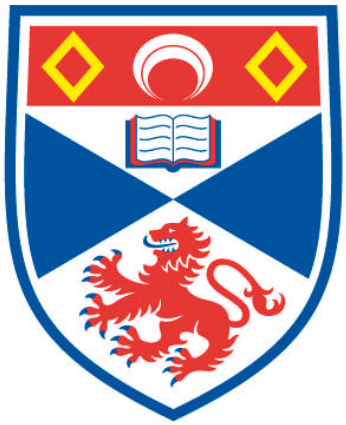
# FACIAL FEATURE DETECTION AND TRACKING WITH A 3D CONSTRAINED LOCAL MODEL

**Meng Yu**

**A Thesis Submitted for the Degree of PhD**
**at the**
**University of St Andrews**

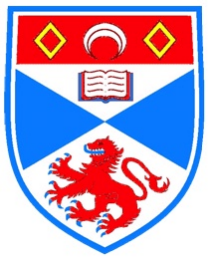


**2010**

# Facial Feature Detection and Tracking with a 3D Constrained Local Model

**Meng Yu**

**supervised by Dr. Bernard Tiddeman**



School of Computer Science

University of St. Andrews

*Submitted in partial fulfilment of the requirements*

*for the degree of Doctor of Philosophy*

*in the field of Computer Science.*

May 2010

# Declaration

I, Meng Yu, hereby certify that this thesis, which is approximately 25,000 words in length, has been written by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in April 2005 and as a candidate for the degree of Doctor of Philosophy in April 2006; the higher study for which this is a record was carried out in the University of St. Andrews between 2005 and 2009.

**Date:**

**Signature of candidate:**

# Certification

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

**Date:**

**Signature of supervisor:**

# Library Declaration

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. We have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the electronic publication of this thesis:

Embargo on both all printed copy and electronic copy for the same fixed period of two years on the following ground: publication would be commercially damaging to the researcher, or to the supervisor, or the University; publication would preclude future publication.

**Date:**

**Signature of candidate:**

**Signature of supervisor:**

# Acknowledgements

First, I would like to thank my supervisor, Dr. Bernard Tiddeman, for his invaluable ideas and guidance throughout my Ph.D. years.

I would also like to express my appreciation to all the people who helped me on my study and research: Alex Bain, David Letham, Paula Whiscombe, Dr. Markus Tauber, Dr. Jingying Chen, Dr. David Hunter, Dr. Weir Mike, Dr. Ishbel Duncan, Dr. Mike Livesey and especially Prof. Steve Linton.

Also, I would like to thank my friends, Ying Zhai, Chunming Tai, Indika Perera, Dulani, Dr. Guogang Xu, Dr. Chun Xiong, Dr. Hai Deng, Dr. Feng Jiao, Dr. Zhuojia Lin, Mrs. Li Meng, Dr. Dengke Xiao and others who have made my time at St Andrews so enjoyable.

Finally, special thanks must also go to my parents, my brother, and the most of all – my dear wife, Dr. Jianli Bao, for their understanding, support and love, without whom I could not have completed this thesis.

# Publication List

[1] B. P. Tiddeman, D. W. Hunter, and M. Yu, "Fibre centred tensor faces," in *British Machine Vision Conference*, vol. 1, pp. 449–458, 2007.

[2] M. Yu and B. P. Tiddeman, "Facial feature detection and tracking with a 3d constrained local model," in *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2010.

[3] M. Yu and B. P. Tiddeman, "Face detection and tracking with 3d PGA CLM," in *International Conferences on Computer Vision Theory and Applications*, 2010.

# Abstract

This thesis establishes a framework for facial feature detection and human face movement tracking. Statistical models of shape and appearance are built to represent the human face structure and interpret target images of human faces. The approach is a patch-based method derived from an earlier proposed method, the constrained local model (CLM) [1] algorithm.

In order to increase the ability to track face movements with large head rotations, a 3D shape model is used in the system. And multiple texture models from different viewpoints are used to model the appearance. During fitting or tracking, the current estimate of pose (shape coordinates) is used to select the appropriate texture model. The algorithm uses the shape model and a texture model to generate a set of region template detectors. A search is then performed in the global pose / shape space using these detectors. Different optimisation frameworks are used in the implementation. The training images are created by rendering expressive 3D face models with different scales, rotations, expressions, brightness, etc.

Experimental results are demonstrated by fitting the model to image sequences with large head rotations to evaluate the performance of the algorithm. To evaluate the stability and selection of factors of the algorithm, more experiments are carried out. The results show that the proposed 3D constrained local model algorithm improves the performance of the original CLM algorithm for videos with large out-of-plane head rotations.

# Table of Contents

Chapter 1

# Introduction

The target of machine vision is image understanding. It is difficult to develop model-based approaches to the interpretation of images with complex and variable structures such as faces. Variability is the key problem at this point. To improve our ability to model such objects, specific models need to be built. In this thesis, a method using a 3D shape model and view-dependent feature templates is presented for locating and tracking human face features. The 3D statistical model is matched to previously unseen 2D video sequences of human faces by applying a shape constrained search method, using an extension of the Constrained Local Model (CLM) algorithm. A set of model parameters learnt from the the training set are used to control modes of shape and texture variation.

The original CLM algorithm [2] works with limited rotations from the front face view. The extension to the algorithm proposed here works not only on the front face view but also on the face with large head rotations. The proposed multi-view CLM consists of a 3D shape model and several 2D texture models from multiple views.

In this implementation, the shape model is first given some suitable initialisation parameters (approximate rigid body alignment, scaling) based

on the initialisation of the feature points locations for the target image (the first frame in the case of tracking). In each subsequent iteration square regions are sampled around each feature point and projected into the allowed appearance model space. The shape and pose parameters are then found that maximise the correlation between the synthesised appearance template patches and patches extracted around the current estimates of the feature points locations in image space. For tracking, these locations can be used as the initialisation for the refinement of the next frame. The proposed algorithm is view based, in that the switching of texture models depends on the current estimate of the face orientation.

## 1.1 Motivation

The problem of face analysis in still images and videos has been extensively studied for years. This intense research activity finds its motivation in the possibility to set up a large range of applications in the medical, psychological and linguistic fields (cognitive studies, expression transfer on an avatar, behaviour / expression analysis, etc. ). Face analysis is a difficult topic since face images vary in identity, pose and expression. The sought-after model should be able to automatically and reliably describe previously unseen faces under any pose and expression.

More potential applications of human face movement capture are the driving force of system development including two major application areas: surveillance and control. The surveillance area covers applications where one or more subjects are being tracked over time and possibly monitored for special actions. The control area relates to applications where the captured motion is used to provide controlling functionality. It could be used as an

interface to games, virtual environments, or animation or to control remotely located objects.

## 1.2 Thesis Outline

In the following sections, a brief overview of the work described here is given along with pointers to the relevant sections of the thesis.

### 1.2.1 Model Building

Cristinacce and Cootes [1; 2] first proposed the Constrained Local Model (CLM) algorithm and described how to build and fit the model. The model building method is similar to building an active appearance model (Cootes *et al.* [3; 4]). Based on these earlier methods, a basic framework is set out to build the proposed multi-view 3D CLM model in *Chapter* 3. In order to allow improved tracking through large head rotations, a 3D shape model is built instead of a 2D one. The shape model is built by finding the main shape variations from the mean using principal component analysis (PCA) on a set of orientation and scale normalised point sets. Multiple PCA appearance models are then built for a number of selected views covering overlapping angles of head orientation. Separate parameters are used for the shape model and texture models to supervise the search process.

To obtain the training data, 3D images of 14 subjects (8 males, 6 females) in 14 posed expressions were captured. Twenty-five facial features around the eyes, eyebrows, nose, lips and the jaw area were manually located on these 3D models. The 3D points are used directly to build the 3D shape model. To build the appearance model, the original 3D models are rendered using one of the 15 viewpoints and a 20x20 block of pixels is extracted around

the projected location of each feature landmark point at each of 3 spatial scales. The extracted patches are used to build an appearance model for each viewpoint. More details can be found in *Chapter* 3.

### 1.2.2 Image Registration

In *Chapter* 4, methods fitting the model to unseen face images are introduced. The Gauss-Newton iterative method [5–9] has been widely used to solve AAM fitting problems [4; 10] and proved to be efficient. More recent work [11; 12] showed that normalised cross correlation (NCC) can be a better metric. Cristinacce and Cootes [1; 2] used the Nelder-Meade method [13] to optimise the NCC which gives good performance. Tiddeman *et al.* [14] used the traditional inverse compositional AAM alignment method to optimise the NCC instead of the standard sum of squared error (SSE).

Using the proposed system, three optimising algorithms (Gauss-Newton iterative method, Powell's Direction Set method and Tiddeman *et al.* 's FastNCC algorithm), are tested for fitting the model to previously unseen face images. These fitting algorithms can be extended for tracking by fitting the model to each frame, initialised from the estimate of the previous frame. An alternative is to use the conditional density propagation (Condensation) algorithm [15] to track using multiple hypotheses for the current parameters. The fitting and tracking algorithms are described in more detail in *Chapter* 4.

### 1.2.3 Results and Discussion

In *Chapter* 5, some experiments are set out in order to evaluate the performance of the proposed algorithm.

First, the testing data sets and methods used for the experiments are described in detail. Then fitting the model to image sequences with large

head rotations is demonstrated. The results show that the multi-view 3D CLM algorithm improves the performance of the original CLM algorithm for videos with large out-of-plane head rotations. More experiments are carried out to evaluate the performance of the system with different settings. The different optimising methods for the fitting are tested for stability and speed. Additional experiments investigate varying the size of the feature patches, the effects of using different multi-resolution settings and methods for estimating which features are hidden to eliminate them from fitting.

The main contribution of this thesis are the improved effectiveness of the CLM algorithm when using a 3D model and multiple texture models, and a thorough investigation into the effects of varying the parameters of the algorithm. A different fitting framework is used in the implementation. And the effects of varying the parameters allows selection of the most suitable values for the algorithm are studied. Further studies are also carried out on other factors such as the patch size, scales, hidden points, background occlusion, etc. It shows that the proper selection of these factors can make the algorithm more stable and efficient.

The proposed algorithm was implemented in Java and tested on a Core Duo 2.10GHz processor under 32bit Microsoft Windows environment. Over 80% of the code comprising four packages, about forty classes, and about fifty thousands lines is freshly written. Part of the work is built on the framework of my colleagues and supervisor. Other parts are obtained from free contributors and text books.

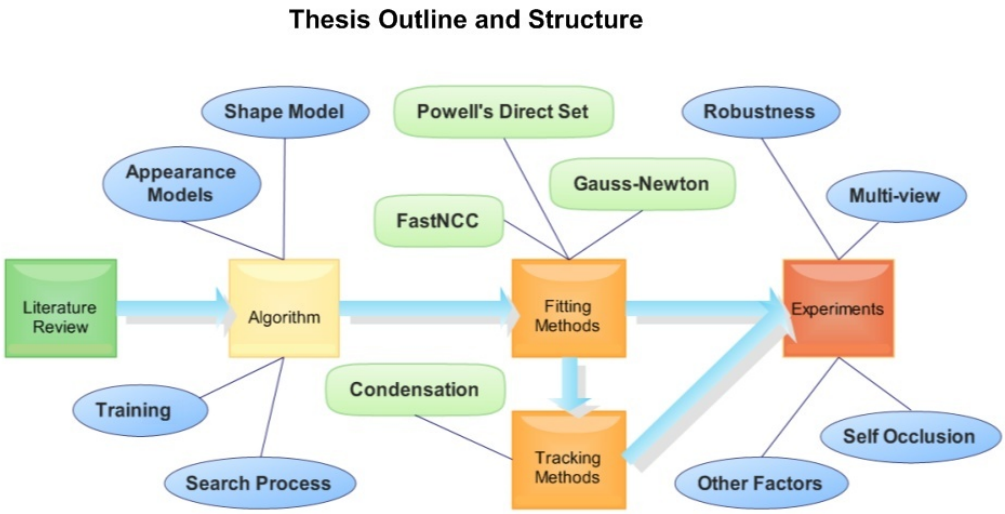



*Figure 1.1: The literature review of the proposed multi-view 3D CLM algorithm.*

## 1.3 Summary

In this chapter, an overview of the work contained in this thesis was given along with its motivational context and major contributions. The next chapter presents a review of face and facial feature detection algorithms relevant to the proposed method. The model building algorithms are described in *Chapter* 3 and fitting methods in *Chapter* 4, followed by experimental results demonstrating the performance of the proposed multi-view CLM method in *Chapter* 5. The final chapter discusses the conclusions that can be drawn from this work and provides pointers to possible future work. Some technical details are included in *Appendix* A. The whole structure and outline of the thesis can be seen in *Figure* 1.1.

Chapter 2

# Literature Review

The problems of facial feature detection and tracking have received a great deal of attention in the literature, here the more immediately relevant works will be covered – Active Shape Models (ASMs) [16], Active Appearance Models (AAMs) [3; 4], Constrained Local Models (CLM) [1; 2], 2D+3D AAMs [17] and 3D Morphable Models [18]. Before AAMs are introduced, some previous relevant methods including the active contour algorithm will be presented. In the final section of the chapter, face tracking methods are briefly reviewed. A diagram illustrating the relationships between the methods covered can be seen in *Figure* 2.1.

## 2.1 Previous Methods

The simplest model of an object is to use a typical example as a 'golden image'. A correlation method can be used to match (or register) the golden image to a new image. If structures in the golden image have been labelled, this match then gives the approximate position of the structures in the new image. For instance, one can determine the approximate locations of many structures in a MR image of a brain by registering a standard image, where

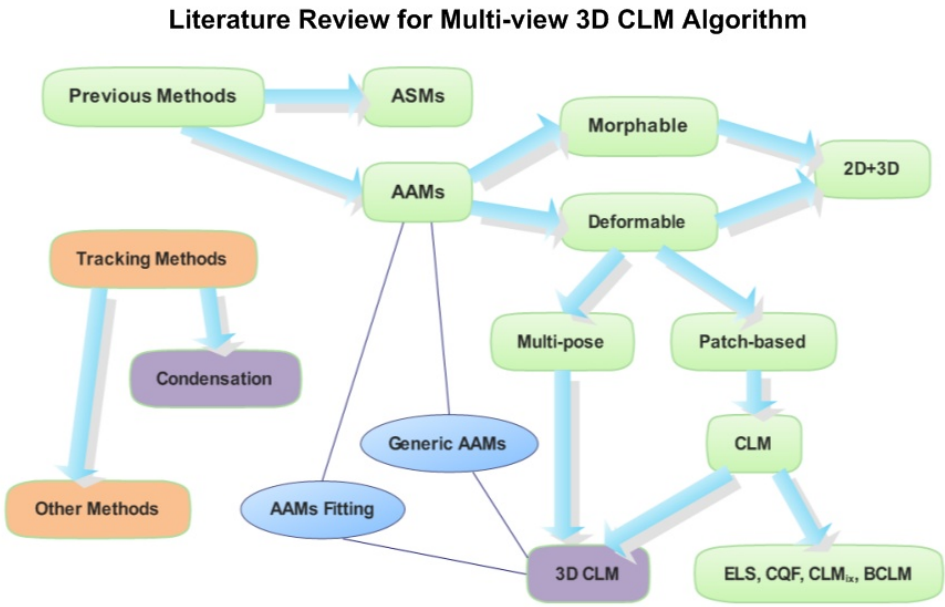



*Figure 2.1: An illustration of relationships between methods relevant to the proposed multi-view 3D CLM algorithm.*

the standard image has been suitably annotated by human experts. However, the variability of both shape and texture of most targets limits the precision of this method.

Staib and Duncan [19] represent the shapes of objects in medical images using Fourier descriptors of closed curves. The choice of coefficients affects the curve complexity. Placing limits on each coefficient constrains the shape somewhat but not in a systematic way. It can be shown that such Fourier models can be made directly equivalent to the statistical models described below [20–22], but are not as general. For instance, they cannot easily represent open boundaries.

Kass *et al.* [20] introduced Active Contour Models (or 'snakes') which are energy minimising curves. In the original formulation, the energy has an internal term which aims to impose smoothness on the curve, and an external term which encourages movement toward image features. They are particularly useful for locating the outline of general amorphous objects. However, since no model (other than smoothness) is imposed, they are not optimal for locating objects which have known shape variability.

Turk and Pentland [21] use principal component analysis to describe the intensity patterns in face images in terms of a set of basis functions, or 'eigenfaces'. Though valid modes of variation are learnt from a training set, and are likely to be more appropriate than a 'physical' model, the representation is not robust to shape changes, and does not deal well with variability in pose and expression. Eigenfaces can, however, be matched to images easily using correlation based methods.

Covell [22] demonstrated that the parameters of an eigen-feature model can be used to drive shape model points to the correct place. The AAM [4] described later is an extension of this idea. Black and Yacoob [23] use local,

hand-crafted models of image flow to track facial features, but do not attempt to model the whole face. The AAM can be thought of as a generalization of this, in which the image difference patterns corresponding to changes in each model parameter are learnt and used to modify a model estimate.

### 2.1.1 Active Shape Models

Cootes *et al.* [24; 25] introduced the Point Distribution Model (PDM) by generating a statistical model of shape and shape variation from a set of example shapes. Each model describes one way in which the shapes in the training set tend to vary from the mean. By analysing sets of real examples a compact description of shape variation can be formed. Cootes *et al.* [26] presented a method combining PDM [25] and finite element methods (FEM) [27–29], which gives better performance.

Active Shape Model (ASM) [16; 30–32] are statistical model of the shape of objects, which iteratively deform to fit to an example of the object in a new image. The shapes are constrained by the PDM [25] Statistical Shape Model to vary only in ways seen in a training set of labelled examples. ASM is similar to Active Contour Models (Snakes) [20], but differs in the applied global shape constraints. ASM uses Principal Component Analysis (PCA) [33] to learn the main axes of variation from a training set of labelled examples. Fitting the shape model to a new image involves local searches for matching features alternated with projection of the shape estimate back into the allowed model space. Results have shown that ASM can be used successfully in image search.

Romdhani *et al.* [34] proposed a multi-view nonlinear active shape model that utilises 2D view-dependent contextual constraint without explicit reference to 3D structures. The model can cope with both nonlinear shape and

grey-level variations around the landmarks by capturing all possible 2D shape variations in the training set and performing a nonlinear transformation of the model during matching. A Kernel PCA [35] based on Support Vector Machines [33] is applied for the transformation.

Hamarneh *et al.* [36] proposed a new method for locating spatio-temporal shapes (ST-shapes) in image sequences. They extend Active Shape Models [16] to include knowledge of temporal shape variations and present a ST-shape modelling and segmentation technique. The method is well suited to model and segment objects with specific motion patterns, as in cardiograph, optical signature motion recognition, and lip-reading for Human Computer Interaction (HCI). The method succeeded in segmenting synthetic spatio-temporal shapes in noisy image sequences.

ASMs have been used successfully in many application areas, including face recognition [37; 38], industrial inspection [16] and medical image interpretation [26]. However, ASMs only use data around the model points, and do not take advantage of all the grey-level information available across an object. The Active Appearance Models (AAMs) manipulates a full model of appearance, which represents both shape variation and the texture of the region covered by the model.

## 2.2 Active Appearance Models

Active Appearance Models (AAMs) [4], first proposed in [3], and the closely related concepts of Active Blobs [39] and Morphable Models [18; 40; 41], are examples of statistical models that are used to characterize the shape and the appearance of the underlying object by a set of model parameters. They are non-linear, generative, and parametric models of a certain visual phe-

nomenon. AAMs may be useful for tracking faces in video [42–44], tracking objects [45–48], modelling and recognising objects [37; 41; 46; 49; 50], and medical image processing [51–53].

Active Appearance Models(AAMs) [3; 4] use the same PCA based shape model as ASMs together with a PCA based model of appearance (i.e. shape normalised texture). Typically, an AAM is first fit to an image of a face; i.e. the model parameters are found to maximise the error metric between the model instance and the input image. The model parameters are then used in the application by passing them to a classifier. Many different classification tasks are possible like face recognition, pose estimation, and expression recognition. A comparison between ASM and AAM can be seen in Cootes *et al.* 's work [51; 54].

Although they are perhaps the most well-known example, Active Appearance Models are just one instance in a large class of closely related linear shape and appearance models (and their associated fitting algorithms). This class contains Active Appearance Models (AAMs) [3; 4; 37], Shape AAMs [10], Active Blobs [39], Morphable Models [18; 40; 41], and Direct Appearance Models [55].

Sclaroff *et al.* [39; 45] demonstrated an Active Blobs method for tracking. The approach is broadly similar in that they use image differences to drive tracking, learning the relationship between the image error and parameter onset in an online processing stage. The main difference is that Active Blobs are derived from a single example, whereas Active Appearance Models use a training set of examples. The former use a single example as the original model template, allowing deformations consistent with low energy mesh deformations (derived using a Finite Element method). A simply polynomial model is used to allow changes in intensity across the object. AAMs learn

what are valid shape and intensity variations from their training set.

Motivated by the AAM algorithm, Hou *et al.* [55] have developed Direct Appearance Models (DAMs) where they use the texture to directly predict the shape during the iterations of the parameter updates. This approach no longer combines the shape and texture parameters into appearance parameters like AAM does. Li *et al.* [56] have later extended this approach for multi-view face alignment by training multiple models for different poses of the human face. In relation to his work, Yan *et al.* [57] have developed texture-constrained ASMs (TC-ASMs), where the shape updates predicted by an ASM is combined with the shape constraint provided by a global texture model like the one in AAM. They show that such an approach performs better than ASM or AAM alone.

One of the main weaknesses of the AAMs is that the reliability generally degrades when the amount of variability increases in the training data. Sources of variability include person identity, expression, pose and shading. Building a rich training database including a large amount of lighting variations [58] is one option. However, it is not easy collecting sufficient training data and the fitting performance will be reduced because of the high model complexity [59]. Pizarro *et al.* [60] proposed a light-invariant AAM fitting algorithm based on the light-invariant transformation [61]. Instead of classically matching the model appearance to the input image in the color space, they project the images into a light-invariant space where the effect of shading is cancelled. The method makes the system useful to color image databases because it does not need the pre-calibrated cameras. According to their experiments, the method outperforms the regular AAM approaches [4].

## 2.3 Optimising AAMs

Fitting an AAM to an image is a non-linear optimisation problem. The usual approach [3; 4; 37; 51] is to iteratively solve for incremental additive updates to the parameters (the shape and appearance coefficients). Given the current estimates of the shape parameters, it is possible to warp the input image backward onto the model coordinate frame and then compute an error image between the current model instance and the image that the AAM is being fitted to. In most previous algorithms, it is simply assumed that there is a constant linear relationship between this error image and the additive incremental updates to the parameters. The constant coefficients in this linear relationship can then be found either by linear regression [37] or by other numerical methods [51]. In Cootes *et al.* 's work in [62], extra constraints are applied to the statistical AAM matching algorithm. The framework allows user interaction to guide the search effectively.

Following the forwards additive algorithm [5], the inverse additive algorithm [6], and the forwards compositional algorithm [7], Baker and Matthews [8] introduced an efficient AAM fitting algorithm for descent image alignment by investigating the inverse compositional algorithm and its extension. They consider an appearance model where the shape and texture coefficients are not combined into appearance coefficients, and show that their approach provides increased efficiency during the matching procedure. Their formulation for the matching algorithm requires the warp functions to satisfy certain properties, which are not satisfied by the warps used in AAM; therefore, they use first-order approximations of the inversion and composition operators.

The original implementation [8; 63] learns a linear model relating the error image (between the model and the image) and the required parameter

updates at each time step. Matthews and Baker [64; 65] derived more mathematically elegant methods in which the updates are always calculated in the average shape and then concatenated with the current guess. This inverse compositional method allows the pre-computation of the gradient images and inverse Hessian matrix for greater efficiency. Later work demonstrated that the inverse compositional algorithm is only really suitable for personal-specific fitting and tracking, and that simultaneous estimation of the shape and appearance parameters was required for robust face fitting [66].

Romdhani *et al.* [67] extends the inverse compositional image alignment algorithm to fit 3D Morphable Models using a mathematical notation which facilitates the formulation of the fitting problem. A mapping from the parameter space to the image frame is built as the generative shape projection. Then two tools are used to solve the inverse fitting problem: the inverse shape projection and an algorithm which separates the model parameters from a set of vertices projection. The formulation avoids a simplification being as efficient and leading to improve fitting precision. Furthermore, the algorithm is robust without sacrificing its efficiency and accuracy.

Batur *et al.* [68] proposed an adaptive AAM where they abandon the fixed gradient matrix approach of the basic AAM, and replace it with a linearly adaptive matrix that is updated according to the composition of the target texture [69]. The adaptation increases the efficient of the fitting process. They showed that the adaptive AAM significantly outperforms the basic AAM.

Papandreous *et al.* pointed out that existing fitting algorithms perform poorly when used in conjunction with models exhibiting significant appear-

ance variation [70]. To overcome the problem, they proposed a fitting algorithm adaptation, by (i) Fitting matrix adjustment and (ii) AAM mean template updates. They also used prior information to incorporate and constrain the AAM fitting process. Both techniques substantially improve AAM fitting performance of models exhibiting significant appearance variation, and give better results while tracking human faces in video, efficiently and robustly.

## 2.4 Alternative Error Metrics

The sum of squared errors provides a simple and efficient error metric as it leads to a minimisation problem with a linear solution. This simple error metric will match the intensity but does not attempt to directly relate important information between the template model and the target image, such as image edges. Other error metrics might provide improved matching at the expense of a non-linear solution. These include the normalised cross correlation and mutual information maximisation.

Cristinacce and Cootes [1; 2] proposed a framework using normalised cross correlation (NCC) as a metric and Nelder-Meade method [13] as the optimising method. This algorithm works by using N+1 samples in the N dimensional parameter space. Each iteration the worst sample is discarded and a new sample is added based on a set of simple heuristics.

Tiddeman *et al.* [14] extended the inverse compositional alignment to use NCC and showed improved results when fitting AAMs to previously unseen face images. It is referred as correlated active appearance models (CAAM) in their paper.

Paterson and Fitzgibbon's experimented with a number of metrics for

tracking using a 3D head model [11] including normalised cross correlation and maximisation of mutual information (MMI), with MMI giving the best performance.

The AAM fitting methods described above are mostly used to solve 2D fitting problems and more efficient used for personal-specific models. However, it has been developed into generic-specific models fitting problems. Based on the technique, more fitting methods are introduced towards that direction including CLM fitting algorithm [1], CAAM fitting algorithm [14] and the proposed algorithm. The AAM fitting methods are also extended to solve 3D problems. They will be reviewed later and used in the proposed algorithm.

## 2.5 Generic AAMs

Although Active Appearance Models (AAMs) have been widely used, the performance of an AAM built to model the variation in appearance of a single person across pose, illumination and expression are substantially better than the performance of an AAM built to model the variation in appearance of many faces, including unseen faces not in the training set. As Gross *et al.* [66] pointed out, it is important to distinguish between the two situations.

