $b_i$, as the $i^{th}$ component of the model. The eigenvectors, $\mathbf{P}$, define a rotated co-ordinate frame, aligned with the cloud of original shape vectors. The vector $\mathbf{b}$ defines points in this rotated frame. Here we are showing one example, where explains the shape model

Figure 3.8: Example face image annotated with landmarks

and the effect of varying the model shape parameters. Figure 3.9 shows example shapes from a training set of 300 labelled faces, which Figure 3.8 is one example of the training set. Each image is annotated with 133 landmarks and there are 36 parameters in the shape model. (More details about this example can be found from

[11])

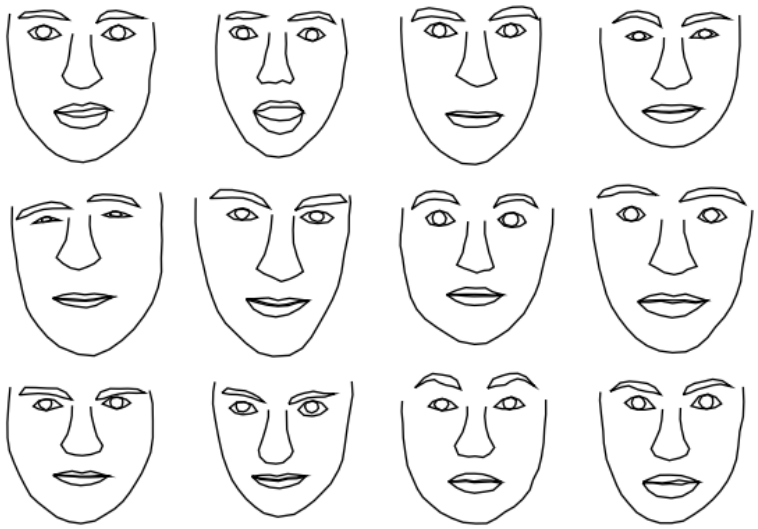


Figure 3.9: Example shapes from training set of faces [2]

While Figure 3.10 shows the effect of varying the first three shape parameters in turn between ±3 standard deviations from the mean value, leaving all other parameters at zero.

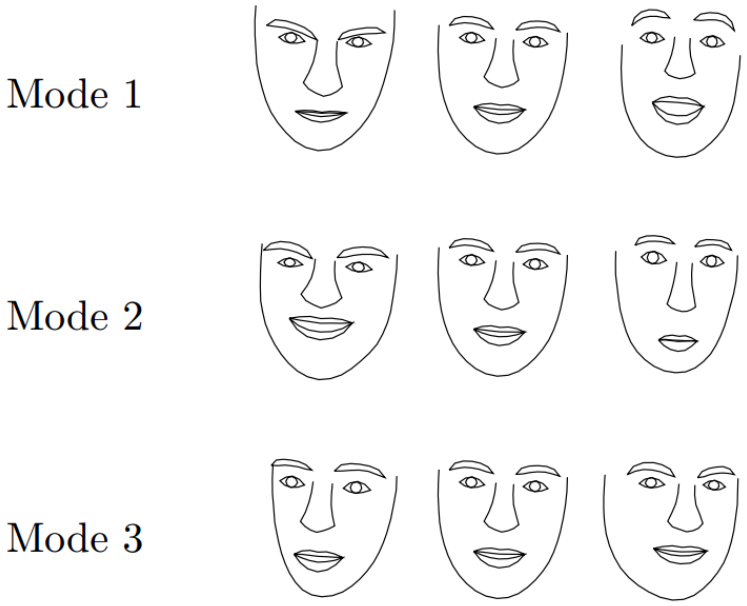



Figure 3.10: Effect of varying each of first three face model shape parameters in turn between ±3 s.d. [2]

### 3.2.2 Fitting a Model to New Points

From the parameters above, a particular value of the shape vector, **b**, corresponds to a point in the rotated space described by **P** which corresponds to an example model. This can be turned into an example shape using the transformation from the model coordinate frame to the image coordinate frame. Typically this will be a Euclidean transformation defining the position, ($X_t$, $Y_t$), orientation, $\theta$, and scale,$S$, of the model in the image.

The positions of the model points in the image, **X**, are then given by

$$\mathbf{X} = T_{X_t,Y_t,S,\theta}(\overline{x} + \mathbf{Pb}) \tag{3.4}$$

Where the function $T_{X_t,Y_t,S,\theta}$ performs a rotation by $\theta$, a scaling by $S$ and a translation by ($X_t$, $Y_t$). For instance, if applied to a single point ($\boldsymbol{x}$, $\boldsymbol{x}$),

$$T_{X_t,Y_t,s,\theta}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} S\cos\theta & -S\sin\theta \\ S\sin\theta & S\sin\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{3.5}$$

Suppose now we wish to find the best pose (translation, scale and rotation) and shape parameters to match a model instance **X** to a new set of image points, **Y**. Minimising the sum of square distances between corresponding model and image points is equivalent to minimising the expression

$$|\mathbf{Y} - T_{X_t,Y_t,S,\theta}(\overline{x} + \mathbf{Pb})|^2 \tag{3.6}$$

A simple iterative approach to achieving this is as follows:

---

1. Initialize the shape parameters, b, to zero (the mean shape).
2. Generate the model point positions using $\mathbf{x} = \overline{x} + \mathbf{Pb}$
3. Minimize 3.6 to find the pose parameters $(X_t, Y_t, S, \theta)$, which best align the model points **x** to the current found points **Y**.
4. Project **Y** into the model coordinate frame by inverting the transformation $T$:

$$\mathbf{y} = T^{-1}_{X_t,Y_t,S,\theta}(\mathbf{Y}) \tag{3.7}$$

5. Project **y** into the tangent plane to $\overline{x}$ by scaling: **y**'=**y**/(**y**.$\overline{x}$).
6. Update the model parameters to match to **y**'

$$\mathbf{b} = \mathbf{P}^T(\mathbf{y}' - \overline{x}) \tag{3.8}$$

7. If not converged, return to step 2.

---

### 3.2.3 Testing How Well the Model Generalises

From the training set, we can see the shape models are described using linear combinations of the shape. In order to generate new versions of the shape to match a new image data, the training set should contain all the possible shape models which a new image data can be expressed. If not, the model will be over constrained and will not be matched to some cases. For instance, if we train a model with all frontal face images, the images has side faces cannot be modeled via our model. Meanwhile, over-fitting is also a problem which should be took into consideration.

Cross validation is one approach to estimating how well the model will perform. 'Leave-one-out' experiments is a common way to use for cross validation. Given a training set of several examples and equally divided the whole training set into $n$ groups, build a model use *n-1* groups of samples, then fit the model to the group that has not been used in training and record the error. Repeat this until all individual group has been tested. If the error is unacceptably large for any example, more training samples should be required. While, small errors for all examples only mean that there is more than one example for each type of shape variation, not that all types are properly covered. It is good to calculate the errors for all points to ensure the maximum error on any point is sufficiently small.

