A real-time facial expression recognition system for affective computing

Anderson, Keith William John

ISSN 1470-5559

A real-time facial expression recognition system for affective computing

Keith William John Anderson

Queen Mary
University of London
Department of Computer Science

RR-04-01 January 2004

A real-time facial expression recognition system for affective computing

Keith William John Anderson

A thesis submitted to the University of London
in partial fulfilment to the degree of
Doctor of Philosophy

Department of Computer Science
Queen Mary, University of London
2003

ABSTRACT

A fully automated multi-stage system for real-time recognition of facial expressions is presented. The novel system described uses facial motion to characterise frontal views of facial expressions and is able to operate effectively in cluttered and dynamic scenes, recognising the six emotions universally associated with unique facial expressions, namely happiness, sadness, disgust, surprise, fear, and anger.

The system has three main components, a face tracker, an optical flow algorithm, and an expression recognition system. The face tracker is required to locate the face in the scene and is a modification of, and an extension to, a spatial ratio template algorithm. Optical flow at the location of the face is determined by a real-time implementation of a multi-channel gradient model, whilst the expression recognition system uses the motion outputs of this model to recognise facial expressions, using Support Vector Machine classifiers.

The completed system is used to drive applications that respond in real-time to the facial expressions of the user, thereby providing improvements in the interaction between a computer user and technology. The completed system also has potential applications in the field of affective computing where it could be used in conjunction with other emotion-based techniques to develop a computing system able to express, recognise, and respond to people in a natural and human-like manner.

DECLARATION

I declare that this thesis has been composed by myself, that it describes my own work, that it has not been accepted in any previous application for a degree, that all verbatim extracts are distinguished by quotation marks and that all sources of information have been specifically acknowledged.

Additionally, some parts of the work presented in this thesis have been published in the following articles:

1. Anderson, K. & McOwan, P.W.: Real-time Emotion Recognition using Biologically Inspired Models, *4th International Conference on Audio and Video Based Biometric Person Authentication*. p119-127 (2003)

2. Anderson, K.W.J. & McOwan, P.W.: Changing Faces: The Science of Facial Expression, *to appear in 'As others see us' edited by Philip J. Hills*. Peter Francis Publishers. In Press 2004

3. Anderson, K. & McOwan, P.W., Robust Real-Time Face Tracker for Cluttered Environments, *Computer Vision & Image Understanding*. In Press 2004

4. Anderson, K. & McOwan, P.W., Enhanced Interaction with Technology: A Real-Time Affective Computing System for Recognising Facial Emotions. *Evolvability & Interaction Symposium. University of Hertfordshire Computer Science Technical Report No. 393*. (2003)

ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor, Peter McOwan, for his enthusiasm and for the time and advice he has provided over the past three years. My wife, Claire, who has patiently waited in Australia for me to complete this work, and my parents, particularly for their assistance in recent months. Also, to Andy Anderson and Adam Sherwood who have made the time spent at the office much more enjoyable than it could have been. All the other members of the Computer Vision group here at Queen Mary, from the present members, Alessio Del Bue, Lourdes de Agapito Vicente, Sean Gong, Andrew Graves, Hayley Hung, Alex Leung, Fabrizio Smeraldi, Tony Xiang, and Lukasz Zalewski, to those who used to work here in the past, Jack Chang, Ong Eng-Jon, Yongmin Li, Jamie Sherrah, Jeffrey Ng and Justin Lim. Finally, other members of the department, particularly Matt Bernstein, Tim Kay, and Pablo Armelin.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Affective computing is an area that addresses issues relating to emotion in computing and has been pioneered by the work of Picard at MIT [Picard 1995]. Picard describes how "affective interaction can have maximal impact when emotion recognition and expression is available to all parties, human and computational" and goes on to say that "if one party cannot recognise or understand emotions then interaction is impaired" [Picard 1998]. Thus, the field of affective computing examines the problem that the relationship between humans and technology currently goes in one direction only, with humans relating to their machines, but the machines giving nothing back due to their ignorance as to the moods and emotions of a human user.

