Animation of a Hierarchical Image Based Facial Model and Perceptual Analysis of Visual Speech

Darren Cosker

Cardiff School of Computer Science
Cardiff University
June 2005

A thesis submitted in partial fullfillment of the requirement for the degree of Doctor of Philosophy.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

In this Thesis a hierarchical image-based 2D talking head model is presented, together with robust automatic and semi-automatic animation techniques, and a novel perceptual method for evaluating visual-speech based on the McGurk effect.

The novelty of the hierarchical facial model stems from the fact that sub-facial areas are modelled individually. To produce a facial animation, animations for a set of chosen facial areas are first produced, either by key-framing sub-facial parameter values, or using a continuous input speech signal, and then combined into a full facial output.

Modelling hierarchically has several attractive qualities. It isolates variation in sub-facial regions from the rest of the face, and therefore provides a high degree of control over different facial parts along with meaningful image based animation parameters.

The automatic synthesis of animations may be achieved using speech not originally included in the training set. The model is also able to automatically animate pauses, hesitations and non-verbal (or non-speech related) sounds and actions. To automatically produce visual-speech, two novel analysis and synthesis methods are proposed. The first method utilises a Speech-Appearance Model (SAM), and the second uses a Hidden Markov Coarticulation Model (HMCM) - based on a Hidden Markov Model (HMM).

To evaluate synthesised animations (irrespective of whether they are rendered semi automatically, or using speech), a new perceptual analysis approach based on the McGurk effect is proposed. This measure provides both an unbiased and quantitative method for evaluating talking head visual speech quality and overall perceptual realism. A combination of this new approach, along with other objective and perceptual evaluation techniques, are employed for a thorough evaluation of hierarchical model animations.

Acknowledgements

Firstly, i would like to thank my PhD supervisors, Dr Dave Marshall and Dr Paul Rosin, for their input, ideas, suggestions and patience in helping me to complete this work. It would have been an impossible task without their advice and support. I would also like to thank Dr Simon Rushton and Susan Paddock for their invaluable knowledge in perception research, and Dr Yulia Hicks for helping me to understanding the numerous techniques I had to become familiar with when I first started, and her general advice throughout.

I would also like to thank my family, friends and other loved ones for their patience and their ears, I have lots of catching up to do and I intend to do it!

This PhD is dedicated to my Grandfather.

Publications

The work contained in this thesis is based on the following articles:

- D. Cosker, D. Marshall, P. L. Rosin, S. Paddock and S. Rushton, "Towards Perceptually Realistic Talking Heads: Models, Metrics and McGurk", ACM Transactions on Applied Perception, vol. 2, no. 3, 2005.
- D. Cosker, D. Marshall, P. L. Rosin, S. Paddock and S. Rushton, "Towards Perceptually Realistic Talking Heads: Models, Metrics and McGurk", ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization, pp. 151-157, 2004.
- D. Cosker, D. Marshall, P. L. Rosin and Y.A. Hicks,"Speech Driven Facial Animation using a Hierarchical Model" , Proc. IEE Vision, Image and Signal Processing, vol. 15, no. 4, pp 314-321, 2004.
- D. Cosker, D. Marshall, P. L. Rosin and Y.A. Hicks, "Speech Driven Facial Animation using a Hidden Markov Coarticulation Model", Proc. IEEE Int. Conf. Pattern Recognition, vol. 1, pp. 128-131, 2004.
- D. Cosker, D. Marshall, P. L. Rosin and Y.A. Hicks, "Speaker-Independent Speech-Driven Facial Animation Using a Hierarchical Facial Model", Proc. of IEE Visual Information Engineering (VIE2003), pp. 169-172, 2003.
- D. Cosker, D. Marshall, P. L. Rosin and Y.A. Hicks, "Video Realistic Talking Heads using Hierarchical Non-Linear Speech-Appearance Models", Proc. of Mirage 2003, pp. 20-27, 2003.

The power of the hierarchical model as a tool for the perceptual analysis (and consequent resynthesis) of facial behaviours such as smiles is demonstrated in several psychological studies:

- Krumhuber, E., Cosker, D., Manstead, A., Marshall, D., and Rosin, P.L. (2005). Synthetic humans for the study of subtle temporal aspects in facial displays. Paper to be presented at the IXth Conference of the International Society for Research on Emotions, Bari, Italy (July 2005).

- Krumhuber, E., Cosker, D., Manstead, A., Marshall, D., and Rosin, P.L. (2005). Temporal dynamics of smiling: Human versus synthetic faces. Poster to be presented at the IXth Conference of the International Society for Research on Emotions, Bari, Italy (July 2005).

- Krumhuber, E., Cosker, D., Manstead, A., Marshall, D., and Rosin, P.L. (2005). Temporal aspects of smiles influence employment decisions: A comparison of human and synthetic faces. Paper to be presented at the 11th European Conference Facial Expressions: Measurement and Meaning, Durham, United Kingdom (September 2005).

The models constructed in this Thesis have also been used in work on blind source separation. I am coauthor on the paper:

- W.Wang, D.P.Cosker, Y.Hicks, S.Sanei, J.A.Chambers, "Video Assisted Speech Source Separation", Proc. IEEE ICASSP, March 2005, Philadelphia, USA.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer graphics, like most other branches of computer science, has matured almost in parallel with advances in computer hardware. As processor speeds get faster, computer displays increase in resolution and graphics cards grow in power we continually witness more impressive graphical feats, almost on a daily basis. One of the ambitions of graphics researchers has always been to create a computer generated human indistinguishable from a real human. A satisfactory solution to this problem is based on the success of two other issues: A means of accurately visually portraying the human and a means of convincingly controlling its movement.

Without question the most sophisticated and expressive external part of the body is a persons face. The face is consequently perhaps the most difficult part of the body to reproduce graphically and animate to a realistic standard. The major reason for this is due to our sensitivity to facial behaviour. Humans are highly sensitive to even the slightest defect in a synthetic facial animation. A facial animation indistinguishable from a real face must therefore be close to being flawless - an very difficult proposition. The problem of creating a perfect computer generated human face is often described as the *Holy Grail* of computer graphics, and is by no means an understatement. The problem, as with the rest of the body, is to create a suitable means of displaying the face and of controlling it. Arguably, both problems are equally difficult to solve, and without an adequate solution to each of them, the whole endeavor will be unsuccessful.

Given a realistic 2D or 3D facial model the next problem is to provide an accurate and realistic means of controlling or animating the face for output. There have been many approaches to

producing facial animation in academia and industry. Typically, the animation of the mouth during lip-synching to speech, especially when automatically generated, is regarded separately from animation of facial expression [88]. Common approaches employed to animate expression in faces include rule-based methods, key-framing/target-morphing, motion-capture data or expression tags. Mouth animation is then typically achieved manually using key-framing/target-morphing or motion-capture data, and automatically using phonetic and continuous speech inputs.

The idea of realistically animating a face using only speech as an input is an attractive one. Such systems are often described as 'Talking Heads', and their development has received a great amount of attention - beginning with the development of the first parametric 3D face model in the 1970's [118]. The majority of existing talking head systems concentrate only on automatic animation of lip-synch of the face to an audio track and ignore emotion. The reason for this is in part due to the methods used to animate the mouth, i.e. Phonetic based methods - which contain no information other than speech units.

1.1 Applications of Facial Animation

The global growth of the media and entertainment industries, with help from the unprecedented growth of the computer industry, has provided a large and varied amount of applications for facial animation.

1.1.1 The Movie and Computer Game Industries

Film studios are highly competitive, and film projects are becoming more and more ambitious. Computer graphics are used today in almost every new major film release. Evidence of computer graphics is obvious in some movies (and is often the movies selling point) [123, 52, 30, 126, 150] and less obvious in others [121, 122]. Computer generated facial animation has also been used extensively in many films [52, 30, 150, 126]. Figure 1.1 shows examples of recent, memorable computer generated characters. In such films the quality of the facial animation is of paramount importance, since the character needs deliver a convincing performance in place of a real actor. It is therefore not surprising to learn that the animation of a computer generated face for such films is highly expensive. Such animations are often created manually by the artists and animators, typically

using key-framing/target-morphing [150, 88]. An alternative animation method is to use motion-capture techniques to map facial locations on an actors face to drive the facial model [30, 88].

Figure 1.1: Recent computer generated characters: (clockwise from top-left) Gollum, Shrek and Sulley. Gollum was animated using a mixture of motion-capture and key-framing, while Shrek and Sulley were both animated using key-framing.

The development of a system capable of providing automatic full facial animation from speech alone, with lip-synch and facial expression, would provide a useful tool for the movie industry. We may also envisage such a system being used to re-dub the lip-synching of computer generated characters into different languages for foreign releases, with minimal resources. Given a suitably realistic model we may also imagine such a system being used to re-dub live action movies: where an actors voice is recorded for a new scene, and then used to animate a facial model which is later projected back into the scene. A realistic but somewhat limited performance driven example of such as system was recently implemented to create mock archive footage of famous (and infamous) politicians during the second world war [34].

Sharing similarities with the movie industry, the computer game industry (or, as is now becoming the fashionable term, the *video game* industry) now makes extensive use of computer generated characters employing facial animation [138]. However, unlike in film animation, real-time facial animation (i.e. not pre-rendered) is often desirable, requiring performance trade-off decisions from

developers with respect to facial model complexity and processing speed [149]. Pre-rendered in-game animations are typically lengthly movie-style cut-scenes, with high resolution graphics [88]. In these situations the quality of the facial animation is often as important to ensuring the sale of a game as it would be in selling a movie - such is the current demand from the video game market for leading edge ascetics.

1.1.2 User Interfaces: Man - Machine Interaction

Facial animations used in user interfaces can provide users with a more comfortable and less alien computing experience. Specifically, facial animations can provide tutorial and help information, or respond to certain system and application events. Facial animation in a user interface can also help the hearing impaired. This is because speech is multi-modal - we decipher it using both our ears and our eyes [109], and in noisy environments can understand speech more clearly given its corresponding facial movements [68].

Animations incorporated in such an application are also unlikely to be pre-rendered, and would likely be delivered in real-time to account for potential novel interactions between human and machine. In this situation, a facial animation model driven by a Text-To-Speech (TTS) system is perhaps the most realistic solution [9, 8, 7].

1.1.3 The Internet

Applications for talking heads on the internet share similarities with those for User Interfaces. However, due to the restricted bandwidth of the internet, talking heads also present novel solutions in other areas. We can imagine a web site having pre-recorded talking head sequences to help guide users around the vastness of the internet, or simply to point them in the right direction on one specific site. Search engines might give their results using talking heads, or a talking head might provide a site introduction - informing the user of what they are likely to find at the site. More importantly, since a talking head may be controlled using only speech, we may envisage streaming video which uses a very small amount of bandwidth. A decoder, downloaded by the client, can receive speech information across the network and subsequently produce real-time animations. This reduces network overheads associated with viewing such video, and also reduces site administrator costs, since new video can be generated from scratch without a professional animator. A news web-site would be an ideal candidate for a dynamic on-line talking head application, utilising a TTS system to translate

written news stories into on-the-fly newsreader broadcasts.

1.1.4 Communication

Sharing the benefits of the potential low-bandwidth nature of talking heads, mobile communication is another application [115]. A single image, or full speaker model, may be downloaded into a receivers handset and then animated using the speakers voice (either warped or parametrically animated). Users may also enjoy the novelty of a conversation with a famous person, as the application is not solely restricted to animating the callers face, but could animate any-ones. Video-conferencing is a similar application where talking heads might be employed, reducing the overhead of streaming real-time video. As well as adding to the immersion of mobile and video communication interactions, the presence of a talking head on a mobile device is also a benefit to the hearing impaired [135].

1.2 Animating 3D and Image Based Talking Heads

Using 3D polygonal head models for animation is the most popular direction taken in the movie, computer game and media industries. This trend is also followed in academia. The reason is in part due to the flexibility that this type of model provides in terms of animation. Using physically based (or muscle/anatomical based) 3D models or non-physical based 3D models (sometimes referred to as blend shape models), control may be achieved by linearly combining a few select facial expressions (or blend shapes), altering the parameters values of facial area articulators, manually adjusting the positions of vertices by hand, or driving the model via an actors performance (this technique is called performance driven animation).

An alternative model for use in facial animation is the 2D or 3D image based model (or Morphable Model). Image based models are trained on real speaker footage, and convey a high degree of static and video realism [77, 22, 40, 36, 11, 15], often to a level where human observers cannot distinguish between real video clips and synthetic animations [62, 15]. The drawback of image based models is that they offer limited control for animation compared to 3D models. As a consequence of this, image based animations can appear zombie like and unnatural looking since the animation of facial areas other than the mouth is ignored [62, 22, 145]. The facial models used in the motion

picture *The Matrix Revolutions* [15] are perhaps the most realistic created to date. However, animation of these models is achieved using manual time-consuming methods - i.e., no flexible means of intuitive animation control is available. In the case where 2D and 3D image based models are able to offer a greater than normal degree of animation flexibility then output realism suffers [65]. For example, many image based models do not allow the control of separate facial areas, and are therefore unsuitable for many traditional animation applications. If fine control is provided then it is typically only geometrical based - involving warping separate parts of static facial images using a 2D or 3D mesh (e.g. raise the eyebrows). Through this type of technique, adaption of the underlying facial texture, in light of geometric deformation, is usually ignored (e.g. raise eyebrow *shape* but do not add forehead creases in the *texture*). This can cause a conflict between static and behavioural realism [65].

The parameters controlling 2D image-based models are also not intuitive. For example, to animate an appearance model [36] or a morphable model [11] the user would have no idea which parameter value to alter in order to raise the eyebrows of the model. This is because in statistical image based models such as these, modes of variation describing the movement of separate facial areas are not orthogonal. For example, in an appearance model, separate parameters for controlling the eyebrows and the mouth do not exist. Instead alteration of a single parameter results in a combined motion in several different facial areas (this is discussed further in Chapter 4. The only way to avoid this problem when using statistical image based models is to collect a training set which contains examples of isolated facial variation. However, to provide a user with the power to combine these variations in any way they wanted would require an unrealistically large training set, which contains every possibly combination of facial variation.

Much work has been done on researching facial animation models which may be automatically animated using only speech. Current speech driven animation systems only automatically animate lip-synching, and do not animate facial expression and non-verbal sounds - such as auditory expressions of joy (e.g. "Woo!"), relief (e.g. "Ahhhhh"), hesitations, and miscellaneous sounds such as *blowing* noises. The reason for this is that most current facial animation systems are driven using *phonemes*. Phonemes are discrete symbols for describing speech sounds, and contain no non-verbal information. Phonemes may be generated manually by transcribing a speech recording or automatically using a TTS synthesiser [9]. In terms of low-bandwidth communication, phonemes are ideal for transmission over a network and subsequent animation of a face. TTS systems appear attractive

for generating phonemes in such an application since they can be used by unskilled operators. However, TTS systems produce synthetic and almost robotic like speech, which can prove distracting if played along with a synthetic animation over a network. To alleviate this problem manual phonetic transcriptions could be used to animate a face, accompanied by a real speech recording. However, manual phonetic transcription of speech is a specialist skill, and can be extremely time-consuming.

1.3 Animation of a Hierarchical Image Based Model

In this thesis a novel hierarchical image-based 2D talking head model is presented. The model is an extension of a flat Appearance Model [36], and offers a high degree of animation flexibility. The novelty of the hierarchical facial model stems from the fact that sub-facial areas such as the mouth, eyes, lower-face and eyebrows, are modelled individually. To produce a facial animation, animations for a set of chosen facial areas are first produced, either using key-framing or an input speech signal, and then combined into a full facial output (see Figure 1.2). Modelling hierarchically has several attractive qualities. It isolates variation in sub-facial regions from the rest of the face and therefore provides a highly general facial model with which to perform analysis and synthesis. This facial decomposition also provides a high degree of control over different facial parts, and gives meaningful image based animation parameters. For example, it is straight forward to find out which parameter is responsible for raising the eyebrows, or closing the eyes of the model.

The thesis describes how hierarchical models may be constructed and animated using a combination of automatic and semi-automatic techniques. The automatic animation method described in this thesis is used to perform *visual-speech* synthesis, while the semi-automatic method is used to animate the eyes and eyebrows by *key-framing* animation parameters. Once trained, a model can produce new lip-synched animations with a high degree of video-realism. Animation of the model is not solely constrained to the techniques described in this thesis. The flexibility of the hierarchical model also allows other facial areas to be animated from speech given a suitable synthesis model.

The automatic synthesis of animations may be achieved using speech not originally included in the training set. Also, since *Phonetic* units are not used, the model is able to automatically animate pauses, hesitations and non-verbal (or non-speech related) sounds and actions such as *sighs* , expressions of relief (e.g. "Ahhhh..") and excitement (e.g. "Oooohh!!").

Automatic animation of the model using continuous speech makes it more attractive than phoneme driven models in low-bandwidth applications, since no phonetic transcriptions are required, and the

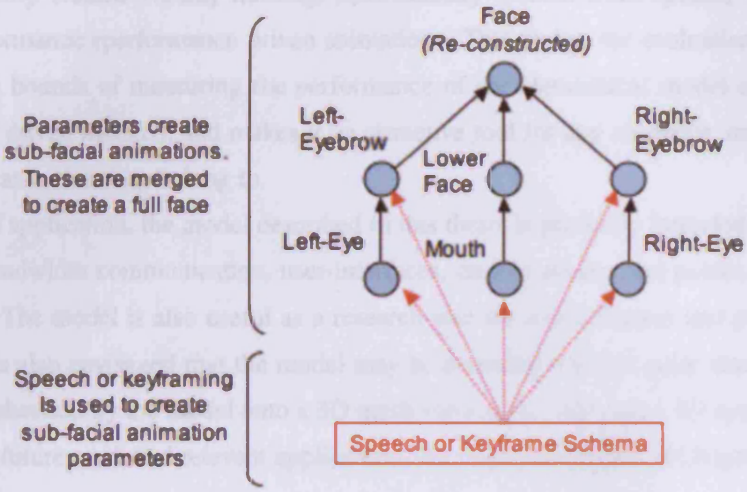

Figure 1.2: A Hierarchical Facial Model: Chosen sub-facial areas are animated via keyframing or a continuous speech signal, and combined to construct a full facial output.

original speech used to animate the model can accompany the animation (for many applications this is more desirable than a robotic voice produced by a TTS system). Thus this thesis, among other contributions, provides an argument for using continuous speech for talking head animations, and not phonemes.

To automatically produce visual-speech, two novel analysis and synthesis methods are proposed. The first method utilises a Speech-Appearance Model (SAM), and the second uses a Hidden Markov Coarticulation Model (HMCM) - based on a Hidden Markov Model (HMM). SAMs are novel statistical models used for estimating appearance parameters from speech parameters. The HMCM is a time based appearance parameter state selection model, primarily intended to incorporate coarticulation in mouth animations.

To evaluate synthesised animations (irrespective of whether they are rendered semi automatically, or using speech), a new perceptual analysis approach based on the McGurk effect [109] is proposed. This measure provides both an unbiased and quantitative method for evaluating talking head visual speech quality and overall perceptual realism. A combination of this new approach, along with other objective and perceptual evaluation techniques, are employed for a thorough evaluation of hierarchical model animations.

The perceptual test described may also be applied to the evaluation of any facial animation,

whether manually created via key-framing, automatically created from speech, or derived from an actors performance (performance driven animation). This makes the evaluation technique useful beyond the bounds of measuring the performance of the hierarchical model alone (or similar parametrically driven models), and makes it an attractive tool for any animator, irrespective of the industry or organisation they belong to.

In terms of application, the model described in this thesis is primarily intended for applications such as low-bandwidth communication, user-interfaces, on-line avatars and guides and multimedia presentations. The model is also useful as a research tool for psychological and physiological experiments. It is also envisaged that the model may be extended into 3D quite simply by applying the texture synthesised by the model onto a 3D mesh - in essence deriving a *3D appearance model.* Such ideas for future work and relevant applications are discussed further in Chapter 13.

1.4 Thesis Overview

The structure of this thesis is as follows:

- In Chapter 2 a review of relevant literature relating to facial animation is given along with a description of how the work in this thesis contributes to the field. In particular the review covers work on 3D polygonal talking heads, image based models, and the Phonetic and continuous techniques which have been employed to animate them.

- In Chapter 3 an overview of the hierarchical facial animation system is given, with descriptions of each of its major processes: 1) Acquisition and Initialisation, 2) Analysis and Learning, 3) Synthesis and Animation, and 4) Reconstruction and Display. The overview also describes how following Chapters in the thesis relate to each of these processes.

- Chapter 4 describes the Acquisition and initialisation process. In particular it addresses how video and audio data is used to construct a hierarchical model. The Chapter also describes techniques for automatic image annotation, constructing appearance models, and obtaining robust and uncorrelated speech features.

- Chapters 5 and 6 are related to the respective processes of Analysis and Learning, and Synthesis and Animation for visual-speech production using SAMs. Chapter 5 describes how

correlated speech and appearance parameters are used to construct a SAM, while Chapter 6 describes how facial areas are animated using a SAM.

- Chapters 7 and 8 are also related to Analysis and Learning, and Synthesis and Animation respectively. However, an alternative approach to producing visual speech is described based on a HMCM. The HMCM builds on the SAM approach, and addresses co-articulation in visual speech synthesis. The two Chapters describe how to construct a HMCM and use it for synthesis.

- In Chapter 9 the process of Reconstruction and Display is addressed. Specifically it describes how sub-facial animations are merged into a full facial animation, and then re-attached onto existing background frames (an optional step) to increase realism.

- Chapter 10 describes the process of Synthesis and Animation in terms of semi-automatic animation, i.e. how separate sub-facial regions may be animated by key-framing animation parameters.

- In Chapter 11 an evaluation of HMCMs is carried out, focusing on their capacity to faithfully reproduce animations from speech and non-verbal sound. SAM performance is also demonstrated in this Chapter, although a more detailed evaluation may be found in [42, 43, 44].

- Chapter 12 outlines a novel perceptual evaluation technique based on the McGurk effect, and describes how it may be used to identify strengths and weaknesses in visual-speech animation in order to guide future development. The method is applied to HMCM based hierarchical model in an attempt to evaluate its ability to faithfully animate lip-synching, and overall behavioral realism

- In Chapter 13 directions for further research are discussed.

- Chapter 14 concludes the thesis by reiterating its major contributions, and demonstrating how these have been proved.

1.5 Main Contributions

The major contributions of this thesis may be summarised as follows:

- An image-based hierarchical facial model for animation. The model improves on previous image-based models in the level of control it offers the animator. The model also offers a greater degree of specificity in terms of statistical modeling, since sub-facial areas are modeled individually as opposed to being encoded as part of a larger model (e.g. one of the whole face).

- Realistic animation of an image based model using continuous speech signals, including automatic animation for lip-synching, natural pauses, hesitations and *non-verbal* articulations. These features are facilitated using two novel analysis and synthesis techniques: SAMs and HMCMs. Both methods allow for the synthesis of accurate animations given relatively little training.

    1. The non-linear SAM offers a novel approach to synthesizing appearance parameters from continuous speech parameters. It achieves this by statistically encoding relations between appearance and speech, allowing the estimation of one parameter from the other to produce animation.
    2. The HMCM plays a similar role to the SAM, in that appearance parameters are estimated from continuous speech parameters. However, the model also analyses and synthesises co-articulation, providing more accurate animations for the mouth. HMCMs provide the most realistic visual-speech animation to date from continuous speech.

- A semi-automatic technique - akin to keyframing - for producing animations using image-based models. The method differs from a classical key-frame approach in that key-moments are not defined using images. Instead, key-moments are defined using appearance parameter values for different sub-facial areas, and *in-betweening* carried out by interpolating these parameters. The hierarchical decomposition of the face provides intuitive animation parameters for different sub-facial areas for the facilitation of this technique.

- A powerful tool for performing facial analysis. The hierarchical model, when applied to the analysis of a face, yields insightful information regarding facial dynamics, and the interaction between different facial areas. This information is particularly interesting not only from an animation point of view, but also from a perceptual one. With respect to animation, knowledge of real facial dynamics - such as onset and offset lengths of behaviours such as blinks - can

improve the realism of synthetic animations. This information can be applied to the animation of any model, whether it is intended to be video realistic or not. From a perceptual point of view, the behaviour of individual facial areas during expressions can be examined, and used as the basis of recognition and classification.

- A novel perceptual evaluation technique based on the McGurk effect for determining the effectiveness of visual-speech synthesis in synthetic facial animation. The approach identifies strengths and weaknesses in synthetic animation algorithms, thus guiding further development. Also, the approach is stand-alone, and may be applied to the evaluation of any facial animation system.

- An automatic landmark placement algorithm for facial feature tracking and subsequent construction of Appearance Models. The algorithm is based on Downhill-Simplex Minimisation (DSM), and incorporates strategies to optimise the search and avoid local minima based on prior knowledge.

Chapter 2

Facial Animation: A Review

The subject of facial animation is extremely broad, and encompasses numerous construction techniques, animation methods and applications. As such this Chapter is by no means an exhaustive review of facial animation as a whole, rather it concentrates on niches of facial animation research and those aspects of statistical modelling relevant to the thesis.

The major aim of this review is to place the contribution of this thesis along side previous speech driven and image-based facial animation work. Today, facial models may arguably be placed into one of three broad categories: 3D Polygonal models, muscle-based models or image-based models. Similarly there have emerged two major means of automatically animating these models using speech: by using Phonetic inputs or continuous speech inputs. In this thesis the former technique encapsulates any method which uses phonemes to drive the model, whether they are generated via Text-To-Speech (TTS) systems or manually labelled according to a voice recording. Continuous speech methods are defined here as methods which use *raw* speech as a basis for input, typically sampled straight from a recording device such as a microphone. To avoid confusion, techniques which utilise continuous speech to generate Phonemes are regarded here as Phonetic methods, and not continuous speech methods.

The Chapter begins by considering areas of study that inspired the design of early facial models, including psychology and anatomy, and considers some relevant landmark systems. It then describes 3D geometric modeling (including muscle, physics and anatomy based models), and image-based modelling of faces. Techniques for facial animation are then reviewed, including key-framing,

motion-capture and speech driven control. Finally, methods for perceptually and objectively evaluating facial animation are considered. The Chapter concludes with an overview and reiteration of the contributions of this thesis in light of previous facial animation work.

2.1 Origins of Facial Modelling and Psychological Motivations

Facial representation in computing began with low order polygon models in the 1970s. Today this representation has developed with advances in technology and computer sophistication allowing more polygons to enhance facial detail. Facial animation methods incorporated today are descendant from the first animation methods invented for early cartoons. The traditional animation process may be described as follows:

"...synchronisation between the drawn images and the speech track is usually achieved through the tedious process of reading the prerecorded speech track to find the frame times of significant speech events. Key frames with corresponding mouth positions and expressions are then drawn to match these key speech events. The mouth positions used are usually based on a canonical mapping of speech sounds into mouth positions."

- F. Parke and K. Waters [120].

These early techniques are still often incorporated in a similar form today for film and computer game animation, i.e. they form the basis of key-frame/target-morphing animation methods [88]. This approach to animation is time consuming, tedious, expensive and requires highly skilled artists and animators. The benefits of a fully automated facial animation approach, driven by speech, are therefore obvious.

The first parameterized 3D facial animation system was introduced by Parke [118]. In Parke's system a single facial topology is controlled by parameters that define facial conformation and control facial expression. At least two facial poses are modelled using the topology and a parameter, acting as an interpolation coefficient, is used as a function of time to change the face from one expression to another. In [66] Parke then demonstrated that one topology is sufficient to represent most facial expressions.

The design of facial models is largely inspired by either anatomical studies [71], psychological

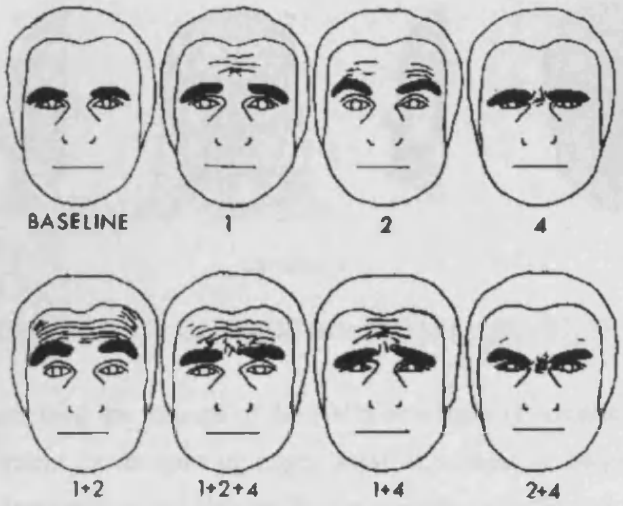

Figure 2.1: AUs 1,2 and 4, along with a neutral face (top) and combinations of these AUs [58].

studies [74, 58] or traditional animation techniques [146]. In 1969 Carl-Herman Hjortsjo developed the *mimic language*, one of the earliest attempts to develop an expression coding system [74]. This psychologically motivated study attempted to introduce a medium by which facial expression (and in a less direct sense, emotion) could be quantified, measured and described.

In 1977 Paul Ekman and his colleagues extended the work of Hjortsjo and introduced what is called the Facial Animation Coding System (FACS) [58]. FACS divides the face into upper and lower facial action and subdivides facial motion into Action Units (AUs). The goal was to develop a system capable of describing all visually discernable facial movements. Many modern parameterized facial models are based on FACS and AU's. The quantification of facial action using these and similar means has led to a number of similarly motivated works [136] and attempts to standardise parametric facial models [1]. The highly impressive creature "Gollum" in the Lord of the Rings Movies [30] is also based on the FACS coding system.

Ekman's FACS contains 46 AU's to account for changes in facial expression and 12 AU's to describe changes in head orientation and gaze. The naming of an AU describes its associated facial action (i.e. Lip Corner Puller, Upper Lip Raiser). Figure 2.1 shows an example of four AU's and how they are used in FACS to describe facial configurations. FACS usually requires trained individuals to analyze facial poses and decompose them into underlying AUs.

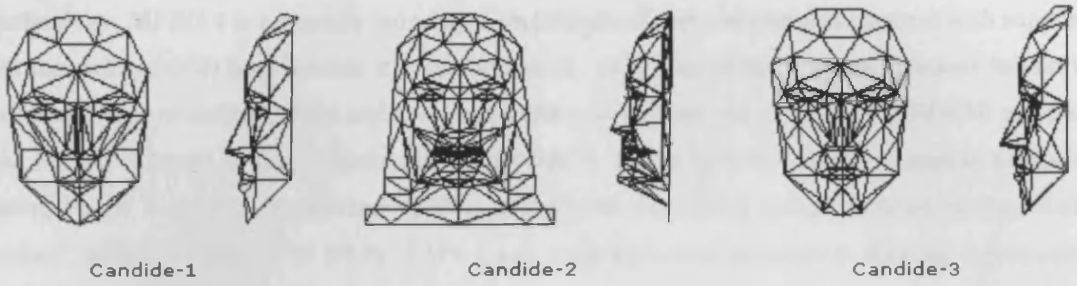

Figure 2.2: The CANDIDE facial models [136, 157, 3] .

Several studies have used the concept of the FACS as a basis of emotion recognition. Lien *et al* [98] developed a system for recognizing upper facial AUs based on image sequences. Feature point tracking, dense flow tracking and high gradient component detection are used to produce input features for a HMM which performs recognition. Their results show a 93% recognition rate using the dense flow tracking method. Ying *et al* [147] attempted to recognise AUs in the lower part of the face using geometric feature vectors extracted from a multi-state model for tracking the lips and the appearance (or non-appearance) of transient facial features such as wrinkles. These features were then used as inputs to a 4-layer (including one hidden-layer) neural network for AU identification. A 100% recognition rate is achieved with almost all the lower AU's apart from a few which cause the network confusion due to their similarity with other AUs.

In [136] Rydfalk described a low-polygon facial mask called the CANDIDE model based on the principle of AUs. The model contained 75 vertices and 100 polygons. It was later implemented by Stromberg and modified to include 79 vertices, 108 polygons and 11 AUs. In [157] Welsh created another version, named CANDIDE 2, with 160 vertices, 238 polygons and 6 AUs. CANDIDE 2 not only modelled the face but also the neck and upper shoulders. The most recent version of CANDIDE, CANDIDE 3, was developed and implemented by Ahlberg [3]. This version contains 113 vertices, 168 polygons and 20 AUs. The motivation behind CANDIDE 3 was not just to bring the *de facto* CANDIDE version implemented by Stromberg up to date, but also to make the CANDIDE model compliant with the emerging MPEG-4 standard [1]. Figure 2.2 displays the various incarnations of CANDIDE.

In [119] Parke suggested that a convincing synthetic representations of a person could operate over low data rate channels and used in applications such as videophones. In today's world the applications are no-longer restricted to videophones but also include the internet and mobile phone

technology. MPEG 4 is a recently introduced multimedia object compression standard with support for the coding of 3D head models at a low bit rate [1, 143]. One of the major motivations behind it is the delivery of online avatars and low-bandwidth teleconferencing. Like the CANDIDE models, the MPEG 4 model is also a descendant of the FACS. In the MPEG 4 model control is achieved using Facial Animation Parameters (FAPs), along with associated activation levels termed FAP values. MPEG 4 contains 68 FAPs. FAPs 1 and 2 are high-level parameters defining expressions and visemes and FAPs 3-68 define low-level facial motion. The naming convention of the latter FAPs hold similarities with the original AU naming (i.e. names such as shift_jaw and push_b_lip).

As a natural progression of the introduction of MPEG 4 as technology intended for low bandwidth broadcast, it has found implementation in a Java based Application Programming Interface (API) named Java Facial Animation Engine (JFAE) developed by Bonamico *et al* [13]. The JFAE is based on the Java 3D API and allows web pages to include Applets hosting computer generated speakers. An example of MPEG 4 coding used for mobile communications may be found in [115]. In this work high resolution 3D scans of real people are used to build parametric models which may be animated in real time and encoded using the MPEG 4 format for delivery on mobile devices. Other implementations of MPEG 4 parameter driven 3D head models also exist for low-bandwidth communications [142, 99].

2.2 Muscle-Based, Physics-Based and Anatomical Modelling

On the bookshelf of any facial animator you are likely to find a text describing facial anatomy. Descriptions and diagrams of facial anatomy are used to help construct facial models which look realistic. For example, when constructing the ear of a character in an animated movie, often achieved by manually shaping a 3D wire-frame mesh, an anatomical diagram will often be used for reference [133]. However, anatomical research is not only applied in modelling the shape of the outerface for 3D polygonal wire-frame models - it is also often used as the basis for the design of more complex models, which include bones, muscle and skin.

Although Platt and Badler [127] were perhaps the first to use anatomical naming conventions in a facial model (they also discuss, but do not implement, physics based bone and skin modelling for realistic animation and simulation), perhaps the earliest full muscle model was by Waters [156]. In this model the skin is regarded as a mesh of springs that deforms under the tension of a muscle layer beneath, and animation is achieved using FACs based control parameters (interestingly, Platt and

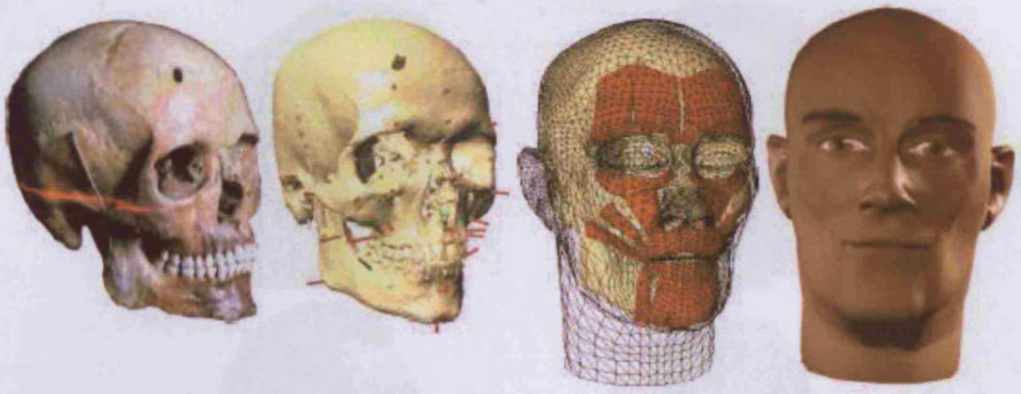

Figure 2.3: Reconstruction of the face from a skull [80].

Badler also incorporate AUs for animation [127]). Waters' model has been extended in many subsequent works to include increased realism of facial expressions [25], the ability to overlap facial areas (such as the lips) and improve computational performance [51], and increased layers, including fatty tissue modelling beneath the skin [144]. It has also been used for animation alone [60].

Advances in computing have lately led to anatomical modelling being used for forensic studies [80, 154], where facial appearance may be reconstructed using only human skulls (see Figure 2.3). Anatomical modelling has also been successfully used to develop facial models for surgical planning and simulation. In [86] Keele *et al* propose an anatomy-based 3D finite element tissue model which allows the precise prediction of soft-tissue changes resulting from the realignment of underlying bone structures.

Muscle based models are now common place in computer graphics for both animation and simulation, and there are many other notable works. Also, since these models are based on anatomy they also incorporate physical laws, with varying levels of sophistication. In [96] Lee *et al* automatically generate models which include the skull, muscles and skin, with sub-models for the eyes, neck and teeth, from laser-scanned range and reflectance data (see Figure 2.4). In [79] Kahler *et al* describe a three-layer animatable muscle based model, these layers corresponding to the skin or tissue, muscles and the skull. This model may also be fitted to laser scan data by aligning a generic skull model. Conversely, when no skull model is present, the data may be used to approximate one. The character of Shrek, in the Dreamworks animated feature of the same name [52], was also based on an anatomical model. An underlying model of bone and muscle was dressed with the outward

Figure 2.4: Animation of a muscle based model generated automatically from laser-scanned range and reflectance data. The model may subsequently be animated by relaxing and contracting the muscles in the model [96].

appearance of the character. Of course in this case design of the underlying bone and muscle is based as much on artistic licence as it is on human anatomy and physics.

The increased complexity associated with muscle-based models adds to computational overheads, and impacts on the potential for real time application, although highly complex examples of such systems do exist [159, 160, 25, 81]. Approaches have also been taken towards using parallel processing to provide real-time animation and interactivity with a model [96].

2.3 Hand-Defined, Multi-View and Laser-Captured 3D Facial Models

To begin, a differentiation should be made between muscle/anatomical/physics based models and the 3D models discussed in this Section. The models discussed in this Section are not necessarily based on physical or anatomical design principles, although they are primarily 3D. Instead this Section contains models which typically only incorporate a superficial polygonal mesh, typically

rigged [1] with axis for animation, and parametrically controlled or performance driven (e.g. using motion-capture data). The detail of these models varies from simple facial outlines in 3D, to models which incorporate the teeth and tongue. However, this is generally where the similarities between models in this and the last Section end [2].

Many 3D models of faces are created by artists and animators by using Computer Aided Drawing (CAD) packages such as MAYA, SoftImage and 3D Studio Max [88, 28]. The process of building and animating the faces of characters for films and computer games often begins by simply moulding primitive shapes such as cylinders, cubes and spheres into desired forms. A *rigging* procedure then defines how the 3D model may be manipulated, and a technique such as motion-capture, key-framing or blend shape animation may then be employed to create a pre-rendered animation.

It has already been seen how 3D laser scan data may aid in the construction of muscle based and anatomical models (Section 2.2). However, 3D laser scan data is also often employed to produce parametric and performance based animation models. In [158] Williams uses a Cyberware laser scanner to gather 3D coordinate data from a plaster cast of a persons head. The subsequent 3D mesh, after post-processing to deal with artifacts such as missing data, is then textured by morphing photographic data onto its surface. In [83] Kalberer and Van gool extract 3D face data using a combination of a 3D shape scanner and facial marker data. Breidt *et al* produce highly realistic facial animation in [23]. In this work a set of high definition 3D laser scans of facial poses are used to create a basis of morphable meshes (see Figure 2.5). Motion capture data is then used to animate the meshes and produce animation. An example of performance driven animation applied to a laser scanned model, where animation data to control the mesh is later extracted from a standard video camera, may be found in the work of Chai *et al* [27].

A less costly alternative to using 3D scanners for building 3D facial models is to use multiple facial views from monocular video cameras. Malvar *et al* [107] combine multiple cameras and facial markers to build a realistic head model employed for performance based animation. A markerless extension of this work by Pighin *et al* [124] fits a generic polygon head to 3D data calculated from multiple views. The resulting model produces a realistic output capable of representing a wide range of facial configurations for animation. In [125] Pighin *et al* use the same model for tracking video sequences and return head pose and shape data for animation re-synthesisation. There are

[1] "Rigging" is a term used in computer animation to describe the process of defining axis of movement for a model [88, 28] (e.g. elbow joints on a full body model)

[2] The type of model described in this Section is also often described as a blend shape model

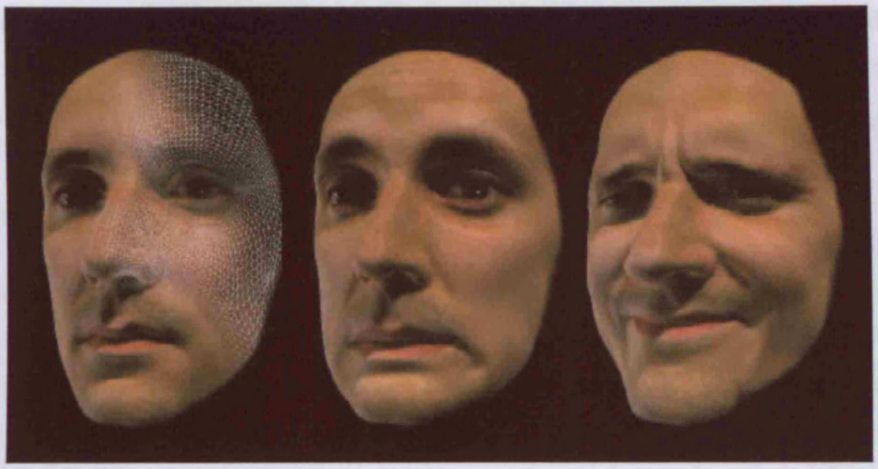

Figure 2.5: Rendered 3D morph shapes. The model is created using a series of high definition 3D laser scans [23].

many other examples of using multi-view camera data to construct 3D models [112, 59, 151]. An example of construction of a 3D head model using only a single camera may be found in the work of Lui *et al* [105]. Lin *et al* [101] take a slightly different approach, and use a single camera with the aid of mirrors and facial markers to gather 3D data. This data is then used to drive a generic head model in performance driven animations. A further example of the initialisation of a 3D model using the same technique may be found in the work of Reveret *et al* [134]. Tracked markers provide the basis of a 3D facial mesh, and polynomial interpolation is used to increase the detail of the lips. A model for the jaw and teeth is then attached to the wire-frame in order to provide inner mouth detail.

The use of hand-defined 3D models for film and video games has been mentioned previously. However, study using such models in academia has also yielded interesting work. BALDI [108] is a parametrically controlled talking head which has been used in dozens of psychological experiments, including a tutor for hard of hearing and autistic children [16] and as an aid for learning new languages [116, 41]. BALDI's original appearance is quite generic, and thus has also been modified by Cohen *et al*to provide personalised output of any speaker's appearance [32]. Facial expression cloning, performance based animation and speech based animation has also been achieved using generic facial meshes in [130, 29] and [93] respectively. The work of Pyun *et al* [130] provides a good example of animation of a wide range of cartoon-like models from tracked facial data. The

major problem addressed is of course the dynamic mapping between tracked features on the participant and corresponding feature movement on the model - since facial dimensions and movement ranges invariably differ between subject and model.

2.4 Image-Based Facial Modelling

The terms image-based and morphable model are sometimes used interchangeably in computer graphics and computer vision. The term image based model often describes a model which uses single images for display and 2D or 3D image warping for animation - where the image may be a static image or a dynamic image which changes its appearance corresponding to the facial expression desired. Image based models also refer to models which re-order existing frames from a training video for animation. Morphable models (including appearance models) are powerful statistical image models, and *always* use dynamic textures and 2D or 3D texture warping to alter shape. In this thesis both types of model are placed under the umbrella of image-based models, unless an author specifically refers to a model as *morphable*.

One type of morphable model, proposed by Breidt *et al*, has already been described, and provides an example of an advantage of morphable models over most other types of 3D model. 3D models are often textured with static textures (e.g. [158, 27, 83, 112, 59, 151]). Deformation of the polygon wire frame under the texture therefore simply deforms the texture, and does not add detail for certain facial configurations (i.e. eyebrow vertex movement may not be accompanied by wrinkles appearing on the forehead). The model of Breidt *et al* [23] includes high-detail laser-captured textures for different facial poses and morphs between the textures during animation, thus providing appropriate texture appearance under varying facial mesh-shape configurations.

Facial animation using real images is often achieved by warping static single images, such as photographs, or reordering a set of images taken from a video. In Voice Puppetry [21] single images of famous people, taken from photographs and paintings, are animated by warping the vertices of a 3D model - where the 3D model is associated with feature points labelled on these images. In Video Rewrite [22] sequence of images corresponding to *tri-phones*, or phoneme triplets, are extracted from an example video and stored in a database. A new phoneme sequence corresponding to a novel speech recording is then used to select the best matching sequence of tri-phones from the database - forming the new animation. As in Voice Puppetry [21] this technique has been successfully used to produce animations of famous people from archive video footage. Figure 2.6 shows example

Figure 2.6: Example video frames from two synthetic animations generated using Video Rewrite [22].

frames extracted from two different synthetically generated video sequences created using the Video Rewrite method.

Lin *et al* [100] animate a single face image using a 2D deformable facial mesh. Landmarks placed on the image correspond to vertices on the mesh, and key-frames are defined by deforming the mesh and the image to create facial expressions. Animation is then achieved by interpolating between key-frames. In the work of Faruquie *et al* [65] facial images representing 12 key visemes and 7 facial expressions are captured and normalised, and optical flow morphs calculated for each combination. Animation is produced using phoneme inputs and coarticulation rules - which define behavioral relationships between the phonemes and the key-frames.

In [40] Cosatto and Graf take an image reordering approach to facial animation. Image recognition is applied to a video of a speaker to extract sub-facial regions such as the mouth, eye-brows and eyes. These regions are then parameterised and stored in a database. During animation these regions are then extracted from the data base, blended and then reconstructed to produce a full facial animation.

The facial model employed by Cosatto and Graf is also based on hierarchical decomposition of the face. However, their hierarchy does not consist of separate statistical models of sub-facial variation which produce different outputs by varying animation parameters. Instead, Cosatto and

Graf produce sub-facial animations by re-ordering images from the training set. Their model is therefore not parametric. For example, using the the hierarchical model described in this thesis an eye can be opened or closed by increasing or decreasing a single parameter value. In Cosatto and Graf's model this can only be done by manually selecting appropriate eye image sequences from the training corpus.

In [63] and [64] Ezzat and Poggio use image morphing to create animations from phonetic inputs. Images corresponding to visemes are extracted manually from a video corpus, and optical flow blending of the images is calculated to provide in-between animation. Visemes are selected for output display using a phoneme to viseme mapping defined by the input speech. An improved version of this system may be found in the work of Ezzat *et al* [62]. In this work a trajectory synthesis algorithm uses phonetic transcripts to generate paths through Multidimensional Morphable Model (MMM) space. The MMM defined consists of a statistical model of texture and optical flow shape information, such that given a shape and texture parameter a unique facial output may be generated.

There has also been much work done using 3D morphable models. The work of Breidt *et al* [23] has already been mentioned in Section 2.3. The earlier work of Blanz *et al* [11] describes another 3D morphable model constructed using 3D laser scans of male and female faces, and controlled using shape and texture parameter vectors (see Figure 2.7). This model has subsequently been used for face recognition [12] and performance driven animation [10]. The model defined by Pighin *et al* [124], although not explicitly described as such, may also be placed into the category of a 3D morphable model. In this work a set of photographs taken from different views of a subject displaying various facial expressions is used to create a realistic 3D model. This is then animated using 3D shape morphing and texture blending.

Appearance Models [36] have been used successfully in computer vision since their introduction in 1998. Applications have included face recognition [55], face tracking [56], medical image analysis [38] and age simulation [95]. Appearance models are joint statistical models of shape and texture, where a single *appearance parameter* defines a corresponding texture and shape vector. Appearance models are used as the basis for the hierarchical facial model described in this thesis, and a full description of them is given in Chapter 4.

Although not originally intended for the purpose, appearance models provide an ideal platform

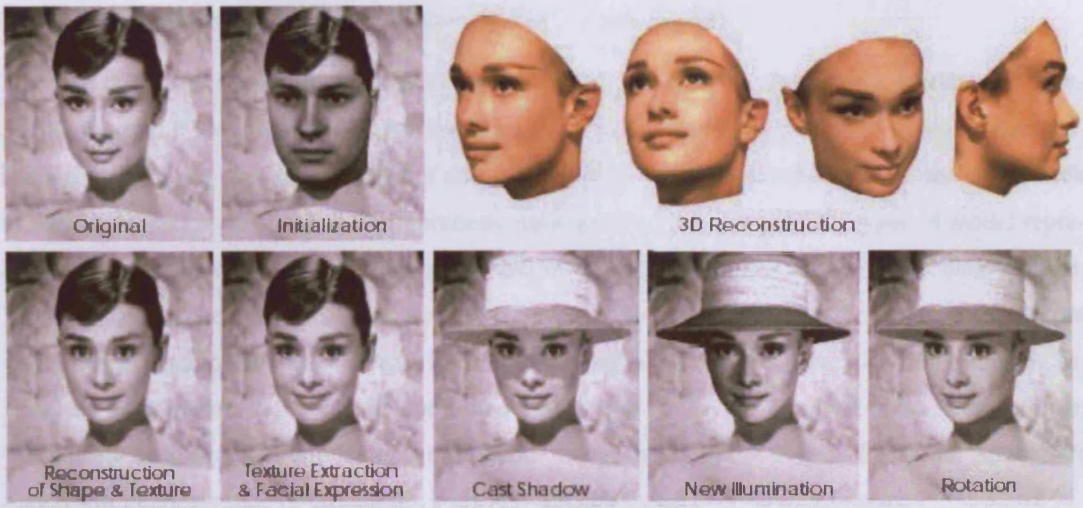

Figure 2.7: Demonstration of a 3D morphable model [11] fitted to a photograph. This allows the alteration of lighting, facial expression, illumination, pose change and 3D reconstruction [11].

to produce animations. By altering the values of an appearance parameter, and rendering corresponding images, facial animations may be synthesised. In [145] Theobald *et al* use an appearance model to produce facial animations given phonetic transcriptions of speech, where appearance parameter trajectories are estimated from phonemes using a code book. Realistic talking head behaviour using appearance models is replicated by Hack and Taylor in [72]. In this work appearance parameters are calculated using a Hidden Markov Model (HMM), trained using an example video of a subject during conversational speech.

An interactive talking head is presented by Devin and Hogg in [49]. Here an appearance model provides the output to an interactive system which produces appropriate facial animations and accompanied speech in response to user messages, relayed via a microphone to the computer. It should be noted that this work was one of the first to model distributions of joint audio-visual parameters. The visual parameters in this work are appearance parameters, while the speech parameters are spectral parameters obtained by performing a Fourier transform on the audio training set. PCA is then applied to the audio parameters in order to parameterise them. The modelling of audio-visual parameters in this way has similarities to the approach used in this thesis. The SAM and HMCM use similar visual features (although only for sub-facial areas), and use Mel-Ceptral speech features parameterised using PCA. Both models are also based on joint audio-visual distributions.

2.4.1 Advantages of a Hierarchical Image Based Model

To close this Section an important point should be made regarding appearance models and morphable models. These powerful image based models allow parametric control of facial appearance. Essentially they are statistical models of shape and texture variation, created using a technique such as a PCA. Hence, a set of facial configurations parameterised via one of these types of model represents a *manifold* in a low dimensional space, and may be approximated using a set of basis vectors. The weakness of these models - with respect to animation - is that the basis vectors do not represent variations in separate sub-facial areas, i.e. sub-facial variation is not *orthogonal* in the model. Instead the basis vectors represent the combined variation of multiple sub-facial areas. For example, when using an appearance model, or a morphable model, alteration of one of the model parameters may result in appearance of the mouth, eyes and forehead changing simultaneously.

One of the advantages of the hierarchical model in this thesis is that sub-facial variation is isolated with respect to the variation of model parameters. This makes it better suited for animation than a standard appearance model or a morphable model; since the animator is aware precisely how certain regions in the face will change given modification of the facial parameters. This lends the hierarchical model to further applications such as manual key-frame animation using sub-facial animation parameters (this technique is described more thoroughly in Section 2.5).

Finally, another advantage of the hierarchical model over flat (i.e. face-only) appearance models or morphable models is that synthetic animations are not accompanied by unwanted facial variation in certain parts of the face. If the aim is to create only lip-synched animations using a morphable model or an appearance model [145], then the designer is required to ensure that any variation appearing in parts of the face other than the mouth is minimised. This reduces the risk of any subsequent lip-synched animations being accompanied by, for example, an unrealistic or mistimed raising of the eyebrows, or shutting of the eyes, whenever a certain mouth pose is made. This artifact exists because current image based animation systems based on morphable models and appearance models relate visemes to *full-facial poses*, and not mouth configurations alone.

The hierarchical model animates different facial areas separately - thus removing this problem, and allowing the subject recorded to build the training set full freedom of expression; without the risk of potentially damaging any synthesised videos.

2.5 Animating Faces

The Chapter now turns solely to the *animation* of facial models. Four broad methods are discussed: keyframe (or target-morph/blend-shape) animation, performance driven animation, automatic animation from phonetic transcriptions and automatic animation from continuous speech.

Keyframe animation may be applied to many different types of facial model, and is the most common animation method used today for film and video games. Key-framing is also the oldest traditional animation method, and in computer graphics emulates the techniques used by both early and current cartoon animators. The philosophy is that key moments for a character are defined for meaningful actions at specific times, and then *interpolated* between to fill the time between these moments (this process is often called *in-betweening*). Key moments include mouth poses, facial expressions, or in the case of full body animation, an action such as opening a door [88].

One major issue of key-framing is the definition of key-frames, and it highlights the flexibility of 3D models over 2D image-based models such as MMMs [62] and appearance models [145, 36]. The definition of key-frames in 3D models may be achieved manually, by setting the position of model vertices, parametrically as in the CANDIDE models [3], or using blend shapes [78] - i.e. by linearly combining specified facial expressions and poses [3].

Performance driven animation, as the name suggests, involves animating a facial model based on data extracted from the performance of a human subject - such as an actor. Data may be obtained by using markerless facial tracking [158], marker based facial tracking [101], model-based tracking [125] or motion capture [23]. An advantage that this animation method carries over key-frame animation is that key-frames and blend-shapes need not be defined, and nuances in the actors performance can be carried directly to the model, without the need for a highly skilled animator. However, solutions do exist where blend-shapes are also applied. In [29] Chuang and Bregler use blend shapes to help map between tracked facial data on a performer and facial poses associated with a cartoon-like 3D facial model. This is required since the dimensions of the cartoon model, and the allowable movement ranges of its facial control points, are not likely to coincide close enough with the corresponding tracked positions of the markers on the performers face. When a 3D model based on the performers face is used, this problem is less prevalent, since the mapping between control points on the mesh and markers on the performers face is likely to be one-to-one [158, 125].

[3]The model of "Gollum", used in the movie "The Lord of the Rings" [30], required 675 blend shapes

2.5.1 Phonetically Driven Speech Animation (and its variants)

The topic now turns to the automatic generation of facial animation using phonetic transcriptions and continuous speech signals. For a clarification of the kinds of models which fall into these categories, the reader is referred to the beginning of this Chapter.

Phonetics was born from psychological studies into speech, and provides a discrete alphabet of speech sounds called Phonemes. The number of phonemes in an alphabet is language dependent, and given an alphabet any spoken sentence may be transcribed into a set of phonemes. The use of phonemes in animation has led to the development of visemes - the visual counterpart of phonemes. To clarify the relation between phonemes and visemes is it helpful to consider the human speech production system.

Speech sound begins with air-flow from the lungs, which is first filtered by the larynx, altering the pitch of the sound, and then the tongue, teeth and lips [120]. Visually, the production of a phoneme is associated with a specific configuration of the tongue, teeth and lips - thus leading to the definition of a viseme, the counterpart of the phoneme. If the mapping between visemes and phonemes were one-to-one then automated animation of the lips from speech would be a more straightforward problem. However this relationship is one-to-many, e.g. the phonemes for /b/ and /p/ are associated with the same viseme configuration, that being a closed mouth. The problem of speech generation, using phonemes or continuous speech, is made more difficult still due to *coarticulation* [31]. Coarticulation is the effect that preceding (backward coarticulation) and forthcoming (forward coarticulation) speech segments have on the articulation of the current speech segment. For example, depending on the next viseme to appear, the current viseme shape will be modified in anticipation. Similarly the configuration of a current viseme is affected by the previous viseme. An effective automatic animation system has to model this phenomena, and this has been approached in several ways.

In the work of Cohen and Massaro [31], and Cohen *et al*, [33] a model descending from Parke's [66] is auomatically animated using phonetic transcriptions and coarticulation rules based on an adaption of the articulatory gesture model of Lofqvist [102]. In this model speech segments have a *dominance* over vocal articulators (or animation parameters), which increases and decreases over time during articulation. The dominance of adjacent speech segments overlaps, and is thus *blended*. The model is therefore parametric in terms of the type of *dominance* and *blending* functions it uses. To produce animation, each phoneme is associated with specific animation parameters

and dominance function characteristics, such as magnitude, time offset, leading attack, and trailing decay rates. The phonetic transcription therefore creates a trajectory of animation parameters, which is formed by the dominance of adjacent phonemes, and appropriately blended.

A popular approach to relating phonemes with visemes for lip-synched facial animation is through rule-based mappings, and some examples are described here. Kalberer and Van Gool [83] approximate mappings based on simplified many-to-one relations, e.g. the phoneme for /b/ and /p/ relates to a single viseme with the appearance of a closed mouth and pursed lips. The phonetic transcription therefore relates to a set of facial model viseme parameters, which are interpolated with splines to provide smooth animation transitions. Faruquie *et al* [65] specify viseme to phoneme mappings using a rule-table. Viseme themselves are defined using images manually extracted from speaker video footage, and viseme transitions calculated using optical-flow warping. The extent of the morphing for viseme transitions (i.e. coarticulation), is calculated using phoneme timing information. In [69] Le Goff and Benoit describe the animation of a parametric facial mesh based model where visemes are defined by matching the model to a corpus of example facial poses. For each pose they record the values of the model parameters. Then, given a stream of phonetic information produced by a TTS synthesiser, they estimate corresponding visemes with their model, again using a set of mapping rules. Viseme transitions are then calculated based on the dominance and blending functions described by Cohen and Massaro [31]. In the work of Lin *et al* [100] a single face image, deformed using a generic 2D polygon mesh, is also animated using rule mappings. Key-frames representing visemes are defined as vertex offsets from a neutral mesh, allowing the animation of arbitrary single facial images. Coarticulation is then controlled by applying an exponential decay model acting to adjacent key-frame parameters. Beskow [6] uses a TTS system to produce 3D facial model parameters via a rule based module named RULSYS. The module converts phonetic transcriptions into animation parameters using a mapping which groups visually equivalent or similar phonemes. Coarticulation is animated in a novel manner by leaving the value of certain visemes unspecified, and calculating them during synthesis based on the proximity of neighbouring visemes.

An alternative to using rules for specifying phoneme and viseme mappings is by learning these relationships from examples. In the work of Ezzat *et al* [62] a MMM, trained from example images, is used to analyse phoneme transcribed speaker footage, and construct phoneme clusters in MMM space. A stream of input phonemes is then used for animation by calculating a trajectory through this space. Coarticulation is solved by directing the path of the trajectory according to its current

proximity to a phoneme cluster. The influence of a phoneme is represented in MMM space using the covariance matrix of a MMM phoneme cluster. In [40] Cosatto and Graf use a training video to construct a data base of facial part samples and associated phonemes. To synthesise new animations a TTS system generates a phonetic transcript and candidate mouth bitmaps are estimated for each output frame, constructing a lattice structure of probable mouths. Animation is then achieved by performing a Viterbi search [131] through the trellis and selecting appropriate mouths for output.

Although the use of single phoneme units is the most common way to obtain robust discrete speech data there has been automatic animation work carried out using phoneme variants. One recent facial animation work by Kshirsagar and Magnenat-Thalmann uses *syllables* and *visyllables* [93] to drive a talking head. The authors define a visyllable as the visual counterpart of a syllable, and describe a syllabification algorithm to convert phonetic transcriptions into sylabic equivalents. During training, a motion capture system records a database of visyllables, which are subsequently parameterised according to the facial model. Parameters are then selected during animation synthesis and smoothed at vowel boundaries to create a smooth animation. The concept of *tri-phones* has previously been mentioned in Section 2.4 in relation to the Video Rewrite system [22]. To reiterate, tri-phones are defined as phoneme triplets. In Video Rewrite a video of a speaker is tracked and each image normalised into a common coordinate frame. A database is then constructed of tri-phones with corresponding short video clips extracted from the training video. This is done according to a phonetic transcription. For synthesis of new animations an input phoneme transcription is analysed for suitable tri-phone segments, which are concatenated to construct an output animation. Performance of the system in terms of animation quality is proportional to the number of tri-phone segments in the database. Tri-phones are also used by Theobald *et al* [145] in the animation of an appearance model. In this work appearance model parameters corresponding to tri-phone segements are stored in a code-book. Synthesis is then achieved by comparing input tri-phone segements to entries in the code-book and selecting the most appropriate appearance parameters. The resulting parameters are then post-processed using smoothing splines to remove any animation artifacts.

2.5.2 Facial Animation using Continuous Speech

Methods for facial animation using continuous speech (i.e. raw speech) are now considered. The problem of animation from continuous speech may be viewed as one of *signal estimation*. For

example, given two corresponding signals (i.e. speech and animation parameters), how can a new example of one of the signals (i.e. new speech parameters) be used to produce a suitable estimate of the other (i.e. new animation parameters).

Given a continuous speech signal it is first processed to acquire robust and uncorrelated continuous speech features. Several techniques exist which allow this, including Linear Predictive Coding (LPC), Mel-Cepstral analysis and Relative Spectral Transform-Perceptual Linear Prediction (RASTA-PLP). These techniques are used frequently in speech recognition [47]. In this thesis Mel-Cepstral analysis is used to provide robust speech features, and a description is given in Chapter 4.

In Voice Puppetry [21] speech parameters are represented using a mixture of LPC and RASTA-PLP coefficients. Mappings between speech and facial parameters are encoded using a dual mapping entropic HMM [21, 20] - essentially a HMM entropically trained using facial parameters, with state observation probabilities calculated using speech parameters. Synthesis is then achieved by finding the most probable sequence of states through the dual-HMM, and calculating the most likely output facial parameters at each state. In this work facial animation is primarily lip-synch orientated. However, the author makes the interesting observation that appropriate facial expressions also accompany the animations.