## 3.3 Model Fitting in Searching Stage of Classic ASM

In order to fit the model to a new image, a set of parameters which defines the shape and position of the target object should be chosen. For a set of model parameters, $c$, we can generate an instance of the model projected into the image. Via comparing this model with the target image, we can get a fit function *F(c)*. What should be done next is finding the suitable parameters to optimize this measure.

Thus, theoretically, all we have to do is choosing a suitable fit function and optimizing the measure.

### 3.3.1 Choosing a Fit Function

A fit function represents the probability how the model parameters fit to the target, *P(c|I)*. So choosing a fit function is to maximize the probability.

As we described above, the parameters that we can vary are the shape parameter, $\mathbf{b}$, and the pose parameters $X_t$,$Y_t$,$S$,$\theta$. Let us assume that the shape model represents the boundaries of the target object, a useful measure is calculating the distance between a given model and the nearest point on the boundary of the object.

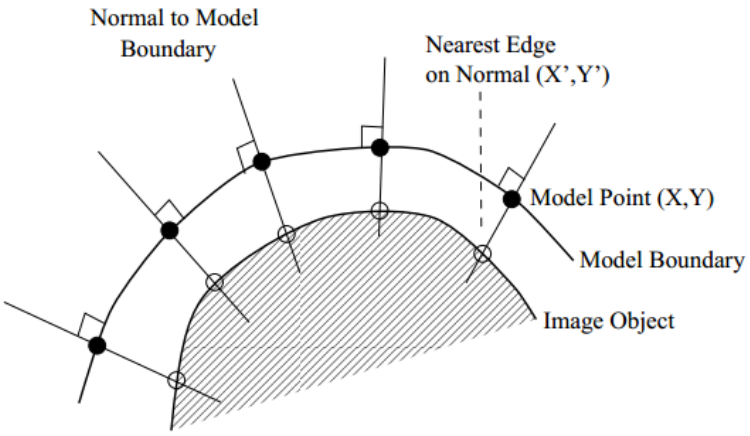



Figure 3.11: One measurement via calculating the distance between the model point and its nearest point on the boundary of the object

If the positions of the model points are given in vector $\mathbf{X}$, and corresponding points on the target are $\mathbf{X'}$, then the measurement is

$$F_{X_t,Y_t,S,\theta} = |\mathbf{X}' - \mathbf{X}|^2 \tag{3.9}$$

### 3.3.2 Optimizing the measurement

Optimizing the fit model is very difficult, if we do not know any initial knowledge about where the target is in the image. However, after the prior processing, we have already get an approximation position. so we can use local optimization techniques.

So far, we can give a general algorithm about the classical ASM method:

1. Examine a region of the image around each point $\mathbf{X}_i$ to find the best nearby match for the point $\mathbf{X}'_i$
2. Update the parameters $X_t$,$Y_t$,$S$,$\theta$,$b$ to best fit the new found points $\mathbf{X}$
3. Apply constraints to the parameters, $\mathbf{b}$, to ensure plausible shapes
4. Repeat until convergence.

## 3.4 One Extension of ASM: Stasm and Its Features

Stasm [19], which is short for "stacked Active Shape Model", is based on the method of Active Shape Model to find the feature in faces. Both the model training and face searching have been already integrated inside the program. Furthermore, Stasm is written by C++ and needs OpenCV implementation [20] for face detectors.

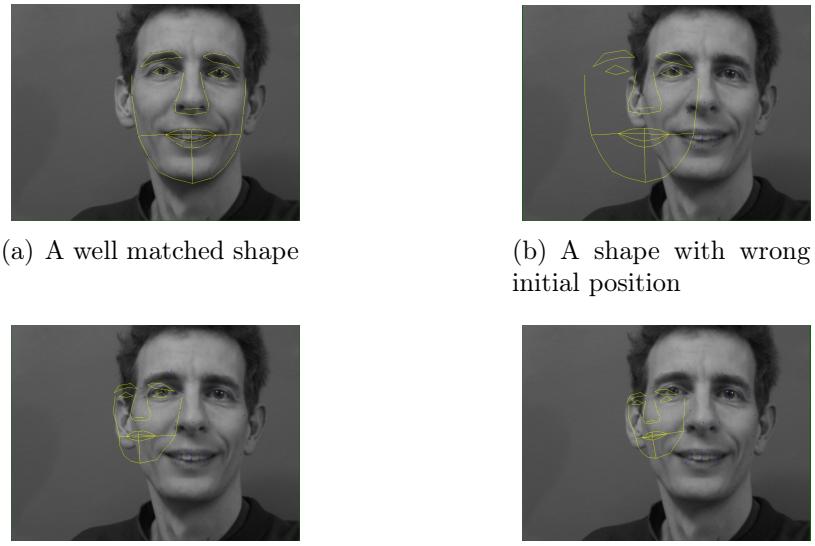
The version we are using in our experiment is 4.1.0. Here are some features about Stasm, compared with the classical ASM method.

### 3.4.1 Two Dimensional Profiles

Different with the classical ASM, which is using one-dimensional profile at each landmark and using two-dimensional "profiles" to improve fits, Stasm samples a square region around each landmarks, instead of sampling a 1D line of pixels. Intuitively, the 2D profile area can obtain more information around the landmark which can also offer a better result on model fitting.

### 3.4.2 Stacking Models

In the classical ASM, the initial location of the shape fitting is crucial. Manually giving a wrong location can cause a final failure from the beginning. Figure 3.12 is showing one failure example that shape fitting on a wrong initial location. In Stasm, a better way to solve the problem is to run two times ASM searches in series, via using the first search result as the start shape for second ones. The stacking model helps the worst fits, where the initial location is often badly misplaced, but has little effect where the start shape is already well positioned.



(a) A well matched shape

(b) A shape with wrong initial position

(c) After 1 iteration

(d) After 10 iterations

Figure 3.12: One example of wrong initial position on shape searching

### 3.4.3 Generating the start shape using face detector

Because of the importance of initial position. In Stasm, in order to avoid the problem stated in Figure 3.12, global face detectors are used before the ASM search begins. It helps to locate a approximate position and size of the face. Rowley face detector [17] is used in the developing stage of Stasm and Viola-Jones detector[18] is more used in the released version 4.1.0.

## 3.5 Another Extension: Component Based ASM

In the classical Active Shape Model, a global shape model has been used to analyse the entire shape and each feature point is sampled by a local profile search. Although we can get quite good results on studio data where the variances are small and test images are similar to training samples, in real life, because of the large variations on face expression, head direction and light illumination, the global ASM is too strict to get a better result.

