Technical Report

UCAM-CL-TR-861
ISSN 1476-2986

Number 861

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Automatic facial expression analysis

Tadas Baltrusaitis

October 2014

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted March 2014 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Fitzwilliam College.

Some figures in this document are best viewed in colour. If you received a black-and-white copy, please consult the online version if necessary.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

Humans spend a large amount of their time interacting with computers of one type or another. However, computers are emotionally blind and indifferent to the affective states of their users. Human-computer interaction which does not consider emotions, ignores a whole channel of available information.

Faces contain a large portion of our emotionally expressive behaviour. We use facial expressions to display our emotional states and to manage our interactions. Furthermore, we express and read emotions in faces effortlessly. However, automatic understanding of facial expressions is a very difficult task computationally, especially in the presence of highly variable pose, expression and illumination. My work furthers the field of automatic facial expression tracking by tackling these issues, bringing emotionally aware computing closer to reality.

Firstly, I present an in-depth analysis of the Constrained Local Model (CLM) for facial expression and head pose tracking. I propose a number of extensions that make location of facial features more accurate.

Secondly, I introduce a 3D Constrained Local Model (CLM-Z) which takes full advantage of depth information available from various range scanners. CLM-Z is robust to changes in illumination and shows better facial tracking performance.

Thirdly, I present the Constrained Local Neural Field (CLNF), a novel instance of CLM that deals with the issues of facial tracking in complex scenes. It achieves this through the use of a novel landmark detector and a novel CLM fitting algorithm. CLNF outperforms state-of-the-art models for facial tracking in presence of difficult illumination and varying pose.

Lastly, I demonstrate how tracked facial expressions can be used for emotion inference from videos. I also show how the tools developed for facial tracking can be applied to emotion inference in music.

Acknowledgements

Many people have supported me during my time as a PhD student, and I would like to thank them all. This dissertation would not have been possible without the guidance of my supervisor, Peter Robinson. I thank him for his constant support and for giving me this wonderful opportunity. I am also very grateful to Louis-Philippe Morency who hosted me during my visit to the Institute for Creative Technologies. His unending energy helped me to stay motivated.

I would like to thank the Rainbow Group and the Computer Laboratory, which provided me with the necessary atmosphere for my work. I enjoyed the daily coffee breaks with my colleagues, especially Leszek, Marwa, Vaiva, Christian, Ntombi, and Ian. I also thank Graham Titmus, our ever helpful system-administrator. I am grateful to Alan Blackwell and Neil Dodgson for keeping me on track during my yearly reports.

My time at the Institute for Creative Technologies at the University of Southern California rekindled my interest in the dissertation topic. I would especially like to thank Julien-Charles, Sylwia, Dimitrios and Geovany for all the wonderful conversations we had.

This work could not have been possible without the financial support of Thales Research and Technology UK. I would like to thank Chris Firth and Mark Ashdown for funding my work and for their support.

My family provided me with an opportunity for growth and education and I am forever indebted to them. I am very grateful that they encouraged me to pursue my education and did not mind me going far away from home.

Finally, I am forever thankful to my soon-to-be wife, Rachael. She has painstakingly proof-read my dissertation and was always supportive and patient during deadlines and my time away from home.

Contents

1 Introduction

Computers are quickly becoming a ubiquitous part of our lives. We spend a great deal of time interacting with computers of one type or another. At the moment the devices we use are indifferent to our affective states. They are emotionally blind. However, successful human-human communication relies on the ability to read affective and emotional signals. Human-computer interaction (HCI) which does not consider the affective states of its users loses a large part of the information available in the interaction.

Recently, affective computing has been widely studied and there is a growing belief that providing computers with the ability to read the affective states of their users would be beneficial (Pantic et al., 2006; Picard, 1997; Robinson and el Kaliouby, 2009). It is believed that in order to make future progress in HCI it is necessary to recognise users' affect. This is informed by the importance of emotion in our daily lives (Cohn, 2006). Affective computing tries to bridge the gap between the emotionally expressive human and the emotionally deficient computer (D'Mello and Calvo, 2013).

There are many application areas that could benefit from the ability to detect affect. These range from interfaces that do not interrupt their users when they are stressed, online learning systems that adapt the teaching if the student is confused, and video games that adapt their difficulty based on the player engagement. Further applications include: assisted living environments that can monitor the users' state and report to medical professionals if the patient is feeling pain; assistive tech-

nologies for diagnosing conditions such as depression; and systems that monitor drivers or pilots for boredom.

Reliable automated recognition of human emotions is crucial before the development of affect sensitive systems is possible (Picard and Klein, 2001). Humans display affective behaviour that is multi-modal, subtle and complex. People are adept at expressing themselves and interpreting others through the use of non-verbal cues such as vocal prosody, facial expressions, eye gaze, various hand gestures, head motion and posture. All of these modalities convey important affective information that humans use to infer the emotional state of each other (Ambady and Rosenthal, 1992).

Out of these modalities, the face has received the most attention from both psychologists and affective computing researchers (Zeng et al., 2009). It is not very surprising as faces are the most visible social part of the human body. They reveal emotions (Ekman and Rosenberg, 2005), communicate intent, and help regulate social interaction (Schmidt and Cohn, 2001). Although not strictly part of the face, head gestures play an important part in human communication as well (Bavelas et al., 2000) and have been investigated for affect detection (Ramirez et al., 2011).

We can look at facial expressions and head gestures from two main perspectives - *message* and *sign* judgement. Message judgement approaches facial expressions in terms of meaning (emotion, intention, etc.), whereas sign judgement looks at the underlying anatomical structure and does not interpret the message. In order to achieve message judgements we need to be able to read the signs. My work mainly concentrates on sign judgement - reliably tracking faces and head pose, but I do present several case studies of message judgement.

Most of the outlined potential uses of affective computing rely fully, or at least partially, on the ability to automatically analyse human facial expressions. In order to do so, an ability to locate certain areas of the face and head pose is necessary. The approaches need to work outside the lab and *in the wild*: outdoor environments, dimly lit rooms, in

presence of harsh shadows, and various other noisy environments. The ideal facial tracker should also be person independent. Furthermore, to be of any use they have to be computationally efficient, especially if large scale monitoring, or analysis of large databases is needed. These requirements combine to present an extremely challenging task for computer vision.

In this dissertation I attempt to bring the state-of-the-art closer to being able to operate *in the wild*. I do this by extending the Constrained Local Model framework to work with depth data from various range scanners, thus reducing the effect of illumination and leading to better tracking. I also develop a novel patch expert that can learn complex non-linear relationships between pixel values and landmark locations leading to more accurate facial tracking, especially under difficult illumination conditions.

Finally, while the main goal of the research was to develop methods for more reliable tracking of faces, I also show how the tracked points can be used for emotion recognition, and how some of the methods developed can even be used in emotion prediction from music.

1.1 Contributions

The main contributions of my dissertation can be split into the following parts:

Constrained Local Model extensions

I present a number of extensions to the existing state-of-the art approach for facial landmark detection. These include a multi-modal, multi-scale formulation together with a novel fitting procedure. I demonstrate the benefits of these extensions on a number of publicly available datasets. Furthermore, I provide a detailed analysis of the issues the model faces.

3D Constrained Local Model

I present an extension of the Constrained Local Model paradigm to include depth information in addition to a regular visible light camera. This approach leads to more accurate and more robust fitting, helping to deal with bad lighting conditions. This was published as Baltrušaitis et al. (2012) in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), and awarded the publication of the year award by the Cambridge Computer Laboratory Ring.

Continuous Conditional Neural Field

I present a novel Continuous Conditional Neural Field model which is able to learn complex relationships between data and output. This model can model the complex non-linear relationships between pixel values and landmark location, and exploit their spatial relationships. This allows for more accurate and reliable face tracking, and is especially helpful for illumination invariant facial tracking. Furthermore, the flexibility of this model is demonstrated through its employment in the task of emotion prediction in music. The description of the model has been published in the *300 faces in-the-wild challenge* workshop in the IEEE International Conference on Computer Vision, 2013 (Baltrušaitis et al., 2013b).

Emotion inference in continuous space

I contribute to the growing body of methods for continuous dimensional emotion prediction from audio-visual data by employing a Continuous Conditional Random Field model, which reliably combines multiple modalities and exploits temporal characteristics of the emotional signal. This was published as Baltrušaitis et al. (2013a) at the IEEE International Conference on Automatic Face and Gesture Recogntition (FG).

1.2 Structure of the dissertation

I will begin with an overview of affective computing in Chapter 2. I will explain the underlying emotion theories and possible application areas. Special focus will be put on facial expressions and head pose.

Chapter 3 will give an overview of the datasets used to evaluate both rigid and non-rigid facial tracking.

In Chapter 4 I will provide a detailed explanation of the Constrained Local Model (CLM) approach to facial landmark detection, including some in-depth analysis of implementation details. I will also describe the several extensions I have developed.

An extension to CLM that uses depth information alongside regular visible light information will be presented in Chapter 5.

A novel Continuous Conditional Neural Field (CCNF) graphical model is introduced in Chapter 6. I will present a particular instance of CCNF which can be used as a novel patch expert that can learn complex non-linear relationships between pixel values and landmark locations. I will also demonstrate how this patch expert can be used to build a CLM landmark detector that outperforms current state-of-the-art detectors in most conditions.

Chapter 7 will demonstrate how such facial expression and head pose tracking can be used to infer emotions in dimensional space while exploiting the temporal properties of the emotional signal.

Another case study will be presented in Chapter 8. Here the CCNF model, developed for landmark detection, is used for emotion prediction in music, outperforming some state-of-the art approaches.

Finally, Chapter 9 will provide the concluding remarks of the dissertation and outline the current limitations together with future research directions.

1.3 Publications

1. Tadas Baltrušaitis, Laurel D. Riek, and Peter Robinson. **Synthesizing Expressions using Facial Feature Point Tracking: How Emotion is Conveyed**, in *ACM Workshop on Affective Interaction in Natural Environments*, October 2010

2. Tadas Baltrušaitis and Peter Robinson. **Analysis of Colour Space Transforms for Person Independent AAMs**, in *ACM / SSPNet 2nd International Symposium on Facial Analysis and Animation*, September 2010

3. Tadas Baltrušaitis, Daniel McDuff, Ntombikayise Banda, Marwa Mahmoud, Rana el Kaliouby, Rosalind Picard, and Peter Robinson. **Real-time inference of mental states from facial expressions and upper body gestures**, in *IEEE International Conference on Automatic Face and Gesture Recognition, Facial Expression Recognition and Analysis Challenge*, June 2011

4. Geovany A. Ramirez, Tadas Baltrušaitis, and Louis-Philippe Morency. **Modeling Latent Discriminative Dynamic of Multi - Dimensional Affective Signals**, in *1st International Audio/Visual Emotion Challenge and Workshop in conjunction with ACII*, October 2011 **(Winner of the video sub-challenge)**

5. Marwa Mahmoud, Tadas Baltrušaitis, and Peter Robinson. **3D corpus of spontaneous complex mental states**, in *International Conference on Affective Computing and Intelligent Interaction (ACII)*, October 2011

6. Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. **3D Constrained local model for rigid and non-rigid facial tracking**, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012

7. Marwa Mahmoud, Tadas Baltrušaitis, and Peter Robinson. **Crowdsourcing in emotion studies across time and culture**, in *Workshop on Crowdsourcing for Multimedia, ACM Multimedia*, October 2012

8. Tadas Baltrušaitis, Ntombikayise Banda, and Peter Robinson. **Dimensional Affect Recognition using Continuous Conditional Random Fields**, in *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, April 2013

9. Vaiva Imbrasaitė, Tadas Baltrušaitis, and Peter Robinson. **Emotion tracking in music using continuous conditional random fields and baseline feature representation**, in *ICME 2013 Workshop on Affective Analysis in Multimedia*, July 2013

10. Vaiva Imbrasaitė, Tadas Baltrušaitis, and Peter Robinson. **What really matters? A study into people's instinctive evaluation metrics for continuous emotion prediction in music**, in *International Conference on Affective Computing and Intelligent Interaction (ACII)*, September 2013

11. Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. **Constrained Local Neural Fields for robust facial landmark detection in the wild**, in *300 Faces in-the-Wild Challenge (300-W), IEEE International Conference on Computer Vision (ICCV)*, December 2013

2 Affective Computing

Affective computing was first popularised by Rosalind Picard's book *"Affective Computing"* which called for research into automatic sensing, detection and interpretation of affect and identified its possible uses in human computer interaction (HCI) contexts (Picard, 1997). Automatic affect sensing has attracted a lot of interest from various fields and research groups, including psychology, cognitive sciences, linguistics, computer vision, speech analysis, and machine learning. The progress in automatic affect recognition depends on the progress in all of these seemingly disparate fields.

Following the lead of Picard (1997) I use the terms *emotion*, *mental state* and *affective state* interchangeably, using them to refer to a dynamic state when a person experiences a feeling.

Affective computing has grown and diversified over the past decades. It now encompasses automatic affect sensing, affect synthesis and the design of emotionally intelligent interfaces. It is a field too broad to describe in detail in this dissertation, but I attempt to provide an overview of the field with an emphasis on affect sensing from facial expressions.

I first provide motivation for automatic affect inference, by giving examples of various application areas. I follow this by a brief overview of the main emotional theories, with special focus on facial expressions. This is followed by an overview of affect sensing and facial expression analysis techniques.

2.1 Application areas

There are a number of areas where the automatic detection and synthesis of affect would be beneficial. I give a number of examples of such potential systems, and outline some of the work that already uses automatic affect analysis.

Automatic tracking of attention, boredom and stress would be highly valuable in safety critical systems where the attentiveness of the operator is crucial. Examples of such systems are air traffic control, nuclear power plant surveillance, and operating a motor vehicle. An automated tracking tool could make these systems more secure and efficient, because early detection of negative affective states could alert the operator or others around him, thus helping to avoid accidents.

Affect sensing systems could also be used to monitor patients in hospitals, or when medical staff are not readily available or overburdened. It could also be used in assisted living scenarios to monitor the patients and inform the medical staff during emergencies. There are some promising developments in medical applications of affective computing. One such development is the automatic detection of pain as proposed by Ashraf et al. (2009). Another promising development is the automatic detection of depression from facial and auditory signals (Cohn et al., 2009).

Automatic detection of affect would not only benefit safety critical and medical environments, it also has its uses in the entertainment industry. One can imagine video games providing the players with a more tailored experience if the affective state of the player was known to the game. In addition, affective information could be used to augment limited channels of communication such as text messaging. One such system was developed by Höök (2009) and is called eMoto. It is a phone with an augmented SMS service where users, besides sending a text message, are allowed to choose its background from colourful and animated shapes. These backgrounds are supposed to represent emotional content along two axes of arousal and valence.

People with autism spectrum disorder have difficulty understanding the emotional states of others and expressing these states themselves (Baron-Cohen et al., 1985; Picard, 2009). Automatic recognition of affect could help autistic people to express their own affective states (Picard, 2009), by allowing them to express outwardly what is being felt inwardly. It could also be possible to build systems that help these people better understand the affective states of others.

In addition, affect synthesis is beneficial for creation of believable virtual characters (avatars) (Cassell, 2000), and robotic platforms (Riek and Robinson, 2011) as it allows these agents to act more like humans. Systems that are able to analyse affect can often be used to synthesise it if generative models are used, hence affect synthesis would benefit from better affect analysis.

Another possible application of automatic affect recognition is in furthering our understanding of human behaviour and emotions. These systems could be used to speed up the currently labour intensive, error prone and tedious task of labelling emotional data. Of special interest is work by Girard et al. (2013), in which automated tools for facial expression analysis (Action Unit detection) are used to support and inform existing theories of depression.

Another example of an affect system already being used is the work by Affectiva for automatic classification of content preference. McDuff et al. (2013) were successful in determining if people liked certain advertisements and were likely to watch them again by analysing their smiling behaviour. This work is potentially very useful for advertising and marketing domains, where new evaluation metrics are constantly sought. The authors collected a dataset in naturalistic environments by using the webcams of the visitors to their website. In total 6729 video segments were collected of people watching a number of selected advertisements. However, even though the authors were using a state-of-the-art tracker, in only 67% of the videos the majority of the frames were tracked successfully, demonstrating the need for facial trackers capable of coping with real life environments.

2.2 Emotions

Emotion research started with Charles Darwin about 140 years ago with his work *The Expression of The Emotions in Man and Animals* (Darwin, 1872). This created a lot of controversy at the time of its publication due to its contentious claim of universality of emotions and their evolutionary origins. Emotions have been a popular research topic ever since.

According to some researchers, emotions developed as an evolutionary advantage (Ekman, 1992). It is thought that emotions evolved for their adaptive value in fundamental life tasks (Ekman, 1992), i.e. that they make us act in a way that was advantageous over the course of evolution.

Affective states and their behavioural expressions are an important part of human life. They influence the way we behave, make decisions and communicate with others (Scherer, 2005). This is because our actions are influenced both by the affective state we are in and the affective states of people around us.

2.2.1 Theories of emotion

Before talking about the automatic detection of affect one has to understand what affect is. Unfortunately, psychologists themselves, have not reached a consensus on the definitions of *emotion* and *affect*. The three most popular ways that affect has been conceptualised in psychology research are as follows: discrete categories, dimensional representation, and appraisal-based. These theories are a good starting point to understanding affect for the purposes of automatic affect recognition as they provide information about the ways affect is expressed and interpreted.

Categorical

A popular way to describe emotion is in terms of discrete categories using the language from daily life (Ekman et al., 1982). The most popular example of such categorisation are the basic emotions proposed by Paul Ekman (Ekman, 1992). These are: happiness, sadness, surprise, fear, anger, and disgust. Ekman suggests that they have evolved in the same

Figure 2.1: Facial expressions of the six basic emotions - happiness, sadness, fear, anger, surprise and disgust, taken from Ekman and Friesen (1976).

way for all mankind and their recognition and expression is independent of nurture. This is supported by a number of cross-cultural studies performed by Ekman et al. (1982), suggesting that the facial expressions of the basic emotions are perceived in the same way, regardless of culture. Facial expressions representative of these emotions can be seen in Figure 2.1.

The problem of using the basic emotions for automatic affect analysis is that they were never intended as an exhaustive list of possible affective states that a person can exhibit Ekman et al. (1982). What makes them basic is their universal expression and recognition, amongst other criteria (Ekman, 1992). Finally, they are not the emotions that appear most often in everyday life (Rozin and Cohen, 2003).

Despite these shortcomings, basic emotions are very influential in automatic recognition of affect, as the majority of research has focused on detecting specifically these emotions, at least until recently (Zeng et al., 2009). However, there is a growing amount of evidence that these emotions are not very suitable for the purposes of affective computing, as they do not appear very often in HCI scenarios (D'Mello and Calvo, 2013).

There exist alternative, categorical representations that include *complex emotions*. An example of such a categorisation is the taxonomy developed by Baron-Cohen et al. (2004). It is a broad taxonomy including 24 groups of 412 different emotions. This taxonomy was created through a linguistic analysis of emotional terms in the English language. In addition to the basic emotions, it includes emotions such as boredom,

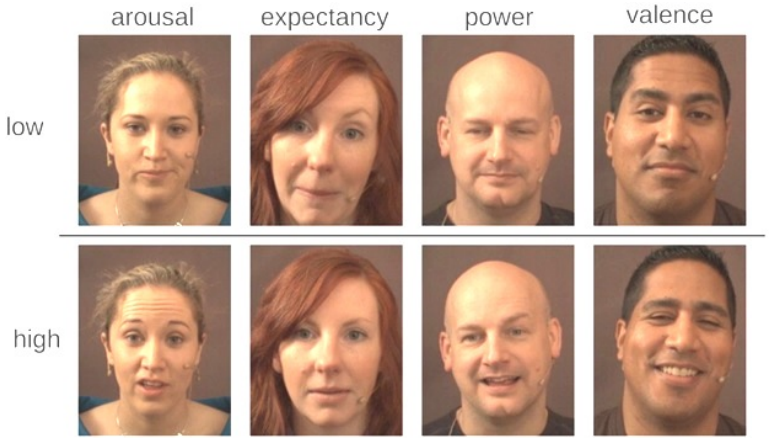


Figure 2.2: Facial expressions that could be attributed to certain values in the dimensional emotion space.

confusion, interest, frustration etc. The emotions belonging to some of these categories such as confusion, thinking and interest, seem to be much more common in everyday human-human and human-computer interactions (D'Mello and Calvo, 2013; Rozin and Cohen, 2003).

Baron-Cohen's taxonomy has been used by a number of researches in automatic recognition (el Kaliouby and Robinson, 2005; Sobol-Shikler and Robinson, 2010) and in description of affect (Mahmoud et al., 2011, 2012), however, it is not nearly as popular as the basic emotion categories. Complex emotions might be a more suitable representation, however, they lack the same level of underlying psychological research when compared to the six basic emotions. Furthermore, little is understood about the universality and cultural specificity of the complex emotions, although there has been some work to suggest the universality of some of them (Baron-Cohen, 1996).

Dimensional

Another way of describing affect is by using a dimensional representation (Russell and Mehrabian, 1977), in which an affective state is characterised as a point in a multi-dimensional space and the axes represent a small number of affective dimensions. These dimensions attempt to ac-

count for similarities and differences in emotional experience (Fontaine et al., 2007). Examples of such affective dimensions are: valence (pleasant vs. unpleasant); power (sense of control, dominance vs. submission); activation (relaxed vs. aroused); and expectancy (anticipation and appraisals of novelty and unpredictability). Fontaine et al. (2007) argue that these four dimensions account for most of the distinctions between everyday emotional experiences, and hence form a good set to analyse. Furthermore, there is some evidence of the cross-cultural generality of these dimensions (Fontaine et al., 2007). Facial expressions which could be associated with certain points in the emotional dimension space can be seen in Figure 2.2.

Dimensional representation allows for more flexibility when analysing emotions when compared to categorical representations. However, problems arise when one tries to use only several dimensions, since some emotions become indistinguishable when projecting high-dimensional emotional states onto lower dimension representations. For example, fear becomes indistinguishable from anger if only valence and activation are used. Furthermore, this representation is not intuitive and requires training in order to label expressive behaviour.

Affective computing researchers have started exploring the dimensional representation of emotion as well. It is often treated as a binary classification problem (Gunes and Schuller, 2013; Schuller et al., 2011) (active vs. passive, positive vs. negative etc.); or even as a four-class one (classification into quadrants of a 2D space). Treating it as a classification problem loses the added flexibility of this representation, hence there has been some recent work, treating it as a regression one (Baltrušaitis et al., 2013a; Imbrasaitė et al., 2013a; Nicolle et al., 2012).

Appraisal based

The third approach for representing emotion, and very influential amongst psychologists, is the appraisal theory (Scherer, 2005). In this representation, an emotion is described through the appraisal of the situation that elicited the emotion, thus accounting for individual differences. Unfor-

tunately this approach does not lend itself well for purposes of automatic affect recognition.

2.2.2 Affect expression and recognition

Humans express their affective states both consciously and unconsciously. Expressive behaviour is often unintended and in some cases even impossible to control. Furthermore, it is neither encoded nor decoded at an intentional, conscious level (Ambady and Rosenthal, 1992).

Most humans are competent mind readers and can attribute complex emotional states to other humans (Ambady and Rosenthal, 1992; Baron-Cohen, 1996). Although one's emotional state cannot be directly observed by another person it can be inferred from expressive behaviour. The modality of expressive behaviour varies. We reveal our affective states through our facial expressions and head gestures (Ekman et al., 1982). Our bodies reveal our emotional states as well, through various bodily postures and gestures (de Gelder, 2009) and even hand-over-face gestures (Mahmoud et al., 2011). The non-verbal features of our speech, such as prosody (Juslin and Scherer, 2005) also contain emotional information. In addition, we reveal our emotional states through various nonlinguistic vocalisations such as sighs, yawns and laughter (Cowie et al., 2001; Russell et al., 2003).

There have been various conflicting studies trying to measure the relative importance of different modalities (facial expressions, speech, posture) for conveying and interpreting affect. One such study, by Ekman et al. (1980), found that relative weight given to facial expression, speech, and body cues depends both on the judgement task and the conditions in which the behaviour occurs. Bugental et al. (1970) suggest that the influence of facial expression, as compared with other sources, depends on the expresser, the perceiver, the message contained in each channel, and previous experience. Another study, by de Gelder and Vroomen (2000), shows that one modality can influence the judgement of another. Furthermore, in a meta-analysis conducted by Ambady and Rosenthal (1992), it was suggested that accuracy of emotional judgement can de-

crease when multiple modalities are present (due to information overload). In addition, they suggested that people rely mostly on facial expressions when interpreting emotional states.

These studies provide a complex picture of the affective signals in different modalities and highlight the difficulties facing the multi-modal fusion of affective signals, especially in the case of conflicting emotional information. It is still an open research question as to what is the best approach for combining different modalities of expressive behaviour.

2.3 Facial expressions of emotion

The face is one of the most important channels of non-verbal communication. Facial expressions figure prominently in research on almost every aspect of emotion (De la Torre and Cohn, 2011). Facial expressions can have non-emotional information associated with them as well: they help with turn taking, convey intent, communicate culture-specific signals (for example winks), and are indicative of certain medical conditions, such as pain or depression. Unsurprisingly, this multi-faceted tool for expression and communication has interested researchers for centuries.

Facial expression of emotion has been a subject of scientific research for more than 150 years. Research began in the nineteenth century with *Mecanisme de la Physionomie Humaine* by the French neurologist Duchenne de Boulogne (Duchenne de Boulogne, 1862). Duchenne tried to identify the specific muscles representing specific emotions, such as the muscle of reflection and the muscle of aggression. His work represents a landmark in scientific writing – it was the first time that photography had been used to illustrate a series of experiments.

Duchenne's work was popularised by Charles Darwin in *The Expression of The Emotions in Man and Animals* (1872), in which photographs from Duchenne's experiments were published (see Figure 2.3 for some examples). Darwin used these photographs of facial expressions to find out if people agreed about the emotion shown by each expression. He showed

Figure 2.3: Example photographs of facial expressions captured by Duchenne and used in Darwin's experiments. The use of electrical probes (seen here being held by Duchenne and his assistant) helped keep the expression still for long enough, and to activate only particular muscles.

the photographs to friends during dinner parties and to a number of his naturalist colleagues. However, the use of electrically elicited facial expressions raises doubts about the accuracy and objectivity of the results. Nevertheless, his studies were cutting edge at the time, because of the use of external observers and *realistic* stimuli such as photographs.

A major step in the research on facial expressions came from Paul Ekman with his work on basic emotions (Ekman et al., 1982), and the Facial Action Coding System (FACS) (Ekman and Friesen, 1977). The latter made it possible for researches to analyse and classify facial expressions in a standardised framework since FACS allows one to encode all the possible, visually discriminable, facial expression combinations on the human face. As indicated by Cohn (2006), it is the most widely used system for the analysis of facial expressions to date.

There are two major approaches to the measurement of facial expressions. The first one is *message judgement* which assumes that the face is a *read out* of emotion, or some other social signal, and thus it should be interpreted as that by the observer. The second type of measurement is *sign judgement*, which assumes nothing about the semantics of the expressions and leaves inferences to higher order decision making (Cohn, 2006). I am more interested in sign judgement as it has broader applicability to various disciplines, including affective computing, psychology, and expression synthesis.

Message judgement attempts to describe expressions in terms of emotions they reveal. *Basic emotions* form the most popular *message* taxonomy. Basic emotions have specific facial expressions associated with them, for example anger is characterised by lowered eyebrows and tightened lips whereas surprise is characterised by raised eyebrows and open mouth (Ekman et al., 1982). Examples of facial expressions of basic emotions can be seen in Figure 2.1.

Universality (both in terms of recognition and expression) of basic emotions is supported by cross-cultural studies conducted by Ekman et al. (1982). More recently, Matsumoto and Willingham (2009) compared facial expressions of athletes in the 2004 Olympic and Paralympic games. They looked at the expressions of athletes after winning or losing a game. Interestingly, expressions shown did not differ between sighted, congenitally blind, and non-congenitally blind athletes. Moreover, the authors found no cultural differences.

There also exist facial signals that are not necessarily related to emotions and are more likely to be culturally specific and learned. One such signal is the eyebrow flash, which might indicate relevance, or help establish eye contact (Frith, 2009). Facial expressions can also reliably communicate physical pain (Prkachin and Solomon, 2008) and depression (Girard et al., 2013).

Together with head nods and eye-gaze, facial expressions are very important in human-human communication. Head nods and eyebrow raises act as illustrators - serving the function of emphasis during conversation (Ekman, 2004). They also act as regulators during the conversation – helping with initiation and termination of speech. Furthermore, they signal the speaker to continue, with what they are saying, through nods, agreement-smiles, forward leans, brow raises etc. (Ekman, 2004).

It is important to note that facial expression understanding is a context-dependent process. Aviezer et al. (2008) conducted studies demonstrating that the same facial expression can mean different things in different contexts (for example, anger may be confused with disgust). It is still an

Figure 2.4: This is an example of the Wollaston illusion. Even though the eyes are the same in both of the images, the perceived gaze direction is affected by the head orientation.

open research question how to incorporate context in expression recognition. Thus it is not enough to just label the expression, in order to get the bigger picture we need both the expression and context.

2.3.1 Head pose and eye gaze

Head pose and eye gaze play a role in expressing affect and communicating social signals. From a computational point of view it sometimes makes sense to treat them together with facial expression, as they all occur in the same place – the human head. Hence, I provide a brief overview of affective and social signals conveyed by these modalities.

Head pose is important when detecting certain emotional states such as interest, where the tilting of the head is important (Ekman et al., 1982). Head pose together with facial expressions also plays a role in the expression of pride and shame (Tracy and Matsumoto, 2008). Furthermore, the expression of embarrassment is accompanied by gaze aversion, downward head motion and a nervous smile (Keltner, 1995), demonstrating the importance to analyse these modalities together.

As mentioned before head nods can act as illustrators and regulators during conversation. In addition, head movements of a listener during a dyadic interaction signal 'yes' or 'no', indicate communicative intentions and help with the synchronisation of interactional rhythm (Hadar et al.,

1985). Finally, head direction and eye gaze are also used to indicate the target of a conversation.

Gaze direction is important when evaluating things like attraction, attentiveness, competence, social skills and mental health, as well as intensity of emotions (Kleinke, 1986). In order to estimate gaze direction, however, we also need to compute head orientation. Lastly, head orientation is used for eye gaze interpretation as well, as demonstrated in Figure 2.4.

2.4 Facial affect analysis

The previous section outlined how affect can be expressed through facial expressions, head pose and other non-verbal signals. These signals are easily understood by humans and there has been much progress in making them readable by computers as well. In this section I outline the work done on automated affect recognition from facial expressions and head pose.

Automatic facial expression analysis has been of interest to researches for over 30 years (Suwa et al., 1978). Most of the initial attempts built systems that relied on very restricted conditions. Faces had to be frontal or profile, in controlled lighting conditions, and the system often had to know the location of the face or facial landmarks (Samal and Iyengar, 1992). The types of facial expressions analysed were also mainly restricted to acted and exaggerated basic emotions. A huge amount of progress has been made in the field of automatic facial expression analysis since then.

The first development was in the type of data analysed. Instead of looking at still images there has been a move to analyse much richer image sequences (el Kaliouby and Robinson, 2005; Ramirez et al., 2011; Zeng et al., 2009). In order to be successful at exploiting such complex temporal signals a number of new statistical learning models have been developed (Lévesque et al., 2013; Song et al., 2012). In addition, there has been a recent move to not only look at the visible light sig-

nal (greyscale, RGB, etc.) but also to use 3D information available from various range scanners (Sandbach et al., 2012). This move has been motivated by the difficulty of dealing with varying illumination in visible light signals. Most of the research so far has concentrated on posed expressions collected using high end scanners (Sandbach et al., 2012). However, datasets of naturalistic expressions of emotion are becoming available as well (Mahmoud et al., 2011; Zhang et al., 2013).

The second major development has been a shift from posed data to evoked or natural expressions. This is a very important step as spontaneous facial expressions of emotion differ from acted and deliberate ones in several ways: onset and offset speed; amplitude of movement; and offset duration (Schmidt et al., 2006; Valstar et al., 2006, 2007). This means that systems trained on posed data might not generalise to spontaneous expressions. In order to achieve generalisation two developments are required. Firstly, the collection of naturalistic datasets. There is a growing number of such datasets, including SEMAINE (McKeown et al., 2010), parts of MMI (Valstar and Pantic, 2010), CK+ (Lucey et al., 2010), and Cam3D (Mahmoud et al., 2011). Secondly, the progress of computer vision and machine learning techniques which can deal with such unconstrained data (more on this in Section 2.5).

A fairly recent trend has started to look at facial expressions beyond the affect they express. One such area is the automatic detection of facial Action Units (AUs) (Valstar, 2008). It analyses the signal conveyed by the expression, not the intended message. The potential benefits of such an approach is that AU detections can later be used for inference of emotional state (Baltrušaitis et al., 2011; el Kaliouby and Robinson, 2005), and medical conditions such as depression (Girard et al., 2013). Furthermore, such an approach allows one to avoid the difficulties of context-dependent or culture-specific expressions.

A large amount of progress has been made recently due to automatic affect recognition competitions. The first of which was the FERA challenge for AU and basic emotion recognition (Valstar et al., 2011), this has been followed by the three Audio/Visual Emotion Challenges (AVEC)

for prediction of emotion in the dimensional space by using multi-modal signals (Schuller et al., 2011, 2012). These competitions both develop the state-of-the-art and make the comparisons between approaches easier.

One thing in common with most of the above outlined systems is their reliance on facial landmark detection. The facial landmarks can be used directly for affect recognition (Jeni et al., 2012) or in conjunction with appearance based features (such as Local Binary Patterns, Gabor Wavelets, and SIFT features). The advantage of using landmark locations directly is the ease of their temporal analysis - how much the eyebrow moves, the speed of onset and offset. However, the accurate location of landmarks is also necessary when using appearance based features, as they rely on face registration to a common reference frame (Chew et al., 2011).

2.5 Facial tracking

I use the term facial tracking as an umbrella term to encompass facial landmark detection, facial landmark tracking and head pose estimation. *Facial landmark detection* refers to locating a certain number of points of interest in an image of a face. *Facial landmark tracking* refers to the tracking of a set of interest points in an image sequence, by either treating each frame in a sequence as independent, or using temporal information. *Head pose estimation* attempts to compute the location and orientation of the head either from a single image or an image sequence. All of these problems are related, and some trackers are able to deal with all of them at once. However, historically these problems were often treated separately. This section provides an overview of the existing facial tracking approaches.

2.5.1 Landmark detection and tracking

Facial landmark tracking is sometimes called non-rigid tracking, as a face is a highly non-rigid object. It is also sometimes called face alignment and face registration. There are three main motivations that spurred research in facial landmark detection and tracking: affective comput-

ing, facial recognition and performance-driven animation (Metaxas and Zhang, 2013; Pantic and Bartlett, 2007; Zeng et al., 2009; Zhao et al., 2003). All of these fields rely on accurate landmark detection. It is important for affective computing and facial recognition as the landmark locations can be used as features, help with face segmentation, and provide locations where appearance features can be computed. For the case of performance-driven animation, the features have to be tracked accurately in order to create believable and realistic animations.

Arguably, the most popular approaches are various deformable model based ones, as they show good results for landmark detection and tracking (Gao et al., 2010). Such approaches include Active Shape Models (Cootes and Taylor, 1992); Active Appearance Models (Cootes et al., 2001); 3D Morphable Models (Blanz and Vetter, 1999); and Constrained Local Models (Cristinacce and Cootes, 2006). A more detailed discussion of these approaches can be found in Section 4.1.1.

There are few approaches which attempt to detect and track facial landmarks using depth data[1], instead of just visible light[2] images. Several approaches use Iterative Closest Point like algorithms for landmark detection and tracking on depth images (Breidt et al., 2011; Cai et al., 2010). Breidt et al. (2011) use depth information to fit an identity and expression 3D morphable model. Cai et al. (2010) use the intensity to guide their 3D deformable model fitting. Another noteworthy example is that of Weise et al. (2011), in which a person-specific deformable model is fit to depth and texture streams for performance based animation.

A slightly different approach to landmark detection uses explicit regression to correct for landmark detection estimates. Such an approach uses a regressor to predict the shape, instead of fitting a model. This avoids the construction of a loss function that is minimised by deformable model based approaches. This approach has been used in the facial

[1] By depth data I refer to scene geometry, or depth images where pixels represent distance to the object, usually acquired through various range scanners or stereoscopy

[2] I use the term visible light to refer to RGB, greyscale intensity images or any of their transformations, such as gradients of greyscale images

point detector by Valstar et al. (2010), in which regressor estimates are combined with a probabilistic graph-based shape model. The shape model is evaluated after each iteration, correcting the predicted point locations, if they do not form a consistent face shape. This is similar to Cao et al. (2012) who use explicit shape regression to minimise the detection error directly.

The above-mentioned approaches are mainly used for landmark detection in images and not tracking. Most of them can be easily converted to landmark trackers by simply reinitialising the detection procedure in the subsequent frame by using the current estimates. Alternatively, various trackers could be used as well, often by first initialising them with landmark detectors (Liwicki and Zafeiriou, 2011; Patras and Pantic, 2004).

2.5.2 Head pose tracking

Head pose estimation is sometimes referred to as rigid tracking, as often the head is treated as a rigid object. These techniques can be grouped based on the type of data they work on: *static*, *dynamic* or *hybrid*. Static methods attempt to determine the head pose from a single intensity or depth image, while dynamic ones estimate the object motion from one frame to another. Static methods are more robust, while dynamic ones show better overall accuracy, but are prone to failure during longer tracking sequences due to accumulation of error (Murphy-Chutorian and Trivedi, 2009). Hybrid approaches attempt to combine the benefits of both static and dynamic tracking.

Recent work also uses depth for static head pose detection (Breitenstein et al., 2008; Fanelli et al., 2011a,b). These approaches are promising, as they are not sensitive to changing illumination. However, they could still benefit from additional temporal information.

2.5.3 Combined landmark and head pose tracking

Recently, approaches that combine head pose estimation together with feature point tracking have become more popular. There have been

several extensions to Active Appearance Models that explicitly model the 3D shape in the formulation of the point distribution model (Xiao et al., 2004), or train several types of models for different view points (Cootes et al., 2000). These approaches show better performance for feature tracking at various poses, but still suffer from low accuracy in estimating the head pose. CLM can also be easily extended for the purpose of rigid and non-rigid tracking by using a 3D point distribution model (Saragih et al., 2011).

More recently, Zhu and Ramanan (2012) have demonstrated the accuracy and robustness of tree-structured models at detection of faces, together with pose estimation and landmark localisation. They show good results on standard benchmarks, as well as on an *in the wild* dataset. However, their approach is very slow (up to 40 seconds per image), and is not suitable for affect analysis in its current state.

3 Facial expression and head pose datasets

This section provides a description of the facial expression and head pose datasets I used throughout the dissertation. They played a major role in evaluating my proposed methods. I used some of the datasets to evaluate facial landmark detection and others for facial tracking evaluation. All of these datasets are available publicly, ensuring the reproducibility of the results. However, some of them have been adapted to fit my experimental needs.

The datasets can be split into two types: image and image sequence. The image datasets are used to evaluate the accuracy of landmark detection methods. The image sequence (video) datasets are used to evaluate facial tracking - head pose estimation and landmark tracking in a sequence.

3.1 Image datasets

I used two image-based datasets for the evaluation of landmark detection algorithms. The first one is the Carnegie Mellon University Multi-PIE (pose, illumination, expression) dataset (Gross et al., 2008), which will be referred to as Multi-PIE. The second one is a subset of the Binghamton University - 4D Facial Expression dataset (Yin et al., 2008), referred to as BU-4DFE. Both of the datasets are used for algorithm training and form the basis for their evaluation.

(a) Camera locations (b) Sample images from different cameras

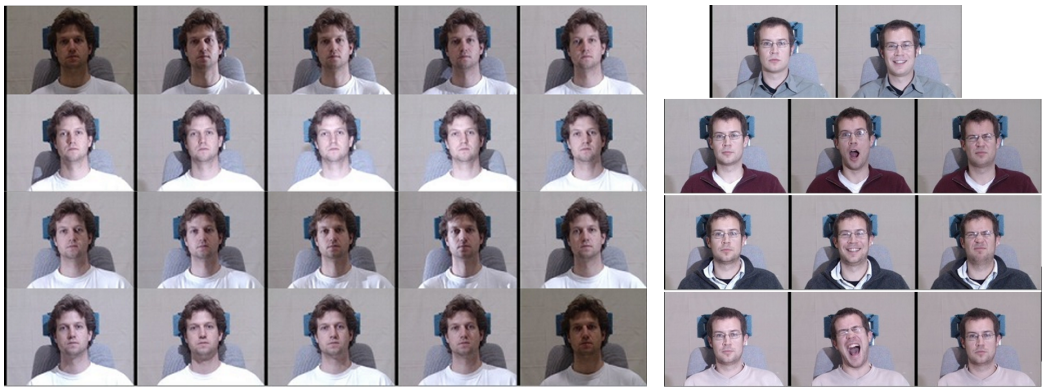
Figure 3.1: Multi-PIE dataset pose variations.

3.1.1 Multi-PIE

The Multi-PIE dataset (Gross et al., 2008) is one of the most extensive datasets of facial images across pose, expression and illumination. It consists of more than 750,000 images of 337 people recorded in up to four sessions over the span of five months. Photographs of subjects were taken from 15 view points and under 19 illumination conditions while displaying a range of facial expressions. The dataset was originally intended for use in face recognition, as it contains images of the same person under different illuminations, poses, and expressions. However, it also proved very popular for evaluating facial landmark detection and facial expression analysis techniques.

The subjects in this dataset were captured using 15 cameras at once, whilst going though 19 predefined illumination conditions. This led to 285 images of the same person with the same expression, but at different poses and under different illumination. Camera locations can be seen in Figure 3.1a, and sample images taken by these cameras in Figure 3.1b.

In addition to multiple poses, Multi-PIE dataset consists of faces from multiple lighting conditions, as seen in Figure 3.2a. Having access to the same expressions under different lighting conditions allowed me to check how well certain approaches cope with different and unseen illumination. Moreover, it allowed me to train landmark detection systems that can work across varying lighting conditions.



(a) Variation in illumination (b) Variation in expressions

Figure 3.2: The available lighting and expression variations across the Multi-PIE dataset. Notice how the same expression appears under different illuminations in Figure 3.2a.

As it is a very big dataset, only a subset of the images have been labelled for facial feature points. I had access to 5874 such manual labels. These labels consist of 5060 fully frontal (or close to frontal images), and 814 images at ± 15, ± 30, ± 45, ± 60, ± 75, ± 90 degrees of yaw. This number was doubled for training purposes by considering mirrored images as well.

An attractive property of the Multi-PIE dataset is that the landmark locations do not change across the illumination conditions, with the exception of possible eye narrowing or blinking following a flash. This allowed me to reuse the ground truth labels from one lighting condition on the others. In this dissertation I explored 4 lighting conditions: frontally lit face, left side lit face, right side lit face, and poorly lit face (Figure 3.3). I restricted myself to 4 out of 19 lighting conditions to save space and reduce computational complexity.

The 5874 frontally lit faces are referred to as *frontal illumination* Multi-PIE, the 17622 side and poorly lit faces *difficult illumination* Multi-PIE, and the combined 23496 images *general illumination* Multi-PIE.