The person-specific context is, where the individual face, lighting and range of movement have been explicitly learnt by the model. The fitting accuracy is usually very good in this context, and reliable for post processing systems. Lucey *et al.* [71] used a person-specific AAM to retrieve the face shape and successfully classify facial deformations into Action Units.

On the other hand, the person-generic context is where the fitted face is not in the training set. As first shown by Gross *et al.* in [66], the fitting

process is much harder than in the person-specific context. Peyras *et al.* [72] demonstrated with carefully chosen experiments that fitting an unseen face with an AAM is much less accurate than fitting a face that belongs to the set of images used to train the model. They pointed out that in the generic context, the appearance counterpart of the model cannot fully explain the appearance of the face in the input image. As an unfortunate consequence, the minimum error of the cost function corresponds to a biased position of the model. Even when initialised in the best possible position (the ground-truth shape), the AAM drifts away.

Gross *et al.* [66] carried out an empirical evaluation comparing the Generic AAM to the Person Specific AAM. They showed that (i) Building a generic shape model is far easier than building a generic appearance model, (ii) The shape component is the main cause of the reduced fitting robustness of Generic AAMs. They also managed to improve the performance with two proposed refinements to Generic AAMs: (i) A refitting procedure to improve the quality of the ground-truth data used to build the AAM, (ii) The Simultaneous Inverse Compositional fitting algorithm [9; 64].

AAMs appear to provide an interesting basis to fit unseen faces. One could think that adding more training data would increase the ability of the model to generalize to unseen faces. Indeed, this ability decreases with the amount of training data.

Peyras *et al.* [73] pointed out that the higher complexity of the AAM makes its fitting unreliable because this induces numerous local minima in the cost function. As a consequence, the solution for reliable and accurate fitting must combine these two contradictory conditions: (i) The complexity of an AAM must be kept as low as possible to preserve a large convergence basin and can find the global cost minimum, (ii) The range of face images

that the AAM can explain must be large, so that the global cost minimum matches the sought after solution.

To bypass the contradiction, they proposed a method by building a pool of AAMs, each one being specialized on a particular pose and expression. These AAMs are fitted to the picture in a multiple fitting fashion and the AAM shows the smallest residual error is retained. The experiments showed that the method is very accurate on unseen faces, which could be used to track the face reliably in real-time.

## 2.6 Machine Learning Algorithms

Rather than using simple linear error minimisation approach, attempts have been made to apply the wide range of statistical and machine learning techniques to face feature detection and tracking.

Donner *et al.* [12] used canonical correlation analysis to improve the learning of the optimal model parameter update from the current error image.

Several other techniques have been proposed for the fitting. RANSAC algorithm (RANdom SAmple Consensus) is first proposed by Fischler *et al.* [74]. It is an iterative method to estimate parameters of a mathematical model from a set of observed data, which contains outliers. It is a non-deterministic algorithm in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are allowed.

The Expectation Maximisation Algorithm was proposed by Dempster *et al.* [75]. It's a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values. It's been widely used ever

since, such as mixture estimation [76–78] and AAM fitting [79; 80]. The EM algorithm has also been used in various motion estimation frameworks [81] and variants of it have been used in multi-frame super-resolution restoration methods which combine motion estimation along the lines of [82].

Boosting is one of the most important classification methods, and has been widely used [83]. The boosting family includes AdaBoost [84; 85], GentleBoost [86], RankBoost [87] etc. Liu [88] proposed a Boosted Appearance Model (BAM) framework that greatly improves the robustness, accuracy and efficiency of alignment on generic faces. Wu *et al.* [89] pointed out that there is no guarantee that moving along its gradient will always improve the alignment. They proposed a Boosted Ranking Model (BRM) to address the limitation of BAM and showed improvements over BAM.

Cristinacce *et al.* [90] used boosted regression approach within the Active Shape Model framework [30]. Instead of using local eigen patches [21], they use non-linear boosted features trained using GentleBoost [86] to model each feature. This boosted regression approach is shown to be much more efficient than the CLM algorithm based on the experiments on BIOID [91] and XM2VTS [92] data sets.

## 2.7 Constrained Local Models

Numerous algorithms have been suggested for facial feature detection. Methods related to this work include Cristinacce *et al.* 's pairwise reinforcement of feature responses [93], Cristinacce and Cootes' Constrained Local Model (CLM) algorithm [1; 2], and Wang *et al.* 's Exhaustive Local Search (ELS) algorithm and generic Convex Quadratic Fitting (CQF) approaches [94; 95]. ELS algorithm and CQF algorithm are extended from the original CLM

algorithm. The CQF method is extended by Paquet *et al.* by applying a Bayesian framework [96]. A "shape-constrained" Markov random field is used by Liang *et al.* to model a face [97]. Another Bayesian generative model is Gu and Kanade's treatment of multiple candidate feature alignment positions as unobserved latent variables, through which the shape-and-pose posterior mode can be found with an expectation maximization algorithm [98]. Liu aligns images by iteratively maximizing the score of a classifier–a boosted appearance model–that distinguishes between correct and incorrect alignments, and updating a low-rank shape parameter [88]; this approach is extended with a boosted ranking model by Wu *et al.* [89].

CLMs are ideally suited to facial feature alignment and general non-rigid object registration, as they merge shape and texture information by coupling (or constraining) an ensemble of local patch or feature detectors at a global shape level [1; 2; 95; 96]. This has proven to outperform AAMs [4] as it is more robust to occlusion and changes in appearance and no texture warps are required.

A major advantage of CLMs over conventional methods for non-rigid registration such as AAMs [4] lies in their ability to: (i) Be discriminative and generalize well to unseen appearance variation; (ii) Offer greater invariance to global illumination variation and occlusion; (iii) Model the non-rigid object as an ensemble of low dimensional independent patch experts; and (iv) Not employ complicated piece-wise affine texture warp operations that might introduce unwanted noise. [95]

The method of building the CLM model is similar to the AAM [4] approach, but instead of modelling the whole object region, they model a set of local feature templates as appearance patches. The feature templates are then matched to the image using an efficient shape constrained search

of the template response surfaces. A CLM can register a non-rigid object through the application of an ensemble of patch / region experts to local search regions within the source image. Given an appropriate non-rigid shape prior to the object, the response surfaces from these local regions are then employed within a joint optimization process to estimate the global non-rigid shape of the object.

Cristinacce and Cootes' method [2] relies on computationally expensive generic optimisers such as the Nelder-Mead simplex [13] method. The canonical Lucas-Kanade algorithm [5; 65] is effective when dealing with a similar optimisation problem. To take advantage of this similarity, Wang *et al.* [95] proposed a couple of extensions to the original CLM [2] and the Lucas-Kanade algorithm to optimise the global warp update in an efficient manner by enforcing convexity at each local patch response surface.

Wang *et al.* [94] proposed another patch-based approach to align unseen images based on an Exhaustive Local Search (ELS), which uses a single appearance template of the target subject differing from the previous methods [2; 99]. The approach attempts to find a local maximum in each patch response surface and then simply constrain these local maxima to be consistent with the global shape prior.

Their method is not limited to intensity values or gradients and therefore, offers a natural framework to integrate multiple local features, such as filter responses, to improve the robustness to large initialisation errors, changing illumination conditions and non-rigid deformations. The experimental results demonstrated that the approach can improve the accuracy and robustness of feature alignment and image registration.

Wang *et al.* [95] proposed a new discriminative approach for non-rigid object registration by making a couple of extensions to the canonical

constrained local models (CLM) [2] framework. Through generic Convex Quadratic Fitting (CQF) and Robust Convex Quadratic Fitting (RCQF), they turn the global CLM warp update into a convex problem. It can jointly optimise the local responses in an efficient manner when estimating the global non-rigid shape of an object by attempting to model each local response using a convex quadratic function. By enforcing this convexity it was possible, through an iterative method, to solve jointly for the global non-rigid shape of the object. Furthermore, their extension of the Lucas-Kanade algorithm leads to an efficient and robust implementation of the CLM method.

By finding convex approximations to the local patch response surfaces of feature alignment classifiers, they circumvented the need for computationally expensive optimizers (except maybe for fitting the convex surfaces). They pointed out that a specific form of the Lucas-Kanade [5] gradient descent image alignment algorithm can be viewed as a generic CQF. It was also shown to be superior to Exhaustive Local Search (ELS) [94], which constrains local patch response maxima to be consistent with the shape prior.

Based on the experimental results using the CMU MultiPIE face database [100] and the UNBC-McMaster archive [101], the RCQF method [95] proposed by Wang *et al.* has better alignment performance than other evaluated CLMs [2; 94] and leading existing holistic methods for alignment / tracking.

Saragih *et al.* [79] proposed an approach ($\text{CLM}_{ix}$) for combining local experts for deformable model fitting based on the patch-based shape models. They combined responses for each landmark from an ensemble of simple local experts in a principled way. To achieve this, the likelihood of each landmark location is approximated using a mixture model. Each component of the mixture is the result of an exhaustive search with a local expert. An

Expectation Maximisation (EM) algorithm is adapted to find the parameters of the shape model. Comparing the ASM algorithm [16] and the CQF algorithm [95], the fitting fidelity is improved with $\text{CLM}_{ix}$. In addition, the computational complexity of the approach was shown to scale only linearly with the number of mixture components used.

Paquet *et al.* [96] proposed a Bayesian Constrained Local Model (BCLM) to improve the performance of the generic CQF method of Wang *et al.* [95]. Similar to LFW [102], the Bayesian framework is built on the assumption that faces are detectable by a Viola-Jones face detection algorithm [103] – an assumption that can be explicitly incorporated into a prior alignment distribution.

They generalised the generic CQF to a Bayesian version, which allows multiple sets of patch alignment classifiers to be used. For even better alignment, the choice of degrees of freedom allows for more accurate fits needs to be traded against the quality and speed of the patch classifiers. The results showed that better fits can be achieved by applying the BCLM over generic CQF and the BCLM is more useful when faces appear in an unconstrained environment.

## 2.8 Multi-Pose Models

Active appearance models (AAMs) [3; 4] were originally formulated as a 2D method and most of the algorithms for AAM fitting have been single-view [10]. Automatically locating detailed facial landmarks across different subjects and viewpoints, i.e. 3D alignment of a face, is a challenging problem. Previous approaches can be divided into three categories: view (2D) based, 3D based and combined 2D+3D based. View based methods

[73; 80; 104; 105], train a set of 2D models, each of which is designed to cope with shape or texture variation within a small range of viewpoints. For some applications switching between 2D views can cause notable artefacts (e.g. in face reanimation [106]). 3D based methods [18; 40; 41; 107–109], in contrast, deal with all views by a single 3D model. The early work of Blanz *et al.* [18] on 3D Morphable models interpret a face by minimising intensity differences between the synthesised image and the given image. 3D Morphable model fitting is an expensive search problem in a high dimensional space with many local minima, which often fails to converge on real data. 2D+3D based methods [17; 110–112] used AAMs and estimated 3D shape models to track faces in videos, but these algorithms are generally most suitable in the person specific context. The proposed multi-view CLM algorithm is a 3D extension of the CLM algorithm [1; 2] which could be more useful for fitting to unseen face images and tracking.

Zhang *et al.* [109] proposed an approach that deforms a 3D mesh model so that the 3D corner points reconstructed from a stereo pair lie on the surface of the model. Dimitrijevic *et al.* [113] proposed the use of a 3D morphable model similar to that of Blanz's, but discarded the texture component from the model in order to reduce the sensitivity to illumination. Both [109] and [113] minimise the shape difference instead of intensity difference, but rely on stereo correspondence.

Zhou *et al.* [80] proposed a multi-modal Bayesian framework for multi-view face alignment. A mixture model is used to describe the shape distribution and point visibility. Within the framework, an EM algorithm for shape regularisation is developed to estimate the parameters for the unknown feature points.

Faggian *et al.* [105] pointed out that it is possible to model real-world

variation of identity using AAMs built from synthetic data because the synthetic 3DMM [18] data provides a number of advantages: (i) The data can be reproduced to match a specific environment, (ii) Many different identity variations can be generated to build more generic AAMs, (iii) An AAM+MM fitting could provide correspondence from the 2D to the 3DMM.

Most of the previous algorithms for AAM or CLM fitting and construction have been single-view [1–4; 9; 10; 37; 94; 95]. (assuming near frontal-parallel views) and tends to break down when presented with large rotations or profile views. Cootes *et al.* [104] studied simultaneous multi-view algorithms using statistical models of shape and appearance to represent the variations in appearance from a particular viewpoint. They used 5 models (3 distinct models), roughly centred on viewpoints at – $90°, -45°, 0°, 45°, 90°$ (where $0°$ corresponds to the frontal-parallel view). They demonstrated that the models can be used to track faces through wide angle changes, and that they can be used to predict appearance from viewpoints given a single image of a person.

Several algorithms have been proposed to build deformable 3D face models and to fit them efficiently ([17; 46; 67; 110; 112; 114–117]) in addition to Cootes *et al.* 's methods [104; 118]. Deformable 3D face models have a wide variety of applications. Not only can they be used for tasks like pose estimation, which just require the estimation of the 3D rigid motion, but also for tasks such as expression recognition and lip-reading, which require, explicit or implicit, estimation of the 3D non-rigid motion.

The main technical challenge is relating the AAM shape parameters in one view with the corresponding parameters in the other views. This relationship is complex for a 2D shape model but is straightforward for a 3D shape model. A 2D+3D AAMs algorithm [17] using both a 2D shape model and a 3D shape

model is presented by Xiao *et al.* Besides the requirement of having a 3D shape model, the main advantage of the algorithm is that a 2D+3D AAMs can be fitted very efficiently in real-time. Xiao *et al.* [17] tried to combine the best features of AAMs [4] and 3DMMs [18]: real-time fitting (AAMs) and a parameterisation consisting of a camera matrix and 3D shape (3DMM). They used an AAM and computed 3D shape models to build the Combined 2D+3D AAM. The number of parameters is increased in constraining an AAM with the corresponding 3D shape modes. Because the AAM parameters and the 3D shape parameters are tightly coupled, the algorithm reduces the flexibility of the model and can lead to faster convergence.

Based on the single-view 2D+3D AAM algorithm presented by Xiao *et al.* 's [17] and Coupled-view AAM (CVAAM) presented by Cootes *et al.* 's [118], Hu *et al.* [110] proposed a multi-view 2D+3D AAM algorithm to fit a single 2D+3D AAM to N concurrent images captured simultaneously by N uncalibrated cameras. The algorithm computes (i) 2D shape parameters for each image, (ii) A single set of global 3D shape parameters, (iii) The weak-perspective camera matrix for each view (3D pose), (iv) Appearance parameters for each image. The algorithm enforces the constraints that all of these quantities are physically consistent in the 3D scene. The results showed the multi-view 2D+3D AAM algorithm to be both more robust and converge more quickly than the single-view 2D+3D AAM algorithm, which is itself more robust than the 2D AAM algorithm [9].

Koterba *et al.* [111] proposed an extension of the multi-view AAM fitting algorithm presented by Hu *et al.* [110]. They used the human face as a calibration grid to calibrate orientations of a set of cameras. They demonstrated that the resulting calibration can be used to improve the performance of multi-view face model fitting.

Gu *et al.* [99] proposed a deformable model consisting of a sparse set of 3D points and view-based patch appearance. With the patch information, it is easier to compensate for the illumination locally; and the variance of texture within a patch is considerably smaller than that of the whole face. However, patch information alone is not enough to localize a facial point. A compact 3D shape is constructed, and is applied to constrain the 2D facial points in different views. Their experiments demonstrate that the approach can effectively handle unseen faces with a variety of pose and illumination variations.

A multi-view algorithm to both fit and build 3D AAMs is presented by Ramnath *et al.* [112]. Fitting an AAM to an image consists of minimizing the error between the input image and the closest model instance. Face models are usually fitted to a single image of a face [2; 4; 9; 66; 94–96]. To obtain better application performance [119], they integrate the information from multiple views by fitting a single AAM to multiple images, captured by cameras with arbitrary locations, rotations, and response functions.

Their proposed multi-view 2D+3D AAM algorithm enforces the constraints that all of these quantities are physically consistent in the 3D scene. Their method is both slightly more robust and converge more quickly than the single-view 2D+3D AAM algorithm [17], which is itself more robust than the single-view 2D AAM algorithm [9]. The results shown that the benefit of using calibrated multi-view over uncalibrated multi-view is in most cases perhaps even bigger than the benefit of using uncalibrated multi-view over single-view. The proposed algorithm shows that using calibrated multi-view motion-stereo can eliminate this ambiguity and yield face models with higher 3D fidelity. So automatic calibration is important in many applications and dramatically improves fitting performance.

Matthews *et al.* [120; 121] carried out several experiments comparing 2D (AAM) and 3D (3DMM) face models along three axes: (i) Representational power, (ii) Construction, (iii) Real-time fitting. They pointed out that 3D models are overall preferable to 2D models. The 3D parameterisation is more compact (up to 6 times fewer parameters), more natural (i.e. head poses and non-rigid shape deformation are separated), 3D model fitting is more robust and requires fewer iterations to converge, and 3D occlusion reasoning is more powerful. They presented two of the frequently cited negatives of 3D models, slow fitting and construction requiring range data [121]. One final benefit of 3D models is that multi-camera model fitting is far easier [111].

## 2.9 Tracking

Detecting and tracking faces in video sequences is a challenging task because faces are non-rigid and their images have a high degree of variability in shape, texture, pose, and imaging conditions. Detecting faces in images has received much attention. A comprehensive survey on face detection methods can be found in [122]. A huge research effort has been devoted to detecting and tracking of heads and facial features in 2D and 3D [43; 123–128].

The formulation of head and facial expression tracking using active appearance models was introduced in Ahlberg's work [44]. An AAM-based tracker is drifting insensitive and flexible. However, since AAMs are essentially representing 2D appearances, the main drawback of this previous work is that the estimated out-of-plane motion is not very accurate due to the nature of the criterion used. Similar results were reported in [129]. Unlike the face interpretation used in [4] which deals with the 2D geometry of faces, Dornaika *et al.* 's [127] proposed a method that dealt with the real-

time estimation of 3D geometry of faces. Their work addresses the real-time 3D tracking of pose and animation of the human face in monocular image sequences using active appearance model search.

If pose estimation is based on feature-based correspondence, those features can be tracked over time [130–135]. On the other hand, if appearance-based templates are used, matching all model views at all possible poses in each frame can be prohibitively expensive. Furthermore, face views at different pose can be confused with one another, leading to false maxi-ma. Therefore, temporal continuity assumptions are used.

Kalman filters [136; 137], Hidden Markov Model (HMM) [138; 139] and Condensation [15; 140–142] provide useful techniques for object tracking and have been used quite widely. Their recursive nature makes them well suited to real-time applications and their ability to predict provides further computational efficiency by reducing image search.

Kalman filters are based on a single Gaussian state density which is unimodal, they cannot represent simultaneous, multiple hypotheses. Condensation, on the other hand, does not make such a strong parametric assumption about the form of the state density, $p(s_t)$, and can therefore, track multiple, ambiguous targets simultaneously over time [140]. The condensation algorithm allows quite general representations of probability. Experimental results show that this increased generality does indeed lead to a marked improvement in tracking performance. In addition to permitting high-quality tracking in clutter, the simplicity of the Condensation algorithm also allows the use of non-linear motion models more complex than those commonly used in Kalman filters. The condensation algorithm is based on sampling the posterior distribution estimated in the previous frame and propagating these samples to form the prior for the current frame. The

method has shown to be a powerful alternative to the Kalman filter [143; 144].

## 2.10 Summary

In this chapter, the relevant works on Active Shape Models (ASMs) [16], Active Appearance Models (AAMs) [4] a 2D+3D AAMs [17], 3DMMs [18; 40; 67; 107] and patch-based methods such as Constrained Local Model (CLM) [1] etc. have been reviewed. First, how the ASMs and AAMs were derived from previous methods was described. Then the AAM literature including generic AAMs, CLM and Multi-Pose models were introduced. Finally, a brief review of face tracking techniques was presented.

The proposed algorithm is derived from the original CLM algorithm [1], which can be further extended using other CLM based methods [94–96]. The CLM methods can be used for tracking human face movements naturally, and they perform quite well. Currently, the original patch fitting technique [1] is used and it can be flexibly replaced or combined with more recently proposed methods [94–96]. View-based methods have been used to model 3D human faces effectively. By applying the technique to the CLM methods, the system could not only retain the ability of tracking as a CLM, but also gain the ability to track human faces with large head movements in video. The 3D extension of the original CLM algorithm is then proposed, which will be introduced from the next chapter.

Chapter 3

# Algorithm

In this chapter, the basic framework of the proposed multi-view 3D CLM algorithm for facial feature detection and tracking is set out. The algorithm is based on Cristinacce and Cootes' Constrained Local Model (CLM) [1; 2] extended to handle large out-of-plane head rotation. A 3D shape model is combined with multiple texture models built from generated sets of region texture patches from selected views. When fitting the model to a new image, first the local appearance is grabbed from around the current estimated location of the feature points. Template patches are then generated from the texture model corresponding to the current estimate of the head rotation. A search is then performed in the global pose / shape space to find the best match between the model patches and the local image appearance.

## 3.1 Overview

The method of building the CLM is similar to the AAM [3; 4] approach, but instead of modelling the whole object region, a set of local feature templates is modelled. The model is fitted to an unseen image in an iterative manner by generating templates using the model and the current parameter estimates,

correlating the templates with the target image to generate response images and optimising the shape parameters to maximise the sum of responses. [1; 2]

The original CLM can be fitted to face images from a certain view (frontal view without significant head rotation). In order to make the model suitable for tracking through large head rotation, a 3D shape model and several texture models trained from different views are used. Another adjustment has been made to the original CLM algorithm which can be seen in *Figure* 3.1. Instead of using a combined model, the shape model and texture models are applied separately. To build a combined model, a further PCA would need to be applied for each view, which means there are different models for each view. Each model requires separate combined parameters $c_0, c_1, \cdots, c_n$ to give shape coordinates, $s$. The parameters are not related to each other because the models are built separately. It's still possible but requires additional steps to decompose and rebuild the model when switching views. So for convenience, in this work the focus is on using only one shape model to simplify make the switching across views.

## 3.2 Shape Model

The shape model is obtained by first manually specifying the locations of a number of landmarks on a set of 3D face models. To eliminate variation due to extrinsic factors (pose and scale) the landmark points are aligned to the average of the aligned points. To find the main axes of shape variation the shape vectors are treated as points in a high dimensional space. Jacobi's method is applied to the covariance matrix of these points to find the eigenvectors and eigenvalue, which represent the principal components and their variances.

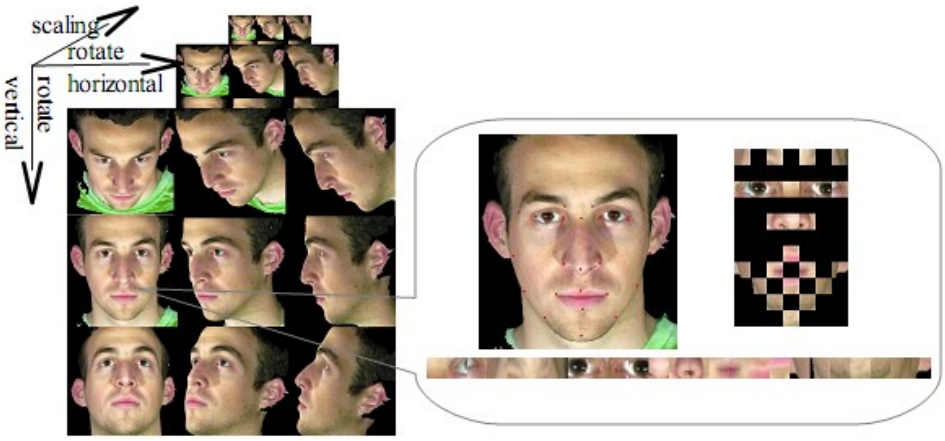



*Figure 3.1: The Multi-view CLM consists of a shape model and several texture models from different views. Fifteen rotations and three scales are used to cover all the likely circumstances in the application. (There are only 9 rotations in the figure because the views from the right side are mirroring copies of the ones from the left side.)*

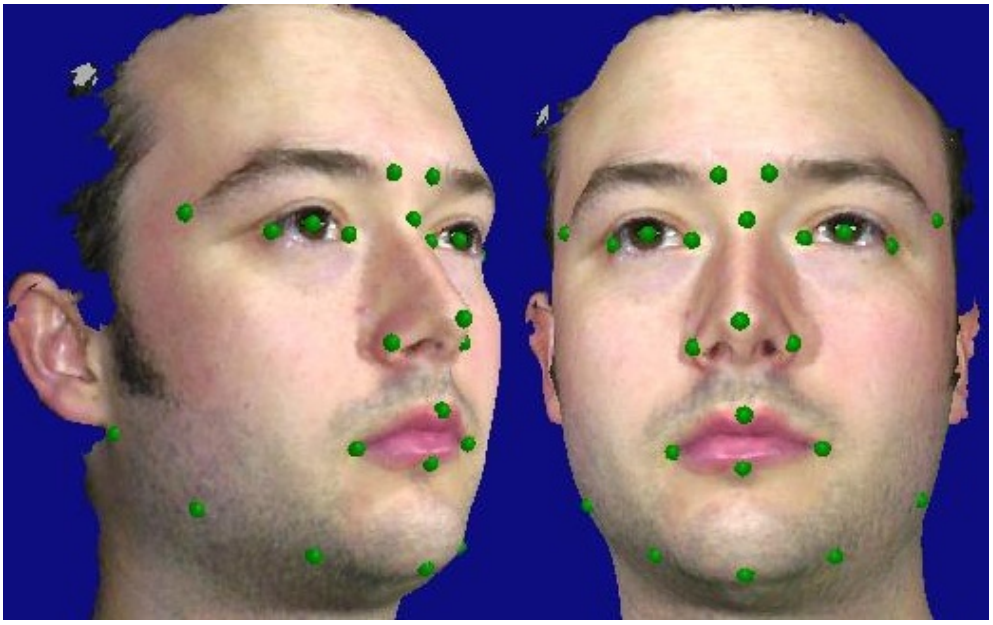


*Figure 3.2: An example of 3D shape model. The shape model is built from the vertices of the points marked on the face.*

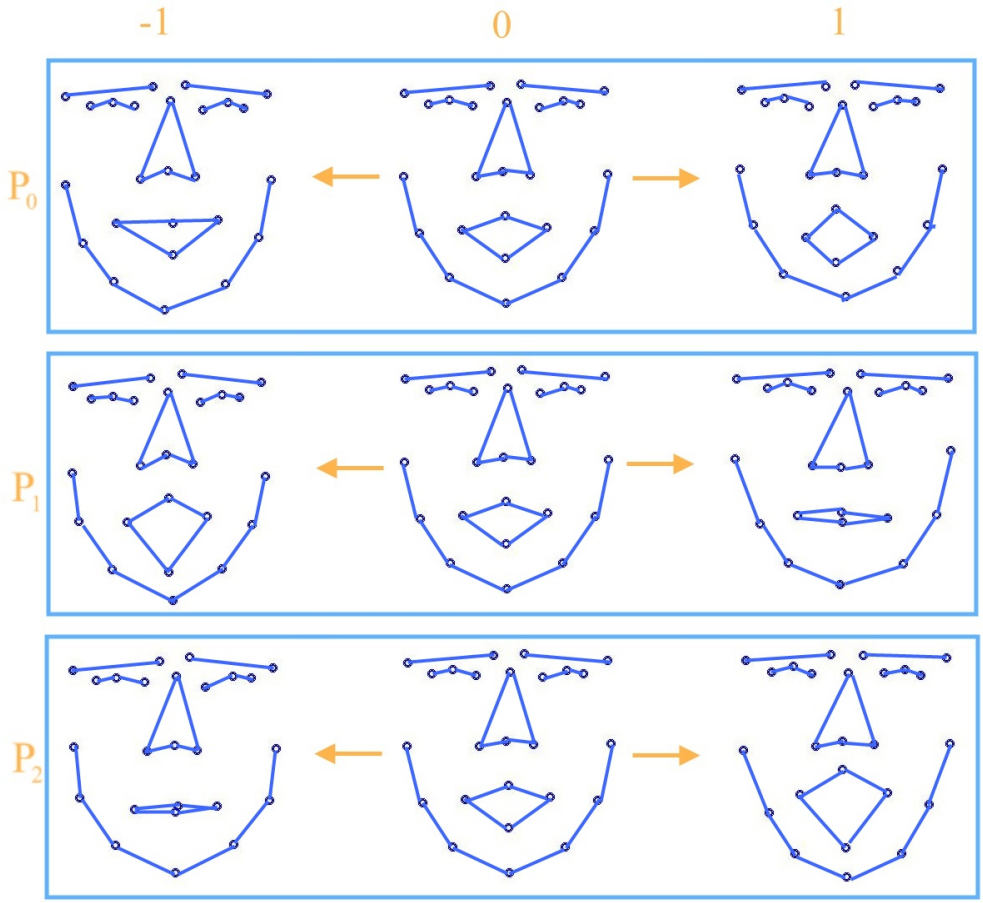



*Figure 3.3: The principal components corresponding to the facial feature movements with variation of the shape parameters.*

The 3D shape model is built from a training set of manually labelled faces as illustrated in *Figure* 3.2. The normalised shape coordinates, $s(x, y, z)$ are used to construct the linear model.

$$s = \bar{s} + P_s b_s \tag{3.1}$$

where $\bar{s}$ is the mean shape, $P_s$ is a set of orthogonal modes of variation and $b_s$ is a set of shape parameters. This equation can then be used to reconstruct a new shape from the given shape parameters.