### 3.5.1 Component Based ASM

Based on the observation which we talked above, the component based ASM [26] has been presented originally in 2007. Instead of using a global shape, a few separated local models have been introduced to describe the whole object. In a face shape for instance (Figure 3.13), the set of landmarks for the shape model has been decomposed into 7 local models, "left brow", "right brow", "left eye", "right eye", "nose", "mouth", "jawline". Those 7 local models are called components. In component based model, each component has its own coordinate system, local frame. and the center of the component are represented in a higher level dominant face frame, configuration frame.

The model that we are talking here is slightly different with the original Component Based ASM. In this thesis, Comp-based ASM refers to the later version, which is published in 2012 [16].

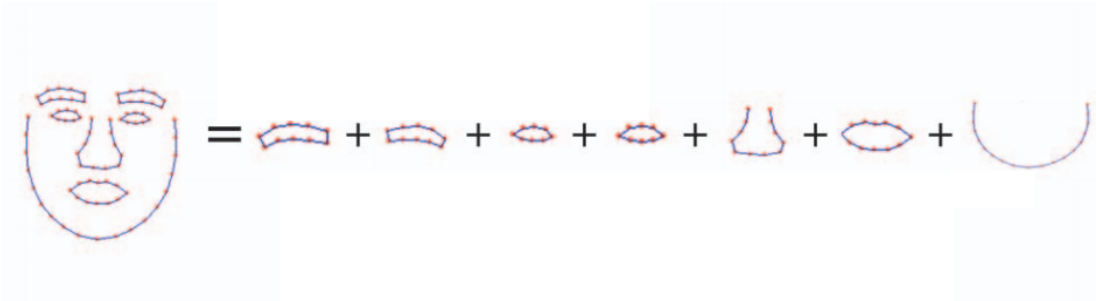


Figure 3.13: Local shape model of a face shape

According to [16], there are three coordinate frames: global, configuration and local frames. In Comp-based ASM, the local models for each component are used to fitting the components' shape model to the relevant components and they are constrained by the configuration model and the configuration model is responsible to find the orientation and scale of the face and to estimate optimal locations of components. In other words, because each local model is fitted independently, Comp-based ASM can handle larger global shape variations.

Another difference with the Classical ASM is the profile model. In Classical ASM, profile model is a one dimensional vector obtained by searching the surrounding target landmark. Because all the landmark locations are chosen individually, it could end up with the two landmarks far away from each other, which they should be neighbors. In Comp-based ASM, there is a binary constraint for each pair of adjacent landmarks, which keep them in a proper distance from each other. Elaborately, the landmark locations fitting is done by maximizing the unary scores at each location and binary scores at each pair of adjacent locations.

In the end, we present the algorithm for Comp-based ASM:

---

1. Detect faces, initialize the shape model based on face rectangle.
2. Do profile search for suggesting new landmark locations
3. Update the landmark locations with local shape and configuration model
4. Form a new global shape by applying the inverse similarity transformation
5. if the locations of points are not stable, go back to step 2.

---

### 3.5.2 Features of Comp-Based ASM

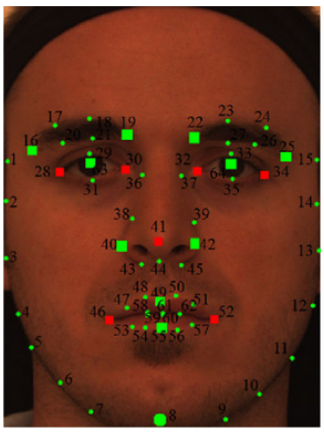
Comp-based ASM is published in C++ code under dlib C++ library. Dlib is a general purpose cross-platform open source software library written in the C++ programming language. [27; 28] The implementation of dlib is well documented in its manual which easy for understanding by users.

Besides, Comp-based ASM uses a face detector which is made using the classic Histogram Oriented Gradient (HOG) feature combined with a linear classifier, an image pyramid and sliding window detection scheme. [28] This face detector has more powerful function compared with Viola-Jones face detector, so that we still can finish the task under extreme situations, but as a drawback, it requires much more computation during the process.

# 4. PERFORMANCE EVALUATION

## 4.1 Type of Landmarks in Performance evaluation

Different with the landmark type in annotation stage, the landmarks which we choose here can be detected relatively easily using low-level image features such as gradient information, cornerness or local information extracted, which means that they are based on the abundance and reliability of image features aiding their detection. These landmarks, easily detectable ones, such as the corners of the eyes, the corners of the mouth, the nose tip and the eyebrows, are referred to as primary ones and they play a more determining role in facial identity and face tracking [13]. Based on this, there are a few different standards. Furthermore, primary landmarks are often guiding the secondary landmarks. The primary and secondary landmarks most commonly used in the literature are shown in Figure 4.1. Compared with Figure 3.5, there is a slightly different in landmark system, but the location of each primary landmark is the same.



| Primary landmarks | | Secondary landmarks | |
|---|---|---|---|
| Number | Definition | Number | Definition |
| 16 | Left eyebrow outer corner | 1 | Left temple |
| 19 | Left eyebrow inner corner | 8 | Chin tip |
| 22 | Right eyebrow inner corner | 2-7, 9-14 | Cheek contours |
| 25 | Right eyebrow inner corner | 15 | Right temple |
| 28 | Left eye outer corner | 16-19 | Left eyebrow contours |
| 30 | Left eye inner corner | 22-25 | Right eyebrow corners |
| 32 | Right eye inner corner | 29, 33 | Upper eyelid centers |
| 34 | Right eye outer corner | 31, 35 | Lower eyelid centers |
| 41 | Nose tip | 36, 37 | Nose saddles |
| 46 | Left mouth corner | 40, 42 | Nose peaks (Nostrils) |
| 52 | Right mouth corner | 38-40, 42-45 | Nose contours |
| 63,64 | Eye centers | 47-51,53-62 | Mouth contours |

Figure 4.1: The most commonly used primary and secondary landmarks *(the green and red squares refer to the primary landmarks, in which red square shows the most fiducial points)* [13]

## 4.2 The Error Measurement

To evaluate the accuracy of feature detection, the locations found by each feature detector are compared with ground truth locations which is the manually annotations. The average point to point error ($m_e$) is calculated as follows:

$$m_e = \frac{1}{ns} \sum_{i=1}^{n} d_i. \tag{4.1}$$

Where $d_i$ is the point to point errors, $s$ is the inter-ocular distance and $n$ is the number of points. Figure is one measure example.