In [75] Hong *et al* use a Multi-Layer Perceptron (MLP) neural network to map Mel Frequency Cepstrum Coefficients (MFCC) to visual Motion-Unit-Parameters (MUPs), where MUPs represent different facial deformations. The synthesis from speech is lip-synch orientated. However, the system also allows users to manually incorporate facial expressions into synthesised animations. In [76], Huang and Chen describe two MFCC to visual mapping methods. The first method uses a joint Gaussian Mixture Model (GMM) to correlate audio and visual parameters and provide a mapping between the two. The second method maps an audio HMM to a joint audio-visual HMM, where each audio-visual HMM state is represented as a 3 mixture GMM. Given novel audio, the audio HMM maps to the audio-visual HMMs states, and an audio to visual mapping is calculated in each GMM (using the technique outlined in the first method). Eisert *et al* [57] use a feed-forward neural network to derive MPEG-4 FAPs given LPC speech parameters. The MPEG-4 parameters are then used to control a facial mesh. Li *et al* [97] animate a cartoon face from speech, where the input signal is represented using MFCC coefficients and a mapping between audio and visual parameters is obtained through a GMM mapping. This work also animates facial emotion from

speech, where emotion features are represented using statistics related to speech rhythm, pitch, derivative and slopes. Speech examples are then used to train an emotion classifier for a neutral emotion, happiness, sadness and anger, and corresponding cartoon face templates, with associated expression magnitudes, are defined. Given new speech it is classified according to the probability of it belonging to a particular emotion. These percentages are then used as weights to mix the output facial emotion. Kakumanu *et al* [82] use a look-up table of RASTA-PLP speech segments and corresponding facial articulator values to produce animation, where the look-up table is trained using data captured from a subject reciting sentences from the TIMIT database [61]. Given new audio, segments are extracted and compared to RASTA-PLP values stored in the database, and corresponding values used for animation.

2.6 Evaluating Facial Animation

The quality of automatically synthesised facial animation has been measured using various approaches. These include subjective assessment [22, 40, 64], visual comparison of synthetic versus ground truth animation parameters [145, 32], measurement of a test participants ability to perceive audio in noisy environments with the aid of synthetic animation [40, 116] and through forced choice experimentation [67, 72].

Subjective evaluation is the most common method and typically entails comments on the animations from the designers and a number of naive test participants. The observations of the participants are then supported using example videos of the animations. Visual comparison of synthetic versus ground truth parameters involves comparing the trajectories of speech synthesised animation parameters, with trajectories of ground truth parameters, typically obtained from a real speaker (see Figure 2.8). The first method provides subjective information on the overall quality of facial animations, but leaves no means of comparison with other systems, or no direct method of determining any strengths or weaknesses inherent in the synthesis algorithm. The second method provides more insight into an algorithm's strengths and weaknesses, and a more quantitative measure of a system's overall effectiveness. However, taken on its own it provides no means of communicating the perceptual quality of an animation.

Measurement of the ability of a synthetic talking head to improve the intelligibility of speech in a noisy environment gives a good indication of the quality of an animation (especially lip-synching)

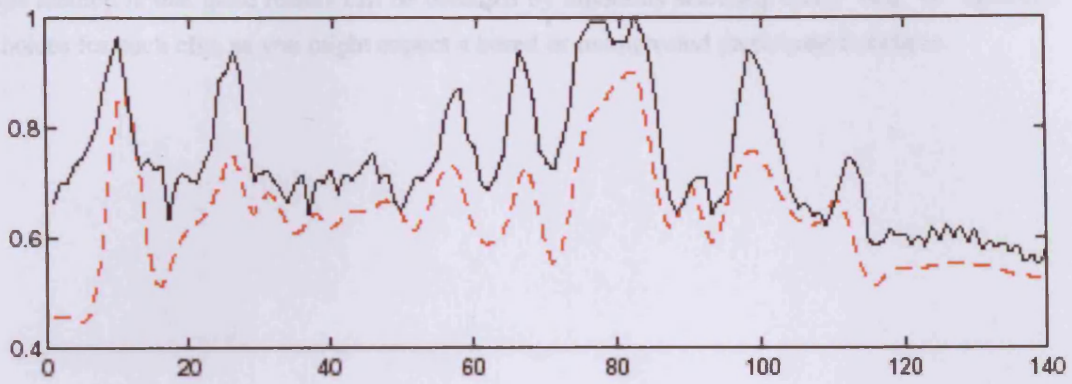

Figure 2.8: Comparison of synthesised animation parameters (dashed line) and ground truth parameters (solid line). In this example the parameters under comparison are appearance parameters. A match between the synthetic and ground parameters suggests good animation quality.

when compared to the performance, in the same circumstances, of real speaker footage. This measure, along with comparisons of synthesized trajectories to ground truths, gives a good overall picture of a talking head's lip-synch ability.

Perhaps the most thorough method used to date to measure the perceptual realism of talking head animation is using forced choice experiments. In the work of Gieger *et al* [67], and Hack and Taylor [72], a series of experiments are carried out where the user is asked to state whether a displayed animation is real or synthetic. If the animations are indistinguishable from real video then the chance of correctly identifying a synthetic animation is 50/50. The test may be thought of as a kind of *Turing Test* for facial animation. Gieger *et al* produce facial animation from phonemes, thus the test determines the quality of lip-synchronization. Animations of the lower part of the face are synthesized and fixed onto real speaker footage to increase the overall impression of realism. In the work of Hack and Taylor it is only facial behavior which is measured (with no audio). Thus forced choice is used in each case to measure different animation characteristics.

A drawback of the forced choice approach is that the participants can develop a *prior* or *opinion* about the animations during testing which may influence their decisions. Any artifacts picked up on in the animations e.g. texture flicker or incorrect co-articulation, cause a participant to develop a picture of what is real and what is not. This is due to the participant's knowledge before the test that some animations will be synthetic and some will be real video footage. A further drawback with

this method is that good results can be obtained by randomly selecting either "real" or "synthetic" choices for each clip, as you might expect a bored or uninterested participant to behave.

Chapter 3

System Overview

This Chapter gives an overview of the procedures used for building and training a hierarchical image-based model, and also for using the model to synthesise novel animations. The overall system may be divided into four separate procedures (see Figure 3.1):

1. Acquisition and Initialisation
2. Analysis and Learning
3. Synthesis and Animation
4. Reconstruction and Display

A brief description of each of these procedures now follows.

3.1 Acquisition and Initialisation

The initial step for building a hierarchical model, and training a visual-speech synthesis model (a SAM or a HMCM), begins with recording a subject speaking front on into a video camera. The hierarchical model is 2D, hence out-of plane head movements are kept to a minimum. Lighting is set up to be as constant across the face as possible (see Section 4.4).

The subsequent recorded video data is then exported into a set of RGB images, and each image is annotated semi-automatically using a tracking algorithm - which defines key-landmark points on the face of the subject (see Section 4.3). Once facial features have been tracked, the images are

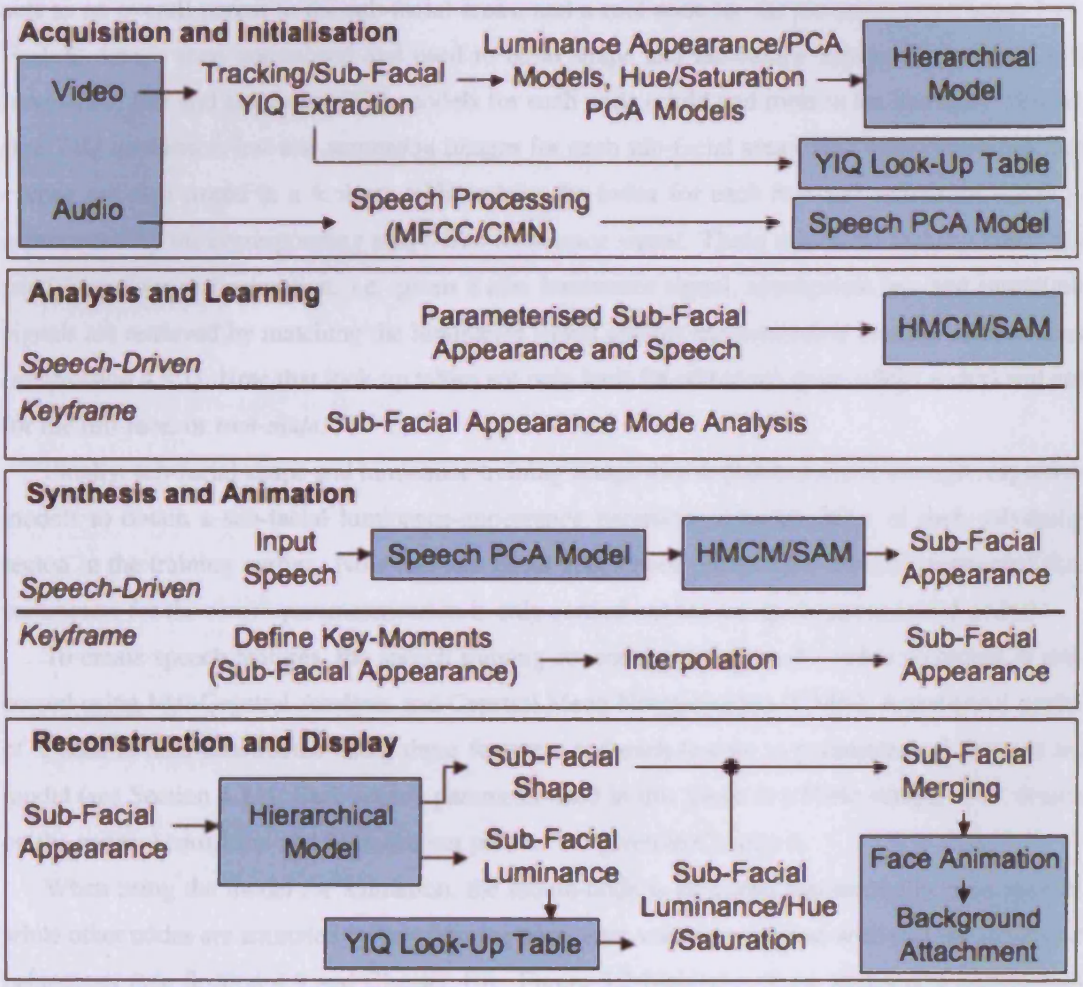

Figure 3.1: An overview of the four major processes for building, training and animating a hierarchical model from speech. The four rows (from top to bottom) correspond to the processes of *Acquisition and Initialisation*, *Analysis and Learning*, *Synthesis and Animation* and *Reconstruction and Display*.

converted into the YIQ colour space (see Section 4.7), and shape and texture data corresponding to different sub-facial regions is extracted from the face. These sub-facial regions (or child nodes) incorporate the mouth, lower-face, eyes and eyebrows. The full face region is also extracted - and acts as an overall parent to the sub-facial areas, and a root node for the hierarchy (see Figure 3.2). Feature data is then normalised and used to build shape and *luminance* appearance models, and *luminance*, *hue* and *saturation* PCA models for each node (child and root) in the hierarchy. Shape-free YIQ *luminance*, *hue* and *saturation* images for each sub-facial area (child node) in the training corpus are also stored in a look-up table, where the index for each *hue* and *saturation* signal is represented by the corresponding shape-free *luminance* signal. These sub-facial look-up tables are used for colour reconstruction, i.e. given a new luminance signal, appropriate hue and saturation signals are retrieved by matching the luminance signal against the luminance look-up table indices (see Section 4.8.1). Note that look-up tables are only built for sub-facial areas (child nodes) and not for the full-face, or *root-node*.

Finally, sub-facial shape and luminance training image data is parameterised through respective models to obtain a sub-facial luminance-appearance parameter representation of each sub-facial region in the training corpus. Note that full facial appearance parameters are not constructed (i.e. parameters for the root) - parameterisation is only carried out for sub-facial areas (child nodes).

To create speech features, the speech training set corresponding to the video recording is processed using Mel-Cepstral Analysis and Cepstral Mean Normalisation (CMN). A statistical model of speech is then constructed using these features, and each feature is parameterised through the model (see Section 4.11). Each speech parameter used in this thesis is a 50Hz sample. Full details on the entire Acquisition and Initialisation process are given in Chapter 4.

When using the model for animation, the mouth-node is animated automatically from speech, while other nodes are animated by key-framing parameter values associated with specific sub-facial behaviours (see Section 4.9 and Chapter 10). Figure 3.2 highlights those nodes of a hierarchical model suited to speech driven animation, and those suited to *key-frame* based animation. Note that there also exists a potential for other nodes to be animated from speech, e.g. the eye-brows. These nodes are therefore also highlighted as potential speech driven nodes in the Figure. Chapter 13 discusses how different speech features may be incorporated into HMCM training in order to facilitate the automatic animation of different sub-facial areas.

Both the face and lower-face act as *dumb* nodes in the hierarchy, in that both are animated only

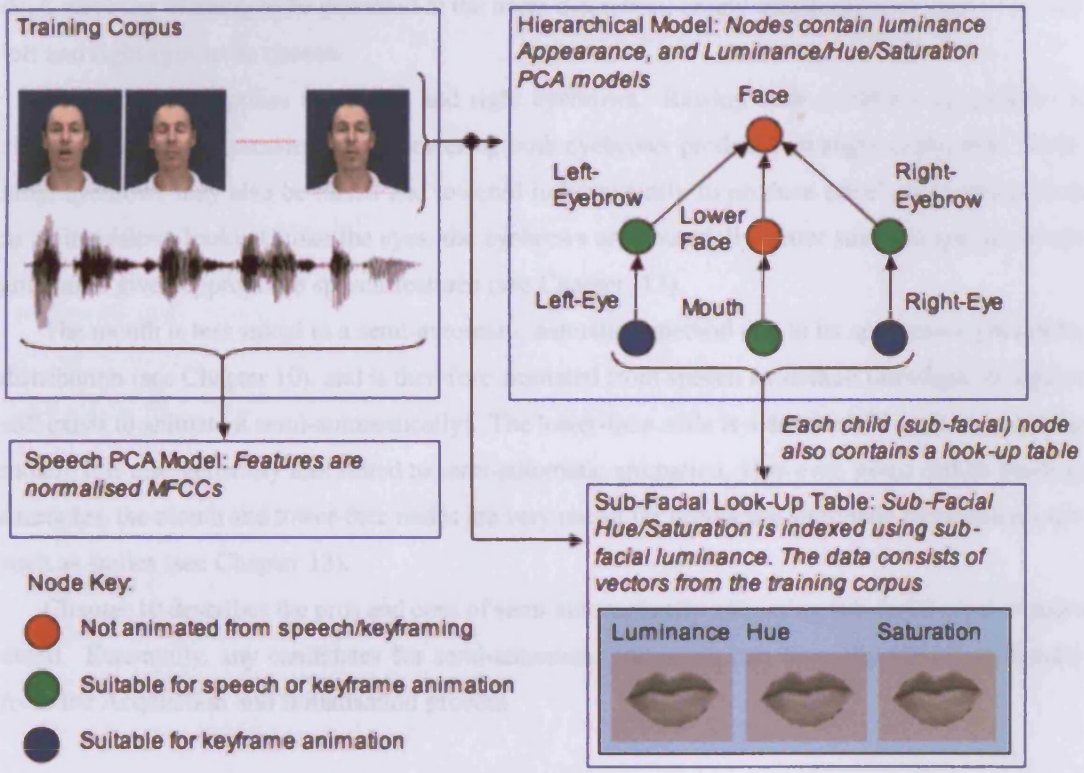

Figure 3.2: An overview of hierarchical model/look-up table initialisation, and node animation suitability

by their children. The mouth therefore animates the lower-face - which in turn animates its corresponding part in the full-face. Any correlation which exists between eye-movement and speech is not specifically modelled in this thesis. Using the models described here-in, animating the eyes from speech often produces unrealistic opening and closing of the eye-lids. Instead, the eyes are more suited to semi-automatic animation. The user may specify eye-parameter values accounting for open eyes and closed eyes (and positions in-between). The left and right eyes are also independent, allowing blinking to be animated at the users discretion, or any combination of open or closed left and right eyes to be chosen.

The same also applies to the left and right eyebrows. Raising both eyebrows can produce a shocked or scared expression, while lowering both eyebrows produces an angry expression. Similarly, eyebrows may also be raised and lowered independently to produce novel expressions (such as an inquisitive look). Unlike the eyes, the eyebrows are potentially better suited to speech-driven animation given appropriate speech features (see Chapter 13).

The mouth is less suited to a semi-automatic animation method due to its appearance parameter distribution (see Chapter 10), and is therefore animated from speech by default (although the option still exists to animate it semi-automatically). The lower-face node is a dumb *child* node. As with the mouth, it is also generally less suited to semi-automatic animation. However, given certain training examples, the mouth and lower-face nodes are very useful for producing simplistic facial behaviours such as smiles (see Chapter 13).

Chapter 10 describes the pros and cons of semi-automatically animating sub-facial areas in more detail. Essentially, any candidates for semi-automatic animation can be easily identified directly from the Acquisition and Initialisation process.

3.2 Analysis and Learning

The analysis and learning step involves training speech driven animation models using parameterised luminance-appearance mouth data and speech data. In order to train synthesis model, corresponding speech and luminance-appearance parameters are required. Two such models are primarily described in this thesis, SAMs and HMCMs. Speech is sampled at 50Hz (50fps), and video at 25Hz (25fps). Therefore, during construction of these models, visual parameters are interpolated from 25Hz to 50Hz to create one-to-one correspondences. The analysis and learning of SAMs and HMCMs is described fully in Chapters 5 and 7.

3.3 Synthesis and Animation

This process concerns the animation of sub-facial areas from either continuous speech, or a predefined semi-automatic schema. Given a new audio recording, sampled at 50Hz, it is first processed using mel-cepstral analysis and parameterised via the speech model. The speech is then projected into an animation model, either a SAM or a HMCM, and appropriate mouth luminance-appearance parameters produced.

Semi-automatic animation initially requires an examination of the modes of appearance variation of each sub-facial area - typically carried out at an early stage (e.g. during Acquisition and Initialisation). This identifies isolated variations in sub-facial areas, the parameters associated with producing a certain type of variation, and the recommended bounds applicable to the parameter. For example, the modes of variation for the left-eye distribution reveal a single parameter, which when varied between certain bounds, causes the eye to open and close. Variation of this parameter can then be used to produce a number of animation effects, such as blinking, winking and talking with the eyes-closed. It is also insightful to study example trajectory dynamics from the training set when preparing such animations in order to improve the emulation of realism.

3.4 Reconstruction and Display

The final process in the system is related to hierarchical reconstruction, and the output of facial animations. At this stage it is assumed that luminance-appearance animation parameters have already been synthesised (by what ever means) for each sub-facial area. Colour information is retrieved by constructing a shape-free luminance image for a sub-facial area (using its appearance parameter value), and then matching this image against the luminance index values in the hue and saturation image look-up tables.

The best matching *luminance*, *hue* and *saturation* images are used to represent an output frame for that sub-facial area, and combining the three shape-free images produces a colour output for a sub-facial image.

To reconstruct a full-face, shape-free sub-facial images, in each colour domain, are merged on a per-frame basis in a top-down manner. The left and right eyebrows are merged with the mean shape-free face first (i.e. the mean face luminance, hue and saturation image). The left and right eyes are then merged over the left and right eyebrows (in the partially completed face). The lower-face is then

merged with the partially completed face, and finally the mouth merged with the lower face. This creates a completed facial image. In order to remove any warping artifacts, and to normalise any lighting variations between sub-facial images, *parent-approximation* is performed on the merged facial image. This involves projecting each facial colour image through its respective PCA model.

Shape is calculated by concatenating sub-facial shape vectors, and offsetting these vectors with respect to the mean face shape. The completed shape-free face image is warped according to this combined new facial shape information.

To increase the static realism of reconstructed facial images they may also be filtered using an un-sharp mask. This compensates for noise lost in images during PCA dimensional reduction. Finally, to further increase the illusion of realism, facial images may also be re-attached to background images in the original training video. The entire reconstruction process is described in Chapter 9. Figure 3.3 gives a graphical overview of this process.

3.5 Summary

The structure of the next six Chapters, in relation to these processes, is as follows:

- Chapter 4 describes Acquisition and Initialisation, i.e. the process of acquiring data, and building a hierarchical model.
- Chapters 5 and 6 relate to the processes of Analysis and Learning, and Synthesis and Animation respectively. The two Chapters focus on using SAMs to create mouth animation parameters from speech.
- Chapters 7 and 8 also relate to both Analysis and Learning, and Synthesis and Animation - but describe these processes in terms of using HMCMs to animate the mouth.
- Chapter 9 relates to Reconstruction and Display, and makes no assumptions concerning whether sub-facial areas are animated from speech, or semi-automatically. Instead, the Chapter describes how synthesised sub-facial regions are merged to create facial images, and how these images may be attached to background images.

The parameter key-framing method for semi-automatic animation, although strictly related to the process of Synthesis and Animation, is described separately in Chapter 10.

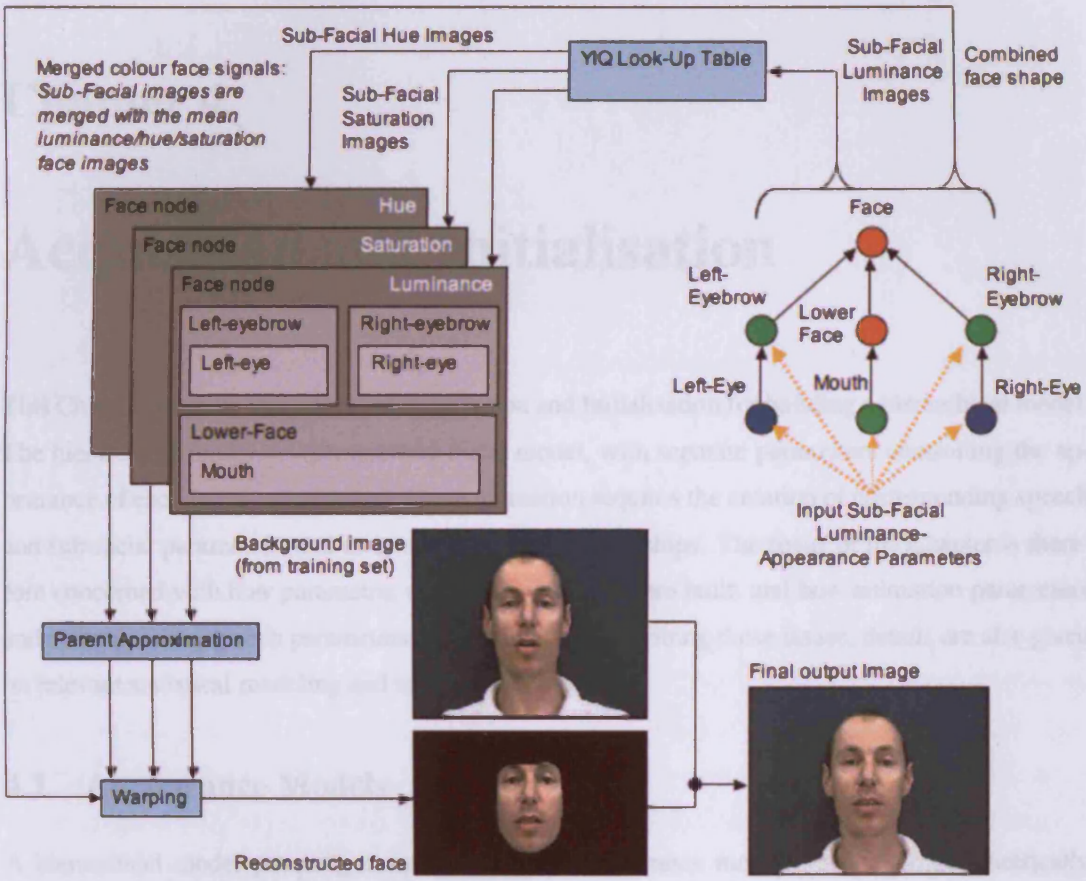

Figure 3.3: Sub-facial (child node) appearance luminance parameters - created from speech or key-framing - are used to synthesise sub-facial luminance images and shape vectors via the hierarchy. The sub-facial luminance images are used to select corresponding hue and saturation images from a sub-facial YIQ look-up table. The face is then reconstructed in a top-down manner for each colour signal by layering sub-facial images over mean (shape-free) luminance, hue and saturation face images. Depending on the hierarchy, sub-facial (child) areas may also be layered over other sub-facial (parent) areas. Parent approximation is performed on each reconstructed full-facial signal, and each signal is warped according to the synthesised combined output facial shape. Synthesised colour images are then combined with background images from the training set to increase realism.

Chapter 4

Acquisition and Initialisation

This Chapter concerns the process of Acquisition and Initialisation for building a hierarchical model. The hierarchical model is a parameteric facial model, with separate parameters controlling the appearance of each facial area. Speech driven animation requires the creation of corresponding speech and sub-facial parameters, and an analysis of their relationships. The focus of the Chapter is therefore concerned with how parametric models for each area are built, and how animation parameters and corresponding speech parameters are obtained. In describing these issues, details are also given on relevant statistical modeling and tracking techniques.

4.1 Appearance Models

A hierarchical model primarily consists of several appearance models [36], each parametrically representing a sub-facial area. In an appearance model, a single *appearance parameter* controls the variation of texture and shape. Training a hierarchical model for animation requires a set of such parameters for each sub-facial area. This Section is therefore concerned with describing the construction of appearance models. Those already familiar with such models should note that several differences exist between constructing classic appearance models [36], and building appearance models for the hierarchical model. These differences are emphasised throughout this Section.

Figure 4.1: Example landmarked images. Note the correspondences between landmarks across the images.

4.1.1 Point Distribution Models (PDM)

Construction of an appearance model begins with the construction of a Point Distribution Model (PDM) [39]. A PDM is a linear model of shape variation, allowing shape to be represented in a parametric and reduced dimensional form. A PDM also allows the generation of new shape vectors, with similar properties to those initially used to train a PDM. In a 2D PDM [1], shape constitutes a vector of image landmark positions $\mathbf{x} = (x_1, y_1, x_2, y_2, \ldots, x_D, y_D)$ where x and y coordinates represent image landmarks, D is the number of landmarks in an image and $\mathbf{x}$ is a column vector (in this thesis all vectors are in column form unless explicitly stated otherwise). Given a set of images, landmarks are placed in correspondence across the entire corpus (see Figure 4.1), to yield a set of 2D dimensional shape vectors.

Landmark placement is very important, and great care should be taken to assign landmarks to regions of consistent and prominent appearance throughout the entire image set. For example, corners provide good landmark placement points, such as those appearing at the edge of the eyes and mouth. However, skin flesh points, such as those found in the middle of the cheeks, are poor placement points.

A PDM captures shape *variation*, i.e. in the example of a face we would be interested in capturing variation which occurs in regions such as the mouth, e.g. when it opens or closes. However,

[1] In this thesis all PDMs are 2D. From this point on the term PDM is therefore used to describe a 2D PDM, unless specifically stated otherwise.

we do not wish to model variation which may occur due to *pose* changes in the face, e.g. when a person moves their face *out-of-plane*.

The shape training set is therefore first normalised with respect to pose and transformed into a common coordinate frame using an alignment algorithm. A popular method to achieve alignment is Procrustes analysis [35] where each shape is aligned such that the sum of the distances of each shape to the mean ($E = \sum_{i=1}^{N}(\mid \mathbf{x}_i - \bar{\mathbf{x}} \mid)$) is minimized, where $\mathbf{x}_i$ is a shape vector, $\bar{\mathbf{x}}$ is the mean shape and N is the number of shape vectors.

When aligning shape data for the construction of a hierarchical model, shape vectors are *not* normalised with respect to *scale*. This is done to preserve certain unique shape characteristics that would otherwise be lost. An example is demonstrated at the end of this Section. In order to reduce potential non-linearities which may emerge as a result of this approach, it is important that the distance between the video camera and the subject recorded for training is kept as stable as possible.

An iterative approach for minimizing E is described in [35], and may be summarised as follows: (note that alignment w.r.t. scale - as described in [35] is omitted from step 4)

1. Translate each shape vector so that its centre of gravity lies at the origin.
2. Choose one vector as an initial estimate of the mean and scale so that $\mid \bar{\mathbf{x}} \mid = 1$.
3. Record this estimate as the vector $\bar{\mathbf{x}_0}$.
4. Align each vector with $\bar{\mathbf{x}_0}$, with respect to translation and rotation only, using Procrustes Analysis.
5. Re-estimate the mean, and align with $\bar{\mathbf{x}_0}$.
6. If the estimate of the mean has not converged then return to step 4. Convergence is declared if the mean does not change significantly between iterations.

Figure 4.2 shows a set of mouth landmarks before and after alignment. Note the reduction in pose variation in the aligned vector set.

The shape vector set now represents a D-dimensional data distribution. By performing PCA on this distribution, variation can be approximated in terms of *principle components*, i.e. orthogonal (and therefore uncorrelated) basis vectors through the distribution. The principle components may be calculated as follows

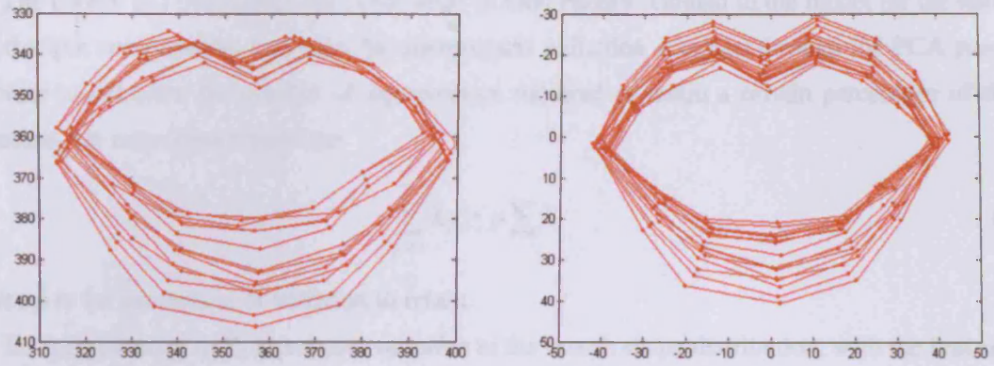

Figure 4.2: A set of mouth landmarks before alignment (left) and after alignment (right). Note how translational pose variation is reduced in the aligned data set.

1. Calculate the mean of the distribution

$$\bar{x} = \frac{1}{s}\sum_{i=1}^{N} \mathbf{x}_i \tag{4.1}$$

2. Calculate the covariance of the distribution

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{4.2}$$

3. Calculate the eigenvectors ϕ and eigenvalues λ of $\mathbf{S}$, sorted so that $\lambda_i > \lambda_{i+1}$, i.e. in descending order of energy.

Step 3 may be achieved by performing a Singular Value Decomposition (SVD) on the matrix $\mathbf{S}$ [111]. Completion of this process allows the distribution to be represented as a linear model

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_x \mathbf{b}_x \tag{4.3}$$

where $\mathbf{P}_x$ contains the first t eigenvectors and $\mathbf{b}_x$ is a shape parameter represented as

$$\mathbf{b} = \mathbf{P}_x^T (\mathbf{x} - \bar{\mathbf{x}}) \tag{4.4}$$

Variation of $\mathbf{b}_x$ allows new shapes to be defined. A constraint of $\pm 2\sqrt{\lambda_i}$ (2 standard deviations) may be applied to the elements b_i of $\mathbf{b}_x$ to ensure that constructed shapes are similar to those present in the training set.

The choice of t determines the percentage of total energy retained in the model (or the number of principle components), and also the dimensional reduction acquired through the PCA process. In order to calculate the number of eigenvectors required to retain a certain percentage of shape variation, we may chose t such that

$$\sum_{j=1}^{t} \lambda_j \geq p \sum \lambda_i \tag{4.5}$$

where p is the percentage of variation to retain.

Each eigenvector in $\mathbf{P}_x$ represents variation in the mouth shape distribution, with the first eigenvector, or highest mode of variation, representing the highest percentage of variation. Figure 4.3 shows the first 2 modes of variation from a training set of 100 left-eyebrow shape vectors, calculated by varying the elements of $\mathbf{b}_x$ between $\pm 2\sqrt{\lambda_i}$ from the mean shape. In the example, 98% of the total model energy is retained. Note that the left-eyebrow PDM also contains a small number of anchor points (not directly assigned to the eyebrow itself). These give positional information about the eye-brow relative to fixed points on the face, and are used to ascertain whether the eye-brow is either raised or lowered.

The left eyebrow PDM provides a perfect example of the importance of why *not* to align the shape of certain facial areas with respect to scale. Figure 4.4 shows the same PDM constructed *with* scale alignment. Note how variation associated with raising and lowering the eyebrow is lost. This is because when the eyebrow raises or lowers, overall landmark shape stays constant - the major variation being due only to the translation of the anchor points. Aligning with respect to scale corrupts the position of these anchors, and as such removes the raising and lowering variation from the PDM.

In relation to facial animation we may animate a PDM by varying elements of the parameter $\mathbf{b}_x$. Conversely, recorded facial shape data produces a trajectory through PDM space, which proves useful in understanding the relation between animation parameters (such as $\mathbf{b}_x$) and speech parameters.

4.1.2 Statistical Models of Texture

After PDM construction, the next stage in building an appearance model is to construct a statistical model of texture - referred to in this thesis as a Grey Level Model (GLM). Shape normalised textures – called *shape-free patches* – are first constructed by warping each image texture from its landmark

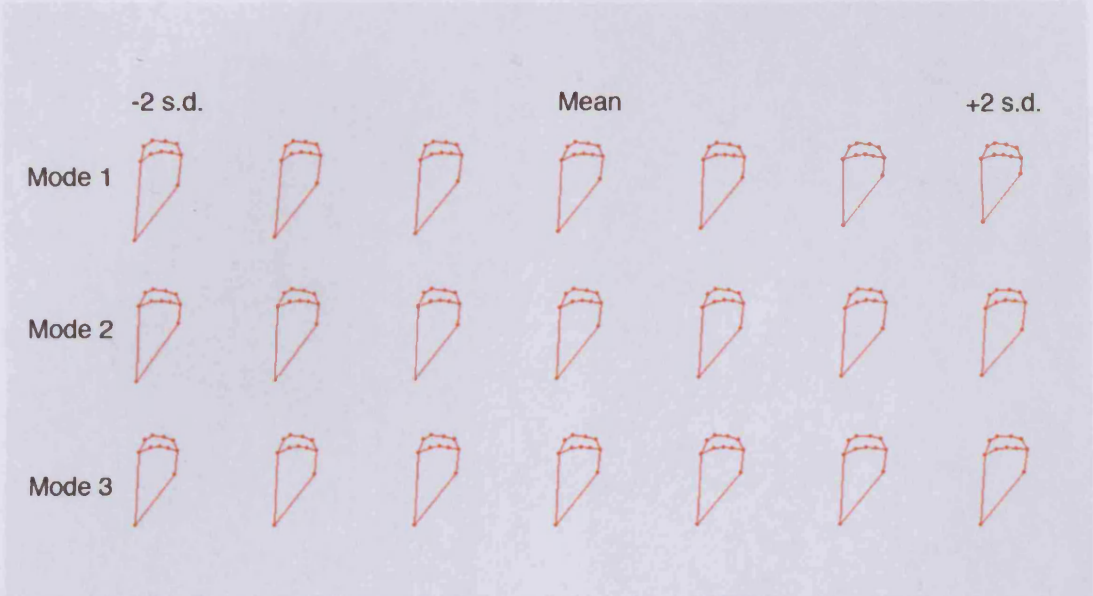

Figure 4.3: First 3 modes of shape variation in a left eyebrow PDM. Modes are offset 2 s.d. from the mean shape.

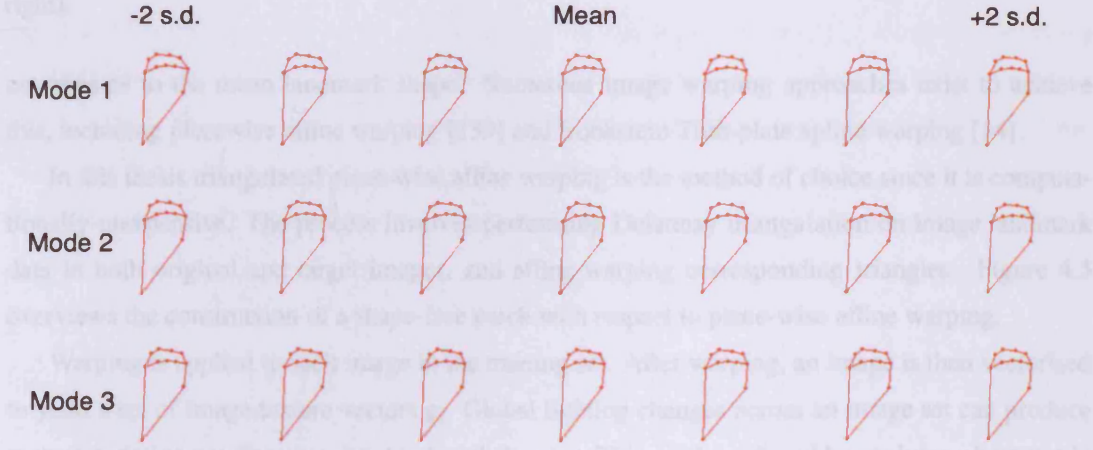

Figure 4.4: First 3 modes of shape variation in a left eyebrow PDM *with* scale alignment. Modes are offset 2 s.d. from the mean shape. Note that variation related to raising and lowering the eyebrows is lost.

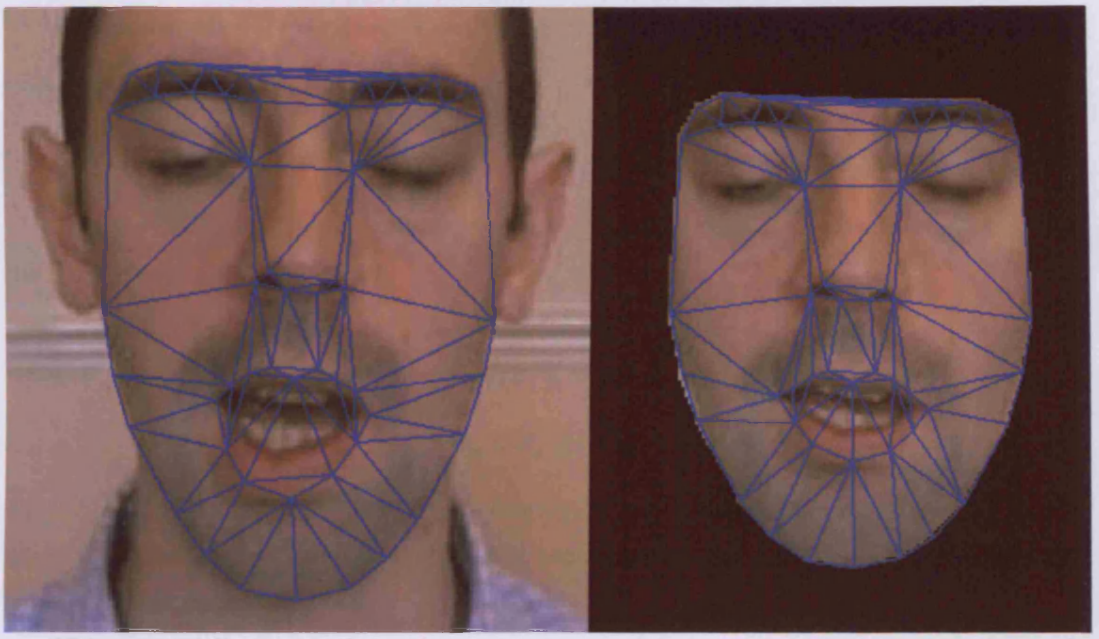

Figure 4.5: Constructing a shape-free patch. Landmarks on a training image (left) and in the mean shape (shown projected onto the image on the right) are first triangulated. The texture in each triangle in the training image (left) is then warped to its corresponding position in the mean shape. Repeating this for each triangle produces a shape-free version of the training image (shown on the right).

coordinates to the mean landmark shape. Numerous image warping approaches exist to achieve this, including piecewise affine warping [139] and Bookstein Thin-plate spline warping [14].

In this thesis triangulated piece-wise affine warping is the method of choice since it is computationally inexpensive. The process involves performing Delaunay triangulation on image landmark data in both original and target images, and affine warping corresponding triangles. Figure 4.5 overviews the construction of a shape-free patch with respect to piece-wise affine warping.

Warping is applied to each image in the training set. After warping, an image is then vectorised to yield a set of image texture vectors $\mathbf{g}_i$. Global lighting changes across an image set can produce texture variation not directly related to facial changes. This can be reduced by applying photometric normalisation, following the procedure described in [36]:

1. Chose a texture from the texture training set as an initial estimate of the mean texture $\bar{\mathbf{g}}$.

2. For each texture $\mathbf{g}_i$ calculate a scaling value $\alpha = \mathbf{g}_i.\bar{\mathbf{g}}$ and an offset value $\beta = (\mathbf{g}_i.\mathbf{1})/n$ (where

n is the number of elements in $\mathbf{g}_i$ and $i = 1,\ldots,N$), and recalculate $\mathbf{g}_i = (\mathbf{g}_i - \beta.\mathbf{1})/\alpha$.

3. Calculate a new estimate for the mean $\bar{\mathbf{g}}$.

4. Repeat from step 2 until $\bar{\mathbf{g}}$ converges, i.e. there is little change in the mean between iterations.

Once the vector set has been normalised PCA is then applied to the texture training set giving the linear model

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \tag{4.6}$$

where $\mathbf{P}_g$ contains the first t eigenvectors, and $\mathbf{b}_g$ is a texture parameter vector such that

$$\mathbf{b}_g = \mathbf{P}_g^T(\mathbf{g} - \bar{\mathbf{G}}) \tag{4.7}$$

As with a PDM, variation of $\mathbf{b}_g$ allows the creation of new texture under a constraint of $\pm 2\sqrt{\lambda_i}$, where λ_i are the first t eigenvalues. The desired percentage of total variation retained in the model can again be estimated using (4.5).

PCA is especially useful when using image texture since vectors can be extremely large. As with a PDM, facial animations may also represent trajectories through GLM space.

4.1.3 Combining Shape and Texture

Shape and texture may now be represented using the parameters $\mathbf{b}_x$ and $\mathbf{b}_g$. However, assuming that both the PDM and GLM are constructed using the same image set, then correlations will exist between the two models. Appearance models encode these correlations and allow related shape and texture vectors to be represented using a single appearance parameter $\mathbf{c}$.

Shape and texture parameters are first parameterised using 4.4 and 4.7, yielding the vector set

$$\mathbf{b}_i = \begin{bmatrix} \mathbf{W}\,\mathbf{b}_x^i \\ \mathbf{b}_g^i \end{bmatrix} \tag{4.8}$$

where $i = 1,\ldots,N$ and N is the number of vectors in the training set. The matrix $\mathbf{W}$ is a diagnonal matrix of elements r, which scales the parameters $\mathbf{b}_x^i$ to lie within the same range as the parameters $\mathbf{b}_g^i$. One method to find r is to entropically scale the values of $\mathbf{b}_x$ [18]. However, a simpler and comparibly effective alternative is to scale the shape parameters using the variance energy ratios of the PDM and GLM.

$$r = \sqrt{\frac{\sum \lambda_i^x}{\sum \lambda_i^g}} \tag{4.9}$$

Performing PCA on this new set of vectors results in a linear appearance model

$$\mathbf{b} = \mathbf{Q}\mathbf{c} \tag{4.10}$$

where $\mathbf{Q}$ contains the first t eigenvectors. Note that unlike the PDM and GLM, there exists no mean offset from the basis vectors $\mathbf{Q}$. This is because the mean is a vector of zero element values (assuming correct normalisation of the shape and texture vector sets). The vector $\mathbf{c}$ is an appearance parameter, the variation of which can be used to produce new values for $\mathbf{x}$ and $\mathbf{g}$.

The matrix $\mathbf{Q}$ consists of both shape and texture related elements, and may be separated to form matrices $\mathbf{Q}_x$ and $\mathbf{Q}_g$. Thus

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x \\ \mathbf{Q}_g \end{bmatrix} \tag{4.11}$$

where $\mathbf{Q}_x$ is of dimension j by t, $\mathbf{Q}_g$ is of dimension k by t, t is the number of eigenvectors in $\mathbf{Q}$, j is the number of eigenvectors in $\mathbf{P}_x$ and k is the number of eigenvectors in $\mathbf{P}_g$. Both $\mathbf{x}$ and $\mathbf{g}$ may now be written as functions of $\mathbf{c}$ using

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_x\mathbf{W}^{-1}\mathbf{Q}_x\mathbf{c} \tag{4.12}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{Q}_g\mathbf{c} \tag{4.13}$$

To synthesise an output image, $\mathbf{c}$ is first used to calculate values for $\mathbf{x}$ and $\mathbf{g}$. The texture $\mathbf{g}$ is then warped from the mean shape $\bar{\mathbf{x}}$ to the new shape $\mathbf{x}$.

Shape and texture vectors are related to $\mathbf{c}$ such that

$$\mathbf{c} = \mathbf{Q}^T\mathbf{b} \tag{4.14}$$

As with a PDM and a GLM, a facial animation may now be regarded as a trajectory through appearance parameter space. Figure 4.6 shows an example trajectory through a distribution of appearance parameters, while Figure 4.7 demonstrates example images resulting from key positions along the trajectory - marked A, B, C and D. Chapters 5, 6, 7 and 8 demonstrate how creating a new animation from speech may be regarded as creating new trajectories through such a space following a set of constraints imposed by the speech.

4.2 Implications of using Appearance Models for Facial Animation

An appearance model built in this manner is described in this thesis as a *flat-appearance model*, and allows a face to be animated by varying the parameter $\mathbf{c}$. However, when building an appearance

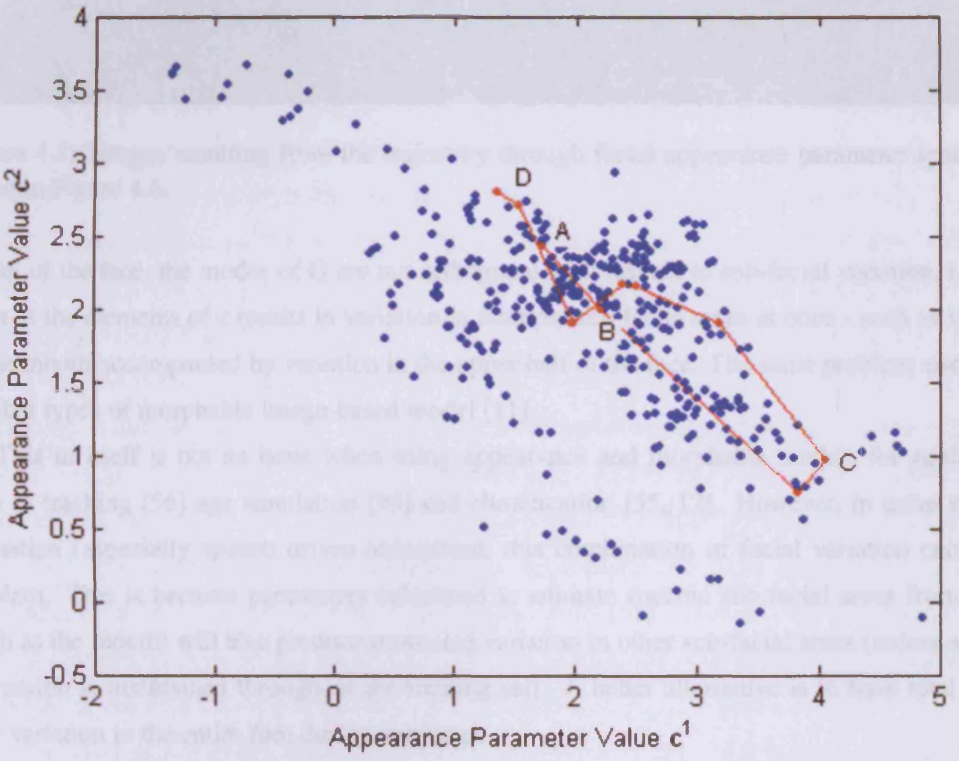

Figure 4.6: A short facial image trajectory through an appearance parameter distribution. Coordinates are represented using parameters responsible for the two highest modes of appearance variation. The trajectory is labelled with four points: A, B, C and D. Facial images resulting to parameter values at these points may be found in Figure 4.7.

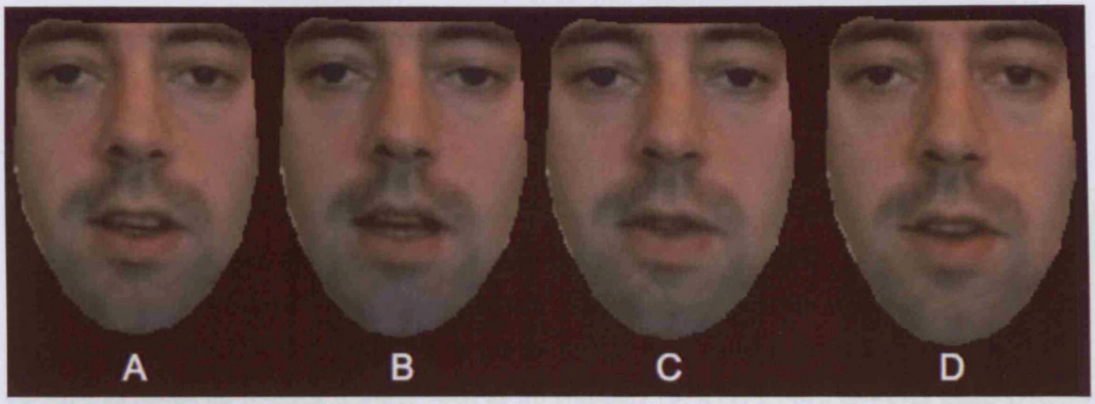

Figure 4.7: Images resulting from the trajectory through facial appearance parameter space illustrated in Figure 4.6.

model of the face, the modes of $\mathbf{Q}$ are not orthogonal with respect to sub-facial variation, i.e. variation of the elements of $\mathbf{c}$ results in variation in multiple sub-facial areas at once - such as variation of the mouth accompanied by variation in the upper half of the face. The same problem also occurs in other types of morphable image-based model [11].

This in itself is not an issue when using appearance and morphable models for applications such as tracking [56] age simulation [95] and classification [55, 12]. However, in terms of facial animation (especially speech driven animation), this combination of facial variation can pose a problem. This is because parameters calculated to animate specific sub-facial areas from speech (such as the mouth) will also produce unwanted variation in other sub-facial areas (unless a neutral expression is maintained throughout the training set). A better alternative is to have total control over variation in the entire face during animation.

The hierarchical model proposes a solution to this problem by constructing appearance models for each sub-facial area. Examination of the modes of each sub-facial model then yields parameters which isolate useful kinds of isolated variation. A full description of a hierarchical model is given in Section 4.4.

4.3 Automatic Landmark Placement

In order to robustly train a hierarchical model of the face, a training set consisting of thousands of video frames is required. Each node in a hierarchical model consists of a PDM, three colour

channel PCA models, and an appearance model. In order to build these models, each image in the hierarchical model training set must be annotated with landmarks. For small models, built using only a few hundred samples, landmarks can be placed manually on training images in a matter of hours. However, it is unrealistic to place landmarks manually on thousands of images. An automatic method for assigning landmarks across a large set of images is therefore required.

Automatic landmark placement is a difficult task. Most facial tracking systems reliably place only a handful of feature points on each image [17][128], or represent different facial areas as thresholded blob regions [137][114]. Other tracking algorithms use geometrically constrained models such as parabolas to track features such as lip contours [87][117][148], however these systems are typically not accurate enough for reliable landmark placement as the templates used for tracking are symmetric - which is inaccurate with respect to facial areas such as the mouth.

Some systems do exist for generic landmark placement, and are considered next. Cootes *et al* [35] describe an Active Shape Model (ASM) for feature placement and classification. A PDM trained on an example set of images is fitted to new instances by iteratively updating the parameter $\mathbf{b}_x$, and a set of pose parameters, until convergence. New landmark positions are chosen at each iteration by examining the texture normal to each landmark and identifying the point of greatest gradient change. Projection of these points through the PDM, and subsequent constraint of the parameters for $\mathbf{b}_x$, results in a candidate for convergence. The ASM approach has been shown to work well when locating objects with a rigid texture appearance over time. However, when tracking features such as lip contours, where lip appearance constantly changes due to deformation and lighting changes (such as the appearance of shadows beneath the bottom lip), the ASM proves less robust. An approach to placing landmarks on lips which directly addresses this problem is proposed by Luettin *et al* in [104][103]. In this work PDM parameters are iteratively estimated by minimizing the error between a global landmark texture model and texture extracted from the image beneath. Parameters that minimise the error may be found using an optimisation algorithm such as Downhill Simplex Minimisation (DSM) [129].

A popular approach to automatically placing landmarks across the entire face, using a method called Eigen-points, is proposed by Covell [45]. In this approach, the location of features such as the mouth and eyes are first estimated using either template or model-based matching. Control points are then placed around that feature (such as along the contours of the lips) using an affine manifold model which couples grey-scale values with control-point locations. Walker *et al* [155]

build appearance models by automatically placing landmarks at salient points across a set of example images ,where salient points are defined as those points which have a low probability as being mis-classified as another feature (e.g. lip-corners). However, the method returns incorrectly placed landmark points given long sequences with high variation, making it inappropriate for building a hierarchical model.

An alternative method for automatic landmark placement is to use the Active Appearance Model search algorithm proposed by Cootes *et al* [36]. In this work an appearance model is fitted to an image by finding the appearance parameter **c** which creates a synthetic image as close to the target image as possible. This is achieved by learning how differences in pose and appearance parameter values, between synthetic images and training images, change in relation to changes in image texture. During training, the appearance parameter for a synthetic representation of an image, as well as the pose of the synthetic image, is deviated by fixed amounts and the subsequent change in texture recorded. This is done for various pose and appearance parameter changes across a set of training images. Linear regression is then used to build matrices which estimate how pose values and appearance values should be changed given a difference in synthetic texture and image texture. The matrix is used in iterative steps until convergence is met. The drawback with this method is that an active appearance model requires construction before searching can begin, and frequent updating as new images are landmarked. This is a time consuming process, although it has been successfully used in several works [145][72].

It should be noted that automatic landmark placement is important in the context of this work since the hierarchical model being built is essentially a collection of appearance models - which rely on landmark data to construct shape free patches and represent shape vectors. Ezzat *et al* construct a morphable model [63] using optical flow data to represent shape, and do not require their training set to be automatically landmarked. They assume planar perspective deformations between images, and solve for the perspective warp by first using optical flow to find correspondences between images, and then using least-squares to solve the underdetermined set of equations. The end result of this process is ultimately the same achieved by landmarking the image set, i.e. that all images can then be put into correspondence with each other, and texture variation will then only assumed to be caused by facial feature movement and not head movement.

4.3.1 Downhill Simplex Landmark Placement

The landmark placement method used in this thesis is a modified version of the DSM method described by Luettin *et al* [103]. This Section first describes the original method before discussing its drawbacks, and how it may be modified to provide more robust tracking.

Perhaps the most difficult part of the face to automatically landmark with both accuracy and reliability is the mouth. This is due to the extreme variability it exhibits. In one frame the mouth may be fully closed, and over two preceding frames may move to fully open. In order to retain accuracy and reliability, a successful tracker needs to be able to deal with extreme pose changes, and also be able to re-initialise itself if it looses track of its target. The bottom lip is perhaps the most difficult part of the mouth to track. This is for two reasons: The boundary between the bottom lip and the skin around the mouth often becomes extremely weak, and is on occasion even difficult to identify by a human observer. Also, protrusion of the bottom lip often causes a shadow to be cast beneath it, which is easy for a tracker to confuse with the lip boundary itself (since it forms a prominent change in image gradient). The eye is also often difficult to track - since a fully open eye and a fully closed eye have a completely different appearance (similar to a fully open mouth and a fully closed mouth). It should be noted, that in finding a solution to tracking the mouth, tracking of the eye is also solved.

Luettin *et al's* DSM tracking method is now described. Given an initial training set of landmarked images, these are used to build a global profile texture model. For each image, the texture *normal* to each landmark is measured and concatenated to form a global profile vector (see Figure 4.8). This is repeated such that $\mathbf{p}_i$ is a global profile vector, where $i = 1,\ldots,N$ and N is the number of images in the training set. Performing PCA on the global profile training set yields the model

$$\mathbf{p} = \bar{\mathbf{p}} + \mathbf{P}_p \mathbf{b}_p \tag{4.15}$$

where $\mathbf{P}_p$ are the first t eigenvectors of the global profile training set, and $\mathbf{b}_p$ is a global profile parameter. The premise behind the global profile model is that it correlates the texture beneath one landmark with the texture beneath the others. For example, if the mouth is open, and teeth visible, then the texture bordering a landmark on the top lip will be correlated with texture bordering a landmark on the bottom lip.

Given a global profile model, and a corresponding PDM, the tracking problem is then defined as finding the model parameters $\mathbf{b}_x$ and pose parameters t_x, t_y, θ and σ (corresponding to x and y

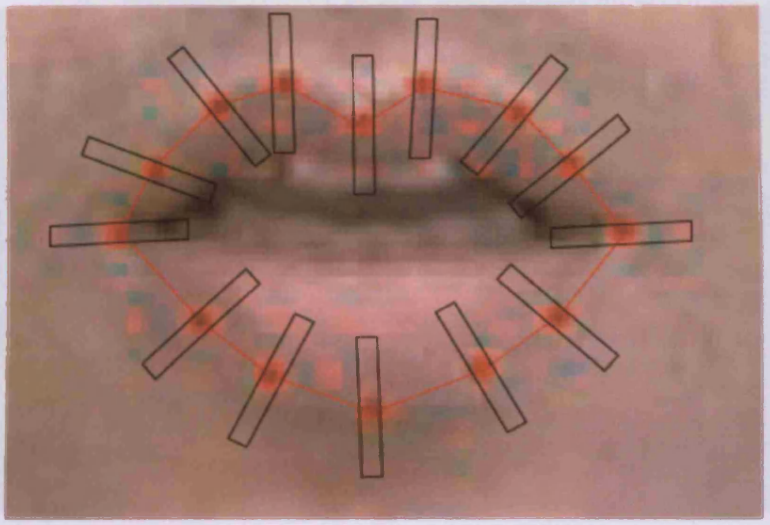

Figure 4.8: Texture samples *normal* to image landmarks are concatenated to form a global profile texture vector. The Figure illustrates example texture sample regions normal to landmarks placed around a mouth image. Hence, the global profile vector contains information related to the entire outline of the mouth.

translation, rotation and scale), that minimize the error function

$$E = (\mathbf{p}_{Im} - \bar{\mathbf{p}})^T (\mathbf{p}_{Im} - \bar{\mathbf{p}}) - \mathbf{b}^{Im\,T} \mathbf{b}^{Im} \tag{4.16}$$

where

$$\mathbf{b}^{Im} = \mathbf{P}_p^T (\mathbf{p}_{Im} - \bar{\mathbf{p}}) \tag{4.17}$$

and $\mathbf{p}_{Im}$ is a concatenated vector of landmark profiles sampled from beneath the current position of the model in the image. These parameters are discovered iteratively using the DSM algorithm, and convergence declared when E stabilises, or the DSM algorithm exceeds a predefined maximum number of iterations. To summarise, the value of E for each iteration of the DSM algorithm is calculated as follows:

1. Using the current values for $\mathbf{b}_{shape}$, t_x, t_y, θ and σ generate a shape vector $\mathbf{x}$ using the PDM and project it into the current image using the pose variables.

2. Sample the image below each landmark normal to the profile boundary and concatenate these to form the vector $\mathbf{p}_{im}$.

Figure 4.9: Mouth tracking using standard DSM approach. Top row (from left to right): Landmark placement is initially satisfactory. However, as the mouth begins to close tracking begins to fail. Bottom row (from left to right): The DSM error becomes stuck in a local minima as the mouth begins to close further, resulting in misplacement of landmarks. In this example, the mouth finally begins to open again re-initialising the tracker.

3. Using (4.16) calculate $\mathbf{b}^{Im}$.

4. Calculate the error E using (4.15).

The initial search parameters may be manually defined, and for future frames the solution to the previous frame provides a reasonable starting position. This method is successful in tracking mouths given certain limits in mouth variation between images. However, given large variations, such as tracking an mouth in one frame and then tracking a closed mouth in the next, the search often fails - becoming stuck in a local minima. This is a symptom primarily of the DSM algorithm itself. Figure 4.9 highlights the problem, showing iterations in an attempt to track two mouths, one fully open and one partially open.

This flaw makes the search unreliable when tracking thousands of frames, as becoming stuck in a local minima in one frame affects tracking performance in preceding frames - often causing the model to loose track of its target completely for hundreds of frames. The search is therefore modified in this thesis, making it more robust to variation and more accurate in general landmark placement.

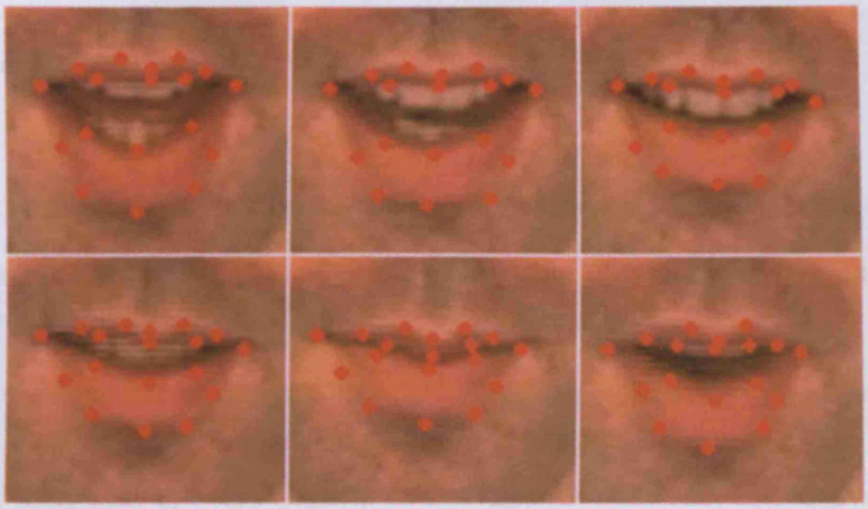

Figure 4.10: Mouth tracking using the modified DSM approach. Top row (from left to right): Landmark placement is satisfactory as the mouth moves from fully open to partially closed. Bottom row (from left to right): Tracking continues successfully as the mouth moves from partially open, to fully closed, and again to partially open. Note that unlike in standard DSM tracking (Figure **??**), the algorithm avoids local minima by tracking each frame using several different starting positions. A combination of this approach, along with a *square* based texture sampling scheme, also improves the general accuracy of landmark placement.

4.3.2 A Modified DSM approach to Automatic Landmark Placement

The modification applied to the basic DSM algorithm is based on repeating the search multiple times for each image using different initial search parameters. The most obvious initial parameters are those from the previous frame's solution. The other initial positions constitute a selection of k parameters $\mathbf{b}_x$ extracted from the existing training set, and chosen using a *k-means* search (initialised using a random seeding). Thus, each parameter represents a different mouth shape, e.g. a closed mouth, an open mouth, a mouth with pursed lips, etc. The algorithm is therefore less likely to become stuck in a local minima, since it is probable that one of the initial sets of search parameters will already be close to the final solution.