Finally, for model training and testing purposes the datasets were split into training and testing partitions, with a quarter of subjects allocated

(a) Frontally lit face (b) Poorly lit (c) Left lit (d) Right lit

Figure 3.3: Face images under varying illumination from the Multi-PIE dataset. The left most image indicates an easy lighting condition: frontally lit face; while the other three more difficult ones: dimly lit face, and two face images with a strong side light.

to training and three quarters to testing. Since I am interested in person independent facial tracking, I ensured that the same subject never appeared in both training and testing.

3.1.2 BU-4DFE subset

Binghamton University 4D (3D + time) Facial Expression (BU-4DFE) database is a 3D dynamic facial expression database (Yin et al., 2008). It consists of 3D video sequences of 101 subjects acting out one of the six basic emotions from neutral to apex, and back to neutral. It was collected using the Di3D[1] dynamic face capturing system, which records sequences of texture images together with 3D models of faces.

In my work I did not use BU-4DFE as a video dataset, but took only a subset of the available video frames and used them as a still image dataset. I chose to do this because labelling videos for facial landmark positions would have been a very labour intensive task, but it was manageable for a smaller subset of images.

I took a subset of 707 frames (each participant with neutral expression and peaks of the 6 basic emotions) and labelled the images with 66 feature points semi-automatically. At first the landmarks were detected using the CLM facial tracker by Saragih et al. (2011), followed by a manual inspection and correction of landmark locations. I discarded some

[1] http://www.di3d.com (accessed Apr. 2012)

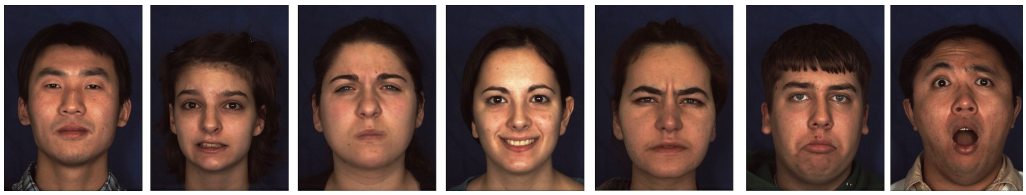


Figure 3.4: Sample of images extracted from the BU-4DFE video dataset.

of the frames because of poor coverage, either by the range scanner, or the cameras (part of the chin missing etc.). Some of the frames were discarded because the corresponding 3D models were too noisy (based on a manual inspection). This led to a total of 554 images together with their corresponding 3D models. In this dissertation, the reduced dataset is referred to as BU-4DFE. Some samples of extracted images can be seen in Figure 3.4.

Synthetic data generation

The great advantage of the BU-4DFE dataset is that each of the colour images has a corresponding 3D model of face geometry (as a triangulated point cloud). This means it is easy to manipulate the dataset to provide more training data for facial landmark detection algorithms. This section outlines the steps I took to generate the extra training data from only a limited number of labelled images.

Firstly, it was possible to generate extra texture images at various poses. This was done by aligning them to a reference frame using a statistical shape model (Section 4.2.5). Then the 3D model was rotated to a different view, where it was rendered using the available texture information. The following orientations (roll, yaw, pitch) were used: $(\pm 75, 0, 0)$; $(\pm 45, 0, 0)$; $(\pm 20, 0, 0)$; $(0, 0, 0)$; $(0, 0, \pm 30)$; $(0, 0, 30)$. Examples of such synthetic images can be seen in Figure 3.5. The advantage of such data generation is that the landmark labels are consistent across pose, which is often difficult to achieve via manual labelling of each pose (especially for the face outline). However, this technique introduces some artifacts due to missing data from the range scanner.

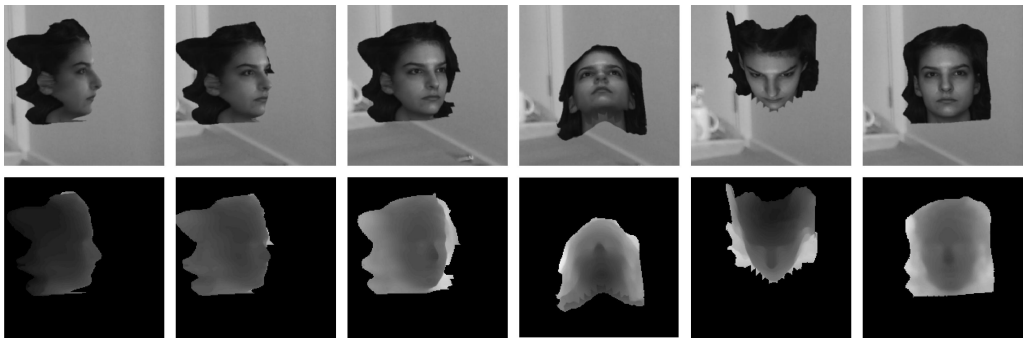


Figure 3.5: Sample of synthetically generated intensity and depth images from BU-4DFE at various poses. For the intensity images a basic background was simulated from various images in order to avoid overfitting during training.

Figure 3.6: Sample of synthetically generated intensity images with slight pose variations and varying lighting conditions: left, right and frontal illumination.

The second advantage of the availability of 3D data, was the ability to generate synthetic images at various illuminations. This was achieved by rendering the 3D scenes with a light source at different positions. Four lighting conditions were generated: frontal, left, right and poorly lit. The rendering was done with the freeglut OpenGL library using the 3D data, normals and texture from the BU-4DFE dataset. Examples of synthetic images of faces under different illuminations can be seen in Figure 3.6.

Finally, access to 3D models allowed for the creation of synthetic depth images similar to those that would be expected from various range scanners (such as Microsoft Kinect or Time-of-flight sensors). They were also rendered at various poses, leading to training data that was used for my experiments. Examples of depth images generated can be seen in Figure 3.5.

Figure 3.7: Sample stills from one of the ICT-3DHP sequences.

All of the synthetic greyscale images, together with the depth images, were used for landmark detector training. In order to avoid overfitting, the dataset was split similarly to Multi-PIE, with a quarter of subjects reserved for training and three quarters for testing (only non-synthetic images were used for testing). Lastly, in all of the experiments the same subject never appeared in both training and testing.

3.2 Image sequence datasets

Three image sequence (video) datasets were used for facial tracking evaluation. One dataset was used to evaluate landmark detection in a sequence (tracking), while three datasets were used for evaluating head pose estimation accuracy. Head pose estimation accuracy can also be seen as a proxy metric for landmark detection accuracy. For example, if the correct head pose is estimated from the detected landmarks, it is very likely that the landmarks were detected accurately as well.

3.2.1 ICT-3DHP

One of the head pose datasets was collected by myself, using the Microsoft Kinect sensor. The dataset contains 10 image sequences with both colour and depth information (RGBD), of around 1400 frames each. It is publicly available for research purposes[2].

The head pose of the individual in each video was labelled using a

[2] http://projects.ict.usc.edu/3dhp/ (accessed August 2013)

(a) RGB images (b) Range data

Figure 3.8: Sample images from the Biwi Kinect Head Pose database

Polhemus Fastrak magnetic position tracker[3]. The tracker consists of a System Electronics Unit which generates and senses the magnetic fields and computes the position and orientation of a sensor with respect to a source unit. The sensor tracked by the system was attached to a baseball cap worn by each participant. The Fastrack system tracks the position and orientation of a small sensor as it moves through space using electromagnetic fields. It demonstrates good accuracy in position - 1.4mm in RMSE, and orientation 0.12° in RMSE at the distance of 1.2 meters from the transmitter. It also has a very high update rate - 120Hz, and low latency - 4ms.

The dataset was recorded in an office environment with unconstrained lighting conditions. The person sitting in front of the camera was instructed to move their head in various ways. Even though this is not a naturalistic dataset, it is still very useful for assessing the accuracy of head pose estimation algorithms. It involves large head motions ranging from ±45° roll, ±75° yaw and ±40° pitch. Samples from one of the sequences can be seen in Figure 3.7.

3.2.2 Biwi Kinect Head Pose

Biwi Kinect Head Pose Database (Fanelli et al., 2011b) is another head pose dataset used in my work. It contains over $15k$ frames of 20 people (6 females and 14 males - with 4 people recorded twice) recorded with a Microsoft Kinect sensor while moving their heads around freely. For each frame, depth and colour images are provided. Sample frames from the dataset can be seen in Figure 3.8.

The head pose was annotated using an automated system that relies on person specific face range scans[4]. This resulted in ground truth in the form of the 3D location of the head and its rotation angles. The head pose ranges from $\pm 75^\circ$ yaw and $\pm 60^\circ$ pitch. The dataset was collected to evaluate a static head pose algorithm proposed by Fanelli et al. (2011b).

The frames from the database were converted to 24 sequences of RGBD images (same format as ICT-3D HP dataset). Since the Biwi Kinect Head Pose database was collected with frame-by-frame estimation in mind the resulting image sequences had a number of frames missing. This made the modified Biwi Kinect Head pose database very difficult for tracking based approaches as they rely on temporal information. Because the approaches used in this dissertation are all tracking based, this dataset allowed me to stress test them.

Biwi for features

In order to create a labelled facial feature point dataset of RGBD sequences I hand-labelled a subset of the Biwi Kinect Head Pose database. I chose 4 sequences of 772, 572, 395, and 634 frames each and manually labelled every 30th frame of those sequences with 66 feature points for frontal images and 37 feature points for profile images. This led to 82 labelled images in total. This is a particularly challenging dataset for a feature point tracker due to large head pose variations ($\pm 75^\circ$ yaw and $\pm 60^\circ$ pitch) and missing frames.

[3] http://www.polhemus.com/?page=Motion_Fastrak (accessed August 2013)
[4] www.faceshift.com

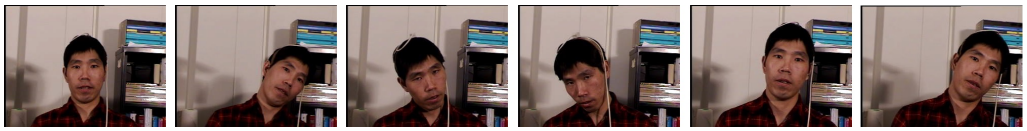


Figure 3.9: Sample stills from one of the Boston University head pose dataset sequences. Note the visible wire that is connecting the flock of birds tracker to the receiver.

3.2.3 Boston University head pose dataset

Lastly, the Boston University head pose dataset was used (Cascia et al., 2000). It contains 45 video sequences, from 5 different people, with 200 frames each. The dataset was labelled using an Ascension Technology "Flock of Birds" tracker (similar tracker to that used for the ICT-3DHP dataset). In each of the sequences a participant moved their head around freely. Sample stills from one of the sequences in the Boston University dataset can be seen in Figure 3.9.

4 Constrained local model

4.1 Introduction

A crucial initial step in many affect sensing, face recognition, and human behaviour understanding systems, is the estimation of head pose and detection of certain facial feature points. The detection of eyebrows, corners of eyes, and lips allows us to analyse their structure and motion. Furthermore, it helps with face alignment for appearance based analysis.

Facial landmark detection is a very difficult problem for several reasons. Firstly, the human face is a non-rigid object; its shape is affected by identity and facial expressions. Secondly, facial appearance is highly affected by lighting conditions; skin tone; facial hair; and various accessories (glasses, hats, scarves etc.). Furthermore, people tend to move their heads when interacting (both as a social signal and general fidgeting), leading to self-occlusion. A certain amount of occlusion also occurs due to hand-over-face gestures which are prevalent during natural communication and interaction with computers (Mahmoud et al., 2011; Pease and Pease, 2006). All of the above means that an algorithm capable of tracking non-rigid objects, with variable shape and appearance, is needed. The algorithm must also cope with a considerable amount of out-of-plane motion, occlusion and lighting variation.

4.1.1 Deformable model approaches to facial tracking

Approaches based on deformable models are commonly used for the task of landmark registration. Notable examples include Active Shape

Models (ASM) (Cootes and Taylor, 1992); Active Appearance Models (AAM) (Cootes et al., 2001); 3D Morphable Models (3DMM) (Blanz and Vetter, 1999); and Constrained Local Models (CLM) (Cristinacce and Cootes, 2006). The problem of fitting a deformable model involves finding the parameters of the model that best match a given image.

All of the above mentioned approaches depend on a parametrised shape model, which controls the possible shape variations of the non-rigid object (see Figure 4.5 for an example of a shape model). The approaches, however, differ in the way they model object appearance. Approaches based on AAM and 3DMM model the appearance holistically (the whole face together), whereas approaches based on CLM and ASM model the appearance in a local fashion (each feature point has its own appearance model).

Given a shape and appearance model, a deformable model fitting process is used to estimate the parameters that could have produced the appearance of a face in an unseen image. The parameters are optimised with respect to an error term which depends on how well the parameters model the appearance of a given image, or how well the current points represent an aligned model. There are two ways of finding the optimal parameters. The first directly minimises an error function through various specialised or general optimisation techniques (Cootes and Taylor, 1992; Cootes et al., 2001; Cristinacce and Cootes, 2006; Saragih et al., 2011; Wang et al., 2008). The second trains a regressor to estimate the model parameter update based on the current state (Cristinacce and Cootes, 2007; Fanelli et al., 2013; Saragih and Goecke, 2007; Sauer et al., 2011). Both of these methods usually employ a regularisation term that penalises complex shapes.

One of the most promising deformable models is the Constrained Local Model (CLM) proposed by Cristinacce and Cootes (2006), and various extensions that followed (Gu and Kanade, 2008; Saragih et al., 2011; Wang et al., 2008). Recent advances in CLM fitting and construction have led to good results in terms of accuracy, convergence rates, and real-time performance in the task of person-independent facial feature

tracking. It has outperformed various instances of AAM and ASM for the task of facial expression tracking (Gu and Kanade, 2008; Saragih et al., 2011; Wang et al., 2008).

There are several naming conventions in the CLM literature, the original term Constrained Local Model was coined by Cristinacce and Cootes (2006), but others have used it to refer to a more general problem (Saragih et al., 2011). In my work, CLM refers to a deformable shape model that follows a statistical shape distribution and models the local appearance of each feature point with the help of a local detector (patch expert).

CLM is able to achieve more generalisable fitting by modelling the appearance of each feature separately (Saragih et al., 2011). This is partly because CLM models the fact that multiple people can share similar local features, e.g. nose, eyebrows etc., while having other features that differ. Furthermore, due to the independent modelling, CLM demonstrates robustness in the presence of uneven lighting and occlusion. For example, a strong shadow on the left side of the face would not affect fitting of the right side. For these reasons local description based approaches have become more popular than holistic ones. However, there has been some work done recently on creating generic Active Appearance Models which are able to deal with person independence (Tzimiropoulos et al., 2012). Unfortunately, their generality is still lacking when compared to CLM based approaches (see Section 6.4.1 for a comparison).

For the reasons discussed, I chose CLM as a starting point for my work. Even though there has been a lot of progress in CLM construction and fitting techniques, many open questions still remain. My work on CLM has led to improved facial tracking, which can be used for emotion recognition.

4.1.2 Problem formulation

A CLM consists of two parts: a statistical shape model and patch experts (also called local detectors). Both the the shape model and patch

Initial | After 2 iterations | After 6 iterations | After 18 iterations

Figure 4.1: An example of initialising a deformable model, and iteratively fitting it until convergence. Taken from Cootes and Taylor (1992).

experts can be trained offline and then used for online landmark detection, which is achieved by fitting the CLM to a given image.

The deformable model is controlled by parameters $\mathbf{p}$ and the instance of a model can be described by the locations of its feature points $\mathbf{x}_i$ in an image $\mathcal{I}$ (usually a greyscale image, but other possible types are described in Section 4.3.2). The CLM fitting algorithms attempt to find the value of $\mathbf{p}$ that minimises the following energy function:

$$\mathcal{E}(\mathbf{p}) = \mathcal{R}(\mathbf{p}) + \sum_{i=1}^{n} \mathcal{D}_i(\mathbf{x}_i; \mathcal{I}). \tag{4.1}$$

$\mathcal{R}$ represents the regularisation term (smoothness term) which penalises overly complex or unlikely shapes, and $\mathcal{D}$ represents the amount of misalignment the i^{th} landmark is experiencing at location $\mathbf{x}_i$ in the image (data term). The value of $\mathbf{x}_i$ is controlled by the parameters $\mathbf{p}$ through the shape model that is described later (Equation 4.10). An example of iterative minimisation of the above energy function for an Active Shape Model (Cootes and Taylor, 1992) can be seen in Figure 4.1.

Equation 4.1 provides an alternative probabilistic interpretation of the error function. Under the probabilistic formulation of CLM, the fitting algorithms look for the maximum *a posteriori* probability (MAP) of the deformable model parameters $\mathbf{p}$:

$$p(\mathbf{p}|\{l_i=1\}_{i=1}^{n},\mathcal{I}) \propto p(\mathbf{p})\prod_{i=1}^{n} p(l_i=1|\mathbf{x}_i,\mathcal{I}), \tag{4.2}$$

where, $l_i \in \{1,-1\}$ is a discrete random variable indicating whether the i^{th} feature point is aligned or misaligned, $p(\mathbf{p})$ is the prior probability of the model parameters $\mathbf{p}$, and $\prod_{i=1}^{n} p(l_i = 1|\mathbf{x}_i,\mathcal{I})$ is the joint probability of the feature points being aligned at locations $\mathbf{x}_i$, given an image $\mathcal{I}$. From the above formulation it can be clearly seen that all of the local detectors are assumed to be conditionally independent of each other, in contrast to other approaches that model appearance holistically.

The Equation 4.2 is equivalent to Equation 4.1 if the regularisation and data terms take the following form:

$$\mathcal{R}(\mathbf{p}) = -\ln\{p(\mathbf{p})\}, \tag{4.3}$$

$$\mathcal{D}_i(\mathbf{x}_i;\mathcal{I}) = -\ln\{p(l_i = 1|x_i,\mathcal{I})\}. \tag{4.4}$$

The probability of a certain feature being aligned at image location $\mathbf{x}_i$ is $p(l_i = 1|\mathbf{x}_i,\mathcal{I})$. It is computed from the response maps created by patch experts. In my work I have explored different patch experts in terms of both the modality and the regressor being used. I have also explored the best training practices for the classifiers, thus furthering the understanding of deformable models.

It is possible to minimise the error in Equation 4.1 by using general mathematical optimisation techniques, such as the Newton method or stochastic optimisation approaches. However, these approaches often exhibit slow convergence, especially in the presence of a complex deformable model with a large number of parameters (Saragih et al., 2011). This makes general optimisation techniques unsuitable for real-time, or close to real-time tracking. Hence, it is more common to use optimisation strategies designed specifically for CLM fitting.

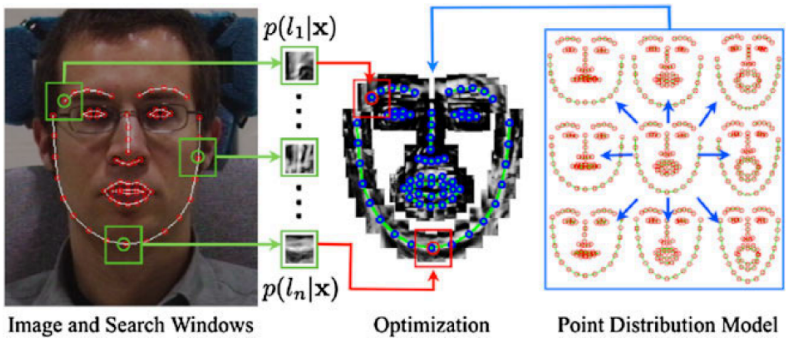


Figure 4.2: Two step CLM fitting: calculation of patch responses in the surrounding region of interest followed by a Point Distribution Model constrained parameter update. Taken from Saragih et al. (2011).

A common approach to CLM fitting is illustrated in Figure 4.2. This approach involves an iteration of two steps. The first step performs an exhaustive local search around each current estimate of a landmark, evaluating the patch expert at each pixel location in an area of interest. This results in response maps around each of the landmarks. The second step involves an optimisation performed over the resulting response maps (taking into account the regularisation term). There are numerous optimisation techniques, such as regularised landmark mean shift (Saragih et al., 2011); exhaustive local search (Wang et al., 2007); and convex quadratic fitting (Wang et al., 2008). None of these techniques optimise across the response maps directly, as that is computationally intractable and susceptible to errors due to noisy response maps. Instead, an approximation over these response maps is used: taking the maximal response value for each landmark (Wang et al., 2007); fitting a Gaussian over the response maps (Wang et al., 2008); fitting Gaussian Mixture Models over response maps (Gu and Kanade, 2008); or using a Kernel Density Estimator (Saragih et al., 2011).

4.1.3 Structure of discussion

CLM-based landmark detection consists of three main parts: the shape model, patch experts and the fitting method. My discussion introduces

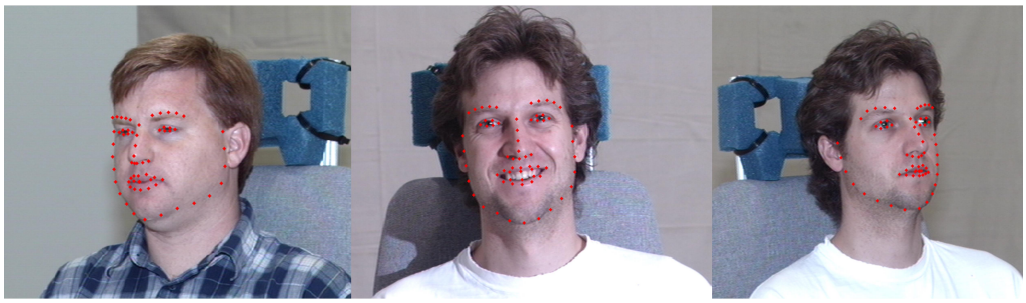


Figure 4.3: Examples of hand labelled feature points, of faces with different expressions and orientations.

each of these parts and gives detailed information on how each of them can be constructed or implemented. Section 4.2 concentrates on the construction and choice of the shape model. Section 4.3 presents the types of patch experts regularly used in CLM fitting and explores some multi-modal extensions. Section 4.5 describes my novel CLM fitting method, together with a multi-scale extension. Section 4.6 outlines how all of these parts, with an addition of a face detector, can be made into a system which can detect and track facial features. Finally, the results of facial tracking on several publicly available datasets are presented in Section 4.7.

4.2 Statistical shape model

A very important part of any model-based landmark detection algorithm is the shape model. Firstly, the model describes the possible deformations of a face, i.e. what constitutes a legal and an illegal face shape. Secondly, the model evaluates the plausibility of that shape in order to guide the fitting (by acting as a prior or a regularisation term).

Several examples of face shapes that the model should be able to describe can be seen in Figure 4.3. Notice how the positions of feature points are affected by both the location and orientation of the head and expression (called global and local parameters respectively).

There exist a number of possible shape models for facial landmark de-

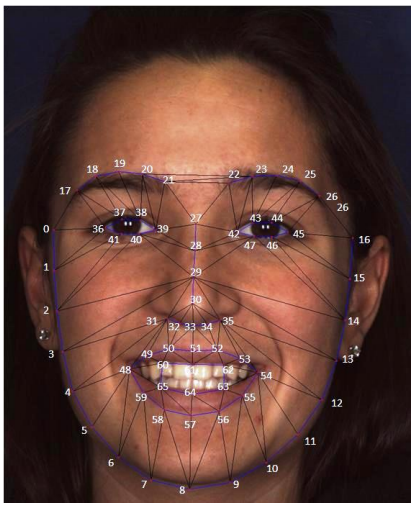


Figure 4.4: Feature points being tracked in my work. Outline of the head and significant internal locations.

tection (Cootes and Taylor, 2004, 1992; Gao et al., 2010; Matthews et al., 2007). Three main variables were chosen for the shape model: the type of the model; the dimensionality of the model; and the projection used for placing the model in the image. The following sections outline the choices I made and the reasons behind them.

4.2.1 Choosing the points

In order to build a model of face geometry, the relevant points have to be identified. Cootes and Taylor suggest that landmarks should represent the boundary or significant internal locations of an object (Cootes and Taylor, 1992). Furthermore, good landmarks are points which can be consistently located from one image to another during annotation of the training set (Cootes and Taylor, 2004). Points could be placed at clear corners of object boundaries, or easily located biological landmarks. However, as there are rarely enough points to give more than a sparse description of the shape, the boundaries are usually augmented with equally spaced points along them (Cootes and Taylor, 2004).

For the above reasons, I chose to model the outline of the face and

the important facial features for emotion recognition: eyebrows (raising, furrowing), lips (smiles, open mouth, yawn, frown etc.), nose (for wrinkling), and eyes (for narrowing or widening). The feature points I used can be seen in Figure 4.4.

Datasets have to be manually or semi-automatically (Sagonas et al., 2013) labelled with the chosen feature points for both training and evaluation. Some of the datasets I used were labelled manually, while others were labelled with help from existing trackers.

4.2.2 Model

The vast majority of deformable shape models use a linear model for non-rigid deformations (Cootes and Taylor, 2004; Gao et al., 2010; Gu and Kanade, 2008; Matthews et al., 2007; Wang et al., 2008). This type of linear model is called a Point Distribution Model (PDM) (Cootes and Taylor, 1992).

The PDM is a linear model which parametrises a class of shapes. It can also be used to estimate the likelihood of the points being in a model, given a set of feature points. This is important for model fitting, as it can act as a prior.

The shape of a face that has n landmark points can be described as a single column vector:

$$\mathbf{X} = [X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n, Z_1, Z_2, \ldots Z_n]^T. \tag{4.5}$$

$\mathbf{X}$ indicates the collection of points in the model. The vector describing a valid instance of a face using the PDM can be represented in the following way:

$$\mathbf{X} = \overline{\mathbf{X}} + \mathbf{\Phi q}. \tag{4.6}$$

Above $\mathbf{X}$ is a model instance of a particular PDM and $\overline{\mathbf{X}}$ is the mean shape of the face (described in the same format as Equation 4.5). The shape is controlled by the m components of linear deformation, described using the $n \times m$ matrix $\mathbf{\Phi}$, and the m dimensional column vector

$\mathbf{q}$, representing the non-rigid deformation parameters. The above definition is in the object coordinate frame. Description of how this can be placed in the image is presented in Section 4.2.4.

Both $\overline{\mathbf{X}}$ and $\mathbf{\Phi}$ can be learned automatically from hand labelled images using Principal Component Analysis (Cootes and Taylor, 2004), through various non-rigid structure from motion methods (Matthews et al., 2007; Torresani et al., 2008), or even defined manually (Cai et al., 2010).

The probability associated with a particular valid shape described by parameters $\mathbf{q}$ can then be expressed as a zero mean Gaussian with Covariance matrix $\mathbf{\Lambda} = \text{diag}([\lambda_1; \ldots; \lambda_m])$ evaluated at $\mathbf{q}$:

$$p(\mathbf{q}) = \mathcal{N}(\mathbf{q}; \mathbf{0}, \mathbf{\Lambda}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{\Lambda}|}} \exp\{-\frac{1}{2}(\mathbf{q}^T \mathbf{\Lambda}^{-1} \mathbf{q})\}. \tag{4.7}$$

$\mathbf{\Lambda}$ is constructed from the training set, based on how much shape variation in the training is explained by the i^{th} parameter, with λ_i corresponding to the q_i parameter.

A Gaussian shape likelihood and CLM formulation in Equation 4.3 leads to the following regularisation term:

$$\begin{aligned} \mathcal{R}(\mathbf{q}) &= -\ln\{\tfrac{1}{\sqrt{(2\pi)^m |\mathbf{\Lambda}|}} \exp\{-\tfrac{1}{2}(\mathbf{q}^T \mathbf{\Lambda}^{-1} \mathbf{q})\}\} \\ &= \ln\{\sqrt{(2\pi)^m |\mathbf{\Lambda}|}\} + \tfrac{1}{2}\mathbf{q}^T \mathbf{\Lambda}^{-1} \mathbf{q}. \end{aligned} \tag{4.8}$$

As constant terms can be ignored, the regularisation term becomes:

$$\mathcal{R}(\mathbf{q}) = \frac{1}{2}\mathbf{q}^T \mathbf{\Lambda}^{-1} \mathbf{q} \propto ||\mathbf{q}||^2_{\mathbf{\Lambda}^{-1}}. \tag{4.9}$$

The notation $||\mathbf{x}||_W$ is a shorthand for $\sqrt{\mathbf{x}^T W \mathbf{x}}$, and represents the Mahalanobis distance with a covariance matrix W, which measures how far a sample is from a mean in a multi-variate Gaussian distribution by calculating the z-value in each dimension . Since, in PDM case, the covariance matrix is diagonal, the Mahalanobis distance reduces to a normalised Euclidean distance.

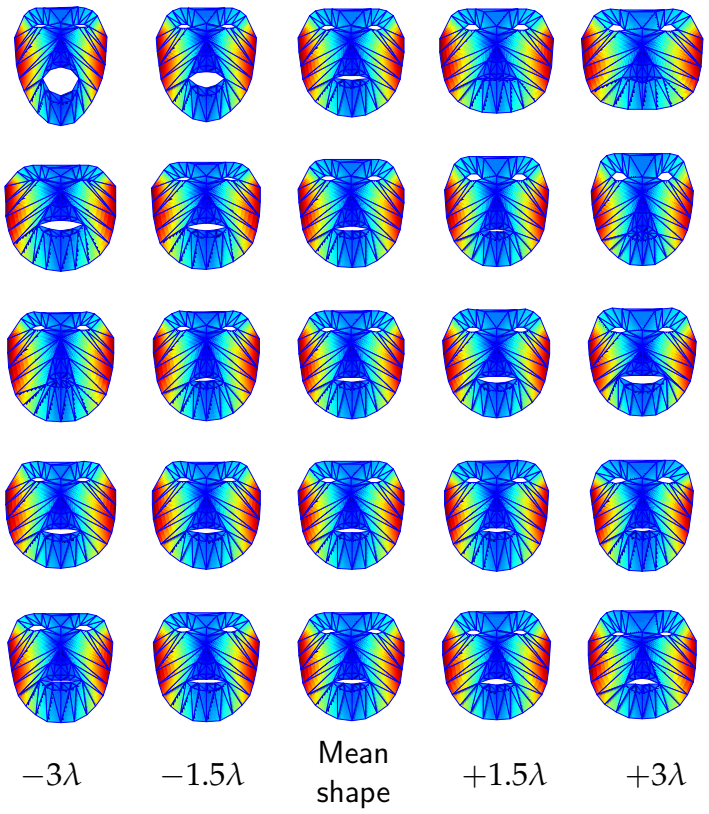


Figure 4.5: The modes of variations of a point distribution model of a face, constructed from the Multi-PIE dataset using non-rigid structure from motion and aligned to frontal orientation. The mean shape is shown together with the five biggest principal components - those with highest variance (λ). The principal components are shown top to bottom. Notice how both the variation in identity (morphology) and expression are captured by the model.

4.2.3 Dimensionality of the model

The previous section defined a 3D PDM (Equation 4.5), 2D versions also exist. 2D models were more popular in the early days of deformable model research due to the inherent difficulty of 3D labelling and unavailability of good 3D sensors. Developments in sensor technology and non-rigid structure from motion approaches increased the popularity of 3D PDMs (Matthews et al., 2007).

Theoretically, a PDM could model shape in any dimension, but for purposes of face modelling the choice is between a 3D or 2D shape model. This choice will affect how the model is placed in an image, how the model is constructed, and the techniques used for fitting. The differences arise because a 3D model needs to be projected, whereas a 2D one only needs to be scaled.

The choice of model dimensionality depends on the problem at hand. For faces which are mainly frontal, a 2D model would suffice, as some variation of out-of-plane rotation could be captured amongst the principal components of the shape model - $\mathbf{\Phi}$. If out-of-plane variation is needed, one could even construct multiple 2D models for desired poses (Cootes et al., 2000). However, this would involve building different PDMs for different orientations and switching between them accordingly, leading to extra computation. Furthermore, numerous labelled training samples at various orientations would be required for such model construction, which is a very time-consuming task.

In order to track a face at various orientations a 3D model is preferable over a 2D one. This is because the orientation is coded explicitly in the former, whereas it has to be controlled by non-rigid shape parameters in the latter. However, 3D model construction is tricky: in order to build a PDM, hand-labelled 3D samples are needed, and labelling faces on 3D surfaces is challenging (Cootes and Taylor, 2004). The task can be made slightly easier if texture images, corresponding to the range data, are also available.

Alternatively, non-rigid structure from motion (NRSFM) approaches can

be used to create the 3D PDM from labelled feature points at various orientations (Matthews et al., 2007; Torresani et al., 2008). However, there are issues facing this approach: labelling the same point at different orientations is tricky especially for the face outline, and hand labelling across pose is very time-consuming.

Even though the construction of a 3D model instead of a 2D one, is potentially more difficult, Matthews et al. (2007) argue that 3D models are preferable for the following reasons. Firstly, their parametrisation is more compact – removing the need to model slight rotations, scalings and translations. Secondly, they are more natural – pose and shape are separated. Lastly, it provides an easier way to deal with self occlusions. Due to the advantages of 3D models over 2D ones, I chose to use a 3D PDM in my experiments.

4.2.4 Placing the model in an image

In order to place the 3D PDM in an image, it needs to be projected. This can be done by either a weak or a full perspective projection. I chose the former as it simplifies the fitting considerably by removing the additional non-linearities introduced by full perspective projection. Weak perspective camera model (otherwise known as *scaled orthographic projection*) assumes that the object of interest lies roughly within a plane and, hence, the projection can be approximated by using the same depth for every point. Use of weak perspective for face tracking is a reasonable approximation because of the relatively small variations of depth along the face plane with respect to distance to the camera.

The following equation is used to place a single feature point of the 3D PDM in an image using weak perspective projection:

$$\mathbf{x}_i = s \cdot R_{2D} \cdot (\overline{\mathbf{X}}_i + \mathbf{\Phi}_i \mathbf{q}) + \mathbf{t}, \tag{4.10}$$

where $\overline{\mathbf{X}}_i = [\overline{x}_i, \overline{y}_i, \overline{z}_i]^T$ is the mean value of the i^{th} feature, $\mathbf{\Phi}_i$ is a $3 \times m$ principal component matrix, and $\mathbf{q}$ is an m dimensional vector of parameters controlling the non-rigid shape.

The rigid shape parameters (or global parameters) in Equation 4.10 can be parametrised using 6 scalars. First of all, s is a scaling term that controls how close the face is to the camera (inversly proportional to average depth $s = \frac{f}{\overline{Z}}$) and $\mathbf{t} = [t_x, t_y]^T$ is the translation term. Finally, $\mathbf{w} = [w_x, w_y, w_z]^T$ is the rotation term that controls the 2×3 matrix R_{2D} - the first two rows of a full 3×3 rotation matrix R (Equation 4.11). The rotation matrix is constructed from an axis-angle representation of rotation. Under axis-angle representation, any 3D rotation can be described using a vector $\mathbf{w} = [w_x, w_y, w_z]^T = \theta\hat{\mathbf{n}}$, where the magnitude of the vector ($|\mathbf{w}| = \theta$) describes the size of rotation in radians around the $\hat{\mathbf{n}}$ axis. Axis-angle representation can be converted to a rotation matrix R using Rodrigues' rotation formula (Szeliski, 2010):

$$R = I + \sin(\theta)[\hat{\mathbf{n}}]_\times + (1 - \cos(\theta))[\hat{\mathbf{n}}]_\times^2, \tag{4.11}$$

where

$$[\hat{\mathbf{n}}]_\times = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}. \tag{4.12}$$

The instance of the face in an image is therefore controlled using the parameter vector $\mathbf{p} = [s, \mathbf{w}, \mathbf{t}, \mathbf{q}]$; where $\mathbf{q}$ represents the local non-rigid deformation, and $s, \mathbf{w}, \mathbf{t}$ are global motion (rigid) parameters.

It is useful to define the function from parameter vector $\mathbf{p}$ to a $2 \times n$ matrix of landmark locations, where the first and second rows are the x and y coordinates of landmarks in and image:

$$\mathrm{P}_{\mathrm{wp}}(\mathbf{p}) = T_{s,\mathbf{w},\mathbf{t}}(\overline{\mathbf{X}} + \mathbf{\Phi}\mathbf{q}), \tag{4.13}$$

where P_{wp} stands for weak perspective projection, which can be defined with the help of homogeneous coordinates:

$$T_{s,\mathbf{w},\mathbf{t}}(\mathbf{X}) = [s \cdot R_{2D}|\mathbf{t}] \cdot \begin{bmatrix} X_1 & X_2 & \dots & X_n \\ Y_1 & Y_2 & \dots & Y_n \\ Z_1 & Z_2 & \dots & Z_n \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{4.14}$$

Prior

Section 4.2.2 demonstrated how to construct a prior for the non-rigid shape, by assuming that non-rigid shape parameters $\mathbf{q}$ follow a Gaussian distribution. For the rigid shape parameters $s, \mathbf{w}, \mathbf{t}$ it is common to use a non-informative prior. This can be achieved by defining $\tilde{\mathbf{\Lambda}}^{-1} = \text{diag}([0;0;0;0;0;0;\lambda_1^{-1};\ldots;\lambda_m^{-1}])$. Leading to the following regularisation term:

$$\mathcal{R}(\mathbf{p}) = ||\mathbf{p}||^2_{\tilde{\mathbf{\Lambda}}^{-1}}. \tag{4.15}$$

Note that this leads to $\tilde{\mathbf{\Lambda}}$ being undefined, due to the division by zero, however, this does not matter as $\tilde{\mathbf{\Lambda}}$ is never used directly.

4.2.5 Point distribution model fitting

A point distribution model (PDM) can be used to do several things: generate realistic examples of a class, guide model fitting, and evaluate the likelihood of the model. It is also possible to find the PDM parameters given an instance of the model. That is, given a set of corresponding points in 2D, it is possible to find which model parameters $\mathbf{p}$ represent the instance, which is useful for two main reasons. Firstly, it makes it possible to estimate the pose of a face (orientation and translation) given only the labelled 2D landmarks. This is useful for both evaluating how well an algorithm performs at different orientations, and for picking which images to use when training different patch experts for different orientations. Secondly, it is useful for the generation of synthetic training data from range scans at different orientations.

However, fitting a 3D model to 2D points is not straightforward, and requires an iterative approach. This can be done using the Gauss-Newton algorithm[1] with a slight correction for rotation parameters.

For the task of aligning a 3D PDM $(\bar{\mathbf{x}}, \Phi)$ to the 2D model instance $\mathbf{y}$, the following function needs to be minimised:

[1] an algorithm for solving non-linear least squares problems of the form that involves sum of squared residuals

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}}\{||\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p})||_2^2 + r||\mathbf{p}||^2_{\tilde{\mathbf{\Lambda}}^{-1}}\}. \tag{4.16}$$

Algorithm 1 Fitting a 3D PDM to new 2D points

Require: Feature points - $\mathbf{y}$, $PDM - \Phi, \overline{\mathbf{X}}$, regularisation terms r, Λ
  Initialise the shape parameters, $\mathbf{p}$, to zero
  while not converged ($||\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p})||_2^2 + r||\mathbf{p}||^2_{\tilde{\mathbf{\Lambda}}^{-1}}$) **do**
    Linearise $||\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p})||_2^2$ around $\mathbf{p}$
    Calculate the Jacobian $\mathbf{J}$ Eq. 4.28
    Solve the linear system for parameter update $\Delta\mathbf{p}$ Eq. 4.19
    Update the rotation parameters
    Update all other parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
  end while
  return $\mathbf{p} = [\mathbf{R}, \mathbf{T}_{2D}, s, \mathbf{q}]$

Above, $\mathrm{P}_{\mathrm{wp}}(\mathbf{p})$ is the scaled orthographic projection of the model described by parameters $\mathbf{p}$ (Equation 4.13). $r||\mathbf{p}||^2_{\tilde{\mathbf{\Lambda}}^{-1}}$ is the regularisation term, which helps avoid overfitting. The parameter r controls the trade-off between penalising unlikely faces and the landmark placement error. The suitable value of r will depend on the noisiness of data, but I experimentally found that values 10–30 work well.

The solution to Equation 4.16 can be found using Algorithm 1, which is a slightly modified version of Gauss-Newton method for non-linear least squares problems.

Derivation

If an initial estimate of $\mathbf{p}$ is available, it is possible to find $\Delta\mathbf{p}$ in the direction of the optimal solution, leading to $\mathbf{p}^* = \mathbf{p} + \Delta\mathbf{p}$. This leads to the next estimate of $\mathbf{p}$, which can be used for the next iteration. In order to find a $\Delta\mathbf{p}$ in the direction of an optimal value of $\mathbf{p}$, Taylor series expansion of P_{wp} around the current estimate of $\mathbf{p}$ can be used:

$$||\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p}^*)||_2^2 + r||\mathbf{p}^*||^2_{\tilde{\mathbf{\Lambda}}^{-1}} \approx ||\mathbf{y} - (\mathrm{P}_{\mathrm{wp}}(\mathbf{p}) + J\Delta\mathbf{p})||_2^2 + r||\mathbf{p}^*||^2_{\tilde{\mathbf{\Lambda}}^{-1}}, \tag{4.17}$$

where $J = \frac{\partial \mathrm{P}_{\mathrm{wp}}(\mathbf{p})}{\partial \mathbf{p}}$ is the Jacobian of $\mathrm{P}_{\mathrm{wp}}(\mathbf{p})$ evaluated at $\mathbf{p}$. The $\Delta\mathbf{p}$ which brings us closer to the solution is:

$$\Delta\mathbf{p} = \underset{\Delta\mathbf{p}}{\operatorname{argmin}}\{||\mathbf{y} - (\mathrm{P}_{\mathrm{wp}}(\mathbf{p}) + J\Delta\mathbf{p})||_2^2 + r||\mathbf{p} + \Delta\mathbf{p}||^2_{\tilde{\mathbf{\Lambda}}^{-1}}\}. \tag{4.18}$$

This can be solved for $\Delta\mathbf{p}$ using the Tikhonov regularised linear least squares (also known as ridge regression):

$$\Delta\mathbf{p} = (J^T J + \tilde{\mathbf{\Lambda}}^{-1})^{-1}(J^T(\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p})) - \tilde{\mathbf{\Lambda}}^{-1}\mathbf{p}). \tag{4.19}$$

Jacobian

Intuitively, the Jacobian describes how the function values are changing based on the infinitesimal changes of its parameters. In the case of PDM, it models the changes of landmark locations – $\mathbf{x}$, based on the parameters – $\mathbf{p}$. The computation of the Jacobian is needed for the PDM fitting and for other parts of CLM landmark detection, hence its derivation is explained in detail.

As a reminder, the location of a landmark point evaluated at $\mathbf{p} = [s, \mathbf{w}, \mathbf{t}, \mathbf{q}]$ is defined as:

$$\mathbf{x}_i = s \cdot R_{2\mathrm{D}} \cdot (\overline{\mathbf{X}}_i + \mathbf{\Phi}_i\mathbf{q}) + \mathbf{t} = s \cdot R_{2\mathrm{D}} \cdot \mathbf{X}'_i + \mathbf{t}, \tag{4.20}$$

where $\mathbf{X}'_i = [X'_i, Y'_i, Z'_i] = \overline{\mathbf{X}}_i + \mathbf{\Phi}_i\mathbf{q}$ for brevity.

The change in x and y landmark locations, based on the changes in the scaling term s, is as follows:

$$\frac{\partial \mathbf{x}_i}{\partial s} = \begin{bmatrix} \frac{\partial x_i}{\partial s} \\ \frac{\partial y_i}{\partial s} \end{bmatrix} = \begin{bmatrix} R_{1,:} \cdot \mathbf{X}'_i \\ R_{2,:} \cdot \mathbf{X}'_i \end{bmatrix}, \tag{4.21}$$

where $R_{1,:}$ and $R_{2,:}$ indicate the first and the second row of the rotation matrix R.