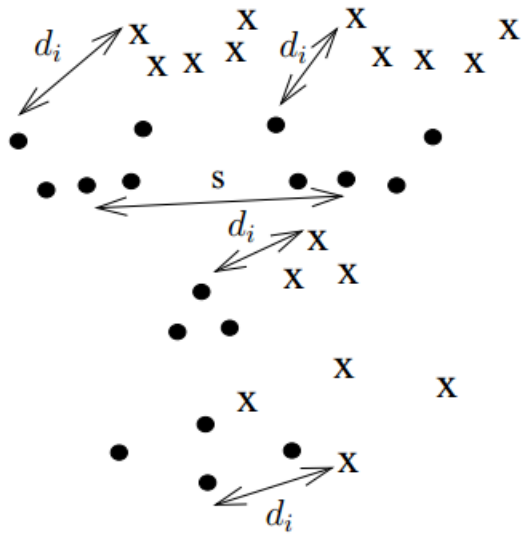



Figure 4.2: Example of error calculating, "." = annotated location and "X" = predicted location [11]

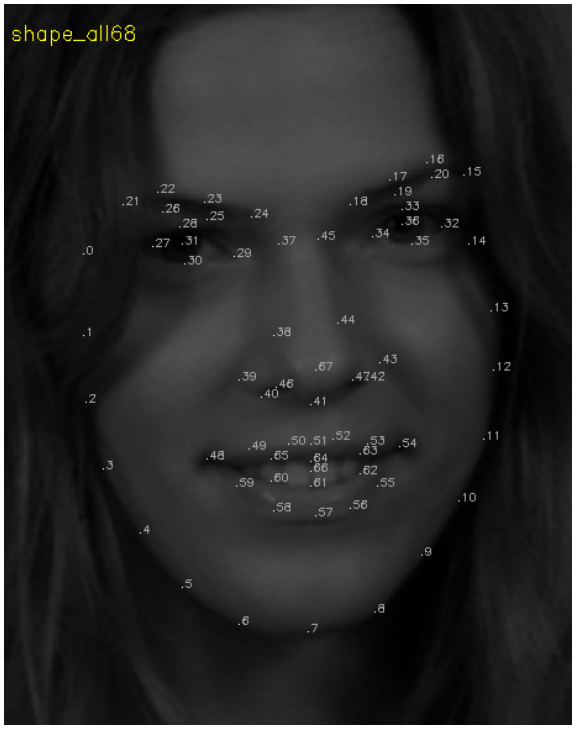
As we discussed above, commonly, we use 17 primary landmarks to refer to the whole shape, so in our case, n equals to 17. This is the so called me17 measure[14].

Besides me17 measure, there are a few other measurements using in researches, for an example me196, basically they are using the same formula as Equation 4.1 showed, the only difference is the number of landmarks they use in calculation.

## 4.3 Stasm Model Performance

In this section, we are going to show some results of the trained Stasm Models. During training stage, we use different numbers of training samples and test the model on the whole MUCT data sets. Some results, we can easily see and draw the conclusion based on the intuitionistic.

In Figure 4.3, we are showing two standard images. Figure 4.3(a) is the standard model of 68 landmarks, which shows the locations of all the 68 landmarks and their relevant positions. Figure 4.3(b) shows the standard locations of the primary landmarks, which will be used for measuring the model performance, like we discussed in last section.




(a) standard location for 68 landmarks shape


(b) A shape with wrong initial position

Figure 4.3: Standard images of shape 68 and 17

Firstly, we are going to train the new model with first 100 images in MUCT. This is a relatively arbitrary model, because the limit of the small amount images in the training set. Figure 4.4 is the mean shape which has been obtained from the first 100 images of MUCT. Like we can see from the image, the landmark locations of the right eye (landmark numbers from 32 to 36) is not accurate. Also, the outline of the shape is obviously larger then the lady's face, while mouth is much smaller than it should be.

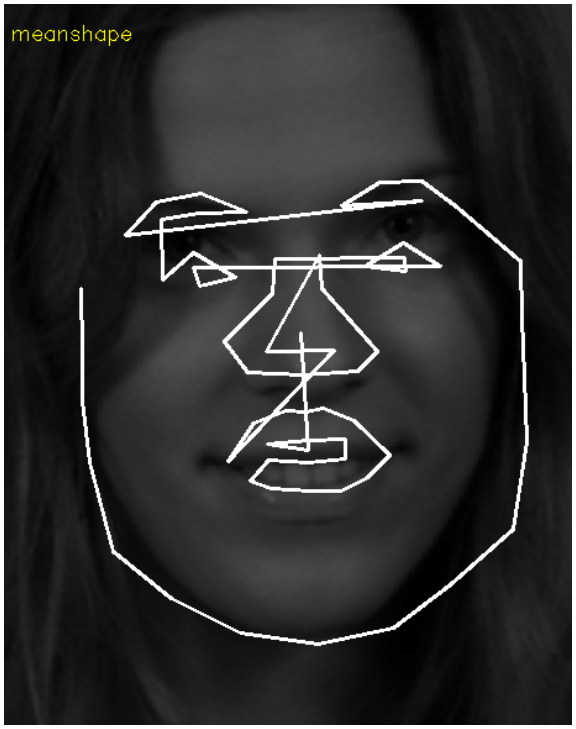
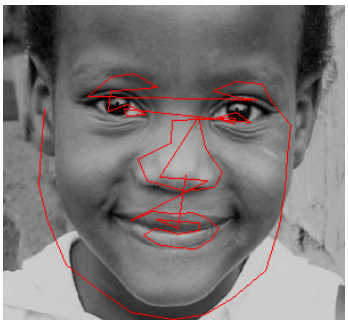



Figure 4.4: Mean shape from the training set with 100 images

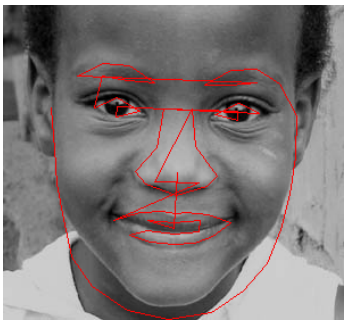
Now let us make a comparison, in Figure 4.5 , the image on the right side (fig:4.5(a)) is the fitting result from the model which is trained by 100 images, while on the left side (fig:4.5(b)) is the fitting result from the model which is trained by 2000 images. As we can see directly from the images, the model which is trained by only 100 images has less fitting accuracy than the other one, because of the locations of the eye brow, right eye and corner of the mouth. Via calculating the me17 error, which are 0.093 (4.5(a)) and 0.072 (4.5(b)), it shows the same result.

Then we fit these two models to the whole MUCT data set, which contains 3755 images, and calculate the performance of them. Figure 4.6 shows the histograms from two different models, where we can see that the error of the model which is trained by 100 images has a wider range, while the model trained by 2000 images has a more converged one.