For an affective computing system to be effectual, it not only has to recognise the emotions of a user, but also has to respond appropriately to these emotions. A response to a particular emotional state should not be hard-wired, but modified by factors such as the relationship between the computer and the user, the situation in which the emotion is felt and by using knowledge of the subjects personality and needs [Picard 1998].

The work presented here addresses this need for improved interaction between humans and computers, describing a fully automated real-time expression recognition system that can be used to drive a broad range of applications in the field of human computer interaction. The particular focus of this work is to address problems with previous solutions, specifically their slowness and/or requirement for some degree of manual intervention. These failings have meant that it has not been realistic for technology to respond in real-time to the facial expressions of a user.

Real-time recognition of expression is achieved in this work as the approach is restricted to recognising only frontal views of expressions at a single scale. However, these restrictions are not weaknesses in the application domains for which the use of this system is envisaged, namely expression recognition of a user seated in front of a personal computer. In such an environment, the user's gaze is generally directed at the

computer monitor and thus recognition from other viewpoints is not of real significance (NB the problem of pose change has not been seriously addressed in other systems where there are fewer computational constraints [Fasel 2003]). Also, the distance of the user's face from the camera is fairly constant, with the system presented here able to handle the limited changes in scale that may occur.

There are three main components to this system: a face tracker, an optical flow algorithm, and an expression recognition system. The face tracker is a modification of, and an extension to, the ratio template algorithm [Sinha 1995]. Optical flow is determined by a real-time version of a multi-channel gradient model [Johnston 1999], whilst the final expression recognition system uses Support Vector Machines. These components are integrated into a single system running at 4fps on a 384x247 image on a 450MHz Pentium III machine with Matrox Genesis DSP boards. The system is summarised in **figure 1.1**. The system developed is able to recognise the facial expressions of any user who sits in front of a computer equipped with a camera in real-time, even in cluttered and dynamic scenes. It recognises six expressions, happiness, sadness, disgust, surprise, fear, and anger.

The need for affective computing systems arises from the fact that, as humans, we are above all social animals. Both our evolutionary and developmental histories are fundamentally social, and thus in order to cope with social complexity and regulate social interaction we possess a vast repertoire of skills allowing us to assess, identify and predict the "internal states" of others. A vital component of this capacity is an ability to recognise and interpret facial expressions.

Interactions where the role played by facial expression is significant occur throughtout human society. A flustered parent flashes a look of anger at a naughty child and he immediately stops misbehaving. A man smiles warmly at a complete stranger at a party, the recipient's mood is lifted and they move towards one another for conversation. A market researcher approaches someone confidently in the street, but this advance is met by a scowl and the researcher instantly withdraws, looking for alternative prey. Facial expressions give richness and subtlety to our social exchanges,

allowing us not only to show what we are feeling, but also to affect the behaviour of those around us. Thus, to interact naturally in a social situation we need to form meaningful facial expressions and also correctly interpret the facial expressions of others.