The change in landmark location based on the translation term $\mathbf{t} = [t_x, t_y]^T$ is straightforward:

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{t}^T} = \begin{bmatrix} \frac{\partial x_i}{\partial t_x}, & \frac{\partial x_i}{\partial t_y} \\ \frac{\partial y_i}{\partial t_x}, & \frac{\partial y_i}{\partial t_y} \end{bmatrix} = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}. \tag{4.22}$$

The least straightforward part of the rigid parameter Jacobian, is the effect of rotation parameters $\mathbf{w} = [w_x, w_y, w_z]^T$ on landmark locations. First, the landmark locations can be expressed in terms of the current rotation matrix R_{2D}, and an infinitesimal rotation R_Δ:

$$\mathbf{x}_i = s \cdot R_{2\mathrm{D}} \cdot R_\Delta \cdot \mathbf{X}'_i + \mathbf{t}. \tag{4.23}$$

The infinitesimal rotation R_Δ can be approximated using the axis-angle representation of rotation, Rodriguez's formula (Equation (4.11), and the small angle assumption of $\sin(\theta) = \theta$ and $\cos(\theta) = 0$ as:

$$R_\Delta = \begin{bmatrix} 1 & -w_z & w_y \\ w_z & 1 & -w_x \\ -w_y & w_x & 1 \end{bmatrix}. \tag{4.24}$$

This leads to:

$$\mathbf{x}_i = s \cdot R_{2D} \cdot \begin{bmatrix} X'_i - w_z Y_i + w_y Z_i \\ w_z X'_i + Y_i - w_x Z_i \\ -w_y X'_i + w_x Y_i + Z_i \end{bmatrix} + \mathbf{t}. \tag{4.25}$$

This can now be used to derive the changes in landmark locations due to changes in rotation parameters:

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{w}^T} = \begin{bmatrix} \frac{\partial \mathbf{x}_i}{\partial w_x}, & \frac{\partial \mathbf{x}_i}{\partial w_y}, & \frac{\partial \mathbf{x}_i}{\partial w_z} \end{bmatrix} = s \cdot R_{2D} \begin{bmatrix} 0 & Z_i & -Y_i \\ -Z_i & 0 & X_i \\ Y_i & -X_i & 0 \end{bmatrix}. \tag{4.26}$$

The changes in landmarks due to changes in non-rigid parameters of shape are as follows:

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{q}^T} = s \cdot R_{2D} \Phi_i. \tag{4.27}$$

These can be combined to get the full Jacobian of interest:

$$J = \begin{bmatrix} \frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial w_x} & \frac{\partial x_1}{\partial w_y} & \frac{\partial x_1}{\partial w_z} & \frac{\partial x_1}{\partial t_x} & \frac{\partial x_1}{\partial t_y} & \frac{\partial x_1}{\partial q_1} & \cdots & \frac{\partial x_1}{\partial q_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial s} & \frac{\partial x_n}{\partial w_x} & \frac{\partial x_n}{\partial w_y} & \frac{\partial x_n}{\partial w_z} & \frac{\partial x_n}{\partial t_x} & \frac{\partial x_n}{\partial t_y} & \frac{\partial x_n}{\partial q_1} & \cdots & \frac{\partial x_n}{\partial q_n} \\ \frac{\partial y_1}{\partial s} & \frac{\partial y_1}{\partial w_x} & \frac{\partial y_1}{\partial w_y} & \frac{\partial y_1}{\partial w_z} & \frac{\partial y_1}{\partial t_x} & \frac{\partial y_1}{\partial t_y} & \frac{\partial y_1}{\partial q_1} & \cdots & \frac{\partial y_1}{\partial q_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial s} & \frac{\partial y_n}{\partial w_x} & \frac{\partial y_n}{\partial w_y} & \frac{\partial y_n}{\partial w_z} & \frac{\partial y_n}{\partial t_x} & \frac{\partial y_n}{\partial t_y} & \frac{\partial y_n}{\partial q_1} & \cdots & \frac{\partial y_n}{\partial q_n} \end{bmatrix}. \tag{4.28}$$

This Jacobian can now be used to solve for the parameter update $\Delta\mathbf{p}$ in Algorithm 1.

Solving for the parameter update $\Delta\mathbf{p}$ using the Equation 4.18 and adding them to the initial parameter estimates, leads to the updated shape parameters closer to the optimal value. An exception, however, is made for rotation parameters.

Rotation parameter update

In order to get the final rotation parametrisation after the update, the current rotation matrix can be multiplied with the updated rotation matrix, leading to $R'_{2D} = R_{2D}R_\Delta$. Equation 4.24 is used to compute R_Δ. The resulting R'_{2D} can now be converted to axis-angle representation, leading to an updated orientation.

However, because of the approximation used to construct R_Δ, it is not guaranteed to be orthogonal. It can be made orthogonal using Singular Value Decomposition (SVD). The decomposition of R_Δ is as follows:

$$USV^T = R_\Delta. \tag{4.29}$$

Here U and V are orthogonal. The corrected R_Δ can now be expressed as:

$$R_{\Delta\text{corrected}} = U \cdot \det(UV^T) \cdot V^T. \tag{4.30}$$

The matrix determinant $\det(UV^T)$ ensures the correct handedness of the new rotation.

This leads to a final rotation matrix $R'_{2D} = R_{2D} \cdot R_{\Delta\text{corrected}}$.

4.2.6 Model construction

This section describes how a 3D PDM can be constructed automatically from a set of labelled examples. The shape model consists of three components: the mean model shape $\overline{\mathbf{X}}$, the main modes of variation $\mathbf{\Phi}$ (principal components), and the variance matrix $\mathbf{\Lambda}$, which describes the amount of variability explained by each of the principal components.

There are multiple ways of creating a PDM. These include: using Principal Component Analysis on the 3D landmark locations; using non-rigid structure from motion (NRSFM) on the 2D landmark locations (Torresani et al., 2008; Xiao et al., 2006); or even defining the shape variation manually (Cai et al., 2010).

Given 2D locations of n feature points across m images, NRSFM recovers the motion of the non-rigid object relative to the camera. The object can be rotating, translating, or undergoing a linear 3D deformation. NRSFM estimates the transformations, affecting the object, and the linear model of deformation (Torresani et al., 2008; Xiao et al., 2006).

In my work I used a model constructed using the NRSFM approach from the labels of the Multi-PIE dataset. It is a 3D deformable model with 24 principal components and can be seen in Figure 4.5.

4.3 Patch experts

Patch experts (also called *local detectors*) are a very important part of the CLM. They evaluate the probability of a landmark being aligned (or alternatively the misalignment error) at a particular pixel location. There have been various patch experts proposed: simple template matching techniques (Cristinacce and Cootes, 2006); logistic regressors (Paquet, 2009); and Support Vector Machines (Jeni et al., 2012; Saragih et al., 2011; Wang et al., 2008).

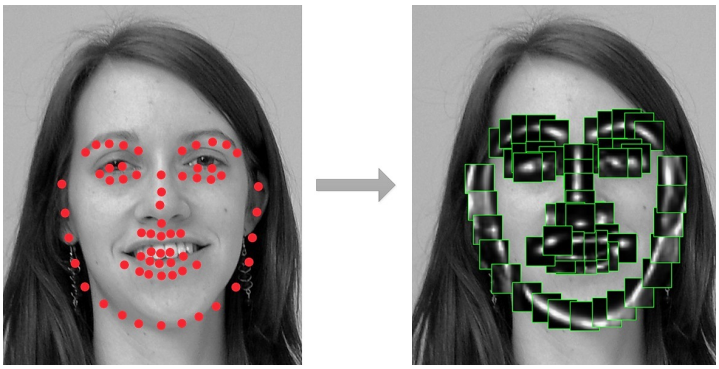


Figure 4.6: Example of 11 × 11px SVR patch experts evaluated on a greyscale image of a face at the locations indicated by red circles and a 21 × 21px area of interest surrounding them. The green bounding boxes represent a particular patch expert response map in the area of interest. The darker response values indicate low probability of alignment, and brighter values indicate high probability of alignment.

Under the probabilistic formulation, patch experts quantify the probability of alignment of a feature i – $p(l_i = 1|\mathbf{x}_i, \mathcal{I})$, at the image location $\mathbf{x}_i$, in an image $\mathcal{I}$, based on the surrounding support region (often an $m \times m$ grid). The image is usually expressed as greyscale pixel values, but other modalities can be used as well. The evaluation of a patch expert in an *area of interest* leads to a *response map*. An example of patch expert response maps can be seen in Figure 4.6.

A very popular patch expert is a Support Vector Regressor (SVR) in combination with a logistic regressor (Jeni et al., 2012; Saragih et al., 2011; Wang et al., 2008). It is defined as follows:

$$p(l_i|\mathbf{x}_i, \mathcal{I}) = \frac{1}{1 + e^{d\mathcal{C}_i(\mathbf{x}_i;\mathcal{I})+c}}. \tag{4.31}$$

Here, $\mathcal{C}_i$ is the output of a SVR regressor for the i^{th} feature, c is the logistic regressor intercept, and d the regression coefficient. The use of a logistic regressor in addition to the Support Vector enforces the output to be within 0 and 1. The advantage of this formulation is its computational simplicity and potential for efficient implementation on images using convolution (see the following section). Furthermore, it

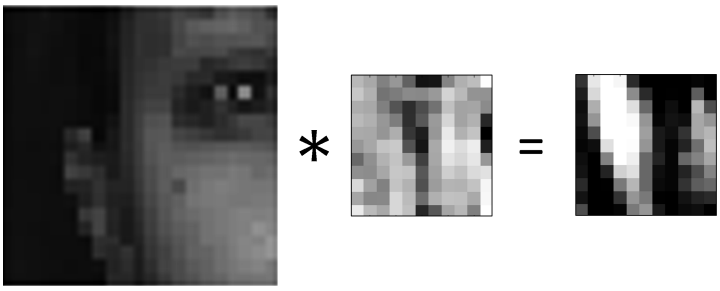


Figure 4.7: Example of a 11×11 pixel area of interest (21×21 pixel grid) being convolved with 11×11 pixel SVR patch expert weights, resulting in a 11×11 pixel response. This response can be used to calculate the final response map using logistic regression. The darker response values indicate low probability of alignment, and brighter values indicate high probability of alignment.

is easy to train, and there are a number of libraries for efficient SVR training (Chang and Lin, 2011; Fan et al., 2008).

The support vector regressor is expressed as:

$$\mathcal{C}_i(\mathbf{x}_i;\mathcal{I}) = \mathbf{w}_i^T \mathcal{P}(\mathcal{W}(\mathbf{x}_i;\mathcal{I})) + b_i, \tag{4.32}$$

where $\{\mathbf{w}_i, b_i\}$ are the weights and biases associated with a particular feature SVR. Here $\mathcal{W}(\mathbf{x}_i;\mathcal{I})$ represents the *support region* – a vectorised version of $n \times n$ image patch centred around $\mathbf{x}_i$. Often an 11×11 support region is used, it is small enough to enable real-time implementations and large enough to capture interesting information. $\mathcal{P}$ is the normalisation function which returns a zero mean and unit L2 norm of the signal:

$$\mathcal{P}(\mathbf{x}) = \frac{\mathbf{x} - \overline{\mathbf{x}}}{||\mathbf{x} - \overline{\mathbf{x}}||_2}. \tag{4.33}$$

The normalisation helps make the patch less sensitive to intensity variations due to changing lighting conditions.

4.3.1 Implementation using convolution

In CLM fitting the patch expert is usually evaluated exhaustively in a rectangular *area of interest*, leading to a *response map*. Each response calculation has to be fast, as the patch experts for each landmark have to be evaluated at every pixel location in the area of interest.

Typical area of interest sizes may range from 11×11 to 21×21 pixels, depending on the difficulty of the scene, precision of initial parameter estimate, and the potential amount of motion in the video sequence. This requires $121 - 441$ patch responses per each of the 66 landmarks to be computed for every iteration of the fitting algorithm.

In order to calculate each of the responses, each support region needs to be normalised and multiplied by the SVR weights. This becomes very computationally expensive if an exhaustive local search around each landmark is performed.

Fortunately, the most computationally expensive tasks during patch response computation are equivalent to the normalised cross-correlation problem on an image $\mathcal{I}$ with a template $\mathcal{T}$. This can be done by reshaping the SVR weights to a $N \times N$ template and flipping it along horizontal and vertical axes and calculating the response as follows:

$$C_i((u,v);\mathcal{I}) = \frac{\sum_{x,y}[\mathcal{I}(x,y) - \overline{\mathcal{I}}_{u,v}][\mathcal{T}(x-u,y-v) - \overline{\mathcal{T}}]}{\{\sum_{x,y}[\mathcal{I}(x,y) - \overline{\mathcal{I}}_{u,v}]^2 \sum_{x,y}[\mathcal{T}(x-y,y-v) - \overline{\mathcal{T}}]^2\}^{\frac{1}{2}}} + b_i. \tag{4.34}$$

Here $\overline{\mathcal{T}}$ is the mean of the weights and $\overline{\mathcal{I}}_{u,v}$ is the mean of $\mathcal{I}$ under the template. An example of using convolution as a step in response calculation is in Figure 4.7.

The correlation response can be calculated efficiently with the use of *fast normalised cross-correlation* (Lewis, 1995). If normalisation is ignored, the evaluation of an SVR regressor (without a bias term) across each area of interest is equivalent to convolution, which can be computed more efficiently in the Fourier domain. Normalisation is then performed using integral images. See Lewis (1995) for more details.

Use of fast normalised cross-correlation is not limited to response map computation from SVR patch experts only; it can be used alongside other regressors as well. The outlined optimisation also presents an alternative and an interesting view of patch experts. Patch experts can be seen as convolution kernels which produce the desired patch responses.

4.3.2 Modalities to use

Although, there has been work exploring the use of gradient intensity images (Stegmann and Larsen, 2003) and different colour channels (Baltrušaitis and Robinson, 2010; Ionita et al., 2009) for Active Appearance and Active Shape Model fitting, previously published work on CLM concentrates on the use of simple greyscale images ($\mathcal{I}$). It is an interesting question to see if an additional modality helps with the fitting accuracy of CLM. In my work I explore the use of squared gradient intensity image as an additional input modality for CLM patch experts.

The squared gradient intensity image is defined as follows:

$$\mathcal{I}_\nabla = \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2. \tag{4.35}$$

In order to retain the speed gained from using normalised cross-correlation, the feature vectors from intensity and gradient images cannot be simply combined. Separate regressors for each of them need to be trained, thus leading to two patch experts:

$$p(l_i|\mathbf{x}_i, \mathcal{I}) = \frac{1}{1 + e^{d\mathcal{C}_i(\mathbf{x}_i;\mathcal{I})+c}}, \tag{4.36}$$

$$p(l_i|\mathbf{x}_i, \mathcal{I}_\nabla) = \frac{1}{1 + e^{d_\nabla\mathcal{C}_{\nabla,i}(\mathbf{x}_i;\mathcal{I}_\nabla)+c_\nabla}}. \tag{4.37}$$

In order to benefit from multiple patch experts, their response maps have to be combined. This can be achieved in a number of ways – multiplication, arithmetic mean, and geometric mean:

$$p(l_i|\mathbf{x}_i, \mathcal{I}, \mathcal{I}_\nabla) = p(l_i|\mathbf{x}_i, \mathcal{I}) \cdot p(l_i|\mathbf{x}_i, \mathcal{I}_\nabla). \tag{4.38}$$

$$p(l_i|\mathbf{x}_i, \mathcal{I}, \mathcal{I}_\nabla) = \frac{p(l_i|\mathbf{x}_i, \mathcal{I}) + p(l_i|\mathbf{x}_i, \mathcal{I}_\nabla)}{2}, \tag{4.39}$$

$$p(l_i|\mathbf{x}_i, \mathcal{I}, \mathcal{I}_\nabla) = \sqrt{p(l_i|\mathbf{x}_i, \mathcal{I}) \cdot p(l_i|\mathbf{x}_i, \mathcal{I}_\nabla)}. \tag{4.40}$$

Experimentally, I found that the multiplication method was the most effective for combining greyscale and gradient intensity based response maps. This led to significantly better results than just using greyscale images. However, the slight disadvantage of using multiple modalities is the increased patch response computation time. The results of experiments using different modality combinations can be found in Section 4.7.2.

4.3.3 Multi-view patch experts

In order to track faces at multiple poses, examples of faces at different poses are needed for patch expert training. However, if a patch expert of a certain feature is trained on all poses, it will not work well due to the complexity of the task. A way to approach this problem is by training separate sets of patch expert for the views of interest. This is similar to View-based Active Appearance models (Cootes et al., 2000), in which a separate view would have a separate associated AAM.

During the model fitting, the set of patch experts to be used is chosen based on the current orientation estimate. Furthermore, if the landmark is invisible at a certain orientation, for example one side of the face in a profile image, the occluded points are excluded from model fitting. Section 4.8.2 experimentally demonstrates the benefits of using more views. However, training more views requires more data and increases the training time needed.

4.4 Patch expert training

The purpose of patch expert training (SVR or logistic regression, etc.) is to learn a mapping from the patch support region (feature vector) to a scalar (response). This section describes both the features used for the

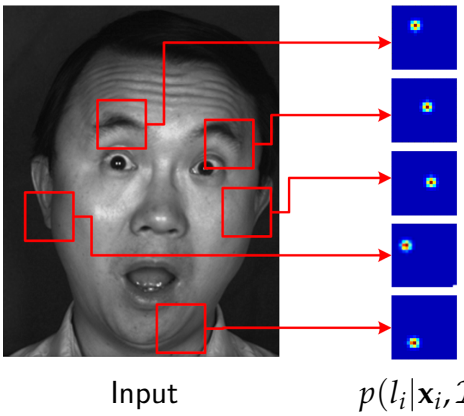


Figure 4.8: Example of the training data. The image on the left indicates the sampling areas, and the response maps on the right indicate the ground truth responses expected from applying a patch expert on them, generated using a Gaussian centred on the ground truth location of the feature point.

patch experts and the generation of their associated labels, and provides other details necessary to train patch experts.

4.4.1 Training data

The patch support region is commonly a vectorised and normalised $n \times n$ pixel grid leading to an $n \times n$ dimensional feature vector. Examples of patch support regions at different scales can be seen in Figure 4.9. I used a 11×11 pixel support region for my experiments.

Given a patch support around the pixel location $\mathbf{x}_i$, the corresponding response is $p(l_i|\mathbf{x}_i, \mathcal{I})$. Taking an image $\mathcal{I}$, with landmark i at $\mathbf{z}_i = [u, v]^T$, the probability of it being aligned at $\mathbf{x}_i$ is modelled as $p(l_i|\mathbf{x}_i, \mathcal{I}) = \mathcal{N}(\mathbf{x}_i; \mathbf{z}_i, \sigma I)$, where I is a 2×2 identity matrix. That is, the alignment probability can be modelled as an isotropic Gaussian with standard deviation σ in both x and y dimensions, centred on the ground truth landmark location.

A good selection of σ is very important for training. If σ is too small the classifier might lead to too many misclassifications. This is because

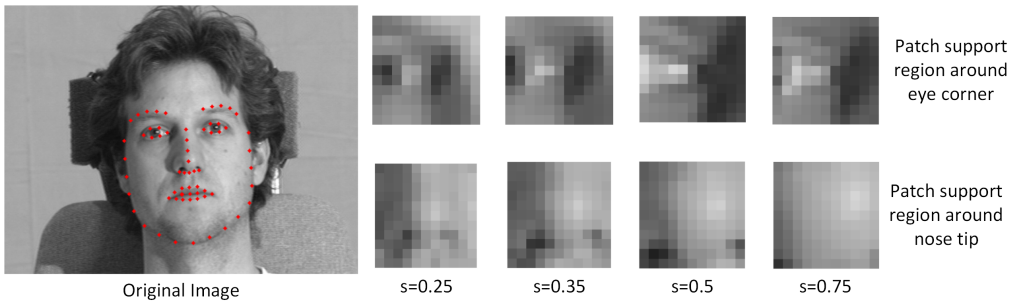


Figure 4.9: Example of extracted support areas around two landmarks (nose tip and eye corner) at different scales. Notice how difficult it becomes to distinguish features from just an 11×11 pixel grid when the scaling term increases.

the patches displaced from each other only by a single pixel will not be very different even though their expected output might. If σ is too large, however, the fine-grained accuracy of the regressor is sacrificed, leading to less accurate landmark detection. I experimentally determined that $\sigma = 1$ leads to best results across multiple datasets.

In order to generate training data, areas close and far away from the ground truth location can be sampled. See Figure 4.8 for examples of such area sampling, together with expected response maps. In my experiments I used a 9×9 pixel area, with a random offset of 0–4 pixels for positive and 10-15 pixels for negative training samples. The ratio of close to far samples was 1 to 20, with many more negative samples.

In order to ensure some lighting invariance, the training data was normalised by taking a z-score of every patch. During the patch response calculation for model fitting, this normalisation was performed implicitly through the use of normalised cross-correlation (see Section 4.3.1).

Patch scaling

For the training to be successful, all of the training data has to be in the same reference scale. However, this is not always the case, especially in the less constrained datasets. In order to align all of the training images

Scale	rms	corr.
0.25	0.074	0.14
0.35	0.075	0.10
0.5	0.076	0.07

Table 4.1: Results comparing SVR patch experts trained on different scales and evaluated on a hold out fold. Notice the decreased performance in terms of root means square error and Pearson correlation coefficient.

to the same reference frame, PDM parameters of their labelled feature points have to be found (see Section 4.2.5 of how to do this). The estimated scaling parameter s can then be used to scale the corresponding face image to the desired scale.

It is important to select a suitable reference scale. If a large scale is used the patch expert can be more accurate, but not as robust; if a small scaling term is used the patch expert loses accuracy. This is demonstrated in my experiments (see Section 4.7.4) that show that CLM landmark detection gains accuracy but loses robustness if the scaling term is increased.

Furthermore, if the scale is too large, an 11×11 patch expert does not have enough information to build an accurate regressor (this can be seen illustrated in Figure 4.9) and Table 4.1.

In my experiments, I used a 3D point distribution model with a projected inter-ocular distance of 65 pixels at $s = 1$, and 11×11 pixel patch experts. I found the best reference scales to be $\{0.25, 0.35, 0.5\}$.

4.5 Constrained Local Model fitting

CLM fitting usually employs a two step strategy (Cristinacce and Cootes, 2006; Gu and Kanade, 2008; Saragih et al., 2011; Wang et al., 2008). The first step evaluates each of the patch experts around the current estimate of its corresponding feature point - leading to a response map around every feature point. The second step iteratively updates the model pa-

rameters to maximise Equation 4.2 until a convergence metric is reached. However, instead of optimising on the patch responses directly, an approximation is used. One such approach is the regularised landmark mean-shift (RLMS) (Saragih et al., 2011).

4.5.1 Regularised landmark mean shift

The RLMS algorithm, first introduced by Saragih et al. (2011), attempts to find the maximum *a posteriori* estimate of $\mathbf{p}$ in the following equation:

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmax}}\{p(\mathbf{p}) \prod_{i=1}^{n} p(l_i = 1|\mathbf{x}_i, \mathcal{I})\}. \tag{4.41}$$

Treating the locations of the true landmarks as hidden variables, they can be marginalised out of the likelihood that the landmarks are aligned:

$$p(l_i = 1|\mathbf{x}_i, \mathcal{I}) = \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} p(l_i = 1|\mathbf{y}_i, \mathcal{I}) \mathcal{N}(\mathbf{y}_i; \mathbf{x}_i, \rho\mathbf{I}), \tag{4.42}$$

where $\mathbf{\Psi}_i$ denotes all integer locations where the patch expert is evaluated (every pixel in an $n \times n$ area of interest around the current estimate). The value of ρ reflects the amount of observational noise expected, and is learned from the data (Saragih et al., 2011). This formulation is equivalent to approximating the likelihood of point alignment using a Gaussian Kernel Density Estimator (KDE):

$$p(l_i = 1|\mathbf{x}_i, \mathcal{I}) = \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} \pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i; \mathbf{y}_i, \rho\mathbf{I}). \tag{4.43}$$

Here $p(l_i = 1|\mathbf{x}_i, \mathcal{I})$ refers to the approximation of the patch response map using KDE, and $\pi_{\mathbf{y}_i} = p(l_i = 1|\mathbf{y}_i, \mathcal{I})$ is the patch expert response at $\mathbf{y}_i$ (Section 4.3).

Substituting Equation 4.43 into Equation 4.41 leads to:

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmax}}\{p(\mathbf{p}) \prod_{i=1}^{n} \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} \pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i; \mathbf{y}_i, \rho\mathbf{I})\}, \tag{4.44}$$

which can be solved using the RLMS algorithm. The approach uses expectation maximisation (Saragih et al., 2011), where the E-step involves

Algorithm 2 RLMS algorithm

Require: Image $\mathcal{I}$, initial parameters $\mathbf{p}$, kernel variance ρ, regularisation term r, and patch experts $\{d_i, c_i, \mathbf{w}_i, b_i\}_{i=1}^{n}$
  Compute affine transform $\mathcal{T}$ from image space to patch space
  while num iterations **do**
    Convert image using the affine transform $\mathcal{T}$
    Compute patch responses (Equation 4.31)
    while not converged($\mathbf{p}$) **do**
      Compute mean-shift vectors $\mathbf{v}$ (Equation 4.45)
      Convert them back to image space using $\mathcal{T}^{-1}$
      Compute global PDM parameter update $\Delta\mathbf{p}$ (Equation 4.46)
      Update global parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
      Compute all PDM parameter update $\Delta\mathbf{p}$ (Equation 4.46)
      Update all parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
    end while
  end while
  return $\mathbf{p}$

evaluating the posterior over the candidates, and the M-step finds the parameter update. The pseudocode for RLMS is shown in Algorithm 2.

As a prior $p(\mathbf{p})$ for parameters $\mathbf{p}$, RLMS assumes that the non-rigid shape parameters $\mathbf{q}$ vary according to a Gaussian distribution (Section 4.2.2); and the rigid parameters s, $\mathbf{w}$, and $\mathbf{t}$ follow a non-informative uniform distribution (Section 4.2.4).

The RLMS approach relies on mean-shift algorithm, which is a common way to maximise over a kernel density estimate. The mean shift vector $\mathbf{v}$, comprising of mean shifts for every landmark under the current estimate of every feature point $\mathbf{x}_i^c$ is defined as:

$$\mathbf{v}_i = \sum_{\mathbf{y}_i \in \Psi_i} \frac{\pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i^c; \mathbf{y}_i, \rho\mathbf{I})}{\sum_{\mathbf{z}_i \in \Psi_i} \pi_{\mathbf{z}_i} \mathcal{N}(\mathbf{x}_i^c; \mathbf{z}_i, \rho\mathbf{I})} - \mathbf{x}_i^c. \tag{4.45}$$

Given the mean shift vector, and incorporating the prior, the parameter update rule is:[2]

[2]the RLMS formulation by Saragih et al. (2011) did not have a separate regularisation term r and instead used the Gaussian KDE variance ρ

$$\Delta \mathbf{p} = -(J^T J + r\Lambda^{-1})(r\Lambda^{-1}\mathbf{p} - J^T \mathbf{v}). \tag{4.46}$$

The Jacobian J in the above equation is the same as defined in Equation 4.28. Furthermore, the same correction of the rotation parameters as used in Section 4.2.5 is used to correct rotation parameters for RLMS. As is often the case in deformable model fitting, the update is first performed on the rigid (global) motion, followed by an update to all of the parameters.

Because the patch experts are trained at a particular scale and orientation, the image needs to be warped to match them. This is done using a 2D affine transform $\mathcal{T}$ from the current feature point estimates to the training reference frame. Finally, as the parameter update should happen in the image, and not the reference space, the mean shift vectors are transformed to the image space using $\mathcal{T}^{-1}$.

Alternative view

Notice the similarity between the parameter update rule in Equation 4.46 and the parameter update rule for PDM fitting in Equation 4.19. They both use Tikhonov regularised linear least squares to determine the parameter update $\Delta \mathbf{p}$.

If the alignment error $\mathbf{y} - \mathrm{P}_{\mathrm{wp}}(\mathbf{p})$ in Equation 4.46 is replaced with a mean shift vector $\mathbf{v}$, the result is the same RLMS parameter update rule.

Treating a mean-shift vector as a misalignment error leads to a slightly different view of the RLMS algorithm. The mean shift vector points in the direction where the feature point should go, but the motion is restricted by the PDM and the regularisation terms. The mean shift becomes constrained by a subspace (Saragih et al., 2009). This interpretation leads to the following RLMS update objective:

$$\underset{\Delta \mathbf{p}}{\operatorname{argmin}}\{||\mathbf{v} - J\Delta \mathbf{p}||_2^2 + r||\mathbf{p} + \Delta \mathbf{p}||^2_{\tilde{\Lambda}^{-1}}\}. \tag{4.47}$$

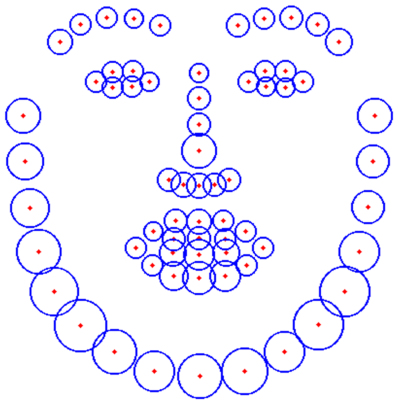


Figure 4.10: The reliabilities of SVR patch experts, smaller circles represent more reliability (less variance).

4.5.2 Non-uniform Regularised landmark mean shift

A problem facing RLMS-based CLM fitting is that each of the patch experts is equally trusted, but this should clearly not be the case. This can be seen illustrated in Figures 4.6 and 6.4, where the response maps of certain features are noisier. To tackle this issue, instead of solving Equation 4.47, I propose minimising the following objective function:

$$\arg\min_{\Delta \mathbf{p}} \{||J\Delta \mathbf{p} - \mathbf{v}||_W^2 + r||\mathbf{p} + \Delta \mathbf{p}||^2_{\tilde{\Lambda}^{-1}}\}. \tag{4.48}$$

The diagonal weight matrix W allows for weighting of mean-shift vectors. Non-linear least squares with Tikhonov Regularisation leads to the following update rule:

$$\Delta \mathbf{p} = -(J^T W J + r\Lambda^{-1})(r\Lambda^{-1}\mathbf{p} - J^T W \mathbf{v}). \tag{4.49}$$

Note that, if we use a non-informative identity $W = I$, the above collapses to the regular RLMS update rule. I call the CLM fitting algorithm that uses this update rule - Non-uniform Regularised Landmark Mean Shift (NU-RLMS).

To construct W, the performance of patch experts on training data is used. The correlation scores of each patch expert on the holdout fold of

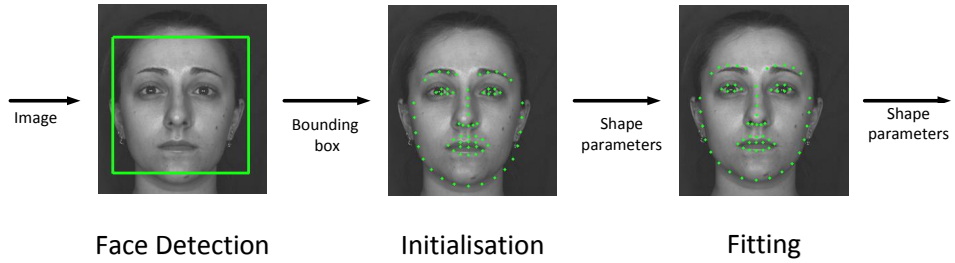


Figure 4.11: The flow diagram for a CLM based landmark detection system. Face detection leads to initial shape parameter (which lead to initial landmark locations), followed by CLM fitting.

training data are computed. This leads to $W = w \cdot \text{diag}(c_1; \ldots; c_n; c_1; \ldots c_n)$, where c_i is the correlation coefficient of the i^{th} patch expert on the hold-out test fold. The i^{th} and $i + n^{\text{th}}$ elements on the diagonal represent the confidence of the i^{th} patch expert. Patch expert reliability matrix W is computed separately for each scale and view. This is a simple but effective way to estimate the error expected from a particular patch. Example reliabilities are displayed in Figure 4.10.

4.5.3 Multi-scale fitting

Patch experts are trained in a particular reference scale (Section 4.4), which is also used to compute the response maps during fitting. There are advantages in both using smaller scales (more robustness) and higher scales (more accuracy) for fitting. I propose a multi-scale approach which combines the benefits of both. That is, instead of using patch experts trained at a single scale, the algorithm can start with patches trained at a lower scale and use higher scales at later iterations.

The multi-scale NU-RLMS approach is defined in Algorithm 3.

4.6 System overview

CLM and most other deformable model fitting techniques are local. Given initial shape parameters, the fitting algorithm looks for optimal

Algorithm 3 Multi scale NU-RLMS algorithm

Require: Image $\mathcal{I}$, initial parameters $\mathbf{p}$, kernel variance ρ, regularisation term r, and patch experts $\{d_i, c_i, \mathbf{w}_i, b_i\}_{i=1}^{n}$
  Compute affine transform $\mathcal{T}$ from image space to patch space
  while num iterations **do**
    Convert image using the affine transform $\mathcal{T}$
    Compute patch responses (Equation 4.31)
    while not converged($\mathbf{p}$) **do**
      Compute mean-shift vectors $\mathbf{v}$ (Equation 4.45)
      Convert them back to image space using $\mathcal{T}^{-1}$
      Compute global PDM parameter update $\Delta\mathbf{p}$ (Equation 4.49)
      Update global parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
      Compute all PDM parameter update $\Delta\mathbf{p}$ (Equation 4.49)
      Update all parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
    end while
    Update $\{d_i, c_i, \mathbf{w}_i, b_i\}_{i=1}^{n}$ to higher scale if available
  end while
  return $\mathbf{p}$

shape parameters within a local area. If the initial parameters are *sufficiently close* to the global optimum, it will be found. This means an extra step is needed to find the initial shape parameters.

In the case of fitting on single images, a face detector (Section 4.6.1) provides a bounding box of the face. This bounding area is used to initialise the rigid shape parameters. As the detector does not provide any information about the facial expression, the non-rigid parameters are all initialised to zero. These initial shape parameters are used as a starting point for CLM fitting. This leads to a full landmark detection system from static images (illustrated in Figure 4.11).

Tracking in video could be done by dealing with each of the images in a sequence independently – applying landmark detection to each of them. This, however, is inefficient as face detection is often more computationally expensive than CLM fitting. Such an approach also ignores the temporal relationships between the shape parameters in an image sequence, as under normal conditions relatively little motion occurs be-

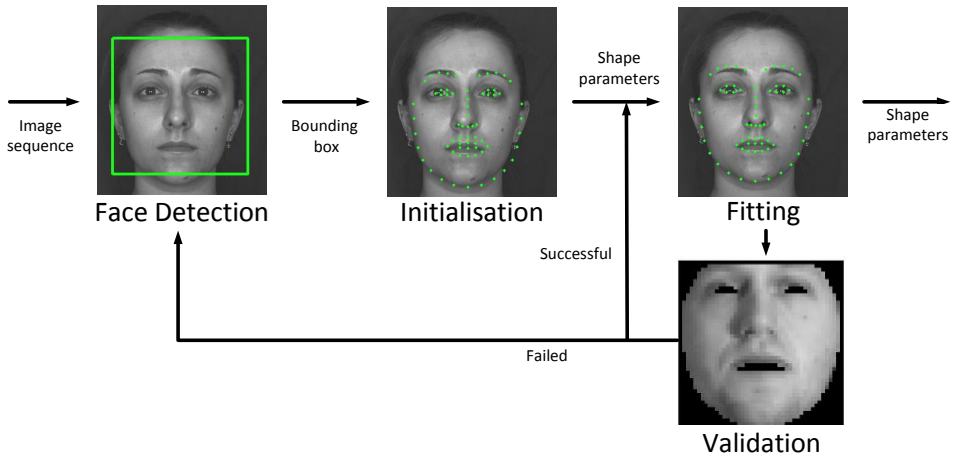
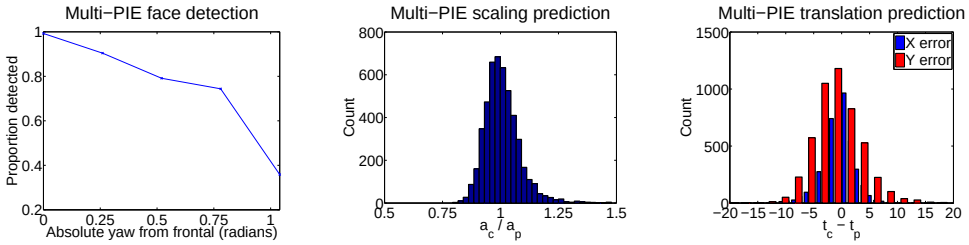


Figure 4.12: The flow diagram for a CLM based landmark tracking system. Face detection leads to initial landmark locations where CLM fitting can start. The success of CLM tracking is checked using a validator, if validation succeeds tracking continues on the next image using the current shape parameters. However, if validation fails tracking reinitialises using a face detector.

tween subsequent frames. Temporal relationships can be exploited by using the estimate of the shape parameters from the previous frame. Such initialisation, however, may eventually lead to drift, as the error will build up and the fitting algorithm will no longer converge on valid feature points. In order to combat drift it is necessary to know if the CLM was successful in locating facial features and inform the tracker to reinitialise using a face detector. This, however, requires an extra validation step that estimates if the landmark detection was successful (Section 4.6.2). A complete facial landmark tracking system is summarised in Figure 4.12.

4.6.1 Face detector

Face detection is a mature field in Computer Vision with a number of off-the-shelf face detectors available in various libraries. Arguably, the most popular face detector to date is the Viola-Jones detector (Viola and



(a) Detection accuracy (b) Scaling prediction accuracy ($s_{\text{corr.}}/s_{\text{pred.}}$) (c) Error in translation ($t_{\text{corr.}} - t_{\text{est.}}$)

Figure 4.13: Evaluation of the Haar-cascade detector. Notice how the accuracy degrades for non-frontal images. Furthermore, the detector often provides bounding areas that are either too small or too large, and often slightly offset. The errors, however, seem to be Gaussian distributed.

Jones, 2004). It is faster than most other approaches and its implementations are readily available in most Computer Vision libraries.

Most of the available implementations of the Viola-Jones detector are for frontal faces, but profile ones exist as well. However, the performance of profile models is usually worse (as illustrated in Figure 4.13a). This is possibly due to a lack of available training data, or the detection of profile faces being an inherently more difficult task.

The Viola-Jones detector provides a list of bounding boxes of faces detected in an image (if multiple faces are detected the biggest is chosen). As most images in the test sets contain a person facing the camera, a frontal face detector is used first, followed by left profile and right profile detections in case of failure. Subsequently, the detected bounding box is used to initialise the rigid shape parameters needed for CLM fitting.

Some detectors have a systematic bias, so a relationship between the bounding box and the rigid shape parameters (scaling, rotation and translation) needs to be learned for each detector individually. This can be easily done by detecting faces in a number of training images and estimating offset and scaling terms.

In my work I used two available implementations of the Viola-Jones

face detector: from OpenCV 2.4.0 and Matlab 2012b Vision toolbox. The OpenCV one is used for landmark tracking in videos and the Matlab one is used for landmark detection in images. The Matlab implementation also provides profile detectors. Since neither of the detectors provide any rotation estimates, the rotation vector $\mathbf{w}$ was initialised to $(0, 0, 0)$ for frontal face detection and $(0, \pm 60^\circ, 0)$ for profile detections.

The evaluation of the Matlab Viola-Jones model on the Multi-PIE dataset is displayed in Figure 4.13a (using a frontal detector followed by a profile one). The errors of rigid parameter estimation from the bounding box can be seen in Figures 4.13b and 4.13c.

4.6.2 Landmark detection validation

In order to combat drift is is necessary to have a way to determine if landmark detection succeeded. I refer to this as *validating* landmark detections. For examples of correct and incorrect landmark detections, see Figures 4.14, and 4.15.

One way to validate landmark detections would be to use the model likelihood (Equation 4.2), however, this measure is not very stable as it differs significantly from image to image and even more from dataset to dataset. Determining the right threshold is very difficult, if not impossible.

Another way to validate landmark detections is to transform the area surrounded by the landmarks to a pre-defined reference shape. The vectorised resulting image can then be used as a feature vector for a classifier which will act as the validator. The use of the reference shape allows for mapping of any landmark configuration to an image of fixed size. It also makes it possible to reduce the effect of facial expression on facial appearance.

As a reference shape the mean shape of the PDM is used (seen in Figure 4.5). This shape is triangulated using Delaunay triangulation. This makes it possible to perform a piece-wise affine warp on each of the

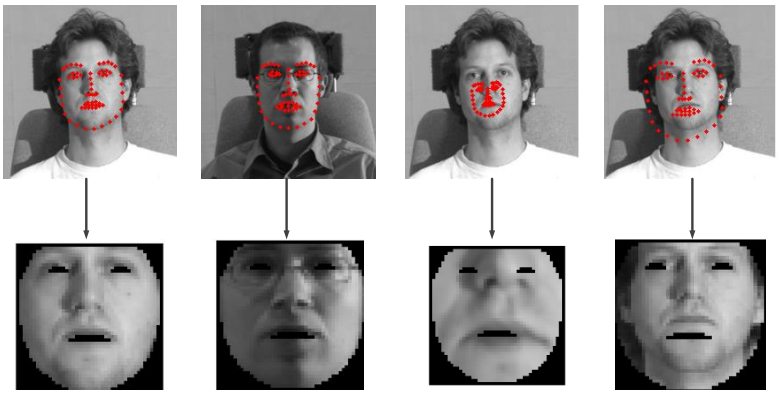


Figure 4.14: Examples of landmark detections and the corresponding warps onto the reference shape.

corresponding triangles in the source and reference shapes. An example of detected feature points and their warps onto a reference shape can be seen in Figure 4.14. The vectorised version of the reference warp can now be used as input into the validator. Furthermore, the per-pixel warping coefficients can be mostly precalculated, with only a limited set of them (per-triangle ones) needing to be recalculated per frame (Matthews and Baker, 2004).

In order to train the classifier on the vectorised reference warp, it needs positive and negative landmark detection examples. Choosing the positive samples is simple, ground truth landmark labels can be used. To generate the negative samples, the ground truth labels can be offset and scaled.

I trained three linear SVM validators using the Multi-PIE and the BU-4DFE datasets. They were trained at three orientations: $(0, 0, 0)$, $(0, 30, 0)$, and $(0, -30, 0)$, so that self occlusions would be easier to deal with.

4.7 Experiments

This section presents the experiments that I conducted in order to explore the effect of the CLM extensions described in this chapter: multi-

modal patch experts, NU-RLMS algorithm, and multi-scale fitting. The benefits of CLM approaches over ASM and AAM have been demonstrated by numerous authors (Cristinacce and Cootes, 2006; Saragih et al., 2011; Wang et al., 2008), hence no comparisons will be made with them in this chapter. A detailed comparison with other state-of-the art landmark detection methods is provided in Chapter 6. Furthermore, this section also demonstrates the usefulness of CLM as a head pose tracker, compared to other dedicated rigid head trackers.

4.7.1 Methodology

Training Data

For patch expert training I used two datasets: BU-4DFE (Section 3.1.2) and the frontally illuminated subset of the Multi-PIE dataset (Section 3.1.1). A quarter of subjects from each of the datasets were reserved for training whereas the rest were used for testing. For Multi-PIE this resulted in 84 subjects and 1713 images for training, and for BU-4DFE in 22 subjects and 130 images – used for synthetic data generation described in Section 3.1.2.