The results in Table 4.1 are collected from 300 random images in MUCT, there are unavoidable some false images which means the face detector cannot found the



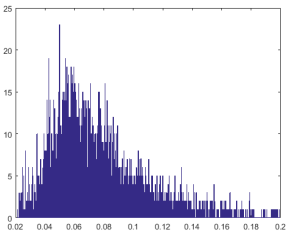
(a) model trained by 100 images



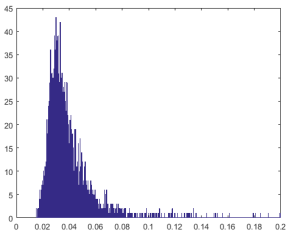
(b) model trained by 2000 images

Figure 4.5: Fitting results from different training models

face. Generally the errors for these kind of images are abnormal, so we removed the false results, in order to calculate the mean me17 error.




(a) model trained by 100 images




(b) model trained by 2000 images

Figure 4.6: Histograms from different training models

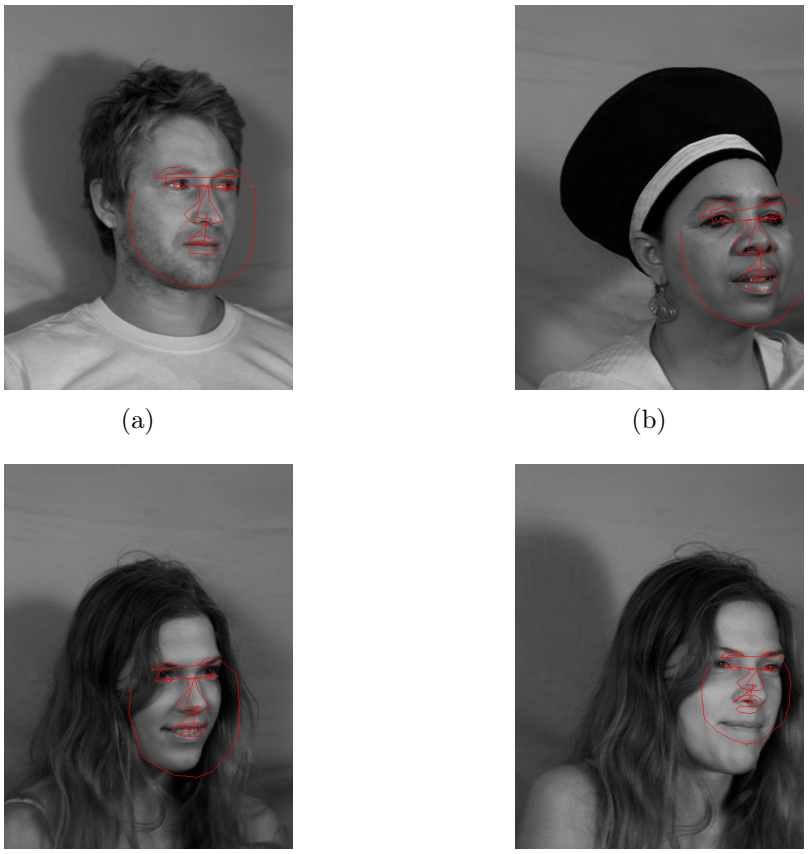
| Model | Mean | Max | Min |
|---|---|---|---|
| Model with 100 images | 0.08 | 0.433 | 0.021 |
| Model with 2000 images | 0.04 | 0.264 | 0.020 |

Table 4.1: Comparison of 100 images model and 2000 images model on random 300 images from MUCT data set under me17 measure

## 4.4 Inaccurate Situations

When we train the model with Stasm, we found there are situations either with very large me17 errors or the errors are smaller than it should be. I think it is necessary to give a few examples here.

In Figure 4.7, these images all have the me17 error higher than 0.1. Because with non-frontal faces, which Stasm has a problem on fitting the model. Also as the writer introduces Stasm, she said Stasm is mainly used to detect frontal face.



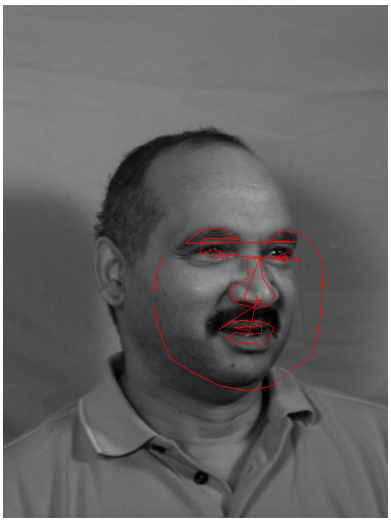
(a) (b)

(c) (d)

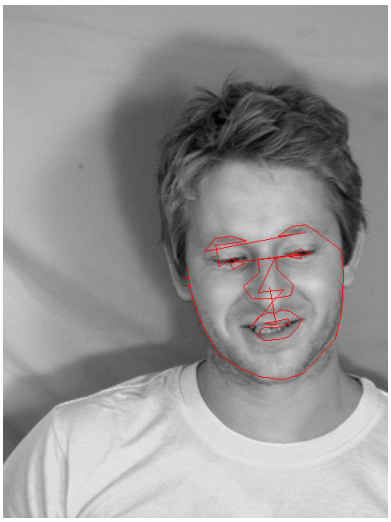
Figure 4.7: Large error with side faces

In Figure 4.8, both of the images are marked with the same me17 error, 0.09. But from the images themselves, we should say Figure 4.8(b) fits much better than Figure 4.8(b). The reason for this situation is that we do use the primary landmarks (check 4.3(b)) to calculate the me17 error, but not the landmarks on the outline of the shape. So, although the fits are quite different with each other, they still have the same me17 error.

Figure 4.9 shows some other situations which can cause abnormal error and mainly they are caused by the face detector, like 4.9(a) caused by face detector cannot find the face and 4.9(b) caused by face detector locate the wrong place. The reason for these situations are quite complicated, such like the light of the place, the texture of the clothes and the accessory of the object.

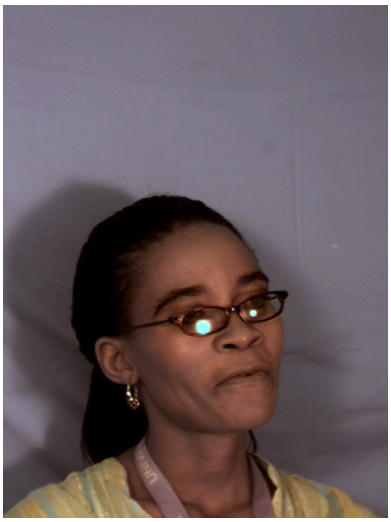


(a)



(b)

Figure 4.8: Images with the same me17 error



(a) Face detector cannot find the face

(b) Face detector locate the wrong place

Figure 4.9: some other situations

## 4.5 Performance Comparison between Two ASMs

In this section, we are going to compare the results from Stasm and Comp-based ASM, based some tested examples.