To improve robustness further, the texture around a landmark is sampled in a square region as opposed to along a straight line normal to the boundary. This makes the error measure more robust to changes in normal direction. The square region is then raster scanned to form a local profile vector for that landmark. Figure 4.10 demonstrates the advantage of using the modified DSM approach to

overcome large shape changes between frames. The complete modified process may therefore be defined thus

1. Set $E_{best} = \infty$ (i.e. some large number).
2. Let the parameters $\mathbf{b}_{shape}$, t_x, t_y, θ and σ equal the solution values for the previous frame.
3. Calculate new values for $\mathbf{b}_{shape}$, t_x, t_y, θ and σ using the DSM algorithm. The modified error term E is now calculated thus:
   (a) Using the current values for $\mathbf{b}_{shape}$, t_x, t_y, θ and σ generate a shape vector $\mathbf{x}$ through the PDM and project it into the current image using the pose variables.
   (b) Sample an r by r square region from the image below each landmark. Raster scan these regions and concatenate these to form the vector $\mathbf{p}_{im}$.
   (c) Using (4.16) calculate $\mathbf{b}^{Im}$.
   (d) Calculate the error E using (4.15).
4. If $E < E_{best}$ then set $E_{best} = E$ and record the values for $\mathbf{b}_{shape}$, t_x, t_y, θ and σ.
5. Choose one of the k alternative starting vectors for $\mathbf{b}_x$ and repeat steps 3-4 until all k vectors have been exhausted.
6. The tracking solution is that which gives the lowest value E_{best}.

The above method will accurately landmark a single image given a well trained PDM, and a well trained global profile model. These may be obtained using a bootstrapping approach. A complete set of images may therefore be landmarked using the following strategy

1. Initially label a set of images with landmarks.
2. Build a PDM and a global profile model using the current labelled image set. Select k parameters $\mathbf{b}_x$ from the PDM training set using *k-means*. These represent alternative starting positions.
3. Automatically landmark a set of images using the modified DSM tracking procedure.
4. Manually check landmark placement and correct any errors by hand.

5. Rebuild the PDM and the global profile model, and chose a new set of k starting parameters using *k-means*.

6. Repeat steps 3-5 until the entire image set is annotated.

As a guide, good performance can be obtained by initially landmarking a set of approximately 40 to 100 images by hand, and then automatically landmarking sets of between 50 and 100 images. As performance increases and tracking errors reduce, the set of automatically landmarked images can be increased until supervision is no longer required. The choice of a value for k is somewhat dependent on the differences in variation between shape in the initial part of the training set and the remainder of the training set. If the beginning of the training set includes examples of highly varying shape then a large value for k, chosen after landmarking an initial image set, may suffice for the remainder of the procedure. However, if the initially landmarked frames exhibit low variation then it is sensible to initially choose k to be small. This value can then be increased further as the PDM training set becomes larger.

The value r is best found through experimentation. A large value for r often yields lower errors and better overall performance. However, this also directly affects the time it takes to landmark a single frame. The same observation also applies to the choice of value for k.

4.4 Capturing the Hierarchical Model Training Corpus

This Section describes the data collection process for building the hierarchical models used in this thesis. Building a 2D morphable model or appearance model requires video footage of a subject speaking into a camera. The quality of synthetic facial animation is typically proportional to both the length and content of the training video [22][145][62][40][21]. For example, in the works of Bregler *et al* [22] and Theobald *et al* [145] animations are constructed by piecing together parts of the original training set - the quality of animation is therefore proportional to the size of the trained database or code-book. Bregler *et al* [22] and Brand [21] record approximately 8 minutes and 3 minutes (respectively) of video footage where a subject is asked to recite a variety of childrens stories and fairytales. Thus the training emphasis is placed on capturing long segments of natural uninterrupted speech. Theobald *et al* [145] capture 12 minutes of footage of a subject reciting 279 sentences specifically chosen to include as many combinations of phoneme transitions as possible. In the work of Ezzat *et al* [62] the training set consists of 15 minutes of a subject speaking 1 and 2

syllable words together with a list of sentences, while Cosatto *et al* [40] capture a corpus consisting of a subject speaking 14 phrases, each 2 to 3 seconds long. As well as training set differences between different models there are also variabilities in the sampling rates using to capture video and audio data. Cosatto *et al* [40] use a frame rate of 60 fps, while Ezzat *et al* [62] and Brand [21] use standard 29.97 fps NTSC sampling rates.

When building 2D image-based models, minimization of out-of-plane head variation is important so as not to introduce non-linearities into the training set. This is handled in different work using several different approaches. These include tracking using a 3D based model [40][21], using a head mounted camera [145], or simply by requesting the subject remain as still as possible during recital of the training speech [62][22]. Lighting set up is also important to avoid undesirable texture variation artifacts. Typical solutions involve setting up front lighting using lamps [40][62].

In this thesis two hierarchical models were trained for evaluation. Each model was trained on a different participant, and consisted of different node structures. Participant 1 was recorded for approximately 13 minutes speaking front on into a standard interlaced Digital Video (DV) camera, using a sampling rate of 40ms (25fps). Audio was recorded in 16 bit mono using a sampling rate of 32KHz. The DV recording was converted into an uncompressed Quicktime movie and exported into a set of JPEG images of dimension 361 $\times$ 289 pixels using Quicktime Pro. Audio was represented uncompressed in WAVE format. Participant 2 was recorded for approximately 5 minutes using the same capture and data representation methods as participant 1.

The content of the training sets for both participants was chosen as a combination of long segments of uninterrupted speech, non-verbal articulations such as whistling and coughing, and single words. In order to capture naturalistic speech both subjects were also asked to recite a children's fairy-tale from memory, thus introducing natural pauses and hesitations into the recording. Both participants were recorded under front-lighting conditions, and were asked to remain as still as possible so as to avoid out-of-plane head movement.

The hierarchical models resulting from the capture of participant 1 and participant 2, are simply named *Hierarchical Model 1* and *Hierarchical Model 2* in this thesis.

4.5 Hierarchical Model Landmarking Strategies

After editing, the training corpus for model 1 consisted of 13012 images in total, equating to approximately 8 minutes and 40 seconds of video. The final corpus for model 2 consisted of 8295

images, equating to approximately 165 seconds of video. For each model, different facial regions were tracked and landmarked separately in order to reduce the number of search parameters being optimised during each iteration. Figure 4.11 shows a landmarked image with the individually tracked facial regions highlighted. Note that the landmarks positioned at the bottom of the left and right ears were used when tracking both the jaw and the eye-corners. Table 4.1 lists the number of landmarks placed on each region, the choice of values for r and k used for tracking different facial areas in each subject corpus, and the number of frames used to bootstrap each facial area tracking model, i.e. the number of frames tracked before supervision was no longer necessary.

Note that there are no *perfect* tracking parameters, i.e. parameters which might work for one subject, will not necessarily work given another. The parameters given in Table 4.1 are also not necessarily *optimum* parameters, and it is entirely likely that given more experimentation better parameters could have been found. When tracking, the best strategy is to initially experiment with values for k and r until a good compromise is found.

Table 4.1: Facial Area Tracking Parameters

Region	Landmarks	M1: No. Manual	M1 - r, k	M2: No Manual	M2 -r, k
Mouth	9	300	41, 5	100	31, 4
Eyes	4	280	41, 2	100	21, 3
Nose	6	140	41, 8	10	21, 1
Jaw	13	100	51, 3	100	21, 3
Left-eye	9	20	51, 1	10	21, 1
Right-eye	9	20	51, 1	10	21, 1

4.6 Hierarchical Model Structure

Given sets of fully landmarked training images, hierarchical nodes can now be identified for each model and extracted. Figure 4.12 shows the facial areas chosen to represent each node in each hierarchical model, along with their position in the hierarchy. Note that these areas do not necessarily map to the individually tracked facial regions.

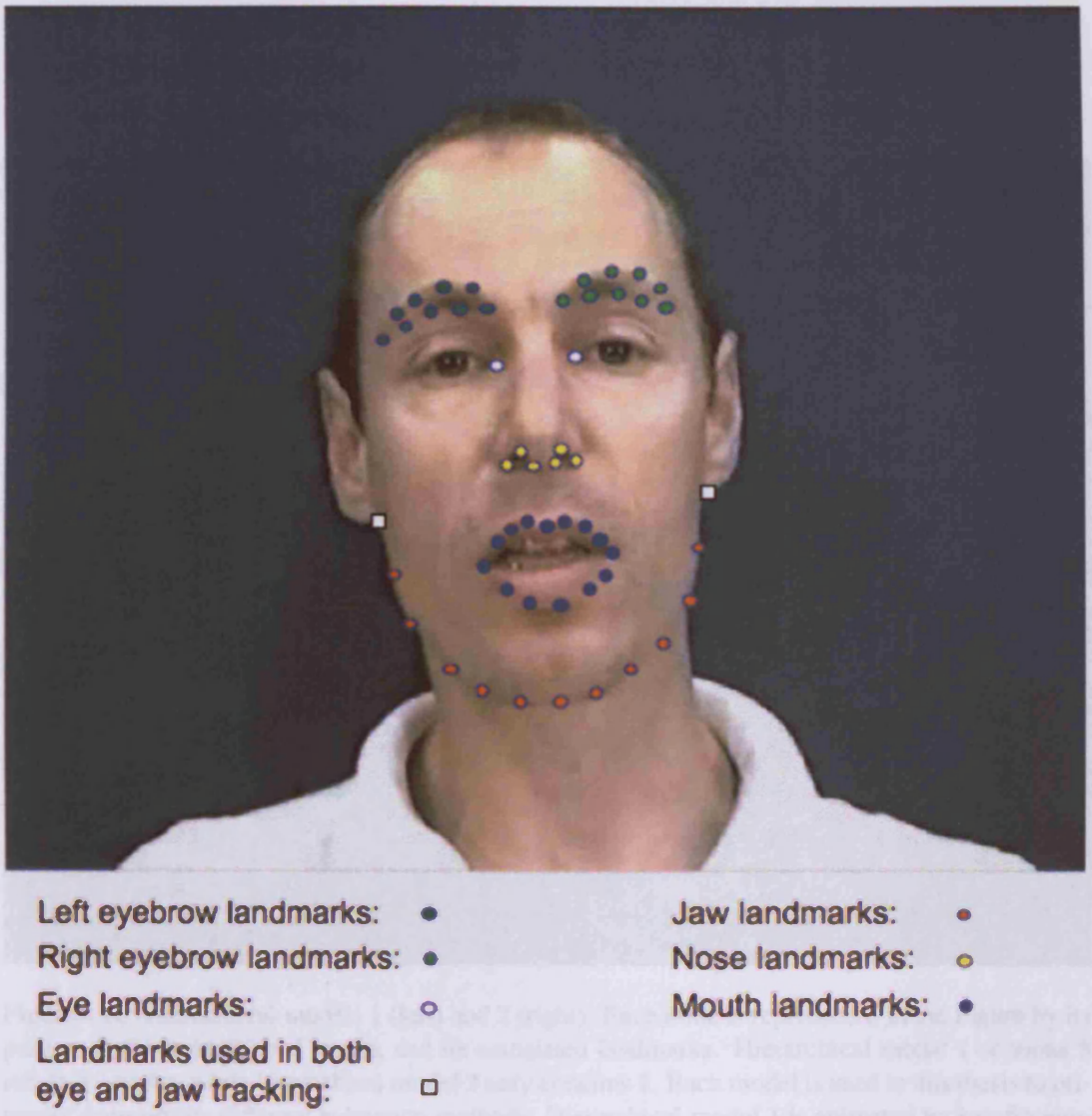

Figure 4.11: Individually tracked facial regions. When tracking, individual facial regions are landmarked separately for optimisation. The Figure shows which regions are tracked separately. Note that the landmarks between the corner of the jaw and the ears are used for tracking the jaw and the eye corners. Also note that these separate regions do not necessarily correspond to separate sub-facial regions in the final hierarchy.

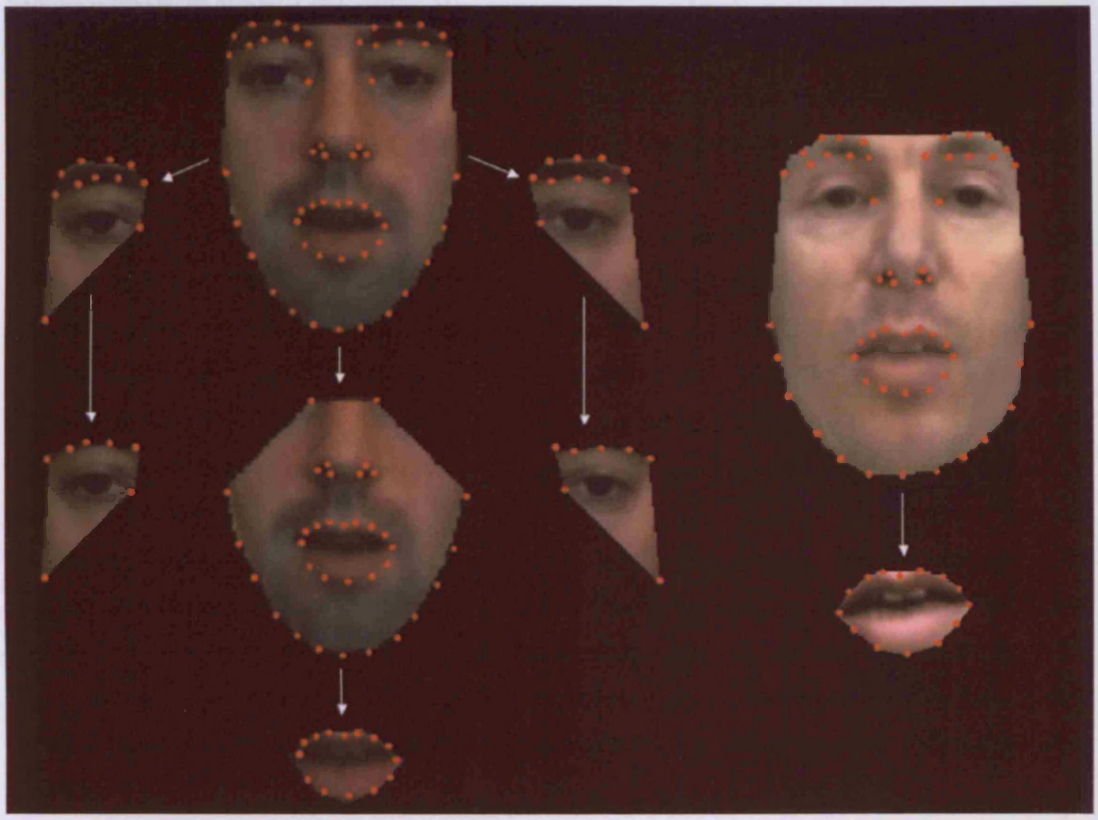

Figure 4.12: Hierarchical models 1 (left) and 2 (right). Each node is represented in the Figure by its portion of the overall facial image, and its associated landmarks. Hierarchical model 1 contains 5 sub-facial nodes, while hierarchical model 2 only contains 1. Each model is used in this thesis to primarily demonstrate different animation methods. Hierarchical model 1 is animated by key-framing sub-facial parameter values, and hierarchical model 2 is animated automatically from speech.

4.6.1 Hierarchical Model 1

The purpose of model 1 is to demonstrate the animation flexibility achievable using a full hierarchical model with multiple sub-facial nodes. The full hierarchy allows the configuration of multiple facial poses through the simple manipulation of sub-facial appearance parameters. This process is demonstrated with Model 1 in Chapter 10.

The rationale behind design of model 1 is now described. The choice of which facial areas to model as nodes is based on subjective observation, technical considerations and perceptual studies [89]. Subjective choice is largely based on selecting nodes which contribute to isolated facial variation.

Technical factors also contribute to the choice of a node, as in the case of the lower-face, and left and right eyebrows. The left and right eyebrows, if landmarked solely around their border, provide little information regarding their position relative to the rest of the face. This information is required in order to ascertain whether or not an eyebrow is raised or not. Therefore, several anchor points are included in the eyebrow model around the upper jaw and the corner of the eye. The eyebrow region, in terms of animation, now becomes a larger proportion of the face (practically speaking, an entire side of the face).

The eyes themselves, although encompassed by the eyebrow region, are still animated using separate parameters. These parameters are responsible for opening or closing the left and right eyelids. An animated eye region is layered on-top of its respective eyebrow region during facial reconstruction. The *shape* of the final merged eye/eyebrow region, as it appears in the face, is then calculated with the aid of shape information taken from the eye-brows position with respect to its anchors.

A lower face node is constrained to include nose landmarks in order to anchor the nose into a stable position whenever the mouth node opens. Without nose landmarks, whenever the mouth opens *its* landmarks will deform the appearance of the nose.

4.6.2 Hierarchical Model 2

The purpose of Model 2 is to demonstrate the ability of the SAM and HMCM for producing visual-speech. Thus, model 2 only contains two nodes - one for the face and one for the mouth. The mouth node in Model 2 is larger (in terms of pixels) than that used in Model 1 in order to demonstrate mouth animations in more detail. Model 2 is also evaluated perceptually using the *McGurk* experiment

described in Chapter 12.

4.7 Hierarchical Model Node Composition

Each node in the hierarchy (the root and each child) contains six models. These models are constructed using the memory efficient PCA techniques described in Appendix A.

Colour modeling in the hierarchy is approached as follows: All texture models use the YIQ colour space, which has three channels, namely *luminance*, *hue* and *saturation*. Appearance models are built for each node using image *luminace* signals and landmark shape data. The *luminance* channel of the YIQ colour space is equivalent to a grey-scale signal, and hence contains enough information for use in analysis and synthesis without the need for the missing colour information. All analysis, processing and animation is therefore carried out using the luminance appearance model *only*.

Models in each node therefore defined as follows:

- Shape Model (PDM)

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_x \mathbf{b}_x \tag{4.18}$$

where $\mathbf{x}$ is a shape vector, $\bar{\mathbf{x}}$ is the mean shape vector, $\mathbf{P}_x$ are the shape PDM eigenvectors and $\mathbf{b}_x$ is the shape PCA weight parameter.

- Appearance Model (Shape):

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_x \mathbf{W}^{-1} \mathbf{Q}_x \mathbf{c} \tag{4.19}$$

where $\mathbf{W}^{-1}$ is the shape parameter scale matrix, $\mathbf{Q}_x$ are the appearance model eigenvectors relating to shape and $\mathbf{c}$ is the appearance parameter.

- Appearance Model (Luminance)

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{Q}_g \mathbf{c} \tag{4.20}$$

where $\mathbf{g}$ is a texture vector, $\bar{\mathbf{g}}$ is the mean texture vector, $\mathbf{P}_g$ are the luminance PCA model eigenvectors, $\mathbf{Q}_g$ are the appearance model eigenvectors relating to texture and $\mathbf{c}$ is the appearance parameter.

- Luminance Signal PCA Model

$$\mathbf{l} = \bar{\mathbf{l}} + \mathbf{L}\mathbf{b}_l \tag{4.21}$$

where $\mathbf{l}$ is a luminance vector, $\bar{\mathbf{l}}$ is the mean luminance vector, $\mathbf{L}$ are the luminance PCA model eigenvectors and $\mathbf{b}_l$ is the luminance PCA weight parameter.

- Hue Signal PCA Model

$$\mathbf{h} = \bar{\mathbf{h}} + \mathbf{H}\mathbf{b}_h \tag{4.22}$$

where $\mathbf{h}$ is a hue vector, $\bar{\mathbf{h}}$ is the mean hue vector, $\mathbf{H}$ are the hue PCA model eigenvectors and $\mathbf{b}_h$ is the hue PCA weight parameter.

- Saturation Signal PCA Model

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{S}\mathbf{b}_s \tag{4.23}$$

where $\mathbf{s}$ is a saturation vector, $\bar{\mathbf{s}}$ is the mean saturation vector, $\mathbf{S}$ are the saturation PCA model eigenvectors and $\mathbf{b}_s$ is the saturation PCA weight parameter.

For each PCA model, 98 % of the total variation is generally retained. This number is reduced further when building SAMs and HMCMs, and discussed in Chapters 5 and 7.

All texture models use the YIQ colour space, which has three channels, namely *luminance*, *hue* and *saturation*. PCA models are constructed for the hue and saturation image signals of each sub-facial area. Appearance models are built for each node using image *luminace* signals and landmark shape data. The *luminance* channel of the YIQ colour space is equivalent to a grey-scale signal, and hence contains enough information for use in analysis and synthesis without the need for extra colour information.

Audio-visual analysis is performed using only luminance appearance parameters. Thus, speech driven animation for the mouth yields a sequence of sub-facial shape and luminance vectors. Semi-automatic animation is also appearance model based, and produces a similar set of vectors. Colour reconstruction takes place after shape and luminance vector construction, using a look-up table (see Section 4.7.1). A full description of the reconstruction process is given in Chapter 9. The hue and saturation PCA models are required for the parent approximation process (also described in Chapter 9).

4.7.1 Look-Up Table Construction

Colour is typically incorporated into appearance models by defining texture vectors as concatenated Red, Green and Blue (RGB) signal vectors [145]. The appearance model therefore explicitly models colour, and variation of the appearance parameter yields a new vector from which RGB signals can be extracted for reconstruction. Constructing a PCA model using this data is very memory intensive. Therefore, in order to save memory a look up-table approach is employed. This is used in conjunction with the memory efficient PCA techniques described in Appendix A.

To reconstruct colour outputs, sub-facial hue and saturation images are retrieved from a look-up table using a luminance image as a key. Look-up tables are constructed for each sub-facial model in the hierarchy, and are used to reconstruct colour for the respective sub-facial area.

Each row of a look-up table consists of a luminance image vector, a hue image vector and a saturation image vector. Each row corresponds to the sub-facial colour information extracted from an image in the training set. The first row contains sub-facial data from the first image, and the n^{th} row contains sub-facial data from the n^{th} image.

Animation produces streams of luminance parameters for sub-facial areas. In order to produce colour, these luminance signals are used as keys by matching them against each luminance signal in the respective look-up table for that sub-facial area. Matching is performed by minimising the Euclidean distance between the synthesised luminance image signal and each luminance signal stored in each row of the look-up table. Once a best match is found, the hue and saturation image signals from that row are combined with the synthesised luminance signal (i.e. the one used as the key) to produce the colour output for that sub-facial area.

4.8 Modes of Variation: Appearance, Shape and Luminance

Figures 4.13 to 4.19 show the four highest modes of variation for each sub-facial appearance model in hierarchical model 1. Figures 4.20 to 4.21 demonstrate the same information for hierarchical model 2. Colour was reconstructed in these Figures by varying sub-facial luminance, hue and saturation PCA weights by the same standard deviation, and combining the resulting images.

Figure 4.13: First four modes of appearance variation for the face node (root) of hierarchical model 1.

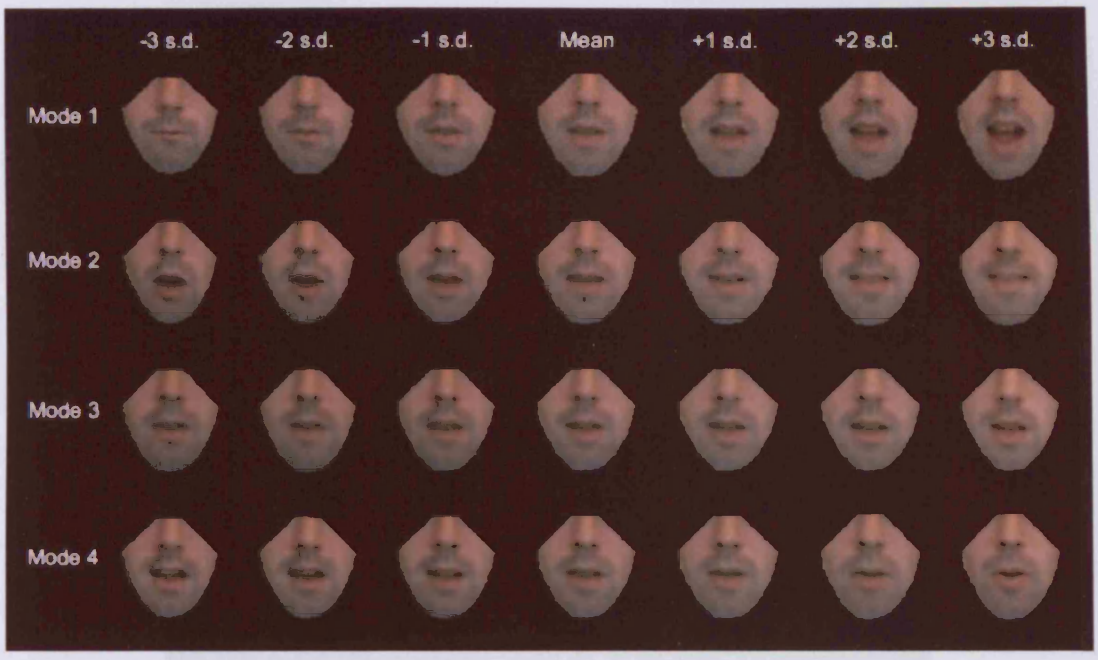

Figure 4.14: First four modes of appearance variation for the lower face node of hierarchical model 1.

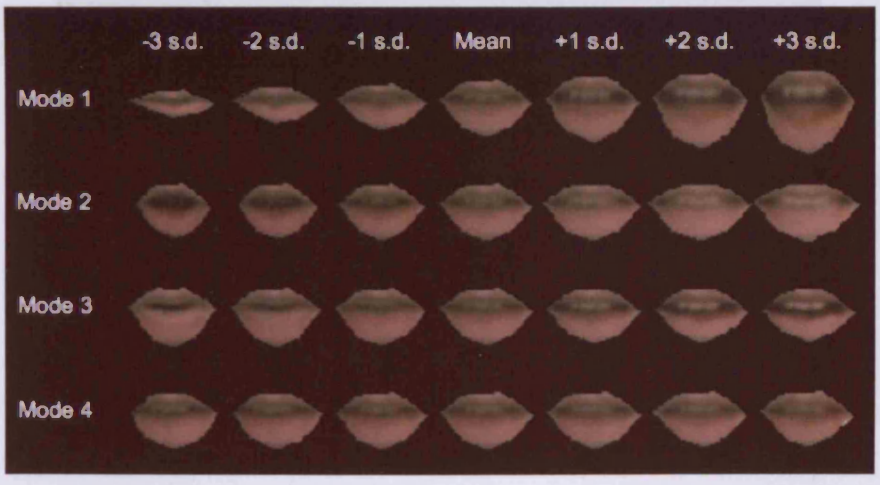

Figure 4.15: First four modes of appearance variation for the mouth node of hierarchical model 1.

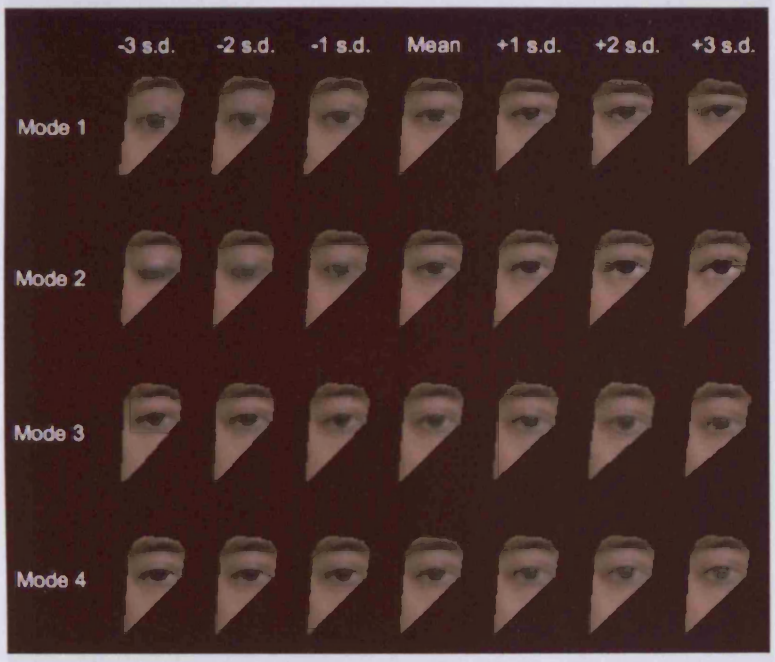

Figure 4.16: First four modes of appearance variation for the left eyebrow node of hierarchical model 1.

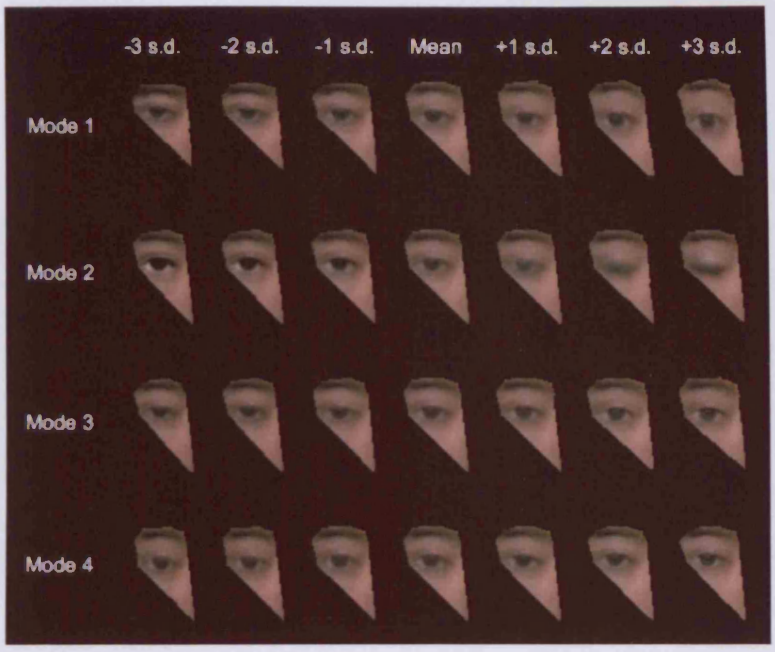

Figure 4.17: First four modes of appearance variation for the right eyebrow node of hierarchical model 1.

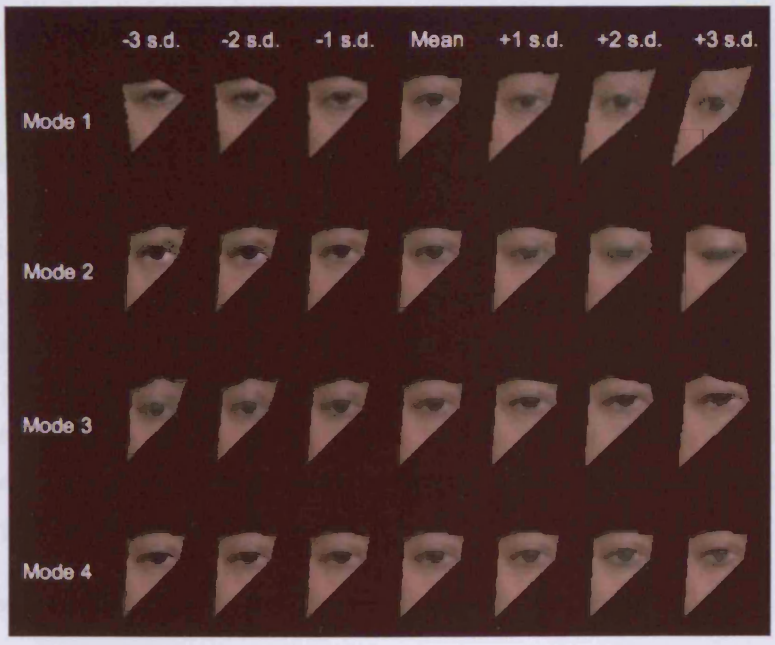

Figure 4.18: First four modes of appearance variation for the left eye node of hierarchical model 1.

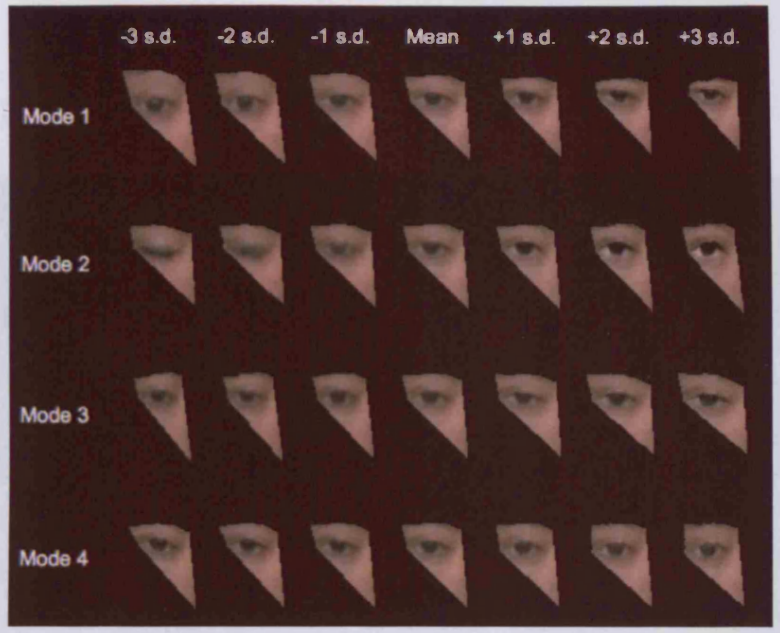

Figure 4.19: First four modes of appearance variation for the right eye node of hierarchical model 1.

4.9 Speech Processing

Now that the visual features have been described the topic turns to acquiring robust and uncorrelated speech features. These features are used in conjunction with the visual features for training the analysis and synthesis models, such that given a new set of speech features new visual features can be obtained. This thesis does not add any new theory to the field of speech analysis. Instead it borrows from techniques developed in the speech recognition community. Background information is therefore kept to a minimum, save for descriptions on why the speech features chosen are appropriate to this studies needs, and how they are obtained and represented.

The primary speech features used in this study are Mel Fequency Cepstrum Coefficients (MFCCs). MFCCs are the most commonly used feature in speech recognition, and exploit the mel scale [140]. MFCCs are typically calculated as follows [26, 47]:

1. Divide an input speech signal into frames, and window each frame.

2. Take the Fourier transform of the signal

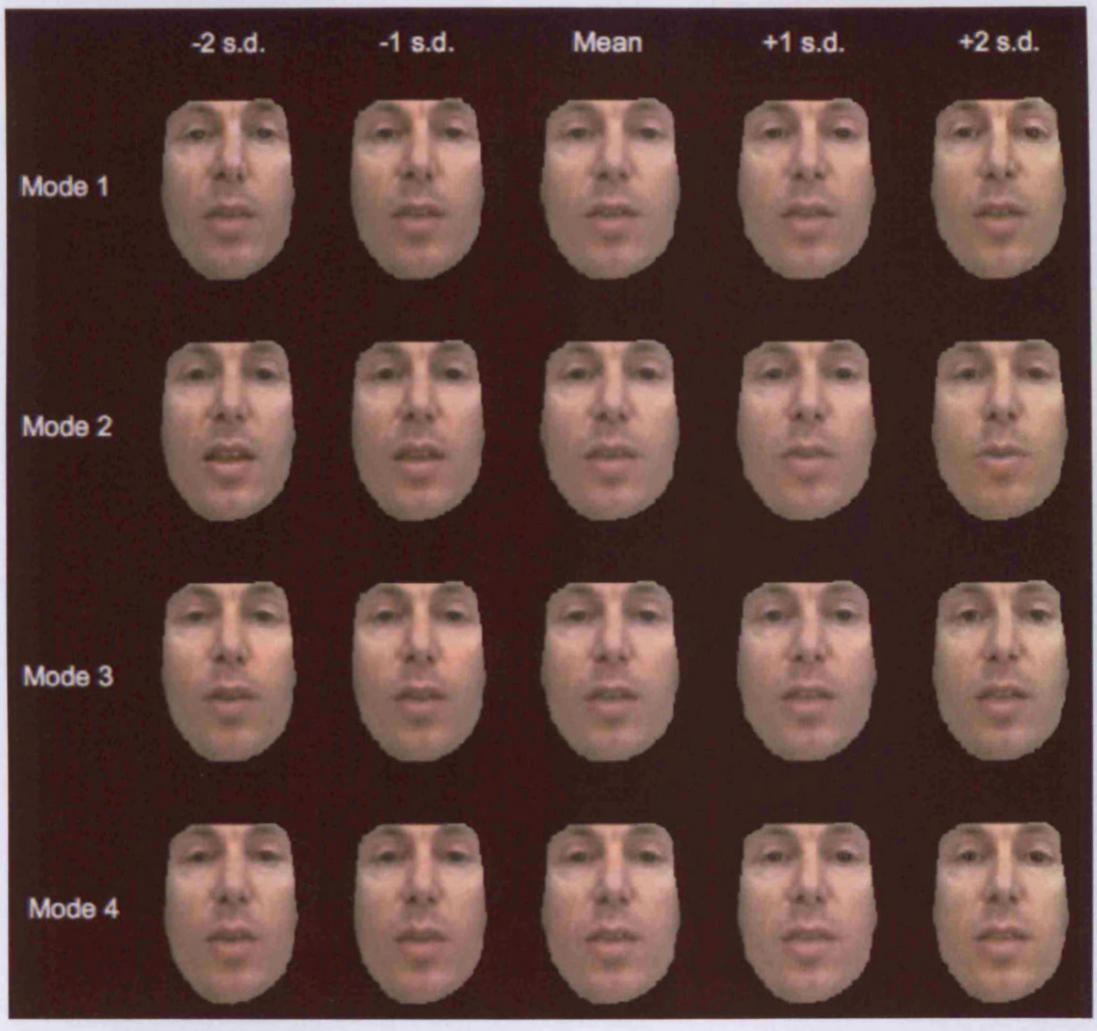

Figure 4.20: First four modes of appearance variation for the face node (root) of hierarchical model 2.

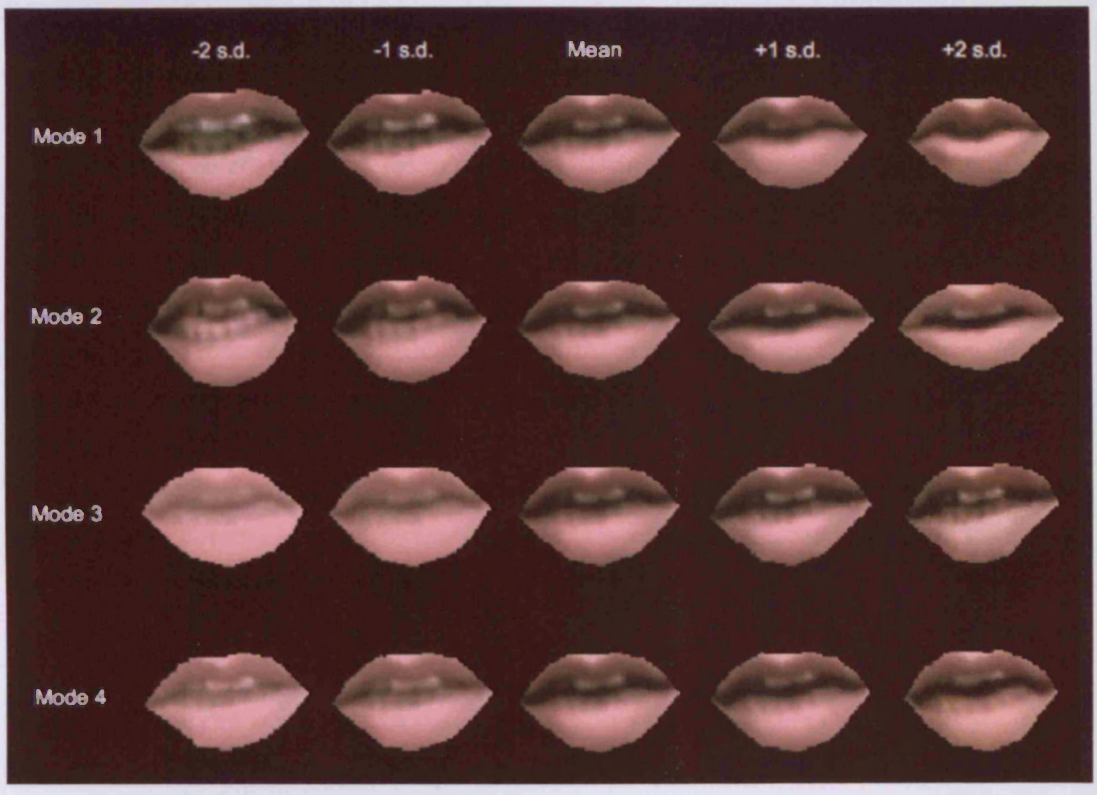

Figure 4.21: First four modes of appearance variation for the mouth node of hierarchical model 2.

3. For each frame, obtain the spectral magnitude and logarithm.

4. Warp the resulting frequencies according to the mel scale

5. Take the inverse Fourier transform.

The frame size used in this thesis is 20ms, since fast speech events such as plosives typically occur over a short time span such as this [70]. The window used to process the frames is a 20ms hamming window.

After obtaining a set of MFCCs corresponding to the training speech signal, they are then normalised using Cepstral Mean Normalisation (CMN). The goal of CMN is to reduce distortion caused by the transmission channel (e.g. the microphone). Using any transmission channel to record an input speech signal translates to multiplying the obtained spectrum by the transfer function of the channel (i.e. the distortion of the channel). Since the Cepstrum of a signal is the Fourier transform of the log spectrum, the logarithm turns the multiplication into a summation. Averaging over time, the mean may be regarded as an MFCC estimate of the input transmission channel. Any distortion imposed by the input channel may therefore be reduced by simply removing the mean MFCC vector from the training set of MFCC vectors [26, 47].

Given a set of normalised MFCCs a linear PCA model is then constructed. This allows both a reduction in the dimensionality of the features and representation of the coefficients in a linear form. The linear PCA model is defined thus

$$\mathbf{m} = \bar{\mathbf{m}} + \mathbf{M}\mathbf{b}_m \tag{4.24}$$

where $\mathbf{m}$ is a MFCC vector, $\bar{\mathbf{m}}$ is the mean normalised MFCC vector, $\mathbf{M}$ are the eigenvectors of the MFCC distribution and $\mathbf{b}_m$ is the speech PCA weight parameter. Given this model the entire normalised MFCC vector set is parameterised by rearranging equation 4.33 to give

$$\mathbf{b}_m = \mathbf{M}^T(\mathbf{m} - \bar{\mathbf{m}}) \tag{4.25}$$

Note that a single speech model is used for the entire hierarchy as opposed to a separate model being associated with each node. In this thesis, 98 percent of the speech models energy is retained - equating to 10 basis vectors in its PCA model. Figure 4.22 shows a continuous speech segment from the training set of hierarchical model 2, along with the speech parameter distribution for the full corpus (visualised in 2D). Figure 4.23 shows the trajectories made by parameters responsible

for the three highest modes of speech variation, and the corresponding first three mel-cepstral coefficients, with respect to the continuous speech segment from Figure 4.22. Note that high PCA and spectral energy is generally correlated with high frequencies in the continuous representation (see Figure 4.22 for a comparison).

4.10 Summary

This Chapter has given descriptions relating to the acquisition and initialisation process required to construct a hierarchical model. In doing so, detailed information has also been given on statistical shape modelling, grey-level modelling, and appearance modelling. Using these techniques a hierarchical model is defined, along with a memory efficient means of constructing large PCA models. The set of PCA models constituting a hierarchy (see Section 4.7), and the speech model defined in Section 4.9 are used in later Chapters for learning correspondences between speech and appearance parameters, synthesising new appearance parameters given new speech parameters, semi-automatically animating a facial area, and reconstructing colour outputs from synthesised appearance parameters.

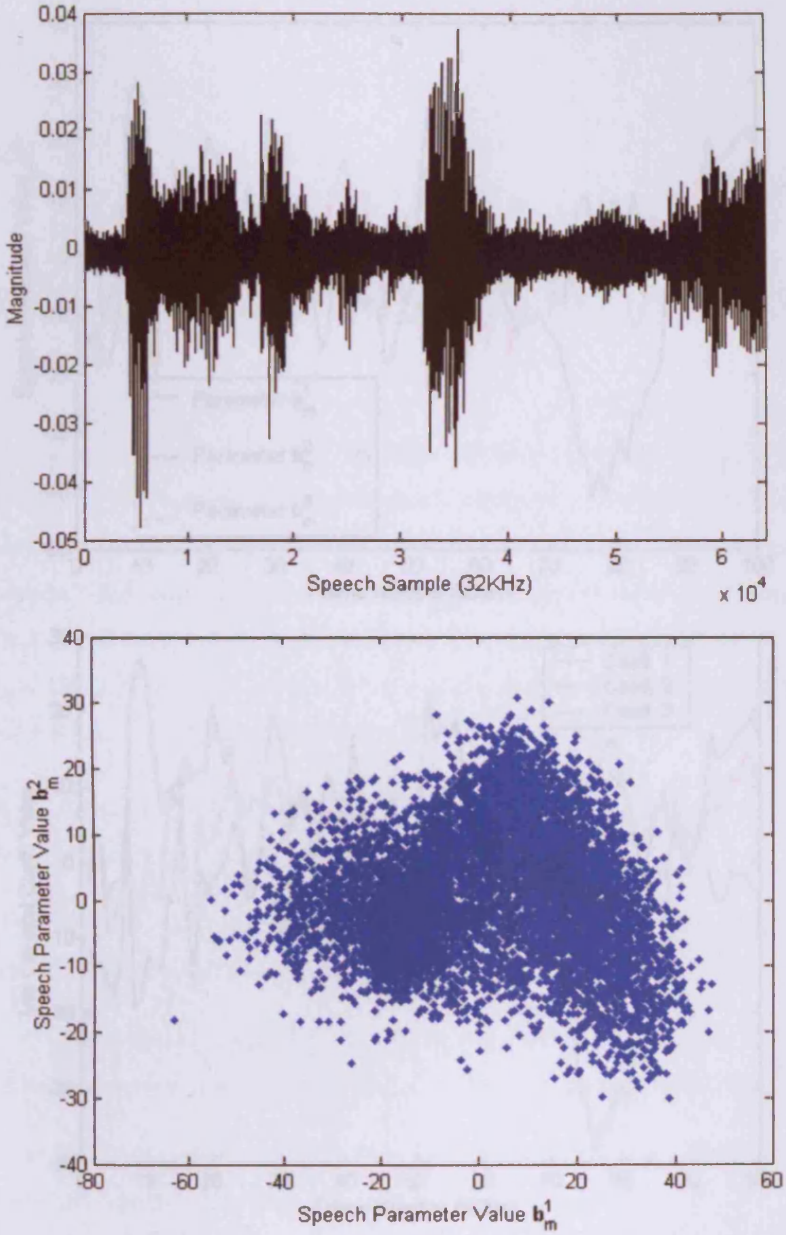

Figure 4.22: A continuous speech segment (top), and the full distribution of speech PCA parameters - visualised in 2D - from hierarchical model 2 (bottom).

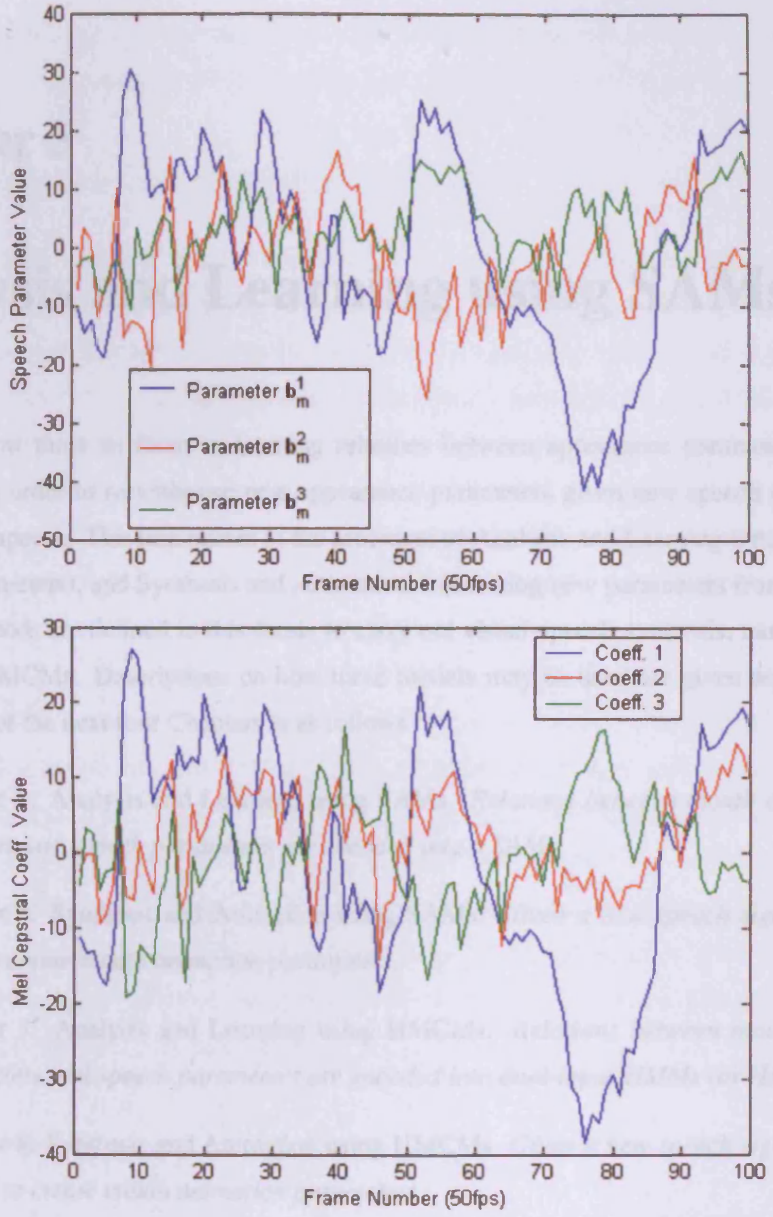

Figure 4.23: Parameter trajectories for $\mathbf{b}_m^1$, $\mathbf{b}_m^2$ and $\mathbf{b}_m^3$, and the first three Mel Cepstral coefficients corresponding to the continuous waveform segment shown in Figure 4.22.

Chapter 5

Analysis and Learning using SAMs

The thesis now turns its focus to learning relations between appearance parameters and speech parameters in order to resynthesise new appearance parameters given new speech parameters, i.e. create visual-speech. This task relates to the processes of Analysis and Learning (encoding relations between parameters), and Synthesis and Animation (estimating new parameters from speech).

Two methods are defined in this thesis to carry out visual-speech synthesis, namely the use of SAMs and HMCMs. Descriptions on how these models may be used are given separately, hence the structure of the next four Chapters is as follows

- Chapter 5: Analysis and Learning using SAMs. *Relations between mouth appearance parameters and speech parameters are encoded into a SAM.*
- Chapter 6: Synthesis and Animation using SAMs. *Given a new speech signal the SAM is used to create mouth animation parameters.*
- Chapter 7: Analysis and Learning using HMCMs. *Relations between mouth appearance parameters and speech parameters are encoded into dual-input HMMs (or HMCMs).*
- Chapter 8: Synthesis and Animation using HMCMs. *Given a new speech signal the HMCM is used to create mouth animation parameters.*

Both SAMs and HMCMs are used to animate the mouth from speech, and a semi-automatic method is used to animate the eyes and eyebrows. The face node and lower-face node are not animated from speech, or semi-automatically. Instead, these nodes are driven directly by animations created

by their children. Details on the semi-automatic key-framing technique for producing animation may be found in Chapter 10.

After animation parameters have been calculated for nodes (by whatever means), the hierarchy is then reconstructed during the Reconstruction and Display process (Chapter 9).

5.1 Appearance Parameter Interpolation

To synthesise a mouth animation from speech, a set of input speech parameters $\mathbf{b}_m^t$ are used to find a set of mouth appearance parameters $\mathbf{c}_t$, where $1 \leq t \leq T$, and T is the number of frames to animate.

Animation is produced at a rate of 50fps (or 50Hz - equivalent to the audio sampling rate) as opposed to 25fps (or 25Hz - the initial video capture rate). Thus, there are more audio samples than video samples in the initial training corpus. To learn relationships between speech and appearance parameters requires a direct correspondence, i.e. both video and audio parameters should have the same sample length, and there should be the same number of video and audio parameters. Therefore, the appearance parameters for the mouth are linearly interpolated to give a 50fps (50Hz) training set.

All appearance parameters $\mathbf{c}$, in the context of SAMs and HMCMs, are now considered in this thesis to be 20ms samples (the sample width resulting from a 50Hz sampling frequency). Thus all luminance and shape vectors reconstructed from appearance parameters, and all hue and saturation vectors reconstructed from the luminance vector, are also considered to be 20ms samples.

Linearly interpolating appearance parameters from 25Hz to 50Hz has no perceptual effect on ground truth reconstructed animations, i.e. animations reconstructed using only the appearance parameters recorded from the training video. Indeed, the visual difference between original parameters and neighboring interpolated parameters is also insignificant - as new parameters are simply averaged *bridges* between their neighbors.

5.2 Non-Linear Appearance Modelling

Given 50Hz vector correspondences between speech and appearance, relationships may now be encoded. Figure 5.1 shows the hierarchical model 2 distribution of mouth appearance parameters visualised in 2D. In this example the distribution appears quite linear, with a dense clustering of parameters along the mean $\mathbf{c}^1$-axis. However, this visualisation does not guarantee linearity

throughout the entire multi-dimensional distribution. The data would therefore be better modelled in a non-linear fashion. This would reduce the possibility of generating illegal data examples, as well as providing clusters of similar parameters - which are required for SAMs and HMCMs.

Many suitable non-linear modelling techniques exist, and two are considered here. Bowden *et al* [19] builds a non-linear PCA model by first building a linear PCA model, reducing the dimensionality of the data using the model, performing cluster analysis on the data, and then building PCA models for each cluster. Cootes and Taylor [37] represent a non-linear PCA model using a mixture of Gaussians, estimated using the Expectation Maximisation (EM) algorithm [48, 53].

In this thesis an approach similar to that of Cootes and Taylor's is taken to non-linear modelling. Given a training distribution of appearance parameters, the EM algorithm is performed to yield a Gaussian Mixture Model (GMM) [53], i.e. a set of means of covariance matrices describing the distribution. However, unlike the methods described by Bowden *et al*, and Cootes and Taylor, PCA is not performed directly on these mixtures once the GMM is trained. Instead the means and covariances are used in the next stage to cluster audio-visual data and build SAMs.

5.2.1 GMM Definition

A k-component GMM, in terms of a distribution of appearance parameters, may be defined as

$$p(\mathbf{c}) = \sum_{j=1}^{k} \alpha_j N(\mu_j, \mathbf{S}_j) \tag{5.1}$$

where $\mathbf{c}$ is an appearance parameter, α_j are the prior probabilities of each Gaussian mixture, μ_j are the means (or centres) of the Gaussians and $\mathbf{S}_j$ are the covariances of the Gaussians. Values for α_j, μ_j and $\mathbf{S}_j$ are estimated using the EM algorithm, with initial centres chosen by carrying out 10 iterations of the k-means algorithm . The term $N(\mu_j, \mathbf{S}_j)$ is a d dimensional multivariate normal (or Gaussian) distribution, and may be defined as

$$\frac{exp(-\frac{1}{2}(\mathbf{c}-\mu_j)^T \mathbf{S}_j^{-1}(\mathbf{c}-\mu_j))}{\sqrt{(2\pi)^d |\mathbf{S}_j|}} \tag{5.2}$$

5.2.2 Selecting an Appropriate Number of GMM Clusters

Selection of an appropriate value for k (i.e. the number of GMM clusters) is important since it acts as a major contributor towards the quality of final animations. The value of k affects both the level of noise in the final animations (i.e. whether an animation is smooth or not), and the accuracy and

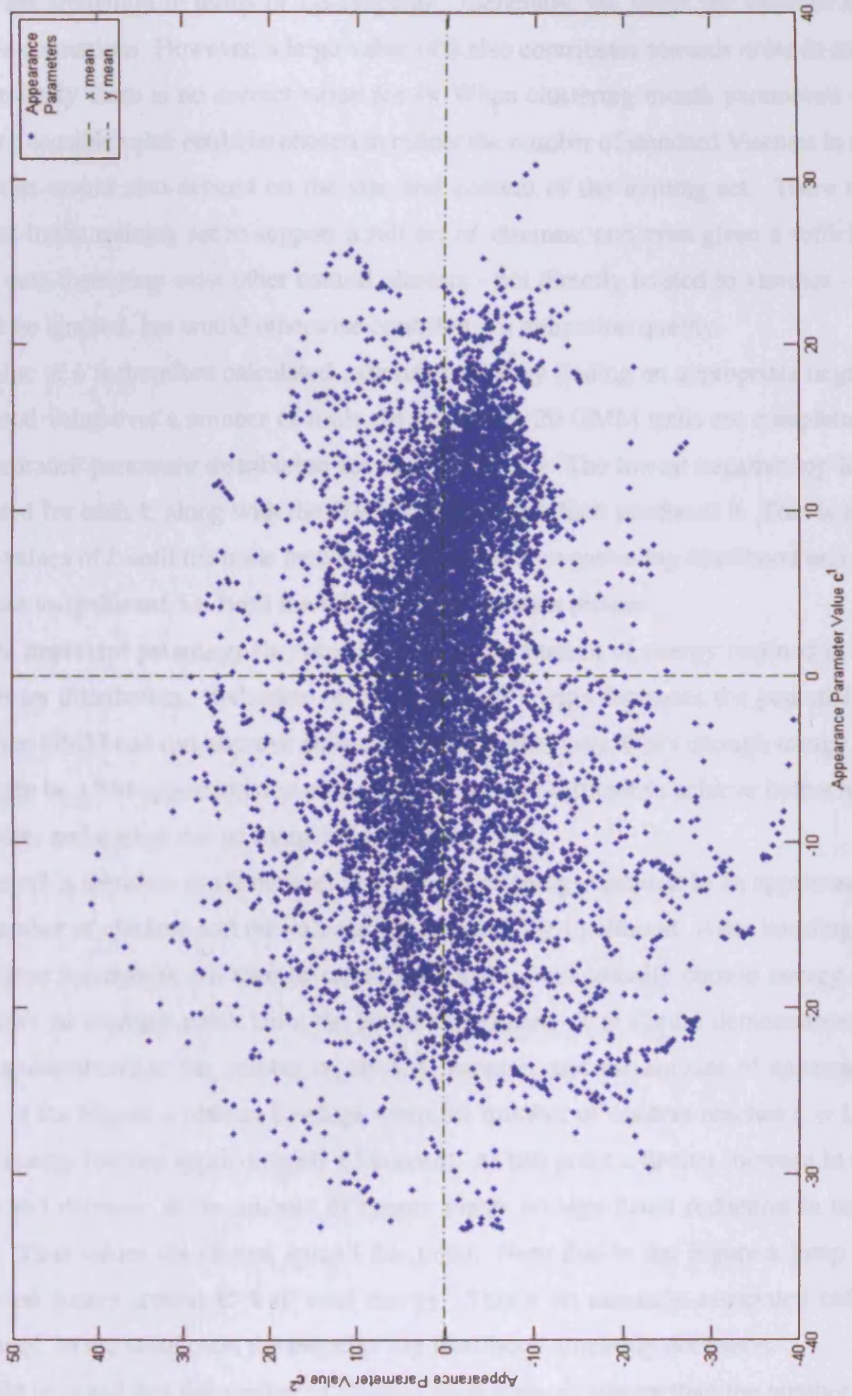

Figure 5.1: Mouth appearance parameter distribution from hierarchical model 2 visualised in 2D. Note the linear clustering along the mean $\mathbf{c}^1$-axis. Nevertheless, for the purpose of SAM and HMCM synthesis, the distribution is fitted with a non-linear model.

realism of the animation in terms of lip-synching. Generally, the larger the value of k, the more accurate the animations. However, a large value of k also contributes towards noise in animations.

Unfortunately there is no *correct* value for k. When clustering mouth parameters it could be argued that a sensible value could be chosen to reflect the number of standard Visemes in a language. However, this would also depend on the size and content of the training set. There may not be enough data in the training set to support a full set of visemes; and even given a sufficiently large amount of data there may exist other natural clusters - not directly related to visemes - that would in this case be ignored, but would otherwise contribute to animation quality.

The value of k is therefore calculated experimentally by finding an appropriate negative GMM log-likelihood value over a number of trails. In this thesis 20 GMM trails are completed using the mouth appearance parameter distribution for each value of k. The lowest negative log-likelihood is then recorded for each k, along with the GMM parameters which produced it. This is repeated for increasing values of k until the trade between a reduction in negative log-likelihood and an increase in k becomes insignificant, i.e. until the relationship reaches a plateau.

Another important parameter for consideration is the amount of energy retained in the appearance parameter distribution. Reduction of the amount of energy increases the potential number of clusters in the GMM and can improve animation quality. However, if not enough energy is retained, the GMM can be a bad approximation of the data. It is often difficult to achieve both a high level of GMM clusters and energy due to computational costs.

A trade-off is therefore made between the amount of energy retained in an appearance distribution, the number of clusters, and the value of the negative log-likelihood. After building the GMM, the appearance parameters can then be represented using their initially chosen energy value. Figure 5.2 shows an example result from the above experiment. The Figure demonstrates the fall in negative log-likelihood as the number of clusters increase, and the amount of appearance energy decreases. In the Figure, a plateau develops when the number of clusters reaches $k = 120$, and the amount of energy reaches approximately 65 percent. At this point a further increase in the number of clusters and decrease in the amount of energy yields no significant reduction in negative log-likelihood. Thus values are chosen around this point. Note that in the Figure a jump in negative log likelihood occurs around 65% of total energy. This is an anomaly associated only with this particular trail. In the usual case, the negative log-likelihood smoothly decreases.

It should be noted that the number of clusters used is much greater than the number of visemes

associated with visual speech. Experimentally it was found that the larger the number of clusters the better the results. Perhaps the benefit is that this allows clusters which represent transitions between visemes.

Figure 5.3 shows a GMM with 120 mixtures fitted to the mouth appearance parameter distribution from hierarchical model 2. In the Figure, blue ellipses represent Gaussians, while red lines show the direction of the highest mode of variation in a mixture resulting from performing SVD on a Gaussians covariance matrix.

5.3 Constructing SAMs

Using the covariances and centres of a trained GMM, 20ms appearance parameters are then classified into one of k clusters by calculating the minimum Mahalanobis distance between each parameter $\mathbf{c}$ and each centre.

$$D(\mathbf{c}) = argmin_{j=1}^{k}[(\mathbf{c}-\mu_j)^T\mathbf{S}_j^{-1}(\mathbf{c}-\mu_j)] \tag{5.3}$$

To build a SAM, correspondences between appearance and speech must be maintained. Therefore parameters $\mathbf{a} = [\mathbf{c}^T\mathbf{b}_m^T]^T$ are defined. Each parameter $\mathbf{a}$ consists of an appearance parameter with its corresponding speech parameter from the training set (using 20Hz sampling). Using the Mahalanobis cluster classification already defined, parameters $\mathbf{a}$ now also belong to one of the k states.

These clusters are used in the synthesis step to synthesise appearance parameters from speech. Due to coarticulation, the relationship between speech and appearance is many-to-many, i.e. one speech parameter may have many mouth postures associated with it, and vice-versa. In the ideal case there would be one-to-one mappings between audio and video, and each cluster would contain correlated audio-visual parameters. Even so, for the synthesis method described in the next chapter it is a sufficient to assume this as a first order approximation, and this has been found to give reasonable results.

Each cluster of parameters $\mathbf{a}$ is now used to build a SAM, and the number of clusters, i.e. the value of k, directly affects animation performance.