The principal components of the shape model correspond to the jaw, lip,

eye and eyebrow movement, etc. These can be varied by adjusting the shape parameters $b_s$ and the effects can be seen in *Figure* 3.3. The two-dimensional coordinates of the shape model can be calculated with the following equation:

$$s_{2d} = M \cdot V \cdot (\bar{s} + P_s \cdot b_s) \tag{3.2}$$

where $V$ is a vector of the pose (translation, rotation, scaling) transforming parameters $T_x, T_y, T_z, S, \theta, \phi, \gamma$ and $M$ is the opengl frustum projection matrix described in Appendix A.

## 3.3 Texture Models

To build a model of the appearance, each 3D face model in the training set is rendered from a particular viewpoint and a square patch is sampled around each feature point (*Figure* 3.1). By transforming the face with different scale, rotation, shift and lighting parameters, a set of texture patches is built. Two sets of example texture patches sampled from the frontal view and side view are shown in *Figure* 3.4 and 3.5. The pixel values of each texture patch image are vectorised as:

$$g = (R_1, G_1, B_1, R_2, G_2, B_2, \cdots, R_n, G_n, B_n)^T \tag{3.3}$$

Then PCA analysis is applied to the texture patches from a particular viewpoint and resolution to build a texture model.

A specific set of patches, $g$ can be generated from the model using:

$$g = \bar{g} + P_g \cdot b_g \tag{3.4}$$

where $\bar{g}$ is the mean normalized grey-level vector, $P_g$ is a set of orthogonal modes of variation and $b_g$ is a set of grey-level parameters. Forty-five texture

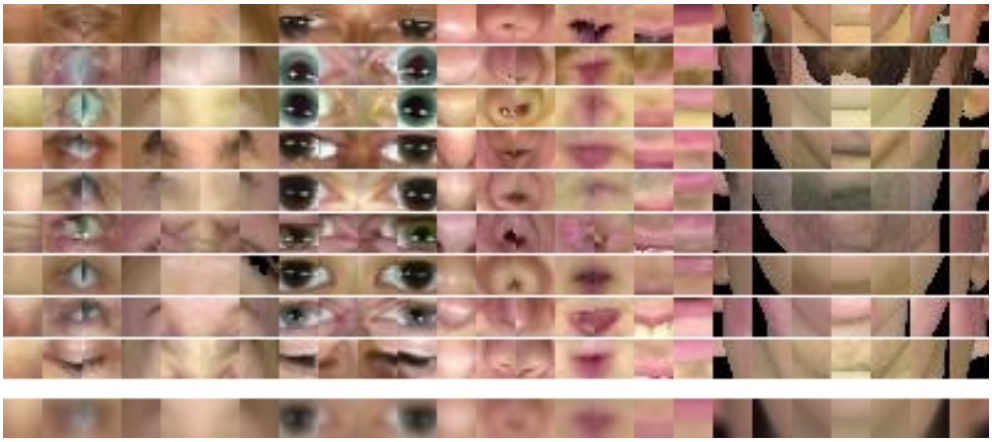


*Figure 3.4: Some example texture patches sampled from the frontal view. The bottom one is the average.*

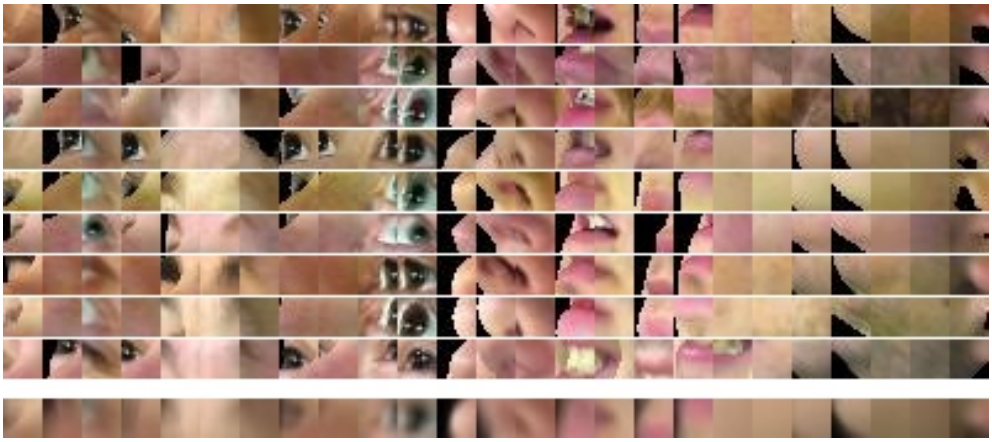


*Figure 3.5: Some example texture patches sampled from the side view. The bottom one is the average.*


*Figure 3.6: Texture model illustration*

models are built, one for each of 15 viewpoints (5 left-right and 3 up-down) across 3 different scales.

For a given face image, the following equation is used to analyse a given input patch set, $g$,

$$b_g = (g - \bar{g}) \cdot P_g^T \tag{3.5}$$

We can then construct a new face with the calculated parameters $b_g$ and the PCA model using *Equation* 3.4. This is shown in *Figure* 3.6.

In *Figure* 3.7, there are some samples of the variation corresponding to the varying of shape parameters, $b_s$, and appearance parameters, $b_g$.

### 3.3.1 Multiple Scale Technique

Multi-scale techniques are standard in computer vision and image processing. They allow short range models to extend over longer ranges and optimisation to be achieved in fewer steps. For each view, after the first texture model

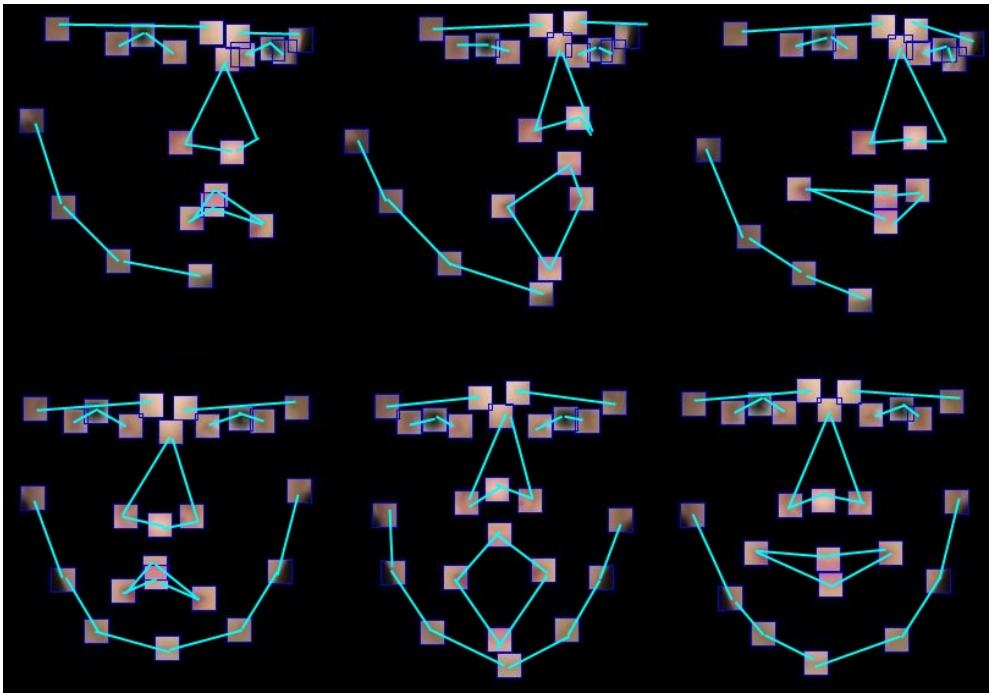


*Figure 3.7: Shape and Texture Variation. Three examples of shape model variation are shown from the side (top row) and front (bottom row) views. For each example, a texture model from the appropriate view is shown.*

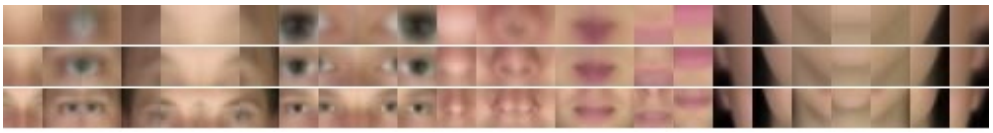


*Figure 3.8: Multiple Scale Texture Patches from the frontal view.*

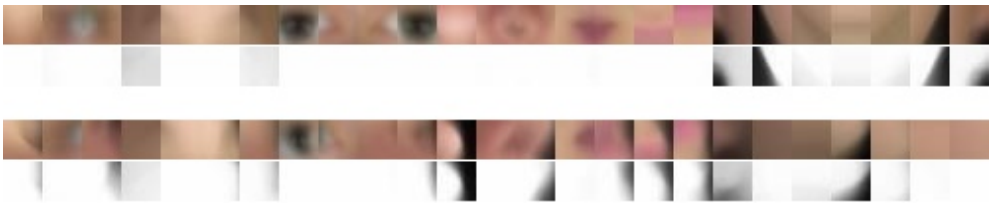


*Figure 3.9: The opengl alpha channel for background self occlusion. The upper pair is the average texture patches image and the alpha channel patches image from the frontal view. The bottom pair is from a side view.*

is built, another texture model is built from the low-pass filtered and sub-sampled training texture images. Several different scales are used for the model as shown in *Figure* 3.8. Because the texture patches have a lower resolution, they cover a wider searching range which could improve the convergence rate. In the algorithm, the lowest resolution image is first fitted to improve the performance.

### 3.3.2 Modelling of Background Pixels

To increase the stability with varied backgrounds, visibility information from the rendered textures is used to estimate occluded pixels. The opengl alpha channel (*Figure* 3.9) from the rendering canvas is grabbed when the texture patches are extracted to mark out the edges between the face and the background.

Because the training face images (see *Section* 3.4) consist of a face part and the transparent background (*Figure* 3.10), a filter patch image (the black

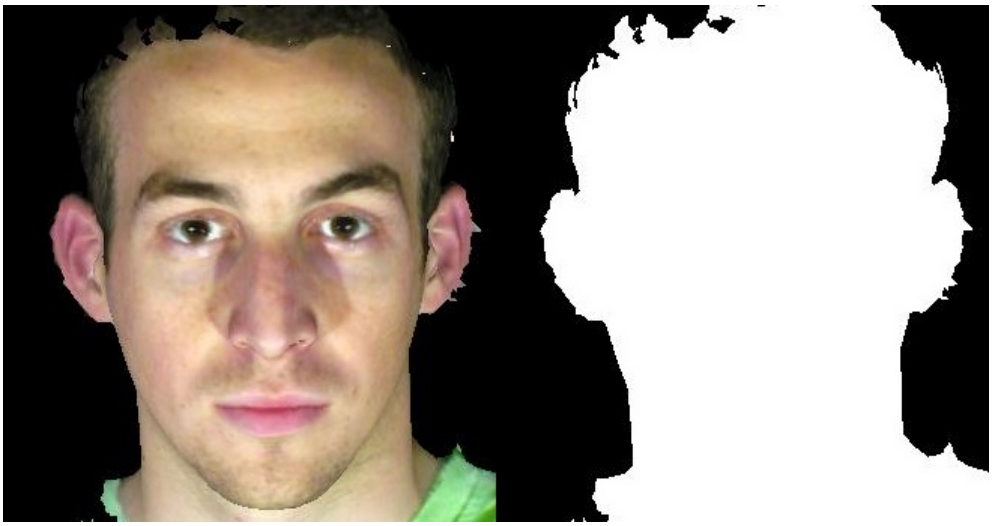


*Figure 3.10: An example of a rendered face. The image on the left is the texture map and the one on the right is the alpha map from the frontal view.*

/ white strip in *Figure* 3.9) can be created from the average image of the alpha value. A simple way is chosen to build the alpha channel information into the model. Before computing the errors between the synthetic $I_{syn}$ and extracted image patches $I_{ext}$, the average image patches of alpha channel $I_{alpha}$ are applied as a mask to both image patches by using pixel-wise multiplication.

### 3.3.3 Modelling of Self Occlusion

During head rotation, facial features can be blocked by other parts of the face. On the facial boundary, the appearance of background pixels can vary significantly (*Figure* 3.11). These effects could result in failure of the matching between the extracted image, $g(x)$ and the synthetic image, $f(x)$. In order to exclude the effects of these points, the texture models for different views are built with different sets of features.

Currently, a fixed visibility model is built for each viewpoint based on the feature points that are typically visible in that view. Self occlusion can

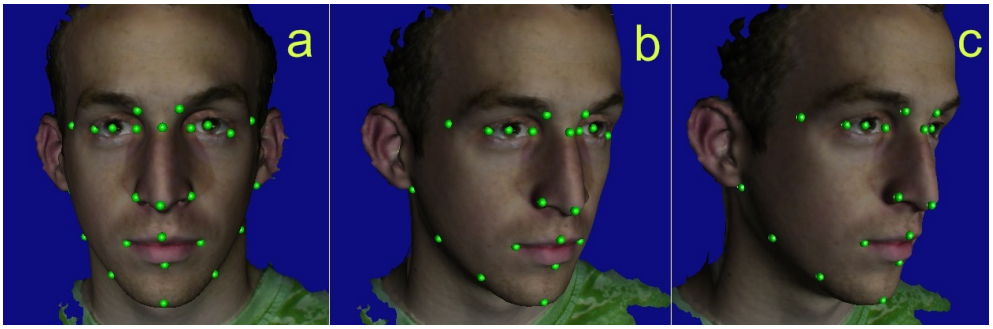



*Figure 3.11: Some features are blocked during head rotation. (a) is sampled from the frontal view; (b) is sampled from a view with a horizontal rotation of* $30°$*; (c) is sampled from a view with a horizontal rotation of* $60°$*.*

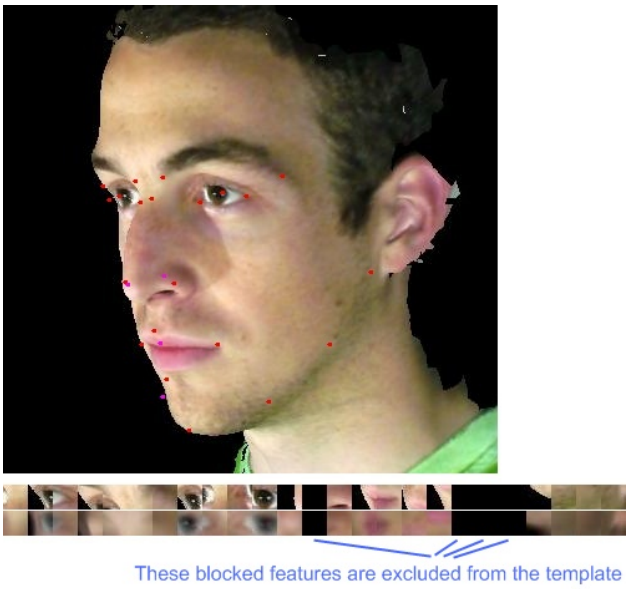



*Figure 3.12: An example of a texture model with hidden features from a side view. The hidden features are not extracted for the model.*

be detected in the training set by identifying model points that are further from the virtual camera than the rendered point, as given by the depth-buffer value. If the point is occluded in more than 50% of the training examples it is excluded from the model for that view. An example of a texture model from a side view can be seen in *Figure* 3.12.

## 3.4 Training Process

The system (*Figure* 3.1) consists of a model of 3D shape variation and fifteen models of the appearance variations in a shape-normalised frame. A training set of labelled images is required, where key landmark points are marked on each example object. Landmark points placed on a set of 3D face models are used to generate the 3D shape model. The texture model for each view is found by rendering the face model from the appropriate viewpoint and sampling square patches from the captured image about the projected location of the feature point. For each face in each view, the variation in the training set is increased by capturing several samples with the rotation varied within five degrees both in the horizontal and vertical direction.

Chen and Davoine use 20 feature points in their work [145], with most features located around the non-rigid motion units (eyes, eyebrows, lips). In order to allow the model to follow the jaw movements, a set of 25 facial features is used (*Table* 3.1). The points are manually labelled on the training set of 3D models. This set of feature points are chosen to make the model capable of tracking face movement and expression variation effectively.

A training set of labelled 3D face images are used, where key landmark points are marked on each example. There are 14 subjects (8 males, 6 females) performing 7 posed expressions (neutral, happy, sad, disgust,

*Table 3.1: The feature point used in the experiments.*

| | |
|---|---|
| 1 | Centre of the two eyes |
| 2 | Outer corner of the left eye |
| 3 | Outer corner of the right eye |
| 4 | Outer corner of the left eyebrow |
| 5 | Inner corner of the left eyebrow |
| 6 | Inner corner of the right eyebrow |
| 7 | Outer corner of the right eyebrow |
| 8 | Centre of the left eye |
| 9 | Inner corner of the left eye |
| 10 | Inner corner of the right eye |
| 11 | Centre of the right eye |
| 12 | Top of the nose |
| 13 | Left base of the nose |
| 14 | Right base of the nose |
| 15 | Left corner of the mouth |
| 16 | Right corner of the mouth |
| 17 | Uppermost point of the top lip |
| 18 | Lowermost point of the bottom lip |
| 19 | Left earlobe |
| 20 | Left cheek |
| 21 | Left chin |
| 22 | Centre chin |
| 23 | Right chin |
| 24 | Right cheek |
| 25 | Right earlobe |

*Figure 3.13: Example of training images.*

surprise, fear, anger) and 7 posed vise mesh (/ah/, /ch/, /ee/, /k/, /oo/, /p/, /th/) captured using a stereophotogrametric system (*www.3dMD.com*). A 20x20 block of pixels is extracted around each feature landmark point at each of 3 spatial scales (*Figure* 3.13). These patches are vectorised and used to build the texture model. All the features are formed into a 500x20 block of pixels strip before the PCA analysis is applied.

## 3.5 Texture Model Selection

In the proposed algorithm, there is a global 3D shape model and fifteen texture models. One additional step to the original algorithm is the selection of the texture model while searching with the Multi-view CLM algorithm. For tracking face movements, the algorithm has to be able to select the proper texture model automatically. To achieve this, a simple model selection

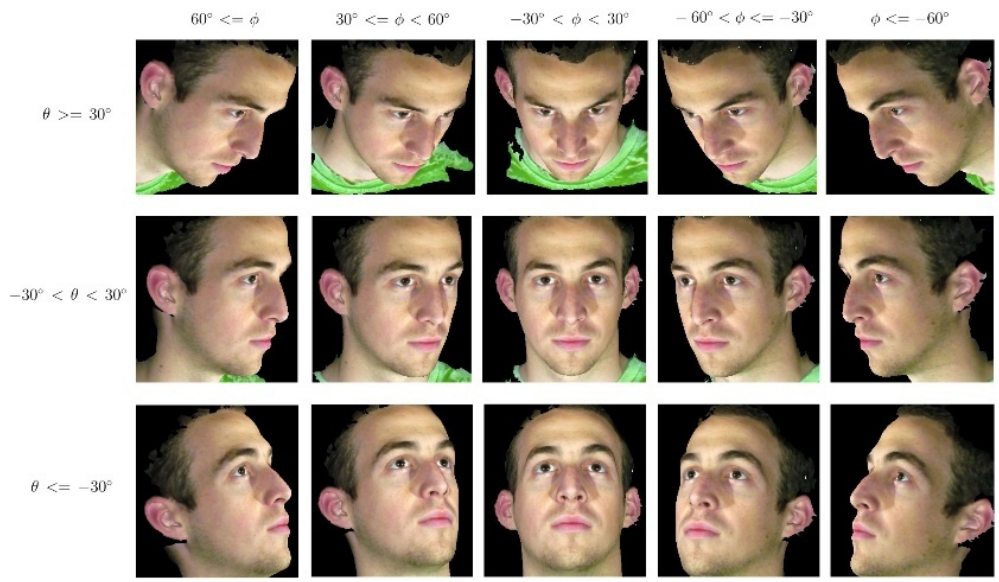



*Figure 3.14: Multiple texture models.*

method is applied. Each texture model covers a certain range of head rotations, the rotation parameters $\theta$ and $\phi$ can be used to estimate the view by testing the criteria shown in *Figure* 3.14. $\theta$ and $\phi$ can be obtained from the current estimate of head rotation using the methods described in the next section (Shape Template Update Methods).

The selection process of texture models is given by the following steps applied repeatedly until the end of the tracking.

1. The multi-view CLM is applied to the given frame accompanied with the initial parameters.

2. A set of new parameters is obtained including $\theta$ and $\phi$ which are the estimated rotation angles for the current face pose.

3. To estimate the next frame, $\theta$ and $\phi$ are then passed into the texture

model selection module to choose the proper appearance model.

## 3.6 Search Algorithm

In this section, the search process used in the original algorithm is briefly described. Then a search algorithm with the 3D model is described with the model switching technique applied.

### 3.6.1 Previous Algorithm

In the previous chapter, the construction of a multi-view constrained local model was described. To fit the model to a new image with a set of initial feature points, a two step iterative algorithm is used, alternating between estimating the local appearance model and then fitting it to the image using a shape constrained search. Below is a summary of the single-view CLM (*Figure* 3.15) search method.

1. Input an initial set of feature points and calculate the initial set of affine and shape parameters.
2. Repeat
    a. Grab the texture from a square area around each feature point
    b. Fit the appearance model to the current set of feature points to generate a set of model appearance patches.
    c. Repeat
        i. Grab the texture from a square area around each feature point.
        ii. Use the grabbed texture patch and the synthetic texture patch to calculate the shape model parameter updates.

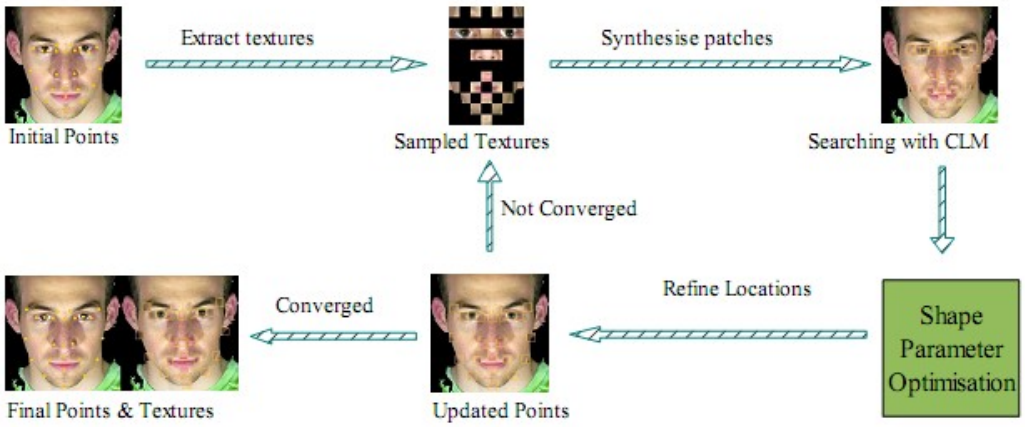



*Figure 3.15: The CLM algorithm uses an initial guess of the feature point positions to extract the grey-scale appearance around the point. The appearance sampled from the test image is projected into linear appearance space learnt from examples. The best match for these is then searched for in the image using a non-linear optimiser. The algorithm is iterated until convergence.*

iii. Synthesise the new shape model and project into the image space using the pose and camera parameters.

End

Until Converged.

### 3.6.2 The Search Process

Using the texture model selection algorithm and the three dimensional shape model, the searching method can be extended to large head rotations.

For a given set of initial points, $X = (x_0, y_0, z_0, x_1, y_1, z_1 \cdots, x_{n-1}, y_{n-1}, z_{n-1})$, the initial pose parameters $V$ are estimated for the shape model built in the previous chapter. Then the multi-view appearance CLM tracking algorithm shown in *Figure* 3.16 is applied. The optimising algorithms will be discussed

in detail in the next chapter.