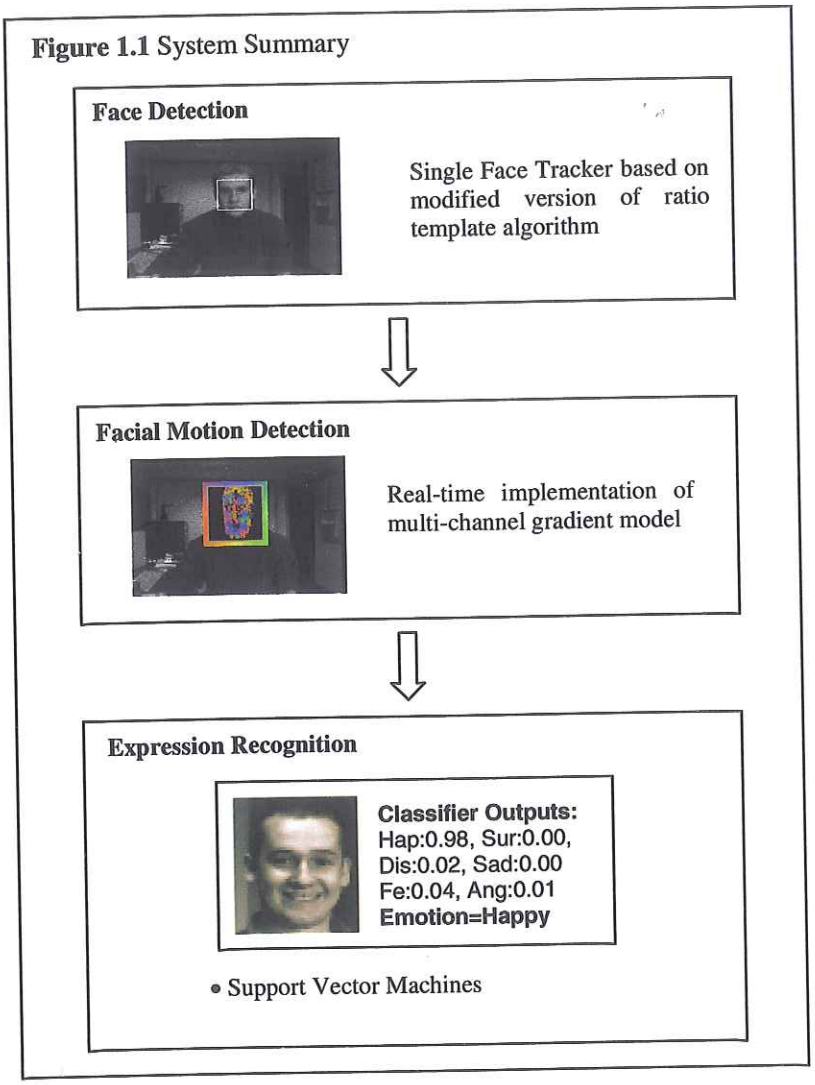


Figure 1.1 System Summary

It therefore follows that for interaction with technology to be improved and become more natural it is crucial that it be provided with human affective abilities such as display and recognition of facial expression. In recent years attempts have been made to do exactly this, with an ability to express emotion being bestowed upon some sociable robots [Breazeal 2003] and also on synthetic characters incorporated into

pieces of computer software. However, less progress has been made in providing such agents with skills enabling them to understand the emotional state of others and this thesis addresses this shortcoming.

To demonstrate the usefulness of the completed expression recognition system, this thesis presents exemplar applications that are driven by the outputs of the expression recognition system. These include an application that automatically inserts emoticons (widely understood symbolic abbreviations such as :) for happy) into the text for a chatroom user and another system that triggers a desktop application that either plays music or opens appropriate web-pages according to the facial expression of the user.

In addition to being a useful tool as a stand-alone system that can drive such applications as those described above, the approach presented here also has uses in the previously described field of affective computing. A vital requirement of any successful affective computing system is the ability to recognise human facial expression accurately, and the approach presented in this thesis clearly meets this criterion. Nonetheless, facial expression alone does not always provide an accurate read-out as to underlying emotional state. Furthermore, emotions are often felt without any obvious signs being shown on the face. Thus, in an affective computing system the expression recognition system could be combined with additional approaches that examine, for example, gesture [Psarrou 2002] and tone of voice [Fernandez 2000].