The SAM construction process is straightforward, and involves performing PCA on each of the k clusters. A SAM for a sub-facial area is therefore defined as

$$\mathbf{a} = \bar{\mathbf{a}}^r + \mathbf{A}^r\mathbf{b}_a^r \qquad r = 1,\ldots,k \tag{5.4}$$

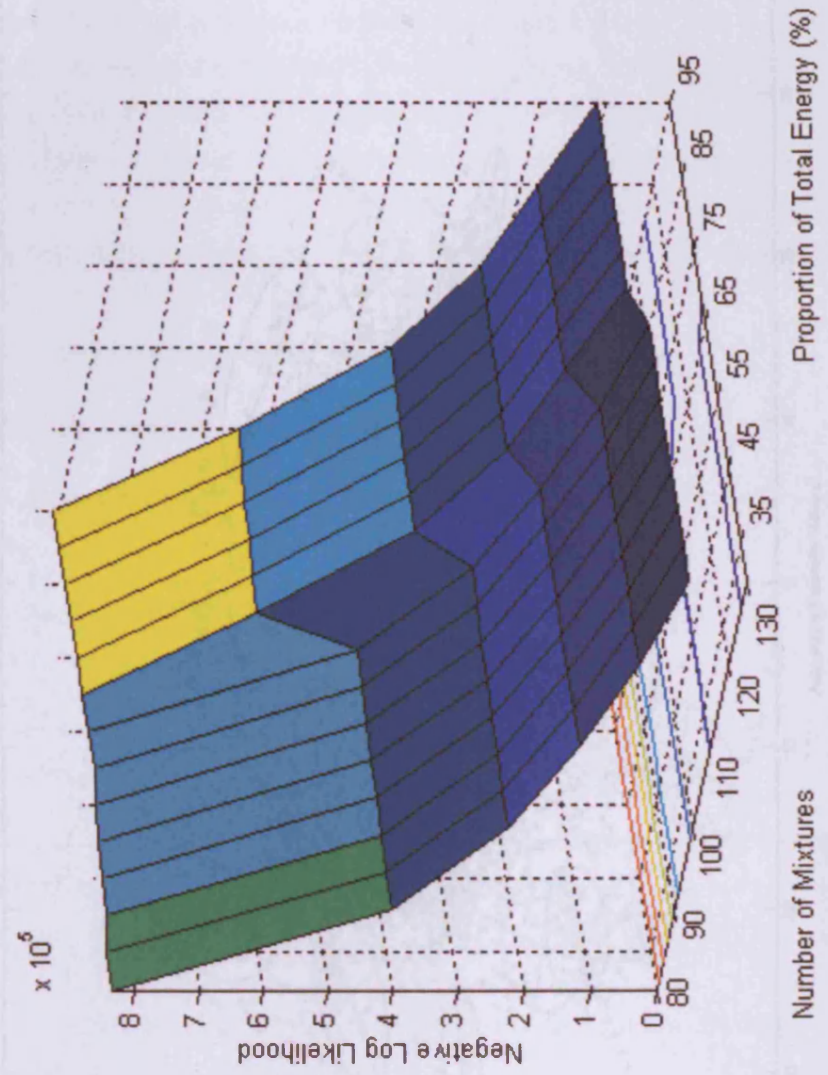

Figure 5.2: GMM trails for estimating values of k and appearance model energy. Negative log-likelihood is shown as a function of k and appearance energy. Note that how negative log-likelihood plateaus as k and energy increase.

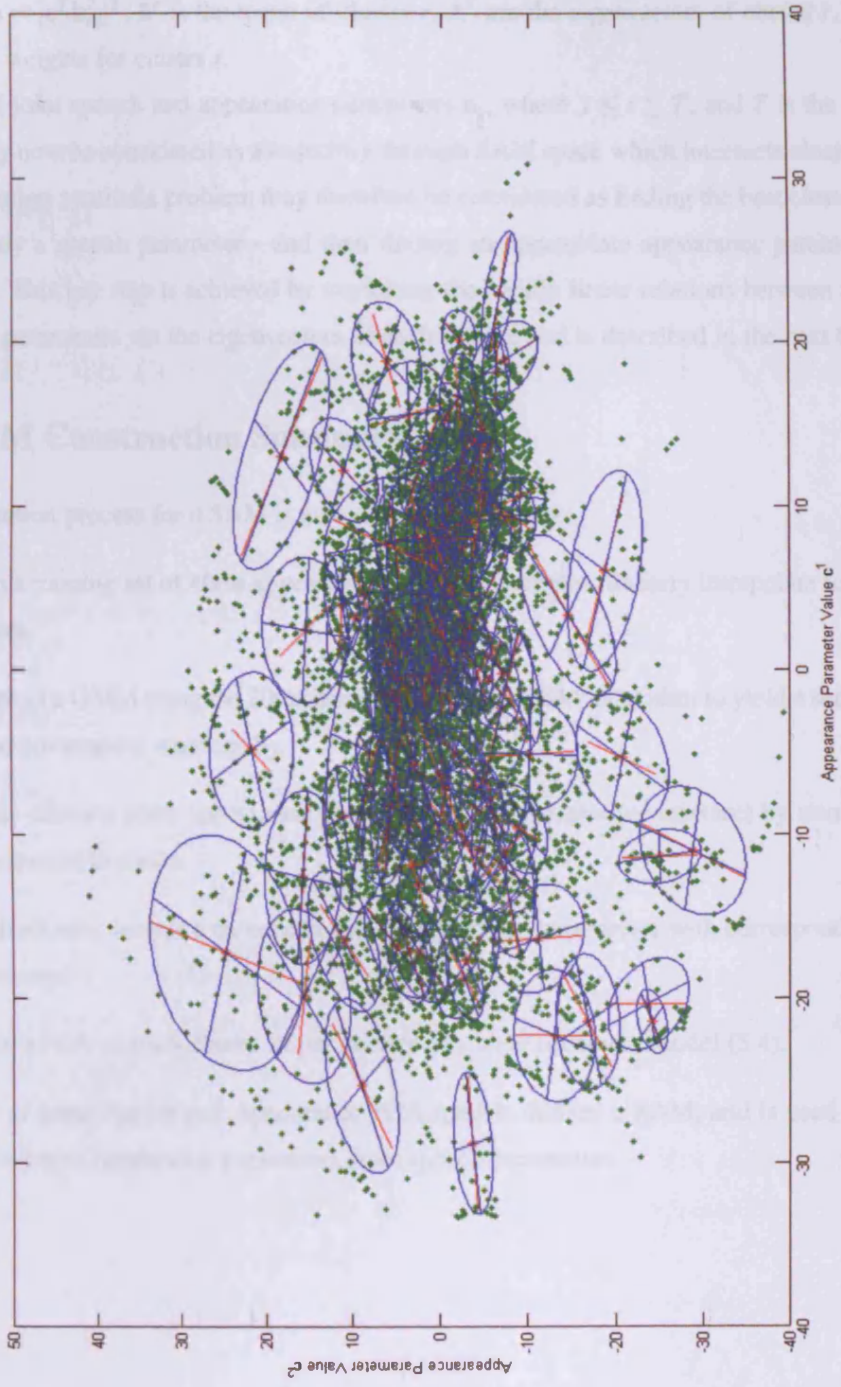

Figure 5.3: Mouth appearance parameter distribution from hierarchical model 2 visualised in 2D, and fitted with a 120 mixture GMM. Blue ellipses represent Gaussians, while red lines show the direction of the highest mode of variation in a mixture resulting from performing SVD on a Gaussian's covariance matrix.

where $\mathbf{a} = [\mathbf{c}^T \mathbf{b}_m^T]^T$, $\bar{\mathbf{a}}^r$ is the mean of cluster r, $\mathbf{A}^r$ are the eigenvectors of cluster r, $\mathbf{b}_a^r$ are the eigenvector weights for cluster r.

A set of joint speech and appearance parameters $\mathbf{a}_t$, where $1 \leq t \leq T$, and T is the number of vectors, may now be considered as a trajectory through SAM space which intersects cluster r at time t. The animation synthesis problem may therefore be considered as finding the best cluster r at time t - using only a speech parameter - and then finding an appropriate appearance parameter within that cluster. This last step is achieved by exploiting the locally linear relations between speech and appearance parameters via the eigenvectors of each cluster, and is described in the next Chapter.

5.4 SAM Construction Summary

The construction process for a SAM is now summarised below.

1. Given a training set of 40ms appearance parameter samples, linearly interpolate to give 20ms samples.

2. Construct a GMM using the 20ms distribution with the EM algorithm to yield a set of k means μ_j and covariances matrices $\mathbf{S}_j$.

3. Strictly allocate 20ms appearance parameters to each cluster (or mixture) by minimizing the Mahalanobis distance.

4. Construct new vectors $\mathbf{a}$ by concatenating appearance parameters with corresponding speech parameters.

5. Perform PCA on each cluster of parameters $\mathbf{a}$ to yield the linear model (5.4).

The set of joint Speech and Appearance PCA models defines a SAM, and is used in the next Chapter to estimate appearance parameters from speech parameters.

Chapter 6

Synthesis and Animation using SAMs

Using a SAM, the mouth may be animated from speech by finding a trajectory of most appropriate appearance parameters given a set of input speech parameters. This Chapter describes how this is achieved, and discusses issues related to the method.

6.1 Local Estimation of Appearance from Speech

The task of synthesis is to estimate a new trajectory of vectors **a** through SAM space given only input speech parameters. This may be considered as two separate problems: 1) Choice of an appropriate cluster r in the SAM at each time t, and 2) Calculation of an appearance parameter **c** using this cluster.

The second problem is addressed first, and it is already assumed that an appropriate cluster has been chosen. Figure 6.1 shows a 2D distribution of parameters **a**, each parameter consisting of a 1D appearance parameter c and a 1D speech parameter b_m. The Figure also shows the highest mode of variation through the distribution (red line). Under the assumption that the distribution is locally linear, a parameter c can be estimated from a parameter b_m by first projecting b_m onto the eigenvectors, and then back-projecting onto the c-axis. To do this the eigenvectors $\mathbf{A}^r$ must first be split into two parts, one relating to speech and the other to appearance.

The eigenvectors $\mathbf{A}^r$ encode both values for **c** and **b**, and are separable such that

$$\mathbf{A}^r = \begin{bmatrix} \mathbf{A}_Q^r \\ \mathbf{A}_M^r \end{bmatrix} \tag{6.1}$$

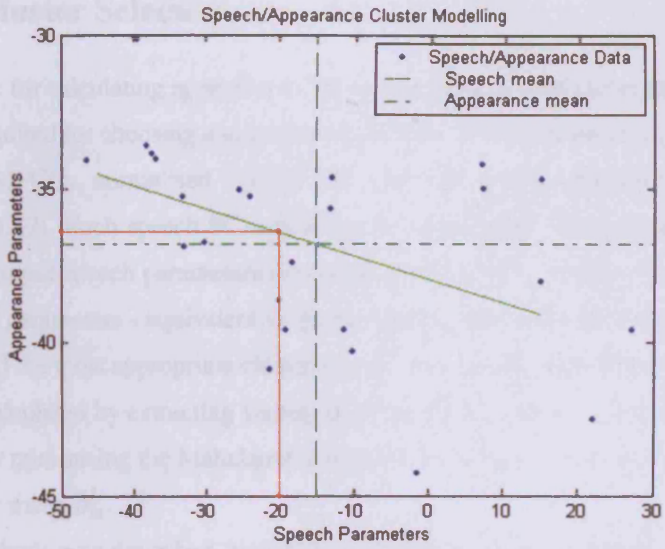

Figure 6.1: A 2D distribution of vectors **a**, where each parameter consists of a 1D appearance parameter and a 1D speech parameter. The basis vectors formed by this distribution allow the estimation of one parameter given the other via a projection onto its axis.

where $\mathbf{A}_Q^r$ are the rows of $\mathbf{A}^r$ relating to the appearance model, and $\mathbf{A}_M^r$ are the rows of $\mathbf{A}^r$ relating to the speech model. This separation of $\mathbf{A}^r$ is equivalent to the separation of $\mathbf{Q}$ into the matrices $\mathbf{Q}_x$ and $\mathbf{Q}_g$, as carried out in equation (4.11).

Therefore, the vectors $\mathbf{c}$ and $\mathbf{b}_m$ may be written as functions of $\mathbf{b}_a$ using

$$\mathbf{c} = \bar{\mathbf{c}}^r + \mathbf{A}_Q^r \mathbf{b}_a \tag{6.2}$$

$$\mathbf{b}_m = \bar{\mathbf{b}}_m^r + \mathbf{A}_M^r \mathbf{b}_a \tag{6.3}$$

where $\bar{\mathbf{c}}^r$ is the mean appearance parameter in cluster r and $\bar{\mathbf{b}}_m^r$ is the mean speech parameter in cluster r. Since the eigenvectors $\mathbf{A}_Q^r$ and $\mathbf{A}_M^r$ are orthogonal, $\mathbf{b}_a$ may be written as

$$\mathbf{b}_a = \mathbf{A}_M^{rT}(\mathbf{b}_m - \bar{\mathbf{b}}_m^r) \tag{6.4}$$

This allows estimation of an appearance parameter $\mathbf{c}$ by using $\mathbf{b}_a$ in equation (6.2), and is carried out for each 20ms input speech parameter $\mathbf{b}_m$ using an appropriate SAM cluster.

The projection is demonstrated visually (in 2D) in Figure 6.1, and may also be applied in reverse, i.e. calculation of $\mathbf{b}_m$ from $\mathbf{c}$.

6.2 SAM Cluster Selection

Given a procedure for calculating appearance parameters from speech parameters within a cluster, a method is now required for choosing a sequence of clusters. An input speech signal is first processed to yield a set of MFCCs, normalised using CMN, and is then projected through the speech PCA model (equation 4.37). Each speech PCA parameter is responsible for synthesis of a single output video frame. The input speech parameters are therefore defined $\mathbf{b}_m^t$, where $t = 1, \ldots, T$ and T is the number of speech parameters - equivalent to the number of output frames to synthesise.

In order to find the most appropriate cluster at each time t, speech covariance matrices and mean vectors are first calculated by extracting vectors $\mathbf{b}_m$ from each cluster of vectors $\mathbf{a}$. Cluster choice is then calculated by minimising the Mahalanobis distance between an input speech parameter at time t, and each cluster mean $\bar{\mathbf{b}}_m^r$.

Using the methods now described, appearance parameters can be estimated from speech to create animations. However, the Mahalanobis cluster selection procedure is somewhat of an oversimplification with regard to mouth animation since it assumes a direct one-to-one mapping between speech and appearance over a very short time frame (20ms). This assumption is in conflict with the rules of co-articulation - which describe how the mouth appearance at any given time is directly influenced by how the mouth has appeared previously and how it will appear in the near future.

Since coarticulation is not modelled, cluster choices at time t are not constrained by choices at times *t-1* and *t+1*. This means that neighbouring clusters can by chosen which are large distances apart. The result of this is that synthesised parameter trajectories, and corresponding output animations, may appear *noisy*.

6.3 Post-Processing

One approach to reduce signal noise, and the effect of incorrect cluster choice, is to smooth the synthetic signal. Smoothing of synthesised animation parameters is not uncommon. A popular approach to achieving this is to fit splines to synthesised animation trajectories [145] [83]. However, the characteristics of the noise inherent in a SAM signal make it unsuitable for spline fitting. This is because the noise causes large changes in the synthesised signal over short time periods (typically one frame), and the spline will be consequently fitted to these large changes, as opposed to removing them.

A better approach is to median filter the SAM signal, thus removing *spikes* in the trajectory caused by the incorrect cluster choice. Figure 6.2 shows an appearance trajectory compared to its ground truth before and after median filtering with a window of size 3. Note the signal spikes in the original trajectory, and their removal in the filtered signal.

A demonstration of synthesis using a SAM is given in Chapter 11, and a more detailed evaluation can be found in [42, 43, 44].

6.4 SAM Synthesis Summary

A summary is now given of the SAM synthesis method. Given a SAM model, an animation is constructed using the following procedure:

1. Given a new speech recording, construct 20ms MFCCs and perform CMN (see Section 4.10).
2. Project the MFCCs through the speech PCA model (equation 4.34)
3. For each resulting speech parameter $\mathbf{b}_m$:
   (a) Chose a SAM cluster by minimizing the Mahalanobis distance between the input speech parameter and the speech mean of each SAM cluster.
   (b) Using the chosen cluster and the input speech parameter, estimate (and store) an appearance parameter (equations 6.2 and 6.4)
4. Post process the resulting appearance parameter trajectory using a median filter.

Using the calculated mouth appearance parameters, luminance and shape information can then be retrieved via its appearance model to produce a animation for that region. Construction of a full facial animation involves the seamless merging of the resulting mouth animation and any other synthesised sub-facial animations. The issues involved with this procedure, along with their solution, are given in Chapter 9, which outlines the process of *Reconstruction and Display*.

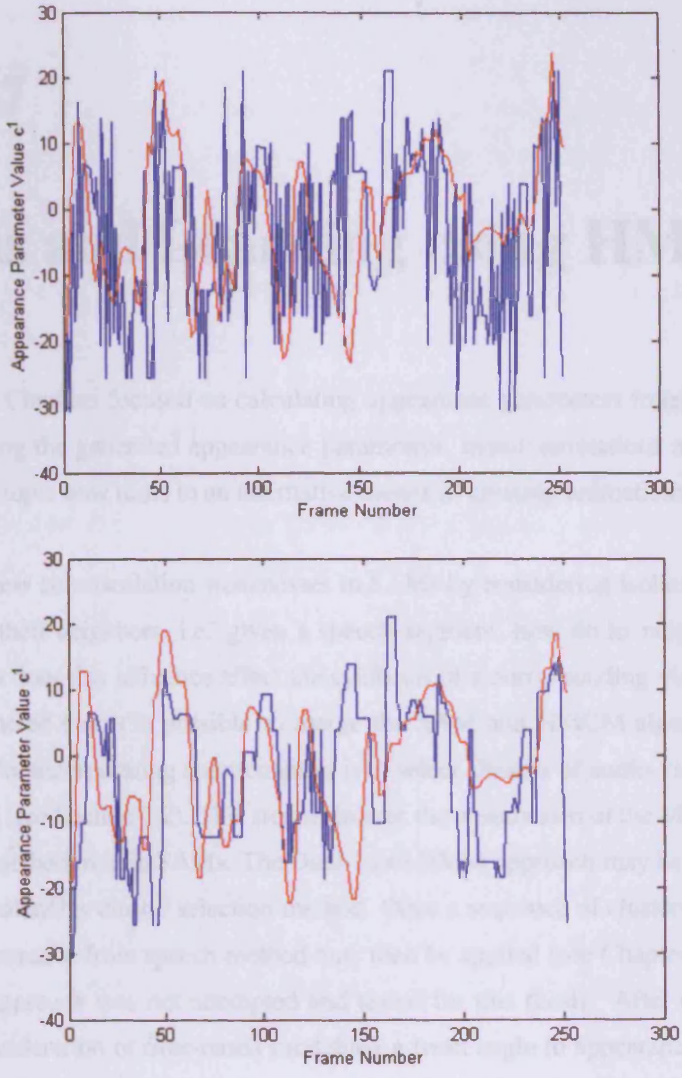

Figure 6.2: An appearance trajectory synthesised by a SAM before median filtering (top) and after median filtering (bottom). The red trajectory is the ground truth signal while the blue trajectory is the synthetic signal.

Chapter 7

Analysis and Learning using HMCMs

The previous two Chapters focused on calculating appearance parameters from speech parameters using SAMs. Using the generated appearance parameters, mouth animations may then be created from speech. The topic now turns to an alternative means of creating animations from speech using HMCMs.

HMCMs address co-articulation weaknesses in SAMs by considering isolated speech segments in the context of their neighbors, i.e. given a speech segment, how do its neighbors influence its meaning, and how does this influence affect the synthesis of a corresponding visual output?

It should be noted that it is possible to merge the SAM and HMCM algorithms. Part of the HMCM solution for incorporating coarticulation is to select clusters of audio-visual vectors using a Dual-Input HMM (see Section 7.2). This step addresses the weaknesses of the Mahalanobis distance cluster selection method used in SAMs. The Dual-Input HMM approach may be used as a substitute for the SAM Mahalanobis cluster selection method. Once a sequence of clusters has been selected, the in-cluster appearance from speech method may then be applied (see Chapter 6).

This hybrid approach was not attempted and tested for this thesis. After development of the SAM and the consideration of time-based modelling a fresh angle to appearance from speech synthesis was sought - hence the development of the HMCM. In the authors opinion, a SAM-HMCM hybrid would perform better than a SAM, but worse than a HMCM.

7.1 Time Based Cluster Selection

Effective animation of the mouth requires that the rules of co-articulation are obeyed. To this end, several approaches have previously been suggested, notably in the works of Cohen and Massaro [31], Parke and Waters [120], Ezzat *et al* [62], Bregler *et al* [22], Cosatto and Graf [40], Brand [21] and [102]. The SAM method does not directly account for co-articulation since it assumes that short (20ms) one-to-one mappings exist between input parameters and output parameters.

The cause of incorrect cluster choice (under the one-to-one mapping assumption), is due to the fact that mouths (or any other facial areas) of similar appearance can be associated with entirely different speech segments. This is due to clustering being performed on appearance parameters. Given that clusters are formed by grouping similar mouths, it is entirely likely that a set of similar mouths with different speech vectors will be clustered together. The dilemma then lies in the fact that given a new speech segment there may be more than one valid cluster to choose from, i.e. the speech may closely match several clusters. However, the appearance of the mouth (or other facial area) in each of these clusters can be completely different.

It is often the case that the best (i.e. the closest) Mahalanobis chosen cluster may be the incorrect one. Indeed, the worst cluster might often be the correct one. This is due to co-articulation. For example, the current speech segment may be silence, but the proceeding segment may be the phoneme /A/, in which case it is more correct to choose an open mouth for the current segment in preparation for the forthcoming /A/. However, based on the speech Mahalanobis distance it is likely that a closed mouth will be chosen, since the current speech segment is silence.

Essentially, what is required is a method of selecting clusters at time t to reflect previous cluster choices at time $t-1$, and to anticipate possible future cluster choices at time $t+1$. One approach to achieving this is to learn probable cluster transitions from the training set - a task well suited to Hidden Markov Models (HMMs).

7.1.1 Hidden Markov Models (HMMs)

HMMs are complimentary to processes that have inherent temporality, i.e. processes which unfold over time. As such they have been used successfully in tasks such as speech recognition [131], tracking [84] and facial animation [76]. Huang and Chen [76] train HMMs on individual words

for real time facial animation using continuous speech. Given a new sentence it is segmented into individual words which are used to select appropriate HMMs. The HMMs then calculate appropriate synthesis states in a pre-trained joint audio-visual GMM. Brand [20] uses an entropically trained HMM (a HMM with an entropic prior [21]) to create a dual-mapping between speech and visual parameters. Given new speech, the dual mapping allows selection of a HMM state sequence and an optimal set of visual parameters for output. Hack and Taylor [72] use a HMM to learn, and then reproduce, appearance parameters for the synthesis of realistic talking head behaviour.

In terms of the cluster selection problem, HMMs are useful since cluster choices at time t can be made based on previous choices and possible future choices - essentially modelling co-articulation.

A brief overview of HMMs is now given. A HMM is a doubly embedded stochastic process where one process is *hidden* and only observed through a second one which produces a set of observations. In terms of a SAM, the cluster selection process can be considered hidden, and the observations are joint audio-visual parameters **a**.

There are several types of HMM, and they vary primarily according to the type of data used to train them. Parameters affecting the construction of the HMM include whether data is discrete or continuous, how many *states* it has, and whether states are modelled using single or multiple Gaussian mixtures.

In this thesis Gaussian output HMMs are used, with one Gaussian mixture per state. In the HMM description that follows it is assumed that a mouth appearance parameter training set is used for construction - with 20ms observation lengths. This is because in the following Sections an appearance based HMM is initially used.

A HMM is characterised by the following

- The number of states k in the model. The states are hidden, and may be represented by Gaussian mixtures for the purposes of a continuous density HMM. The states are defined $S = S_1, S_2, \ldots, S_k$, and the state at time t is q_t.

- The data on which the model is based. This data set is defined here as a set of N 20ms appearance parameters **c**.

- The state transition probability distribution $\zeta = Z_{ij}$, where

$$Z_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \qquad 1 \leq i, j \leq k \tag{7.1}$$

Therefore Z_{ij} encodes the probability of reaching state S_j at time $t+1$ given that the state at time t is S_i.

- The observation probability distribution in state j, $\Upsilon = \upsilon_j(i)$, where

$$\upsilon_j(i) = P[\mathbf{c}_i \; at \; t \mid q_t = S_j], \qquad 1 \leq j \leq k, 1 \leq i \leq N. \tag{7.2}$$

- The initial state distribution $\pi = \{\pi_j\}$ where

$$\pi_j = P[q_1 = S_j], \qquad 1 \leq j \leq k. \tag{7.3}$$

For notational ease, a HMM is often defined as the tuple

$$\lambda = (\zeta, \Upsilon, \pi) \tag{7.4}$$

Defining an observation sequence of appearance parameters $C = \mathbf{c}_1\mathbf{c}_2 \ldots \mathbf{c}_T$, where c_t is an observation at time t, there are three basic problems associated with a HMM:

1. Given the observation sequence $C = \mathbf{c}_1\mathbf{c}_2 \ldots \mathbf{c}_T$, and a model $\lambda = (\zeta, \Upsilon, \pi)$ how can we efficiently compute $P(C \mid \lambda)$, the probability of the observation sequence given the model?

2. Given the observation sequence $C = \mathbf{c}_1\mathbf{c}_2 \ldots \mathbf{c}_T$ and the model $\lambda = (\zeta, \Upsilon, \pi)$, how can we optimally choose the hidden state sequence $Q = q_1q_2 \ldots q_T$ which best *explains* the observations?

3. How can the model parameters $\lambda = (\zeta, \Upsilon, \pi)$ be found which optimise $P(C \mid \lambda)$?

Problems 2) and 3) are the most relevant to this thesis, and relate to cluster selection and HMM training respectively. Several solutions to Problem 2) exist depending on the optimisation criteria. However, a good solution may be found using the Viterbi Algorithm [131], which attempts to optimally find the best *single* state (or cluster) sequence given the observations (e.g. new appearance parameters). Problem 3) (training a HMM) is the most difficult to solve optimally since it is computationally intractable. However, an efficient solution may be found using the Baum-Welch algorithm [131].

To summarise, the aim is to improve on the SAM cluster selection method by considering previous and possible future cluster choices - hence encoding co-articulation. A SAM can easily be

extended to include state transition probabilites and observation probabilities, allowing an appropriate sequence of clusters to be generated using the Viterbi algorithm and a new observation of input vectors **a** (problem 2).

However, when producing an animation from speech, the only observation available is a sequence of speech vectors $\mathbf{b}_m$, i.e. the appearance information **c** required to complete the vector **a** is missing (indeed, this is the information which eventually must be estimated).

Fortunately a solution can be found using a Dual-Input HMM, which conveniently combines both appearance and speech information, allowing estimation of an appropriate appearance parameter HMM state sequence given an observation sequence of speech parameters.

7.2 Constructing a Dual-Input HMM

Dual-input HMMs were first described by Brand in [20]. In his work audio parameters are mapped to visual shape and velocity parameters through an entropic HMM, allowing estimation of the hidden visual state sequence from a new speech observation. In this thesis the method differs both in the type of HMM used (a standard HMM is employed here), and in the parameters used for training - which do not model velocity, but do model texture (as opposed to only shape). The process used after the HMM stage for calculating visual parameters from a state sequence also differs significantly from that employed by Brand. In this thesis a novel approach akin to *unit based selection* is used to calculate the final visual parameters (see Chapter 8).

The Dual-Input strategy employed in this thesis is based on the following premise. The set of all possible mouths can be considered to lie on a low dimensional manifold, and may be segmented into different categories (e.g., these categories may relate to Visemes). When a person speaks, a trajectory is formed through the manifold, intersecting one of the mouth categories at each time t. Using a standard HMM, the mouth categories may be considered as HMM states, and the *hidden* HMM process considered to be the sequence of states visited when a person speaks. The observations from the HMM then relate to the mouth displayed at each state visit.

This idea may be expressed further by considering a HMM built from an appearance parameter training set for a facial area. When using a mouth appearance parameter training set, the HMM observations relate to appearance parameters at time t (the mouth observed visually), and the state sequence relates to the Gaussian mixture (or cluster) which best explains the parameter. Figure 7.1 shows a 120 state HMM constructed using the full mouth appearance training set from hierarchical

model 2. Given a new observation of appearance parameters the Viterbi algorithm may then be used to find the associated state sequence (problem 2). However, in the context of speech driven animation, the only available observation signal is the input speech signal. Hence, a method must be found to find the best appearance based HMM state sequence using a speech observation.

A Dual-Input HMM allows this problem to be solved by creating new *speech based* means and covariances for each *appearance* HMM state. The Viterbi algorithm may then be applied to find the best *appearance* HMM state sequence (or cluster sequence) using a new *speech* observation.

The construction process is as follows. First a k state HMM defined $\lambda_A = (\zeta_A, \Upsilon_A, \pi_A)$ is constructed (see Section 7.1.1 for more details) using an appearance parameter training set. The means and covariance's of each state in λ_A are defined as $\bar{\mu}_A^j$ and $\bar{\Sigma}_A^j$, where $j = 1, \ldots k$.

After training this HMM, the matrix $\gamma_t(j)$ is also automatically obtained, which defines the probability of being in state j at time t. Note that during training, $T = N$, since the length of the observation sequence is equal to the number of training vectors.

Using $\gamma_t(j)$, the new k state HMM $\lambda_M = (\zeta_A, \Upsilon_A, \pi_A)$ may be constructed, with transition, observation, and prior probabilities equal to those in λ_A, but with means $\bar{\mu}_M^j$ and covariances $\bar{\Sigma}_M^j$ defined using a *speech* parameter training set such that

$$\bar{\mu}_M^j = \frac{\sum_{t=1}^{T} \gamma_t(j).\mathbf{b}_m^t}{\sum_{t=1}^{T} \gamma_t(j)} \qquad 1 \leq j \leq k, \;\; T = N \tag{7.5}$$

and

$$\bar{\Sigma}_M^j = \frac{\sum_{t=1}^{T} \gamma_t(j).(\mathbf{b}_m^t - \bar{\mu}_M^j)(\mathbf{b}_m^t - \bar{\mu}_M^j)^T}{\sum_{t=1}^{T} \gamma_t(j)} \qquad 1 \leq j \leq k, \;\; T = N \tag{7.6}$$

Given λ_M, and a sequence of new speech parameters, the Viterbi algorithm may now be used to find the best state sequence Q through λ_M. Given the shared transition, observation, and initial probabilities between λ_M and λ_A, and the fact that the the means and covariance's of λ_M are constructed using the *appearance* state observation matrix $\gamma_t(j)$, the state sequence Q now also corresponds to a state sequence through λ_A.

The state sequence Q, as well as defining a set of hidden states through both λ_M and λ_A, also relates to a unique set of speech mean and covariance matrices for each time t, and a unique set of appearance mean and covariance matrices for each time t. That is, the shared properties between λ_M and λ_A associate Q_t (i.e. the state selected at time t) with $\bar{\mu}_M^t$ and $\bar{\Sigma}_M^t$, as well as $\bar{\mu}_A^t$ and $\bar{\Sigma}_A^t$.

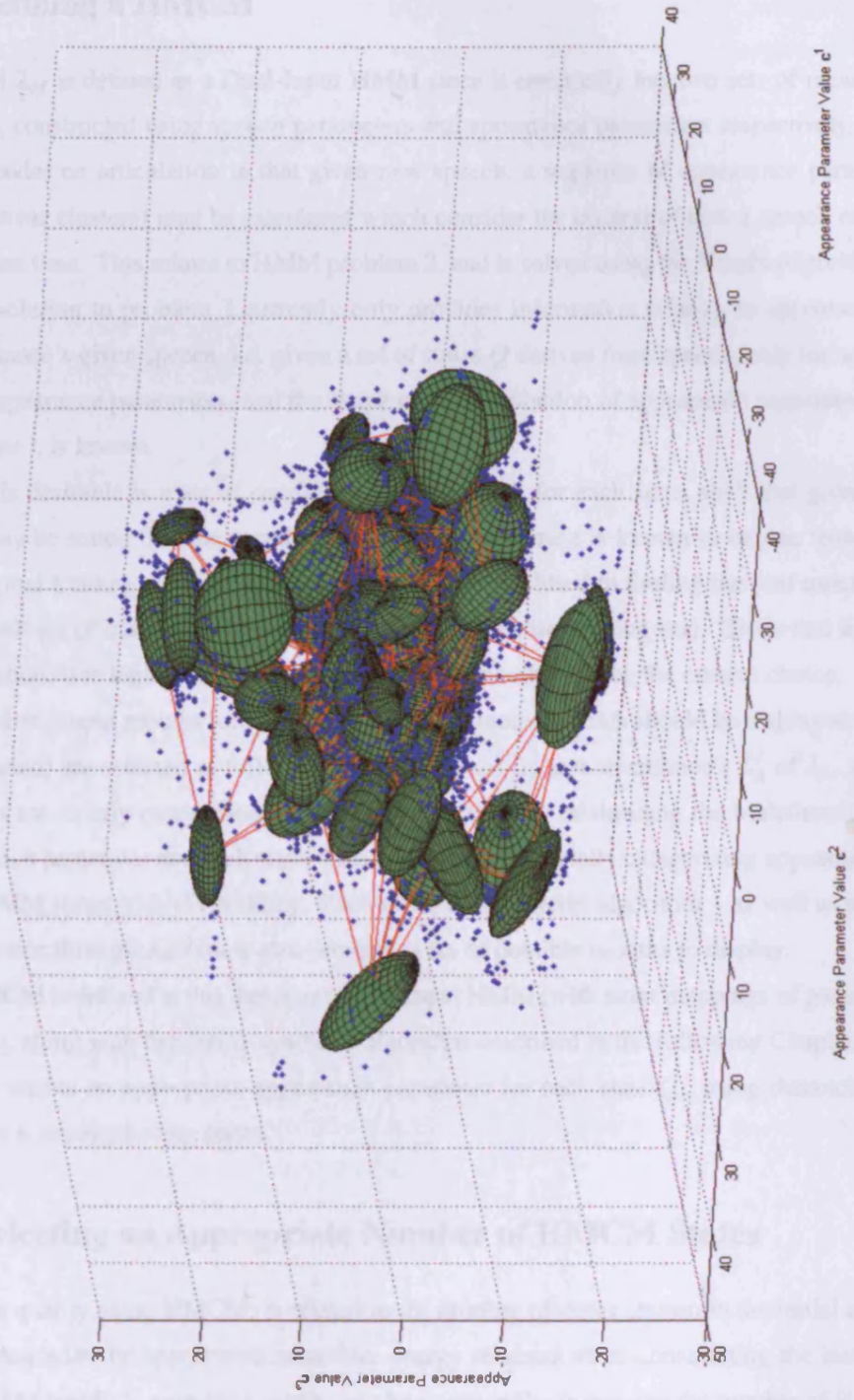

Figure 7.1: A HMM fitted to the mouth appearance parameter distribution of hierarchical model 2. Green ellipsoids represent Gaussians associated with a HMM state, while red lines represent transitions between states.

7.3 Defining a HMCM

The HMM λ_M is defined as a Dual-Input HMM since it essentially has two sets of means and covariance's, constructed using speech parameters and appearance parameters respectively. This new HMM encodes co-articulation in that given new speech, a sequence of appearance parameter (visual) states (or clusters) may be calculated which consider the context of how a speech observation evolves over time. This relates to HMM problem 2, and is solved using the Viterbi Algorithm. However, the solution to problem 2 currently only provides information relating to appearance means and covariance's given speech, i.e. given a set of states Q derived from speech, only the sequence of average appearance parameters, and the shape of the distribution of appearance parameters for each state at time t, is known.

What is desirable is a set of *candidate* output mouths for each state, such that given a speech input it may be stated that the mouth to be displayed at time t is known to be one from a sub-set of the original training set. The problem may then be simplified to finding the best mouth at time t from a small set of candidates (as opposed to from the entire training set). Given that λ_M encodes co-articulation, it is highly probable that one of these mouths will be the correct choice.

Candidate output mouths for each state (i.e. likely mouths which should be displayed if a certain state is visited) are selected as follows. Given the means $\bar{\mu}_A^j$ and covariance's $\bar{\Sigma}_A^j$ of λ_A, appearance parameters are strictly categorised into one of the k states by minimizing the Mahalanobis distance between each parameter and each state. This is equivalent to strictly categorising appearance parameters to GMM states in SAM building. The output of the Viterbi algorithm – as well as providing a state sequence through λ_M – now also provides a set of possible mouths to display.

A HMCM is defined in this thesis as a Dual-Input HMM (with strict mappings of parameters **a** to each state), along with the *Trellis* synthesis algorithm described in the following Chapter. This new algorithm selects an appropriate appearance parameter for each state Q_t, using distances between parameters **a** in neighboring states.

7.4 Selecting an Appropriate Number of HMCM States

Animation quality using HMCMs is related to the number of states chosen in the initial appearance HMM, along with the appearance parameter energy retained when constructing the initial model. As with SAM building, animation quality can be improved by increasing the number of HMM states

(providing the data is not over-fitted with states). However, animation quality is also increased by retaining a high amount of appearance parameter energy in the HMM. Unfortunately, as with SAM construction, it is computationally very expensive to construct a high state HMCM which contains a large amount of appearance parameter energy. The same trade off - between the number of clusters/states versus the amount of energy contained - posed by SAM GMM construction is therefore also faced in HMCM HMM construction, and the same negative log-likelihood minimisation strategy employed (see Section 5.2.2).

Chapter 11 thoroughly evaluates the HMCM with respect to visual-speech synthesis, and contains results of HMM construction experiments.

7.5 Dual Input HMM (and initial HMCM) Construction Summary

A summary is now given for initial HMCM construction.

1. Given an appearance parameter training set with 20ms observations, build the HMM λ_A, and retain the matrix $\gamma_t(j)$.

2. Using $\gamma_t(j)$, the speech training set corresponding to the appearance training set used in λ_A construction, and equations 7.5 and 7.6, calculate new HMM speech means $\bar{\mu}_M^j$ and covariances $\bar{\Sigma}_M^j$.

3. Using the new means and covariances define the Dual-Input HMM λ_M, with transition, observation and prior probabilities equal to those in λ_A.

4. Using λ_A, classify appearance parameters for each observation to a state by minimising the Mahalanobis distance.

5. Using the one-to-one speech-and-appearance correspondences from the training set, form clusters of vectors **a** for each HMM state.

Chapter 8

Synthesis and Animation using HMCMs

This Chapter describes a new algorithm for synthesising appearance parameters using a HMCM. The method improves upon SAM synthesis by further enforcing co-articulation in animations.

8.1 Re-addressing SAM Cluster Synthesis

The SAM synthesis method assumes there exist locally linear relationships between appearance and speech, and estimates appearance by projecting speech parameters onto linear PCA axis. Theoretically, the estimation method is accurate since there exists a closed form solution to a set of linear equations. However, similar appearance parameters in clusters can have entirely different speech parameters associated with them (due to clustering being performed on appearance parameters alone). This clustering attribute can affect SAM synthesis, as the fitting of a least squares axis (i.e. performing PCA) to a group of uncorrelated speech vectors is likely to be a poor approximation - since the chance of the distribution being linear in such an instance is lower. Thus projection onto, and then back off these axis, will give inaccurate results. However, reconsideration of the appearance-to-speech problem given a HMCM highlights certain new constraints on the problem, which may be exploited in order to create better animations.

The Dual-Input HMM, given new speech, provides a group of candidate vectors **a** for each time t. That is, for each time t, it is known that the most probable output vector is one of those allocated to state Q_t. Using SAM Mahalanobis cluster selection, any of the k clusters could be chosen at each time t. Therefore, it could be argued that the output parameter at time t could be selected from any of the parameters in the training set, highlighting a potential cause of animation errors since this

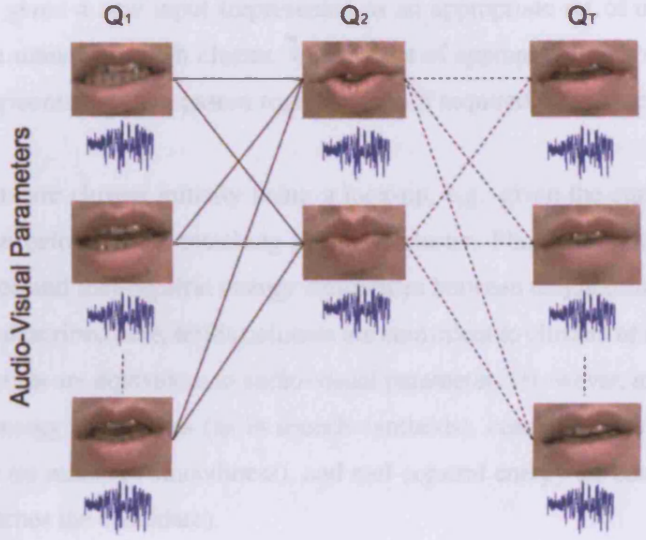

Figure 8.1: A Trellis of audio-visual parameters **a**. Each column contains the parameters **a** associated with the HMM state chosen for time t using the Viterbi algorithm. For synthesis, errors are allocated to each node (audio visual parameter) to represent the cost of visiting a particular parameter at time t. For each column, the visual parameter associated the lowest error is displayed at time t.

approach is in conflict with the rules of co-articulation.

The output of the Dual-Input HMM is a sequence of clusters with parameters $\mathbf{a} = [\mathbf{c}, \mathbf{b}_m]^T$, and can be represented as a trellis data structure. Figure 8.1 shows such a structure, where each column represents the parameters **a** associated with the cluster state Q_t. The synthesis problem may now be redefined as finding the best vector **a** (and therefore the best appearance parameter **c**), from each column at time t. In other words, the problem is now to find the best *path* through the trellis, for $1 \leq t \leq T$.

This trellis search strategy is comparable to *unit based selection* (or selection based synthesis). Unit based selection is typically employed in speech synthesis to create synthetic voices given a set of phonemes, typically generated using a TTS system [8, 7]. Given a phonetically labeled speech training set, it is first segmented into individual phones or diphones (sub-word units), and similar units clustered such that each cluster contains a set of similar phones or diphones. Within clusters, units are differentiated by factors such as whether or not a phoneme appears at the beginning or end of a word, and differences in prosodics (emphasis). Each of the clustered units corresponds to an

audio segment, and given a new input (represented as an appropriate set of units) the problem is to select appropriate units from each cluster. Given a set of appropriately chosen units, the corresponding speech segments are then pasted together (and if required, also time-warped) to create a synthetic voice.

Appropriate units are chosen initially using a look-up, e.g. given the current input phoneme, candidates for output belong to the matching phoneme cluster. Phonemes within clusters are then chosen based on pitch and mel-cepstral energy similarities between neighboring phonemes.

In the algorithm described here, trellis columns are equivalent to clusters of candidate phonemes, and candidate phonemes are equivalent to audio-visual parameters. However, instead of considering pitch and spectral energy similarities (as in speech synthesis), considerations now become visual parameter similarity (to maintain smoothness), and mel-cepstral energy (to ensure the input speech segment closely matches the candidate).

Unit based selection has already been used successfully for audio-visual synthesis in the work of Cosatto *et al* [40]. However, in this work, phonemes are used for look up in a similar way to the speech synthesis method already described, and speech is generated from a set of input phonemes generated using a TTS system. Output is then delivered as a set of visual parameter as opposed to previously recorded speech segments. Note that these visual parameters are not of a uniform length, and may also be time-warped (again, in a similar way to speech synthesis). Also note that coarticulation is less of an issue in the work of Cosatto *et al*, since the input data (a set of phonemes) already inherently encodes coarticulation. The problem addressed by Cosatto *et al* is primarly concerned with unit-based selection, as opposed to coarticulation.

Another application of unit based selection is in the dynamic delivery of animation in response to application data [141]. A series of application events can be used to select a set of candidate output audio-visual animations to give feedback to a user. Appropriate candidates may then be chosen for each output time segment by blending similar neighboring candidates.

8.2 A Trellis Based HMCM Search

Given a trellis of parameters $\mathbf{a}$, and a new input speech signal $M = \mathbf{b}_m^1 \mathbf{b}_m^2 \ldots \mathbf{b}_m^T$, the animation problem is defined as finding the best path through the trellis minimising some error function E. Two properties are desirable in such an error function, 1) that the distances between appearance parameters chosen at time t, and adjacent parameters, are minimised, and 2) that the trellis speech

parameter chosen at time t is similar to the input speech parameter at time t. The first criteria ensures that smoothness is considered in the animations, while the second criteria ensures that selected appearance parameters are associated with speech parameters similar to the input signal. If the Dual-Input HMM is satisfactorily encoding co-articulation, then speech parameters associated with appearance parameters in trellis columns at time t will be similar to the input speech parameter at time t.

By assigning error costs to each cell (or parameter) in the trellis based on the criteria defined, the lowest costing path - and hence the optimal animation trajectory - may be chosen by selecting the cell in each column at each time with the lowest error, for $1 \leq t \leq T$. Using the covariances Σ_M^j of λ_M, an error E may be assigned to cells in the column at $t=1$ using

$$E(\mathbf{a}_i, Q_1) = ((\mathbf{b}_m^i - \mathbf{b}_m^1)^T \Sigma_M^{Q_1\,-1} (\mathbf{b}_m^i - \mathbf{b}_m^1)) \qquad i = 1, \ldots, p \tag{8.1}$$

where $E(\mathbf{a}_i, Q_1)$ is the error associated with parameter $\mathbf{a}_i$ in state Q_1, $\mathbf{b}_m^i$ is the speech part of parameter $\mathbf{a}_i$, $\mathbf{b}_m^1$ is the input speech vector observed at time $t=1$, and p is the number of parameters $\mathbf{a}_i$ in state Q_1 (i.e. the number of nodes in the first trellis column).

With this formulation it can be seen that errors are simply allocated to the first state based on the similarity of the speech part $\mathbf{b}_m^j$ of $\mathbf{a}_i$, to the input $\mathbf{b}_m^1$. Thus, smoothness is not considered in this calculation. Rather, the initial error calculation for time t serves more to *prior* the trellis with errors for further calculations.

Using the covariances Σ_A^i of λ_A, and the covariances Σ_M^i of λ_M, an error E may be assigned to cells in columns for $2 \leq t \leq T$ with the following formulation

$$E(\mathbf{a}_j, Q_t) = \left[\sum_{i=1}^{r} (\mathbf{c}_{t-1}^i - \mathbf{c}_t^j) \Sigma_A^{Q_t\,-1} (\mathbf{c}_{t-1}^i - \mathbf{c}_t^j) \right] \times ((\mathbf{b}_m^j - \mathbf{b}_m^1)^T \Sigma_M^{Q_1\,-1} (\mathbf{b}_m^j - \mathbf{b}_m^1)) \qquad j = 1, \ldots, p \tag{8.2}$$

where $E(\mathbf{a}_j, Q_t)$ is the error associated with parameter $\mathbf{a}_j$ in state Q_j, p is the number of parameters $\mathbf{a}_j$ in state Q_t, r is the number of parameters $\mathbf{a}_i$ in state Q_{t-1}, $\mathbf{c}_{t-1}^i$ is the appearance part of parameter $\mathbf{a}_i$ in state Q_{t-1} and $\mathbf{c}_t^j$ is the appearance part of parameter $\mathbf{a}_j$ in state Q_t.

Figure 8.2 demonstrates the calculation for $E(\mathbf{a}_j, Q_t)$. Essentially, node errors are calculated by summing the distances between a nodes appearance part, and the appearance parts of nodes in the previous column; and then multiplying by the distance between the current speech segment,

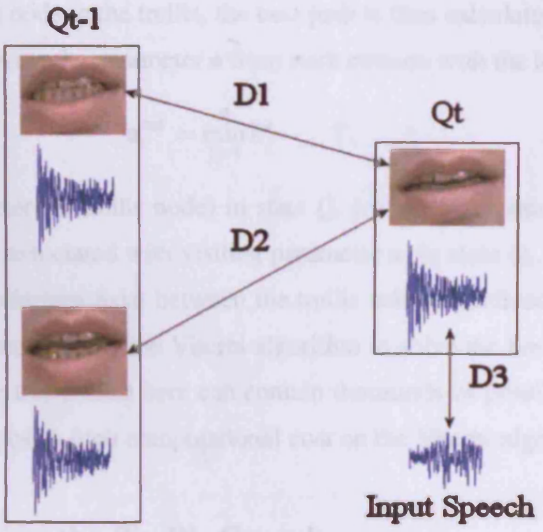

Figure 8.2: A visual representation of error calculation for nodes in the trellis (see Figure 8.1) at time *t*. Errors associated with a node are calculated by summing the mahalanobis distances between the visual parameter in a node at time *t*, and each node at time *t-1*. The resulting sum is then multiplied by the distance between the speech parameter in a node at time *t*, and the input speech parameter observed at time *t*. The overall error in this example is therefore $E = (D1 + D2) \times D3$.

and the speech part of the current node. Errors for a node therefore implicitly encode information concerning all the nodes in the previous column, making calculation of a best path feasible. Without encoding information about previous nodes, the solution would be computationally unrealistic, since solving the best path would involve considering every single possible path through the trellis.

This scheme of error calculation is a heuristic approach inspired by the forward-part of the forward-backward algorithm [4, 5]. In this procedure nodes represent states, and an optimal state sequence is sought through a trellis of states (similar to the Viterbi algorithm [131]). The cost of visiting a state at time *t* is defined as the sum of the probabilities of travelling to that state from each state at time *t-1* (the transition probabilities), multiplied by the probability that the output observation at time *t* belongs to the current state. In the approach described here, transition probabilities are represented using distances between neighbouring mouths. The output observation is defined here as a speech vector, and instead of calculating the probability that the observation belongs to some state, the distance between the observed speech vector and the speech vector associated with the current visual feature is used.

Given errors for each node in the trellis, the best path is then calculated by working backwards, from $T \geq t \geq 1$, and choosing the parameter **a** from each column with the lowest error value. Hence,

$$\mathbf{a}_t^{out} = \min_{j=1}^{n} E_j^t \qquad T, \ldots, 1 \tag{8.3}$$

where $\mathbf{a}_t^{out}$ is the parameter (or trellis node) in state Q_t (or trellis column t) with the lowest error value, and E_j^t is the error associated with visiting parameter $\mathbf{a}_j$ in state Q_t.

Note that certain similarities exist between the trellis solution defined here and the Viterbi algorithm. It is also possible to apply the Viterbi algorithm to solve the best path through the trellis. However, columns in the trellis used here can contain thousands of possible candidate mouths for each time t. This can impose a high computational cost on the Viterbi algorithm.

8.3 Post-processing the Trellis Search

Using the calculated parameters $\mathbf{a}_t^{out}$, the appearance part $\mathbf{c}_t$ of each vector can be extracted and used for animation of the mouth.

From a closer examination of the HMCM trellis algorithm, it is apparent that that the method essentially re-orders data from the training set for animations. Therefore, although smoothness is considered in the search, it is still largely dependent on the content and size of the training set, i.e. a large training set will usually require less smoothing than a smaller training set. Hence, as with SAM synthesis, post-processing is employed to intelligently smooth the signal.

Noise characteristics in a HMCM signal differ from that of a SAM signal. In a SAM signal noise is characterised by large changes over a short time period (typically 1 frame). This is caused by incorrect cluster choice. In a HMCM, coarticulation is encoded - thus cluster choice is less erratic and far more accurate. Therefore, large changes in a HMCM signal are not related to noise, and are instead associated with important articulatory detail.

The choice of filter used to smooth the animation signal should discriminate between articulatory detail and spurious noise. Important detail in the signal, corresponding to events such as fast changes in the animation due to the onset of certain speech events, should be preserved in the output animations. Noise in the signal during periods of mouth *inactivity* (such as when the mouth is at rest) results in artifacts such as texture flicker in output animations. In considering a filter, it is necessary to ensure that articulatory detail is not mistaken for noise and thus eliminated.

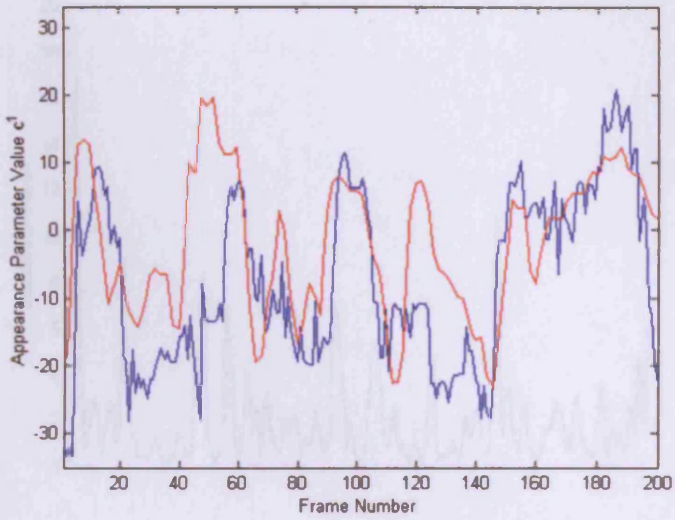

Figure 8.3: An appearance parameter trajectory synthesised by a HMCM before filtering. The blue line represents the synthetic trajectory, while the red line represents its ground truth.

Figure 8.3 shows a mouth appearance parameter trajectory after HMCM synthesis (blue line) in comparison to its ground truth (red line). Note that the signal is generally quite noisy, especially in areas where we might expect the signal to otherwise remain steady (e.g. between frames 20 and 30). These steady periods are associated with parts of the animation where the mouth might remain still, or hold a certain pose for a short time. Noise can be reduced in the signal (and consequently in the animation) by averaging it over a small window. However, this is somewhat of an *ad-hoc* approach which causes resulting animations to appear *sluggish* - i.e. although noise and texture flicker is removed, the timing of co-articulation is damaged and the magnitude of key mouth poses reduced.

A better approach to reducing noise and retaining detail is to filter the appearance parameter signal based on its *local standard deviation*. If the standard deviation of the signal over a small window is high, then this would be related to an important part of the signal which should be preserved, e.g. the onset of an articulatory event such as a plosive (i.e. mouth-closed to mouth-open over a very short time span). Noise will generally be associated with a low change in standard deviation over a short window, i.e. where the signal is fluxuating by a small amount over a short period.

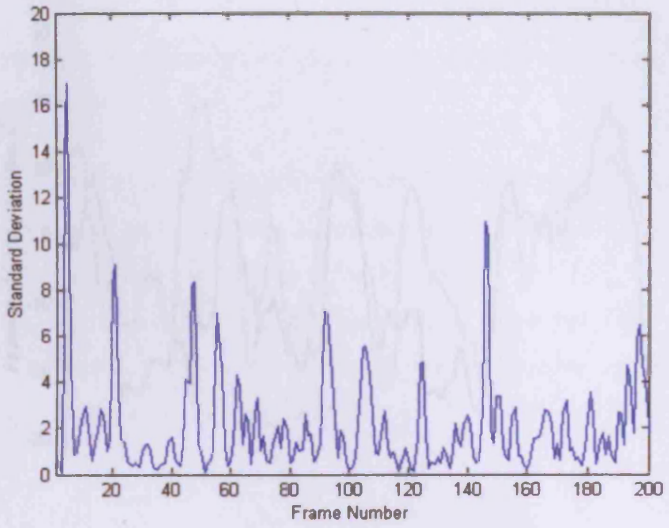

Figure 8.4: The standard deviation of the synthetic signal shown in Figure 8.3, calculated using a local 3-frame window.

The aim is therefore to smooth the signal by different amounts depending on whether the local standard deviation is high or low. If the standard deviation is high, then the signal is smoothed by only a small amount. If the standard deviation is low then the signal is smoothed by a greater amount.

Figure 8.4 shows the local standard deviation of the signal in Figure 8.3, where each value represents the standard deviation of the signal at time *t-1*, *t* and *t+1* (i.e. a 3 frame window). Note that the standard deviation in noisy parts of the signal is low, compared to a high standard deviation in regions of large change.

Once the standard deviation has been calculated, the entire signal is then normalised between zero and 1, and its inverse taken. The result of this is simply that large values of standard deviation are now attributed with noisy regions, are small values with parts that should be preserved.

This new set of values is used to filter the appearance parameter signal using a 1D Gaussian convolution mask 3 frames wide. The standard deviation of the Gaussian filter in a given local window is equivalent to the corresponding normalised-inversed standard deviation value for that window. Therefore, a large (in magnitude) Gaussian filter will be applied to noisy regions (thus smoothing it), and a small (in magnitude) Gaussian filter will be applied to regions expected to

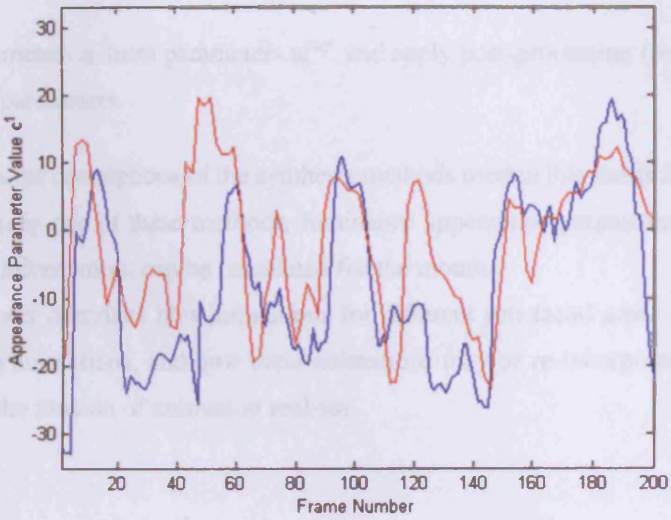

Figure 8.5: The synthetic HMCM appearance trajectory (blue line) after filtering in comparison to its ground truth (red line).

contain important articulatory detail (thus preserving the signal).

Figure 8.5 shows the appearance trajectory from Figure 8.3 (blue line) in comparison to its ground truth (red line) after filtering. After filtering, the stability of the signal is generally far more satisfactory. Note the reduction in noise across regions where the signal would normally be expected to be still (e.g. from frames 20-30), and the preservation of fast changes in the signal associated with articulatory detail.

8.4 Trellis Summary

A summary of the trellis search is now given.

1. Calculate the best state sequence Q through the HMCM using the input speech signal b^i_M, where $t = 1,\ldots,T$.
2. Using Q, construct a trellis of parameters **a**.
3. Assign errors to each trellis node at $t=1$, using equation (8.1).
4. Assign errors to each trellis node at $2 \leq t \leq T$, using equation (8.2).

5. Choose the best parameter $\mathbf{a}_t^{out}$, for $T \geq 1 \geq t$, using equation (8.3).

6. Extract parameters $\mathbf{c}_t$ from parameters $\mathbf{a}_t^{out}$, and apply post-processing (see Section 8.3) to the appearance parameters.

This concludes the descriptions of the synthesis methods used in this thesis for producing visual-speech. Hence, using one of these methods, luminance appearance parameters, and consequently shape and texture information, can be calculated for the mouth.

The next Chapter describes how animations for different sub-facial areas are reconstructed to produce full facial animations, and how these animations may be re-incorporated into the training video to increase the illusion of animation realism.

Chapter 9

Reconstruction and Display

The previous four Chapters described how SAMs and HMCMs may be used for calculating luminance appearance parameters from input speech parameters for animation of the mouth. In the Chapter 10, manual estimation of luminance appearance parameters, via a key-framing based approach, is also considered.

This Chapter describes how sub-facial animations may be merged to create full-facial animations, and how the full-facial animations may be re-integrated into background footage to increase the illusion of realism.

9.1 Facial Reconstruction

Irrespective of the appearance parameter synthesis method used, either automatic or manual, luminance-appearance parameters are now available for animation of different sub-facial areas. If a sub-facial area has not been animated, then the appearance parameter values for that area may be set to zero for each animation frame. The effect of this is simply that the mean shape free image for that sub-facial area will be used instead.

Using appearance parameters, luminance images and shape coordinate vectors may be calculated for a sub-facial area using equations (4.26) and (4.27). Luminance vectors are then used to access hue and saturation vectors in a sub-facial models look-up table. Since HMCMs essentially re-order frames from the training set, using a look-up table to retrieve colour has no adverse effect on animation quality. SAMs generate new luminance images upon synthesis, which are not guaranteed to perfectly resemble images from the training corpus. However, this also does not cause

a significant reduction in the quality of output animations since good matches are still commonly found.

The only artifact which can result from using this look-up table method is a lack of smoothness in the animations. In order to account for this possibility, the inverse-standard deviation Gaussian filter approach described in Section 8.3 is applied to sub-facial output images.

At this point it is now assumed that shape, luminance, hue, saturation sub-facial image vectors are available for output. The goal of reconstruction is to merge each sub-facial animation to produce a full-facial animation. Essentially what is required is facial luminance, saturation, hue and shape information which contains all the information from the sub-facial areas.

9.1.1 Sub-Facial Warping

Bregler *et al* [22] and Cosatto *et al* [40] create facial animations by warping individually animated facial areas over target images. Bregler *et al* animate solely the mouth region, and warp onto original background footage, while Cosatto *et al* perform a similar task using multiple sub-facial region animations. Both methods use prior pose and shape information relating to the likely position of the sub-facial area on the target image to direct warping. Much therefore depends on the accuracy of this prior knowledge, and inaccuracy can cause animations to *float* over the background image.

The problem in terms of the hierarchical model is initially somewhat different in that synthesised regions are first merged with other synthesised regions. Warping information at this stage is typically very accurate. For example, when warping a mouth image onto a lower-face image, the position of the mean mouth on the lower-face is already known, since a lower face node contains landmarks for the mouth, the jaw line and the nose. The mouth texture being warped may then be simply warped over the mean mouth position on the lower face. Thus there is no danger of the mouth floating over the lower-face, as it will be firmly fixed onto a robust target. Shape may be integrated by substituting coordinates for the mouth region present in the lower-face node, with shape coordinates calculated for the new mouth.

A warping-based procedure for reconstructing a face using a hierarchical model may be summarised in two steps. First, synthesised luminance, hue and saturation vectors for each node are warped together in a top-down manner. The left and right eyebrows are merged with the mean shape-free face first (i.e. the mean face luminance, hue and saturation image). The left and right

eyes are then merged over the left and right eyebrows (in the partially completed face). The lower-face is then merged with the partially completed face, and finally the mouth merged with the lower face. This creates a completed facial image. Full facial shape is then constructed by calculating an *offset* vector of shape values, and applying them to the face nodes mean shape. In order to successfully reproduce eyebrow movement, anchor points are retained from the mean face shape. Eyebrow shape also has precedence over eye shape, resulting in shape generated by the eye nodes being ignored. Finally, mouth node shape also has precedence over any mouth shape generated by the lower face. To summarise:

1. Construct luminance, hue and saturation shape-free vectors for each hierarchical node, including the root node (full face). The root node vectors are the mean face vectors. Note that *all* sub-facial images at this point are still *shape-free*.

2. In a top-down manner (from the first child level to the bottom child level), warp each child node shape-free texture onto its corresponding facial area on the parent node shape-free texture.

The final merged luminance, hue and saturation facial images are shape-free with respect to the mean facial shape. The next step is to calculate a new shape vector which combines all the synthesised sub-facial area shape information.