### 4.5.1 Overview

In Table 4.2, we can see that Stasm has a slightly better result than Comp-based ASM. But we should say that MUCT data set is easy to deal with the fitting algorithm, because the images are obtained under the same studio environment. While Helen data set is more close to the real life images data set, this data set contains images with larger variations on pose, lighting and facial expression. As we can see from the result, Comp-based ASM has outperformed Stasm on these complicated situations.

Table 4.2: Comparison of Stasm and Comp-based ASM on two test data sets.

| Dataset | Model | Mean | Max | Min |
|---|---|---|---|---|
| MUCT | Stasm | 0.043 | 0.19 | 0.020 |
| MUCT | Comp-based ASM | 0.045 | 0.23 | 0.021 |
| Helen | Stasm | 0.111 | 0.411 | 0.037 |
| Helen | Comp-based ASM | 0.091 | 0.402 | 0.035 |

### 4.5.2 Computational Efficiency

In this thesis, all the experiments are executed under Ubuntu 14.04.4 LTS 64bit system via using VMware Workstation as virtual machine. The CPU of the laptop which distributes to the virtual machine is one processor of Intel Core I5 2.6 GHz together with 1 GB RAM. All the calculations have been done under the same hardware condition.

During the training session, we found, Stasm needs 166ms per image, while Comp-Based ASM needs 6.5 seconds per image (ASM model training needs 1 second and 5.5s for face detector). There are a huge time difference on two methods.

While training the comp-based ASM model, we calculated the test error for different number of training images. From Figure 4.10, comp-based ASM needs a very small train data set to achieve a quite good fitting model, while from the previous Table 4.1, Stasm needs a training set which contains 100 faces to reach the mean test error 0.08, however, only 18 images are needed in comp-based model training.

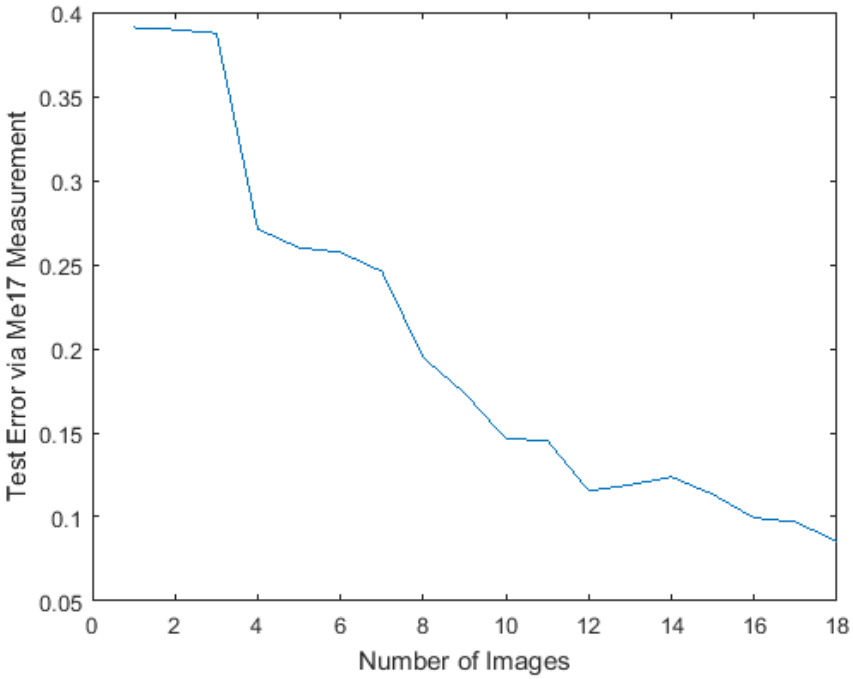



Figure 4.10: Test Error result in Comp-Based ASM

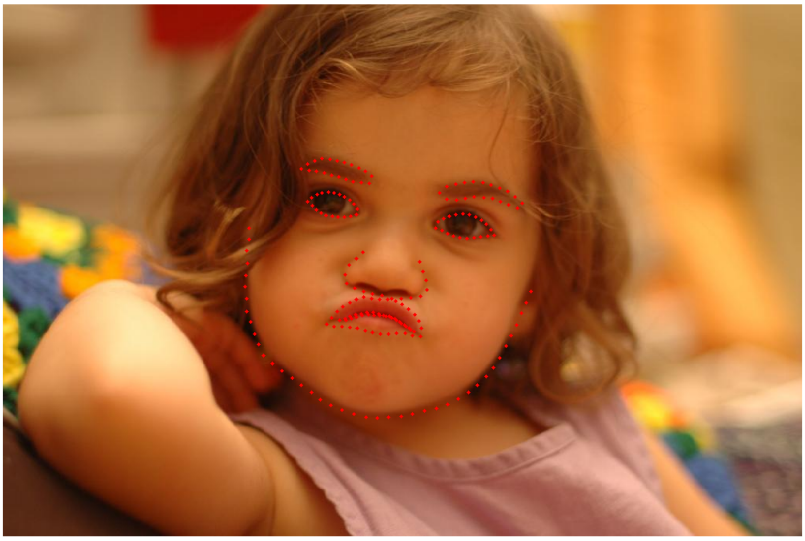
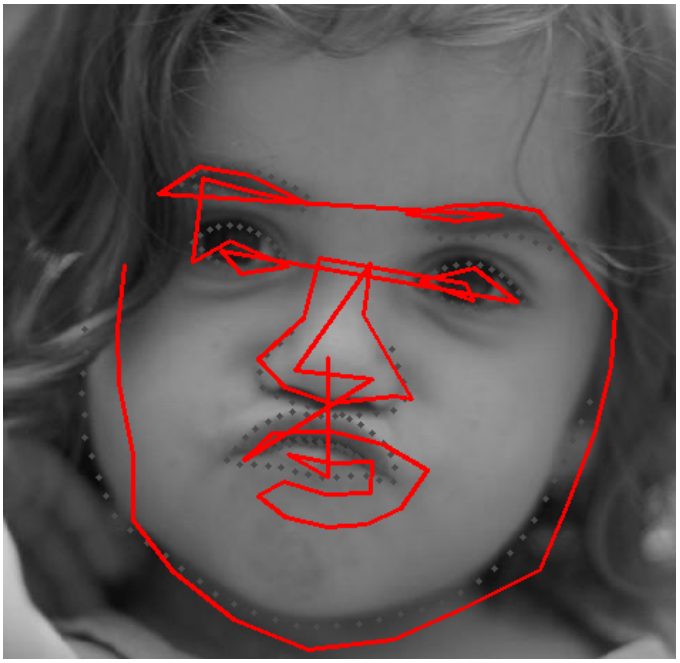