1. Initialise with the global face detector.
2. Estimate the initial pose parameters $V$.
3. Repeat
   (a) Repeat
      i. Compute the feature coordinates, $s$, and extracts the feature patches, $g$.
      ii. Estimate the texture model from the pose parameters $V$.
      iii. Synthesise the feature patches from the updated coordinates and the selected texture model.
      iv. Apply the alpha channel feature and the hidden points feature to the extracted and synthetic feature patches.
      v. Optimise the error metrics with the shape template updating methods and gives a new set of pose and shape parameters, $V$, $b_s$.
   (b) Until converged.
4. Until converged for all selected scale.

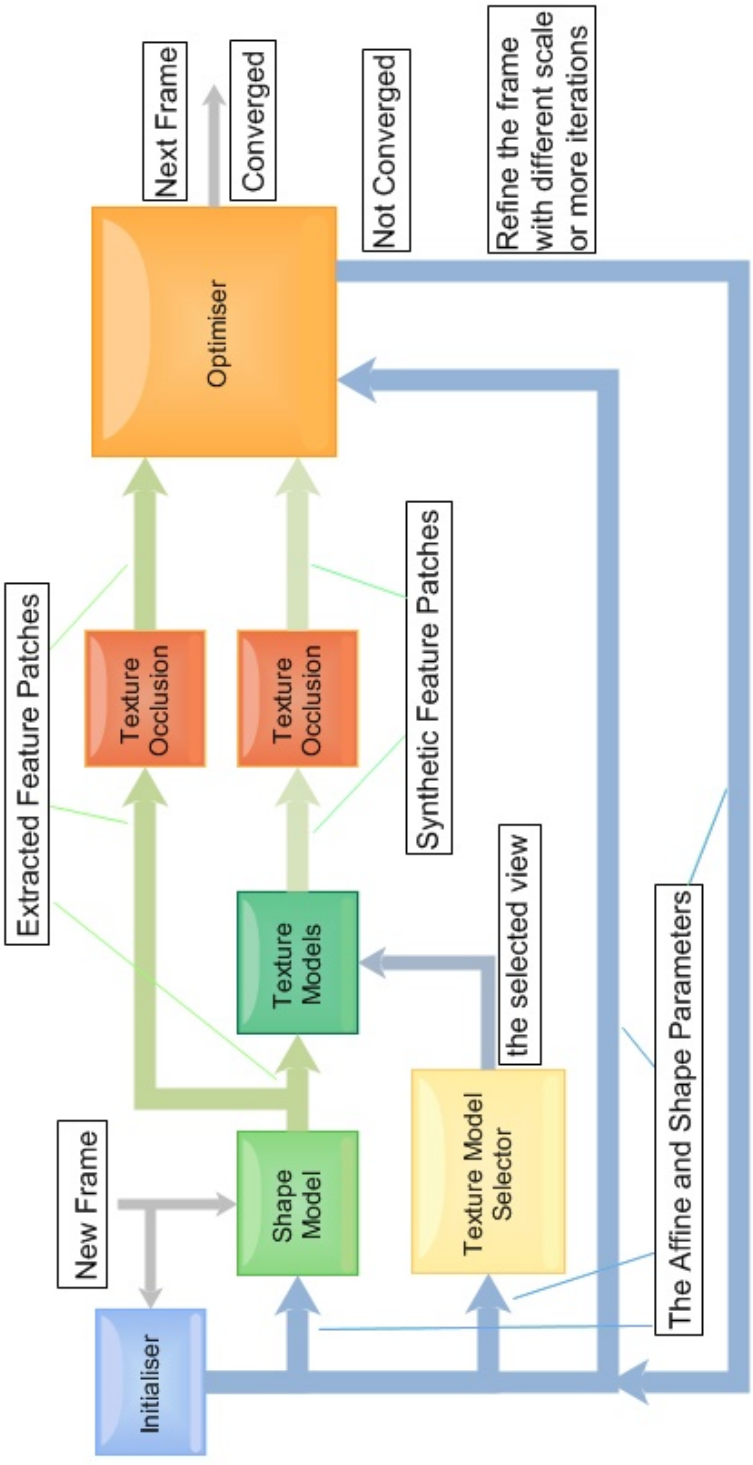



*Figure 3.16: Multi-view CLM tracking process*

## 3.7 Summary

The presented face model is a 3D constrained local model obtained by extending the original CLM algorithm [2]. A 3D shape model and fifteen texture models from different facial views are built to make the model suitable for 3D face feature location and tracking. For each view, a constrained local search on the given image is applied with the selected texture model and the shape model. Only one shape model is used for all the views, it's more convenient to apply the shape and appearance models separately. In that way, the face feature location is retained during the variation and switching of the texture model. The 3D face templates gathered by a 3DMD system are used as the training data. A principal component analysis is applied to the shape model and each texture model to reduce the number of parameters. A multi-scale technique is used to make the model fitting more efficient and effective. The alpha channel feature is also used to exclude background effects when fitting the model to the faces. For different views, different features are excluded from the model due to self-occlusion in the training set to improve the performance.

In the following chapter, the fitting procedure and the performance of the model will be discussed. The fitting metric and the estimation methods will also be discussed.

Chapter 4

# Optimisation Methods for Fitting and Tracking

In this chapter, the optimisation methods used to fit the model built in the previous chapter to a new face image is described. The aim is to find the best match between the model and the patches extracted from the target image. Four different shape updating methods are tested to select the most appropriate for use in the proposed algorithm – Powell's direction set method [146], Gauss-Newton iterative method [6; 9], a FastNCC algorithm [14] and the Condensation tracking algorithm [15].

## 4.1 Overview

The original CLM algorithm [2] used the Nelder-Meade simplex algorithm [13] to optimise the error function. This algorithm works by using N+1 samples in the N dimensional parameter space. In each iteration, the worst sample is discarded and a new sample is added based on a set of simple heuristics.

A generic non-linear optimisation of the NCC [146] is evaluated for the

shape and pose updating search. This is referred to as Powell's Direction set method. Another efficient direct method for maximising the NCC [14] is also applied. It is referred to as FastNCC in this work and Correlated Active Appearance Models (CAAM) in the original implementation.

The techniques described above are also compared with minimisation of the sum of squared errors (SSE) as an error metric. This is similar to the above, requiring the Jacobean and inverse Hessian matrices and solution of a linear system. This method is equivalent to the inverse additive AAM alignment,[8; 9] but with a different appearance model and wrapped in a slightly different fitting algorithm.

For tracking it is possible to use one of the above fitting methods for each frame, initialised using the result from the previous frame. An alternative is to use a method specifically designed for tracking across frames. To this end, a Condensation (CONditional DENSity propagATION) tracking method [15] is tested. The Condensation algorithm retains multiple estimates at each time point and a statistical dynamics model is used to generate pose estimates in the next frame from the current estimates. Estimates ranked higher in the current frame give rise to more samples in the proceeding frame in a selection process similar to genetic algorithm search. A simple implementation using independent Gaussian distributions for the dynamics model is tested.

## 4.2 CLM Fitting Algorithms

Like fitting an AAM to an image, fitting the proposed multi-view 3D CLM to an image is a non-linear optimisation problem. Previous approaches [4; 9; 37] iteratively solve for incremental updates to the parameters (the shape and appearance coefficients.) Given the current estimates of the shape

parameters, it is possible to warp the input image backward into the model coordinate frame and then compute an error image and the incremental updates to the parameters.

Cristinacce *et al.* [1; 2] introduced another framework to optimise the Normalised Cross Correlation (NCC) instead of the standard Sum of Squared Error (SSE). During the process, the Nelder-Meade simplex method is used to find the global rigid and non-rigid shape parameters that optimise the sum of NCCs across all the patches. It attempts to find a local maximum in each patch response surface and constrain these local maxima to be consistent with the global shape prior.

### 4.2.1 Powell's Direction Set Method

The original CLM implementation [1; 2] used Nelder-Meade simplex method [13; 146] for optimisation. Typically, Nelder-Meads method is thought to require more function evaluations than Powell's method [146]. This implies it could be more efficient to use Powell's method as the non-linear optimiser.

Powell's method [146] uses a set of orthogonal directions and performs a 1D minimisation along each. After the first iteration, a better set of directions can be found and the process is repeated until convergence. Powell's direction set method takes a simple strategy to minimise a multidimensional function $f(P)$, where $P$ is a concatenated set of transformation. $T_T$ and shape, $b_s$ parameters. It takes the unit vectors $u_0, u_1, \cdots, u_{N-1}$ as a set of directions. (*Figure* 4.1) Using Brent's method, [146] a one-dimensional line minimisation is used along the first direction to its minimum, then from there along the second direction to its minimum, and so on, and a new set of directions is given at the same time, which is used in the next iteration of minimisation. This can be seen in *Figure* 4.2.

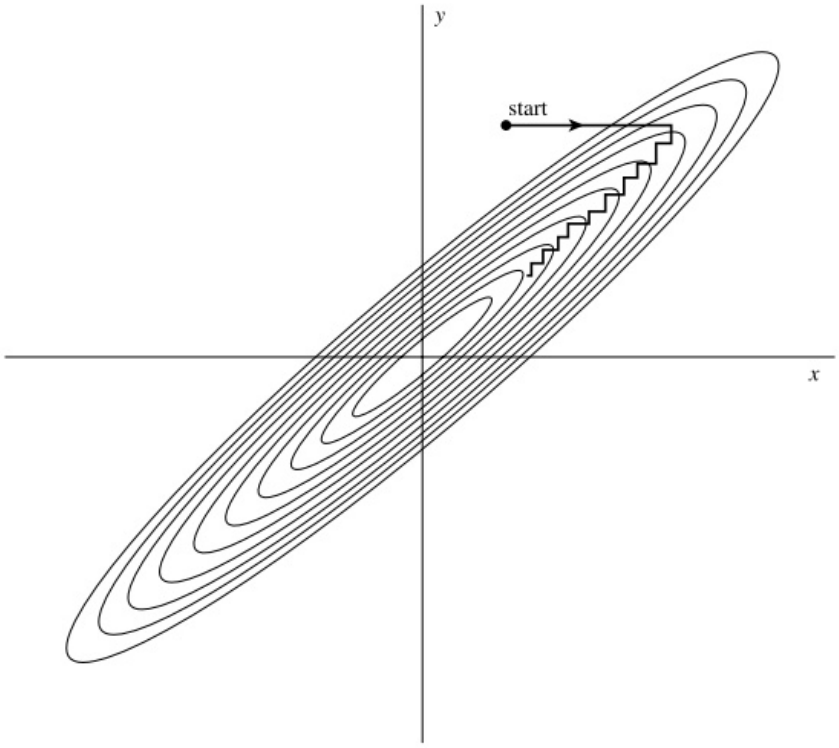



*Figure 4.1: Successive minimisation along coordinate directions in a long, narrow "valley" (shown as contour lines). The method takes many tiny steps to get to the minimum, crossing and re-crossing the principal axis. (Adapted from Numerical Recipes [146])*

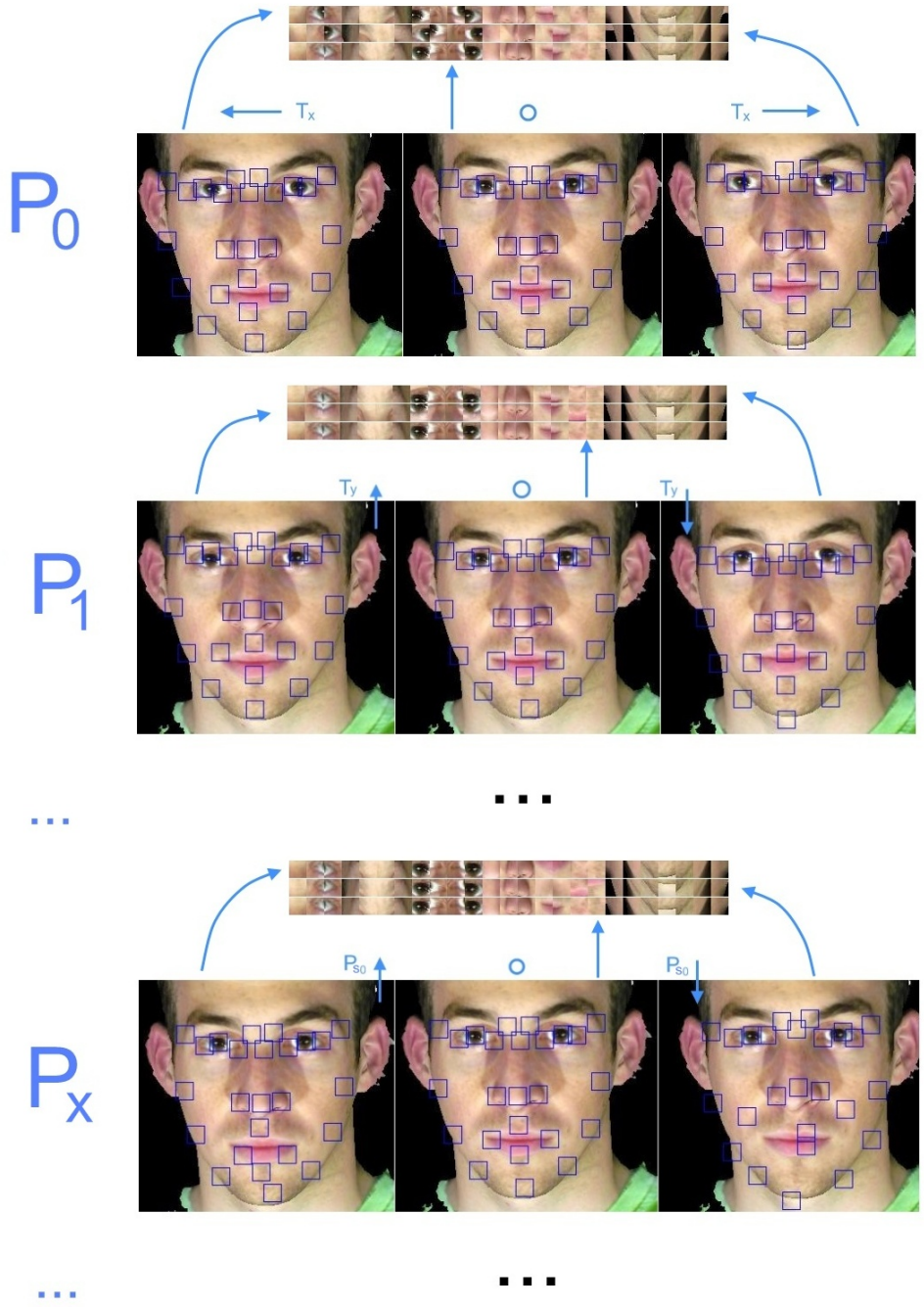



*Figure 4.2: The movement of the feature coordinates along with the change of the parameters during one iteration.*

The matching function is the cross correlation between two images f(x,y) and g(x,y).

In order to eliminate the effects from varying brightness of the images due to different lighting and exposure conditions, the images are first normalised. This is done by subtracting the means $\bar{f}$, and $\bar{g}$, and dividing by the standard deviations, $\sigma_f$ and $\sigma_g$. The normalised cross-correlation of the synthetic image $f(x, y)$ with the extracted image $g(x, y)$ is:

$$E = \frac{1}{n-1} \sum_{x,y} \frac{(f(x,y) - \bar{f})(g(x,y) - \bar{g})}{\sigma_f \sigma_g} \tag{4.1}$$

where n is the number of pixels in $g(x, y)$ and $f(x, y)$.

To search with Powell's method, initialise the set of directions $u_i$ to the basis vectors,

$$u_i = e_i \quad i = 0, \cdots, N-1 \tag{4.2}$$

Then repeat the following steps until the function stops decreasing:

1) Save the starting position as $P_0$

2) For $i = 0, \cdots, N-1$, move $P_i$ to the minimum along direction $u_i$ and call this point $P_{i+1}$.

3) For $i = 0, \cdots, N-2$, set $u_i \leftarrow u_{i+1}$.

4) Set $u_{N-1} \leftarrow P_N - P_0$.

5) Move $P_N$ to the minimum along direction $u_{N-1}$ and call this point $P_0$.

### 4.2.2 Gauss-Newton Iterative Method

Lucas-kanade's image alignment method [5] has been adapted for efficient AAM fitting [4; 9; 10]. In this work, the inverse additive template matching

Gauss-Newton Iteration Method

Pre-compute:

3) Evaluate the gradient $\nabla f$ of the synthetic image $f(x)$

4) Evaluate the Jacobian $\frac{\partial f}{\partial b_s}$ at $(x; 0)$

5) Compute the steepest descent images $\nabla f \frac{\partial f}{\partial b_s}$

6) Compute the Hessian matrix $\sum_x \left[\nabla f \frac{\partial f}{\partial b_s}\right]\left[\nabla f \frac{\partial f}{\partial b_s}\right]$

Iterate:

1) Extract image $g(x)$ from the face image, $I(x)$

2) Compute the error image $g(x) - f(x)$

7) Compute $\sum_x \left[\nabla f \frac{\partial f}{\partial b_s}\right]^T [g(x; b_s) - f(x)]$

8) Computer $b_s \leftarrow b_s + \Delta b_s$

9) Update the feature coordinates $s = \bar{s} + P_s b_s$

Until $||\Delta b_s|| \leq \varepsilon$

approach [6; 9] is adapted for fitting the 3D multi-view model. To improve the accuracy of the feature position and keep track on them while the face is moving, the approach estimates the parameters (both shape and pose) iteratively to minimise the squared errors given by A.14 in *Appendix A*. The searching process can be seen in *Figure* 4.3 and more details of the algorithm can be found in *Appendix A*. The update algorithm is summarised in *Figure* 4.3.

### Derivatives Computation

For the Gauss-Newton template update methods, the gradients $\nabla f$ needed to be computed followed by the Jacobian $\frac{\partial f}{\partial b_s}$ and an approximation to the Hessian matrix $H$. The kernel used to compute the template gradients

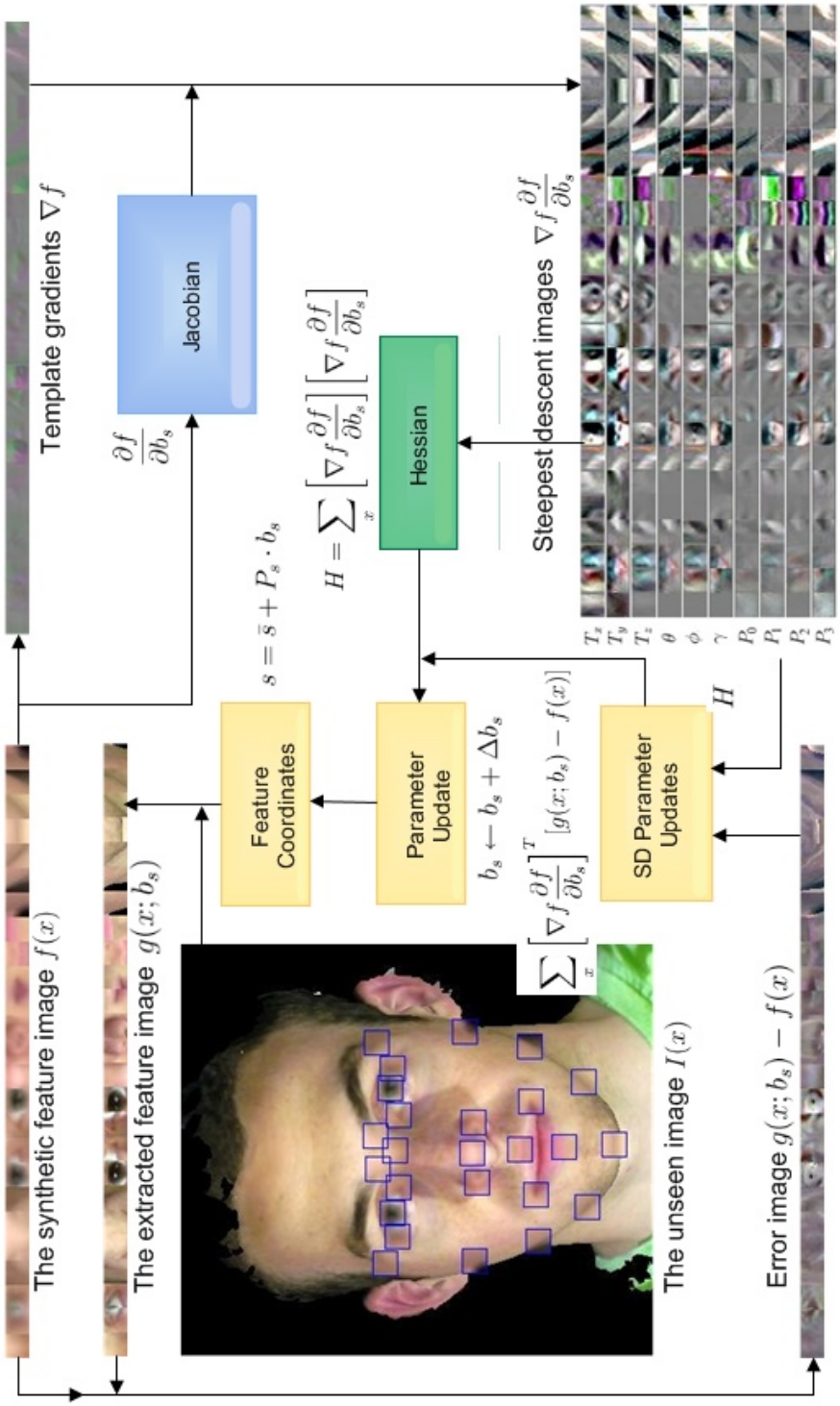



*Figure 4.3: Update parameters with Gauss-Newton algorithm.*

(derivatives) will be discussed in detail.

The image gradients, $\nabla f$, which is the partial derivative of $f(x, y)$ with respect to $x$ and $y$, can be approximated using a convolution process.

$$\begin{aligned} \frac{\partial f}{\partial x} &= \lim_{\delta} \frac{f(x+\delta, y) - f(x, y)}{\delta} \\ \frac{\partial f}{\partial y} &= \lim_{\delta} \frac{f(x, y+\delta) - f(x, y)}{\delta} \end{aligned} \tag{4.3}$$

a partial derivative can be estimated as a symmetric finite difference:

$$\begin{aligned} \frac{\partial h}{\partial x} &\approx h_{i,j+1} - h_{i,j-1} \\ \frac{\partial h}{\partial y} &\approx h_{i,j+1} - h_{i,j-1} \end{aligned} \tag{4.4}$$

Smoother estimates of the derivative are often desirable, and can be obtained by convolution with the derivative of a smooth sampling function. The following is the kernel used in this work:

$$derivativefilter = [-^1/_4, -^1/_2, 0, ^1/_2, ^1/_4] \tag{4.5}$$

which is the derivative of a cubic polynomial curve approximating a Gaussian [147].

A set of steepest descent images calculated from sampled derivatives, and the Jacobian of the feature patches are shown in *Figure* 4.4.

### 4.2.3 FastNCC Algorithm

Tiddeman *et al.* [14] proposed an algorithm which gives better results than SSE when fitting AAMs to unseen faces.

They derived an analytic solution to the NCC minimisation problem for AAM alignment rather than relying on an off-the-shelf non-linear optimisa-

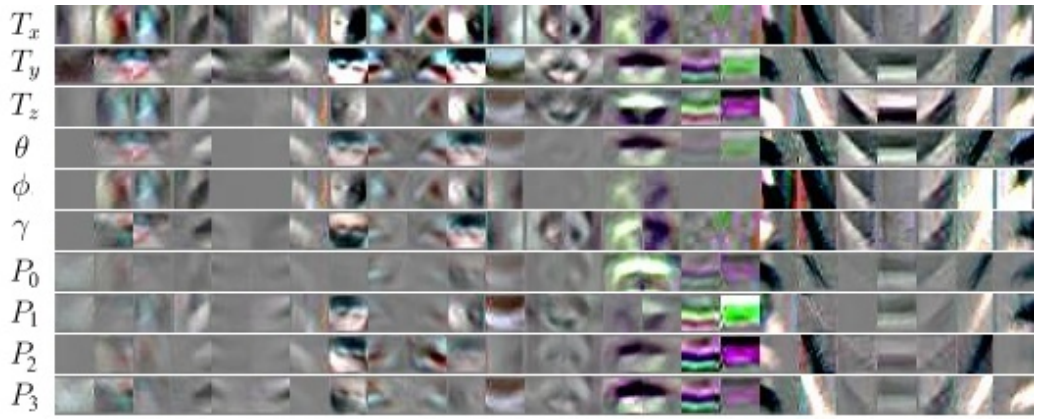



*Figure 4.4: The Steepest descent image,* $\nabla f \frac{\partial f}{\partial b_s}$.

tion routine. The method differs from the Gauss-Newton method described above, because the method takes the normalised cross correlation as the error metric. Optimisation techniques based on off-the-shelf non-linear optimisers like those described above are typically slow to converge. It is possible to optimise the global NCC directly using an estimate of the Jacobean and Hessian matrices and solving a linear system and a quadratic equation.

FastNCC is a stable and efficient method for solving AAM fitting problems. It could prove suitable for solving CLM fitting problems.

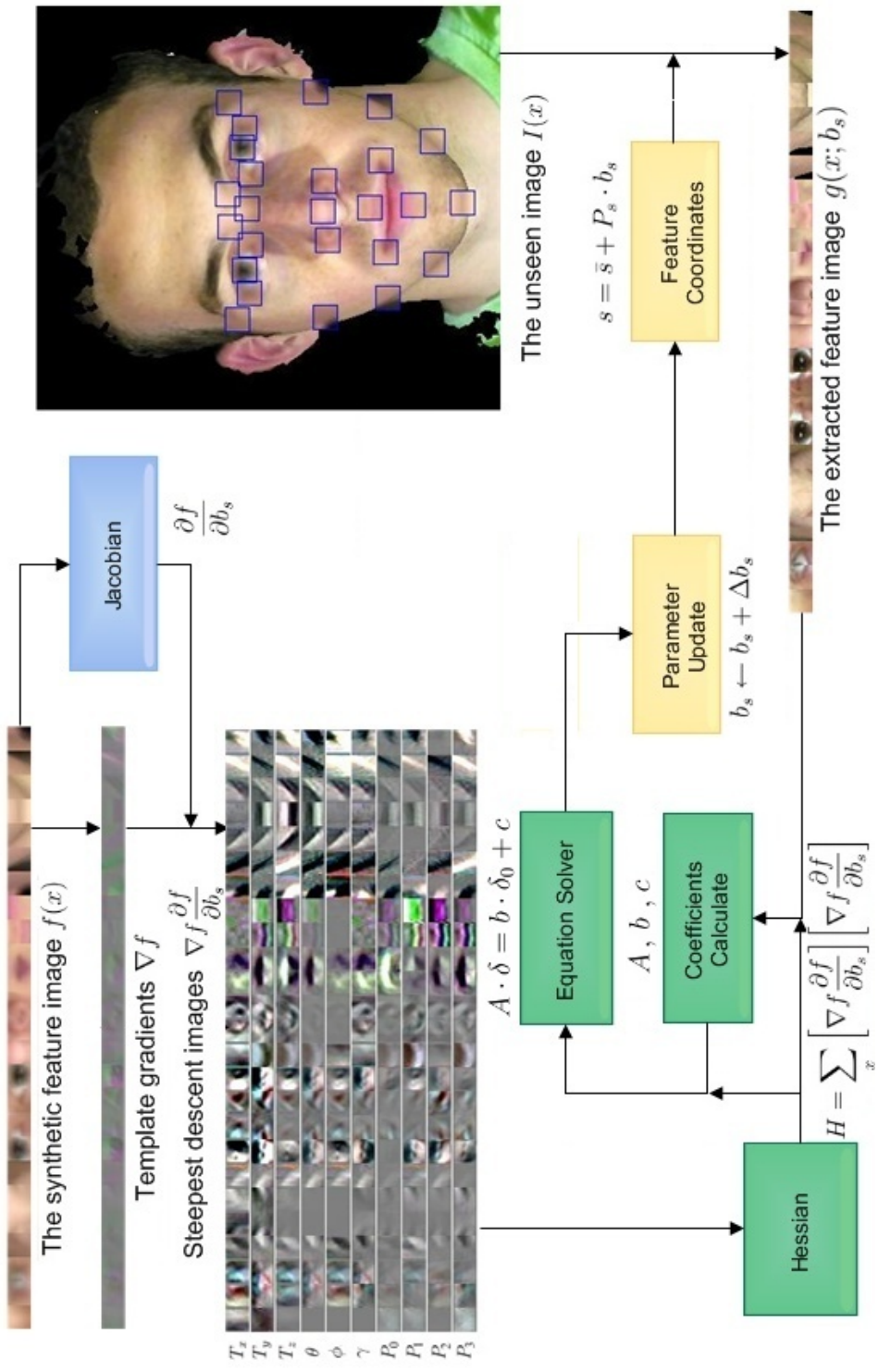



*Figure 4.5: Update parameters with FastNCC algorithm.*

FastNCC Iteration Method

Pre-compute:

3) Evaluate the gradient $\nabla f$ of the synthetic image $f(x, y)$

4) Evaluate the Jacobian $\frac{\partial f}{\partial b_s}$ at $(x; 0)$

5) Compute the steepest descent images $\nabla f \frac{\partial f}{\partial b_s}$

6) Compute the Hessian matrix $\sum_x \left[\nabla f \frac{\partial f}{\partial b_s}\right]\left[\nabla f \frac{\partial f}{\partial b_s}\right]$

Iterate:

1) Extract image $g(x, y)$ from the face image, $I(x)$

2) Compute the coefficients $A, b, c$

7) Compute $\Delta b_s$ by solving *Equation* A.26

8) Computer $b_s \leftarrow b_s + \Delta b_s$

9) Update the feature coordinates $s = \bar{s} + P_s b_s$

Until $||\Delta b_s|| \leq \varepsilon$

## 4.3 Tracking

The CLM algorithm automatically adjusts the feature templates to match the current image. Therefore, it is a natural tracking method in the sense that the templates learn to match the image, but are also constrained by the shape and texture model to remain plausible feature templates.

One approach to extend the algorithm and the optimising methods described above for tracking faces in videos is to apply the algorithm simply to each frame, initialising the search as the output from the previous frame. This multi-view appearance CLM algorithm for tracking can be illustrated in the following flow diagram (*Figure* 4.6). For each frame, a Gaussian pyramid [148] is built and then a CLM search is applied at each layer from the coarsest

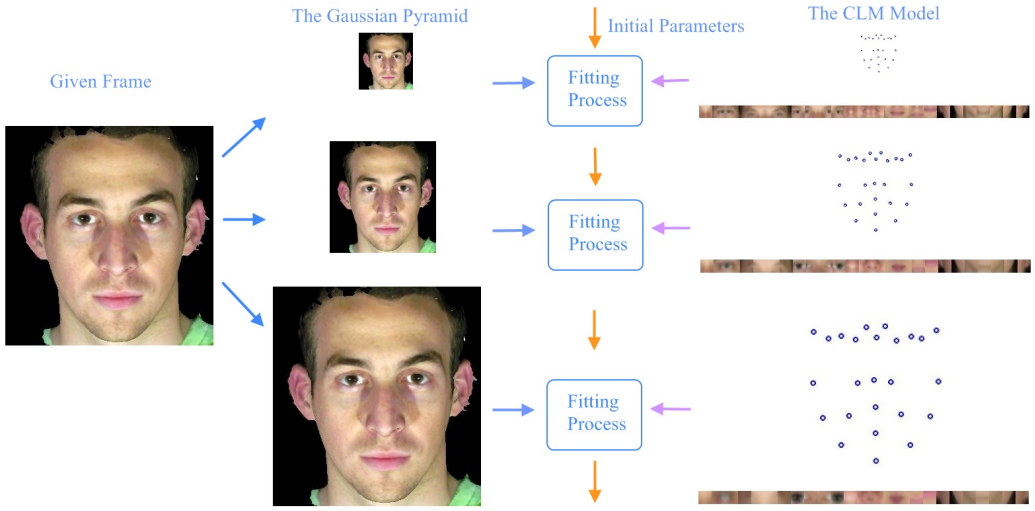



*Figure 4.6: A skeleton of the three scales image searching technique.*

to the finest.

An alternative approach is to use a model of the dynamics to predict the configuration in the following frame. This can allow for more localised searches and so faster and more robust tracking. A commonly used method for recursively predicting and updating the model parameters of a linear dynamic system is the Kalman filter [149], which is restricted to situations where the probability distribution of the state-parameters is unimodal. In the presence of occlusion, cluttered background resembling the tracked objects, and complex dynamics, the distribution is likely to be multi-modal. The Condensation algorithm [15] is a better alternative tracking algorithm, which is based on sampling the posterior distribution estimated in the previous frame and propagating these samples to form the prior one for the current frame. The method has shown to be a powerful alternative to the Kalman filter [143; 144].

### 4.3.1 Condensation

Object tracking is one of the more basic and difficult aspects of computer vision and is generally a prerequisite for object recognition. Being able to identify and follow the pixels in an image that make up the contour of an object is a non-trivial problem. Condensation is a probabilistic algorithm that attempts to solve this problem. The principal application of the condensation algorithm [15] is to detect and track the contour of objects moving in a cluttered environment. The original authors have implemented a mixed discrete/continuous tracker in the Condensation framework which switches between multiple continuous Auto-Regressive Process motion models according to a discrete transition matrix.

One of the advantages of the condensation algorithm is that it can propagate the probability densities of many states in the search space, it is particularly useful for multiple tracking problems. Generalisation of a particular movement is defined by the shape parameters $b_s$ and pose parameters $T_x, T_y, T_z, S, \theta, \phi, \gamma$. Each state in the search space contains values for these parameters. It's not necessary to record all the states during tracking for this case. Only the current state and the next state are needed. A state is defined as $s_n$.

$$s'_n = s_n + G \cdot v_n \tag{4.6}$$

where $s'_n$ is the new set of parameters to be generated. $s_n$ is the current set of parameters. $G$ is a random generated Gaussian distributed number. $v_n$ is the initialised weights for each parameter.

Essentially, the condensation algorithm consists of four basic steps; initialisation, selection, prediction and updating, as shown in *Figure* 4.7. A brief description of the condensation algorithm can be found in [150]. The

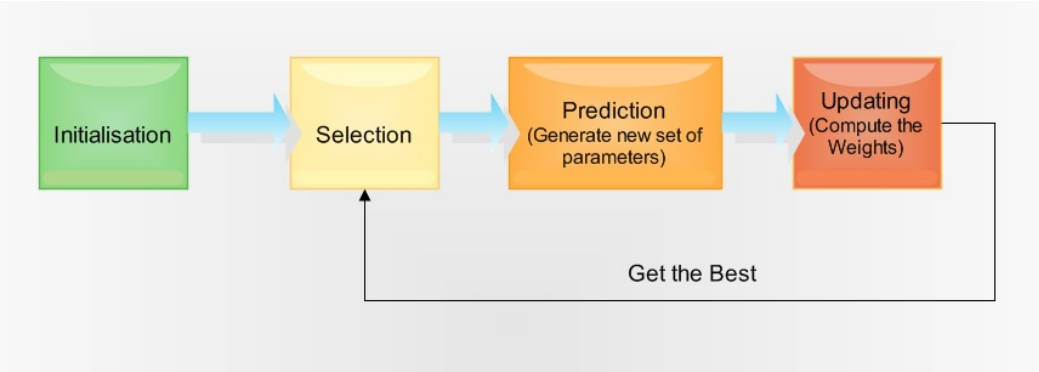



*Figure 4.7: High level block diagram of the condensation algorithm.*

search space is first initialised by choosing likely sample states. This produces a set of samples, $s_n$. The purpose of the algorithm is to find the most likely state that creates the best match for the input or observation data. This is calculated as the probability of the current observation $z$ given the state $s_n$.

To measure and weight the new position in terms of the measured features $z$, *Equation* 4.7 is used to compute the weightings. A new state can then be given.

$$r_n = E/r \tag{4.7}$$

where $r = \sum_i^n r_i$. $r_i$ is the normalised cross correlation, $E$, calculated by *Equation* 4.1.

1. Initialise
2. Repeat
   (a) Repeat
      i. Generate a new set of parameters from *Equation* 4.6.

ii. Compute the normalised cross correlation, $r_i$, with the new set of parameters.

(b) Until $i = n\_samples$.

(c) Compute the weights from *Equation* 4.7.

(d) Get the Best parameters set with the smallest weight.

3. Until $n = n\_iteration$.

4. Converged.

## 4.4 Summary

In this chapter, a fitting algorithm is derived from Cristinacce *et al.* [1; 2] and inverse additive algorithm [6; 9]. Three optimising methods including Powell's Direction Set method [146], Gauss-Newton method [6; 9] and FastNCC algorithm [14] were described. A simple condensation algorithm [15; 140] for tracking human faces was also described.

In the next chapter, the model and the search algorithms are tested for fitting to the given images and tracking human face feature with large head rotations. Factors and parameter settings that affect the performance of the model are also tested.

Chapter 5

# Experiments

In previous chapters, the structure of the model and the processes of fitting and tracking were discussed. In this chapter, experiments are described in order to evaluate the performance of the proposed algorithms.

In the first section, the testing data sets and error metric for the experiments are introduced. The statistical methods that will be used for analysing the data gained from the experiments are also discussed. In the following sections, results of the experiments are discussed. The effectiveness of the proposed multi-view 3D CLM algorithm is compared to the single-view 2D CLM algorithm by fitting to synthetic data. The effectiveness and performance of the proposed multi-view 3D CLM algorithm is tested with different optimising methods by fitting to real data. The factors that influence the performance of the system are also investigated.

## 5.1 Overview

To evaluate the algorithm described in the previous chapters, a set of experiments were conducted, these are described in this chapter. Before describing the experiments, the testing data sets and several measures for

*Figure 5.1: Example frames from the test sequences synthesised from rendered 3DMD data.*

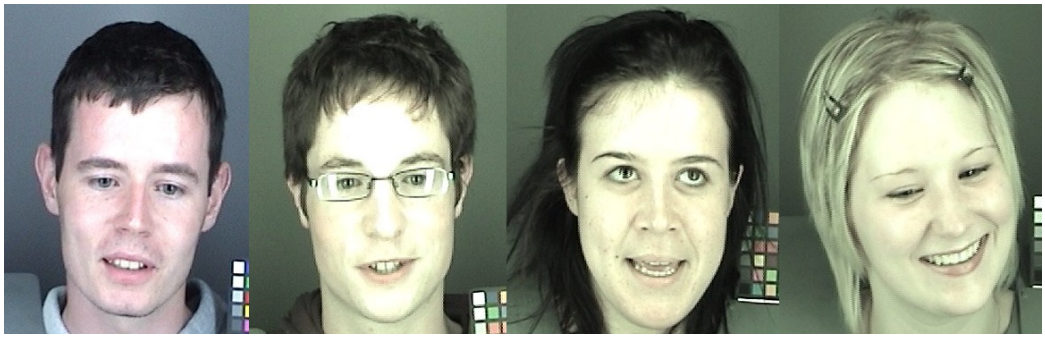


*Figure 5.2: Example frames from the test sequences from the real video data.*

evaluating the algorithm are described.

### Testing Data Sets

A mixture of synthetic and real data has been used for the experiments. Synthetic data (*Figure* 5.1) is generated by rendering multiple 3D face scans (from a 3DMD system) from different viewpoints. It is helpful in measuring the performance of the fitting because the 3D models provide accurate ground-truth data. Evaluation of the algorithm is also conducted using real video data (*Figure* 5.2) with hand-labelled feature points.

### Error Metric

To evaluate the accuracy of the fitting, the difference between the projected locations of the model's synthesised feature points and their true location in the target image is calculated. This is measured with the following equation:

$$d_e = \frac{\sum \sqrt{(x_{std} - x_c)^2 + (y_{std} - y_c)^2}}{Nd} \tag{5.1}$$

where $x_{std}$, $y_{std}$ represent the manually placed "ground truth" feature points locations, $x_c$, $y_c$ represent the tracked feature points locations, $d$ represents the distance between the centre of the eyes and $N$ is the number of features.

### Facial Region

The quality of the fitting may vary in different facial regions. The fitting performance could be improved by putting additional constraints on the worse fitted regions. To investigate the fitting performance of the algorithm, the face is split into jaw, mouth, nose and two eyes regions and the errors are calculated separately as illustrated in *Figure* 5.3.

### Statistical Analysis

A number of simple, standard statistical parameters are calculated on the data to aid understanding of the behaviour of the algorithm while fitting. In the following experiments, the mean and standard deviation of the data are calculated and a cumulative distribution function is used to analyse the data. To assess the significance of the experimental results, an ANOVA test including Student's t-test is applied to the data. An ANOVA test is used to test for differences among two or more independent groups and Student's t-test is used to test for differences among two independent groups. The

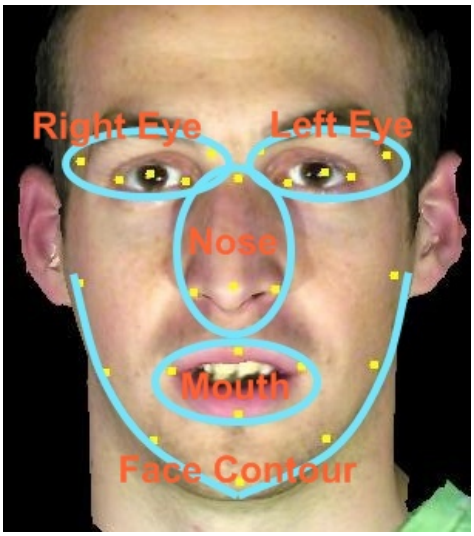



*Figure 5.3: The average distances between the tracked feature points and the corrected feature points of the faces divided by parts. The horizontal line at each group is the average distance of all the face feature points.*

program looks to see what the variation (variance) is within the groups, then works out how that variation would translate into variation (i.e. differences) between the groups, taking into account how many subjects there are in the groups. If the observed differences are a lot larger than expected by chance, they are considered statistically significant. More details can be found at [151].

## 5.2 Multi-view CLM Experiments

The original CLM algorithm was aiming to detect and track facial features for the frontal view with small head rotations. In order to make the algorithm work with large head rotations, the algorithm is extended to three-dimensions by building several texture models from different views. This experiment aims to compare the performance of the proposed multi-view 3D CLM

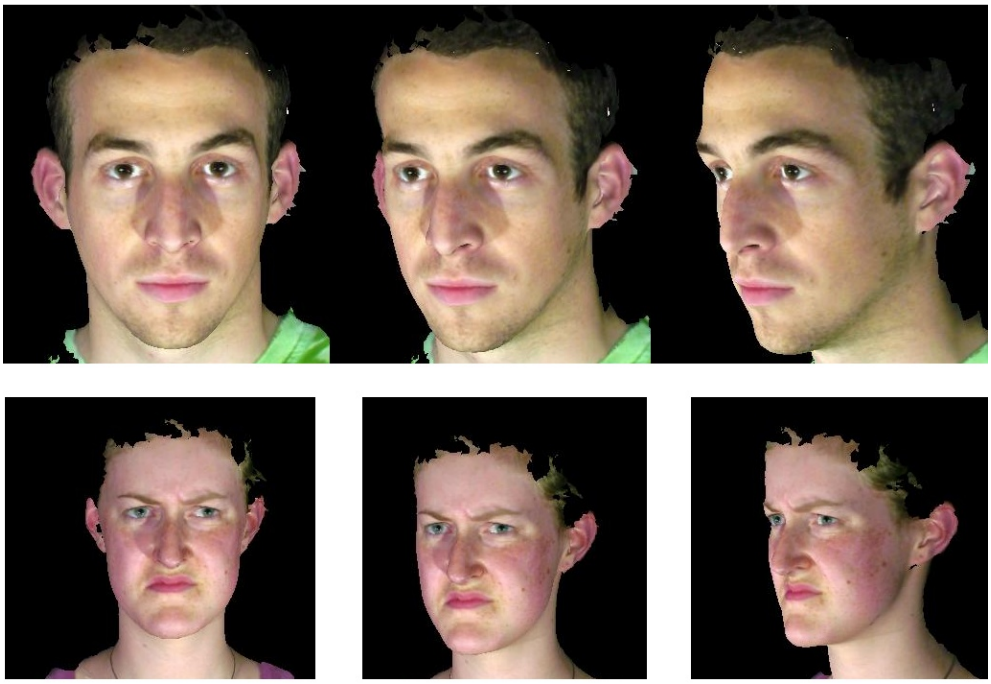


*Figure 5.4: Examples of the image sequence set used in the view setting experiments. From the left to right are frontal view, 25° side view and 45° side view.*

algorithm to the single-view 2D CLM algorithm [1; 2].

A set of face sequences (*Figure* 5.4) with fixed expression and head rotation of over 40 ° from the front are rendered using 3D data captured from a 3DMD stereoscopic capture system. Ten sequences comprising 700 images are used in the experiment. Both 2D single-view and 3D multi-view methods are applied to the same set of face sequences using the FastNCC algorithm [14] as the optimisation method. An illustration can be seen in *Figure* 5.5. In this section, the limit for single-view 2D CLM tracking of human face sequences with large head movements is also described.

The statistical results are shown in *Figure* 5.6 and *Table* 5.1. It can be seen that the fitting with the multi-view CLM algorithm converges better. With the multi-view model, it takes fewer steps to converge. The fitting

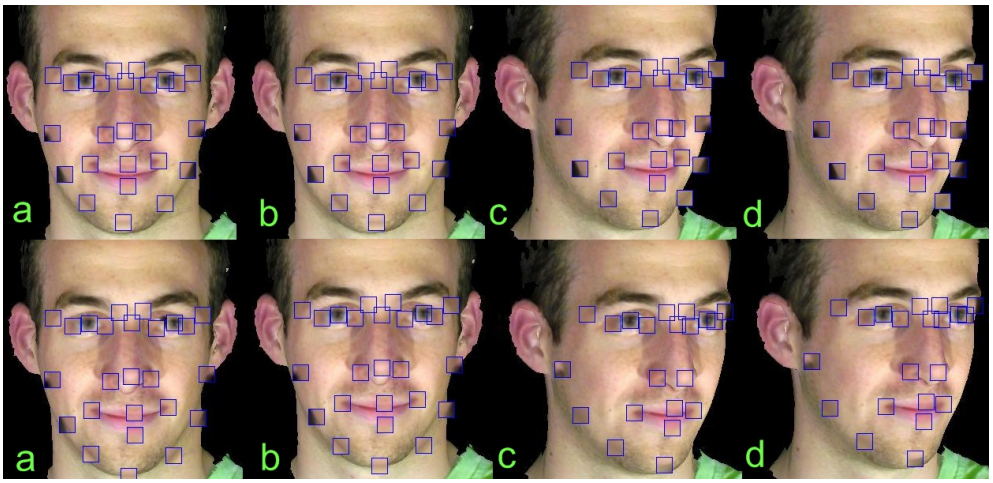



*Figure 5.5: Each row consists of a set of selected frames from a tracking sequence with the synthetic texture patches drawn on, indicating the location of the features. The results in the first row are from the single-view approach and the second row are from the multi-view approach. When the rotating angle reaches certain angles (b,c), the algorithm continues tracking the face well by automatically switching the texture model to a side view while the patches start drifting off the correct position with the single-view model.*

becomes more efficient and the fitting speed is improved. To investigate the difference further between the two methods, a Student's t-test (p=0.05) has been carried out for the error metric from both methods. The hypothesis is $\mu_0 < \mu_1$, where $\mu_0$ is the mean eye distance with the single-view CLM algorithm and $\mu_1$ is with the multi-view CLM algorithm. The Sig. is $8.16e-5$ showing that the fitting process is more stable with the multi-view CLM algorithm.

To investigate the performance of the two algorithms further, the average errors are compared frame by frame as can be seen in *Figure* 5.7. In these experiments, the first frame is the frontal face image and the face rotates one degree per frame. From the results, the errors calculated from different

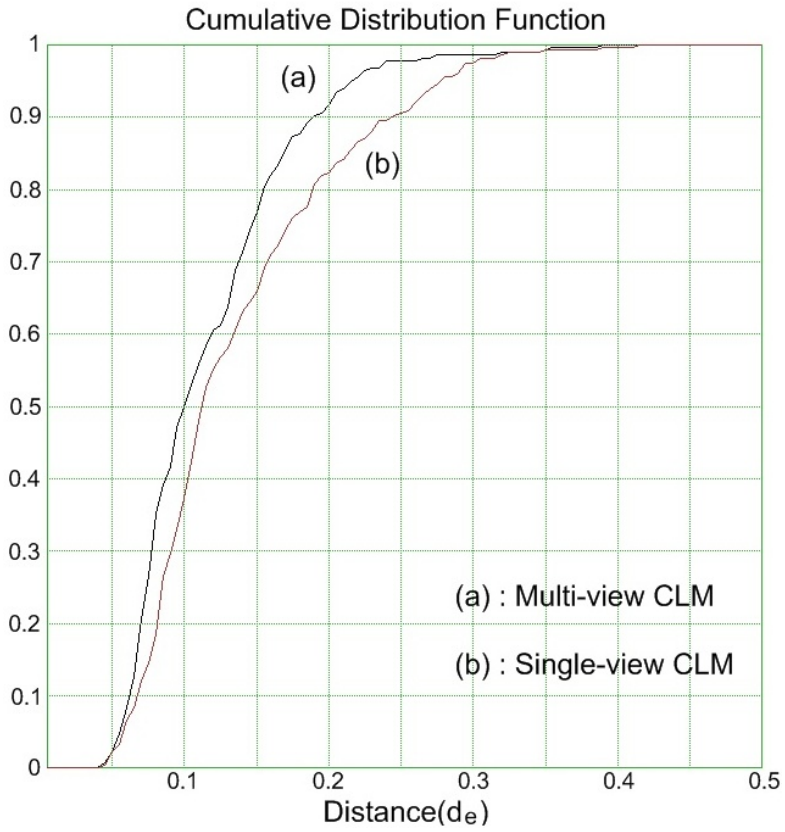



*Figure 5.6: The error comparison between the multi-view 3D CLM algorithm and the single-view 2D CLM algorithm. The experiments are applied to synthetic images.*

views remain approximately constant for the multi-view CLM algorithm.

A texture model from one view could possibly cover the rotation range around 20°–25°. However, one has to be cautious about the criterion area because the fitting could fail at that area referring to *Figure* 5.8. The two adjacent texture models should overlap on some range. In order to do that, a different texture model is chosen for every 30°. This means each texture model covers 15° in each direction. To meet this requirement, a fifteen texture model system is built, which covers about 100° in the vertical direction and 160° in the horizontal direction.

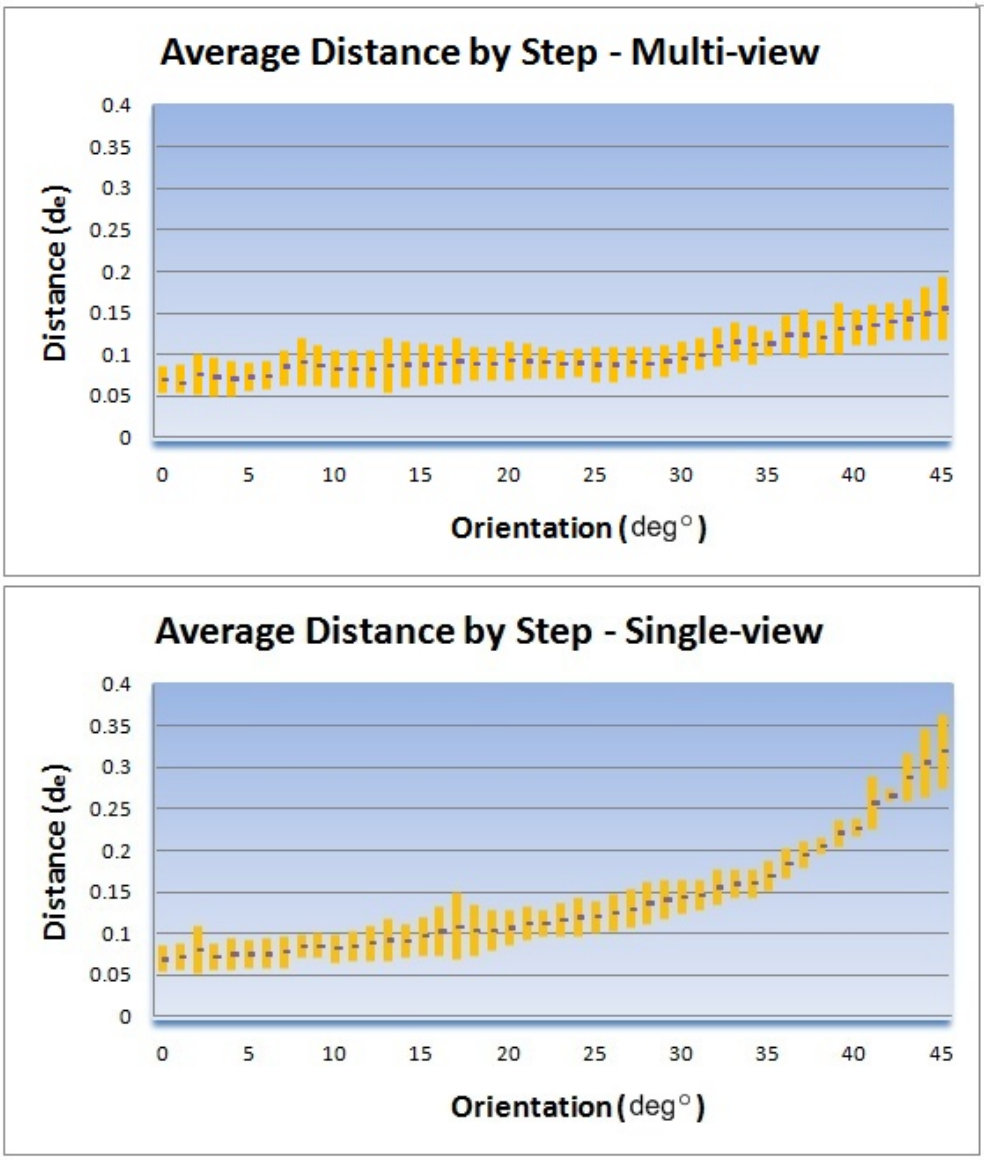



*Figure 5.7: The figures contain the average fitting errors for each step (degree). The upper figure is for the Multi-view algorithm and the lower is for the Single-view algorithm.*

| Time ($ms$) | Multi-view CLM | Single-view CLM |
|---|---|---|
| mean | 445.4 | 457.5 |
| variance | 46.3 | 65.1 |

*Table 5.1: The speed comparison between the multi-view CLM algorithm and the Single-view CLM algorithm. The experiments are applied to synthetic images.*

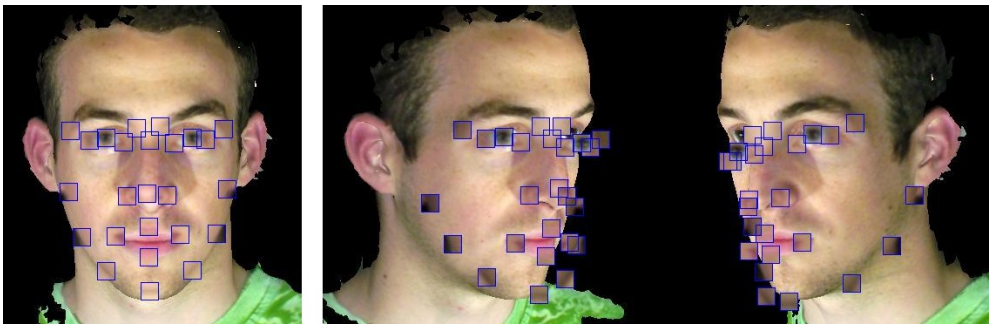


*Figure 5.8: The tracking starts from the frontal view - the image on the left. The fitting gives bad results (images on the right) when the head rotates to large angles from the frontal view with the Single-view CLM algorithm.*

From *Figure* 5.9, it can be seen that the multi-view 3D CLM algorithm significantly improves the fitting for the face contour features than the single-view 2D algorithm. The proposed algorithm improves mostly the performance on the face contour. The right eye area is somehow hidden during rotation. The performance in this area is improved because of the fixed visibility model applied to the multiple appearance models.

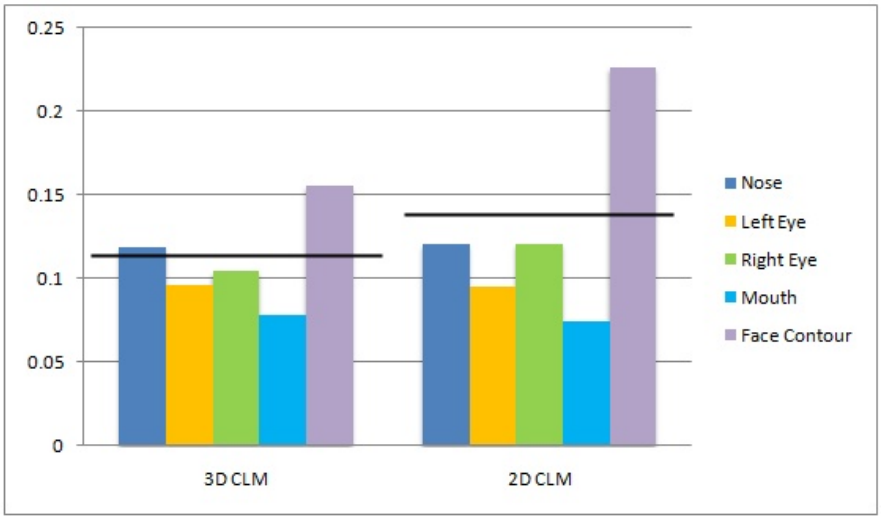



*Figure 5.9: The average distances between the tracked feature points and the corrected feature points of the faces divided by parts. The horizontal line at each group is the average distance of all the face feature points.*

## 5.3 Robustness Experiment

This experiment is carried out to investigate the accuracy and stability of the proposed algorithm for real data. The estimation methods are applied to video clips of 4 subjects showing expression, speech and some head rotation (1208 frames in total) (*Figure* 5.2) to test the localisation accuracy and stability. These images and subjects are independent of the training sets.

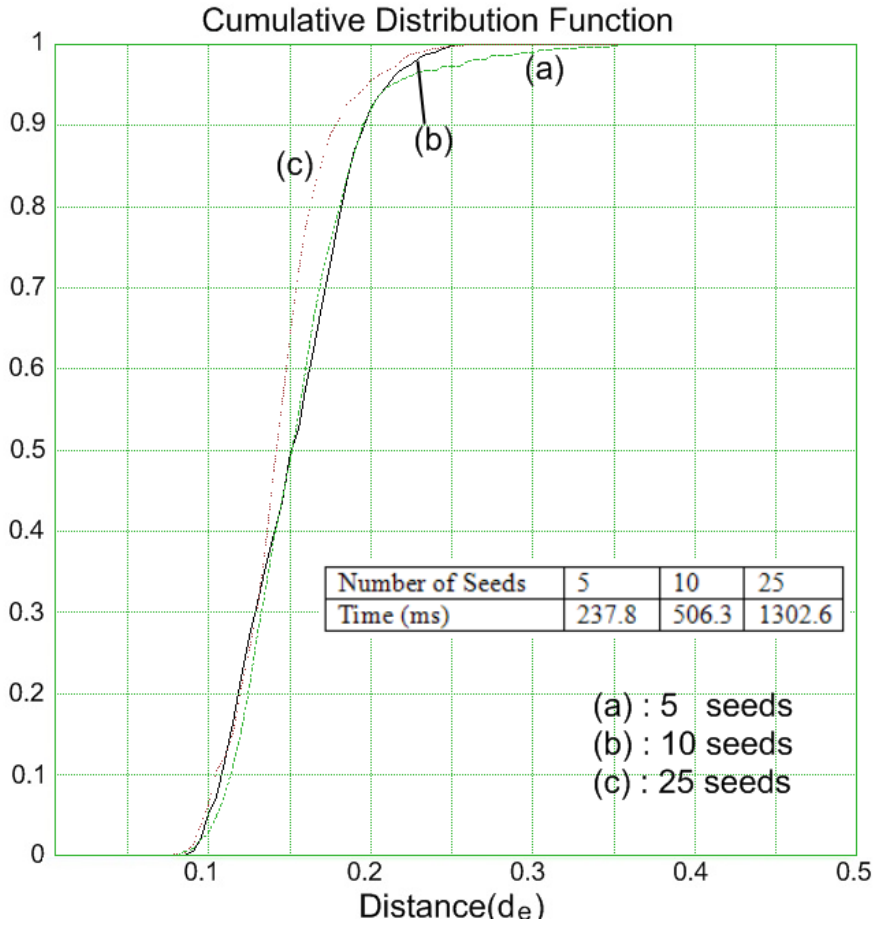



*Figure 5.10: The Condensation algorithm is applied to a set of real face images with three different number of seed settings. The fitting errors using 5 seeds and 10 seeds are close, but it takes twice much time using 10 seeds than 5 seeds. The fitting errors are smaller using 25 seeds.*

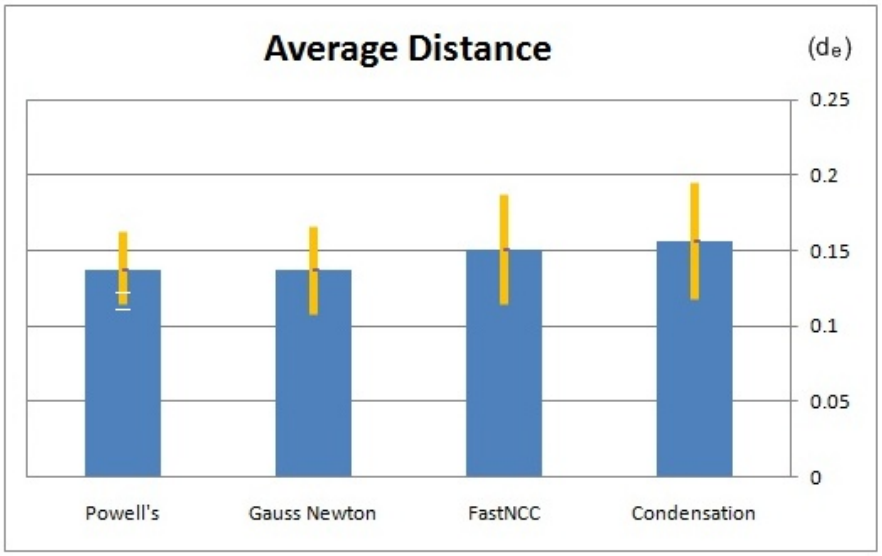



*Figure 5.11: Average errors and variances of fitting.*

The image sequences are roughly initialised with the face detector described in [152], then a CLM search (*Figure* 3.15) is applied to the first frame and the following frames while tracking. Four optimising methods are used for the experiments. For Powell's method [146], FastNCC [14] and the Gauss-Newton iterative method [6; 9], a maximum of 4 iterations is taken. For the Condensation based tracking method [15] 25 seeds are used while generating new samples from the current sample. Comparison experiments of the selection of the number of seeds for the Condensation algorithm are carried out and the results can be seen in *Figure* 5.10. A two different resolution search is used for this experiment.

For Powell's method, Gauss-Newton algorithm and Condensation algorithm with 25 seeds, nearly 80 % to 90 % of the points are within 0.18 of the eye separation, with 70 % to 80 % for FastNCC algorithm. This can be seen in *Figure* 5.12. However, the performance of the algorithms are not statistically significantly different as shown in *Figure* 5.11.

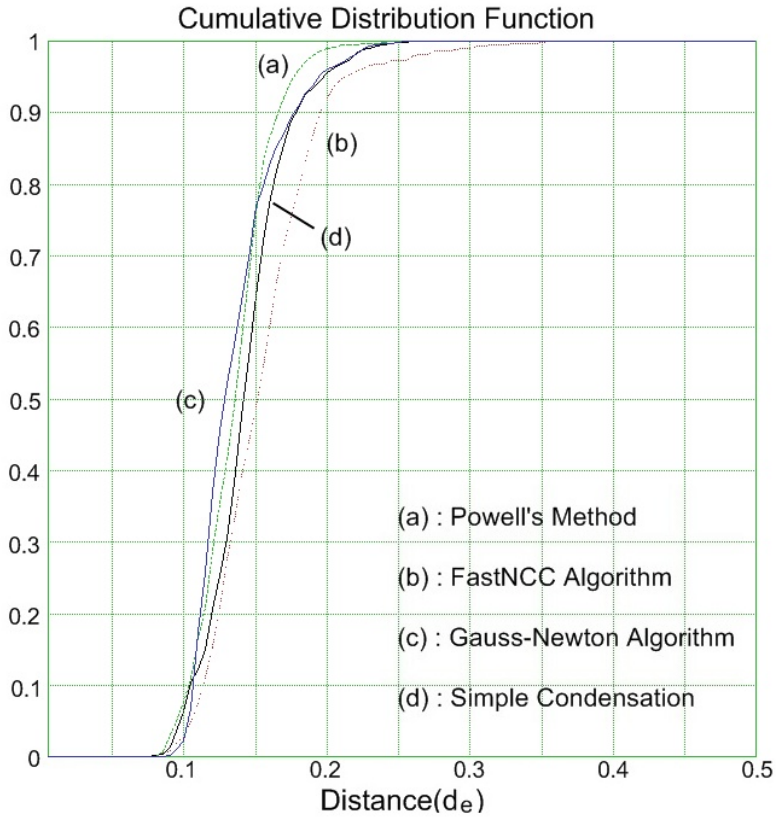



*Figure 5.12: Fitting results of four optimising algorithms on sets of real images.*

The errors are measured separately for each of the facial parts in the previous section. The results are shown in *Figure* 5.13, the performance is better around the eyes and nose area. The feature points in the mouth area and the face contour have not converged as well as the others.

The fitting speed is used to give a rough indication of the relative performance of the algorithms, but all the algorithms could be further optimised. The results are shown in *Figure* 5.14. The difference between Gauss Newton algorithm and the FastNCC algorithm are not significant, and they perform best. The Condensation algorithm performs worse than those two methods. Powell's method takes significant longer than the other three methods.

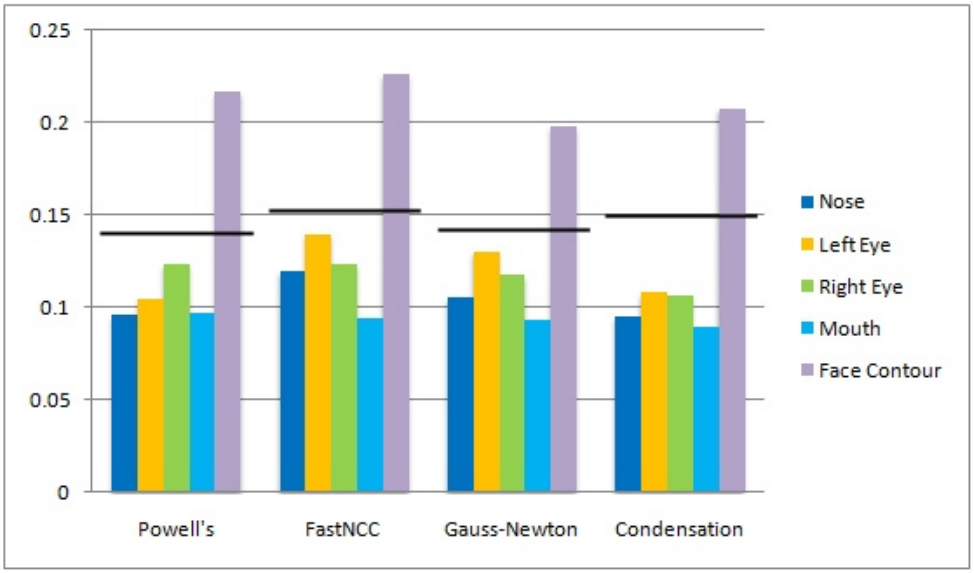



*Figure 5.13: The average distances between the tracked feature points and the corrected feature points of the faces divided by parts. The horizontal line at each group is the average distance of all the face feature points.*

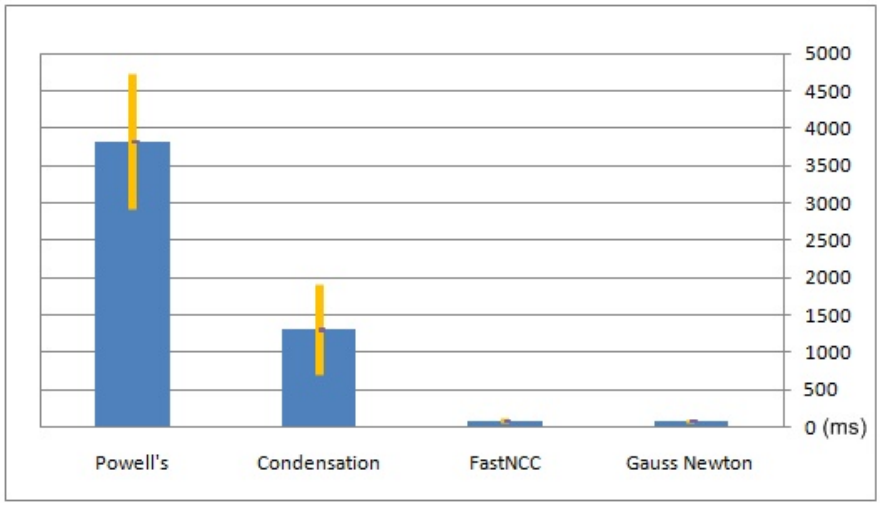



*Figure 5.14: Average fitting speed.*

## 5.4 Factors Influencing the Performance of the System

There are several factors that influence the performance of the system such as the size of the feature patches, the estimation of background and hidden points, the scale settings, etc. To investigate the influence of these factors, several experiments are carried out and described in the following sections. During these experiments, FastNCC algorithm [14] is chosen to estimate the fitting process as this demonstrated a good balance between fitting speed and accuracy in the earlier experiments.

### 5.4.1 Choosing the Feature Patch Sizes

Another issue to solve while building the model is the selection of the feature patch size. With bigger patches, the searching process will be more accurate with the sacrifice of the speed. The experiments are implemented to help give a balance between the two factors and find a proper size for the patches.

The average eye distance in the training pool is close to 104 pixels. For the experiments, 20x20, 16x16 and 12x12 pixel image patches are tested. The effect of image resolution is also tested, as the same size patch will cover a larger face area in a lower resolution image. For each patch size, a model from a reduced resolution version of the training images is built. The shrink rate is 75% which means the average eye distance is around 78 pixels. They are declared as 20x20b, 16x16b and 12x12b and the original ones as 20x20a, 16x16a and 12x12a.

To describe the performance for different sized patches, the Patch Ratio,

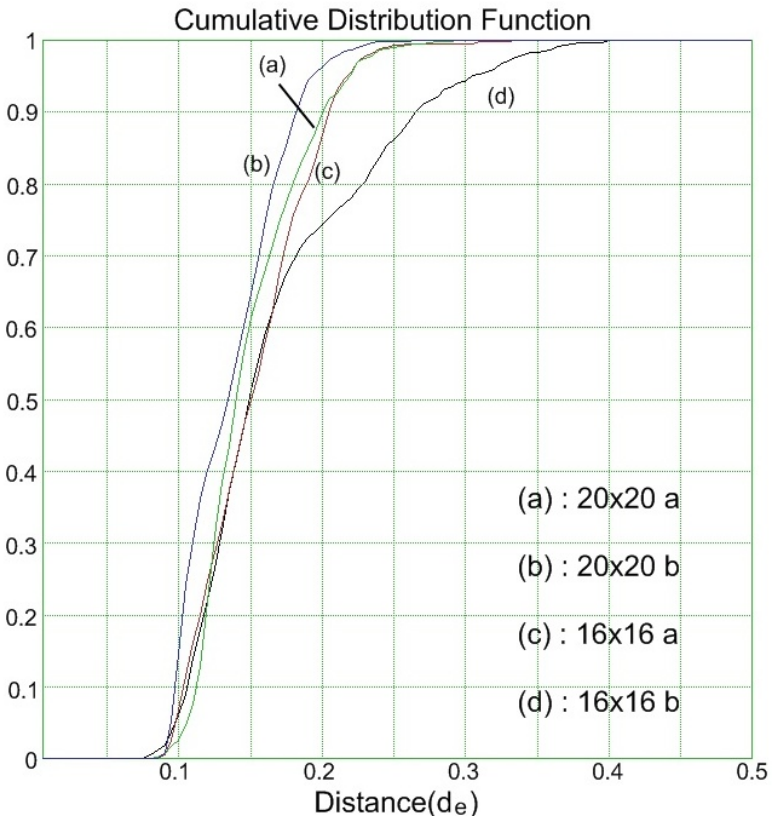



*Figure 5.15: Comparison results of the model with different patch sizes. This is from the sequence of real video images.*

$R$, is defined with the following equation:

$$R = \frac{d_p}{d_e} \tag{5.2}$$

where $d_p$ is the size of the square patches and $d_e$ is the eye distance described in *Equation* 5.1.

FastNCC algorithm [14] is applied as the optimising method and two scales are used for the fitting. In the experiments, the model with the patch size of 12x12 frequently failed to converge during the fitting. The convergence rates are too poor to compute the errors. So only 20x20 and 16x16 pixel patches are compared. Referring to the results in *Figure* 5.15 and *Figure* 5.16,

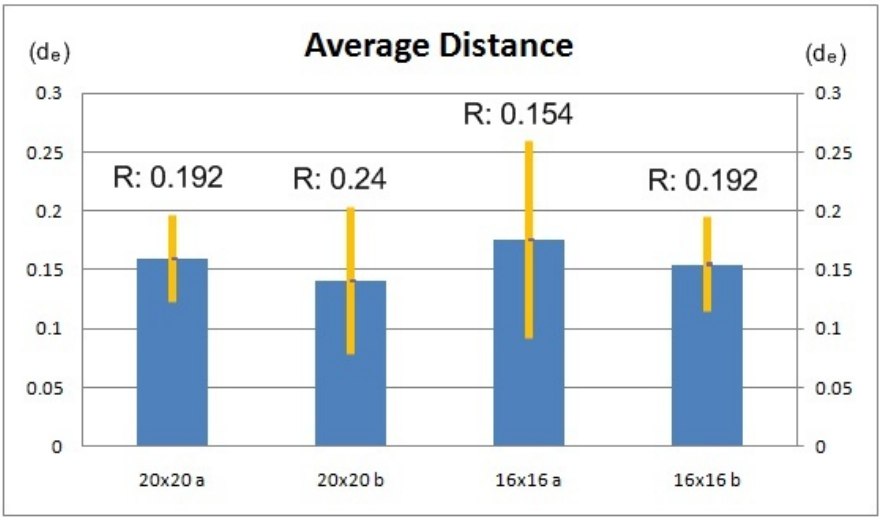



*Figure 5.16: The fitting error with different patch settings applied. The experiments are applied to the real images. Patch ratio (R) equals to the size of the patch divided by the average eye distance referring to Equation 5.2.*

20x20 patches with higher patch ratio, $R$, give the best results over the other three which has the highest patch ratio. The 20x20 patches perform better than the 16x16 patches. However, the patch ratio is an important factor for the performance of the algorithm. 20x20a patch and 16x16b patch has the same patch ratio. Although the size of the patch is different, they give nearly the same results.

As shown in *Figure* 5.17, the system with larger patch size takes more time for the fitting. Although it takes more time for the larger patch setting, it gives the better fitting results.

### 5.4.2 Reducing the Influence of Background Pixels

The stencil and depth occlusion data for the patch images available from the training data can be used to reduce the influence of the background pixels and occluded points. The following experiments are designed to compare

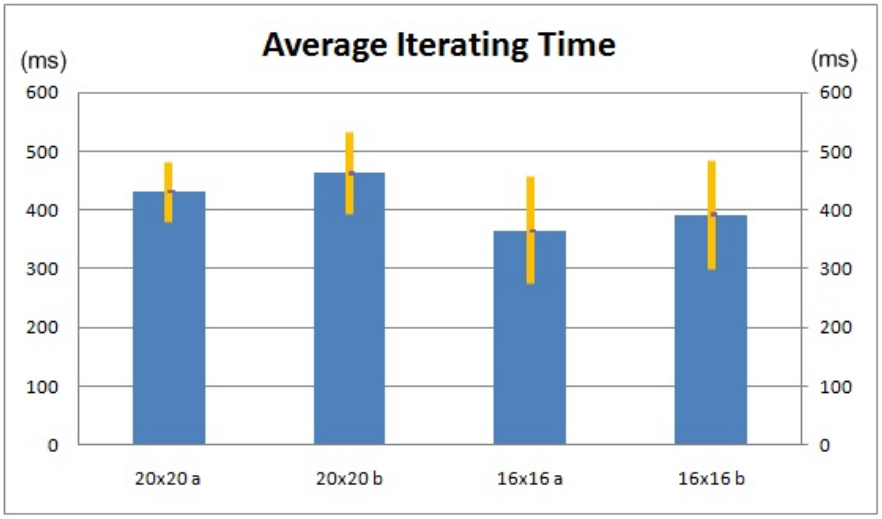



*Figure 5.17: The fitting speed with different patch settings applied. The experiments are applied to the realistic video images.*

the performance between the system with and without the use of visibility/ background information. In this section, we investigate the labelling of background pixels using data extracted from the stencil buffer of the training data. In the following, this information is referred to as the alpha channel as it is modelled by labelling pixels in the patches with an opacity value. Opacity is usually referred to as the alpha channel in computer graphics.

The algorithm with and without alpha channel is applied to the same set of face sequences comprising 1208 images. The results are shown in *Figure* 5.18. The accuracy is improved by applying the alpha channel feature to the model. The processing time is $430.9 \pm 51.2ms$ with alpha feature applied and $489.3 \pm 87.2ms$ without. The speed of the fitting process is increased by 8.43% with the extra alpha channel step applied. Although an extra step is included in the process, the efficiency is improved with the alpha feature. It takes fewer steps to converge.

In addition the system becomes more stable with the alpha feature. In the

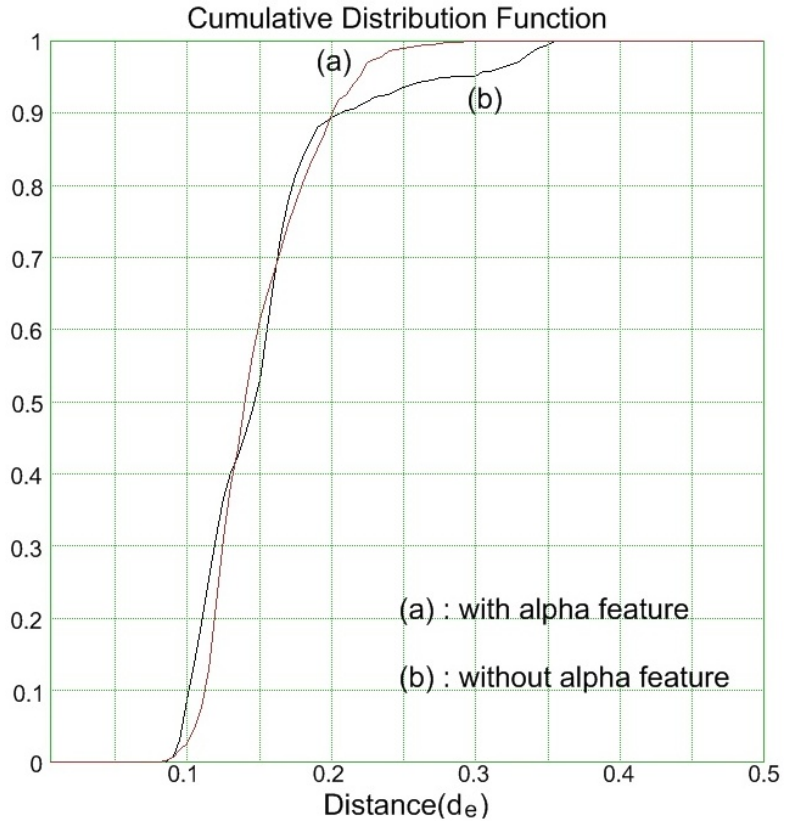



*Figure 5.18: The fitting error comparison between the system with and without alpha channel feature applied. The experiments are applied to synthetic images.*

experiments, the convergence rate is higher with the alpha feature. For the testing images, the searching converges well for 1067 images without alpha feature, and for 1208 images using the alpha feature. 141 more images failed to converge during the fitting without the alpha feature enabled. There is an example of bad convergence in *Figure* 5.19.

### 5.4.3 Reducing the Influence of Hidden Points

Facial features can be occluded under different head poses by other parts of the face. When building the model, points are labelled as hidden when their

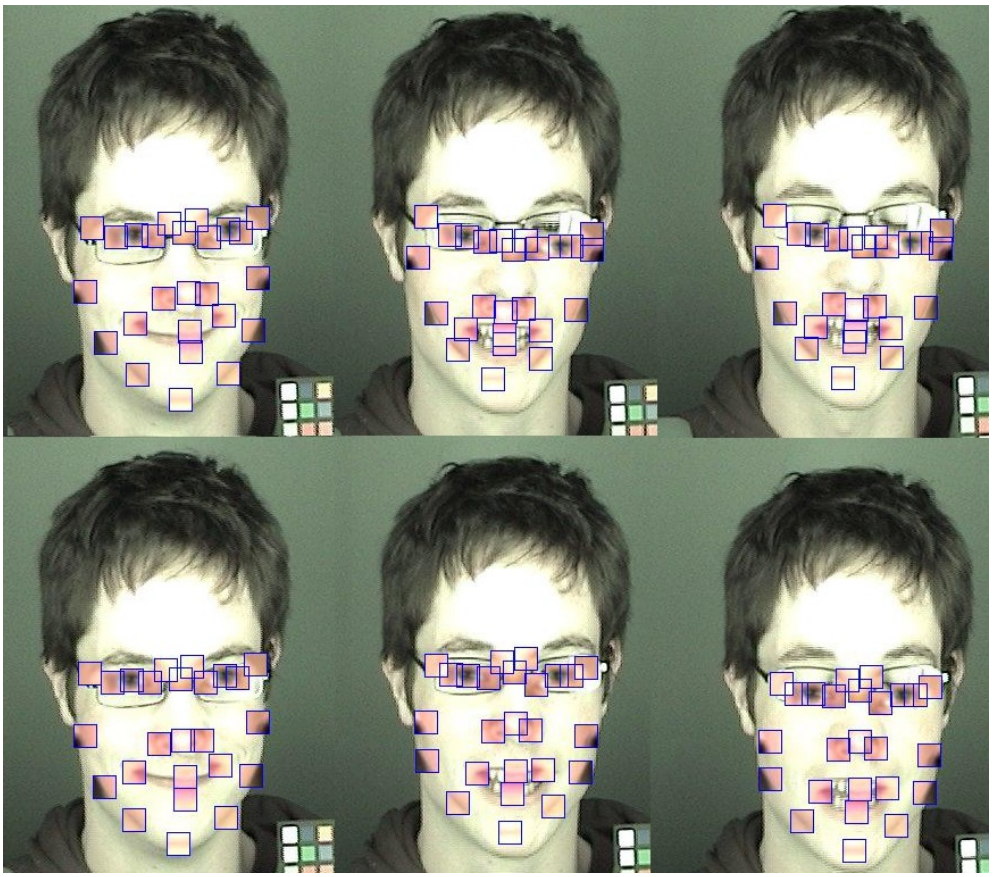


*Figure 5.19: The tracking is tested on the same set of image sequences. The images on the bottom are the results using the background occlusion. The images on the top are the results without.*

depth is greater than the value in the depth buffer for each view to reduce the negative effects of the detection and tracking. The following experiments are designed to compare the performance between the system with and without the hidden features excluded.

A set of face image sequences with fixed expression and head rotation over 40° generated from 3D models captured using the 3DMD system is used for the test. There are 726 images used in the experiments. As we can see in *Table* 5.2, the average error is smaller with the hidden feature exclusion

enabled, and it gives more robust performance with better convergence rate. It converges faster as well because fewer features are used in the fitting process (*Table* 5.3).

| Distance ($d_e$) | without hidden points | with hidden points |
|---|---|---|
| mean | 0.1380 | 0.1520 |
| variances | 0.0320 | 0.0484 |

*Table 5.2: The fitting error comparison between the system with and without hidden feature removal. The experiments are applied to synthetic images.*

| Time ($ms$) | without hidden points | with hidden points |
|---|---|---|
| mean | 445.4 | 489.0 |
| variances | 46.3 | 51.4 |

*Table 5.3: The speed comparison between the system with and without hidden feature removal. The experiments are applied to synthetic images.*

### 5.4.4 Choosing the Model Scaling

Multi-resolution methods are a standard technique for improving the optimisation speed and basin of convergence in computer vision problems. Coarser scale images contain less feature information but can reduce the noise by smoothing the high frequencies. The searching range is shortened in this level, which can help improve the performance. Conversely, the use of only low-resolution images can reduce the alignment accuracy available using high-resolution features.

To investigate the effects to the system with different scale settings, the multi-view CLM algorithm is applied to a set of face image sequences including 1208 images. FastNCC algorithm is chosen as the optimising

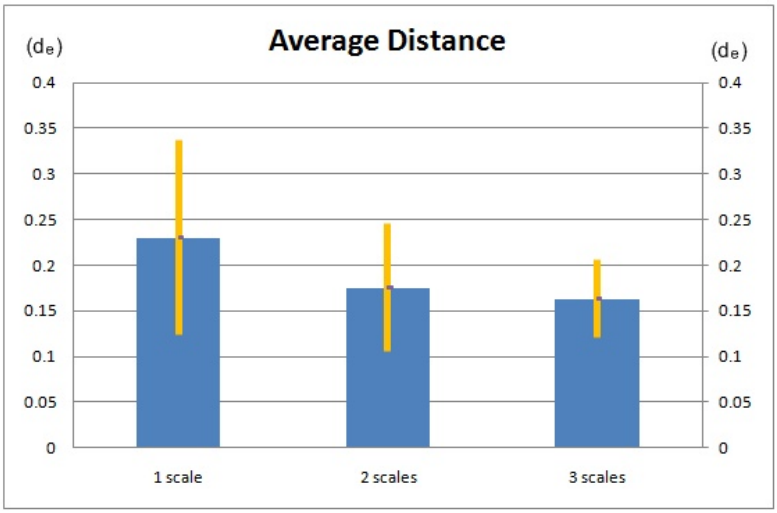



*Figure 5.20: The error comparison among the systems with different scale settings. The experiments are applied to the real image data.*

method. In the experiments, a fixed number of iterations is used for all three systems.

Three experiments are implemented to compare the performance of the system using one scale, two scales and three scales. The results are shown in *Figures* 5.20, 5.21 and 5.22. The system with one scale gives poor fitting results. Better results are obtained with more scales applied to the system and there is a significant improvement of choosing two scales over only one scale. As for speed, the system with one scale performs best and the system with three scales performs worst.

## 5.5 Discussion

In this chapter, the performance of the proposed multi-view 3D CLM algorithm was investigated using a set of experiments to test 3D CLM vs 2D CLM, the choice of the optimisation algorithm, the effects of hidden

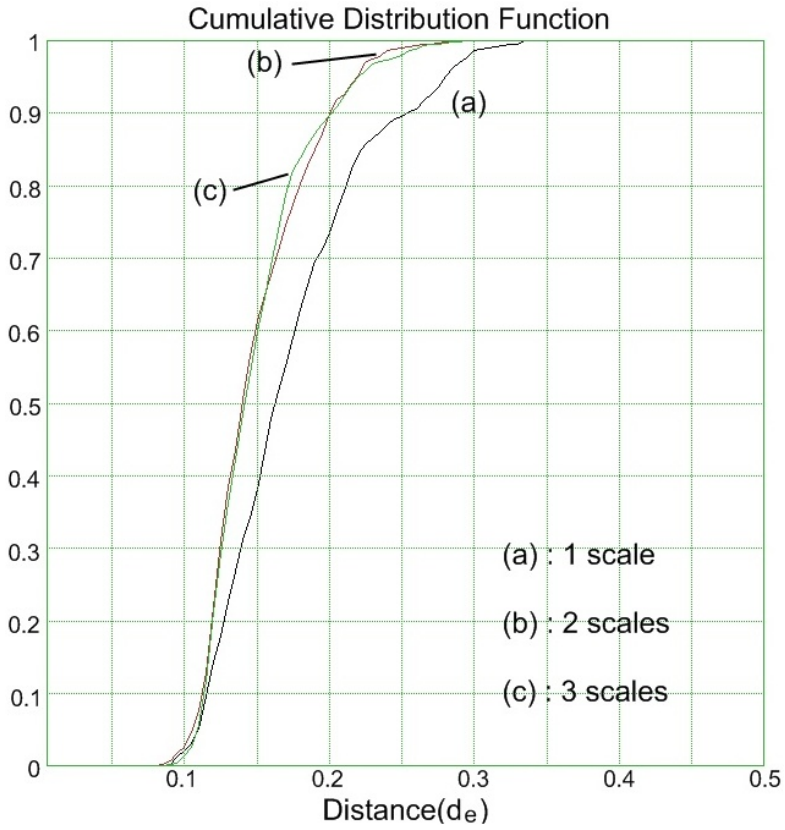



*Figure 5.21: The error comparison among the systems with different scale settings. The experiments are applied to the real image data.*

features and the effects of multi-scale.

First, the experiments showed that the proposed multi-view 3D CLM algorithm gives better results fitting to unseen images with large head rotations (the images are rendered from 3D models). The fitting is more robust (*Figures* 5.6, 5.9) and efficient (*Table* 5.1) than the single-view 2D CLM algorithm [1; 2]. The effects of choice of the optimisation algorithm were marginal for the data tested here, but further experimentation with a wider range of video data would be needed to confirm this. These experiments also identified a suitable range for a single-view CLM algorithm while tracking rotating faces (*Figure* 5.7). This explains the choice of fifteen texture models

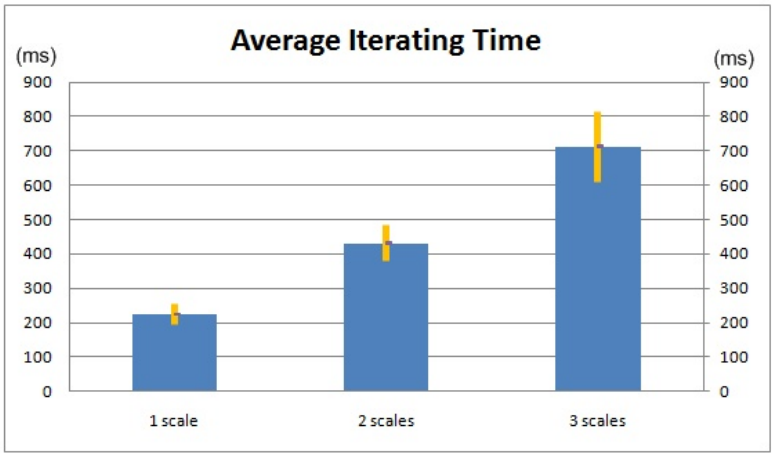



*Figure 5.22: The speed comparison among the systems with different scale settings. The experiments are applied to the real images.*

for the final system.

Next some important factors in determining the algorithm's behaviour were investigated. This helps to understand the effects of several additional parameters of the algorithm and to find a good balance to improve performance.

To find a balance between the robustness and speed, several different sized patches were studied. The best performance was found to be with the patch ratio, $R$ (*Equation* 5.2) of 0.24 (*Figure* 5.16). The 20x20b patches gave the most promising performance over the others in the experiments.

To reduce the effects of background pixels, an alpha channel feature was introduced to the system, which uses information from the stencil buffer when rendering the 3D training set. With this feature enabled, the fitting becomes more robust and takes less time to converge (*Figure* 5.18). The error is improved slightly as well. The alpha channel feature improves the performance of the algorithm while fitting to previously unseen real images.

The hidden points feature is introduced to reduce the negative effects

from the self-occluded facial features while tracking. From the results shown in *Tables* 5.2, 5.3, it can be seen that the system with the hidden points feature gives better results in both robustness and speed.

Multi scale is a general method for image processing problems. In the experiments described here (*Figures* 5.20, 5.21, 5.22), an extra scale (two scales) for the algorithm gave significant improvements to the fitting. The system with three scales gives a slight further improvement in performance, but suffers from a more significant cost in time.

Chapter 6

# Conclusions and Future Work

Facial feature detection and tracking have a long history in computer vision. AAMs [3; 4] are one of the most widely used and researched methods over recent years. Although they were originally designed for fitting 2D images, they have been extended to 3D forms. In addition, some patch based methods have been derived from AAMs such as the CLM work extended here. The presented multi-view 3D CLM algorithm is derived from the Cristinacce *et al.* 's constrained local model algorithm [1; 2]. The original algorithm combines a 2D shape model and a texture model which shows some limitation when tracking head movements with large rotations (*Figure* 5.7). The proposed multi-view 3D CLM algorithm extends the original algorithm by using a global 3D shape model and fifteen texture models built from different views of faces. During the fitting, a texture model is selected, and the shape model is projected to 2D. Patches grabbed around the current estimate of the feature point locations are projected into the appearance model space. Then a search is used to estimate updates to the model's shape and pose parameters.

This thesis uses separate models instead of combined models like the original CLM [1; 2] does. In that way, the face feature location is retained

during the variation and switching of the texture model. A Principal Components Analysis is applied to the shape model and each texture model to reduce the number of parameters. Multiple resolution techniques are used to make the model fitting more efficient and effective. The alpha channel feature is used to exclude the effects of background pixels when fitting the model to the faces. For different views, different features are chosen based on their likely visibility to improve the performance.

Based on the experiments carried out, the proposed multi-view 3D algorithm gives better results when fitting to unseen images with large head rotations. The fitting is more robust (*Figures* 5.6, 5.7, 5.9) and efficient (*Table* 5.1) than the single-view CLM algorithm [2].

The multiple resolution technique is applied to the fitting algorithm. It is a general method for solving image processing problems. The effects with different scaling settings are studied for the proposed algorithm. This feature improves the performance of the fitting according to the experiments (*Figures* 5.20, 5.21, 5.22).

To build the model, the synthetic three-dimensional face templates gathered by a 3DMD system are used as the training data. The single-view 2D CLM algorithm has improved the fitting to unseen images over AAMs. Based on the experiments carried out, the multi-view 3D CLM built from the synthetic images can fit unseen face images well (*Figure* 5.12). The effects of choice of the optimisation algorithm were marginal for the data tested here, but further experimentation with a wider range of video data would be needed to confirm this.

To understand the algorithm better and find a good balance between speed and accuracy, more experiments have been carried out to investigate some important factors in determining the algorithm's behaviour.

The synthetic images for training (grabbed from the OpenGL canvas) have a stencil channel, labelling points that have been drawn to. This can be used to exclude the effects of (estimated) background pixels when fitting to real images. Results showed that the fitting becomes more robust and takes less time to converge with the alpha features referring to *Figure* 5.18.

Another feature introduced in the thesis is the hidden points feature. It reduces the negative effects from the blocked facial feature while tracking. Referring to the results shown in *Table* 5.2 and *Table* 5.3, the fitting becomes more robust and fast with the hidden points feature applied.

### Future Work

Currently, separate shape and texture models are used in the proposed algorithm. Many works [2; 4] etc. have shown that combined shape and texture models give efficient and improved performance. It is reasonable to suppose that using combined models could improve the performance of the fitting for the multi-view 3D CLM algorithm as well. Future work will include training the models into several combined models for each view.

There are some recently proposed methods which outperform Cristinacce and Cootes' original Constrained Local Model (CLM) algorithm [1; 2] including Wang *et al.* 's Exhaustive Local Search (ELS) CLM algorithm [94], generic Convex Quadratic Fitting (CQF) CLM approach [95] and Paquet *et al.* 's Bayesian Constrained Local Model (BCLM). These methods plus the inverse compositional method and their extensions could be extended to solve 3D problems and adapted to improve the performance of the proposed multi-view 3D CLM algorithm.

The results (*Figures* 5.9, 5.13) showed that the face contour part is not matched as well as internal facial features. Improved background modelling

and contour alignment methods should be investigated to improve facial contour alignment and tracking.

Pizarro *et al.* [60] pointed out that combining the Light-Invariant theory with AAMs can fit AAMs to face images efficiently for which the lighting conditions are uncontrolled. Future research could involve lighting and colour and more self occlusion factors, which could improve the matching rates under variant lighting, colour conditions and even with unexpected occlusions.

Facial feature tracking has been studied for years. The proposed multi-view 3D CLM algorithm benefits from the advantages of the original CLM algorithm and deformable multi-pose techniques. This means the algorithm can track 3D faces in video more effectively. Large head movements and expression variance can be involved in the tracking process. The algorithm can also be further studied and adapted to face recognition, pose estimation and face morphing etc.

Appendix A

# Standard Techniques

## A.1 Shape Model

The shape model is built from normalised shape coordinates, as illustrated in Figure 3.2. The three-dimensional positions of the vertices of the features, $(x, y, z)$, are then concatenated to form a shape vector,

$$s = (x_1, y_1, z_1, x_2, y_2, z_2, \cdots, x_n, y_n, z_n)^T \tag{A.1}$$

where n is the number of vertex feature points in the face model.

To calculate the principal components, the templates are aligned to the average template (in a standard position) by performing a rigid body and scale transform. The covariance matrix of the vectorised points (templates) is created using the formula.

$$C = \frac{1}{n-1} \sum_{i=0,j=0}^{m,n} (s_i - \overline{s})(s_j - \overline{s}) \tag{A.2}$$

The model is obtained by applying Jacobi's method [146] to the covariance matrix to find the eigenvectors and eigenvalue, which represent the principal components and their distributions. Then the PCA model is built with the sorted eigenvalue.

$$s = \bar{s} + P_s \cdot b_s \tag{A.3}$$

where vector, $s$, describes the facial shape and $b$ represents a set of shape parameters, which are linearly independent. $\bar{s}$ is the mean shape vector, $P_s$ is a set of orthogonal modes of variation computed from the covariance matrix, $b_s$ is a set of shape parameters.

### A.1.1 Parameters

The three-dimensional shape is modelled by projecting the model onto a two dimensional surface via a linear transform, describing as the view and perspective transform. The view matrix combines a three dimensional rotation and translation to map the shape vectors $s$ from a feature centered position to one relative to the view point.

$$V = T \cdot S \cdot R_\theta \cdot R_\phi \cdot R_\gamma. \tag{A.4}$$

where

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{pmatrix} \tag{A.5}$$

$$S = \begin{pmatrix} S & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.6}$$

$$R_\theta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.7}$$

$$R_\phi = \begin{pmatrix} \cos(\phi) & 0 & -\sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.8}$$

$$R_\gamma = \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.9}$$

The angles $\theta$, $\phi$ and $\gamma$ are rotations around the first feature which is the center point between the eyes. The concatenated parameters $T_x, T_y, T_z, S, \theta, \phi, \gamma$ are denoted as $T_t$.

### A.1.2 Project Points from 3D to 2D using Frustum

The shape model is three-dimensional and the computer monitor is a two-dimensional surface. (Figure A.1) It is needed to transform the three-dimensional scene into the two-dimensional scene in order to display and extract the textures at the right coordinates. The following perspective matrix produces a perspective projection. This matrix is applied to the three-dimensional points $(x_e, y_e, z_e)$ in the experiments and convert the points from the eye coordinates to the clip coordinates, $(x_c, y_c, z_c)$.

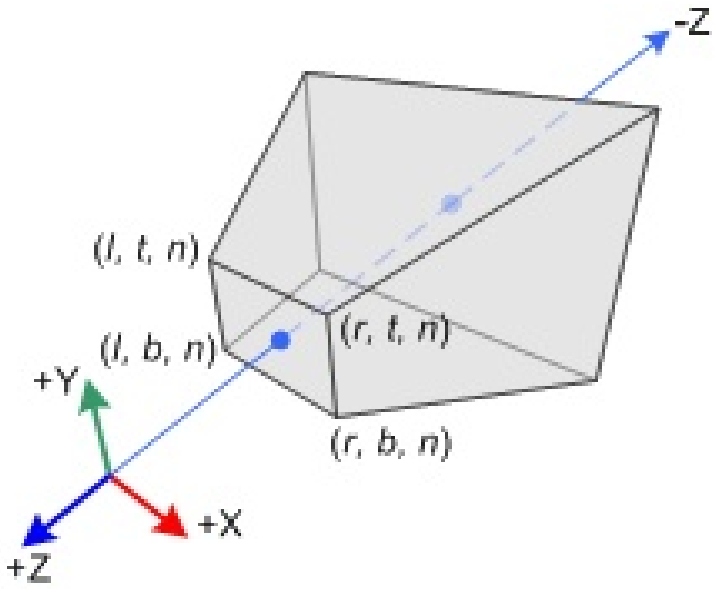



*Figure A.1: Perspective Frustum*

$$M = \begin{pmatrix} \frac{2n}{r-l} & 0 & A & 0 \\ 0 & \frac{2n}{t-b} & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{pmatrix} \qquad \begin{aligned} A &= \frac{r+l}{r-l} \\ B &= \frac{t+b}{t-b} \\ C &= -\frac{f+n}{f-n} \\ D &= -\frac{2f*n}{f-n} \end{aligned} \tag{A.10}$$

l, r specify the coordinates for the left and right vertical clipping planes.

b, t specify the coordinates for the bottom and top horizontal clipping planes.

n, f specify the distances to the near and far depth clipping planes. Both values must be positive.

The projection matrix is for general frustum. The viewing volume is

symmetric here, which is $r = -l$ and $t = -b$, then it can be simplified as,

$$\begin{cases} r + l = 0 \\ r - l = 2r \quad (width) \end{cases}, \quad \begin{cases} t + b = 0 \\ t - b = 2t \quad (height) \end{cases} \tag{A.11}$$

The projected two-dimensional coordinates, $(x_n, y_n)$, then are given by the following equation.

$$\begin{aligned} x_n &= \frac{nx_e}{rz_e} \\ y_n &= \frac{ny_e}{tz_e} \end{aligned} \tag{A.12}$$

## A.2 Gaussian Pyramid

To reduce the size of an image, we cannot just sub-sample the image by taking, for example, every second pixel in every second line. If we did so, we would disregard the *sampling theorem*. [153] The combined smoothing and size reduction can be expressed in a single operator by using a symmetric Gaussian kernel. If we repeat the smoothing and sub-sampling operations iteratively, we obtain a series of images, which is called the Gaussian pyramid. From level to level, the resolution decreases by a factor of two; the size of the images decreases correspondingly. Consequently, we can think of the series of images as being arranged in the form of a pyramid as illustrated in Figure A.2.

These pyramids are most convenient if the image dimensions are a power of two, or a multiple of a power of two. The smallest image is the most heavily smoothed; the layers are often referred to as coarse scale versions of the image. In the implementation of the proposed system, the original image is square, with image dimensions $2^k$. The operator $S^{\downarrow}$ down-samples an image; in particular, the $j, k$'th element of $S^{\downarrow}(\mathcal{I})$ is the $2j$, $2k$'th element of $\mathcal{I}$. The $n$'th level of a pyramid $P(\mathcal{I})$ is denoted $P(\mathcal{I})_n$.

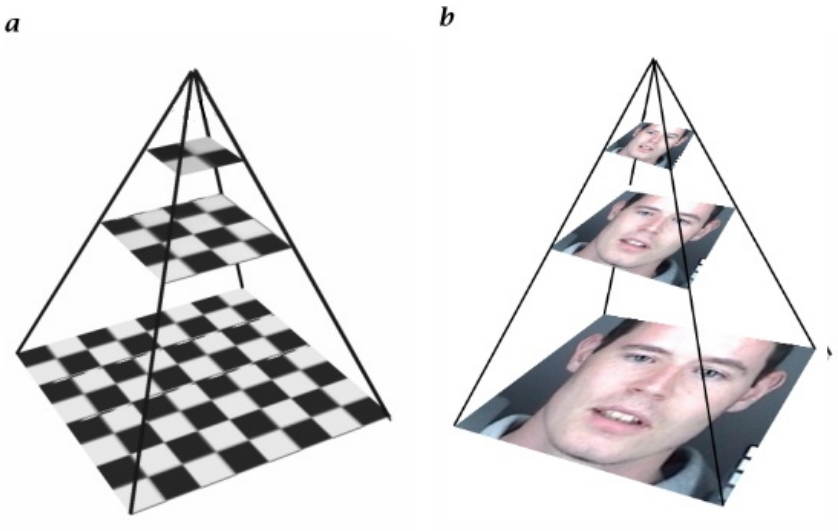



*Figure A.2: Gaussian pyramid: a) schematic representation, the squares of the checkerboard corresponding to pixels; b) example.*

We can now write simple expressions for the layers of a Gaussian pyramid:

$$\begin{aligned} P_{Gaussian}(\mathcal{I})_{n+1} &= S^{\downarrow}(G_\sigma * * P_{Gaussian}(\mathcal{I})_n) \\ &= (S^{\downarrow}G_\sigma)P_{Gaussian}(\mathcal{I})_n) \end{aligned} \tag{A.13}$$

where we have written $G_\sigma$ for the linear operator that takes an image to the convolution of that image with a Gaussian. The finest scale layer is the original image.

$$P_{Gaussian}(\mathcal{I})_1 = \mathcal{I}$$

The pyramid brings large scales into the range of local neighborhood operations with small kernels. Moreover, these operations are performed efficiently. Once the pyramid has been computed, we can perform neighborhood operations on large scales in the upper levels of the pyramid – because of the smaller image sizes – much more efficiently than for finer scales.

The application in this case is the spatial search, a common theme in computer vision. Searching for a match in the original pairs of images is

Forming a Gaussian pyramid

Set the finest scale layer to the image

For each layer

Obtain this layer by smoothing the next finest layer with a Gaussian, and then sub-sampling it

End

inefficient, because we may have to wade through a great deal of detail. The approach using a Gaussian pyramid is to look for a match in the top level of the pyramid (heavily smoothed and re-sampled image), and then refine that match by looking at increasingly detailed versions of the image. For example, in the proposed implementation, 20x20 feature patch images are reduced down to 5x5 versions, match those and then look at 10x10 versions, etc. match those and finally look at 20x20 versions.

## A.3 Gauss-Newton Iterative Method

Let $I(x)$ stand for the $n^{th}$ image in a given video sequence, where $x = (x, y)^T$ are the pixel coordinates and $n = 0, 1, 2, \cdots$ is the frame number. The coordinates, $x$ are synthesised with *Equation* A.3 in *Chapter* 3. The extracted patch image, $g(x; b_s)$, is sampled from the coordinate frame of the image $I_n(x)$. The synthetic patch image, $f(x)$, is synthesised with *Equation* 3.5 and *Equation* 3.4 in *Chapter* 3.

The sum of the squared errors is given by:

$$E(x) = \sum_x (g(x; b_s) - f(x))^2, \tag{A.14}$$

The minimisation of the SSD function is performed with respect to the shape parameters $b_s$. Due to its non-linear nature, optimisation is done

iteratively solving for increments $\Delta b_s$ to the already known parameters $b_s$.

$$E(x) = \sum_x [g(x; b_s + \Delta b_s) - f(x)]^2. \tag{A.15}$$

The more efficient algorithm, where the roles of the synthetic image and the extracted image are switched. In this case, the following expression is to be iteratively minimised,

$$E(x) = \sum_x [f(x; \Delta b_s) - g(x; b_s)]^2. \tag{A.16}$$

Performing a first order Taylor expansion on *Equation* A.16 gives:

$$E(x) = \sum_x \left[ f(x; 0) + \nabla f \frac{\partial f}{\partial b_s} \Delta b_s - g(x; b_s) \right]^2. \tag{A.17}$$

Minimising *Equation* A.17 is a least-square problem. One solution is obtained by taking the partial derivative of *Equation* A.17 and then setting it to equal zero. The solution obtained is:

$$\Delta b_s = H^{-1} \sum_x \left[ \nabla f \frac{\partial f}{\partial b_s} \right]^T [g(x; b_s) - f(x)], \tag{A.18}$$

where $H$ is the Gauss-Newton approximation [154] to the Hessian matrix:

The parameters of the warp, $b_s$ are found by solving *Equation* A.18 and additively updating $b_s$ by,

$$b_s \leftarrow b_s + \Delta b_s. \tag{A.19}$$

$$H = \sum_x \left[ \nabla f \frac{\partial f}{\partial b_s} \right] \left[ \nabla f \frac{\partial f}{\partial b_s} \right], \tag{A.20}$$

which does not depend on the warping parameters and thus can be pre-computed. The Jacobian $\dfrac{\partial f}{\partial b_s}$ is evaluated at $(x; 0)$.

## A.4 FastNCC

Let $g(x,y) = G(x,y) - \overline{G(x,y)}$, be the image fitting to and $f(x,y) = F(x,y) - \overline{F(x,y)}$, be the template image synthesised from the model and $g(x,y)$. The normalised cross correlation between two images $F(x,y)$ and $G(x,y)$, can be written as:

$$E = \frac{\sum_x f(x,y) \cdot g(x,y)}{\sqrt{\sum_x f^2(x,y) \cdot g^2(x,y)}} \tag{A.21}$$

Performing a first order Taylor series expansion to approximate the function $f(x,y;b_s + \Delta b_s)$, where bs are the shape and pose parameters of the model and $\Delta b_s$ are the (small) updates to the parameters:

$$f(x;b_s + \Delta b_s) \approx f(x;b_s) + \sum_i \Delta b_s \frac{\partial f}{\partial b_s} \tag{A.22}$$

To minimise $E$, the derivatives with respect to each $b_s$ is estimated, and the results are set to zero. The derivative of the NCC function is,

$$\frac{\sum_j f(x;\Delta b_s)F - \dfrac{\sum_j \big(f(x;b_s) + \sum_i \dfrac{\partial f}{\partial b_s} f(x;\Delta b_s)\big)\big(f(x;\Delta b_s) + g(x;b_s)\big)}{F}}{F\sqrt{g^2(x;b_s)}}, \tag{A.23}$$

where

$$F = F(x;b_s) = \sqrt{\sum_k \big(f(x;b_s) + \sum_i \frac{\partial f}{\partial b_s} f(x;\Delta b_s)\big)^2}. \tag{A.24}$$

Equating the derivatives to zero at the maxi-ma gives a set of non-linear equations in the unknown update parameters $\Delta b_s$. Dividing any two of these equations gives a linear equation of the form:

$$\frac{\sum_j f(x;\Delta b_s) g(x;b_s)}{\sum_k f(x;\Delta b_s) g(x;b_s)} = \frac{\sum_j f\big(x;\Delta b_s\big)\big(f(x;b_s) + \sum_i \dfrac{\partial f}{\partial b_s} f(x;\Delta b_s)\big)}{\sum_k f\big(x;\Delta b_s\big)\big(f(x;b_s) + \sum_i \dfrac{\partial f}{\partial b_s} f(x;\Delta b_s)\big)} \tag{A.25}$$

only $N-1$ of these equations will be linearly independent, so $k = j+1$ is used with $j$ from 0 to $N-2$. To solve for all the parameters, $\Delta b_{s0}$ is chosen as the unknown variable and let variable $\delta$, $\delta_0$ replace variable $\Delta b_s$, $\Delta b_{s0}$. Then the $N-1$ linear equations are rearranged into the form:

$$A \cdot \delta = b \cdot \delta_0 + c \tag{A.26}$$

where $A$ are the coefficients of the unknown $\delta_i$ with $i$ from 1 to $N-1$, $\delta$ is the column vector of the unknown $\delta_i$ with $i$ from 1 to $N-1$, $b$ is the column vector of coefficients of $\delta_0$ and $c$ is the column vector of constants.

$$\begin{aligned} A &= [a_{ij}]_{i,j\in[0,N-2]} \\ &= \Big[g \cdot df_i \cdot H_{i+1,j+1} - g \cdot df_{i+1} \cdot H_{i,j+1}\Big] \\ b &= [b_i]_{i\in[0,N-2]} \\ &= \Big[g \cdot df_{i+1} \cdot H_{0,i} - g \cdot df_i \cdot H_{0,i+1}\Big] \\ c &= [c_i]_{i\in[0,N-2]} \\ &= \Big[g \cdot df_{i+1} \cdot f \cdot df_i - g \cdot df_i \cdot f \cdot df_{i+1}\Big] \end{aligned} \tag{A.27}$$

where,

$$\begin{aligned} g &= g(x, y; b_s) \\ f &= f(x, y; b_s) \end{aligned} \tag{A.28}$$

$$df_i = \nabla f \frac{\partial f}{\partial b_{si}} \tag{A.29}$$

$$H_{i,j} = df_i \cdot df_j \tag{A.30}$$

# References


[1] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008.

[2] D. Cristinacce and T. Cootes, "Feature detection and tracking with constrained local models," in *British Machine Vision Conference*, vol. 3, pp. 929–938, 2006.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *European Conference on Computer Vision*, vol. 2, pp. 484–498, 1998.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 681–685, 2001.

[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[6] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions*

*on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1025–1039, 1998.

[7] H. Y. Shum and R. Szeliski, *Panoramic vision: sensors, theory, and applications*, ch. Construction of panoramic image mosaics with global and local alignment, pp. 227–268. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[8] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *IEEE IEEE Transactions on Computer Vision and Pattern Recognition*, pp. 1090–1097, 2001.

[9] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, pp. 135–164, 2004.

[10] T. F. Cootes and P. Kittipanyangam, "Comparing variations on the active appearance model algorithm," in *British Machine Vision Conference*, vol. 2, pp. 837–846, 2002.

[11] J. Paterson and A. Fitzgibbon, "3d head tracking using non-linear optimization," in *British Machine Vision Conference*, 2003.

[12] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast active appearance model search using canonical correlation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1690–1694, 2006.

[13] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[14] B. P. Tiddeman and J. Chen, "Correlated active appearance models," in *IEEE Transactions on Signal-Image Technology & Internet-Based Systems*, pp. 832–838, 2007.

[15] M. Isard and A. Blake, "Condensation - c conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[16] 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, pp. 38–59, 1995.

[17] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2d+3d active appearance models," in *the IEEE computer society conference on computer vision and pattern recognition*, vol. 2, pp. 535–542, 2004.

[18] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Computer graphics, annual conference series (SIG-GRAPH)*, pp. 187–194, 1999.

[19] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1061–1075, 1992.

[20] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.

[21] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[22] M. Covell, "Eigen-points: Control-point location using principal component analysis," in *International Conference on Automatic Face and Gesture Recognition*, pp. 122–127, 1996.

[23] M. J. Black and Y. Yacoob, "Recognizing facial expressions under rigid and non-rigid facial motions," in *International Workshop on Automatic Face and Gesture Recognition*, pp. 12–17, 1995.

[24] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham, "Training models of shape from sets of examples.," in *British Machine Vision Conference*, pp. 9–18, 1992.

[25] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," in *International Conference on Information Processing in Medical Imaging*, (London, UK), pp. 33–47, Springer-Verlag, 1994.

[26] T. F. Cootes and C. J. Taylor, "Combining point distribution models with shape models based on finite element analysis," *Image and Vision Computing*, vol. 13, pp. 419–428, 1995.

[27] P. Karaolani, G. D. Sullivan, K. D. Baker, and M. J. Baines, "A finite element method for deformable models," in *Alvey Vision Conference*, pp. 73–78, 1989.

[28] A. P. Pentland, "Automatic extraction of deformable part models," *International Journal Computer Vision*, vol. 4, pp. 107–126, 1990.

[29] A. P. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 715–729, 1991.

[30] T. F. Cootes and C. J. Taylor, "Active shape models - smart snakes," in *British Machine Vision Conference*, pp. 266–275, Springer-Verlag, 1992.

[31] T. Cootes and C. J. Taylor, "Active shape model search using local grey-level models: A quantitative evaluation," in *British Machine Vision Conference*, pp. 639–648, BMVA Press, 1993.

[32] A. Lanitis, C. J. Taylor, T. F. Cootes, and T. Ahmed, "Automatic interpretation of human faces and hand gestures using flexible models," in *International Workshop on Automatic Face- and Gesture-Recognition*, pp. 98–103, 1995.

[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, Inc., 2 ed., 2000.

[34] S. Romdhani, S. Gong, A. Psarrou, and R. Psarrou, "A multi-view nonlinear active shape model using kernel pca," in *British Machine Vision Conference*, pp. 483–492, BMVA Press, 1999.

[35] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computing*, vol. 10, no. 5, pp. 1299–1319, 1998.

[36] G. Hamarneh and T. Gustavsson, "Deformable spatio-temporal shape models: Extending asm to 2d+time," in *British Machine Vision Conference*, pp. 13–22, 2001.

[37] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic interpretation and codeing of face images using flexible models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(7), pp. 742–756, 1997.

[38] T. Mcinerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, vol. 1, pp. 91–108, 1996.

[39] S. Sclaroff and J. Isidoro, "Active blobs," in *The International Conference on Computer Vision*, pp. 1146–1153, 1998.

[40] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," in *Pattern Analysis and Machine Intelligence*, vol. 19(7), pp. 733–742, 1997.

[41] M. J. Jones and T. Poggio, "Multidimensional morphable models: A framework for representing and matching object classes," in *International Journal of Computer Vision*, vol. 29, pp. 107–131, Springer Netherlands, 1998.

[42] R. Gross, I. Matthews, and S. Baker, "Active appearance models with occlusion," *Image and Vision Computing*, vol. 24(6), pp. 593–604, 2006.

[43] F. Dornaika and J. Ahlberg, "Face model adaptation for tracking and active appearance model training," in *British Machine Vision Conference*, 2003.

[44] J. Ahlberg, "An active model for facial feature tracking," *EURASIP Journal on Applied Signal processing*, vol. 2002, pp. 566–571, 2001.

[45] S. Sclaroff and J. Isidoro, "Active blobs: region-based, deformable appearance models," *Computer Vision and Image Understanding*, vol. 89(2/3), pp. 197–225, 2003.

[46] J. Ahlberg, "Using the active appearance algorithm for face and facial feature tracking," in *International Conference on Computer Vision Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-time Systems*, pp. 68–72, 2001.

[47] M. B. Stegmann, "Object tracking using active appearance models,"

in *Danish Conference Pattern Recognition and Image Analysis*, vol. 1, pp. 54–60, 2001.

[48] F. Dornaika and A. D. Sappa, "Rigid and non-rigid face motion tracking by aligning texture maps and stereo 3d models," *Pattern Recognition Letter*, vol. 28, no. 15, pp. 2116–2126, 2007.

[49] N. Faggian, A. Paplinski, and T. Chin, "Face recognition from video using active appearance model segmentation," in *International Conference on Pattern Recognition*, vol. 1, pp. 287–290, 2006.

[50] G. J. Edwards, T. F. Cootes, C. J. Taylor, H. Burkhardt, and B. Neumann, "Face recognition using active appearance models," in *Europe Conference of Computer Vision*, pp. 581–591, 1998.

[51] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for medical image analysis and computer vision," in *SPIE Medical Imaging*, pp. 236–248, 2001.

[52] S. C. Mitchell, B. P. F. Lelieveldt, R. J. Geest, J. G. Bosch, J. H. C. Reiber, and M. Sonka, "Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac mr images," in *IEEE Transanctions on Medical Image*, vol. 20, pp. 415–423, 2001.

[53] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "3-d active appearance models: Segmentation of cardiac mr and ultrasound images," in *IEEE Transactions on Medical Imaging*, pp. 1167–1178, 2002.

[54] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Comparing active

shape models with active appearance models," in *British Machine Vision Conference*, pp. 173–182, 1999.

[55] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng, "Direct appearance models," in *Computer Vision and Pattern Recognition*, pp. 828–833, 2001.

[56] S. Yan and Q. Cheng, "Multi-view face alignment using direct appearance models," in *IEEE International Conference on Automatic Face and Gesture Recognition*, p. 324, 2002.

[57] S. Yan, C. Liu, S. Z. Li, H. Zhang, H. Shum, and Q. Cheng, "Face alignment using texture-constrained active shape models," *Image and Vision Computing*, vol. 21, pp. 69–75, 2003.

[58] T. Sim and T. Kanade, "Combining models and exemplars for face recognition: An illuminating example," in *Computer Vision and Pattern Recognition Workshop on Models versus Exemplars in Computer Vision*, December 2001.

[59] X. Liu, P. Tu, and F. Wheeler, "Face model fitting on low resolution images," in *British Machine Vision Conference*, vol. 3, pp. 1079–1088, 2006.

[60] D. Pizarro, J. Peyras, and A. Bartoli, "Light-invariant fitting of active appearance models," in *IEEE Conference on Computer Vision and Patter Recognition*, pp. 1–6, 2008.

[61] G. D. Finlayson, S. D. Hordley, L. Cheng, and M. S. Drew, "On the removal of shadows from images," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 59–68, 2006.

[62] T. F. Cootes and C. J. Taylor, "Constrained active appearance

models," in *International Conference on Computer Vision*, pp. 748–754, 2001.

[63] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework: Part 1," Tech. Rep. CMU-RI-TR-02-16, Robotics Institute, University of Carnegie Mellon, Pittsburgh, PA, July 2002.

[64] S. Baker, R. Gross, and I. Matthews, "Lucas-kanade 20 years on: A unifying framework: Part 3," tech. rep., Robotics Institute, University of Carnegie Mellon, Pittsburgh, PA, November 2003.

[65] S. Baker, R. Gross, and I. Matthews, "Lucas-kanade 20 years on: A unifying framework: Part 4," *International Journal of Computer Vision*, vol. 56, pp. 221–255, 2004.

[66] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23(11), pp. 1080–1093, November 2005.

[67] S. Romdhani and T. Vetter, "Efficient, robust and accurate fitting of a 3d morphable model," in *International Conference on Computer Vision*, pp. 59–66, 2003.

[68] A. U. Batur and M. H. Hayes, "Adaptive active appearance models," in *IEEE Transactions on Image Processing*, vol. 4, pp. 1707–1721, 2005.

[69] A. U. Batur and M. H. Hayes, "A novel convergence scheme for active appearance models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 359–466, 2003.

[70] G. Papandreou and P. Maragos, "Adaptive and constrained algorithms for inverse compositional active appearance model fitting," in *IEEE*

*Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[71] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. Frade, and J. Cohn, "Aam derived face representations for robust facial action recognition," in *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 155–160, 2006.

[72] J. Peyras, A. Bartoli, H. Mercier, and P. Dalle, "Segmented aams improve person-independent face fitting," in *British Machine Vision Conference*, 2007.

[73] J. Peyras, A. J. Bartoli, and S. K. Khoualed, "Pools of aams: Towards automatically fitting any face image," in *British Machine Vision Conference*, 2008.

[74] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Computer Vision: Issues, Problems, Principles, and Paradigms*, pp. 726–740, 1987.

[75] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[76] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2 ed., 2008.

[77] G. J. McLachlan and T. Krishnan, *Finite Mixture Models*. John Wiley & Sons, 2000.

[78] T. F. Cootes and C. J. Taylor, "A mixture model for representing shape

variation," in *Image and Vision Computing*, pp. 110–119, BMVA Press, 1997.

[79] J. M. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting with a mixture of local experts," in *International Conference on Computer Vision*, August 2009.

[80] Y. Zhou, W. Zhang, X. Tang, and H. Shum, "A bayesian mixture model for multi-view face alignment," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Washington, DC, USA), pp. 741–746, IEEE Computer Society, 2005.

[81] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1621–1633, 1997.

[82] Y. Weiss, *Bayesian motion estimation and segmentation*. PhD thesis, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, 1998.

[83] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," *Advanced lectures on machine learning*, pp. 118–183, 2003.

[84] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory*, (London, UK), pp. 23–37, Springer-Verlag, 1995.

[85] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, 1996.

[86] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, pp. 337–374, 2000.

[87] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of Machine Learning Research*, vol. 4, pp. 933–969, 2003.

[88] X. Liu, "Generic face alignment using boosted appearance model," in *Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[89] H. Wu, X. Liu, and G. Doretto, "Face alignment via boosted ranking model," in *Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[90] D. Cristinacce and T. F. Cootes, "Boosted regression active shape models," in *British Machine Vision Conference*, vol. 2, pp. 880–889, 2007.

[91] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 90–95, Springer, 2001.

[92] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database," in *International Conference on Audio and Video-based Biometric Person Authentication*, pp. 72–77, 1999.

[93] D. Cristinacce, T. F. Cootes, and I. Scott, "A multi-stage approach to facial feature," in *British Machine Vision Conference*, pp. 231–240, 2004.

[94] Y. Wang, S. Lucey, J. Cohn, and J. M. Saragih, "Non-rigid face tracking with local appearance consistency constraint," in *IEEE International*

*Conference on Automatic Face and Gesture Recognition*, September 2007.

[95] Y. Wang, S. Lucey, and J. F. Cohn, "Enforcing convexity for improved alignment with constrained local models," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. Issue 23–28, pp. 1–8, June 2008.

[96] U. Paquet, "Convexity and bayesian constrained local models," *IEEE Transactions on Computer Vision and Pattern Recognition*, pp. 1193–1199, 2009.

[97] L. Liang, F. Wen, Y. Xu, X. Tang, and H. Y. Shum, "Accurate face alignment using shape constrained markov network," in *Computer Vision and Pattern Recognition*, pp. 1313–1319, 2006.

[98] L. Gu and T. Kanade, "A generative shape regularization model for robust face alignment," in *European Conference on Computer Vision*, pp. 413–426, 2008.

[99] L. Gu and T. Kanade, "3d alignment of face in a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1305–1312, 2006.

[100] R. Gross, I. M. S. Baker, and T. Kanade, "The cmu multi-ple pose, illumination and expression (multipie) database," tech. rep., Robotics Institute, Carnegie Mellon University, 2007.

[101] A. B. Ashraf, S. Lucey, T. Chen, K. Prkachin, P. Solomon, Z. Ambadar, , and J. Cohn, "The painful face: Pain expression recognition using active appearance models," in *ACM International Conference on Multimodal Interfaces*, pp. 9–14, 2007.

[102] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," tech. rep., University of Massachusetts, 2007.

[103] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *CVPR*, vol. 1, pp. 511–518, 2001.

[104] T. F. Cootes, K. Walker, and C. Taylor, "View-based active appearance models," in *IEEE International Conference on Automatic Face and Gesture Recognition*, (Washington, DC, USA), pp. 227–232, IEEE Computer Society, 2000.

[105] N. Faggian, S. Romdhani, J. Sherrah, and A. Paplinski, "Color active appearance model analysis using a 3d morphable model," in *Digital Image Computing on Techniques and Applications*, (Washington, DC, USA), p. 59, IEEE Computer Society, 2005.

[106] B. P. Tiddeman and D. I. Perrett, "Moving facial image transformations using static 2d prototypes," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 260–266, 2001.

[107] S. Romdhani, V. Blanz, and T. Vetter, "Face identification by fitting a 3d morphable model using linear shape and texture error functions," in *European Conference on Computer Vision*, pp. 3–19, 2002.

[108] M. Brand, "Morphable 3d models from video," in *IEEE computer society conference on computer vision and pattern recognition*, vol. 2, pp. 456–463, 2001.

[109] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, and Y. Shan, "Robust and rapid generation of animated faces from video images: A model-based modeling approach," *International Journal of Computer Vision*, vol. 58, no. 2, pp. 93–119, 2004.

[110] C. Hu, J. Xiao, I. Matthews, S. Baker, J. Cohn, and T. Kanade, "Fitting a single active appearance model simultaneously to multiple images," in *British Machine Vision Conference*, September 2004.

[111] S. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J. Cohn, and T. Kanade, "Multi-view aam fitting and camera calibration," in *IEEE International Conference on Computer Vision*, (Washington, DC, USA), pp. 511–518, IEEE Computer Society, 2005.

[112] K. Ramnath, S. Koterba, J. Xiao, C. B. Hu, I. Matthews, S. Baker, J. F. Cohn, and T. Kanade, "Multi-view aam fitting and construction," in *International Journal of Computer Vision*, vol. 76, pp. 183–204, 2008.

[113] M. Dimitrijevic, S. Ilic, and P. Fua, "Accurate face models from uncalibrated and ill-lit video sequences," in *IEEE computer society conference on computer vision and pattern recognition*, vol. II, pp. 1034–1041, 2004.

[114] J. Sung and D. Kim, "Extension of aam with 3d shape model for facial shape tracking," in *IEEE International Conference on Image Processing*, pp. 3363–3366, 2004.

[115] Z. Wen and T. S. Huang, "Capturing subtle facial motions in 3d face tracking," in *International Conference on Computer Vision*, vol. 2, pp. 1343–1350, 2003.

[116] F. H. Pighin, R. Szeliski, and D. Salesin, "Resynthesizing facial animation through 3d model-based tracking," in *International Conference on Computer Vision*, pp. 143–150, 1999.

[117] F. Dornaika and J. Ahlberg, "Fast and reliable active appearance model search for 3d face tracking," in *IEEE transactions on systems, man and cybernetics*, vol. 34, pp. 1838–1853, 2004.

[118] T. F. Cootes, G. Wheeler, K. Walker, and C. Taylor, "Coupled-view active appearance models," in *the British machine vision conference*, vol. 1, pp. 52–61, 2000.

[119] R. Gross, I. Matthews, and S. Baker, "Appearance-based face recognition and light-fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26(4), pp. 449–465, 2004.

[120] I. Matthews, J. Xiao, and S. Baker, "On the dimensionality of deformable face models," tech. rep., Robotics Institute, University of Carnegie Mellon, Pittsburgh, PA, March 2006.

[121] I. Matthews, J. Xiao, and S. Baker, "2d vs. 3d deformable face models: Representational power, construction, and real-time fitting," in *International Journal of Computer Vision*, vol. 75, pp. 93–113, 2007.

[122] M. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, 2002.

[123] M. Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 322–336, 1999.

[124] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Learning to identify and track faces in image sequences," in *International Conference on Computer Vision*, (Washington, DC, USA), p. 317, IEEE Computer Society, 1998.

[125] S. Gokturk, J. Bouguet, and R. Grzeszczuk, "A data driven model for monocular face tracking," in *International Conference on Computer Vision*, pp. 701–708, 2001.

[126] J. Ström, "Model-based real-time head tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2002, pp. 1039–1052, 2002.

[127] F. Dornaika and J. Ahlberg, "Efficient active appearance model for real-time head and facial feature tracking," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, (Washington, DC, USA), p. 173, IEEE Computer Society, 2003.

[128] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 232–237, 1998.

[129] Y. Li, S. Gong, and H. Liddell, "Modelling faces dynamically across views and over time," in *IEEE International Conference on Computer Vision*, pp. 554–559, 2001.

[130] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 602–605, 1993.

[131] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, 1994.

[132] S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," in *International Conference on Pattern Recognition*, pp. 611–616, 1996.

[133] T. Maurer and C. Malsburg, "Tracking and learning graphs and pose on image sequences of faces," in *International Conference on Automatic Face and Gesture Recognition*, (Washington, DC, USA), p. 76, IEEE Computer Society, 1996.

[134] S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, and D. Banin, "Tracking facial feature points with gabor wavelets and shape models," in *International Conference on Audio- and Video-Based Biometric Person Authentication*, (London, UK), pp. 35–42, Springer-Verlag, 1997.

[135] T. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking," 1999.

[136] C. Y. Seong, B. D. Kang, J. H. Kim, and S. K. Kim, "Effective detector and kalman filter based robust face tracking system," in *Advances in Image and Video Technology*, pp. 453–462, 2006.

[137] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME – Journal of Basic Engineering*, pp. 35–45, 1960.

[138] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Speech Recognition*, pp. 267–296, 1989.

[139] R. D. Rimey and C. M. Brown, "Selective attention as sequential

behavior: Modeling eye movements with an augmented hidden markov model," in *DaRPA Image Understanding Workshop*, pp. 840–849, 1990.

[140] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *European Conference on Computer Vision*, pp. 343–356, 1996.

[141] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *International Conference on Computer Vision*, pp. 107–112, 1998.

[142] S. Gong and M. Walter, "Recognition of temporal structures: Learning prior and propagating observation augmented densities via hidden markov states," in *IEEE International Conference on Computer Vision*, pp. 157–162, 1999.

[143] E. Ong and S. Gong, "Tracking hybrid 2d-3d human models from multiple views," in *International Workshop on Modeling People at International Conference of Computer Vision*, pp. 11–18, 1999.

[144] H. Sidenbladh, F. Torre, and M. J. Black, "A framework for modeling the appearance of 3d articulated figures," in *IEEE International Conference on Automatic Face and Gesture Recognition*, p. 368, 2000.

[145] Y. Chen and F. Davoine, "Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically," in *British Machine Vision Conference*, vol. 2, pp. 609–618, 2006.

[146] W. H. Press., S. A. Teukolsky., W. T. Vetterling, and B. P. Flannery, *Numerical recipes - The art of scientific computing.* Cambridge University Press, 2007.

[147] J. C. Russ, *The Image Processing Handbook.* CRC, 4 ed., 2002.

[148] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," 1984.

[149] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[150] M. J. Black and A. D. Jepson, "Recognizing temporal trajectories using the condensation algorithm," in *International Conference on Face & Gesture Recognition*, (Washington, DC, USA), p. 16, IEEE Computer Society, 1998.

[151] W. Hopkins, *A New View of Statistics*. http://newstatsi.org, 2000.

[152] J. Chen and B. P. Tiddeman, "Multi-cue facial feature detection and tracking," in *International Conference on Image and Signal Processing*, pp. 356–367, 2008.

[153] B. Jähne, *Digital Image Processing*. Springer, 5 ed., 2002.

[154] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.