For all of my experiments I used 10^6 training samples per view from both the BU-4DFE and Multi-PIE datasets. Approximately 1701 training samples (81 samples from 1 positive and 20 negative areas) came from a single image, resulting in 587 images in total used for training each of the views. If possible, an equal split of BU-4DFE and Multi-PIE images was used. However, as there were insufficient labelled examples of images from the Multi-PIE dataset at certain views, more BU-4DFE images were used in some cases.

In order to build a model that could work at different orientations I trained separate patch experts at different orientations. In total, 9 sets of patch experts were trained at the following orientations: $(\pm 75, 0, 0)$; $(\pm 45, 0, 0)$; $(\pm 20, 0, 0)$; $(0, 0, 0)$; and $(0, 0, \pm 30)$. The orientation is described in degrees of roll, yaw, and pitch respectively.

Test data

For **landmark detection** experiments I used the remaining 4161 images from the frontally lit Multi-PIE dataset and 424 from the BU-4DFE dataset.

For **head pose estimation** experiments I used three datasets with labelled head pose ground truth: Boston University, Biwi Kinect, and ICT-3DHP head pose datasets (Sections 3.2.3, 3.2.2, and 3.2.1 respectively).

Initialisation

For landmark detection in images, the model parameters were initialised with the use of an off-the-shelf face detector available with Matlab Computer Vision toolbox (Section 4.6.1). The procedure of converting the detected bounding box to initial shape parameters is described in Section 4.6.1.

In the cases where the detector failed (255 out of 4161 images in the Multi-PIE test data, and none in the BU-4DFE data), the rigid shape parameters were initialised by taking the correct values and adding some Gaussian noise. The amount of noise expected was determined by evaluating the face detector used on the Multi-PIE dataset. This gave a realistic initialisation and allowed for the analysis of the CLM approach, without the results being affected by failed face detection.

For video sequence tracking, an OpenCV 2.4.0 the Viola-Jones frontal face detector was used to both initialise and reinitialise tracking if it failed.

Model parameters

The parameter values I used for CLM fitting are provided in this section in order to assist with the reproducibility of the experiments.

For **landmark detection** during RLMS and NU-RLMS fitting the following parameter values were used: regularisation term $r = 25$, mean shift kernel variance $\rho = 1.5$, number of RLMS or NU-RLMS iterations - 3,

area of interest for all three iterations 11×11 pixels, and number of iterations for $\Delta\mathbf{p}$ calculation - 10. The patch training scales used for training and fitting were $s = \{0.25, 0.35, 0.5\}$. In addition, in all but the modality experiments a single modality approach on greyscale intensity images was used.

For **head pose estimation** all of the same parameters were used, with a couple of exceptions. If landmark detection in the previous frame was successful only two RLMS iterations were used of 9×9 and 7×7 pixel areas of interest, respectively. The number of iterations for $\Delta\mathbf{p}$ calculation was 5. The reason for these changes was to speed up the approach to be real-time. Furthermore, surprisingly the smaller areas of interest following successfully tracked frames seemed to lead to better head pose estimation. This may be due to the fact that the face moves little between neighbouring frames and smaller search regions help avoid local optima.

Unless otherwise stated, the experiments in this chapter used the parameter values detailed in this section.

Landmark detection and tracking error metric

In order to measure fitting accuracy, an error metric was needed. I used the Root Mean Square Error (RMSE) between the detected landmarks and the known ground truth locations:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((x'_i - x_i)^2 + (y'_i - y_i)^2)}. \tag{4.50}$$

Above x', y' are the ground truth locations of landmarks; x, y are the detected locations; and N is the number of feature points used. I used a size normalised version of the above error so that it would be possible to compare errors across datasets, and to avoid bias caused by face size. In order to do this, the resulting error was divided by the average of the

width and height of the ground truth shape:

$$\text{RMSE}_{\text{normed}} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left((x'_i - x_i)^2 + (y'_i - y_i)^2\right)}}{0.5 \cdot (\text{width} + \text{height})}. \tag{4.51}$$

Not all of the landmarks were used for RMSE computation. For profile images only the visible points were used for error estimation (e.g. if a person's face fas turned to the left, the left part of the face was not used). The outline of the face (points 1–17) was also unused because of two reasons. First, the outline is very difficult to label consistently across different views leading to inconsistent ground truth. Second, feature spacing in outline can dominate the error – even if all of the detected points are on the face outline they might not correspond well to the ground truth.

Landmark detection error is often visualised in the deformable model fitting community as a convergence vs. error curve (for examples see Figures 4.16, 4.17). The curve is constructed by computing the proportion of images in which the error was below a certain value. The closer the curve is to the top left corner of the graph - the better the fitting. The curve can also reveal accuracy and robustness trade-offs between approaches, and is arguably more informative than median or mean error values.

To aid the understanding of error values some example landmark detections with their RMSE can be seen in Figure 4.15. These examples also give an idea of the RMSE for which the landmark detection could be considered successful and help with the interpretation of the error curves:

- RMSE < 0.02, all landmarks were detected very accurately
- RMSE < 0.05, detection can be considered successful as all of the features have been located, but not necessarily very accurately
- RMSE < 0.1, detection still manages locating most of the facial features, but not all of them

Figure 4.15: Comparing different RMS errors. Notice how the landmark detection can no longer reliably identify all of the regions of the face, with errors above 0.05.

- RMSE > 0.1, detection has failed or is very unreliable

Finally, RMSE is extremely unlikely to be normally distributed (at best it might be skew-normal). Therefore, in all of the cases where statistics were used to compare landmark detection approaches non-parametric tests were used to compare the medians.

Head pose estimation error metric

Most approaches to head pose estimation use the mean absolute angular error for one of the three rotation axes (yaw, pitch and roll). Usually an

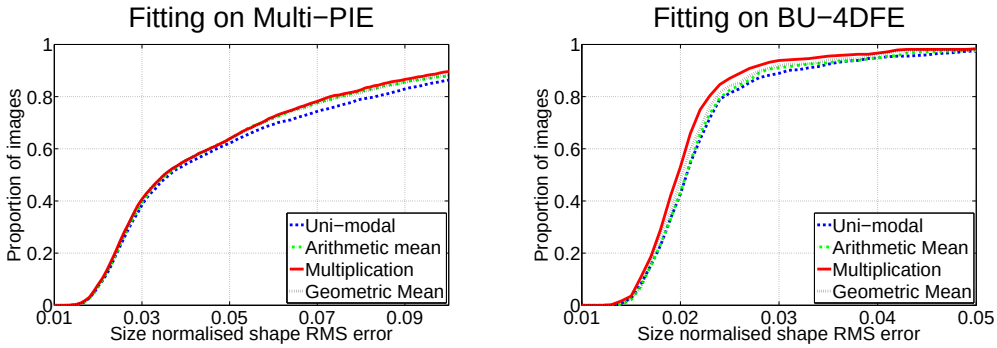


Figure 4.16: Error fitting curves when using uni-modal and multi-modal patch experts for CLM fitting. The gradient intensity combined with greyscale shows clear benefits for fitting on both datasets. Out of the methods for combining the patch responses, the one that multiplies them together fared best.

average error across the orientations is reported as well, resulting in a single statistic which provides insight into the accuracy of competing methods (Murphy-Chutorian and Trivedi, 2009). The main problem of this measure is that error distributions are unlikely to be normal. Thus, the mean error would be negatively influenced by outliers due to drift or occasional miss-classifications. Nevertheless, this metric allows for easy comparison with other researchers' results. Hence, it was computed in my experiments.

In addition to the mean error, the median error was computed too. In head pose tracking, median error is often more informative than the mean, because the latter is strongly affected by outliers. For example, if estimated head pose is off by 100 degrees – it is incorrect; how incorrect it is will affect the mean but not the median error. The median error, thus, reflects the accuracy whereas the mean error reflects the robustness. However, as most authors only provide mean values, using median errors makes it difficult to compare approaches.

4.7.2 Multi-modal patch experts

I conducted a set of experiments to determine if the use of multi-modal patch experts is helpful for CLM fitting. In addition, this set of experiments explored the effect of different patch response map aggregation techniques: multiplication, arithmetic mean or geometric mean. The experiments were carried out on Multi-PIE and BU-4DFE datasets.

Results

The results of the modality experiments on both of the datasets can be seen in Figure 4.16.

A Friedman's ANOVA was conducted to compare the effect of adding a gradient intensity modality on the RMS errors on the BU-4DFE dataset. There was a significant effect of modality, $\chi^2(3) = 205.2, p < 0.001$. Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that multiplication (Mdn $= 0.0197$) and geometric mean (Mdn $= 0.0202$) outperformed the uni-modal version (Mdn $= 0.0205$) and the arithmetic mean version (Mdn $= 0.0204$) at significance level $p < 0.001$. Furthermore, combining the response maps using multiplication outperformed the geometric mean version $p < 0.001$.

A Friedman's ANOVA was conducted to compare the effect of adding a gradient intensity modality on the RMS errors on the Multi-PIE dataset. There was a significant effect of modality, $\chi^2(3) = 300.5, p < 0.001$. Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the multimodal approaches: multiplication (Mdn $= 0.0350$), arithmetic mean (Mdn $= 0.0356$) and geometric mean (Mdn $= 0.0350$) outperformed the uni-modal version (Mdn $= 0.0363$) at significance level $p < 0.001$. Furthermore, combining the response maps using multiplication outperformed the two other multi-modal approaches $p < 0.001$.

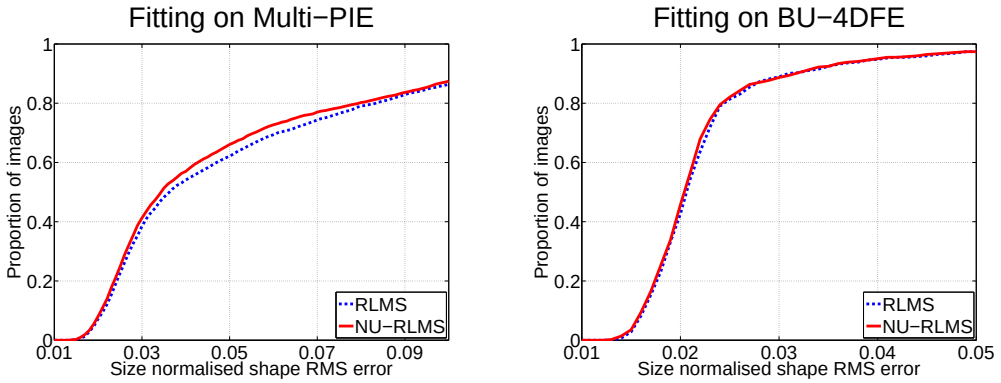


Figure 4.17: Error fitting curves when using RLMS and NU-RLMS algorithms. NU-RLMS increases landmark detection accuracy on both of the datasets, although the increased accuracy is marginal on the BU-4DFE dataset.

Discussion

The results show that the patch responses from an additional patch expert lead to an improved fitting accuracy on both of the datasets. Finally, they demonstrate that the multiplication method for aggregating the patch response maps significantly outperforms the other methods on both of the datasets.

4.7.3 Non-Uniform Regularised Landmark Mean Shift

To see the effect of my new NU-RLMS algorithm on fitting accuracy, I conducted a fitting experiment on the Multi-PIE and BU-4DFE datasets. In order to construct the weight matrix W, I used patch expert correlations with $w = 7$.

Results

The comparison of NU-RLMS with RLMS can be seen in Figure 4.17. A Wilcoxon signed rank test was performed to compare the RMS errors under the different fitting strategies for different datasets. For Multi-PIE fitting there was a significant difference in the errors for RLMS (Mdn $= 0.0363$) and NU-RLMS (Mdn $= 0.0343$), $z = -19.1, p < 0.001$. For

BU-4DFE dataset there was also a significant difference in the errors for RLMS (Mdn $= 0.0232$) and NU-RLMS (Mdn $= 0.0229$), $z = -6.26, p < 0.001$.

Discussion

The results indicate that, on both Multi-PIE and BU-4DFE, NU-RLMS is statistically significantly more accurate than RLMS. In conclusion, the above results demonstrate the benefits of not treating each of the patch experts equally and taking their reliability into account. However, the amount of improvement seems to depend on the dataset used, and in the case of BU-4DFE the improvement was quite small. This can possibly be explained by the simplicity of the dataset, leaving very little room for improvement.

4.7.4 Multi-scale fitting

I conducted a set of experiments to evaluate how the CLM fitting is affected by the scaling term of the patch expert. Patch experts were trained using the following scales: $s = \{0.25, 0.35, 0.5\}$. First, fitting was done using only one of the scales during all three RLMS iterations. For fairness, the area of interest was adjusted for each scale, so that all of the conditions saw the same amount of the image (resulting in an added computational cost for larger scales). Secondly, I wanted to see if a multi-scale approach improved performance over a single-scale approach. The area of interest used for the multi-scale approach was 11×11 pixels (same as the $s = 0.25$ case), hence the same computational cost.

Results

The results of the experiments can be seen in Figure 4.18.

A Friedman's ANOVA was conducted to compare the effect of the scaling used on the RMS errors on the BU-4DFE dataset. There was a significant effect of scaling, $\chi^2(3) = 850.9, p < 0.001$. Friedman's ANOVAs

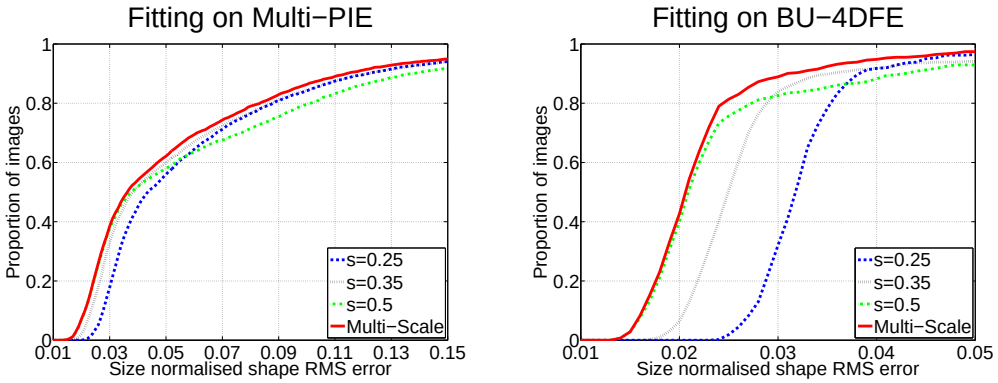


Figure 4.18: Error fitting curves when using differnt scale patch experts for CLM fitting. Notice how the robustness degrades as the scaling term grows larger (more images with high error), but the accuracy improves (more images with low error) leading to a trade-off. Multi-scale formulation, however, manages to retain both robustness and accuracy leading to better performance.

were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed significant differences between all of the conditions ($p < 0.01$). The median RMSE error values for different conditions are as follows: scaling of 0.5, Mdn $= 0.0208$; scaling of 0.35, Mdn $= 0.0249$; scaling of 0.25 - Mdn $= 0.0319$; and the multi-scale approach, Mdn $= 0.0205$.

A Friedman's ANOVA was conducted to compare the effect of the scaling used on the RMS errors on the Multi-PIE dataset. There was a significant effect of scaling, $\chi^2(3) = 2954.4, p < 0.001$. Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed significant differences between all of the conditions ($p < 0.001$). The median RMSE error values for different conditions are as follows: scaling of 0.5, Mdn $= 0.0374$; scaling of 0.35, Mdn $= 0.0382$; scaling of 0.25, Mdn $= 0.0435$; and the multi-scale approach, Mdn $= 0.0363$.

Discussion

The above graphs (Figure 4.18) show the trade-off expected from using patches trained at different scales – lower scale patch experts are more robust but less accurate than higher scale ones. This distinction is particularly clear on the BU-4DFE dataset, where the lowest scale reaches similar accuracy to a multi-scale formulation for RMSE below 0.05. Furthermore, the results also demonstrate that multi-scale based fitting manages to capture both robustness and accuracy.

Another interesting result is that different scalings perform best on different datasets. On BU-4DFE the $s = 0.5$ performed best, whereas on the Multi-PIE $s = 0.35$ led to the best accuracy. This is possibly because BU-4DFE faces are frontal, leading to better initialisation and reducing the need of lower scale search. In Multi-PIE, on the other hand, equivalent initialisation is difficult to provide, making initial lower scale search more important.

4.7.5 Head pose estimation

As a final test I wanted to see how CLM facial tracking compares to other methods for estimating head pose. This also acted as a proxy evaluation for feature point tracking in video sequences: good feature point detection leads to accurate head pose estimation.

I compared the CLM model to several state-of-the-art dedicated head pose trackers: Generalised Adaptive View-based Appearance Model (Morency et al., 2008), and regression forests on depth maps (Fanelli et al., 2011a). The CLM used is described in the Methodology section (Section 4.7.1).

The results of head pose tracking experiments can be seen in Table 4.2. CLM exhibits similar or superior performance to head pose trackers that rely purely on intensity information. This indicates that CLM can act successfully as a head pose tracker. However, it seems to slightly under perform when compared to trackers that take depth information into account as well (when looking at the mean errors but not median

Model	Yaw	Pitch	Roll	Mean	Mdn.
	Boston University				
GAVAM (Morency et al., 2008)	**3.85**	4.55	**2.20**	**3.53**	**2.12**
CLM	4.31	**4.00**	2.50	3.60	2.26
	ICT-3DHP				
GAVAM (Morency et al., 2008)	6.58	5.01	**3.50**	5.03	3.08
CLM	5.41	**4.32**	4.83	4.85	**2.45**
	ICT-3DHP with depth				
GAVAM (Morency et al., 2008)	**3.76**	5.24	4.93	**4.64**	2.91
Reg. for. (Fanelli et al., 2011a)	7.69	10.66	8.72	9.03	5.02
	Biwi-HP				
GAVAM (Morency et al., 2008)	14.16	9.17	12.41	11.91	5.63
CLM	10.32	10.27	9.01	9.87	**3.58**
	Biwi-HP with depth				
GAVAM (Morency et al., 2008)	**6.75**	**5.53**	10.66	**7.65**	3.92
Reg. for. (Fanelli et al., 2011a)	9.2	8.5	**8.0**	8.6	NA

Table 4.2: Estimating head pose using CLM and other baselines. The datasets tested on were: ICT-3DHP, the Biwi Kinect head pose, and the Boston University dataset. Notice the comparable accuracy of the CLM approach.

errors). In sum, CLM can be used for head pose estimation, but if depth data is available other approaches might be preferable.

4.7.6 Conclusions

The experimental results demonstrate the error rates expected from CLM landmark detection and head pose tracking. Furthermore, they show the benefits of three proposed extensions to CLM facial tracking accuracy. First, the CLM approach can be extended to use multiple visible light based channels (greyscale and gradient intensity) for better landmark detection accuracy. Second, treating each of the patch response maps with different reliability by using NU-RLMS leads to more accurate tracking. Finally, using a multi-scale CLM fitting leads to more robust and accurate landmark detection.

4.8 CLM issues

Much progress has been made to make CLM fitting and tracking more accurate, including several extensions outlined in the previous sections. However, the CLM approach described in this chapter is not without limitations. There are three main identifiable situations in which CLM landmark detection and tracking fails or is inaccurate: large variations in pose, illumination, and expression. The same factors affect most face tracking and landmark detection approaches, and are notoriously difficult to solve.

This section describes the experiments I performed to better understand the limitations of the CLM landmark detector. Specifically, I explored how landmark detection accuracy is affected by pose, illumination, and expression. The observations which follow help us to understand the existing limitations of CLM.

4.8.1 Illumination

Even if the face is in the same pose and has the same expression, the captured image will depend very much on the illumination present in the scene (see Figure 3.3). However, landmark detection and pose estimation approaches should not depend on the illumination, as it does not reveal affective information. Due to the huge effect that illumination has, it is very difficult to build landmark detectors which are completely illumination independent. It is, however, worthwhile making them as robust to illumination changes as possible, especially if they need to work in unconstrained and naturalistic environments.

CLM based approaches tend to generalise well to unseen faces under the same illumination, however, the landmark detection accuracy degrades rapidly in unseen illumination (not present in the patch training data). Some examples of a CLM trained on frontally lit faces (such as Figure 3.3a), but tested on different illuminations can be seen in Figure 4.19. Note how landmark detection is affected by the shadows and uneven lighting.

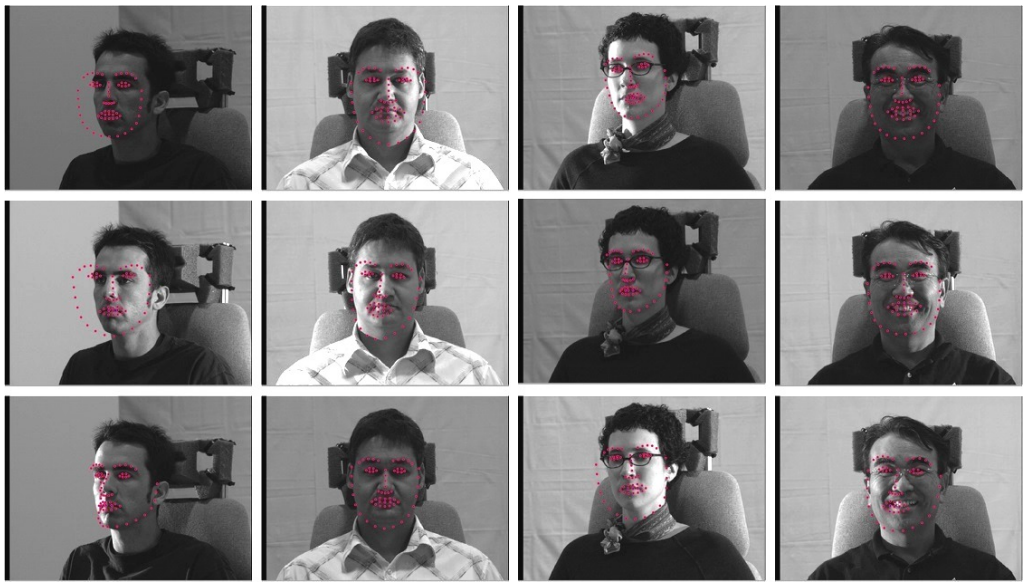
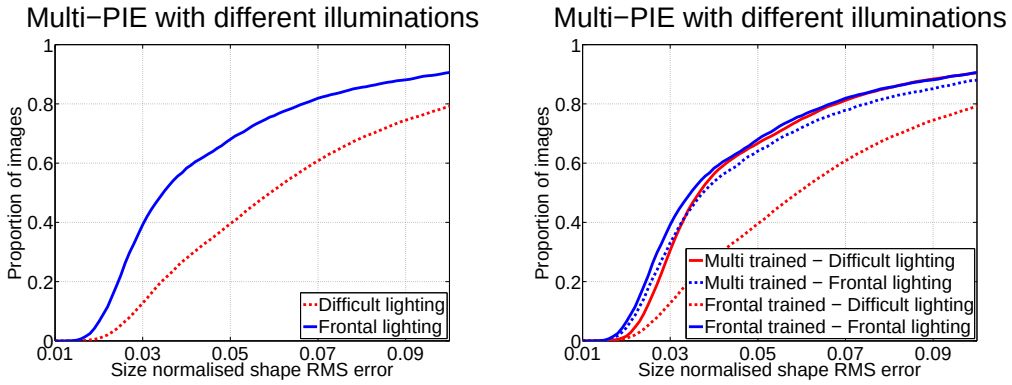


Figure 4.19: Some common failure cases across different illumination fitting. Note how the strong shadow on the face affects fitting, with the shadow being identified as the nose ridge and tip.

I constructed a set of experiments to demonstrate the effect of illumination on the Multi-PIE dataset. For the following experiment, SVR based patch experts were trained on frontally lit faces (using the same data as in Section 4.7.2) and using NU-RLMS multi-scale and multi-modal CLM fitting. The fitting was performed on frontally lit faces and on three difficult lighting conditions (Section 3.1.1) to test the CLMs ability to generalise to unseen illumination (examples of such lighting can be seen in Figure 3.3).

The experimental methodology was the same as outlined in Section 4.7, but with an additional test set that included unseen lighting conditions: left, right, and poorly lit.

The results of fitting on seen and unseen lighting are given in Figure 4.20a. A Wilcoxon rank sum test was performed on the RMSE errors on the two different test sets. It revealed significantly worse performance of CLM on the general illumination test set ($z = 37.7, p < 0.001$).



(a) (b)

Figure 4.20: Fitting on the Multi-PIE dataset using differently trained SVR experts. (a) Fitting single light trained CLM on different lighting conditions. Observe a huge degradation in accuracy when fitting to unseen lighting. The performance degrades significantly when fitting on unseen illumination. (b) Fitting CLM on different lighting conditions together with extra training at different light conditions. Pay special attention to the dashed blue curve and how fitting on frontal lighting degrades because of more general training. However, if a more general expert is trained, performance generally is improved, as shown by the solid red curve.

Even though CLM tries mitigating the effect of lighting variation through using intensity normalised patch experts the above result suggests this does not solve the problem. Given the results, CLM clearly shows limited generalisability across illumination for landmark detection in images.

Simple approach to lighting issues

A naïve approach to solving the lighting invariance issue would be to use more varied lighting conditions when training the patch experts. For example, left, right and poorly lit faces could be included alongside frontally lit ones during the patch expert training. I conducted an experiment to see if this general illumination training helps with fitting on difficult lighting conditions. Instead of training on just frontally lit

faces, the SVR based patch experts were trained on the four lighting conditions (frontal, dim, left and right). The same experimental conditions as in previous sections were used. In order for the results not to be affected by the accuracy of the face detector, the bounding box from a detector run on a frontally lit face was used for all four images of different illumination.

In Figure 4.20b one can see the results of fitting on different illuminations when using frontal and more general training. A Wilcoxon sign rank test revealed that there was an improvement on fitting accuracy on the difficult illumination case when a general illumination training was used ($z = 77.5, p < 0.001$). However, the performance on the frontal lighting case decreased, when more general patch experts were trained ($z = -21.4, p < 0.001$).

The extra training helps on the difficult lighting condition, however, it still does not reach the performance that is achieved by using frontal trained patches on frontal lit faces. Also, the improved performance on difficult lighting comes at the expense of degraded performance on frontally lit faces. That is, if the CLM is made more robust, it is at the expense of accuracy. A possible reason for this is that a simple linear SVR patch expert can not learn the complex relationships between pixel values under different illuminations and the landmark alignment probabilities.

I also wanted to see the effect of the differently trained patch experts on a single lighting condition dataset. The results of using the single and general illumination trained experts on the BU-4DFE dataset can be seen in Figure 4.21. A Wilcoxon sign rank test reveals significantly worse performance of the general illumination patch experts ($z = -13.69, p < 0.001$). The results confirm that the use of more general patch experts leads to worse fitting performance.

The above results highlight two major problems of the SVR patch expert based CLM approach on visible light images. These are an inability to generalise to unseen lighting conditions and a reduced overall accuracy

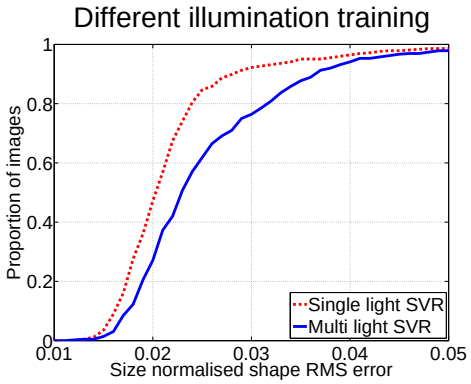
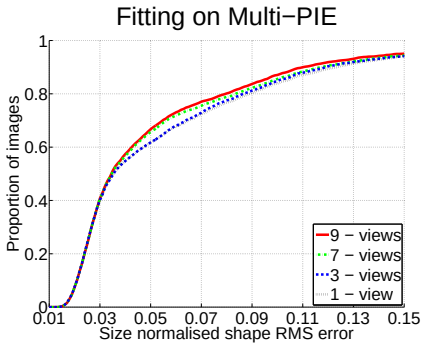
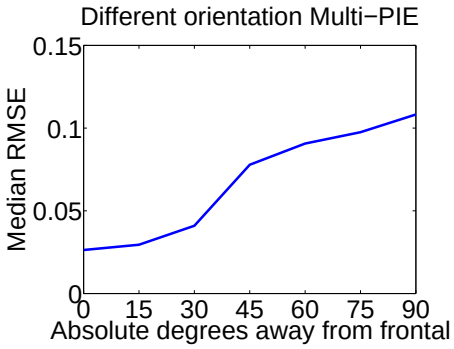


Figure 4.21: Fitting with single and multi-light patch experts on BU-4DFE dataset. Note the decreased performance when a more general patch expert is used.



(a) Fitting on Multi-PIE using different numbers of views. Observe how accuracy increases with additional views.



(b) Error distribution across different orientations. Observe how error increases with off frontal views even with multi-view patch experts.

Figure 4.22: Analysis of CLM fitting at different orientations and with additional views

if more general training is provided. Ideally, landmark detection would work equally well, or at least comparatively, under different lighting conditions. This is not the case for CLM.

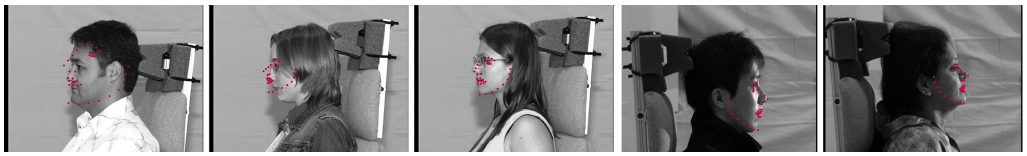


Figure 4.23: Some common failure cases with across pose CLM landmark detection.

4.8.2 Pose

Another major issue that CLM faces is the degradation of landmark detection accuracy on non-frontal images. I analysed the fitting results from Section 4.7.2 to see how CLM accuracy depends on the orientation of the face being tracked. Figure 4.22b shows landmark detection accuracy based on the distance of the pose (in degrees) from a frontal one. It can be seen that landmark detection accuracy degrades with more non-frontal poses.

There are multiple reasons for the degradation of results at different orientations. First, with fewer points available for tracking, accurate estimation becomes more difficult. Second, fewer non-synthetic profile training images were used when compared to frontal training, potentially leading to worse performance. Some examples that illustrate the difficulty of fitting on non-frontal images of faces can be seen in Figure 4.22b.

Adding extra views

An additional test was performed to see if the addition of extra views to training is beneficial to CLM landmark detection. In total nine sets of patch experts were trained: $(0, \pm 75, 0)$, $(0, \pm 45, 0)$, $(0, \pm 20, 0)$, $(0, 0, \pm 30)$, and $(0, 0, 0)$.

I performed landmark detection on the Multi-PIE dataset under the following conditions: single frontal view; three views - frontal and profiles; 7 views – frontal, up-down, and side views (without $(0, \pm 45, 0)$; and all 9 views.

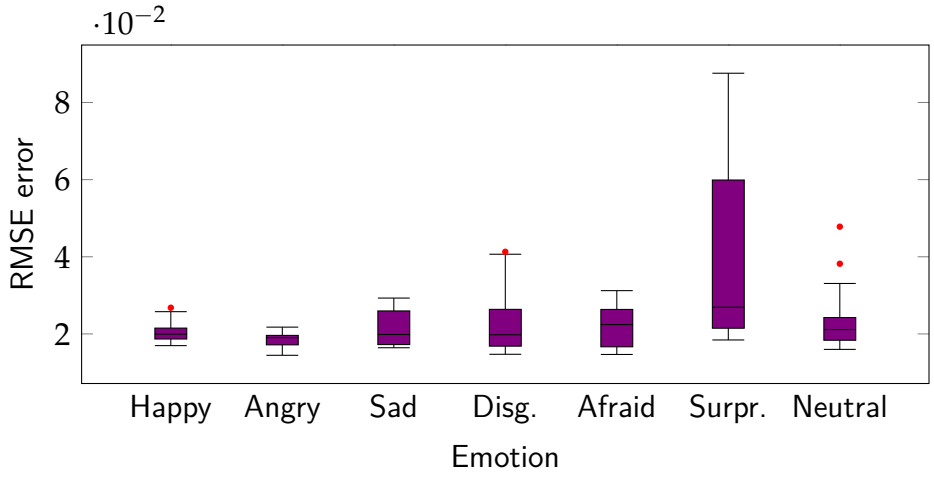


Figure 4.24: Error rates on the BU-4DFE dataset on different emotion images. Observe the difference in error in expression of surprise. This is due to the widely opened mouth which CLM finds difficult to detect correctly.

A Friedman's ANOVA was conducted to compare the effect of the additional views on the RMS errors on the Multi-PIE dataset. There was a significant effect of views ($\chi^2(3) = 340.7, p < 0.001$). Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed significant differences ($p < 0.001$) between all but two conditions: 9-views vs. 7-views and 3-views vs. 1-view . The median RMSE error values for different conditions were as follows: single view - Mdn = 0.0353, 3 views - Mdn = 0.0353, 7 views - Mdn = 0.0346 and 9-views - Mdn = 0.0345. Please note that the small improvement of adding extra views is affected by the fact that the majority of test images are close to frontal.

The results of this experiment are shown in Figure 4.22a, where it can be clearly seen that adding extra views is beneficial. However, although it helps with landmark detection across pose it still does not solve it fully.

4.8.3 Expression issues

Illumination and head pose variations are not the only things affecting the CLM landmark detection accuracy. CLM fitting accuracy also suffers

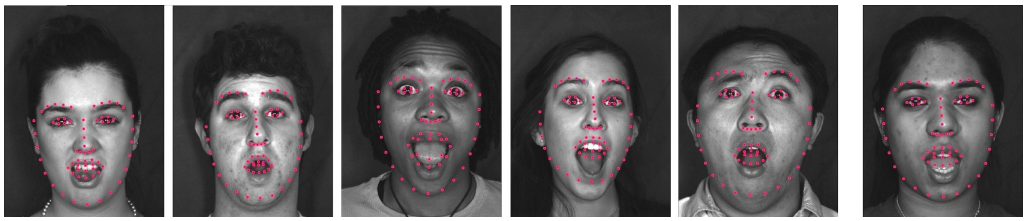


Figure 4.25: Some common failure cases with landmark estimation of surprised expression. Notice how most of the errors come from the inability to reliably detect the lower lip.

in the presence of extreme variations of expression.

To illustrate this, I analysed the results from Section 4.7.2 on the BU-4DFE dataset which had $\approx$ 60 images for each of the basic emotions and of neutral faces. Since the emotional expressions in this dataset are posed based on the Ekman basic emotions, they have similar feature point configurations. This reveals how the fitting accuracy is affected by the type of expression.

Results from this analysis can be seen in Figure 4.24. It can be seen that landmark detection accuracy degrades for the expression of surprise, which is a clear outlier. Informal analysis reveals that this is mostly due to large mouth opening present in the expression (see Figure 4.25).

There are two main reasons for the degradation of performance in expression of surprise. Firstly, the patch experts for lower lip might not be able to capture the many possible variations of appearance: closed lips, open lips with teeth present, and open lips without teeth present. Secondly, a prior imposed on the parameters of the shape model, penalises shapes which are complex and far away from neutral face. This results in worse performance on large expressions.

4.8.4 Discussion

The pose and illumination issues outlined above are addressed in the following chapters, suggesting how CLM fitting can be made more robust in the presence of lighting and pose variations. The work I have

done on this can be split into three parts. The first one deals with a way of approaching lighting and pose difficulties by using depth/range scanner data in addition to visual light, leading to a CLM-Z tracker (Chapter 5). Second, I describe how a CLM tracker can be combined with a dedicated pose tracker to lead to better head pose tracking accuracies (Section 5.6). Finally, I present my CLNF model for face tracking which exploits an advanced patch expert that addresses all of the issues outlined, and especially with the illumination generalisation (Chapter 6).

4.9 General discussion

In this chapter, I provided a detailed description of CLM. The model can be used for facial landmark detection in images, facial landmark tracking in video sequences and head pose estimation. As most of the work in Chapters 5 and 6 is based on CLM, this chapter also serves as detailed background explanation for this type of deformable model.

Also in this chapter, I presented and explored three extensions to the model to make it more accurate and robust: multi-modal patch experts, multi-scale fitting and NU-RLMS.

Finally, I outlined the issues facing CLM facial tracking that need to be resolved for it to be useful in real-world environments: failure to generalise across illuminations, poor performance across pose, and decreased accuracy for extreme expressions. These problems guided my work, leading to CLM extensions presented in later chapters.

5 CLM-Z

In this chapter, I present a 3D Constrained Local Model (CLM-Z) which takes advantage of both 3D geometry (depth data) and visible light images to detect facial features in images, track them across video sequences and estimate head pose. The use of depth data allows the approach to mitigate the effect of illumination and make the tracking more robust to pose variations. An additional advantage of CLM-Z is the option to use only depth information when no visible light signal is available or lighting conditions are inadequate.

The benefits of CLM-Z over regular CLM are demonstrated by evaluating it on four publicly available datasets: the Binghamton University 3D dynamic facial expression database (BU-4DFE) (Yin et al., 2008), the Biwi Kinect head pose database (Biwi) (Fanelli et al., 2011b), the Boston University head pose database (BU) (Cascia et al., 2000), and my collected dataset ICT-3DHP (Baltrušaitis et al., 2012). A more detailed description of the datasets can be found in Section 3. The experiments show that the CLM-Z method significantly outperforms existing state-of-the-art approaches both for person-independent facial feature tracking (convergence and accuracy) and head pose estimation accuracy.

Finally, this chapter presents a way to combine CLM landmark detector with a dedicated head pose tracker, leading to better head pose estimation accuracy.

5.1 Depth data

It is possible to recover 3D scene geometry through use of specialised hardware or algorithms. There are many ways to capture such information, and I briefly describe some of the most popular ones: multi-view stereo, active stereo, and time-of-flight cameras.

Multi-view stereo approaches compute the disparity between corresponding points in stereo images, which can then be converted to sparse or dense depth maps. This approach requires more than one calibrated camera (or a single moving camera and a static scene). Sometimes, multi-view stereo techniques use more than two cameras leading to more accurate reconstructed scenes. High-end range scanners use this technique to recover very accurate scene representations.

Active stereo techniques combine a camera with a projector which projects known light patterns onto the scene. Using suitable known patterns, it is possible to work out the depth of the pixel seen in the devices imaging camera. An example of such a system is Microsoft Kinect (Zhang, 2012), which is the first mass-market product to combine an infra-red-based active stereo system with a colour video camera in a single case for a competitive price. The depth sensor operates in infra-red light to avoid interference with the scene which is captured by the colour video camera.

Time-of-flight cameras record depth by measuring the time it takes a light signal to travel from the camera to an object and back. Current models cannot record colour images, but only an infra-red intensity image. Time-of-flight cameras have many industrial applications, for example in robotics and automotive industry, and they are also increasingly used in computer graphics.

5.1.1 Representation

A common way to represent the captured scene geometry is by using a depth map, where each pixel in the depth map represents how far away the object is from the camera plane. Figure 5.1, from Shim and

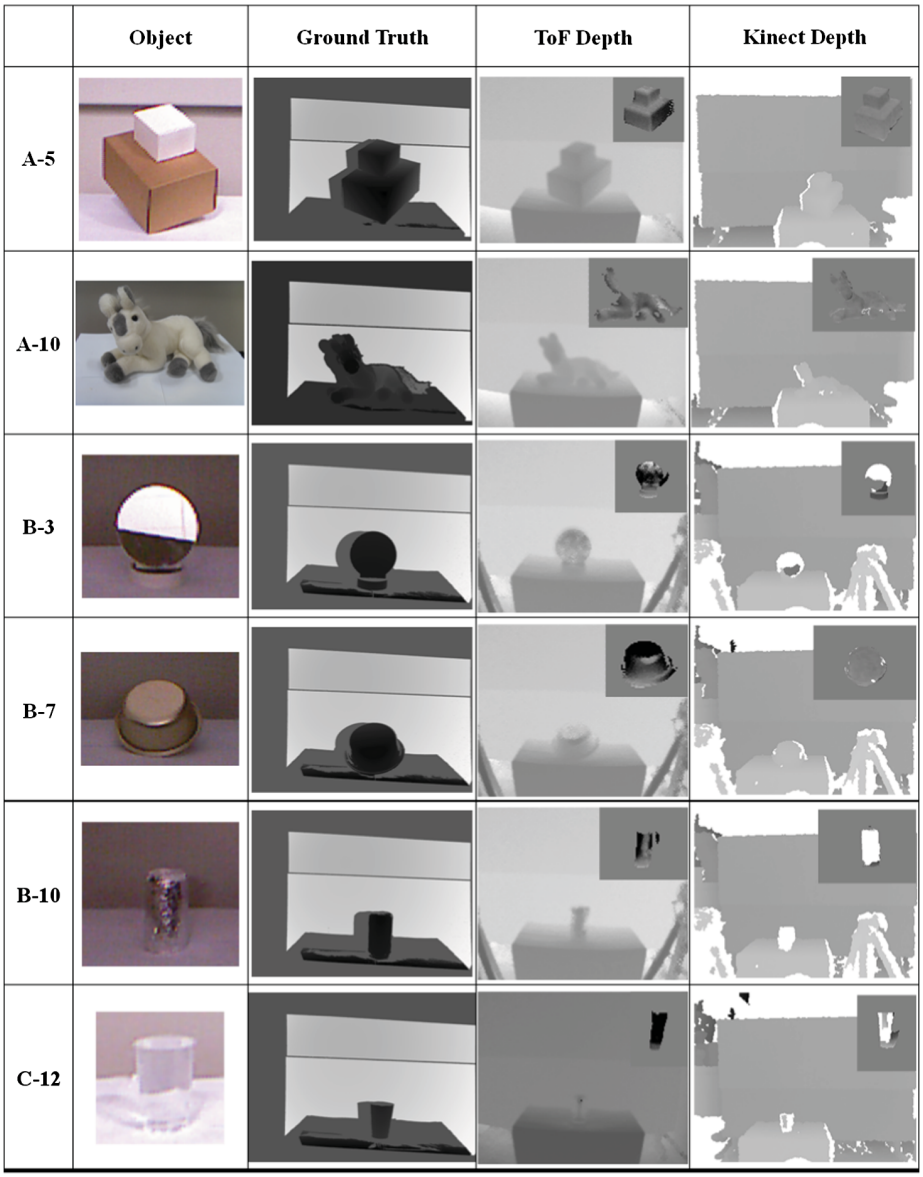


Figure 5.1: Samples of the actual scene, the geometry and the reconstructions of the geometry using two range sensing devices. Taken from Shim and Lee (2012)

Lee (2012), shows examples of depth maps of the same scene produced using an active stereo and a Time-of-flight camera.

The main benefit of such a scene imaging is that it is not affected by the illumination which visible light images are particularly sensitive to. Moreover, it provides an alternative view of the scene, allowing for additional analysis. However, depth maps of scenes are not without issues - there might be gaps appearing in the depth image due to occlusions, reflections and shadows. Hence, they require specialised algorithms for efficient analysis.

In my work I used depth data in the form of depth maps collected using two of the above listed techniques: multi-view stereo and Microsoft Kinect sensor.

5.2 Model

The CLM-Z model is a special instance of CLM introduced in the previous chapter. It uses the same Point Distribution model which can be described by parameters $\mathbf{p} = [s, \mathbf{w}, \mathbf{q}, \mathbf{t}]$: the scale factor s, object rotation $\mathbf{w}$, 2D translation $\mathbf{t}$, and a vector describing non-rigid variation of the shape $\mathbf{q}$. See Section 4.2.2 for more details.

5.3 Patch experts

The main difference between CLM and CLM-Z is the patch experts used. I introduce a novel patch expert that is based on a depth map and not on a visible light image (greyscale or the gradient intensity of greyscale). The patch expert is similar to the SVR based ones used in the previous chapter, however with one crucial difference: the normalisation function that deals with missing data present in depth data.

The depth based patch expert evaluated on a depth map $\mathcal{Z}$ at a pixel location $\mathbf{x}_i$ can be defined as follows:

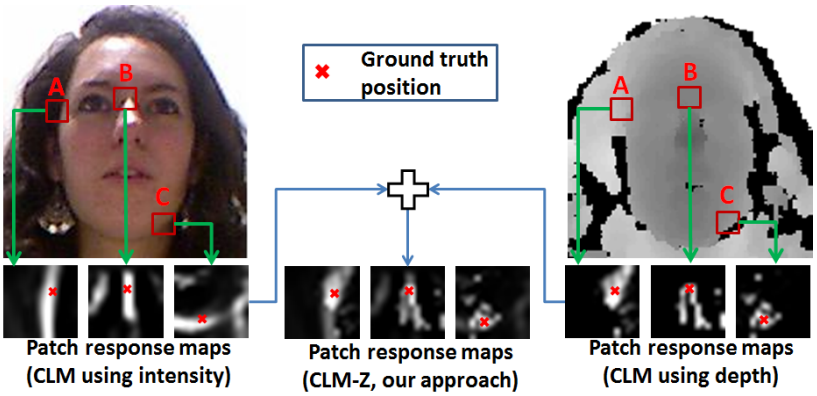


Figure 5.2: Response maps of three patch experts: (A) face outline, (B) nose ridge and (C) part of the chin. SVR patch expert response maps using greyscale intensity contain strong responses along the edges, making it hard to find the actual feature position. By integrating response maps from both intensity and depth images, the CLM-Z approach mitigates the aperture problem.

$$p(l_i|\mathbf{x}_i, \mathcal{Z}) = \frac{1}{1 + e^{d\mathcal{C}_{\mathcal{Z},i}(\mathbf{x}_i;\mathcal{Z})+c}}, \tag{5.1}$$

where $\mathcal{C}_{\mathcal{Z},i}$ is the outputs of depth patch regressor (SVR), for the i^{th} feature, c is the logistic regressor intercept, and d the regression coefficient.

$$\mathcal{C}_{\mathcal{Z},i}(\mathbf{x}_i;\mathcal{Z}) = \mathbf{w}_{\mathcal{Z},i}^T \mathcal{P}_{\mathcal{Z}}(\mathcal{W}(\mathbf{x}_i;\mathcal{Z})) + b_{\mathcal{Z},i}, \tag{5.2}$$

where $\{\mathbf{w}_i, b_i\}$ are the weights and biases associated with a particular SVR. Here $\mathcal{W}(\mathbf{x}_i;\mathcal{Z})$ is a vectorised version of $n \times n$ image patch centered around $\mathbf{x}_i$.

$\mathcal{P}_{\mathcal{Z}}$ ignores missing values in the patch when calculating the mean. It then subtracts that mean from the patch and sets the missing values to zero. Finally, the resulting patch is normalised to unit variance. This is crucial when dealing with depth data which can have missing values.

For intensity and gradient images $\mathcal{P}_{\mathcal{I}}$ is used, which normalises the vectorised patch to zero mean and unit variance. Due to the potential of

missing data caused by occlusions, reflections, and background elimination, $\mathcal{P}_{\mathcal{I}}$ is not used on depth data. Instead, a robust $\mathcal{P}_{\mathcal{Z}}$ is used. Using $\mathcal{P}_{\mathcal{I}}$ on depth data leads to missing values skewing the normalised patch (especially around the face outline), resulting in decreased performance (see Figure 5.4).

Depth patch experts are insensitive to lighting conditions, making them robust. However, they are not as accurate as the intensity based ones for fine grained feature detection, as demonstrated in the experiments section. Therefore, they need to work with the visible light patch experts for best performance.

There are several options for combining the response maps from the depth and visible light patch experts. This is similar to the multi-modal patch expert case in Section 4.3.2. There are three main options, as before: arithmetic mean, geometric mean and multiplication. In my experiments I found little difference between these methods, so arithmetic mean can be chosen for speed.

Example images of intensity, depth and combined response maps (the patch expert function evaluated around the pixels of an initial estimate) can be seen in Figure 5.2. A major issue that CLMs face is the aperture problem, where detection confidence across the edge is better than along it. This is especially apparent for nose ridge and face outline in the case of intensity response maps. Addition of the depth information helps with solving this problem, as the strong edges in both images do not correspond exactly, providing further disambiguation for points along strong edges.

5.4 Fitting

For CLM-Z fitting, the same strategy as used in the previous chapter for CLM fitting can be used. However, it requires an additional step of calculating responses from depth patch experts. The CLM-Z fitting approach is summarised in Algorithm 4. For better accuracy, depth

based patch experts should not be used during the last RLMS iteration. They are more robust, but less accurate (especially at larger scales).

Algorithm 4 CLM-Z RLMS algorithm

Require: $\mathcal{I}, \mathcal{Z}$ and $\mathbf{p}$, kernel variance ρ, regularisation term r, patch experts
  Compute affine transform $\mathcal{T}$ from image space to patch space
  while num iterations **do**
    Convert image to using the affine transform $\mathcal{T}$
    Compute intensity patch responses (Equation 4.31)
    Compute depth patch responses (Equation 5.1)
    Combine the response maps
    while not converged($\mathbf{p}$) **do**
      Compute mean-shift vectors $\mathbf{v}$ (Equation 4.45)
      Convert them back to image space using $\mathcal{T}^{-1}$
      Compute PDM parameter update $\Delta\mathbf{p}$ (Equation 4.46)
      Update parameters $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$
    end while
  end while
  return $\mathbf{p}$

5.5 Training data

The synthetic depth images can be used to train the patch experts in the same way that visible light images are (except for the different normalisation). The same sampling technique can be used to generate expected patch expert responses (with same $\sigma = 1$).

For training the depth based patch experts I used the synthetic depth data generated from the BU-4DFE dataset described in Section 3.1.2, an example of such synthetic images with landmark labels can be seen in Figure 5.3. For training the visible light based patch experts Multi-PIE and BU-4DFE datasets were used (same as in Section 4.7.1). For all of my experiments I used 10^6 training samples in total from both the BU-4DFE and Multi-PIE datasets for visible light patch experts, and only BU-4DFE for the depth patch experts. As before, patch experts were

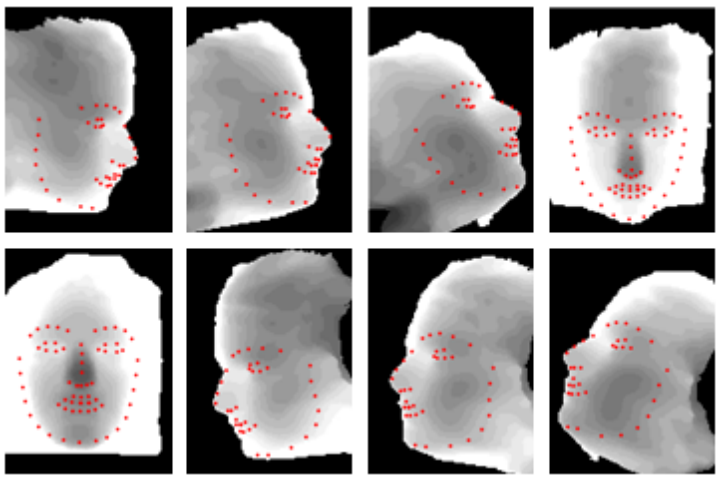


Figure 5.3: Examples of synthetic depth images used for training. Closer pixels are darker, and black is missing data. Notice how face outline and certain feature points are difficult to identify from the depth images.

trained at the following orientations: $(\pm 75, 0, 0)$; $(\pm 45, 0, 0)$; $(\pm 20, 0, 0)$; $(0, 0, 0)$; $(0, 0, \pm 30)$; $(0, 0, 30)$.

5.6 Combining rigid and non-rigid tracking

Because non-rigid shape based approaches, such as CLM, do not provide an accurate pose estimate on their own (see Section 5.7.6), it is possible to combine a CLM tracker with an existing rigid pose tracker. For a rigid head pose tracker a Generalised Adaptive View-based Appearance Model (GAVAM) introduced by Morency et al. (2008) can be used. The tracker works on image sequences and estimates the translation and orientation of the head in three dimensions with respect to the camera, in addition to providing an uncertainty associated with each estimate.

GAVAM is an adaptive keyframe based differential tracker. It uses 3D scene flow (Vedula et al., 1999) to estimate the motion of the frame from keyframes. The keyframes are collected and adapted using a Kalman filter throughout the video stream. This leads to good accuracy track-

ing and limited drift. The tracker works on both intensity and depth video streams. It is also capable of working without depth information by approximating the head using an ellipsoid. I introduce three extensions to GAVAM in order to combine rigid and non-rigid tracking, hence improving pose estimation accuracy both in the 2D and 3D cases.

Firstly, I replace the simple ellipsoid model used in 2D tracking with a person specific triangular mesh. The mesh is constructed from the first frame of the tracking sequence using the 3D PDM of the fitted CLM. Since different projection is assumed by CLM (weak-perspective) and GAVAM (full perspective), to convert from the CLM landmark positions to GAVAM reference frame:

$$Z_g = \frac{1}{s} + Z_p, X_g = Z_g \frac{x_i - c_x}{f}, Y_g = Z_g \frac{y_i - c_y}{f}, \tag{5.3}$$

where f is the camera focal length, c_x, c_y the camera central points, s is the PDM scaling factor (inverse average depth for the weak perspective model), Z_p the Z component of a feature point in PDM reference frame x_i, y_i the feature points in image plane, and X_g, Y_g, Z_g the vertex locations in the GAVAM frame of reference.

Secondly, I use the CLM tracker to provide a better estimate of initial head pose than is provided by the static head pose detector used in GAVAM. Furthermore, the initial estimate of head distance from the camera used in GAVAM (assuming that the head is 20 cm wide), is replaced with a more stable assumption of interpupillary distance of 62 mm (Dodgson, 2004), based on the tracked eye corners using the CLM-Z or CLM trackers.

Lastly, an additional hypothesis – the current head pose estimate from CLM-Z (CLM in 2D case), is provided to aid the GAVAM tracker with the selection of keyframes to be used for differential tracking.

5.7 Experiments

I performed a number of experiments to analyse and validate the CLM-Z approach. Firstly, the necessity for a new normalisation function was

tested (Section 5.7.2). Secondly, the approaches for fusing the depth and visible light based patch expert responses were investigated (Section 5.7.3). Finally, the CLM-Z approach was compared to CLM for the task of facial landmark detection (Section 5.7.4), landmark tracking (Section 5.7.5) and head pose estimation (Section 5.7.6).

5.7.1 Methodology

The experimental methodology used in this section is identical to that used in Section 4.7, with the exception of the normalisation experiment (Section 5.7.2), results of which are from Baltrušaitis et al. (2012). The training data sampling and landmark detection procedures are not described here as they are similar to those used in Section 4.7.

For video based feature point and head pose tracking, a slight addition is made over the regular CLM approach. Depth is used, in addition to greyscale, to check that the fitting has converged (similar to validation in Section 4.6). Furthermore, if no depth signal is present in the converged area, tracking is assumed to have failed, leading to attempts at reinitialisation using a face detector (Section 4.6).

5.7.2 Normalisation

In order to see the effect the normalisation $\mathcal{P}_{\mathcal{Z}}$ has on the CLM-Z performance, I conducted experiments on landmark detection and tracking when using only depth information. Two sets of patch experts using $\mathcal{P}_{\mathcal{I}}$ and $\mathcal{P}_{\mathcal{Z}}$ normalisation techniques were trained and then used on the test sets.

As a test set, the depth maps from the BU-4DFE and Biwi feature point datasets were used (the initialisation was performed on the visible light images). For BU-4DFE the task was landmark detection, whereas for Biwi it was landmark tracking. The results on both of the datasets can be seen in Figure 5.4. These results demonstrate the need for robust normalisation for landmark detection and tracking on depth images.

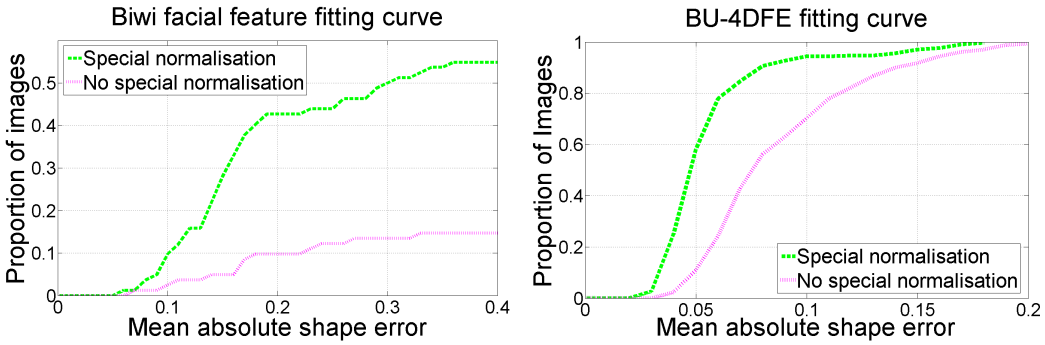


Figure 5.4: The fitting curve of CLM-Z comparing how the use of the specialised depth normalisation affects the landmark tracking accuracy. Note the much higher fitting accuracy on depth images using the normalisation scheme $\mathcal{P}_{\mathcal{Z}}$, as opposed to zero mean unit variance one. This is especially evident on Biwi dataset which has a much noisier signal due to Kinect sensor.

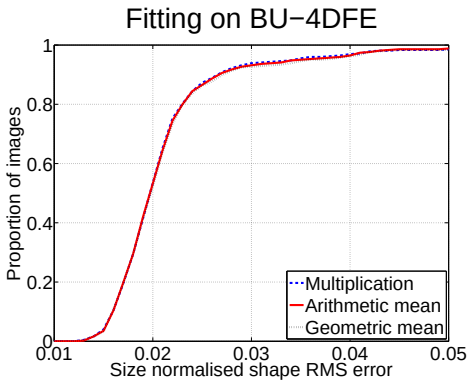


Figure 5.5: Using different methods to combine depth and intensity based information. Observe how there is no difference between the ways of combining the response maps.

5.7.3 Patch response combination

An experiment was conducted to assess which of the patch combination methods works best for combining visible light based response maps with the response maps from depth patch experts for the task of landmark detection. The same techniques that were used to combine greyscale and gradient intensity patch responses were explored (see Sec-

tion 4.7.2). These are: multiplication, geometric mean, and arithmetic mean.

The experiment was conducted on the BU-4DFE dataset (using both the greyscale and depth images). The results of this experiment can be seen in Figure 5.5. Friedman's ANOVA revealed no significant differences between the methods for the task of landmark detection on this dataset ($p > 0.05$). This is an interesting result, because for the fusion of greyscale and gradient intensity patch responses, the best approach to use was multiplication of the response maps. However, there seems to be little, if any, difference in the way that the patch responses are fused for CLM-Z (no statistically significant differences). In further experiments, I used the arithmetic mean because it is quicker to calculate.

5.7.4 Landmark detection in images

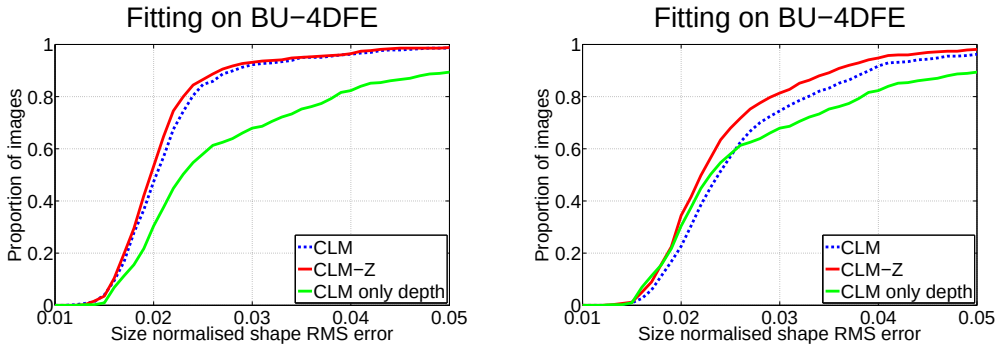
In order to assess the effect of adding depth information for landmark detection in images, I evaluated the CLM-Z approach on the BU-4DFE dataset.

Landmark detection accuracy was assessed in the following three conditions: intensity and gradient only (CLM); depth only, intensity and gradient with depth (CLM-Z). In addition, two types of greyscale and gradient intensity patch experts were considered: trained on only frontally lit images (less general but more accurate), and on multiple lighting conditions (more general but less accurate). Description of such training data and its implications are discussed in Section 4.8.1.

Results

The results of the CLM-Z experiment for facial landmark detection can be seen in Figure 5.6.

A Friedman's ANOVA was conducted to compare the effect of channels being used (depth, visible light, depth+visible light) on the RMS errors on the BU-4DFE dataset when using frontal illumination for training. There was a significant effect of the channel, $\chi^2(2) = 136.5, p < 0.001$.



(a) Using frontal light patch experts (b) Using general illumination patch experts

Figure 5.6: Comparing landmark detection accuracy on BU-4DFE database when using CLM, CLM-Z and CLM using only depth data. Observe how addition of depth data improves detection accuracy, especially when more general/robust intensity patch experts are used. Furthermore, only using depth data still leads to good convergence rate with 90% of images converged (RMSE$>$ 0.05).

Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the channels: depth (Mdn $=$ 0.0230), visible light (Mdn $=$ 0.0204) and depth with visible-light (Mdn $=$ 0.0198) were significantly different from each other at $p < 0.001$.

A Friedman's ANOVA was conducted to compare the effect of channels being used (depth, visible light, depth+visible light) on the RMS errors on the BU-4DFE dataset when using general illumination for training. There was a significant effect of the channel, $\chi^2(2) = 68.8, p < 0.001$. Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the channels: depth (Mdn $=$ 0.0230), visible light (Mdn $=$ 0.0243) and depth with visible-light (Mdn $=$ 0.0220) were significantly different from each other at $p < 0.01$.

Figure 5.7: Examples of facial expression tracking on Biwi dataset. Top row CLM, bottom row CLM-Z.

Discussion

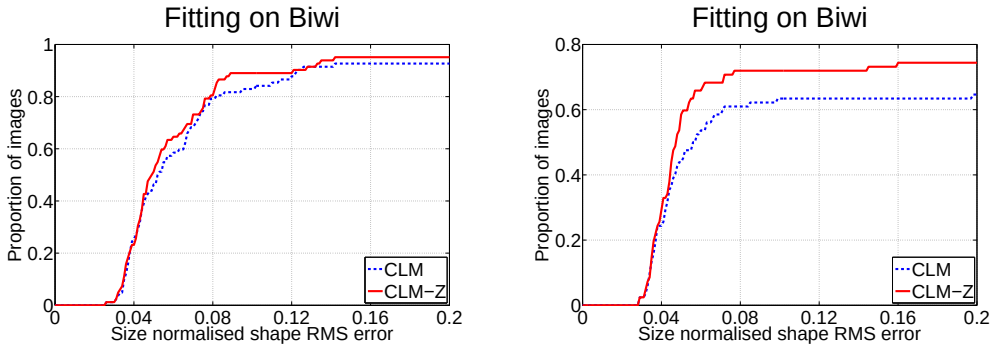
The experiment was designed to see the effect of using CLM-Z over the CLM approach with both frontal and general illumination patch experts. In both cases CLM-Z achieved statistically significantly lower error rates.

Secondly, note how performance degrades when using more general patch experts (Figure 5.6b) when compared to specific frontal patch experts (Figure 5.6a). However, the degradation was smaller when the depth signal was available. This illustrates the added benefit of CLM-Z when the visible light signal is not as reliable or unpredictable (needing multi-light training).

Finally, the depth modality on its own was still able to track the feature points reasonably well (with 90% of images converging at a 0.05 threshold). This demonstrates the usefulness of depth when there is no intensity information available. Furthermore, using only the depth signal was more accurate in the general illumination patch expert case, demonstrating its effectiveness.

5.7.5 Evaluation on image sequences

The effect of CLM-Z on tracking feature points in a video sequence was also assessed. For this I used a subset of the Biwi head pose dataset that is labelled for feature points (see Section 3.2.2).



(a) With reinitialisation (b) Without reinitialisation

Figure 5.8: Using CLM-Z on four Biwi video sequences. Notice how the benefit of CLM-Z becomes more apparent when no reinitialisation is performed, suggesting it is a more robust approach.

The training and fitting strategies used were the same as for the previous experiments on video tracking (Section 4.7.5). The depth and visible light patches used were the same as in the above sections. For feature tracking in a sequence the model parameters from the previous frame were used as starting parameters for tracking the next frame. Two experiments were performed, one with a reinitialisation scheme after validation (Section 4.6), and one without, in order to test both accuracy and robustness (a window size of 11×11 was used in all iterations).

Results

The results of the first experiment with reinitialisation can be seen in Figure 5.8a. A Wilcoxon signed rank test on effect of the model on RMS errors revealed no significant difference in error rates ($Z = -0.98, p > 0.1$) between CLM-Z (Mdn $= 0.049$) and CLM (Mdn $= 0.052$).

The results of the second experiment without reinitialisation can be seen in Figure 5.8b. A Wilcoxon signed rank test on effect of the model on RMS errors revealed a significant difference in error rates ($Z = -2.88, p < 0.01$) between CLM-Z (Mdn $= 0.047$) and CLM (Mdn $= 0.056$).

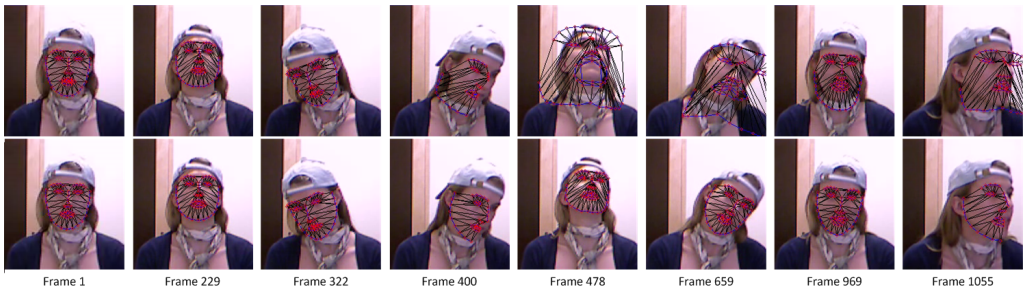


Figure 5.9: Examples of facial expression tracking on the ICT-3DHP dataset. Top row CLM, bottom row the CLM-Z approach.

Discussion

The CLM-Z approach managed to generalise well on a dataset not used for training, and improved the performance of a regular CLM when no reinitialisation was used. This was despite the training and testing datasets being quite different: high resolution range scanner data for training, low resolution noisy Kinect data for testing.

During the experiment without any reinitialisation CLM-Z demonstrated the ability to keep tracking and recover from failure, whereas CLM accuracy decreased. This demonstrates the benefit of CLM-Z over CLM for the task of facial feature tracking in videos.

5.7.6 Head pose tracking using depth data

To measure the performance of CLM-Z as a head pose tracker and the combination of the CLM-Z with a head pose tracker I evaluated them on two publicly available datasets: Biwi head pose dataset and ICT-3DHP. Both datasets contain labelled ground truth head pose data and aligned RGBD data.

ICT-3DHP dataset has 10 video sequences with head motion and very few missing frames, the lighting is also reasonably static. On the other hand, Biwi head pose dataset was collected with a frame based algorithm in mind so it has numerous occasions of lost frames and occasional

mismatch between colour and depth frames. This makes the dataset especially difficult for tracking based algorithms.

Baseline methods

The first baseline used is the CLM approach described in Chapter 4. This allowed me to evaluate the effect of depth data for the task of head pose estimation.

The second baseline used is that of Random Regression Forests (Fanelli et al., 2011b) (using the implementation provided by the authors), which is a detection based method and provides head pose estimates on a per frame level.

Another baseline used for the Biwi head pose dataset is that from Marras et al. (2013). Their method fuses greyscale and depth data using angle based features. It combines image gradient orientations as extracted from intensity images with the directions of surface normals computed from depth images. Lastly, it is a tracking based approach.

The final baseline was the GAVAM (Morency et al., 2008) tracker which can use visible light alone, or can include depth information for improved performance.

Results

The results of the head pose tracking experiment are shown in Table 5.1. They include the results from: baseline methods, CLM-Z, and a combined rigid and non-rigid tracker.

Discussion

Firstly, it is clear that tracking based approaches performed much better on the ICT-3DHP dataset. The different tracker results on Biwi and ICT-3DHP datasets are because the former was not collected with a tracking approach in mind and has a number of frames missing. Nevertheless, CLM tracking approaches managed to show good performance.

Model	Yaw	Pitch	Roll	Mean	Mdn.
	ICT-3DHP				
GAVAM (Morency et al., 2008)	6.58	5.01	**3.50**	5.03	3.08
CLM	5.41	4.32	4.83	4.85	2.45
CLM with GAVAM	6.02	5.52	3.84	5.13	2.95
	ICT-3DHP with depth				
Reg. for. (Fanelli et al., 2011b)	7.69	10.66	8.72	9.03	5.02
GAVAM (Morency et al., 2008)	**3.76**	5.24	4.93	4.64	2.91
CLM-Z	4.73	**4.10**	4.66	**4.50**	**2.35**
CLM-Z with GAVAM	4.41	5.22	5.36	5.00	2.81
	Biwi-HP				
GAVAM (Morency et al., 2008)	14.16	9.17	12.41	11.91	5.63
CLM	10.32	10.27	9.01	9.87	3.58
CLM with GAVAM	13.19	9.46	10.81	11.15	4.94
	Biwi-HP with depth				
Marras et al. (2013)	9.2	9.0	**8.0**	8.7	NA
Reg. for. (Fanelli et al., 2011b)	9.2	8.5	**8.0**	8.6	NA
GAVAM (Morency et al., 2008)	6.75	**5.53**	10.66	7.65	3.92
CLM-Z	10.52	7.98	8.14	8.88	**3.20**
CLM-Z with GAVAM	**6.74**	6.07	9.64	**7.48**	4.02

Table 5.1: Estimating the head pose using CLM-Z and CLM-Z with GAVAM approaches alongside some other state-of-the-art methods. It can be seen that CLM-Z outperformed CLM in terms of mean and median errors.

CLM-Z outperformed the regular CLM approach on both of the datasets, demonstrating the benefit of the depth signal. Furthermore, the head pose estimation accuracy of CLM-Z is comparable to that of dedicated head pose trackers, indicating the usefulness of CLM-Z as a head pose tracker.

The combination of CLM-Z together with a head pose tracker had little effect on overall performance (with slightly better performance in 2D case and slightly worse performance in 3D case when compared to GAVAM). This suggests that a better fusion technique is needed for the

Model	Yaw	Pitch	Roll	Mean	Mdn.
GAVAM (Morency et al., 2008)	3.79	4.45	2.15	3.47	2.12
CLM	3.68	4.26	2.50	3.48	2.26
CLM with GAVAM	**2.69**	**3.84**	**2.10**	**2.88**	**1.83**

Table 5.2: Head pose estimation results on the BU dataset, measured in mean absolute error.

rigid and non-rigid head pose estimation.

Finally, it is evident that CLM and CLM-Z approaches either perform equally well, or better, than dedicated head pose trackers (especially when looking at the median error metric). These approaches look even more promising when considering that CLM runs at 18–22 fps; CLM-Z at 11–15 fps; GAVAM at 8–10 fps; and CLM-Z with GAVAM at 4–7 fps on these dataset. The experiments were performed on a 3.06 GHz dual core Intel i3 CPU.

5.7.7 Head pose tracking on 2D data

One extension of CLM proposed in this chapter was its combination with a rigid head pose tracker such as GAVAM. In addition to evaluating it on the datasets which include depth it was also evaluated on a 2D dataset – Boston University head pose dataset.

The results of the combined approach can be seen in Table 5.2. The approach that combines both of the trackers outperforms the separate GAVAM and CLM methods in all of the orientation dimensions, which is not the case when depth is available. Combination of rigid and non-rigid trackers seems to benefit the 2D case much more. However, there is a performance cost: CLM runs at 40 fps, GAVAM at 20 fps, and CLM with GAVAM at 15 fps on this dataset. The experiments were performed on a 3.06 GHz dual core Intel i3 CPU.

5.8 Conclusion

In this chapter I presented CLM-Z, a Constrained Local Model approach that fully integrates depth information alongside intensity for facial feature point tracking and detection. This approach was evaluated on publicly available datasets and shows better performance both in terms of convergence and accuracy for feature point tracking from a single image and in a video sequence. The approach is especially helpful when the visible light signal is noisy or unreliable. CLM-Z is especially relevant due to recent availability of cheap consumer depth sensors, which can be used to improve existing computer vision techniques.

6 Constrained Local Neural Field

A big issue in CLM based landmark detection is the performance of patch experts, which are rarely more complex than linear SVRs or logistic regressors. Due to their simplicity, they may fail to learn complex non-linear relationships between pixel values and response maps. This is especially true if a more complex task of illumination invariant landmark detection is considered (see Section 4.8.1 and Figures 4.20b and 4.21). Because of its simplicity, it is difficult to expect a linear patch expert to work equally well in different illuminations. However, they are commonly used because they are simple to train and have fast implementations (using the convolution trick described in the Section 4.3.1) leading to real-time tracking speeds.

Patch experts do not need to be limited to simple linear SVRs or logistic regressors. However, ones based on more complex regressors (for example RBF kernel SVRs) can be very slow (under 1fps), making them unusable for application where large amounts of data needs to be processed. This is especially true for affect inference, as some datasets contain hour long recordings leading to hundreds of thousands of frames per single recording.

In this chapter, I present a Local Neural Field (LNF) patch expert which is an instance of the more general Continuous Conditional Neural Field (CCNF). It deals with the issues of learning complex scenes by using a hidden non-linear layer, and by exploiting spatial relationships between pixels. An additional advantage of the new LNF patch expert is that it can also be implemented by using simple convolutions for the most expensive part of regression, resulting in close to real-time tracking. Con-

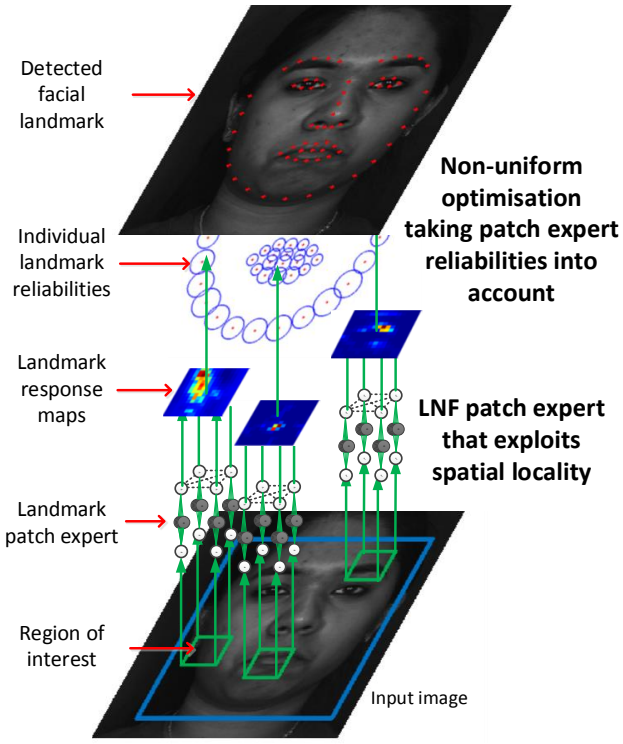
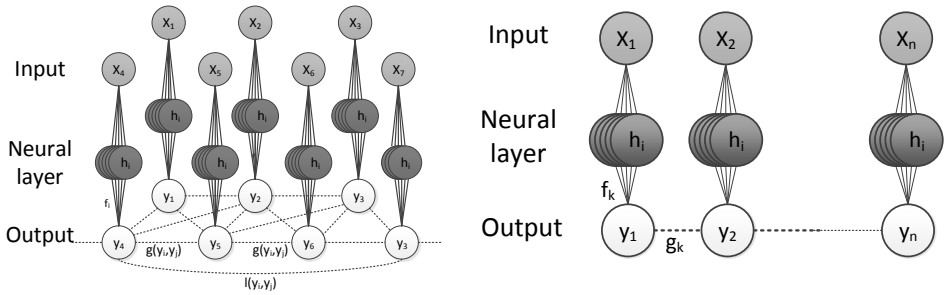


Figure 6.1: Overview of the CLNF model. LNF patch expert is used to calculate patch response maps, which leads to more reliable responses. Optimisation over the patch responses is performed using the Non-Uniform Regularised Mean-Shift method that takes the reliability of each patch expert into account leading to more accurate fitting. Only 3 out of 66 patch experts are displayed for clarity.

strained Local Neural Field (CLNF) is the name given to a CLM instance that uses LNF patch experts and, the previously introduced, NU-RLMS fitting method. An overview of CLNF can be seen in Figure 6.1.

I demonstrate the benefit of the LNF patch expert by comparing it to state-of-the-art methods on three tasks: detection of facial landmarks in images, tracking landmarks in videos, and head pose estimation in videos. By using just the visible light images CLNF achieves comparable, or better, performance than CLM-Z and outperforms the CLM in all of these tasks, while still achieving close to real time speeds ($\approx$ 20fps). Finally, I compare the CLNF approach with other state-of-the-art ap-



(a) Local Neural Field (b) Linear-chain CCNF

Figure 6.2: Examples of two instances of CCNF. Solid lines represent vertex features (f_k), dashed lines represent edge features (g_k or l_k). The input vector $\mathbf{x}_i$ is connected to the relevant output scalar y_i through the vertex features that combine the neural layer (Θ) and the vertex weights $\boldsymbol{\alpha}$. The outputs are further connected with edge features g_k (similarity) or l_k (sparsity). Only direct links from $\mathbf{x}_i$ to y_i are presented here, but extensions are straightforward.

proaches for facial landmark detection, demonstrating its benefits.

The experiments were conducted on four publicly available datasets. For facial landmark detection I used Multi-PIE and BU-4DFE datasets. For landmark tracking in videos I used the Biwi feature point dataset. Finally, for head pose estimation I used Boston University, Biwi and ICT-3DHP datasets. The test sets have extreme pose variation and difficult (Multi-PIE) and uncontrolled lighting (ICT-3DHP, and Biwi).

The discussion in this chapter is structured as follows. First, the CCNF model is introduced (Section 6.1). Then LNF patch expert, a special case of CCNF, is presented in Section 6.2. Finally, the experiments used to validate the new patch experts are presented.

6.1 Continuous Conditional Neural Field

Continuous Conditional Neural Field (CCNF) is an undirected graphical model which models the conditional probability of a continuous valued vector $\mathbf{y}$ depending on continuous $\mathbf{x}$. CCNF is a general graphical model which can be used as a patch expert, but also for time series modelling

(see Chapter 8). CCNF combines the non-linearity of Conditional Neural Fields (Peng et al., 2009) with the flexibility and continuous output of Continuous Conditional Random Fields (Qin et al., 2008). The model also bears close resemblance to the Discriminative Random Fields (DRF) model proposed by Kumar and Hebert (2003). DRF, however, is a classification rather that regression model and it uses slightly different vertex (association) and edge (interaction) features (potentials) that are more suited for classification.

In the discussion the following notation is used: $\boldsymbol{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is a set of observed input variables, $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ is a set of output variables to be predicted, and n is the length of a sequence. The output variable y_i is a scalar ($y_i \in \mathcal{R}$) and input $\mathbf{x}_i$ is an m dimensional feature vector ($\mathbf{x}_i \in \mathcal{R}^m$).

CCNF model for a particular set of observations is a conditional probability distribution with the probability density function:

$$P(\mathbf{y}|\boldsymbol{x}) = \frac{\exp(\Psi)}{\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}}, \tag{6.1}$$

where Ψ is a set of potential functions which control the model and $\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}$ is the normalisation (partition) function which makes the probability distribution a valid one (by making it sum to 1). The following section describes the potential function than can be used in the CCNF model.

Figure 6.2 demonstrates two CCNF instances: LNF and linear-chain CCNF. The former can be used as a patch expert and the latter for time-series modelling. CCNF is flexible enough to be used for such different tasks.

6.1.1 Potential functions

Three types of potential functions are defined for the model: vertex features (f_k) and edge features (g_k, and l_k). CCNF potential function is

defined as:

$$\Psi = \sum_{i} \sum_{k=1}^{K1} \alpha_k f_k(y_i, \boldsymbol{x}, \boldsymbol{\theta}_k) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j) + \sum_{i,j} \sum_{k=1}^{K3} \gamma_k l_k(y_i, y_j), \quad (6.2)$$

where model parameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots \alpha_{K1}\}$, $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots \boldsymbol{\theta}_{K1}\}$, and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots \beta_{K2}\}$, $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \dots \gamma_{K3}\}$ are learned and used for inference during testing. The individual potential functions are defined as follows:

$$f_k(y_i, \boldsymbol{x}, \boldsymbol{\theta}_k) = -(y_i - h(\boldsymbol{\theta}_k, \mathbf{x}_i))^2, \quad (6.3)$$

$$h(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}, \quad (6.4)$$

$$g_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(g_k)} (y_i - y_j)^2, \quad (6.5)$$

$$l_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(l_k)} (y_i + y_j)^2. \quad (6.6)$$

Vertex features f_k represent the mapping from the $\mathbf{x}_i$ to y_i through a single layer neural network, where $\boldsymbol{\theta}_k$ is the weight vector for a particular neuron k. The corresponding α_k for vertex feature f_k represents the reliability of the k^{th} neuron. The number of neurons used will depend on the problem and can be determined during cross-validation.

Edge features g_k represent the similarities between observations y_i and y_j, enforcing smoothness between connected nodes. Neighbourhood measure $S^{(g_k)}$ is used to create (and potentially weigh) the similarity connections between nodes.

Edge features l_k represent the sparsity (or inhibition) constraint between connected observations y_i and y_j. They penalise the model if both y_i and y_j are large, but do not if both of them are zero. However, l_k edge features have an unwanted consequence of slightly penalising the model if one of y_i or y_j is large. This, of course, only works for positive y values, which is the case for response maps. Neighbourhood measure $S^{(l_k)}$ is used to create (and potentially weigh) the sparsity connections between nodes.

6.1.2 Learning and Inference

In this section I describe how to estimate the parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}\}$ given training data $\{\boldsymbol{x}^{(q)}, \mathbf{y}^{(q)}\}_{q=1}^{M}$ of M observations (sequences, areas of interest, etc.) where each $\boldsymbol{x}^{(q)} = \{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a set of inputs (pixel values under the response, features describing facial appearance), and each $\mathbf{y}^{(q)} = \{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a corresponding set of real valued labels (expected response from a patch expert, point in continuous emotional space).

CCNF learning picks the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\Theta}$ values which maximise the conditional log-likelihood of the model on the training observations:

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}) = \sum_{q=1}^{M} \log P(\mathbf{y}^{(q)} | \boldsymbol{x}^{(q)}), \tag{6.7}$$

$$(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \bar{\boldsymbol{\gamma}}, \bar{\boldsymbol{\Theta}}) = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta}} (L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Theta})). \tag{6.8}$$

During inference a value of $\mathbf{y}$ that maximises the probability distribution given an observation $\boldsymbol{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is found:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} (P(\mathbf{y} | \boldsymbol{x})). \tag{6.9}$$

This can be computed using the learned parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\Theta}$.

Multi-variate Gaussian form

It helps with the derivation of inference and the partial derivatives used for training, if the probability density function (Equation 6.1) is converted into multivariate Gaussian form:

$$P(\mathbf{y} | \boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu})), \tag{6.10}$$

$$\Sigma^{-1} = 2(A + B + C). \tag{6.11}$$

The diagonal matrix A represents the contribution of $\boldsymbol{\alpha}$ terms (vertex features) to the covariance matrix, and the symmetric B and C represent

the contribution of the β, γ terms (edge features):

$$A_{i,j} = \begin{cases} \sum_{k=1}^{K1} \alpha_k, & i = j \\ 0, & i \neq j \end{cases}, \tag{6.12}$$

$$B_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \beta_k \sum_{r=1}^{n} S_{i,r}^{(g_k)}) - (\sum_{k=1}^{K2} \beta_k S_{i,j}^{(g_k)}), & i = j \\ -\sum_{k=1}^{K2} \beta_k S_{i,j}^{(g_k)}, & i \neq j \end{cases}, \tag{6.13}$$

$$C_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \gamma_k \sum_{r=1}^{n} S_{i,r}^{(l_k)}) + (\sum_{k=1}^{K2} \gamma_k S_{i,j}^{(l_k)}), & i = j \\ \sum_{k=1}^{K2} \gamma_k S_{i,j}^{(l_k)}, & i \neq j \end{cases}. \tag{6.14}$$

It is useful to define a vector $\boldsymbol{d}$, which describes the linear terms in the distribution, and $\boldsymbol{\mu}$ which is the mean value of the Gaussian form of the CCNF distribution:

$$\mathbf{d} = 2\alpha^T h(\mathbf{\Theta X}). \tag{6.15}$$

$$\boldsymbol{\mu} = \Sigma \boldsymbol{d}, \tag{6.16}$$

where $\mathbf{X}$ is a matrix in which the i^{th} column represents $\mathbf{x}_i$, and $\mathbf{\Theta}$ represents the combined neural network weights, and $h(M)$ is an element-wise application of sigmoid (activation function) on each element of M. Thus, $h(\mathbf{\Theta X})$ represents the response of each of the gates (neural layers) at each $\mathbf{x}_i$.

Intuitively $\boldsymbol{d}$ is the contribution from the the vertex features, which contribute directly from input features $\mathbf{x}$ towards $\mathbf{y}$. Σ on the other hand, controls the influence of the edge features to the output. Finally, $\boldsymbol{\mu}$ is the expected value of the distribution, hence it is the value of $\mathbf{y}$ that maximises $P(\mathbf{y}|\boldsymbol{x})$:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}}(P(\mathbf{y}|\boldsymbol{x})) = \boldsymbol{\mu} = \Sigma \boldsymbol{d}. \tag{6.17}$$

Having defined all the necessary variables, it is now possible to demonstrate the equivalence between probability density in Equation 6.1 and the multivariate Gaussian in Equation 6.10. First, combining the feature functions from Equations 6.3, 6.5, and 6.6 with the potential Equation 6.2, leads to:

$$\begin{aligned}\Psi &= \sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, \boldsymbol{x}, \boldsymbol{\theta}_k) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j, \boldsymbol{x}) + \sum_{i,j} \sum_{k=1}^{K3} \gamma_k l_k(y_i, y_j, \boldsymbol{x}) \\ &= -\sum_i \sum_{k=1}^{K1} \alpha_k (y_i - h(\boldsymbol{\theta}_k^T \boldsymbol{x}_i))^2 - \frac{1}{2} \sum_{i,j} \sum_{k=1}^{K2} \beta_k S_{i,j}^{g_k} (y_i - y_j)^2 \\ &\quad -\frac{1}{2} \sum_{i,j} \sum_{k=1}^{K3} \gamma_k S_{i,j}^{(l_k)} (y_i + y_j)^2.\end{aligned} \tag{6.18}$$

The factor Ψ can now be expressed in terms of A, B and $\mathbf{d}$ defined previously. This is done in parts, starting with terms containing $\boldsymbol{\alpha}$ parameters from Equation 6.18:

$$\begin{aligned}&-\sum_i \sum_{k=1}^{K1} \alpha_k (y_i - h(\boldsymbol{\theta}_k^T \mathbf{x}_i))^2 = -\sum_i \sum_{k=1}^{K1} \alpha_k (y_i^2 - 2 y_i h(\boldsymbol{\theta}_k^T \mathbf{x}_i) + h(\boldsymbol{\theta}_k^T \mathbf{x}_i)^2) \\ &= -\sum_i \sum_{k=1}^{K1} \alpha_k y_i^2 + \sum_i \sum_{k=1}^{K1} \alpha_k 2 y_i h(\boldsymbol{\theta}_k^T \mathbf{x}_i) - \sum_i \sum_{k=1}^{K1} \alpha_k h(\boldsymbol{\theta}_k^T \mathbf{x}_i)^2 \\ &= -\mathbf{y}^T A \mathbf{y} + \mathbf{y}^T \boldsymbol{d} - \sum_i \sum_{k=1}^{K1} \alpha_k h(\boldsymbol{\theta}_k^T \mathbf{x}_i)^2.\end{aligned} \tag{6.19}$$

Collecting terms with β and γ parameters in Equation 6.18 leads to:

$$
\begin{aligned}
&-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K2}\beta_k S_{i,j}^{(g_k)}(y_i-y_j)^2-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K3}\gamma_k S_{i,j}^{(l_k)}(y_i+y_j)^2\\
&=-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K2}\beta_k S_{i,j}^{(g_k)}(y_i^2-2y_iy_j+y_j^2)-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K3}\gamma_k S_{i,j}^{(l_k)}(y_i^2+2y_iy_j+y_j^2)\\
&=-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K2}\beta_k S_{i,j}^{(g_k)}(y_i^2+y_j^2)+\sum_{i,j}\sum_{k=1}^{K2}\beta_k S_{i,j}^{(g_k)}y_iy_j+\\
&\quad-\frac{1}{2}\sum_{i,j}\sum_{k=1}^{K2}\gamma_k S_{i,j}^{(l_k)}(y_i^2+y_j^2)-\sum_{i,j}\sum_{k=1}^{K2}\gamma_k S_{i,j}^{(l_k)}y_iy_j\\
&=-\sum_{k=1}^{K2}\beta_k\sum_{i,j}S_{i,j}^{(g_k)}y_i^2+\sum_{k=1}^{K2}\beta_k S_{i,j}^{(g_k)}\sum_{i,j}y_iy_j+\\
&\quad-\sum_{k=1}^{K2}\gamma_k\sum_{i,j}S_{i,j}^{(l_k)}y_i^2-\sum_{k=1}^{K2}\gamma_k S_{i,j}^{(l_k)}\sum_{i,j}y_iy_j\\
&=-\mathbf{y}^TB\mathbf{y}-\mathbf{y}^TC\mathbf{y}.
\end{aligned}
\tag{6.20}
$$

Recall that every $S^{(k)}$ is a symmetric matrix by construction.

Combining Equations 6.18, 6.19, and 6.20, and having $\mathbf{d}=\Sigma^{-1}\boldsymbol{\mu}$ from the definition of $\boldsymbol{\mu}$ in Equation 6.16 results in:

$$
\Psi=-\mathbf{y}^TA\mathbf{y}+\mathbf{y}^T\boldsymbol{d}-\mathbf{y}^TB\mathbf{y}-\mathbf{y}^TC\mathbf{y}-e=-\frac{1}{2}(\mathbf{y}^T\Sigma^{-1}\mathbf{y})+\mathbf{y}\Sigma^{-1}\boldsymbol{\mu}-e.
\tag{6.21}
$$

Above, $e=\sum_i\sum_{k=1}^{K1}\alpha_k h(\boldsymbol{\theta}_k^T\boldsymbol{x}_i)^2$, it cancels out eventually, so is not written out fully.

Replacing Ψ in Equation 6.1 with Ψ defined in Equation 6.21 leads to:

$$
\begin{aligned}
P(\mathbf{y}|\boldsymbol{x}) &= \frac{\exp(\Psi)}{\int_{-\infty}^{\infty}\exp(\Psi)d\mathbf{y}}=\\
&= \frac{\exp(-\frac{1}{2}(\mathbf{y}^T\Sigma^{-1}\mathbf{y})+\mathbf{y}\Sigma^{-1}\boldsymbol{\mu})\exp(-e)}{\int_{-\infty}^{\infty}\{\exp(-\frac{1}{2}(\mathbf{y}^T\Sigma^{-1}\mathbf{y})+\mathbf{y}\Sigma^{-1}\boldsymbol{\mu})\exp(-e)\}d\mathbf{y}}\\
&= \frac{\exp(-\frac{1}{2}(\boldsymbol{y}^T\Sigma^{-1}\boldsymbol{y})+\boldsymbol{y}\Sigma^{-1}\boldsymbol{\mu})}{\int_{-\infty}^{\infty}\{\exp(-\frac{1}{2}(\mathbf{y}^T\Sigma^{-1}\mathbf{y})+\mathbf{y}\Sigma^{-1}\boldsymbol{\mu})\}d\mathbf{y}}.
\end{aligned}
\tag{6.22}
$$

As e does not depend on $\mathbf{y}$, it can be taken out of the integral, leading to it cancelling out.

The partition function can be integrated using the integral of an exponential with square and linear terms[1]:

$$\int_{-\infty}^{\infty} \{\exp(-\frac{1}{2}(\mathbf{y}^T\Sigma^{-1}\mathbf{y}) + \mathbf{y}\Sigma^{-1}\boldsymbol{\mu})\}d\mathbf{y} = \frac{(2\pi)^{\frac{n}{2}}}{|\Sigma^{-1}|^{\frac{1}{2}}}\exp(\frac{1}{2}\boldsymbol{\mu}\Sigma^{-1}\boldsymbol{\mu}). \quad (6.23)$$

In order to guarantee that the partition function is integrable the following constraints have to be met: $\alpha_k > 0$ and $\beta_k > 0, \gamma_k > 0$ (Qin et al., 2008). This is because if the constraints are not held Σ^{-1} is not guaranteed to be positive semi-definite, and this is required for the integral.

Finally, plugging Equations 6.21 and 6.23 into Equation 6.1 leads to:

$$\begin{aligned} P(\mathbf{y}|\boldsymbol{x}) &= \frac{\exp(-\frac{1}{2}\mathbf{y}^T\Sigma^{-1}\mathbf{y} + \mathbf{y}\Sigma^{-1}\boldsymbol{\mu})}{\frac{(2\pi)^{\frac{n}{2}}}{|\Sigma^{-1}|^{\frac{1}{2}}}\exp(\frac{1}{2}\boldsymbol{\mu}\Sigma^{-1}\boldsymbol{\mu})} \\ &= \frac{\exp(-\frac{1}{2}\mathbf{y}^T\Sigma^{-1}\mathbf{y} + \mathbf{y}\Sigma^{-1}\boldsymbol{\mu})\exp(-\frac{1}{2}\boldsymbol{\mu}\Sigma^{-1}\boldsymbol{\mu})}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \\ &= \frac{\exp(-\frac{1}{2}\mathbf{y}^T\Sigma^{-1}\mathbf{y} + \mathbf{y}\Sigma^{-1}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\mu}\Sigma^{-1}\boldsymbol{\mu})}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \\ &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})). \end{aligned} \quad (6.24)$$

This demonstrates that the CCNF probability density function can be expressed as a multivariate Gaussian.

Partial derivatives

Having defined the probability distribution, it is now possible to find the partial derivatives of it with respect to the model parameters. These partial derivatives can then be used in a gradient based optimisation method to help in finding locally optimal model parameters faster and more accurately.

Firstly, it is more convenient to express the problem in terms of log-likelihood, as this does not affect the minima or maxima of the objective. Applying a logarithm on the CCNF model from Equation 6.24 leads to:

[1] http://www.weylmann.com/gaussian.pdf

$$\begin{aligned}\log(P(\mathbf{y}|\boldsymbol{x})) &= -\tfrac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) - \log((2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}) \\ &= -\tfrac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) - (\tfrac{n}{2}\log(2\pi) + \tfrac{1}{2}\log|\Sigma|) \\ &= -\tfrac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) + \tfrac{1}{2}\log|\Sigma^{-1}| - \tfrac{n}{2}\log(2\pi) \\ &= -\tfrac{1}{2}\mathbf{y}^T\Sigma^{-1}\mathbf{y} + \mathbf{y}^T\boldsymbol{d} - \tfrac{1}{2}\boldsymbol{d}^T\Sigma\boldsymbol{d} + \tfrac{1}{2}\log|\Sigma^{-1}| - \tfrac{n}{2}\log(2\pi).\end{aligned} \tag{6.25}$$

Recall that $\boldsymbol{d} = \Sigma^{-1}\boldsymbol{\mu}$, and $|\Sigma| = \frac{1}{|\Sigma^{-1}|}$, where $|\Sigma|$ denotes the determinant of the covariance matrix Σ. Furthermore, because Σ^{-1} is symmetric by construction, $\Sigma^{-1} = (\Sigma^{-1})^T$ and $\Sigma = \Sigma^T$.

First, the derivation of partial derivatives with respect to the $\boldsymbol{\alpha}$ parameters is demonstrated. Recall that A is only dependent on $\boldsymbol{\alpha}$, B on $\boldsymbol{\beta}$, and C on $\boldsymbol{\gamma}$; $\boldsymbol{d}$, however, depends on both $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$, hence:

$$\frac{\partial\Sigma^{-1}}{\partial\alpha_k} = \frac{\partial 2A + 2B + 2C}{\partial\alpha_k} = \frac{\partial 2A}{\partial\alpha_k} = 2I, \tag{6.26}$$

$$\frac{\partial d_i}{\partial\alpha_k} = 2h(\Theta\mathbf{X})_{k,i}, \tag{6.27}$$

$$\frac{\partial\boldsymbol{d}}{\partial\alpha_k} = (2h(\Theta\mathbf{X})_{k,*})^T. \tag{6.28}$$

I is the identity matrix of size $n \times n$, where n is the number of elements in a sequence. $X_{k,*}$ notation refers to a row vector corresponding to the k^{th} row of a matrix X. For brevity, $D = h(\Theta\mathbf{X})$

In the derivation below the partial derivative of a matrix inverse $\frac{\partial M^{-1}}{\partial\alpha} =$

$-M^{-1}\frac{\partial M}{\partial \alpha}M^{-1}$ is used, to get the partial derivative of Σ.

$$\begin{aligned}\frac{\partial \boldsymbol{d}^T\Sigma\boldsymbol{d}}{\partial \alpha_k} &= \frac{\partial \boldsymbol{d}^T}{\partial \alpha_k}\Sigma\boldsymbol{d} + \boldsymbol{d}^T\frac{\partial \Sigma\boldsymbol{d}}{\partial \alpha_k} = 2D_{k,*}\boldsymbol{\mu} + \boldsymbol{d}^T(\frac{\partial \Sigma}{\partial \alpha_k}\boldsymbol{d} + \Sigma\frac{\partial \boldsymbol{d}}{\partial \alpha_k}) \\ &= 2D_{k,*}\boldsymbol{\mu} + \boldsymbol{d}^T\frac{\partial \Sigma}{\partial \alpha_k}\boldsymbol{d} + \boldsymbol{d}^T\Sigma 2(D_{k,*})^T = 4D_{k,*}\boldsymbol{\mu} + \boldsymbol{d}^T\frac{\partial \Sigma}{\partial \alpha_k}\boldsymbol{d} \\ &= 4D_{k,*}\boldsymbol{\mu} + \boldsymbol{d}^T(-\Sigma\frac{\partial \Sigma^{-1}}{\partial \alpha_k}\Sigma)\boldsymbol{d} = 4D_{k,*}\boldsymbol{\mu} - 2\boldsymbol{d}^T\Sigma\Sigma\boldsymbol{d} \\ &= 4D_{k,*}\boldsymbol{\mu} - 2\boldsymbol{\mu}^T\boldsymbol{\mu} \end{aligned} \tag{6.29}$$

$$\begin{aligned}\frac{\partial \log|\Sigma^{-1}|}{\partial \alpha_k} &= \frac{1}{|\Sigma^{-1}|}\frac{\partial|\Sigma^{-1}|}{\partial \alpha_k} = \frac{1}{|\Sigma^{-1}|}|\Sigma^{-1}| \times \text{trace}(\Sigma\frac{\partial \Sigma^{-1}}{\alpha_k}) \\ &= 2 \times \text{trace}(\Sigma I) = 2 \times \text{trace}(\Sigma)\end{aligned} \tag{6.30}$$

These can be combined to get the partial derivative of log-likelihood with respect to $\boldsymbol{\alpha}$ terms:

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\alpha_k} = -\mathbf{y}^T\mathbf{y} + 2\mathbf{y}^T D_{k,*}^T - 2D_{*,k}\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{\mu} + \text{trace}(\Sigma) \tag{6.31}$$

Now the derivation of the partial derivatives of the likelihood with respect to β and γ parameters is shown (they are discussed together as they are very similar):

$$\frac{\partial \Sigma^{-1}}{\partial \beta_k} = 2B^{(k)}, \tag{6.32}$$

$$\frac{\partial \Sigma^{-1}}{\partial \gamma_k} = 2C^{(k)}, \tag{6.33}$$

$$B^{(k)} = \begin{cases} (\sum_{r=1}^{n} S_{i,r}^{(g_k)}) - S_{i,j}^{(g_k)}, & i = j \\ -S_{i,j}^{(g_k)}, & i \neq j \end{cases}, \tag{6.34}$$

$$C^{(k)} = \begin{cases} (\sum_{r=1}^{n} S_{i,r}^{(l_k)}) + S_{i,j}^{(l_k)}, & i = j \\ S_{i,j}^{(l_k)}, & i \neq j \end{cases}, \tag{6.35}$$

$$\frac{\partial \boldsymbol{d}}{\partial \beta_k} = 0, \tag{6.36}$$

$$\frac{\partial \boldsymbol{d}}{\partial \gamma_k} = 0, \tag{6.37}$$

$$\frac{\boldsymbol{d}^T \Sigma \boldsymbol{d}}{\beta_k} = -\boldsymbol{d}^T (\Sigma \frac{\partial \Sigma^{-1}}{\partial \beta_k} \Sigma) \boldsymbol{d} = -2\boldsymbol{d}^T \Sigma B^{(k)} \Sigma \boldsymbol{d} = -2\boldsymbol{\mu}^T B^{(k)} \boldsymbol{\mu}, \tag{6.38}$$

$$\frac{\boldsymbol{d}^T \Sigma \boldsymbol{d}}{\gamma_k} = -\boldsymbol{d}^T (\Sigma \frac{\partial \Sigma^{-1}}{\partial \gamma_k} \Sigma) \boldsymbol{d} = -2\boldsymbol{d}^T \Sigma C^{(k)} \Sigma \boldsymbol{d} = -2\boldsymbol{\mu}^T C^{(k)} \boldsymbol{\mu}, \tag{6.39}$$

$$\begin{aligned} \frac{\partial \log |\Sigma^{-1}|}{\partial \beta_k} &= \frac{1}{|\Sigma^{-1}|} \frac{\partial |\Sigma^{-1}|}{\partial \beta_k} = \frac{1}{|\Sigma^{-1}|} |\Sigma^{-1}| \times \text{trace}(\Sigma \frac{\partial \Sigma^{-1}}{\beta_k}) \\ &= 2 \times \text{trace}(\Sigma B^{(k)}) = 2 \times \text{Vec}(\Sigma)^T \text{Vec}(B^{(k)}), \end{aligned} \tag{6.40}$$

$$\begin{aligned} \frac{\partial \log |\Sigma^{-1}|}{\partial \gamma_k} &= \frac{1}{|\Sigma^{-1}|} \frac{\partial |\Sigma^{-1}|}{\partial \gamma_k} = \frac{1}{|\Sigma^{-1}|} |\Sigma^{-1}| \times \text{trace}(\Sigma \frac{\partial \Sigma^{-1}}{\gamma_k}) \\ &= 2 \times \text{trace}(\Sigma C^{(k)}) = 2 \times \text{Vec}(\Sigma)^T \text{Vec}(C^{(k)}). \end{aligned} \tag{6.41}$$

Here the matrix trace property - $\text{trace}(AB) = \text{Vec}(A)^T \text{Vec}(B)$ was used, where Vec refers to the matrix vectorisation operation which stacks up columns of a matrix together to form a single column matrix[2]. The derivative of inverse matrix as in the case with α_k version, was also used.

This can now be combined to lead to:

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\beta_k} = -\mathbf{y}^T B^{(k)} \mathbf{y} + \boldsymbol{\mu}^T B^{(k)} \boldsymbol{\mu} + \text{Vec}(\Sigma)^T \text{Vec}(B^{(k)}), \tag{6.42}$$

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\gamma_k} = -\mathbf{y}^T C^{(k)} \mathbf{y} + \boldsymbol{\mu}^T C^{(k)} \boldsymbol{\mu} + \text{Vec}(\Sigma)^T \text{Vec}(C^{(k)}). \tag{6.43}$$

Finally, the partial derivatives of the likelihood with respect to the $\boldsymbol{\theta}$ parameters (the neural network weights) are derived. The notation is abused slightly for clarity and brevity, $h(A)$ on a $n \times m$ size matrix A produces a $n \times m$ matrix with the activation function applied on each element.

$$\frac{\partial \Sigma^{-1}}{\partial \theta_{i,j}} = 0, \tag{6.44}$$

[2]This leads to much faster calculation of the trace as a multiplication of potentially very big matrices is avoided

If the sigmoid activation function $h(z) = \frac{1}{1+e^{-z}}$ is used:

$$\frac{\partial h(z)}{\partial z} = h(z)(1 - h(z)), \tag{6.45}$$

$$b_r = 2 \sum_{k=1}^{K1} \alpha_k h(\theta_k^T \mathbf{x}_r), \tag{6.46}$$

$$\frac{\partial b_r}{\partial \theta_{i,j}} = 2\alpha_i h(\boldsymbol{\theta}_i^T \mathbf{x}_r)(1 - h(\boldsymbol{\theta}_i^T \mathbf{x}_r))\mathbf{x}_{r,j}, \tag{6.47}$$

$$\frac{\partial \boldsymbol{d}}{\partial \theta_{i,j}} = 2\alpha_i \{h(\boldsymbol{\theta}_i^T X) \circ (1 - h(\boldsymbol{\theta}_i^T X))\}\mathbf{X}_{*,j}. \tag{6.48}$$

$$\begin{aligned} \frac{\partial \boldsymbol{d}^T \Sigma \boldsymbol{d}}{\partial \theta_{i,j}} &= \frac{\partial \boldsymbol{d}^T}{\partial \theta_{i,j}} \Sigma \boldsymbol{d} + \boldsymbol{d}^T \frac{\partial \Sigma \boldsymbol{d}}{\partial \theta_{i,j}} = \frac{\partial \boldsymbol{d}^T}{\partial \theta_{i,j}} \boldsymbol{\mu} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{d}}{\partial \theta_{i,j}} \\ &= 2\boldsymbol{\mu}^T \frac{\partial \boldsymbol{d}}{\partial \theta_{i,j}} \end{aligned} \tag{6.49}$$

Above, $\circ$ is the Hadamard or element-wise product.

These are now combined to get :

$$\begin{aligned} \frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\theta_{i,j}} &= \mathbf{y}^T \frac{\partial \boldsymbol{d}}{\partial \theta_{i,j}} - \boldsymbol{\mu}^T \frac{\partial \boldsymbol{d}}{\partial \theta_{i,j}} \\ &= (\mathbf{y} - \boldsymbol{\mu})^T (2\alpha_i \{h(\boldsymbol{\theta}_i^T X) \circ (1 - h(\boldsymbol{\theta}_i^T X))\} X_{*,j}) \end{aligned} \tag{6.50}$$

Above, is basically the update of a single layer neural network (back propagation) with sigmoid activation where the current feed-forward prediction is $\boldsymbol{\mu}$ and error is $(\mathbf{y} - \boldsymbol{\mu})$.

Learning

The partial derivatives in Equations 6.31, 6.42, 6.43, and 6.50 can now be used in the CCNF learning algorithm.

In order to avoid overfitting L_2 norm regularisation terms are added to the likelihood function for each of the parameter types ($\lambda_\alpha ||\boldsymbol{\alpha}||_2^2$,$\lambda_\beta ||\boldsymbol{\beta}||_2^2$, $\lambda_\beta ||\boldsymbol{\gamma}||_2^2$,$\lambda_\theta ||\boldsymbol{\Theta}||_2^2$), with alpha and beta sharing the regularisation weight. The values of $\lambda_\alpha, \lambda_\beta, \lambda_\Theta$ are determined during cross-validation, as is the number of neural layers.

I used the constrained BroydenFletcherGoldfarbShanno (BFGS) algorithm for finding locally optimal model parameters. I used the standard Matlab implementation of the algorithm.

Inference

Since the CCNF model can be viewed as a multivariate Gaussian, inferring $\mathbf{y}$ values that maximise $P(\mathbf{y}|\mathbf{x})$ is straightforward. The prediction is the mean value of the distribution:

$$\mathbf{y}' = \arg\max_{\mathbf{y}}(P(\mathbf{y}|\boldsymbol{x})) = \boldsymbol{\mu} = \Sigma \boldsymbol{d}. \tag{6.51}$$

6.2 Local Neural Field

In this section I present the novel LNF patch expert, also called the Grid-CCNF. Firstly, it learns complex non-linear relationships between the pixel values and the patch response maps. Secondly, it learns the relationships between nearby pixels in the response map. The two types of spatial relationships captured by the LNF model are: spatial similarity and sparsity. Spatial similarity ensures that pixels nearby should have similar alignment probabilities; sparsity reduces the number of peaks in the response map.

LNF is an instance of CCNF presented in the previous section. Input $\mathbf{x}$ is the set of support regions in the area of interest; the output $\mathbf{y}$ is the response map. The LNF patch expert with the edge features used is summarised in Figure 6.3.

The LNF patch expert uses two similarity edge features g_k to enforce smoothness on connected nodes. $S^{(g_1)}$ is defined to return 1 (otherwise return 0) only when the two nodes i and j are direct (horizontal/vertical) neighbours in a grid. $S^{(g_2)}$ is defined to return 1 (otherwise 0) when i and j are diagonal neighbours in a grid.

A single sparsity enforcing edge feature l_k is used. A neighbourhood region $S^{(l)}$ is defined to return 1 only when two nodes i and j are between

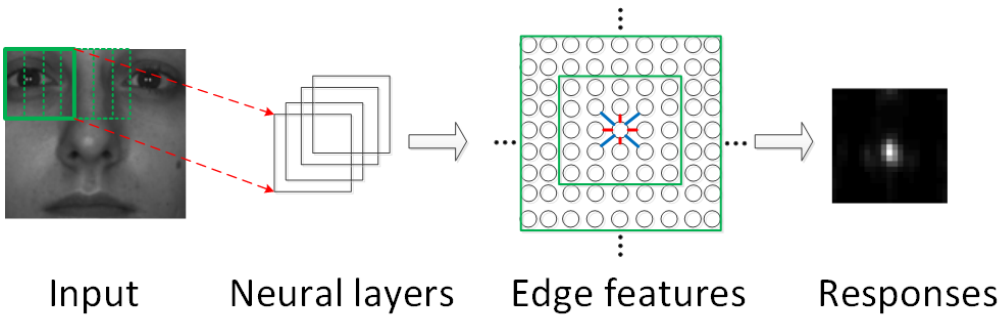


Figure 6.3: A visualisation of the LNF patch expert. The three types of edge features used are displayed, sparsity is enforced between the central node and the nodes surrounded by the green rectangle, similarity is enforced between the central nodes and the neighbours (two similarities illustrated in red and blue).

3 and 5 edges apart (where edges are counted from the grid layout of the LNF patch expert Figure 6.2).

LNF without edge features reduces to something similar to a three layer perceptron with a sigmoid activation function followed by a weighted sum of the hidden layers. It is also similar to the first layer of a Convolutional Neural Network (LeCun et al., 2010).

Figure 6.4 demonstrates the advantages of modelling spatial dependencies and input non-linearities by comparing LNF (with and without edge features) and SVR patch experts. Note how the response maps of LNF patch experts with edge features have fewer peaks and are smoother than the ones without edge features. Furthermore, the LNF patch experts are more accurate than the SVR ones.

6.2.1 Training

In order to collect training data the SVR sampling method can be used (Section 4.4). However, instead of each sample being considered a separate instance, they are grouped into 'sequences' of 9×9 samples, leading to a single observation $\boldsymbol{x} = \{x_1, \ldots, x_{81}\}$.

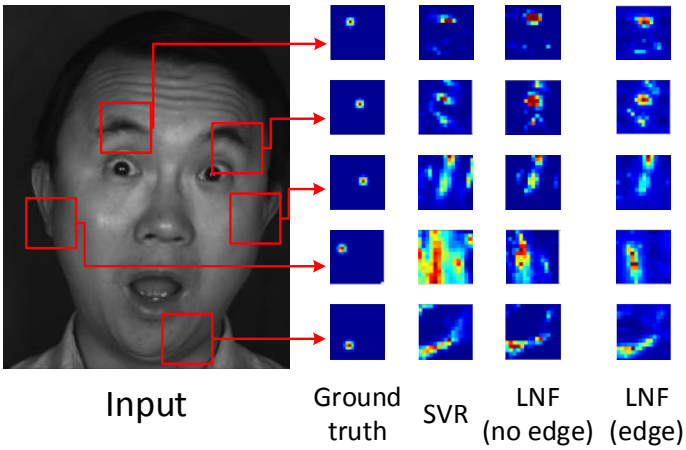


Figure 6.4: Different response maps from patch experts next to the actual landmark location (hotter is higher probability). The ideal response is shown above Ground Truth (used for training). SVR refers to the usual patch expert used by CLM approaches. Two instances of the LNF model are also shown: one without spatial features, that is without g_k and l_k, and one with. Note how the edge features lead to fewer peaks and a smoother response, improving the patch response convexity. Furthermore, note the noisiness of the SVR response.

6.3 Patch expert experiments

I conducted the following experiments to asses the properties of the new LNF patch expert. Firstly, the effect of spatial constraints on landmark detection was assessed. Secondly, LNF patch experts were compared to the SVR ones under easy illumination conditions. Finally, the ability of LNF to generalise across illuminations was evaluated; recall that in Section 4.8.1 it was shown that SVR based patch experts do not generalise well.

6.3.1 Methodology

I used the same data to train LNF patch experts as was used to train the SVR patch experts in Section 4.7.1. The fitting parameters used for CLNF are: $r = 25, \rho = 1.5, w = 5$ for landmark detection in images; $r = 20, \rho = 1.25, w = 10$ for tracking in videos.

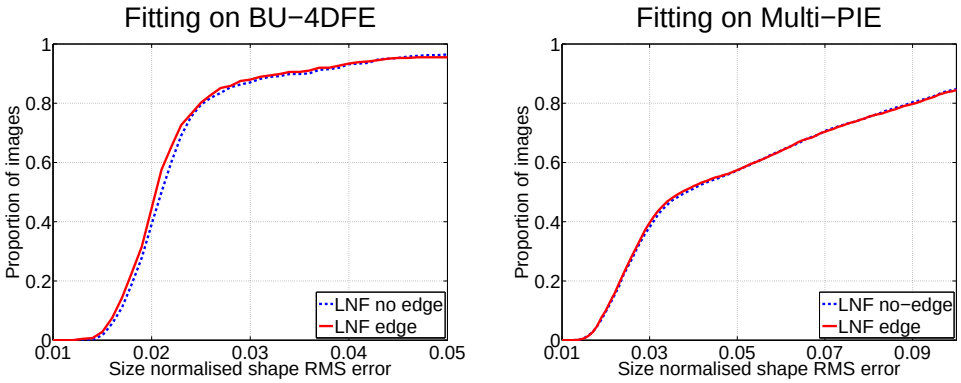


Figure 6.5: Comparison of using LNF patch experts with and without edge features. Observe the very slightly improved fitting accuracy with edge features.

The training and testing data never overlap, and the same participant never appears in both of the sets. The term *general illumination* refers to training data from both BU-4DFE and Multi-PIE datasets which includes four different lighting conditions: frontal, left, right and poorly lit. The term *frontal illumination* is used to refer to frontally lit faces used for training from both BU-4DFE and Multi-PIE datasets.

Two Multi-PIE subsets were used for testing: easy lighting and difficult lighting. Easy lighting includes frontally lit faces; difficult lighting includes left, right and poorly lit faces.

6.3.2 Importance of edge features

The first experiment explored the effect of modelling the relationships between output responses through the use of edge features in LNF patch experts. This was done by training general illumination LNF patches with and without edge features and comparing their performance on BU-4DFE and Multi-PIE frontal light datasets. Patch experts of three views were trained for the experiment - frontal and two profiles.

Results

The results of using LNF with and without edge features on the two datasets can be seen in Figure 6.5.

Wilcoxon sign rank test revealed significant differences ($z = 6.48, p < 0.001$) between LNF patch experts which use the edge features (Mdn $= 0.0204$) versus those which do not (Mdn $= 0.0210$) on the BU-4DFE.

Wilcoxon sign rank test revealed significant differences ($z = 3.09, p < 0.01$) between LNF patch experts which use the edge features (Mdn $= 0.0373$) versus those which do not (Mdn $= 0.0386$) on the Multi-PIE frontally lit dataset.

Discussion

Edge features provide a very small but statistically significant improvement to fitting accuracy both on BU-4DFE and Multi-PIE datasets. These results provide support for the use of LNF patch experts with edge features for landmark detection in images. Therefore, in all of the further sections LNF refers to LNF patch experts with edge features.

6.3.3 Facial landmark detection under easy illumination

I conducted an experiment to see how well the LNF patch experts perform for landmark detection on frontally lit faces, when only such faces were seen in training. This is a task on which SVR based patch experts perform well, but I wanted to see if LNF can improve on these results. For this experiment both SVR and LNF patch experts were trained on frontal illumination and tested on the Multi-PIE frontal illumination subset and the BU-4DFE dataset.

As shown in previous sections using multi-modal SVR patch experts helps with performance. In this section, a uni-modal LNF patch expert is compared with an already improved multi-modal SVR patch expert.

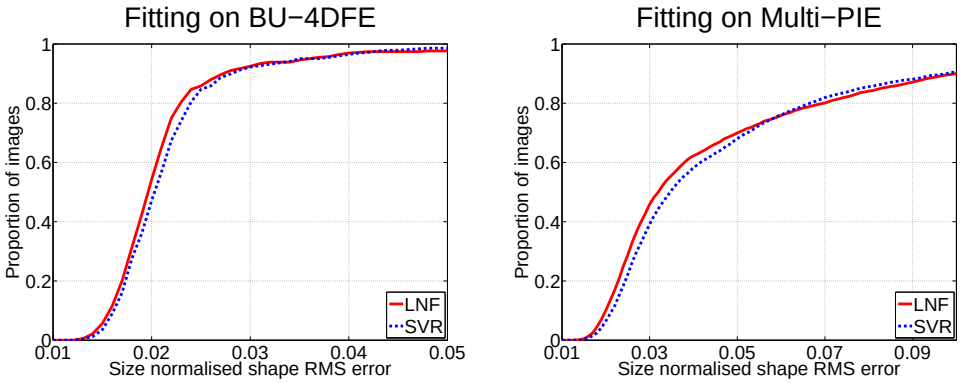


Figure 6.6: Fitting on the Multi-PIE dataset, observe how the LNF patch experts outperform the SVR ones.

Results

Results of this experiment can be found in Figure 6.6.

Wilcoxon sign rank test revealed significant differences ($z = -5.60, p < 0.001$) between LNF patch experts (Mdn $= 0.0197$) versus SVR ones (Mdn $= 0.0204$) on the BU-4DFE dataset.

Wilcoxon sign rank test revealed significant differences ($z = -19.2, p < 0.001$) between LNF patch experts (Mdn $= 0.0320$) versus SVR ones (Mdn $= 0.0347$) on the Multi-PIE single light dataset.

Discussion

LNF based patch experts statistically significantly outperformed SVR patch experts on both of the Multi-PIE and BU-4DFE datasets when trained and tested on the same illumination. Furthermore, the LNF patch experts used were uni-modal, but they still outperformed the multi-modal SVR ones which used both intensity and gradient intensity images.

6.3.4 Facial landmark detection under general illumination

The big problem with SVR patch experts and the main motivation behind LNF patch experts is the inability of the former to learn under a

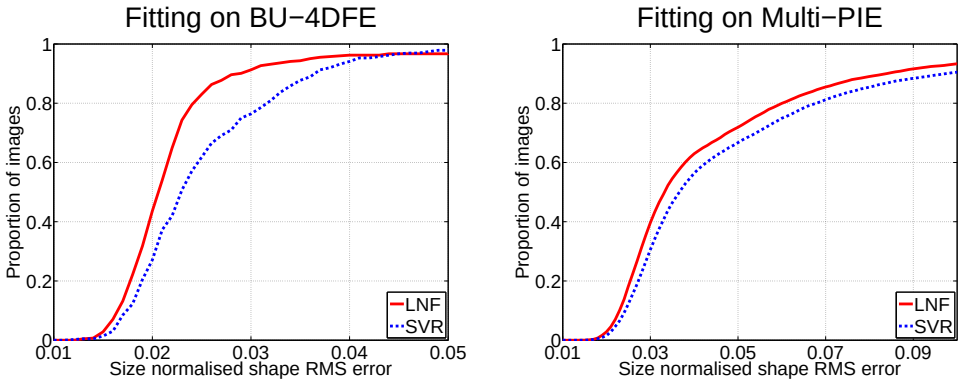


Figure 6.7: Fitting on the Multi-PIE dataset and BU-4DFE datasets when training on the general illumination case. Observe a marked improvement when using LNF patch experts.

number of illumination conditions. In this experiment, I wanted to see how the LNF patch experts affect the landmark detection accuracy on the Multi-PIE difficult lighting subset, Multi-PIE frontal lighting subset and the BU-4DFE dataset. Unless otherwise stated, both the SVR and LNF patch experts are trained on the general illumination training set.

Furthermore, I wanted to see the effect of training more general patch experts has on a frontal illumination test set. In the SVR case this results in degraded performance, forcing a trade-off between robustness and accuracy.

Results

The results of using general illumination trained SVR and LNF patch experts on the Multi-PIE difficult lighting subset and the BU-4DFE dataset can be seen in Figure 6.7.

Wilcoxon sign rank test revealed significant differences ($z = -9.65, p < 0.001$) between LNF patch experts (Mdn $= 0.0207$) versus SVR ones (Mdn $= 0.0229$) on the BU-4DFE dataset.

Wilcoxon sign rank test revealed significant differences ($z = -68.06, p < 0.001$) between LNF patch experts (Mdn $= 0.0332$) versus SVR ones

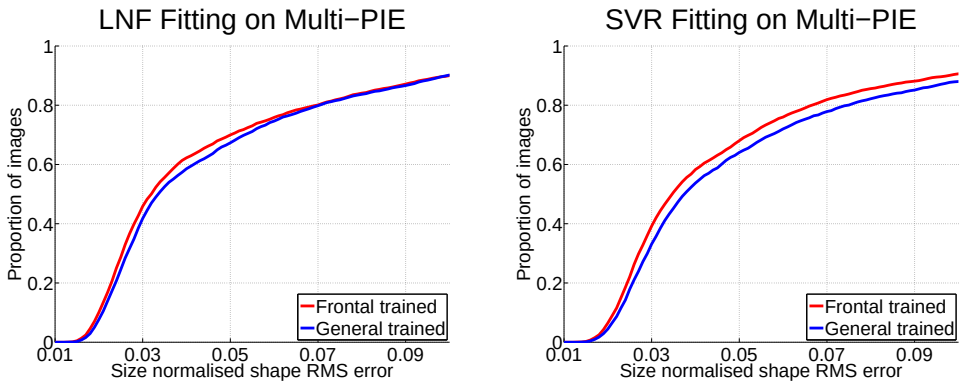


Figure 6.8: The effect of using patch experts trained on general versus specific illumination cases. Observe how when using SVR patch experts the use of general training has a big negative effect on the specific case. This is not the case when using LNF patch experts, which can learn a general model without sacrificing accuracy.

(Mdn = 0.0366) on the Multi-PIE difficult light dataset.

The effect of different training sets (specific and general illumination) when testing on the frontally lit Multi-PIE subset can be seen in Figure 6.8. Observe the bigger drop in performance when using general training with SVR patch experts: for LNF the drop in accuracy is from Mdn = 0.0320 to Mdn = 0.0335 and for SVR the drop in accuracy is from Mdn = 0.0347 to Mdn = 0.0374 (both statistically significant according to a Wilcoxon sign rank test). This demonstrates that LNF patch experts are better at learning a number of illuminations.

Discussion

The above experiments show the great benefit of LNF patch experts for a model which needs to work in more general environments. They substantially outperform SVR patch experts when testing on both Multi-PIE and BU-4DFE datasets. Furthermore, there is less degradation of performance when using a general model, compared to the SVR case. The novel LNF patch expert tackles the big problem that SVR patch experts face – difficulty of illumination independent landmark detection. LNF

patch experts make it easier to train a general tracker without sacrificing performance in a specific case.

6.4 General experiments

The previous section explored the use of LNF patch experts compared to the SVR based ones, demonstrating their advantages for landmark detection. In this section, I perform a broader set of experiments demonstrating the performance of CLNF (LNF with NU-RLMS) against other state-of-the-art approaches in landmark detection and head pose estimation.

6.4.1 Facial landmark detection

This section compares the performance of the CLNF model versus other state-of-the-art approaches for landmark detection on the BU-4DFE and the frontal and general illumination Multi-PIE datasets. I explored how CLNF compares to other approaches: when detecting landmarks on the same dataset under the same illumination; and when generalising to unseen lighting conditions and unseen datasets.

Baselines

One of the baselines used is the Zhu and Ramanan (2012) tree based model. It is a joint face detector, head pose estimator and landmark detector. The tree based model has shown very good performance at locating the face and the landmark features on a number of datasets. I used the trained models and the detection code provided by the authors. Zhu and Ramanan's approach has been trained on the Multi-PIE dataset using 900 faces from 13 viewpoints. 300 of those faces are frontal, while the remaining 600 are evenly distributed among the remaining viewpoints. The authors provide two models: a more accurate independent model which takes $\approx$ 40 seconds per image, and a less accurate fully-shared model which takes $\approx$ 5 seconds per Multi-PIE image (on a 3.06 GHz dual core Intel i3 CPU). I refer to these models as *tree based indepen-*

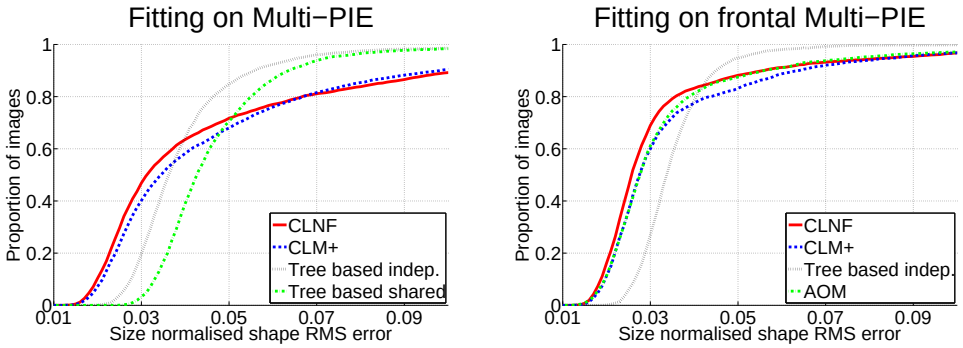
dent and *tree based shared* respectively. The amount of training data the authors used is comparable to 587 training images used for every view of LNF patch expert, making it a fair comparison.

Active Orientation Model (AOM) (Tzimiropoulos et al., 2012) is a generative model of facial shape and appearance. It is similar AAM, however, there are two differences: a different model of appearance and a robust algorithm for model fitting and parameter estimation. Therefore, it generalises better to unseen faces and variations. I used the trained model and the landmark detection code provided by the authors. Tzimiropoulos et al. (2012) trained and evaluated their model on close-to-frontal faces (-15 to 15 degrees yaw). They trained their models using 432 images from 54 different subjects. For each subject 8 images were used: 1 image for frontal (0 degrees) neutral expression, 2 images for 2 different viewpoints (-15 and 15 degrees of yaw) displaying neutral expression; and 5 frontal images (0 degrees) displaying the remaining 5 expressions. This is comparable to the number of training samples used for CLNF training.

As a final baseline, I used the CLM model introduced in the previous chapters: a multi-modal, multi-scale formulation with NU-RLMS fitting. For CLM the same training data was used as for the CLNF approach. Note this is a more accurate version of the CLM model presented by Saragih et al. (2011), and it is called CLM+ in this section.

Methodology

As CLNF was compared to other state-of-the art approaches that were not optimised for speed, I used CLNF parameters that make fitting slower, but more accurate. I used bigger areas of interest - 21×21 pixels and more NU-RLMS iterations - 4. This ensured a fair comparison. Other parameters used are as follows: $r = 25, \rho = 1.5, w = 5$. Furthermore, even with increased complexity, CLM and CLNF approaches still performed faster that the baselines they were being compared against. With these accuracy tuned parameters CLNF runs at 3 frames per second on a 3.06GHz dual core Intel i3 CPU.



(a) (b)

Figure 6.9: Comparison of CLNF and other landmark detectors on the frontally lit Multi-PIE dataset. a) The performance on the whole dataset. b) The performance on the close to frontal images (from -15° to 15° yaw). Notice how CLNF outperforms all of the other approaches in terms of accuracy, but the tree based models of Zhu and Ramanan (2012) are better in terms of robustness.

Performance on the same dataset and same illumination

In order to compare the CLNF method against the other state-of-the-art approaches, I performed landmark detection on the Multi-PIE dataset with frontal lighting. All of the approaches were trained on the frontally lit Multi-PIE dataset, on roughly the same amount of images (potentially on different subjects). CLNF and CLM approaches had BU-4DFE training data as well.

The results of the experiments can be see in Figure 6.9. The AOM model was only trained on close to frontal images, so a separate graph displaying the performance of baselines and CLNF is shown in Figure 6.9b.

To see the effect of landmark detectors on the RMSE, in the case of fitting on all of the Multi-PIE frontal lit images, a Friedman's ANOVA was performed. It revealed a significant effect of the method ($\chi^2(3) = 1822.8, p < 0.001$). Next, Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the landmark detection methods were significantly different ($p < 0.001$) from each other. The rankings from

most to least accurate are as follows (according to mean test rankings assigned during the ANOVA): CLNF (Mdn = 0.0312), tree based independent model (Mdn = 0.0362), CLM+ (Mdn = 0.0344), and tree based fully shared model (Mdn = 0.0426).

In order to assess the effect of landmark detectors on the RMSE, in the case of fitting on close to frontal Multi-PIE images, a Friedman's ANOVA was performed. It revealed a significant effect of the method ($\chi^2(3) = 1487.9, p < 0.001$). Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the landmark detection methods were significantly different ($p < 0.001$) from each other. This led to the following rankings from most to least accurate (according to test rankings assigned during the ANOVA): CLNF (Mdn = 0.0257), AOM (Mdn = 0.0276), CLM+ (Mdn = 0.0275), and tree based independent model (Mdn = 0.0337).