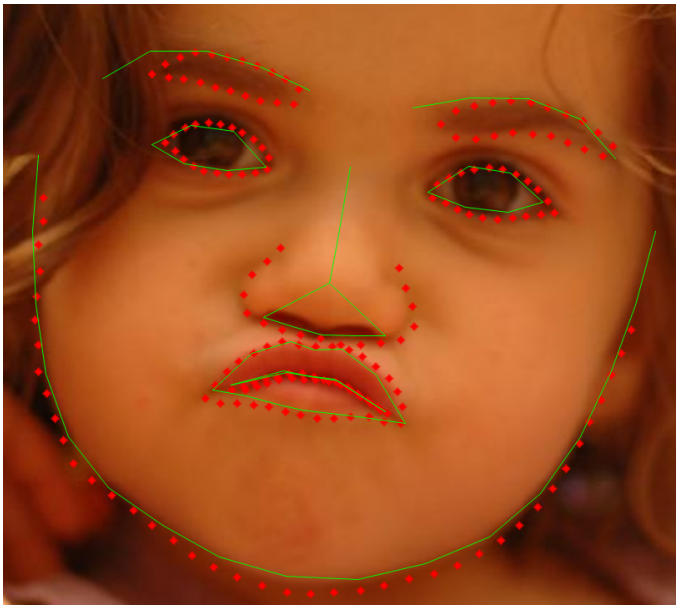
Figure 4.11: An example image from Helen data set with face gesture

### 4.5.3 Situation 1: Face with expression

In Figure 4.11, we show one example image from Helen data set which has a large variation on mouth shape and Figure 4.12 are the fitting result from both of the models. Images with non-neutral face are always causing quite a lot of fitting problems via using the classical ASM. Because of the global constrained parameters in Stasm, as it shows in 4.12(a) , the shapes for local mouth area is still in a neutral shape, while the girl is pouting in fact. As a comparison, in 4.12(b), we can see the local shape for mouth is almost perfectly overlapped with the original landmark points, which improves the fitting result quite a lot.



(a) Fitting result from Stasm. Red line shows the shape fitting by stasm



(b) Fitting result from Comp-based ASM. Green line shows the shape fitting by Comp-based ASM

Figure 4.12: Fitting results from two models. The red dots are the original landmark points.

### 4.5.4 Situation 2: Non-frontal faces

As we mentioned in last section, where we showed that Stasm doesn't work very well with the non frontal images. In comp-based ASM, this problem can also be well solved.

(a) Fitting result from Stasm

(b) Fitting result from Comp-based ASM

Figure 4.13: Stasm and Comp-based ASM on non-frontal face

### 4.5.5 Situation 3: Image with multi-faces

In real life, a commonly problem that we need to deal with is group photo (same like in Figure 4.14), which means there are more than one face in the image. Because of the large variations between each face, for this kind of images, before we match the single shape to the faces, a high sensitive face detector is highly needed.


(a)

(b)

Figure 4.14: Example images with multi-faces

Active Shape Model is originally trained with one images which contains one face only, so does Stasm, and it has the ability on matching multi faces. But, because of the limitations from the shape and profile models, the effect on this task by Stasm is really poor. Just like Figure 4.14, Stasm almost cannot work on these kind of images. Result are showed in Figure 4.15, the face detector can recognize the three

faces in the image 4.15(a), but the location is not correct, so that the fitting results confuse the outline of the face and in 4.15(b), the face detector cannot location the faces, because the size of each face is too small to detect by the used face detector, which is viola-jones detector.


Figure 4.15: Fitting result from Stasm. White dots are representing the fitting results

Now, let us show the results from Comp-based ASM model. Compared with Stasm model, the shapes constructed via using global and local Component models have been generated correctly.


Figure 4.16: Fitting result from Comp-based ASM. Green lines are representing the fitting results

# 5. DISCUSSION

In this chapter, the observed results from the presented data and image are discussed and a few ideas for further research are written.

As we can see from the examples above, the classical ASM (Stasm) has a good result on frontal facial landmark localization. Based on the training data, the shape model is well constructed with relatively flexible model.

But we should still be aware of the limitation of Stasm. Because we do not work only with frontal face image, and in real life problems, most of the images are taken in a more freedom way then sitting in front of the camera straightly. The Helen data set is one of the challenging task for Stasm.

Like in the situation 1, when the image contains face with exaggerated facial expression, the Stasm will not offer a good fitting solution based on that. On the contrary, the entire shape will be re-scaled by the little influencing factor. Parameters which construct the shapes are too strict. Once a small changing in the local area, the global shape will be changed. A better result from Comp-Based ASM just proves this idea, Where the entire face has been divided into seven components, the changing in one component will be evaluated before changing the parameters constraint the whole shape model.

Situation 2 indicates the most commonly problem in real life, which also partly results from situation 1. Furthermore, the Stasm using landmark pairs in shape model, which defines the relationship between two individual landmarks. For an example, landmark on the right corner of mouth is paired with the landmark on the left corner. For non-frontal images, the shape is not totally symmetrical. Hence, the parameter should reduce the binding of paired landmarks. And on the other hand, Stasm has a elementary implementation on facial pose estimation. While the model which is recovered from the Comp-based ASM performs very good.

Another field of multi-face face landmark localization is the idea of situation 3. And the factor which affects the result highly is the facial detector. As the first step of classical face detection, a face detector determines the final result quality. Both Stasm and Comp-Based ASM are using face detector to offering a bounding box for each face in the image. Intuitively, we can see from the result, the face detector which uses HOG feature and pose estimator work quite well for this task. However, Stasm is about a failure. After analyzing the implementation of Stasm, we found a

few reasons. Firstly, an improper face detector. During the training and test stage, we found a lot of the false examples caused by the face detector. Secondly, the profile file used in Stasm does not obtain the feature for each landmark accurately.

Besides all these situations we discussed above, there are a few other ideas which i think can improve the performance of model fitting and user experience. As we have noticed, the images processed by Stasm are all in gray level, which makes it more difficult if the surrounding is dark. And also, some information based on the colors are missed automatically.

On the other hand, Stasm is not a very handy tool for new model training. As we can found from the website, the manual for training a new model is around 30 pages. And quite a lot of constrained parameters have been used in the training session which we need to manually modify tons of files before the training start.

At the end, let us summarize a few further work for Stasm:

---

1. Separate the global shape into a few local shape for different part of the object for a better fitting in local areas
2. Include a proper face pose estimator, in order to dealing with side face images and images with other positions
3. Update to a new face detector instead of using the current one for a better detection under extreme situation, Or revisiting the start shape which is generated by Viola-Jones detector
4. Using color images to obtain more information for each landmark points
5. Improve the user experience on training new models

---

# 6. CONCLUSION

The idea of this thesis is going getting familiar with the classical shape describe method, Active Shape Model, which was originally published in 1992. The method used shape and profile models for locating the shape for new images. During the training stage, shape model, which is obtained from the Point Distribution Model, forms the entire shape of the object, while profile models collects the features around each landmark points. Afterwards, in the searching step, a new shape will be generated based on the shape of the new object. Parameters in shape model keeps the entire model in a certain shape, for instance, a face model will always be in a face shape and eyes, nose and other feature landmarks are in a solid relationship, while each landmark matches the relevant point, based on the feature which is collected beforehand in profile model.