1. Construct the vector $\mathbf{x}_{concat}$, by concatenating each sub-facial shape vector such that its coordinate ordering corresponds to the coordinate ordering in the mean face shape, i.e. each coordinate in $\mathbf{x}_{concat}$ should correspond to the same coordinate in the mean face shape. Eyebrow node shape has precedence over eye node shape. Similarly, mouth node shape has precedence of mouth data synthesised by the lower-face node. The result is that eye node shape is ignored, as well as mouth shape generated by the lower-face.

2. Construct the vector $\bar{\mathbf{x}}_{concat}$, by concatenating each mean sub-facial shape vector such that its coordinate ordering corresponds to the coordinate ordering in the mean face shape (see step 1). Precedence between nodes is adhered to as in step 1.

3. Calculate the offset vector $\mathbf{x}_{offset} = \mathbf{x}_{concat} - \bar{\mathbf{x}}_{concat}$.

4. Calculate the final output facial shape vector $\mathbf{x}_{final} = \mathbf{x}_{offset} + \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the mean facial shape. Certain key *anchor* shape coordinates in the resulting face vector are given values from the original mean face node shape. These are highlighted in Figure 9.2.

Shape reproduction is therefore achieved by adding the offset between a sub-facial shape and its mean, to the respective sub-facial area on the mean face shape. As described, eye node shape data is not used. Instead, eyebrow shape data has precedence. Certain eyebrow landmarks are also treated as a special case, and are substituted for coordinates on the mean face shape. These special cases act as anchor points for the eyebrows. Figures 9.1 and 9.2 visually illustrate these texture and shape reconstruction procedures.

By examining the example final output image in Figure 9.1, it is clear that the method described thus far does not produce an entirely satisfactory output. The final image shown includes artifacts such as inconsistent lighting variation across certain sub-facial areas, and inaccurate warping of child sub-facial areas over parent facial areas. Lighting variation across sub-facial areas is caused by the fact that the images used to train each sub-facial area are normalised with respect to *each other*, and not with respect to a global illumination. This means it is never guaranteed that the illumination rendered for one facial area, will match that of a neighboring area.

Inaccurate warping artifacts are caused primarily by inaccurate landmark placement in the training corpus, or head pose variations which introduce non-linearity into the models. Unfortunately, without extending into 3D, this latter problem is difficult to solve with respect to the warping technique described.

In order to address these warping artifacts, a novel procedure called *Parent Approximation* is applied to the combined face textures. The method is surprisingly simple, but produces very satisfactory results.

9.2 Parent Approximation

In Parent Approximation, texture information generated by a child node, is approximated by its parent, i.e. a new parent texture is estimated which contains the texture of its child. This is not the same as simply warping the child texture over the parent texture. In this previous Section it was seen that this approach can cause warping artifacts. The new texture generated by parent approximation is a single continuous texture which contains a representation of what the combined child and parent

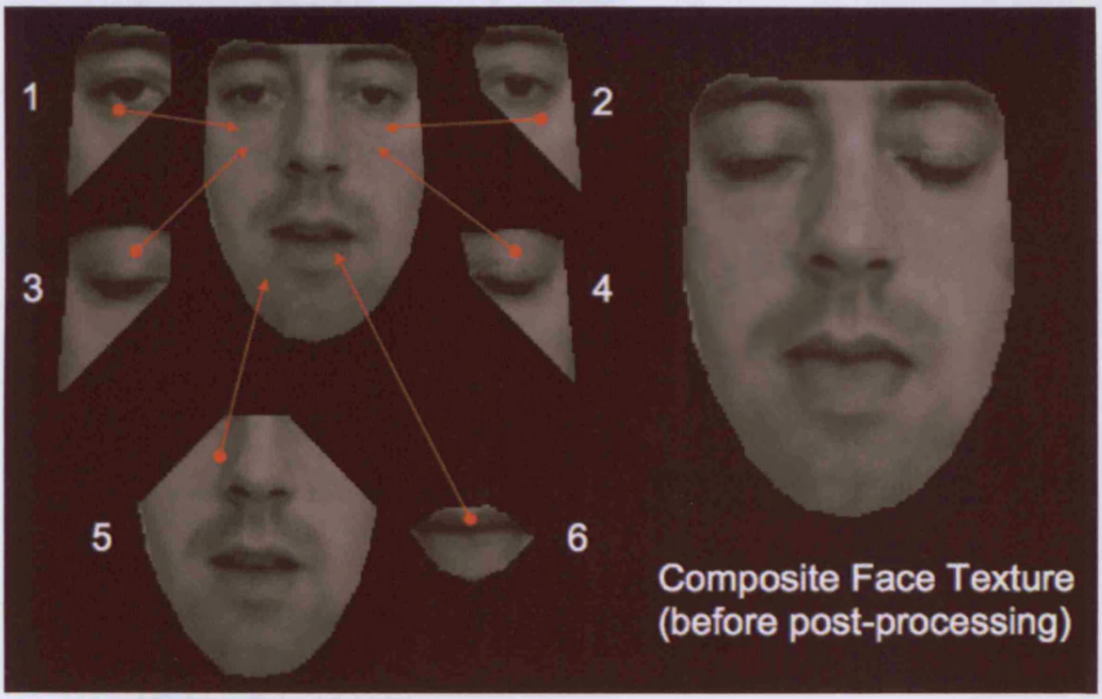

Figure 9.1: An initial face texture is reconstructed by warping synthesised sub-facial areas over the mean shape free face texture. The ordering is follows: (1) warp the left eyebrow texture over the mean face, (2) warp the right eyebrow texture over the mean face, (3) warp the left eye texture over the left eyebrow texture (now on the mean face), (4) warp the right eye texture over the right eyebrow texture (now on the mean face), (5) warp the lower face texture over the mean face, (6) warp the mouth texture over the lower face texture (now on the mean face). At this point, all sub-facial textures are shape-free. The constructed face texture at this stage is not entirely satisfactory, and may contain illumination and warping artifacts. Note illumination variation between the right eye texture and the lower face texture in the reconstructed face. Warping artifacts are also visible along the top lip and over the right eye.

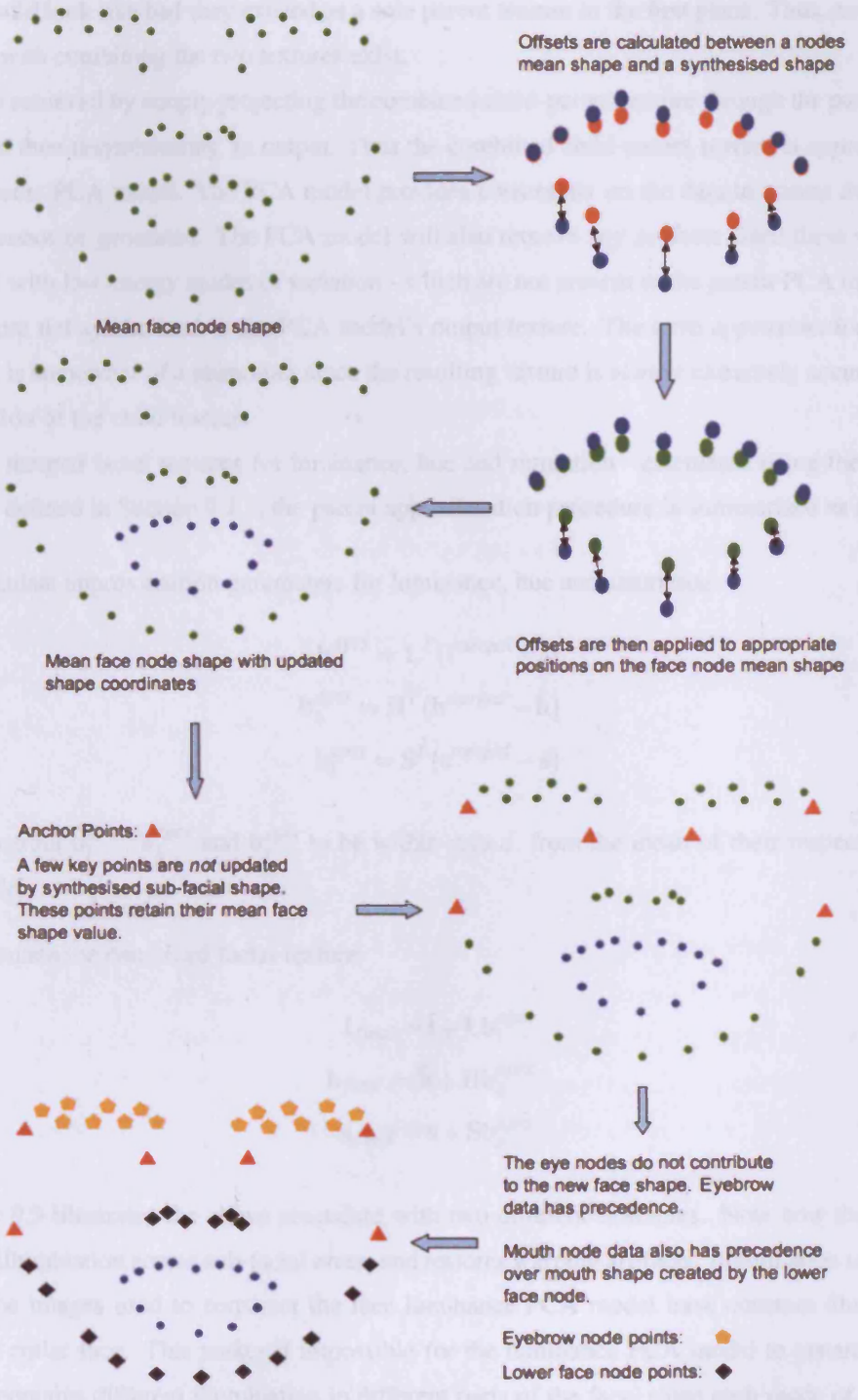

Figure 9.2: Shape is reconstructed by offsetting mean face coordinates. Certain landmark information is also retained from the mean face shape (*anchor* points). Rules of precedence also apply, i.e. eyebrow node data has precedence over eye node data, and mouth node shape data has precedence over mouth shape data synthesised by the lower-face.

texture would look like had they existed as a sole parent texture in the first place. Thus, no artifacts attributed with combining the two textures exist.

This is achieved by simply projecting the combined child-parent texture through the parent PCA model, and then resynthesising an output. Thus the combined child-parent texture is *approximated* by the parents PCA model. The PCA model provides constraints on the data to ensure that illegal textures cannot be generated. The PCA model will also remove any artifacts since these would be associated with low energy modes of variation - which are not present in the parent PCA model -are are therefore not synthesised in the PCA model's output texture. The term *approximation*, as it is used here, is somewhat of a misnomer since the resulting texture is always extremely accurate in its interpretation of the child texture.

Given merged facial textures for luminance, hue and saturation - calculated using the warping procedure defined in Section 9.1.1, the parent approximation procedure is summarised as follows:

1. Calculate approximation parameters for luminance, hue and saturation

$$\mathbf{b}_l^{aprx} = \mathbf{L}^T(\mathbf{l}^{merged} - \bar{\mathbf{l}}) \tag{9.1}$$

$$\mathbf{b}_h^{aprx} = \mathbf{H}^T(\mathbf{h}^{merged} - \bar{\mathbf{h}}) \tag{9.2}$$

$$\mathbf{b}_s^{aprx} = \mathbf{S}^T(\mathbf{s}^{merged} - \bar{\mathbf{s}}) \tag{9.3}$$

2. Constrain $\mathbf{b}_l^{aprx}$, $\mathbf{b}_h^{aprx}$ and $\mathbf{b}_s^{aprx}$ to be within $\pm 2s.d.$ from the mean of their respective PCA model.

3. Calculate the combined facial texture

$$\mathbf{l}_{final} = \bar{\mathbf{l}} + \mathbf{L}\mathbf{b}_l^{aprx} \tag{9.4}$$

$$\mathbf{h}_{final} = \bar{\mathbf{h}} + \mathbf{H}\mathbf{b}_h^{aprx} \tag{9.5}$$

$$\mathbf{s}_{final} = \bar{\mathbf{s}} + \mathbf{S}\mathbf{b}_s^{aprx} \tag{9.6}$$

Figure 9.3 illustrates the above procedure with two different examples. Note how the method equalises illumination across sub-facial areas, and restores warping artifacts. Illumination is equalised because the images used to construct the face luminance PCA model have constant illumination across the entire face. This makes it impossible for the luminance PCA model to generate a texture with contains different illumination in different parts of the face, since each mode of the PCA

model accounts for variation in facial appearance. If illumination is not constant across the training set (perhaps due to incorrect normalisation), then modes can exist which account for illumination intensity. However, variation of these modes will only produce variation across the entire face, and not in individual facial regions.

After parent approximation, the merged luminance, hue and saturation textures $\mathbf{l}_{final}$, $\mathbf{h}_{final}$ and $\mathbf{s}_{final}$ are then warped from the mean face node shape to the combined synthesised shape $\mathbf{x}_{final}$.

9.3 Image Noise Simulation

One of the products of synthesising images with appearance models, or any PCA based model, is that the removal of certain (lower energy) modes of variation also removes noise. In many cases this may not be an issue. However, the aim in this thesis is to re-attach synthesised facial images with background footage, which inherently contains some form of noise. It is therefore necessary - in certain cases - that in order to retain the illusion of realism some form of noise needs to be applied to them. Surprisingly effective results can be achieved by simply performing unsharp masking on the synthetic images. Figure 9.4 demonstrates two different synthetic images before unsharp marking, and after unsharp masking with the following filter

$$\frac{1}{(\alpha+1)}\begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix} \tag{9.7}$$

In both images the value of α is set to 0.1. The Figure highlights that while this step is not always necessary, in certain cases it can radically improve facial detail. In the image generated by hierarchical model 1 (top row), unsharp masking produces too much distortion in the output image, and is therefore not applied to synthetic animations. However,unsharp masking is applied to animations produced by hierarchical model 2, since they greatly improve image detail. The decision to apply unsharp masking is therefore primarily subjective, and is best on the ascetic quality of the resulting images. Experimentation with the value of α is also required in order to maximise this quality.

Figure 9.3: Image post-processing using parent approximation. Initial recombined facial images (left), facial images after parent approximation (middle), final images with colour included (right). Top row: note the illumination difference between the right eyebrow region and lower face region, and its subsequent normalisation. Bottom row: The primary error in the left figure is a mouth warping artifact, i.e. the join between the mouth and the lower face is corrupt. Parent approximation perfectly blends this join.

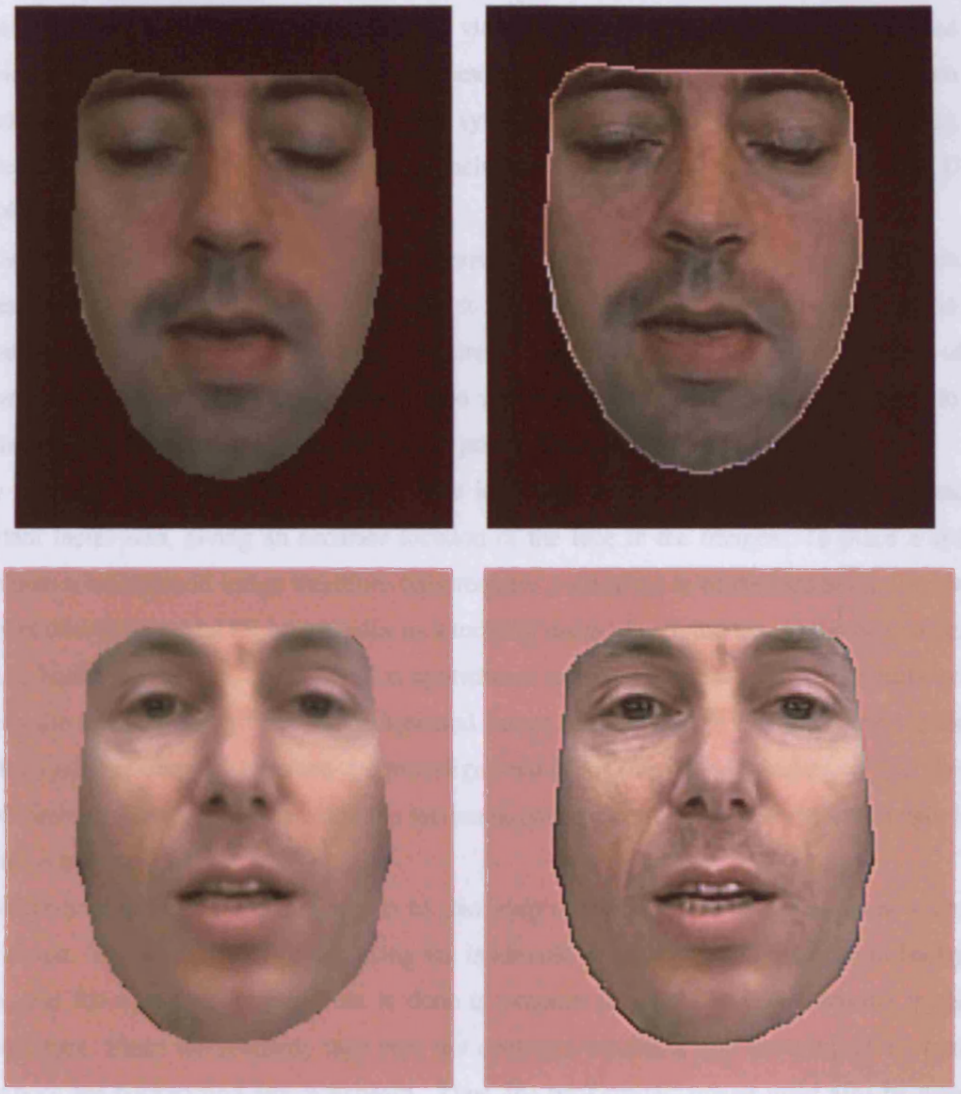

Figure 9.4: Unsharp masking to improve facial detail. The image produced by hierarchical model 2 (top right) appears slightly distorted after unsharp masking. This processing step is therefore not applied to hierarchical model 2 animations. The image produced by hierarchical model 1 (bottom right) is greatly enhanced by unsharp masking. This process is therefore applied by default to all hierarchical model 1 animations. The decision to apply unsharp masking is therefore based solely on ascetic considerations.

9.4 Background Addition

Full facial animations may now be constructed via the hierarchy. These images may be used alone for animation - in an application such as low-bandwidth communication - or re-inserted into background footage to give the impression that the synthetic animation is a real video recording. This technique has been adopted in several works, including Bregler *et al* [22], Ezzat *et al* [62], Cosatto *et al* [40] and Blanz and Vetter [11].

The common approach is to register synthesised images over their counterpart facial area on a segment of the original training video. In order to achieve this, information as to the position of the desired facial area in the training video is required. However, this is often not a problem of great measure, since in most cases the training video will already contain information related to facial area locations in the form of landmarks, and/or pose information.

In this case of this thesis the training video is already annotated with landmarks around each important facial area, giving an accurate location of the face in the images. To place a synthetic image onto a background image therefore only requires a mapping to be defined using the synthetic landmark coordinates, and the coordinates on a training image. Since the face in the original training frame is highly unlikely to correspond in appearance to the synthesised face, a straight warping between the synthetic image and the background image cannot be used - as this would incorrectly warp the synthetic image. Therefore, the best alignment of rotation θ and translation (t_x, t_y) between the synthesised shape coordinates, and the landmark coordinates on the background image, is used to define a mapping.

Before final insertion, one final step is to also warp the background image to the new synthetic facial shape. The warp is carried out using the landmarks associated with the current background image, and the new face shape. This is done to account for any eyebrow lowering in the new synthetic face. Since the synthetic face does not contain a forehead, any lowering of the eyebrows will expose the background image beneath. Thus, the background image must also be warped in order to map the eyebrows on that particular image to the eyebrows on the new synthetic image.

Once a synthesised image has been registered and inserted into a background image, its is then linearly blended around its borders to provide a smooth join. Pixels at the border of the synthetic image are smoothed heavily with the background, while pixels further away from the border are smoothed less. Blending is performed up to 10 pixels in from the edge of the border so as not to disrupt too much of the synthetic image's detail.

Figure 9.5 demonstrates the procedure for adding synthetic images on to background images. Note how the facial mask blends the inserted face into the background. Figure 9.6 shows the final image resulting from the procedure in Figure 9.6.

9.5 Summary

This Chapter has described a procedure for reconstructing full facial images from sub-facial areas, and attaching facial images to background images. The next Chapter describes how sub-facial areas may be animated by key-framing sub-facial appearance parameters.

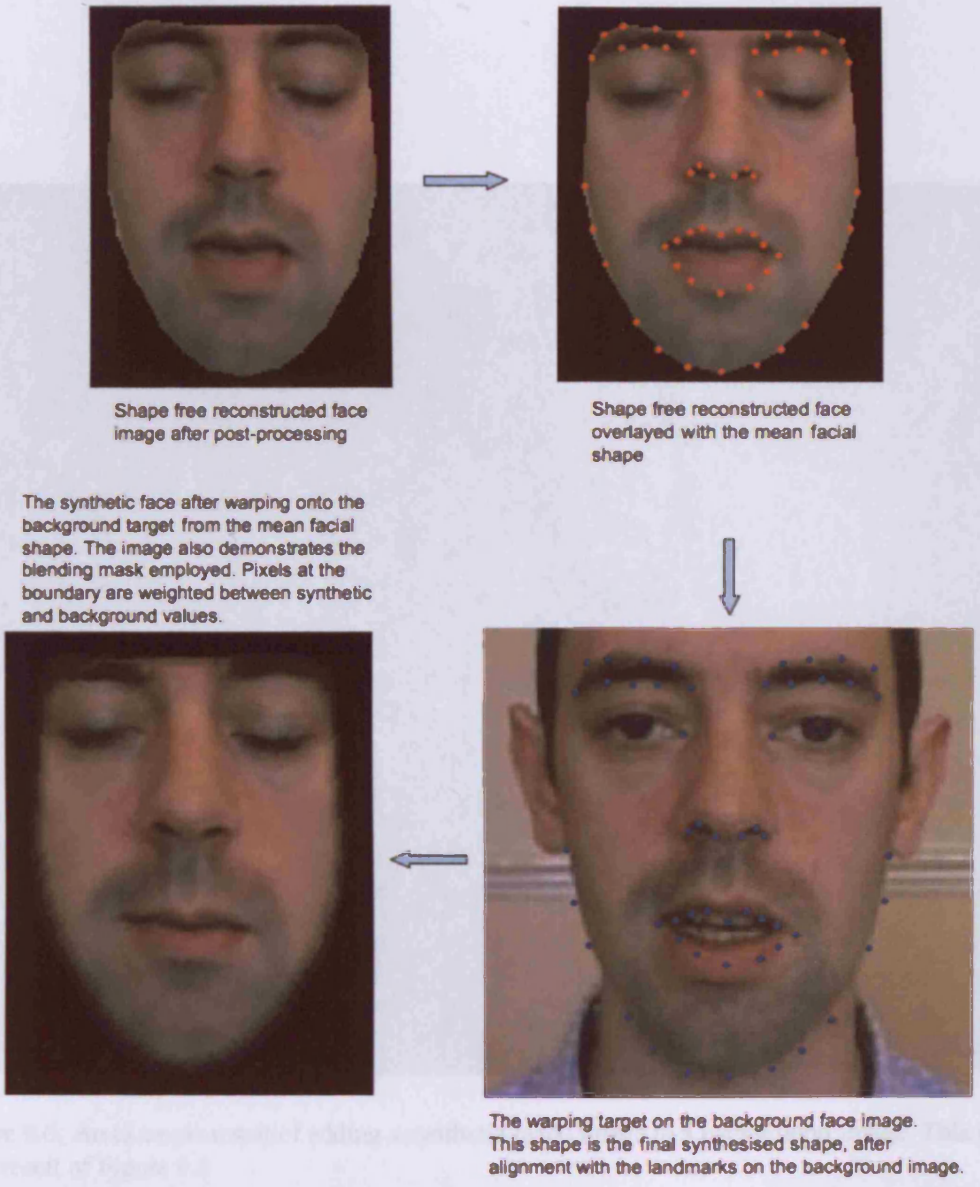

Figure 9.5: The procedure for adding synthetic face images to background images. The synthetic image is initially warped according to its target on the background image. This target is the synthetic face shape after alignment with landmark coordinates on the background image. The image is then blended into the background image around its border.

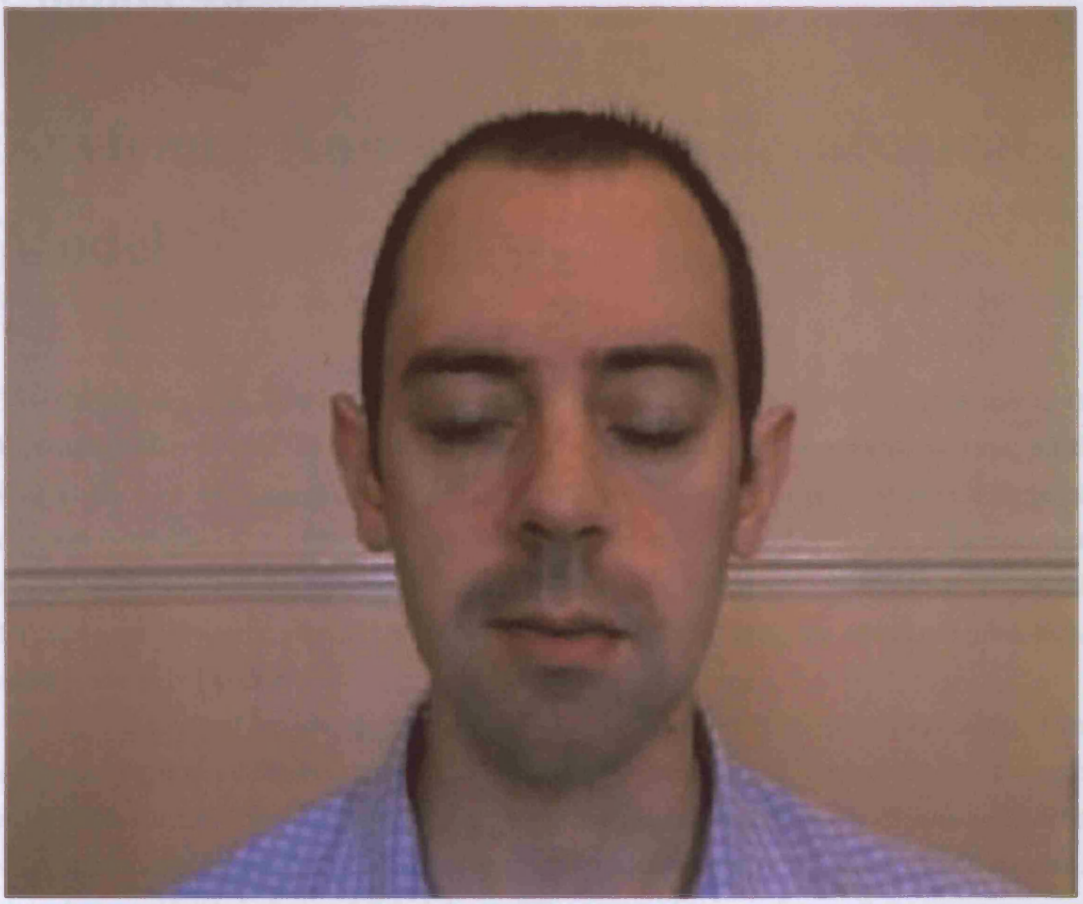

Figure 9.6: An example result of adding a synthetic facial image to a background image. This is the final result of Figure 9.5

Chapter 10

Keyframe Animation of a Hierarchical Model

This Chapter describes how a hierarchical model may be animated using a technique similar to keyframing. The hierarchical model yields intuitive parameters for the animation of facial areas such as the eyes and eyebrows, and variation of these parameters can produce a wide range of different facial poses and expressions. Lip-synching of the mouth using keyframing is currently not supported, as the hierarchy does not automatically produce uncorrelated mouth parameters useful for such a task. However, other key-frame tasks may be performed using the mouth and lower-face, and are considered in Chapter 13.

It should be noted that although the term *key-framing* is used extensively in this chapter, there are several important differences from the standard industrial approach to key-framing 3D geometric models (see Chapter 2 for more information). In the method described here, key-frames are defined using sub-facial parameters. These parameters control major sub-facial image and shape variations, and when combined produce a full-facial pose. To create in-between frames these parameters are interpolated, and the values they take during this period are meaningful with respect to major sub-facial variations. Sub-facial images are created directly from this set of sub-facial parameters, and combined to create full facial outputs.

In the standard industrial approach, parameters control the weight of *morph-targets*. These morph-targets are defined by the animator, and the weight defines the morph-targets variation from some neutral face. Morph-targets are geometry based, and only affect vertices (i.e. shape and not

texture). Interpolation between morph-targets therefore only involves interpolating between facial mesh configurations and not image and shape variation parameters (as in the new method described here). The underlying image is then warped according to the geometry.

Different approaches to animating PCA based statistical models also exist. In [11] Blanz and Vetter describe a method for transferring facial expressions between morphable models of different people. The expression for a person is defined as the difference between an expression and a neutral pose. This expression may be transferred onto the morphable model of a new person by adding the difference vector to the neutral pose of the model. Using this approach, expressions are defined in the first place by fitting the morphable model to a 2D image of a person. Thus, new expressions cannot be created from *scratch*, i.e. an animator cannot tweak the various facial parts of a morphable model as an animator might when creating morph targets. This chapter describes an approach for overcoming this limitation, where expressions may be defined by selecting sub-facial parameter values.

Image based facial models are also used in the Movie and Video Game industries. In these environments markerless performance driven animation is the animation method of choice. Appearance models are fitted to the face of an actor giving a performance and used to analyse and record shape and texture variations (e.g. wrinkles and mouth poses). This information is then transferred onto a geometric character model. This approach has an advantage over shape-based performance driven animation in that subtle skin wrinkles and deformations may be applied to the synthetic character [110].

The Chapter begins by investigating how an analysis of the training corpus gives insights into how animation parameters *naturally* behave. A keyframe approach using the extracted animation parameters is then considered in light of these observations

10.1 Sub-facial Variation Analysis

The modes of variation for an appearance model extract the major variation from a set of images, and are ordered according to the proportion of total variation encoded. When we look at a face, we expect to see certain variations in different facial areas more than others. For example, when we observe the behavior of a persons eye, we would expect that the major variation exhibited to be blinking.

Therefore, by building an appearance model solely using eye images, we would expect the major

modes of variation for the appearance model to encode an opening and closing behavior. In fact, this is exactly what happens. This illustrates one of the advantages of using a hierarchical model. It also illustrates how the model is able to provide a set of useful parameters for animation.

By decomposing the face into a set of sub-facial appearance models, each appearance model will encode the major variation for that region in its modes. The modes of each sub-facial region then give the animation parameters. These parameters can be identified by examining the modes of variation for each sub-facial appearance model. Figures 4.14 to 4.19 show the first four modes of variation for each sub-facial appearance model. Examination of these modes identifies the most useful animation parameters.

As can be seen by examining the modes of the mouth appearance model (4.15), the behavior of the mouth is not generalised well enough to allow any mouth configuration to be easily composed by combining the modes of the mouth. The same is true for the lower-face appearance model, and is especially true for the full face appearance model (Figure 4.13). This is in contrast to the highest 2 modes for the eyes and the eyebrows, which satisfactorily separate out opening and closing (for the eyes), widening and narrowing (for the eyes), and raising and lowering (for the eyebrows) behaviors in individual modes.

Putting this into perspective, it is straightforward to see how a flat appearance model of the face is less suitable for keyframe animation than a hierarchical model. If a user was required to keyframe animate using only an appearance model of the face, he/she would find that the modes of the appearance model would not separate out different facial behaviors, unless that is, the training set of the model contained a large number of example separate behaviors. For example, unless the images used to train the model contained many examples of a person winking - with both eyes - none of the modes of variation would open and close the left and right eyes individually. Add to this the requirement that the animation should contain left and right winking, with raising or lowering of the eyebrows, or opening and closing of the mouth, and a combinatorial explosion begins in terms of the facial poses required in the appearance model training video. It is also not guaranteed that the modes will indeed finally contain these variations. The hierarchical model, by decomposing the face, gives immediate and intuitive access to such a set of parameters - and is therefore far more suitable for keyframe animation.

Figure 10.1 summarises important eye and eyebrow appearance modes of variation for ease of exposition in the remainder of this Chapter. Before considering how to achieve keyframe animation

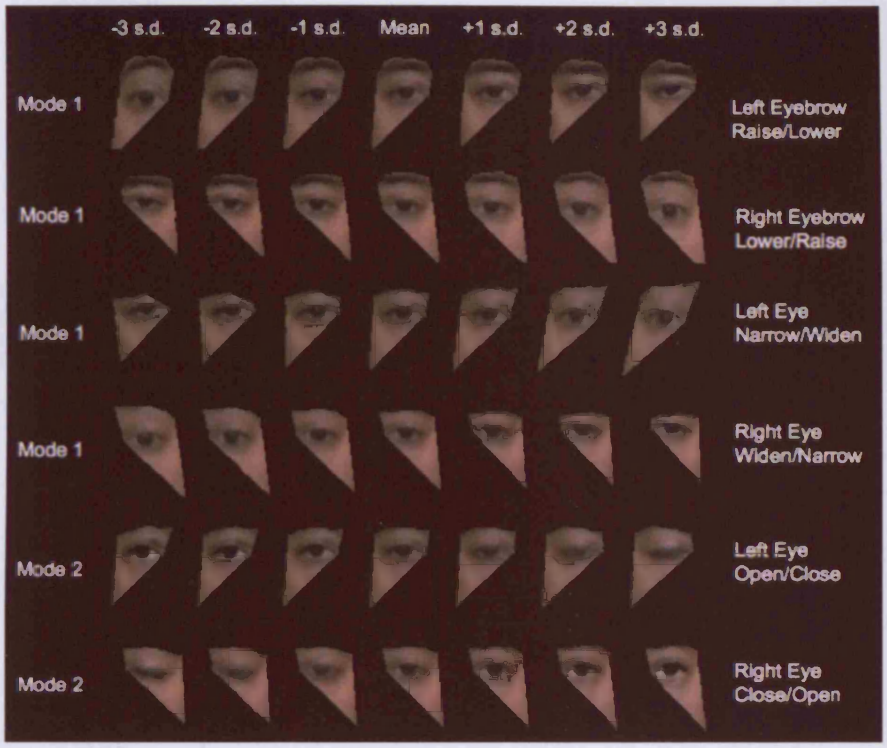

Figure 10.1: An overview of the significant modes of variation resulting from appearance models built for the left and right eyes, and the left and right eyebrows. The Figure shows the ranking of each mode in its respective appearance model and the results of positively or negatively varying its weight.

using this information, it is interesting and insightful to consider the trajectory paths of appearance values for these sub-facial areas as they appear from examples in a training corpus.

Figure 10.2 demonstrates two blinks extracted from the training corpus, along with the behavior of the modes of variation responsible for opening and closing the eyes during these motions. Figure 10.3 shows frames corresponding to the first blink in Figure 10.2. Note that examination of the trajectories proves very insightful into the behavior of an eye during a blink. For example, the eye closing and opening durations differ (i.e. the onset time for a blink (approximately 3 frames) is faster than its offset time (approximately 7 frames)). This information is insightful when attempting to synthetically re-create blinks. Also useful is knowledge of the overall duration of a blink - approximately 10 frames (or $\frac{2}{5}$ of a second) - and the true parameter magnitudes during a blink. For example, a left-eye blink causes the left eye parameter to move between a value of approximately

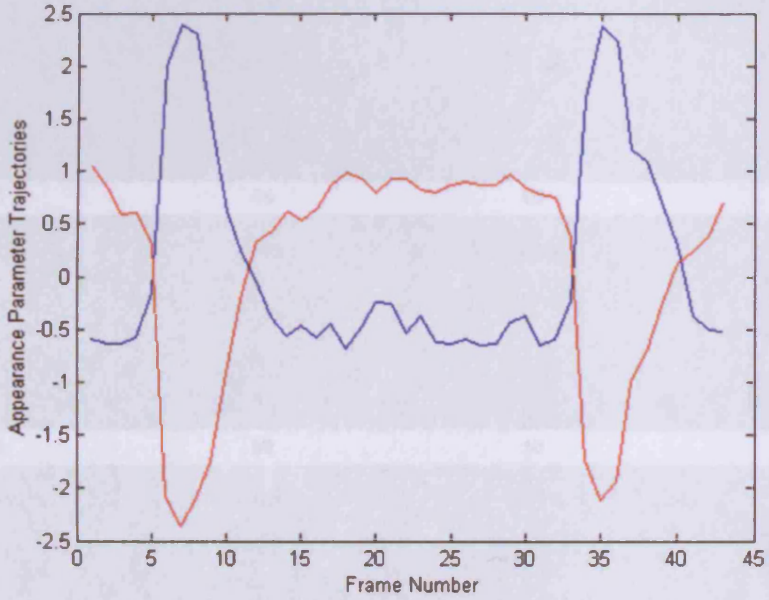

Figure 10.2: An appearance trajectory for a blink, where the blue line represents the behaviour of the left eye, and the red line represents the behaviour of the right eye. Note the important differences in blink offset and onset lengths. The mirroring of the blink trajectories is not significant, and is caused by chance, i.e. when modelling these particular eye distributions, the positive mode of variation for the left eye happened to account for an eye-close, and negative mode of variation for the right eye happened to account for an eye-open. In other words, the sign of the value has no meaning in the context of a mode of variation.

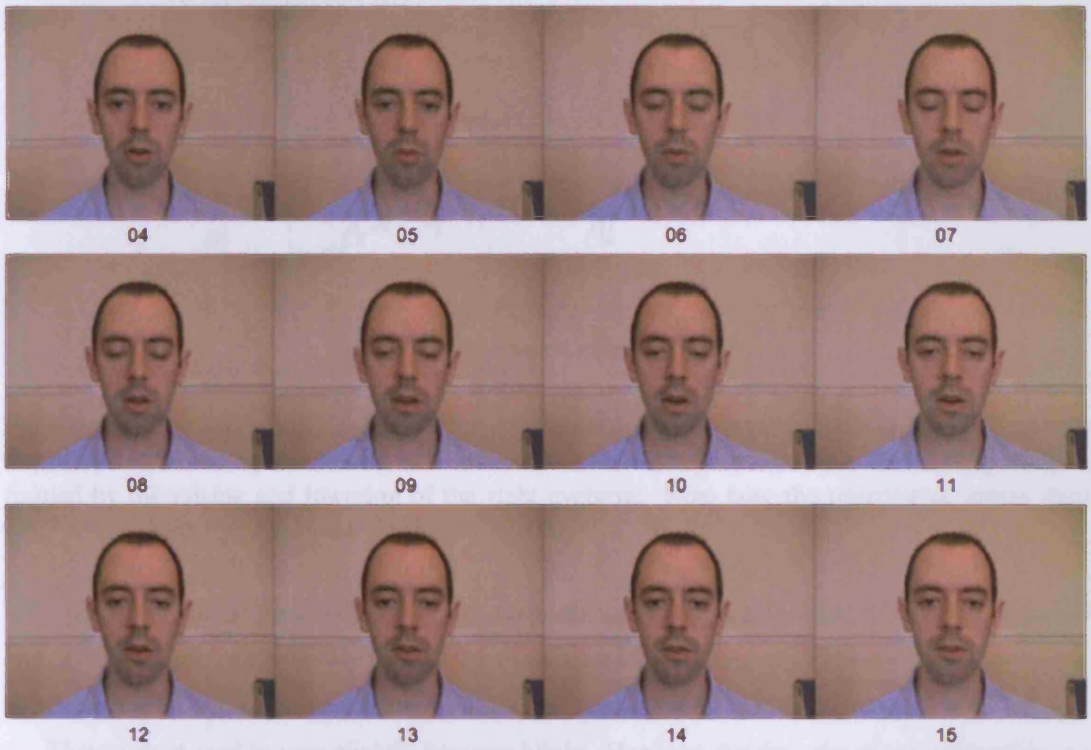

Figure 10.3: Output frames corresponding to the first blink in Figure 10.2. At frame 4 the eyes are still open and blink has not begun. The onset of the blink begins at frame 5 and peaks at frame 7. The longer offset of the blink lasts from frame 8 until approximately frame 14. The short onset, and long offset are also highlighted in the blink trajectories in Figure 10.2

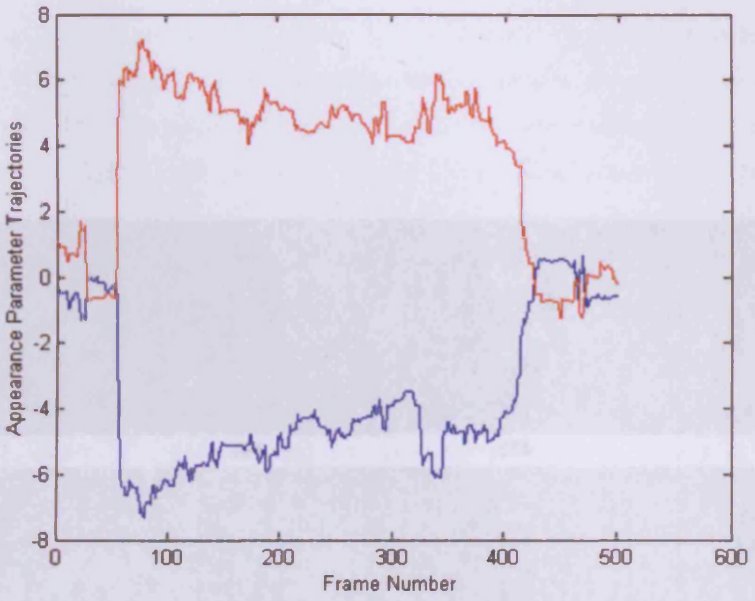

Figure 10.4: An appearance trajectory for a left/right eyebrow raise/lower behaviour. The blue trajectory is formed by the raising and lowering of the left eyebrow, while the red trajectory is formed by the raising and lowering of the right eyebrow. Note how the trajectories mirror each other as each eyebrow raises and lowers at the same time.

-0.5 and 2.5, while the parameter for a right-eye blink moves between a value of approximately 0.5 and -2.5. For these particular eye models, these values translate to ranges of between approximately -0.65 s.d. and 3.25 s.d. for the left eye, and 0.52 s.d. and -2.64 s.d. for the right eye.

These values tend to vary slightly between blinks. However, for the sake of animation they are suitable since they consistently create the appropriate visual effect of an eye opening, or closing. The same applies to onset and offset times. These differ slightly depending on the current blink under examination. However, as with the eye magnitudes, applying constant onset and offset durations of 3 and 7 respectively faithfully re-creates convincing blinks and other eye opening and closing behaviours.

Analysis of real eyebrow trajectories is also insightful. Figure 10.4 shows the trajectories made by left (the blue trajectory) and right (the red trajectory) eyebrow parameters during a raising and subsequent lowering action. The trajectories corresponding to each eyebrow mirror the others behaviour as the action runs through its course. In the corresponding video the raising and lowering is

Figure 10.5: Output images corresponding to the simultaneous eyebrow raising and eye widening trajectories in Figures 10.4 and 10.6. At frame 53 the eyebrows and eyes are at a neutral position. During the onset - from frames 54 to 59 - the eyebrows begin to raise, and the eyes begin to widen. Frames 418, 421, 423, 425 and 427 are selected from the eyebrow and eye offset motion, i.e. as they return from raised/widened to neutral states.

premeditated, and is not correlated with the context of the speech.

The eyebrows move from a neutral position to a fully raised position over an onset period of 6 frames, and each eyebrows appearance parameter reaches a magnitude of approximately ± 7 units. The eyebrows are held in this position for around 6 seconds, although during this time their parameter magnitudes fall slightly between frames 75 and 325 to a low of around ± 4 units. After this point they increase again to a magnitude of ± 6, and subsequently fall off for a second time before the final offset. This is consistent with the behaviour of the participant, whose eyebrows begin to relax after their initial raised position - which is quite extreme - and then begin to lower slightly. At frame 325 the participant forces the eyebrows up once again into an exaggerated position, and the relaxing process repeats itself.

When the eyebrows finally lower back into their neutral positions, a clear offset is observed lasting approximately 10 frames. As with the blinking behaviour, this offset time is longer than the onset time. Figure 10.5 displays several output frames corresponding to the trajectory in Figure 10.4.

As with the analysis of the eye nodes, the magnitudes reached during this animation are also very interesting. A magnitude value of ± 7 creates an exaggerated eyebrow raise, while a less forced raise coincides with a value of ± 4. At rest, the eyebrows have magnitudes of approximately ± 0.5. In terms of standard deviations, these values correspond to -4.66 s.d. for an exaggerated left eyebrow raise, -2.66 for a more relaxed and natural left eyebrow raise and 0.33 s.d for a neutral left eyebrow. For the right eyebrow these values correspond to 4.37 s.d. , 2.5 s.d. and -0.31s.d. respectively.

Also noticeable in Figure 10.5 is the widening of the eyes during the eyebrow raise. An eye widening action is not directly related to an eyebrow raise. In the video, the participant purposefully widens the eyes while raising the eyebrows, producing an effect similar to a shocked expression. An interesting relationship is demonstrated between the eyebrow raise parameters, and the eye widening parameters shown in Figure 10.6. The eye widening parameters follow a trajectory almost identical to the eyebrow raise parameters in terms of onset and offset lengths, and magnitudes (see Figure 10.4). The fact the onset and offset lengths are very similar is perhaps not surprising, since in order to produce the shocked expression, both the eyebrows and the eyes are initially raised and brought to rest at the same time. Reaffirmation of the expression, around frame 350, also coincides with a synchronous re-raising of the eyebrows, and re-widening of the eyes.

The similarities between the magnitudes of the eyebrow raise and eye widen parameters is perhaps more interesting, as it suggests a linear relationship between these two types of behaviour.

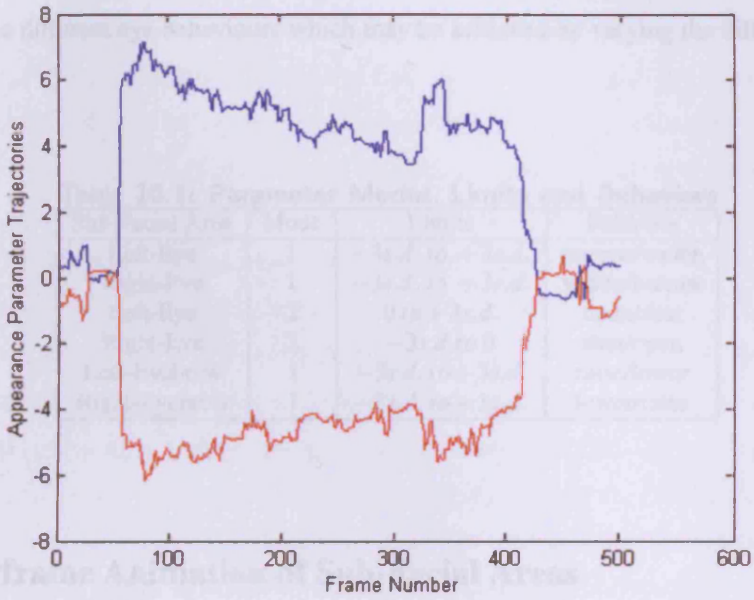

Figure 10.6: An appearance trajectory for a left/right eye widening/narrowing behaviour. The blue trajectory is formed by the widening and narrowing of the left eye, while the red trajectory is formed by the widening and narrowing of the right eye. Note how the trajectories mirror each other as each eye widens and narrows at the same time. Also note the correlation between the left and right eye trajectories shown here, and the left and right eyebrow trajectories shown in Figure 10.4

However, a closer examination of Figure 10.1 reveals that this observation may not be as surprising as it first appears. This is because the images initially created by the eyebrow raise parameters already include a widening of the eyes. These parameters are also the highest modes of variation for the eyebrow nodes, suggesting that this type of variation is the most common eye variation appearing in the training set. Since the *eye* nodes are essentially constructed from sub-images extracted from the *eyebrow* images, then it follows that the highest mode of variation for the eye nodes will be a widening of the eyes - in fact this is exactly what happens. Since there are the same number of eyebrow raise images and eye widening images, and also because these are essentially correlated in a one-to-one manner, it follows that an almost perfect linear relationship between the two modes should exist.

When constructing animations, the magnitudes observed for different eye behaviours do not have to be strictly enforced in order to achieve a satisfactory effect. By examining Figure 10.1 it

can be seen that equally good visual results can be obtained by setting limits at ± 3 s.d.. Table 10.1 summarises the different eye behaviours which may be achieved by varying the different parameter values.

Table 10.1: Parameter Modes, Limits and Behaviors

Sub-Facial Area	Mode	Limits	Behavior
Left-Eye	1	$-3s.d.\ to\ +3s.d.$	narrow/widen
Right-Eye	1	$-3s.d.\ to\ +3s.d.$	widen/narrow
Left-Eye	2	$0\ to + 3s.d.$	open/shut
Right-Eye	2	$-3s.d.to\ 0$	shut/open
Left-Eyebrow	1	$-3s.d.\ to + 3s.d.$	raise/lower
Right-Eyebrow	1	$-3s.d.\ to + 3s.d.$	lower/raise

10.2 Keyframe Animation of Sub-Facial Areas

Keyframe animation for the face is achieved by first generating appearance parameter trajectories for separate sub-facial areas. Sub-facial images generated by these trajectories are then combined to produce final full-facial animations (see Chapter 9).

10.2.1 Eye Animation: Blinks and Winks

New animations for the left and right eyes are generated by setting appearance parameter values for key-moments, and then linearly interpolating in-between. Figure 10.8 shows an example key-frame facial animation demonstrating actions for blinking, winking, and holding both eyes closed for an extended period of time. The animation was created by varying the value of mode 2 for each eye node between limits defined in Table 10.1. Specifically, mode 2 values for the eyes were defined for key-moments, and in between values linearly interpolated. The key-framed trajectory corresponding to Figure 10.8 is given in Figure 10.7. The blue line corresponds to the left eye, while the red line corresponds to the right eye. An increase in the left eye value, or a decrease in the right eye value closes the eye (and vice-versa). By examining Figure 10.7 it is therefore straight forward to guess the content of a resulting video, i.e. it begins with a blink, followed by a wink of the left eye, then a wink of the right eye, and finishes with both eyes held closed for 10 frames. The onset and offset durations used are 3 and 7 frames respectively (see Section 10.1).

Visually, blinks generated in the resulting video are convincing. The onset and offset durations, as well as the magnitudes applied, produce a realistic effect. The only artifact appears when an eye is fully closed - where evidence of the eye-ball texture beneath the eye-lid is faintly visible. This may be caused either by an incorrect peak-magnitude, or the need for the inclusion of another mode of variation in the eye. It is unlikely that an incorrect peak magnitude alone would cause the problem on its own - since a lower magnitude would result in the eye not being properly closed (even though this would fix the problem), and a higher mode would only make the eye behind the eyelid more visible. The solution is therefore to find a second eye mode, which when varied along with the current eye open/close mode, will increase the intensity and population of eyelid (or skin colour) pixels in that region. This could be approached by examining the trajectories taken by smaller (in terms of energy) modes in the eyebrow appearance models, and looking for magnitude changes corresponding to eye open/close trajectory magnitude changes. Another alternative would be to create a second eyebrow PCA model using images calculated as the difference between a ground truth blink and one of the synthetic blinks. These images would contain the required extra pixels. Therefore a model constructed with these images would represent this missing information as its highest mode of variation. Varying this mode in unison with the eye open/close mode would generate the *missing data* image, which could then be added to the *eye close image*.

10.2.2 Combining Eye and Eyebrow Animations: Creating Expressions

Eyebrow animations are generated in a similar fashion to eye animations, i.e. left and right eyebrow parameters are given values for key-moments, and then interpolated in-between. Two main behaviours are considered in this Section: raising and lowering of the eyebrows, and widening and narrowing of the eyes. With just this simple level of control, a wide number of facial configurations can be created.

Figure 10.9 shows a key-framed trajectory sequence which generates combinations of left and right eyebrow raise/lower combinations. The blue line represents the trajectory of left eyebrow mode 1, while the red line represents the trajectory of right eyebrow mode 1. Eyebrows are raised given a positive value for the left eyebrow, or a negative value for the right eyebrow (and vice-versa). Example frames from the video are shown in Figure 10.10. Frame 1 demonstrates a neutral face. The first three actions in the video show the participant lowering both eyebrows at different onset and apex velocities. Frame 35 shows one of the resulting images displaying lowered eyebrows.

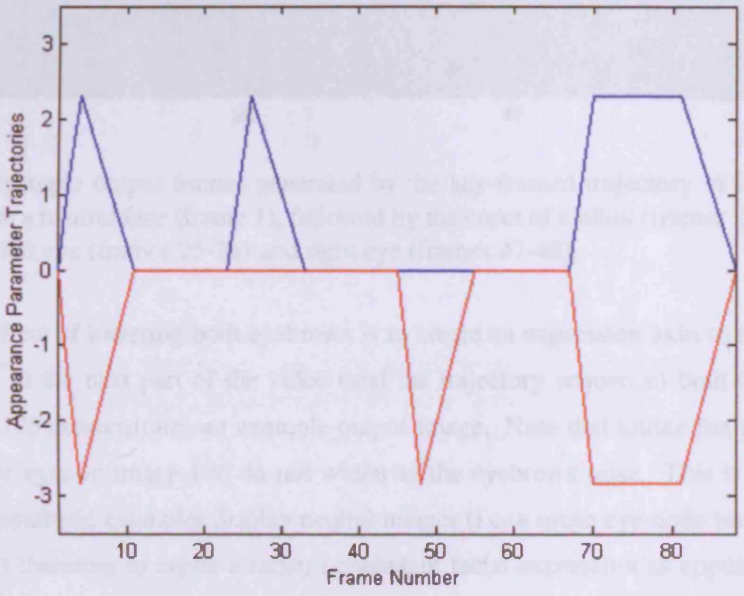

Figure 10.7: A key-framed appearance trajectory for a synthetic animation demonstrating blinking, winking and hold both eyes closed for an extended period of time. The blue line represents the behaviour of the left eye, while the red line represents the behaviour of the right eye. Corresponding output frames are shown in Figure 10.8

Figure 10.8: Example output frames generated by the key-framed trajectory in Figure 10.7. The images illustrate a neutral face (frame 1), followed by the onset of a blink (frames 2-4), and winking actions for the left eye (frames 25-26) and right eye (frames 47-48).

Note that the effect of lowering both eyebrows is to create an expression akin to frustration, anger or annoyance. In the next part of the video (and the trajectory sequence) both the eyebrows are raised. Image 196 demonstrates an example output image. Note that unlike the images shown in Figure 10.5, the eyes in image 196 do not widen as the eyebrows raise. This is because the eye nodes in these synthetic examples display neutral images (i.e. a mean eye-node images). The effect of this action is therefore to create a rather nonchalant facial expression as opposed to a shocked, scared or surprised expression. The remaining trajectory in Figure 10.9 synthesises combinations of raising and lowering the left and right eyebrows independently. Image 715 shows the right eyebrow raised, while image 794 shows a left eyebrow raised. The effect of raising and lowering eyebrows independently is more prominent in images 850 and 910. In image 850 the left eyebrow is lowered and the right eyebrow raised, while in image 910 the left eyebrow is raised and the right eyebrow lowered. Image 850 provides perhaps the best example of a questioning or doubtful facial expression, while in image 910 the expression appears quite arrogant.

Animations displaying these combinations of eyebrow movement are quite smooth, and display minimum artifacts. However, it appears that a right eyebrow raise (images 715 and 850) is not as strong as a left eyebrow raise (image 794 and 910). This appears due to the fact that the neutral position of the right eyebrow is initially lower than that of the left eyebrow (image 001) - and is

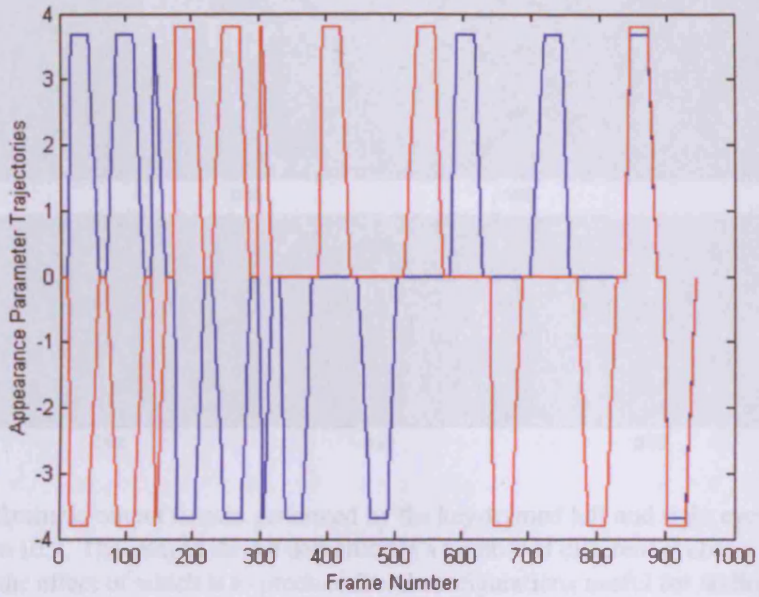

Figure 10.9: Appearance trajectories for left (blue line) and right (red line) eyebrow raise/lower parameters. The resulting animation - example frames from which are shown in Figure 10.10 - displays combinations of raising and lowering the left and right eyebrows. The effect of this is to generate facial configurations useful for adding expression to synthetic performances.

therefore not a flaw in the animation method. In order to provide a better balance, magnitudes applied to the right eyebrow could be increased by the animator to suit their needs.

By adding a widening or narrowing of the eyes to a raised or lowered eyebrow configuration, further expression can be added to animations. Figure 10.11 shows trajectories for eye widen/narrow parameters (top plot) and eyebrow raise/lower parameters (bottom plot). In each plot, the left eye/eyebrow is characterised by the blue line, and the right eye/eyebrow by the red line. Increasing the narrow/widen parameter value for the left eye, or decreasing this parameter for the right eye, produces a widening of the eye (and vice-versa). In the first half of the resulting video - example frames of which are displayed in Figure 10.12 - the eyebrows remain neutral, and the eyes first widen (frame 25) and then narrow (frame 100). The second half of the video shows the effect of widening the eyes while raising the eyebrows (frame 175), and narrowing the eyes while lowering the eyebrows (frame 250).

The result of a combination is to emphasise a frustrated, angry or annoyed expression (frame

Figure 10.10: Example output frames generated by the key-framed left and right eyebrow trajectory shown in Figure 10.9. The images shown demonstrate a number of different eyebrow raise and lower combinations, the effect of which is to produce facial configurations useful for adding expression to performances.

250), or to produce a shocked, scared or worried expression (frame 175). Widening or narrowing the eyes without raising or lowering the eyebrows creates more subtle versions of these expressions.

In Figure 10.1 it can be seen that the eye widen/narrow modes are accompanied by a change in shape, i.e. it appears as if a widening or narrowing of the eyes is automatically accompanied by a respective raising or lowering of the eyebrows. Since overall eye shape (i.e. combined eye and eyebrow node shape) is dictated by the eyebrow node (see Chapter 9), this change in eye-only node shape is removed from the animations if they are not simultaneously accompanied by a change in eyebrow configuration (raised or lowered). Images 25 and 100 demonstrate this, i.e. when the eyes widen or narrow the eyebrows do not raise. The effect of this is that the eyes do not widen or narrow to their full potential unless accompanied by an eyebrow movement - even though the effect synthesised here is still unmistakably a widening or narrowing. In order to uncouple this dependency, the training set would have to contain example eye widen and narrow behaviours independent from eyebrow movement. However, as the training set for hierarchical model 1 in this thesis does not contain such examples, this limitation is currently unavoidable.

One noticable *artifact* in the synthetic eye widen/narrow videos is that the eyes can appear vacant when widened. Eye gaze direction and eye focus is not modeled in this thesis, so this effect

is currently unavoidable. However, given enough controlled examples of eye gaze change in the training video, it is envisaged that a mode responsible for changing these parameters would automatically be encoded in the eye model, and could therefore be exploited to improve the realism of the animations.

10.3 Discussion and Summary

Having the ability to combine wide and narrow eyes - or closed and open eyes - with raised and lowered eyebrows demonstrates the advantage of having two nodes for each eye. The same affect could also be achieved with one eye node. However, in order to do this the training set would have to contain independent examples of many different types of eye behaviour, e.g. eyebrow raised with a blink, eyebrow raised with a wide eye, eyebrow raised with a neutral eye, eyebrow lowered with a blink, eyebrow lowered with a narrow eye and eyebrow lowered with a neutral eye. This is not a difficult task, but it places constraints on the training set which may not be realistic. For example, the source video may not contain all the necessary examples, and it may not be possible to obtain such a video.

The structure of a hierarchical model should therefore be determined by the content of the training video, and some experimentation may be required in order to decide which nodes are important, and which nodes are not. Using the eye and eyebrow node example to illustrate another point, of course one eye node could theoretically model both blinking, eyebrow raising/lowering and eye widening/narrowing. But if further behaviours across the face are considered, such as eye gaze direction, then a combinatorial explosion begins in terms of training video requirements. Now, instead of only having the eyebrow raised with a blink/wide-eye/neutral-eye, the video would have to contain every permutation of raised/look-left/look-right/look-up/look-down/eye-blink/wide-eye/neutral-eye.

This problem can by solved by increasing the number of nodes in the model. However, it should be remembered that by increasing the number of nodes in a model, the complexity of the model is also increased. For example, the trajectory and mode analysis required for each node in order to identify valid animation limits, onset lengths, apex lengths and offset lengths, becomes more time consuming. Potential interactions between nodes also become more complex. One simple example already seen is the interaction between the eye widening parameter and the eyebrow raise parameter.

The best hope is that an automatic method could be developed for determining the optimal

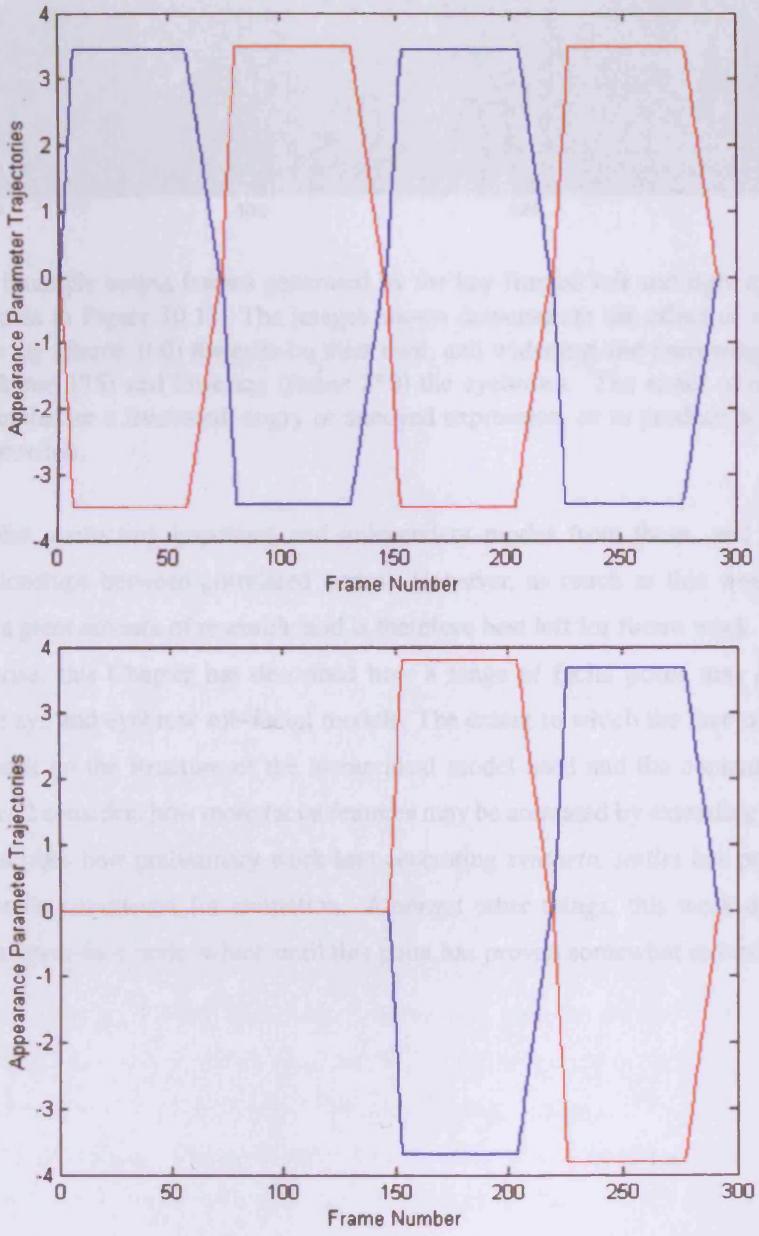

Figure 10.11: Keyframe appearance trajectories for eye widen/narrow parameters (top plot) and eyebrow raise/lower parameters (bottom plot). Example frames from the resulting synthetic animation can be found in Figure 10.12. An increase in the left eye parameter, or a decrease in the right eye parameter causes the eyes to widen (and vice-versa). Combining an eye widening or narrowing action with an eyebrow raise or lower emphasises an expression. The trajectories produce a video where the eyes first widen and then narrow with neutral eyebrows, and widen and narrow with raised and lowered eyebrows.

Figure 10.12: Example output frames generated by the key-framed left and right eye and eyebrow trajectories shown in Figure 10.11. The images shown demonstrate the effect of widening (frame 25) and narrowing (frame 100) the eyes on their own, and widening and narrowing the eyes along with raising (frame 175) and lowering (frame 250) the eyebrows. The result of combining these actions is to emphasise a frustrated, angry or annoyed expression, or to produce a shocked, scared or worried expression.

number of nodes, extracting important and independent modes from these, and identifying and encoding relationships between correlated nodes. However, as much as this would be useful, it would require a great amount of research, and is therefore best left for future work.

To summarise, this Chapter has described how a range of facial poses may be animated by keyframing the eye and eyebrow sub-facial models. The extent to which the face may be keyframe animated depends on the structure of the hierarchical model used and the content of the training video. Chapter 12 considers how more facial features may be animated by extending the hierarchical model, and describes how preliminary work into generating *synthetic smiles* has proven successful in extracting smile parameters for animation. Amongst other things, this work demonstrates the usefulness of a lower-face node, which until this point has proven somewhat redundant.

Chapter 11

An Evaluation of HMCMs and SAMs

This Chapter provides a quantitative evaluation of SAMs and HMCMs for synthesising visual-speech, and eyebrow motions from new speech signals. A thorough analysis of SAMs is omitted from this Thesis. Instead, only an overview of its performance is given. A more detailed analysis of SAMs may be found in [42, 43, 44].

The Chapter begins by first performing a thorough analysis of HMCMs for synthesising mouth and eyebrow animation from speech. Visual-speech synthesis is considered using hierarchical model 2, while eyebrow animation is considered using hierarchical model 1. Both models are described in detail in Chapter 4. Hierarchical model 1 contains a hierarchy with nodes for the face, lower-face, left and right eyebrows, left and right eyes, and the mouth. To recap, this model is also used in the evaluation of key-frame animation. Hierarchical model 2 only contains nodes for the face and the mouth, and is designed solely for the evaluation of visual speech synthesis.

As well as evaluating a HMCMs ability to synthesise visual-speech, the synthesis of natural pauses, hesitations, non-verbal articulations and independent speech is also considered. It has already been demonstrated how the hierarchical model has advantages over a flat model for applications such as key-framing. The advantage of performing visual speech synthesis using a hierarchical model over a flat model of the face is also described in this Chapter. In the evaluation, it is demonstrated how synthesis accuracy (as defined in the next section) is improved by concentrating a synthesis model on a smaller specific facial area as opposed to a large facial area (e.g. the entire face).

11.1 HMCM Test Strategy

The test strategy employed in evaluation of the HMCM for visual-speech synthesis is based on a *leave one observation out* approach. The training corpus of model 2 is segmented into a set of 13 observations. Each observation contains a speech signal, a set of ground truth luminance intensity and shape coordinate vectors, and a set of ground truth PCA parameter trajectories for luminance appearance, and luminance and shape. Observation sizes range from 408 to 808 frames in length.

The synthesis model is tested by:

1. Leaving one observation out of the training set.
2. Building a HMCM using the remaining observations.
3. Synthesising a new set of output images, and their corresponding appearance, luminance and shape PCA parameters using the speech associated with the observation left out.
4. Comparing the synthesised values with their ground truths.

Using this approach, strengths and weaknesses can be identified in each synthesised observation, and an overall picture of performance across all observations can be painted.

In order to get an initial impression of synthetic performance - for each observation and across the training set - the Root-Mean-Square (RMS) error between synthetic and ground truth data can be viewed. The RMS error gives the normalised difference between the sum of the elements in a synthetic input vector, and a ground truth vector:

$$e = \sqrt{\frac{\sum_{i=1}^{n}(\mathbf{x}_i^S - \mathbf{x}_i^G)^2}{n}} \tag{11.1}$$

where $\mathbf{x}_i^S$ is a synthetic vector and $\mathbf{x}_i^G$ is a ground truth vector.

It should be noted that an RMS error for a single observed synthetic and ground truth mouth (shape or texture) is an average pixel error for that frame. The error measure is therefore not perfect since it is unclear why some errors are large and some small in a particular case. Errors may be due solely to illumination differences between vectors, or differences in texture around one particular part of the mouth. Some differences are obvious when visually examining the images produced by vectors. However, this difficult to do for thousands of images. The fact remains that a perfect

quantitative measure for animation quality does not currently exist, and in the absence of a better alternative the RMS measure is used.

Using respective ground truths, the RMS error is measured across the entire set of observations for synthesised luminance images, shape coordinates, luminance PCA parameters, shape PCA parameters, and appearance parameters. Using this information and a general picture of performance formed (taking into account the inherent drawbacks of this approach previously addressed).

A more detailed picture of performance is obtained by examining smaller segments of the synthesised animations. This gives an insight into what visemes are articulated well, and if coarticulation between them is satisfactory or not. This more detailed analysis is done on a sentence level, and comparisons are made versus the ground truth signal for a particular sentence using:

- The RMS luminance pixel intensity error of the synthetic mouth animation .
- The RMS shape coordinate pixel error of the synthetic mouth animation.
- The trajectory produced by the highest PCA mode of luminance variation.
- The trajectory produced by the highest PCA mode of shape variation.
- The trajectory produced by the highest PCA mode of luminance-appearance variation.

The RMS error produced by synthesised luminance mouth pixels gives a good initial indication of synthesis quality in a particular sentence. An overall low luminance RMS error typically points to a good overall animation for a sentence. An increase in the RMS error at a particular part of the sentence would indicate a fault in the synthetic animation. This is also typically accompanied by a discrepancy between the ground truth appearance, luminance and shape PCA signals and their synthetic counterparts. For example, given an increase in RMS error at a particular point, the ground truth and synthetic appearance trajectories - along with corresponding luminance and shape PCA trajectories - often diverge. However, as will be seen from the examples examined, this is not always an indication of a poor synthetic animation, and highlights a fault in analytical analysis of mouth animation. The data could perhaps indicate a bad animation, but a closer inspection might indicate that in that particular part of the animation there is silence, and while the ground truth mouth might be slightly open during this event, the synthetic mouth might move to fully closed. From a perceptual point of view, it could therefore be stated that the synthetic animation is actually

valid, and not at fault as the RMS measure might suggest. In Chapter 12, a perceptual test for visual speech synthesis is proposed and applied in evaluation of synthesised animations.

RMS errors, and comparisons between ground truth and synthetic animation trajectories (on an overall level and a per-sentence level), should therefore be approached with a degree of caution - and should always be examined alongside the animation itself. However, Discrepancies (such as the one described) do not always mean the synthetic animation will still be valid, and will often be found, for example, when the synthetic animation has unsuccessfully articulated a certain sound. In these cases the synthetic animation typically *catches-up* a few frames later - as is demonstrated in the given examples. However, this is not always the case. So again, in order to effectively evaluate synthesis from an analytical point of view, a certain degree of perceptual observation is also essential.

11.2 Visual-Speech Synthesis using a HMCM

This Section analytically evaluates the HMCM for visual-speech synthesis given new speech observation inputs. For this evaluation, hierarchical model 2 was used (see Section 4.6.2). A HMCM is constructed for the mouth and used to produce new animations from new speech observations by first synthesising the mouth, and then attaching the mouth to the face node. The completed full face image is then added to a background image from the original training set (see Chapter 9).

11.2.1 Mouth HMCM: HMMs

In Section 7.4 it was described how HMCM synthesis quality is related to the number of states in the HMM and the amount of appearance PCA energy retained. Due to computational constraints, it is difficult to achieve an optimum value for both, so a compromise is found via experimentation.

Since evaluation is performed in a *leave one observation out* manner, HMMs - and consequently HMCMs - are constructed using the remaining set of observations. A new HMCM is therefore constructed to test each observation, and a new HMM constructed for each HMCM.

In order to find sensible values of k (i.e. the number of states in the HMM), and the proportion of total appearance model energy retained for constructing the HMM, the experimental approach described in Section 5.2.2 was carried out. To recap, the negative log-likelihood is calculated for a HMM as a function of both k and the proportion of appearance model energy. That is, HMMs

are constructed for multiple values of k and appearance energy, and the negative log-likelihood recorded. Since the initial means of each HMM are chosen at random (using k-means), 20 HMMs are constructed for each value of k, and the lowest negative log-likelihood recorded along with the mean starting positions which produced the value.

Results from this trial - carried on on the entire mouth training set of hierarchical model 2 - were almost a mirror of the GMM results obtained out in Section 5.2.2, i.e. values of $k=120$ and 65 percent appearance energy are deemed appropriate choices. As observations are left out of the training corpus, and new HMM trails performed on the remaining data, these values for k and appearance energy still appear satisfactory - and are therefore used for each *leave one observation out* test.

11.2.2 Mouth HMCM: Overall RMS Errors

Using the *leave one observation out* test strategy, mouth animations were synthesised using a HMCM for each observation and RMS errors recorded against ground truths for pixel intensities, shape coordinates, and parameter trajectories. Figure 11.1 shows the RMS error recorded for each synthetic mouth luminance vector $\mathbf{l}$ versus its ground truth from the training corpus. Each vector contains luminance value pixels with ranges of between 0 and 255. The average RMS error is 32.5, and the standard deviation is 10.2. Although this can only give a general indication of performance, the RMS error recorded is relatively low. This would suggest that overall synthesis performance is satisfactory. However, as mentioned in Section 11.1, a firm conclusion cannot be made based solely on these results, and a more detailed examination is required in order to paint an overall picture of performance.

Figure 11.2 shows the RMS errors recorded for each synthetic mouth shape vector $\mathbf{x}$ versus its ground truth from the training corpus. Each vector contains pixel coordinate values for the mouth. The average RMS error is 1.9 pixels, while the standard deviation is 0.7 pixels. Again, it is difficult to judge overall performance based only on these values. However the low RMS results suggest that, in the worst case, shape errors should not have too much of an adverse effect on the quality of output animations.

There appears to be no correlation between the luminance RMS errors and shape RMS errors. This is perhaps not surprising since two similar mouth shape vectors can be associated with two entirely different luminance vectors (i.e. mouths with an entirely different *appearance*). A high

RMS error for luminance intensity may therefore coincide with a low RMS error for shape.

Figures 11.3, 11.4 and 11.5 show RMS errors for synthesised PCA model weights representing luminance $\mathbf{b}_l$, shape $\mathbf{b}_x$ and appearance $\mathbf{c}$, versus their respective ground truths. RMS errors for PCA weights are more difficult to interpret than RMS errors for pixel intensities and shape coordinates since it is unclear what a *good* error is and what a *bad* error is. However, interesting observations may still be drawn. For example, the shape parameter RMS error in Figure 11.4 is correlated with the shape coordinate RMS error in Figure 11.2. Surprisingly, the same correlation does not seem to exist between the luminance parameter RMS error and the luminance intensity RMS error.

Since RMS errors are influenced by values with potentially large limits, and also since the highest modes of variation in a PCA model tend to have the largest variance of weight value, this would suggest that the lack of a correlation is caused by errors in PCA model modes accounting for a lesser proportion of total model energy, but which are nonetheless significant in terms of the texture they produce. This would make sense, and would also explain the correlation between the shape parameter errors and the shape coordinate errors. Since the shape model inherently contains less variance to begin with, most of its variation is included in only a handful of modes (i.e. less modes than in for example the luminance PCA model). An error in a shape parameter with a potentially large variance (i.e. a significant mode) would therefore have a more drastic effect on the error in the respective reconstructed shape output. Since a luminance model contains *more* significant modes (in terms of the texture they produce), but importantly also more significant modes with less variance, a high parameter error (which in terms of an RMS error is made up of the sum of its mode weights), could be associated with a low luminance pixel intensity error, and vice-versa.

A better method for evaluating parameter errors is to view them alongside ground truth parameters. Discrepancies between synthetic and ground truth parameters, and their correlation with pixel intensity and shape coordinate errors, can then be interpreted more clearly. In the next Section evaluation is performed on a sentence level, which allows for a closer inspection of errors, and encourages a more thorough method of evaluation.

11.2.3 Mouth HMCM: Example Animations

Figures 11.6, 11.7 and 11.8 show RMS error and trajectory comparisons for the sentence "When she got inside she saw a large wooden table and three wooden chairs".

Overall, the animation is of a perceptually satisfactory standard. However, as the RMS error

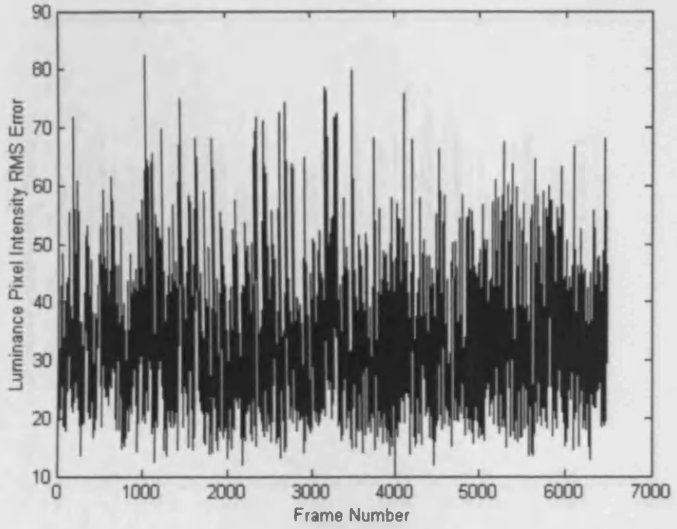

Figure 11.1: RMS errors recorded for HMCM generated mouth luminance vectors **l** versus ground truth vectors extracted from the training corpus. Luminance vectors contain luminance intensity values, and are ranged between values of 0 and 255.

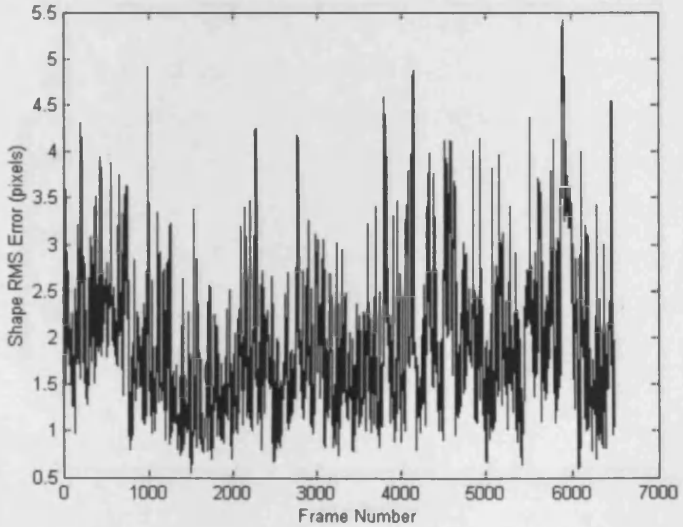

Figure 11.2: RMS errors recorded for HMCM generated mouth shape vectors **x** versus ground truth vectors extracted from the training corpus. Shape vectors contain pixel coordinates.

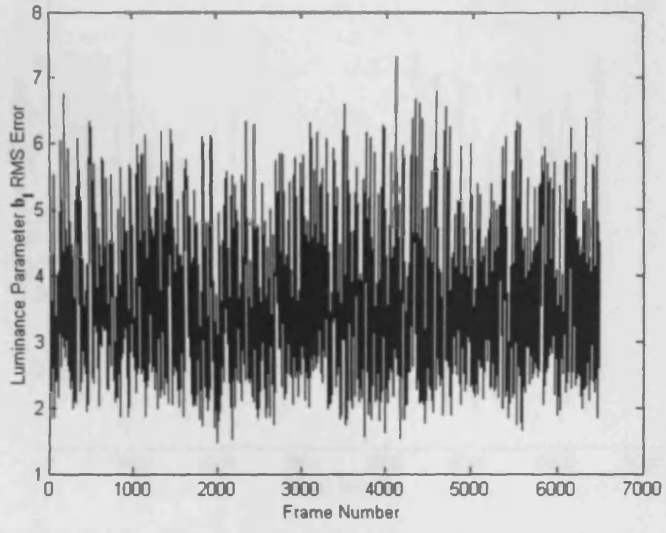

Figure 11.3: RMS errors recorded for HMCM generated mouth luminance PCA model parameters $\mathbf{b}_l$ versus ground truth parameters extracted from the training corpus.

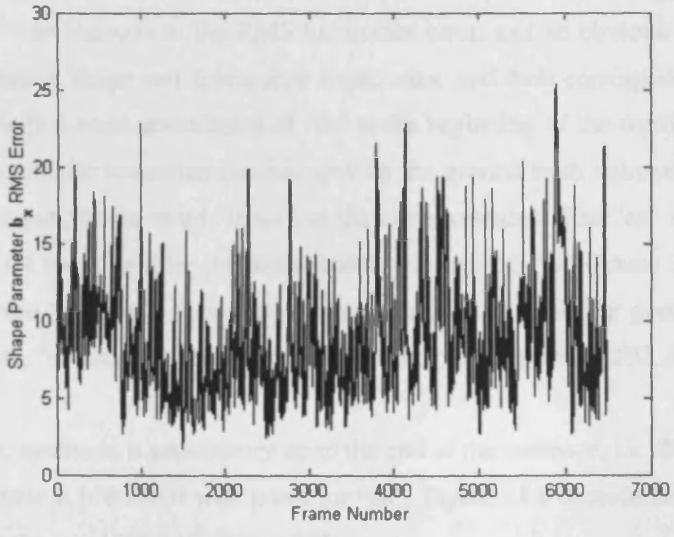

Figure 11.4: RMS errors recorded for HMCM generated mouth shape PCA model parameters $\mathbf{b}_x$ versus ground truth parameters extracted from the training corpus.

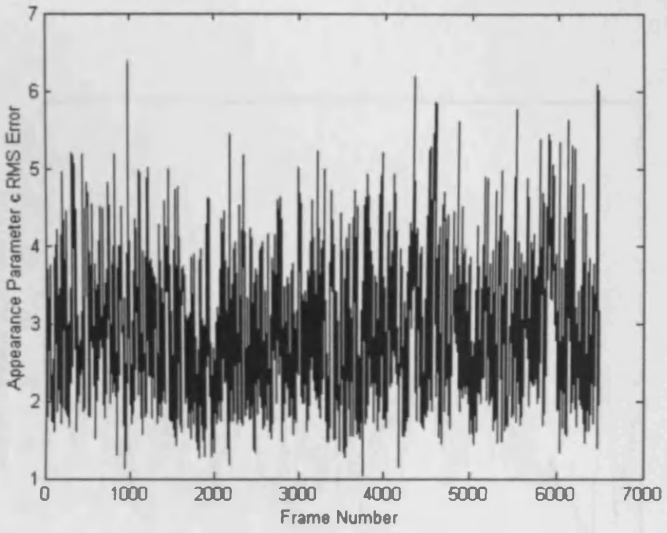

Figure 11.5: RMS errors recorded for HMCM generated mouth appearance PCA model parameters **c** versus ground truth parameters extracted from the training corpus.

plots and trajectories suggest, parts of the animation are not perfect and warrant further investigation. Figure 11.9 highlights selected frames from the animation. Between approximately frames 140 and 170 there is an increase in the RMS luminance error, and an obvious difference between the synthetic luminance, shape and appearance trajectories, and their corresponding ground truths. This is associated with a weak articulation of "th" at the beginning of the word "three". However, by frame 177 the synthetic animation catches up with the ground truth animation, and both begin synthesis of "ee" (again, in the word 'three') at the same moment. The "ee" in the synthetic animation continues for too long after the initial onset, missing the "w" (frame 182) and "o" at the beginning of the word "wooden". However, synthesis catches up again for good articulation at the end of the word ("d", "e" and "n"). Figure 11.9 shows the output at frame 192, during the synthesis of "n".

After this point, synthesis is satisfactory up to the end of the sentence, i.e. during the final word "chairs", which begins at frame 201 with poses for "ch". Figure 11.9 demonstrates synthesis of 'ai' (pronounced *eh*) in the word "chairs" (frame 207).

Figures 11.10, 11.11 and 11.12 show RMS error and trajectory comparisons for the sentence "As she went closer she saw the three bowls had porridge in it".

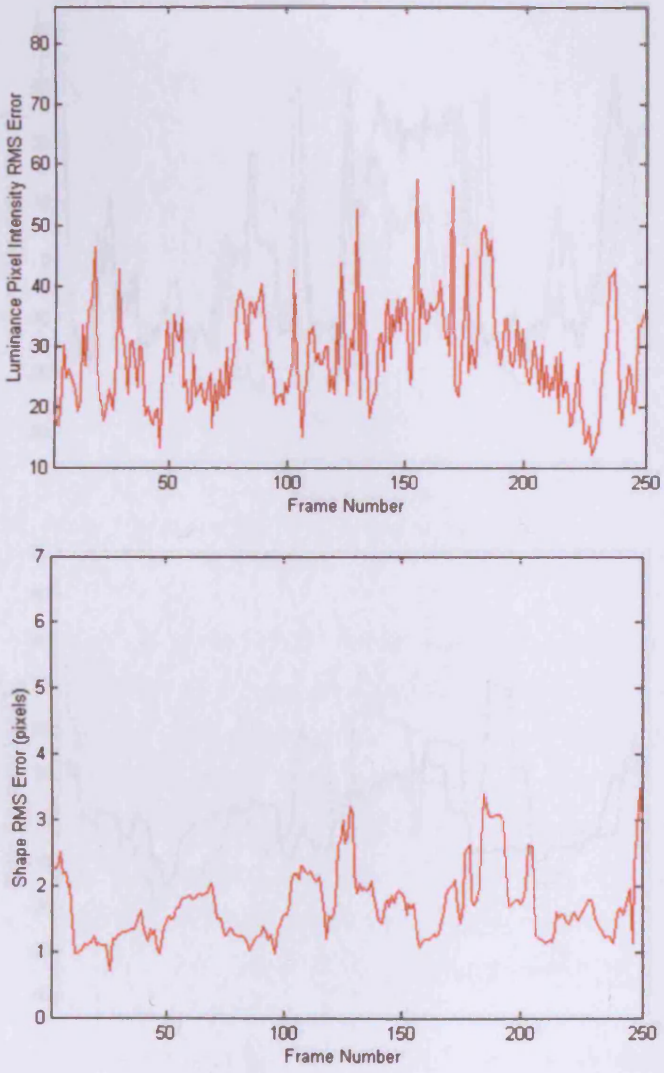

Figure 11.6: Model 2 mouth HMCM luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "When she got inside she saw a large wooden table and three wooden chairs.".

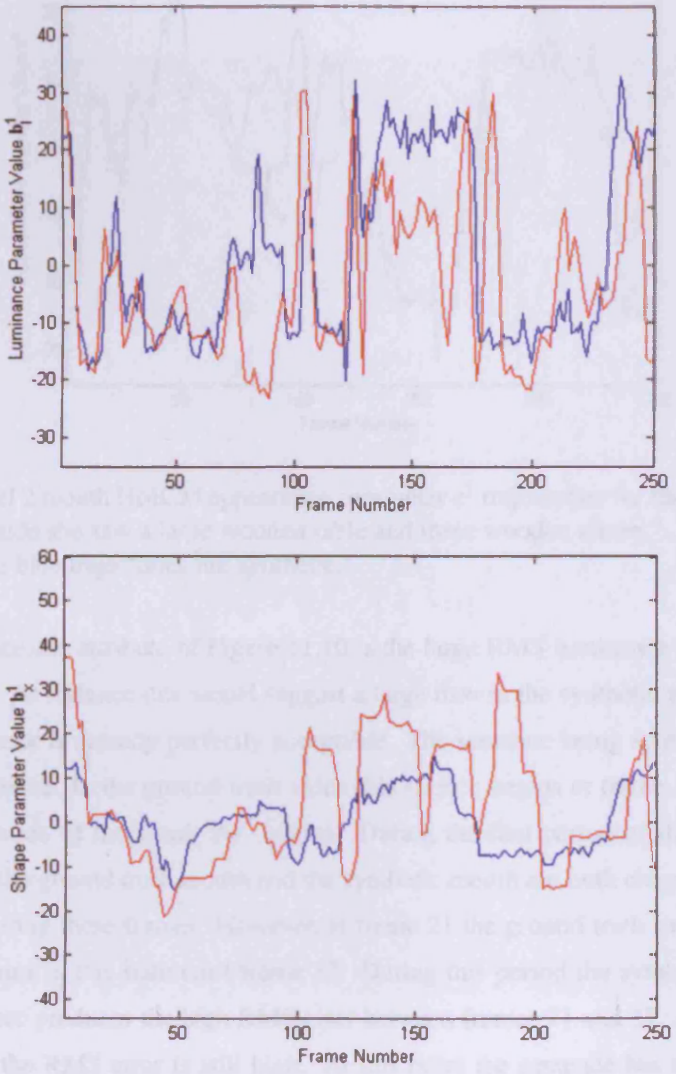

Figure 11.7: Model 2 mouth HMCM luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "When she got inside she saw a large wooden table and three wooden chairs.". Red trajectories are ground truth while blue trajectories are synthetic.

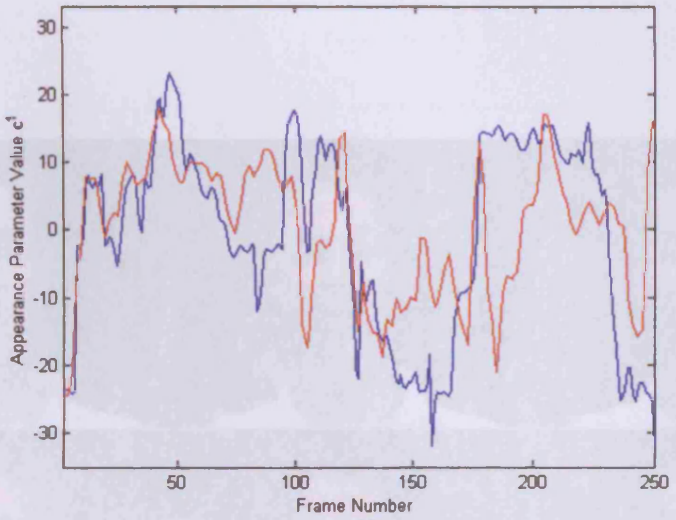

Figure 11.8: Model 2 mouth HMCM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "When she got inside she saw a large wooden table and three wooden chairs.". Red trajectories are ground truth while blue trajectories are synthetic.

The most noticeable attribute of Figure 11.10 is the large RMS luminance pixel error between frames 21 and 45. At a glance this would suggest a large flaw in the synthetic animation - however the cause of the error is actually perfectly acceptable. The sentence being synthesised begins with a a segment of silence. In the ground truth video this silence begins at frame 1 and ends at frame 37, where articulation of the word "As" begins. During the first period of this silence (between frames 1 and 20) the ground truth mouth and the synthetic mouth are both closed - hence the initial low RMS error during these frames. However, at frame 21 the ground truth mouth begins to open slightly, and remains in this state until frame 37. During this period the synthetic mouth remains closed and therefore produces the high RMS error between frames 21 and 37. After frame 37, and before frame 45, the RMS error is still high. At this point the sentence has already begun. The cause of the high RMS error can be found by examining the synthetic animation, where it is seen that articulation of the word "As" (the first word of the sentence) actually begins late, and does not begin until frame 45. At this point the RMS error decreases again. This delay in articulation onset is hardly noticeable in the synthetic video from a perceptual point of view. Figures 11.11 and 11.12 highlight this change in the ground truth mouth during the silent period. The red trajectory (ground

Figure 11.9: Selected frames from the synthesised sentence "When she got inside she saw a large wooden table and three wooden chairs.". Synthesis is achieved using a mouth HMCM constructed with hierarchical model 2.

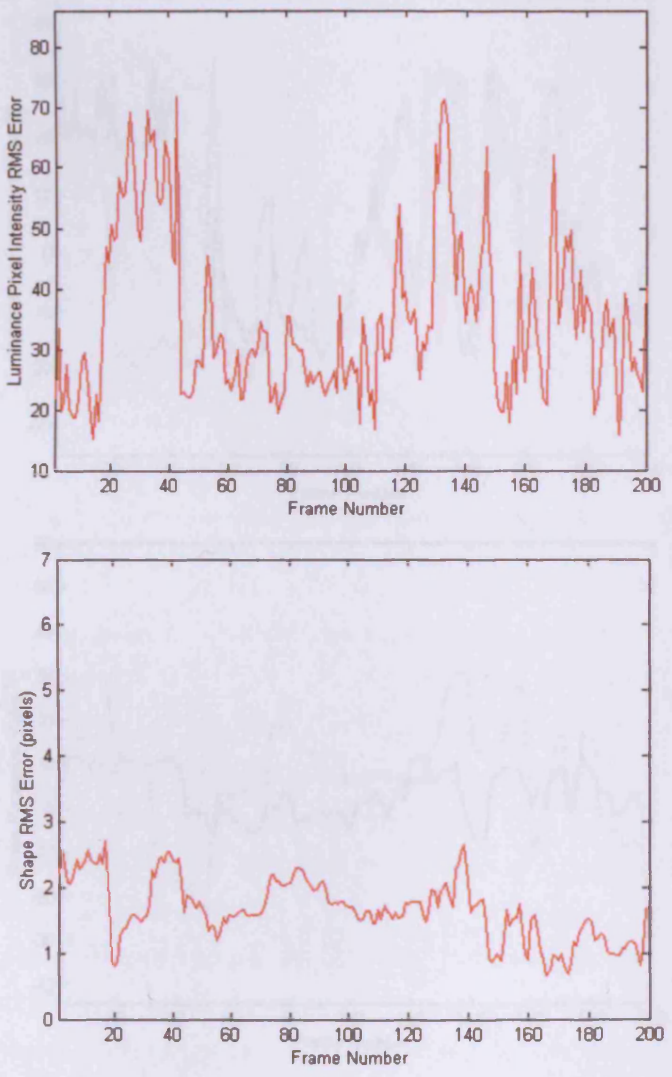

Figure 11.10: Model 2 mouth HMCM luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "As she went closer she saw the three bowls had porridge in it.".

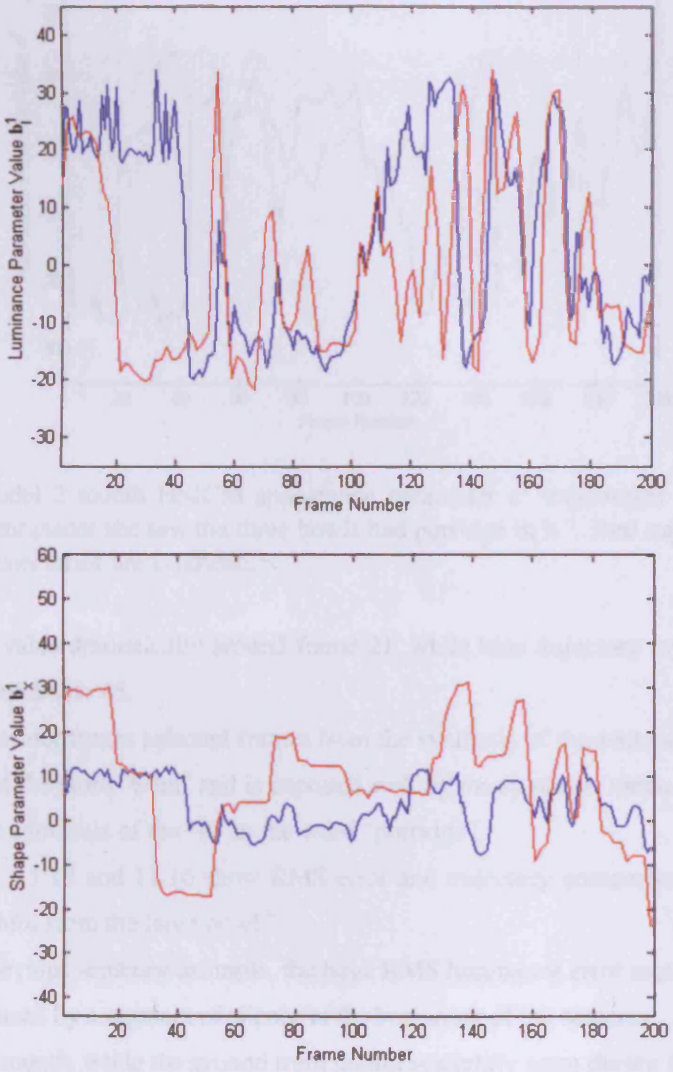

Figure 11.11: Model 2 mouth HMCM luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "As she went closer she saw the three bowls had porridge in it.". Red trajectories are ground truth while blue trajectories are synthetic.

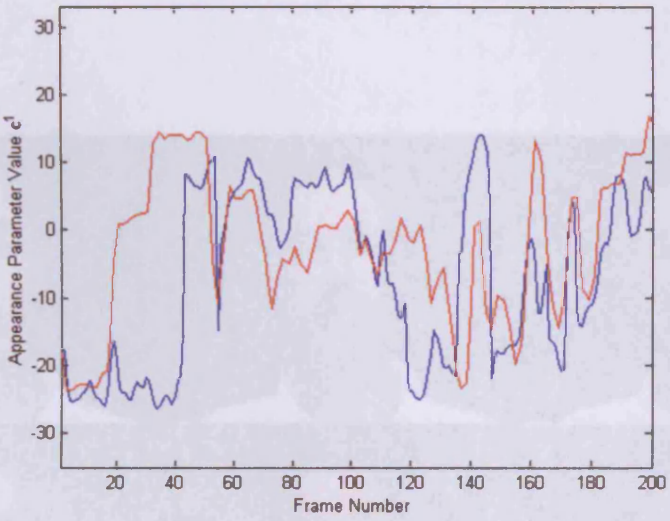

Figure 11.12: Model 2 mouth HMCM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "As she went closer she saw the three bowls had porridge in it.". Red trajectories are ground truth while blue trajectories are synthetic.

truth) changes its value dramatically around frame 21, while blue trajectory (synthetic) remains at its initial value until frame 45.

Figure 11.13 demonstrates selected frames from the synthesis of the sentence. Frame 56 occurs at the beginning of the word 'went' and is captured well by the synthetic video, while frame 177 is extracted from the synthesis of the "o" in the word "porridge".

Figures 11.14, 11.15 and 11.16 show RMS error and trajectory comparisons for the sentence "She took a mouthful from the large bowl.".

As with the previous sentence example, the large RMS luminance error seen at the beginning of Figure 11.14 is caused by a segment of silence at the beginning of the sentence. Again, the synthetic mouth is a closed mouth, while the ground truth mouth is slightly open during this period - causing the large RMS error. This period of silence ends at frame 3. The RMS luminance error for the remainder of the sentence is relatively low. However there are several points of interest.

As the low RMS errors indicate, articulation in the synthetic sentence is relatively strong throughout. One observation - which although interesting from an analytical perspective has no detrimental effect on the animations - is that synthetic articulations are often late in comparison to the ground

Figure 11.13: Selected frames from the synthesised sentence "As she went closer she saw the three bowls had porridge in it.". Synthesis is achieved using a mouth HMCM constructed with hierarchical model 2.

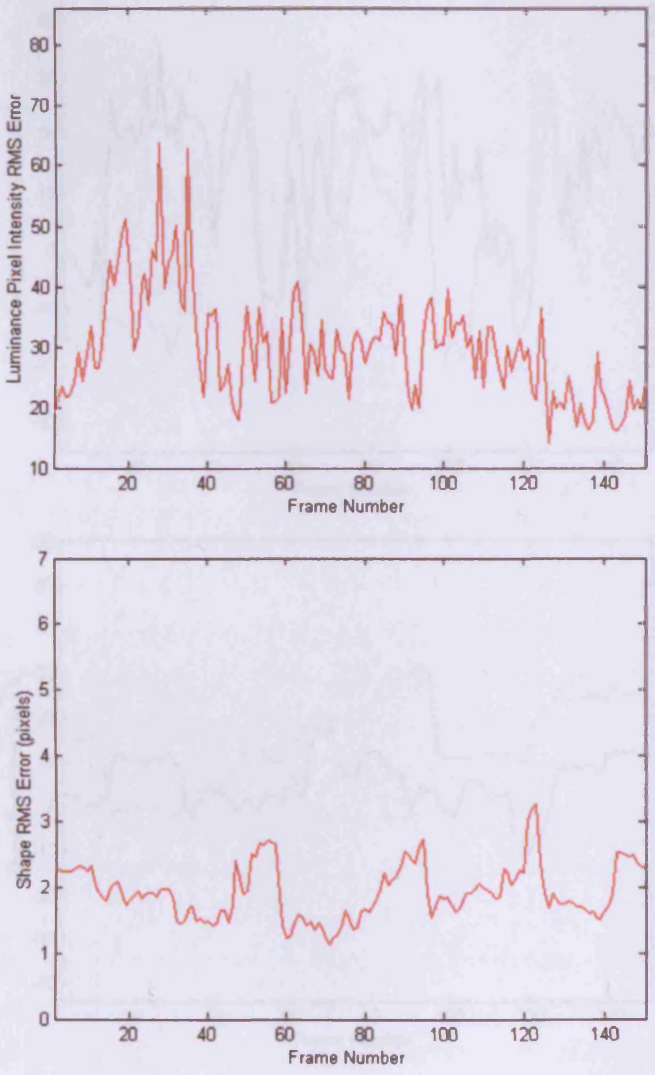

Figure 11.14: Model 2 mouth HMCM luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "She took a mouthful from the large bowl.".

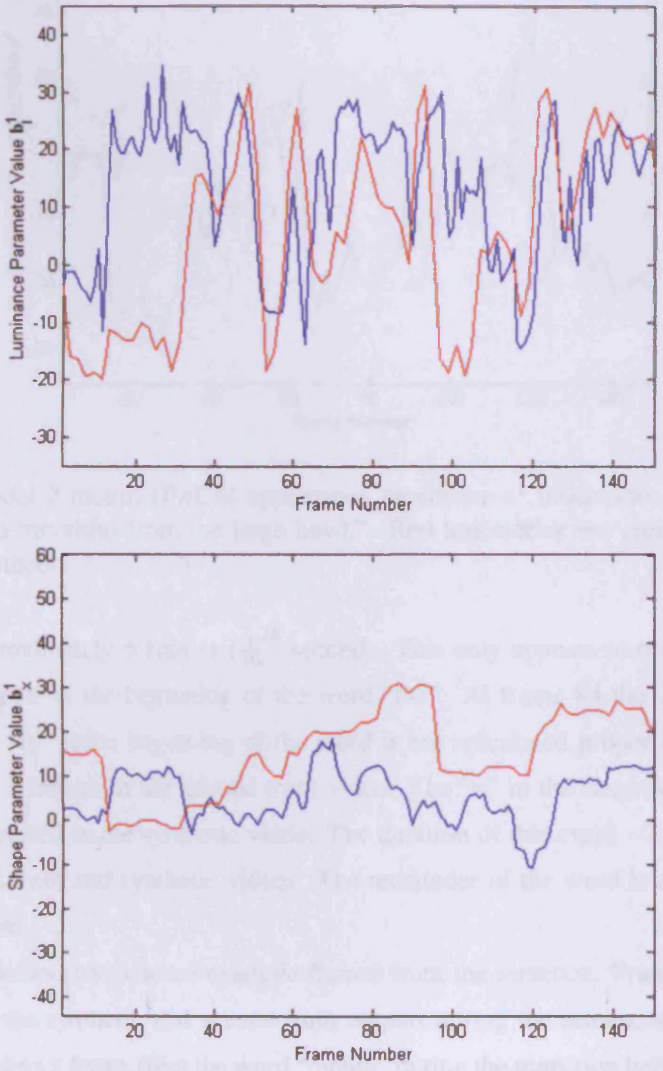

Figure 11.15: Model 2 mouth HMCM luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "She took a mouthful from the large bowl.". Red trajectories are ground truth while blue trajectories are synthetic.

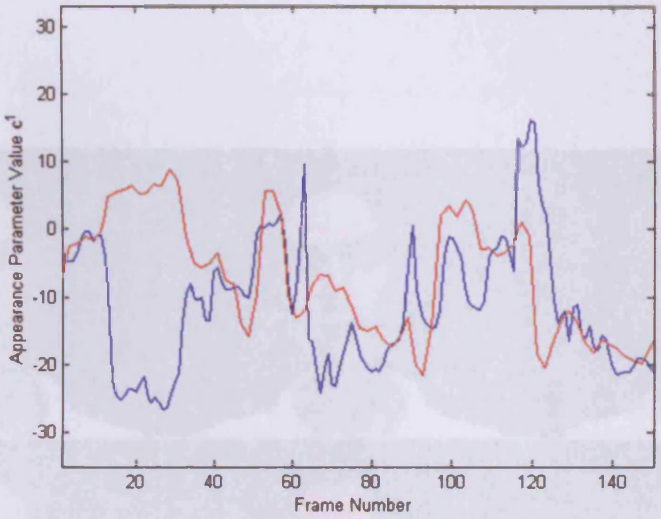

Figure 11.16: Model 2 mouth HMCM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "She took a mouthful from the large bowl.". Red trajectories are ground truth while blue trajectories are synthetic.

truth by up to approximately 5 frames ($\frac{1}{10}^{th}$ second). This only appears to occur at the beginning of words, for example at the beginning of the word "the". At frame 84 the word "from" begins. Unfortunately, the "fr" at the beginning of the word is not articulated properly. This event occurs for approximately 5 frames in the ground truth video. The "o" in the same word occurs at frame 89, and is well captured in the synthetic video. The duration of this event - 2 frames - is identical in both the ground truth and synthetic videos. The remainder of the word is also synthesised to a satisfactory standard.

Figure 11.17 demonstrates some example frames from the sentence. Frame 20 shows the discrepancy between the synthetic and ground truth outputs during the sentences initial period of silence. Frame 56 shows a frame from the word "mouth" during the transition between the articulation of "ou" and "th". Frame 89 shows the articulation of "o" in the word "from".

11.2.4 Mouth HMCM: Pause and Hesitation Synthesis

Figures 11.18, 11.19 and 11.20 show RMS error and trajectory comparisons for the sentence "And tried a mouthful of the next bowl of porridge. Too sour she said.".

Figure 11.17: Model 2 mouth HMCM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "She took a mouthful from the large bowl.".

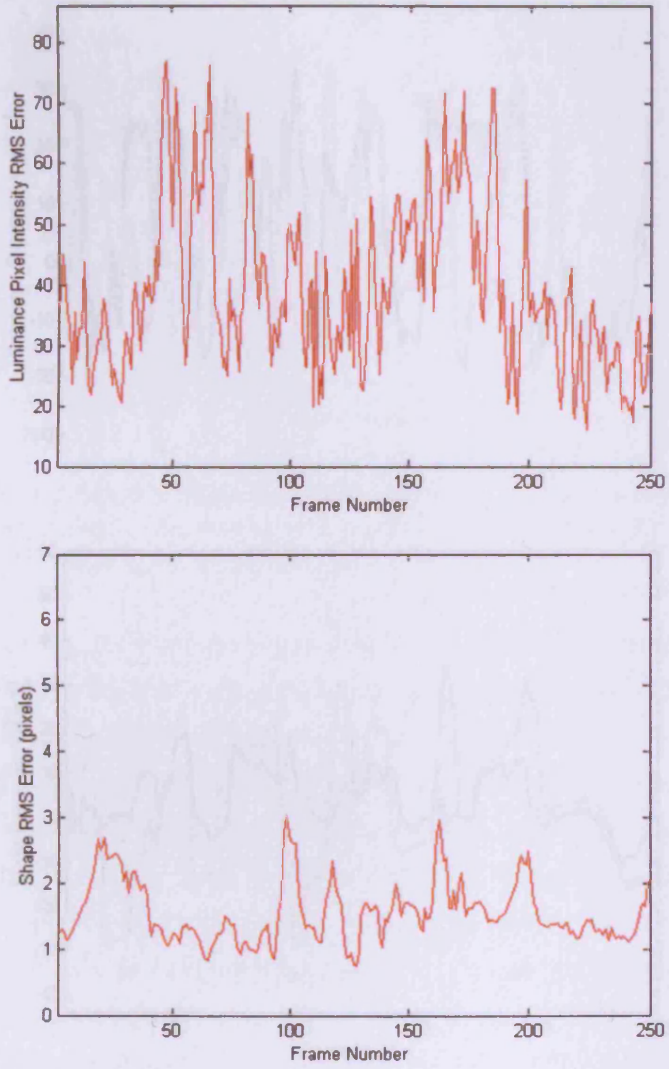

Figure 11.18: Model 2 mouth HMCM luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "And tried a mouthful of the next bowl of porridge. Too sour she said.".

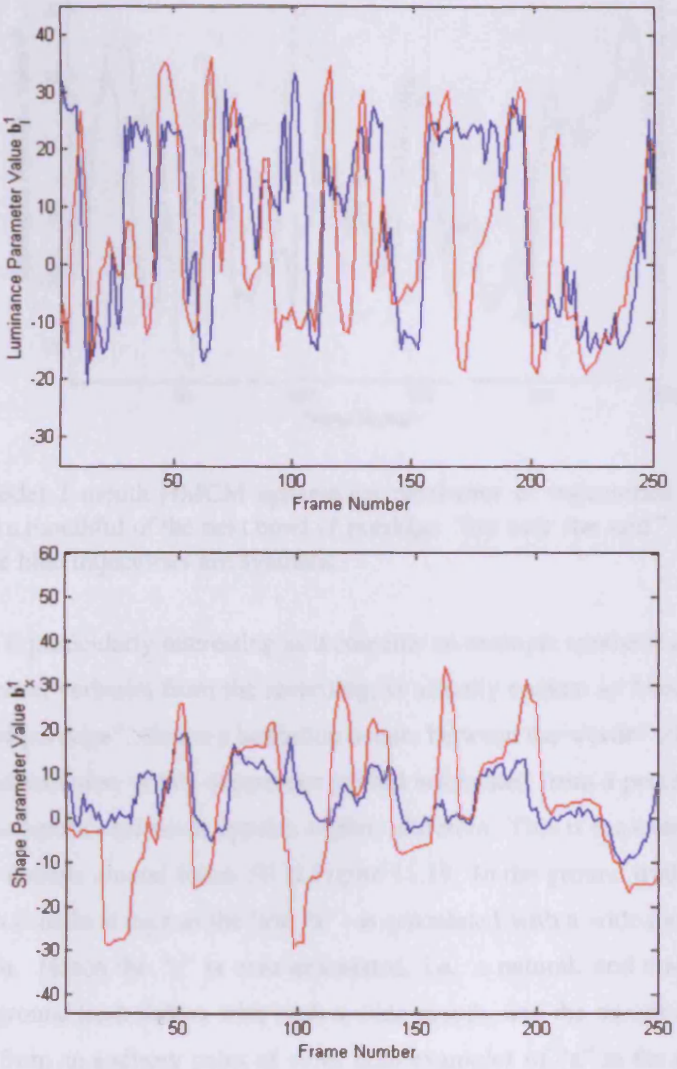

Figure 11.19: Model 2 mouth HMCM luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "And tried a mouthful of the next bowl of porridge. Too sour she said.". Red trajectories are ground truth while blue trajectories are synthetic.

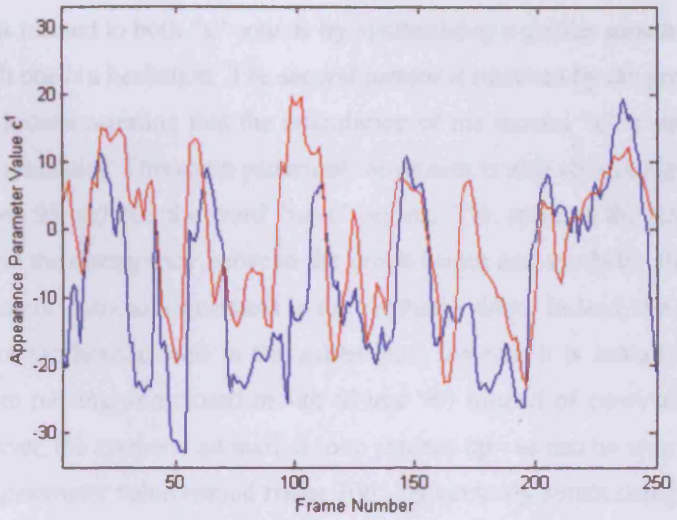

Figure 11.20: Model 2 mouth HMCM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "And tried a mouthful of the next bowl of porridge. Too sour she said.". Red trajectories are ground truth while blue trajectories are synthetic.

This sentence is particularly interesting as it contains an example synthetic articulation of a hesitation. The sentence, verbatim from the recording, is actually spoken as "And tried a, a spoonful of the next bowl of porridge". Hence a hesitation occurs between the words "tried" and "spoonful". The synthetic representation of this occurrence is well articulated from a perceptual point of view, although the ground truth equivalent appears slightly different. This is the cause of the RMS luminance pixel error notable around frame 50 in Figure 11.18. In the ground truth representation, the hesitation - which is defined here as the first "a" - is articulated with a wide mouth moving quickly to a closed mouth. Hence the "a" is over-articulated, i.e. a natural, and un-hesitated "a" is not attributed in the ground truth videos with such a wide mouth, and the mouth does not completely close. However, from an auditory point of view, both examples of "a" in the sentence sound very similar. In the synthetic video, both of the "a" sounds therefore appear almost identical. The difference in articulation between both of the "a" sounds in the ground truth video and the synthetic video are the cause of the RMS luminance errors near frame 50, and can also be seen in evidence in the luminance trajectory of Figure 11.19 and the appearance trajectory of Figure 11.20. In Figure 11.19, the same synthetic pattern (i.e. the blue trajectory moving up and down over the course

of approximately 10 frames) is seen just before and around frame 50. This demonstrates how the synthetic video has reacted to both "a" sounds by synthesising a similar mouth articulation in both cases - even though one is a hesitation. The second pattern is matched by the ground truth trajectory (the red trajectory), demonstrating that the articulation of the second "a" is similar in both videos (ground truth and synthetic). This same pattern of movement is also seen in Figure 11.20.

Between frames 95 and 109 the word "next" begins. The spike in the RMS luminance error at this moment, and the discrepancy between the ground truth and synthetic shape, luminance and appearance parameters, point to a weakness in the synthetic video. Indeed, the 'n' at the beginning of this word is not synthesised well in the animation. Instead, it is initially formed accurately in frame 94, before moving to a closed mouth (frame 96) instead of continuing with the correct articulation. However, the synthetic animation soon catches up - as can be seen by the late increase in the appearance parameter value around frame 106 - by correctly synthesising the "xt" part of the word.

The RMS luminance error between (approximately) frames 160 and 190 is due to the synthetic mouth coming to a rest before pronunciation of the phrase "too sour she said", and the ground truth video preparing articulation of the word "too". After this short segment of silence the remaining part of the sentence - "too sour she said" - is articulated in the synthetic video very well. Figures 11.21 and 11.21, as well as demonstrating frames mentioned in some of the earlier commentary on this sentence, show the articulation of the "t" in the word "too" (frame 187), the "s" in the word "sour" (frame 199) and the "ai" segment of the word "said" - pronounced *eh* (frame 235).

11.2.5 Mouth HMCM: Non-Verbal Animation Synthesis

Synthetic animations can be greatly enriched by the inclusion of non-verbal articulations. Since the HMCM is trained on *sound* as opposed to relationships between discrete symbols - i.e. phonemes - it has the ability to realistically synthesise non-verbal animations given non-verbal speech. In this thesis the term non-verbal does not include emotion. Instead it refers to sounds such as *sighs* and other exclamations of relief (e.g. "Ahhhhh"), exclamations of joy or excitement (e.g. "woo hoo!"), and also more comic sounds such as *lip-smacks*.

Four main non-verbal sentences are synthesised and demonstrated in this thesis:

- "Tasty! < *lipsmack* > < *lipsmack* >".

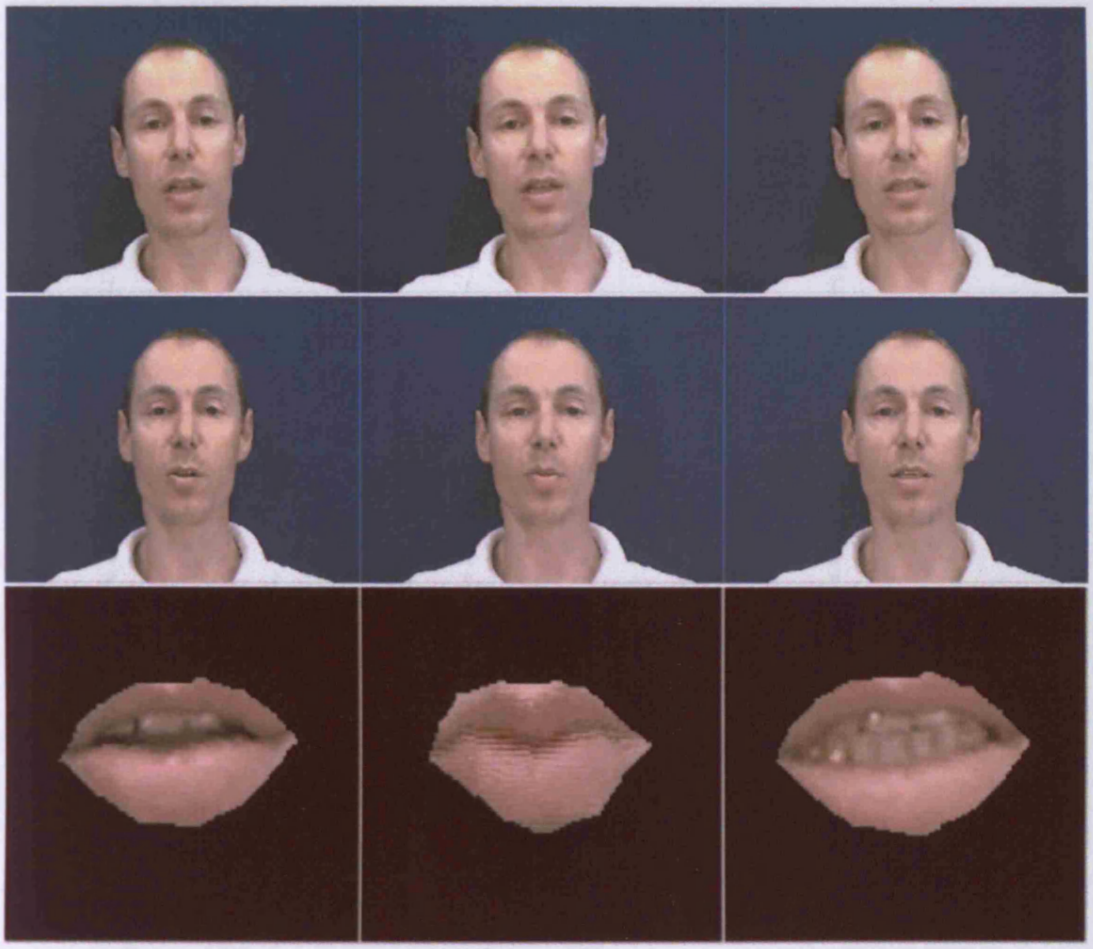

Figure 11.21: Selected frames from the synthesised sentence "And tried a mouthful of the next bowl of porridge. Too sour she said.". Synthesis is achieved using a mouth HMCM constructed with hierarchical model 2.

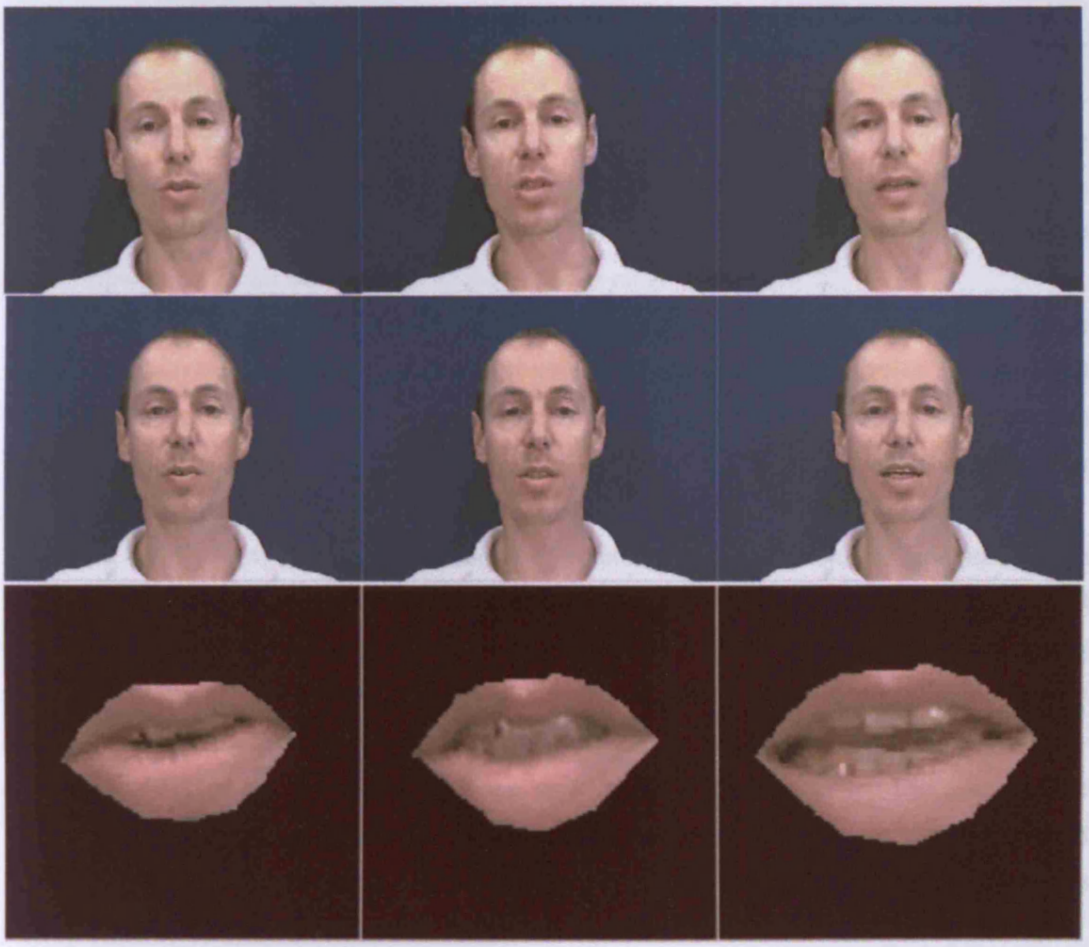

Figure 11.22: Selected frames from the synthesised sentence "And tried a mouthful of the next bowl of porridge. Too sour she said.". Synthesis is achieved using a mouth HMCM constructed with hierarchical model 2.

- "Fantastic! $< whistle >$".
- "Huuuuh...Ahhhh.....It's a hard life!".
- "Choo choo!!! $< blow >$, $< blow >$, $< blow >$, $< blow > < blow >$...".

The $< lipsmack >$ sound is created by the participant tightly closing the lips, and then opening them quickly while *sucking in air* creating a *smacking* sound (similar sounding to the word "muh"). The $< whistle >$ sound begins at a low pitch, and reaches a high pitch half way through before reaching a lower pitch one again at the end. The $< blow >$ label given to the last sentence denotes a blowing sound made to resemble a train engine. Thus, the first and last sentence are somewhat comic. However, they do well to illustrate the synthesis range of the HMCM.

The above sentences were synthesised using hierarchical model 2. An important point to make is that *no non-verbal articulations* appear in this training set. Synthesis has therefore been achieved in the HMCM by finding similar *sounds* in the training set, and piecing these together to create a best fit animation. This immediately demonstrates an advantage of using continuous speech for animation - as this cannot be done with phonemes.

Figure 11.23, 11.24, 11.25 and 11.26 illustrate mouth synthesis of the above sentences from a HMCM. In the Figures, an underscore is used to denote silence. Unfortunately, acceptable ground truth data for trajectories and texture is not available in order to run a direct comparison with synthetic equivalents. This is because, as previously mentioned, the synthesised non-verbal sentences did not appear in the original training set, and were instead recorded later. As a result, the appearance of the participant in the ground truth non verbal video is different from when capturing the hierarchical model 2 corpus. The *visual* recording conditions were also different, e.g. with respect to illumination. However, the *auditory* recording conditions remained the same. Synthesised videos given the above sentences can therefore be found at

- $http://www.cs.cf.ac.uk/users/D.P.Cosker/NonVerbal.html$.

Figure 11.23 shows synthesis of the first non-verbal sentence. Synthesis of the word "Tasty" is satisfactory as the Figure suggests. The $< lipsmack >$ is represented in the Figure as "muh", and is shown stretched over a number of frames. The first $< lipsmack >$ is shown in its entirety, while only the beginning of the second $< lipsmack >$ is shown.

Figure 11.24 shows synthesis of the second non-verbal sentence. Again, synthesis of the word before the non-verbal sound is convincing. The $< whistle >$ is represented here using the word

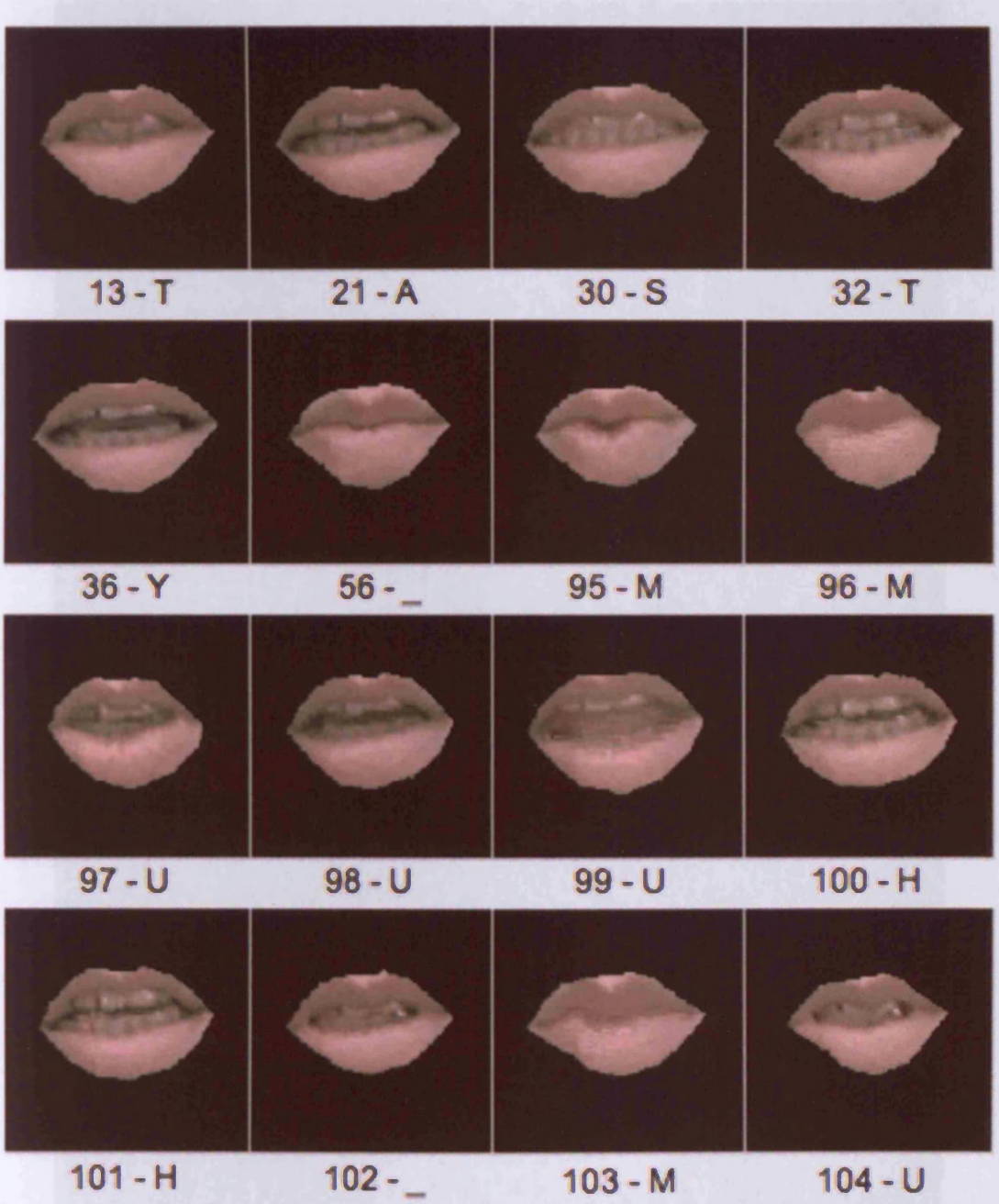

Figure 11.23: Synthesis of the phrase "Tasty! $<$ *lipsmack* $>$ $<$ *lipsmack* $>$" using a mouth HMCM constructed with the training corpus of hierarchical model 2.

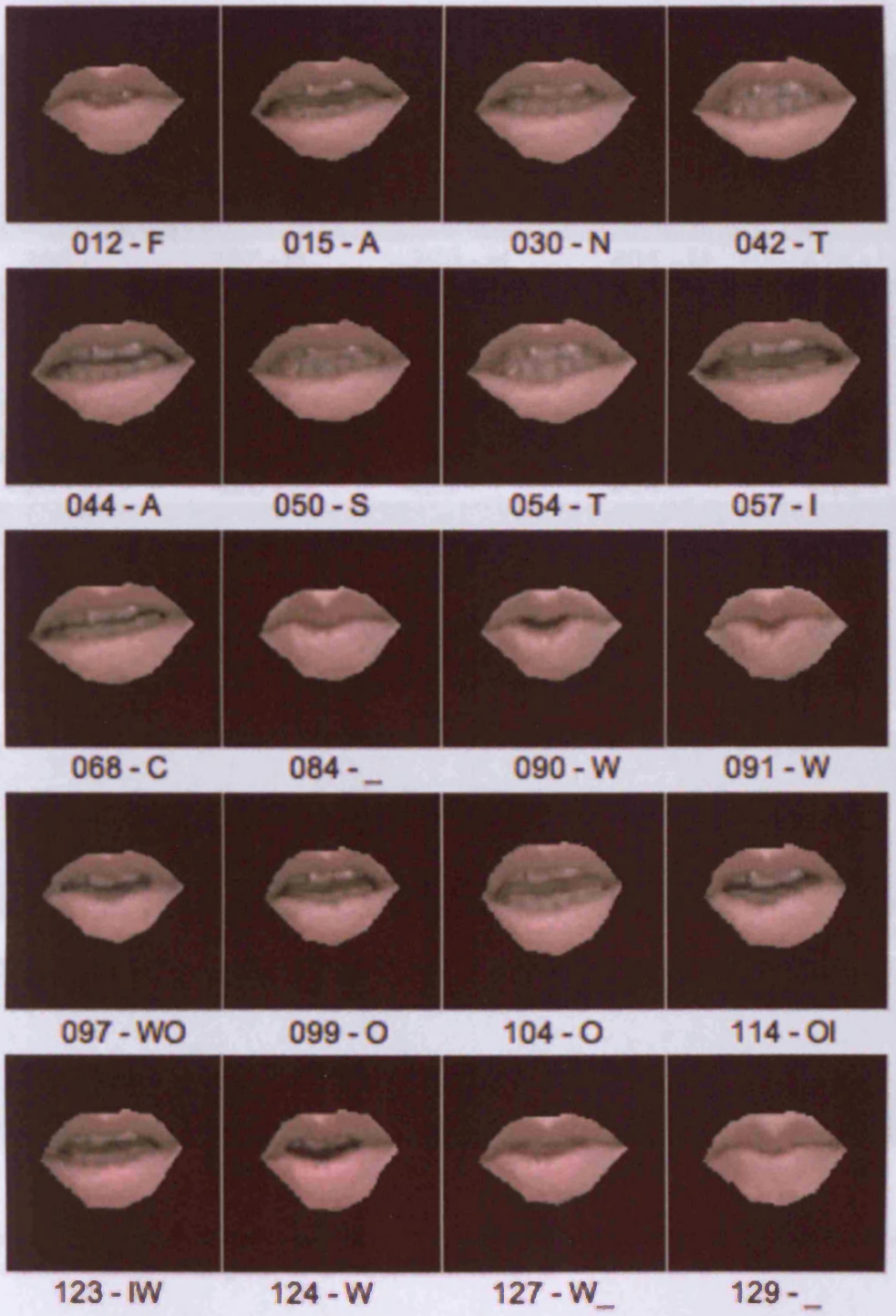

Figure 11.24: Synthesis of the phrase "Fantastic! $<$ *whistle* $>$!" using a mouth HMCM constructed with the training corpus of hierarchical model 2.

Figure 11.25: Synthesis of the phrase "Huuuuh....Ahhhh.....It's a hard life!" using a mouth HMCM constructed with the training corpus of hierarchical model 2.

Figure 11.26: Synthesis of the phrase "Choo choo!!! $< blow >$, $< blow >$, $< blow >$, $< blow >$ $< blow >$...". using a mouth HMCM constructed with the training corpus of hierarchical model 2.

"woiw". Synthesis of the $< whistle >$ is generally satisfactory - however in order to create a real whistle it would be expected that either the teeth would be in a closed state, or the lips would be pursed throughout. However, considering that there are no examples of whistling in the training set, the HMCM produces a surprisingly good approximation.

The non-verbal "Huuuuh.....Ahhhhh" in Figure 11.25 is synthesised well, and is visually convincing. The following phrase "It's a hard life" is generally articulated well. However, the "f" in the word "life" is weakly produced.

The final comic train noise "Choo choo!! $< blow >, < blow >, < blow >< blow >, < blow >$" is surprisingly accurate given that there are no examples of *blowing* sounds in the original training set. However, one flaw appears to be under-articulation of the second "ch" (frame 629), where the teeth are not closed enough. The $< blow >$ sounds at the end of the phrase are pleasingly represented by a relaxed mouth (e.g. frames 679 and 683), followed by a pursed mouth during exhalation (e.g. frames 680 and 684).

11.3 Animating a Face Model using a HMCM

As well as providing meaningful parameters for sub-facial animation, the hierarchical model also improves visual-speech synthesis by considering only speech and *mouth* data. That is, when learning relationships between audio and visual data, only the mouth is considered as opposed to a larger part of the face. This strategy is also followed in several other works, but the motivation is generally different. In [62, 22] only visual-speech synthesis is attempted, and animation of the rest of the face is ignored. Hence, any visual information not directly associated with the mouth is redundant to the visual-speech learning and synthesis algorithm. Also, since only visual-speech information is produced, animation of the remainder of the face is not required.

However, irrespective of these motivations to use only mouth-related visual information, there is another good reason why this is the best strategy for training visual-speech synthesis algorithms. If a larger portion of the face is used, containing redundant information, then certain mouth poses will always reproduce this information when displayed. For example, if a large facial region is considered for the visual data, and if one of the closed mouth images also contains a closed eye, then whenever the closed mouth image is chosen, the eye will also close - invariably at an untimely moment. The only way to avoid this effect would be to ensure the face outside of the mouth region remains neutral throughout the training set. This may sound like an obvious observation, but again

it highlights why a hierarchy of sub-facial appearance models is better for animation purposes than a flat facial appearance model.

In terms of SAMs and HMCMs, there is also another good reason why only mouth-related visual information is used. In order to achieve good performance, data clustering in SAMs and HMCMs is important. Clusters in both algorithms should contain – whenever possible – distinct groups of visual data. Since visual-speech synthesis is the goal, the aim is to cluster similar mouths. If *face* images are used as opposed to mouth images, then clustering will be biased by information outside the mouth, i.e. by variation in the upper face. Clustering will therefore be sub-optimal.

Figures 11.27 and 11.28 show mouth RMS errors and parameter trajectories created using a HMCM trained with face images from the hierarchical model 2 training corpus. The RMS errors and trajectories were produced by extracting the mouth from the subsequent full-facial HMCM animation. In a comparison with the mouth only HMCM equivalent (Figures 11.18 and 11.19), this sample result demonstrates a generally poor synthesis, with little visual correlation between the synthesised and ground truth trajectories.

Another issue when training a SAM or a HMCM using full facial data is the strain imposed on the GMMs and HMCMs. Since face parameters are larger than mouth parameters, it is more difficult to balance the amount of energy retained against the number of clusters allowed before memory becomes an issue. If too little energy is retained in the face parameters, then not enough facial variation will be modelled and synthesised. If too much energy is used, then it is difficult to build a GMM or a HMM with a satisfactory number of clusters/states.

Nevertheless, the best approach is still to use mouth-related visual-speech parameters due the animation related advantages in the remainder of the face, and the reduction of redundant data incorporated into the visual-speech synthesis algorithm.

11.4 Visual-Speech Synthesis using a SAM

An overview of SAM performance using hierarchical model 2 is now given. To reiterate, further evaluation details may be found in [42, 43, 44]. This Section demonstrates the performance of an example SAM synthesis given speech not originally included in the training set using comparisons with ground truth data.

Leaving one observation out, a SAM was constructed with 120 Gaussian mixtures and an appearance model energy proportion of X. The observation omitted from the training set consisted of

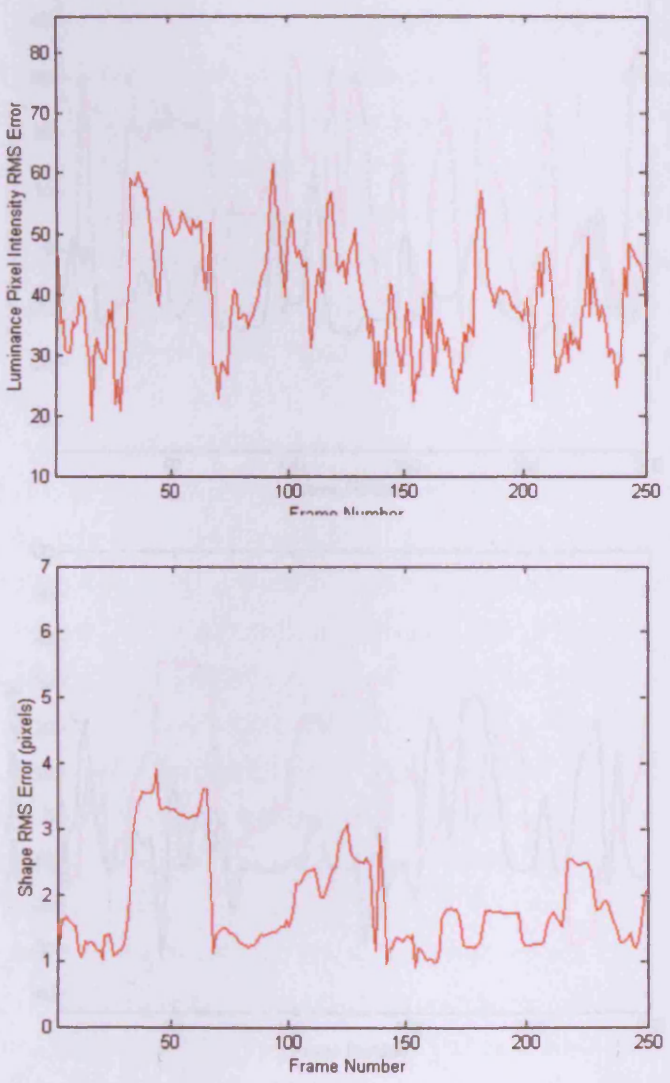

Figure 11.27: Model 2 mouth luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "And tried a mouthful of the next bowl of porridge. Too sour she said.". The results were generated using mouth data extracted from a face HMCM.

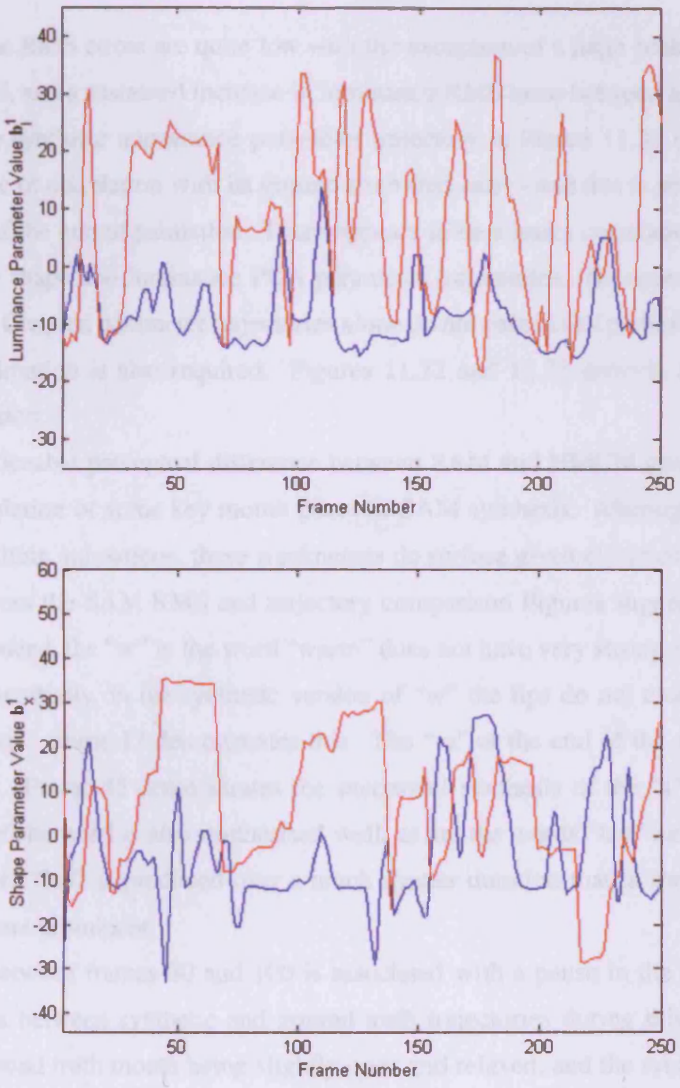

Figure 11.28: Model 2 mouth luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "And tried a mouthful of the next bowl of porridge. Too sour she said.". The results were generated using mouth data extracted from a face HMCM. Red trajectories are ground truth while blue trajectories are synthetic.

the sentence "It looked nice and warm and as she lay down, she closed her eyes and fell asleep". Figures 11.29, 11.30 and 11.31 show RMS error and trajectory comparisons resulting from the synthesis.

In general, the RMS errors are quite low with the exception of a large peak in shape RMS error between frame 40, and a sustained increase in luminance RMS error between approximately frames 80 and 110. The synthetic appearance parameter trajectory in Figure 11.31 (blue line) displays a reasonable degree of correlation with its ground truth (red line) - and this is promising with respect to the accuracy of the output animation. There appears to be a lesser correlation between synthetic and ground truth shape and luminance PCA parameter trajectories. However, as discussed at the beginning of this Chapter, parameter trajectories alone do not paint a full picture, and an examination of the output animation is also required. Figures 11.32 and 11.33 provide an illustration to the following discussion.

The most noticeable perceptual difference between SAM and HMCM generated animations is the weaker articulation of some key mouth poses in SAM synthesis. Although this is not entirely damaging to resulting animations, these weaknesses do surface given closer scrutiny.

Evidence across the SAM RMS and trajectory comparison Figures suggests an initial error in the animation. Indeed, the "w" in the word "warm" does not have very strong articulation (although it is present). Essentially, in the synthetic version of "w" the lips do not close as much as in the ground truth video - frame 17 demonstrates this. The "m" at the end of the word "warm" is also under articulated. Frame 45 demonstrates the successful synthesis of the "s" at the beginning of "she". The rest of the word is also synthesised well, as are the words "lay" and "down". However, the "l" in the word "lay" is produced over a much shorter duration that in the ground truth video, and is therefore less prominent.

The period between frames 80 and 100 is associated with a pause in the speech. RMS errors and discrepancies between synthetic and ground truth trajectories during this event are primarily caused by the ground truth mouth being slightly open and relaxed, and the synthetic mouth moving between a relaxed and open mouth, and a fully closed mouth.

After this pause, articulation is good for the word "she", which begins in frame 106 with the synthesis of "s". The word "closed" is also synthesised satisfactorily - frame 117 demonstrates the synthesis of "o" in this word. The "f" in the word "fell' is weakly articulated (frame 152), while the word "asleep" is synthesised well. Frames 175 and 182 demonstrate synthesis of "e" and "p" in the

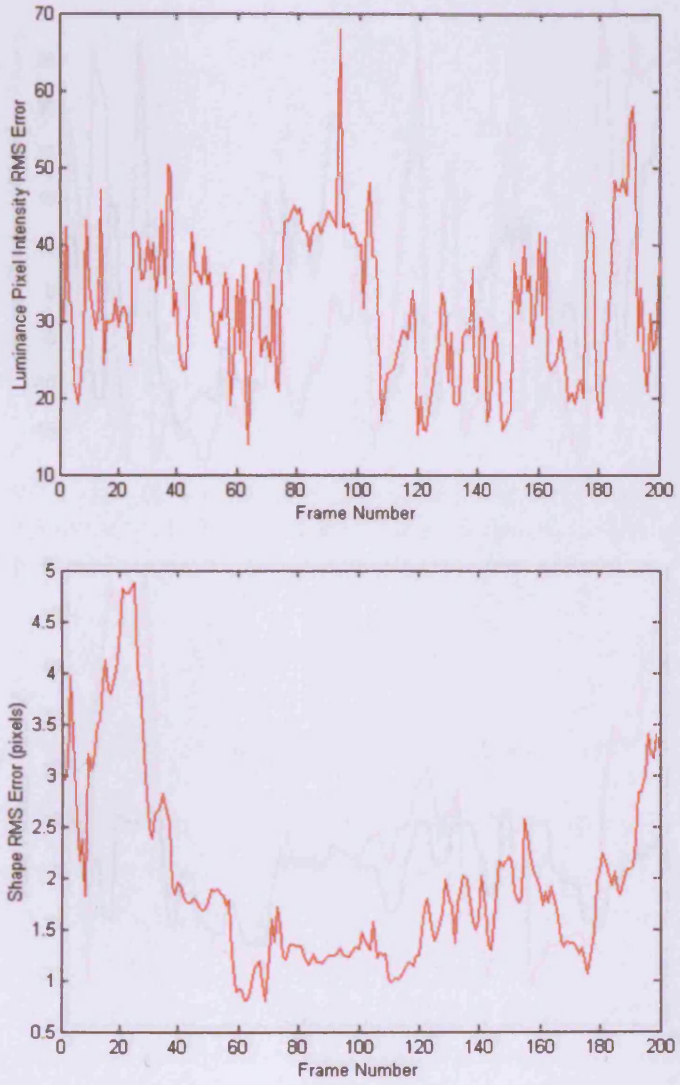

Figure 11.29: Model 2 mouth SAM luminance intensity (0-255) and shape coordinate pixel RMS errors for the synthesised phrase "It looked nice and warm and as she lay down, she closed her eyes and fell asleep".

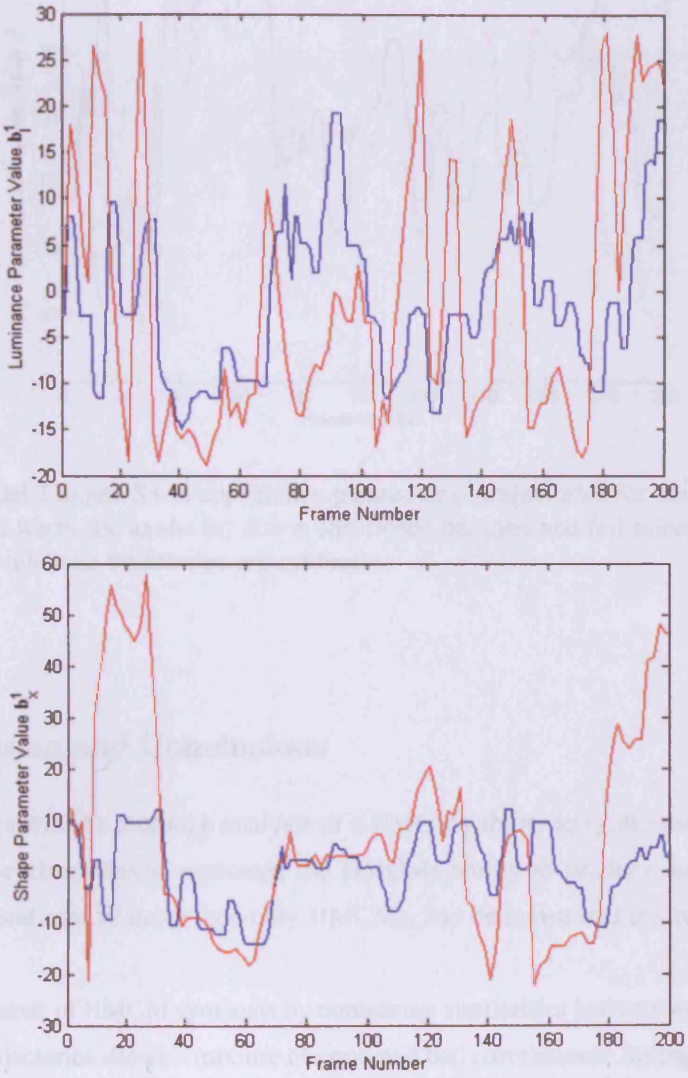

Figure 11.30: Model 2 mouth SAM luminance $\mathbf{b}_l^1$ and shape $\mathbf{b}_x^1$ parameter trajectories for the synthesised phrase "It looked nice and warm and as she lay down, she closed her eyes and fell asleep". Red trajectories are ground truth while blue trajectories are synthetic.

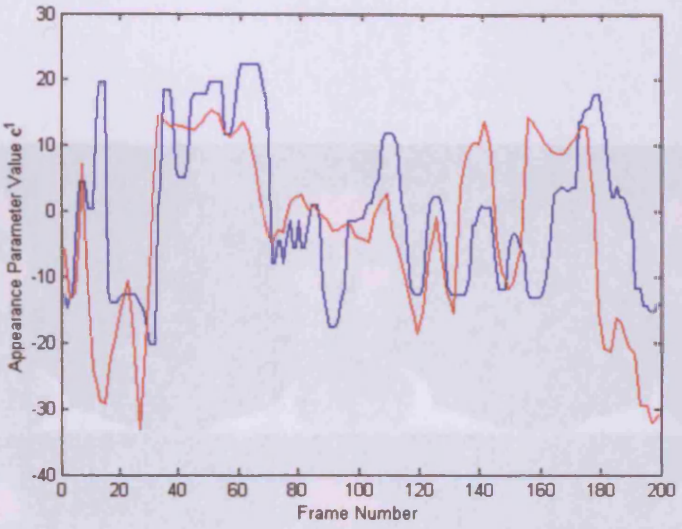

Figure 11.31: Model 2 mouth SAM appearance parameter $\mathbf{c}^1$ trajectories for the synthesised phrase "It looked nice and warm and as she lay down, she closed her eyes and fell asleep". Red trajectories are ground truth while blue trajectories are synthetic.

word "asleep".

11.5 Discussion and Conclusions

This Chapter has provided a thorough analysis of a HMCMs ability to synthesise animation parameters for visual-speech synthesis, evaluated the HMCMs ability to produce non-verbal synthesis, weighed the pros and cons of using face-only HMCMs, and demonstrated the synthesis standard of a SAM.

The demonstration of HMCM synthesis by comparing similarities between synthetic and ground truth parameter trajectories shows a mixture of good and bad correlations. Strong visual correlations do exist, but in general synthetic trajectories have weak visual correlations to ground truth trajectories. Some of these weaknesses can be explained through visual analysis of the output animations, and these have already been considered (they are also summarised below). However, since from a perceptual standpoint the animations appear for the most part convincing, it suggests a flaw in the trajectory comparison approach employed. Therefore, even though synthetic trajectories may often

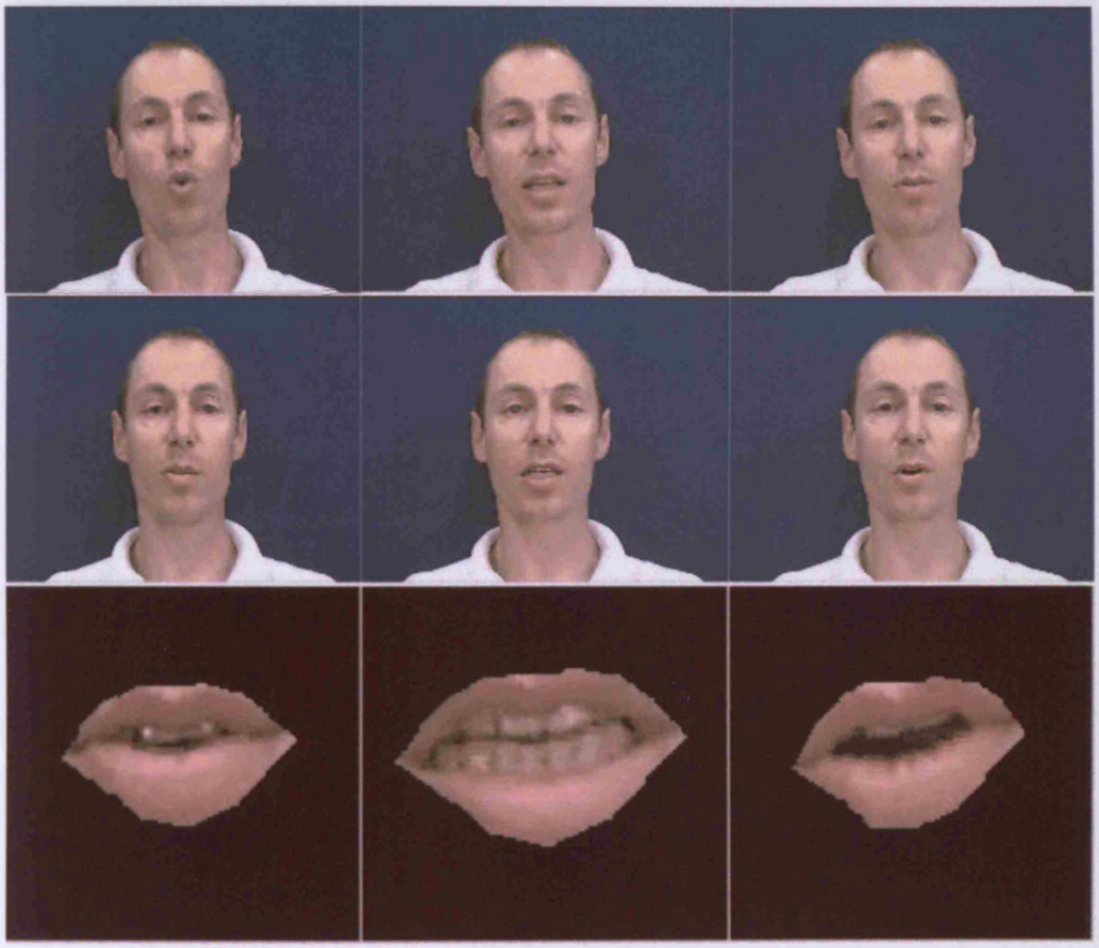

Figure 11.32: Selected frames from the synthesised sentence "It looked nice and warm and as she lay down, she closed her eyes and fell asleep". Synthesis is achieved using a mouth SAM constructed with hierarchical model 2.

Figure 11.33: Selected frames from the synthesised sentence "It looked nice and warm and as she lay down, she closed her eyes and fell asleep". Synthesis is achieved using a mouth SAM constructed with hierarchical model 2.

appear different from ground truth trajectories, this in fact may not matter for a perceptual point of view.

Periods of silence appear to be synthesised differently from ground truth equivalents - however this has no real detrimental effect in perceptual quality. The only criticism would be that the mouth almost always closes during periods of silence in synthetic videos, whereas in ground truth videos the mouth will move between fully closed, and a relaxed open state. One suggested approach to rectifying this might be to include longer examples of silence in the training corpus in order to bolster the HMCM with alternative *silence routes* through its HMM.

One disadvantage of using continuous speech appears to be that quickly occurring speech sounds do not stand out enough to be robustly encoded in the HMCM. For example, in synthesis of the sentence "When she got inside she saw a large wooden table and three wooden chairs", the "th" in the word "three" is pronounced very quickly, and consequently synthesised weakly. On the other hand the "th" appearing in the word "mouthful", taken from synthesis of the sentence "She took a mouthful from the large bowl", is synthesised well - and is much more noticeable in the audio. In some respects these fast speech articulations are almost *silent*, so it is no surprise that they are missed out of synthesised animations. Based on this analysis it is therefore difficult to come to a solid conclusion as to which articulations are commonly synthesised well, and which are synthesised poorly. The only firm conclusion that can be made at this point is that if speech events are observed as *silent* in a speech signal, there is an increased chance of them being under-articulated in a synthetic animation.

This is an advantage of phonetic systems since each speech sound (or unit) is labelled with a phoneme irrespective of whether or not it is audible in the respective sound track. It could be argued that automatic phoneme label systems might also suffer from this disability, since they too use continuous speech inputs. However, transcriptions resulting from such a system can be manually checked for such errors.

From a continuous speech point of view, the solution is then perhaps to use smaller speech samples for training and synthesis, e.g. 100Hz as opposed to 50Hz. This could improve synthesis since barely audible sounds should still leave a trace signature in the speech signal. These smaller signals should then theoretically be noticed by the HMCM, and be encoded more robustly into its HMM.

One other observation from the results is that some synthetic articulations begin later than their

ground truth counterparts. Generally, this is hardly perceivable in the resulting animations, but it does on occasion tend to make some articulations appear rushed. A delay in synthetic articulation is associated with a period of silence in the voice track, where the mouth might usually prepare itself for the onset of speech (e.g. forward articulation). Since this preparation appears to be omitted in the synthesised animations, it means that the HMCM is not looking *forward* enough along the speech track in anticipation of some sounds. This might be improved by modifying the Viterbi algorithm to place more consideration on future states when defining state-visit error ratings in its forward-backward algorithm. Another possibility might be to consider more future states in the Trellis search algorithm. However, perhaps a more successful option would be to use *pathlets* of visual-speech features instead of single frame visual-speech features. Pathlets [72] are short continuous segments of visual observations - typically 3 to 5 frames in length. By using pathlets, penalties imposed by late articulation in the HMCM algorithms would be reduced since the error induced by choosing an inactive (closed) mouth during the onset of an articulation would become greater than the error induced by choosing an active (preparatory) mouth. This is because the speech segment corresponding to the *active* pathlet (which would contain silence, and then speech) would correspond better with the input speech signal (which would contain silence, and then speech) than the *inactive* pathlet (which would just contain silence). Use of pathlets would also improve the smoothness of output animations, since pathlets contain consecutive visual features taken from the training corpus. Smoothness would then depend more on the transition between pathlets, and in a worst case scenario it is envisaged that a pre-processed pathlet signal would always be more stable than a pre-processed *single-frame* signal.

The non-verbal animation results are very pleasing considering that there are no similar examples in the training corpus of hierarchical model 2. The capacity to create behaviours such as these adds life to synthetic animations, and highlights a potential advantage of using continuous speech for animation as opposed to phonemes. However, that is not to say that a phoneme based method using additional predefined tags for producing non-verbal animation could not perform equally as well. But the attractive aspect of the continuous speech method is in its ease of use - no extra work is required in order to predefine non-verbal articulations. The next step is obviously to perform a comparison with ground truth data, and to explore the possibility of producing other articulations for actions, e.g. yawning and coughing.

The demonstration of SAM synthesis highlights the level of robustness offered by the speech

and visual features used throughout this thesis. Since the SAM method does not incorporate coarticulation, it relies heavily on good matches between single isolated audio samples in order to produce good animations. The level of correspondence demonstrated by the trajectory outputs in the SAM example show that good matches are still obtained even without coarticulation. It is therefore not surprising that the HMCM method - which essentially builds on the SAM approach - performs better since it reduces errors caused by choosing visual vectors in isolation.

Coarticulation is therefore the major reason why the HMCM performs better than the SAM. Mouth animations which include coarticulation will always be visually more realistic than those that do not.

One symptom of a system which does not encode coarticulation is the mouth returning to rest after the synthesis of each individual sound, or no smooth visual transitions between different mouth postures. This is seen more in SAM synthesis than in HMCM synthesis. HMCM animations of the mouth will not always return to rest (neutral mouth) because the system looks ahead to see what speech event will occur in the future and prepares accordingly. In a SAM this does not happen because clusters are chosen based on the audio witnessed at that specific moment. HMCM animations are not always perfect in this respect however, and will occasionally return the mouth to rest during synthesis of a word where it would normally not be expected.

The SAM is good at capturing vowel sounds and there appears to be a good reason for this. Mel-Cepstral coefficients are well suited to vowel recognition. Since SAM synthesis is largely based on classifying the current speech sound (represented using a Mel-Cepstral feature) into one of a number of speech clusters – many of which will largely contain Mel-Cepstral features representing specific vowel sounds – this good synthesis is not surprising. Vowel sounds are usually very audibly distinct, meaning that Mel-Cepstral analysis should be effective in picking them out from a speech recording.

HMCMs are also good at synthesising vowels. Like SAMs, HMCMs also use Mel-Cepstral features and similarly also enjoy all the benefits that these features possess in encoding vowel sounds. If one were to synthesise an audio track consisting solely of individual vowels using a SAM and a HMCM, then both would output the correct mouth from a visual point of view. However, the transition between the mouth at rest, to the peak of the vowel sound and back to rest, would be far smoother and visually more appealing in the HMCM animation.

Consonant synthesis using SAMs depends on the length and magnitude of the consonant sound. For example, a /V/ sound spoken quickly would probably not be recognised and therefore not be

synthesised by the SAM. On the other hand an audibly distinct /B/ sound occurring over a longer number of frames will almost certainly be synthesised. The same is true in HMCM synthesis. However, in HMCM synthesis any transitions between different (successfully recognised) consonant and vowel combinations are also better articulated, and appear smoother. The SAM will often attempt to move the mouth to a rest position in between synthesising these different sounds - incorporating a jerkiness into the animations. HMCM recognition of consonants is also perhaps slightly better than SAM recognition since the audio is analysed over time. This means that the influence of a future consonant - no matter how weakly audible - will have an influence on recognition during the current time frame (no matter how weak this might be).

Improved synthesis of consonants in both systems depends on successfully recognising a consonant in the first place. Since both systems are often weak in synthesising fast consonants (or plosives) then shortening the speech sampling window from 50Hz to 100HZ would likely improve animations.

To conclude, the main advantage of HMCMs over SAMs is the lower overall perceptual error rate (i.e. less visually incorrect mouth postures), better correlations between synthetic and ground truth trajectories, better transitions between visual postures and better capture and synthesis of consonants.

One general defect in synthesised animations at this time is an occasional texture flicker. Part of this is caused by the current warping method employed - piecewise affine warping. The size of the facial images used is quite small, and there are a relatively large number of landmarks. This means that delaunay triangulation of the face produces small triangles, and each triangle contains a significant amount of visual face information. The piecewise affine warping method does not preserve straight lines across triangles. In the face, these straight lines correspond to facial detail. Therefore, after warping, it is often the case that triangles may not correctly match up with respect to detail across their boundaries - and this can cause texture flicker. A solution to this problem would be to use larger images, and a more stable warping technique.

Another contributor to texture flicker is a lack of smoothness in some parts of the animation. This could be improved on by using *pathlets* as observations as opposed to single frame sized visual observations (this has already been discussed earlier in this Section). Another solution would be to improve post-processing of the synthetic appearance signal. The current method reduces most of the signal noise, and preserves signal detail in an intelligent manner. However, some noise still remains

in parts of the signal which should otherwise be stable. Using the current approach, these parts of the signal are difficult to remove without reapplying the filter several times. The disadvantage of this approach is that important detail could be *blurred* or over-smoothed. A better approach might be to fit splines to the data after initial filtering. Given a stable mean across noisy periods of the signal, a spline should fit between the noisy peaks - thus eliminating noise and preserving signal stability. At regions where the mean changes by a large magnitude over a short window, i.e. during periods accounting for important detail, spline fitting could be ignored. Thus, the approach would intelligently apply splines based on local signal information just as the current approach Gaussian filters the signal based on local standard deviation.

The next Chapter takes a perceptual approach to evaluation, and introduces a novel experiment based on the McGurk effect for evaluating the effectiveness of visual-speech synthesis. The experiment is applied to HMCM generated animations, and uses hierarchical model 2 as a foundation.

Chapter 12

Perceptual Evaluation of Facial Animation

The drawback of analytical evaluation methods is that they do not address the question of whether animations are convincing from a perceptual point of view, i.e. do the animations actually *look* convincing and/or realistic, and what exactly *makes* them look convincing? This has already been seen with the use of RMS errors to evaluate animations in the previous Chapter.

This Chapter evaluates the perceptual realism of visual-speech in synthesised animations using a novel McGurk based test. The test is general, in that it may be applied to the evaluation of the visual-speech in any facial animation, either cartoon based or video realistic. The test is also extremely subtle, in that participants are not made directly aware that they are actually evaluating the performance of visual-speech.

Using this test it is shown how strengths and weaknesses in visual-speech synthesis algorithms may be identified perceptually, and how improvements and directions for future work may be found based on an analysis of the results. The test also considers optimum display settings for the presentation of synthetic facial animations, and how the variation of display size affects perceptual performance.

The McGurk test is applied in this Chapter to the evaluation of HMCM generated facial animations using hierarchical model 2. In order to evaluate the graphical and behavioral realism of animations, the test is ended by asking participants whether or not they noticed any of the presented animations were computer generated. This provides an additional measure to the McGurk

test specific to the chosen medium of output - which is intended to be as video-realistic as possible.

12.1 Perceptual Evaluation Techniques

The quality of synthetic facial animation, produced solely from speech, has been measured using various approaches. These include subjective assessment [22, 40, 64], visual comparison of synthetic versus ground truth animation parameters [145, 32], measurement of a test participant's ability to perceive audio in noisy environments with the aid of synthetic animation [40, 116] and through specific forced choice testing where a participant is asked to decide whether or not they believe a presented animation to be synthetic or real [67, 72]. Each of these tests primarily assesses computer generated lip-synchronization to speech - a good result is obtained given strong visual-speech synthesis just as poor results will follow from weak visual-speech synthesis.

Subjective evaluation is the most common method and typically entails comments on the animations from the designers and a number of naive test participants. The observations of the participants are then demonstrated using example videos of the visual-speech animations. Visual comparison of synthetic versus ground truth parameters involves comparing the trajectories of speech synthesized mouth animation parameters, with trajectories of ground truth mouth animation parameters, typically obtained from a real speaker. The first method provides subjective information on the overall quality of lip-synching, but leaves no means of comparison with other systems, or no direct method of determining any strengths or weaknesses inherent in the synthesis algorithm, e.g. which Visemes are correctly synthesised? The second method provides more insight into an algorithm's strengths and weaknesses, and a more quantitative measure of a system's overall effectiveness. However, taken on its own it provides no means of communicating the perceptual quality of an animation, i.e. is mouth animation visually convincing?

Measurement of the ability of a speech-driven synthetic talking head to improve the intelligibility of speech in a noisy environment gives a good indication of the overall quality of lip-synching when compared to the performance, in the same circumstances, of real or ground truth speaker footage. This measure, along with comparisons of synthesized trajectories to ground truths, gives a good overall picture of a talking head's lip-synch ability. However, perceptually we are still unaware as to what visual-speech segments, or Visemes, are synthesized well, and which are synthesized poorly. For example, if poor results are obtained, which Visemes contribute to this? Similarly, which Visemes are responsible for a strong a result?

The specific forced choice experiments performed by Ezzat *et al* [64] provide perhaps the most thorough and rigorous talking head evaluations to date. In these tests, a series of experiments are carried out where participants are asked to state whether a displayed animation is real or synthetic. If the animations are indistinguishable from real video then the chance of correctly identifying a synthetic animation is 50/50. The test may be thought of as a kind of *Turing Test* for facial animation. In [64, 67] the facial animation is produced from phonemes, thus the test determines the quality of lip-synchronization, and a final positive or negative result will follow based on the overall quality of the visual-speech.

A drawback of this particular experiment is that it simply asks whether or not synthetic lip-synched animations *look* realistic - the tests provide no insight into what exactly makes the animations *look* good, and what exactly might make them *look* bad (e.g. what Visemes are synthesized well? How does the correct - or incorrect - synthesis of a Viseme contribute to the overall quality of the animations?).

12.2 A McGurk Test for Perceptual Evaluation

McGurk and MacDonald [109] noted that auditory syllables such as /ba/ dubbed onto a videotape of talkers articulating different syllables such as /ga/ were perceived as an entirely different syllable, e.g. /da/. During such a test, when a participant closes his or her eyes the illusion created by the integration of both stimuli vanishes, leaving the participant with perception of the auditory signal alone. This raises important questions in audio-visual analysis, such as how do humans integrate and combine auditory and visual stimulus, and why do we combine such information when the auditory signal is by itself sufficient?

MacDonald and McGurk argue that when information from both visual and auditory sources is available, it is combined and synthesized to produce the "auditory" perception of a best-fit solution. The *McGurk effect* (as the phenomena is now widely known) has been replicated several times [106, 46, 50] using varieties of visual and auditory stimuli. An interesting summary of expected misinterpreted audio syllables, given the influence of a differing visual syllable, may be found in [50].

In this Chapter a perceptual evaluation technique based on the McGurk effect is proposed, and used to evaluate the visual-speech quality of HMCM generated talking head animations using hierarchical model 2. The test allows for the evaluation of any synthetic lip-synch generated automatically

from speech, or for the comparison of lip-synch created by one method against another. In the tests, participants are shown real and synthesized *McGurk tuples* [1], one video clip at a time, and chosen at random from a database of video clips.

Real McGurk tuples are generated by redubbing a video of a person speaking a word with audio taken from the same person speaking a different word. To generate *synthetic* McGurk tuples, a speech input is used to generate a video sequence using the HMCM, and it is then re-dubbed with the audio of a different word. In the experiments, a selection of reliable McGurk tuples taken from previous studies are used as a basis for coding participant responses [46, 50, 54].

For each video the participant is simply asked what word they hear while watching. The participants are not informed that some of the clips are real and some are synthetic - thus lip-synch performance is assessed in an extremely subtle and indirect manner. The fact that some of the clips *are* generated synthetically should have no influence on the response from the test participant other than their perception of the McGurk tuple due to the quality of the animations lip-synch. If the lip-synching algorithm (in this case the HMCM) produces a poor animation then the audio cue would be expected to dominate the visual cue [109], and if good lip-synching is produced from the algorithm then a *combined* or *McGurk* response would be expected, i.e. where the audio and visual data are confused to give a response other than the audio or visual stimuli. It can be stated that the algorithm's lip-synch is effective given either a *combined* or *McGurk* response since the presence of one of these responses depends on correct articulation from the synthetic video, where this video is created from audio alone. Also the illusion of reality produced by the synthetic videos should not be broken given a bad lip-synch animation from the synthesis algorithm, since the participants find, during the course of the experiment, that the audio is *supposed to be* different from the video (i.e. according to the McGurk tuple). As noted above, a participant's objective knowledge of the illusion (i.e. presence of the McGurk effect) should also not affect his or her perception of the words [109]).

After the McGurk test, the participants are then asked a series of forced-choice questions in order to determine whether or not they noticed anything *unnatural* about the videos. These questions help evaluate the performance of the animations in terms of their behavioral output and video-realism. Given sufficiently realistic synthetic video clips, no prior concerning the source of the videos should be developed by the participants, since they are not told before-hand to expect a mixture of real and synthetic clips. Again, this is a subtle test, and given any artifacts in the synthetic videos a participant

[1] For ease of exposition, a triplet consisting of a visual syllable or word, dubbed with a different audio syllable or word, along with its expected new perceived syllable or word, is referred to as a McGurk tuple.

would be expected to conclude that that some of the videos are indeed computer generated.

Positive feedback resulting from the questioning is interesting on two different levels: firstly it suggests that the synthetic animations are visually convincing to a satisfactory standard, and secondly - even if there are artifacts in the videos which the participants did not notice - it suggests that they are still convincing enough to make a naive participant believe they are real. This second point is very insightful, as it points to the possibility that computer generated animations do not have to be perfect in order to convince a person that they are real. Rather, it is only necessary for a person to *not expect* to see a computer generated animation in a given situation.

The results of the questioning, along with the success rate of the McGurk effect responses, then form an overall evaluation of the visual-speech and rendering algorithms. The tests may be regarded as both a quantitative measure of lip-synch performance and a Turing test for realism. If the animations under assessment are not intended to be video realistic, then the questioning used may not apply. Instead questioning may be modified to reflect the stimuli used.

To ensure the validity of the McGurk tuples, and also to provide a baseline for comparison, real video footage is displayed as well as synthesised footage. The real and synthesized footage include the same McGurk tuples. If a McGurk response is obtained for a real tuple then it would be hoped that a similar response would be recorded given the synthetic equivalent. If the participant responds with similar McGurk responses to both the the real and synthetic tuples, it can be stated that the synthetic lip-synching algorithm is effective. (Logically the same conclusions cannot be drawn if a McGurk effect is not perceived in either video). As a McGurk response from a participant depends on the correct articulation from the lip-synching algorithm, a non-McGurk response with the synthesized video and a McGurk response with the real video points to a weakness in the algorithm at co-articulating that specific mouth behavior or Viseme.

This allows a developer to analyze the overall results and concentrate on algorithmic development in a guided direction, whether that be by providing more training data of certain phrases, or by fine tuning the algorithm to be more sensitive to certain articulatory actions.

12.3 Experiments

The visual-speech algorithm evaluated in this Chapter is the HMCM. The HMCM was constructed for the mouth node of hierarchical model 2. In order to evaluate the talking head model using the McGurk perception test, 20 psychology undergraduate volunteers were used. This group consisted

of 4 males and 16 females, aged between 18 and 29 years (mean age 19.9). All volunteers had normal vision and hearing.

Ten McGurk tuples were used in the experiment. These were chosen through pilot testing from a larger collection of monosyllabic words taken from past research into the McGurk effect [46, 50, 54]. Table 1 gives the tuples which were chosen. These were used to construct 30 real and 30 synthetic videos, consisting of each tuple, real and synthetic, presented at three different resolutions - 72x75 pixel resolution, 361x289 pixel resolution, and 720x576 pixel resolution. These sizes were chosen as a result of pilot testing, which showed that the 72x75 pixel image produced a McGurk effect roughly 50% of the time in the real video condition, and that the three sizes produced differences between video type conditions (i.e. real or synthetic) and (non-significant) differences between sizes in the proportion of McGurk effects produced. Each video was encoded in the Quicktime movie format. Figure 12.1 gives an overview of the construction of a synthetic video tuple.

Table 12.1: McGurk Tuples

Dubbed Audio	Source Video	McGurk Response
Bat	Vet	Vat
Bent	Vest	Vent
Bet	Vat	Vet
Boat	Vow	Vote
Fame	Face	Feign
Mail	Deal	Nail
Mat	Dead	Gnat
Met	Gal	Net
Mock	Dock	Knock
Beige	Gaze	Daige

The 60 videos (30 real, 30 synthetic) were presented in a random order on a standard PC using a program written in Macromedia Director MX. This program also included two example videos (using the word sets "might-die-night" and "boat-goat-dote", which were not used in the experimental trials) and the option to replay each of the 60 experimental videos before proceeding to the next. Speakers with adjustable volume were plugged in to the PC (adjusted during the example video phase to provide a clear acoustic level) and the experiment took place in a soundproofed laboratory

with artificial lighting.

The number of McGurk responses was assessed using an open response paradigm as used by Dodd [50], requiring each participant to write down the word they had heard after viewing each video. This removes any interpretation bias which may arise if the experimenter transcribes the verbal responses. Participants were also directed to fix their attention on the mouth of the video during play back, so as to avoid the experience of an audio only stimuli. Participant responses did not have to be real words as the aim was to find exactly what they were hearing, and it could not be guaranteed that all McGurk effects would produce real words.

After viewing all 60 clips each participant was finally asked 3 questions: 1) "Did you notice anything about the videos that you can comment on?", 2) "Could you tell that some of the videos were computer generated?" and 3) "Did you use the replay button at all?". All the videos used in the test, along with the Macromedia Director MX program used to present them, may be found at *http://www.cs.cf.ac.uk/user/D.P.Cosker/McGurk/*.

12.4 Results

To code the results, two different interpretation formats were used: *Any Audio - Any McGurk* and *Expected Audio - Expected McGurk - Other*. Tables 2 and 3 list the words recorded from the test participants which constituted and warranted these formats. In the first format words which were homonyms of the audio, or which sounded very similar and could easily be confused (for example a slightly different vowel sound) were coded as "audio", while all others were coded as "McGurk". In the second format the *Expected Audio* category contained only the audio words and alternative spellings of these. The *Expected McGurk* category included McGurk words, alternative spellings of these, and words with the same consonant sound when presence or absence of voice was ignored (after [50] for example, "fent" was accepted as a response to the word set "bent-vest-vent"). All other responses were placed in the "other" category.

The rationale behind the *Any Audio - Any McGurk* coding method is that it follows in the spirit of the original McGurk observation, i.e. that a different audio word will be heard under misleading visual stimuli. The *Expected Audio - Expected McGurk - Other* category more closely follows word tuples suggested in previous work. Given the variability in different peoples accents and articulatory behavior, it is unrealistic to assume that a fixed McGurk effect response word would apply to every living person. Therefore the *Any Audio - Any McGurk* format is perhaps more reflective of a general

McGurk effect.

Table 12.2: Any Audio - Any McGurk

Tuple	Accepted Audio	Accepted McGurk
Bat-Vet-Vat	Bat	Vat, Fat
Bent-Vest-Vent	Bent,Bint	Vent,Fent,Fint
Bet-Vat-Vet	Bet	Vet,Fet
Boat-Vow-Vote	Boat,Bolt,Bought,But,Boot Booked,Port	Vote,Fault,Foot,Fought,Fot Thought,Faught,Vault,Caught Caugh,Vought
Fame-Face-Feign	Fame	Feign,Fein,Fain,Vain,Vein Fin,Fiend,Feeind,Thin
Mail-Deal-Nail	Main,Male,Meal,Mayo	Nail,Kneel,Neil,Neal
Mat-Dead-Gnat	Mat	Gnat,Nat,Knat
Met-Gat-Net	Met	Net
Mock-Dock-Knock	Mock,Muck	Knock,Nock,Hock
Biege-Gaze-Daige	Beige,Beidge,Beege,Bij Peege	Daige,Deige,Dij,Dage,Dej Age,Stage,Fish,Eige,Eege Vij,Thage,Veige,These, Theign,Vis,Beign

Figure 12.2 gives the total number of "McGurk" and "Audio" responses, using the *Any Audio - Any McGurk* coding format, given by participants under all conditions (i.e. all video sizes, real and synthetic videos). The results show large variabilities in participant responses, e.g. participants 4, 7, 18, and 19 showed relatively few McGurk perceptions, while participants 6 and 16 tended to favour McGurk responses over audio. Not surprisingly the same results, coded using the *Expected Audio - Expected McGurk - Other* format (Figure 12.3), show a change in the number of McGurk responses given by participants. This shows the variability which occurs when *strictly* enforcing previously recorded McGurk responses. As previously mentioned, this variability is to be expected across McGurk experiments using clips built from different people. Therefore, it is sensible to base the overall interpretation of the results more on evidence from the *Any Audio - Any McGurk* coding format.

Figure 12.4 shows the mean number of "McGurk" responses, for real and synthetic videos, under each video size, and coded using the *Any Audio - Any McGurk* format. It clearly shows that the number of "McGurk" responses increased with video size using real video, and stayed fairly constant using synthetic video. The graph also indicates that more McGurk responses were

Table 12.3: Expected Audio - Expected McGurk - Other

Tuple	Accepted Audio	Accepted McGurk	Other
Bat-Vet-Vat	Bat	Vat, Fat	
Bent-Vest-Vent	Bent,Bint	Vent,Fent,Fint	
Bet-Vat-Vet	Bet	Vet,Fet	
Boat-Vow-Vote	Boat	Vote	Bolt,Bought,But, Boot,Booked,Port, Fault,Foot,Fought, Fot,Thought,Faught, Vault,Caught,Caugh
Fame-Face-Feign	Fame	Feign,Fein,Fain Vain,Vein	Fin,Fiend,Thin
Main-Deal-Nail	Main,Male	Nail	Meal,Mayo,Kneel, Neil, Neal
Mat-Dead-Gnat	Mat	Gnat,Nat,Knat	
Met-Gal-Net	Met	Net	
Mock-Dock-Knock	Mock	Knock,Nock	Muck,Hock
Beige-Gaze-Daige	Beige, Beege Beidge	Daige,Deige,Dij, Dage,Dej	Bij,Peege,Age Stage,Fish,Eige Eege,Vij,Thage, Veige,These, Theign,Vis,Beign

recorded using the real video clips. Under *t*-test conditions we see that effect of video type (i.e. real or synthetic) was significant ($F(1, 19) = 315.81$, $p < .01$). We also see that the main effect of video size was significant ($F(2, 38) = 75.48$, $p < .01$), with more McGurk responses in the medium and large conditions than the small condition. The interaction between video type and size was also found to be significant ($F(2, 38) = 44.05$, $p < .01$). Figure 12.5 repeats these observations, showing the mean number of "McGurk", "Audio" and "Other" responses, for real and synthetic videos, under each video size, and coded with the *Expected Audio - Expected McGurk - Other* format.

In terms of directions for further development of talking heads, or any other facial animation system, the most useful perspective on the results comes from the number of McGurk effects reported for each real tuple versus effects for synthetic tuples. Figure 12.6 gives the normalized total number of McGurk and Audio responses for each synthetic tuple under large video conditions, while Figure 12.7 shows the same information for real tuples. Both figures use the *Any Audio - Any McGurk* coding format. The small and medium video results are omitted from these figures since

the goal is to analyze the McGurk responses under the best viewing conditions (see Figures 12.4 and 12.5). The figures confirm the observation that the McGurk effect is stronger in the real videos than in the synthetic ones. Using the real videos as a baseline, a bias is also seen in the number of McGurk responses recorded for certain tuples. This is helpful for future experiments as it allows the identification of weak tuples which can be removed in future tests.

Concerning the end of test questions the following feedback was received: In response to Question 1 most of the participants noticed that the audio did not always match the video, as would be expected given the construction of the McGurk tuples. In response to Question 2 none of the participants noticed that any of the clips were computer generated, although one participant did comment that he thought some of the clips appeared somehow "unnatural". Concerning the use of the replay button (Question 3) most participants chose not to use it, with only a handful opting to use it once, and a single participant using it twice.

The next Section discusses how the results may be interpreted to identify possible strengths and weakness in synthetic videos, and how these may be exploited to direct further development of the talking head synthesis algorithms.

12.5 Discussion

In terms of overall performance the real video clips produced more McGurk effects from the participants than the synthetic ones. Thus, the synthetic animations are not as convincing as the real video clips. Based on the assumption that production of a McGurk effect implies good lip-synch from the animation algorithm, this points to some lip-synching weakness in the talking head. Figures 12.6 and 12.7 are considered in an attempt to identify these weaknesses. The tuple *Beige-Gaze-Daige* performed poorly in the real video and synthetic video trails, with over twice as many "Audio" responses than "McGurk" responses for both video types. This suggests an inherently weak McGurk response for that tuple. The same weakness would also appear to apply to the tuple *Mock-Dock-Knock*, with half the overall responses being "McGurk" and the other half "Audio". These McGurk effect weaknesses are most likely due to the accent of the participant used to create the tuples, and makes the performance of the synthetic versions of these tuples, in comparison to the real tuples, difficult to correctly interpret.

The other real video tuples scored sufficiently high enough to warrant further analysis. The only

two synthetic tuples to generate a higher number "McGurk" responses than "Audio" ones were *Mat-Dead-Gnat* and *Fame-Face-Feign*. This suggests satisfactory lip-synch generation for the visemes /D/ and /F/, as well as good articulation throughout the rest of the words (these being *Dead* and *Face* respectively). The high scoring of the real videos of these tuples point to the conclusion that this observation is valid. The content, and poor synthetic video performance, of the remaining synthetic tuples suggests that the animation algorithm currently has difficulty in generating lip-synching for the viseme /V/, as seen in the words *Vet, Vest, Vat and Vow*. To overcome this it is believed that modification of the training set, to include exaggerated articulation of certain words and consonants, may sensitise the HMCM to the presence of these sounds in new talking head input speech signals. Another possibility may be to increase the sampling rate of the speech analysis, in an attempt to better capture short-term speech sounds.

A deeper analysis of the results suggests that the HMCM is successfully modelling and then synthesizing visemes such as /S/, /A/ and /E/, these being present in the high scoring synthetic tuples.

Feedback from the questions posed to the participants was particularly encouraging. Out of the 20 participants none stated that they thought any of the clips were computer generated, with only one participant mentioning that "some of the clips seemed somehow unnatural". This points to an overall realistic output, and realistic behavior in the talking head. It also helps support the hypothesis that given no prior, or at best an undeveloped one, a person is less likely to notice a synthetic talking head animation over a real one.

The increase and subsequent plateau in the mean number of McGurk responses to the real video tuples under varying sizes (Figure 12.4) suggests that intelligibility of the clips degrades between video resolutions of approximately 361x289 pixels and 72x75 pixels, and reaches an optimal level at a resolution between 361x289 pixels and 720x576 pixels - suggesting that an increase in resolution at this point does not contribute to talking head intelligibility. The effect however is less dramatic for the synthetic tuples, and is likely due to the generally low number McGurk responses given in relation to the synthetic clips.

Figure 12.1: Example preparation of a synthetic McGurk tuple: Audio of the word "Mat" is recorded (Top Box). Video and Audio of the word "Dead" is recorded (Middle Box). Audio for the word "Mat" is dubbed onto video for the word "Dead", producing the McGurk effect "Gnat" (Bottom Box).

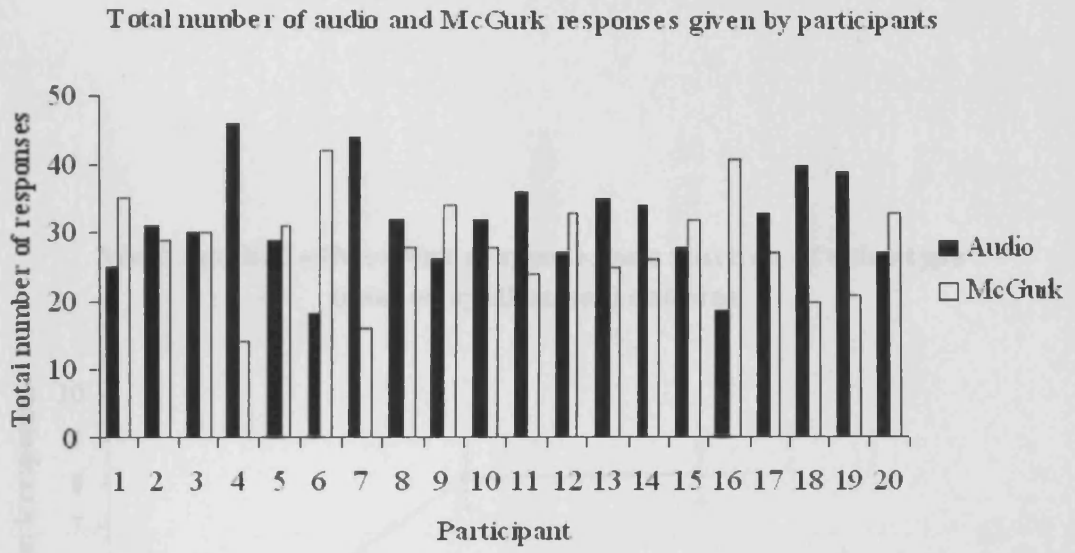

Figure 12.2: Total Number of McGurk and Audio responses given by participants under all conditions, and using the *Any Audio - Any McGurk* coding format.

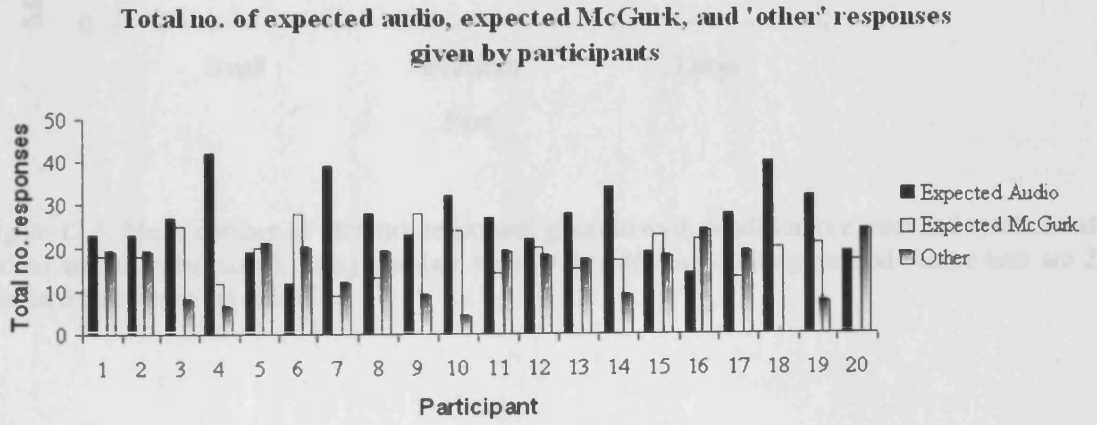

Figure 12.3: Total Number of McGurk, Audio and Other responses given by participants under all conditions, and using the *Expected Audio - Expected McGurk - Other* coding format.

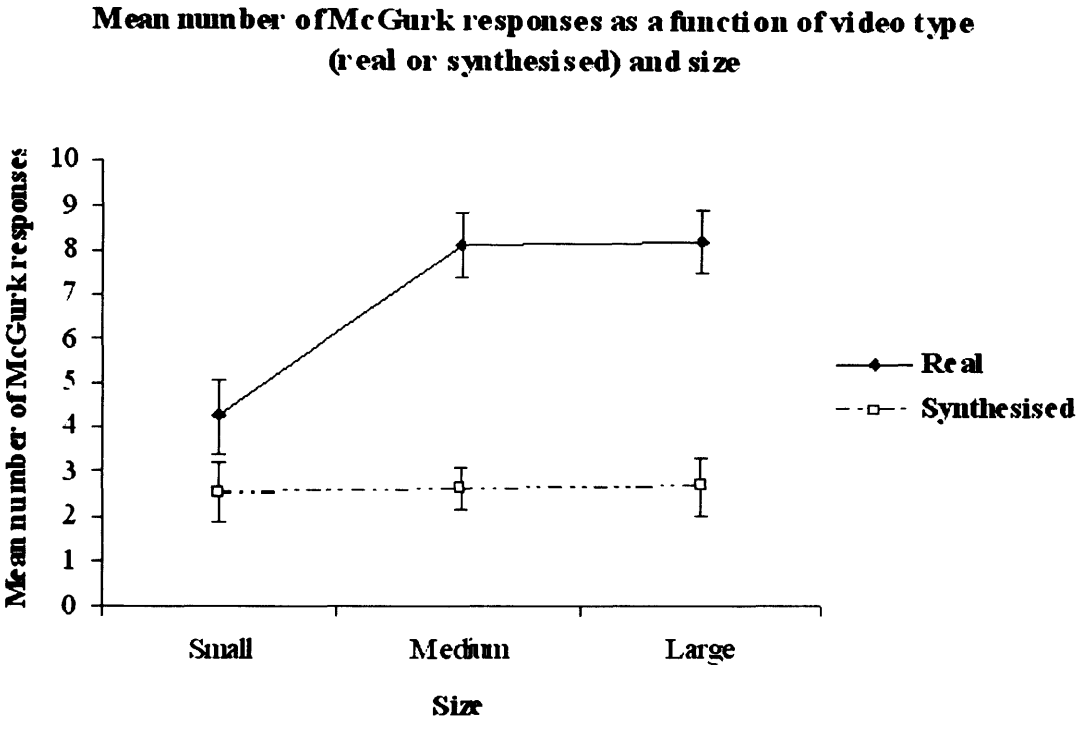

Figure 12.4: Mean number of McGurk responses given in each condition (i.e. real and synthesized videos and all video sizes), using the *Any Audio - Any McGurk* coding method. Error bars are 2 standard error from the mean.

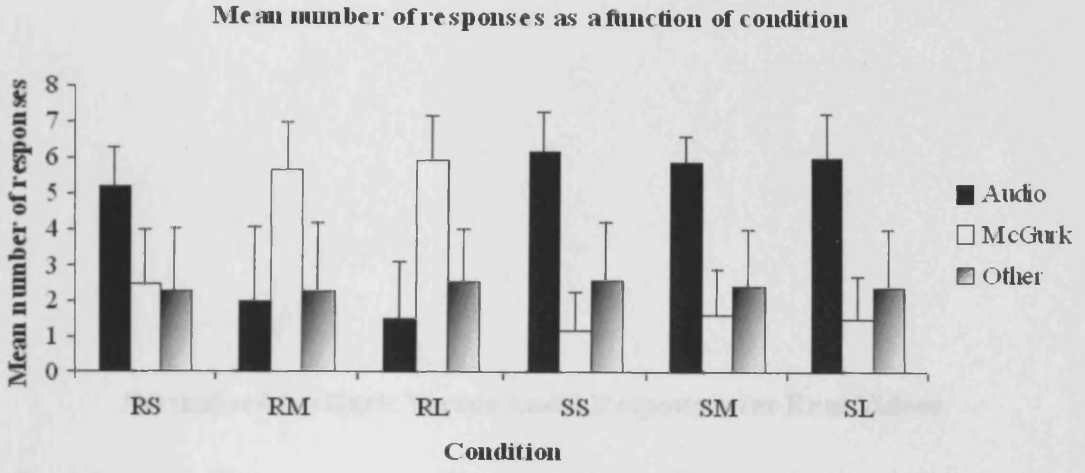

Figure 12.5: Mean number of responses for real and synthetic videos, under different video size, using the *Expected Audio - Expected McGurk - Other* coding format. *RS, RM and RL* relate to small, medium and large sized real videos respectively, while *SS, SM and SL* relate to small, medium and large sized synthetic videos respectively. Error bars are +1 standard error from the mean.

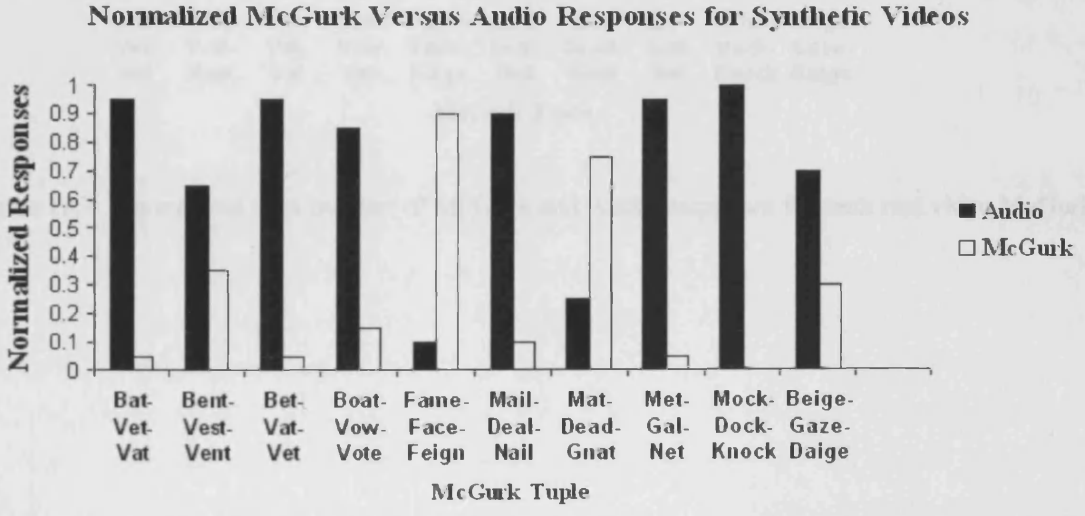

Figure 12.6: Normalized total number of McGurk and Audio responses for each synthetic video McGurk tuple.

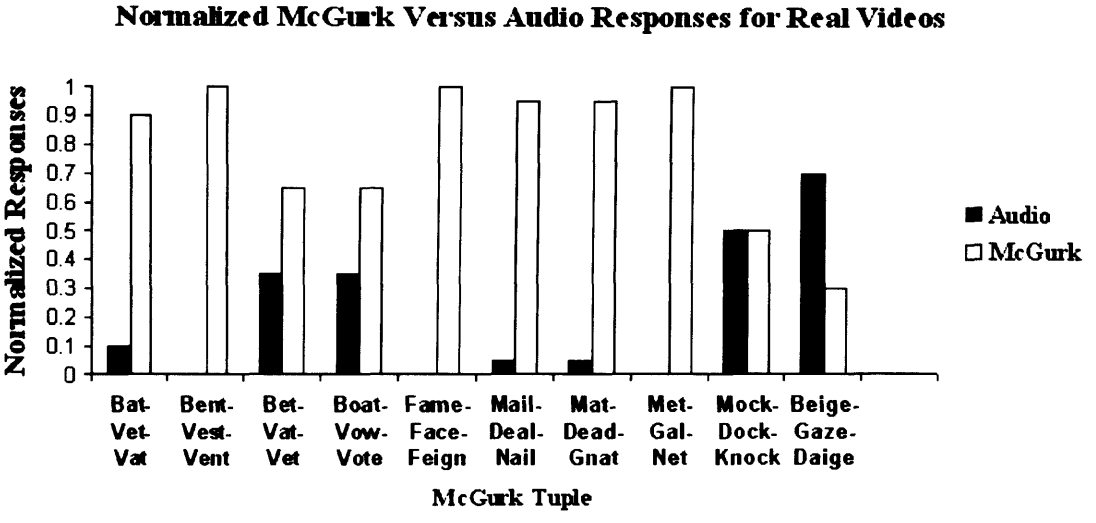

Figure 12.7: Normalized total number of McGurk and Audio responses for each real video McGurk tuple.

Chapter 13

Future Work

This Chapter considers directions for future research in light of suggested improvements noted in the main body of the thesis, and the numerous ideas that have resulted from this study.

13.1 Further HMCM Applications

In this thesis, a HMCM is used to animate the mouth from speech. Speech parameters and animation parameters for the mouth are essentially correlated signals, and the HMCM may be regarded as a *black box* for signal estimation. As such, it is also suitable for the estimation of other types of signal, and subsequently may be employed for the animation of other sub-facial areas – as opposed to solely the mouth – from speech

Animation of the eyebrows from speech has already been attempted. A HMCM was constructed using eyebrow and speech data from the hierarchical model 1 training corpus. Given new speech observations, left and right eyebrow trajectories and corresponding synthetic images were then generated and compared against ground truths. Figure 13.1 shows ground truth (top plot) left (blue line) and right (red line) eyebrow trajectories versus corresponding synthetic trajectories (bottom plot).

From the Figures it is clear that the synthetic trajectories do not correspond with the ground truth trajectories. Also note the correlation between the left and right eyebrow ground truth trajectories, and the disjoint behaviour between the left and right eyebrow trajectories. For training the left and right eyebrow HMCMs, mel-cepstral speech feature were used in this example. The lack of correspondence between the synthetic and ground truth trajectories is therefore perhaps not surprising, since mel-cepstral features are typically associated with voice spectral energy, and there is no direct

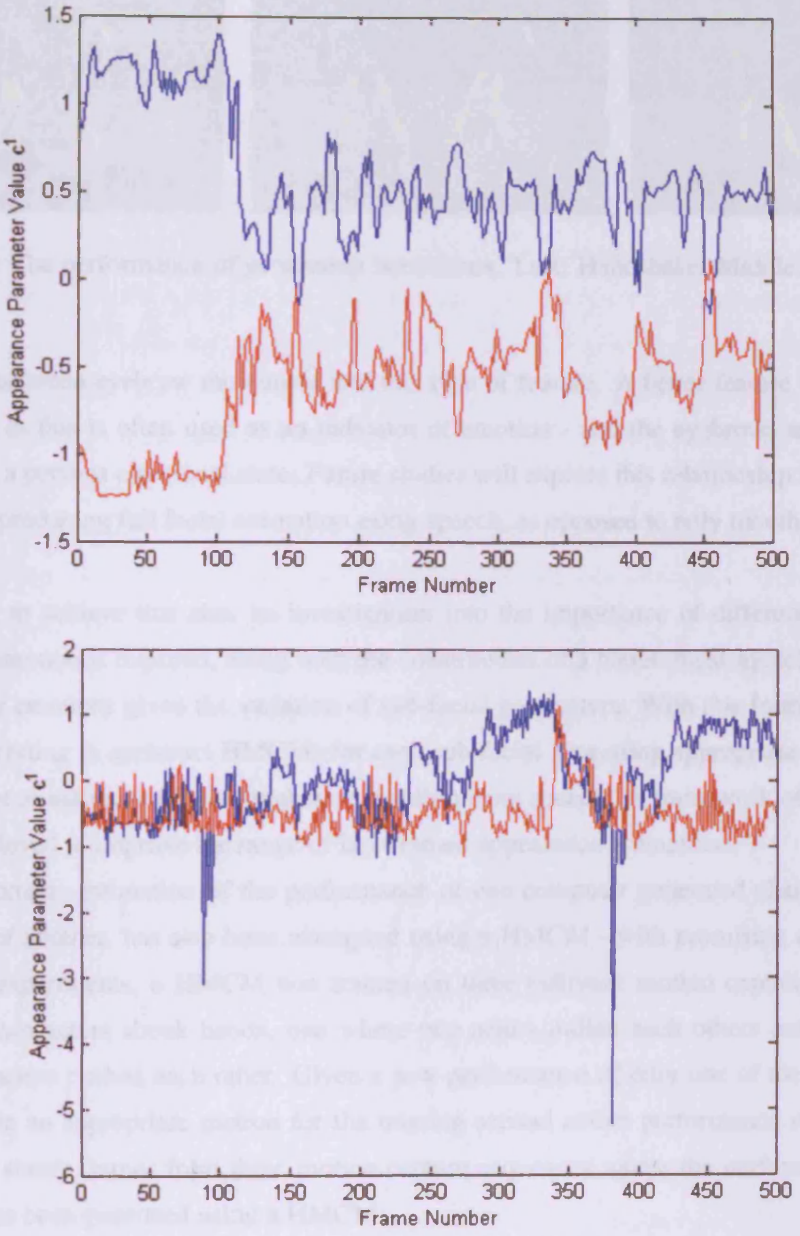

Figure 13.1: Ground truth (top plot) and synthetic (bottom plot) left (blue line) and right (red line) eye brow trajectories. The synthetic trajectories were generated using separate left and right eye-brow HMCMs, trained using the hierarchical model 2 corpus.

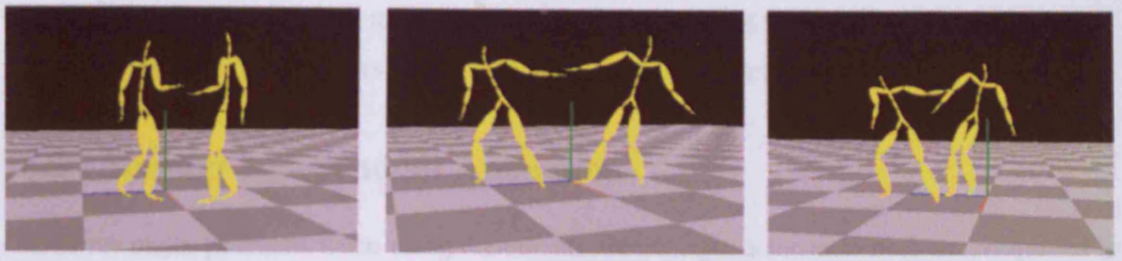

Figure 13.2: The performance of generating behaviours. Left: Handshake; Middle: Pulling; Right: Pushing.

correlation between eyebrow movement and this type of feature. A better feature to use would be voice pitch, as this is often used as an indicator of emotion - and the eyebrows are often used to help convey a persons emotional state. Future studies will explore this relationship further, with the final aim of producing full facial animation using speech, as opposed to only mouth animation from speech.

In order to achieve this aim, an investigation into the importance of different facial areas in expressing emotion is required, along with the construction of a hierarchical model capable of emulating these emotions given the variation of sub-facial parameters. With this framework, it would be very interesting to construct HMCMs for each sub-facial area using appropriate speech features in an attempt to automatically re-synthesise emotion from speech. A framework of this kind could also be employed to improve the range of key-framed appearance animations.

The automatic estimation of the performance of one computer generated character, given the interaction of another, has also been attempted using a HMCM - with promising early results. In a series of experiments, a HMCM was trained on three different motion capture performances: one where two actors shook hands, one where two actors pulled each others arms, and another where two actors pushed each other. Given a new performance of only one of the actors, the aim is to generate an appropriate motion for the missing second actors performance using a HMCM. Figure 13.2 shows frames from three motion capture sequences where the performance of one of the actors has been generated using a HMCM.

A technique such as this can greatly aid an animator when designing complex scenes where the performance of many computer generated characters - all interacting together - is required. For example, the performances of a few key-characters could be used to automatically animate the performances of all the *extras*. Similar work has already been demonstrated successfully in the

motion picture "The Lord of the Rings" through implementation of Weta's *MASSIVE* program [30]. Animation for large crowd scenes is also a popular line of study in this area [153]

13.2 Speaker Independent Animation

One of the major problems when using continuous speech signals for tasks such as recognition is that of speaker normalisation, especially given independent speaker signals. Speaker normalisation is one of the key research areas in speech recognition, and the quality of animation produced from a SAM or a HMCM, given an independent speaker signal, depends largely on the normalisation techniques employed. Unfortunately, a study into speaker normalisation as part of this thesis is beyond its scope This is an area in animation where phoneme driven methods have an advantage - as they are by their nature speaker independent.

In the context of SAMs and HMCMs, if the continuous speech distributions for two different speakers overlap substantially, then good animations can still be obtained. However, given too great a difference between the distributions animations can often be poor [43]. Figure 13.3 shows the speech distribution from hierarchical model 2 with respect to the distribution from a different speaker, where audio was captured under the same recording conditions. Note that the speaker recorded for hierarchical model 2 is male and has a Scottish accent, while the new speaker is female and has a Welsh accent. In the Figure, speech is represented by parameters corresponding to the two highest modes of speech variation.

In this case the two distributions have strong similarities. The means of both distributions are zero, and the standard deviations are approximately 18 and 9 for each respective speech parameter. The new speaker was recorded reciting the alphabet, and Figure 13.4 shows some example frames. Of the letters synthesised, productions for *B, E, G, H, I J, L, N, O, Q, R, T, U, V, X, Y* and *Z* were satisfactory, while productions for *A, C, K, M, P, W* and *S* were slightly weaker.

The results are quite encouraging for this particular speaker, although strong normalisation between two speech distributions does not always exist. One difficulty with evaluating independent speaker animations using continuous speech is the difficulty in obtaining a suitable ground truth. For example, if recitation of the alphabet does not exist in the hierarchical model 2 corpus then there is no basis for comparison. Even if such an utterance did exist, it would be very unlikely that pronunciation lengths for individual letters, and articulation timings would correspond - making comparison more difficult still. Accent effects might also make comparisons more difficult,

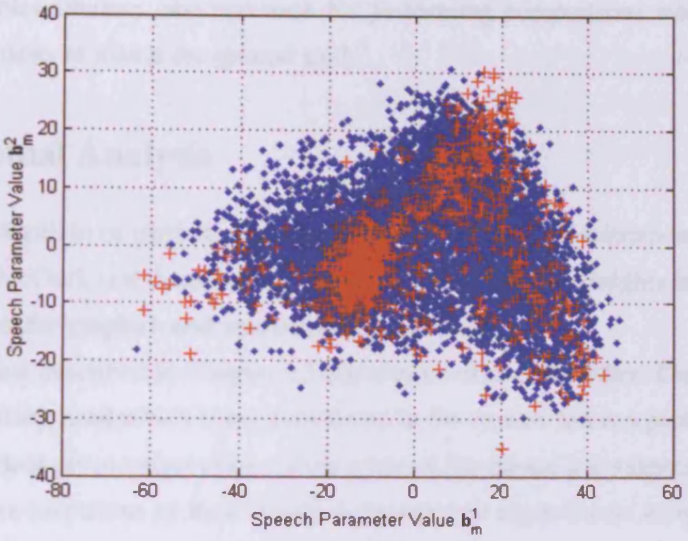

Figure 13.3: Speech Distributions for two speakers represented by parameters for the two highest modes of speech variation. The speech distribution for the hierarchical model 2 corpus is represented using blue dots, while the speech distribution for the second speaker is represented using red crosses.

Figure 13.4: Independent speaker synthesis of the letters "H" (top row, frames 256 – 281) and "L" (bottom row, frames 338 – 350) using a mouth HMCM trained using the hierarchical model 2 corpus.

since the pronunciations of certain sounds can differ, causing a significant difference in visual feature trajectories. Nevertheless, one approach for performing comparisons would be to time-warp synthesised animations to match the ground truth.

13.3 Perceptual Analysis

There is much work still to be carried out in the perceptual analysis of animation. Subtle and indirect tests, such as the McGurk test described in Chapter 12, can provide insights into where directions for the development for graphics and animation algorithms lie.

The McGurk test described in Chapter 12 can also be extended further. One matter concerning the hierarchical talking head which is not considered in the current test is a participant's opinion towards synthetic videos given longer clips. Given a test of this nature it is suspected that a participant might become more suspicious of their origin, as reported in experiments using a different talking head in [62]. In order to perform a test of this nature, McGurk *sentences* could be employed, one such example being the audio "My bab pop me poo brive", dubbed onto the video "My gag kok me koo grive", with the expected McGurk effect of perceiving "My dad taught me too drive". Another alternative is to simply create synthetic clips of the talking head speaking a long list of single words (with corresponding real video clips as a baseline).

An alternative to using the McGurk effect as a perceptual test in the manner proposed might also be to simply play real and synthetic clips with the *correct* dubbing, and to ask participants what words they hear. Given incorrect dubbing, McGurk responses would be expected from the participants. The attractive aspect of this approach is that the participants are not informed that some clips are real and some synthetic, again reducing their development of a prior.

Given a fully developed McGurk talking head test one thing that is not expected is that it could be modified to test the synthesis of facial emotion and expression. However, one approach to achieve this kind of testing, from a perceptual point of view, is to play random real and synthetic clips and ask a participant what emotion they are witnessing [85]. An attractive quality of this type of test is that it reduces prior opinion from the participants. A test of this nature also addresses strengths and weaknesses in the emotion algorithm since it allows a developer to identify which synthetic emotions are confused with real ones (if any), in order to direct the algorithm in rectifying these ambiguities.

One final point which should be made concerns the selection of McGurk tuples, which does

not yet include tests for every type of Viseme. Performing a test of this nature would require the investigation and construction of new McGurk tuples - since the tuples described in perception literature do not cover all Visemes. The work of Dodd [50] would be particularly useful in such a study, as it attempts to identify combinations of individual *sounds* which produce McGurk effects. Words which incorporate these sounds could be carefully chosen to achieve a full Viseme test set.

13.4 Extending the Hierarchy

The configuration of the hierarchical model described in this thesis is by no means set in stone, and could easily be extended to enable the creation of new and more detailed facial poses. In order to achieve this, extra nodes could be added for different facial areas (e.g. for the forehead), or existing nodes could be decomposed further.

For example, eye-nodes may be decomposed to include child nodes for creating eye-corner wrinkles (or crows-feet). Of course, the appearance of such facial features is often dependent on the configuration of the rest of the face. For example, eye-wrinkles usually appear when the cheeks are raised, or the eyebrows are lowered. This leads to the possibility of also including nodes for the left and right cheeks.

A discussion such as this raises an entirely new concept - that of building dependencies between the nodes of the hierarchy. For example, if the eyebrows are raised, then a forehead node should automatically produce wrinkles. Similarly, if the forehead is wrinkled, then the eyebrows should automatically raise. The same relationship would also have to exist between nodes for eye-corners and nodes for cheeks. One such dependency between the eyebrows and the eyes has already been noted in Chapter 10. However, this dependency seems to only exist on certain occasions, e.g. when the participant *chooses* to raise the eyebrows and widen the eyes at the same time. The dependency does not always exist when just the eyebrows are raised. Therefore, the application of dependencies should be an optional feature to be employed at the animators discretion.

It has already been seen that the complexity of the modes of variation extracted for the mouth and lower-face nodes make them unsuitable for keyframe animation. This is because it is difficult to easily construct a desired poses by manually combining these mode parameter values. Linear interpolation of these parameter for creating in-between frames is also not suitable, and leads to illegal poses. However, other mouth and lower-face behaviors may be modeled apart from those related to lip-synching. Work has already been carried out for animating *smiles* using lower-face

modes of variation [90, 92, 91]. Given a training video of different smile examples, the modes of variation of a lower face appearance model will naturally extract smile variation, from a closed mouth to a full smile. By varying the value of the smile parameter, different intensity smiles can be produced. Smile characteristics such as onset, apex and offset length can also be controlled by keyframing the smile parameter. Figure 13.5 demonstrates example frames from a smile animation as well as the smile parameter trajectory used to generate it.

13.5 A 3D Hierarchical Model

The current hierarchical model is constructed using 2D video image data, and as such has certain limitations. During training, a subject is required to minimise out-of-plane head movement, and to remain a fixed distance from the camera. If these directions are not followed, non-linearities may be introduced into the model, and animation quality may suffer. For example, if a subject performs an out-of-plane head movement, then mouth images captured during this phase will be skewed. Normalisation of these mouth images is performed without out-of-plane considerations (i.e. it is assumed that images are subject only to translation, rotation and translation), therefore out-of-plane distortions are not corrected. The HMCM assumes all images are front-on, and will therefore regenerate skewed mouth images during synthesis.

Since front-on images are required in order to robustly train the model, output animations are restricted in that they cannot synthesise out-of-plane movements, or cannot be satisfactorily re-inserted into a background video where the head is out-of-plane. This is obviously a great limitation, as it restricts our model to *news-reader* like applications. It would be desirable to allow animations to be re-inserted into any kind of background video, with no constraints on head pose. This would make it suitable for applications such as film and T.V. re-dubbing.

The solution to these problems is to build a hierarchical model in 3D. In order to achieve this, the capture of a set of 3D facial scans exhibiting a wide range of facial poses would be necessary. Such a model could then be key-frame animated and effectively projected into any background footage. To train a HMCM in 3D, a real-time 3D camera would be required in order to capture short (20-40ms) audio-video correspondences. Luckily, such cameras now exist [2], allowing the 2D speech driven ideas proposed in this thesis to be applied straightforwardly in 3D. Figure 13.6 demonstrates a 3D scan of a persons head using a 3DMD real time passive stereo 3D camera.

Figure 13.5: A synthetically generated smile using a lower-face node. The trajectory shows the path of the lower-face parameter - or smile parameter. The red marker on the trajectory shows the value of the parameter used to generate the corresponding output image. From the example, it can be seen that different smiles can easily be generated by varying onset, apex and offset lengths.

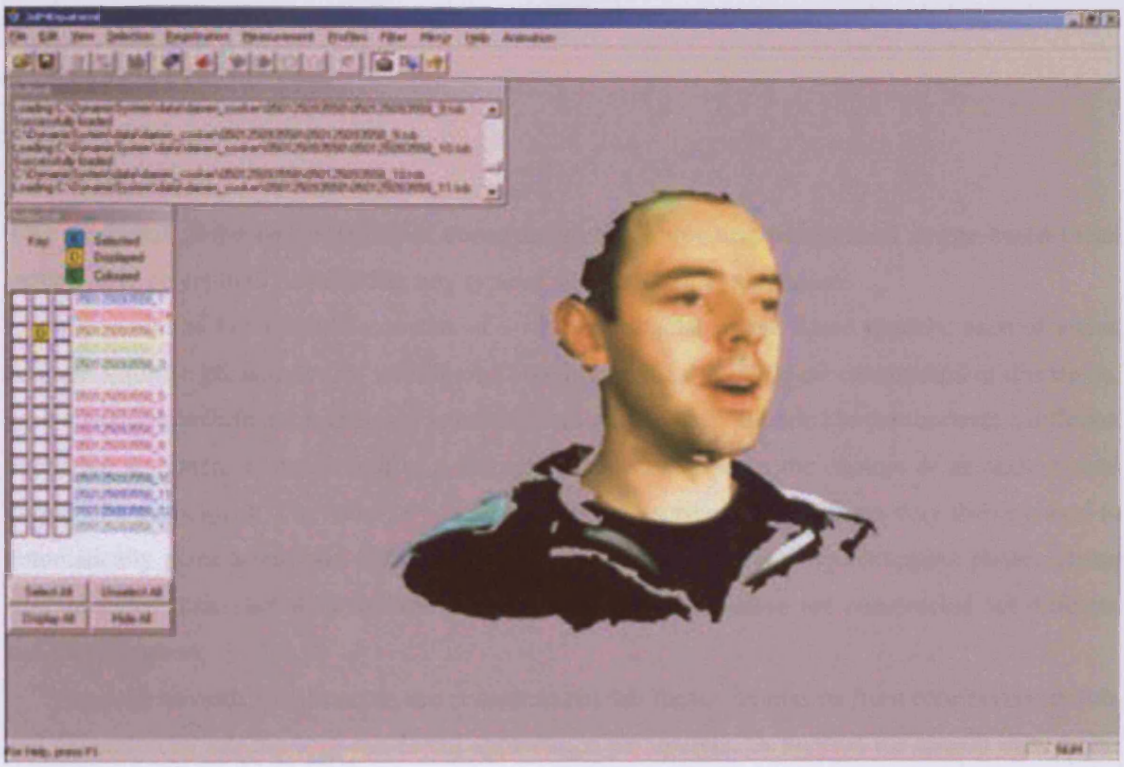

Figure 13.6: A 3D scan taken using a real time passive stereo camera. Using this technology, a 3D hierarchical model can be constructed and animated using a HMCM.

Chapter 14

Conclusions

This thesis has proposed methods for constructing and animating hierarchical image based facial models, and perceptually evaluating any type of synthetic facial animation.

A hierarchical facial model consists of a set of sub-facial image based models, each of which contains intuitive parameters for animation. Two hierarchical models are constructed in this thesis, each containing a different number of sub-facial *nodes*, and each intended to demonstrate a different animation approach. Construction of a hierarchical model requires the capture of an audio-visual corpus of a participant. The robust tracking algorithm described in this thesis may then by used to automatically place landmarks across the visual data set after a short bootstrapping phase. Using this annotated data, statistical models of shape and texture variation are constructed for different sub-facial regions.

Two such animation approaches are considered in this thesis: animation from continuous speech, and animation by *key-framing* sub-facial appearance parameters. A method for speech driven animation is presented through the proposal of two different models: SAMs and HMCMs. Both models synthesise mouth animation parameters given speech parameters. The HMCM approach builds on the SAM approach by encoding coarticulation, and thus improves on synthesis. Animations may be constructed from speech examples not originally present in the training set, and from non-verbal articulations, making it useful for a wide range of applications. Input speech is processed using known techniques to achieve robust and uncorrelated features.

Since the hierarchical model concentrates on building statistical models for individual facial areas, these models produce intuitive animation parameters. The alternative animation approach

described in this thesis works by key-framing these sub-facial parameters. Using this technique, a variety of expressive animations may be constructed.

The hierarchical model has also been shown to be useful in facial behaviour analysis. Since sub-facial models are associated with their own descriptive parameters, analysis of the training set provides interesting insights into the behaviour and interaction of these parameters during the production of certain facial actions.

In order to assess the animation methods described, a thorough analytical evaluation is performed of HMCMs. A perceptual perspective on performance is then measured by applying a novel McGurk test to synthetic animations. This test is general in that it may be applied to the evaluation of any synthetic facial animation, irrespective of how the animation was produced.

To conclude this thesis, its major contributions are again reiterated with respect to their assertion in the main body of the report..

- An image-based hierarchical facial model for animation (Section 4.6). The model improves on previous image-based models in the level of control it offers the animator (Section 10.2). The model also offers a greater degree of specificity in terms of statistical modeling, since sub-facial areas are modeled individually as opposed to being encoded as part of a larger model (e.g. one of the whole face - see Section 11.3).

- Realistic animation of an image based model using continuous speech signals, including automatic animation for lip-synching, natural pauses, hesitations and *non-verbal* articulations (Section 11.2). These features are facilitated using two novel analysis and synthesis techniques: SAMs (Chapters 4 and 5) and HMCMs (Chapters 7 and 8). Both models produce accurate animations given relatively little training (Chapter 11).

- A semi-automatic technique - akin to keyframing - for producing animations using image-based models (Section 10.2). The method differs from a classical key-frame approach in that key-moments are not defined using images. Instead, key-moments are defined using appearance parameter values for different sub-facial areas, and *in-betweening* carried out by interpolating these parameters. The hierarchical decomposition of the face provides intuitive animation parameters for different sub-facial areas for the facilitation of this technique.

- A powerful tool for performing facial analysis (Section 10.1). The hierarchical model, when applied to the analysis of a face, yields very insightful information regarding facial dynamics,

and the interaction between different facial areas. This information is particularly interesting not only from an animation point of view, but also from a perceptual one. With respect to animation, knowledge of real facial dynamics - such as onset and offset lengths of behaviours such as blinks - can improve the realism of synthetic animations. This information can be applied to the animation of any model, whether it is intended to be video realistic or not. From a perceptual point of view, the behaviour of individual facial areas during expressions can be examined, and used as the basis of recognition and classification (Section 13.4).

- A novel perceptual evaluation technique based on the McGurk effect for determining the effectiveness of visual-speech synthesis in synthetic facial animation (Chapter 12). The approach identifies strengths and weaknesses in synthetic animation algorithms, thus guiding further development. Also, the approach is stand-alone, and may be applied to the evaluation of any facial animation system.

- An automatic landmark placement algorithm for facial feature tracking and subsequent construction of Appearance Models (Section 4.3.2). The algorithm is based on Downhill-Simplex Minimisation (DSM), and incorporates strategies to optimise the search and avoid local minima based on prior knowledge.

The hierarchical model has also been shown to be useful in perceptual studies, such as the analysis and synthesis of fake and genuine smiles (Section 13.4). The HMCM also shows promise in performing tasks other than mouth animation, such as the animation of other facial areas, and the automatic creation of a synthetic performance given a real input one (Section 13.1).

The obvious next step is to apply the techniques described in this thesis to a 3D model, to further extend the hierarchy, and to encode dependencies between sub-facial areas.

Appendix A

PCA Memory Issues

This Appendix describes the various memory issues relevant when building the PCA models required for a hierarchical model, and gives solutions.

A.1 Memory Issues and Colour Encoding

Given a set of defined sub-facial regions, appearance models are constructed for each facial area in each hierarchical model. However several issues arise when building these models due to their size, and the consequent effect on computer memory. For example, the face node training set in hierarchical model 2 consists of 8295 texture vectors of dimension 13038 by 1. Performing standard SVD on this data set in order to build a GLM is extremely costly in terms of memory, and is not practical on some slower computers. Another contributor to the processing cost of building this model is the addition of colour information into the appearance model.

Colour is typically incorporated into appearance models by defining texture vectors as concatenated Red, Green and Blue (RGB) signal vectors [145]. The appearance model therefore explicitly models colour, and variation of the appearance parameter yields a new vector from which RGB signals can be extracted for reconstruction. Following this strategy increases the size of the texture data set in model 2 to 39114 by 8295, i.e. 8295 RGB vectors of dimension 39114. This provides an even worse computer memory problem.

The memory efficient solution given in this thesis incorporates the use of Turk and Pentland's fast PCA method [152] and Hall *et als* [73] eigenspace addition algorithm. Colour reconstruction is then handled using a look-up table, as described in Chapter 4.

A.1.1 Fast, Memory Efficient PCA

Given a training set of dimension $N \times D$, where D is the number of pixels in an image and N is the number of images, the covariance matrix is of size D^2. Performing SVD on a matrix of this size is often an intensive task, e.g. in the case of a facial training set where each image may contain a large number of pixels. Fortunately it is often true that $N < D$, in which case the problem can be reduced to performing SVD on a matrix of size N^2. This is the method proposed by Turk and Pentland in [152], and its application is demonstrated with a set of image vectors.

Let $\mathbf{G} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N]$ be the training set of image textures with the mean image $\bar{\mathbf{g}}$ subtracted from each vector. The covariance matrix may now be written as $\mathbf{S} = \mathbf{G}\mathbf{G}^T$. Now consider the eigenvectors $\mathbf{V}_t$ of $\mathbf{G}^T\mathbf{G}$ such that

$$\mathbf{G}^T\mathbf{G}\mathbf{V}_t = \lambda_t \mathbf{V}_t \tag{A.1}$$

By multiplying both sides by $\mathbf{G}$ we have

$$\mathbf{G}\mathbf{G}^T\mathbf{G}\mathbf{V}_t = \lambda_t \mathbf{G}\mathbf{V}_t \tag{A.2}$$

from which we see that $\mathbf{G}\mathbf{V}_t$ are the eigenvectors of $\mathbf{S} = \mathbf{G}\mathbf{G}^T$. Following this the N by N matrix $\mathbf{L} = \mathbf{G}^T\mathbf{G}$ may be constructed and the eigenvectors $\mathbf{V}_t$ calculated from this. The matrix $\mathbf{L}$ is then used to calculate the final eigenvectors

$$\mathbf{P}_t = \sum_{k=1}^{N} \mathbf{V}_{lk}\mathbf{G}_k, \qquad l = 1, \ldots, N \tag{A.3}$$

This method is efficient if the number of samples is less than the number of dimensions in each image (i.e. if $N < D$). However, given a suitably large value for N the calculation may still be memory intensive. Thus the eigenspace addition method of Hall *et al* [73] is employed along with Turk and Pentland's method.

A.1.2 Adding Eigen-Spaces

The premise behind this method is that eigenmodels may be added together. Thus, given memory constraints in a system, construction and subsequent addition of a number of smaller PCA models is more efficient than construction of one large model. The problem is defined thus: Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ be collections of N data points, each D dimensional. Their respective eigenmodels are then defined as

$$\Omega(\mathbf{X}) = (\mu(\mathbf{X}), U(\mathbf{X})_{mp}, \Lambda(\mathbf{X}), N(\mathbf{X})) \tag{A.4}$$

$$\Omega(\mathbf{Y}) = (\mu(\mathbf{Y}), U(\mathbf{Y})_{mq}, \Lambda(\mathbf{Y}), N(\mathbf{Y})) \tag{A.5}$$

where p and q are the number of eigenvalues in each respective model, μ is the mean vector in each distribution, U are the eigenvectors of each distribution, Λ are the eigenvalues of each distribution and N is the number of data points in each distribution. The problem is to compute the new model

$$\Omega(\mathbf{Z}) = (\mu(\mathbf{Z}), U(\mathbf{Z})_{nr}, \Lambda(\mathbf{Z}), N(\mathbf{Z})) \tag{A.6}$$

$$= \Omega(\mathbf{X}) \bigoplus \Omega(\mathbf{Y}) \tag{A.7}$$

with reference only to $\Omega(\mathbf{X})$ and $\Omega(\mathbf{Y})$, i.e. with no reference to the original training data. The only constraint on the process is that D is similar for $\mathbf{X}$ and $\mathbf{Y}$. Thus the heart of the problem is to define the operator $\bigoplus$. Estimation of $N(\mathbf{Z})$ and $\mu(\mathbf{Z})$ is trivial

$$N(\mathbf{Z}) = N(\mathbf{X}) + N(\mathbf{Y}) \tag{A.8}$$

$$\mu(\mathbf{Z}) = (N(\mathbf{X})\mu(\mathbf{X}) + N(\mathbf{X})\mu(\mathbf{X}))/N(\mathbf{Z}) \tag{A.9}$$

Estimation of the new eigenvectors involves taking a linear combination of the vectors in the two input models, and complete details are given in [73].

A.1.3 PCA Construction Strategy

Given these two techniques, a formal strategy is now outlined for building all PCA models in this thesis. The process is defined thus

1. Given a vector training set extract 1000 vectors. If $D < 1000$ (where D is the number of elements in a training vector) then calculate the eigenvectors and eigenvalues using standard SVD. If $D >= 1000$ then perform PCA using Turk and Pentlands PCA method.
2. Define this model as $\Omega(\mathbf{X})$
3. Extract 1000 more vectors and repeat step 1 (i.e. perform SVD or efficient PCA based on data dimensions). Define this model as $\Omega(\mathbf{Y})$.
4. Construct $\Omega(\mathbf{Z}) = \Omega(\mathbf{X}) \bigoplus \Omega(\mathbf{Y})$
5. Extract 1000 more vectors (or the maximum available if there are less than 1000 vectors remaining) and repeat step 1 (i.e. perform SVD or efficient PCA based on data dimensions). Define this model as $\Omega(\mathbf{X})$.

6. Update $\Omega(\mathbf{Z}) = \Omega(\mathbf{X}) \bigoplus \Omega(\mathbf{Z})$

7. Repeat steps 5-7 until all vectors in the training set have been processed.

Bibliography

[1] Mpeg working group on visual, international standard on coding of audio-visual objects, part 2(visual). *ISO-14496-2*, 1999.

[2] 3DMD. http://www.3dmd.com/.

[3] J. Ahlberg. Candide-3, an updated parameterised face. *Technical Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linkping University, Sweden.*, 2001.

[4] L. Baum and J. Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *Bull. Amer. Meteorol. Soc*, 73:360–363, 1967.

[5] L. Baum and G. Sell. Growth functions for transformations on manifolds. *Pac. J. Math*, 27(2):211–227, 1968.

[6] J. Beskow. Rule-based visual speech synthesis. In *Proc. of Eurospeech*, pages 299–302, 1995.

[7] A Black and K. Lenzo. Limited domain synthesis. In *Proc. of ICSLP*, volume 2, pages 411–414, 2000.

[8] A Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Proc. of EUROSPEECH*, pages 601–604, 1997.

[9] A. Black, P. Taylor, R. Caley, and R. Clark. Festival tts system. centre for speech technology, university of edinburgh.

[10] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Proc. of EUROGRAPHICS*, 2003.

[11] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of ACM Siggraph*, 1999.

[12] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1063 – 1074, 2003.

[13] C. Bonamico, R. Pockaj, and C. Braccini. A java based mpeg-4 facial animation player. In *Proc. of ICAV3D*, pages 335 – 338, 2001.

[14] F. Bookstein. Principle warps: Thin plate splines and the decomposition of deformations. *IEEE Pattern Analysis and Machine Intelligence*, 11:567–585, 1989.

[15] G. Borshukov, K. Sabourin, M. Suzuki, O. Larsen, T. Mihashi, K. Faiman, S. Schinderman, O. James, and J. Jack. Making of the super punch. In *ACM SIGGRAPH Sketch*, 2004.

[16] A. Bosseler and D. Massaro. Development and evaluation of a computer-animated tutor for vocabulary and language learning for children with autism. *Journal of Autism and Development Disorders*, 2002.

[17] F. Bourel, C. Chibelushi, and A. Low. Robust facial feature tracking. In *Proc. of BMVC*, 2000.

[18] R. Bowden. Learning non-linear models of shape and motion. *PhD Thesis. Dept. Systems Engineering, Brunel University*, 2000.

[19] R. Bowden, T. Mitchell, and M Sarhadi. Cluster based nonlinear principle component analysis. *IEE Electronic Letters*, 33(22):1858–1859, 1997.

[20] M. Brand. An entropic estimator for structure discovery. In *Proc. of Neural Information Processing Systems*, pages 723–729, 1998.

[21] Matthew Brand. Voice puppetry. In *Proc. Computer graphics and interactive techniques*, pages 21–28. ACM Press, 1999.

[22] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: driving visual speech with audio. In *Proc. of 24th conf. on Computer graphics and interactive techniques*, pages 353–360. ACM Press, 1997.

[23] M. Breidt, C. Wallraven, D. Cunningham, and H. Bulthoff. Facial animation based on 3d scans and motion capture. In *SIGGRAPH Sketches and Applications*, 2003.

[24] Warner Bros. Matrix reloaded, 2003.

[25] T. D. Bui, Dirk Heylen, and Anton Nijholt. Improvements on a simple muscle-based face for realistic facial expressions. In *Proc. of 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, 2003.

[26] J. Campbell. Speaker recognition: A tutorial. *Proc. of the IEEE*, 85(9):1437–1462, 1997.

[27] J. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 193 – 206, 2003.

[28] J. J. Choi. *MAYA Character Animation, Second Edition*. SYBEX, 2004.

[29] E. Chuang and C. Bregler. Performance driven facial animation using blendshape interpolation. *Technical report CS-TR-2002-02, Stanford University*, 2002.

[30] New Line Cinema. The lord of the rings: The return of the king, 2003.

[31] M. Cohen and D. Massaro. Modelling coarticulation in synthetic visual speech. In *Proc. of Computer Animation, Geneve, Suisse*, 1993.

[32] M. Cohen, D. Massaro, and R. Clark. Training a talking head. In *Proc. of of the IEEE Fourth International Conference on Multimodal Interfaces*, pages 499–510, 2002.

[33] M. Cohen, R. Walker, and D. Massaro. Perception of synthetic visual speech. *Speechreading by Man and Machine: Models, Systems and Applications, NATO Advanced Study Institute 940584*, 1995.

[34] The Moving Picture Company. Virtual history: The secret plot to kill hitler, 2004.

[35] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[36] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–684, 2001.

[37] T. Cootes and C. Taylor. A mixture model for representing shape variation. In *Proc. of BMVC*, 1999.

[38] T. Cootes and C. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. of SPIE Medical Imaging*, 2001.

[39] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. of British Machine Vision Conference*, pages 9–18, 1992.

[40] E. Cosatto and H. P. Graf. Photo-realistic talking-heads from image samples. *IEEE Transactions on Multimedia*, 2(3):152–163, 2000.

[41] P. Cosi, M. Cohen, and D. Massaro. Baldini: Baldi speaks italian. In *Proc. of 7th International Conference on Spoken Language Processing*, pages 2349–2352, 2002.

[42] D. Cosker, D. Marshall, P. L. Rosin, and Y. A. Hicks. Speech driven facial animation using a hierarchical model. *IEE Vision Image and Signal Processing*, 15(4):314–321, 128-131.

[43] D. Cosker, D. Marshall, P. L. Rosin, and Y. A. Hicks. Speaker-independent speech-driven facial animation using a hierarchical facial model. In *Proc. of IEE Visual Information Engineering*, pages 169–172, 2003.

[44] D. Cosker, D. Marshall, P. L. Rosin, and Y. A. Hicks. Video realistic talking heads using hierarchical non-linear speech-appearance models. In *Proc. of Mirage*, pages 20–27, 2003.

[45] M. Covell. Eigen-points: Control-point location using principle component analysis. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, 1996.

[46] D. Dekle, C. Fowler, and M. Funnell. Audiovisual integration in perception of real words. *Perception and Psychophysics*, 51(4):355–362, 1992.

[47] J. Deller, J. Hansen, and J. Proakis. *Discrete-Time Processing of Speech Signals*. Wiley-IEEE Press, 1999.

[48] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[49] V. Devin and D. Hogg. Reactive memories: An interactive talking head. In *Proc. of British Machine Vision Conference (BMVC)*, 2001.

[50] B. Dodd. The role of vision in the perception of speech. *Perception*, 6:31–40, 1977.

[51] P. Drahos and P. Kapec. Animating human faces using a modified waters muscle model. In *Proc. of CEDCG*, 2004.

[52] Dreamworks. Shrek, 2001.

[53] R. Duda, P. Hart, and D. Stork. *Pattern Classification: Second Edition*. John-Wiley and Sons, 2001.

[54] R. Easton and M. Basala. Perceptual dominance during lipreading. *Perception and Psychophysics*, 32(6):562–570, 1982.

[55] G. Edwards, C. Taylor, and T. Cootes. Face recognition using the active appearance model. In *Proc. of 5th European Conference on Computer Vision (ECCV)*, volume 2, pages 581–695, 1998.

[56] G. Edwards, C. Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In *Proc. of 3rd International Conference on Automatic Face and Gesture Recognition*, pages 260–265, 1998.

[57] P. Eisert, S. Chaudhuri, and B. Girod. Speech driven synthesis of talking head sequences. In *Proc. of 3D Image Analysis and Synthesis*, pages 51–56, 1997.

[58] P. Ekman and W. Friesen. *Facial Action Coding System*. Consulting Psychologist Press, 1977.

[59] R. Enciso, J. Li, D. Fidaleo, Tae-Yong Kim, J. Noh, and U. Neumann. Synthesis of 3d faces. In *Proc. of Digital and Computational Video*, 1999.

[60] F. Erol. An interactive facial animation system. In *Proc. of WSCG*, pages 5 – 8, 2001.

[61] J. Garofalo et al. Darpa timit cd-rom: An acoustic phonetic continuous speech database. *National Institute of Standards and Technology, Gaithersburg, MD*, 1998.

[62] T. Ezzat, G. Geiger, and T. Poggio. Trainable video realistic speech animation. In *Proc. of ACM SIGGRAPH*, 2002.

[63] T. Ezzat and T. Poggio. Videorealistic talking faces: A morphing approach. In *Proc. of AVSP'97 Workshop*, 1997.

[64] T. Ezzat and T. Poggio. Miketalk: A talking facial display based on morphing visemes. In *Proc. of the Computer Animation Conference*, 1998.

[65] T. Faruquie, A. Kapoor, R. Kate, N. Rajput, and L. Subramaniam. Audio driven facial animation for audio-visual reality. In *Proc. of IEEE International Conference on Multimedia and Expo Tokyo*, 2001.

[66] F.Parke. *A parametric model for human faces*. University of Utah, Department of Computer Science, 1974.

[67] G. Geiger, T. Ezzat, and T. Poggio. Perceptual evaulation of video-realistic speech. *MIT AI Memo 2003-003, CBCL Memo 224*, 2003.

[68] B. Goff, T. Guiard-Marigny, M. Cohen, and C. Benoit. Real-time analysis-synthesis and intelligibility of talking faces. In *Proc. of 2nd ESCA/IEEE Workshop on Speech Synthesis*, 1994.

[69] B. Le Goff and C. Benoit. A text-to-audiovisual speech synthesiser for french. In *Proc. of ICSLP96*, 1996.

[70] G. Gold and N. Morgan. *Speech and Audio Signal Processing : Processing and Perception of Speech and Music*. Wiley, 1999.

[71] H. Gray, L. H. Bannister, M. M. Berry, and P. L. Williams. *Gray's Anatomy: The Anatomical Basis of Medicine and Surgery*. Churchill Livingstone, 1995.

[72] C. Hack and C. Taylor. Modelling talking head behaviour. In *Proc. of British Machine Vision Conference (BMVC)*, 2003.

[73] P. Hall, D. Marshall, and R. Martin. Adding and subtracing eigenspaces. In *Proc. of BMVC*, 1999.

[74] C. Hjortsjo. *Mans Face and the Mimic Language*. Studentlietur, 1969.

[75] P. Hong, Z. Wen, and T. S. Huang. Real-time speech-driven face animation with expressions using neural networks. *IEEE Transactions on Neural Networks*, 13(4):916–927, 2002.

[76] F. J. Huang and T. Chen. Real-time lip-synch face animation driven by a human voice. In *Proc. of IEEE Workshop on Multimedia Signal Processing, Los Angeles, California*, 1998.

[77] M. Jones and T. Poggio. Multidimensional morphable models. In *Proc. of 6th International Conference on Computer Vision (ICCV)*, pages 683–688. IEEE Computer Society, 1998.

[78] P. Joshi, W. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *Proc. of Eurographics*, 2003.

[79] K. Kahler, J. Haber, and H. P. Seidel. Geometry-based muscle modelling for facial animation. In *Proc. of Graphics Interface*, pages 37 – 47, 2001.

[80] K. Kahler, J. Haber, and H. P. Seidel. Reanimating the dead: Reconstruction of expressive faces from skull data. *ACM Transactions on Graphics*, 22(3):554–561, 2003.

[81] K. Kahler, J. Haber, H. Yamauchi, and H. P. Seidel. Head shop: generating animated head models with anatomical structure. In *Proc. of 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 55–63. ACM Press, 2002.

[82] P. Kakumanu, R. Gutierrez-Osuna, A. Esposito, R. Bryll, A. Goshtasby, and O. Garcia. Speech driven facial animation. *ACM Multimedia*, 2001.

[83] G. Kalberer and L. Van Gool. Face animation based on observed 3d speech dynamics. In *Proc. of 14th IEEE. Conf. on Computer Animation (CA'01), IEEE Computer Society*, pages 20 – 27, 2001.

[84] Y. Karaulova, P. Hall, and A.D. Marshall. Tracking people in three dimensions using a hierarchical model of dynamics. *Image and Vision Computing*, 20(9), 2002.

[85] J. Katsyri, V. Klucharev, M. Frydrych, and M. Sams. Identification of synthetic and natural emotional facial expressions. In *Proc. of International Conference on Audio-Visual Speech Processing*, pages 239–243, 2003.

[86] E. Keele, S. Girod, P Pfeifle, and B. Girod. Anatomy-based facial tissue modelling using the finite element method. In *Proc. of Visualisation*, 1996.

[87] P. Kelly, E. Hunter, K. Kreutz-Delgado, and R. Jain. Lip posture estimation using kinematically constrained mixture models. In *Proc. of BMVC*, 1998.

[88] I. Kerlow. *The Art of 3D Computer Animation and Effects, 3rd Edition.* Wiley, 2003.

[89] M. Kleiner, A. Schwaninger, D. Cunningham, and B. Knappmeyer. Using facial texture manipulation to study facial motion perception. In *Proc of ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, 2004.

[90] E. Krumhuber, D. Cosker, A. Manstead, D. Marshal, and P. Rosin. Synthetic humans for the study of subtle temporal aspects in facial displays. In *Proc. of the 9th Conference of the International Society for Research on Emotions*, 2005.

[91] E. Krumhuber, D. Cosker, A. Manstead, D. Marshal, and P. Rosin. Temporal aspects of smiles influence employment decisions: A comparison of human and synthetic faces. In *Proc. of the 11th European Conference on Facial Expressions: Measurement and Meaning*, 2005.

[92] E. Krumhuber, D. Cosker, A. Manstead, D. Marshal, and P. Rosin. Temporal dynamics of smiling: Human versus synthetic faces. In *Proc. of the 9th Conference of the International Society for Research on Emotions*, 2005.

[93] S. Kshirsagar and N. Magnenat-Thalmann. Visyllable based speech animation. *Computer Graphics Forum*, 22(3), 2003.

[94] Curious Labs. Poser 5, 2004.

[95] A. Lanitis, C. Taylor, and T. Cootes. Towards automatic simulation of ageing effects on face images. *IEEE PAMI*, 24(4):442–455, 2002.

[96] Y. Lee, D Terzopoulos, and K. Waters. Realistic modelling for facial animation. In *Proc. ACM SIGGRAPH*, 1995.

[97] Y. Li, F. Yu, Y. Xu, E. Chang, and H. Shum. Speech-driven cartoon animation with emotions. *ACM Multimedia*, 2001.

[98] J. Lien, T. Kanade, J. Cohn, and C. Li. Automated facial expression recognition based on facs action units. In *Proc. of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pages 390 – 395, 1998.

[99] I. Lin, C. Huang, J. Wu, and M. Ouhyoung. A low bit-rate web-enabled synthetic head with speech-driven facial animation. In *Proc. of Eurographics Workshop on Computer Animation and Sumulation'2000 (CAS 2000)*, pages 29 – 40, 2000.

[100] I. Lin, C. Hung, T. Yang, and M. Ouhyoung. A speech driven talking head system based on a single face image. In *Proc. of 7th Pacific Conference on Computer Graphics and Applications*, 1999.

[101] I. Lin, J. Yeh, and M. Ouhyoung. Realistic 3d facial animation parameters from mirror-reflected multi-view video. In *Proc. Computer Animation 2001 (CA 2001)(IEEE ISBN 0-7803-7237-9), IEEE Computer Society*, pages 2 – 11, 2001.

[102] A. Lofqvist. *Speech as Audible Gestures. In W. Hardcaste and A. Marchal (Eds.) Speech Production and Speech Modelling*. Dordrecht: Kluwer Academic Publishers, 289-322, 1990.

[103] J. Luettin, N. Thacker, and S. Beet. Locating and tracking facial speech features. In *Proc. of International Conference on Pattern Recognition ICPR*, 1996.

[104] J. Luettin and N. A. Thacker. Speechreading using probabilistic models. *CVIU*, 65(2):163–178, 1997.

[105] Z. Lui, Z. Zhang, C. Jacobs, and M. Cohen. Rapid modeling of animated faces from video. *Technical Report MSR-TR-2000-11, Microsoft Research*, 2000.

[106] J. MacDonald, S. Anderson, and T. Bachmann. Hearing by eye: How much spatial degradation can be tolerated? *Perception*, 29(10):1155–1168, 2000.

[107] H. Malvar, F. Pighin, C. Grimm, and D. Wood. Making faces. In *Proc. of ACM SIGGRAPH*, 1998.

[108] D. Massaro. *Percieving Talking Faces: from Speech Perception to a Behavioral Principle*. MIT Press, 1998.

[109] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[110] Image Metrics. www.image-metrics.com, 2005.

[111] J. Nash. *The Singular-Value Decomposition and its Use to Solve Least-Squares Problems.* Adam Hilger. 2nd Edition, 1990.

[112] U. Neumann, J. Li, R. Enciso, J. Noh, D. Fidalea, and T. Kim. Constructing a realistic head animation mesh for a specific person. *Tech Report USC-TR 99-691, University of Southern California*, 1999.

[113] S. Ohman. Numerical model of coarticulation. *Journal of the Acoustical Society of America*, 41:310–320, 1967.

[114] N. Oliver, A. Pentland, and F. Berard. Lafter: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33, 1369-1382.

[115] P. Omedas, F. Berrizbeitia, G. Szijarto, B. Kiss, and B. Takacs. Model-based facial animation for mobile communication. In *Proc. of 1st Ibero-American Symposium on Computer Graphics (SIACG)*, 2002.

[116] S. Ouni, D. Massaro, M. Cohen, K. Young, and A. Jesse. Internationalization of a talking head. In *Proc. of the 15th International Congress of Phonetic Sciences (ICPhS03) Poster Session (CD-ROM)*, 2003.

[117] M. Pantic, M. Tomc, and L. Rothkrantz. A hybrid approach to mouth feature detection. In *Proc. of IEEE Systems, Man and Cybernetics*, 2001.

[118] F. Parke. Computer generated animation of faces. In *Proc. of ACM National Conference*, 1972.

[119] F. Parke. Parameterised models for face animation. *IEEE Computer Graphics and Applications*, 1982.

[120] Frederic I. Parke and Keith Waters. *Computer Facial Animation.* A. K. Peters, 1996.

[121] Paramount Pictures. Forrest gump, 1994.

[122] Paramount Pictures. Enemy at the gates, 2001.

[123] Universal Pictures. Terminator 2: Judgement day, 1991.

[124] F. Pighin, J. Hecker, D. Lischinski, and R. Szeliski. Synthesising realistic facial expressions from photographs. In *Proc. of ACM SIGGRAPH*, 1998.

[125] F. Pighin, R. Szeliski, and D. Salesin. Resynthesising facial animation through 3d model-based tracking. In *Proc. of ICCV*, 1999.

[126] Pixar. Monsters inc., 2001.

[127] S. Platt and N. Badler. Animating facial expressions. *Computer Graphics*, 15(3):245 – 252, 1981.

[128] G. Potamianos, H. Graf, and E. Cosatto. An image transform approach for hmm based automatic lipreading. In *Proc. of International Conference on Image Processing*, volume 3, pages 173 – 177, 1998.

[129] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press; 2 edition (October 30, 1992).

[130] H. Pyun, Y. Kim, and W. Chae. An example-based approach for facial expression cloning. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 167 – 176, 2003.

[131] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–285, February 1989.

[132] R. Rao and T. Chen. Exploiting audio-visual correlations in coding of talking head sequences. In *Proc. PCS*, 1996.

[133] P. Ratner. *3D Human Modelling and Animation*. John Wiley and Sons Inc, 2003.

[134] L. Reveret, G. Bailly, and P. Badin. Mother: A new generation of talking heads providing a flexible articulatory control for video-realistic speech animation. In *Proc. of the 6th Int. Conference of Spoken Language Processing, ICSLP'2000, Beijing, China*, pages 16–20, 2000.

[135] R. Rodman, D. McAllister, D. Blitzer, and A. Freeman. Automated lip-sync animation as a telecommunications aid for the hearing impaired. In *Proc. of Interactive Voice Technology for Telecommunications Applications (IVTTA)*, pages 204–209, 1998.

[136] M. Rydfalk. Candide, a parameterised face. *Technical Report No. LiTH-ISY-I-0866, Dept. of Electrical Engineering, Linkping University, Sweden.*, 1987.

[137] M. Sadeghi, J. Kittler, and K. Messer. Modelling and segmentation of lip area in face images. *IEE Vision Image and Signal Processing*, 149(3), 2002.

[138] Valve Software. Half-life 2, 2004.

[139] M. Stegmann. *Active Appearance Models: Theory, Extensions and Cases.* Masters Thesis. Technical University of Denmark, 2000.

[140] S. Stevens, J. Volkman, and E. Newman. A scale measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937.

[141] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics*, 23(3):506–513, 2004.

[142] G. Szijartoo, B. Kiss, and B. Takcs. Model-based real-time facial animation: Design and implementation. In *Proc of. Computer Animation and Geometric Modeling*, 2002.

[143] A. Tekalp and J. Ostermann. *Face and 2-D Mesh Animation in MPEG-4.* Telecom Italia Lab: White Paper, 2000.

[144] D. Tersopoulos and K. Waters. Physically based facial modelling, analysis and animation. *Visualisation and Computer Animation*, 1(2):73–80, 1990.

[145] B. Theobald, G. Cawley, J. Glauert, and A. Bangham. 2.5d visual speech synthesis using appearance models. In *Proc. of British Machine Vision Conference (BMVC)*, 2003.

[146] F. Thomas and O. Johnston. *The Illusion of Life: Disney Animation.* Hyperion, 1995.

[147] Y. Tian, T. Kanade, and J. Cohn. Recognising lower face action units for facial expression analysis. In *Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 484 – 490, 2000.

[148] Y. Tian, T. Kanade, and J. Cohn. Robust lip tracking by combining shape, colour and motion. In *Proc. of ACCV*, 2000.

[149] S. E. Tice and J. Lander. Character animation engines for interactive game and web applications using hardward assisted functionality. In *Proc. of International Workshop on Human Modelling and Animation (HMA), Seoul National University*, 2000.

[150] Columbia Tri-Star. Final fantasy: The spirits within, 2001.

[151] P. Tu, I. Lin, J. Yeh, R. Liang, and M. Ouhyoung. Expression detail mapping for realistic facial animation. In *Proc. of CADCG*, pages 20–25, 2003.

[152] M. Turk and A. Pentland. Eigenfaces for recognition. *Neuroscience*, 3(1), 1991.

[153] B. Ulicny, P. Ciechomsky, and D. Thalmann. Crowdbrush: interactive authoriing of real-time crowd scenes. In *Proc. of 2004 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, pages 243–252, 2004.

[154] P. Vanezis, M. Vanezis, G. McCombe, and T. Niblett. Facial reconstruction using 3-d computer graphics. *Forensic Science International*, 1999.

[155] K. Walker, T. Cootes, and C. Taylor. Automatically building appearance models from image sequences using salient features. In *Proc. of BMVC'99*, 1999.

[156] K. Waters. A muscle model for animating 3d facial expression. *ACM Computer Graphics*, 21(4):17–24, 1987.

[157] B. Welsh. *Model-Based Coding of Images*. PhD Thesis, British Telecom Research Lab, 1991.

[158] L. Williams. Performance driven facial animation. *Computer Graphics*, 24(4):235 – 242, 1990.

[159] Y. Zhang, E. Prakash, and E. Sung. Anatomy-based 3d facial modelling for expression animation. *Machine Graphics and Vision International*, 11(1):53–76, 2002.

[160] Y. Zhang, E. Prakash, and E. Sung. Efficient modelling of an anatomy-based face and fast 3d facial expression synthesis. *Computer Graphics Forum*, 22(2):159 – 164, 2003.