In sum, CLNF outperforms all of the other approaches in terms of error rates, demonstrating its ability to learn the Multi-PIE dataset well. However, it is also clear that the tree based approach is more robust. This shows the potential of combining the two approaches.

Performance on the same dataset but different illumination

The following experiment explored the ability of CLNF and other baselines to generalise to unseen illuminations. All of the detectors were trained on a frontal illumination and tested on the difficult illumination Multi-PIE dataset. Furthermore, the results of multi-illumination trained CLNF (CLNF-ML) were included, for a reference to a detector that generalises well.

The results of CLNF based landmark detection against previous baselines can be seen in Figure 6.10. It can be clearly seen that the results drop substantially when performing landmark detection on unseen lighting conditions, even on the same dataset. However, if we use multi-light training for CLNF the good results can be preserved. Furthermore,

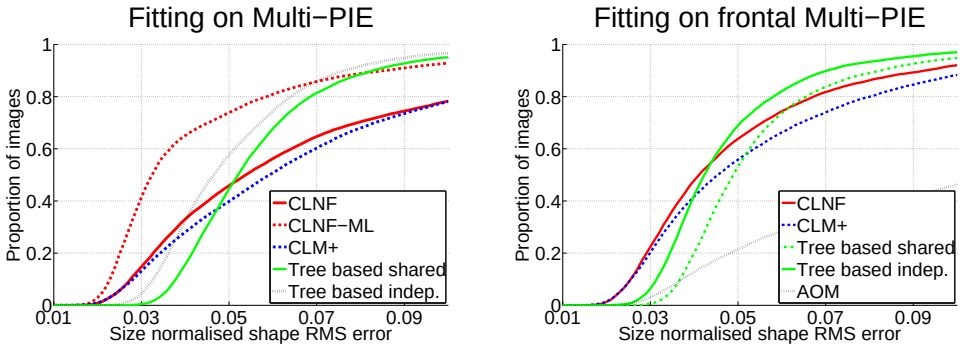


Figure 6.10: Comparison of CLNF and other landmark detectors on the Multi-PIE dataset, but on unseen to training illumination (except for CLNF-ML case which is there only for reference). Notice how the performance drops on unseen illumination, however the drop for a holistic approach (AOM) is the greatest.

even though it works well on frontal lighting, the holistic AOM fails to generalise to unseen lighting conditions with only 22% convergence. In contrast part based models perform much better with over 50% convergence.

To compare the ability of landmark detectors to generalise across illumination, a Friedman's ANOVA was performed on RMSE of all of the Multi-PIE general illumination images. It revealed a significant effect of the method ($\chi^2(3) = 1210.3, p < 0.001$). Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the landmark detection methods were significantly different ($p < 0.001$) except for the CLM+ and fully shared tree model. This led to the following rankings from most to least accurate (according to mean test rankings assigned during the ANOVA): tree based independent model (Mdn $= 0.0465$), CLNF (Mdn $= 0.0538$), tree based fully shared model (Mdn $= 0.0521$), and CLM+ (Mdn $= 0.0592$).

To compare the ability of landmark detectors to generalise across illumination on frontal images, a Friedman's ANOVA was performed on RMSE of frontal Multi-PIE general illumination images. It revealed a

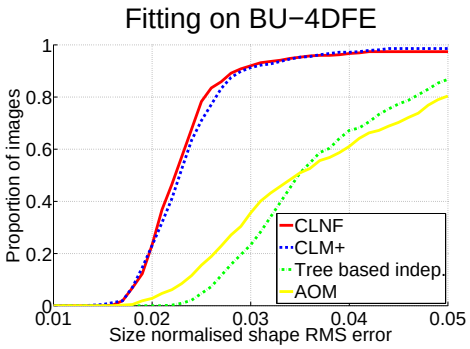


Figure 6.11: Comparison of CLNF and other landmark detectors on BU-4DFE datasets, when none of the detectors were trained on it.

significant effect of the method ($\chi^2(4) = 9849.1, p < 0.001$). Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the landmark detection methods were significantly different ($p < 0.001$). This led to the following rankings from most to least accurate (according to mean test rankings assigned during the ANOVA): CLNF (Mdn = 0.0411), tree based independent model (Mdn = 0.0423), CLM+ (Mdn = 0.0453), tree based fully shared model (Mdn = 0.0489), and AOM (Mdn = 0.111).

The results indicate that part based approaches are more suitable for generalisations to unseen illuminations. Furthermore, CLNF is not only orders of magnitude faster, but often outperforms the tree based methods of Zhu and Ramanan (2012) in terms of generalisability.

Performance on a different dataset

The final experiment assessed how well the different landmark detectors generalise to an unseen dataset, a crucial requirement for truly robust detectors. All of the detectors were trained on the frontal light Multi-PIE dataset and evaluated on the BU-4DFE dataset.

The results of this experiment can be seen in Figure 6.11. It can be seen that the CLNF and CLM approaches outperform the tree based

method of Zhu and Ramanan (2012) and the person independent AOM of Tzimiropoulos et al. (2012) by a large margin.

In order to examine the generalisability of landmark detectors across datasets, a Friedman's ANOVA was performed on the RMS errors on BU-4DFE images. Detectors were significantly different ($\chi^2(3) = 770.6, p < 0.001$). Friedman's ANOVAs were used to follow up the findings (a Bonferroni correction to p values was applied). The comparisons revealed that all of the landmarks were significantly different ($p < 0.001$) from each other, except for CLNF vs. CLM+ and AOM vs. independent tree based method. The following ordering was produced by the test from most to least accurate (according to test rankings assigned during the ANOVA): CLNF (Mdn = 0.0223), CLM+ (Mdn = 0.0230), AOM (Mdn = 0.0345) and tree based independent model (Mdn = 0.0349).

In conclusion, this experiment illustrates the better generalisation of CLNF and CLM to unseen datasets, highlighting their usefulness for real life scenarios.

Discussion

The experiments have shown that CLNF performs much better than other landmark detectors both when fitting on the same dataset, and when generalising to unseen data and unseen lighting conditions. Even the simpler CLM with my extensions displays better generalisability. This suggests that one can achieve better generalisability by modelling parts of a face rather than the whole face, as is the case with AOM or AAM.

Finally, CLNF (3fps) and CLM (5fps) approaches are faster than the AOM (2fps) and tree based models (0.03fps) for landmark detection in images. In all of the cases Matlab implementations were used, without much explicit optimisation running on a 3.06 GHz dual core Intel i3 CPU. These reported frame-rates are worse than ones reported for tracking in image sequences as they are run using Matlab code and with much bigger areas of interest. In conclusion, my CLNF model is both

faster and more accurate than models proposed by Zhu and Ramanan (2012) and Tzimiropoulos et al. (2012).

6.4.2 Facial landmark tracking

The effect of the CLNF model on tracking feature points in a video sequence was also assessed. For this, the subset of the Biwi head pose dataset that is labelled for feature points (see Section 3.2.2) was used. This is the same dataset that was used to evaluate the CLM-Z approach for facial feature tracking.

The training and fitting strategies were the same as in the previous experiments on video tracking (Section 4.7.5), but with slightly different fitting parameters. I used general illumination training, as the lighting in the dataset is varied and uncontrolled. For feature tracking in an image sequence using CLNF the model parameters were as follows: $r = 20$, $\rho = 1.25$, and $w = 10$. The approach was tested with and without reinitialisation strategies to assess both robustness and accuracy.

Baselines

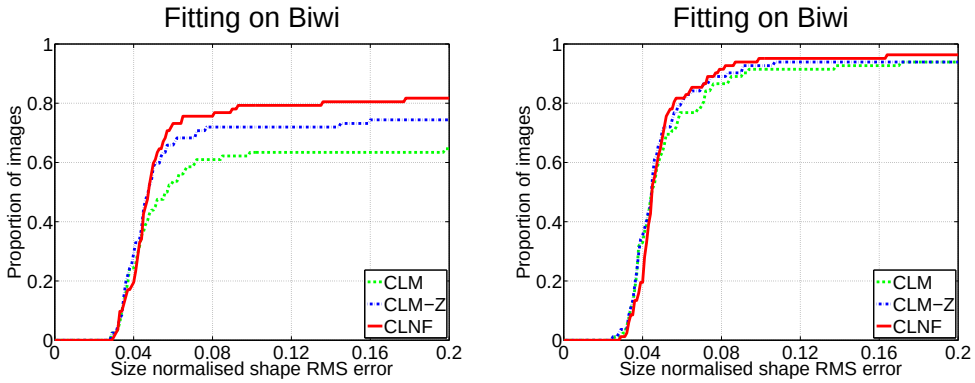
I compared CLNF to CLM and CLM-Z methods. Note that CLM-Z used both intensity and depth data, hence it was a slightly unfair comparison.

Results and Discussion

The results of the experiment are shown in Figure 6.12. The graphs indicate that the CLNF model is more robust than CLM and CLM-Z with and without reinitialisation. However, Friedman's ANOVA did not reveal any significant differences $p > 0.05$ in the no-reinitialisation case and $p > 0.1$ when reinitialisation was used.

6.4.3 Head pose estimation

In order to asses the ability of CLNF to track head pose I used the same datasets and baselines that were used to assess the performance of CLM-Z (Chapter 5). Furthermore, as one of the biggest advantages of CLNF is



(a) Without reinitialisation (b) With reinitialisation

Figure 6.12: Error fitting curves on the Biwi Kinect head pose dataset using the CLNF approach versus CLM and CLM-Z. Normalised by interocular distance.

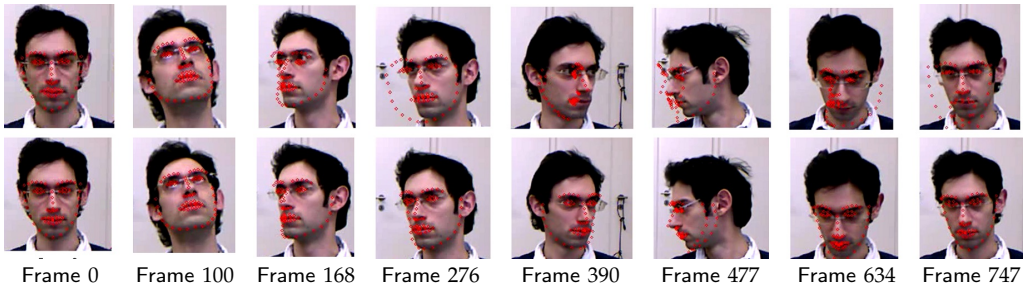


Figure 6.13: Sample point fitting on Biwi head pose dataset. Top row is CLM, bottom row is CLNF.

its ability to effectively learn illumination variations if they are provided in training, I trained two CLNF versions: CLNF-SL (single frontal light) and CLNF-ML (four lighting conditions).

Results

The results of the head pose estimation are displayed in Table 6.1. It is evident that the CLNF-ML model outperforms the other approaches based both on RGB and RGBD. These results highlight the importance of capturing lighting variation in training in order to fit on unseen datasets.

Model	Yaw	Pitch	Roll	Mean	Mdn.
	ICT-3DHP				
GAVAM (Morency et al., 2008)	6.58	5.01	3.50	5.03	3.08
CLM	5.41	4.32	4.83	4.85	2.45
CLNF-SL	4.72	3.51	4.47	4.24	2.41
CLNF-ML	**4.46**	**3.21**	**4.16**	**3.94**	**2.23**
	ICT-3DHP with depth				
GAVAM (Morency et al., 2008)	3.76	5.24	4.93	4.64	2.91
Reg. for. (Fanelli et al., 2011b)	7.69	10.66	8.72	9.03	5.02
CLM-Z	4.73	4.10	4.66	4.50	2.35
	Biwi-HP				
GAVAM (Morency et al., 2008)	14.16	9.17	12.41	11.91	5.63
CLM	10.32	10.27	9.01	9.87	3.58
CLNF-SL	9.89	9.37	7.84	9.03	3.73
CLNF-ML	8.14	7.37	**6.68**	**7.40**	**3.20**
	Biwi-HP with depth				
GAVAM (Morency et al., 2008)	**6.75**	**5.53**	10.66	7.65	3.92
Reg. for. (Fanelli et al., 2011b)	9.2	8.5	8.0	8.6	NA
CLM-Z	10.52	7.98	8.14	8.88	**3.20**

Table 6.1: Estimating head pose using the CLNF approach on the ICT-3DHP and the Biwi Kinect head pose datasets. The clear benefit of CLNF approach can be seen, even though it used only visible light.

Furthermore, the results show that the CLNF demonstrates comparable, or better, performance than dedicated rigid head pose trackers.

6.5 Conclusions

I presented a Constrained Local Neural Field model for facial landmark detection and tracking. The approach leads to more accurate landmark detection and head pose estimation, when compared to a number of state-of-the-art approaches.

The new LNF patch expert can exploit spatial relationships between patch response values, and learn non-linear relationships between pixel

values and patch responses. It is able to learn from multiple illuminations and retain accuracy. This becomes important when creating landmark detectors and trackers that are expected to work in unseen environments and on unseen people. The improvement in accuracy comes from the modelling of non-linear relationships, rather than spatial relationships (vertex rather than edge features). However, the edge features have a beneficial effect when modelling time series in Chapter 8 demonstrating the broader applicability of the CCNF model.

Finally, it achieves close-to-real-time/real-time performance ($\approx$ 20fps) bringing us closer to facial tracking in naturalistic environments.

7 Case study: Automatic expression analysis

7.1 Introduction

The previous chapters concentrated on making landmark detection and facial expression tracking more robust to changes in illumination and pose. This chapter will discuss how such tracked feature points could be used for the task of emotion recognition.

Most work in automated emotion recognition so far has focused on analysing the six discrete basic emotions: happiness, sadness, surprise, fear, anger and disgust. However, a single label (or multiple discrete labels from a small set) may not be sufficient for describing the complexity of an affective state. Consequently, There has been a move to analyse emotional signals along a small set of latent dimensions, providing a continuous rather than a categorical view of emotions. Examples of such affective dimensions are power (sense of control); valence (pleasant vs. unpleasant); activation (relaxed vs. aroused); and expectancy (anticipation).

Affective computing researchers have started exploring the dimensional representation of emotion (Gunes and Pantic, 2010; Ramirez et al., 2011; Schuller et al., 2011). The problem of dimensional affect recognition is often posed as a binary classification problem – active vs. passive; or even as a four-class one – classification into quadrants of a 2D space. In my work, however, the problem of dimensional affect recognition is one of regression.

In addition, most of the work so far has concentrated on analysing different modalities in isolation rather than looking for ways to fuse them (Gunes and Pantic, 2010; Zeng et al., 2009). This is partly due to the limited availability of suitably labelled multi-modal datasets and the difficulty of fusion itself: the optimal level at which the features should be fused is still an open research question (Gunes and Pantic, 2010; Lalanne et al., 2009; Zeng et al., 2009).

Conditional Random Fields (CRF) (Lafferty et al., 2001) and various extensions have proven very useful for emotion recognition tasks (Ramirez et al., 2011; Wöllmer et al., 2008). However, conventional CRF cannot be directly applied to continuous emotion prediction, as they model the output as discrete rather than continuous. I propose the use of Continuous Conditional Random Fields (CCRF) (Qin et al., 2008) in combination with SVRs for the task of continuous emotion recognition.

The CCRF model is applied to the task of continuous dimensional emotion prediction on the AVEC 2012 subset of the SEMAINE dataset (Schuller et al., 2012). The benefits of using this approach for emotion recognition are demonstrated by comparing it to the SVR baseline. Furthermore, the Correlation Aware Continuous Conditional Random Field model (CA-CCRF), which exploits the correlations between the emotion dimensions, is proposed.

My work also demonstrates the benefit of using facial geometry features for spontaneous affect recognition from video sequences. Such features are often ignored in favour of appearance based features, thus losing useful emotional information (Jeni et al., 2012). The reason geometry features are rarely used is the difficulty of acquiring a neutral expression from which facial shape deformation can be measured. My work shows how to extract a neutral expression and demonstrates the utility of geometry alongside appearance for emotion prediction.

The main contributions of this chapter are as follows:

- A fully continuous CCRF emotion prediction model which exploits temporal properties of the emotion signal

- A CA-CCRF model which can exploit correlations between emotional dimensions
- A novel way to fuse multi-modal emotional data
- A demonstration of the utility of facial geometry for continuous affect recognition

The work presented in this Chapter is the result of a collaboration with Ntombikayise Banda. I was responsible for the facial expression tracking and emotion modelling. Ntombikayise Banda extracted the appearance and audio features.

7.2 Background

Nicolaou et al. (2010) present a coupled Hidden Markov Model for classification of spontaneous affect based on Audio-Visual features. The model allows them to capture temporal correlations between different cues and modalities. They also show the benefits of using the likelihoods produced from separate (C)HMMs as input to another classifier. Such fusion is more accurate than picking the label with a maximum likelihood. The problem is formed as a classification rather than regression one.

Nicolaou et al. (2012) propose the use of the Output-Associative Relevance Vector Machine (OA-RVM) for dimensional and continuous prediction of emotions based on automatically tracked facial feature points. Their proposed regression framework exploits the inter-correlation between valence and arousal dimensions by including in their mode the initial output estimation together with their input features. In addition, OA-RVM regression attempts to capture the temporal dynamics of output by employing a window that covers a set of past and future outputs.

Of particular relevance to my dissertation is the work done by Wöllmer et al. (2008). They use Conditional Random Fields (CRF) for discrete emotion recognition by quantising the continuous labels for valence and

arousal based on a selection of acoustic features. In addition, they use Long Short-Term Memory Recurrent Neural Networks to perform regression analysis on these two dimensions. Both of their approaches demonstrate the benefits of including temporal information for the prediction emotions.

More recently Ramirez et al. (2011) proposed the use of Latent Dynamic Conditional Random Fields (LDCRF). Their approach attempts to learn the hidden dynamics between input features by incorporating hidden state variables which can model the sub-structure of gesture sequences. Their approach was particularly successful in predicting dimensional emotions from the visual signal. However, the LDCRF model can model only discrete output variables, hence the problem was posed as a classification one.

7.3 Continuous CRF

In my work I model affect continuously, rather than quantising it. Furthermore, my models capture the temporal relationships between each time step, since emotion has temporal properties and is not instantaneous (el Kaliouby et al., 2003). A recent and promising approach which attempts to model such temporal relationships is the Continuous Conditional Random Fields (CCRF) (Qin et al., 2008). It is an extension of the classic Conditional Random Fields (CRF) (Lafferty et al., 2001) to the continuous case. I adapt the original CCRF model so it can be used for continuous emotion prediction.

7.3.1 Model definition

CCRF is a discriminative undirected graphical model in which conditional probability $P(\mathbf{y}|\mathbf{x})$ is modelled explicitly. This is in contrast to generative models where a joint distribution $P(\mathbf{y}, \mathbf{x})$ is modelled. Discriminative approaches have shown promising results for sequence labelling and segmentation (Sutton and McCallum, 2006). The graphical

model that represents the linear-chain CCRF for emotion prediction is shown in Figure 7.1.

The notation is as follows: $\boldsymbol{x}^{(q)} = \{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a set of observed input variables, $\{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a set of output variables to be predicted, and n is the number of frames/time-steps in a sequence. $\mathbf{x}_i^{(q)} \in \mathcal{R}^m$ and $y_i^{(q)} \in \mathcal{R}$, where m is the number of predictors used. q indicates the q^{th} sequence of interest.

The CCRF model for a particular sequence is a conditional probability distribution with the probability density function:

$$P(\mathbf{y}|\boldsymbol{x}) = \frac{\exp(\Psi)}{\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}} \tag{7.1}$$

$$\Psi = \sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, \boldsymbol{x}) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j, \boldsymbol{x}) \tag{7.2}$$

Above, $\boldsymbol{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is the set of input feature vectors (it can be represented as a matrix with per frame observations as rows) and $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ is the unobserved variable. $\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}$ is the normalisation (partition) function which makes the probability distribution a valid one (by making it sum to 1). Following the convention of Qin et al. (2008), f_k refers to vertex features, and g_k to edge features. The model parameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots \alpha_{K1}\}$ and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots \beta_{K2}\}$ are used for inference and need to be estimated during learning. This model is very similar to CCNF introduced in Section 6.1.

7.3.2 Feature functions

Two types of features are defined for the linear-chain CCRF model: vertex features f_k and edge features g_k.

$$f_k(y_i, \boldsymbol{x}) = -(y_i - \mathbf{x}_{i,k})^2, \tag{7.3}$$

$$g_k(y_i, y_j, \boldsymbol{x}) = -\frac{1}{2} S_{i,j}^{(k)} (y_i - y_j)^2. \tag{7.4}$$

Vertex features f_k represent the dependency between the $\mathbf{x}_{i,k}$ (the k^{th} element of $\mathbf{x}_i$) and y_k, for example dependency between a static emotion

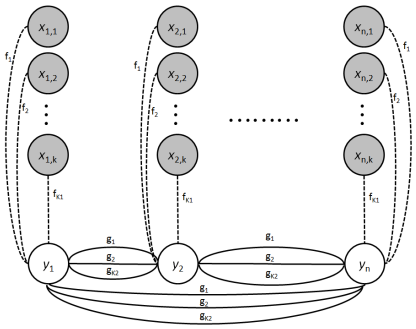


Figure 7.1: Graphical representation of the CCRF model. $\mathbf{x}_{i,k}$ represents the k^{th} feature of the i^{th} observation, and y_i is the unobserved variable to be predicted. Dashed lines represent the connection of observed to unobserved variables (f_k vertex features), so the first predictor is connected using f_1, whilst the k^{th} predictor is connected using f_k. The solid lines show connections between the unobserved variables (edge features): the first connection is controlled by g_1, the k^{th} connection is controlled by g_k. In the CCRF model all the output variables y_i are connected to each other (edge functions can break connections by setting the appropriate $S_{i,j}$ to 0)

prediction from a regressor and the actual emotion label. Intuitively, the corresponding α_k for vertex feature f_k represents the reliability of the k^{th} predictor. This is particularly useful for multi-modal fusion, as it models the reliability of a particular signal for a particular emotion. For example, the CCRF model could learn that the facial appearance might be more important in predicting valence than the audio signal.

Edge features g_k represent the dependencies between observations y_i and y_j, for example how related is the emotion prediction at time step j to the one at time step i. This is also affected by the similarity measure $S^{(k)}$. As a fully connected model is used, the similarities $S^{(k)}$ allow us to control the strength or existence of such connections. Two types of similarities are defined for emotion modelling:

$$S_{i,j}^{(\text{neighbour})} = \begin{cases} 1, & |i-j| = n \\ 0, & \text{otherwise} \end{cases} \tag{7.5}$$

$$S_{i,j}^{(\text{distance})} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{\sigma}\right) \tag{7.6}$$

By varying n it is possible to construct a family of similarities and to connect the observation y_i, not only to y_{i-1}, but also to y_{i-2}, and so on. By varying σ another set of similarities is created. These similarities control how strong the connections between y terms should be based on how similar the x terms are. This framework enables the easy creation of different similarity measures, which could be used in other applications.

The learning phase of CCRF determines which of the similarities are important for the dataset of interest. For example, it can learn that for one emotion dimension eighbour similarities are more important than for others.

Following Radosavljevic et al. (2010) and Qin et al. (2008), the feature functions model the square error between prediction and a feature. Therefore, each element of the feature vector $\mathbf{x}_i$ should already be predicting the unobserved variable y_i. This can be achieved using any regression technique, such as: Support Vector Regression, linear regression, or neural networks.

7.3.3 Learning

This section describes how to estimate the parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ of a CCRF with quadratic vertex and edge functions. Given training data $\{\boldsymbol{x}^{(q)}, \mathbf{y}^{(q)}\}_{q=1}^{M}$ of M sequences, where each $\boldsymbol{x}^{(q)} = \{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a sequence of inputs and each $\mathbf{y}^{(q)} = \{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a sequence of real valued outputs. Matrix $\mathbf{X}$ denotes the concatenated sequence of inputs.

CCRF learning picks the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values that optimise the conditional log-likelihood of the CCRF on the training sequences:

$$L(\alpha, \beta) = \sum_{q=1}^{M} \log P(\mathbf{y}^{(q)} | \boldsymbol{x}^{(q)}) \tag{7.7}$$

$$(\bar{\alpha}, \bar{\beta}) = \arg\max_{\alpha, \beta}(L(\alpha, \beta)) \tag{7.8}$$

As the problem is convex (Qin et al., 2008), the optimal parameter values can be determined using standard techniques such as stochastic gradient ascent.

The probability density of CCRF (Equation 7.1) can be expressed as a multivariate Gaussian. This form helps with explanation of inference and makes the derivation of partial derivatives of Equation 7.7 easier. The process of converting CCRF to Gaussian form is very similar to that of CCNF in Section 6.1:

$$P(\mathbf{y}|\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})), \tag{7.9}$$

$$\Sigma^{-1} = 2(A+B). \tag{7.10}$$

The diagonal matrix A represents the contribution of $\boldsymbol{\alpha}$ terms (vertex features) to the covariance matrix. The symmetric matrix B represents the contribution of the $\boldsymbol{\beta}$ terms (edge features):

$$A_{i,j} = \begin{cases} \sum_{k=1}^{K1} \alpha_k, & i = j \\ 0, & i \neq j \end{cases}, \tag{7.11}$$

$$B_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \beta_k \sum_{r=1}^{n} S_{i,r}^k) - (\sum_{k=1}^{K2} \beta_k S_{i,j}^k), & i = j \\ -\sum_{k=1}^{K2} \beta_k S_{i,j}^k, & i \neq j \end{cases}. \tag{7.12}$$

A further vector $\mathbf{b}$ is defined, which describes the linear terms in the distribution:

$$\mathbf{b}_i = 2\sum_{k=1}^{K1} \alpha_k \mathbf{X}_{i,k}. \tag{7.13}$$

$\boldsymbol{\mu}$ is the mean value of the Gaussian CCRF distribution:

$$\mu = \Sigma\mathbf{b}. \tag{7.14}$$

The partial derivatives of the $\log P(\mathbf{y}|\boldsymbol{x})$ can now be derived (see Section 6.1 for a similar derivation of partial derivatives of CCNF):

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \alpha_k} = -\mathbf{y}^T\mathbf{y} + 2\mathbf{y}^T\mathbf{X}_{*,k}^T - 2\mathbf{X}_{*,k}\mu + \mu^T\mu + \mathrm{tr}(\Sigma), \tag{7.15}$$

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \beta_k} = -\mathbf{y}^T B^{(k)}\mathbf{y} + \mu^T B^{(k)}\mu + \mathrm{Vec}(\Sigma)^T\mathrm{Vec}(B^{(k)}), \tag{7.16}$$

$$B^{(k)} = \begin{cases} (\sum_{r=1}^{n} S_{i,r}^{(k)}) - S_{i,j}^{(k)}, & i = j \\ -S_{i,j}^{(k)}, & i \neq j \end{cases}. \tag{7.17}$$

In order to guarantee that the partition function is integrable, the following constraints have to hold: $\alpha_k > 0$ and $\beta_k > 0$ (Qin et al., 2008; Radosavljevic et al., 2010). Such constrained optimisation can be achieved by using partial derivatives with respect to $\log \alpha_k$ and $\log \beta_k$ instead of just α_k and β_k. A regularisation term is also added in order to avoid overfitting. The regularisation is controlled by λ_α and λ_β hyper-parameters (determined during cross-validation):

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \log \alpha_k} = \alpha_k \left(\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \alpha_k} - \lambda_\alpha \alpha_k\right), \tag{7.18}$$

$$\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \log \beta_k} = \beta_k \left(\frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \beta_k} - \lambda_\beta \beta_k\right). \tag{7.19}$$

Using these partial derivatives a CCRF learning algorithm that uses stochastic gradient ascent can be derived. The pseudocode is provided in Algorithm 5.

Algorithm 5 CCRF learning algorithm

Require: $\{\boldsymbol{x}^{(q)}, \mathbf{y}^{(q)}, S_q^{(1)}, S_q^{(2)}, \ldots, S_q^{(K)}\}_{q=1}^{M}$
  Params: number of iterations T, learning rate ν, $\lambda_\alpha, \lambda_\beta$
  Initialise parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$
  for r = 1 **to** T **do**
    for i = 1 **to** N **do**
      Compute gradients of current query (Eqs.(7.18),(7.19))
      $\log \alpha_k = \log \alpha_k + \nu \frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \log \alpha_k}$
      $\log \beta_k = \log \beta_k + \nu \frac{\partial \log(P(\mathbf{y}|\boldsymbol{x}))}{\partial \log \beta_k}$
      Update $\{\alpha, \beta\}$
    end for
  end for
  return $\{\bar{\alpha}, \bar{\beta}\} = \{\alpha, \beta\}$

7.3.4 Inference

Because the CCRF model can be viewed as a multivariate Gaussian, inferring $\mathbf{y}$ values that maximise $P(\mathbf{y}|\mathbf{x})$ is straightforward. The prediction is the mean value of the distribution.

$$\mathbf{y}^* = \arg\max_{\mathbf{y}}(P(\mathbf{y}|\boldsymbol{x})) = \boldsymbol{\mu} = \Sigma\mathbf{b} \tag{7.20}$$

One of the downsides of similarity constraints in CCRF is that inference can sometimes lead to an over-smoothed and dampened signal. This is more likely to happen if the input variables $\boldsymbol{x}$ are very noisy, leading to CCRF learning to trust temporal consistency much more than the $\mathbf{x}$ observations. To combat the over-smoothing, one could use very high λ_β values to force the training to rely on the $\boldsymbol{\alpha}$ predictions more than on temporal elements controlled by $\boldsymbol{\beta}$. However, this would be at a cost of retaining a noisy signal. Alternatively, the scaling term s can be learned from the same training data (after the CCRF training is finished). Then the inference becomes $\mathbf{y}^* = s \cdot \boldsymbol{\mu}$, leading to a correctly scaled signal.

Furthermore, if multiple CCRF models are to be trained (as is the case for dimensional emotions), the Z-score of both input $\boldsymbol{x}$ and output $\mathbf{y}$ variables can be used. This means that the same learning rate can be used on all of them. Normalisation also helps if one wants to use predictions from other dimensions in a single CCRF, as is done by CA-CCRF.

7.4 Video Features

In order to infer emotional state from the face one needs to track facial feature points and the head pose. In addition, knowing the landmark locations makes it possible to analyse the appearance around them. For tracking faces the CLM-GAVAM tracker (see Section 5.6) can be used.

7.4.1 Geometric features

In order to extract the geometric/shape features of facial expressions one needs to establish the neutral facial expression from which the expression is measured. The geometric configuration of the initial frame

is not always reliable, as not all video sequences start with a neutral expression. In order to extract a neutral expression, a PDM which separates the expression and morphology subspaces can be used (Equation 7.21). Such a PDM is needed to decouple shape deformations arising from identity and expression.

The CLM model is described by parameters $\mathbf{p} = [s, \mathbf{w}, \mathbf{q_m}, \mathbf{q_e}, \mathbf{t}]$, which can be varied to acquire various instances of the model: the scale factor s; object rotation $\mathbf{w}$ (axis angle rotation); 2D translation $\mathbf{t}$; a vector describing non-rigid variation of the identity shape $\mathbf{q_m}$; and expression shape $\mathbf{q_e}$ (similar to a model used by Amberg et al. (2008)). The point distribution model (PDM) is:

$$\mathbf{x}_i = s \cdot \mathbf{R}(\bar{\mathbf{x}}_\mathbf{i} + \mathbf{\Phi}_i \mathbf{q_m} + \mathbf{\Psi}_i \mathbf{q_e}) + \mathbf{t}, \tag{7.21}$$

here $\mathbf{x}_i = (x, y)$ denotes the 2D location of the i^{th} feature point in an image and $\bar{\mathbf{x}}_i = (X, Y, Z)$ is the mean value of the i^{th} element of the PDM in the 3D reference frame. The vector $\mathbf{\Phi}_i$ is the i^{th} eigenvector obtained from the training set that describes the linear variations of non-rigid shape of this feature point in morphology space (constructed from the Basel 3DMM dataset (Paysan et al., 2009)). The vector $\mathbf{\Psi}_i$ is the i^{th} eigenvector obtained from the training set that describes the linear variations of non-rigid shape in expression space (constructed from BU-4DFE (Yin et al., 2008)).

In order to fit CLM using the split PDM, the model is first optimised with respect to the morphology parameters q_m, followed by expression parameters q_e. After a frame is successfully tracked in a video sequence the morphology parameters are fixed and only expression parameters are optimised. Such optimisation can be performed using RLMS or NU-RLMS methods. After the fitting has been performed, the expression parameters $\mathbf{q}_e$ describe the deformations due to expression.

7.4.2 Appearance-based features

In addition to face geometry, appearance captures emotional information as well. Appearance can be described using local binary patterns

(LBPs), which have been widely used in facial analysis tasks due to their tolerance against illumination variations and their computational simplicity (Shan et al., 2009). The local binary code, as introduced by Ojala and Pietikainen (1996), can be defined for each pixel with respect to its neighbours as:

$$\begin{aligned} LBP_P(x_c, y_c) &= \sum_{n=0}^{P-1} s(i_n - i_c)2^n, \\ s(x) &= \begin{cases} 1 & x >= 0 \\ 0 & x < 0 \end{cases}, \end{aligned} \tag{7.22}$$

where (x_c, y_c) is pixel centre position, P represents the number of neighbouring pixels, i_n the intensity value of a neighbour pixel and i_c the intensity value of the centre pixel.

One extension of the LBP operator seeks to combine motion features with appearance features, thus incorporating the temporal dynamics of an image sequence (Zhao and Pietikainen, 2007). This is achieved by concatenating local binary pattens on three orthogonal planes (LBP-TOP): XY, XT and YT. The operator is expressed as:

$$LBP - TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T},$$

where the notation $(P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T)$ denotes a neighbourhood of P points equally sampled on a circle of radius R on XY, XT and YT planes, respectively. An LBP code is extracted from the XY, XT and YT planes for all pixels, and statistics of the three different planes are obtained and then concatenated into a single histogram. This is demonstrated in Figure 7.2. This technique incorporates spatial domain information through the XY plane, and spatio-temporal co-occurrence statistics through the XT and YT planes. A detailed explanation of the LBP-TOP feature can be found in Zhao and Pietikainen (2007).

In my work, I used the facial feature points from the CLM-GAVAM tracker to extract frontal faces from an image sequence. In order to extract a frontal face, perspective warping was used from the current

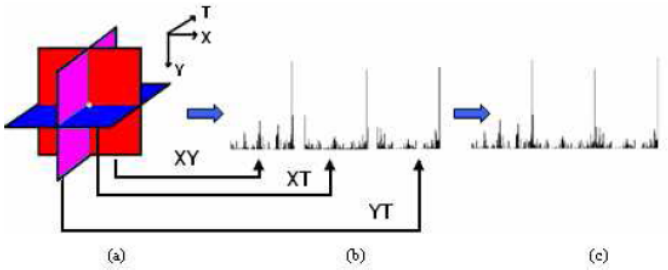


Figure 7.2: a) Three planes from which spatio-temporal local features are extracted. b) LBP histogram from each plane. c) Concatenated feature histogram. Taken from Zhao and Pietikainen (2007)

tracked points to the neutral reference frame, also ensuring size uniformity. The extracted faces were divided into a 3x3 non-overlapping grid, and LBP-TOP features were extracted for each block in the grid. Uniform patterns were applied, producing $P(P-1)+3$ output labels (instead of 2^P), resulting in a significant dimension reduction to a 59-dimensional histogram per image block (for $P = 8, R = 3$). A complete feature vector was obtained by concatenating the block histograms for each plane, resulting in a 1593-dimensional vector.

7.4.3 Motion features

Head gestures are an integral part of human communication as they convey a range of meanings and emotion. They involve a range of dynamics such as head orientation, rhythmic patterns, amplitude and speed of movement which act as indicators of affective states.

The CLM-GAVAM tracker estimates 6 degrees-of-freedom of head pose corresponding to head rotation and translation. The variation intensity of head motion was tracked by calculating the standard deviation of the rotational and translational parameters, a measure which takes into account the amplitude range and speed of change in head motion. In addition to these statistics, the Euclidean norm of all rotational parameters and that of translational parameters were added to describe the overall head movement. This resulted in the following 8-dimensional

feature vector:

$$\left[\sigma_{r_x}, \sigma_{r_y}, \sigma_{r_z}, \sigma_{t_x}, \sigma_{t_y}, \sigma_{t_z}, \sigma_{r_{xyz}}, \sigma_{t_{xyz}}\right],$$

where r corresponds to rotation parameters and t to translation parameters.

7.5 Audio Features

Vocal affect recognition analyses how things are said by extracting non-verbal information from speech. Scherer et al. (1989) states that emotion may produce changes in respiration, phonation and articulation, which in turn affect the acoustic features of the signal. Therefore, variations in acoustic measures contribute to our ability to discriminate between different emotional states. I adopted prosodic features used by Ozkan et al. (2012). Table 7.1 lists these features and provides motivations for their choice. Details of their extraction algorithms can be found in Ozkan et al. (2012).

7.6 Final system

The final emotion prediction system proposed is shown in Figure 7.3. The model depends on the per time step predictions from the previous layer. SVR is used, but this could be replaced by any other continuous predictor, such as linear regression or an artificial neural network. The features that were used with each SVR are explained in more detail in Sections 7.4 and 7.5. The CCRF model used is explained in Section 7.3.

CCRF can employ any number of SVR predictors, and the various combinations of them are explored in the evaluation section. Firstly, a system that just uses a prediction from an audio-visual SVR as its input ($K = 1$) was tested. Secondly, four SVR predictors (audio, shape, appearance, and pose) of the same dimension ($K = 4$) were used. Finally, as the emotional dimensions do not form an orthogonal set, the correlations between them were exploited using a Correlation Aware CCRF

Table 7.1: Description of the audio features used in this work.

Feature	Description	Motivation
Energy (in dB)	reflects the perceived loudness of the speech signal	has been found to have a high, positive correlation with arousal (Pereira, 2000), with increased intensity correlating well with valence (Schröder, 2004)
Articulation rate	is calculated by identifying the number of syllables per second	has been found to be positively correlated with arousal (Schröder, 2004)
Fundamental frequency (f_0)	is the base frequency of the speech signal (that is, the frequency the vocal folds are vibrating at during voiced speech segments)	has been found to have a high, positive correlation with arousal (Pereira, 2000); and a positive correlation between lower f_0 and power (Schröder, 2004)
Peak slope	is a measure suitable for the identification of breathy to strained voice qualities	there is evidence of a positive correlation between 'warm' voice quality and valence (Schröder, 2004)
Spectral stationarity	captures the fluctuations and changes in the voice signal; a measure of the speech monotonicity	monotonicity in speech is associated with low activity and negative valence (Davidson et al., 2003)

(CA-CCRF). This was achieved by including SVR predictions from other dimensions alongside the corresponding SVRs. Both the original and negated SVR predictions (from valence, arousal, expectancy and power dimensions) were used when training the four CA-CCRFs ($K = 32$ for each). This allowed me to capture both positive and negative correla-

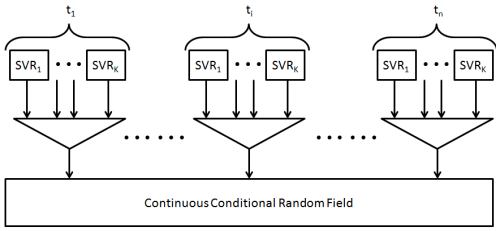


Figure 7.3: Final continuous emotion recognition system, which combines support vector regressors with continuous conditional random fields. The number of SVRs used can be varied, and depends on the experiment.

tions. In order to account for the fact that the dimensions have different scalings and different offsets, the Z-scores of the $\mathbf{X}^{(\mathbf{q})}_{*,\mathbf{k}}$ and $\mathbf{y}^{(\mathbf{q})}$ were used instead of raw values for training and inference.

7.7 Evaluation

7.7.1 Database

The proposed CCRF framework was evaluated using the dataset distributed through the AVEC 2012 Emotion Challenge (Schuller et al., 2012). This dataset forms part of the Solid SAL section of SEMAINE database, which contains naturalistic dialogues between two human participants, one of the participants simulating an artificial listener agent. The dataset was, however, partitioned differently from the challenge. The recordings were split into three partitions: training set I (for SVR training), training set II (for CCRF training) and a test set (for evaluation) with 21, 20 and 18 video sessions in each partition, respectively. The interactions were annotated by at least two raters along the dimensions arousal, valence, power and expectancy.

7.7.2 Methodology

The video features were extracted at a frame rate of 50 frames per second and down-sampled by employing the block averaging technique with a block size of 25 frames. The audio features were computed at 100Hz

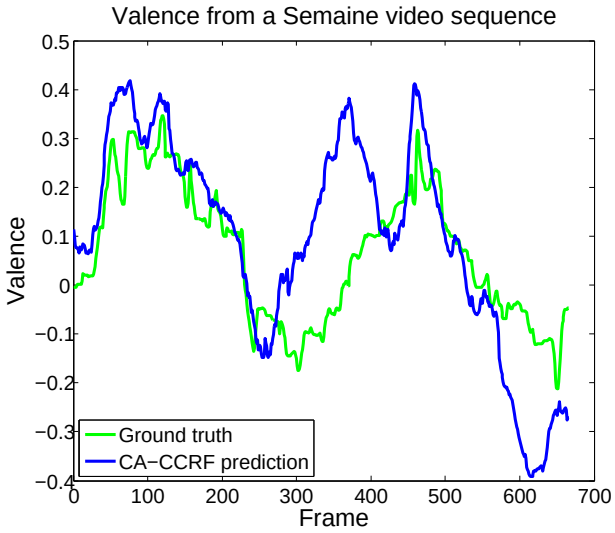


Figure 7.4: A plot of a standardized CA-CCRF valence prediction against the ground truth from the test partition

and down-sampled for alignment purposes. Linear kernel L2 loss ϵ-SVRs with L2 regularisation were used as initial CCRF layers. The training was performed using the Liblinear package (Fan et al., 2008). The SVR hyper-parameters were optimized using five-fold cross validation on training set I. Prediction labels were generated from each feature-type SVR model for the remaining two partitions for further CCRF training and inference. The training set II was used to determine the CCRF and CA-CCRF parameters ($\bar{\alpha}$, $\bar{\beta}$) and to cross-validate the regularization hyper-parameters λ_α and λ_β (ranging in $[10^{-2}, 10^0, 10^2, 10^4, 10^6]$). Ten edge features ($g_k$) were used for all experiments: 5 neighbour $n = \{1, 2, \cdots, 5\}$ and 5 distance $\sigma = \{2^{-6}, 2^{-7}, \cdots, 2^{-11}\}$. The learned $\bar{\beta}$ weights that model the temporal and spatial similarities of the signals, the channel reliability measures $\bar{\alpha}$, and SVR predictions were used to predict unseen data (test set). The continuous emotion label predictions were then up-sampled to the original video frame rate through linear interpolation. An example of a CCRF prediction is shown in Figure 7.4.

Baseline

SVR models were trained using both training sets I and II to ensure that the baseline and the CCRF models were exposed to the same training data. Uni-modal SVR models and a multi-modal SVR model (through early fusion) were trained for comparison with the CCRF framework.

7.7.3 Results

The model's performance was measured using Pearson's correlation coefficient (r) following the AVEC 2012 emotion challenge evaluation strategy. The results were obtained by computing the correlation coefficient between the predicted labels and ground truth labels per character interaction and per dimension, and calculating the average over all sessions. The following sections present the results of experiments conducted to evaluate CCRF ability to predict emotions.

Feature-type analysis

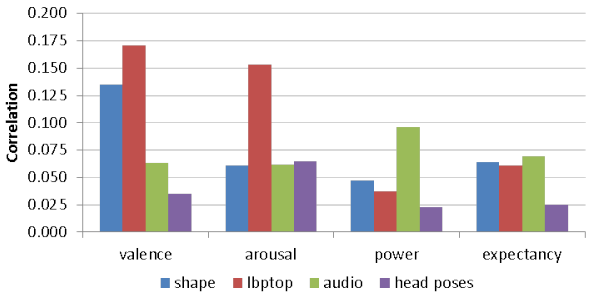


Figure 7.5: Comparison of the correlation results for each feature SVR model per dimensional affect

Figure 7.5 illustrates the performance of the feature SVR models for each dimensional emotion. The graph suggests that the appearance based features (temporal LBPs) are a better estimator of valence and arousal, and that audio features provide better predictions for power and expectancy. Apart from the arousal dimension, the shape (geometry) features do not perform much worse than the appearance ones, highlighting their potential as comparable estimators of emotion. The

Table 7.2: Correlation results of baseline SVR and CCRF models evaluated on the test partition

Model	Val.	Arous.	Pow.	Expect.	Mean
	video features				
SVR	0.176	0.234	0.100	0.120	0.158
CCRF	**0.311**	**0.294**	**0.171**	**0.214**	**0.248**
	audio features				
SVR	0.062	0.053	0.103	0.104	0.081
CCRF	**0.064**	**0.166**	**0.297**	**0.277**	**0.201**
	audio-visual features				
SVR	0.170	0.241	0.132	0.127	0.168
CCRF	**0.326**	**0.341**	**0.273**	**0.248**	**0.297**

Table 7.3: Investigating the fusion ability of CCRF with fused and non-fused predictor inputs

CCRF Inputs	Val.	Arous.	Pow.	Expect.	Mean
1 fused SVR	0.305	0.239	0.110	**0.275**	0.232
4 feature SVRs	**0.326**	**0.341**	**0.273**	0.248	**0.297**

Table 7.4: Comparison of CCRF and CA-CCRF model performances on test partition

	Val.	Arous.	Pow.	Expect.	Mean
CCRF	0.326	**0.341**	0.273	**0.248**	0.297
CA-CCRF	**0.343**	0.333	**0.309**	0.218	**0.301**

generally low correlation is indicative of the challenging task of working with naturalistic data and the variety of expressions that can be associated with an affective state.

Model and modality comparisons

Three types of modalities were investigated for both the baseline SVR and CCRF models: audio, video and audio-visual. Table 7.2 presents a comparative view of the three modalities for each model type. The results show that the CCRF model significantly outperforms the baseline

SVR in all modalities and dimensions. This attests to the importance of temporal data in the analysis and recognition of emotion, and the success of the CCRF model in capturing these dynamics.

Consistent with other studies (Nicolaou et al., 2012; Ozkan et al., 2012), it can be seen that visual features are better predictors of valence and that audio features perform better for the power dimension. However, in contrast to previous findings arousal state seems to be better predicted by visual rather than audio features.

Lastly, the CCRF model succeeded in fusing valence and arousal dimension; with overall results of the audio-visual CCRF outperforming the individual CCRFs.

Fusion strength of CCRF

Table 7.3 contrasts the use of a fused audio-visual SVR ($K = 1$) over the use of several SVR predictors ($K = 4$) as input to the CCRF model. The results show that fusing within the CCRF framework is better than providing fused predictors, therefore highlighting one of the strengths of the CCRF model: the information gain from using signal dynamics for fusion.

Correlations between dimensions

With reference to Table 7.4, it can be seen that the CA-CCRF model outperforms the regular CCRF for some dimensions. The effect of using CA-CCRF is especially beneficial for power dimension. This is not surprising as, in the dataset used, power correlates with other dimensions ($r = 0.25$ with valence, $r = 0.43$ with arousal and $r = -0.46$ with expectancy).

7.8 Conclusion

This chapter presented a CCRF model that can be used to model continuous dimensional emotion. It can easily incorporate multiple simple

predictors and exploits temporal correlations between time steps and different modalities. Furthermore, the model can easily be extended to include various other similarity functions that capture the dynamic nature of the signals. It also allows for high-order paths to be defined, exploiting long and short range dependencies of time series. The model is also able to exploit correlations between emotional dimensions leading to better prediction for some dimensions. The compact and simple CCRF design allows for applications in other domains with dynamic properties.

Further work

Future research might benefit from the comparison of using CCNF instead of a CCRF model, for emotion prediction from audio-visual signals. This would make the training more straightforward as only one model would need to be trained. Furthermore, joint learning of parameters might benefit the training. However, it is unclear how correlations between dimensions could be exploited by a CCNF model.

8 Case study: Emotion analysis in music

8.1 Introduction

So far in this dissertation I have concentrated on describing emotion recognition in humans (mainly from facial expressions and head pose). In this chapter I demonstrate how some of the tools developed in the previous chapters can also be used for emotion prediction in music.

Music surrounds us every day, and people's interaction with it is becoming increasingly digitized—buying digital music albums, streaming music, etc (BPI, 2013). This introduces a need to develop better tools for music search, playlist generation and the general management of music libraries. People use a number of different descriptors for songs, including emotion (Bainbridge et al., 2003).

In the same way as automatic emotion analysis from faces, the field of emotion recognition in music began by focusing on assigning a single categorical label to an entire piece of music. However, it has been slowly moving towards more complex techniques, with an increasing focus on dimensional representation of emotion, and continuous emotion tracking. Both of these, especially when combined, require more advanced machine learning techniques. However, until recently, there have been only a few approaches that could tackle this problem (see Section 8.2).

This chapter demonstrates how the Continuous Conditional Neural Fields (CCNF) model (Section 6.1) can be applied to the problem of continuous

dimensional emotion tracking in music. I compared the performance of CCNF with SVR and CCRF models. The experiments demonstrate that CCNF outperforms the other models in most cases.

The work presented in this chapter is the result of a collaboration with Vaiva Imbrasaitė. I was responsible for the emotion modelling; Vaiva Imbrasaitė extracted the audio features.

8.2 Background

Dimensional emotion representation describes emotion using several axes. In the field of emotion in music, the most commonly used dimensions are arousal and valence (AV). Adding other axes (such as expectancy and power) has also been considered, but it has repeatedly been shown that they add little to the description or the recognition of emotion in music (MacDorman, 2007).

Most of the approaches to dimensional continuous emotion tracking in music have focused on inferring the emotion label over a time window, which is independent of the surrounding music (bag-of-frames approach) (Korhonen et al., 2006; Panda and Paiva, 2011; Schmidt and Kim, 2010a; Schmidt et al., 2010). However, these approaches failed to exploit the temporal properties of music.

Some research has been done on trying to incorporate temporal information in to the feature vector—either by using features extracted over varying window length for each second/sample (Schubert, 2004), or by using machine learning techniques adapted for sequential learning. Examples include the sequential stacking algorithm used by Cohen and Carvahlo (2005), Kalman filtering or Conditional Random Fields (CRF) used by Schmidt and Kim (2010b, 2011).

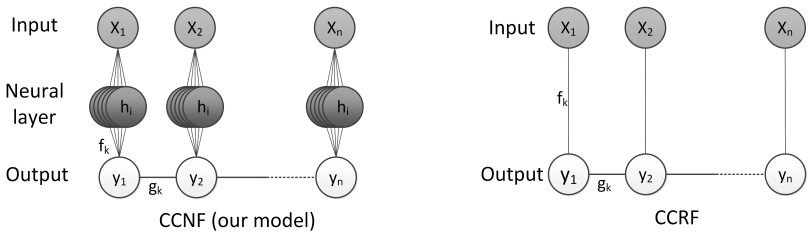


Figure 8.1: The linear-chain CCNF model compared to the linear-chain CCRF one (Section 7.3). The input vector $\mathbf{x}_i$ is connected to the relevant output scalar y_i through the vertex features that combine the h_i neural layers (gate functions) and the vertex weights $\boldsymbol{\alpha}$. The outputs are further connected with edge features g_k

8.3 Linear-chain Continuous Conditional Neural Fields

8.3.1 Model definition

It is possible to adapt the CCNF model introduced in Section 6.1 to use temporal rather than spatial relationships. The model is illustrated in Figure 8.1, and is called linear-chain CCNF. This model is also an extension of the linear-chain CCRF introduced in Section 7.3.

The model has two types of features: vertex features f_k and edge features g_k. The potential function is defined as:

$$\Psi = \sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, \mathbf{x}_i, \boldsymbol{\theta}_k) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j). \tag{8.1}$$

The vertex features f_k represent the mapping from the $\mathbf{x}_i$ to y_i through a one layer neural network, where $\boldsymbol{\theta}_k$ is the weight vector for a particular neuron k:

$$f_k(y_i, \mathbf{x}_i, \boldsymbol{\theta}_k) = -(y_i - h(\boldsymbol{\theta}_k, \mathbf{x}_i))^2, \tag{8.2}$$

$$h(\boldsymbol{\theta}, \mathbf{x}_i) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}_i}}. \tag{8.3}$$

The number of vertex features $K1$ is determined experimentally during cross-validation. The values tried during cross-validation were $K1 = \{5, 10, 20, 30\}$.

The edge features g_k represent the similarities between observations y_i and y_j. The existence and strength of the edge connections is controlled by the neighbourhood measure $S^{(k)}$.

In the linear-chain CCNF model, g_k enforces smoothness between neighbouring nodes. A single edge feature is defined, i.e. $K2 = 1$. $S^{(1)}$ is defined to be 1 only when the two nodes i and j are neighbours in a chain, otherwise it is 0:

$$g_k(y_i, y_j) = -\frac{1}{2} S^{(k)}_{i,j} (y_i - y_j)^2. \tag{8.4}$$

The linear-chain CCNF was used for emotion prediction in music. For training, song samples together with their corresponding dimensional continuous emotion labels were used. The dimensions were trained separately. See Section 6.1 for more details on the model and on its learning and inference.

8.4 Evaluation

A number of experiments were performed to assess the accuracy of the CCNF model when compared to several other baselines.

8.4.1 Dataset

The dataset used in the experiments is the only publicly available emotion tracking dataset of music extracts labelled on the arousal-valence dimensional space. The data (Speck et al., 2011) has been labelled using Mechanical Turk (MTurk)[1]. The paid participants were asked to label 15-second excerpts with continuous emotion ratings on the AV space, with another 15 seconds given as a practice for each song. The songs in the dataset cover a wide range of genres: pop, various types of rock and

[1] https://www.mturk.com/ - accessed May 2013

hip-hop/rap, and are drawn from the "uspop2002"[2] data-base containing popular songs. The dataset consists of 240 15-second clips (without the practice run), with $\mu = 16.9$, $\sigma = 2.7$ ratings for each clip. In addition, the dataset contains a standard set of features extracted from those musical clips: MFCCs, octave-based spectral contrast, statistical spectrum descriptors (SSD), chromagram, and a set of EchoNest[3] features.

8.4.2 Baselines

Several baselines were used for comparing against the CCNF model. The first baseline was the linear-chain CCRF. A single neighbour similarity feature was defined (as in the case of CCNF).

As an additional baseline, a linear and RBF kernel Support Vector Regressors were used. They have been used extensively for emotion prediction in music.

8.4.3 Error Metrics

Three different evaluation metrics were used in the experiments: correlation, root-mean-square error (RMSE) and Euclidean distance. Both the correlation coefficient and RMSE were calculated in two modes: *short* and *long*. Long evaluation metrics were calculated over the span of the whole dataset, essentially concatenating all of the songs into one. Short evaluation metrics were calculated over each song and then averaged over all of the songs. The short correlation metric is non-squared, so as not to hide any potential negative correlation. Short metrics might be better suited for evaluation of emotion recognition in music, as there has been some evidence that people agree more with algorithms that optimise the short RMSE (Imbrasaitė et al., 2013b).

Long metrics are reported as well, since these are usually reported in the literature. The average Euclidean distance was calculated as the distance between the two-dimensional position of the original label and

[2]http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html - accessed May 2013

[3]http://developer.echonest.com/ - accessed May 2013

the predicted label in the normalized AV space (each axis normalized to span between 0 and 1). Each metric was calculated for each fold and the average over 5 folds is reported.

Observe that lower RMSE and Euclidean distance values correspond to better performance, while the opposite is true for correlation.

8.4.4 Design of the experiments

For the purpose of this study, only the non EchoNest features provided in the MTurk dataset (Section 8.4.1) were used. Features were averaged over a one second window and the average of the labels for that second was used as the ground truth label. A separate model was trained for each emotional dimension: one for arousal and one for valence.

Features

Four types of features were used: MFCC, chromagram, spectral contrast and SSD. They were concatenated into a single vector. Their Z-scores were calculated on the training set (and the scalings were used on the test set).

Cross-validation

A 5-fold cross-validation was used for all of the experiments. The dataset was split into two parts: 4/5 for training and 1/5 for testing; this process was repeated 5 times. When splitting the dataset into folds, it was made sure that all of the feature vectors from a single song were in the same fold. The album and artist information was ignored, as it has been shown that they have no effect on this particular dataset (Imbrasaitė et al., 2013a). The reported results were averaged over 5 folds.

For SVR-based experiments, 2-fold cross-validation (splitting into equal parts) was used on the training dataset to choose the hyper-parameters. These were then used for training on the whole training dataset.

The process for the CCRF-based experiments contained an extra step. The training dataset was split into two parts—one for SVR and one for

Model	Arousal		Valence		Euclidean
	rms	corr.	rms	corr.	distance
SVR-Lin	0.196	0.634	0.222	0.173	0.130
SVR-RBF	0.194	0.645	0.220	0.211	0.128
CCRF	0.204	0.721	0.223	0.247	0.136
CCNF	**0.166**	**0.739**	**0.205**	**0.301**	**0.116**

Table 8.1: Results comparing the CCNF approach to the CCRF and SVR with linear and RBF kernels.

Model	Arousal		Valence	
	rms	corr.	rms	corr.
SVR-Lin	0.180	0.012	0.189	0.036
SVR-RBF	0.178	0.011	0.186	0.007
CCRF	0.176	0.049	0.183	**0.090**
CCNF	**0.143**	**0.072**	**0.170**	0.019

Table 8.2: Results comparing the CCNF approach to the CCRF and SVR with linear and RBF kernels. The metrics used are short correlation and root mean square error.

CCRF, and 2-fold cross-validation was performed on them individually to learn the hyper-parameters.

For CCNF-based experiments, 2-fold cross-validation was used to pick the hyper-parameters, but the results were averaged over 4 random seed initializations. The chosen hyper-parameters were used for training on the whole dataset. The model was randomly initialized 20 times (using the best hyper-parameters) and the model with the highest likelihood (Equation 6.7) was picked for testing.

It is important to note that the same folds were used for all of the experiments, and that the testing data were always kept separate from the training process.

Model	Arousal		Valence		Euclidean
	rms	corr.	rms	corr.	distance
W/o Chroma	0.167	0.737	0.207	0.285	**0.116**
W/o Contrast	**0.164**	**0.743**	0.208	0.285	**0.116**
W/o MFCC	0.175	0.707	**0.200**	**0.315**	0.117

Table 8.3: Results of CCNF with smaller feature vectors.

Model	Arousal		Valence	
	rms	corr.	rms	corr.
W/o Chroma	0.144	**0.068**	0.172	0.046
W/o Contrast	**0.143**	0.047	0.169	0.040
W/o MFCC	0.150	0.032	**0.164**	**0.089**

Table 8.4: Results (short evaluation metrics) of CCNF with smaller feature vectors.

8.4.5 Results

CCNF consistently outperforms all of the other methods on all the evaluation metrics except for short correlation for valence, where CCRF performs better (Tables 8.1 and 8.2). Not only is performance improved, but the results are substantially better than those of the other methods.

Since neural network-based models are particularly sensitive to the size of the feature vector used, the effect of a smaller feature vector was explored by omitting a class of features. As can be seen from Tables 8.3 and 8.4, not including chromagram, octave-based spectral contrast or MFCC features improves the results even further (compare with Tables 8.1 and 8.2).

8.5 Discussion

This chapter introduced an adaptation of CCNF — linear-chain CCNF. This model is particularly well suited for dimensional continuous emotion tracking.

The results achieved with this linear-chain CCNF are encouraging. It consistently outperformed the other models—both the standard baseline used in the field (SVR) and the more advanced CCRF model. These experiments demonstrate the applicability of CCNF for time-series modelling alongside its usefulness as a patch expert.

Schmidt and Kim (2010b) used the same dataset for their experiments and had a similar experimental design to the one presented in this chapter. They reported mean Euclidean distances of 0.160-0.169, which is within the same order of magnitude as the best average Euclidean distance of 0.116 achieved in the CCNF experiments. Unfortunately, even though the same dataset is used, the experimental design is slightly different so concrete conclusions are difficult to draw.

The experiments with smaller feature vectors showed that the size of the feature vector plays a major role in the performance of CCNF. The fact that better results were achieved by omitting a whole class of features shows that there may be some redundancy between the features. Thus, future work could investigate feature selection or sparsity enforcing techniques.

9 Conclusions

9.1 Contributions

The main goal of this dissertation was to bring facial expression analysis in real world environments closer to reality. My work has demonstrated multiple ways of making facial tracking techniques work better under varying illumination and pose.

The main contributions of this dissertation are as follows. Firstly, the exploration and extension of the Constrained Local Model (CLM) for facial tracking in difficult conditions. Secondly, the presentation of 3D Constrained Local Model (CLM-Z), a CLM based tracker that takes full advantage of depth information alongside visible light data. Thirdly, a Constrained Local Neural Field (CLNF) facial tracking model that is especially suited for tracking in difficult lighting conditions – when pose and illumination variations are expected. Finally, I demonstrated how these trackers can be used for emotion recognition in dimensional space using the tools developed. A brief description of these contributions follows.

9.1.1 Constrained Local Model extensions

I have presented a detailed analysis of the CLM for facial tracking. I extended it to use a multi-scale formulation and demonstrated how it can take into account different reliabilities of patch experts by using non-uniform regularised landmark mean shift. Finally, I identified a number of challenges still facing CLM based landmark detection: changes in illumination, extreme pose and extreme expressions.

9.1.2 3D Constrained Local Model

I introduced CLM-Z, which uses depth information alongside visible light data. I demonstrated how the training data for such a model could be generated synthetically, and used for training. Furthermore, I presented a novel normalisation function that allows CLM-Z to deal with missing data in the depth signal. This model was extensively evaluated on public datasets demonstrating its superiority over the regular CLM and a number of other head pose trackers.

9.1.3 Continuous Conditional Neural Field

I introduced the Continuous Conditional Neural Field graphical model. It is a regressor that can learn complex non-linear relationships and exploit some of the temporal and spatial characteristics of a signal. One of its instances – Local Neural Field can be used as a patch expert in the CLM framework, and is particularly suited for landmark detection under difficult illumination. Furthermore, another instance – linear-chain CCNF can be successfully used for emotion prediction from music.

9.1.4 Emotion inference in continuous space

I demonstrated how the facial landmark and head pose tracker developed throughout this dissertation can be used for emotion inference in continuous space. This was accomplished using the Continuous Conditional Random Fields on features extracted from the face together with some acoustic features.

9.2 Future work

My work addressed a number of existing issues in the field of facial tracking. It also points to certain areas which could benefit from further research.

First of all, my work did not explicitly address landmark detection under extreme expression, such as screaming and yawning. Exploring suit-

able priors or even different shape models might address this problem.

CLM is very dependant on good initialisation. There are few face detectors that deal with faces at various poses effectively. The face detectors that do exist are too slow to be of practical use (Zhu and Ramanan, 2012). Fast face detection in the wild and across pose is still an unsolved problem.

Finally, my work evaluated facial tracking on laboratory collected data. Further research is needed to see how well the algorithms would generalise in completely unconstrained environments.

Bibliography

Nalini Ambady and Robert Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences : a meta-analysis. *Psychological Bulletin*, 111(2):256–274, 1992.

Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3d face recognition with a morphable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

Ahmed Bilal Ashraf, Simon Lucey, Jeffrey F. Cohn, Tsuhan Chen, Zara Ambadar, Kenneth M. Prkachin, and Patricia E. Solomon. The painful face: Pain expression recognition using active appearance models. *Image and Vision Computing*, 27:1788–1796, 2009.

Hillel Aviezer, Ran R. Hassin, Jennifer Ryan, Cheryl Grady, Josh Susskind, Adam Anderson, Morris Moscovitch, and Shlomo Bentin. Angry, disgusted, or afraid? studies on the malleability of emotion perception. *Psychological science*, 19(7):724–732, 2008.

David Bainbridge, Sally Jo Cunningham, and J. Stephen Downie. How people describe their music information needs : A grounded theory analysis of music queries. In *International Conference on Music Information Retrieval*, 2003.

Tadas Baltrušaitis and Peter Robinson. Analysis of colour space transforms for person independent AAMs. In *The ACM / SSPNET 2nd International Symposium on Facial Analysis and Animation*, page 21, 2010.

Tadas Baltrušaitis, Daniel McDuff, Ntombikayise Banda, Marwa M. Mahmoud, Rana el Kaliouby, Rosalind W. Picard, and Peter Robinson. Real-time inference of mental states from facial expressions and upper body gestures. In *IEEE International Conference on Automatic Face and Gesture Recognition, Facial Expression Recognition and Analysis Challenge*, 2011.

Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2617, 2012.

Tadas Baltrušaitis, Ntombikayise Banda, and Peter Robinson. Dimensional affect recognition using continuous conditional random fields. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013a.

Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *International Conference on Computer Vision workshops*, 2013b.

Simon Baron-Cohen. Reading the mind in the face: A cross-cultural and developmental study. *Visual Cognition*, 3(1):3960, Mar 1996.

Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46, 1985.

Simon Baron-Cohen, Ofer Golan, Sally Wheelwright, and Jacqueline J. Hill. Mind reading: the interactive guide to emotions. 2004.

Janet B. Bavelas, Linda Coates, and Trudy Johnson. Listeners as co-narrators. *Journal of Personality and Social Psychology*, 79(6):941–952, 2000.

Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

BPI. Digital Music Nation. Technical report, 2013.

Martin Breidt, Heinrich H. Biilthoff, and Cristóbal Curio. Robust semantic analysis by synthesis of 3D facial motion. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.

Michael D. Breitenstein, Daniel Kuettel, Thibaut Weise, and Luc van Gool. Real-time face pose estimation from single range images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Daphne Blunt Bugental, Jaques W. Kaswan, and Leonore R. Love. Perception of contradictory meanings conveyed by verbal and nonverbal channels. *Journal of personality and social psychology*, 16(4):647–655, 1970.

Qin Cai, David Gallup, Cha Zhang, and Zhengyou Zhang. 3D deformable face tracking with a commodity depth camera. In *European Conference on Computer Vision*, 2010.

Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2894. IEEE, 2012.

Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000.

Justine Cassell. *Nudge nudge wink wink: elements of face-to-face conversation for embodied conversational agents*, volume 1, chapter 1, pages 1–27. MIT Press, 2000.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:127:27, 2011.

Sien W. Chew, Patrick Lucey, Simon Lucey, Jason M. Saragih, Jeffrey F. Cohn, and Sridha Sridharan. *Person-Independent Facial Expression Detection using Constrained Local Models*. 2011.

William W. Cohen and Vitor R. Carvahlo. Stacked sequential learning. In *International Joint Conference on Artificial Intelligence*, 2005.

Jeffrey F. Cohn. Foundations of human computing: Facial expression and emotion. In *ACM International Conference on Multimodal Interfaces*, pages 233–238, 2006.

Jeffrey F. Cohn, Tomas Simon Kruez, Iain Matthews, Ying Yang, Minh Hoai Nguyen, Margara Tejera Padilla, Feng Zhou, and Fernando De la Torre. Detecting depression from facial actions and vocal prosody. In *Affective Computing and Intelligent Interaction*, 2009.

Timothy F. Cootes and C. J. Taylor. *Statistical Models of Appearance for Computer Vision*. 2004.

Timothy F. Cootes and Christopher J. Taylor. Active Shape Models - "Smart Snakes". In *British Machine Vision Conference*, 1992.

Timothy F. Cootes, Kevin N. Walker, and Christopher J. Taylor. View-based active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

Timothy F. Cootes, Gareth Edwards, and Christopher Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, 2001.

Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George N. Votsis, Stefanos D. Kollias, Winfried A. Fellenz, and John Gerald Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):32–80, 2001.

David Cristinacce and Timothy F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, 2006.

David Cristinacce and Timothy F. Cootes. Boosted regression active shape models. In *British Machine Vision Conference*, 2007.

Charles Darwin. *The Expression of The Emotions in Man and Animals*. London, John Murray, 1872.

Richard J. Davidson, Klaus R. Scherer, and H. Hill Goldsmith. *Handbook of Affective Sciences*. 2003.

Beatrice de Gelder. Why bodies? twelve reasons for including bodily expressions in affective neuroscience. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1535):3475–3484, 2009.

Beatrice de Gelder and Jean Vroomen. The perception of emotions by ear and by eye. *Cognition and Emotion*, pages 289–311, 2000.

Fernando De la Torre and Jeffrey F Cohn. *Guide to Visual Analysis of Humans: Looking at People*, chapter Facial Expression Analysis. Springer, 2011.

Sidney D'Mello and Rafael Calvo. Beyond the basic emotions : What should affective computing compute? In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 2287–2294, 2013.

Neil A Dodgson. Variation and extrema of human interpupillary distance. In *Stereoscopic Displays and Virtual Reality Systems*, pages 36–46, 2004.

Guillaume-Benjamin-Amand Duchenne de Boulogne. *In Mecanisme de la Physionomie Humaine*. Cambridge University Press, 1862. Reprinting of the original 1862 dissertation.

Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6 (3):169–200, 1992.

Paul Ekman. *Language, knowledge, and representation*, chapter Emotional and conversational nonverbal signals, pages 39–50. Kluwer Academic Publishers, 2004.

Paul Ekman and Wallace V. Friesen. *Pictures of Facial Affect*. Consulting Psychologists Press, 1976.

Paul Ekman and Wallace V. Friesen. *Manual for the Facial Action Coding System*. Palo Alto: Consulting Psychologists Press, 1977.

Paul Ekman and Erika L. Rosenberg. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression using the Facial Action Coding System*. 2005.

Paul Ekman, Wallace V. Friesen, Maureen O'Sullivan, and Klaus R. Scherer. Relative importance of face, body, and speech in judgments of personality and affect. *Journal of Personality and Social Psychology*, 38:270–277, 1980.

Paul Ekman, Wallace V. Friesen, and Phoebe Ellsworth. *Emotion in the Human Face*. Cambridge University Press, second edition, 1982.

Rana el Kaliouby and Peter Robinson. *Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures*, pages 181–200. Springer US, 2005.

Rana el Kaliouby, Peter Robinson, and Simeon Keates. Temporal context and the recognition of emotion from facial expression. In *HCI International Conference*, pages 2–6, 2003.

Rong-En Fan, Chang Kai-Wei, Hsieh Cho-Jui, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real Time Head Pose Estimation with Random Regression Forests. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 617–624, 2011a.

Gabriele Fanelli, Thibaut Weise, Juergen Gall, and Luc Van Gool. Real time head pose estimation from consumer depth cameras. In *Deutsche Arbeitsgemeinschaft für Mustererkennung*, 2011b.

Gabriele Fanelli, Matthias Dantone, and Luc Van Gool. Real time 3d face alignment with random forests-based active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

Johnny R. J. Fontaine, Klaus R. Scherer, Etienne B. Roesch, and Phoebe C. Ellsworth. The world of emotions is not two-dimensional. *Psychological science*, 18(12):10501057, 2007.

Chris Frith. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3453–3457, 2009.

Xinbo Gao, Ya Su, Xuelong Li, and Dacheng Tao. A review of active appearance models. In *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, volume 40, pages 145–158, 2010.

Jeffrey M. Girard, Jeffrey F. Cohn, Mohammad H. Mahoor, Seyedmohammad Mavadati, and Dean P. Rosenwald. Social risk and depression : Evidence from manual and automatic facial expression analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

Leon Gu and Takeo Kanade. A generative shape regularization model for robust face alignment. In *IEEE European Conference on Computer Vision*, pages 413–426. Springer, 2008.

Hatice Gunes and Maja Pantic. Automatic, dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions*, 1(1): 68–99, 2010.

Hatice Gunes and Björn Schuller. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image and Vision Computing*, 31(2):120–136, 2013.

Uri Hadar, Timothy J. Steiner, and Frank Clifford Rose. Head movement during listening turns in conversation. *Journal of Nonverbal Behavior*, 9 (4):214–228, 1985.

Kristina Höök. Affective loop experiences: designing for interactional embodiment. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535), 2009.

Vaiva Imbrasaitė, Tadas Baltrušaitis, and Peter Robinson. Emotion tracking in music using Continuous Conditional Random Fields and relative feature representation. In *IEEE International Conference on Multimedia and Expo*, 2013a.

Vaiva Imbrasaitė, Tadas Baltrušaitis, and Peter Robinson. What really matters? a study into peoples instinctive evaluation metrics for continuous emotion prediction in music. In *Affective Computing and Intelligent Interaction*, 2013b.

Mircea C. Ionita, Peter Corcoran, and Vasile Buzuloiu. On color texture normalization for active appearance models. *IEEE Transactions on Image Processing*, 18(6):1372–1378, 2009.

László Jeni, András Lörincz, Tamás Nagy, Zsolt Palotai, Judit Sebök, Zoltán Szabó, and Dániel Takács. 3D shape estimation in video sequences provides high precision evaluation of facial expressions. *Image and Vision Computing*, 2012.

Patrik Juslin and Klaus R Scherer. *The New Handbook of Methods in Nonverbal Behavior Research*, chapter Vocal expression of affect, pages 65–135. 2005.

Dacher Keltner. Signs of appeasement: Evidence for the distinct displays of embarrassment, amusement, and shame. *Journal of Personality and Social Psychology*, 68(3):441–454, 1995.

Chris L. Kleinke. Gaze and eye contact: a research review. *Psychological bulletin*, 100(1):78–100, 1986.

Mark Korhonen, David A. Clausi, and Ed Jernigan. Modeling emotional content of music using system identification. *IEEE Transactions on Systems Man and Cybernetics Part B - Cybernetics*, 36(3), 2006.

Sanjiv Kumar and Martial Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification. volume 2, pages 1150–1157, 2003.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.

Denis Lalanne, Laurence Nigay, Philippe Palanque, Peter Robinson, Jean Vanderdonckt, and Jean-Francois Ladry. Fusion engines for multimodal input: A survey. In *International Conference on Multimodal Interfaces*, pages 153–160, 2009.

Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *International Symposium on Circuits and Systems*, pages 253–256, 2010.

Julien-Charles Lévesque, Louis-Philippe Morency, and Christian Gagné. *Sequential Emotion Recognition using Latent-Dynamic Conditional Neural Fields*. 2013.

John P. Lewis. Fast template matching. *Vision Interface*, 10:120–123, 1995.

Stephan Liwicki and Stefanos Zafeiriou. Fast and robust appearance-based tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 507–513, 2011.

Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason M. Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010.

Karl F. MacDorman, Stuart Ough Chin-Chang H. Automatic emotion prediction of song excerpts: Index construction, algorithm design, and empirical comparison. *Journal of New Music Research*, 36(4):281299, Dec 2007. doi: 10.1080/09298210801927846.

Marwa M. Mahmoud, Tadas Baltrušaitis, Peter Robinson, and Laurel D. Riek. 3D corpus of spontaneous complex mental states. In *Affective Computing and Intelligent Interaction*, 2011.

Marwa M. Mahmoud, Tadas Baltrušaitis, and Peter Robinson. Crowdsouring in emotion studies across time and culture. In *Proceedings of the ACM Multimedia workshop on Crowdsourcing for multimedia*. ACM Press, 2012.

Ioannis Marras, Joan Alabort-i Medina, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. Online Learning and Fusion of Orientation Appearance Models for Robust Rigid Object Tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

David Matsumoto and Bob Willingham. Spontaneous facial expressions of emotion of congenitally and noncongenitally blind individuals. *Journal of Personality*, 96(1):1–10, 2009.

Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

Iain Matthews, Jing Xiao, and Simon Baker. 2D vs. 3D Deformable Face Models: Representational Power, Construction, and Real-Time Fitting. *International Journal of Computer Vision*, 75(1):93–113, 2007.

Daniel McDuff, Rana el Kaliouby, David Demirdjian, and Rosalind W. Picard. Predicting online media effectiveness based on smile responses gathered over the internet. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

Gary McKeown, Michel F. Valstar, Roddy Cowie, and Maja Pantic. The SEMAINE corpus of emotionally coloured character interactions. In *IEEE International Conference on Multimedia and Expo*, 2010.

Dimitris Metaxas and Shaoting Zhang. A review of motion analysis methods for human nonverbal communication computing. *Image and Vision Computing*, 31:421–433, 2013.

Louis-Philippe Morency, Jacob Whitehill, and Javier Movellan. Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4), 2009.

Mihalis A. Nicolaou, Hatice Gunes, and Maja Pantic. Audio-visual classification and fusion of spontaneous affective data in likelihood space. In *International Conference on Pattern Recognition*, pages 3695–3699, 2010.

Mihalis A. Nicolaou, Hatice Gunes, and Maja Pantic. Output-associative RVM regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 30(3):186–196, 2012.

Jérémie Nicolle, Vincent Rapp, Kvin Baily, and Lionel Prevost. Robust continuous prediction of human emotions using multiscale dynamic cues categories and subject descriptors. In *ACM International Conference on Multimodal Interaction*, pages 501–508, 2012.

Timo Ojala and Matti Pietikainen. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

Derya Ozkan, Stefan Scherer, and Louis-Philippe Morency. Step-wise Emotion Recognition Using Concatenated-HMM. In *ACM International Conference on Multimodal Interaction*, pages 477–484, 2012.

Renato Panda and Rui Pedro Paiva. Using support vector machines for automatic mood tracking in audio music. In *130th Audio Engineering Society Convention*, 2011.

Maja Pantic and Marian Stewart Bartlett. *Face Recognition*, chapter Machine Analysis of Facial Expressions, pages 377–416. I-Tech Education and Publishing, 2007.

Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas Huang. Human computing and machine understanding of human behavior: A survey. In *ACM International Conference on Multimodal Interfaces*, pages 239–248, 2006.

Ulrich Paquet. Convexity and Bayesian constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1193–1199, 2009.

Ioannis Patras and Maja Pantic. Particle filtering with factorized likelihoods for tracking facial features. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 97–102, 2004.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveilance*, pages 296–301, 2009.

Allan Pease and Barbara Pease. *The Definitive Book of Body Language*. Orion, 2006.

Jian Peng, Liefeng Bo, and Jinbo Xu. Conditional neural fields. In *Advances in Neural Information Processing Systems*, pages 1419–1427, 2009.

Cécile Pereira. Dimensions of emotional meaning in speech. In *International Speech Communication Association Workshop on Speech and Emotion*, pages 25–28, 2000.

Rosalind W. Picard. *Affective Computing*. The MIT Press, 1997.

Rosalind W. Picard. Future affective technology for autism and emotion communication. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1535):3575–3584, Dec 2009.

Rosalind W. Picard and Jonathan Klein. Computers that recognise and respond to user emotion: Theoretical and practical implications. *Interacting with Computers*, 14(2):141–169, 2001.

Kenneth M. Prkachin and Patricia E. Solomon. The structure, reliability and validity of pain expression: evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.

Tao Qin, Tie-yan Liu, Xu-dong Zhang, De-sheng Wang, and Hang Li. Global ranking using continuous conditional random fields. In *Advances in Neural Information Processing Systems*, pages 1281–1288, 2008.

Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for regression in remote sensing. In *European Conference on Artificial Intelligence*, pages 809–814, 2010.

Geovany A. Ramirez, Tadas Baltrušaitis, and Louis-Philippe Morency. Modeling latent discriminative dynamic of multi-dimensional affective signals. In *1st International Audio/Visual Emotion Challenge and Workshop in conjunction with Affective Computing and Intelligent Interaction*, 2011.

Laurel D. Riek and Peter Robinson. *Using robots to help people habituate to visible disabilities.* 2011.

Peter Robinson and Rana el Kaliouby. Computation of emotions in man and machines. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3441–3447, 2009.

Paul Rozin and Adam B. Cohen. High frequency of facial expressions corresponding to confusion, concentration, and worry in an analysis of naturally occurring facial expressions of americans. *Emotion*, 3(1): 68–75, 2003.

James A. Russell and Albert Mehrabian. Evidence for a three-factor theory of emotions. *Journal of Research in Personality*, 11(3):273–294, 1977.

James A. Russell, Jo-Anne Bachorowski, and Jose-Miguel Fernandez-Dols. Facial and vocal expressions of emotion. *Annual review of psychology*, 54:329–349, 2003.

Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *Workshop on Analysis and Modeling of Faces and Gestures*, 2013.

Ashok Samal and Prasana A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: a survey. *Pattern recognition*, 25(1):65–77, 1992.

Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic, and Lijun Yin. Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697, 2012.

Jason M. Saragih and Roland Goecke. A Nonlinear Discriminative Approach to AAM Fitting. In *International Conference on Computer Vision*, 2007.

Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Face alignment through subspace constrained mean-shifts. In *IEEE International Conference on Computer Vision*, pages 1034–1041, 2009.

Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

Patrick Sauer, Timothy F. Cootes, and Christopher J. Taylor. Accurate Regression Procedures for Active Appearance Models. In *British Machine Vision Conference*, 2011.

Klaus R. Scherer. *Handbook of Cognition and Emotion*, chapter Approaisal Theory, pages 637–663. Wiley-Blackwell, 2005.

Klaus R. Scherer, H. Wagner, and A. Manstead. *Handbook of Psychophysiology: Emotion and social behavior*, chapter Vocal correlates of emotional arousal and affective disturbance, pages 165–197. 1989.

Erik M. Schmidt and Youngmoo E. Kim. Prediction of time-varying musical mood distributions from audio. In *International Society for Music Information Retrieval Conference*, pages 465–470, 2010a.

Erik M. Schmidt and Youngmoo E. Kim. Prediction of time-varying musical mood distributions using kalman filtering. In *International Conference on Machine Learning and Applications*, pages 655–660, 2010b.

Erik M. Schmidt and Youngmoo E. Kim. Modeling musical emotion dynamics with conditional random fields. In *International Society for Music Information Retrieval Conference*, pages 777–782, 2011.

Erik M. Schmidt, Douglas Turnbull, and Youngmoo E. Kim. Feature selection for content-based, time-varying musical emotion regression. In *International Society for Music Information Retrieval Conference*, pages 267–273, 2010.

Karen L. Schmidt and Jeffrey F. Cohn. Human facial expressions as adaptations: Evolutionary questions in facial expression research. *Yearbook of Physical Anthropology*, 44:3–24, 2001.

Karen L. Schmidt, Zara Ambadar, Jeffrey F. Cohn, and L. Ian Reed. Movement differences between deliberate and spontaneous facial expressions: Zygomaticus major action in smiling. *Journal of Nonverbal Behavior*, 30(1):37–52, 2006.

Marc Schröder. *Affective Dialogue Systems*, chapter Dimensional emotion representation as a basis for speech synthesis with non-extreme emotions, pages 209–221. 2004.

Emery Schubert. Modeling Perceived Emotion With Continuous Musical Features. *Music Perception*, 21(4), 2004.

Björn Schuller, Michel F. Valstar, Florian Eyben, Gary McKeown, Roddy Cowie, and Maja Pantic. AVEC 2011 - the first international audio / visual emotion challenge. In *Affective Computing and Intelligent Interaction*, 2011.

Björn Schuller, Michel F. Valstar, Roddy Cowie, and Maja Pantic. Avec 2012 - the continuous audio / visual emotion challenge - an introduction. In *ACM International Conference on Multimodal Interaction*, pages 361–362, 2012.

Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

Hyunjung Shim and Seungkyu Lee. Performance evaluation of time-of-flight and structured light depth sensors in radiometric/geometric variations. *Optical Engineering*, 51(9), 2012.

Tal Sobol-Shikler and Peter Robinson. Classification of complex information : Inference of co-occurring affective states from their expressions in speech. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1284–1297, 2010.

Yale Song, Louis-Philippe Morency, and Randall Davis. *Multimodal Human Behavior Analysis: Learning Correlation and Interaction Across Modalities*. 2012.

Jacquelin A. Speck, Erik M. Schmidt, Brandon G. Morton, and Youngmoo E. Kim. A comparative study of collaborative vs. traditional musical mood annotation, 2011.

Mikkel B. Stegmann and Rasmus Larsen. Multi-band Modelling of Appearance. *Image and Vision Computing*, 21(1):61–67, 2003.

Charles Sutton and Andrew McCallum. *Introduction to Statistical Relational Learning*, chapter Introduction to Conditional Random Fields for Relational Learning. MIT Press, 2006.

Motoi Suwa, Noboru Sugie, and Keisuke Fujimura. A Preliminary Note on Pattern Recognition of Human Emotional Expression. In *International Joint Conference on Pattern Recognition*, pages 408–410, 1978.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York Inc, 2010.

Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, May 2008.

Jessica L. Tracy and David Matsumoto. The spontaneous expression of pride and shame: evidence for biologically innate nonverbal displays. *Proceedings of the National Academy of Sciences of the United States of America*, 105(33):11655–11660, 2008.

Georgios Tzimiropoulos, Joan Alabort-i Medina, Stefanos Zafeiriou, and Maja Pantic. Generic Active Appearance Models Revisited. In *Asian Conference on Computer Vision*, pages 650–663, 2012.

Michel F. Valstar. *Timing is everything A spatio-temporal approach to the analysis of facial actions*. PhD thesis, 2008.

Michel F. Valstar and Maja Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *International Conference on Language Resources and Evaluation, Workshop on EMOTION*, pages 65–70, 2010.

Michel F. Valstar, Maja Pantic, Zara Ambadar, and Jeffrey F. Cohn. Spontaneous vs. posed facial behavior: Automatic analysis of brow actions. In *ACM International Conference on Multimodal Interfaces*, pages 162–170, 2006.

Michel F. Valstar, Hatice Gunes, and Maja Pantic. How to Distinguish Posed from Spontaneous Smiles using Geometric Features. In *ACM International Conference on Multimodal Interfaces*, 2007.

Michel F. Valstar, Brais Martinez, Xavier Binefa, and Maja Pantic. Facial point detection using boosted regression and graph models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2729–2736, 2010.

Michel F. Valstar, Bihan Jiang, Marc Mehu, Maja Pantic, and Klaus R. Scherer. The First Facial Expression Recognition and Analysis Challenge. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.

Sundar Vedula, Peter Rander, Robert T. Collins, and Takeo Kanade. Three-dimensional scene flow. In *IEEE International Conference on Computer Vision*, pages 722–729, 1999.

Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

Yang Wang, Simon Lucey, and Jeffrey Cohn. Non-rigid object alignment with a mismatch template based on exhaustive local search. In *IEEE International Conference on Computer Vision*, 2007.

Yang Wang, Simon Lucey, and Jeffrey F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *SIGGRAPH*, 2011.

Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *Interspeech*, pages 597–600, 2008.

Jing Xiao, Simon Baker, Ian Matthews, and Takeo Kanade. Real-time combined 2D + 3D active appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 535–542, 2004.

Jing Xiao, Jinxiang Chai, and Takeo Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, 2006.

Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A high-resolution 3D dynamic facial expression database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu. *A High-Resolution Spontaneous 3D Dynamic Facial Expression Database*. 2013.

Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–12, 2012.

Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6): 915–928, 2007.

Wen-Yi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, 2012.