In this thesis, two extended version of Active Shape Model are emphatically introduced. They are both based on the original Active Shape Model, while improved the model performance via using different way to generate the shape model and profile model. Meanwhile, both of the extended methods are using face detector as the first step to locate the face in the image, in order to segment the target from it surrounding environment. in Stacked Active Shape Models, an stacked model via using two times of ASM is introduced to the new method, so more accurate position of the start shape will be located. Besides, a 2D profile for one single landmark will be obtained, in order to get more efficient and correct solution for new target shape fitting.

Another method, which is included here, is Component Based Active Shape Models. Different with the classical Active Shape Model and Stacked Model, this methods forms the shape model via dividing the entire shape into a few separate local shapes and parameters are used to constrain the local shapes inside the global shape. It offers a more flexible fitting method, especially for the faces with large variations on facial gestures, face directions and environment factors.

Besides all the introductions, a few experiments based on the mentioned method have been established. One popular error measurement, which is based on calculating the primary facial landmarks distance, is used to analyzing the results. Via the comparison between the two methods, we get to know more about the advantage and disadvantage of the Active Shape Models. We do believe that adding more

landmark points which gives a more integrated shape can helps for more accurate facial landmark localization. On the other hand, how to take full advantage of the profile information is another aspect to increase the success rate of finding correct facial landmarks.

Nowadays, facial landmark localization and face detection are commonly used in the applications in computer vision and virtual reality, even in robotics. The method, Comp-based ASM, has already been used into detection based on real time and video stream. With the development of relevant field, like face detector techniques and face alignment methods, Active Shape Models will be more in used as a primary method.

# REFERENCES

[1] F.L.Bookstein, Morphometric Tools for Landmark Data, Cambridge University Press, London/New York, 1991.

[2] T.F.Cootes, C.J.Taylor, D.H.Cooper, and J.Graham, Active Shape Models– Their Training and Application, Computer Vision and Image Understanding, Vol.61, No.1,January, 1995, pp.38-59.

[3] T.F.Cootes, and C.J.Taylor,Statistical Models of Appearance for Computer Vision. The University of Manchester school of Imaging Science and Biomedical Engineering, 2004.

[4] T.F.Cootes, C.J.Taylor, D.H.Cooper and J.Graham, Training Models of Shape from Sets of Examples. University of Manchester, department of Medical Biophysics.

[5] A.Hill, T.Thornham, C.J.Taylor, Model Based Interpretation of 3D medical Image. British Machine Vision Conference.BMVA Press, 1993, pp 339-348.

[6] T.F.Cootes, A.Hill, C.J.Taylor and J.Haslam. The Use of Active Shape Models For Locating Structures in Medical Images. Image and Vision Computing Vol.12 No.6 July 1994.pp. 355-366.

[7] A.Lanitis, C.J.Taylor, T.F.Cootes. A Generic System for Classifying Variable Objects Using Flexible Template Matching. Proc. British Ma Conference 1993. pub. BMVA Press. pp. 329-338.

[8] T.F.Cootes, G.J.Edwards and C.J.Taylor. Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.23, No.6, June 2001.pp 681-685.

[9] T.F.Cootes, G.Edwards and C.J.Taylor. Comparing Active Shape Models with Active Appearance Models. British Machine Vision Conference. BMVA Press, 1999, pp 173-182.

[10] S.Milborrow, J.Morkel and F.Nicolls. The MUCT Landmarked Face Database. available at: $http://www.milbo.org/much/The-MUCT-Landmarked-Face-Database.pdf$

[11] T.F.Cootes. An Introduction to Active Shape Models.Appears as Chapter 7: "Model-Based Methods in Analysis of Biomedical Images" in "Image Processing and Analysis", Ed.R.Baldock and J.Graham, Oxford university Press, 2000, pp223-248.

[12] R.O.Duda, P.E.Hart and D.G.Stork. Pattern Classification. New York, NY, USA, John Wiley and Sons. 2001. 654p.

[13] O.Çeliktutan, S.Ulukaya and B.Sankur. A Comparative Study of Face Landmarking Techniques. European Association for Signal Processing, 2013:13.

[14] D.Cristinacce and T.Cootes. Feature detection and tracking with constrained local models. British Machine Vision Conference, pp 929-938, 2006.

[15] O.Jesorsky, K.J.Kirchberg and R.Frischholz. Robust Face Detection Using the Hausdorff Distance. In: AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-based Biometric Person Authentication, London, UK, Springer-Verlag. 2001, pp 90-95.

[16] V.Le, J.Brandt, Z.Lin, L.Bourdev and T.S.Huang. Interactive Facial Feature localization. In Europen Conference on Computer Vision, 2012, Volume 7574, pp 679-692

[17] H.A.Rowley, S.Baluja and Takeo kanade. Neural Network-Based Face Detection. Institute of Electrical and Electronic Engineers Transactions on Pattern Analysis and Machine Intelligence, Volume 20, 1998, pp 23-38.

[18] P.Viola and M.Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition Conference, Volume 1, 2001.

[19] S.Miborrow and F.Nicolls. Active Shape Model with SIFT Descriptors and MARS. VISAPP, 2014

[20] Open Source Computer Vision. Available at: $http://opencv.org$

[21] S.Miborrow and F.Nicolls. Locating Facial Features with an Extended Active Shape Model. ECCV, 2008.

[22] S.Yan, C.Liu, S.Z.Li, H.Zhang, H.Shum and Q.Cheng. Face Alignment Uing Texture-constrained Active Shape Model. Image and Vision Computing 21, 2003, pp 69-75.

[23] Interpreting Face Images Using Active Appearance Models. proc. Third International Conference. Automatic Face and Gesture Recognition, 1998, PP 300-305

[24] A.kasinski, A.Florek, A.Schmidt. The PUT face database. Image Processing and Communication,2008 pp 59-64.

[25] K.Messer, J.Matas, J.Kittler, J.Luettin and G.Maitre. XM2VTS: The Extended M2VTS Database. Proceedings 2nd Conference on Audio and Video-base Biometric Personal Verification (AVBPA99) Springer Verlag, New York, 1999.

[26] Y.Huang, Q.Liu and D.Metaxas. A Component Based Deformable Model for Generalized Face Alignment. IEEE International Conference on Computer Vision. 2007.

[27] D.E.King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, 2009, pp. 1755-1758.

[28] V.Kazemi and J.Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. CVPR, 2014

[29] K.Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine 2, 1901, pp. 559-572.

[30] L.Smith. A tutorial on Principal Components Analysis. 2002. available at: $http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf$