Cardiff University
School of Computer Science and Informatics
Visual Computing Group

4D (3D Dynamic) Statistical Models of Conversational Expressions and the Synthesis of Highly-Realistic 4D Facial Expression Sequences

Jason Vandeventer

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science at Cardiff University, July 24th, 2015

Abstract

In this thesis, a novel approach for modelling 4D (3D Dynamic) conversational interactions and synthesising highly-realistic expression sequences is described.

To achieve these goals, a fully-automatic, fast, and robust pre-processing pipeline was developed, along with an approach for tracking and inter-subject registering 3D sequences (4D data). A method for modelling and representing sequences as single entities is also introduced. These sequences can be manipulated and used for synthesising new expression sequences. Classification experiments and perceptual studies were performed to validate the methods and models developed in this work.

To achieve the goals described above, a 4D database of natural, synced, dyadic conversations was captured. This database is the first of its kind in the world.

Another contribution of this thesis is the development of a novel method for modelling conversational interactions. Our approach takes into account the time-sequential nature of the interactions, and encompasses the characteristics of each expression in an interaction, as well as information about the interaction itself.

Classification experiments were performed to evaluate the quality of our tracking, inter-subject registration, and modelling methods. To evaluate our ability to model, manipulate, and synthesise new expression sequences, we conducted perceptual experiments. For these perceptual studies, we manipulated modelled sequences by modifying their amplitudes, and had human observers evaluate the level of expression realism and image quality.

To evaluate our coupled modelling approach for conversational facial expression interactions, we performed a classification experiment that differentiated predicted frontchannel and backchannel sequences, using the original sequences in the training set. We also used the predicted backchannel sequences in a perceptual study in which human observers rated the level of similarity of the predicted and original sequences. The results of these experiments help support our methods and our claim of our ability to produce 4D, highly-realistic expression sequences that compete with state-of-the-art methods.

Dedication

Dedicated to my parents, Debbie and Larry Vandeventer. You always encouraged me to reach for the stars, while letting me know I always had a safe place to fall.

Acknowledgements

I would like to express my deep gratitude to my primary supervisor, Professor David Marshall. The quality of his research and passion for pursuing new ideas gave me the confidence to move to a new country to pursue my PhD. A great deal of thanks to my co-supervisor, Professor Paul Rosin, whose ideas and suggestions always offered insightful perspectives and shaped the way I approach research challenges.

Many thanks to Dr. Andrew Aubrey for helping me settle in to a new country, as well as for helping me understand the long-term vision of my supervisors early on in my PhD career. My gratitude also goes to Dr. David Pickup for his countless help in Matlab, and troubleshooting issues during one of the most challenging sections of my PhD. His assistance with Blender related issues, including the creation of research images, was priceless.

I would also like to thank Professor Tony Manstead and Dr. Job van der Schalk for including me in their Social Psychology seminar group. It has provided me a perspective that I feel has made me a more well-rounded research scientist. Thank you as well to everyone in the *VLunch* seminar group. Your feedback on my presentations taught me how to better analyse my research and the research of others.

Thank you to my research colleagues, Lukas Gräser and Dr. Magdalena Rychlowska. Without you two, many of these accomplishments would not have been achievable.

To my academic friends, Ben Barbour and Dijana Tralić. Thank you for many nights of discussion, debate, and analysis, but also evenings of laughter and enjoyment. You provided me with perspective when I needed it the most, and for that, I cannot thank you enough.

A special thank you to Professor Curry Guinn. He not only convinced me to study computer science, but also encouraged me to pursue a PhD very early on in my academic career.

Finally, thank you to my family and friends. You have supported me unconditionally since I began this journey in a strange and foreign land... even though I "don't spell words correctly anymore".

Quote

"Science is much more than a body of knowledge. It is a way of thinking. This is central to its success. Science invites us to let the facts in, even when they don't conform to our preconceptions. It counsels us to carry alternative hypotheses in our heads and see which best match the facts. It urges on us a fine balance between no-holds-barred openness to new ideas, however heretical, and the most rigorous skeptical scrutiny of everything – new ideas and established wisdom." -Carl Sagan

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation, Objectives, and Challenges

Facial expressions in human face-to-face conversation are an important aspect of social communication [36, 61, 62, 167]. They communicate a variety of social signals [35, 63, 90, 233], from agreement and confusion, to anger and disgust. Understanding the effects of facial expressions in conversational interactions is important for applications in the fields of affective computing, Human-Computer Interaction, social psychology, digital animation, biometrics, computer vision, and a variety of other research fields.

Face-to-face interactions play an important role in dyadic conversations [61, 167]. Facial expressions can inform a viewer of a variety of things, such as an individual's cognitive state (e.g. thinking, confusion), affective state (e.g. joy, fear), and their engagement level in a conversation (e.g. attention, agreement/disagreement). Recent research has shown that the listener in a conversation provides quite a bit of control in both conversational flow and information discussed [36, 35, 47, 56, 62, 63, 131, 187, 233, 248]. Each expression interaction can potentially have an effect on the flow, content, and even mood of a conversation. By analysing conversational interactions, we can better understand the mechanisms involved in this important social activity. The work in this thesis looks to provide an approach for modelling these interactions, so they may be studied further.

For obvious reasons, these fields would prefer to use faces that look and move realistically. This is most easily achieved by capturing an individual with a video camera and using those captured clips to produce an intended output. More recently, 2D statistical models have been used for synthesising (i.e. creating new frames that were not captured) new expression output [7, 8, 14, 115, 239, 252, 254].

While 2D data and models are certainly useful, 3D video (4D) data captures offer a "real" representation of the characteristics of facial expressions and head movements. That is, 3D data offers the advantage of providing intrinsic geometry which is invariant to pose and lighting, and 3D dynamic (4D) data includes temporal information, which is very important for modelling and synthesising highly-realistic facial expressions. Perceptual studies [79] have shown that, overall, preserving the temporal dynamics of synthesised facial expressions increases perceived naturalness.

Systems with the ability to capture high-resolution 3D video are very useful for providing data that can be used to build statistical models of appearance. Systems with the ability to capture two people in conversation allow for the analysis and modelling of the interactions between two people. Statistical modelling of such interactions is a particularly promising research area because it not only allows us to quantify qualitative data, but can also be used to interpolate new data. Thus, these models can be used to synthesise new 3D facial expression frames. As well, by manipulating model parameters it is possible to synthesise new facial expression frames (i.e. frames that were not captured during data acquisition). This ability is extremely useful when the data capturing process may not record every aspect of an expression that may be needed for an application. For instance, having the ability to capture a single smile sequence, modify its intensity, and synthesise a highly-realistic, reduced-intensity smile could be very useful for researchers interested in the perceptions of smiles based on their varying characteristics (i.e. expression movements and timings).

Until very recently, model manipulation and synthesis methods for 3D data produced relatively unrealistic looking expression frames (i.e. those which were created rather

than captured). An additional shortcoming in this research area is the lack of temporal information in the synthesis of expression sequences. Most work only synthesises important frames (such as a neutral expression and the highest intensity of a smile sequence) and morphs from one target frame to the next (see [79, 249] for examples and discussion on this issue). The work in this thesis looks to address these issues.

One of the objectives of this work is to model, manipulate, and synthesise facial expression sequences to a level of realism not before achieved using a statistical modelling approach. Highly-realistic facial expression sequences synthesised from statistical models, especially those which retain the subject's expression characteristics, can be very useful for enhancing research in many key areas. These areas include research in affective computing (e.g. interactive virtual agents [95, 234]), entertainment (e.g. more realistic facial expressions for digital characters [9, 112]), medicine (e.g. planning surgeries and how a patient's face moves before and after surgery [188, 189]), business (e.g. negotiation tactics [77, 142, 220]), clinical psychology (e.g. helping individuals with Autism Spectrum Disorder (ASD) syndrome better understand facial expressions in social settings [69, 108, 190]), and social psychology (e.g. perceptions of trustworthiness [178, 223]).

Building on the first objective, the second objective is to develop coupled statistical models of conversational interactions, for both analysing interactions and synthesising new expression interactions. Models of how individuals interact in conversation is useful for identifying mechanisms of conversations. Having the ability to modify and synthesise parts of an interaction, or even predict what part of the interaction would look like given the complementary part, could be very useful in a variety of research fields. This includes research in facial mimicry [35], expression reciprocation [47], and control in dyadic conversations [233].

Developing the methods needed for achieving these objectives will provide applications and techniques that will be useful for other researchers in the community, specifically those working with 4D face data. This could result in more realistic

virtual humans, digitally animated faces with accurate temporal dynamics, artificially intelligent systems capable of recognising conversational expressions, and many related applications.

For any of these statistical modelling-based applications to be successful, quality data of facial expressions, specifically people in conversation, is needed. For 3D/4D modelling of conversational expressions, this means 4D (3D video) captures. Before this work no such database existed of 4D conversations. Therefore, the first major challenge to overcome was to develop a 4D database of natural conversations. This proved to be no simple task, as the amount of data captured required time-consuming processing and a large amount of storage space (on the order of terabytes). The data acquisition process is described in Chapter 3.

The next challenge arose from the capture system's data format. For each 3D frame, the capture system provides an OBJ and a three-view texture map (i.e. a texture map with one view per colour camera). When modifying a vertex its corresponding texture map coordinate is also modified. This often resulted in texture artefacts. In many cases, the modification of the vertex's texture coordinates resulted in the three texture coordinates for each face (i.e. polygon) being located in separate images of the three-view texture map. The resulting texture was a blob of colour due to the texture triangle overlapping images in the texture map. For this reason, a *cleaning* approach was developed to remove problematic items from the original mesh, such as non-manifolds, and to create a single-view, Unified Texture Map (UTM) so that there would be no overlapping issues. Chapter 4 describes the development and approaches used in this cleaning process.

The 3D frames, even the *cleaned* data, are completely independent from each other. The vertices, faces, and texture maps do not correspond in anyway. In fact, they contain a different number of vertices and faces from one frame to the next. In order to use this data in statistical models, the frames need to correspond. Corresponding data can be compared, which is what PCA-based approaches like Active Appearance Models (AAM) [72, 73] do; they describe variances in the data and model its

characteristics. Accurate 3D tracking of facial feature points, actually using the 3D data and not simplifying it down to a 2D problem, is still an active area of research. Neither the tracking approaches provided by third-party software tools, nor the approaches described in the literature, were sufficient to provide the level of accuracy and semi-automatic method that we desired. For this reason a sparse tracking approach was developed which uses both 3D shape (curvature) and texture information. Since the goal is to synthesise highly-realistic expression sequences, a dense registration approach was required. Our Thin Plate Spline (TPS) [48, 99] based approach uses the sparse tracking points as anchors in a dense registration approach. The output allows for an inter-subject registered mesh that is nearly identical in shape, resolution, and appearance to the original 3D frame. Chapter 5 describes the development of these methods.

A 3D Active Appearance Model can be built using the registered data. The problem is that this modelling approach describes individual frames of a sequence, but not the sequence itself. A method for representing an entire 3D sequence as a single entity was necessary to allow for analysis at the sequence level, as well as for modification of sequences as a whole. Retaining an individual's temporal dynamics was an important goal. Most research in the area of 4D expression synthesis uses warping techniques to morph from a neutral expression to a target expression, in a pre-determined number of morphing steps, such as in [249]. This temporal information is an important detail of facial expressions [79], especially in conversations. This challenge was overcome by using polynomial regression to fit a curve to our nicely structured Duchenne/Non-Duchenne smile data, as well as the much more variable conversational expression data we captured. Chapter 3 describes these datasets and Chapter 6 describes our methods for modelling and expression sequence synthesis.

The final major challenge that needed to be solved for this research was how to create coupled statistical models of time-sequential actions. While other coupled statistical modelling approaches are described in the literature, they all focus on actions that are co-occurring, such as one person nodding while another shakes their head. The conversation data, however, includes interactions where the social

signal (facial expression) of *Person A* can elicit a response by *Person B*. *Person B's* response may, in-turn, elicit a reaction from the *Person A*. This back-and-forth can occur any number of times throughout a conversation. The solution was developed, in-part, by looking at how financial markets in different locations can affect each other through a day of trading [70]. For a single interaction, the feature vector of the frontchannel signal is concatenated with the feature vector of the backchannel signal, with some additional information concatenated as well. Each new feature vector describes an interaction, and all interactions of a certain type make up the coupled model. This coupled model approach is described in greater detail in Chapter 6.

1.2 Main Contributions

One of the major contributions of this work is the world's first publicly available 4D database of natural, dyadic conversations [229]. This high-resolution database was captured at 60fps and consists of 17-minutes of conversations. The database has also been fully annotated for speaker and listener activity: conversational facial expressions, head motion, and verbal/non-verbal utterances. The annotated data consists of 764 frontchannel/backchannel expression periods (329 Frontchannel, 435 Backchannel). The annotations, 2D videos from the centre camera, the 3D original frames, as well as the 3D *cleaned* frames, have been made available to the research community. We believe this expression-rich, audio-visual database of natural conversations will make a useful contribution to the computer vision, affective computing, and cognitive science communities by providing raw data, features, annotations, and baseline comparisons.

The robust, *fully-automated* pre-processing (cleaning) pipeline is also a contribution of this work. While the individual steps used in the pipeline are not novel, the way in which they are combined and the order in which they are implemented allows for the solution to a challenging issue that many in the research community have seen as a barrier to entry, in regards to modifying 3D sequence data. While private companies may have developed in-house solutions, this work provides specific steps to the

research community for solving this issue, using Matlab [163] and the freely-available, open source software program Blender [1]. The issues solved by this pipeline are ones which any researcher with multi-view texture maps may face when modifying the vertices and texture coordinates of their data. The code for this robust pre-processing pipeline will be made publicly available, and can be used by researchers who have multi-view 3D and 4D capture systems or who have this type of data, as it does not require any system-specific information (e.g. calibration information, camera configuration, etc.). See Section 4.7 for evaluation of this pipeline, which includes meshes from previously-unseen data captures from a variety of capture systems.

The approach of using polynomial regression on the principal component values for each frame in a sequence, to represent the sequence as a single entity, is also a contribution of this work. This single entity can be modified in various ways and a reverse-process can be used for synthesising new sequence output. Polynomial regression is not new, but the way in which it was used, along with 3D AAM data, offers a guide to how researchers may be able to start including the temporal dynamics information for their 4D data.

The coupled statistical models described in this work is another novel contribution. Modelling time-sequential interactions is a challenging problem and the coupled model approach here is a sound method for creating such models of conversational expression interactions. The classification and perceptual experiments described in Section 7.6 (in Chapter 7) help support this claim. While not exhaustively explored, it offers a method other researchers in the community can use, specifically with the data from our conversation database.

The creation of highly-realistic, synthesised expression sequences is also a major contribution of this work. Using statistical modelling approaches, 4D synthesised expression sequences created in this work competes with the quality and realism (dynamic expressions and image quality) of current state-of-the-art approaches [9, 10, 38, 54, 80, 91, 112, 143, 238, 243, 249]. For descriptions and analysis, please see Section 2. This contribution includes the ability to manipulate the created models

and use them for generating perceptually convincing output sequences, which were validated by human observers in perceptual experiments. The perceptual experiments described in Section 7.5 help support this claim.

1.3 List of Relevant Publications

Published

- A. J. Aubrey, D. Marshall, P. L. Rosin, J. Vandeventer, D. W. Cunningham, and C. Wallraven. "Cardiff conversation database (CCDb): A database of natural dyadic conversations", In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 277-282, 2013.

Accepted

- J. Vandeventer, L. Gräser, M. Rychlowska, P. L. Rosin, D. Marshall, "Towards 4D Coupled Models of Conversational Facial Expression Interactions", In *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, 2015.
- J. Vandeventer, A. J. Aubrey, P. L. Rosin, D. Marshall, "4D Cardiff Conversation Database (4D CCDb): A 4D Database of Natural, Dyadic Conversations", In *The 1st Joint Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing (FAAVSP)*, 2015.

In Preparation

- L. Gräser, J. Vandeventer, J. van der Schalk, P. L. Rosin, D. Marshall, "4D Tracking and Inter-Subject Registration for the Synthesis of Realistic Facial Expression Sequences", 2015

1.4 Thesis Outline

Chapter 2 discusses previous work in the areas of facial expression databases, facial features and expression recognition, expression synthesis, and dynamic statistical models of facial expressions. Chapter 3 describes the data acquisition system and process for capturing the conversation and smile data used for model building and the experiments. In Chapter 4 the *fully-automated* process for producing cleaned meshes and their single-view, Unified Texture Maps is covered. Chapter 5 describes the multiple 3D landmarking, tracking, and registration approaches that were developed and used over the course of this work. Chapter 6 explains the process of modelling expression sequences, the methods used to manipulate and synthesise new, highly-realistic expression sequences, and the coupled statistical modelling approach we propose for modelling expression interactions. This coupled model is used for predicting conversational interaction expression sequences. Chapter 7 describes in detail the experimental approach and results for the classification and perceptual experiments, which were conducted using two separate databases. Chapter 8 concludes by covering the achievements of this work, potential applications of this work, and future research to be conducted.

Chapter 2

Background

2.1 Introduction

Facial expressions are an important aspect of human face-to-face conversation. Facial expressions can convey emotions, provide feedback, and communicate a variety of other social signals [35, 63, 90, 233]. While most humans can identify and understand facial expressions without any special training, it is more complex for machines; specifically those used in computer vision applications. The first barrier for creating systems that understand facial expressions is the lack of useful data. Over the past 10-15 years many databases of facial expressions have been built [80, 135, 158, 165, 203, 204, 253]. These include posed and spontaneous expressions, and many of them have been systematically annotated according to a standard for specifying facial movements, the most popular being the Facial Action Coding System (FACS) [105]. These databases include 2D still images, 2D videos, 3D still images, and 3D videos (or *4D*), and are used in a variety of research applications, such as identifying features in the face (such as FACS action units), tracking features across a video sequence, and building models of facial expressions.

For 2D/3D static data, manual annotation of facial feature points is very common. While it is a time-consuming process, landmarking various peak facial expressions for each subject still provides a useful amount of data. With 4D data (and a frame

rate of 60fps), manual annotation of the entire sequence is simply not feasible. For this reason, there is quite a bit of research in developing more automated approaches. Statistical models, specifically the Active Appearance Models (AAMs) [72, 73] used in this work, require corresponding data. There are several registration techniques one can use. Certain techniques work better for achieving a registered mesh that closely resembles the original 3D frame. Various tracking and registration techniques are described in Section 2.6.

A major focus of computer science researchers has been the detection and classification of FACS action units (AU's). The ability to identify action units and their intensities is of use specifically to the areas of expression recognition and biometrics. Applications include the ability to identify people in pain [15, 151] (useful for tele-medicine applications) or identify individuals with non-neutral facial expressions [39] (biometrics). The wide variety of research in this area is described in detail in Section 2.8.

Those who build statistical models of facial expressions are interested in the characteristics of those expressions, including the temporal dynamics. Statistical models of appearance also allow for researchers to synthesise new expression sequences [45, 73, 79, 249]. These are of particular use for developing stimuli for psychology experiments or increasing realism in virtual humans. Coupled models allow researchers to couple data and behaviours, such as a front-view of a face with the corresponding side (profile) view [76]. This area of research is covered in Section 2.9.

The literature of previous research will, among other things, lay a basis for why a new 4D conversational database was required for the research conducted in this work, as well as show state-of-the-art examples of realistic, synthesised faces which can be used as a comparison to the synthesised facial expressions developed in this work.

2.2 Conversational Interactions

Face-to-face conversations are a frequent and important part of social communication. These conversations, whether with well-known friends or complete strangers, consist of a variety of verbal and non-verbal signals (e.g. expressions, gestures), which determine the tone, content, and flow of a conversation [36, 56, 62, 248].

A dyadic conversation is not simply made up of two independent individuals. There exists a mutual-dependence which affects the structure of the conversation and interaction. Therefore, analysis of dyadic conversations should not focus on these individuals as two separate entities, but rather as *non-independent* entities [136, 137].

During face-to-face conversations, there is a considerable degree of communication from the listener to the speaker, which often serves to control conversational flow [35, 36, 47, 56, 62, 63, 131, 187, 233, 248]. In [248], Yngve coined the term *backchannel* to describe the signals being sent from the listener(s) to the speaker (Figure 2.1). This feedback can indicate comprehension (e.g. a look of confusion), provide an assessment (e.g. saying "correct"), control conversational flow, or even add new content (e.g. sentence completion). Conversely, the term *frontchannel* is used to describe the speaker's behaviour. Obviously, the frontchannel and backchannel roles may be swapped multiple times during a conversation; this dynamic relationship is what allows for the conversation's path to be altered based on expressed and received conversational expressions.

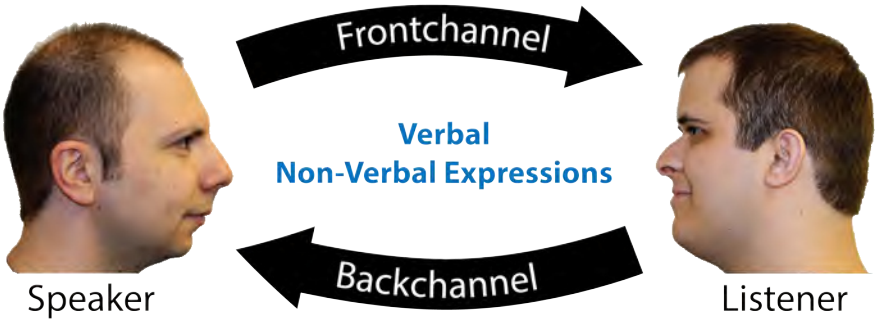


Figure 2.1: Frontchannel-Backchannel Interaction

2.3 Facial Expressions

Understanding emotion is an invaluable skill for normal communication in day-to-day interactions for human beings. Emotions are the psycho-physiological states that an individual can experience, such as happiness, sadness, anger, fear, disgust, contempt, and surprise. Emotions are expressed in a variety of ways including body gestures, audible noises, and facial expressions.

In 1978 Dr. Ekman and Dr. Wallace Friesen developed the Facial Action Coding System (FACS) [104, 105]. FACS is a tool used to measure and describe facial movements and has become the de facto standard for measuring and describing facial expressions. The face is capable of making over 10,000 expressions. Not all expressions are linked to emotions. Using FACS, the 7 "prototypical" expressions (happiness, sadness, fear, anger, disgust, surprise, and contempt) and their intensities can be described. A FACS-trained individual can spot expressions and use the information to make decisions about the individual displaying them. Many employers, such as law enforcement and government agencies, use FACS to train their employees in order to increase effectiveness by increasing their ability to detect deception.

While a substantial amount of research has focused on recognizing the so-called *prototypical expressions*, they are not common in everyday situations. Expressions that are common are those related to conversations. Face-to-face communication is essential to daily life, but most research has only focused on the 7 universal emotions proposed by Ekman. The *conversational expressions* that occur frequently are expressions such as agreement, disagreement, thinking, pleasant surprise, and confusion. While the expressions of the speaker (main channel) can control the flow of a conversation, recent research has shown that the expressions of the listener (backchannel) are just as important [84, 85, 86, 87]. In [36], Bavelas et al. showed the difference between generic and specific responses to a story-teller. The specific responses increased the confidence of the speaker, resulting in more-detailed, longer stories. The engagement enhanced the situation and showed that the listener is a vital part to a conversation. Understanding the dynamics and interactions of these types

of signals is important for not only better understanding human communication, but also for the various types of computing applications that can be developed; for instance, creating better artificial intelligence agents whose purpose is to realistically communicate with humans [62].

2.3.1 Conversational Expressions

Early work on conversational modelling focused on written transcripts of conversations. As a result, traditional models of communication assumed that in any dyadic conversation one person was active (the speaker) and one was passive (the listener). Since at least 1970, however, it has been repeatedly shown that human conversations are very much multimodal. In addition to the words chosen, it has been found that prosody, facial expressions, hand and body gestures, and gaze all convey conversational information. For example, Birdwhistell has shown that speech conveys only about one-third of the information in a conversation [42]. The rest of the information is distributed throughout a number of non-verbal semiotic channels, such as hand or facial motions [90]. It has also been shown that non-verbal information is often given a greater weight than spoken information: when the spoken message conflicts with facial expressions, the information from the face tends to dominate [61, 167].

In most conversations the role of the speaker and listener changes from person to person throughout the conversation. One moment an individual may be the speaker and producing frontchannel expressions, while in the next moment their role has shifted to listener and their expressions are of the backchannel type. This dynamic relationship is what allows for the conversation's path to be altered based on expressed and received conversational expressions.

Dyadic conversations play an important role in social cognitive development [145]. Studies have looked at viewing the interactions as dyads, rather than simply one-sided speech, and more specifically, dominance, control, and coherence in dyadic interactions [148]. Things like gaze can affect the control or flow of a conversation [175].

While dialogue and vocal cues are an important part of conversational interactions, facial expressions have been shown to have a stronger impact on perceptions of trustworthiness [223].

In [86], the effect of image size on the perception of facial expressions was investigated. Using 9 conversational expressions for 6 human subjects, the authors conducted a psychophysical experiment by showing observers the expressions at varying sizes. The expressions shown were as follows: agreement, disagreement, disgust, thinking, pleased/happy, sadness, pleasantly surprised, clueless (as if the actor did not know the answer to a question), and confusion (as if the actor did not understand what was just said). The sizes shown were as follows (in pixels): 512×384, 256×192, 128×96, 64×48, 32×24, and 16×12. The researchers found that the expressions were able to be easily recognized all the way down to the 64×48 pixel size before recognition performance dropped significantly.

In [84, 85] the components of the face that are necessary for certain conversational expressions were examined. 6 human subjects were recorded performing 9 conversational expressions, using the *method acting* technique. These expressions were: agreement, disagreement, disgust, thinking, pleased/happy, sadness, pleasantly surprised, clueless (as if the actor did not know the answer to a question), and confusion (as if the actor did not understand what was just said). The original footage, along with five *freeze face* conditions were shown to participants in a perception study. The *freeze frame* condition consisted of freezing areas of the face, using the subject's neutral expression as the source. The regions of the face frozen were the eyes (direction of gaze and blinking), eye and eyebrow region, and the mouth region. In every instance, rigid head motion was left intact. This methodology allowed the researchers to examine the importance of certain regions of the face in the recognition of conversational facial expressions. There were 9 participants in the perception study. Their task was to identify the expressions, and rate both the believability and naturalness of the expressions. The rating scale ranged from 1 to 7. Participants were told to rate any expressions deemed to contain noticeable artefacts from the manipulation techniques as 'unnatural'.

The results showed that overall participants were able to recognize the expressions and that the expressions were for the most part believable and natural. They found that rigid head motion contained a lot of information about the expressions. They found the addition of eye motion improved accuracy in some conditions. Furthermore, the addition of eyebrow motion improved recognition accuracy even more. The mouth also contained much information regarding certain expressions, such as happiness.

Thinking relied heavily on eye motion and confusion was defined quite a bit by eyebrow motion. Happiness and pleasant surprise relied on mouth motion mainly, but the eyes and eyebrows were necessary for the accurate recognition for the pleasantly surprised expression. Sadness and disgust required all four types of motion (rigid head movement, eye, eyebrow, and mouth motion). These four types of motion have the ability to produce the movements necessary for accurate recognition of the conversational expressions examined in this work, as much so as the original recordings. They also found that the motion of other facial regions, such as the cheeks and forehead, seemed to have little to no impact on the recognition accuracy for the expressions examined in this study.

Cunningham et al. [87] conducted 5 experiments, with the purpose of proving the importance of dynamic information in the recognition of conversational expressions. This work is one of the most useful for understanding the importance of temporal dynamics with conversational expressions. As such, this work will be described in full detail.

The 9 expressions used were: agreement, disagreement, disgust, thinking, pleased/happy, sadness, pleasantly surprised, clueless (as if the actor did not know the answer to a question), and confusion (as if the actor did not understand what was just said).

The first experiment compared the peak versions of dynamic and static conversational expressions. In the dynamic condition, the expression was shown as a dynamic sequence (video) from the neutral expression to the peak frame. In the static condition, only the peak frame was shown (for the same amount of time as the dynamic video). 10 participants were asked to recognize the expressions in a 10-

alternative, non-forced-choice task (one option was *none of the above*). Dynamic expressions had an accuracy rate of 78%, while static expressions had an accuracy rate of 52%. Agreement and disagreement seemed to be the two expressions that relied heavily on the dynamic information for accurate recognition. Sadness, disgust, clueless, confused, and surprise also relied on dynamic information as well, but not to the extent of agreement and disagreement. Happy and thinking were the two expressions where static information displayed an advantage over dynamic information, in regards to recognition accuracy. There was an overall advantage with dynamic expressions over static expressions, as seen in the accuracy percentage, and the results suggest that there is some information present in the dynamic sequences that is not available in the static peak frame.

The second experiment attempted to show the importance of spatio-temporal information by scrambling the order of the frames in the dynamic expressions. It tested 2 hypotheses using 5 conditions. The first two are the same as experiment 1 (dynamic and static). The third condition shows the last 16 frames as a video (*dynamic 16*). The fourth condition shows 16 frames in a 4x4 grid, in order, that is, first frame and so forth from left-to-right, top-to-bottom (*static 16 ordered*). The fifth and final condition is similar to the fourth condition, however there is no order (*static 16 scrambled*).

9 new participants were tasked with recognizing the conversational expressions. The two dynamic conditions produced higher recognition rates than the three static conditions. Dynamic conditions had a 76% accuracy rate, while static conditions only had a 52% accuracy rate. The dynamic-or-static advantages found in experiment 1 were mirrored in experiment 2. The experiment also showed that dynamic information, in regards to expression recognition accuracy, is not simply looking at a set of static images, mentally converting the spatial order of the frames into the proper temporal sequence, and then inferring the critical dynamic information.

Experiment 3 attempted to show that there is information present in the normal temporal development of an expression. That is to say, it is not simply the motion that

is responsible for the increased recognition rates, but that there is some information present in the dynamic condition that is the cause. For the experiment the frames were shown as a video, but in a random order (*scrambled condition*). The same frames were shown in each condition for the same amount of time (scrambled dynamic condition and regular dynamic condition).

Ten participants were tasked with recognizing the conversational expressions. The regular dynamic condition had an accuracy rate of 76%, while the scrambled dynamic condition had an accuracy rate of 56%. Agreement, disagreement, and surprise relied heavily on the dynamic information, while the other expressions only relied somewhat on the dynamic information. This experiment, along with experiments 1 and 2, would seem to indicate that the advantage of dynamic information is not due to the presence of multiple images nor the presence of face-related motion signals.

Experiment 4 examined the effect playing an expression backward has on the recognition of conversational expressions. Ten participants were used for recognizing the conversational expressions. The forward sequences resulted in a slightly higher recognition rate than the backward sequences: 79% to 72% recognition accuracy, respectively. This experiment supports the authors' statement that the dynamic advantage is due, at least in part, to some form of characteristic dynamic information.

The fifth, and final experiment estimated the length of the temporal integration window. That is, how recognition accuracy changed over the amount of time the expression was visible. One dynamic condition (which had been used in the previous experiments) and four scrambled dynamic conditions were used in this experiment. The first scrambled condition was identical to that of experiment 3, with a single frame being the preserved unit (*scrambled 1*). For the second scrambled condition the preserved unit was subsequent image pairs (e.g. frames 1 and 2, 3 and 4, etc.) (*Scrambled 2*). The order of the pairs was randomized. For the third scrambled condition the preserved unit was 4 subsequent frames (e.g. frames 1, 2, 3, and 4; frames 5, 6, 7, and 8; etc.) (*Scrambled 4*). In the fourth scrambled condition the preserved unit was 6 sequences (*scrambled 6*).

Overall, as the number of frames kept intact increased, the recognition accuracy also increased. The results for the dynamic and scrambled 1 conditions were similar to that found in experiment 3 (Experiment 5: 73% and 47%; Experiment 3: 76% and 56%). The integration window was determined to be at least 4 frames (100 ms) long. This work showed that dynamic information is important for the accurate recognition of conversational expressions, and the advantages of dynamic information, including its robustness, over static information cannot be accounted for by simple, static-based explanations.

2.4 Facial Expression Databases

In order to conduct research in facial expression analysis, recognition, modelling, and synthesis it is important to have a large catalogue of facial data with which to work. Unfortunately, capturing data is time-consuming, expensive, and with constantly changing technology, challenging. It is of little surprise then that most of the research conducted in the area of facial recognition and facial expressions over the past 10 years has occurred using a few popular databases. One of the best known 2-D databases is the Cohn-Kanade (CK) database and Extended Cohn-Kanade (CK+) database [135, 158]. It is popular because of its substantial number of subjects, its consistent data capturing techniques, and because it is FACS coded. The Bosphorus 3D database [203, 204] is a very popular 3D database for similar reasons. It has 2D and 3D scans available and also includes scans with occlusions, for researchers attempting to solve the problem of occlusions in facial data. There are relatively few 4D (or "3D Dynamic" as it is also referred to) publicly available facial expression databases. A relatively up-to-date survey of 3D/4D databases can be found in [202]. One good example that has focused on basic expressions, basic articulations, and varying speech pronunciation intensities is the ADSIP facial database [165]. Cosker et al. [80] provides the first 4D database of FACS-coded expression sequences. One of the most recent and popular 4D databases of spontaneous, basic facial expressions can be found in [253].

Databases	Format	Images/Sequences	Expressions	Colour/Gray	Resolution	Subjects	Year
FERET	2-D Static	14,051	2	Gray	256x384	1199	1996
CMU-PIE	2-D Static	41,367	4	Colour	384x286	68	2000
Multi-PIE	2-D Static	750,000	4	Colour	3072x2048	337	2009
MMI	2-D Static	200	6	Colour	720x576	52	2005
JAFFE	2-D Static	213	7	Gray	256x256	10	1998
Cohn-Kanade	2-D Dynamic	486	6	Gray	640x490	97	2000
MPI	2-D Dynamic	60	4	Colour	450x400	8	2003
DaFEx	2-D Dynamic	1,008	6	Colour	360x288	8	2005
FG-NET	2-D Dynamic	399	6	Colour	320x240	18	2006
BU-3DFE	3-D Static	2,500	7	Colour	1040x1329	100	2006
Bosphorus	3-D Static	4,666	6	Colour	1600x1200	105	2008
ZJU-3DFE	3-D Static	360	4	Colour	-	40	2006
ADSIP	3-D Dynamic	210	7	Colour	601x549	10	2009
BU-4DFE	3-D Dynamic	606	6	Colour	1024x681	101	2008
Hi4D-ADSIP	3-D Dynamic	3360	14	Colour	2352x1728	80	2011

Table 2.1: Table of well-known Facial Expression Databases [165]

While some conversational databases exist (e.g. [89, 166, 169, 177, 235, 236]), the general lack of interaction between participants, lack of 4D data, or poor visibility of the face make these unsuitable for our research. In [89], pre-defined speaker/listener roles are assigned, which constrains the naturalness of the conversation. In [166], one side of the conversation contains an operator-controlled synthesised face. In [169, 235] the subjects are often too far from the camera for the face to be visible. While [192] is a multi-modal dyadic behaviour database, it is between adult confederates and toddlers. Finally, the works of [177, 236] focus more on the gestures and body movement than the facial expressions of the individuals in the conversations.

While the 2D Cardiff Conversation Database (CCDb) [22] does contain audiovisual, annotated, natural dyadic conversations, only the 2D data has been processed and made available.

One major contribution of this thesis is the development of the first 4D database of natural, dyadic conversations [229]. This publicly available audio-visual database contains 2D videos, 3D dynamic data, and is annotated for frontchannel and backchannel signals, which include conversational facial expressions, head motion, and verbal/non-verbal utterances. A baseline experiment classifying frontchannel and backchannel smile interactions was performed. This database is useful for analysing, modelling, and synthesising facial expressions as well as for analysing and modelling human conversations. Details of this database can be found in Chapter 3.

2.4.1 2D Databases

The Cohn-Kanade AU-Coded Facial Expression Database (CK) was compiled in an effort to advance the field of facial expression analysis [135]. The CK database contains a substantial number of images with a range of expressions and uses FACS to define those expressions. Certified FACS coders were used to code sequences in the database. 201 participants ranging from 18 to 50 years old, with an ethnic distribution of 81% Euro-American, 13% African-American, and 6% other, took part in the original data collection process. 69% of the subjects were female.

The initial release of the CK database contained 486 FACS-coded sequences across 97 subjects. A sequence included all frames from the neutral expression to the peak expression. Each peak expression frame was FACS AU-coded. Emotion labels were added, but not validated. The 97 subjects were university students ranging from 18 to 30 years old. 65% of the participants were female. Each participant was asked to perform 23 different facial expressions. These expressions included single action unit movements, such as AU 12 for lip corner puller, and also combinations of action unit movements, such as AU 1 + 2 for inner and outer brow raiser. Full details about the data collection process for the full database can be found in [135].

Figure 2.2: Example of an image from the CK+ Database [158]

Although the Cohn-Kanade database is one of the most used face databases in the

world, facial expression recognition and related areas of research have advanced substantially since the original release of the database. As a result, three main limitations of the CK database became apparent. The first limitation was that the original emotion labels were not properly validated. The expression labels were those requested of the subject, not the actual expression performed. The second shortcoming was the lack of a common performance metric. It was difficult for researchers to evaluate new algorithms with no existing benchmark with which to compare. The third limitation was that to make quantitative meta-analysis possible a standard protocol for common databases was needed. These three issues were addressed in the release of Version 2, also known as the Extended Cohn-Kanade Database or CK+ [158].

Version 2, also known as CK+, added 107 new sequences from 26 new subjects, bringing the total for the Cohn-Kanade database (CK and CK+) to 593 sequences across 123 subjects. Action Unit labels and emotion labels have been revised and validated. Non-posed smiles and related meta-data have also been added. Full details about the CK+ database and related studies can be found in [158].

2.4.2 3D Databases

The Bosphorus database, developed at Bŏgaziçi University in Istanbul, Turkey, is a FACS AU-coded 3D facial database consisting of various facial expressions, facial occlusions, and head poses [203]. Corresponding 2D high-resolution colour images are also available. 20 lower face action units, 5 upper face, and 3 combinations of FACS action units were used for the database, and for Version 2, six prototypical emotions of happiness, sadness, anger, fear, disgust, and surprise were captured. The current database consists of 105 subjects, with 4666 face images in a variety of poses, expressions, and occlusion conditions.

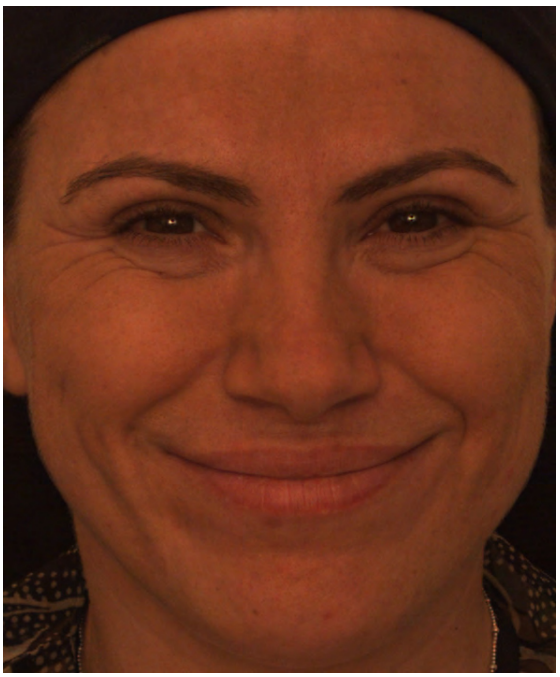


Figure 2.3: Example of an image from the Bosphorus Database [204]

2.4.3 4D Databases

There have been quite a few 4D databases released in recent years. There are three, however, that have become popular due to the number of examples, quality of scans, and variety of posed and spontaneous expressions captured.

The Hi4D-ADSIP database [165], is an extension to the ADSIP (Applied Digital Signal and Image Processing) facial database [111]. The additions include 7 basic expressions (anger, disgust, fear, happiness, sadness, surprise, and pain), basic-articulations including mouth and eyebrow motion, and phrasing being read at 3 different pronunciation intensities. All expressions were posed rather than spontaneous. Expressions of 80 subjects were captured on a Dimensional Imaging system which consisted of 6 cameras. These cameras captured images at a resolution of 2352×1728, at 60fps. Basic expressions had a duration of approximately 3 seconds, basic articulations lasted approximately 6 seconds, and phrase reading sequences were around 10 seconds. The human evaluation approach used data of 12 actors posing 7 expressions at 3 different intensity levels: mild, normal, and extreme. The normal intensity level averaged better confidence scores, and happiness expressions were given the highest confidence scores. Anger-disgust, as well as fear-surprise, were

often dyads that were mistaken for one another. The main short-comings of this work were that untrained human observers were used for validation, the data consists of posed expressions, the approaches used are outdated (Eigenfaces/Fisherfaces), and, perhaps as a result, they produced relatively low recognition rates.

In [80], Cosker et al. introduce the first 4D (3D video) database of FACS-coded expression sequences. 10 subjects performed between 19 and 97 different action units (individually and in combination) for a total of 519 AU sequences. Each expression sequence was on average 90 frames long and the peak expression frame was manually FACS-coded by a FACS-certified coder. During capture each subject had the use of a mirror to practice their movements. Of the 10 subjects, 4 were FACS-certified coders and the remaining were FACS-untrained participants.

The other major contribution of the paper, apart from the 4D FACS-coded data, was the introduction of a novel approach for tracking and registering 3D sequence data. While optical flow is a popular technique for tracking 3D sequences using multiple 2D stereo images from capture systems [49, 54, 251], the approach is very prone to drift due to small errors in the tracking that accumulate over time. An AAM and TPS based approach (AAM+TPS) was introduced by the authors for tracking and dense registration of sequence data. This AAM+TPS approach, like the optical flow-based techniques, treats the 3D correspondence issue as a 2D image registration problem. The 2D stereo views are joined to create a single-view uv texture map, as seen in the bottom row of Figure 2.4).

A mapping between this single view texture map and the 3D mesh is computed. This mapping allows any operations performed on the 2D texture map to be performed on the corresponding 3D mesh as well. The authors evaluated their AAM+TPS tracking and registration approach against optical flow by using the well-known pyramidal Lukas-Kanade algorithm [157]. 47 landmarked points around the eyes, nose and mouth were used for the evaluation experiment. Every frame in a sequence was manually annotated and the AAM was trained using three frames; typically the first, middle, and last frame of the sequence. Ground truth manually landmarked

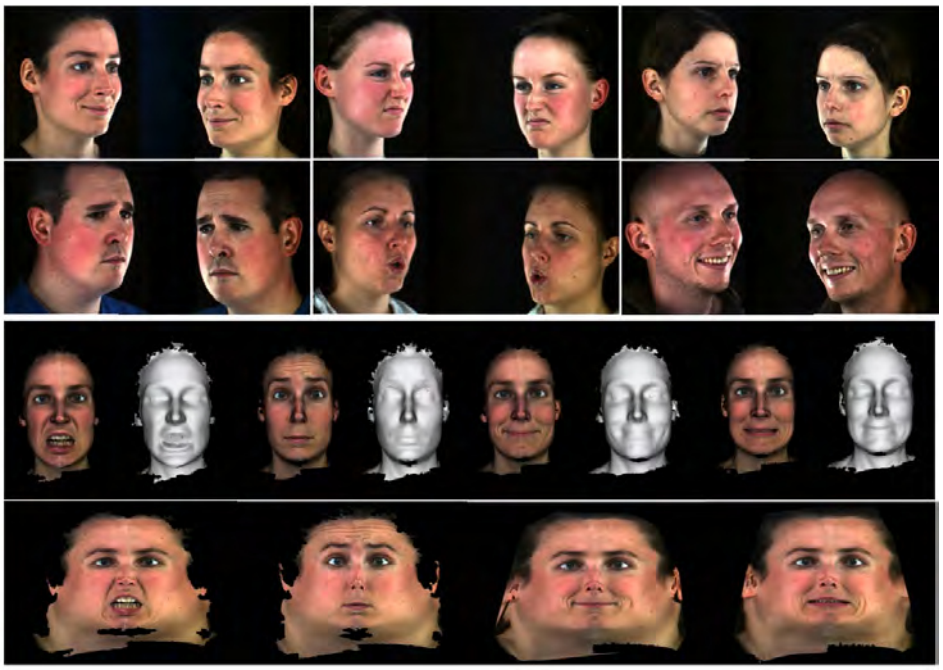


Figure 2.4: Row 1 & 2: Camera views of the 6 participants. Row 3: 3D mesh data. Row 4: Single-View *uv* texture maps. [80]

points were used and Euclidean distance error between the tracked points and ground truths was computed for each sequence, for each approach (LK optical flow and AAM+TPS). The results support the authors' claim that their AAM+TPS technique is less prone to drift than an optical-flow based technique. To evaluate the qualitative aspects of their approach, the authors used the registered data to build 3D Morphable Models [45, 46] to re-synthesise the facial expression sequences. They provide a visual comparison between their approach and an LK and LK+TPS approach. It is clear in the examples shown that their approach produced the least amount of artefacts of the tested approaches. They also claim that in a related perceptual study [79] participants did not notice any issues with the models they viewed.

The BP4D-Spontaneous database [253] is a high-resolution database of basic facial expressions and contains 41 subjects (23 women, 18 men), ranging from 18-29 years old, and from various ethnic backgrounds (11 Asian, 6 African-American, 4 Hispanic, 20 Euro-American). Subjects were captured using a Dimensional Imaging passive stereo capture system which captured at 25fps. The 2D texture images were 1040×1392 pixels and the 3D frames produced consisted of 30,000-50,000 vertices. Subjects were required to take part in eight tasks, each intended to elicit spontaneous

emotions. These tasks included listening to a joke (laughter), watching a documentary (sadness), hearing a sudden, unexpected sound (surprise), a game of improvising a "silly" song (embarrassment), anticipation of a physical threat (Nervous/Fear), submerging their hand in ice water (physical pain), experiencing harsh insults from an interviewer (Anger/Upset), and an unpleasant smell (disgust). There were a total of 328 sequences captured. The expression sequences were manually FACS AU coded for 27 action units by two certified coders. A cylindrical head tracker was used on the 2D videos for tracking rigid head motion. 83 facial feature landmarks were tracked using a 2D Constrained Local Model (CLM) [83] for the 2D videos and a 3D Temporal Deformable Shape Model (TDSM) for the 3D sequences. A Hidden Markov Model (HMM) [33, 34] was used to classify the 6 prototypical emotion expressions (happiness, sadness, surprise, fear, anger, disgust), for each subject (of 16). This classifier used both spatial and temporal features of the expression sequences. 14 subjects were used in the training set and 2 for testing. Average classification accuracy was 70.2%. The authors claim spontaneous expressions are more difficult to automatically classify, referencing their posed database which used the same classification approach and achieved classification accuracy of 80%.

Figure 2.5: 3D Sequence from the BP4D-Spontaneous Database [253]

These publicly available database are the state-of-the-art for 4D posed and spontaneous facial expressions. These databases provide high quality data, with a relatively high number of participants and scans of 4D data. They provide some or all of the following: 2D videos, 3D sequences, head pose information, 2D/3D tracked features, and FACS AU-coded expressions. These databases will allow for a variety of interesting applications in computer science, HCI, Psychology, and other fields. However, they lack the everyday facial expressions of dyadic, face-to-face conversations. It is for this reason we chose not to use these databases for the our research. Instead, we have developed our own 4D database of natural, dyadic conversations. This publicly

available database [229] allows for analysis and model building of conversational facial expressions and expression interactions. Details of this database can be found in Chapter 3.

2.5 Facial Analysis Methods

One of the main challenges to using sequence data, instead of simple still images, is identifying corresponding facial feature points and where they are located in each frame. In a simple capture, such as a smile, the location of the lip corners will change throughout the sequence. This can be further complicated if there is rigid head movement, such as slightly nodding the head as many people do when they laugh. For this reason many *tracking* methods have been developed. Through typically manual or semi-automatic processes, feature points are identified for each frame in a sequence. These points can be used in a variety of ways, such as for analysing low-resolution to using the sparse feature points as anchor points for a dense registration method. Creating corresponding data across a sequence is also vital for building statistical models.

In the following sections, 4D tracking and registration methods will be discussed, along with statistical modelling approaches which use such corresponding data.

2.6 Feature Point Correspondence

In order to analyse facial expressions and create statistical models for 4D facial expression data, feature points for each frame are needed. These feature points provide a sparse correspondence between frames and can assist dense registration techniques. Rather than annotate each frame individually, which would be very time consuming and labour intensive, it is beneficial to automate this process. While many strides in 2D tracking have occurred over the past five years, 4D tracking and registration is still very much an open area of research. Most approaches devolve the

4D tracking and registration problem down to a 2D problem. While this may make solving the problem easier, it often neglects using the rich information provided with 3D data.

Therefore, a semi-automatic 4D tracking and registration approach was developed and is described in this work (Chapter 5) that retains the 3D information for each frame and results in high-resolution, accurate, inter-subject densely registered 3D frames.

2.6.1 Tracking Feature Points and Dense Mesh Registration

Many 2D tracking and registration approaches currently exist for feature point detection and tracking [83, 147, 219]. Others have approached tracking and registration of 3D data by reducing it to a 2D problem.

In [38, 54], a 2D image-based tracking and registration approach is used on their 3D, passive stereo data. This is a popular approach, and is used in other works [41], as it does not require active stereo or marker-based capturing techniques. In [242], tracking of 3D frames is done using harmonic maps. This reduces the 3D tracking and registration problem to a 2D image-matching problem. These approaches have limitations in both the quality of the tracking and the quality of the registration. This is due to these methods not using the original 3D shape and texture information. In [80], a novel tracking and registration approach is described (AAM+TPS) which makes use of a 2D, single-view texture map. The tracking approach described does not require physical markers and is less prone to drift than previous techniques, however their evaluation was only tested on relatively short sequences (roughly 65 frames/1 second long). As well, the visual output from their 3DMM contained less artefacts than the works they compared against, but it still produced meshes that differed noticeably from the original (pre-registered) in certain areas of the face, specifically the mouth region.

In [23], a Constrained Local Model approach is used for tracking 3D sequences of faces

and utilises a non-rigid head pose tracker that is aided by a rigid head pose tracker (Generalised Adaptive View-Based Appearance Model (GAVAM)). This approach uses both intensity and depth information to track facial features in 3D frames and is referred to as *CLM-Z*. Unfortunately, this approach requires either a system that captures depth information (e.g. Microsoft Kinect) or manually annotated depth data. The former currently does not provide high-resolution 4D meshes, and the latter is time and resource intensive.

Huang et al. [129] performed 3D tracking by aligning each frame with a generic 3D mesh model and using a hierarchical tracking scheme. By forcing the 3D frames onto a generic template mesh, however, the approach loses accuracy in the facial shape and movements, along with quality of the face image.

Sidorov et al. [210, 211] proposed a novel method for performing non-rigid groupwise registration of textured surfaces. This method is fully automatic and described as being efficient and reliable. The method can be used to build 3D models of appearance and is robust enough to even handle inter-subject registration. The author's approach is that the deformation model is defined on the original meshes and so is explicitly aware of the 3D geometry. The main idea is to maintain the correspondences between surfaces and to operate with textures in a common flat reference space while performing optimisation on the original 3D surfaces. This technique is, however, limited to diffeomorphic deformations of the face (such as requiring a closed mouth).

Another popular approach for dense registration is using Thin Plate Splines (TPS) [48, 99, 205, 206]. TPS is a technique for data interpolation and is used in various applications for fitting one surface to another. In [123], TPS is used to create a dense correspondence across 3D face scans, in order to analyse facial morphology. In a more recent work [124], it was used to register scans for modelling and analysing individuals with medical conditions. The strength of a TPS-based approach for dense registration over the groupwise technique described previously is that it is not limited to only diffeomorphic deformations. This added robustness is useful for inter-subject

registration of frames which include a variety of face expression dynamics.

A comprehensive survey of 2D and 3D databases and affect recognition approaches can be found in [250]. To the best of our knowledge, none of these approaches provide robust 4D inter-subject registration, which allows for high-resolution, realistic facial expression synthesis.

2.7 Statistical Modelling

2.7.1 Principal Component Analysis

Principal Component Analysis (PCA) was developed by Karl Pearson in 1901 [183] and is a popular technique for reducing dimensionality in data using orthogonal transformation. Data is converted from potentially correlated variables to a set of uncorrelated variables, called *principal components*, that represent the level of variance in the data. Principal components are ordered by decreasing level of variance [212]. This ordering allows the number of principal components to be truncated, while still maintaining the most significant information. Each principal component removed represents the removal of a single dimension. Reducing data dimensionality, and thus complexity, while retaining necessary information is one of the main reasons PCA is a popular technique.

2.7.2 Point Distribution Model

A Point Distribution Model (PDM) is developed using landmarked images from a training set [75]. The points in the landmarked image compose the shape of an object. Procrustes analysis is used to align the points of each image in the training set in a coordinate frame, with each image's points being represented by a vector [73]. For each point cluster, the average is computed and results in an average shape and, together, an average shape model of the PCA.

Figure 2.6: The PDM, the average point for each cluster, and the resulting average shape [232]

2.7.3 Active Appearance Models

Active Appearance Models (AAMs) is a PCA-based statistical approach to modeling appearance developed by Cootes, Taylor, and Edwards [72] in the late 1990's. AAMs build on the idea of the Active Shape Models (ASMs), which is a model-based statistical approach to locating shapes in an image [71]. Active Appearance Models use annotated images where key areas are labelled using landmark points. Landmark schemes vary by the number and location of points. For a specific AAM the landmark scheme used should remain consistent across all images in the dataset. Active Appearance Models include both shape and texture information by combining a model of shape variation with a model of texture variation. Texture refers to the pattern of intensities or the colours in an image [73].

Once the PDM has been created, each image in the training set is warped to the average shape. Using triangulation on both a training image and the average shape model, where each triangle in the original training image corresponds to a triangle in the average shape model, the original image is warped to the average shape using piecewise affine warping [164]. This results in a *shape-free* texture image. Once this has been completed for all of the images in the training set, it is easy to calculate the average texture across the training set, resulting in a texture model. PCA is then performed on the shape and texture models. The combined appearance model is computed by concatenating the resulting shape and texture vectors, and performing

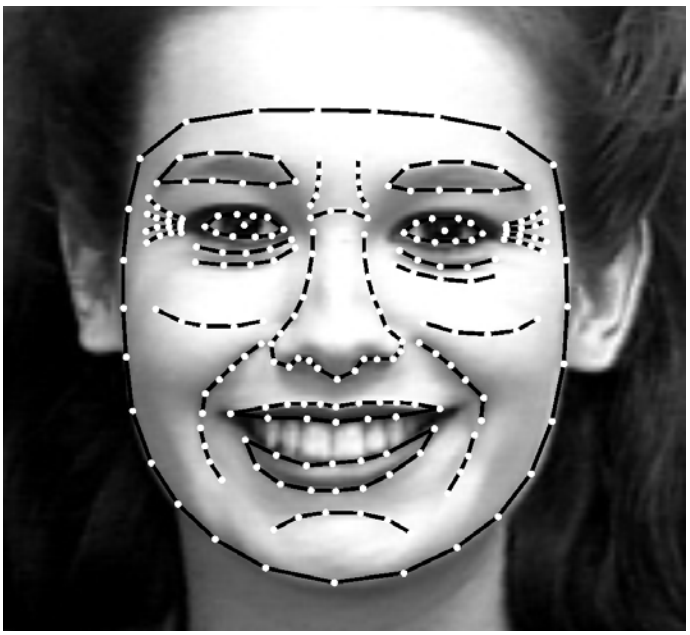


Figure 2.7: Example of a 227-point landmark scheme using an image from the CK+ database [158]

PCA a final time to account for possible shape-texture correlations.

The combined (shape and appearance) model can be used to interpret a previously unseen image. A test image can be explained statistically by its relation to the combined model in a vector of feature weights. Feature weights define a test image in terms of features existent in the combined model. If the features of a test image are well-represented by the training set, a test image can be accurately expressed. That is, any feature that exists in the training set can be represented statistically. For faces, this means that even if an individual with a specific combination of features does not exist in the training set, a test image can be faithfully expressed statistically as long as those features exist across the training set.

Now that the various databases, feature point correspondence, and statistical modelling approaches have been described, the various works which utilise these types of approaches may be discussed.

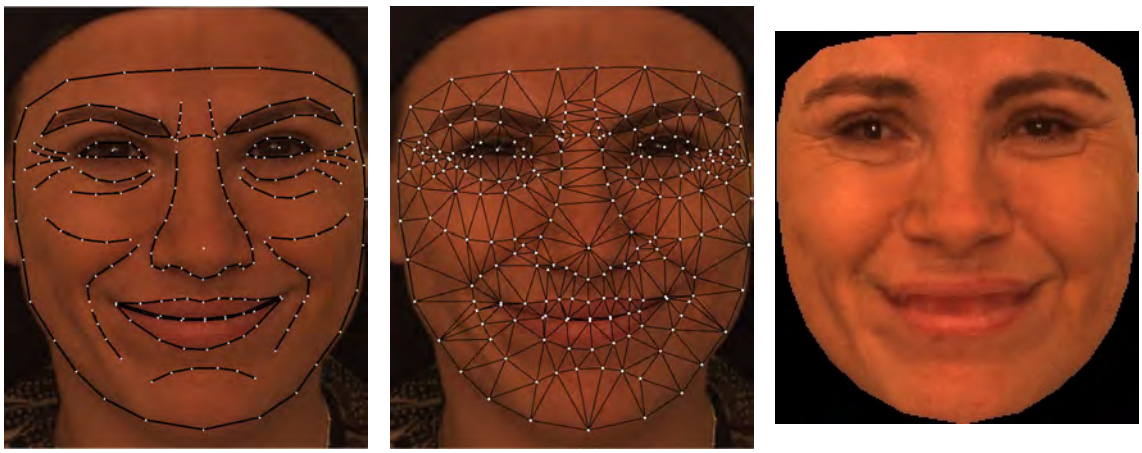


Figure 2.8: A landmarked image from the Bosphorus database, the corresponding triangulation, and the shape-free texture [204]

2.8 Automatic Extraction, Recognition, and Classification of Facial Expressions

Much focus over the past 10 years has been given to the automatic extraction and recognition of facial expressions. Many different extraction and classification techniques have been developed and modified over the years to increase the recognition of facial expressions. Some have focused on recognising, in real-time, the 7 "prototypical emotions", while others have focused on identifying expressions in a variety of environments, including those with lighting variations and facial occlusion issues.

2.8.1 Action Unit Detection and Classification

Bartlett et al. [31, 51] used feature extraction techniques such as Gabor filters, Support Vector Machines (SVMs), and Hidden Markov Models (HMMs) to automatically FACS AU-code facial expressions. In one study they attempted to automatically code 12 (6 upper face and 6 lower face) action units for posed facial expressions for 20 subjects. Gabor Wavelet decomposition and independent component analysis performed best with a 95.5% accuracy rate over five other approaches, which were optical flow (85.6%), local feature analysis (81.1%), and eigenfaces (79.3%), Fisher's linear discriminant (75.7%), and explicit features (wrinkles) (57.1%). Another study

involved 17 subjects in a high-stake mock crime experiment conducted by Mark Frank and Paul Ekman. To address out-of-plane head rotation, which is a common element of spontaneous expressions, they used deformable 3D models to warp a subject's face to a canonical face geometry. Blinks (AU 45), brow raiser (AU 1 + 2), and brow lowerer (AU 4) were chosen as the three actions to recognize. Best results were achieved using HMMs trained on the outputs of the SVM for Blink versus Non-Blink actions, with an accuracy rate of 98.2%.

In a later study by Bartlett et al. [28, 29], a real-time automatic classification system for both posed and spontaneous facial expressions was evaluated. The image databases used were RU-FACS from Rutgers University, the Cohn-Kanade database, and a database of images collected by Ekman and Hager (Ekman-Hager Database). RU-FACS contains FACS AU-coded spontaneous facial expressions from 100 subjects. The Cohn-Kanade and Ekman-Hager databases both contain FACS AU-coded posed facial expressions, and for this study 119 subjects were included. The technique that provided the best results, in both speed and accuracy, was the method of selecting a subset of Gabor filters using Adaptive Boosting (Adaboost) and using those outputs to train SVMs. Automatic FACS labeling for spontaneous expressions and posed expressions achieved 90.5% and 94.8% agreement with human FACS codes, respectively. Recognition of full facial expressions of emotion in a 7-way forced choice achieved 93.3% accuracy.

The Computer Expression Recognition Toolbox (CERT) [26, 27, 150] is a real-time, fully automated software tool that uses appearance-based discriminative approaches, such as Gabor filters and SVMs, to code real-time video with respect to "40 continuous dimensions, including basic expressions of anger, disgust, fear, joy, sadness, contempt, a continuous measure of head pose (yaw, pitch, and roll), as well as 30 facial action units (AU's) from the Facial Action Coding System". Although CERT is a relatively new tool it has already been used in a variety of facial expression recognition studies including drowsiness detection, affect-sensitive adaptive tutoring, and games for children with *Autism Spectrum Disorder* (ASD) [57, 69, 153, 237, 244].

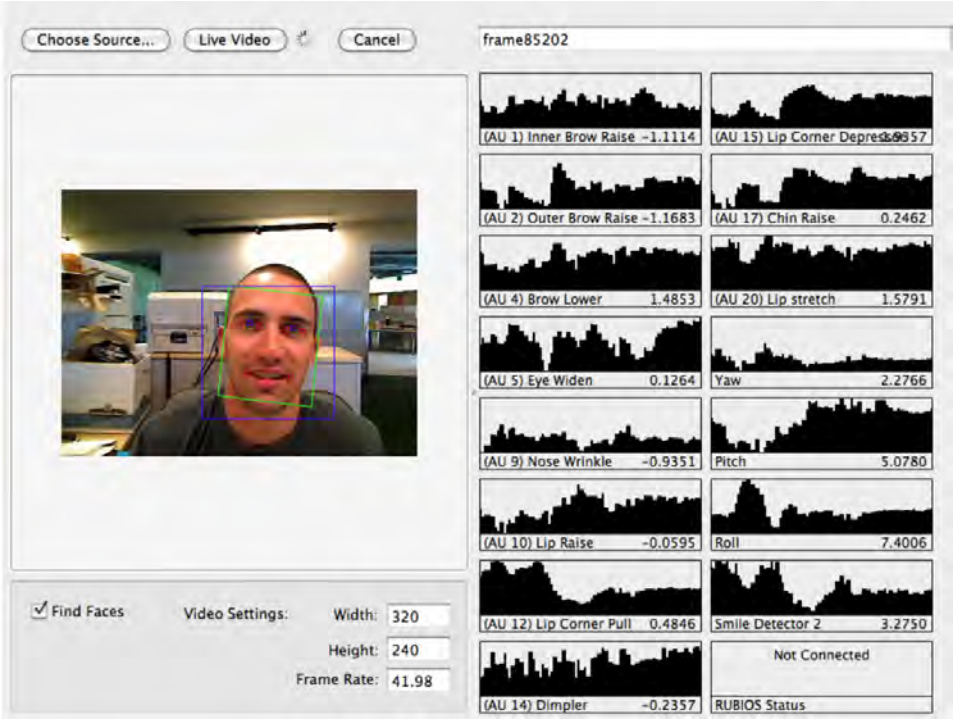


Figure 2.9: A Screenshot of the CERT Software [150]

CERT was used in a study by Littlewort et al. [151] to distinguish real versus faked expressions of pain. 48 subjects took part in an experiment which used cold pressor pain to induce real expressions of pain. A subject's forearm was submerged in 5°C ice water to induce real pain expressions. The water was 20°C for baseline and faked pain conditions. For the faked pain portion of the experiment, subjects were asked to produce facial expressions that would convince an expert they were genuinely experiencing pain. To test human accuracy in differentiating real from faked expression of pain, 170 naïve observers viewed videos of the experiment and were asked to choose whether the subject's pain expression was real or faked. Naïve observer mean accuracy was 52%. The automatic system used a two-stage process to differentiate real from faked pain. It first automatically detected facial actions and then provided that data to a machine learning classifier. A few meaningful differences in action unit activity and intensity for real versus faked pain were observed, including more brow lowerer (AU 4), cheek raiser (AU 6), and inner brow raiser (AU 1) activity during faked pain conditions. The automatic system achieved .72 area under the ROC curve, which is equivalent to 72% accuracy on a 2-alternative forced choice of fake versus real pain. In a later study by the same authors [152],

naïve human observer mean accuracy was 49.1% with a standard deviation of 13.7%. The automatic system achieved 88% accuracy.

Lucey et al. [159] performed experiments in an attempt to recognize facial action units for posed and spontaneous facial expressions. The Cohn-Kanade dataset was used for posed expressions and the RU-FACS dataset was used for spontaneous expressions. The experiment used AAMs for feature extraction and SVMs for classification. The use of a Nearest Neighbor (NN) classifier based on either PCA or LDA subspaces was considered, but preliminary tests showed it offered no advantage over the SVM approach. Results for posed action unit recognition showed an improvement over previous research. Most notably, this study's approach of using Active Appearance Models outperformed the approach of using Gabor filters with AdaBoost from Bartlett et al. [30], when using the same database and set of action units. Results from the spontaneous expression experiment were poor, mainly due to the problems resulting from the amount of head movement in subjects. Complete details, including statistical analysis of each experiment, can be found in [159].

Using 364 sequences across 94 subjects from the Cohn-Kanade database, Gonzalez et al. [119] developed a system to detect individual lower facial action units using a geometry-based approach. The shape model used consisted of 83 facial feature points. AdaBoost and overlapping coefficients (OVL) were used for feature extraction and AdaBoost and SVMs were used for classification (AdaBoost is both a feature selection and classification technique). An average accuracy of 94.55% was achieved.

Lucey et al. [158] added to the Cohn-Kanade Database in order to address a few limitations. One of these limitations was the lack of a common performance metric, or benchmark, to which other researchers could compare their own algorithms. Therefore, the authors conducted their own experiment using the new data. They used Active Appearance Models to track faces and extract features. A linear one-vs-all, two-class SVM was used to detect the presence of each AU. 17 AU's were included in this study along with seven stereotypical emotions, as defined by FACS. These emotions are anger, contempt, disgust, fear, happiness, sadness, and surprise.

The authors used a 3-step process for selecting the appropriate emotion label for each peak frame in a sequence. The first step took a strict approach by evaluating a sequence by the action units present, based on the Emotion Prediction Table from the FACS manual [104]. The Emotion Prediction Table lists the action units required for each prototypical emotion and the major variants of each emotion. If a sequence satisfied these requirements for any emotion, it was provisionally given that emotion label. The first step in the selection process was considered "strict" because the presence of, or lack of, any action unit not included in the table would result in the clip being excluded. The second step took a more loose approach by allowing action units not included in the prototypical emotions, or major variants of the emotions, if they were consistent with the emotion displayed. For instance, AU 4 is consistent with negative emotions such as anger, but not with positive emotions such as happiness. The third step was a visual evaluation of a sequence to determine if the expression resembled the targeted emotion category. 327 of the 593 sequences met criteria for one of the seven prototypical expressions. Complete details about the selection process can be found in [158].

AU detection and expression detection were the two experiments conducted on the CK+ database. Leave-one-subject-out-cross-validation resulted in 123 different training and testing sets for AU detection and 118 different training and testing sets for emotion detection.

Similarity-normalized shape (SPTS) and canonical-normalized appearance (CAPP) were derived once the shape and appearance AAM parameters were computed. SPTS refers to the shape information of an image and CAPP refers to the texture information. SPTS performed well for expressions that caused distinct changes in the shape of the AAM mesh, such as disgust and happiness. Conversely, CAPP performed well for expressions that caused texture changes, such as anger and sadness. The combination of features (SPTS + CAPP) produced the best recognition rates for all emotions except surprise. These recognition rates were Anger - 75%, Disgust - 94.7%, Fear - 65.2%, Happiness - 100%, Sadness - 68%, and Contempt - 84.4%. SPTS achieved a 100% recognition rate for surprise.

2.8.2 Expression Recognition

Valstar et al. [227] examined spontaneous versus posed facial expressions by developing a geometry-based system capable of automatic analysis of brow actions. To automatically distinguish posed from spontaneous brow actions they focused on the temporal dynamics of the brow actions. Temporal dynamics refers to the neutral, onset, apex, and offset stage of facial actions. Brow actions were chosen because the brow is active in a variety of facial expressions. A shape model consisting of 8 feature points representing the eyebrow, eye, and nostril regions of the face was used in the experiment. FACS action units 1, 2, and 4 were represented by the locations of these points. The MMI Facial database, Cohn-Kanade Facial database and DS118 dataset were used in this study [179, 195, 226]. The MMI Facial Expression and Cohn-Kanade Facial Expression databases consist of 123 posed facial expressions and the DS118 dataset consists of 139 samples of spontaneous facial expressions. Gentle Boost and SVMs were used to detect AU activity and Relevance Vector Machines (RVM) was used to classify the data. A correct classification rate of 90.7% was achieved during testing across the three datasets.

A similar experiment was conducted by Sebe et al. [208] using a variety of classification techniques such as Bayesian networks, decision trees, SVMs, and k-Nearest Neighbor for affect recognition.

Papatheodorou and Rueckert [180] defined "4D" as 3D shape plus texture (texture being the fourth dimension). Using low resolution scans, they attempted to recognise faces. The authors used Iterative Closest Point (ICP) for rigid alignment and 3D distances for determining face similarity. This simple approach is not very robust, but it was a decent step towards automatic face recognition using 3D/4D data. In [214], spatio-temporal HMMs take both spatial and temporal information into account when performing face recognition. The work focused on identifying the 6 prototypical expressions and at the time used a private, in-house database. The results showed improvement over a similar 2D approach. In [222], 3D geometry and 2D texture were used for identifying AU activations with a rule-based approach for

expression measurements.

Huang et al. [130] proposed a method for reconstructing human face models from 3D scan data. They present a new approach for 3D feature detection and a hybrid approach using two vertex mapping algorithms to reconstruct facial surface detail. They tested their new approach using the static 3DFE and dynamic 4DFE databases. 60 subjects made up the database used in the experiment. 50 subjects were used for training and 10 for testing. Six expressions were classified: anger, disgust, fear, happiness, sadness, and surprise. Using LDA to classify the dynamic facial expressions, a 20-fold cross-validation approach was used and resulted in an average recognition rate of 85.6%.

2.8.3 Expressions as a Biometric

Benedikt et al. [39] explored the idea of using facial emotions as a behavioural biometric using the Facial Action Coding System (FACS) for expression definition and Active Appearance Models (AAMs) for feature extraction. 3-D video data was collected over an extended time interval for both verbal and non-verbal facial actions. The experiment examined identification and verifications problems and results showed that while nonverbal facial actions (emotional expressions) were not sufficiently reliable for identity recognition, verbal facial actions (speech-related) showed potential for future use in biometric applications.

Patterson et al. [182] also explored the idea of using facial actions as a biometric. They chose to use a blink as the facial action, AAMs for feature extraction, and HMMs for classification. 23 landmark points marked the eyebrow, eye socket, and eyelid regions. Video data resulting in 1100 image sequences across 12 individuals was collected. They achieved 100% accuracy for all tests.

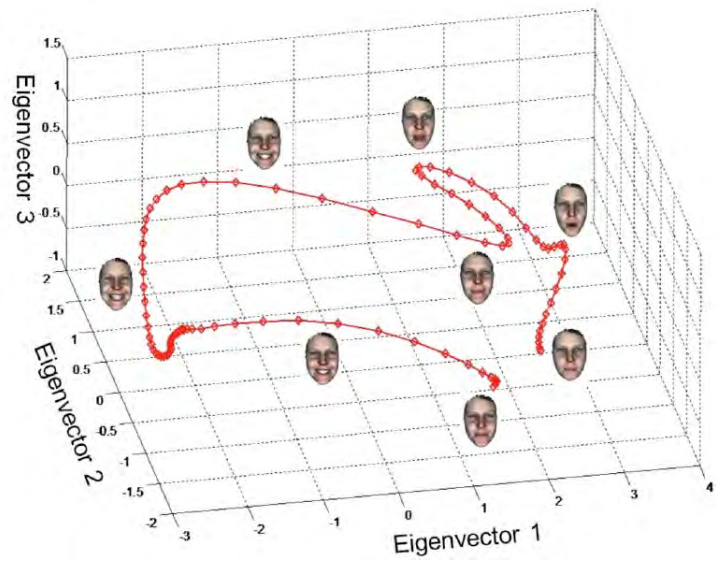


Figure 2.10: Smile dynamics of a subject plotted in the subspace spanned by the first three Eigenvectors [39]

2.8.4 Emotional Expression Detection and Classification

Ratliff [191] explored the use of AAMs and three popular classification techniques for recognizing facial emotional expressions. The classification techniques he employed include Euclidean distance, Gaussian Mixture Models (GMMs), and SVMs and the AAM landmark scheme used 113 points. The experiment used sequences from 18 subjects displaying emotions of fear, joy, surprise, anger, disgust, and sadness from the Face and Gesture Recognition Research Network (FG-Net) database. A neutral expression was also included. Euclidean distance measure, Gaussian Mixture Models (GMMs), and SVMs achieved a mean accuracy of 83.9%, 87.1%, and 91.3%, respectively.

Ashraf et al. [15] used the UNBC-McMaster shoulder pain expression archive in a study using AAMs for feature extraction and SVMs for classification, to differentiate real from faked expressions of pain. The study found that decoupling the face into separate non-rigid shape and appearance components increased performance for their study. Unlike Littlewort et al., they attempted to detect pain expressions not at the action unit level but rather from shape and appearance features. Many previous approaches applied techniques that used rigid registration, however, they claim, based on their findings, that rigid registration of appearance may not be necessary

for some applications.

Tam et al. [215] used interactive visualization for better understanding of facial dynamics data. Using visual analytics, the researchers were able to develop a decision tree as a classification algorithm, using time-series data in parameter space. The researchers took time-series data and used it to visualize the correlation between the algorithm space and classifying facial dynamics. They compared their approach with a well-known decision tree building tool and found it outperformed the tool, although the data set is small and future experiments are needed to validate the usefulness of their approach.

Fang et al. [107] introduced two facial expressions recognition frameworks for 3D and 4D data. The first is an improvement to the Annotated Face Model (AFM) in which AFM was improved by introducing Procrustes Analysis and Thin Plate Spline to the fitting process. The second is a fully automatic pipeline for classifying 3D/4D data. Two methods were used for finding vertex correspondences between two meshes. One method was based on spin image similarities, and the other on Euclidean distances between MeshHOG descriptors. Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) was used for expressions analysis. The pipeline framework was tested using the publicly available database BU-3DFE [247]. Support Vector Machines with a Radial Basis Function kernel was used for classification.

2.9 Dynamic Models of Facial Expressions

Building facial expression models allows researchers to better understand the intricacies of the relationships between different expressions across different subjects, through different analysis techniques. Facial expression models allow for a computer vision system to understand (classify) new, previously "unseen" (in the computer vision sense of the word) data. By understanding the relationships between data, it is possible to manipulate the data, in order to create synthetic, but realistic, facial expressions. These synthetic expressions can be used in a variety of applications,

most notably, in perceptual psychology experiments. The ability to build models of facial expressions from real-world data is relatively useless if we lack the ability to validate and, thus, understand the meanings of the facial expressions. Being able to manipulate models of real-world data is key to progressing the field of computer vision, in regards to recognising and understanding human facial expressions.

2.9.1 Model Manipulation

Theobald et al. [218] developed techniques for not only transferring, but manipulating, facial expressions from video sequences, from one individual to another. Active Appearance Models (AAM) is a popular feature extraction technique for reasons such as the ease of mapping model parameters from one face to another, and for its ability to account for a high degree of variability in data. Using AAMs, near-video realistic avatars can be synthesized with visual speech and expression information from other faces, without the need to record new subjects. The mapped expressions can be constrained so they best match the appearance of the target which is producing the expression and the model parameters can be easily manipulated to exaggerate or attenuate the mapped expressions.

The aim of the researchers was the create a real-time system where subjects could have a conversation with another subject via video conferencing software and the facial expressions of each could be manipulated in real-time without the knowledge of either subject. This allowed for perception studies in conversational situations. The process of capturing a video frame, extracting the AAM parameters automatically, applying the manipulation, re-rendering the face, and producing the manipulated face in the display takes approximately 150ms. The manipulations can be precisely controlled (e.g. smile exactly half as much). The authors first showed the copying of facial expressions from one face directly onto another face. This was a direct transfer of the expression model parameters, thus, even the source face's nasolabial furrows and mole on the cheek is transferred to the target faces.

The authors then showed the mapping of facial expressions from a source face

to a target face. In this, the expressions are mapped to the target face but the characteristics of the target are retained. In essence, it is the expression made by the source, but in the way the source would make the expression. The expressions can be constrained so that only realistic expressions are made, and has been determined to lie within 3 standard deviation from the mean.

Cosker et al. [78] used expression mapping techniques (also known as *performance-driven animation*) to create models for facial expression animation. The complexity of the face is difficult to capture and expression mapping helps to capture the subtleties of the face. The authors describe a method they created to facilitate re-mapping animation parameters between multiple types of facial models. A performance can be analysed in terms of parameter trajectories and then those trajectories can be used to animate other facial models.

Aubrey et al. [21] developed a novel approach for building and manipulating models of temporal dynamics. The goal was to be able to generate stimuli for perception studies involving trustworthiness. They used a modified version of Dynamic Time Warping (Graph-based Weighted Dynamic Time Warping) to bring the video sequences into alignment. The modified DTW algorithm augments the signal magnitudes by incorporating derivatives. It also includes a scheme for estimating weights in the cost function. Furthermore, the video sequences are used in the building of a graph where each node represents a sequence and each edge indicates the cost of applying the modified DTW to align pairs of sequences. This method can also be used to warp the audio of the sequences. Once aligned, a model of the dynamics can be built. This is used to identify and manipulate dynamics of interest (e.g. exaggerate or attenuate facial dynamics). Data capture consisted of 4 subjects speaking the phrase "once upon a time". 7 sequences of this sentence were recorded with each frame consisting of 49 landmarks. Groupwise registration was applied to align every image in a sequence. This results in a mean image. Using this mean image, each frame, and the pixels therein, can be described by their variation from the mean, creating the deformation map. The mean image is landmarked, and using the deformation map, the landmarks can be reverse-mapped onto the original frames. Thus making the

process a semi-automatic landmarking process. In the Active Appearance Model for each subject, 95% variance in the shape model and 99% variance in the appearance model were retained. Using these models the authors were able to manipulate the sentences spoken to elongate certain words. The manipulation of these temporal dynamics will serve as a core feature in the planned perception studies involving trustworthiness.

Boker et al. [47] examined the role of facial expressions and head movements in dyadic conversations. In the experiment, a confederate and a naive subject took part in a videoconference-style conversation. They were in two separate rooms, and were recorded by colour video cameras. Each subject viewed the other's face on a projection screen. The confederate's face was a cut-out of the face region only (i.e. no body, neck, hair) (Figure 2.11).

An AAM was built using 40-50 images of various facial expressions. The face being viewed by the naive participant was a synthesised avatar, which used Active Appearance Models for tracking and parameterisation. Facial expressions of the confederate were manipulated, and the effects of these manipulations were the main focus of the experiments in this work.

The naive participants only see a re-rendering of the original video that has been manipulated by attenuating or exaggerating the overall face by scaling the AAM parameters. This was done in "real-time", with a 165ms delay (opposed to the unmodified video feed from the naive participant to the confederate which is only 66ms). The confederates were informed of the purpose of the experiment and the type of manipulations, but had no knowledge of the order or timings of manipulations. This is important because as seen later, the way in which a naive observer reacts to the (modified) confederate can affect the confederate's actions. Obviously, the naive participants were completely unaware of the purpose of the experiment or manipulations. None of the naive participants noted anything unnatural about the videoconference; that is, they all believed the video feed was a normal, unmodified source.

Figure 2.11: Illustration of the videoconference paradigm. Top left: Video of the confederate. Top Right: AAM tracking of confederates expression. Bottom Left: AAM reconstruction that is viewed by the naive participant. Bottom Right: Video of the naive participant [47]

Three modifications were performed: head pitch and turn, facial expression attenuation, and voice frequency range modification. For the first, image coordinates for translation and rotation were attenuated from their canonical values by either 1.0 or .5. These same values were used to attenuate the facial expressions by multiplying them by the AAM shape parameters of the canonical expression. VoicePro was used to "either restrict or not restrict the range of the fundamental frequency of the voice". Figure 2.12 shows examples of these attenuated, synthesised faces.

27 naive participants and 6 confederates each took part in two, 8 minute, dyadic conversations (one with a male confederate, one with a female confederate). Head sensors were used to track and record head movements by both participants.

Results showed that head movement had significant effects on the participants and their behaviour. Increased anteriorposterior (A-P) and lateral movement (i.e. pitch and yaw) in the confederate resulted in a decreased amount of movement in the naive participant. Decreased A-P and attenuated facial expressions of the confederate resulted in increased A-P and lateral angular velocity of the naive participant. In

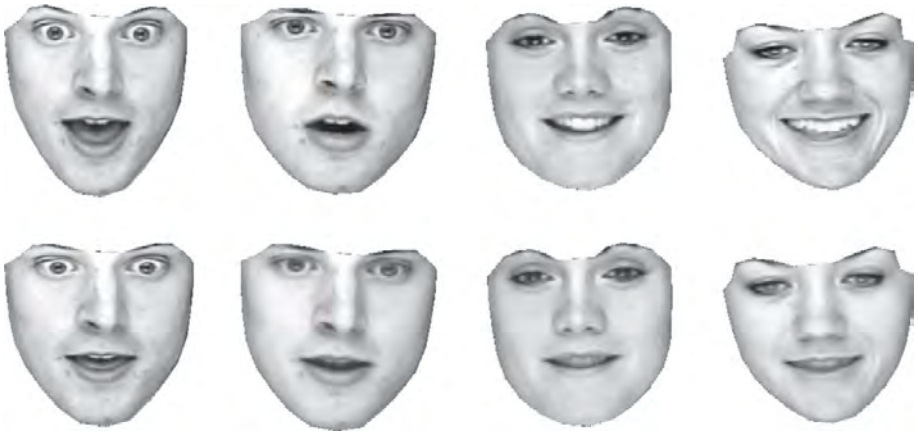


Figure 2.12: Facial expression attenuation using an AAM. Top Row: Four faces re-synthesized from their respective AAM models showing expressions from tracked video frames. Bottom Row: The same video frames displayed at 25% of their AAM parameter difference from each individual's mean facial expression (i.e. $\beta = 0.25$) [47]

this case, the naive participant's increase in head movement could be due to a variety of factors, one of which is attempting to elicit more expressive responses from the confederate. The authors propose a hypothesis that explains this balance of head movements between the participants as a "shared equilibrium"; one-side providing less energy when the other was engaged, or they supply more energy when the movement of the other is less (such as in the attenuated case). Attenuating the facial expression and modifying the voice of the confederate alone did not seem to have any significant effect on the behaviour or perception of the naive participant. Only when head movement was involved, such as in the example given above, was there an effect on the conversation participants.

These results follow logic, as a slightly less expressive face is not typically a concern for change in behaviour, but an individual whose focus is elsewhere (as indicated by head movement away from a subject) or an individual who is enthusiastically moving their head in response to another, will cause an effect on conversational dynamics.

While this work made important strides into the role of head movements and facial expressions in dyadic conversations, it has quite a few short-comings. The first short-coming is that the paper makes no real mention of the content of the conversations. That is, were the individuals given topics to discuss, did the confederate follow a script, or did the two simply exchange facial expressions? This is an extremely

important detail to leave out (or at worst, overlook) of a paper whose purpose was to understand the effect of head and facial movement in conversations. The structure of the experiment and content of the conversation would not only put the results in a different light, but also the modifications they applied. For instance, in a natural, dyadic conversation there is a substantial amount of mouth movement for both the speaker and the listener (who may respond with short-verbal signals). One cannot imagine that attenuating the entire face as the researchers did, would result in a spoken sentence looking natural to the naive participants.

The second short-coming is related to the first. By not describing the conversations themselves, it is impossible to know the context in which the reported observations occurred. Simply showing that the attenuated lateral head movement and facial expressions of the confederate resulted in increased head movement of the naive participant, is relatively meaningless without some context. There are a multitude of reasons this could occur apart from the physical actions of the head and face. Was the confederate telling a story to the naive participant about his/her sick puppy? Was the confederate describing his/her game winning sports performance? Although the authors state, of the naive participants, that "none mentioned that they thought they were speaking with a computer generated face or noted the experimental manipulations", this does not imply that the interactions were perfectly natural. There are plenty of "peculiar" individuals with whom we interact on a regular basis. The 27 naive participants were all recruited from a psychology department at a university. It could very well be that they acted in a non-natural manner to a somewhat odd behaving face cut-out, without going so far as to make an assumption about the capabilities of computer-generated faces. The authors' statement is therefore only in support that their model did not produce extreme artefacts, and not that the manipulations they made were consistent or natural.

The third short-coming is, perhaps, explained by the first two conditions. This short-coming is that the analysis focused more on frequency than other meaningful results. The only manipulations performed were that of 1.0 and .5 from canonical values. Given that the focus is on frequency and not context, intensity or timings,

it is impossible to understand the driving factors of the interactions and how they varied. Lateral angular velocity increasing while the other signal decreases could be interesting but, as stated in the paper, there are a handful of hypotheses as to what that interaction means and how that knowledge could be of any use. The researchers do themselves a disservice by ignoring the temporal dynamics and timings (as well as the more qualitative context attributes, as stated previously). There is also no real 'ground truth' with the head and face movements. Take two confederates, one whose natural movements are greatly above the norm and one who is naturally much calmer. It is going to be difficult to draw any conclusions about the reactions of naive participants when the attenuation of the head movements and/or facial expressions is inconsistent and, more importantly, not recorded/reported for analysis. One could argue these details are not important because we regularly interact with individuals with varying degrees of expressiveness, but the point of this work was to observe the effects of damping head movement and facial expressions in dyadic conversations. The effects are surely more than the simple frequency and velocity of head movement exchanges.

The limitations of having to use a floating head, person-specific AAM models, or that the AAM model sometimes ignored the eye blinks of the confederate, can easily be forgiven as items restricted by current technology. The lack of detailing the conversational interactions, in a paper so focused on the effects of head movement and facial expressions in conversations, is harder to understand. With so much rich information in the temporal dynamics, timings, and context of conversations, it is a shame this work fell short of its full potential.

2.9.2 Facial Expression Synthesis

Synthesising realistic facial expressions is important for a variety of fields and applications. From biomedical applications, to use in perceptual psychology experiments, it is useful to be able to produce realistic facial expressions, some of which may never have been directly captured.

Wallraven et al. [238] examined the realism of facial expressions from a perceptual psychology view-point. Understanding when expressions are "real enough" should be based on perception, not "physical accuracy" in Computer Graphics, they argue. It is for this reason they evaluate a variety of animation techniques to determine which lend themselves best to producing realistic synthesised facial expressions, using psychophysical experiments. They use 3D facial animation techniques on 4D (3D dynamic), structured light sequence data. The 4D scanner lacks high spatial resolution, which is why a 3D static scanner captured the peak expression frame of a sequence. A correspondence between the two types of scans is computed for creating a "set of morphable scans".

Seven posed facial expressions were captured from an amateur actor: happy, sad, fear, disgust, confusion, pleasantly surprised, and thinking. Six animation techniques were evaluated: Avatar, AvaNoR, AvaLin, AvaClu, 4DScan, and 4DPeak. *Avatar* used a hybrid approach of utilising the 3D scans of neutral and peak expressions along with the temporal timings from the motion capture. *AvaNoR* use of rigid head motion was "turned off' so as to evaluate its impact in the perception of the facial expressions. *AvaLin* produced a linear morph from neutral to peak expressions (3D static frames) and included the rigid head motion. *AvaClu* used cluster animation and weight maps for morphing from static neutral expressions to the different captured facial expressions. *4DScan* simply consisted of the 4D, low resolution captured data, which included shape and texture. In *4DPeak*, only the peak frame of each expression from the 4D scan was shown, in order to evaluate the perceptual qualities of dynamic sequences and static images.

In the first experiment, 12 human observers attempted to identify the avatar's facial expressions in an eight-alternative non-forced choice approach (the eighth choice being "none of the above"). As well, the observers were asked to rate the intensity, sincerity, and typicality of the expressions, using a scale ranging from 1 to 7. The average recognition accuracy was 60%. AvaClu was the animation technique that resulted in the best recognition accuracy, with 70% accuracy, as well as the highest ratings of intensity, sincerity, and typicality. However, the authors note that the

approach of AvaClu introduced jitter for the *confusion* and fear expressions, which might have helped the human observers identify these expressions.

In the second experiment, the shape, texture, and motion channels were manipulated as a means for identifying how each information channel can affect facial expression recognition and perception. For the 7 expressions, and 2 motion types (dynamic and static peak frames), a Gaussian blur was applied at five different levels for a gradual degradation of the signal, for the 3 channel types (shape, texture, and motion). This resulted in 175 different trial versions. 12 human observers (not the same 12 from Experiment 1) evaluated the trials. Unsurprisingly, degrading the information channels did have an effect on the perceptual measures. So long as the motion channel was present, blurring the shape and texture channels had no effect on recognition rates. If the motion channel was not present (static, peak frame data), then blurring of the shape strongly affected the recognition rates. Blurring the shape and texture affected the intensity evaluations, and blurring of the shape affected the perceived sincerity of the facial expressions. Examples of the avatar and different blurring levels can be seen in Figure 2.13.

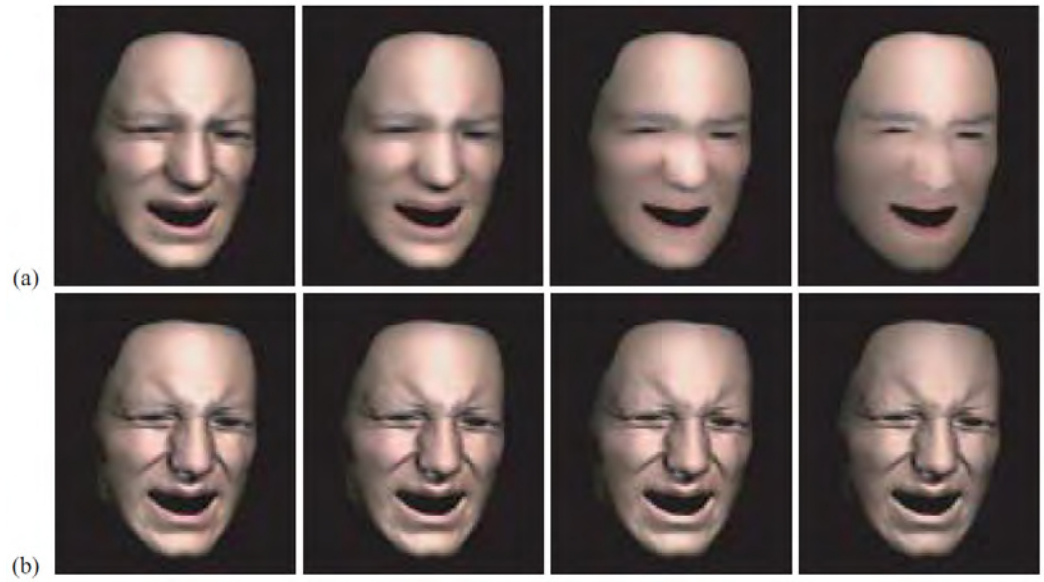


Figure 2.13: Examples of the four (a) shape and (b) texture blur levels for the disgust expression used in Experiment 2 [238]

Experiment 3 looked at the effect of adding in eyes and teeth to the avatar, which had only holes for the mouth and eyes previously. Unsurprisingly, they found that

the addition of these features increased the recognition accuracy, intensity ratings, and sincerity ratings.

This work was an important step in understanding some of the important aspects in manipulating and synthesising facial expressions. While the approaches were quite simple and the avatars could not be mistaken as "real" data captures of humans (like in a video), the research indicated, among other things, that motion is a key mechanism for the recognition and perception of facial expression characteristics, including sincerity, intensity, and typicality.

A similar work, with more realistic 3D synthesised faces, is that of Cosker et al. [79]. This work used 3D FACS-validated data to examine the importance of natural temporal information in the recognition of different facial actions. Twenty FACS-validated AUs were captured using a 3dMD active stereo system, and used in a perceptual experiment. Optical flow was used for tracking 2D features through each sequence and registration was achieved by aligning each frame's texture map to a reference texture map, calculating the barycentric coordinates for each texture map in relation to the reference meshes coordinates, and using this information to make correspondent the 3D mesh for each frame. Rigid head motion was also removed.

While many animation techniques (e.g. blend shape models) produce geometrically linear animations of facial actions, natural face deformations are geometrically non-linear. Figure 2.14 shows the difference between a linear blend shape model animation of a facial expression (AU 9 - Nose Wrinkler) and the natural, geometrically non-linear movement from the original data.

In a forced-choice perceptual experiment linear geometric motions are compared to non-linear geometric motions. Participants viewed the linear and non-linear sequences side-by-side and were asked to identify which of the two had the more natural motion. They rated their confidence in this choice using a Likert-scale that ranged from 1 (Not confident at all) to 5 (Very confident). The results of the experiment showed that, overall, participants perceived the non-linear animations as more natural than the linear animations. For certain sequences, such as AU's 4, 9, and 17, the non-linear

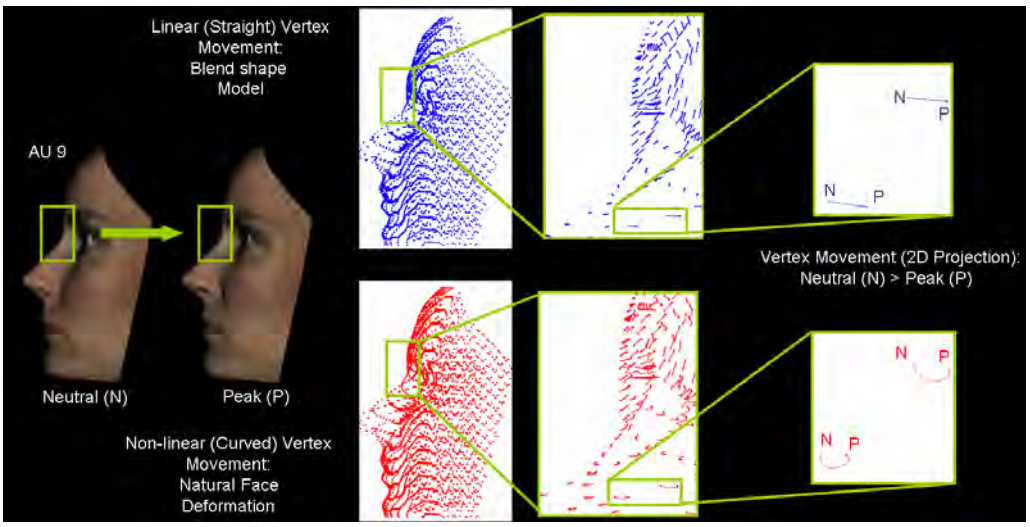
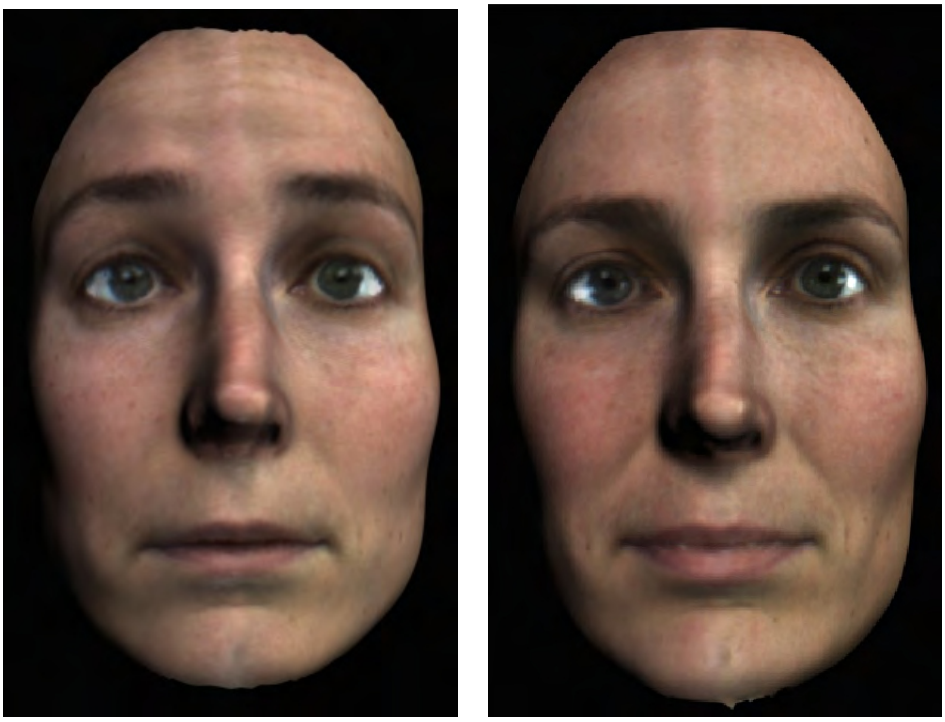


Figure 2.14: Linear vs non-linear geometric vertex motion for AU 9. Note that the vertices (sub-sampled) follow a curve in the non-linear animations (recorded from a real facial performance) and a straight line in the linear animations (created using a blend shape model). [79]

animations were strongly preferred over the linear ones. For other sequences, such as AU's 1, 10, 20, and 25, the linear animations were preferred. The authors noticed that the non-linear preference correlated with actions that included wrinkling of the skin, and that it is not surprising that perceptions of realism would be negatively affected by linear animations of these types of deformations. There was no significant difference in the ratings of confidence for the two animation types. This study is the first to numerically evaluate the effect of geometrically linear and non-linear motion on perceptions of realism for synthesised sequences of facial actions.

One drawback of this work is that it was only focused on single AU movements and ignored AU combinations, which represent a majority of common facial expressions such as happiness, surprise, sadness, anger, and disgust. Given that this work is aimed towards creating more perceptually realistic animations of facial expressions, understanding how linear versus non-linear animations of "complex" expressions affect the perception of realism and naturalness is of great importance. It is also of interest to social psychology researchers, as complex facial actions are common in everyday social communication. Another shortcoming of this work is technical in nature. The authors do not seem to address it, but there is a visible seam down the middle of the synthesised meshes, most likely an artefact from the capturing and statistical

modelling process. Figure 2.15 shows two meshes. The seam is apparent in the wrinkles of the forehead in Figure 2.15(a). Figure 2.15(b) shows how the differences in colouration make this seam quite obvious. For any experiments evaluating the naturalness of a synthesised expression, this detail is important. However, if both the linear and non-linear animations used these captured sequences, as is implied in the paper, it is less of an issue and one that can be addressed in the future.



(a) AU 1 (b) AU 6

Figure 2.15: Example of Synthesised Frames [79]

Overall, this paper provides a better understanding of the importance of geometrically non-linear movements of faces for the perception of naturalness, and opens up new areas of exploration for researchers interested in the perception of realism and naturalness of synthesised facial expressions.

In [249], Yu et al. attempted to synthesise the 6 prototypical expressions using a three-view Di3D system. Optical flow was used to track feature points through a sequence and 3D Morphable Models (3DMM) used for modelling. An expression sequence was synthesised by animating from neutral to a peak, target AU, and back to neutral. This approach, unfortunately, ignored a key aspect of realistic expressions: temporal dynamics. As well, from the synthesised faces shown in the paper, it would

be hard to argue the faces looked natural and were not in the "uncanny valley" area of facial realism (Figure 2.16). Human observers made evaluations on the emotional expression and intensities in a forced-choice approach. The evaluation ignored the realism of the expressions or of the sequence image quality. While a decent attempt, this work lacked the realism necessary, both in image quality and temporal dynamics of natural facial movements, for synthesising realistic facial expressions.

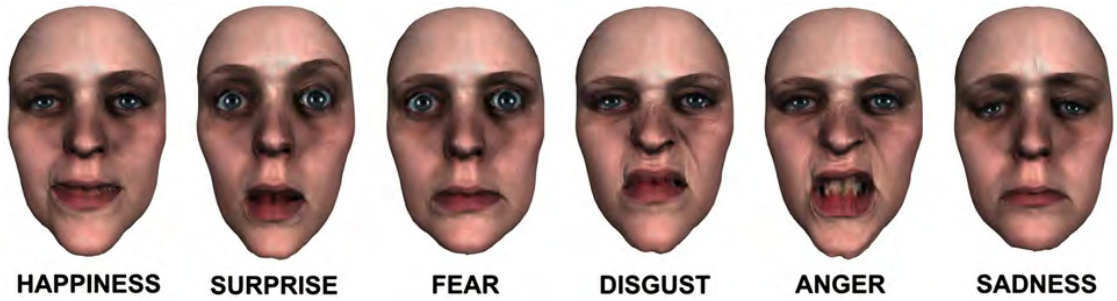
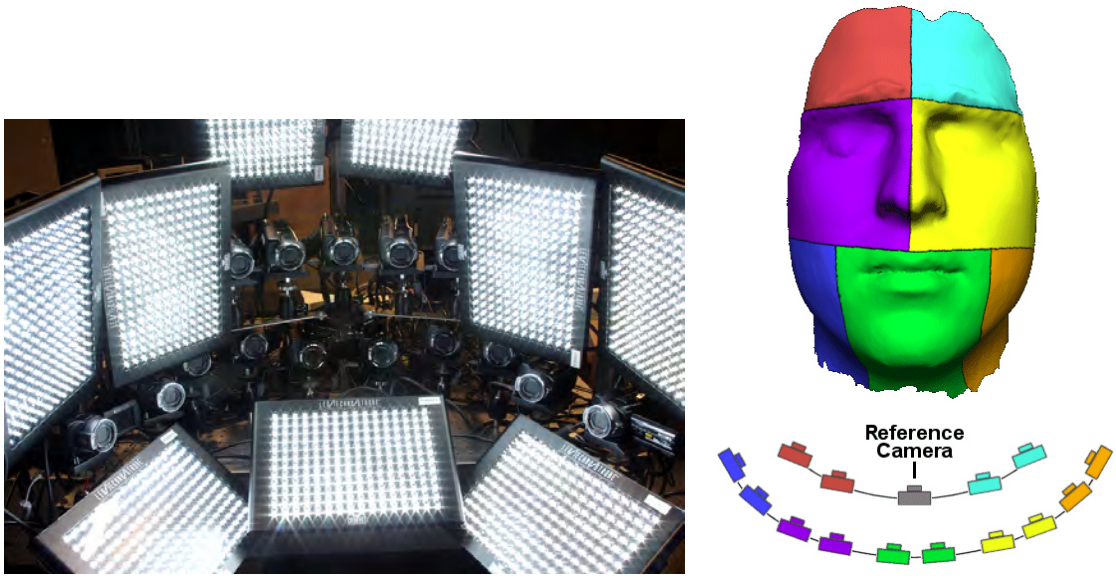


Figure 2.16: Example of Synthesised AU Peak Expression Frames [249]

In [54], Bradley et al. introduced an approach to passive stereo face capture that requires no facial markers or active lighting. They used 14 HD cameras arranged in binocular stereo pairs. These cameras were geometrically calibrated and each of the binocular stereo pairs were zoomed-in on a different part of the face. Figure 2.17 shows the system configuration and parts of the face captured.



(a) Capture System - 14 HD Cameras and 9 LED Panels (b) Parts of the face captured

Figure 2.17: Capture System Setup in [54]

The views did overlap somewhat, and the zooming of the cameras allowed them

to capture minute details of the face, such as skin pores, hair follicles, and skin blemishes. The actors were captured under uniform illumination conditions using 9 LED light panels that had 192 LEDs each.

Multi-view reconstruction was performed using the seven binocular stereo views. The resulting seven depth images were merged into a single mesh using an iterative process for multi-view reconstruction; an extension to a technique first described in [53]. The original approach was designed for 360-degree data. The data captured in this paper only uses a front view of the face. For this reason the authors implemented an *iteratively constrained binocular reconstruction* approach. This is an approach for removing outliers during reconstruction and works by iteratively tightening depth constraints until the pixels are within an acceptable distance to the surface mesh. The authors found this process tended to only need three iterations. After this process is complete the seven point clouds are combined into a single, dense point cloud with approximately 500,000 vertices and 1 million triangles.

After creating the 3D mesh from their multi-view reconstruction method, they create a 2D, single-view texture map by projecting the 3D triangles captured from each camera view to a 2D plane. The different cameras of the system capture slightly different skin tones and this must be dealt with when creating the single-view texture map. This issue is addressed by applying Poisson image editing techniques to create a consistent texture across the single-view texture map. Figure 2.18 shows this process.

Figure 2.18: 2D texture generation. From left to right: reference image, camera contribution image, initial texture, final texture [54]

The authors use an optical flow based approach for feature tracking and correspondence. To overcome the drift issues associated with optical flow, they implement

mouth tracking and texture-based drift correction. This texture-based drift correction is used on the assumption that the single-view texture maps do not differ greatly in their appearance, and as such, any temporal drift in the 3D mesh will appear as a small 2D shift in the texture images, which can easily be detected using optical flow. This correction step, along with the mouth tracking used for adding in the teeth from the reference frame, produced 3D, feature correspondent frames. Figure 2.19 shows the original capture, the 3D mesh without teeth, and the 3D mesh with the teeth added back in.

Figure 2.19: From left to right: original capture, mesh without teeth, final mesh with teeth added back in [54]

One of the main drawbacks of the approach described by the authors is that it struggles with fast facial motions. The motion blur associated with fast motion causes problems for their optical flow method. To use such a system for capturing spontaneous expressions or natural expressions of individuals in conversation may prove to be problematic as this type of data is often associated with fast, dynamic movements.

The other main shortcoming of this work is that, given the image examples in the paper, the 3D output is not convincingly realistic. While their approach preserves minute details, such as skin pores, the overall mesh lacks visual realism (i.e. the 3D meshes would not be mistaken as real video captures of people). It could be a side effect of the makeup applied to the actors, or the smoothing techniques used in the multi-view reconstruction and single-view texture map approaches, but the 3D faces shown have an overly smooth appearance and odd colour tone. This results

in the overall face lacking visual realism, as can easily be seen in the final frame of Figure 2.19. Even with these drawbacks, the approaches described in this work offer promising techniques for future research in the area of markerless, passive stereo performance capture.

Beeler et al. [38] also offer an approach to markerless, passive stereo facial performance capture. They describe an approach that can work on any stereo capture system, but they do require the cameras to be focused on the entire face, unlike in [54]. In this work *anchor frames*, which are frames considered close (similar in image and expression appearance) to the reference mesh are selected. These anchor frames are used to create *clips*, which are short sequences that occur between two anchor frames. Feature tracking is performed by finding a correspondence between the anchor frames for a clip and the reference mesh, and then tracking through the clip starting from the anchor frame. The mapping of the reference frame to the anchor frame and the anchor frame to the individual clip allows for accurate feature tracking. This mapping is used for propagating the reference mesh to each frame in the sequence. By using individual clips for this tracking, they are able to avoid drifting issues and can process a sequence in parallel (process clips separately). By working directly in the 2D image space (as opposed to a 2.5D approach) these techniques avoid the issue motion blur can cause, and produce a more detailed geometry than the main work they were comparing against: Bradley et al. [54]. Figure 2.20 shows the original frames and the 3D meshes generated using their approach. Figure 2.21 shows an overview of their approach.

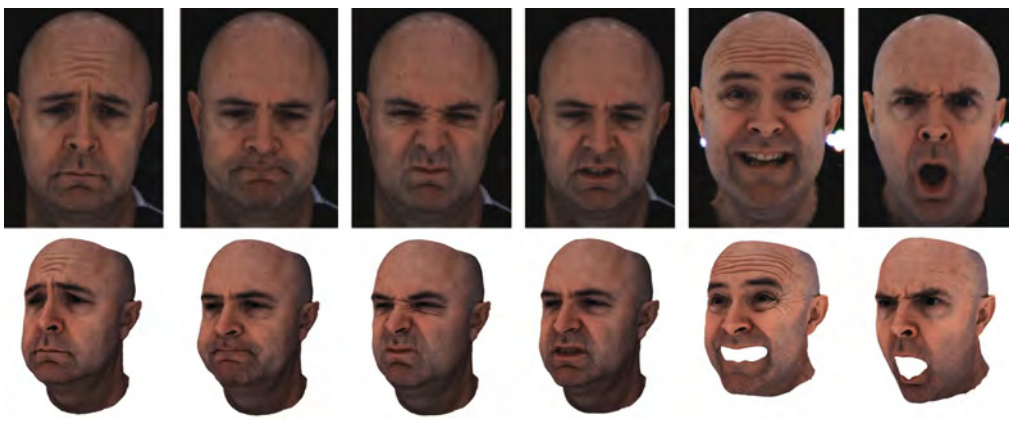


Figure 2.20: Original data captured and the corresponding 3D meshes [38]

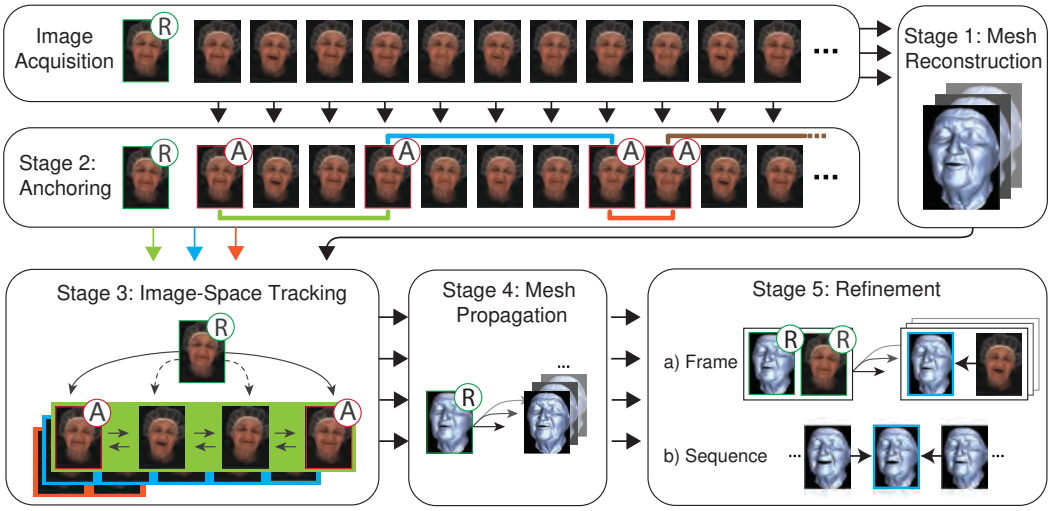


Figure 2.21: System overview. Image acquisition is followed by mesh reconstruction (Stage 1) and anchor frames are detected to partition the sequence (Stage 2). The image-space tracking step matches the reference frame to all frames in the sequence (Stage 3), and then the reference mesh is propagated to each frame (Stage 4). Finally, the meshes are refined for a high-quality result (Stage 5). Image and information from [38]

Their technique does allow for intra-subject registration, but without additional steps, it does not allow for 4D inter-subject registration. This is not an issue given the purpose of this work (facial performance capture for entertainment purposes), but could be for researchers wanting an approach that includes the ability to produce inter-subject registered 3D meshes. The static realism of the captured faces is impressive and a variation of this capturing approach for dyadic, dynamic video captures could be utilised by our lab for creating more detailed faces during our capturing process.

The University of Southern California (USC) Institute for Creative Technologies (ICT) Light Stage [91] capturing technology is a state-of-the-art approach for capturing highly-realistic facial scans (and full body, as of Light Stage 6). The system works by capturing high-resolution static geometry using multi-view stereo and gradient-based photometric stereo techniques [116, 117] which results in highly-detailed, highly-realistic facial captures. The system captures a scene in a variety of lighting conditions and using proprietary software to create 3D highly-detailed faces, which are mainly used in entertainment media.

Two specific projects, *Digital Emily* [10] and *Digital Ira* [9, 112], showed the quality

of this approach for capturing and facial animation. The Digital Emily project used the USC ICT's *Light Stage 5* technology to capture facial performances. 40 manual markers were applied to the actresses face and 37 facial scans were captured. The Light Stage used had 156 white LED lights that fired at different times to cover a range of lighting conditions. Only 15 photographs were required per facial expression and used Canon EOS 1D Mark III digital (still) cameras. This set up required the actress to hold an expression for 3-seconds. 3D multi-view reconstruction was performed using a stereo correspondence algorithm and the photographs that included the coloured stripe patterns from a video projector. A 3D mesh model of the actress's neutral expression was built and a facial performance capture was used to drive the models and blend shapes. The teeth were modelled using a plaster cast and digital model and placed back into the rigged mesh. The eyes were also separately modelled.

Many manual steps were required after the initial capture using a variety of software programs (both publicly available and in-lab only). The process to rig, animate, and render the captured sequence took approximately a combined 7 months. 3 of those months were dedicated to rigging the face model (which had 75 blend shapes) and 3 months were dedicated to rendering and compositing. 3 capture system technicians, 3 artists and 2 animators were required for the process.

While the Light Stage system and processing approaches in these papers result in highly-detailed and highly-realistic facial textures (more static detail/realism than the meshes of this work), they use the established blend shape approach for animation. This approach lacks the specific details of dynamic movement and minute/micro expressions found in normal facial expressions. In their approach there are many manual steps for ensuring the animation matches the desired output. This is a time-consuming and labour intensive approach and it is likely that these manual steps remove or reduce the naturalness of the facial performance, specifically due to interpretation by artists and animators whose goal is to produce a desired output that may not align perfectly with the most accurate output. Unfortunately, there have been no published perceptual experiments evaluating the realism of these facial

performances using this system and approach to original video facial performance captures. The blend shape modelling approach also suffers from "combinatorial explosion in representing the complex manifold of facial expression" [200, 201], which makes it an unlikely approach for producing highly-realistic facial performances (both globally and minute/micro expressions) for use in perceptual experiments and for modelling conversational interactions.

The approaches from these papers are suitable and set up for entertainment media creation, but are not suitable for our research needs. Aside from requiring a (currently) not commercially available, highly-specialised capturing system and proprietary processing software, the amount of manual labour required to transform static captures and facial performances into 3D sequences makes this approach unsuitable for our research. We look to capture many subjects in natural conversation with one another and wish to capture all major and minute facial movements, use a fully-automated processing pipeline, and a statistical modelling approach that allows us to model characteristics from any number of subjects. The static realism of the captured faces is impressive and a variation of this capturing approach for dyadic, dynamic video captures could be utilised by our lab in the future for capturing more detailed faces during our acquisition phase.

The work of this thesis in modelling and synthesising highly-realistic, modified 4D facial expression sequences provides the next step in this area of research. Chapters 6 and 7 provide details of our 3D tracking and inter-subject registration, statistical modelling, sequence modification and synthesis approaches; and the perceptual studies that validated these approaches.

2.9.3 Predictive and Coupled Statistical Models

Classifying behaviour, such as facial expressions, is not just limited to rule-based or machine learning approaches. Statistical models can also be used to classify and predict behaviour.

Galata et al. [113] used 2D and 3D data of body gestures to build stochastic models of high-level structure of activities without assumption of prior knowledge. By breaking down complex movements into components of movement, and using Variable Length Markov Models (VLMMs), the authors were able to represent behaviours efficiently, as well as predict and synthesise new movements. In a similar work [133], human interactions were statistically modelled (handshakes) and used for predicting and synthesising appropriate output given input gesture of human hand shake.

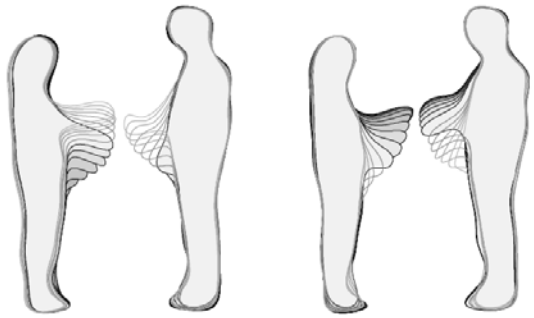


Figure 2.22: Examples of behaviour extracted, at each time interval, for two people shaking hands. This was used for predicting the behaviour of the person on the right given input from the person on the left [133]

Many statistical based approaches exist for learning and modelling behaviour. Such models that "couple" one behaviour model with another are termed "coupled statistical models". Coupled statistical models are often used to enhance the information contained in data.

Hogg et al. [126] was one of the first works to build coupled models of facial expressions. Using tracked data of head shakes and nod movements, a 2D coupled statistical model was built and used to synthesise a nodding head, which would be displayed when a real (non-synthesised) head was making a shaking motion (Figure 2.23).

In [71], 2D coupled-view Active Appearance Models were used to determine the relationship of the frontal-view and profile of the face. Castelan et al. used 2D frontal photographs to approximate 3D face shape, by coupling intensity and height information [64]. In [198], Coupled Scaled Gaussian Process Regression (CSGPR) models are used for head pose normalisation, with the goal of head-pose invariant

Figure 2.23: Left: Tracked faces for behavioural training data. Right: Reconstructed face (right) responding to a real face (left) [126]

facial expression recognition.

All of these approaches couple actions which occur in the same time instance. The coupled models we build in this paper are sequential in time, as one action influences another. To our knowledge, no work on 4D coupled models of conversational expressions currently exists.

2.10 Summary

This chapter explored the various facial expression databases, facial expression extraction, recognition, and classification techniques, and methods for building dynamic statistical models of facial expressions. Many of these research applications require tracking and registration methods for creating corresponding sequence data. Some also use this corresponding data to build statistical models for use in analysing and synthesising facial expressions. Much of the work over the past 5-10 years has focused on action unit recognition and face recognition (such as for biometric tasks). While this is a challenging problem and useful to solve, most work has ignored an important characteristic of facial expressions: temporal dynamics. The amount of information conveyed in how the face moves when making these expressions is quite important and should not be overlooked. The majority of these works also focus on a single individual making *prototypical* expressions, such as anger, fear, or disgust. Whether posed or spontaneous, the fact remains that these expression are not common occurrences in day-to-day interactions. *Conversational* expressions such as agreement, thinking, and confusion are common and of great interest to, among

others, social psychology researchers. There are, however, no 4D databases which consist of natural, dyadic, synced conversations between two people. A database such as this would allow for interesting research not only in how people interact or the conversational expressions they use, but also, given the 4D nature of the data, how the face moves during the expression interactions. The data being synced would also allow researchers to understand how one person's social signals affect the other, and even how the reaction to that signal can have an effect. Coupled statistical models would provide a good method for modelling these interactions and could even be used for predicting expression interaction characteristics.

Given all of the useful and interesting research that could be conducted, it was surprising to learn that no 4D databases of natural conversations existed. As the following chapters will demonstrate, this may be because of the challenging nature of capturing and processing 4D, synced data (Chapter 3). While the active stereo system used is one of the most advanced capture systems in the world, the data it produces can not be easily manipulated, without negative consequences (Chapter 4). Even once the massive amount of data is pre-processed, the next challenge of accurately tracking feature points through the 60fps sequence proves to be tricky. Finally, most dense registration techniques, as described previously, lack the ability to create highly-realistic corresponding 3D frames, especially across multiple subjects (Chapter 5). In order to build statistical models of individuals in conversations, at least two subjects need to have registered (corresponding) data (Chapter 6). In the research performed later in this work, all captured subjects will need to be registered (Chapter 7).

Chapter 3

4D Expression and Conversation Data Capture

3.1 Introduction

As described in the previous chapter (Chapter 2), a database of natural, dyadic conversations would be useful for a variety of applications in affective computing, perceptual psychology, HCI, and computer vision. Some research topics include better understanding of interactions in natural conversations and identifying conversational expressions. Some applications include developing more natural virtual humans and synthesising highly-realistic conversational facial expressions.

Such a database was captured at Cardiff University in May 2014. It is the first database of its kind in the world. This database was captured because existing conversational databases (e.g. [89, 166, 169, 177, 235, 236]) did not meet our requirement of high-quality, natural, 4D captures of the faces of individuals in conversations. Our database was captured with the intent of providing the research community with natural, synced, annotated data that could be used in a variety of research areas. This database has been made publicly available [229]. While the research community may use the database for a variety of applications, we intend to use it to build statistical models of appearance, for use in analysing conversational

interactions as well as synthesising new conversational facial expressions.

The following sections describe the capturing process, as well as the annotation process, in detail. The information below also described a separate data collection session where Duchenne and Non-Duchenne smiles were captured. This data was validated and used to build models for creating highly-realistic synthesised facial expression sequences. This model creation is described in detail in Chapter 6 and used in Chapter 7 for classification and perceptual experiments.

3.2 4D Capture System

We used 3dMD 4D (3D video) systems for our data captures. Two 4D, 3dMD, synced, 60fps capture systems were placed back-to-back (Figure 3.1) and used for the conversational data acquisition (Section 3.3). A single system was used for capturing Duchenne and Non-Duchenne smiles (Section 3.4).

Figure 3.1: 3dMD Synced 4D Capture Systems

The system consisted of 7 cameras, 3 LED light panels, 2 projectors, and a PC with 3dMD capturing and processing software. The PC was a 64-bit, Windows 7, 3.60 Ghz, Intel Core i7-3820 with 8GB of memory (64GB of memory in one system after upgrade), with a GeForce 620 GPU, and 7 250GB Samsung EVO SSDs (one for each camera). Each system consists of 7 Allied Vision Technologies Prosilica GT cameras:

4 monochrome (GT1660) and 3 colour (GT1660C) (Figure 3.2). These cameras are 2 megapixel cameras with gigabit Ethernet interfaces, Truesense KAI-02050, 2/3" CCD Progressive sensors, Auto Iris (P-Iris and DC), have a resolution of 1600×1200, a bit depth of 14-bit (mono) and 12-bit (colour), and a maximum frame rate of 62fps.

Figure 3.2: Prosilica GT 1660 Camera

The layout of these cameras consisted of two pods containing two mono cameras on the top and bottom, with a colour camera in the middle. These pods were on the left and right side of the subject, with the seventh camera, a single colour camera, in the centre of the subject. This camera was in front and slightly above the subject.

The speckle projectors were modified Moritex Schott LLS 3 LED light sources, which have an optical output of 550 lumens and fast triggered strobe capabilities.

Figure 3.3: Moritex Schott LLS 3 LED light source

The mono cameras and speckle projector were slightly offset with the colour cameras and 3 LED light panels. The LED panels oscillate at 120 Hz. When the LED panels are on, each colour camera captures a texture frame (Figure 3.4). When off, the projectors project a speckle pattern onto the subject and the mono cameras capture a frame (Figure 3.5). This results in 7 images per frame that are used with the camera calibration information to create a 3D frame using 3dMD's proprietary *MStereo*

software program. The two file formats provided by MStereo are TSB (proprietary) and OBJ (open). A lapel microphone was used for audio capture (44.1 KHz).

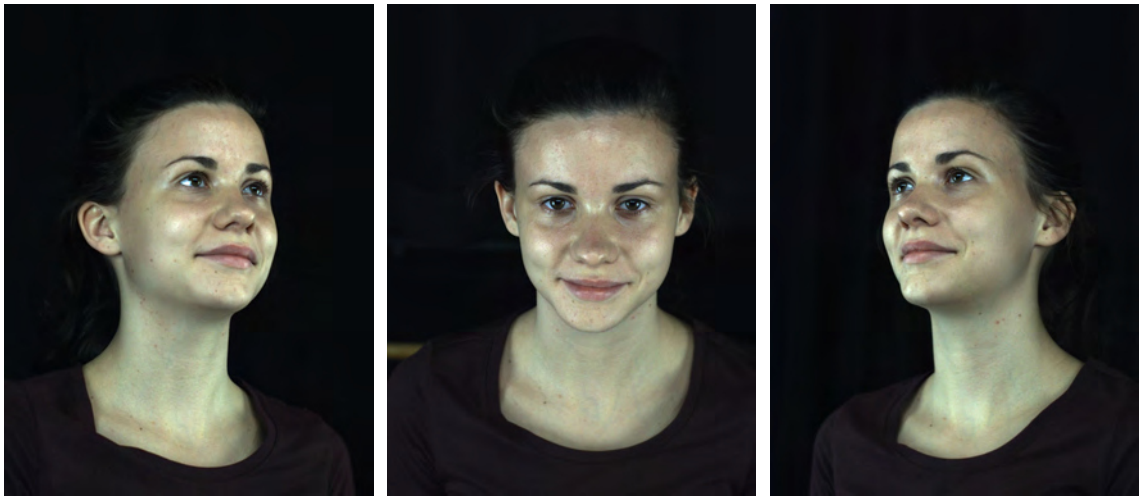


Figure 3.4: The 3 Colour Images of a Single Frame

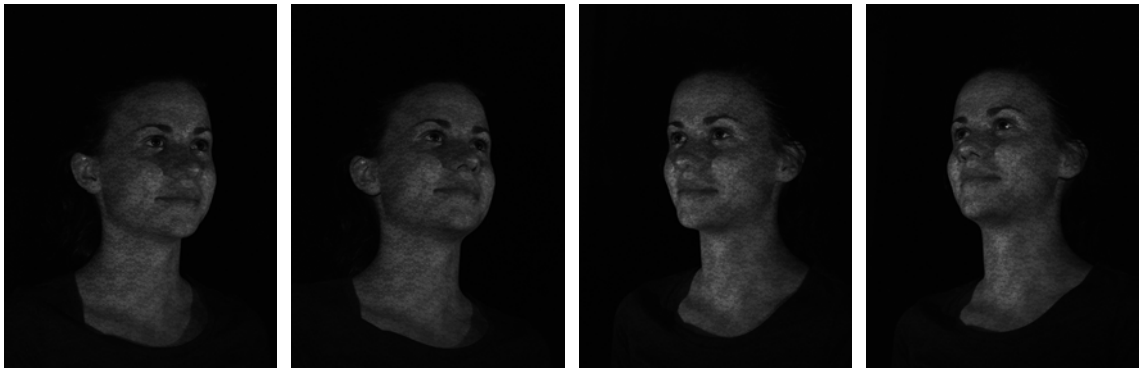


Figure 3.5: The 4 Mono Images of a Single Frame

3.2.1 Capture System Enhancements

Development of the data processing pipeline was a long and arduous task. There were countless unforeseen issues, many varying algorithms and approaches to research and consider, and more than a number of technical obstacles to conquer. In the end, the pipeline is as described below. Note: Per frame processing time will be given as well as 'total time' in parentheses, which is the calculated time for processing both sides of a 1-minute conversation (or 7200 frames).

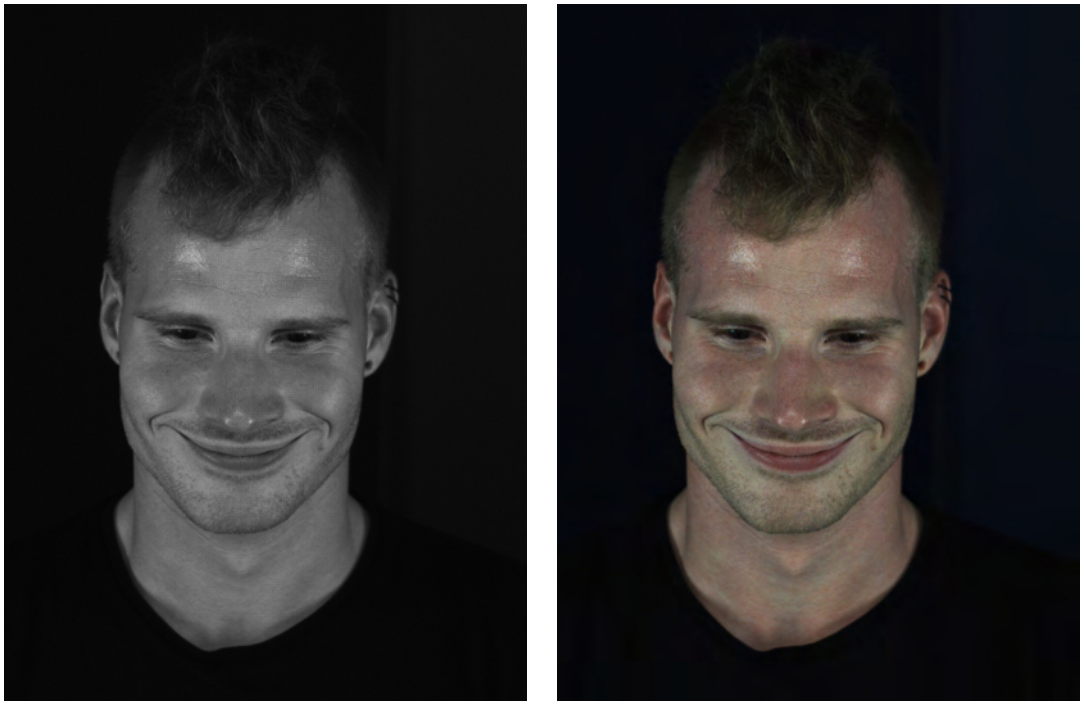
The capture system came with a third-party software tool, Norpix's *StreamPix* [2] for the actual capture process. It assists in viewing the scene and saving the

captured data to disk. This software system was originally capturing the data in a proprietary format, for each of the seven cameras. Producing the 7 raw image files (4 mono, 3 colour) from the capture software's proprietary file format consisted of a painstakingly slow, manual process. This caused a few issues. Firstly, given that this export process was being done in the capture software, a significant amount of time was being taken up during capture sessions (20-30 minutes for a 2-minute capture). This meant that subjects would have to sit and wait for the export to finish before the next capture could begin, which was unacceptable. It was possible to wait until after the capturing process to reload the proprietary files and perform the export process (for producing the 7 images), however, this was also a very time-consuming and manual process. It required loading each of the 7 proprietary files (one for each camera) separately into the capture software. A more automatic approach was greatly sought after.

In the pursuit of efficiency and saving time a new approach was developed. The third-party software had the ability to write BMPs, instead of the proprietary file, straight to each camera's solid-state drive (SSD). Given the capture rate of 60fps, there was initially an issue with dropping frames. 60fps was achieved by upgrading the camera's firmware and skipping the frame colourisation process. A colour reconstruction would occur offline by applying a Bayer demosaicing algorithm to the 3 texture BMP images. The original capturing approach resulted in a post-processing time of approximately 2 seconds per frame (4 hours total). The new approach resulted in processing time being equivalent to real-time capturing, of 2 minutes, plus approximately 15 minutes total to apply the Bayer demosaicing algorithm (Figure 3.6).

This new approach caused a slight difficulty in data organisation. Instead of a single proprietary file for each camera, there was now one BMP for each frame captured, for each of the 7 cameras (7200 BMPs per camera, 50,400 BMPs total) . A Python [3, 228] script was written for moving these BMPs to a destination folder and renaming each file as appropriate to follow the 3dMD file format guidelines. Along with these BMPs, system calibration information and other pertinent files were copied to the destination folder. From here it would be possible to run 3dMD's

proprietary MStereo program to produce the 3D frames.



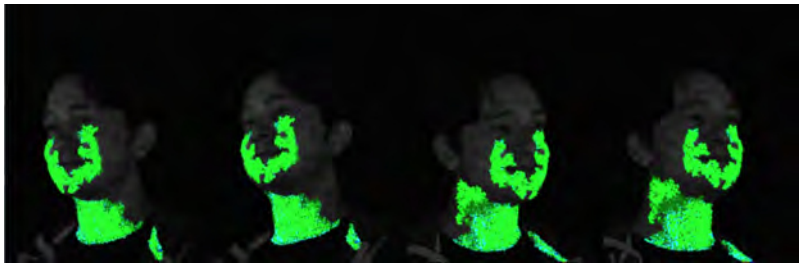
(a) Before Bayer Demosaicing (b) After Bayer Demosaicing

Figure 3.6: Before and After Applying Bayer Demosaicing Algorithm

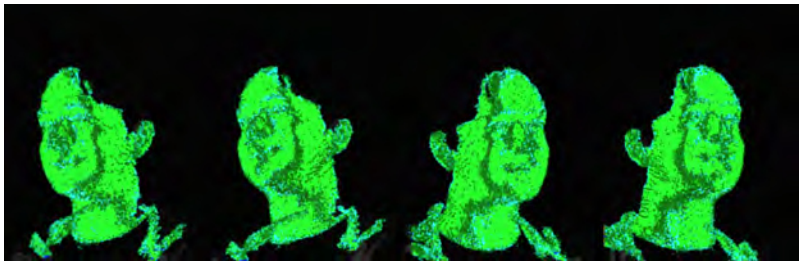
3dMD's *MStereo* program, which uses the 7 BMPs and calibration information to create the 3D frames, was used. Slight modifications were made for speeding up processing. For each frame, the 7 BMPs along with calibration information was fed into the MStereo program and as output a TSB file was created. The original MStereo program took 9 seconds per frame (18 hours total). The modifications, which included a simple multiple-processing approach and bypassing the visualisation feature (Figure 3.7), reduced the per frame processing time to just 3.8 seconds per frame (7.5 hours total).

The TSB file is a 3dMD proprietary format and is useful for use in a variety of 3dMD software. However, because of its closed format it would not be useful for reading into viewing programs such as Blender [1] or Meshlab [68], nor would it allow for editing of information. Using a 3dMD program the TSB files were converted to OBJs (with a three-view texture map BMP and 3dMD MTL file). The same modification was made to this program as was made to MStereo, which resulted in a reduction of processing time from 2 seconds per TSB (4 hours) to 1/4th of a second per TSB (30

(a) First Few Frames of Process



(b) Halfway Through Process



(c) Near Completion

Figure 3.7: MStereo Visualisation Stages

minutes).

3.3 Conversational Expressions Database

Not only does capturing conversations in 4D require a lot of data storage space, but the additional processing steps needed to convert the data into a usable format adds to the time and data storage cost. For instance, a single minute of a 4D conversation can take up 180 GB for the raw data, 80 GB for the proprietary files, and 125 GB for the OBJ files. This is a total of 385 GB of data for a single minute of 4D conversation data and does not include the files from the pre-processing pipeline (Chapter 4) step, which converts the data into a usable format for our research. For this reason, we needed to determine how many subjects would be required for varied and interesting conversational captures, while keeping in mind the time and data

restrictions involved with capturing 4D data. We decided that four subjects would allow us six unique conversation pairs and provide us four unique perspectives that would help to make the conversational captures varied when it came to content and topics.

Dr. Job van der Schalk from the Cardiff School of Psychology was able to recruit four unpaid volunteers: two male and two female, all Caucasian, ranging from approximately 20 to 50 years of age (Figure 3.8). They were recruited from the general public and had no tie to the lab performing the data acquisition. Three of these volunteers had a background in acting, although for the purpose of our experiment we specified they should act naturally. Participants with acting backgrounds were recruited because two separate data captures were occurring on the day, with the other experiment requiring directed and controlled facial expressions (Section 3.4). The participants were unaware of the specifics of the research and only given details at the conclusion of each capture session.

Figure 3.8: 3D Capture Examples of the 4 Participants

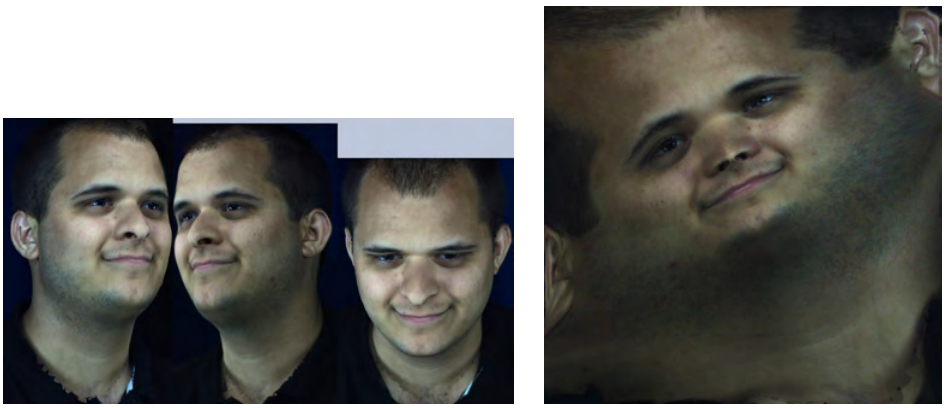
Six 3-minute conversation sessions were captured using two identical, synced, 4D 3dMD systems. These six sessions consisted of each pair of the four subjects. The 3dMD capture systems were placed back-to-back to allow the subjects to sit face-to-face with an unobstructed line-of-sight (Figure 3.1). Participants were given roughly a minute before each conversation capture session to allow them to become comfortable with the environment and the other participant. They were given an indication of when recording began and then three consecutive one-minute captures were recorded. One-minute captures were performed because one system was consistently dropping

frames near the 2-minute mark. It is important to note that the participants did not break from their conversation. Three consecutive captures were made within one long conversation. The subjects were rotated to allow them resting time in-between capture sessions, as well as to ensure they were captured on both systems. One conversation capture had to be discarded due to corrupted, out-of-sync data, resulting in a total of 17 one-minute conversations (34 sequences) being captured over the span of 2 hours.

3.3.1 Frame Processing Overview

The database consists of 17 one-minute, conversation captures (34 Sequences) at 60fps. Therefore, each sequence consists of approximately 3500-4000 frames, with 7 camera images for each frame: 4 mono and 3 colour (Figure 3.5 and Figure 3.4, respectively).

The 7 images are used with the camera calibration information to create 3D frames. The frames are 3D surface object OBJs, with a three-view texture map (BMP) (Figure 3.9(a)). Each OBJ consists of approximately 30,000 vertices, normals, and texture coordinates; and 55,000 faces (polygons). The total size per 3D frame (OBJ and BMP) is typically around 20 MB. A *cleaned* OBJ is then produced using an in-lab tool which removes non-manifold vertices and edges, isolated vertices, and small components, and then produces a unified (single-view) texture map (PNG) (Figure 3.9(b)). Full details of this process can be found in Chapter 4. The total size of the new OBJ and PNG is typically around 4.5 MB. Aside from taking up much less space, the single-view texture map resolves texture *uv*-coordinate issues researchers will have when they make certain modifications to the original 3D object, such as tracking non-vertex feature points through a sequence. In that specific case, new *uv* texture coordinates will often be located in separate images of the three-view texture map, resulting in errant texture patches for the affected faces. In the publicly available database both the original and cleaned OBJ data is provided.



(a) Three-View Texture Map

(b) Single-View, Unified Texture Map (UTM)

Figure 3.9: Three-View and Single-View Texture Maps

3.3.2 Database Annotations

A manual annotation approach was chosen for identifying frontchannel and backchannel interactions, conversational expressions, rigid head motion, and verbal/non-verbal utterances in the conversational data. These annotations will be useful for a variety of reasons to researchers accessing the database, but for our purposes, the annotations identify conversational interactions. Manual annotation of the sequences was carried out in ELAN [245]. ELAN is a publicly available, easy to use software tool that allows for multiple annotation tracks and hierarchical tracks. It also allows for time-accurate text annotation of speech sections (Figure 3.10). Due to time and training costs, four individuals were chosen to annotate this data. Two individuals with experience in annotating conversational data using ELAN helped to train the other two annotators. One pair of annotators (one experience and one new) were assigned to half of the conversations, while the other pair were assigned to the remaining conversations. A minimum of two annotators for such data is the *de facto* standard in the field, and splitting the work load helped to reduce annotation fatigue and the time required to complete the annotations.

A variety of facial expressions and gestures were annotated. The annotators were instructed to mark a backchannel signal as any expression or gesture made in response to verbal or non-verbal action from the other speaker. These backchannel signals can occur during or after the action. The annotation tracks included are (based on

those discussed in [176]) :

- **Frontchannel:** Main speaker periods.
- **Backchannel:** Expressions, gestures, and utterances that would be classified as backchannels.
- **Agree:** Up/down rigid head motion and/or vocalisation (e.g. 'yeah').
- **Disagree:** Left/right rigid head motion and/or vocalisation (e.g. 'no').
- **Utterance:** The periods of speaker activity, including all verbal and non-verbal activity.
  - **Verbal:** Whole or partial words spoken.
  - **Non-Verbal:** Verbal fillers (e.g. 'umm', 'err') and other non-verbal sounds (e.g. 'uh-huh')
- **Happy:** Smile or laugh.
  - **Smile:** Lip corners move upwards.
  - **Laugh:** Spontaneous smile and sound.
- **Interesting-Backchannel:** Eyebrows slightly raised, lip corners move downwards, slight head nod.
- **Surprise-Positive:** Mouth opening and/or raised eyebrows and/or widening of eyes. Upward motion of lip corners.
- **Surprise-Negative:** Mouth opening and/or raised eyebrows and/or widening of eyes. Wrinkled brow. Downward and/or backward-pull of mouth corners.
- **Thinking:** Eye gaze goes up and left/right.
- **Confusion:** Slight squint of the eyes, eyebrows move towards each other.
- **Head Nodding:** Up/down rigid head motion. This can be agreement or motion made during speech.
- **Head Shake:** Left/right rigid head motion. This can be disagreement or motion made during speech.
- **Head Tilt:** In plane left/right rotation of the head.

- **Other:** Expressions not included in the list, but are interesting, such as consistent facial mannerisms of an individual

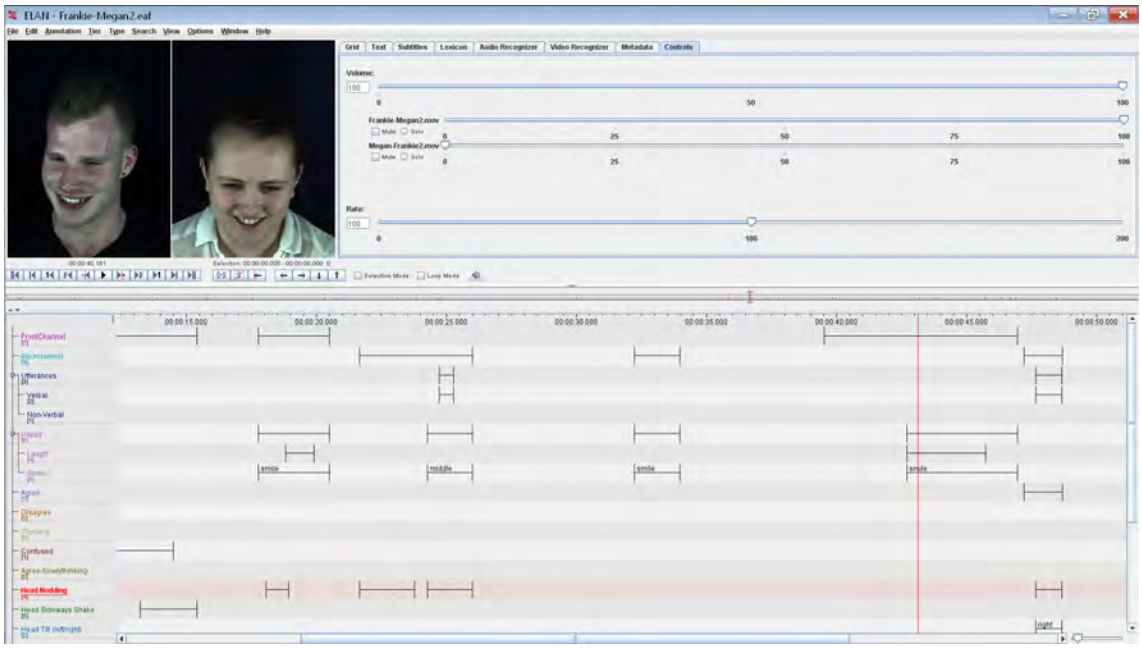


Figure 3.10: Screenshot of ELAN Software

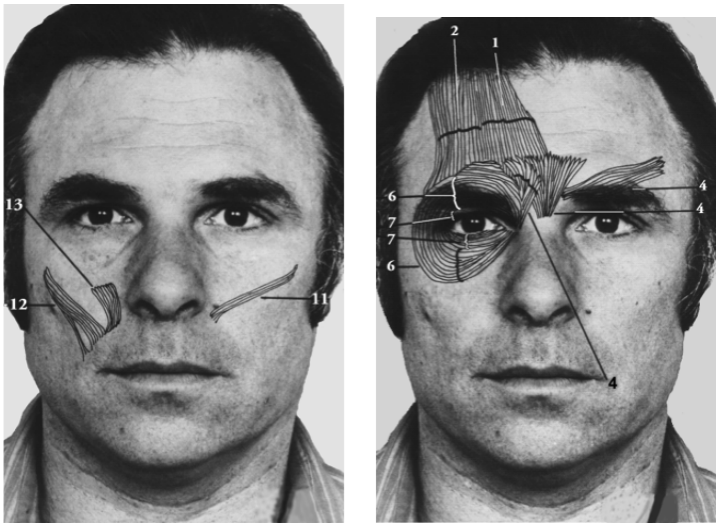
Hereafter, *expression periods* refers to specific annotated instances. The annotations consist of 764 Frontchannel/Backchannel expression periods (329 Frontchannel, 435 Backchannel. Note: Multiple annotation types can fall under the same annotated period), 433 rigid expression periods (e.g. head nod), 450 non-rigid expression periods (e.g. smiles), 305 verbal/non-verbal utterance periods, and 307 'Other' expression periods.

3.4 Psychology Smile Database

As touched on in Section 3.3, a second data capture session was performed on the day, and was in collaboration with the Cardiff School of Psychology. The researchers from the School of Psychology are interested in evaluating the characteristics of facial expressions and how they are perceived. Specifically, how they are perceived if the spatial and temporal characteristics are altered slightly. They wanted to use Duchenne and Non-Duchenne smile sequences for this work. While there are many 2D [135, 158], 3D [203, 204], and 4D [80, 253] databases that have been FACS annotated

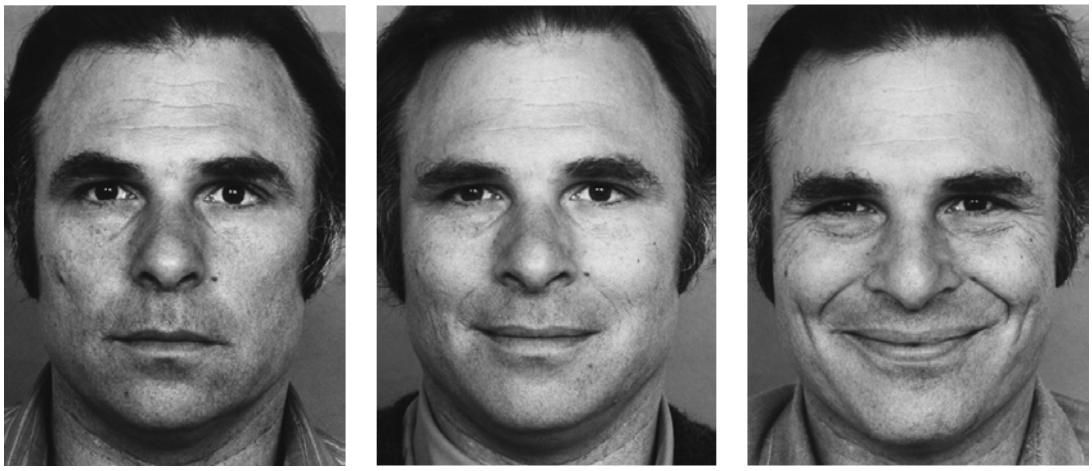
for these types of smiles, we preferred the advantages of having full control over the capture process. Using one of the newest 4D capture systems, the quality of the 4D data competes with, if not exceeds, the quality of older facial expression datasets. Another advantage of capturing the data ourselves was the ability to seamlessly process the captures using our in-lab pre-processing, modelling, and synthesising approaches. These approaches, described in detail in the following chapters of this work, give us the ability to model multiple individuals and manipulate the shape, texture, and facial movements in various ways. This aligns with the research goals of the School of Psychology researchers.

This dataset is referred to as the *Psych Database*. The same four subjects from the conversation data capture took part. They were shown examples of Duchenne and Non-Duchenne smiles, as well as instructed on how to perform the smiles by a FACS-certified coder. Duchenne and Non-Duchenne smiles can be described using Action Units (AU's) using the Facial Action Coding System [103, 104, 105]. The *zygomaticus major* muscle is activated in both smile types and is represented by AU 12 (Figure 3.11(a)), termed *Lip Corner Puller*. What separates a Duchenne from a Non-Duchenne smile is the aptly named *Duchenne Marker*; the activation of the *orbicularis oculi - pars lateralis* represented by AU 6 (Figure 3.12(c)), termed *Cheek Raiser*. The physical attributes of these types of smiles can be seen in Figure 3.12.



(a) AU 12 (b) AU 6

Figure 3.11: Underlying muscles and action unit numbers [105]



(a) Neutral Expression (No AU's Activated)

(b) Non-Duchenne Smile Expression (AU 12 Only)

(c) Duchenne Smile Expression (AU 6 + 12)

Figure 3.12: Neutral, Non-Duchenne, and Duchenne Smile Expression Examples [105]

One-by-one they were captured performing Duchenne and Non-Duchenne smiles, from neutral to peak and back to neutral. This was done for both closed and open mouth smiles. Later, these sequences were validated by the FACS-certified coder. 107 smile sequences were captured. Of those, 42 were validated as being the *requested* smile type (Duchenne or Non-Duchenne). A breakdown of the captured and validate smile types, per subject, can be seen in Table 3.1.

Subject A	**Captured**	**Validated**
D-Closed	7	3
D-Open	8	4
ND-Closed	8	2
ND-Open	8	0
Total	**31**	**9**

Subject B	**Captured**	**Validated**
D-Closed	6	5
D-Open	8	2
ND-Closed	6	2
ND-Open	7	0
Total	**27**	**9**

Subject C	**Captured**	**Validated**
D-Closed	6	5
D-Open	6	5
ND-Closed	6	1
ND-Open	6	1
Total	**24**	**12**

Subject D	**Captured**	**Validated**
D-Closed	6	2
D-Open	7	3
ND-Closed	5	3
ND-Open	7	4
Total	**25**	**12**

Table 3.1: Breakdown of captured and validated smile types per subject

This data will be used in a larger Psychology study led by our colleagues. We have used the data for our own research. Details of how we used this data can be found in Chapter 7.

3.5 Summary

Data collection for the conversation database consisted of two, back-to-back, synced 4D (3D Video) capture systems. Four subjects took part in conversation captures, resulting in 17 minutes of conversations (34 sequences). The data was annotated for conversational expressions by four experienced annotators (two assigned to each conversation). This database is the first of its kind in the world and has been made publicly available to researchers in the community [229]. This data will assist researchers in many novel projects and applications, from analysing the mechanisms of conversational interactions, to synthesising more realistic digital character interactions. A Duchenne and Non-Duchenne dataset was also captured using our 3dMD 4D capture system. This dataset will allow for novel research in the perception of facial expressions and their characteristics.

The 4D tracking and inter-subject registration techniques described in Chapter 5, and shared with the research community in [121], were used on both databases. These techniques allow us to build 3D AAMs and 4D models of expression sequences (Chapter 6). These models are used in Chapter 7 in classification experiments for differentiating frontchannel and backchannel sequences, as well as Duchenne and Non-Duchenne sequences. We also use them in experiments that evaluate our ability to modify and synthesise highly-realistic expression sequences. Finally, a novel approach to modelling conversational interactions using coupled statistical models is described [230], and results from a classification and perceptual experiment are reported.

Before the tracking and registration steps are performed, however, a pre-processing step is required. During development of our processing pipeline, it was discovered that any manipulation of the original data resulted in undesired artefacts. The main reason for these artefacts was due to the multi-view texture maps texture coordinates. As the mesh vertices were updated, so were their corresponding texture coordinates. In many cases, this resulted in texture coordinates for faces being located in separate images of the multi-view texture map. The resulting texture was a blob of colour

due to the texture triangle overlapping images in the texture map. For this reason, a *cleaning* approach was developed to remove problematic items from the original mesh, such as non-manifolds, and to create a single-view, Unified Texture Map (UTM) so that there would be no overlapping issues. The following chapter (Chapter 4) describes the development and approaches used in this cleaning and UTM creation process.

Chapter 4

Mesh Cleaning and Single-View Texture Map Creation

4.1 Introduction

This chapter describes our cleaning process for removing unwanted mesh issues (e.g. non-manifolds) and the process for creating a single-view, Unified Texture Map (UTM). This new texture map allows us to modify a mesh (i.e. modify mesh vertex locations) without creating mesh texture problems. The UTM cannot be created however until the unwanted mesh issues are removed through the cleaning process. The sections below explain why these issues occur and the solutions we've implemented for fixing these issues. The general approach is outlined below and in Figure 4.1. A more detailed *step-by-step* explanation is given near the end of the chapter in Section 4.6.

Pipeline Overview

1. Clean Mesh
    - Remove Non-Manifold Edges (Section 4.3.2)
        - Fill any created holes (Section 4.4.5)

  - Remove Isolated Vertices and Small Components (Section 4.3.1)
- Remove Non-Manifold Vertices (Section 4.3.3)
  - Fill any created holes (Section 4.4.5)
  - Remove Isolated Vertices and Small Components (Section 4.3.1)

2. Create Single-View, Unified Texture Map (UTM) (Section 4.5)
   - Flatten Cleaned Mesh (2D Parameterisation)
   - Calculate UTM *uv* Texture Coordinates for Cleaned Mesh

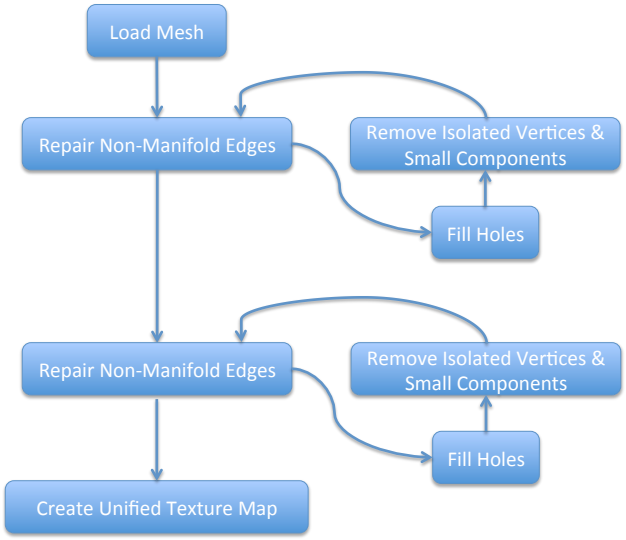


Figure 4.1: General overview of the cleaning and UTM creation process. Step 2 continues in a loop until all its issues are removed, as does Step 3.

One of the major issues with the captured meshes is the existence of *duplicate vertices*. The two leading vendors of 4D capture systems, 3dMD [4] and Di4D (Dimensional Imaging) [5], diverge from the standard OBJ format and implement a duplicate-vertex method. This method allows a single vertex location to be a member of multiple faces and, this, refer to multiple texture coordinates. This approach is used for their state-of-the-art 4D, multi-view capture systems. This approach allows them to seamlessly join the separate camera-views into a single 3D mesh. Figure 4.2 shows the seams involved in a 3dMD frame.

While this approach allows for a very nice and smooth joining of multiple parts of the capture, it becomes a very problematic item when modifying the locations of

Figure 4.2: 3dMD Texture Seams

those vertices, as their texture coordinates will change. Having the ability to modify these meshes is necessary for the type of research we conduct. When the vertices of a mesh are modified, their corresponding texture coordinates are modified, which may result in texture UV coordinates in separate images of a *multi-view* texture map (Figure 4.3) and if this is the case it can cause issues with the mesh texture (Figure 4.4).

For instance, one triangle on the left-side of the nose ridge has Vertex One (V1), Vertex Two (V2), and Vertex Three (V3) with uv Coordinate One ($uv1$), uv Coordinate Two ($uv2$), uv Coordinate Three ($uv3$), respectively. These uv coordinates are contained within one of the three texture images. Also, there is a triangle on the right-side of the ridge of the nose consisting of Vertex 3 (V3-Duplicate), Vertex 4 (V4), and Vertex 5 (V5) with uv Coordinate Four ($uv4$), uv Coordinate Five ($uv5$), uv Coordinate Six ($uv6$), which can be contained within a different texture map image than $uv1/uv2/uv3$. V3, the duplicated vertex will not cause any issues if the initial location stays the same. If the vertex is modified, it will often result in one of

the new texture coordinates being located in a different texture map image than the face for which it is a member. This is visualised in Figure 4.3.

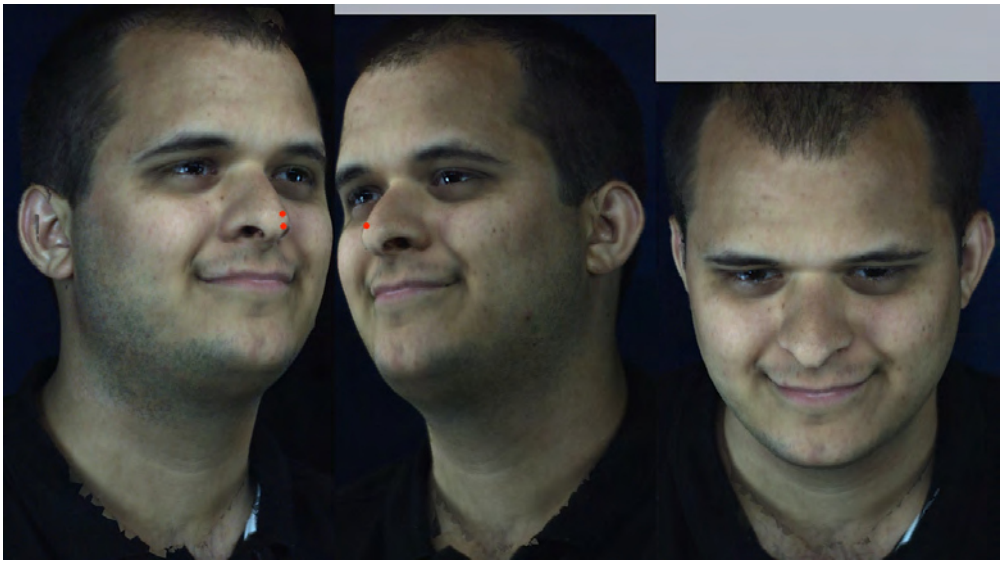


Figure 4.3: Cross-Image *uv* Coordinate Issue

This issue results in a very visually obvious problem: texture blobs. Figure 4.4 shows an example of these black-and-flesh-coloured faces, whose texture coordinates are located in different images of a multi-view texture map. Modifying the location of the vertices is necessary for the registration approaches we used/developed (Chapter 5), and this registered data is necessary for building the statistical models used in this research (Chapter 6). Therefore, it was imperative that we have the ability to modify these vertices without producing new issues with the data.

4.1.1 Potential Solutions

As 3dMD and Dimensional Imaging are the two leading suppliers of 4D scanners in the world, we felt there must be a third-party solution for these issues. While exploring previous research, we did not come across any publicly available information or software for robustly dealing with this specific issue. There was a cleaning approach developed by a lab colleague that would have helped with part of the issue, however, it was developed for relatively low-resolution meshes. When we ran it on our data it took more than half a day to process a single mesh. Given that our data had over 135,000 meshes to process, and this approach did not solve the entire problem, we

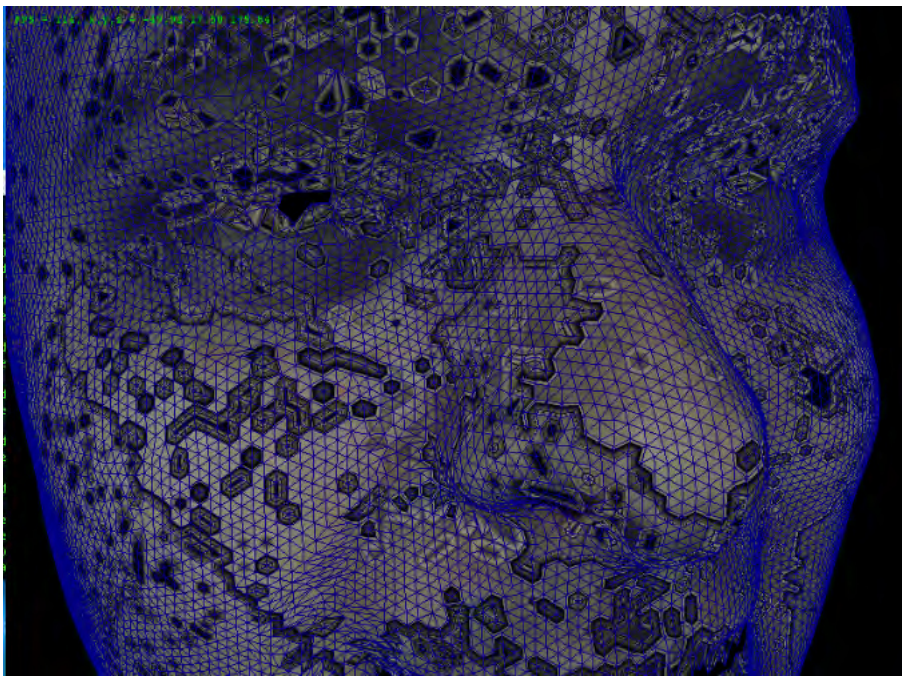
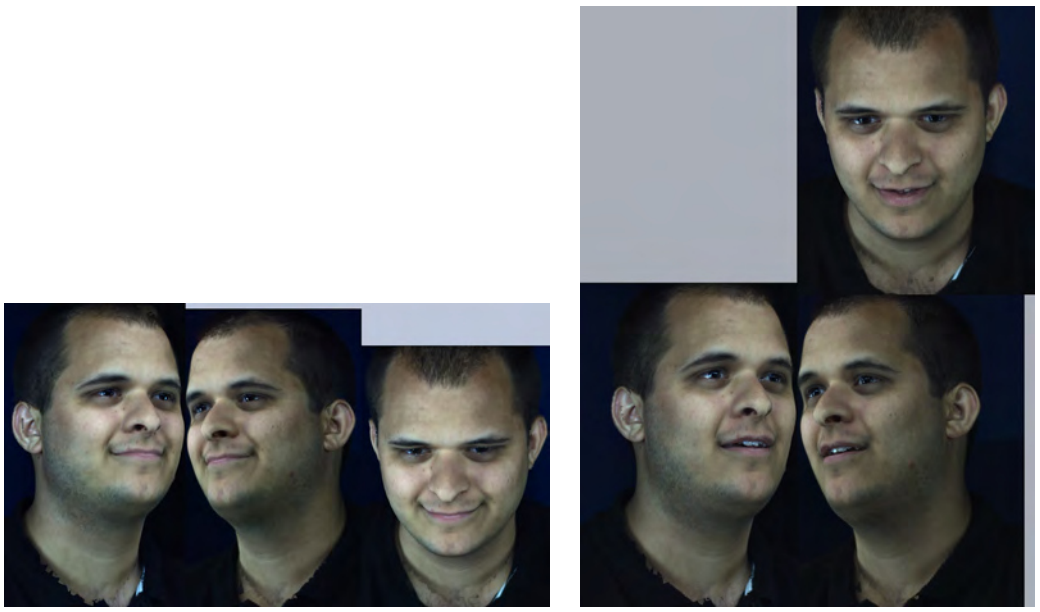


Figure 4.4: Texture Issues on Initial Implementation

decided to develop our own robust, fast, and fully-automatic solution.

Many possible solutions for this issue were considered. One such solution was to break the multi-view texture map into thirds and identify when the texture coordinate of a modified vertex was part of a *uv* triangle that crossed over one of the image boundaries (such as in Figure 4.3). Various corrections could be applied when this issue was identified. The problem with this approach is that the layout of the multi-view texture map is inconsistent. 3dMD's proprietary processing software places the images side-by-side in a single BMP file in such a way as to reduce the total file size. This results in an inconsistent placement of the images, with varying blank spaces. An example of this can be seen in Figure 4.5.

Another solution considered was to remove all duplicate vertices and reconnect the parts of the face by creating close, but non-duplicated vertices. Figure 4.6 shows what the mesh looks like without duplicated vertices. The issue with this approach is that the original values are placed so precisely that any modification of them results in either warping the texture on the face (thus producing an obvious texture seam) or creating *uv* triangles that span multiple texture map images, which is the same issue the process was attempting to fix.



(a) Three-Wide Texture Map (b) Stacked Image Texture Map

Figure 4.5: 3dMD Texture Map Layouts

After exploring different solutions it became clear that for this process, as well as for methods used later in the research, a single-view texture map (we refer to as the *Unified Texture Map*) would be the best solution. That is why the ultimate purpose of the cleaning pipeline, aside from cleaning the mesh of unwanted issues, is the creation of the single-view, Unified Texture Map (UTM). While a *multi-view* texture map (Figure 4.7(a)) contains separate views from each camera, the UTM is a texture map that contains all of the information about the scan in a single, continuous image, as seen in Figure 4.7(b).

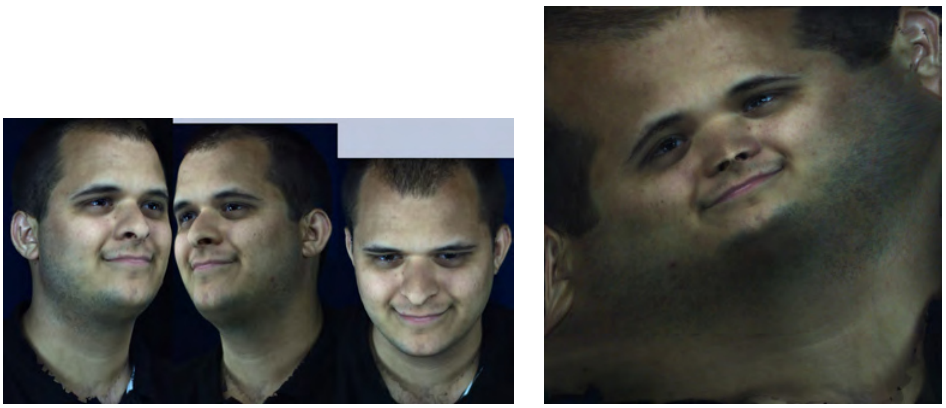
4.1.2 Single-View, Unified Texture Map (UTM) Creation in the Literature

Quite a few works use systems to capture point clouds from one or more cameras and create single-view texture maps through multi-view reconstruction techniques. In [54] a 14-camera and LED light array capture system is used for high-resolution, 30fps captures. 7 stereo pairs are *zoomed-in* and focused on specific parts of the face. This results in an extremely high-resolution capture (1 million polygons) and provides surface texture details which include skin pores, hair follicles, and blemishes.

Figure 4.6: Duplicate Vertices

Beeler et al. [38] describes a similar multi-camera capture and processing approach. In [243], a single Microsoft Kinect camera is used for capturing a point cloud and create a single-view texture map. While these approaches are interesting and provide a step forward in markerless performance-capturing of faces, their systems do not face the issue we, and thousands of other researchers, must overcome when using the state-of-the-art, high-resolution 3D/4D capture systems, from either of the leading vendors. These systems currently provide a 3D mesh and a multi-view texture map.

There are many works that are able to create a single-view, UTM from these types of captures, however their approaches require system calibration information and raw captured images [59, 80, 120, 224, 225]. The most common approach for achieving this with said data and information is to calculate the direction of the triangle normals, then determine the closest camera view of that triangle, and use that information to back-project those triangles onto a plane. In [120], this is used for creating texture maps from multi-views of non-face items (e.g. buildings). In [224], a GPU-based processing approach is used for creating a 3D mesh and texture maps from a multi-view capture of human hand data. Cosker et al. [79, 80] uses this technique for creating single-view *uv* texture maps from two-view captures of faces.



(a) Three-View Texture Map

(b) Single-View, Unified Texture Map (UTM)

Figure 4.7: Three-View Texture Map and UTM

These two works are the closest to ours, because they use facial expression data captured on a similar 3dMD, 4D (3D dynamic) capture system.

These approaches are vendor-specific and require extra information from the capturing process (often times information that is not made available in the release of public datasets). A robust multi-view texture map to single-view texture map processing approach that only requires the 3D mesh and texture map is necessary for a variety of reasons. First, while computation time is not reported in the works which use these techniques for texture maps of faces, it can be expected that working on a triangle-by-triangle basis is computationally intensive. Also, such an approach is only useful if working with data captured in-lab. As well, a change of systems or vendors may require a substantial change to the pipeline. Acquiring data from other researchers might also require a new approach if they have not implemented a multi-view to single-view approach. The novelty of our approach (Section 4.1.3) is that it creates the same type of single-view output as these other approaches without requiring any system information, which may have been lost, corrupted, or otherwise not available (as described above).

The approaches listed above were performed on systems that provide consistent layout for multi-view *uv* texture maps. In newer capture systems, specifically those with more than two cameras, the layout of the texture map, for each frame, is unknown. The overlapping of views is also more complex with the additional views.

It is reasonable to expect problems implementing the approaches described in [79, 80]. The technique described in this chapter is system-agnostic and does not require any configuration information about the capture system used (e.g. camera calibration information). This means any 3D surface mesh and multi-view texture map can be processed, resulting in a single-view texture map. As an added bonus, the processed mesh is cleaned of imperfections (e.g. non-manifolds, isolated vertices, and small components). This may be particularly useful to researchers who require such meshes. To our best knowledge, this is the only fully-automatic, robust approach for creating single-view texture maps from multi-view texture maps from 3dMD and Di4D capture systems that does not require system-specific information.

4.1.3 Our Approach

The UTM is created by simply flattening the surface mesh onto a plane (2D planar parameterisation, see Section 4.5 for details), filling the mesh faces with the appropriate texture values, and rendering the mesh as an image. This new image becomes the texture map used by the cleaned 3D mesh. In order for this flattening to occur the mesh needs to be cleaned of problematic issues, such as non-manifold vertices and edges, isolated (floating) vertices, and small components (separate mesh areas disconnected from the *main* mesh). Unfortunately, due to the complex and unpredictable geometries of high-resolution (30,000 vertices, 50,000 faces) 3D meshes, the process of finding and fixing these issues can be very challenging. There are so many combinations of issues that can occur that a robust solution is needed, as to deal with both common issues and those which exist but are not experienced during development of the cleaning process. The proposed solution, while not perfect, is a fully-automated process that has been used to clean over 135,000 3D frames. These frames consist of four subjects captured using two separate 3dMD systems. While the individual steps used in this pipeline are trivial, both the data store structures used and the combination in which the steps are taken are novel approaches. The challenge in creating a robust and fully-automated cleaning pipeline was in determining the

order of the steps used for solving the complex issues that do (and could) occur in meshes, as well as keeping track of the original and duplicated mesh items.

To summarise: a Unified Texture Map will allow us to bypass the issues caused with duplicate vertices. The UTM is created by flattening each mesh. The mesh can only be flattened after issues, such as non-manifolds, are removed. Therefore, the goal of this cleaning pipeline is to remove those issues, flatten the mesh, and create the cleaned mesh with its UTM. The output of the cleaning step of the processing pipeline are meshes that do not contain issues (such as non-manifolds) and do not contain duplicate vertices or faces. The output of the parameterisation process of the processing pipeline is the creation of the single-view, UTM. Please note that the figures in this chapter (those not of a person) have been manually created for clearly explaining concepts, and were not taken from any actual meshes that were cleaned using the pipeline. Section 4.7 evaluates our approach on seen and previously unseen 3D/4D data from 3dMD and Di4D systems.

4.1.4 Global and Local Cleaning Methods

In order to flatten the mesh, it must first be cleaned of imperfections that will interfere with the flattening process. These issues include non-manifold vertices and edges, isolated vertices, and floating vertices. The original vertices of the mesh should persist throughout the cleaning process, since they are used with the multi-image texture map to texturise the flattened mesh. A few vertices, such as non-manifold vertices, may be removed during the cleaning process, but no new vertices are created. Creating new vertices would, in many cases, introduce new texture mapping issues. By removing problematic vertices and polygons, and then repairing the holes created by those removals, we can fix the specific issues that are keeping the mesh from being flattened. In the cleaning process vertices are never added, only removed if problematic. Any resulting holes are filled by creating polygons which use existing vertices. It is important to not add new vertices because doing this is likely to add texture issues (due to overlapping texture triangles), which is what we are trying to

avoid in the first place. Given that, for our data, most of these issues occur in areas like the neck and ears, a visually-appealing filled hole is not a priority. Repairing the issue (identifying, fixing, and filling any resulting holes) needs to be fast, robust, and efficient.

There are many global and local approaches for repairing meshes. A comprehensive overview of global and local approaches for different mesh issues can be found in [19]. A few are described below, including the approach we adopted for our cleaning pipeline.

Global strategies [43, 44, 50, 134] for mesh repair often provide robust solutions, but tend to require the mesh to have certain characteristics (e.g. free of singularities, self-intersections, and degeneracies; consistent polygon orientation, etc.). These strategies also tend to modify the original meshes vertices and faces due to things such as smoothing and remeshing.

A very popular global mesh repairing approach is that of [134]. It is an efficient and robust approach for repairing volumetric meshes. The author claims that many previous approaches for mesh repair were either too computationally expensive [172] or fail in specific instances [110, 174]. The approach used by the author creates an intermediate volume grid and generates an output surface by contouring the grid. This approach works for volumetric meshes, specifically polygon soups. This approach is "memory-less" and uses a divide-and-conquer approach. While it is reported to be robust and efficient, it solves a different problem than we require for our meshes. We are unable to use this approach for our data and application for three main reasons. First, our 3D frames are surface meshes, not volumetric meshes. Second, they are not polygon soups; the polygons have a relationship with one another in regards to the corresponding texture map. Third, the approach outlined in [134] does not retain the exact vertices of the existing polygons, the approach used constructs new polygons to create a perfectly repaired mesh, through the use of a volumetric grid. For our data we require the original vertices to persist throughout the cleaning process, as they are crucial in our final step. In this final

step we parameterise (flatten) the cleaned mesh to create the single-view unified texture map. It is for these reasons that we have not implemented this popular mesh repair method.

The approaches of [43, 44, 50, 172, 186] are similar global approaches for repairing volumetric meshes, but their techniques do not preserve the original mesh vertices and faces. These approaches work on the mesh globally, that is, most do not identify problematic areas (e.g. non-manifold vertices, isolated vertices, etc.) and attempt to repair those specific instances. They freely remove or add vertices and polygons to achieve a clean mesh. This is fine for their applications, but not for our use. The multi-image texture map that each mesh corresponds to has specific faces and by removing surface mesh polygons.

It is clear that we require a local approach for repairing issues with our mesh frames without modifying the entirety of the original mesh (thus preserving the original vertices and polygons). Local strategies for mesh repair allow for specific issues to be fixed without compromising the vertices and faces of the original mesh. One downside of using local methods is that they are known to have the possibility of producing new flaws. By creating an intelligent and iterative process however, these issues can typically be resolved.

One well-known and widely-used local approach to repairing meshes is that of Dr. Marco Attene [18]. The approach outlined only modifies the mesh in areas where issues (e.g. degenerate and intersecting elements) have been detected. Such an approach preserves the original mesh as much as possible. The algorithm described in the paper iteratively removes undesirable elements and patches any holes created by the repairing process. This approach showed to be computationally efficient on the low resolution meshes that were tested and sufficient for repairing issues. This local, iterative approach is the type used in our work for repairing mesh issues. That is, finding an issue, removing it, and then filling any holes created by the process. Removing non-manifold edges and small components is done as described in this work (using a triangle-adjacency matrix). The removal of non-manifold vertices is

achieved using the *JMeshLib* library [17], and was developed by the same author [18].

The only divergence from the approaches used in [18] from our approach is the filling of mesh holes. In [18], an approach from Barequet and Sharir [25] is used. While this approach is useful because it does not create new vertices to fill the hole, it requires many steps for hole identification and filling (e.g. re-orientating the mesh polygons, matching border portions, etc.) and is more focused on CAD volumetric mesh models. We required a much simpler approach, which is why we implemented the *anchor-vertex* based approach, as described in 4.4.5.

Our approach for filling the hole is simple and efficient, but like many other local mesh repair approaches it is not immune to producing self-intersecting faces. In certain circumstances, such as a 'V' shaped hole, self-intersecting faces can be produced by this hole filling approach. While other issues that are created from hole filling, such as non-manifold edges, will be identified in a subsequent check (our process is iterative until all issues are fixed), we do not explicitly check for self-intersecting faces. Although the cleaning pipeline was sufficiently successful without checking for self-intersecting polygons, this is a step that should be added in a future version of the cleaning pipeline. This should be a simple addition, as the functionality exists within the third-party software currently used for identifying mesh issues (iso2mesh [106] and JMeshLib [17]). The *minimum area triangulation* approach for hole filling of [139] should also be considered as a future implementation for the cleaning pipeline. While it does not offer any significant processing time advantages nor avoid the self-intersection issue, it is an approach that fill holes in a more natural and even manner than our current approach.

The following sections describe in detail the mesh repair approach we implemented.

4.2 Pipeline Storage Structures

Surface mesh data is the type of data to be processed using this cleaning pipeline. In this data, a *vertex* consists of three values (x, y, z) which describe its location in 3D space. The vertices are indexed. These index values are used to define the faces structure. A *face* consists of three vertex indices, which describe how edges connect the three vertices. Each vertex also has a *texture coordinate* which specifies the u and v location in a 2D texture map image. A vertex index and the texture coordinate index correspond (i.e. they have the same value). Therefore, any time there is a modification to the data which changes the vertex indices, an update to the faces structure and texture coordinate structure is required.

4.2.1 Pipeline Structure Overview

The first step in the pipeline is to create the data structures used for the cleaning process. This includes a structure for the "kept" vertices, faces, and uv texture coordinates, along with structures for "duplicate" vertices, faces, and uv texture coordinates. These structures are created in such a way as to keep track of the relationships between the "kept data" (i.e. the single version of the duplicate that is kept) and the "duplicate data" (i.e. the duplicate vertices and their corresponding faces and texture coordinates). The duplicate vertices/faces/texture coordinates will be used in the final step, once the mesh is flattened, to provide texture coordinate information for the flattened vertices. However, identifying mesh issues and fixing them is much more complex if duplicate vertices and faces are a part of the mesh being cleaned. For this reason, the duplicate values are stored in separate structures and updated when their corresponding "kept" values are modified.

To create the *kept data* structures, the first instance of a vertex with duplicates is stored, along with its corresponding faces and texture coordinates. The other (duplicate) vertices are moved to the *duplicate vertices* structure (as index values which "point" to the kept vertex's index) and information about their corresponding

faces and texture coordinates are added to their respective structures (i.e. duplicate faces and duplicate texture coordinates). As previously stated, the "kept" vertex is used as a key for the duplicate vertices structure. The *duplicate texture coordinates* structure has a 1:1 correspondence with the duplicate vertices structure, via their indices. The *duplicate faces structure* contains pointers to the face index in the *kept face* structure, for the face it was a member of; the position (e.g. 1, 2, or 3) in that face where it was located (which is now taken up by the kept vertex's index), and the index of the vertex in the *duplicate vertices* structure. This identifies which duplicate vertex should be in the face and is used later when adding back duplicate values. The duplicate face structure does not have a 1:1 correspondence with the duplicate vertices structure because duplicate vertices can be part of multiple faces.

Given this structure, the kept vertex information is treated as the main item to modify or delete, and the related structures are updated based on the addition, modification, or removal of vertices. Many different approaches to data storage for the cleaning pipeline were evaluated, however, this method provided a balance between structures whose data and connections could be computed quickly and efficiently, with a structure which was easy to follow at a higher-level (which also assisted in debugging issues).

When a vertex is deleted, the corresponding texture coordinate must also be deleted and any faces referencing that vertex are removed. This is known as the *updating* process. Since these structures use indexing of vertices, when a vertex is deleted all the values greater than it must be *shifted* down by one, in the structures which reference vertex indices. This is known as the *shifting process*. One of the reasons the kept/duplicate structure approach was chosen is because it made this process, which is logically simple but can be quite complex to implement, much easier. Figure 4.8 shows an example of when this shifting process is not done properly for all structures. The information that describes which vertices are connected (the mesh faces) is incorrect due to a failure in the shifting process (i.e. the vertex indices in the face structure were not updated to reflect the removal of vertices from the mesh). This results in odd and incorrect faces that connect vertices from around the

mesh. The *deletion and shifting process* is quickly and efficiently completed using highly-vectorised Matlab code.

Figure 4.8: Shifting Update Error (Front and Back)

4.2.2 Pipeline Structure Examples

The storage structures are an important part of this cleaning pipeline. They are referred to often in the following sections. Therefore, this section will give visual examples to help better explain how the storage structures are organised and used. Figure 4.9 shows an example surface mesh structure (2D view) that will be used to help explain the storage structures that were created for use in the cleaning pipeline. The examples in this section have been created to clearly describe the approach and are not taken from any "real" mesh data.

The vertices of this example mesh are stored in the *kept vertices* structure (Table

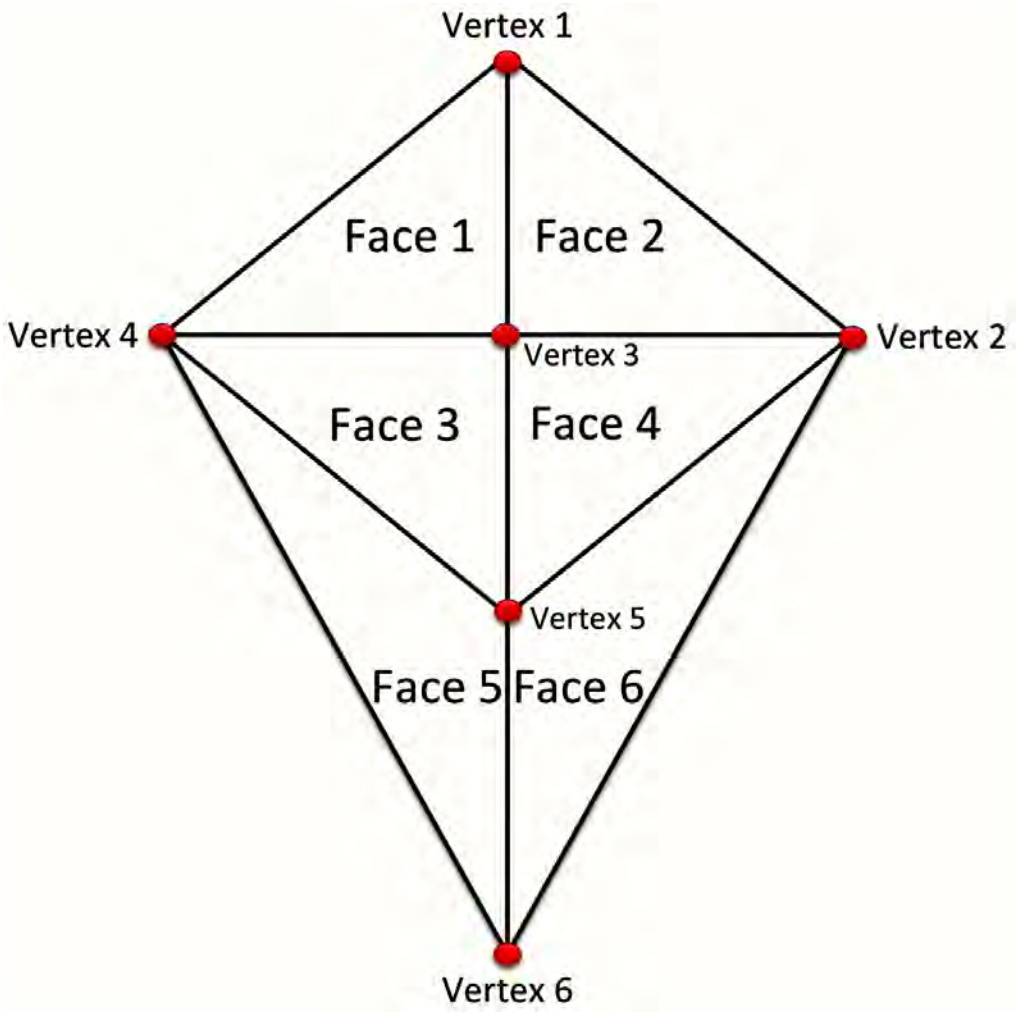


Figure 4.9: Mesh created for describing mesh cleaning pipeline structures

4.0(a)). Each vertex has an x, y, z value and is indexed by its location in the structure. The *kept texture coordinates* structure (Table 4.0(b)) contains the uv coordinate values for the texture map. Each index in this structure corresponds with the indices in the vertex structure.

The *kept faces* structure (Table 4.2) contains the vertex indices which make up the faces of the mesh. For instance, *Face 1* (whose index value is 1), is made up of *Vertex 1*, *Vertex 3* , and *Vertex 4*.

The duplicate structures are important because they will allow us to clean the mesh of issues that will keep us from flattening it. It is important to keep track of where the duplicates were because they will need to be added back at the end of the pipeline when the Unified Texture Map is created. For the *duplicate vertices* structure (Table 4.2(a)), each entry represents one duplicate vertex, where the value is that of the

(a) *Kept Vertices* Structure

Vertex Index	x**-value**	y**-Value**	z**-Value**
1	27.56	22.36	-82.31
2	44.67	18.66	-74.34
3	38.21	28.45	-62.32
4	32.44	22.14	-22.67
5	18.32	44.23	14.72
6	88.62	24.11	44.24

(b) *Kept Texture Coordinates* Structure

T.C. Index	u**-value**	v**-Value**
1	12.45	44.24
2	14.65	42.22
3	12.98	44.11
4	9.42	34.64
5	8.55	32.45
6	10.22	34.21

Table 4.1: Example of the *kept vertices* and *kept texture coordinates* structures

Face Index	**Vertex 1 (Index)**	**Vertex 2 (Index)**	**Vertex 3 (Index)**
1	1	3	4
2	1	2	3
3	3	4	5
4	2	3	5
5	4	5	6
6	2	5	6

Table 4.2: *Kept Faces* Structure

kept vertex's index (i.e. it acts as a pointer, of sorts, to the kept vertex).

The *duplicate texture coordinates* structure (Table 4.2(b)) contains the texture coordinate values for the duplicated vertices. Each index in the duplicate texture coordinates structure corresponds to the indices of the duplicate vertices structure. There are three images in the texture map used in this work, which means a vertex could have up to three different texture coordinates (one for each image). Therefore, the duplicated vertices structure may have (up to 2) repeating values (e.g. Index 1: 1, Index 2: 3, Index 3: 3).

The *duplicate faces* structure (Table 4.2(c)) is not like the kept faces structure. It contains three rows, but these rows do not contain three vertex indices. The first row specifies the index of the face in the "kept" faces structure that the duplicate vertex was a member of (and where the kept vertex now resides). The second row identifies

the index of the kept vertex in the face (e.g. Vertex 1, Vertex 2, or Vertex 3). The third row contains the duplicate vertices index value for the vertex being kept track of. The duplicate faces structure does not contain a 1:1 mapping of indices to the duplicate vertices structure because a duplicate vertex can be a member of many faces; therefore multiple entries in the face structure may point to the same index in the duplicate vertices structure.

(a) *Duplicate Vertices* Structure

Structure Index	**1**	**2**	**3**	**4**
***Kept Vertices* Index**	1	3	3	5

(b) *Duplicate Texture Coordinates* Structure

T.C. Index	u**-value**	v**-Value**
1	12.45	44.24
2	14.65	42.22
3	12.98	44.11
4	9.42	34.64

(c) *Duplicate Faces* Structure

	1	**2**	**3**	**4**	**5**
***Kept* Face Index**	2	2	3	4	5
***Kept* Face Position**	1	3	1	2	2
***Duplicate* Vertex Index**	1	2	2	3	4

Table 4.3: Examples of the *Duplicate* Structures

4.3 Cleaning Process

The following subsections discuss the mesh issues that the cleaning pipeline was created to resolve. Section 4.4 describes how the pipeline deals with holes created by the removal of the mesh issues, and Section 4.5 describes how the Unified Texture Map is created using the newly cleaned mesh. Section 4.6 describes the specific order in which these steps are taken.

4.3.1 Small Components and Isolated Vertices

Two issues that are not difficult to identify and fix are *small components* and *isolated vertices*. Small components are sections of the mesh that are made up of legitimate vertices and faces, but are not connected to the main mesh. These small component issues commonly occur in messy areas of the mesh capture, such as the neck and

shoulders. These issues are often caused by clothing. It is relatively simple to remove these items. A vertex-face adjacency matrix is computed. This provides information about which faces are connected to which vertices. Components are identified this way and the largest component is kept, while all others (*small components*) are deleted. Unless there is an error in the capturing process, the largest component will always be the head of the subject. The update and shifting process is then performed.

Isolated vertices (also commonly referred to as *floating vertices*) are vertices which are not members of any mesh faces. Using the vertex structure and faces structure, any vertex structure indices that do no exist in the faces structure are identified as isolated vertices. These vertices are deleted and the update and shifting process is performed.

4.3.2 Non-Manifold Edges

One of the most challenging issues to identify and fix is that of non-manifold edges. A non-manifold edge exists when a single edge is occupied by more than two faces (Figure 4.10). A non-manifold edge therefore consists of two vertices. To find non-manifold edges, a vertex-face adjacency matrix is computed. This provides information about which faces are connected to which vertices. If any pair of vertices are connected to more than two faces, they contain non-manifold edges (i.e. a NME for every connected face, after two faces). Given the complexity of the mesh geometry a generic approach for determining which face (or faces) makes up the offending non-manifold edge is difficult. For this reason, all vertices and faces connected to the non-manifold edge are removed. The deletion and shifting process then occurs.

After the deletion and shifting process, any holes resulting from the deletion process are identified and filled (as described in Section 4.4.5). The hole identification process is much more challenging than at first glance. The non-manifold edge cleaning process is an iterative process that repeats until non-manifold edges are no longer detected.

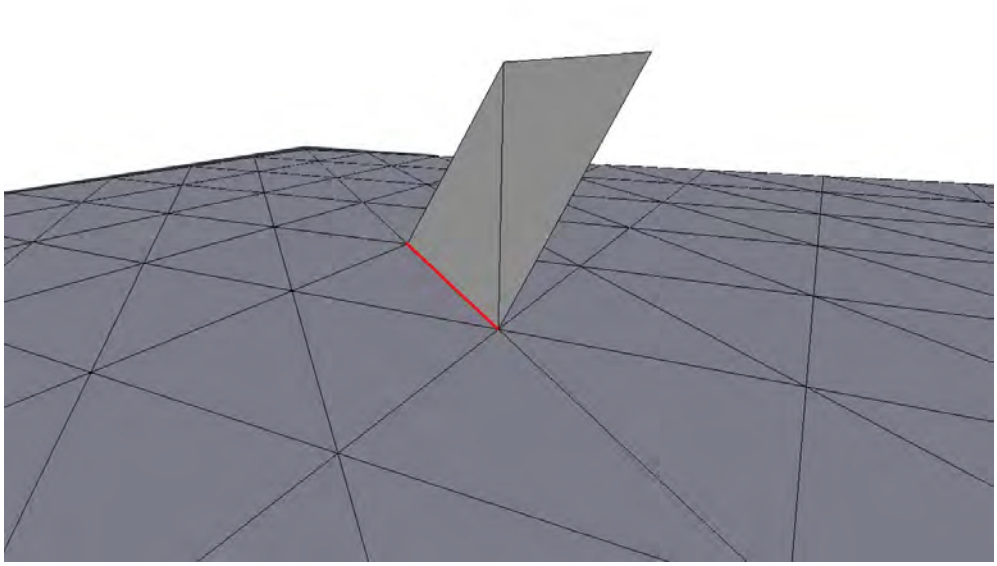


Figure 4.10: Example of a Non-Manifold Edge

4.3.3 Non-Manifold Vertices

Identifying and removing non-manifold vertices can also be challenging. A non-manifold vertex exists when a vertex connects two faces that do not share an edge (Figure 4.11). Non-manifold vertices are identified using the third-party Matlab toolbox *Iso2Mesh*, [106] which uses the *JMeshLib* library [17] to identify non-manifold vertices. When provided with the mesh, this function returns a vertex structure with duplicate vertices for those identified as non-manifold vertices. Any vertex with a duplicate is identified and the matching vertex values in the *kept vertices* structure are found. This identifies the non-manifold vertices and at this point the repair process can commence.

The repairing process is straightforward. The non-manifold vertex and faces connected to that version of the vertex are deleted. All structures are updated and the shifting process described previously is performed. After the deletion and shifting process, any holes resulting from the deletion process are identified and filled (as described in Section 4.4.5). The generic approach to this issue is able to remove the problematic items without negatively affecting the mesh. As with the non-manifold edge cleaning process, the non-manifold vertex cleaning process is an iterative approach that

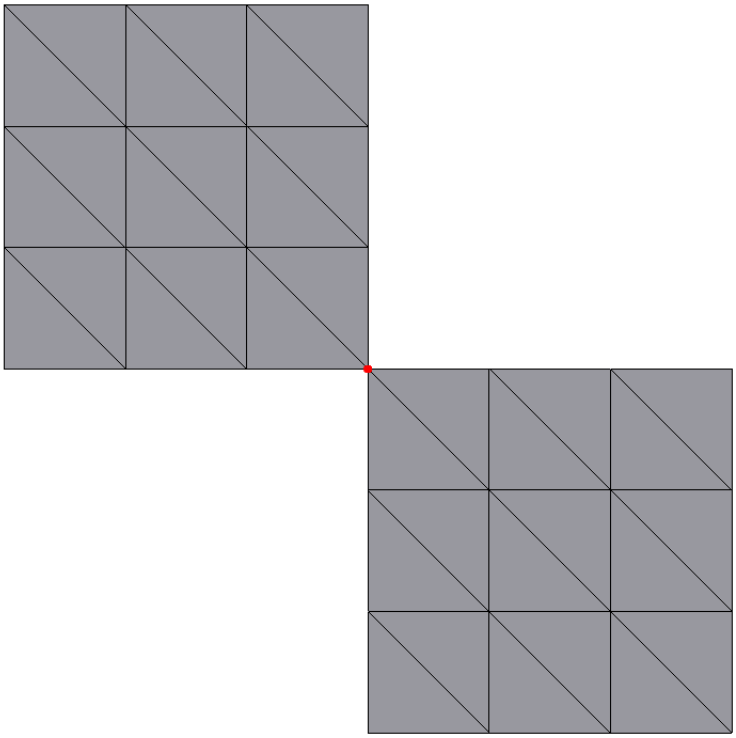


Figure 4.11: Examples of a Non-Manifold Vertex

repeats until no more non-manifold vertices are detected.

4.3.4 Non-Manifold Repairing Approach

One important item to note is that a single non-manifold issue is fully fixed (identified, deleted, hole filled if necessary) and then the identification function is called again. This continues until there are no more non-manifold issues. This one-by-one approach is done instead of removing all non-manifolds and *then* fixing the holes they may create. The removal of non-manifolds can produce new mesh issues, which is why the all-at-once approach can result in unwanted complications, such as *mesh erosion*. Figure 4.12 shows the gaps in the mesh that can be created with this approach and Figure 4.13 shows another instance where the holes have been filled only after all non-manifolds were removed. By not filling holes when they are created, the new issues may cause the cleaning process to "eat away" at the mesh.

This approach, of deletion and hole filling was found to be the most generalised but successful way of dealing with the variety of complex issues arising from non-manifolds

Figure 4.12: Mesh Erosion Along the Texture Seam

in a dense mesh. Other approaches, such as only deleting the faces connected to non-manifolds, were evaluated but they consistently failed to repair many issues in different situations. Often times the removal of a face produced new, sometimes more complex, issues to identify and repair. For this reason, the approach of deleting vertices and their connected faces, and then filling the resulting holes was chosen.

4.4 Identifying and Filling Holes

As discussed in Section 4.3.4, holes are often the result of the deletion of non-manifolds. There are many types and configurations of issues and resulting holes that occur when removing non-manifold edges. This section will discuss some of the most common issues and their implemented solutions. This is not a comprehensive list of hole issues, but the generic approach developed for identifying and fixing hole issues appears to be robust to the many different types of hole issues that occur in the meshes used in this work.

Figure 4.13: Result of Filling a Large Hole Resulting from Erosion

4.4.1 Detection of Regular Holes

A hole is not necessarily created every time a non-manifold is removed. For instance, removal of a non-manifold edge located on the mesh boundary would result in the boundary of the mesh being reduced, without a hole being created. For reasons such as this, an approach for identifying mesh holes was developed. Figure 4.14 shows an example of two holes in a surface mesh.

Holes are detected by storing the vertex indices of the removed non-manifold connected vertices and faces (i.e. the vertices that are immediately connected to the non-manifold's vertices and faces referencing those vertices). These vertices are all potential members of a possible hole. Outright identification of a mesh hole is challenging. A straight-forward and efficient way to identify holes is to compute *boundary vertices*. These are vertices which are part of a pair of vertices that are only connected to one mesh face. This will identify vertices that are part of a hole, however, every vertex on the edge of the mesh is also a boundary vertex. By

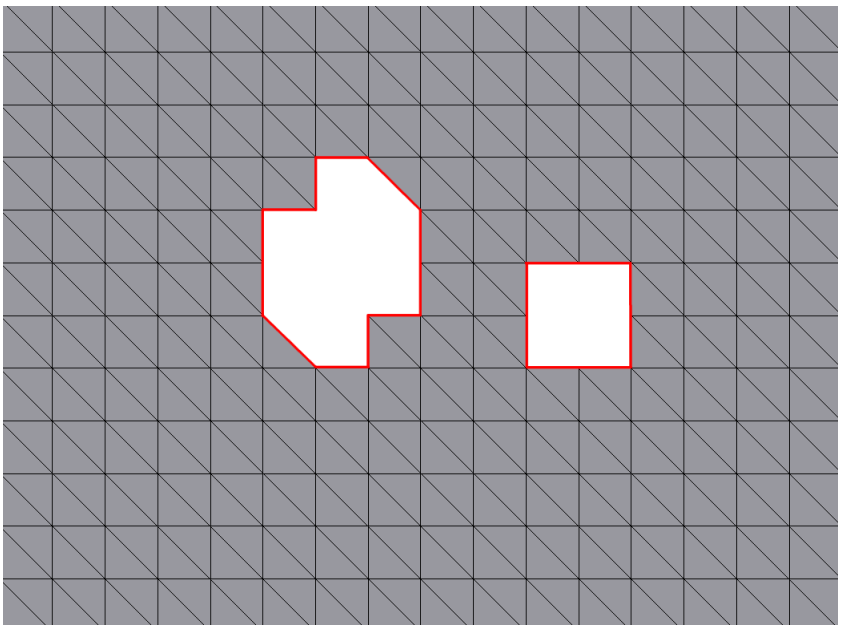


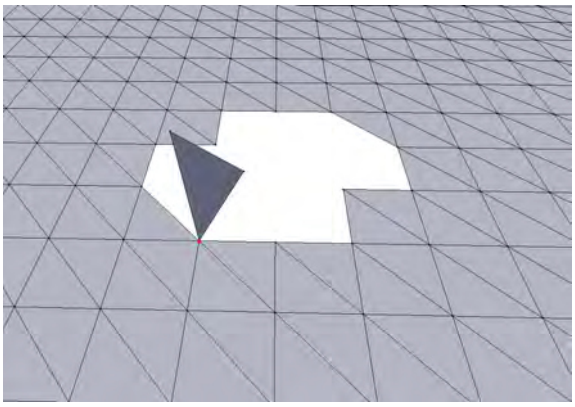
Figure 4.14: Example of "Regular" Holes to be Filled

computing all boundary vertices for the mesh and then cross-referencing that list with the potential members list, the vertices which *may* make up a hole are identified. This list is called the *neighbour list*. An iterative process of traversing the neighbour list vertices to identify holes is performed to determine which vertices are part of a *closed loop* (greater than 2 vertices). This recursive process traverses the loop and removes branches (a vertex in the neighbour list that is only connected to one other vertex) until a closed loop exists. If there are less than three vertices left at the end of this process, no hole has been detected. This is possible, among many other instances, if no hole was created but the neighbour list contained vertices that were part of the removed issue and also on the boundary. This closed loop hole is then filled (see Section 4.4.5 for details).

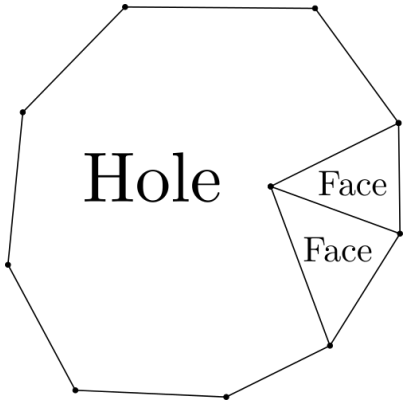
This approach is sound for "regular" holes, but does not work for every type of hole. In many instances the holes created by removing non-manifolds consist of new or not-yet-repaired non-manifold vertices or edges that are part of the hole in some way. The variations of similar issues which occur for different reasons makes it very challenging to develop a general approach to detect, repair, and fill holes; not to mention the inconsistent geometries and the vast number of vertices and faces in each mesh. The following sections will discuss the identification of and repairing solution for some of the more common "abnormal" hole types.

4.4.2 Holes with Faces

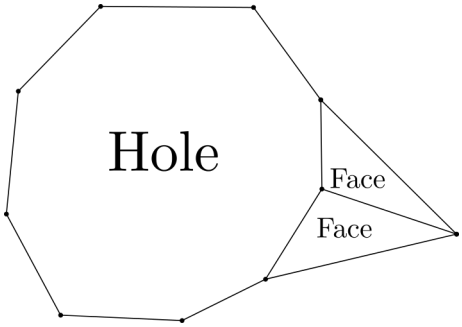
There are various instances when the holes from removing non-manifolds result in "holes with faces". That is, the neighbour list contains vertices which are part of the hole, but are also connected in such a way that includes faces as part of the hole. Traversing the hole, which is a way of identifying a closed hole that is ready to be filled, is for obvious reasons problematic. Issues of this type can be detected by identifying any three vertices of the neighbour list that are connected to one another. In Figure 4.15(a), a face is connected to the hole by a non-manifold vertex. Figure 4.15(b) and Figure 4.15(c) are instances of otherwise allowable faces, but where one of the neighbour list vertices should not technically be part of the hole.



(a) Face Connected by Non-Manifold Vertex



(b) Faces in a Hole



(c) Faces Outside a Hole

Figure 4.15: Hole Filling Issues

If three vertices of the neighbour list are found to be connected to one another, the face(s) which contain those three vertices are deleted. An updating and shifting process is performed and the new hole is checked for the issues described in this section, such as isolated vertices. For Figure 4.15(b) and Figure 4.15(c), an isolated vertex will be detected and removed. Once no further issues are detected with the

hole, and it can be traversed (i.e. it is a closed loop), it is ready to be filled.

The method described was found to be a strong approach for dealing with the variety of combinations that can occur when three vertices of a hole are connected to each other. The density of the mesh (approximately 30,000 vertices and 50,000 faces) means that the few faces that are removed in the process are inconsequential to the shape of the mesh. As well, the hole filling process replaces those areas with new faces.

4.4.3 Pinched Holes

Pinched holes occur when a non-manifold vertex separates two holes that need to be filled (Figure 4.16). This type of hole is identified by checking if any neighbour list vertex is connected by an edge to more than two other vertices in the neighbour list.

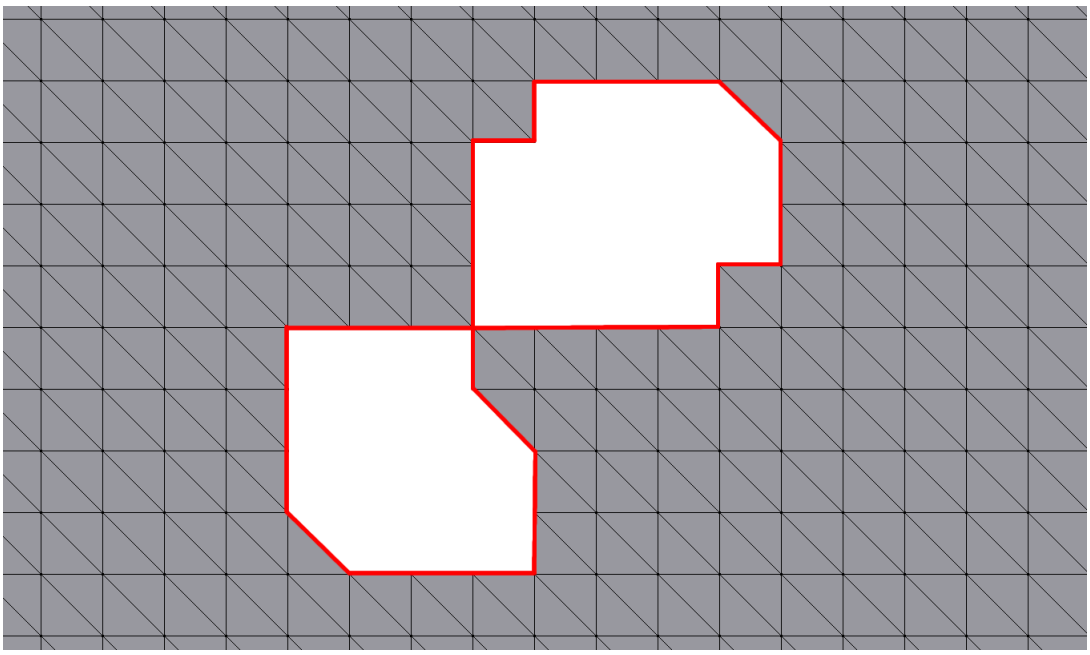


Figure 4.16: Pinched Holes

Many approaches to solving this issue were considered. The difficulty is, there is no useful information on a structured order of the vertices and traversing such a structure can be done in a variety of ways (e.g. one hole loop, figure-eight, etc.). The example figure (Figure 4.16) is just the simplest example of this issue. The decision was made to not attempt to traverse this type of hole at all, but rather approach it

as a non-manifold vertex issue. That is, the non-manifold vertex connecting the two holes is removed, similar to the process described in Section 4.3.3. This new, slightly larger hole can now (typically) be traversed like a "normal hole" and if successful, this hole will be part of a *closed loop*. The hole can then be filled. If it is not a closed loop, it is checked for the issues discussed in this chapter and the repairing process continues until it is a closed loop.

4.4.4 Disconnected Vertices

One of the more complex common issues to solve is that of *disconnected vertices*. These issues were extremely challenging to identify manually (i.e. visualising using a 3D mesh viewer) and developing an automatic, generic approach required attempting many different possible solutions.

Disconnected vertices exist when a non-manifold item is removed and the resulting neighbour list contains multiple vertices which are connected through other neighbour list vertices to the main hole, but are not part of the hole themselves. This typically occurs when vertices from the neighbour list are also on the boundary (so they would match the cross-referencing step used to identify neighbour list vertices), but are not part of the actual hole.

Figure 4.17 shows a 2D example of another version of this issue. The red vertices are those which are part of the neighbour list, either because they were part of the original issue or because they are on the boundary. The blue vertices are not part of the neighbour list. With this issue it is difficult to determine if the vertices on the neighbour list should be part of the hole or not, partially because of the location of the blue vertices. The location of these blue vertices creates an issue similar to the *pinched holes* example (Section 4.4.3), except that one of the two "holes" is filled with faces. Therefore, the vertex "pinching" the two together is not actually a non-manifold vertex, like in the pinched holes example. One of the challenges here is that it is difficult to detect these issues because these vertices are not part of any "illegal" items, such as non-manifolds (which can be detected and repaired).

This issue is also different from the issues discussed in Section 4.4.2 because these faces are not connected by three vertices from the neighbour list, which makes it challenging to automatically detect these faces.

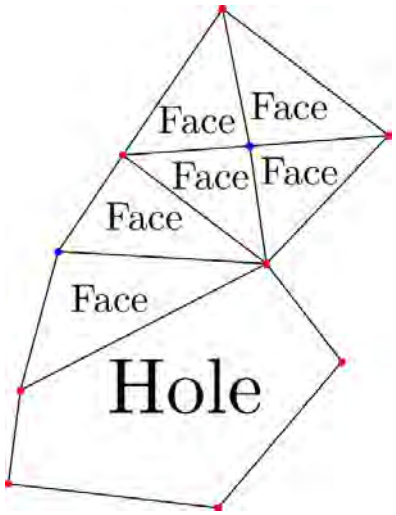


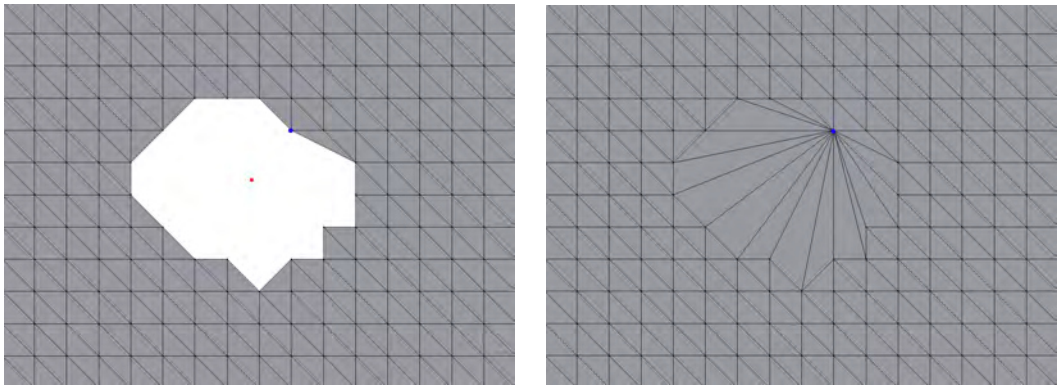
Figure 4.17: Disconnected Vertices - 2D Example

The issue of disconnected vertices can be solved using the rule that no two vertices, that are not directly connected by an edge, can be members of the same face. The top part of Figure 4.17 shows how red vertices that are opposite each other are part of the same face because of the centre, blue vertex. By using a vertex-face adjacency matrix to calculate which neighbour list vertices are members of the same faces but are not connected by an edge, the vertices that should not be part of the hole can be eliminated. If the remaining vertices are part of a closed loop, the hole filling process can begin. Otherwise, the neighbour list is checked again for issues.

4.4.5 Filling Holes

Fixing non-manifolds and identifying/fixing the holes created by them is the challenging part of the cleaning process. Filling holes is very simple. To help ensure that each hole was filled completely and in a structured manner (as not to produce unnecessarily large faces) the approach of using an *anchor vertex* to fill the hole was chosen. To do this, the hole centroid is calculated (the red dot in Figure 4.18(a)) and the closest vertex, the anchor vertex, is found (the blue dot in Figure 4.18(a)). New faces are created by connecting, in order (based on the list created when the hole was

successfully traversed) each pair of vertices to the anchor vertex, iteratively, until the entire hole has been completely filled. Figure 4.18(b) shows the hole after it has been filled. Originally, each pair of vertices were connected to the centroid itself. However, given the high variability of hole geometry and the number of vertices involved, this approach was not robust enough. The "anchor vertex" approach was chosen due to its ability to fix holes of all types without creating new issues (non-manifold edges/vertices, isolated vertices, small components, etc.). If there was more than one hole to fill, as in Figure 4.14, the process filled one hole, then the next, and so on until all holes in that cleaning iteration were filled.



(a) Hole Centroid (Red) and Closest Hole Vertex (Blue), termed *anchor vertex*

(b) A Filled Hole

Figure 4.18: Filling a Hole

4.4.6 Calculating New Hole uv's

One of the biggest challenges with filling the holes was selecting the correct uv coordinate to the newly created faces, when the faces consisted of vertices which had duplicates. Duplicate vertices have the same location (x, y, z values) but different uv texture coordinates. Due to the use of a three-view texture map, texture seams exist (where the three images meet on the mesh). Figure 4.2 shows these seams. To avoid these seams being visually noticeable, 3dMD (the company that makes the scanner), chose to use duplicate vertices. For instance, a single vertex could be connected to two faces. One face may have texture located in one of the images of the texture map, while the other face has texture which comes from a separate image in the texture map. While this allows for a smooth texture seam, it causes problems when

attempting to determine the best texture coordinate for a newly created face. The method used for determining the best texture coordinates for the vertices of newly created faces is discussed in the following paragraphs.

For each of the three vertices of a face, a search for duplicates (up to 2 other values) is performed. In the worst case situation, this means 27 potential *uv* coordinate combinations (3^3). The hole filling process creates new faces, so there is no pre-set "correct" answer for which vertex texture coordinates should be used. In fact, using the kept vertex texture coordinate values often results in *uv* coordinates existing in separate texture map images, and so, the texture would span across two or more images (Figure 4.3). Given that this is one of the main issues the cleaning pipeline is attempting to solve, a solution is needed for finding the best texture coordinates for the new faces.

First, before any cleaning is done to the mesh the average perimeter of the *uv* triangles is calculated using a Euclidean distance measure. That value is multiplied by a scaling value (we chose 10 as the value after testing with various scaling values) resulting in a *perimeter threshold* which is used for avoiding *uv* triangles that cross image boundaries. We observed that the *uv* triangles with a larger perimeter than the threshold were most likely to be those which crossed image boundaries. When a new face is created, if there are no *duplicate vertices* then the original texture coordinates for each vertex of a face are chosen and the face structure is updated. If there are duplicate vertices, then the *uv* texture coordinate combination that results in the smallest *uv* triangle perimeter is selected. If the resulting *uv* triangle is less than the threshold it is chosen as the "best fit" with no further modification. In the majority of instances this choice was best, both for the texture chosen and the efficiency and speed of the pipeline. The complexities of the dense meshes used in this work means that it is possible that the smallest *uv* triangle is not necessarily the "optimal" solution, and that a slightly larger triangle might be a better choice, technically speaking (e.g. slightly more accurate texture values). In these rare cases, we found the differences to be so insignificant that there were no noticeable differences between the choices.

When a *uv* triangle is above the perimeter threshold it is typically because it is a thin, long triangle where one point of the triangle lies in a separate image from the two other points of the triangle. For instance, two of the points may be on the edge of the nose in one image and the other point may be near the edge of the nose in one of the other images in the texture map (e.g. Figure 4.3). This typically happens when one of the *uv* coordinates is located in a separate image than the other two and there are no other *uv* coordinate options for that vertex. In cases where the *uv* triangle is larger than the threshold, the two closest vertex *uv* coordinates of the triangle are selected. The third *uv* coordinate is scaled along a vector so that it is located in a position that results in the *uv* triangle being within the threshold perimeter. An example of the process is shown in Figure 4.19 (Note: not a real instance). The vast majority of the time this results in the *uv* triangle residing in only one image of the three-view texture map, thus resolving the cross-image issue. In the case where the best texture coordinate came from a duplicate vertex (not the kept vertex), the face structure was updated accordingly to contain the duplicate vertex as the member of the new face.

Figure 4.19: Scaling of Larger-than-Threshold *uv* Triangle

The approach implemented was chosen for its simplicity and robustness. In the chosen implementation, the best option was selected a large majority of the time.

There were certain instances were no good choice was available, but in these instance the *uv* coordinates chosen tended to not make a significant difference to the texture area mathematically, visually, or otherwise. Often these were in messy areas of the mesh (e.g. the neck area) and not in important areas (e.g. the subject's face).

4.5 Unified Texture Map Creation

Once the cleaning process is completed, the Unified Texture Map can be created. This is the process of taking the newly cleaned mesh (Figure 4.20(a)) and flattening it (Figure 4.20(b)) (2D planar parameterisation)so that a UTM image can be rendered (Figure 4.20(c)).

(a) Cleaned Mesh with Texture

(b) Flattened, Cleaned Mesh (UTM with *uv* triangles and texture)

(c) UTM (Texture Only)

Figure 4.20: Flattening a Cleaned Mesh

There have been quite a few approaches to mesh flattening (mesh parameterisation) over the years. One of the first approaches was the work of Eck et al. [101] which used harmonic mapping approaches to achieve parameterisation. Graph-drawing theory has also been a popular approach and one such implementation can be found in [109]. Sheffer et al. [209] provides a survey and comparison of parameterisation techniques. These methods, while accurate, lack the ability to perform quickly on meshes with a large number of polygons.

Other popular recent approaches to nonlinear dimensionality reduction for mesh

parameterisation includes IsoMap [217] and Locally Linear Embedding (LLE) [196]. IsoMap is an approach that attempts to parametrise a mesh while minimise distortions. To reduce distortions this algorithm attempts to preserve the (Euclidean) distances between mesh points. This approach requires the computation of pairwise distances and, thus, one of the major issues of this approach is the computation time, specifically for meshes with a high number of points. Zigelman et al. [256] modifies this approach by using the Fast Marching algorithm for computing the geodesic distances between points and restrict computation to a small set of points, which helps to reduce the computation time issue of the IsoMap approach.

The work of Roweis and Saul [196] introduce Locally Linear Embedding (LLE) as an approach to find a low-dimensional embedding of a set of points. This approach provides a speed increase compared to the methods in [217, 256] mainly due to its ability to use sparse matrix algorithms. The work of Peyré and Cohen [185] extends this approach by performing uniform sampling of the points, computing the geodesic distances instead of Euclidean distances, and modifying the appropriate equations for use with the newly used geodesic distances. The result of these modifications to the classical parameterisation methods is faster computation time, preservation of the small scale variations (bumps or noise), and provides a "desirable trade-off between the conservation of angle and area".

Like many other approaches [109, 146, 154, 170, 196, 217, 256], the approach of Peyré and Cohen [185] is not invulnerable to issues (e.g. overlapping triangles) and failures (see Figure 4.25 for an example of a failure that is due to an incorrect calculation of the mesh boundary). This is acceptable as it is easy for us to identify these relatively rare parameterisation failures and to fix them (we replace the frame with the previous frame).

The benefit of the approach of [185] is the easily-available and widely-used third-party Matlab toolbox, *Toolbox Graph* [184], written by Gabriel Peyré. This freely-available toolbox provides an implementation of the parameterisation approach described in [185] and is well-documented and well-supported in the community. We required a

robust, fast, well-tested approach for our mesh parameterisation and this method and toolbox provided a sufficient solution. While more recent parameterisation techniques exist (e.g. [146, 154, 170], these approaches do not provide readily available solutions and, thus, have not been sufficiently tested by use in the research community, unlike [184].

This fast and robust method for flattening our 3D mesh data is one of the few third-party tools used in the cleaning pipeline. It was implemented for its ability to efficiently and accurately flatten the cleaned meshes. This approach allows us to choose the boundary shape (circle, square, or triangle) and the type of Laplacian used (combinatorial or conformal). Our chosen implemented parameterisation approach uses a square boundary and a *Combinatorial* Laplacian approach for spectral embedding [184, 185]. We chose the *Combinatorial* option which, through evaluation, provided a balance between running time and output quality. It calculates the mesh weight which is used in the equation by using a triangle-adjacency matrix. A basic description of this process is that it works by "pinning" the boundary of the mesh to a plane. One of the reasons the cleaning process was necessary is because non-manifolds will cause this flattening step to fail.

The contents of the cleaned mesh are written to an OBJ file, then duplicate vertices and faces are added back into the mesh structure so that the flattened (2D parameterised) mesh can be filled with the texture from the original three-view texture map. This creates the unified texture map seen in Figure 4.20(b). Adding back in the duplicates at this point will not create any mesh issues. These items are used so that the correct *uv* texture coordinates can be referenced. The square-boundary output from the 2D parameterisation process [184] creates a 2D mesh of the same dimensions each time. This makes it simple to calculate where the camera should be placed for rendering. This flattened mesh is then rendered using an orthographic projection. This rendering functionality comes from the freely-available *Psychophysics Toolbox* [55], which provides access to OpenGL calls in Matlab. The output is a single-view, Unified Texture Map, as seen in Figure 4.20(c). This concludes the cleaning process of the pipeline and results in a cleaned-of-imperfections mesh (OBJ) and a Unified

Texture Map (PNG). This Unified Texture Map (UTM) is pivotal for the successful implementation of the tracking and registration approaches discussed in Chapter 5.

4.6 Pipeline Step-by-Step Guide

The following list provides a step-by-step guide of the implementation of the cleaning pipeline. A few items should be noted. First, if a vertex is deleted it is done for all versions of that vertex (i.e. both the *Kept* and *Duplicate* Structures). Any faces or texture coordinates which reference that vertex are also deleted. This is known as the *updating* process. After this deletion a *shifting* process must be performed to update the values in the faces structure, as they reference the index values of the vertices. The deletion of a vertex means every vertex index greater than the deleted item is decreased by one.

This entire process is referred to below as: *Perform Updating/Shifting Process*. Note that this shifting must also occur for any other structure that references the index values of the vertices, such as the *neighbour list*.

This step-by-step guide is not meant to be an exhaustive explanation, but rather a concise description of the steps of this pipeline. Where appropriate, information on the functions implemented is given. While it is out of the scope of this thesis to provide the formulae and algorithms for all of these functions (although the goal is to make this code available to the research community in due time), this guide allows the user to understand the simple steps in the order they were taken and examine the code of the less straightforward steps (as all of the functions/toolboxes described are publicly available).

Pipeline Steps

i) Set uv triangle perimeter threshold, t

   a) Calculate average uv triangle perimeter, $\overline{p}$

b) Choose scaling value, sv

c) $t = \overline{p} \times sv$

ii) Create Kept and Duplicate Structures (Section 4.2.1)

a) Identify duplicate mesh vertices (Matlab's *unique* function with 'rows' flag set)

b) Move all but the first instance of each into the *duplicate vertices* structure (See 4.2.2 for details and examples)

1) *Duplicate vertices* structure contains the index for each vertex's corresponding kept vertex

2) Move corresponding uv texture coordinates in the *duplicate texture coordinates* structure

3) Find faces referencing duplicate vertices and move them to the *duplicate faces* structure

c) Shift values in faces structures (kept and duplicate) to reflect the new indices of the vertices

iii) Remove Non-Manifold Edges (NME)

a) Compute the triangle adjacency matrix, $\mathbf{A}$

1) We used Gabriel Peyré's *triangulation2adjacency* function from *Toolbox Graph* [184]. It describes which pair of vertices connect which faces.

b) Remove Small Components

1) Using $\mathbf{A}$, find largest connected object, remove smaller objects

c) Remove Isolated Vertices

1) Find and remove vertices that are not members of any faces

d) While Non-Manifold Edges Exist:

1) Identify a NME

(i) In **A**, this is a pair of vertices (an edge) with more than two connected faces

2) Repair a NME

(i) Remove faces connected to the NME

(ii) Perform Updating/Shifting Process

(iii) Delete the vertices of the NME and faces connected to those vertices

(iv) Perform Updating/Shifting Process

(v) Remove Small Components and Isolated Vertices

(vi) Identify and Fill Holes (Section 4.4.5)

(a) Identify hole issues and fix as necessary

(b) Fill holes

1) Create New Faces

2) Calculate new face *uv* texture coordinates (Note: If any texture coordinates are from duplicate vertices, update duplicate structures as appropriate)

(vii) Remove Small Components and Isolated Vertices

iv) Remove Non-Manifold Vertices (NMV)

a) Remove Small Components and Isolated Vertices

b) While Non-Manifold Vertices Exist:

1) Identify a NMV

(i) We implemented the *meshcheckrepair* function from [106] which finds NMVs using the *JMeshLib* library [17].

2) Repair a NMV

(i) Delete the non-manifold vertex and the faces connected to that vertex

(ii) Perform Updating/Shifting Process

(iii) Remove Small Components and Isolated Vertices

(iv) Identify and Fill Holes (Section 4.4.5)

(a) Identify hole issues and fix as necessary

(b) Fill holes

1) Create New Faces

2) Calculate new face *uv* texture coordinates

3) Note: If any texture coordinates are from duplicate vertices, update duplicate structures as appropriate

(v) Remove Small Components and Isolated Vertices

v) Calculate edges for *kept faces* structure

a) We implemented Gabriel Peyré's *compute_edges* function from *Toolbox Graph* [184].

vi) Create Unified Texture Map (UTM)

a) Flatten cleaned mesh

1) We used Gabriel Peyré's *compute_parameterization* function from *Toolbox Graph* [184, 185]. Details can be found in Section 4.5

b) Write kept vertices and faces to an OBJ file, using the flattened mesh 2D vertex values as texture map *uv* texture coordinates

c) Add back duplicate vertices, faces, and texture coordinates to flattened mesh

d) Render the flattened mesh to create the UTM

1) We used the *Psychophysics Toolbox* [55] to render the images of the UTM

4.7 Pipeline Evaluation

This section speaks to the speed and robustness of the fully-automatic pre-processing pipeline.

Using an Intel Xeon E5-2407 with 2 processors (4 cores each), 2.20 GHz per processor with 32 GB RAM machine; the cleaning process takes approximately 40-50 seconds per mesh. However, parallelisation has been implemented, which uses 8-cores, resulting in it effectively taking approximately 6.5 seconds per mesh. There are roughly 5-10 problematic (i.e. failed to create a cleaned OBJ and texture map) meshes per 500 meshes. These numbers vary based on the characteristics of the meshes being cleaned (i.e. how many issues there are, how noisy the capture was, etc.), but provides an idea of the speed and robustness of this approach. Failure can occur for a variety of reasons, but the main reasons for failure during processing of our data included a poor initial captured mesh (which results in a problematic mesh OBJ, such as the mesh having too many or too few faces) or a failure to properly flatten the mesh (which results in an incorrect UTM). At 60 frames per second, if a mesh fails it is replaced with the previous mesh frame (or closest successfully cleaned mesh in either direction) using an automatic *check-and-replace* function. This approach was selected because there is a negligible difference between adjacent frames due to the 60fps capture method, and during development we observed that failures tend to not occur sequentially. This meant the previous/next mesh was a near perfect match of what the failed mesh should be and could be used to replace it without any noticeable differences.

A concern during development was that the solutions being implemented would only work well on the data used during development (i.e. 3dMD, 4D system captures). For this reason, the pipeline was tested using data from other 3D scanning systems, both from 3DMD, as well as from the other leading provider for 3D/4D face scanners, Dimensional Imaging (Di4D).

In the following sections, two example scans are tested for each capture system. The first scan comes from a *two-view* system, where two texture cameras are used and, thus, the resulting multi-view texture map contains two images. The other is from a *three-view* system, which contain three images in the texture map. A screenshot of the 3D mesh before cleaning, the original multi-view texture map, the new single-view texture map (UTM), and cleaned mesh is provided. This is to show

that the pre-processing pipeline was successful in cleaning the 3D mesh and creating the single-view texture map, without negatively altering the original mesh in any way. The cleaned 3D mesh also shows that the cleaned mesh and single-view texture map correspond properly so as to produce a legitimate 3D mesh.

The pre-processing pipeline worked on the first attempt, without any issue, for these tests. No modifications needed to be (or were) made to the pre-processing pipeline.

4.7.1 3dMD Scanners

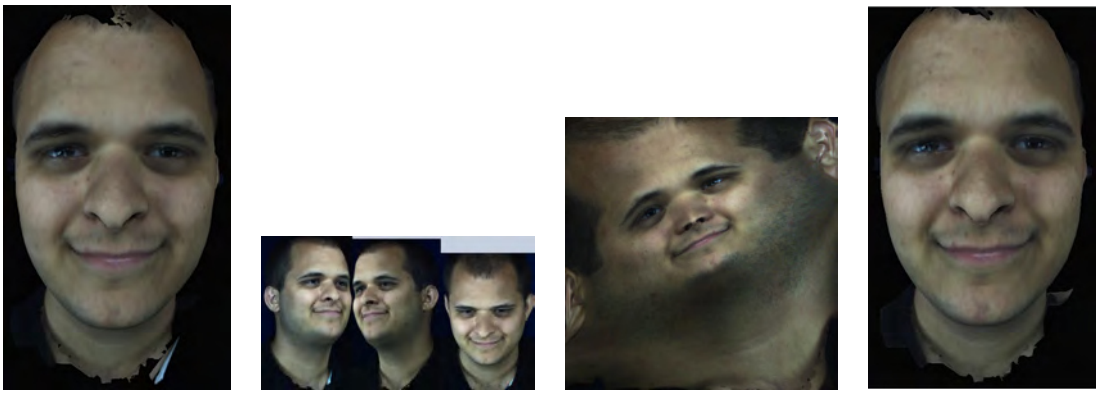
3dMD is one of the leading providers of 3D and 4D scanners in the world [4]. With over 1,500 systems in use worldwide, primarily in the medical field, 3dMD provides 3D and 4D scanners for both face and body captures. Hundreds of peer-reviewed academic papers, whose research has used 3dMD scanners, have been published. One of 3dMD's most well-known academic clients is the Max Planck Institute for Intelligent Systems, Perceiving Systems, in Tübingen, Germany, whose work uses a 22-viewpoint 4D body scanner.

Figure 4.21 shows an example of a two-view texture map and the resulting UTM from a 3D mesh provided by research colleagues at the University of North Carolina - Wilmington (UNCW) *Institute for Interdisciplinary Identity Sciences (I3S)*. This data was captured using a 3dMD, two-view, 3D static scanner. This mesh contained approximately 60,000 vertices and 115,000 faces.

(a) Original Mesh (b) Multi-View Texture Map (c) Unified Texture Map (d) Cleaned Mesh

Figure 4.21: UNCW 3dMD Example

Figure 4.22 shows an example of the three-view texture map and the resulting UTM. This data was captured using a 3dMD, three-view, 3D dynamic scanner (4D). This mesh contained approximately 30,000 vertices and 60,000 faces.



(a) Original Mesh (b) Multi-View Texture Map (c) Unified Texture Map (d) Cleaned Mesh

Figure 4.22: Cardiff University 3dMD Example

4.7.2 Di4D Scanners

Di4D (Dimensional Imaging) is one of the leading providers of 3D and 4D face scanners in the world [5]. Di4D scanners are used in the areas of facial performance capture, medical research (e.g. orofacial), and Psychology research. Hundreds of peer-reviewed academic papers, whose research has used Di4D scanners, have been published. Well-known industry clients include Sony, Electronic Arts. Well-known academic clients include Imperial College London and the University of Glasgow.

No data from Di4D scanners was used during the development of the pre-processing pipeline. The frames tested here consist of both triangle and quad faces and were downloaded from [6]. Blender [1] was used to convert the mesh so that all faces were triangles, as that is what the pipeline expects. Once this simple step is performed, the pipeline processes the Di4D meshes in the same manner as the 3dMD meshes. The output shown below was achieved during the first attempt. No internal modifications to the pre-processing pipeline were necessary.

Figure 4.23 shows a frame that was captured using a Di4D, two-view, 3D static scanner. This mesh contained approximately 108,000 vertices and 212,000 faces.

(a) Original Mesh (b) Multi-View Texture Map (c) Unified Texture Map (d) Cleaned Mesh

Figure 4.23: Di4D Two-Pod Example

Figure 4.24 shows a frame that was captured on a Di4D, three-view, 3D static scanner. This mesh contained approximately 130,000 vertices and 260,000 faces.

(a) Original Mesh (b) Multi-View Texture Map (c) Unified Texture Map (d) Cleaned Mesh

Figure 4.24: Di4D Three-Pod Example

4.7.3 Discussion

The fully-automatic pre-processing pipeline was able to process over 135,000 3D meshes from our 3dMD, 3D dynamic (4D) capture system. The pipeline very rarely failed to properly process a mesh. Failure to properly flatten the mesh typically appeared to occur because the parameterisation step would fail when the area of the ear in a mesh was particularly complex. The technique would attempt to use the curve of the ear as the "pinning" points for the boundary of the texture map

(as described in Section 4.5). This resulted in incorrect texture maps whose texture emanated from the ear on a mesh, as seen in Figure 4.25.

Figure 4.25: Pipeline Error - Result of Flattening Mesh Failure

This failure is due to the use of a third-part function in our pipeline. In the future, we would like to implement our own flattening approach, so that it can correct these types of issues during the UTM creation process.

Our pipeline was able to successfully process 3D mesh data from two types of scanners from two separate vendors: 3dMD and Di4D. These two companies are the leading providers of face scanners in the world, in both academia and industry. No modifications to the pipeline were necessary. The resulting *cleaned* meshes appeared identical (aside from the removed issues) to their original versions.

The pre-processing pipeline was a means to an end: providing data in the format needed to perform tracking, registration, and model building. It is positive to see its successful results using other 3D data from different systems and scanning companies. Further development of the pipeline will occur to make it even more robust. We would like to be able to provide the code to other researchers to help them move past

the barrier of manipulating meshes that contain duplicate vertices and multi-view texture maps.

4.8 Summary

In this section, a fully-automatic, robust pre-processing pipeline was described that allows for the cleaning of 3D meshes and, more importantly, the creation of a single-view texture map (UTM) from systems which provide multi-view texture maps and meshes with duplicate vertices. The UTM allows for the modification of vertices without texture issues being created. The UTM is created by flattening the mesh, which can only be done once imperfections are removed. The imperfections are removed using a specific structure of cleaning steps. These include removing non-manifold edges and vertices, along with small components and isolated vertices. The holes created from the removal of these imperfections are identified and filled. Identifying and filling holes is not always straightforward, which is why different techniques for identifying and fixing hole issues were developed. These approaches were developed to be generic enough to identify and fix issues, while also being specific enough to properly repair the issues and not create new imperfections.

The techniques for repairing issues with the mesh are not novel; they can be found in popular open-source software tools, such as Meshlab [68] or Blender [1]. The novelty of this contribution comes in the assembly of these techniques, in such a way as to provide a fully-automatic, fast, generalised, and robust approach, which includes the challenge of developing a storage structure capable of easily and accurately keeping track of the *Kept* and *Duplicate* vertices/faces/texture coordinates, throughout every step of the cleaning process.

To the best of our knowledge, no publicly-available, system-agnostic, fully-automatic tools are available for the creation of single-view texture maps (UTMs) from systems which provide multi-view texture maps and 3D surface meshes with duplicate vertices. Unlike previous works [59, 80, 120, 224, 225], the approach described in this work

does not require system-specific information. It is able to produce a single-view texture map given only the 3D surface mesh and multi-view texture map.

The pre-processing pipeline described in this chapter was able to, on the first attempt, process 3D mesh frames from multiple 3D/4D systems, from different vendors. Creating a generic-but-robust approach that was able to identify and repair the various complex geometric issues of the 135,000+ meshes used in this research, as well as those meshes from other 3D scanners, is a major achievement. The code for this pre-processing pipeline will be made publicly available to the research community. It is our hope that the code and these steps will be used by other researchers facing this issue, thus, removing the barrier many researchers with these types of systems have had when attempting to modify the initial captured meshes.

Once the pre-processing step is performed on the data, it can be manipulated for a variety of research purposes. In our work, we use it with our 4D tracking and inter-subject registration techniques (Chapter 5) for creating corresponding meshes and texture maps. These corresponding meshes are used for building statistical models (Chapter 6). The 4D tracking and registration approaches are described in the following chapter.

Chapter 5

Tracking and Registration of Sequence Feature Points

5.1 Introduction

As explained in Section 2.6, tracking feature points through a sequence is important for a variety of reasons. In this work, we use the tracked feature points for our registration method. The output of our registration method provides us with data we can use to build our statistical models, which are used to analyse data and synthesise new data. However, tracking high-resolution 3D frames through a sequence and making the sequence data correspondent (registered) is still a challenging problem in computer vision research. In [23], a Constrained Local Model is used for tracking 3D sequences, but it has the limitation that it requires depth information, such as from a Microsoft Kinect camera (2.5D). It can use data from 3D dynamic (4D) systems but requires manual steps to calculate depth information using the texture images of the 3D models (as described in Section 3.3 [23]). Therefore, it is not an acceptable approach for researchers with 4D data who require an automated system. The approach is best for researchers using 2.5D data, such as data captured using a Kinect sensor. Sidorov et al. [210, 211] use a non-rigid groupwise registration approach, but that method is limited to diffeomorphic deformations, such as closed

mouths, which is not an ideal solution for our data of people in conversation. In [123], Hammond et al. used Thin Plate Splines (TPS) to achieve dense registration for static captures (non-sequence data). Eleven manually placed landmarks were used for surface shape analysis: the corners of each eye (4), each lip corner (2), the nose bridge, tip, and base (3); the middle part of the top lip, and the chin. Models used for visualisations could have "up to 25" landmarks. Benedikt et al. [40, 39] have used a TPS approach for densely inter-subject registering meshes for use in statistical models. In our work, we use TPS for dense registration of multiple 3D, high-resolution, high frame-rate sequences across a group of subjects, allowing for inter-subject registered shape and texture frames, which is necessary for building statistical models of appearance of conversational expression interactions.

For the goals of this work to be achieved, a semi/fully-automatic method of tracking and densely registering sequences is required. One of the main goals of this research is to produce highly-realistic, synthesised facial expression sequences, which means the tracking and registration methods used should not unnecessarily warp the data (i.e. registered frames that no longer look like the original frames).

The sections below describe the first tracking approach that was considered. Although there were many issues to solve, eventually this approach proved useful for creating 3D frames with corresponding (registered) vertices and faces, for a single sequence. The goal was to use the multiple sequences of multiple subjects in a statistical model of appearance. Unfortunately, when the process of *inter-subject registration* was performed, it was clear that the sequence-registered meshes could not be used. The topologies and number of vertices were so different that a many-to-one correspondence across sequences occurred. This resulted in problematic (holes, misshapen, non-manifolds, etc.) meshes that were not useful for our statistical modelling approach. Therefore, a new tracking approach was developed, with a slightly modified registration approach. The general approach is outlined below and in Figure 5.1. The following sections explain the need for these approaches and the details of our implementation of these methods.

Tracking and Registration Methods Overview

1. Tracking Approach
    - Annotate a single mesh from the sequence (Sections 5.5.1 and 5.5.3)
    - Track the points through the sequence (Section 5.5.4)
        - Method tracks forwards and backwards from the annotated mesh
2. Create Registration Mask (Section 5.6.1)
    - Must have the same tracking point scheme as the tracked meshes
        - Previously created registration masks can be used if they have the same tracking point scheme
3. Register the Tracked Meshes (Section 5.6)
    - Method uses tracked meshes and registration mask
    - Method registers the meshes and their texture maps

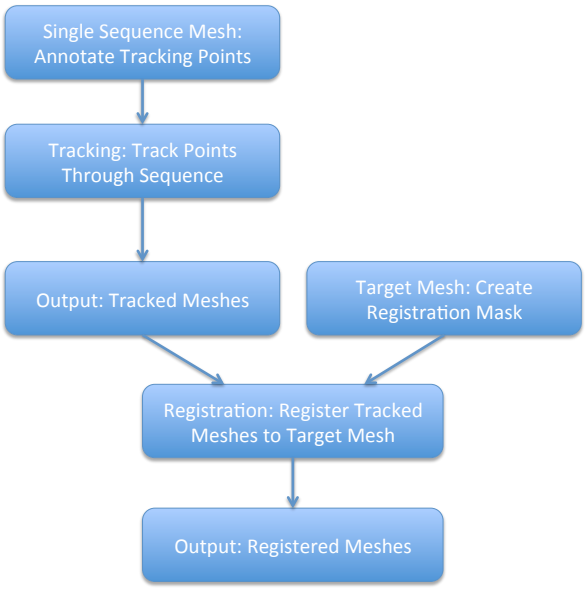


Figure 5.1: General overview of the tracking and registration process. Only one mesh per sequence needs to be annotated with tracking points and any mesh, from any subject, can be used as the *target mesh* if it has been annotated with the same tracking point scheme.

In Section 5.2, the first approach to tracking we explored, which used a third-party tool, is described. In Section 5.5 the second, and final, approach to tracking is

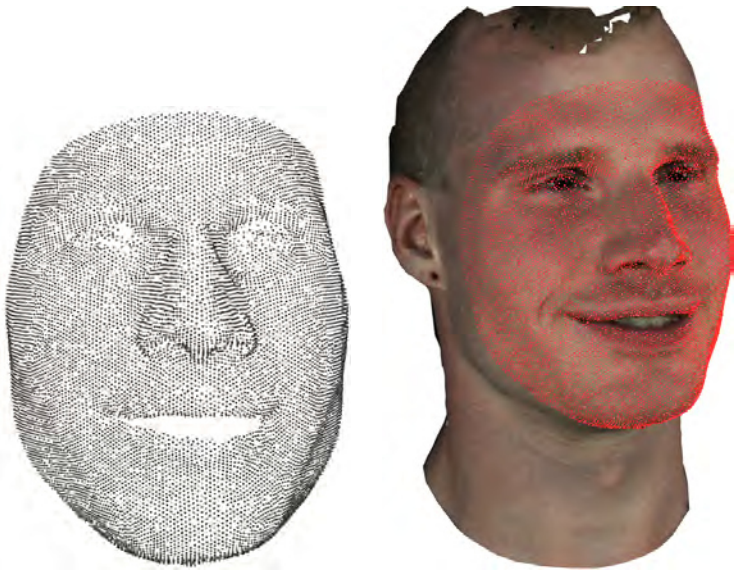
described, along with information about the tracking landmark scheme and software used for tracking. Section 5.6 details the approach used in this work for creating dense correspondence between meshes (i.e. inter-subject registration). The tracking and registration methods used were developed mainly by Lukas Gräser, a 2014 IAESTE (The International Association for the Exchange of Students for Technical Experience) student who spent the summer conducting research at Cardiff University, under the guidance of Professor David Marshall, Professor Paul Rosin, and myself. After the summer he continued to collaborate on the development of these methods and software.

5.2 Tracking: Optical Flow-based Method

Optical flow [37, 118] is an approach that calculates the apparent movement of pixels (or regions) from one image to the next, across a sequence, where the amount of time between frames is small. The movement of a pixel is described using a velocity vector. There are quite a few methods used for optical flow [58, 114, 127, 132, 157]. Well-known facial expression capture research has used optical flow for creating a correspondence between frames, for the purposes of tracking feature points, dense registration, or point cloud reconstruction [54, 79, 93, 243, 251]. The wide-use of optical flow methods and the availability of a third-party tool capable of using the raw data from our capture system to perform optical flow made this method an attractive choice for calculating feature correspondence.

Using a proprietary third-party tool provided by the capture system company, dense feature points are "sprayed" onto the first frame of a sequence. Their placements are arbitrary and determined by the software. These points are tracked through a sequence using a hybrid 2D/3D optical-flow based algorithm. The output consists of X, Y, Z points for each frame, which correspond across the sequence. This data is converted into OBJ format, to be used as a mask (of sorts) with the "cleaned" 3D frames (Section 4.1).

Each cleaned 3D frame provides useful vertices, faces, and texture coordinates and a Unified Texture Map (UTM), while the tracked points from this optical flow approach provides vertices which correspond through the sequence (Figure 5.2). Finding where each tracked point lies on the cleaned mesh allows us to calculate the tracked point's uv-texture coordinate in the UTM.



(a) 3D Point Cloud of Tracking Mesh (b) Original Scan with Tracking Mesh Overlaid

Figure 5.2: Tracking Mesh

Figure 5.3 shows an example of 3 frames that have been tracked.

5.2.1 Tracked Point uv-Coordinates

To calculate the uv texture map coordinates of tracked vertices (Figure 5.2(a)) we use a 3D barycentric calculation technique. The *Barycentric Coordinate System* [168] is a coordinate system for use with simplexes (Figure 5.4). It allows any point within a simplex to be described in a consistent, quantifiable manner.

The location of a point p in triangle T on the 3D surface mesh can be described by a convex combination c of the triangle vertices, v_1, v_2, and v_3, as shown in Equation 5.1.

$$c = \alpha v_1 + \beta v_2 + \gamma v_2 \tag{5.1}$$

Figure 5.3: Three Tracked Frames

where α, β, and γ are the barycentric coordinates of p with respect to T.

Each vertex in the triangle T has a corresponding uv texture map coordinate. To calculate the uv texture map coordinate for p, referred to here as p_{uv}, these barycentric coordinates are used as a scaling factor for the the uv values of the vertex texture map values, $v_{1_{uv}}$, $v_{2_{uv}}$, and $v_{3_{uv}}$, as shown in Equation 5.2.

$$p_{uv} = \alpha v_{1_{uv}} + \beta v_{2_{uv}} + \gamma v_{3_{uv}} \tag{5.2}$$

When this process is completed the tracked vertices will have corresponding texture map uv coordinates, and will only require a triangulation step to create the mesh faces. As it turns out, the uv-coordinates for the tracked vertices are also very important for creating corresponding faces (triangles) for the tracked points, which is explained in Section 5.2.3.

The approach for calculating the uv-texture coordinates for each tracked point was much more challenging than initially anticipated. This was due to two main reasons: (a) issues with the exact tracking point location in relation to the cleaned mesh and (b) not every tracked point was located *on* the cleaned mesh. This was due

to inadequacies in the closed-source, proprietary tracking software. Aside from very bad drift problems in certain parts of the face (e.g. cheeks, eyebrows), one major short-coming is that it often produced what we have termed "floating tracking points". Details of these issues and the developed solutions are as follows.

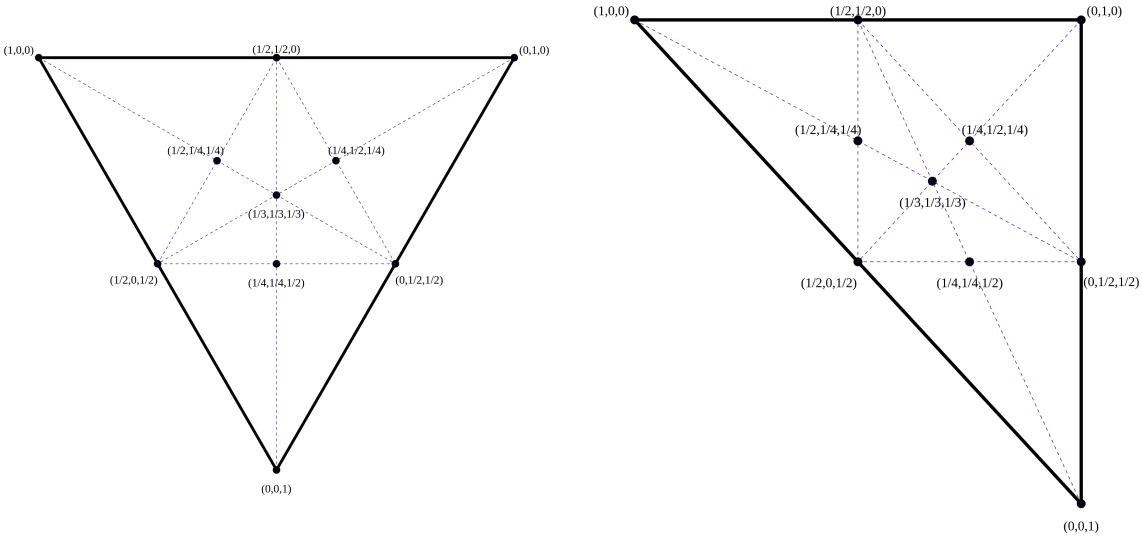


Figure 5.4: Barycentric Coordinate System [197]

The first issue to solve occurred when a tracked point was located on a face edge. Technically, it is not inside any triangle. This situation is relatively easy to identify because one face vertex's Barycentric distances will be 1.0. If so, a simple distance calculation between the two vertices of the edge is performed for determining the factor each vertices' texture coordinates should be multiplied.

In cases where the tracked point has the exact same location as a cleaned mesh vertex, the *uv* coordinates of that vertex are copied. The second issue occurs when a tracked vertex does not mathematically lie on a cleaned mesh vertex, but does so for all intents and purposes (precision issues). If the vertex points are within an arbitrary distance of $1e^{-3}$, then the cleaned mesh vertex coordinates are considered close enough to use, as it is only a matter of mathematical precision, not location.

The third issue is by far the most tricky scenario. There are instances where tracked vertices "float" off of the mesh. This can occur for a variety of reasons, but most commonly occurs with tracked points on hair. The eyebrows and eyelashes particularly can be problematic areas. The sides of the head are not too important for

the analysis being performed, but noisy eyebrow/eyelash regions can be problematic and cause unsettled geometry and texture issues. An example of this type of issue can be seen in Figure 5.5.

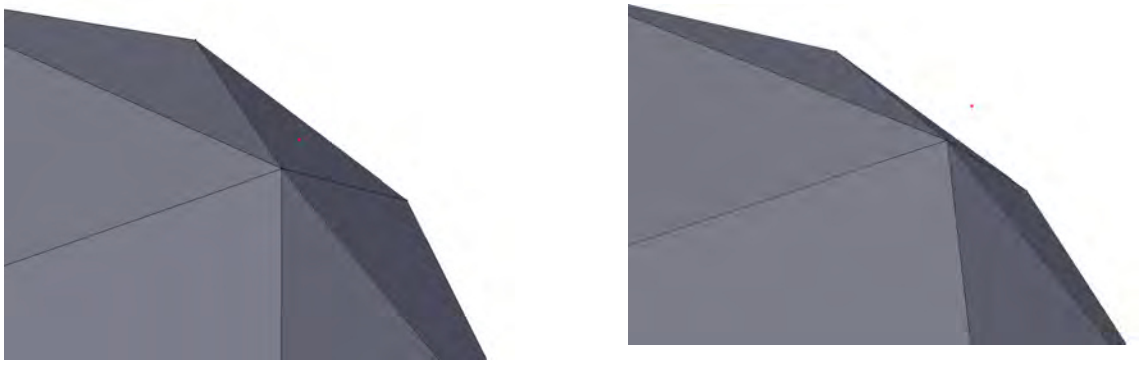


Figure 5.5: Floating Tracking Point (Red) - Two Angles

Floating tracked points were handled by projecting the points back onto the mesh in the direction of the normals. This placed the tracked points on the mesh, where they would have been if they had not drifted off the mesh. At this point the normal process of using Barycentric calculations was performed to determine the texture uv coordinates. It is important to note that the tracked point was not updated to the projected location. It was found that this would introduce geometric issues, such as non-manifolds. In most-to-all cases the original vertex location and projected vertex location were close enough that the calculated uv coordinates were indistinguishably correct.

5.2.2 Point Cloud Triangulation

To use the data for building statistical models of appearance (e.g. AAMs), which is a requirement of our research, it is not enough for the vertices of the meshes in a sequence to correspond. The faces (triangles) must also correspond. With the uv-coordinates calculated for the tracked points (vertices), the only step remaining is to produce a consistent triangulation, to create the correspondence of faces across a sequence. 3D Triangulation methods, such as Ball Point Pivoting [68], were attempted on the vertices, but resulted in poor triangulations, as seen in Figure 5.6.

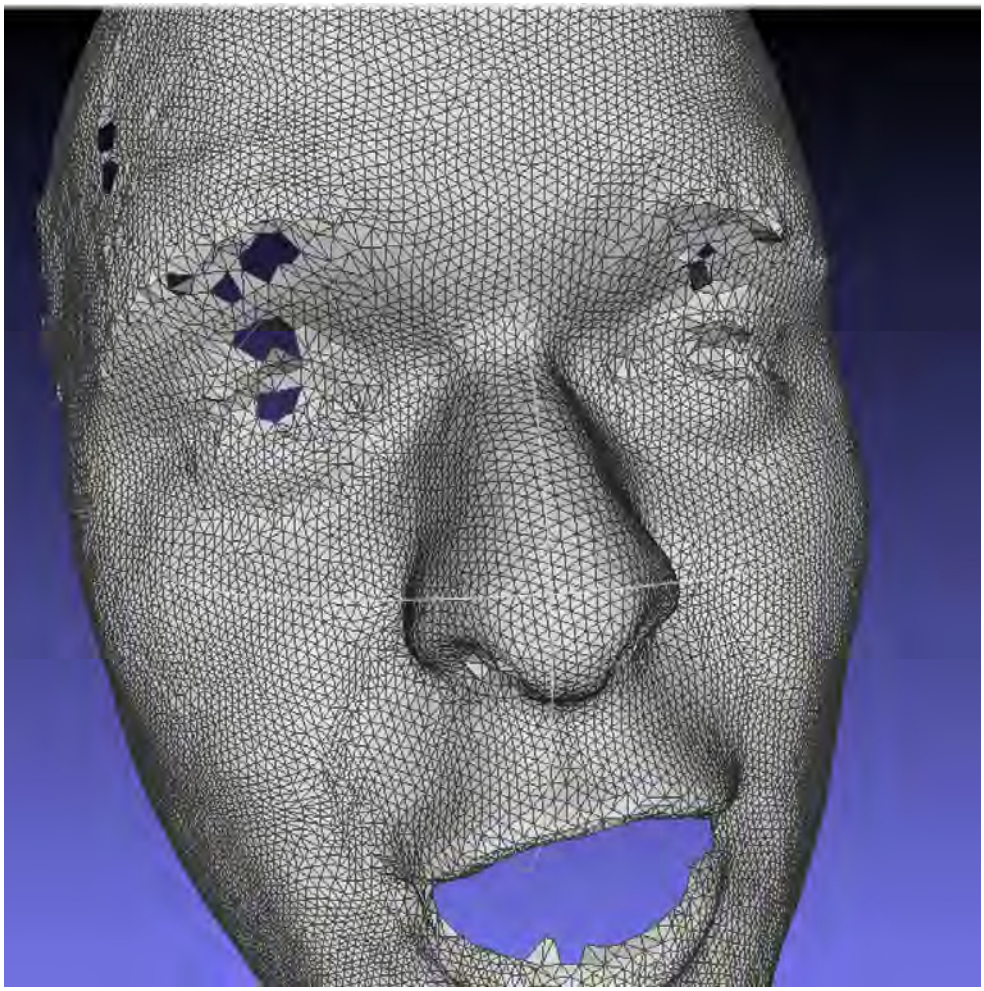


Figure 5.6: Ball Point Pivoting Triangulation

A better approach was necessary. While 3D Delaunay triangulation [94] is one of the main methods for triangulating meshes, 3D Delaunay-based approaches do not meet our requirements for triangulation quality and computation (run-time) required. The method of Dey and Tathagata [96, 97], as implemented in the software solution *SurfRemesh* [97], is a commonly used option. However, as described in their paper [97], one of the main shortcomings is the speed of the approach. Other well-known 3D Delaunay/Voronoi based surface triangulation algorithms, such as the *PowerCrust* algorithm from Amenta et al. [12, 13], produces issues if the point cloud is not dense enough, if there is noise (such as with the neck, ear, and eye areas of our 3D scans), and can often reconstruct the "wrong" surface. Additionally, these approaches are relatively time-consuming (many seconds per mesh), compared to 2D approaches. Tang et al. [216] describe and compare a number of 3D surface triangulation approaches and point out their shortcomings. We required a robust

and fast solution for triangulating our 3D meshes and it proved to be a challenge to find a robust solution for triangulating 3D point clouds of 20,000+ points. Therefore, a simple, more intelligent solution to our problem was devised.

The tracked points have *uv*-texture coordinates in the unified texture map. There is a 1:1 correspondence between the vertices and texture coordinates, meaning that a triangulation of the 2D *uv*-coordinates could give vertex-face relationships for the 3D tracked points. A 2D Delaunay triangulation using concave hull [160] was performed on the texture coordinates (Figure 5.7) and resulted in accurately triangulated meshes (Figure 5.7). In addition to producing nicely triangulated meshes, this approach was very fast (less than half a second per mesh).

Figure 5.7: Triangulated *uv* Coordinates on Unified Texture Map

Along with being a robust solution, it also solved many issues that arose from the previously used triangulation implementation, such as triangulation holes, failed meshes, running time issues, and multiple file *i/o* calls. To give some perspective in the increased efficiency: the original implementation, with Ball Point Pivoting using multiple iterations for best outcome, took 5 second per frame (5 hours per 1-minute sequence). The new implementation, which used optimised and efficient code to solve a 2D problem, took 17 seconds total (per 1-minute sequence).

Figure 5.8 shows the 3D point cloud triangulated, triangulated with texture, and only texture.

Figure 5.8: Tracked Mesh

5.2.3 Optical Flow Approach Discussion

Unfortunately, while this approach produced decently triangulated 3D meshes, with corresponding vertices and *uv* texture map coordinates, it was insufficient for use in creating accurately registered meshes. For one, the amount of drift experienced when using this approach was substantial for certain individuals. As well, this substantial drift would occur for any individual after about 10-20 seconds. Given that the data to be tracked are minute long captures, this was an issue we could not ignore. When the vertices would drift they would end up bunching together in certain regions of the face (e.g. cheeks, eyebrows). This drift/bunching contributed to poor inter-subject registration (described in Section 5.6) due to a many-to-one mapping. That is, when searching for similar mesh vertices between two subjects, bunched vertices from the target mesh would map to a single vertex of the reference mesh. This would often result in chunks of the face missing in the 'registered' meshes. Clearly this approach was not sufficient for creating nicely tracked and accurately registered meshes.

5.3 Optical Flow Issues

The problems experienced using the third-party tool are not unexpected, as it is well-known that optical flow is prone to drifting issues due to the accumulation of

errors [49, 54, 92, 181]. There have been some modifications to popular optical flow methods that attempt to reduce or avoid this issue of the drifting error. Decarlo and Metaxas [92] propose a model-based, least-squares method that reduces drift by combining optical flow and edge information. Other approaches seek to reduce drifting error by using a manual landmarking approach, such as the methods used in the well-known works of Borshukov et al. [49] and Patel and Smith [181]. While these may reduce drift they are also time consuming methods that are not feasible for research contain a large number of long, high-frame rate sequences. Bradley et al. [54], does introduce a texture-based approach for automatic drift correction. It works well, unless there are significant changes in appearance and also requires a consistent (same orientation for each texture map) 2D texture map. Our cleaned meshes use single-view texture maps, however the frames still do not correspond, and the texture maps have different orientations (e.g. unknown rotations). This prevents us from being able to implement a 2D tracking approach using the UTMs since the search space for each tracking point would change drastically frame-by-frame given the inconsistent UTM orientations.

Unfortunately, the tool we used for optical flow is proprietary and closed-source. Therefore, the specific method (e.g. one of the common methods described in [58, 114, 127, 132, 157]) implemented is unknown and there is no way for us to modify the implementation to correct for such issues. Add to this the data and time intensive nature of this specific implementation (100's of GB and 20 hours of processing for a minute-long sequence) and it is clear to see why we decided to explore our own feature correspondence approach.

5.4 Alternative Approaches

At this point we wanted to explore approaches that would not require optical flow, as reducing drift is still an open-problem. Cosker et al. [80] is a particularly interesting work because they use a similar capture system and have describe a novel approach to feature correspondence that does not use optical flow. They introduce

an AAM and TPS-based approach for calculating feature correspondence across a sequence. This approach reduces the 3D feature correspondence problem to a 2D image registration problem, where the 2D texture maps contain a mapping to their corresponding 3D meshes. This approach seems to result in less drift than popular optical flow approaches, however no information is provided in regards to the computation time. Other than the quality of tracking and registration, one major issue we encountered while exploring other tracking and registration techniques was the prohibitive computation time. Without any indication of the computation time of the creation of the single-view 2D *uv* texture map, 2D-to-3D mapping, and AAM+TPS approach, it is difficult to know if this approach would warrant further investigation as a possible approach for our data and applications. As well, no information is given about how their techniques perform on sequences much longer than the average sequence length of 65 frames. This is understandable given the length their sequences, but given the sequences in our conversation data are approximately 3600 frames. Even just focusing on the annotated frames used in our experiments (Chapter 7), our conversational expression interactions range from 50-300 frames and smile data ranges from 180-550 frames.

Additionally, this approach requires (similarly to the promising approaches in [38, 54]) single-view texture maps that have a consistent texture map orientation and layout. The method used for creating the single-view texture map in [80] allows for the same texture map orientation per frame (left and right image merged into one, in an "unwrapped" *uv* texture map). The approach we use (mesh flattening) results in arbitrarily orientated single-view texture maps, which only correspond with their 3D mesh frames. Thus, the frames of a sequence have different texture map orientations and cannot easily use the 2D tracking approach described in [80]. Finally, these works focus on single sequences and appear to be unable to (without additional steps) perform inter-subject registration. Our goal is to build statistical models of multiple individuals and, thus, we require a robust, highly-accurate inter-subject registration approach for our work. Given that we have 3D information, it seemed appropriate to use this information, rather than ignore it, for developing a strong

approach to creating feature correspondence (i.e. tracking and registration) across sequences and subjects.

5.5 Tracking: Local Neighbourhood-based Method

It was clear from the results of the third-party tracking method that a more efficient, accurate, and faster approach to tracking 3D sequence data was needed. The new tracking method would need to be (at least) semi-automatic, and produce accurately tracked points which could be used by our inter-subject dense registration method to create accurately registered, high-resolution meshes. We also wanted to use the 3D information for these approaches for producing accurate tracking points and registered meshes. That is, we did not want to reduce the 3D problem down to a 2D image registration problem.

We made the choice to develop an approach that could use both 3D shape and texture information for tracking feature points across a sequence. This approach only requires a single, manually landmarked mesh (per sequence) and is able to track these points forwards and backwards through a sequence. These landmark points identify important areas of the face, such as the eye and mouth regions, and is required for each frame in a sequence, as these points are used by our inter-subject registration approach for creating corresponding data. Manually-landmarking every frame in a sequence is time-consuming (e.g. one minute of 3D video would be 3600 frames to landmark). Automatically landmarking every frame accurately is not currently feasible without some information about the feature points of the 3D frames. Therefore, an initialisation step is required, which allows for a *semi-automatic* landmarking approach. In this approach, a single frame of the sequence is landmarked with the appropriate points and used as a starting point for our tracking method. Landmark points are identified in other sequence frames using our 3D tracking method. This approach, while not as perfect as manually-landmarking each frame, is sufficiently accurate and much less time-consuming. This semi-automatic approach we have developed includes parameters that allow the user to determine

how much information they would like to use for tracking, to allow users to find a balance between tracking accuracy and computation requirements/time.

In the following sections, we described the landmarking schemes evaluated and chosen (Section 5.5.1), the landmarking tools that were developed for placing the manual annotations on the meshes (Sections 5.5.2 and 5.5.3), and we introduce our 4D tracking approach (Section 5.5.4).

5.5.1 Landmarking Scheme

The landmarking scheme in this work was determined empirically. Details of the different types of landmarking points (mathematical, anatomical, pseudo/intermediate) and descriptions of what make up "good" landmark points (e.g. areas of high curvature, common anatomical areas such as eye corners) can be found in [24]. Based on landmarking schemes used in the literature, nine different landmarking schemes were tested. These were made up of 9, 19, 23, 24, 26, 28, 32, 33, 51 points. These schemes were evaluated to determine the optimal number and location of landmarking points for tracking feature points accurately through a sequence. They were also evaluated to determine if they were good control points for the inter-subject registration step (described in Section 5.6), since it is possible they would be good tracking point choices, but not be helpful in the registration step for producing accurate and clean registered meshes. In the paragraphs below, the scheme locations are described, and for those schemes not chosen, their short-comings are explained.

The 51-point scheme (Figure 5.9(h)) had two main issues. First, the landmark points along the face contour tracked poorly, due to the lack of obvious feature points (such as lip or eye corners). The more troubling issue, related to the first issue, was that it was very challenging to consistently place landmarks for each sequence across all subjects. This issue became visually evident after performing inter-subject registration (described in detail in Section 5.6). The edges of the face for most registered meshes were warped both in shape and texture. Although mentioned first, this scheme was not hypothesised to be the best option, mainly due to computation

time, which increases substantially for each landmarking point that must be tracked. This scheme was only tested so that a wide range of landmarking schemes could be evaluated.

Most of the issues with the other unused landmarking schemes were not down to the ability of the tracker, but more the result of the dense registration step. The 9-point (Figure 5.9(a)) and 19-point (Figure 5.9(b)) landmarking schemes provided too few landmarks which resulted in stretched faces (and therefore warped textures). As well, the eye and eyebrow region contained very "messy" faces (i.e. large-than-average faces of inconsistent sizes; this can affect both shape and texture).

The 26-point (Figure 5.9(d)) landmarking scheme attempted to resolve these issues by adding two more points for each eye (top and bottom of eyelids), as well as three points evenly spread across the eyebrows. The 28-point (Figure 5.9(e)) scheme had these same additions, with one point added for the centre of each eye. The eyebrow feature points helped with the messy shape (triangulation) of this region, however being located on facial hair, the three points tracked very poorly/inaccurately. The feature points in the centre of the eye had no effect on the quality of the (inner) eye mesh; it was as messy with it as it was without it.

The 32-point (Figure 5.9(f)) landmarking scheme kept the eyebrow feature points, but removed the points at the centre of the eye. This new scheme added to two important areas that were also seen to lack enough feature points for good, consistent meshes (after registration). First, the nose, which previously only had a single point on the nose tip, was given two new points: one on the bridge of the nose, and one halfway between the bridge and the nose tip. Second, the lips, which previously had only had four/six points, now had eight. The upper lip now had two points placed at the highest point of the so-called "Cupid's Bow". The third point was placed in-between these two. The bottom lip now had three evenly spaced feature points from the centre of the lip. The 33-point (Figure 5.9(g)) scheme had this same structure but added another feature point to the middle of the chin. The nose feature points helped with the messy faces in the nose region, however the middle of the

(a) 9-Point Scheme (b) 19-Point Scheme (c) 23-Point Scheme (d) 26-Point Scheme

(e) 28-Point Scheme (f) 32-Point Scheme (g) 33-Point Scheme (h) 51-Point Scheme

Figure 5.9: Examples of Examined (Unused) Landmarking Schemes

three points was very erratic in tracking, moving back and forth and falling off the side of the nose slightly. The feature points on the upper lip also tracked poorly, as they would swap places or fall down the lip edge towards the mouth corners.

When the newly added feature points tracked well, they tended to solve the issue of the messy faces for the registered meshes. However, the tracking issues, both for accuracy and run-time, required a balance and an intelligent design. The final landmarking scheme devised was built using the knowledge gained from testing the other schemes.

This was the 23-point (Figure 5.9(c)) scheme (The 24-point scheme is the same as the 23-point scheme, but with a chin feature point as well). This scheme abandoned the three eyebrow feature points and instead modified the eyelid feature points. These eyelid points were having trouble during tracking due to the thin and fast movement of the eyelids. By shifting these points away from the eye, to rest on bone structures,

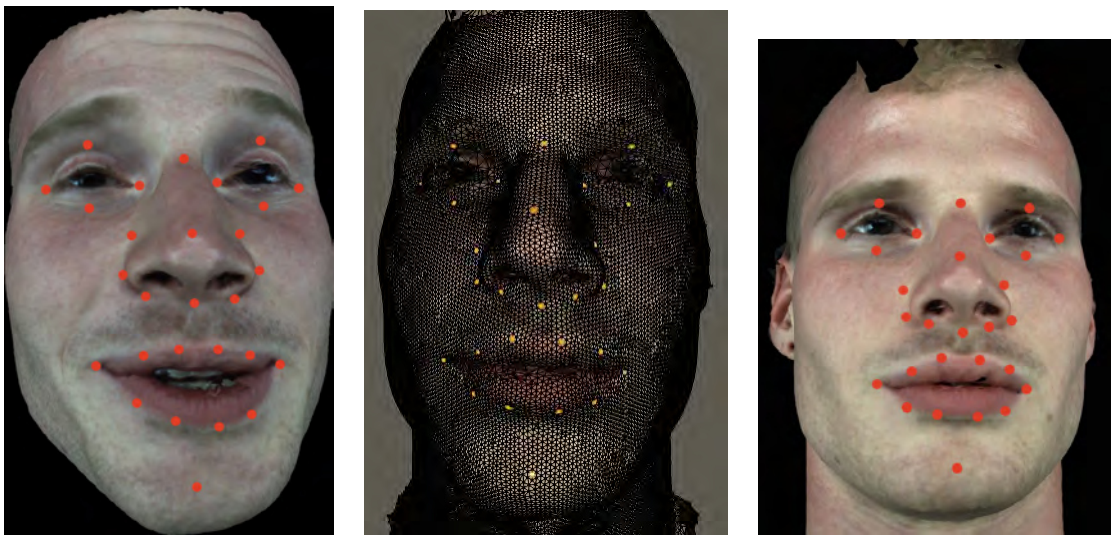
the feature points became much more consistent in tracking, which resulted in a higher-quality registered mesh. Another benefit is that the upper eyelid feature point was now high enough to positively affect the eyebrow region, which resulted in the stability in shape we desired. This new scheme also abandoned the middle feature point on the nose, as it was found to be unnecessary. With the consistent placement of the nose contour feature points, this area of the face (nasolabial furrow) produced a consistent, quality registered mesh. Like the lip corner feature points, the feature points around the nose contour and on the chin were consistently strong choices and needed no large modifications throughout the different schemes. Another modification was the balance of lip feature point placements. While the lip corners proved time and again to be very good feature point locations, it was a challenge to find feature point locations along the lip. The 23-point scheme chose a six-point scheme, where the points are evenly spaced across the lip, on both the top and bottom. This was sufficient in tracking and for most frames for registration. An eight-point scheme for the lips was found to be optimal for both tracking and registration. This resulted in a (newer) 28-point (Figure 5.15(a)) landmarking scheme used for the research in this thesis (the 24-point scheme with the 4 extra lip feature points).

All tested (but unused) schemes can be seen in Figure 5.9. Various views of the chosen 28-point scheme can be seen in Figure 5.10.

The following sections describe the 2D and 3D landmarking tools that were developed. The 2D tool (Section 5.5.2) was initially used but was not sufficiently accurate for our requirements, which is why the 3D tool (Section 5.5.3) was created.

5.5.2 3D-to-2D Landmarking Software

The 2D annotation tool was built using Matlab [163]. Its main purpose is to allow users to load a single OBJ and easily manually annotate a frame with landmarking points. The specified 3D frame (OBJ) is projected down to a 2D image (Figure 5.11) using perspective projection. Each manually landmarked point, x, y, in the annotation tool's (Figure 5.12) 2D image corresponds with an x, y, z of the 3D mesh.



(a) 28-point landmarking scheme (Final Choice)

(b) View of Blender-based [1] landmarking tool, with annotated Frame points

(c) View of the 3D Landmarked

Figure 5.10: Chosen Landmarking Scheme and Blender-based Annotation Tool

These 3D points are used in our tracking and registration approaches. This tool allows for a user to add landmarks, import existing landmarks for the mesh, create or import an existing landmarking scheme, show the feature point numbers (this is important to ensure the points are in a consistent order, following the example in the corner), as well as exporting the 2D image to a PNG.

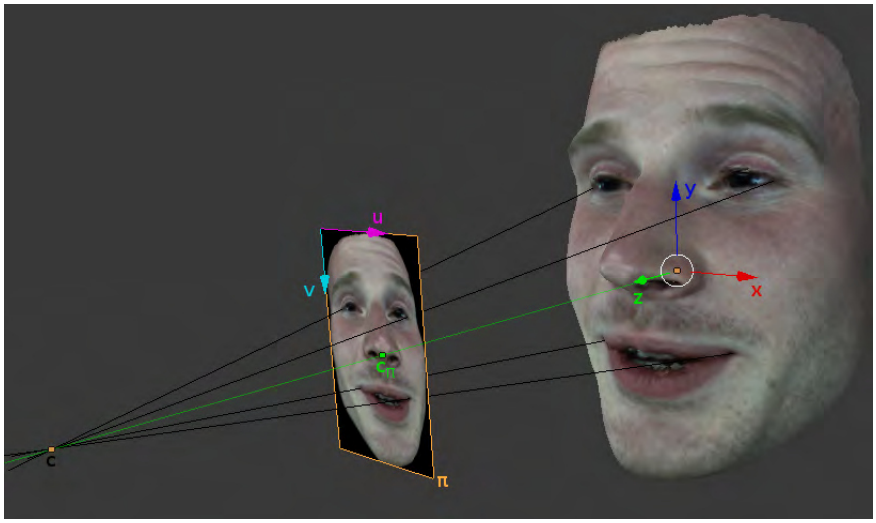


Figure 5.11: Example of projection from 3D mesh to 2D parameterisation image

While this tool works well for simplifying the annotation process of 3D meshes, it does have a few drawbacks. The first is that because it is projected to 2D, the user cannot manipulate the view of the face for placing landmarks. This lack of freedom

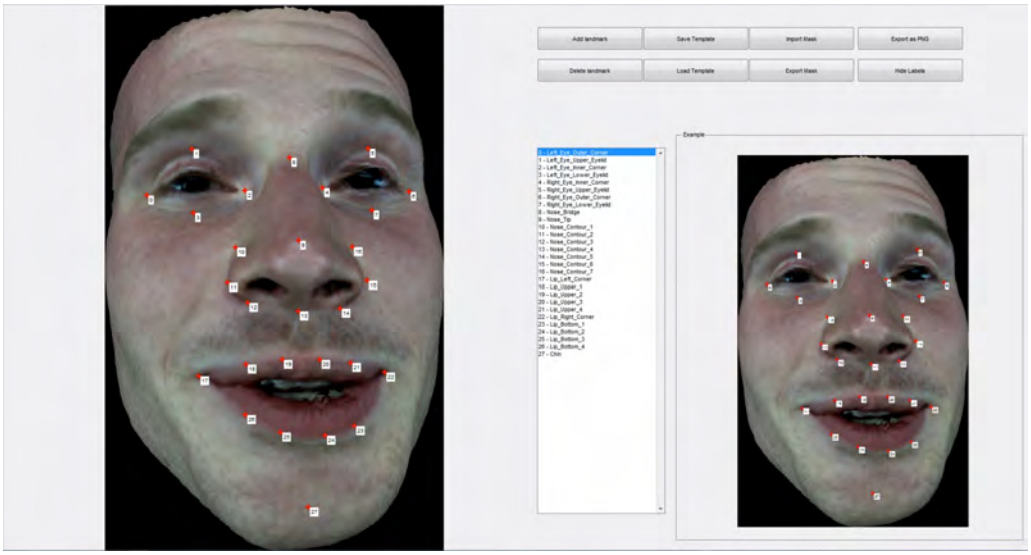


Figure 5.12: Screenshot of the 3D-to-2D annotation tool

means that it is possible that parts of the face occlude other parts. For instance, often times the nostrils occlude part of the nasolabial furrow, which is used in the landmarking scheme. This makes it difficult to provide consistent manual landmarks across all frames (for all subjects). While this issue does not affect the tracking process, it does negatively impact the inter-subject registration process, warping the parts of the face around the nose region. This software tool also limits the control the user has for specific annotation schemes. For example, it would be nearly impossible to accurately annotate all the points of the 51-point annotation scheme for all sequences and subjects, due to the projection method used. A 3D approach would make this possible, as the user would have much more control over the mesh (i.e. zoom, rotate) and of the placement of the landmarking points. The final drawback is the run-time. It takes between one to two minutes to calculate the projection. It takes approximately 30 seconds to load and move the landmarking points to their correct positions. To compare, it is possible to annotate all the sequences using the 3D tool described in 5.5.3 in less than the time it takes to load the 2D projection for each mesh, using this 2D tool.

5.5.3 3D Landmarking Approach

The 3D annotation tool consisted of a Matlab function for selecting the files and launching a Blender [1] file from the Command Line. Blender is a freely-available, open-source, 3D computer graphics software program. The Matlab function opened the Blender program with pre-set options, loaded the mesh file to be annotated, and provided the view of two Python scripts. Figure 5.13 shows a screenshot of this tool. The bottom left window is the Python script responsible for reading in the Matlab-sent commands. The user selects the annotation points in order (using shift and right-click) and then selects the "Run Script" button in the bottom right window. This Python script is responsible for determining the landmarks the user has selected and in which order they were selected, and then writes these out to a file. This file has the same name as the mesh and is used for creating the file needed for the tracking step (another Matlab function calculates a few other needed items, based on the mesh and this landmark file).

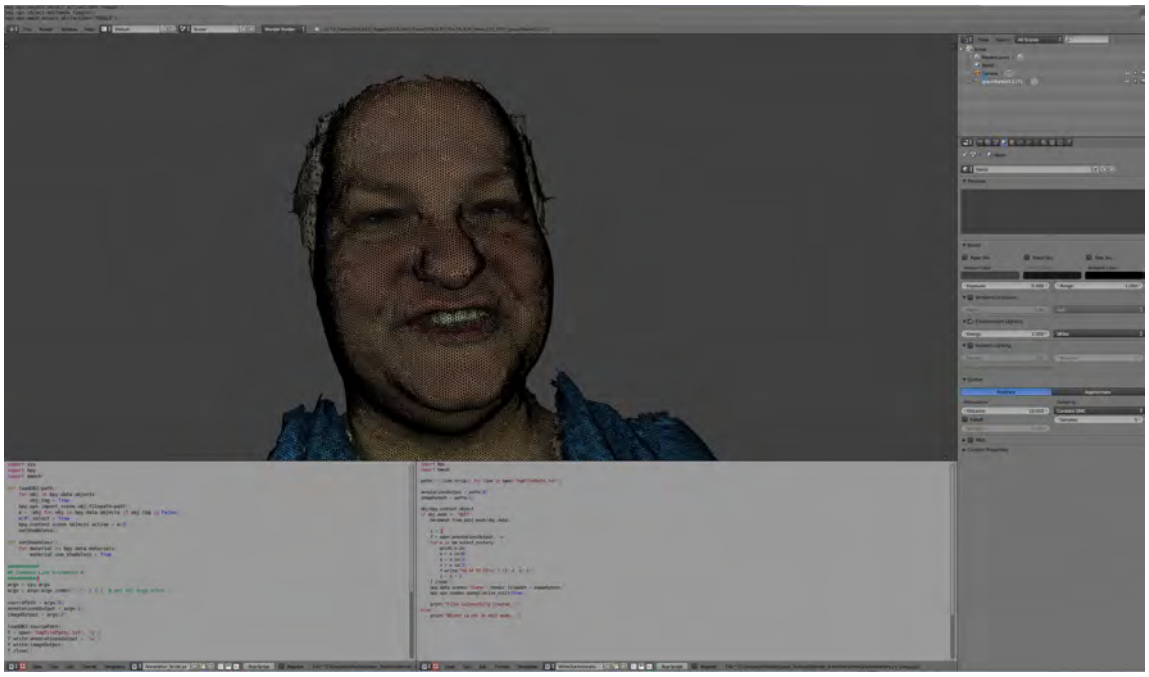


Figure 5.13: Screenshot of 3D Blender [1] Annotation Tool

This 3D annotation approach proved to be a fast, accurate tool that allowed the annotator to have full control of the manipulation of the 3D mesh while placing the landmarks. It allowed for simple adding/removal of landmark points, and the way

in which the Matlab function was written, allowed for seamless transition from one mesh to the next. The speed of annotation was now limited to the ability of the annotator, unlike with the 2D tool. Being able to move the 3D mesh also allowed for consistently placed landmarks across sequences and subjects. The quality of this approach was evident in the output of the inter-subject registered meshes. The issue with the nose region having warped faces due to the 2D tool limitations is not seen in the registered output of 3D tool annotated meshes. Again, the limitation to accuracy is down to the human annotator, and not the software tool used for landmarking.

5.5.4 4D Tracking

Many "4D" tracking approaches actually use *so-called* "2.5D" data [23, 213]. This is data that is comprised of 2D pixel intensity and depth information. These approaches do often use 2D and 3D models for fitting to the captured data, but the tracking is not genuinely "4D", given the type of input data used (2D video with pixel depth information). "Genuine" 4D data, such as data captured on 4D active stereo systems, produces "proper" 4D data, but is typically tracked in 2D (i.e. the 3D tracking problem is simplified to a 2D image-based problem) [38, 41, 54, 80, 242]. There are approaches that do not simplify the problem to 2D, such as Huang et al. [129], but these works fit generic models to the input data which results in unrealistic looking meshes and meshes that have lost fine detail, in favour of matching the global facial expression and head pose. To produce realistic and accurately tracked 4D sequences, we require a tracking approach that uses actual 3D information (3D texture and shape), where "3D texture" comes from the texture information from the 3D mesh faces (not from the 2D texture map) and the "3D shape" comes from mesh surface curvature information. To the best of our knowledge, there are currently no approaches that can use both 3D texture and shape information to perform feature tracking. It is for this reason that we developed our own 3D tracking method.

The feature points described in the previous sections act as anchor points for the dense registration approach described in Section 5.6. These landmark points act as a

sparse correspondence across all annotated meshes and can be used for tracking these points through a 3D sequence. Only one manually annotated frame is required, per sequence, for the tracking approach described in this thesis. These feature points are then tracked in either direction of the annotated frame until all frames of a sequence have been tracked. The feature point is described using both shape and texture feature descriptors.

The tracking of these landmark points through each frame of a sequence is achieved using a similarity search, in a 3D space, using the feature descriptors. Certain constraints (parameter value choices) are used to limit the problem complexity and run-time, as it is unnecessary to search the entire space. With most 4D capture systems, the frame rate is such that there is little change between frames. Thus, the localised region around the tracked point should remain very similar and provides a good initial estimate of the new tracking point's location. There are three approaches that can be used for the similarity search: Two-Frame Sum of Squared Differences (SSD), Linear Combination SSD, and Local Principal Component Analysis (PCA) [183]. The appearance of the surrounding surface of a feature point is given by the texture map. The shape is represented by the curvature. Both texture and curvature information can, and should, be used to form a meaningful descriptor.

5.5.5 Local Neighbourhood Descriptor

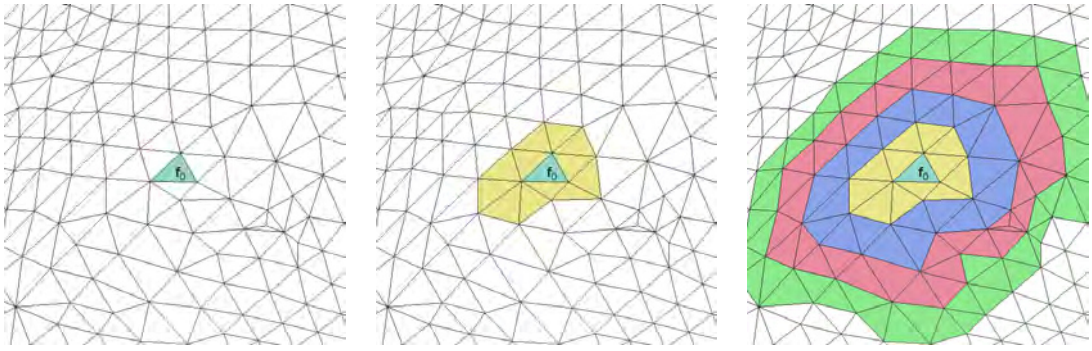
In order to perform a similarity search, a consistent search space that corresponds across sequence frames is required. Many previous works [38, 54, 79, 80, 242, 241] have approached the 3D feature point tracking problem as a 2D image registration problem, in which they attempt to extract meaningful descriptors from the 2D version of the data (e.g. using the 2D image texture maps for tracking). They may use the 3D meshes to help with their tracking approach, but they do not track based on the 3D features. In order to produce more accurate 4D tracking, the 3D information should be utilised. There are some works which use the 3D data for tracking [129], but their their approach of using deformable models results in poorly registered

meshes (i.e. features of the face and the face shape appear slightly different). A 3D SIFT [156] approach might have been a sufficient choice, unfortunately it was not considered by the developers for this tracking method.

We believed one way to achieve accurate tracking (and later inter-subject registration) is to use local neighbourhoods to describe regions of the mesh. Each feature point (tracking point) contains a local neighbourhood descriptor which is built using either/both the shape (curvature) and texture information. Defining a similarity measure for these descriptors is challenging because they are non-corresponding 3D surface meshes. For this reason, the local neighbourhood is orthogonally projected onto a plane tangent to point P.

To achieve this goal of producing consistent, corresponding, geodesic local neighbourhood feature descriptors, an n-ring neighbourhood is calculated (Figure 5.14) and orthogonally projected onto a plane; one which is tangential to the feature tracking point. An approximation of the tangent plane is calculated using the average normal vector of the faces in the local neighbourhood.

Figure 5.14(a) shows feature point P. Figure 5.14(b) shows the first ring of adjacent faces for feature point P. Figure 5.14(c) visualises an n-ring neighbourhood, where $n = 4$, and is constructed in a geodesic manner, by iteratively adding adjacent faces. The value of n is a parameter defined by the user.



(a) Finding the face, f_0, in which feature point P (not pictured) is located

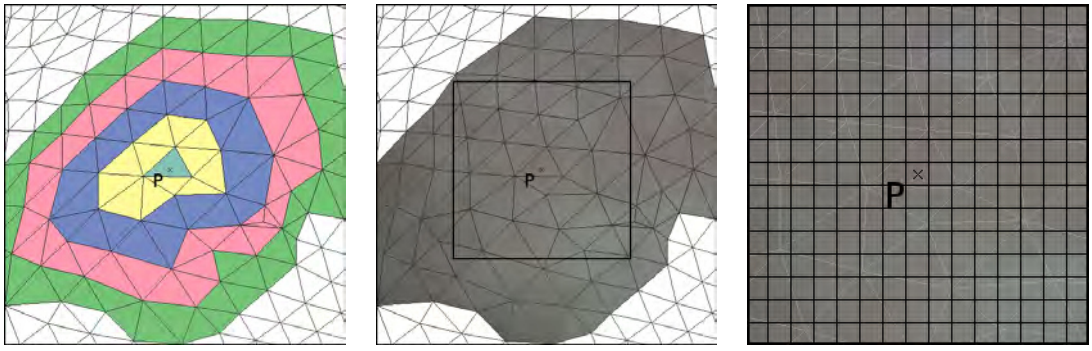
(b) Calculate adjacent faces of f_0; form the first ring of the neighbourhood

(c) Example of a 4-ring neighbourhood around f_0

Figure 5.14: Building an n-ring neighbourhood around a face f_0

For the texture-based feature point descriptor, the local neighbourhood is orthogonally

projected onto the plane using piecewise affine warping of faces (Fig. 5.15(b)). This effectively transfers the uv texture values to the xy-based plane. The 2D-projection of the neighbourhood is then cropped to a uniform shape, such as a square, centred around the tracking point, P. The shape's resolution, r, is specified by a user-set parameter. The optimal r value will balance the quality needed to allow an accurate description of the neighbourhood with computation time. The square is discretised into $r \times r$ pixels (Fig. 5.15(c)). The resulting texture descriptor, D, is an $r \times r \times 3$ matrix because each pixel is described by three intensity values (RGB). This resulting matrix is referred to as $D(P)$, and an example of this process can be seen in Figure 5.15.



(a) Feature Point P and its local $4-ring$ neighbourhood

(b) Warped texture of $4-ring$ neighbourhood on tangential plane and cutout square

(c) Cropped and discretised descriptor matrix $D(P)$

Figure 5.15: Calculating Texture-based Feature Descriptor

The shape-based feature point descriptor is calculated in a very similar manner, except it uses curvature information. The term *curvature* refers to a tensor field on smooth surfaces [82]. The mesh data used in this thesis are surface meshes, and, thus, are polygonal approximations of smooth surfaces. Therefore, an estimator for the curvature tensor is needed. In [11], such an estimator is described in detail and [184] provides a popular implementation. For each vertex, the minimum (κ_{min}) and maximum (κ_{max}) curvature is computed using the directional vectors U_{min} and U_{max}, and the Gaussian curvature κ_{gauss}. The Gaussian curvature is defined as the product of the maximum and minimum curvature:

$$\kappa_{gauss} = \kappa_{min} \cdot \kappa_{max} \tag{5.3}$$

The Gaussian curvature represents the surrounding curvature of a control point and the values for the $r \times r$ descriptor matrix are determined by Barycentric interpolation.

To summarise: The texture feature is described by an $r \times r \times 3$ matrix ($r \times r \times RGB$). The shape feature is described by an $r \times r$ matrix where Gaussian curvature values are calculated using bilinear interpolation of the vertices' curvature values. This information (texture and shape) is projected into 2D space for the search portion of the tracking approach (See Section 5.5.6). Thus, the search is done in 2D using 3D information. This is different from the works that use the 2D information for tracking (e.g. tracking using the *uv* texture map) and use the output to correspond to 3D locations.

The use of texture-based descriptors does not exclude curvature-based descriptors, and vice versa. However, combining both to one descriptor that can be used for local search would require proper weighting as the two are using incompatible units of measurement. In the current implementation the local search is conducted separately for both descriptor types. The results of both searches are then merged in a weighted sum by computing the mean of both similarity maximums. These descriptors are not invariant to rotation, however, given the frame rate of the sequences any rotation will be insignificantly small. Therefore, for this implementation, rotation invariance is not an issue.

5.5.6 Local Similarity Search

The process described in the previous section (Section 5.5.5) is used to compute the search pattern $D(P_k)$ and the search space X_S. The former is a neighbourhood descriptor of a tracked point P_k in frame k and the latter is a descriptor of a larger

neighbourhood centred around the initial estimate of P_{k+1}. The initial estimate is the point on the surface of frame $k+1$ that is closest to P_k.

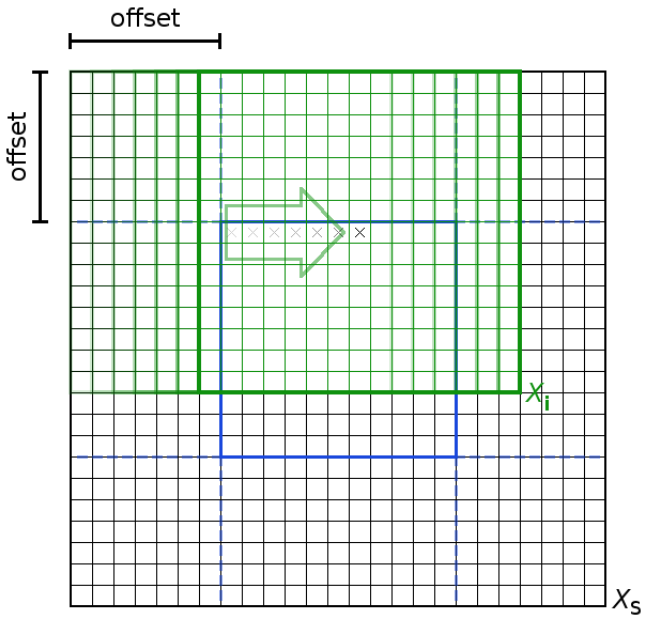


Figure 5.16: Local Search Scheme

The tracking algorithm provides three approaches of similarity measurement. The first approach calculates the sum of squared differences (SSD) between the neighbourhood descriptor $D(P_k)$ and the current subset, X_i, of the search space X_s. The second approach computes the SSD between a linear combination of n descriptors of the previously tracked frames $\sum_{j=k-n+1}^{k} a_j D(P_j)$ and X_i. The larger the value n, a parameter set by the user, the more probable it becomes that any warping distortions will be averaged out. The third approach creates a local PCA model, $D(P) = PCA(P_0, \ldots, P_{k-1})$, for the neighbourhood of each tracking point P. This requires a different similarity function, as the search pattern is projected into PCA space. Every X_i also has to be projected into PCA space. The norm of the resulting vector is the Euclidean distance from X_i to the mean of $D(P)$ in PCA space. Its reciprocal is used as a similarity measure.

Figure 5.17 shows the output of three tracked frames from a smile sequence.

(a) Neutral Expression Frame (b) Middle Frame (c) Peak Expression Frame

Figure 5.17: Tracked Smile Sequence Frames

5.6 Dense 4D Registration

An initial avenue explored for dense 4D registration was the approach of Sidorov et al. [211]. This approach is a groupwise non-rigid registration approach that allows for the inter-subject registration of high-resolution meshes. This technique is limited to diffeomorphic deformations of the face (such as requiring a closed mouth), which makes it unsuitable for use with conversational facial data. Additionally, using the code provided by the Sidorov et al., this approach proved to be a very time-consuming, computationally-intensive method for processing large amounts of sequence data. So much so that a new approach was necessary.

The works of [80, 123, 124] utilise TPS approaches to achieve dense feature correspondence. The strength of a TPS-based approach for dense registration over the groupwise technique is that it is not limited to only diffeomorphic deformations. This added robustness is useful not only for registering frames for a variety of facial expressions (such as those where the subject is talking), but also for inter-subject registration of 3D meshes. For these reason, we chose to use the TPS technique for our inter-subject registration approach.

5.6.1 Inter-Subject Registration

The feature points tracked in Section 5.5 are the control points used for creating a dense correspondence between meshes (i.e. creating registered meshes). Any sequences which contain the same control point annotation scheme can be made correspondent. This easily allows for both intra-subject and inter-subject registration (Figure 5.18).

For our research we are only interested in the facial area of the scans of the subjects. To remove unwanted areas (e.g. chest, shoulders, neck, hair) we select a frame from the subject whom we wanted to be the reference mesh (i.e. the mesh all other meshes are registered to). This has been termed the *registration mask*. Using Blender, we select the vertices from the section of the face we want to keep, and deleted the rest. This step could be done using a variety of tools, but this manual editing in Blender is fast, accurate, and reliable. This *registration mask* is annotated using the chosen landmarking scheme and is used as the reference mesh (also know as the target mesh) for the registration process. Figure 5.18(a) shows the original cleaned mesh used to create the reference mesh (Figure 5.18(b)). Figure 5.18(c) shows a cleaned mesh that was tracked using our tracking method, and Figure 5.18(d) shows the inter-subject registered mesh, which was created by registering the cleaned mesh to the reference mesh. The following paragraphs provide details of this process.

(a) Subject A - Annotated (b) Subject A - Registered (c) Subject B - Annotated (d) Subject B - Registered

Figure 5.18: Two Annotated and Registered Subjects. The annotated points in (a) and (c) are used to register the frames to the reference mesh, (b). Therefore, (b) and (d) are inter-subject registered.

A reference mesh frame, M_{ref}, is chosen as the target mesh for registering any

other mesh frame, M_k. Each vertex in M_k must have a corresponding vertex in M_{ref}. This is achieved using Thin Plate Splines (TPS), which is an approach for interpolating arbitrarily spaced, multidimensional data points, while minimising the bending energy [48, 99].

The registered vertices for M_k, called M_{new}, are interpolated from M_{ref}, using each frame's control points and M_{ref}'s vertices [246]. The interpolated vertices of M_{new} closely match the original shape of M_k, and contain the triangulation of M_{ref}. Remaining displacements between the surfaces of M_{new} and M_k are removed by "snapping" each vertex of M_{new} onto the surface of M_k. This step also allows for the calculation of M_{new}'s texture coordinates.

(a) Subject A (b) Subject B (c) Subject C (d) Subject D

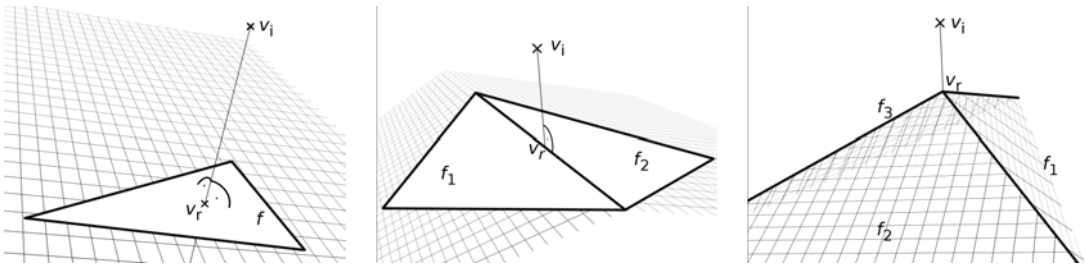
Figure 5.19: Textured Wireframe of Inter-Subject Registered Frames

5.6.2 Snapping

The naive approach of simply snapping each vertex of M_{new} to M_k fails when multiple vertices of M_{new} are assigned to the same vertex of M_k. For this reason, a more advanced approach is needed, where the vertices from M_{new} are snapped to the surface of M_k.

For each vertex, v_i, in M_{new}, the closest vertex, V_c, in M_k is found. Only the faces and edges attached to V_c are considered in the snapping algorithm, as it is improbable other faces/edges will provide a better match. The result is a vertex, v_r, which lies on the surface of M_k.

Figure 5.20 illustrates the three cases that occur when performing this process.



(a) (b) (c)

Figure 5.20: Three cases of mapping reference vertex v_i onto the target mesh (mesh grid used for 3D depth visualisation): (a) For the vertex v_i a face f in the target mesh can be found that is intersected at v_r by an orthogonal line l with $v_i \in l$; (b) if no such face can be found the algorithm searches for an edge in the target mesh that is intersected by an orthogonal line l with $v_i \in l$; (c) if neither a face according to the first case nor an edge according to the second case can be found we follow the naive approach and map v_i onto the closest vertex of the target mesh.

Barycentric coordinates are computed from the intersection point of v_i and the surface. These coordinates are used for calculating the exact x, y, z-position in 3-D space, as well as the uv-position in M_k's texture map. Once every vertex, v_i, of M_{new} is snapped onto M_k, the frames have corresponding (registered) meshes. If the original mesh frames have congruent texture maps, the registration process is complete. If not, one simple, final step needs to be performed: use piecewise affine warping for each mesh, so that all faces in M_{new}'s texture map are warped to correspond with M_{ref}'s texture map. Figure 5.21 shows, on the top row the original (cleaned) Unified Texture Maps for a single frame for each subject. The orientation of the UTM, like the 3D frame at the time, has no correspondence across a sequence or across subjects. On the bottom row is the result of the texture map registration step (piecewise affine warping for the faces in each mesh). The orientations are consistent and the triangulation corresponds to the registered 3D frames.

(a) Subject A - UTM (b) Subject B - UTM (c) Subject C - UTM (d) Subject D - UTM

(e) Subject A - Registered (f) Subject B - Registered (g) Subject C - Registered (h) Subject D - Registered

Figure 5.21: Top Row: Unified Texture Maps from pre-processing pipeline. Bottom Row: Registered Texture Maps

5.7 Tracker and Inter-Subject Registration Evaluation

5.7.1 Tracker Evaluation

In order to assess the quality of tracking, our colleague Lukas Gräser performed both a detailed and general evaluation. For both evaluations, the 28-point landmarking scheme and 3D Blender-based landmarking tool described in Section 5.5 were used. For the detailed validation, four validated smile sequences (see Section 3.4 for details on the data used) were annotated every 20^{th} frame to determine the frame ground truths. The average distance to the ground truth for the four sequences was 4.08% of the inter-ocular distance (IOD). For a more general evaluation, all 42 validated smile sequences were annotated at the first, middle, and last frame (see Figure 7.1 in Chapter 7 for details about the sequences). The average distance to the ground truth was 3.96% of the IOD. Note that the ground truth is prone to displacement as well, as manual annotation is not 100% accurate. To check for this type of error, one sequence was annotated twice, by the same annotator, resulting in an average

distance of 3.52% of the IOD.

More detailed evaluation, including using multiple 4D datasets and different expression types and sequence lengths, is still required to prove the robustness of this tracking method, but these preliminary results, coupled with the output evaluated in Chapter 7, supports the quality of this approach.

5.7.2 Inter-Subject Registration Evaluation

The main evaluation of our the inter-subject registration approach comes from the classification and perceptual experiments in Chapter 7. Specifically, Section 7.3.1 shows how the minute appearance differences of two different smile types is preserved through the registration (and modelling) process. Section 7.5 also supports these methods with results that show that the registered (and modelled) smile sequences are, overall, perceived as highly-realistic.

The figures below (Figures 5.22, 5.23, 5.24, and 5.25) provide a visual comparison of the original (cleaned) mesh with tracking points and the mesh after dense registration. Subject A was used as the reference mesh (as described in Section 5.6). Therefore, Figure 5.22 shows intra-subject registered meshes, and the other figures show inter-subject registered meshes. It is important to note that these examples were not "cherry-picked" (i.e. selectively chosen). They represent frames from a variety of sequences and from different stages of the sequence (i.e. beginning, middle, end).

It is clear to see that our tracking and registration approach produces registered meshes nearly identical to the original meshes. Our approach produces similar visual results to recent, state-of-the-art approaches [38, 54, 79, 80] for producing accurate, high-resolution, registered meshes. The appearance of the eyes and the mouth is occasionally slightly differently than the original capture, but this is mainly due to the nature of the capturing technology. The specularity of the teeth and eyes results in these areas not being captured well by our active stereo system. This is a common issue and we will be exploring solutions to this problem, such as the mouth tracking

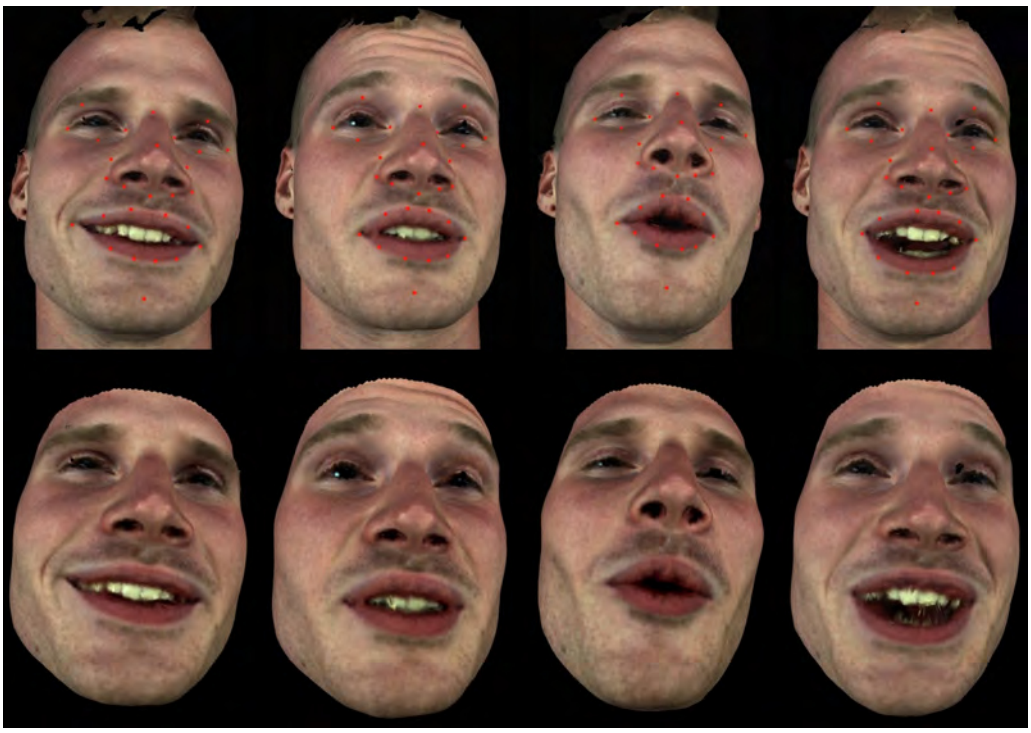


Figure 5.22: Subject A - Top Row: Original Meshes with Tracked Points, Bottom Row: Registered Meshes

and edge detection approach described in [54].

5.8 Summary

In this chapter we described the development of a semi-automatic sparse tracking approach and a dense registration method, which allows for the creation of highly-realistic, inter-subject registered sequences. A 3D annotation tool was created which allows for quick landmarking of a single frame from a sequence. This annotated frame is used as the starting point for a similarity search based on local neighbourhood feature descriptors. This tracking method did not appear to suffer significant drift errors for any of the subject sequences that were tracked. For the data used in this research, only 28-points were required for producing corresponding data that was nearly identical to the original 3D data. This was achieved by using the 28-points as anchors for a Thin Plate Spline based dense registration approach, which included an additional *snapping* step for increased accuracy.

With the ability to now inter-subject register sequences of data, the next step for our

Figure 5.23: Subject B - Top Row: Original Meshes with Tracked Points, Bottom Row: Registered Meshes

research is building statistical models using the registered data. Chapter 6 describes our process for building 3D models and representing our sequences as single entities. Chapter 6 also introduces our coupled statistical model approach for modelling conversational interactions. In Chapter 7 these approaches are used in classification experiments and perceptual studies.

Figure 5.24: Subject C - Top Row: Original Meshes with Tracked Points, Bottom Row: Registered Meshes

Figure 5.25: Subject D - Top Row: Original Meshes with Tracked Points, Bottom Row: Registered Meshes

Chapter 6

3D/4D Statistical Modelling, Modification, and Synthesis of Facial Expression Data

6.1 Overview

In this chapter we introduce two novel contributions of this thesis. The first is covered in Sections 6.2 and 6.5 and discusses our technique for modelling and representing 4D sequences as single entities, and modifying these sequences in various ways. This approach is used for creating 4D, highly-realistic, modified expression sequences, which competes with state-of-the-art approaches [38, 54, 80, 143, 238, 243, 249]. In this section we explain the decisions that were made to allow us to achieve this goal, and explain our process for creating these highly-realistic sequences.

The second novel contribution discussed is our coupled statistical modelling approach for modelling and synthesising conversational interactions (Section 6.6). Using the single-entity expression sequences from our conversational data, we develop a coupled model approach that uses the feature vectors of conversational interactions. This approach allows for the analysis of conversational interactions and the synthesis of sequences.

Evaluations of our methods are performed in classification and perceptual experiments, discussed in detail in Chapter 7.

6.2 Modelling Introduction

Statistical models, specifically Active Appearance Models [72, 73], are a popular method for face-based research, such as for identifying pain facial expressions [15], text-to-speech visualisation [14], face tracking [98, 255], face recognition [16, 144], and 2D, real-time modifications of facial expressions [218]. AAMs can be used in a variety of ways for facial expression analysis and synthesis, and our interest is to use AAMs for modelling facial expression sequences, so that we can analyse the facial characteristics of an expression, as well as synthesise new expression frames by interpolating values in the model. By slightly altering the principal component values of a projected frame, one *could* unproject and render a slightly altered, but realistic, facial expression frame. This ability to synthesise new, uncaptured data makes AAMs a very powerful tool for stimuli creation. Their quantification of data also makes them well-suited for analytical applications. Polynomial regression techniques are useful for representing a sequence of data as a single entity. In this work, a single entity is represented using a feature vector, which consists of the polynomial coefficients. These coefficients can be used in analysis methods, or modified and used to synthesise new expression sequence data. If these coefficients represent the frontchannel and backchannel parts of a conversational interaction, they can be concatenated as a single feature vector and used to build a *coupled statistical model* (See Section 6.6 for full details). By including other useful information, such as an *offset value*, which is the number of frames from the start of the frontchannel expression to the start of the backchannel expression, this coupled model can describe the characteristics of the facial expressions used in an interaction. This model can also be analysed, modified, and used for synthesising new expression interactions.

While Active Appearance Models have been a popular method for face-related research over the past 15 years, to the best of our knowledge, no other work has

used AAMs for feature extraction and curve fitting techniques, such as polynomial regression, for 4D sequence representation. As well, no other work has used the polynomial coefficients of frontchannel and backchannel interactions from 3D AAM principal component values in a coupled statistical model. The following sections describe the process used in this research for doing so.

6.3 AAMs of Conversational Expressions

To build 3D/4D statistical models of conversational expressions and expression interactions, a database of dyadic, synced conversations is required. Such a database was developed for this thesis. The conversations were annotated for frontchannel and backchannel conversational expressions. Full details can be found in Chapter 3.

To build an Active Appearance Model (AAM) [73] of conversation sequences for all subjects, the data must be inter-subject registered. Sparse correspondence was achieved using a 4D tracking approach which can use both 3D shape and texture (Section 5.5.4) (Note: The data used for our experiments (Chapter 7 uses only the 3D texture information for tracking). Dense correspondence is achieved using a Thin Plate Spline (TPS) based algorithm (Section 5.6).

This registered data is used as input to an AAM. The AAM performs 3D Procrustes analysis, and PCA for shape, texture, and combined, weighted shape/texture (Section 2.7.3). The steps below describe the steps for calculating PCA (See [74] for details)

1. Calculate the covariance of the data

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T \tag{6.1}$$

where x_i is a data point in x and N is the number of elements in x.

2. Compute the eigenvectors ϕ_i and eigenvalues λ_i of C

When performing AAM, the eigenvalues λ_i are sorted in descending order (highest

variation to lowest). Thus, Principal Component (PC) 1 contains the highest variation, PC 2 the second highest, and so on. This allows one to easily retain a reduced number of principal components k that represent a percentage of the total variance ($V_{total} = \sum \lambda_i$), based on an arbitrarily chosen threshold value ϵ.

Thus, k is the smallest number that satisfies the following formula (Equation 6.2)

$$\sum_{i=1}^{k} \lambda_i \geq \epsilon V_{total} \tag{6.2}$$

3. Use the eigenvalues to approximate a new data by varying $\mathbf{b}$ in $x \approx \bar{x} + \Phi\mathbf{b}$ where $\mathbf{b}$ is constrained by $b_i = \pm 3\sqrt{\lambda_i}$.

In our work, Step 3 is used to describe a new 3D frame in relation to the model. These new values are described using its *Principal Component* weights, or $\mathbf{b}$ (referred to as *bVectors* in the rest of this thesis).

Using a 3D frame's *bVectors* and the AAM, we can *project out* of the AAM (model space) to get a reconstructed 3D frame (Figure 6.1(b)). For our implementation, this process will not perfectly reconstruct the original frame (Figure 6.1(a)), because we do not keep all of the AAM parameters (for space and computation reasons we only we keep 95%, or $\epsilon = 0.95$). Thus, the projection out of the model is not a lossless process. However, if the frame was used to build the model, it should be very close to the original. If the frame was not used during the building phase of the AAM, the reconstructed frame will be the model's best representation of the original frame.

Aside from being able to describe a large, complex 3D frame by a handful of numbers (thus, reducing the dimensionality), the benefit of this type of representation is that it allows for the data in the model to be compared, as well as modified. Each frame is described by the model by its deviation from the mean. Values which fall in-between these frames are interpolated values. For example, if there were two frames of facial expressions, one neutral and one a peak smile expressions, the AAM could interpolate the values of a half-smile, and unprojecting those *bVectors* could produce a 3D frame of the face performing a half-smile. This serves only

(a) Registered Frame (b) Unprojected Frame

Figure 6.1: Before (Registered Frame) and After (Unprojected Frame) AAM Projection

as a simplified example, but Figure 6.2 shows a similar example. Frames from a neutral-peak-neutral smile expression were used to build an AAM model. The individual images in Figure 6.2 show the deviations from the AAM model mean. -3 to +3 standard deviations is a commonly used range for observing realistic deviations from the mean. All of these images were produced by calculating the deviation from the mean and unprojecting the resulting *bVector* values. This ability is one reason why AAMs are not only useful for analysis, but also synthesis of data. This makes AAMs a powerful tool for synthesising plausible data. With enough examples of conversational facial expressions and expression interactions, we hope we can synthesise realistic expressions.

Figure 6.2: AAM Model: -3 to +3 Standard Deviation for Mode 1

Figure 6.3 shows the AAM mean for an AAM built using all frames from every smile interaction sequence from every subject. The softer texture for this mean mesh is

an effect of the averaging that is done. This AAM model is interesting because it has the ability to model a variety of facial movements and expressions from all four subjects who were involved in the data capture. This is possible because of the inter-subject registration methods described in Section 5.6.

Figure 6.3: AAM Mean from all conversation smile interaction frames for all subjects

One shortcoming of AAMs when wanting to model sequences is that AAMs only model in the space dimension, but not the time dimension. To utilise the strengths of the AAM for model building, analysis, and synthesis, while including a time dimension, a solution was devised that would represent a sequence as a continuous, length-invariant entity: polynomial fitting. In Section 6.4, we explain why this is the best approach for achieving sequence representation for use in modifying sequences and building coupled statistical models for analysis of conversational interactions and predicting parts of conversational interactions.

6.4 Expression Sequence Modelling

Conversations are filled with dynamic facial expressions. These expressions can differ greatly in intensity and length, and their variations may encode different,

important aspects of conversational interactions. Since our goal is to build coupled statistical models of conversational expression interactions, the sequences will need to be comparable, while still maintaining their expression characteristics.

As first explained in Section 6.3, by projecting a 3D frame into the AAM, we get parameters that describe the 3D frame in relation to the AAM. These values are known as *bVectors* and are the principal component (PC) weights for the projected frame.

Equation 6.3 is used to describe a sequence of *bVectors* B.

$$\mathbf{B} = \begin{bmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_n^1 \end{bmatrix} \| \begin{bmatrix} b_1^2 \\ b_2^2 \\ \vdots \\ b_n^2 \end{bmatrix} \| \cdots \| \begin{bmatrix} b_1^m \\ b_2^m \\ \vdots \\ b_n^m \end{bmatrix} \tag{6.3}$$

where m is the number of frames in the sequence.

Figure 6.4 shows the *bVector* values (Y-axis), for the first principal component (Legend: PC1), for each frame in a sequence (X-axis) (which is the first row in B). In this instance, the sequence is a controlled smile expression (neutral-peak-neutral). Please note, future figures in this work will describe 4D sequences in the same manner (*bVector* value on the Y-axis, sequence frame number on the X-axis).

Finding an appropriate method of representing sequences of varying lengths and characteristics, as single, comparable entities, is a challenge. This representation would not only need to describe the sequence in such a way that would allow for easy manipulation of its characteristics (e.g. amplifying a smile sequence, decreasing the speed of an expression sequences), but also allow sequences to be comparable. The latter requirement is essential for building our coupled statistical model (further details can be found in Section 6.6).

An initial avenue of exploration was to use a popular time-series analysis technique (e.g. DTW, HMMs) or Gaussian Mixture Model (GMM) for sequence representation

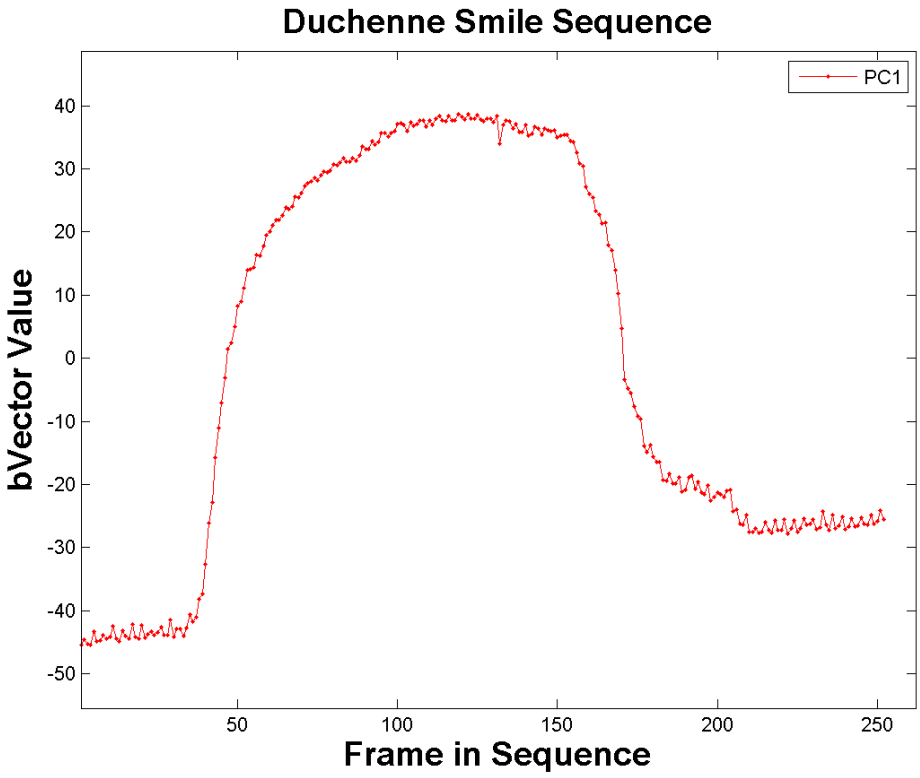


Figure 6.4: *bVector* Values of a Smile Sequence - PC 1

and comparison. Section 6.4.1 explains why these approaches are insufficient for our requirements. The next area explored was popular curve fitting techniques (e.g. B-Splines and polynomial regression). Sections 6.4.3 and 6.4.2 describe these approaches and why we decided to use polynomial regression for our sequence representation method.

6.4.1 Time-Series Analysis and Mixture Models

Time-series analysis techniques, such as Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) [33, 34], are popular approaches for comparing signals.

DTW and its variants (e.g. Derivative Dynamic Time Warping [138]), work well for aligning time-dependent signals which are fairly similar in their characteristics, such as two similar audio signals or two similar facial expressions made by different people [20, 138, 171]. An HMM is a piecewise stationary modelling approach. That is, for discrete states with instantaneous transitions, an HMM attempts to model the likelihood of these transitions. HMMs and its variants (e.g. adaptive HMMs) are

useful for a variety of applications, such as speech recognition [193, 194] and face recognition [155, 173].

Both approaches have the same drawback for use in our research, which uses conversational data. Time-series analysis techniques do not tend to work well when the signals to be compared are not of similar length and similar characteristics. In conversations, even the same type of expressions, such as smiles, may greatly differ in their trajectories due to factors such as the individual speaking before/during the expression, the individual holding what we have termed a "resting smile" (an individual maintaining a masking smile that remains unchanged for long periods of time), or because the individual is transitioning from a different expression to a smile. As an aside, it is for these reasons that many machine learning approaches also fall short. If we were attempting to match similar expression sequences and compare the two, such as in [20], a DTW approach would be a reasonable choice. If our goal was to build models for predicting the next frame in a 3D sequence, then an HMM approach could be very useful. However, our requirement is the modelling of variable-length, variable-characteristic expression sequences for use in modification methods and our coupled statistical modelling approach. These coupled models will allow us, among other things, to understand how a part of one subject's sequence in an interaction can have an effect on the other subject (and vice-versa).

Gaussian Mixture Models (GMM) is another approach we explored. A GMM is a probability model for density estimation and clustering and is very popular for classification applications, such as speaker verification [60, 240]. However, this approach assumes a Gaussian distribution, which is inconsistent with the conversation data we are using for our conversational interaction modelling.

These approaches, while useful for many applications, are insufficient for our research requirements. We require an approach that allows us to represent each variable-length, variable-characteristic sequence as a singe, set-length entity. This approach should allow us to interpolate new values in the sequence (e.g. decrease the expression speed to between-frame values), be easily modifiable (e.g. amplify the expression

sequence) and also allow for equal comparison of sequences (for use in our coupled statistical model). Given these requirements, the next reasonable area of exploration was curve-fitting techniques.

6.4.2 Curve Fitting: Polynomial Fitting of Expression Sequences

Polynomial regression is performed on each row of B (Equation 6.3) resulting in n formulae. Each regression formula models the behaviour of each principal component across the sequence. By selecting a degree d for all sequences such that $d << m$, we reduce dimensionality and gain the ability to directly compare sequences of variable length m, which is important for our coupled statistical models. In fact, these formulae coefficients are used as part of the feature vectors which are used as input to our coupled statistical model (Details given in Section 6.6). The other components of the feature vector are described later in this section.

Figure 6.5 shows a 14^{th} degree polynomial fit on a 252-frame long sequence. This process can be done for each row of B, however in our experiments (Chapter 7) we only used the first 3 rows (PCs), as they describe the amount of variation we required for modelling and producing accurate looking synthesised meshes. That is, PCs after the first 3 (for our data) contained such little visually important information ($< 1\%$ variation for each PC) that we determined that they were not required for producing accurate, realistic-looking meshes.

In this work, before a polynomial fit is calculated for each row of B, the row is first standard score normalised [141] (Equation 6.4) and shifted to the mean (Equation 6.5).

$$\frac{X - \mu}{\sigma} \tag{6.4}$$

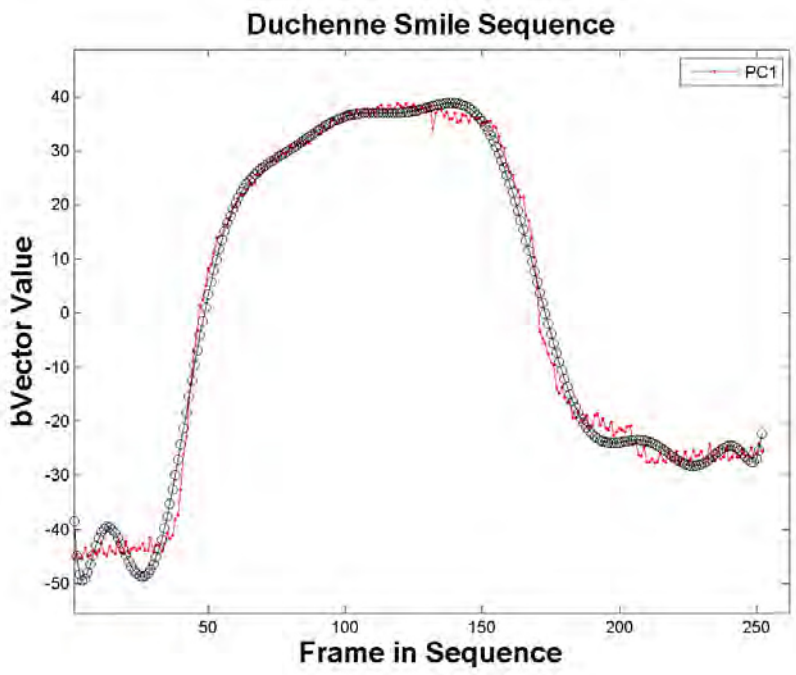


Figure 6.5: Polynomial Fit Example

$$X - \mu \tag{6.5}$$

This is an important step for retaining each sequence's characteristics, while also ensuring all of the sequences reside in the same polynomial space. The importance of this becomes evident when we use the coupled models to predict sequence values (see Section 6.6 for more information).

The polynomial coefficients calculated for each row in B are concatenated with the normalisation and shifting values, as well as the number of frames in the sequence. This information helps for reversing the process when synthesising the sequence. The feature vectors for each row (PC) are concatenated to produce a single, combined feature vector, which describes all parts of the expression sequence.

One of the major benefits to this approach is that it allows for the easy modification of sequences. By projecting new frames using the regression formulae for B, we can modify the sequence, such as changing its length or amplitude. For instance, to amplify facial expressions you simply amplify the polynomial curve and calculate the new $b\mathit{Vector}$ values, as seen in Figure 6.6 as "70% Decreased" and "70% Increased"

(Note: Figure values have been inverted for a more intuitive example of a smile sequence and its modified curves). Projecting the *bVectors* out of the AAM models allows for the synthesis of newly modified sequences while, importantly, retaining the expression characteristics. This is extremely useful for modelling and synthesising realistic, believable facial expressions.

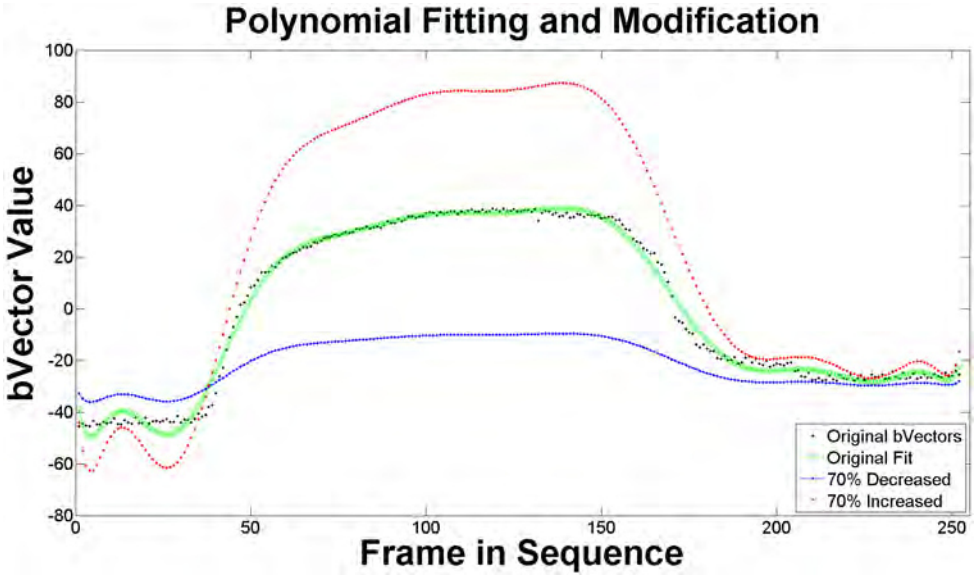


Figure 6.6: Example of Modifying a Polynomial Fit Curve

The main strength of this approach, however, is that it allows sequences of different lengths and characteristics to be represented by the same number of values, which is critical for building our coupled models of conversational interactions.

While it is true that this approach is susceptible to *Runge's phenomenon* [199], which states that oscillations have a tendency to occur at the edges of a polynomial curve for high-order polynomials, at the time we felt the benefits of a single polynomial fit over a piecewise polynomial or spline-based fitting approach (See Section 6.4.3) were still greater. While this oscillation caused slight perceivable movements in the synthesised sequences (i.e. it gave the appearance of the lip corners moving slightly into a smile and back to neutral before the full smile began) it did not seem to have a significant effect on the perceptual experiment results (See Chapter 7) and no subjects from the pilot study commented on these movements. For future experiments that aim to understand perception of expression dynamics this issue will need to be completely resolved. For the experiments performed in Chapter 7,

the single polynomial fit approach was a sufficient choice. The modification of these curves and the synthesis of new sequence data (which were used in the experiments in Chapter 7) is discussed next, in Section 6.5.

6.4.3 Curve Fitting: B-Spline Fitting of Expression Sequences

A popular approach for curve fitting is the use of B-Splines [66, 88, 161, 162]. A B-Spline is a piecewise polynomial fitting approach where the individual pieces are connected by "knots" (also know as *control points*). Low-order polynomials are used to connect these knots and, thus, if the same number of knots (and polynomial degree) are used for all sequences, the resulting feature vectors for each sequence will be the same length. Naturally, this approach appeared to be a good solution to our problem of representing sequences as single, equal-length entities.

Figure 6.7 shows an example of such a fit where a least squares cubic interpolation approach has been used on PC 1 (mode with the largest variation). 30 equi-spaced control points were used for this smile sequence. This number was empirically chosen when considering the balance between fitting the *bVector* data properly while avoiding over-fitting. Another benefit to the B-Spline fitting approach is that it allows for each knot section to be fit with a lower-order polynomial (in this example, a 3^{rd} order polynomial) than would be needed if fitting a single curve, which avoids an oscillation issue common with high-order polynomials [199].

This is a strong approach for curve fitting, but at the time we believed this approach would produce issues when we we went to build our coupled models. Each polynomial section should represent the same action types. So, for the smile data, one could break the smile curve into three parts: onset, peak, and offset. Then, for each smile sequence, each polynomial section would represent, regardless of its length, the same type of action. Unfortunately, the conversation data is very variable and creating an automatic approach for determining the optimal location of control points is a complex issue (see [102] for details), and not feasible given our data. Figure 6.8 shows four examples of this data diversity, which is very common for each principal

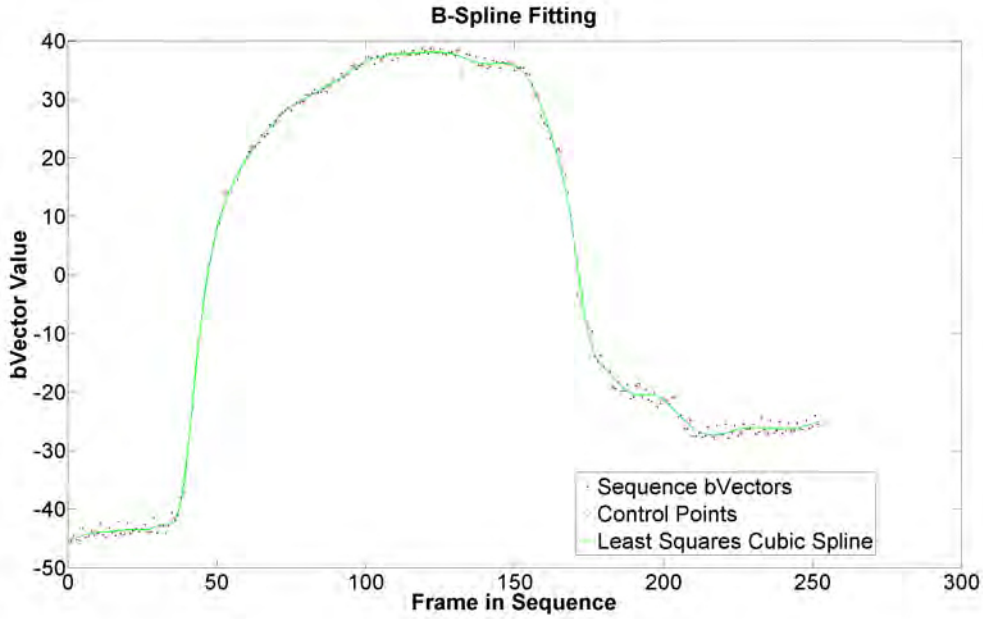


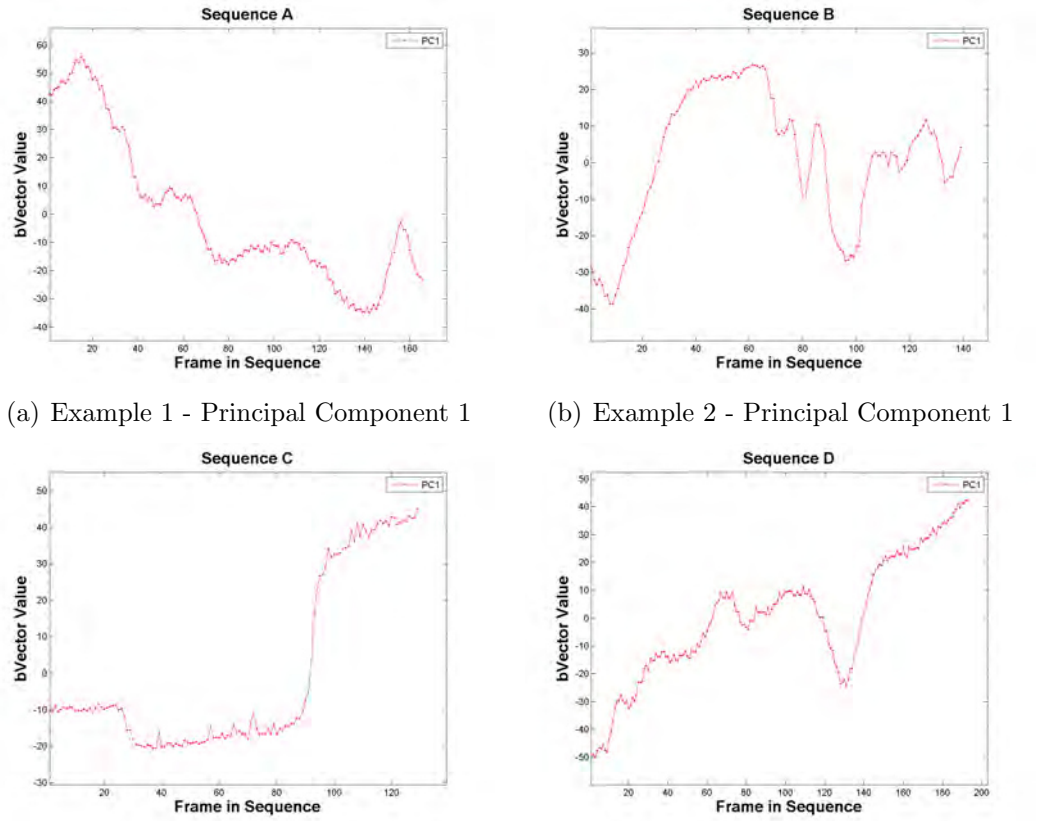
Figure 6.7: B-Spline Fit (30 Control Points)

component of the conversation data.

Equi-spaced control points allow sequences of different lengths to have the same number of polynomial sections, and they are spaced proportionally, but this does not ensure the polynomial sections represent similar actions. The coupled model requires the parts of the data to represent the same types of actions, for analysing interactions as well as for predicting new interaction characteristics. When we were developing our sequence representation approach we believed that this approach would not allow us to have comparable sections of the different sequences. In hindsight, the B-Spline fitting approach is the better choice for curve fitting and subsequently, producing the input for the coupled model, as the equi-spaced sections represent the data as well as, or better, than the single polynomial fit. We plan to implement this curve fitting approach in our future work.

6.5 Sequence Manipulation and Synthesis

Synthesising highly-realistic facial expression sequences has a variety of uses, from creating realistic digital characters to creating new stimuli for use in perceptual studies. As seen in Section 2.9.2 the level of realism of synthesised facial expressions



(a) Example 1 - Principal Component 1

(b) Example 2 - Principal Component 1

(c) Example 3 - Principal Component 1

(d) Example 4 - Principal Component 1

Figure 6.8: Conversation Data - Smile Interaction Sequence *bVector* Examples

has still to reach its peak [238, 249]. The other thing the literature has shown is the common lack of important temporal dynamics in expression sequence synthesis [79]. Both are items this work aims to address. First, the tracking and registration methods, along with the use of AAMs for modelling, help produce highly-realistic, synthesised expression frames. The realism of these synthesised individual frames (static frames) competes with some state-of-the-art work [38, 54, 80, 243], while arguably falling short when compared to other state-of-the-art work in the area of facial modelling [9, 10, 112, 116, 117] (although these works use, in part, hand created face meshes). The polynomial regression approach used for representing expression sequences allows for the modification and synthesis of entire sequences, while importantly retaining the characteristics of the expression. This is a novel contribution of this thesis, as no other work has done this using 4D inter-subject registered, modified, expression sequence data. The details of the model manipulation and expression synthesis methods are described in the following sections.

6.5.1 Sequence Manipulation

The main motivation for manipulating model parameters is for synthesising new, realistic facial expressions. These new sequences can be useful in a variety of psychology-based experiments in affective computing, autism research, deception detection, social group dynamics, and a variety of other areas. The higher the realism, the more likely the findings are based on real human perception and not hampered by the medium of conducting the experiment. For instance, while FACSGen [143] allows for great control over synthesising facial expressions, it is still far from anyone perceiving the synthesised character as a real person (Figure 6.9).

Figure 6.9: Facial expressions of anger, sadness, happiness, surprise, embarrassment, and pride (from left to right side) as synthesized in three-dimensional form by FACSGen 2.0 animation software [143]

As described in Chapter 6, we can represent smile sequences as single entities. It stands to reason then, that we can manipulate these entities to slightly change the characteristics of the expression sequences. The Psychology collaborators we work with are interested in manipulating both the duration and amplitude of the captured, validated smile sequences. It is for this reason we developed an approach for modifying the original sequences, while retaining the smile characteristics for both the expression and expression characteristics.

To modify the duration of the smile, we simply sampled more (slow down) or less (speed up) points along the polynomial curve. While shortening the duration could be achieved without the use of statistical models by removing video frames, lengthening the sequence does require the models for realistic, accurate synthesis. For instance, to double the length of a sequence, the models allows us to interpolate the model parameters between frame 1 and 2 (i.e. “frame 1.5”), and render out the newly

synthesised frame. This is powerful, because rather than simply slowing down playback (thus repeating frame 1 or 2, and so forth), we are able to synthesise the characteristics of what would occur between frame 1 and 2, for the subject in the sequence. This also means we can retain the natural expression dynamics of the subject in the sequence. An example of a smile sequence's duration being shortened by 30% and 70% and lengthened by 30% and 70% can be seen in Figure 6.10. This ability shows the usefulness of the AAM model for interpolating/extrapolating new data (Note: This capability does not come from the polynomial curve, but rather the AAM model).

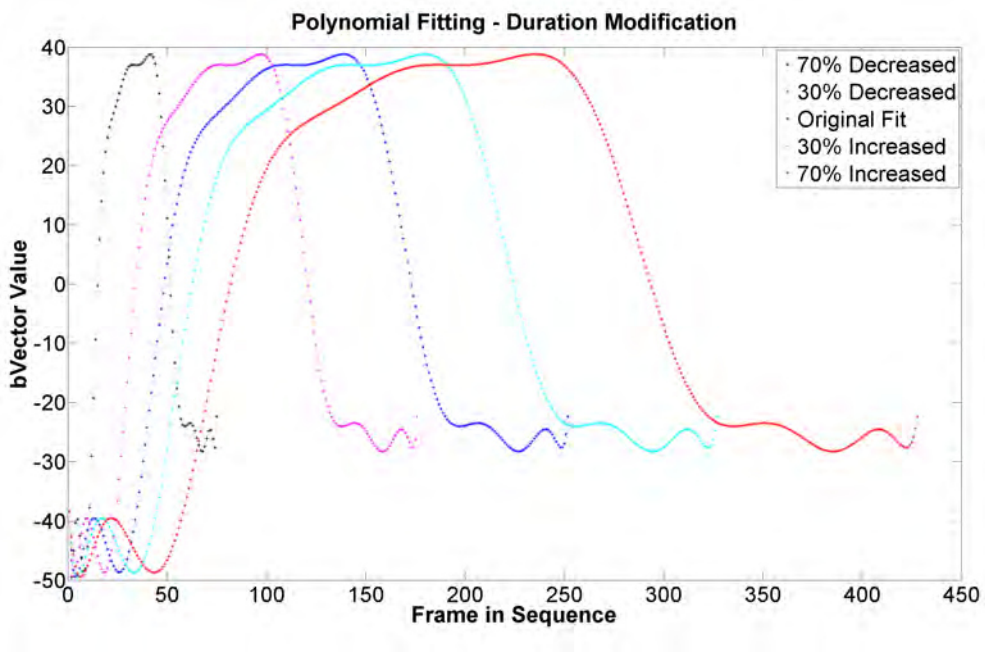


Figure 6.10: Four Duration Modification Levels and the Original Polynomial Fit

A similar approach is used for modifying the smile amplitude. However, unlike the duration modifications which could be produced by slowing down or speeding up a capture video, the smile amplitude modifications require a statistical model for creation. The polynomial curve that has been fitted to the sequence of b*Vector* values contains a polynomial equation. The coefficients of this equation are multiplied by a value to either decrease or increase the amplitude. New b*Vector* values are calculated for each frame in the sequence by using the frame number value for the variables of the polynomial equation. This visually results in the intensity of the smile sequences being modified. This approach is an implementation decision and one alternative

is to multiply the original *bVector* values before performing the polynomial fit. An example of a smile sequence's amplitude being decreased by 30% and 70% and increased by 30% and 70% can be seen in Figure 6.11. To ensure each sequence begins at the same point (neutral expression), each modified curve is shifted so that their start point is aligned with that of the original curve. Note that to simplify the curve examples, the original data has been inverted.

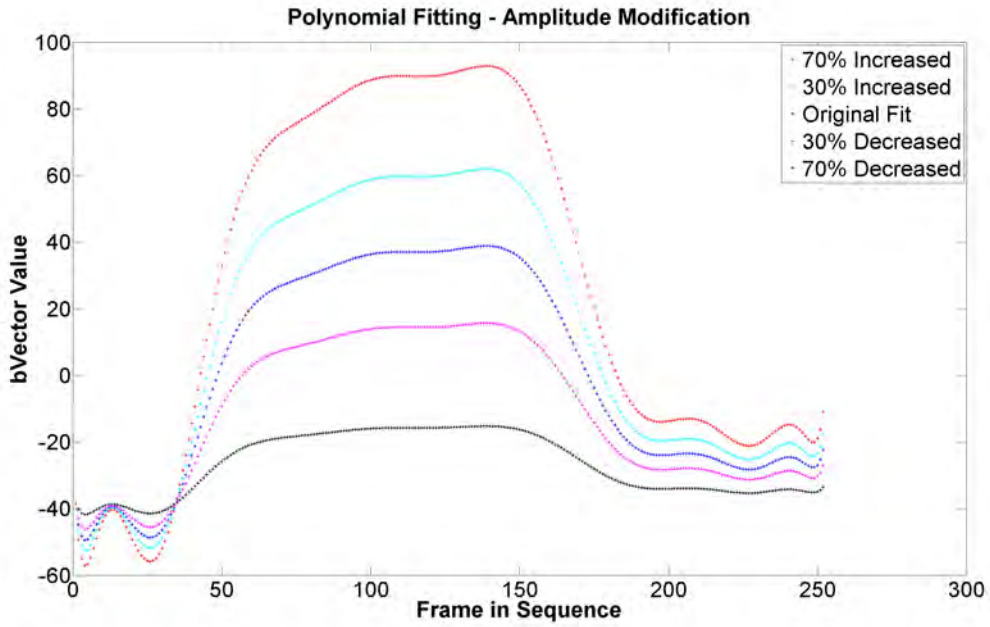


Figure 6.11: Four Amplitude Modification Levels and the Original Polynomial Fit

These modifications discussed in this chapter have been used for synthesising new expression sequences. This ability shows the usefulness of the AAM model for interpolating/extrapolating new data (Note: This capability does not come from the polynomial curve, but rather the AAM model). The polynomial curve becomes useful when we need to compare sequences, such is our application in Section 6.6. This process is described in Section 6.5.2.

6.5.2 Expression Synthesis

Our School of Psychology colleagues wish to explore the characteristics of trustworthy and genuine smiles (using the smile data captured in Section 3.4), by having observers evaluate different expression sequences and then using reverse correlation

to understand the mechanisms behind smile expressions perceived as trustworthy or genuine. While the specific stimuli for the experiment the Psychology collaborators will conduct has not been developed yet, sequences have been modified and synthesised for use in the experiments described in Chapter 7. This served not only as an evaluation of the pipeline and modelling process described in this thesis, but also as a pilot study of sorts for our Psychology collaborators.

The modification values used were empirically determined, so as to provide a slight deviation from the original and a larger deviation which would border on perceived realism. For both duration and amplitude these values were 70% decreased, 30% decreased, 100% (original), 30% increased, and 70% increased.

These modifications were performed for every Psych Database validated sequence, and for every backchannel sequence of the specified interactions. These interactions were frontchannel smile-backchannel smile interactions, where the backchannel responded within 2 seconds (arbitrarily determined). To identify these interactions the annotations described in Section 3.3.2 were used. There were 42 Psych Database smile sequences to modify and 24 Conversational Database (henceforth referred to as the *Convo Database*) backchannel sequences to modify. For the Convo Database videos, the unmodified frontchannel sequence was included in a side-by-side video to give context during the evaluation.

Figure 6.12 shows the peak smile frame for the 5 amplitude modification levels, for a single smile sequence. The interesting item to note is that this expression occurs at the same frame number; the duration is the same while the smile intensity has changed. In the video of this expression it is easy to see the preserved expression characteristics of the smile sequences for the subject.

Please see Chapter 7, specifically Section 7.5, to view results from a perceptual study of perceived realism, which used amplitude modified sequences.

Apart from the creation of 4D, highly-realistic, modified sequences, we also wanted to use the conversational interactions to build a coupled statistical model. This type of model allows for the analysis of conversational interactions as well as for

(a) 70% Decreased (b) 30% Decreased (c) Original (d) 30% Increased (e) 70% Increased

Figure 6.12: The peak expression frame for each modified sequence. This is the same frame number in all sequences.

the creation of new interaction data. Details of this novel contribution are in the following section, Section 6.6.

6.6 Coupled Models of Conversational Expression Interactions

The coupled statistical models of previous works, described in Section 2.9.3, all focus on actions that occur in the same time instance. The coupled models we build in this thesis are sequential in time, as one action influences another. To our knowledge, no work on 4D coupled models of conversational expressions currently exists.

A coupled statistical model is a simple approach that allows us the freedom to use it in various ways, such as for the analysis of conversational interactions or the synthesis of new interactions (such as predicting conversation interaction signals).

Again, as originally covered in Section 6.4.1, an approach like HMMs is insufficient as the the sequences do not follow each other (i.e. it is not a transition from frontchannel to backchannel). They occur simultaneously and one signal in an interaction has an effect on the other signal (and vice-versa). For this reason, we chose a simple concatenation approach for representing the conversational interactions.

For each frontchannel-backchannel interaction, the combined feature vectors for the frontchannel signal, an offset value for the interaction, and the combined feature vectors of the backchannel signal are concatenated to produce a coupled feature vector.

This modelling approach is similar, in spirit, to other joint statistical modelling approaches, such as Active Appearance Models. For AAMs [72, 73], shape and texture features are concatenated together and PCA is performed to find closer correlation. For our models, we concatenate lower-dimensional representations of conversational sequences (plus a time offset) and use imputation (Section 6.6.1) to discover and recover data. Table 6.1 shows an example of the coupled model.

In Table 6.1, "PC" stands for *Principal Component* and is the feature vector that is comprised of four types of information. Each "PC" cell contains thee following values: the polynomial coefficients, the number of sequence frames, the normalisation values, and the mean shift value. These values are concatenated to create a single feature vector for each PC. For example, a PC feature vector of a 4^{th} degree polynomial fit will have 9 values. The polynomial coefficients (5), the number of frames (1), the pre-polynomial fit normalisation values (2), and the pre-polynomial fit *mean shift* value (1).

As explained in Section 6.4.2, each sequence is first standard score normalised [141] and then shifted to the mean (for each PC's *bVectors*). This is an important step for retaining each sequence's characteristics, while also ensuring all of the sequences reside in the same polynomial space. The normalisation step is described by Equation 6.6. The mean and standard deviation of the current PC's *bVectors*, X, are the two "normalisation" values used in the feature vector.

$$\frac{X - \mu}{\sigma} \tag{6.6}$$

After this step, the mean is again computed for the PC's *bVectors*, X, and the *bVector* sequence is shifted to the mean (Equation 6.7).

$$X - \mu \tag{6.7}$$

The calculated mean value μ is the "mean shift" value used in the PC feature vector.

The polynomial coefficients of the PC feature vector are that of the fitted polynomial for the sequence (after the normalisation and mean shift steps), and the number of sequence frames value is obvious. The "Offset" cell represents the number of frames from the start of the (manually annotated) frontchannel signal to the start of the (manually annotated) backchannel signal.

Once the imputation is complete, these values are used for reversing the shift and normalisation steps, and the *bVector* values can be used to project out of the AAM. By projecting out of the AAM a 3D mesh frame can be produced.

	Sequences			
	1	2	3	4
FC	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3
	Offset	Offset	Offset	Offset
BC	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3	PC 1 PC 2 PC 3

Table 6.1: Example of the coupled model's feature vectors.

The combined feature vectors were calculated according to the process described in Section 6.4.2, and the *offset* value is the number of frames from the start of the frontchannel expression to the start of the reacting backchannel expression. The coupled model consists of these coupled feature vectors and now describes the interactions between a frontchannel signal and a backchannel signal, as a single feature vector. This model structure can be extended to represent multiple conversational interaction exchanges. For instance, a *frontchannel-backchannel-frontchannel* interaction (i.e. the speaker's signal caused a reaction by the listener, whose reaction in turn caused a direct response by the speaker) could be modelled by adding the offset value of the second exchange and concatenating its feature vector to the larger interaction feature vector. This model can be used for a variety of applications, including the analysis of interactions and synthesis of interaction data. In this work

we use it to predict backchannel responses to frontchannel signals (and vice versa), using k-Nearest Neighbour (kNN) imputation [32, 81]. This approach is described in Section 6.6.1.

6.6.1 Predicting FC/BC Signals

kNN imputation is a popular approach, especially when using sparse data sets, for replacing missing data [32, 67, 125, 221]. This approach replaces missing data with a weighted-mean of the k-nearest columns, where the weight is inversely proportional to the distance from those neighbours. The steps for using kNN to impute missing data can be found below.

1. Calculate the Euclidean distances for the sequences

$$E(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i \in C} (x_{a_i} - x_{b_i})^2} \tag{6.8}$$

where $E(\mathbf{a}, \mathbf{b})$ is the distance, d, between two of the sequences (columns) in the coupled model, x_{a_i} and x_{b_i} are the values (rows) in the sequences, and C is the set of sequence (column) rows with non-missing values in both $\mathbf{a}$ and $\mathbf{b}$.

2. Impute the missing values, val, for column c_m by calculating a weighted mean of the k-nearest columns k, where the weight is inversely proportional to the distance from those neighbours.

$$val = \frac{\sum_{i \in k} w_i k_i}{\sum_{i \in k} w_i} \tag{6.9}$$

where

$$w_i = \sum_{i \in k} \frac{1}{|d_i|} \tag{6.10}$$

In our coupled model, each column is the feature vector for a single interaction. Given the feature vector values of a frontchannel sequence and using a coupled model of conversational interactions, kNN imputation can be used to predict the values of a backchannel feature vector. The same is true for predicting frontchannel feature vector values. Table 6.2 provides example sequences whose missing values could be imputed using a coupled statistical model of interactions. The first column represents a frontchannel signal that is missing a corresponding backchannel signal. The second column represents the opposite situation. The third and fourth column are less likely scenarios for what we intend the coupled model to be used for when predicting values, but nonetheless the coupled model could help for imputing these values. Column three is missing the offset value for the interaction, and column four is sporadically missing parts of each sequence for the interaction.

	Sequence Values to be Imputed			
	1	2	3	4
FC	PC 1	Missing	PC 1	Missing
	PC 2	Missing	PC 2	PC 2
	PC 3	Missing	PC 3	Missing
	Offset	Offset	Missing	Offset
BC	Missing	PC 1	PC 1	PC 1
	Missing	PC 2	PC 2	Missing
	Missing	PC 3	PC 3	PC 3

Table 6.2: Example of the the types of interactions with missing data that the coupled model could help impute

This imputation process is a simple, but effective, approach for producing data for missing values. We perform it on our conversational data (see Section 7.6 for details) and use the predicted sequences in a classification experiment (Experiment 3) and a perceptual study (Experiment 4), in Chapter 7.

6.7 Summary

In this chapter we introduced two novel contributions of this thesis. The first is our technique for creating 4D, highly-realistic, modified expression sequences, which competes with state-of-the-art approaches [38, 54, 80, 143, 238, 243, 249]. These sequences have been used in classification experiments for differentiating types of smiles (Frontchannel/Backchannel, Duchenne/Non-Duchenne) (Section 7.3), as well as been evaluated by human observers for both expression realism and image quality (Section 7.5).

The second novel contribution is our coupled statistical modelling approach for modelling and synthesising conversational interaction. Using the single-entity expression sequences from our conversational data, we developed a coupled model approach that uses the feature vectors of conversational interactions. This approach allows for the analysis of conversational interactions and the synthesis of sequences. We use a kNN approach for predicting missing values of an interaction, given our coupled model (e.g. given a frontchannel signal, we can predict the values of the backchannel sequence). Evaluation of our method is performed in a classification experiment (Section 7.7) in which we differentiate predicted frontchannel and backchannel sequences. We also evaluate our approach in a perceptual study where human observers rate the similarity between original and predicted backchannel sequences (Section 7.8).

Full details about the stimuli creation and related experiments are provided in the following Chapter (Chapter 7).

Chapter 7

Experiments and Evaluation

7.1 Introduction

In order to evaluate the tracking, inter-subject registration, modelling, and synthesising methods used in this work, three experiments were devised. These experiments were conducted using data from two separate datasets. The first is the Duchenne/Non-Duchenne (D/ND) dataset described in Section 3.4. This is referred to below as the *Psych Data*. The second database is the conversational expression database, which is the main data used in this work and described in detail in Chapter 3. It is referred to below as the *Convo Data*. To simplify this chapter, the introduction for each experiment will describe the experiment methodology both databases (Psych and Convo) followed. Subsections will describe items specific to each database, as well as the results.

In Experiment 1 (Section 7.3), Duchenne smile sequences were differentiated from Non-Duchenne smile sequences for the Psych Experiments; and frontchannel smile sequences were differentiated from backchannel expression smiles for the Convo Experiments. These classification experiments were conducted to help us evaluate that our in-lab tracking (Section 5.5.4) and inter-subject registration (Section 5.6) approaches were sound, and to demonstrate that both Duchenne/Non-Duchenne and frontchannel/backchannel signals have intrinsic, separable properties. As previous

research has shown that there are separable properties between Duchenne and Non-Duchenne smiles [231], it was important to determine that the tracking and registration approaches used did not warp the data in such a way that it lost the characteristics which define it, such as the minute differences between Duchenne and Non-Duchenne smiles.

In Experiment 2 (Section 7.5) we built 4D models of smile sequences (Psych) and of backchannel sequences (Convo), manipulated their amplitudes, and synthesised the manipulated sequences. These sequences were then evaluated by human observers to measure the effect of the modification on perceptions of realism, both for facial expression and image quality. Results of this experiment show that the models we built are perceived as realistic and believable, even after being modified. While the first experiment focuses on the tracking and registration methods, the results of this experiment help us evaluate the modelling and synthesising approaches used. By modifying the sequences themselves, highly-realistic expression sequences could be synthesised while retaining the expression characteristics of the subject's facial expression.

In Experiment 3 (Section 7.7) we built a 4D coupled statistical model of smile interactions from the annotated conversational database. We used the imputation approach described in Section 6.6.1 with our coupled model to predict frontchannel feature vectors given their corresponding backchannel feature vectors. The reverse was done for predicting backchannel feature vectors. The classification training set was comprised of the original frontchannel and backchannel feature vectors and the test set was comprised of predicted frontchannel and backchannel sequence feature vectors. These predicted sequences were classified as either frontchannel and backchannel sequences. Experiment 1 (Section 7.3) showed that FC and BC signals could be differentiated. Experiment 3 was performed to observe if the predicted FC and BC sequences from our coupled model could also be differentiated with high accuracy. This assisted us in evaluating our coupled model and its imputation abilities before we synthesised the predicted output to be used in the perceptual study in Experiment 4.

In Experiment 4 (Section 7.8), we built 4D coupled statistical models of smiles (Psych) and smiles reciprocated during conversations (Convo). We used these models to predict, for each sequence, the characteristics of similar smiles (Pysch) and backchannel reactions to frontchannel signals (Convo). Human observers viewed the original and predicted sequences. Their task was to rate the extent to which a predicted sequence was similar to the original. This experiment explored the ability to use coupled statistical models to predict characteristics of a sequence/interaction based on a model of similar sequences/interactions. Ratings of high similarity would be supportive of this approach and allow for future experiments that would utilise coupled models of conversational expressions.

In the case of the conversational expression interactions, no other work has previously explored the use of coupled statistical models for analysing, predicting, and synthesising new 3D sequences.

All result analysis for Experiment 2 and Experiment 4 were performed in collaboration with our research colleague Dr. Magdalena Rychlowska, a postdoctoral researcher in the Cardiff School of Psychology.

Table 7.1 shows details about the sequence data used for these experiments (D/ND is Duchenne/Non-Duchenne).

7.2 Tracking Parameters and Registration Mask

The data used in these experiments were tracked using the 28-point landmarking scheme described in Section 5.5.1. Only texture information was used for tracking and the resulting tracked points were sufficiently accurate. Curvature (shape) information was not used because it increased the run-time to a level that was undesirable for the amount of data we needed to process in a limited amount of time. While the tracking approach has the ability to use both texture and shape information, we implemented only the texture-based tracking approach for these experiments for the reason listed above. The *linear combination* option was used for the similarity search,

Frontchannel Sequences	**Data**
Number of Sequences:	22
Number of Frames:	3,347
Mean:	2.5356 Secs
Standard Deviation:	1.1979 Secs
Minimum Length:	0.8167 Secs
Maximum Length:	5.2 Secs
Backchannel Sequences	
Number of Sequences:	22
Number of Frames:	2,249
Mean:	1.7038 Secs
Standard Deviation:	0.6476 Secs
Minimum Length:	0.8667 Secs
Maximum Length:	2.98 Secs
D/ND Smile Sequences	
Number of Sequences:	42
Number of Frames:	12,659
Mean:	5.0234 Secs
Standard Deviation:	1.5947 Secs
Minimum Length:	3.0167 Secs
Maximum Length:	9.0333 Secs

Table 7.1: Experiment Sequence Details

with a coefficient of 2 (i.e. quadratically decreasing weights), and a buffer size of 10 (i.e. how many frames were part of the combination). 8 rings made up the local neighbourhood. Details about the tracking method can be found in Section 5.5.4.

The *registration mask* used as the reference mesh in the registration process was created from a frame of Subject A, one of the male subjects (Figure 7.1. He was a good candidate for being the registration mask subject partially because he had nice mesh topologies that required the least amount of cleaning, but also because this subject was in the majority of conversation interaction sequences. Details about the registration method can be found in Section 5.6.

(a) Textured (b) Textured Wireframe

Figure 7.1: Registration Mask

7.3 Experiment 1 - Classification

In this experiment we attempted to differentiate smile sequences (Duchenne/Non-Duchenne and Frontchannel/Backchannel), using 3D AAMs for feature extraction, polynomial fitting for 4D sequence representation, and Support Vector Machines (SVMs) for classifying the 4D sequences.

For each subject, $\text{Sub}_{\text{target}}$, a 3D AAM was built using all sequence frames from every other subject, $\text{Sub}_{\text{others}}$. 95% of the eigenenergy was kept. For each sequence, *bVectors* were calculated by projecting every frame into the AAM.

An n^{th} degree polynomial fit (See 'Poly Degree' in Table 7.2 and Table 7.4) was performed on each sequence of *bVectors*. This approach allowed for a 4D representation of 3D discrete data. A grid search was performed to empirically find an appropriate polynomial degree and number of principal components to use for fitting, for each $\text{Sub}_{\text{target}}$ AAM model.

The polynomial coefficients were used as input into a Support Vector Machine (SVM) classifier, *libSVM* [65], where $\text{Sub}_{\text{others}}$ sequences comprised the training set, and

Sub_{target} sequences comprised the testing set. A ν-SVM with a Gaussian RBF kernel was used, and a grid search was performed for parameter optimisation, as suggested in [128, 207]. As stated above, these steps were performed for each subject, so as to provide a fully-generalised approach to classification.

For classification accuracy, Area Under the ROC Curve (AUC) was chosen as the performance metric because it has been shown to be more reliable and contain more preferable properties than raw classification accuracy, as described in [29, 52, 149].

7.3.1 Psych Results

The average classification accuracy for all four subjects was 98.75% AUC. Details of the scores, polynomial degrees, number of principal components used, SVM parameters, and whether polynomial fitting mu values and the number of frames were part of the feature vector used, can be seen in Table 7.2. A classification confusion matrix for each subject can be seen in Table 7.3.

Psych Classification Results				
	Subject A	**Subject B**	**Subject C**	**Subject D**
Num. of Sequences	9	9	12	12
AUC Score	100%	100%	95%	100%
Raw Score	100%	100%	91.67%	100%
Poly Degrees	4	4	6	7
Num of PCs	2	2	4	4
SVM Cost	0.000977	0.000977	0.000977	0.000977
SVM Nu Value	0.185	0.2717	0.1404	0.1858
Poly Mu Values?	No	No	No	No
Num. of Frames?	Yes	Yes	Yes	No

Table 7.2: Psych Classification - Details for Each Subject

		Sub. A			**Sub. B**			**Sub. C**			**Sub. D**	
Actual		D	ND		D	ND		D	ND		D	ND
Predicted	D	7	0		7	0		9	0		5	0
	ND	0	2		0	2		1	2		0	7

Table 7.3: Psych Classification - Confusion Matrix for Each Subject

7.3.2 Convo Results

The average classification accuracy for all four subjects was 97.54% AUC. Details of the scores, polynomial degrees, number of principal components used, SVM parameters, and whether polynomial fitting mu values and the number of frames were part of the feature vector used, can be seen in Table 7.4. A classification confusion matrix for each subject can be seen in Table 7.5 (where N and ND are Duchenne and Non-Duchenne, respectively).

Convo Classification Results				
	Subject A	**Subject B**	**Subject C**	**Subject D**
Num. of Sequences	22	5	7	10
AUC Score	96.43%	100%	100%	93.75%
Raw Score	95.45%	100%	100%	90%
Poly Degree	4	5	5	3
Num of PCs	4	3	4	3
SVM Cost	0.000977	0.000977	0.000977	0.000977
SVM Nu Value	0.32	0.5697	0.4081	0.10
Poly Mu Values?	Yes	No	No	No
Num. of Frames?	Yes	No	Yes	Yes

Table 7.4: Convo Classification - Details for Each Subject

		Sub. A			**Sub. B**			**Sub. C**			**Sub. D**	
Actual		FC	BC		FC	BC		FC	BC		FC	BC
Predicted	FC	13	0		3	0		3	0		2	1
	BC	1	8		0	2		0	4		0	7

Table 7.5: Convo Classification - Confusion Matrix for Each Subject

7.3.3 Results Discussion

Experiment 1 was able to validate that the classes of each smile sequences classified contain characteristics which allow them to be differentiated. For the Duchenne/Non-Duchenne data this was most likely the area around the eyes, given this characteristic is what separates Duchenne smile sequences from Non-Duchenne sequences. For the Frontchannel/Backchannel smile sequences, the characteristic differentiating the two types of sequences is most likely the vertical movement of the mouth of the speaker (frontchannel signal). This belief stems from a visual examination of

the smile interactions and their corresponding trajectories, as well as the modes of variation for each class of sequences. However, given the complexity of the data and the multiple steps involved (tracking, registration, model building, polynomial fitting, and then classification), more experiments would need to be conducted to confirm that those are the properties that have caused the positive classification results. Overall, we were pleased with the results of these initial tests. It was extremely important to confirm the quality of these approaches before undertaking the modelling and synthesis steps described in Sections 7.5, 7.7, and 7.8.

7.4 Pilot Study for Experiments 2 & 4

We conducted a pilot study for Experiment 2 and Experiment 4 involving individuals who were not eligible to take part in the actual study (e.g., supervisors, collaborators, etc.). This was done to receive feedback on the structure of the experiments as well as to work out any technical issues, before launching the website to a much larger audience.

For the 42 Psych sequences there were 5 modification levels for duration and 5 modification levels for amplitude (Experiment 2), as well as 2 videos (original and predicted) to evaluate (Experiment 4), for a total of 504 videos to evaluate. The sequences were split across four groups in a balanced manner (similar number of specific subjects and smile types) so that each participant only had to evaluate 126 videos each, and so that all modified (Section 7.5) and predicted (Section 7.8) sequences could be evaluated.

A similar approach was used for the Convo sequences. The 24 interaction sequences consisted of an unmodified frontchannel video and a backchannel video with one of the five modification levels for duration and amplitude (Experiment 2). There were also 2 videos (original backchannel and predicted backchannel) to evaluate (Experiment 4). The original *offset* value (delay between the frontchannel's expression and the backchannel's response) was always preserved. These modifications resulted in a

total of 288 interactions to evaluate. The sequences were split across two groups in a balanced manner (similar number of specific subjects and smile types) so that each participant only had to evaluate 144 videos each.

Two main points of feedback were received. The first had to do with the instructions and examples. This feedback was used to make slight modifications in the wording and display of information. The second piece of feedback required a larger change. Many felt the experiment was too long (roughly 30 minutes) and that the sequences they were evaluating were far too similar. Some even commented that they believed they were evaluating the same exact sequence multiple times. While this was not the case, it was obvious that the number of videos to evaluate, along with the sequences having only slight differences, was becoming tedious and affecting the participant's ability to evaluate the sequences. Given that the participants in the real study would be volunteers, who are less likely to finish a long, tedious experiment, changes needed to be made.

It was important to us that all sequences were evaluated. Given the relatively small number of sequences in each database, using all of the sequences gives us a larger, and arguably more diverse data for participants to evaluate. It also became clear, through feedback, that the variation to the duration did not produce very meaningful results. The variation levels were either too subtle for most to realise, or too extreme that individuals thought the videos were freezing (in the case of longer duration). As well, longer duration sequences would increase the experiment time greatly (e.g., 5 second sequence was now 8.5 seconds), without providing very meaningful results (as explained previously). The modification in amplitude values was much more interesting for the purposes of this work. Therefore, the decision was made to remove all duration-modified videos from the experiment. This left 5 modification levels for amplitude (Experiment 2), and 2 videos for the original/predicted experiment (Experiment 4).

The same number of groups for each database type was kept. The new numbers were as follows: 294 Psych videos or 77/70 for each group (2 x 77, 2 x 70); 144

Convo videos or 77 per group (2 x 77). The changes reduced the time it took to complete the experiment from approximately 30 minutes to approximately 15 minutes. By-and-large, the comments from the actual research cohort were positive, with regards to the instruction layout and the length of the experiment.

7.5 Experiment 2 - Modified Expressions

Over the span of one-week, 100 volunteers were recruited for a web-based perceptual study (160 registered, only 100 fully completed the study). Individuals were assigned to their group (described in Section 7.4) on a rotating basis, in this order: *Convo - Group 1*, *Psych - Group 1*, *Convo - Group 2*, *Psych - Group 2*, *Psych - Group 3*, *Psych - Group 4*. Only participants who completed the study were used in the evaluation, to ensure the comparisons of ratings could be done for all sequences, for all participants. The slightly unbalanced groups reflect those individuals who began but did not finish the experiment. The participants per group are shown in Table 7.6.

Convo Smiles	**# of Participants**
Group 1:	18
Group 2:	10
Psych Smiles	
Group 1:	18
Group 2:	16
Group 3:	21
Group 4:	17

Table 7.6: Group Splits - Convo and Psych

This web-based study allowed English-speaking individuals from across the world to participate. These individuals were recruited through contacts of those in the pilot study, as well as social media posts by the Cardiff School of Computer Science. This provided a wide-variety of backgrounds of participants. Table 7.7 show the breakdown of gender and age range and Table 7.8 shows the breakdown of nationalities, for 71 Psych* and 28 Convo Participants (*all demographics were self-reported. One Psych participant opted-out of supplying this information).

	Psych	**Convo**
Gender		
Male	45	17
Female	26	11
Age Range		
18-25	33	12
26-33	16	5
34-40	3	4
41-48	5	3
49-60	10	4
60+	4	0

Table 7.7: Gender and Age Range Demographics

Experiment 2 focused on the realism of synthesised smile sequences (Duchenne/Non-Dcuhenne and Backchannel), for both facial expression and rendered sequence image quality. 3D AAMs were built for every smile sequence (95% eigenenergy kept). The fitting process described in Section 6.4.2 was performed using a 14^{th} degree polynomial for 3 principal components. These 3 PCs represented approximately 90% and 85% of the remaining model energy, for the Psych database and Convo database, respectively. After observing the output from various combinations of PCs, we chose to use the top three PCs because they provided the best balance between the number of PCs used and capturing the data and variability required for our experiments. That is, by reducing the number of PCs we are able to reduce processing time, while still retaining the quality and variation of data we require for our work. For this approach, since each sequence is completely independent, no normalisation or shifting step is performed. In addition to the independent nature, and since the goal is the creation of realistic stimuli, over-fitting is not a concern. This subject-sequence-specific approach allows us to better focus on the variations of a single person and their expressions [122]. Each smile sequence's polynomial fit was modified using four amplitude values: 70% decreased, 30% decreased, 30% increased, and 70% increased. From these new polynomial curves, *bVector* values for each PC were calculated for each modification type, as described in Section 6.4.2 and shown in Figure 6.6. The original amplitude sequence was produced using the original polynomial coefficients and acted as a ground-truth of sorts. Figure 6.12 shows

	Psych	Convo
Nationality		
Great Britain	28	10
United States	21	5
Germany	7	1
Poland	7	2
China	1	3
Canada	1	1
Italy	1	0
Bulgaria	1	0
Pakistan	1	0
Estonia	0	1
Brunei	1	0
New Zealand	0	1
Czech Republic	0	1
Greece	0	1
Taiwan	0	1
France	1	0
Switzerland	1	0
Nigeria	0	1

Table 7.8: Nationality Demographics

examples of the same peak frame for each modified sequence. For the conversational data, the frontchannel sequence uses its original polynomial fitted values. It is worth noting that while the amplitude of the backchannel smile expressions were modified, the expression dynamics of the individual were preserved.

These videos of modified smiles and smile interactions were used in a subsequent experiment, in which 100 participants viewed the video sequences and evaluated the realism of the smile sequence (or backchannel sequence in the case of the conversational data) for both for expression and for image quality, using a 4-point Likert-type scale ranging from (1) Not at all realistic to (4) Highly realistic.

Before starting the task, participants were shown examples of realistic and unrealistic example sequences. These example sequences were not included in the ones evaluated by the participants during the task. The sequences were shown to participants in a random order, using the efficient and unbiased *Durstenfeld-Knuth* shuffling algorithm [100, 140].

7.5.1 Psych Results

For the Duchenne/Non-Duchenne data, the participants evaluated a single smile sequence each time. 72 participants (46 male, 26 female) watched the video sequences and evaluated the realism of the facial expression and of the video itself, using 4-point Likert-type scales ranging from (1) Not at all realistic) to (4) Highly realistic. Each participant rated 5 versions of each sequence – the original and the four modified versions – for a total of 3770 trials.

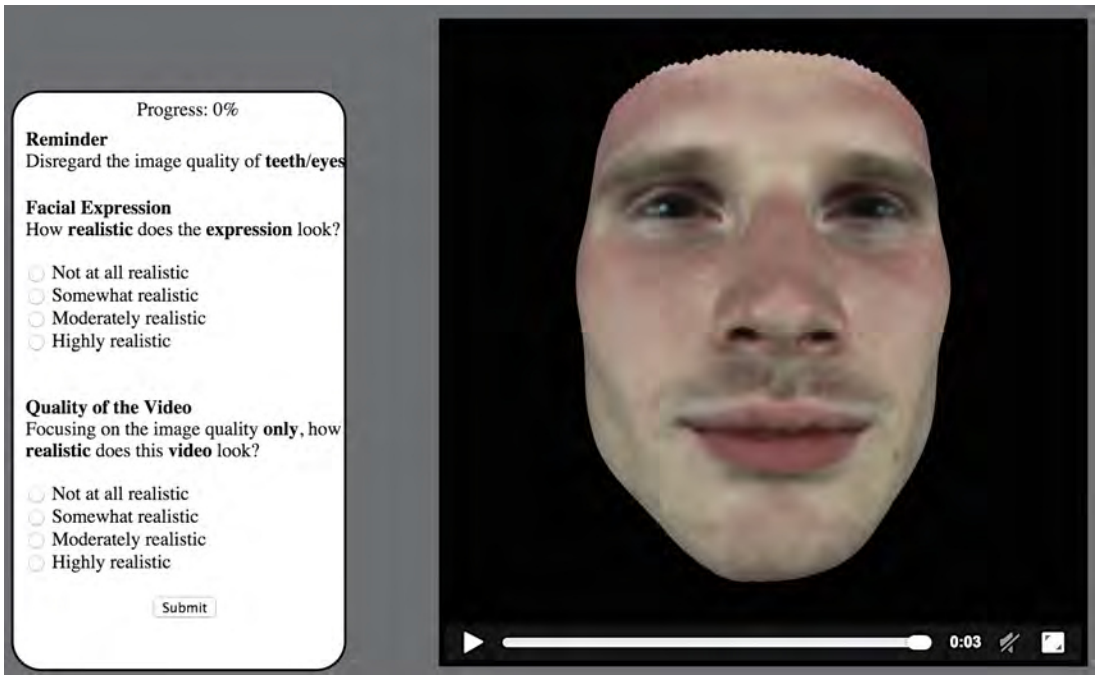


Figure 7.2: Screenshot: Psych Group - Modified Smile Sequence

Two trials from two participants were discarded from the analysis due to web-based input errors which resulted in missing data. Ratings of facial expressions and of image quality were both significantly affected by the manipulation of amplitude, $F(4, 284) = 24.74$, $p < .001$ and $F(4, 284) = 148.81$, $p < .001$, respectively. Perceived realism of facial expressions displayed in the original videos and in the videos using lower levels of amplitude was significantly higher than the scale midpoint (all $t's > 8$, $p's < .01$, Bonferroni corrected). These expressions were thus perceived as highly realistic. The versions using higher amplitudes were perceived as less realistic and were not higher than the scale midpoint (130%: $t(71) = -.38$, $p > .1$, 170%: $t(71) = -14.34$, $p < .01$). Table 7.9 shows the average rating and standard deviation

for each modification level, for expression realism.

Modification Level	Avg. Rating	S.D.
30%	3.18	.45
70%	3.34	.50
100%	3.03	.50
130%	2.47	.56
170%	1.63	.51

Table 7.9: Psych Data - Modified - Expression Realism

A similar pattern was observed for the perceived image quality of video sequences: the original video and the two versions using decreased amplitude were rated as highly realistic (all $t's > 9$, all $p's < .01$). This time, however, the version using 130% amplitude level was also rated as highly realistic ($t(71) = 3.19$, $p = .01$). Ratings of the modification using the highest amplitudes were again lower than the scale midpoint ($t(71) = -5.72$, $p < .01$.) Table 7.10 shows the average rating and standard deviation for each modification level, for image quality.

Modification Level	Avg. Rating	S.D.
30%	3.22	.54
70%	3.22	.55
100%	3.05	.51
130%	2.71	.56
170%	2.06	.64

Table 7.10: Psych Data - Modified - Image Quality

Figure 7.3 shows the values plotted for each modification level, for expression realism ratings and image quality ratings. The red line in the figure represents the rating scale's mid-point. Thus, any values above this line represent stimuli that were perceived as realistic.

For these results the effects of modification level varied as a function of the questionnaire group (marginally significantly for *expression realism*, $F(12, 272) = 1.65$, $p = .08$, significantly for *image quality*, $F(12, 272) = 3.59$, $p < .001$), results, per group, are reported below.

There were 72 participants across four groups. Figure 7.4 shows the average rating for *expression realism*, per group, for each modification level. The corresponding table (Table 7.11) provides the actual values.

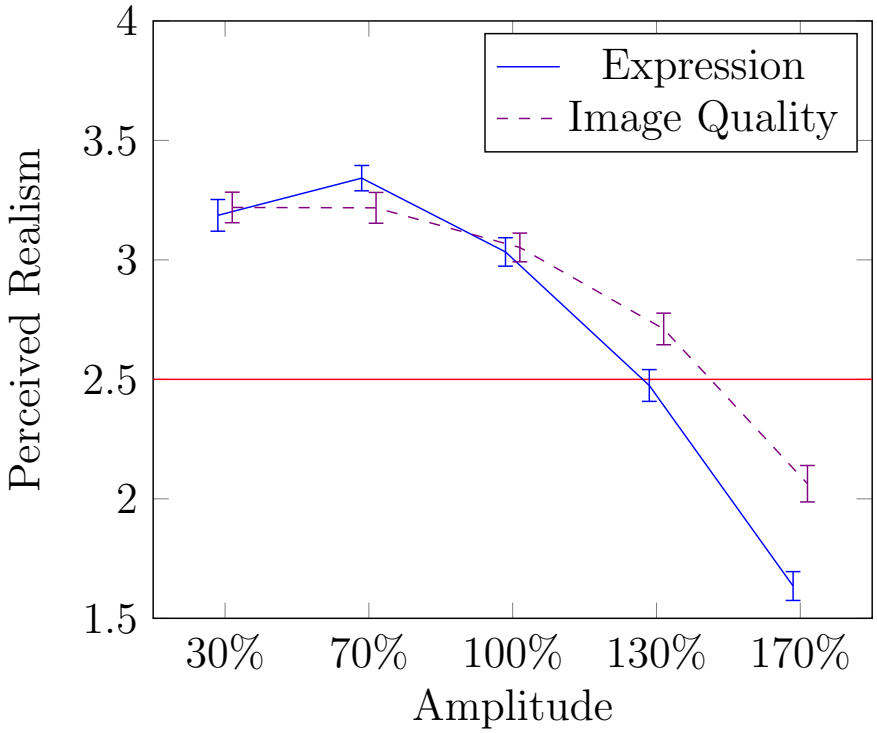


Figure 7.3: Experiment 2 Psych Results

	30%	**70%**	**100%**	**130%**	**170%**
Group 1	3.0444	3.3316	3.0229	2.4049	1.5535
Group 2	3.2159	3.2599	2.8343	2.1548	1.3977
Group 3	3.2836	3.3056	3.0827	2.6922	1.8675
Group 4	3.1889	3.4754	3.1706	2.5797	1.6578
Combined Average	**3.1864**	**3.3421**	**3.0333**	**2.4744**	**1.6351**

Table 7.11: Expression Realism - Average rating per group for each modification level

The tables that make up Table 7.5 give the minimum, maximum, mean, and standard deviation results, per group, for *expression realism*. These results have been averaged across all the participant responses.

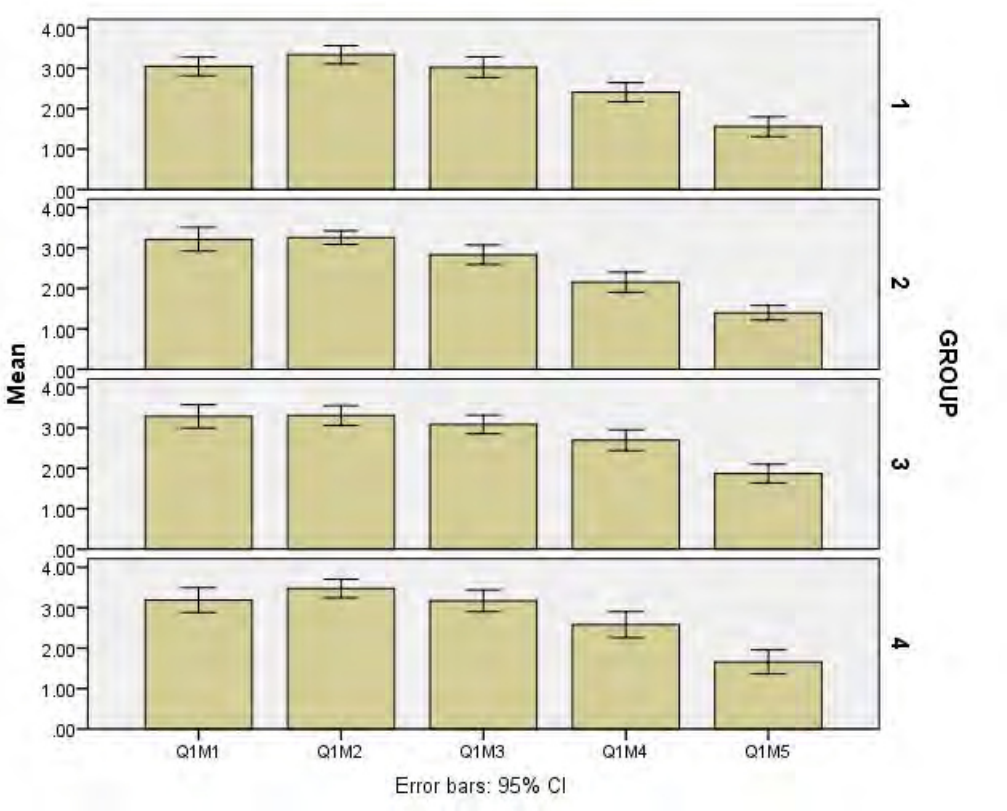


Figure 7.4: Expression Realism - Rating per Modification Level and Group

Modification	Min.	Max.	Mean	S.D.
30%	2.27	3.70	3.0444	.46322
70%	2.27	3.91	3.3316	.45394
100%	2.00	3.91	3.0229	.51528
130%	1.50	3.08	2.4049	.47162
170%	1.00	2.73	1.5535	.49640

(a) Group 1 - 18 Participants

Modification	Min.	Max.	Mean	S.D.
30%	2.00	4.00	3.2159	.55240
70%	2.91	3.82	3.2599	.31707
100%	2.18	3.73	2.8343	.44968
130%	1.55	3.18	2.1548	.46512
170%	1.00	2.36	1.3977	.34157

(b) Group 2 - 16 Participants

Modification	Min.	Max.	Mean	S.D.
30%	2.11	4.00	3.2836	.62866
70%	1.90	4.00	3.3056	.52812
100%	2.09	3.73	3.0827	.50505
130%	1.70	3.50	2.6922	.56629
170%	1.10	3.10	1.8675	.50329

(c) Group 3 - 21 Participants

Modification	Min.	Max.	Mean	S.D.
30%	1.67	4.00	3.1889	.60041
70%	2.30	4.00	3.4754	.44166
100%	1.90	3.80	2.1706	.52719
130%	1.50	3.50	2.5797	.62566
170%	1.00	2.82	1.6578	.58263

(d) Group 4 - 17 Participants

Figure 7.6 shows the average rating for *image quality*, per group, for each modification level. The corresponding table (Table 7.12) provides the actual values.

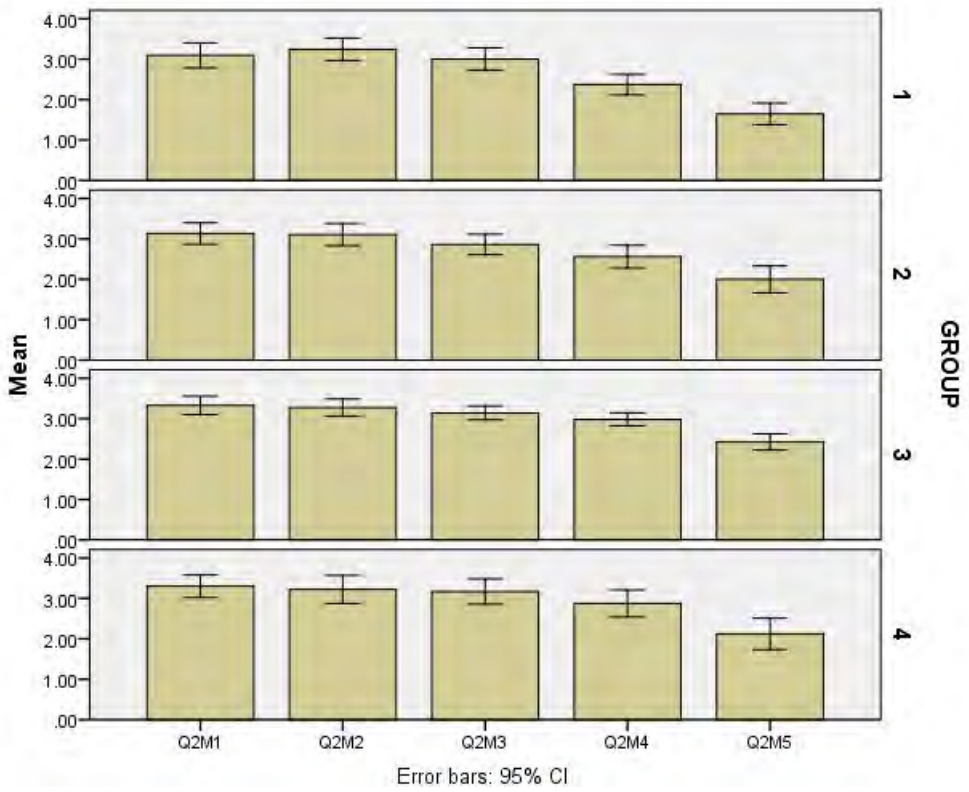


Figure 7.6: Image Quality - Rating per Modification Level and Group

	30%	**70%**	**100%**	**130%**	**170%**
Group 1	3.0934	3.2441	3.0037	2.3716	1.6452
Group 2	3.1335	3.1070	2.8627	2.5620	2.0000
Group 3	3.3275	3.2749	3.1416	2.9827	2.4229
Group 4	3.3007	3.2235	3.1706	2.8749	2.1214
Combined Average	**3.2195**	**3.2178**	**3.0520**	**2.7110**	**2.0633**

Table 7.12: Image Quality - Average rating per group for each modification level

The tables that make up Table 7.7 give the minimum, maximum, mean, and standard deviation results, per group, for *image quality*. These results have been averaged across all the participant responses.

Modification	Min.	Max.	Mean	S.D.
30%	2.00	4.00	3.0934	.62736
70%	2.27	3.92	3.2441	.56125
100%	1.92	3.75	3.0037	.55347
130%	1.36	3.08	2.3716	.51405
170%	1.00	3.00	1.6452	.54026

(a) Group 1 - 18 Participants

Modification	Min.	Max.	Mean	S.D.
30%	2.20	3.91	3.1335	.49946
70%	2.45	3.82	3.1070	.51525
100%	2.09	3.67	2.8627	.47560
130%	1.64	3.45	2.5620	.52909
170%	1.27	3.55	2.0000	.62721

(b) Group 2 - 16 Participants

Modification	Min.	Max.	Mean	S.D.
30%	2.33	4.00	3.3275	.49687
70%	2.36	3.80	3.2749	.46348
100%	2.40	3.90	3.1416	.37897
130%	2.30	3.60	2.9827	.35270
170%	1.64	3.00	2.4229	.43842

(c) Group 3 - 21 Participants

Modification	Min.	Max.	Mean	S.D.
30%	2.33	4.00	3.3007	.54636
70%	1.70	4.00	3.2235	.67595
100%	2.00	4.00	3.1706	.60968
130%	1.60	3.91	2.8749	.64833
170%	1.10	3.60	2.1214	.75338

(d) Group 4 - 17 Participants

Figure 7.7: Image Quality Result Details (per group)

7.5.2 Convo Results

Each sequence evaluated showed the original frontchannel sequence and one of the five amplitude-modified sequences, with the offset preserved. Each participant thus rated 5 versions of 11 sequences - the original and the four modified versions - for a total of 1540 trials (55 trials per person).

Four trials from three participants were discarded from the analysis due to web-based input errors which resulted in missing data. Both expression realism and image quality ratings were significantly affected by the manipulation of amplitude, $F(4, 108) = 24.69$, $p < .001$ and $F(4, 108) = 26.04$, $p < .001$, respectively. Perceived

Figure 7.8: Screenshot: Convo Group - Modified Smile Sequence

realism of facial expressions displayed in the original videos, in the videos using lower levels of amplitude, and in the videos using the 130% amplitude level was significantly higher than the scale midpoint (2.5, all $t's > 3.5$, $p's < .01$, Bonferroni corrected). These versions were therefore perceived as highly realistic. The high-amplitude (170%) level, however, was perceived as less realistic and not significantly higher than the scale midpoint ($t(27) = -1.19$, $p > .1$). Table 7.13 shows the average rating and standard deviation for expression realism, for each modification level.

Modification Level	Avg. Rating	S.D.
30%	2.95	.62
70%	3.14	.49
100%	3.00	.43
130%	2.83	.41
170%	2.40	.43

Table 7.13: Convo Data - Modified - Expression Realism

An identical pattern was observed for the ratings of image quality (Table 7.14).

Modification Level	Avg. Rating	S.D.
30%	2.98	.53
70%	3.04	.44
100%	2.99	.45
130%	2.84	.44
170%	2.43	.52

Table 7.14: Convo Data - Modified - Image Quality Realism

Figure 7.9 shows the values plotted for each modification level, for expression realism ratings and image quality ratings. The red line in the figure represents the rating

scale's mid-point. Thus, any values above this line represent stimuli that were perceived as realistic.

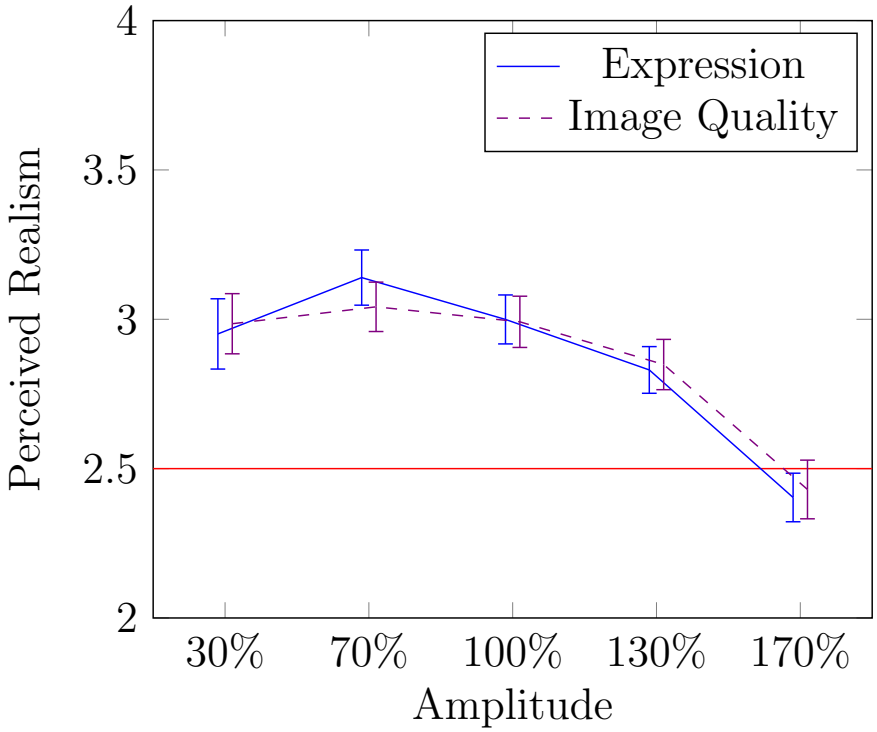


Figure 7.9: Experiment 2 Convo Results

The effects of modification level did not significantly vary as a function of the questionnaire group, therefore no per group reporting is necessary.

7.5.3 Results Discussion

The results of this experiment were overwhelmingly positive. One of the main goals of this work is to produce highly-realistic, synthesised 3D sequences of facial expressions and expression interactions. The results from these perceptual studies show that our methods were able to produce synthesised sequences that were perceived as highly realistic, both for expression realism and image quality. These sequences also retained the subject's expression characteristics, which is an important detail that should not be overlooked when creating synthesised expression sequences. The observed decline in perceived realism of smiles as the amplitude/intensity increased (to extreme levels) was not unexpected and helped show us the level to which we could modify the sequences. Having the ability to modify expression sequences and

retain a decent level of realism is important for our future work with our psychology collaborators. Their goal is to modify parts of expression sequences and observe any changes in the perception of these expressions based on the change of expression characteristics. The results of this experiment support our ability to produce realistic stimuli for their future perceptual studies. Having shown that we can model, modify, and synthesise realistic expression sequences separately, we moved on to joining them in a coupled statistical model, which is used in this work for predicting smile expressions, specifically, backchannel expressions given frontchannel stimuli.

7.6 Coupled Model Experiments

Experiment 3 and Experiment 4 focused on evaluating our coupled statistical modelling approach, specifically the similarity of original and predicted backchannel smile expressions. We built an AAM which contained all sequence frames for every subject (95% eigenenergy kept). The fitting process described in Section 6.4.2 was performed for three principal components. After observing the output from various combinations of PCs, we chose to use the top three PCs because they provided the best balance between the number of PCs used and capturing the data and variability required for our experiments. That is, by reducing the number of PCs we are able to reduce processing time, while still retaining the quality and variation of data we require for our work. The polynomial degree chosen was empirically determined, so as to avoid over-fitting. The coupled model was built according to the steps described in Section 6.6, and the imputation step was performed for the polynomial coefficients using an empirically determined k value of 15 for kNN. The shift and normalisation values (discussed in Section 6.4.2) were not removed as part of the imputation step. In the future we would like to be able to impute all parts (polynomial coefficients, shifting, and normalisation values) of the sequence feature vectors (each 'PC' in Table 7.10), however, more data is needed for predicting accurate enough shifting and normalisation values for re-synthesis purposes. These values are useful for better matching the appearance of the identity of the subjects.

For these experiments we used the data from our 4D conversational database (Convo data) and Duchenne/Non-Duchenne smile dataset (Psych data) (Chapter 3). Clearly, for these interactions there is no missing data. Therefore, to predict a sequence's PCs, the data to be predicted is removed before imputation. That is, the implementation is such that the data is imputed on a PC-by-PC basis (i.e. PC 1, then PC 2, then PC 3, until all PC values have been imputed). Figure 7.10 shows an example where PC 2 of Sequence 2 has been removed and is ready to be imputed. This current implementation is a shortcoming in that we only impute one PC (the blank cell in Table 7.10) at a time, rather than all PCs at once, for an FC or BC signal. Again, we require more data and better fitted sequences to produce adequate results for imputing all parts of a FC or BC sequence and Section 8.4 discusses potential solutions for achieving this goal.

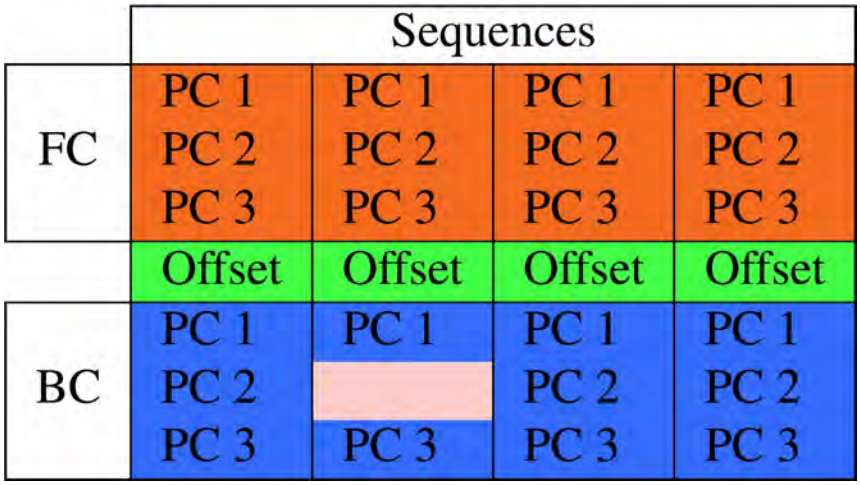


Figure 7.10: Example of the coupled model feature vectors. The blank cell represents the missing feature vector to be imputed.

The original data that has been removed acts as a ground truth and is used to help evaluate how well we predicted the values. The ground truth is used in the classification experiment (Section 7.7) as the training set, and is used to synthesise the original backchannel sequences for the human perceptual study (Section 7.8).

7.7 Experiment 3 - Classifying Predicted Sequences

To analytically evaluate our coupled model approach we classified frontchannel and backchannel predicted sequences. A 7^{th} degree polynomial was used for 3 PCs for the Convo database, which represented 77.63% of the remaining model energy. Creation of the original and predicted frontchannel and backchannel sequences followed the process described in Section 7.6. The polynomial coefficients of the FC and BC feature vectors were used as input into a Support Vector Machine (SVM) classifier, *libSVM* [65]. A ν-SVM with a Gaussian RBF kernel was used, and a grid search was performed for parameter optimisation, as suggested in [128, 207]. The training set was comprised of 22 frontchannel and 22 backchannel original sequences (i.e. polynomial coefficients used for building the coupled model). The testing set was comprised of 22 frontchannel and 22 backchannel predicted sequences. The optimal ν-SVM parameters were: $Cost = -10$, $\nu = 0.95455$. Classification accuracy (Raw and AUC) was 95.45%. The two incorrectly classified sequences were false-positives (i.e. predicted as frontchannel when actually backchannel sequences). Table 7.15 shows the classification confusion matrix. This can be compared to Table 7.16 which shows the combined results for the original FC/BC classification experiment from Experiment 1 (Section 7.3).

Confusion Matrix			
Actual		FC	BC
Predicted	FC	22	2
	BC	0	20

Table 7.15: Coupled Model - Predicted FC/BC Classification

Confusion Matrix			
Actual		FC	BC
Predicted	FC	21	1
	BC	1	21

Table 7.16: Experiment 1 (Section 7.3) - Original FC/BC Classification

If a predicted sequence has been imputed accurately, it reasonable to expect it to be close enough to its original to be classified as the same class (FC or BC). If not, we would hope that it would have similar characteristics of the target class. Apart from

the perceptual studies in Experiment 4 (Section 7.8), this is really the only way to evaluate the quality of prediction. Distance measures for error mean very little as there is no real threshold for what is acceptable. A better measurement, given our modelling and synthesis goals, of the quality of imputation using our coupled model comes from correct classification of sequences based on their characteristics, as well as a qualitative assessment, such as that done in Section 7.8.

7.8 Experiment 4 - Predicted Expressions

Experiment 4 focused on the similarity of original and predicted smile sequences. A 6^{th} degree polynomial was used for 3 PCs for the Psych data, which represented 81.38% of the remaining model energy. After observing the output from various combinations of PCs, we chose to use the top three PCs because they provided the best balance between the number of PCs used and capturing the data and variability required for our experiments. That is, by reducing the number of PCs we are able to reduce processing time, while still retaining the quality and variation of data we require for our work. A 7^{th} degree polynomial was used for 3 PCs for the Convo database, which represented 77.63% of the remaining model energy. These polynomial degrees were empirically determined, so as to avoid over-fitting. The coupled model was built according to the steps described in Section 6.6, and the imputation performed for the polynomial coefficients, as described in Section 6.6.1, with an empirically determined k value of 15 for kNN. The imputed/predicted backchannel sequences were synthesised by projecting the new *bVector* values out of the AAM. The frontchannel sequences and the original backchannel sequences were synthesised using the original polynomial fit values. The offset mean was 22.8636 frames, and the offset standard deviation was 22.1924 frames. Figure 7.11 shows the *bVectors* of Principal Component 1 (PC 1), along with the original polynomial curve and predicted (imputed) curve, for a frontchannel smile sequence (Convo data). Figure 7.12 shows the same type of data for PC 3 of a frontchannel smile sequence.

Figure 7.13 shows the *bVectors* of Principal Component 1 (PC 1), along with the

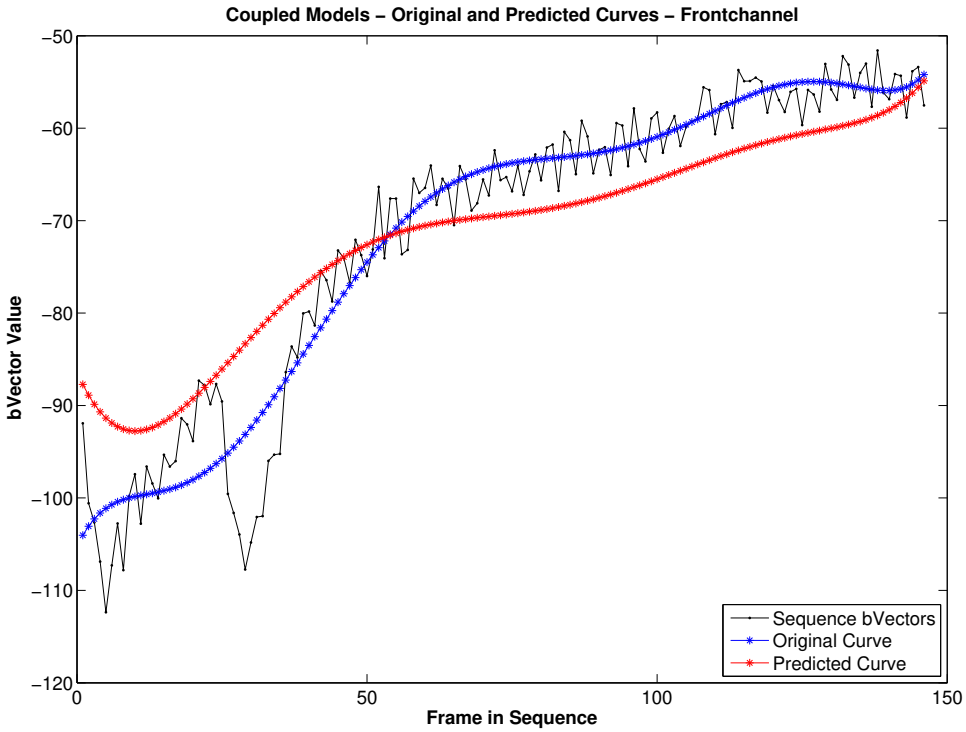


Figure 7.11: Example of Frontchannel Original and Predicted Curves (PC 1)

original polynomial curve and predicted (imputed) curve, for a backchannel smile sequence (Convo data). Figure 7.14 shows the same type of data for PC 3 of a backchannel smile sequence.

Our approach for predicting frontchannel/backchannel sequences is not attempting to replicate the original sequence perfectly, but rather faithfully reproduce the characteristics of these sequences. For this reason, our perceptual experiments (Section 7.8.1 and Section 7.8.2) focus on the similarity between the original and predicted sequences (i.e. the synthesised frames derived from the polynomial curve values and AAM). To model and predict interaction characteristics, our coupled model requires an AAM that contains all frame sequences from all subjects. This results in a *softening* of the texture, through the averaging that occurs in the AAM. The textures could have been made much more realistic (similar to those in Experiment 2 (Section 7.5) by simply unprojecting the shape from the All-Sequences AAM model and then projecting that into the subject-sequence specific AAM models (those built for use in Experiment 2). Unprojecting the combined parameters from that model would result in a much more realistic looking face. Since our main focus was to

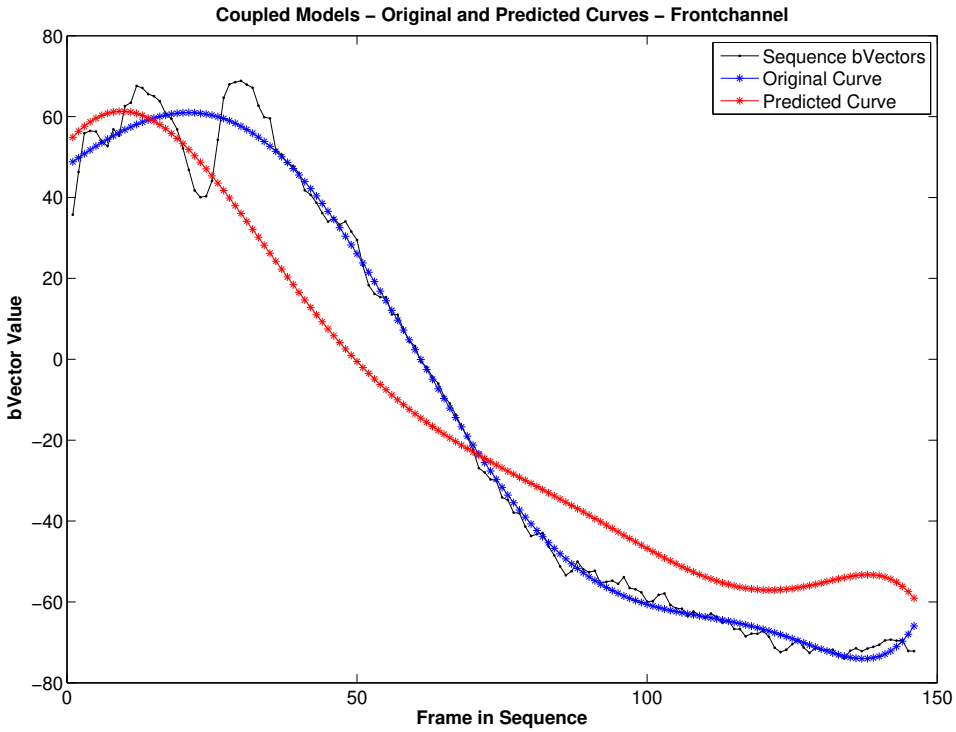


Figure 7.12: Example of Frontchannel Original and Predicted Curves (PC 3)

test the overall abilities of the coupled model (i.e. predicting accurate *bVectors* that would produce legitimate shape and texture frames when synthesis was performed) we decided to use the All-Sequences AAM only.

7.8.1 Psych Results

For the Psych data there was only one "side" of the data; the smile sequence. Therefore, the "coupled model" created for the perceptual experiment contained a feature vector (column) for each sequence. The PC-by-PC imputation was performed for each sequence and the predicted values used for synthesising a predicted smile sequence. These were the sequences used for the perceptual study and allowed us to evaluate the imputation approach using much "cleaner" data than the conversational interactions.

Seventy-two participants (described in Section 7.5) watched the video sequences and evaluated the extent to which they perceived the imputed backchannel expressions as similar to the original expressions. Figure 7.15 shows a screenshot from the

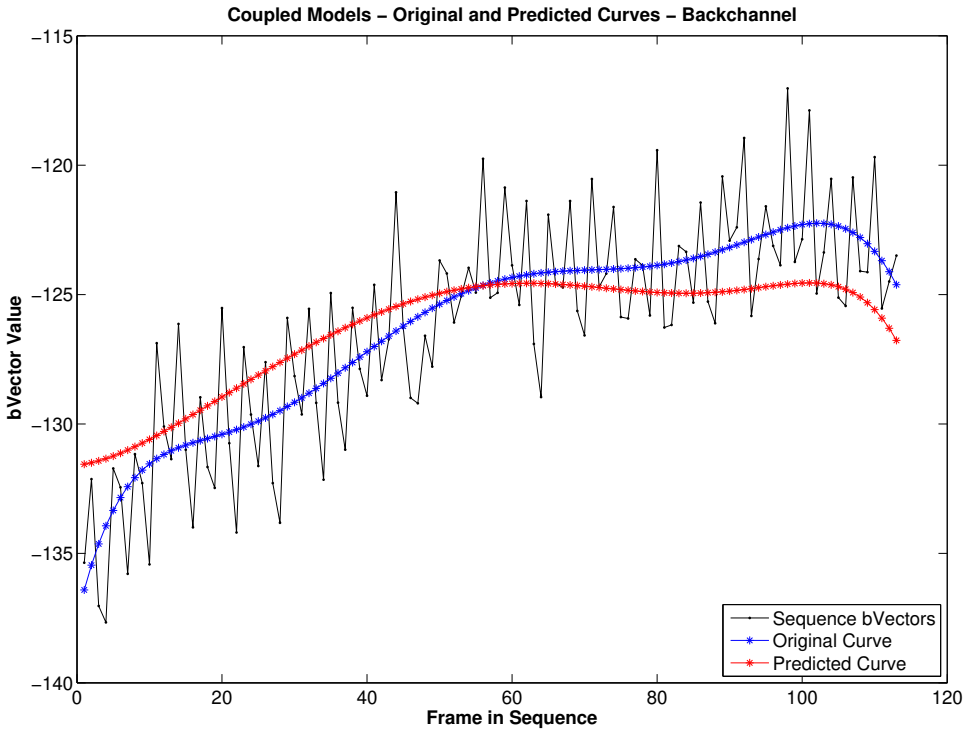


Figure 7.13: Example of Backchannel Original and Predicted Curves (PC 1)

experiment of a predicted smile sequence.

Participants rated the videos on a 4-point Likert-type scale ranging from (1) Very Dissimilar to (4) Very Similar. Each participant rated 10 or 11 videos depending on their assigned group, for a total of 754 trials. One trial was discarded from the analysis due to web-based input errors which resulted in missing data. We also discarded 11 trials for which the participants spent less than 3.02 seconds (this being the length of the shortest video to rate), for a final sample of 742 trials. Similarity ratings, averaged within participants, were marginally significantly different depending on the experiment group, $F(3, 68) = 2.73$, $p = .05$ (Table 7.17). All of them, however, were significantly higher than the scale midpoint (2.5), $M = 3.36$, $S.D. = 0.38$, $p < .001$, suggesting that participants perceived the imputed videos as highly similar to the original versions.

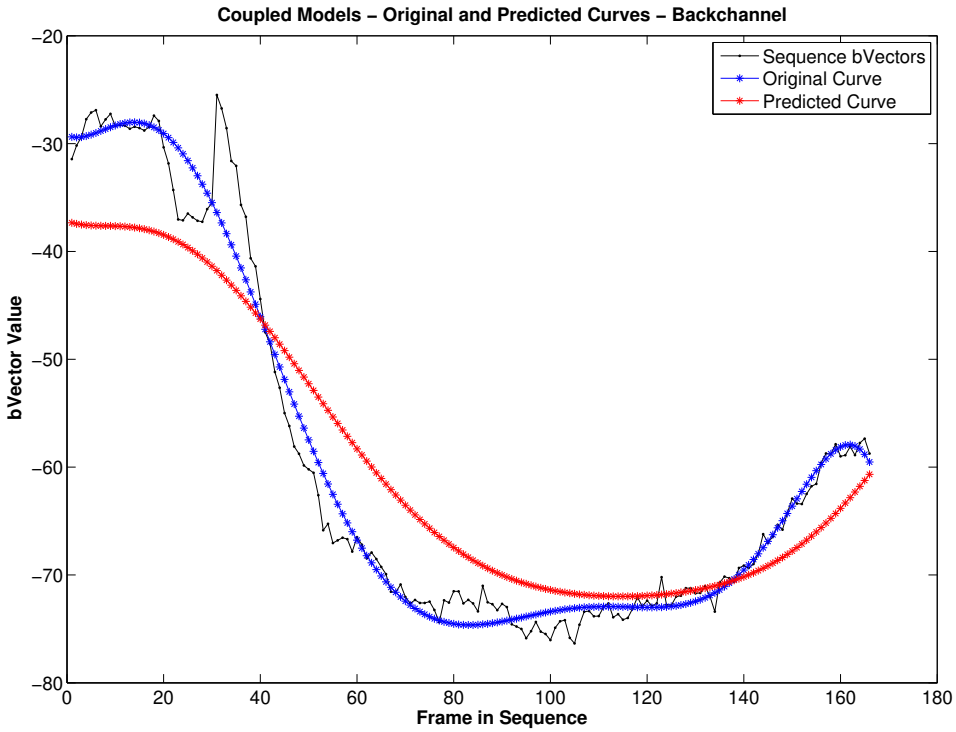


Figure 7.14: Example of Backchannel Original and Predicted Curves (PC 3)

Group	Num. Participants	Avg. Rating	S.D.
Group 1	18	3.18	.41
Group 2	16	3.32	.39
Group 3	21	3.41	.37
Group 4	17	3.52	.38

Table 7.17: Psych Data - Imputed - Average Ratings per Group

7.8.2 Convo Results

Using the conversational smile interactions a coupled model was built and used for predicting and synthesising backchannel sequences. These were used for testing the similarity of original and predicted backchannel smile expressions.

Twenty-eight participants (described in Section 7.5), across two groups, watched the video sequences and evaluated the extent to which they perceived the imputed backchannel expressions as similar to the original expressions. Figure 7.16 shows a screenshot from the experiment of a predicted smile sequence.

Participants rated the videos on a 4-point Likert-type scale ranging from (1) Very Dissimilar to (4) Very Similar. Each participant rated 11 sequences for a total of 308

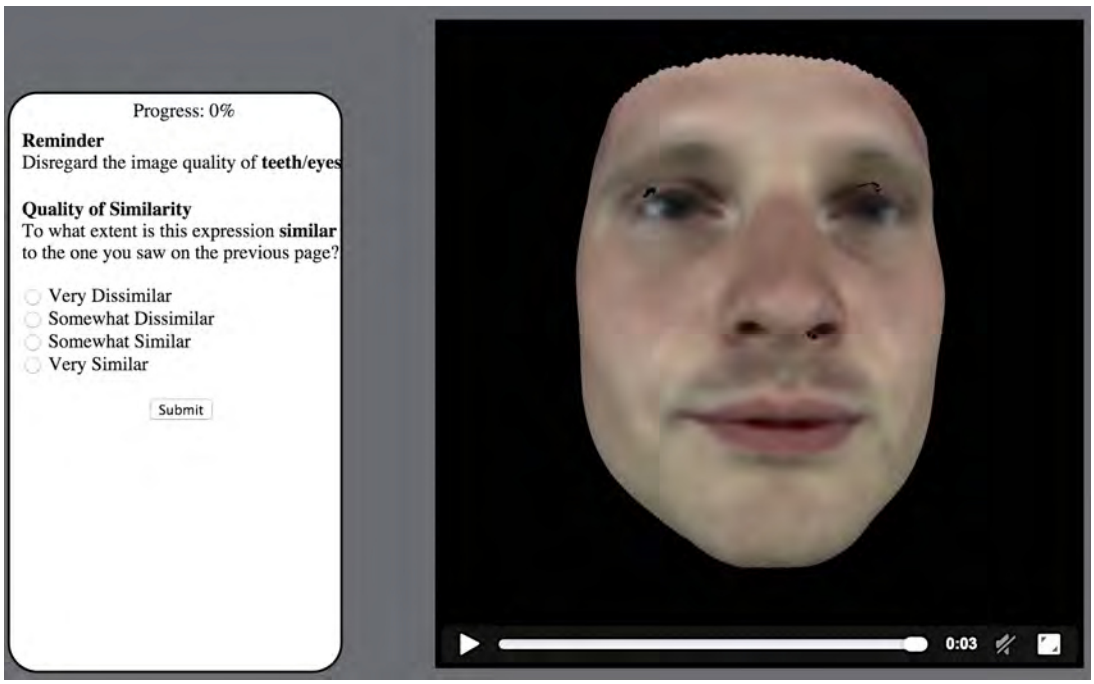


Figure 7.15: Screenshot: Psych Group - Predicted Smile Sequence

Figure 7.16: Screenshot: Convo Group - Predicted Smile Sequence

trials. One trial was discarded from the analysis due to web-based input errors which resulted in missing data. We also discarded one trial for which the participant spent only 1.34 seconds (1.667 seconds being the length of the shortest backchannel video to rate), for a final sample of 306 trials. Similarity ratings, averaged within participants, were significantly higher than the scale midpoint (2.5), $M = 2.90$, $S.D. = 0.31$, $t(27) = 7.00$, $p < .001$, suggesting that participants consistently perceived the imputed videos as similar to the original versions. Ratings of similarity did not significantly differ across the two versions of the questionnaire, $t(26) = .44$, $p > .60$ (Table 7.18).

Group	Num. Participants	Avg. Rating	S.D.
Group 1	18	2.92	.32
Group 2	10	2.87	.30

Table 7.18: Convo Data - Imputed - Average Ratings per Group

7.8.3 Results Discussion

The results support the approach outlined in this thesis for building coupled statistical models of conversational facial expression. This opens the door for using coupled models to analyse more aspects of conversational expression interactions, as well as for synthesising new expression sequences. This includes looking at the perception of backchannel timings, expression intensities, and a variety of other interesting research challenges.

7.9 Discussion

The use of the 3D Active Appearance Model proved to be a strong choice for feature representation. The polynomial regression approach for representing the sequences as a single entity was a sufficient choice, but for future work, specifically the larger perceptual psychology study that is planned (effect of smile characteristic modifications on perceived genuineness and trustworthiness) piecewise polynomial fitting will likely be a better choice. For the smile data which has clear onset, peak, and offset sections, a better approach would be to break the sequences down into these three sections and use a low-order polynomial fit for each section. The slight oscillations which were observed with the higher-order, single-fit modified data are not an issue with this approach. While these oscillations did not affect the perception of realism in our experiments, it could be argued that in a study about the perception of characteristics (e.g. genuineness), such oscillations could have an effect. For the conversational data, the polynomial regression approach was acceptable and fairly optimal, because the structure of the principal component sequence data was highly variable. No standard “break it into parts” approach would work optimally across all the data.

The approach chosen for the coupled models and prediction of backchannel expression sequences was also sufficient. There is an argument that can be made about the context in which expressions are made, but the purpose of the coupled model is to have examples of interactions that are used to inform on missing values of an interaction. Whether or not those expressions are linked to the intrinsic and extrinsic motivations of the individuals in conversation is a much deeper research question which has strong arguments on either side. The other argument that can be made is about the amount and type of interactions used in the coupled models. The data used in these experiments were smile sequences. Not only because this type of data is the most abundant in our data set, but more importantly, it is the expression type our psychology collaborators wish to use for their perceptual studies. Given both of these items, using smile expressions became an obvious choice. There is no reason to believe our approach would not work just as well for other conversational expression types, such as positive-surprise, thinking, and confusion. However, we note that for the coupled model approach to work successfully there needs to be a decent number of these expression examples. We believe our current database does not contain an adequate number of these other expression interaction types for use in our coupled models and we would need to capture more specific data in order to adequately test the other expression interactions with our coupled models. This approach was developed for relatively short conversational facial expression interactions and not for speech interactions or long sequences. We therefore believe that it is not an optimal choice for modelling speech interactions or long sequences. However, further experiments to validate this need to be done and it may be possible to modify this approach in such a way to successfully use the coupled model approach for modelling speech interactions. With more data, more accurate predictions *might* be made, however, the opposite could be argued too; the more data there is, the more generalised the predictions become and as such, the more dissimilar they are to the original. The only way to determine this will be to develop larger coupled models of various conversational expression interactions. This will, however, require more conversational data to be captured and processed. It is a time-and-data-storage

intensive challenge that will hopefully be addressed in the future.

Overall, the results from both the human evaluation of the similarity of predicted backchannel sequences to their original sequences, as well as the classification of frontchannel/backchannel sequences when using the original sequences to train the SVM, support our coupled model and imputation approach. It showed that we can synthesise predicted sequences that are very close to their original versions. Since we were interested in similarity, and not realism, our approach produced frames that had soft textures; a result of the averaging process inherent in AAMs. With an additional step of using the subject-sequence AAMs from Experiment 2 (Section 7.5), the quality of the texture can be brought closer to the level of the sequences in Experiment 2. For each frame, this is achieved by simply unprojecting the shape parameters from the All-subject-sequences AAM and projecting those into the correct subject-specific AAM, and then unproject the combined parameters. For future experiments that want to evaluate the coupled model's ability to produce predicted sequences that are perceived as realistic, this approach is easily and readily available. While not perfect, our coupled modelling approach has demonstrated a novel method for 4D modelling of conversational facial expression interactions between two people.

7.10 Summary

In this chapter, four experiments used for evaluating the tracking, registration, modelling, and synthesising methods detailed in this thesis were described. The first experiment classified Duchenne and Non-Duchenne smile sequences and frontchannel from backchannel smile sequences. The high classification accuracy from these classification experiments support the tracking and registration methods used, as they preserved the quality of the data, as such the minute differences between the two classes of data were still statistically identifiable. In the second experiment human evaluators determined the expression realism and image quality of modified, synthesised expression sequences. All of the sequences, apart from the most extreme (which was expected to be perceived as unrealistic, based on previous research in

psychology), were perceived as realistic by human observers for both expression realism and image quality. Thus, these results supported our modelling, modification, and synthesising approaches. In the third experiment, human observers rated the level of similarity between a predicted sequence and its original version. Results showed that these observers perceived the predicted sequences as similar to the original. These results supported our coupled model and imputation approaches, as they allowed for the prediction of backchannel values which when unprojected from the AAM and rendered, produced a sequence that appeared very similar to the original sequence. A final experiment was performed which classified frontchannel and backchannel predicted sequences. The results of this classification experiment also support our coupled model and imputation methods.

Chapter 8

Conclusion

8.1 Introduction

In this thesis, methods for creating 4D statistical models of conversational expressions and expression interactions, as well as the synthesis of highly-realistic 4D expression sequences, were described.

To achieve these research goals, we developed the world's first 4D database of natural, dyadic conversations [229]. A common issue with 3D/4D multi-view capture systems is the creation of a single-view texture map to allow the modification of the originally captured 3D meshes. A fully-automatic, robust pre-processing pipeline was developed to overcome this issue and can be used by any researchers facing similar issues with such systems. The steps for this pre-processing pipeline are laid out in detail in Chapter 4. A 4D tracking and inter-subject registration approach was developed [121], primarily by our collaborator Lukas Gräser. This tracking and inter-subject registration approach allowed for the building of statistical models, which was necessary for conducting our research. Using Active Appearance Models and polynomial regression, we were able to model expression sequences and synthesise highly-realistic, modified expression sequences that compete with state-of-the-art approaches [38, 54, 80, 143, 238, 243, 249]. Sections 7.3 and 7.5 evaluate our methods in a classification experiment and perceptual study.

This thesis also proposed a novel approach for modelling 4D, time-sequential, conversational facial expression interactions [230]. This coupled statistic model can be used in a variety of applications for analysing and synthesising expression interactions. We used such a model for predicting frontchannel and backchannel sequences of conversational smile expression interactions and conducted a classification experiment (Section 7.7) and a perceptual study (Section 7.8), for evaluating the similarity between the original sequences and the predicted sequences.

Below is a summary of the achievements of this thesis.

8.2 Summary of Thesis Achievements

- The world's first publicly-available, 4D database of natural, dyadic conversations [229]
- A fully-automated, robust, approach for creating single-view texture maps (UTMs) from multi-view texture maps
- The creation of highly-realistic, 4D synthesised facial expression sequences using statistical models, which competes with the quality and realism of current state-of-the-art approaches
- The development of an approach for representing and manipulating model sequences, for generating perceptually convincing synthesised sequences, which were validated by human observers in perceptual experiments [121, 230]
- A novel approach for statistically modelling conversational interactions [230]

8.3 Applications

There are a variety of applications that can be developed using the methods and techniques outlined in this thesis. One such application is creating stimuli for use in psychology experiments. Section 8.4 describes such an application.

The statistical models described in this work, specifically the coupled models of conversational interactions, can be used to increase the naturalness and realism of interactions between humans and virtual agents (i.e. virtual humans). These models could assist in research into the timings of responses, the intensity of responsive expressions, normal expression dynamics for speakers, natural levels of mimicry, and other interesting characteristics of conversational interactions. The believability of these virtual humans is extremely important for applications such as virtual therapists, tutors, or assistants. The information about these conversation characteristics is also of direct use to those in psychology. Whether for better understanding social interactions or helping individuals with conditions that prevent them from interacting normally in social situations (e.g. individuals with Asperger syndrome).

Another area of interest is in enhancing the facial expression realism of digital characters. The texture realism of digital characters is very high, sometimes arguably imperceivable from real-life. The movement of digital character faces, however, still leaves much to be desired. The modelling approaches here could be used to better inform about natural movements of the face and how the face deforms over time, when making these movements. These types of applications are extremely useful for cognitive and social psychologists, as well as affective computing researchers. For obvious reasons, increased realism in temporal dynamics is also of interest to the game and film industry.

While these are the main applications of realistic synthesised facial expressions and conversational interactions, there are many other applications (e.g. deception detection, expression recognition, etc.) and research areas (e.g. biometrics, machine learning, etc.) that could make use of the methods described in this work.

8.4 Future Work

This section will cover future work and research we plan to conduct. Much of the future work focuses on areas in which our methods could be improved. This section

will be structured similarly to the chapter topics of this thesis.

We would like to make publicly available the fully-automatic, robust pre-processing pipeline for cleaning 3D meshes and create single-view texture maps. Before this release, however, there are few additions and modifications we would like to make. The first is to change the process of rendering the Unified Texture Map (Section 4.5). Currently, the pipeline uses the freely-available *Psychophysics Toolbox* [55]. While this toolbox allows us to render the UTM, it is not an ideal implementation. It's main purpose is not for rendering images, it lacks strong developer support, and requires the installation of a few third-party tools. A much better implementation would be to use Blender [1] for rendering, as we do for creating the 2D videos of our 4D sequences. The next modification we would like to make to the pre-processing pipeline is to remove all dependence on third-party tools, specifically the function used for flattening the cleaned mesh into the UTM mesh. This is currently done using an older Matlab toolbox written by Gabriel Peyré (*Toolbox Graph* [184]) and while it works well, there are some instances where it fails. By implementing our own version we will have a function specific to our task and full-control on how it integrates with our pipeline. This includes error checking for when the parameterisation process fails. Finally, we would like to explore the possibility of implementing a version of our pipeline that utilises distributed computing.

We would like to evaluate our tracking and inter-subject registration methods on multiple 4D datasets. The approach worked well for our data and experiments, but to fully evaluate its abilities (and weaknesses), we need to test it on other datasets. This is a time-consuming task, both for the annotation step and processing stage, but it will allow us to identify areas of improvement. Tracking and registration is an integral part of our statistical modelling approach for facial expression research, and an open area of research, so there are quite a few improvements that still can be made.

A better approach, specifically for the creation of psychology stimuli, needs to be developed for representing sequences as single entities. The single polynomial fitting

approach used in our experiments (Chapter 7) was sufficient for our needs, but will need to be improved for future research. One main drawback to our polynomial regression approach is the oscillation that tend to occur at the edges of our high-order polynomial (*Runge's phenomenon* [199]). B-Splines proved to be an excellent curve fitting approach, and upon further exploration we believe we can use this approach to produce the input for our coupled models. It is important that we represent the sequences as single entities, but also that we represent the sequences in a way that allows them to be properly comparable. An improvement in this area would allow us to synthesise even more accurate expression sequences and conduct perceptual studies without the worry that the oscillation issue is causing a negative effect on the perceptions of human observers.

The improvement described above is especially important given the much larger psychology study that is planned. The experiment will show smile sequences whose onset, peak, or offset values have been modified, both for duration and amplitude. Evaluations from human observers will be used in a reverse correlation approach for determining the characteristics that are involved in trustworthy and genuine smile sequences. This experiment will utilise the Duchenne and Non-Duchenne FACS-validated smiles described in Section 3.4.

While the results in the perceptual experiments conducted in Chapter 7 supported our approach for producing realistic-looking stimuli for the purposes of perceptual studies, we will need to further develop our methods and perform further experiments in order to claim full ("real-life") realism and believability. That is, instead of having our human observers rating the realism of expressions and image quality, we should show real video examples of expressions and have them rate the synthesised sequences against those. This would be the ultimate test of synthesised sequence realism. Our methods are not currently at that point, but we hope to make strides towards that level of modification and realism in our future work.

Further development on the coupled models of conversational interactions will be conducted. The data used in these experiments were smile sequences, mainly because

of the abundance of this type of interaction and the research requirements of our psychology collaborators. The coupled model approach for modelling interactions may have limitations when it comes to other expressions or speech interactions. While we believe it will perform adequately for other types of expression interactions we will need to perform further experiments to validate this belief. This approach was not designed for speech interactions or long sequences, so potential users should not expect it to perform well in those instances. Multi-expression sequences (e.g. a backchannel sequence of a smile to surprise to confusion) might also not work well. This modelling method currently works best with single expression interactions but we will attempt to expand the abilities of the coupled model approach and validate these new developments experimentally. As well, more imputation approaches need to be considered to understand if certain methods perform better in certain situations. The kNN approach used in this thesis was sufficient for our usage, but when creating coupled models of different conversational interactions, or perhaps extended interactions (i.e. *FC-BC-FC*, *FC-BC-FC-BC*, etc.), this may not be the case. As well, we would like to use our coupled model with more conversational interactions, perhaps some from the 2D CCDb database [22], if we are able to process the data for creating 3D sequences. Adding more data may also allow us to predict entire sequence feature vectors, rather than just the polynomial coefficients that were used in our experiments (Chapter 7). Exploring a variety of interactions is of interest. There are many types of conversational expressions and not all are captured in 3-minute, natural conversation captures. Developing a data capture session that elicits a variety of conversational expressions may be of use for future research, especially for the evaluation of coupled models of conversational interactions. Finally, we would like to assist our psychology collaborators in using our conversation database with annotated interactions and coupled statistical modelling approach to explore the mechanisms of conversational interactions that could exist and not yet been discovered (i.e. not just the obvious and well-known prototypical or conversational facial expressions).

The newly formed *Cardiff Face Researcher's Group* brings together researchers from

many different disciplines at Cardiff University. It is the hope of the author of this work that the data and methods described in this work will continue to be used for a variety of research, experiments, and applications in the areas currently represented by the group's members: computer science, psychology, neuroscience, dentistry (orthodontics), and optometry. Future research projects being discussed include building models from data captured of individuals speaking the same sentences and making the same facial expressions, over the span of many years, for understanding how facial dynamics and appearances change over time (computer science); using fMRI to map brain functions to a diverse set of elicited reactions based on facial expressions (neuroscience); understanding the dynamics of face movement pre-surgery and exploring techniques for preserving or improving these dynamics post-surgery (dentistry); preserving the recorded facial expressions of a patient while removing identifying features, thus making the subject anonymous while retaining important expression information, for use in various experiments (clinical psychology); understanding the temporal dynamics of trustworthy and genuine smiles, along with developing a better understanding of the characteristics of various types of smiles expressed during conversations (social psychology); and how the dynamics of different parts of the face, and visual occlusions, may affect the perception of facial expressions and conversational signals, while being viewed in 3D (optometry). The improvements to our methods and the future work described in this chapter will provide a means for creating a better understanding of the role of facial expressions in social communication.

Bibliography

[1] Blender. http://www.blender.org.

[2] StreamPix. https://www.norpix.com/products/streampix/streampix.php.

[3] Python. https://www.python.org.

[4] 3dMD. http://www.3dmd.com.

[5] Di4D. http://www.di4d.com.

[6] Di4D Sample Data. http://www.di4d.com/sample-data.

[7] B. Abboud and F. Davoine. Appearance factorization based facial expression recognition and synthesis. In *Pattern Recognition (ICPR 2004), Proceedings of the 17th International Conference on*, volume 4, pages 163–166. IEEE, 2004.

[8] B. Abboud, F. Davoine, and M. Dang. Facial expression recognition and synthesis based on an appearance model. *Signal Processing: Image Communication*, 19(8):723–740, 2004.

[9] O. Alexander, G. Fyffe, J. Busch, X. Yu, R. Ichikari, A. Jones, P. Debevec, J. Jimenez, E. Danvoye, B. Antionazzi, et al. Digital Ira: Creating a real-time photoreal digital actor. In *2013 ACM SIGGRAPH Posters*, page 1. ACM, 2013.

[10] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. Creating a photoreal digital actor: The digital emily project. In *Conference for Visual Media Production (CVMP 2009)*, pages 176–187. IEEE, 2009.

[11] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics (TOG)*, 22(3):485–493, 2003.

[12] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of the 6th ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001.

[13] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2):127–153, 2001.

[14] R. Anderson, B. Stenger, V. Wan, and R. Cipolla. Expressive visual text-to-speech using active appearance models. In *Computer Vision and Pattern Recognition (CVPR 2013), IEEE Conference on*, pages 3382–3389. IEEE, 2013.

[15] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon. The painful Face – Pain expression recognition using active appearance models. *Image and Vision Computing*, 27(12):1788–1796, 2009.

[16] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3D pose normalization. In *Computer Vision (ICCV 2011), IEEE International Conference on*, pages 937–944. IEEE, 2011.

[17] M. Attene. The jmeshlib library. `http://jmeshlib.sourceforge.net`, 2006.

[18] M. Attene. A lightweight approach to repairing digitized polygon meshes. *The Visual Computer*, 26(11):1393–1406, 2010.

[19] M. Attene, M. Campen, and L. Kobbelt. Polygon mesh repairing: An application perspective. *ACM Computing Surveys (CSUR)*, 45(2):15, 2013.

[20] A. Aubrey, D. Marshall, and P. Rosin. Behaviour transfer between expressive talking heads. In *Proceedings of the SSPNET 2nd International Symposium on Facial Analysis and Animation*, page 9. ACM, 2010.

[21] A. J. Aubrey, V. Kajić, I. Cingovska, P. L. Rosin, and D. Marshall. Mapping and manipulating facial dynamics. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE International Conference on*, pages 639–645. IEEE, 2011.

[22] A. J. Aubrey, D. Marshall, P. L. Rosin, J. Vandeventer, D. W. Cunningham, and C. Wallraven. Cardiff Conversation Database (CCDb): A database of natural dyadic conversations. In *Computer Vision and Pattern Recognition Workshops (CVPRW 2013), IEEE Conference on*, pages 277–282. IEEE, 2013.

[23] T. Baltrušaitis, P. Robinson, and L.-P. Morency. 3D constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR 2012), IEEE Conference on*, pages 2610–2617. IEEE, 2012.

[24] B. Barbour and K. Ricanek Jr. An interactive tool for extremely dense landmarking of faces. In *Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications*, page 13. ACM, 2012.

[25] G. Barequet and M. Sharir. Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design*, 12(2):207–229, 1995.

[26] M. Bartlett, G. Littlewort, J. Whitehill, E. Vural, T. Wu, K. Lee, A. Erçil, M. Cetin, and J. Movellan. Insights on spontaneous facial expressions from automatic expression measurement. In M. A. Giese, C. Curio, and H. H. Bülthoff, editors, *Dynamic Faces: Insights from Experiments and Computation*, chapter 14, pages 211–238. MIT Press, 2010.

[27] M. Bartlett and J. Whitehill. Automated facial expression measurement: Recent applications to basic research in human behavior, learning, and education. In A. Calder, G. Rhodes, J. V. Haxby, and M. H. Johnson, editors, *Oxford Handbook of Face Perception*, pages 489–513. Oxford University Press, 2011.

[28] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous

behavior. In *Computer Vision and Pattern Recognition (CVPR 2005), IEEE Conference on*, volume 2, pages 568–573. IEEE, 2005.

[29] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In *Automatic Face and Gesture Recognition (FG 2006), IEEE International Conference on*, pages 223–230. IEEE, 2006.

[30] M. S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 1, pages 592–597. IEEE, 2004.

[31] M. S. Bartlett, J. R. Movellan, G. Littlewort, B. Braathen, M. G. Frank, and T. J. Sejnowski. Towards automatic recognition of spontaneous facial actions. In P. Ekman and E. L. Rosenberg, editors, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS), Second Edition*, pages 393–426. Oxford University Press, 2005.

[32] G. E. Batista and M. C. Monard. A study of k-nearest neighbour as an imputation method. In *Hybrid Intelligent Systems (HIS), 2002 International Conference on*, volume 87, pages 251–260. IOS Press, 2002.

[33] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, pages 1554–1563, 1966.

[34] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, pages 164–171, 1970.

[35] J. B. Bavelas, A. Black, C. R. Lemery, and J. Mullett. "I show how you feel": Motor mimicry as a communicative act. *Journal of Personality and Social Psychology*, 50(2):322–329, 1986.

[36] J. B. Bavelas, L. Coates, and T. Johnson. Listeners as co-narrators. *Journal of Personality and Social Psychology*, 79(6):941–952, 2000.

[37] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[38] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (TOG)*, 30(4):75:1–75:10, 2011.

[39] L. Benedikt, D. Cosker, P. L. Rosin, and D. Marshall. Assessing the uniqueness and permanence of facial actions for use in biometric applications. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(3):449–460, 2010.

[40] L. Benedikt, V. Kajic, D. Cosker, P. L. Rosin, and A. D. Marshall. Facial dynamics in biometric identification. In *BMVC*, pages 1–10, 2008.

[41] A. H. Bermano, D. Bradley, T. Beeler, F. Zund, D. Nowrouzezahrai, I. Baran, O. Sorkine-Hornung, H. Pfister, R. W. Sumner, B. Bickel, and M. Gross. Facial performance enhancement using dynamic shape space analysis. *ACM Transactions on Graphics (TOG)*, 33(2):13:1–12, 2014.

[42] R. L. Birdwhistell. *Kinesics and Context: Essays on body motion communication.* University of Pennsylvania Press, 1970.

[43] S. Bischoff and L. Kobbelt. Structure preserving cad model repair. In *Computer Graphics Forum*, volume 24, pages 527–536. Wiley Online Library, 2005.

[44] S. Bischoff, D. Pavic, and L. Kobbelt. Automatic restoration of polygon models. *ACM Transactions on Graphics (TOG)*, 24(4):1332–1352, 2005.

[45] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 187–194. ACM Press, 1999.

[46] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 25(9):1063–1074, 2003.

[47] S. M. Boker, J. F. Cohn, B.-J. Theobald, I. Matthews, T. R. Brick, and J. R. Spies. Effects of damping head movement and facial expression in dyadic conversation using real–time facial expression tracking and synthesized avatars. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3485–3495, 2009.

[48] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *Pattern Analysis & Machine Intelligence (TPAMI), IEEE Transactions on*, 11(6):567–585, 1989.

[49] G. Borshukov, D. Piponi, O. Larsen, J. P. Lewis, and C. Tempelaar-Lietz. Universal capture-image-based facial animation for the matrix reloaded. In *ACM SIGGRAPH*, page 16. ACM, 2005.

[50] M. Botsch and L. Kobbelt. A robust procedure to eliminate degenerate faces from triangle meshes. In *VMV*, pages 283–290, 2001.

[51] B. Braathen, M. S. Bartlett, G. Littlewort, E. Smith, and J. R. Movellan. An approach to automatic recognition of spontaneous facial actions. In *Automatic Face and Gesture Recognition (FG 2002), IEEE International Conference on*, pages 360–365. IEEE, 2002.

[52] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[53] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Computer Vision and Pattern Recognition (CVPR 2008), IEEE Conference on*, pages 1–8. IEEE, 2008.

[54] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics (TOG)*, 29(4):41:1–10, 2010.

[55] D. H. Brainard. The psychophysics toolbox. *Spatial Vision*, 10:433–436, 1997.

[56] P. Bull. State of the art: Nonverbal communication. *The Psychologist*, 14(12):644–647, 2001.

[57] N. J. Butko, G. Theocharous, M. Philipose, and J. R. Movellan. Automated facial affect analysis for one-on-one tutoring applications. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE International Conference on*, pages 382–387. IEEE, 2011.

[58] B. F. Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2(2):59–75, 1984.

[59] A. R. Calvo, F. T. Ruiz, J. Rurainsky, and P. Eisert. 2D-3D mixed face recognition schemes. *Recent Advances in Face Recognition*, pages 125–48, 2008.

[60] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *Signal Processing Letters, IEEE*, 13(5):308–311, 2006.

[61] P. Carrera-Levillain and J.-M. Fernandez-Dols. Neutral faces in context: Their emotional meaning and their function. *Journal of Nonverbal Behavior*, 18(4):281–299, 1994.

[62] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjalmsson, and H. Yan. More than just a pretty face: conversational protocols and the affordances of embodiment. *Knowledge-Based Systems*, 14(1):55–64, 2001.

[63] J. Cassell and K. R. Thorisson. The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, 13(4-5):519–538, 1999.

[64] M. Castelán. *Face Shape Recovery from a Single Image View*. PhD thesis, University of York, 2006.

[65] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *Intelligent Systems and Technology (TIST), ACM Transactions on*, 2(3):27:1–27, 2011.

[66] G. Chen. Edge detection by regularized cubic B-spline fitting. *Systems, Man and Cybernetics, IEEE Transactions on*, 25(4):636–643, 1995.

[67] H. Chipman, T. J. Hastie, and R. Tibshirani. Clustering microarray data. In T. Speed, editor, *Statistical Analysis of Gene Expression Microarray Data*, chapter 4, pages 159–200. Chapman & Hall/CRC, 2003.

[68] P. Cignoni, M. Corsini, and G. Ranzuglia. Meshlab: An open-source 3D mesh processing system. *European Research Consortium for Informatics and Mathematics (ERCIM) News*, 73:45–46, 2008.

[69] J. Cockburn, M. Bartlett, J. Tanaka, J. Movellan, M. Pierce, and R. Schultz. SmileMaze: A tutoring system in real-time facial expression perception and production in children with Autism Spectrum Disorder. In *Automatic Face & Gesture Recognition (FG 2008), Workshop on Facial and Bodily expressions for Control and Adaptation of Games (ECAG), IEEE International Conference on*, pages 978–986, 2008.

[70] T. Coleman. Estimating the correlation of non-contemporaneous time-series. *Social Science Research Network (SSRN)*, 2007.

[71] T. F. Cootes. Model-based methods in analysis of biomedical images. In R. Baldock and J. Graham, editors, *Image Processing and Analysis: A Practical approach*, chapter 7, pages 223–248. Oxford University Press, 1999.

[72] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the 5th European Conference on Computer Vision (ECCV 1998)*, volume 1407, pages 484–498. Springer, 1998.

[73] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 23(6):681–685, 2001.

[74] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision, 2004.

[75] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995.

[76] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. Coupled-view active appearance models. In *Proceedings of the 11th British Machine Vision Conference (BMVC 2000)*, volume 1, pages 52–61, 2000.

[77] M. Core, D. Traum, H. C. Lane, W. Swartout, J. Gratch, M. Van Lent, and S. Marsella. Teaching negotiation skills through practice and reflection with virtual humans. *Simulation*, 82(11):685–701, 2006.

[78] D. Cosker, R. Borkett, D. Marshall, and P. L. Rosin. Towards automatic performance-driven animation between multiple types of facial model. *Computer Vision, The Institution of Engineering and Technology (IET)*, 2(3):129–141, 2008.

[79] D. Cosker, E. Krumhuber, and A. Hilton. Perception of linear and nonlinear motion properties using a FACS validated 3D facial model. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 101–108. ACM, 2010.

[80] D. Cosker, E. Krumhuber, and A. Hilton. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2296–2303. IEEE, 2011.

[81] B. G. Cox. The weighted sequential hot deck imputation procedure. In *Proceedings of the American Statistical Association, Section on Survey Research Methods*, pages 721–726. American Statistical Association, 1980.

[82] A. Cray. *Modern Differential Geometry of Curves and Surfaces*. CRC Press, 2nd edition, 1998.

[83] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *Proceedings of the 17th British Machine Vision Conference (BMVC 2006)*, pages 95:1–10. BMVA Press, 2006.

[84] D. W. Cunningham, M. Kleiner, H. H. Bülthoff, and C. Wallraven. The components of conversational facial expressions. In *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization (APGV 2004)*, pages 143–150. ACM, 2004.

[85] D. W. Cunningham, M. Kleiner, C. Wallraven, and H. H. Bülthoff. Manipulating video sequences to determine the components of conversational facial expressions. *ACM Transactions on Applied Perception (TAP)*, 2(3):251–269, 2005.

[86] D. W. Cunningham, M. Nusseck, C. Wallraven, and H. H. Bülthoff. The role of image size in the recognition of conversational facial expressions. *Computer Animation and Virtual Worlds*, 15(3-4):305–310, 2004.

[87] D. W. Cunningham and C. Wallraven. Dynamic information for the recognition of conversational expressions. *Journal of Vision*, 9(13):7:1–7:17, 2009.

[88] C. De Boor. A practical guide to splines. *Mathematics of Computation*, 1978.

[89] I. de Kok and D. Heylen. The MultiLis corpus – Dealing with individual differences in nonverbal listening behavior. In A. Esposito, A. M. Esposito, R. Martone, V. Müller, and G. Scarpetta, editors, *Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues*, volume 6456, pages 362–375. Springer, 2011.

[90] J. P. De Ruiter, S. Rossignol, L. Vuurpijl, D. W. Cunningham, and W. J. Levelt. SLOT: A research platform for investigating multimodal communication. *Behavior Research Methods, Instruments, & Computers*, 35(3):408–419, 2003.

[91] P. Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia Technical Briefs*, 2012.

[92] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Computer Vision and Pattern Recognition (CVPR 1996), IEEE Conference on*, pages 231–238. IEEE, 1996.

[93] D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, 2000.

[94] B. Delaunay. Sur la sphere vide (On the empty sphere). *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 6:793–800, 1934.

[95] D. DeVault, R. Artstein, G. Benn, T. Dey, E. Fast, A. Gainer, K. Georgila, J. Gratch, A. Hartholt, M. Lhommet, et al. SimSensei Kiosk: A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1061–1068. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[96] T. K. Dey. *Curve and surface reconstruction: algorithms with mathematical analysis*, volume 23. Cambridge University Press, 2006.

[97] T. K. Dey and T. Ray. Polygonal surface remeshing with delaunay refinement. *Engineering with Computers*, 26(3):289–301, 2010.

[98] F. Dornaika and J. Ahlberg. Fast and reliable active appearance model search for 3-D face tracking. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions On*, 34(4):1838–1853, 2004.

[99] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive theory of functions of several variables*, volume 571, pages 85–100. Springer, 1977.

[100] R. Durstenfeld. Algorithm 235: Random Permutation. *Communications of the ACM*, 7(7):420, 1964.

[101] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques*, pages 173–182. ACM, 1995.

[102] P. H. Eilers and B. D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, pages 89–102, 1996.

[103] P. Ekman, R. J. Davidson, and W. V. Friesen. The Duchenne Smile: Emotional expression and brain physiology: II. *Journal of Personality and Social Psychology*, 58(2):342–353, 1990.

[104] P. Ekman, W. Friesen, and J. Hager. *Facial Action Coding System - The Manual on CD-ROM*. A Human Face, 2002.

[105] P. Ekman and W. V. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.

[106] Q. Fang, D. Boas, et al. Tetrahedral mesh generation from volumetric binary and grayscale images. In *Biomedical Imaging (ISBI): From Nano to Macro, IEEE International Symposium on*, pages 1142–1145. IEEE, 2009.

[107] T. Fang, X. Zhao, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris. 3D/4D facial expression analysis: An advanced annotated face model approach. *Image and Vision Computing*, 30(10):738–749, 2012.

[108] T. Fernandes, S. Alves, J. Miranda, C. Queirós, and V. Orvalho. LIFEisGAME: A facial character animation system to help recognize facial expressions. In *Enterprise information systems*, pages 423–432. Springer, 2011.

[109] M. S. Floater, K. Hormann, and M. Reimers. Parameterization of manifold triangulations. In *Approximation Theory X: Abstract and Classical Analysis*, pages 197–209. Vanderbilt University Press, Nashville, 2002.

[110] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 249–254. ACM Press, 2000.

[111] C. D. Frowd, B. J. Matuszewski, L.-K. Shark, W. Quan, I. Rudas, M. Demiralp, and N. Mastorakis. Towards a comprehensive 3D dynamic facial expression database. In *Proceedings of the 9th WSEAS International Conference on Multimedia, Internet and Video Technology*, number 9, pages 113–119. World Scientific and Engineering Academy and Society (WSEAS), 2009.

[112] G. Fyffe, A. Jones, O. Alexander, R. Ichikari, and P. Debevec. Driving high-resolution facial scans with video performance capture. *ACM Transactions on Graphics (TOG)*, 34(1):8, 2014.

[113] A. Galata, N. Johnson, and D. Hogg. Learning structured behaviour models using variable length markov models. In *Modelling People, IEEE International Workshop on*, pages 95–102. IEEE, 1999.

[114] T. Gautama and M. M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *Neural Networks, IEEE Transactions on*, 13(5):1127–1136, 2002.

[115] J. Ghent and J. McDonald. Photo-realistic facial expression synthesis. *Image and Vision Computing*, 23(12):1041–1050, 2005.

[116] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG)*, 30(6):129, 2011.

[117] A. Ghosh, T. Hawkins, P. Peers, S. Frederiksen, and P. Debevec. Practical modeling and acquisition of layered facial reflectance. In *ACM Transactions on Graphics (TOG)*, volume 27, page 139. ACM, 2008.

[118] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, 1950.

[119] I. Gonzalez, H. Sahli, and W. Verhelst. Automatic recognition of lower facial action units. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, pages 8:1–8:4. ACM, 2010.

[120] L. Grammatikopoulos, I. Kalisperakis, G. Karras, and E. Petsa. Automatic multi-view texture mapping of 3D surface projections. In *Proceedings of the 2nd ISPRS International Workshop 3D-ARCH*, pages 1–6, 2007.

[121] L. Gräser, J. Vandeventer, J. van der Schalk, P. L. Rosin, and D. Marshall. 4D tracking and inter-subject registration for the synthesis of realistic facial expression sequences. *In Prep.*, 2015.

[122] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[123] P. Hammond, T. J. Hutton, J. E. Allanson, L. E. Campbell, R. Hennekam, S. Holden, M. A. Patton, A. Shaw, I. K. Temple, M. Trotter, et al. 3D analysis of facial morphology. *American Journal of Medical Genetics Part A*, 126(4):339–348, 2004.

[124] P. Hammond, M. Suttie, R. C. Hennekam, J. Allanson, E. M. Shore, and F. S. Kaplan. The face signature of fibrodysplasia ossificans progressiva. *American Journal of Medical Genetics Part A*, 158(6):1368–1380, 2012.

[125] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein. Imputing missing data for gene expression arrays. Technical report, Division of Biostatistics, Stanford University, 1999. `http://www.web.stanford.edu/~hastie/Papers/missing.pdf`.

[126] D. Hogg, N. Johnson, R. Morris, D. Buesching, and A. Galata. Visual models of interaction. In *Proceedings of the 2nd International Workshop on Cooperative Distributed Vision*, 1998.

[127] B. K. Horn and B. G. Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[128] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[129] X. Huang, S. Zhang, Y. Wang, D. Metaxas, and D. Samaras. A hierarchical framework for high resolution facial expression tracking. In *Computer Vision and Pattern Recognition Workshop (CVPRW 2004), Conference on*, pages 22–29. IEEE, 2004.

[130] Y. Huang, X. Zhang, Y. Fan, L. Yin, L. Seversky, J. Allen, T. Lei, and W. Dong. Reshaping 3D facial scans for facial appearance modeling and 3D facial expression analysis. *Image and Vision Computing*, 30(10):750–761, 2012.

[131] E. A. Isaacs and J. C. Tang. What video can and cannot do for collaboration: A case study. *Multimedia Systems*, 2(2):63–73, 1994.

[132] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Computer Vision and Pattern Recognition (CVPR 1993), IEEE Conference on*, pages 760–761. IEEE, 1993.

[133] N. Johnson, A. Galata, and D. Hogg. The acquisition and use of interaction behaviour models. In *Computer Vision and Pattern Recognition (CVPR 1998), IEEE Conference on*, pages 866–871. IEEE, 1998.

[134] T. Ju. Robust repair of polygonal models. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 888–895. ACM, 2004.

[135] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition (FG 2000), IEEE International Conference on*, pages 46–53. IEEE, 2000.

[136] D. A. Kashy and D. A. Kenny. The analysis of data from dyads and groups. *Handbook of research methods in social and personality psychology*, pages 451–477, 2000.

[137] D. A. Kenny, D. A. Kashy, and W. L. Cook. *Dyadic data analysis*. Guilford Press, 2006.

[138] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001.

[139] G. Klincsek. Minimal triangulations of polygonal domains. *Ann. Discrete Math*, 9:121–123, 1980.

[140] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley, Third edition, 1997.

[141] E. Kreyszig and E. J. Norminton. *Advanced engineering mathematics*. Wiley, fourth edition, 1979.

[142] R. Krovi, A. C. Graesser, and W. E. Pracht. Agent behaviors in virtual negotiation environments. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 29(1):15–25, 1999.

[143] E. G. Krumhuber, L. Tamarit, E. B. Roesch, and K. R. Scherer. FACSGen 2.0 animation software: Generating three-dimensional FACS-valid facial expressions for emotion research. *Emotion*, 12(2):351–363, 2012.

[144] H.-S. Lee and D. Kim. Tensor-based AAM with continuous variation estimation: Application to variation-robust face recognition. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 31(6):1102–1116, 2009.

[145] M. Legerstee. The role of dyadic communication in social cognitive development. *Advances in Child Development and Behavior*, 37:1–53, 2009.

[146] B. Li, X. Zhang, P. Zhou, and P. Hu. Mesh parameterization based on one-step inverse forming. *Computer-Aided Design*, 42(7):633–640, 2010.

[147] J. J. Lien, T. Kanade, J. F. Cohn, and C.-C. Li. Automated facial expression recognition based on facs action units. In *Automatic Face and Gesture Recognition (FG 1998), IEEE International Conference on*, pages 390–395. IEEE, 1998.

[148] P. Linell, L. Gustavsson, and P. Juvonen. Interactional dominance in dyadic communication: A presentation of initiative-response analysis. *Linguistics*, 26(3):415–442, 1988.

[149] C. X. Ling, J. Huang, and H. Zhang. AUC: A statistically consistent and more discriminating measure than accuracy. In *IJCAI*, volume 3, pages 519–524, 2003.

[150] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. The Computer Expression Recognition Toolbox (CERT). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE International Conference on*, pages 298–305. IEEE, 2011.

[151] G. C. Littlewort, M. S. Bartlett, and K. Lee. Faces of Pain: Automated measurement of spontaneous facial expressions of genuine and posed pain. In *Proceedings of the 9^{th} International Conference on Multimodal Interfaces (ICMI 2007)*, pages 15–21. ACM, 2007.

[152] G. C. Littlewort, M. S. Bartlett, and K. Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27(12):1797–1803, 2009.

[153] G. C. Littlewort, M. S. Bartlett, L. P. Salamanca, and J. Reilly. Automated measurement of children's facial expressions during problem solving tasks.

In *Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE International Conference on*, pages 30–35. IEEE, 2011.

[154] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. In *Computer Graphics Forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.

[155] X. Liu and T. Chen. Video-based face recognition using adaptive hidden Markov models. In *Computer Vision and Pattern Recognition (CVPR 2003), IEEE Conference on*, volume 1, pages 340–345. IEEE, 2003.

[156] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, The proceedings of the 7th IEEE International Conference on*, volume 2, pages 1150–1157. IEEE, 1999.

[157] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI 1981)*, volume 81, pages 674–679, 1981.

[158] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW 2010), IEEE Conference on*, pages 94–101. IEEE, 2010.

[159] S. Lucey, A. B. Ashraf, and J. F. Cohn. Investigating spontaneous facial action recognition through AAM representations of the face. In K. Delac and M. Grgic, editors, *Face Recognition*. InTech, 2007.

[160] J. Lundgren. Alpha shapes of 2D/3D point set. `http://uk.mathworks.com/matlabcentral/fileexchange/28851-alpha-shapes`, 2010.

[161] W. Ma and J.-P. Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.

[162] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. In *Proceedings of the 2008 ACM symposium on Solid and Physical Modeling (SPM)*, pages 269–280. ACM, 2008.

[163] MATLAB. *version 8.2.0.701 (R2013b)*. The MathWorks Inc., 2013.

[164] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004.

[165] B. J. Matuszewski, W. Quan, L.-K. Shark, A. S. Mcloughlin, C. E. Lightbody, H. C. Emsley, and C. L. Watkins. Hi4D-ADSIP 3-D dynamic facial articulation database. *Image and Vision Computing*, 30(10):713–727, 2012.

[166] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schröder. The SEMAINE Database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *Affective Computing, IEEE Transactions on*, 3(1):5–17, 2012.

[167] A. Mehrabian and S. R. Ferris. Inference of attitudes from nonverbal communication in two channels. *Journal of Consulting Psychology*, 31(3):248–252, 1967.

[168] A. F. Möbius. *Der barycentrische calcul (The barycentric calculation)*. Johann Ambrosius Barth Verlag, 1827.

[169] L.-P. Morency, I. de Kok, and J. Gratch. Predicting Listener Backchannels: A probabilistic multimodal approach. In *Intelligent Virtual Agents*, volume 5208, pages 176–190. Springer, 2008.

[170] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. In *Computer Graphics Forum*, volume 27, pages 1487–1494. Wiley Online Library, 2008.

[171] M. Müller. Dynamic time warping. In *Information retrieval for music and motion*, volume 2, pages 69–84. Springer, 2007.

[172] T. Murali and T. A. Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics (I3D)*, pages 155–162. ACM, 1997.

[173] A. V. Nefian and M. H. Hayes III. Face detection and recognition using hidden Markov models. In *Image Processing (ICIP 1998), International Conference on*, volume 1, pages 141–145. IEEE, 1998.

[174] F. S. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):191–205, 2003.

[175] D. G. Novick, B. Hansen, and K. Ward. Coordinating turn-taking with gaze. In *Proceedings of the 4th International Conference on Spoken Language (ICSLP 1996)*, volume 3, pages 1888–1891. IEEE, 1996.

[176] M. Nusseck, D. W. Cunningham, C. Wallraven, and H. H. Bülthoff. The contribution of different facial regions to the recognition of conversational expressions. *Journal of vision*, 8(8):1–23, 2008.

[177] C. Oertel, F. Cummins, J. Edlund, P. Wagner, and N. Campbell. D64: A corpus of richly recorded conversational interaction. *Journal on Multimodal User Interfaces*, 7(1-2):19–28, 2013.

[178] N. N. Oosterhof and A. Todorov. Shared perceptual basis of emotional expressions and trustworthiness impressions from faces. *Emotion*, 9(1):128–133, 2009.

[179] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Multimedia and Expo (ICME), 2005 IEEE International Conference on*, pages 317–321. IEEE, 2005.

[180] T. Papatheodorou and D. Rueckert. Evaluation of automatic 4D face recognition using surface and texture registration. In *Automatic Face and Gesture Recognition (FG 2004), IEEE International Conference on*, pages 321–326. IEEE, 2004.

[181] A. Patel and W. Smith. 3d morphable face models revisited. In *Computer Vision and Pattern Recognition (CVPR 2009), IEEE Conference on*, pages 1327–1334. IEEE, 2009.

[182] E. K. Patterson and A. Gaweda. Toward using dynamics of facial expressions and gestures for person identification. In *Proceedings of the 5th International Association of Science and Technology for Development (IASTED 2010) International Conference*, volume 711, pages 26–58, 2010.

[183] K. Person. On lines and planes of closest fit to system of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[184] G. Peyré. Graph theory toolbox. `http://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph`.

[185] G. Peyré and L. Cohen. Geodesic computations for fast and accurate surface remeshing and parameterization. In *Elliptic and Parabolic Problems*, pages 157–171. Springer, 2005.

[186] J. Podolak and S. Rusinkiewicz. Atomic volumes for mesh completion. In *Symposium on Geometry Processing*, pages 33–41. Citeseer, 2005.

[187] I. Poggi and C. Pelachaud. Performative facial expressions in animated faces. In J. Cassell, J. Sullivan, S. Prevost, and E. F. Churchill, editors, *Embodied Conversational Agents*, chapter 6, pages 155–189. MIT Press, 2000.

[188] H. Popat and S. Richmond. New developments in: Three-dimensional planning for orthognathic surgery. *Journal of Orthodontics*, 37(1):62–71, 2010.

[189] H. Popat, S. Richmond, D. Marshall, and P. L. Rosin. Three-dimensional assessment of functional change following Class 3 orthognathic correction – A preliminary report. *Journal of Cranio-Maxillofacial Surgery*, 40(1):36–42, 2012.

[190] C. Queirós, S. Alves, A. J. Marques, M. Oliveira, and V. Orvalho. Serious Games and Emotion Teaching in Autism Spectrum Disorders: A comparison with LIFEisGAME project. http://hdl.handle.net/10216/64635, 2012.

[191] M. Ratliff. Active appearance models for affect recognition using facial expressions. Master's thesis, University of North Carolina Wilmington, 2010.

[192] J. M. Rehg, G. D. Abowd, A. Rozga, M. Romero, M. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, et al. Decoding children's social behavior. In *Computer Vision and Pattern Recognition (CVPR 2013), IEEE Conference on*, pages 3414–3421. IEEE, 2013.

[193] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco. Connectionist probability estimators in HMM speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 2(1):161–174, 1994.

[194] G. Rigoll. Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems. *Speech and Audio Processing, IEEE Transactions on*, 2(1):175–184, 1994.

[195] E. L. Rosenberg, P. Ekman, and J. A. Blumenthal. Facial expression and the affective component of cynical hostility in male coronary heart disease patients. *Health Psychology*, 17(4):376–380, 1998.

[196] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[197] Rubybrian. Barycentric coordinates in a triangle. http://commons.wikimedia.org/wiki/File:TriangleBarycentricCoordinates.svg, 2008.

[198] O. Rudovic, M. Pantic, and I. Patras. Coupled gaussian processes for pose-invariant facial expression recognition. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 35(6):1357–1369, 2013.

[199] C. Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten (About empirical functions and the interpolation

between equidistant ordinates. *Zeitschrift für Mathematik und Physik (Journal for Maths and Physics)*, 46(224-243):20, 1901.

[200] M. Sagar, D. Bullivant, O. Efimov, M. Jawed, R. Kalarot, P. Robertson, and T.-F. Wu. Embodying models of expressive behaviour and learning with a biomimetic virtual infant. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pages 62–67. IEEE, 2014.

[201] M. Sagar, D. Bullivant, P. Robertson, O. Efimov, K. Jawed, R. Kalarot, and T. Wu. A neurobehavioural framework for autonomous animation of virtual human faces. In *SIGGRAPH Asia 2014 Autonomous Virtual Humans and Social Robot for Telepresence*, page 2. ACM, 2014.

[202] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697, 2012.

[203] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In B. Schouten, N. C. Juul, A. Drygajlo, and M. Tistarelli, editors, *Biometrics and Identity Management*, volume 5732, pages 47–56. Springer, 2008.

[204] A. Savran, B. Sankur, and T. M. Bilge. Estimation of facial action intensities on 2D and 3D data. In *Proceedings of the 19th European Signal Processing Conference (EUSIPCO 2011)*, pages 1969–1973. IEEE, 2011.

[205] D. C. Schneider and P. Eisert. Automatic and robust semantic registration of 3D head scans. In *Visual Media Production (CVMP 2008), 5th European Conference on*, pages 1–7. IET, 2008.

[206] D. C. Schneider, P. Eisert, J. Herder, M. Magnor, and O. Grau. Algorithms for automatic and robust registration of 3D head scans. *Journal of Virtual Reality and Broadcasting (JVRB)*, 7(7):1–15, 2010.

[207] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[208] N. Sebe, M. S. Lew, Y. Sun, I. Cohen, T. Gevers, and T. S. Huang. Authentic facial expression analysis. *Image and Vision Computing*, 25(12):1856–1863, 2007.

[209] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171, 2006.

[210] K. A. Sidorov, S. Richmond, and D. Marshall. An efficient stochastic approach to groupwise non-rigid image registration. In *Computer Vision and Pattern Recognition (CVPR 2009), IEEE Conference on*, pages 2208–2213. IEEE, 2009.

[211] K. A. Sidorov, S. Richmond, and D. Marshall. Efficient groupwise non-rigid registration of textured surfaces. In *Computer Vision and Pattern Recognition (CVPR 2011), IEEE Conference on*, pages 2401–2408. IEEE, 2011.

[212] L. I. Smith. A tutorial on principal component analysis. `http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf`, 2002.

[213] N. Smolyanskiy, C. Huitema, L. Liang, and S. E. Anderson. Real-time 3D face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860–869, 2014.

[214] Y. Sun and L. Yin. 3D spatio-temporal face recognition using dynamic range model sequences. In *Computer Vision and Pattern Recognition Workshops (CVPRW 2008), IEEE Conference on*, pages 1–7. IEEE, 2008.

[215] G. K. L. Tam, H. Fang, A. J. Aubrey, P. W. Grant, P. L. Rosin, D. Marshall, and M. Chen. Visualization of time-series data in parameter space for understanding facial dynamics. *Computer Graphics Forum*, 30(3):901–910, 2011.

[216] R. Tang, S. Halim, and M. Zulkepli. Surface reconstruction algorithms: review and comparisons. *The 8th International Symposium On Digital Earth (ISDE 2013)*, 2013.

[217] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[218] B.-J. Theobald, I. Matthews, M. Mangini, J. R. Spies, T. R. Brick, J. F. Cohn, and S. M. Boker. Mapping and manipulating facial expression. *Language and Speech*, 52(2-3):369–386, 2009.

[219] Y.-l. Tian, T. Kanade, and J. Cohn. Robust lip tracking by combining shape, color and motion. In *Proceedings of the 4th Asian Conference on Computer Vision (ACCV 2000)*, pages 1040–1045, 2000.

[220] D. Traum, S. C. Marsella, J. Gratch, J. Lee, and A. Hartholt. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Intelligent Virtual Agents*, pages 117–130. Springer, 2008.

[221] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[222] F. Tsalakanidou and S. Malassiotis. Real-time 2D+3D facial action and expression recognition. *Pattern Recognition*, 43(5):1763–1775, 2010.

[223] E. Tsankova, A. J. Aubrey, E. Krumhuber, G. Möllering, A. Kappas, D. Marshall, and P. L. Rosin. Facial and vocal cues in perceptions of trustworthiness. In *Proceedings of the 11th Asian Conference on Computer Vision Workshops (ACCVW 2012)*, volume 7729, pages 308–319. Springer, 2013.

[224] K. Tzevanidis, X. Zabulis, T. Sarmis, P. Koutlemanis, N. Kyriazis, and A. Argyros. From multiple views to textured 3D meshes: a gpu-powered approach. In *Trends and Topics in Computer Vision*, pages 384–397. Springer, 2012.

[225] R. Valkenburg and N. Alwesh. Seamless texture map generation from multiple images. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 7–12. ACM, 2012.

[226] M. Valstar and M. Pantic. Induced Disgust, Happiness and Surprise: An addition to the MMI facial expression database. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, Workshop on EMOTION*, pages 65–70, 2010.

[227] M. F. Valstar, M. Pantic, Z. Ambadar, and J. F. Cohn. Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *Proceedings of the 8th International Conference on Multimodal Interfaces (ICMI 2006)*, pages 162–170. ACM, 2006.

[228] G. Van Rossum. *Python tutorial*. Centrum voor Wiskunde en Informatica, 1995.

[229] J. Vandeventer, A. J. Aubrey, P. L. Rosin, and D. Marshall. 4D Cardiff Conversation Database (4D CCDb): A 4D database of natural, dyadic conversations. In *Proceedings of the 1st Joint Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing (FAAVSP 2015)*, 2015.

[230] J. Vandeventer, L. Gräser, M. Rychlowska, P. L. Rosin, and D. Marshall. Towards 4D coupled models of conversational facial expression interactions. In *Proceedings of the British Machine Vision Conference (BMVC 2015)*. British Machine Vision Association (BMVA), 2015.

[231] J. Vandeventer and E. Patterson. Differentiating Duchenne from non-Duchenne smiles using active appearance models. In *Biometrics: Theory, Applications and Systems (BTAS 2012), IEEE 5th International Conference on*, pages 319–324, 2012.

[232] J. Vandeventer, E. Patterson, B. Reinicke, and C. Guinn. Differentiating Duchenne from non-Duchenne smiles using active appearance models and the

Facial Action Coding System. Master's thesis, University of North Carolina Wilmington, 2012.

[233] R. Vertegaal. Conversational awareness in multiparty VMC. In *Extended Abstracts on Human Factors in Computing Systems (CHI 1997)*, pages 496–503. ACM, 1997.

[234] V. Vinayagamoorthy, M. Garau, A. Steed, and M. Slater. An eye gaze model for dyadic interaction in an immersive virtual environment: Practice and experience. In *Computer Graphics Forum*, volume 23, pages 1–11. Wiley Online Library, 2004.

[235] A. Vinciarelli, A. Dielmann, S. Favre, and H. Salamin. Canal9: A database of political debates for analysis of social interactions. In *Affective Computing and Intelligent Interaction and Workshops (ACII 2009), 3rd International Conference on*, pages 1–4. IEEE, 2009.

[236] E. P. Volkova, B. J. Mohler, T. J. Dodds, J. Tesch, and H. H. Bülthoff. Emotion categorization of body expressions in narrative scenarios. *Frontiers in Psychology*, 5:623:1–11, 2014.

[237] E. Vural, M. Bartlett, G. Littlewort, M. Cetin, A. Ercil, and J. Movellan. Discrimination of moderate and acute drowsiness based on spontaneous facial expressions. In *Pattern Recognition (ICPR 2010), 20th International Conference on*, pages 3874–3877. IEEE, 2010.

[238] C. Wallraven, M. Breidt, D. W. Cunningham, and H. H. Bülthoff. Evaluating the perceptual realism of animated facial expressions. *ACM Transactions on Applied Perception (TAP)*, 4(4):4:1–20, 2008.

[239] V. Wan, R. Anderson, A. Blokland, N. Braunschweiler, L. Chen, B. Kolluru, J. Latorre, R. Maia, B. Stenger, K. Yanagisawa, et al. Photo-realistic expressive text to talking head synthesis. In *Proceeding of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2667–2669, 2013.

[240] V. Wan and S. Renals. Speaker verification using sequence discriminant support vector machines. *Speech and Audio Processing, IEEE Transactions on*, 13(2):203–210, 2005.

[241] S. Wang, X. D. Gu, and H. Qin. Automatic non-rigid registration of 3D dynamic data for facial expression synthesis and transfer. In *Computer Vision and Pattern Recognition (CVPR 2008), IEEE Conference on*, pages 1–8. IEEE, 2008.

[242] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps. *International Journal of Computer Vision (IJCV)*, 76(3):283–300, 2008.

[243] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, pages 77:1–9. ACM, 2011.

[244] J. Whitehill, M. Bartlett, and J. Movellan. Measuring the perceived difficulty of a lecture using automatic facial expression recognition. 5091:668–670, 2008.

[245] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. ELAN: A professional framework for multimodality research. In *Language Resources and Evaluation (LREC 2006), 5th International Conference on*, pages 1556–1559, 2006.

[246] Y. Yang. 3D thin plate spline warping function. `http://www.mathworks.com/matlabcentral/fileexchange/37576-3d-thin-plate-spline-warping-function`.

[247] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *Automatic Face and Gesture Recognition (FG 2006), IEEE International Conference on*, pages 211–216. IEEE, 2006.

[248] V. H. Yngve. On getting a word in edgewise. In *Papers from the 6th Regional Meeting of the Chicago Linguistic Society*, pages 567–578, 1970.

[249] H. Yu, O. G. Garrod, and P. G. Schyns. Perception-driven facial expression synthesis. *Computers & Graphics*, 36(3):152–162, 2012.

[250] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, et al. A Survey of Affect Recognition Methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on*, 31(1):39–58, 2009.

[251] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime Faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*, pages 248–276. Springer, 2008.

[252] Q. Zhang, Z. Liu, G. Quo, D. Terzopoulos, and H.-Y. Shum. Geometry-driven photorealistic facial expression synthesis. *Visualization and Computer Graphics (TVCG), IEEE Transactions on*, 12(1):48–60, 2006.

[253] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3D dynamic facial expression database. In *Automatic Face and Gesture Recognition (FG 2013), IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.

[254] C. Zhou and X. Lin. Facial expressional image synthesis controlled by emotional parameters. *Pattern Recognition Letters*, 26(16):2611–2627, 2005.

[255] M. Zhou, L. Liang, J. Sun, and Y. Wang. AAM based face tracking with temporal matching and face segmentation. In *Computer Vision and Pattern Recognition (CVPR 2010), IEEE Conference on*, pages 701–708. IEEE, 2010.

[256] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *Visualization and Computer Graphics (TVCG), IEEE Transactions on*, 8(2):198–207, 2002.