1.1 Thesis Overview

The work presented in this thesis has a number of original features. In summary, it:

- *introduces a novel face tracker that improves upon and extends the ratio template algorithm [Anderson & McOwan 2003]*
- *presents a fully automated system requiring no manual intervention (even in the training process)*
- *presents a system that runs in real-time*

- *provides new techniques for representing motion information for classification of expression*
- *presents a novel ratio-based approach to removing rigid head motion*
- *presents a new approach for determining regions of the face that are important in expression recognition*
- *gives examples of new applications for which automated expression recognition can be used*

The remaining chapters in this thesis cover the following topics. **Chapter 2** gives a brief introduction to the anatomy and psychology of facial expression of emotion. **Chapter 3** provides an extensive review of past research in the field of automated expression recognition. This is followed in **Chapters 4, 5, 6, 7 & 8** by a detailed description of the technology solution introduced here, giving detailed results and motivations for the computational approaches taken. **Chapter 9** summarises and discusses the prototype applications to which this system has been put to use, whilst **Chapter 10** discusses and summarises the work of this thesis, also describing avenues open for further investigation.

2 THE ANATOMY & PSYCHOLOGY OF FACIAL EXPRESSION

Before going into detail on automated facial expression, it is first useful to give some grounding in the psychology of facial expression and also introduce the facial muscles responsible for facial expression. This grounding is useful as, by understanding biology, it can help one build better technology that more closely fits human expectations. This chapter comprises several sections. **Section 2.1** provides a summary of the anatomy of the muscles involved in facial expression, describes how these muscles are controlled and introduces some conditions where processing or display of facial expression is aberrant. In **section 2.2**, arguments for and against the existence of universal expression of emotion in humans are then given, followed by a discussion of the link between expression, emotion and social interaction in **section 2.3**. Finally, the findings of this chapter are summarised in **section 2.4**. The bulk of this chapter is taken from the book, "As Others See Us" [Anderson & McOwan 2004].

2.1 The Anatomy of the Face

As humans we have an incredible potential for making different faces, with an orchestra of facial muscles producing the changes required to form our facial expressions. The facial muscles are highly variable in shape and strength and unusual in that they insert into skin and other muscles rather than into bones. By contracting and extending the facial muscles, we not only cause motion of the skin surface, but also cause the formation of various furrows and lines, such as crow's feet. **Figure 2.1** shows the location of the muscles most important in facial expression and **table 2.1** summarises the actions for which each is responsible.

2.1.1 Facial Muscle Control

The facial muscles receive motor impulses from the 7^{th} cranial nerve, a nerve that plays a role not only in facial expression but also in taste. This nerve begins in the pons (connecting the medulla to the midbrain, linking the upper and lower levels of the central nervous system) of the brain. It then enters the temporal bone before reaching the lateral aspect of the face, where it splits into five major branches that serve the whole face. These are the temporal; zygomatic; buccal; mandibular; and cervical

branches. As well as conveying motor impulses to the muscles of the face, the 7th cranial nerve also plays a role in advising the brain of our own facial movements. This it does by transmitting impulses from proprioreceptors (monitoring the degree of stretch of our muscles) back to the brain. A more detailed introduction to the nerves and muscles of facial expression has been written by Sinha [2001].

Figure 2.1 The major muscles responsible for the generation of facial expression. Approximate locations of each muscle indicated by an arrow

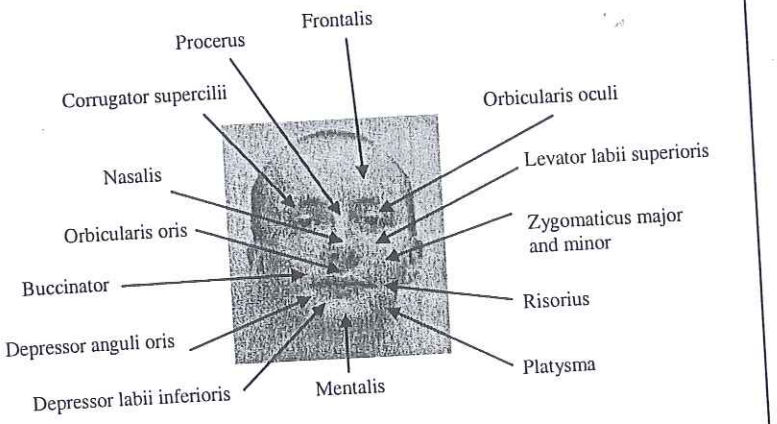


Table 2.1 Actions for which facial muscles are responsible

Muscle name	Position in face and role in expression
Buccinator	Draws corners of cheek laterally; compresses cheek
Corrugator supercilii	Pulls eyebrows together; wrinkles forehead skin
Depressor anguli oris	Draws corners of mouth down and laterally
Depressor labii inferioris	Pulls lower lip down
Frontalis	Raises the eyebrows; used for frowning
Levator labii superioris	Opens lips; flares nostril; raises upper lip
Mentalis	Elevates skin over chin; used for anger
Nasalis	Flares the nostrils
Platysma	Pulls lower lip back and down
Procerus	Bridge of nose, used in frowning
Risorius	Draws corners of lip laterally; tenses lips; used for grinning
Zygomaticus major & minor	Raises corners of mouth upwards; used for smiling and laughing

Whilst communicating face to face with others, we have to both control facial muscles to form meaningful expressions and also watch and interpret the facial displays of others. It is hard to imagine a world where the depth afforded by facial expression to our social exchanges is lacking, but a number of medical conditions lead to either aberrant processing or abnormal display of facial expression.

Those with autism and Turner syndrome have been shown to have deficiencies in processing the facial expressions of others [Critchley 2003, Lawrence et al 2003]. In the case of autism this abnormal processing of emotional cues is partly to blame for their extensive social and emotional problems [Critchley 2003]. Similarly this inability to process facial expressions is thought to be responsible for the difficulty that many people with Turner Syndrome have in forming and maintaining relationships [Lawrence et al, 2003]. In both cases the problems in expressional processing have been attributed to abnormal activity of the amygdalae. The amygdalae, so named due to their almond-like shape, are located just beneath the surface of the front, medial part of the temporal lobe under both brain hemispheres. They are a vital component of the limbic system that is thought to play an important role in the processing of emotional cues and generating emotional states, feelings and moods.

The alternate problem of being unable to express emotion has also been shown to effect peer relationships. Mobius Syndrome, or congenital facial diplegia, is a rare genetic disorder characterised by facial paralysis. This condition is caused by the lack or underdevelopment of the 6th and 7th cranial nerves that control eye movements and facial expression. Sufferers of this condition have reported extreme difficulty in developing and maintaining even casual relationships [Ekman, 1999]. Similar reports exist from those who have suffered a stroke resulting in partial paralysis of the face [Elks 1990], with this paralysis sometimes even being misinterpreted by others as mental deficiency [Hoos and Devriese, 1985]. The inability to display the interactional semaphore of facial expression in these cases can therefore lead to social exclusion and a diminished quality of life.

2.1.2 Measurement of Muscle Activity – The Facial Action Coding System

In 1978, Ekman and Friesen [1978] came up with an anatomically-based approach to describing the different movements of parts of the face, and they called it the Facial Action Coding System (FACS). By using underlying knowledge of facial anatomy and by studying videotapes and photographs showing how facial appearance alters with movement of the facial muscles, they came up with a set of 44 unique and visually distinguishable movements. These movements, called Action Units (AUs), correspond to each independent movement of the face [Bartlett 1999b], and allow one to accurately describe the appearance and dynamics of the human face.

Appendix 4 summarises the different AUs and the muscles underlying the movements. Each AU is assigned a numerical code, and range from clearly defined actions, such as AU1, the inner brow raiser, to more loosely defined actions, such as AU19, tongue out. The FACS coding procedure not only provides the AUs themselves, but also allows for the coding of facial action intensity (on a five-point scale) and the timing of facial actions. Facial expressions are described in terms of events, an AU-based description of a facial expression. These events consist of either a single AU acting alone, or a number of AUs acting together.

As FACS is purely descriptive it is of use in a broad range of fields. Therefore, not only is it of use in clinical [Ekman 1998a], developmental [Oster 1992], and social [Frank 1997] studies of emotion, but is also used in fields such as computer vision [Bartlett 1999] and neuroscience [Katsikitis 1988].

2.2 Universality of Facial Expression

The demonstration that facial expressions are common to all humans is considered important because, by showing the existence of universality, it proves that expressions are innate and have evolved over time and play a part in our evolutionary ability to survive, and are not simply learnt from social interaction during our lifetime [Ekman 1999]. Universality argues that aspects of human emotion are characteristic of the species as a whole.

2.2.1 The Case for Universality

The opinion that facial expressions themselves are universal has long been held, with the first detailed evidence provided by Charles Darwin in his 1872 book, "The Expression of the Emotions in Man and Animals" [Darwin 1872]. Darwin gave a questionnaire to 36 English observers who lived in, or travelled to, different parts of the world. The observers were asked to study the native people and answer questions such as '*is astonishment expressed by the eyes and mouth being opened wide, and by the eyebrows being raised?*'. Darwin concluded that races around the world expressed their state of mind in a like fashion.

Since then more comprehensive studies have been undertaken to ascertain whether facial expressions are truly universal. In general, these studies have involved the showing of photographs of faces with differing posed facial expressions to people from a range of races. The observers have then been asked to describe the emotion of the person within the image, often from a list of possible emotion words. Such work has provided a wealth of evidence for the existence of universal facial expressions but, despite this, many are still to be fully convinced, pointing to a range of perceived problems in the methodology of these investigations [Russell 1994].

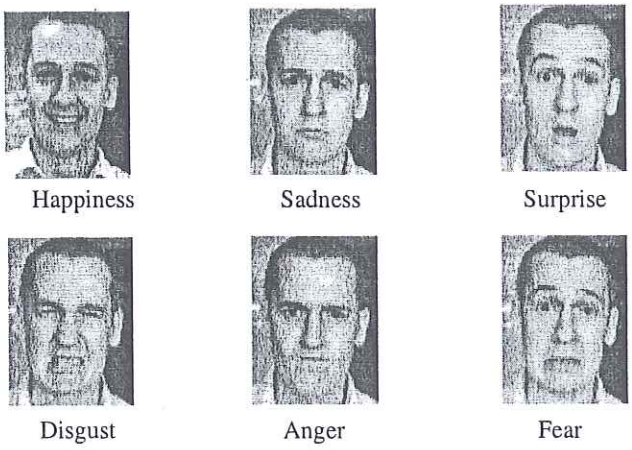
The highest profile advocate of universal facial expression is Paul Ekman. Perhaps his most significant contribution to the field of universality was his 1967 study of the South Fore people in Papua New Guinea [Ekman 1999]. Importantly, at that time these people were extremely isolated, having seen few or no outsiders, and were therefore unexposed to external cultural influences such as magazines, films and television. This work was instigated in direct response to the criticism of previous studies that people from around the world may have 'learned' facial expressions from each other or from other sources, such as media representations of emotion.

Ekman showed the South Fore people photographs of facial expressions and, rather than giving them a list of potential emotion words from which to select, asked them to make up a story that described the events that had caused each expression. This modification was necessary as the South Fore people had no written form of their

language. His results showed that even these isolated people chose the same expressions for each emotion as had people from the rest of the world, with the one exception that they failed to distinguish between fear and surprise. In addition, he then asked the South Fore people to demonstrate to him what their faces would look like if they were the protagonist in one of the stories. He recorded these expressions and showed the sequences to Americans who were able to determine what emotion they were expressing, although again fear and surprise were indistinguishable [Ekman 1999].

One outcome of such studies was the establishment of a number of 'basic' expressions universally associated with specific emotional situations. The six candidates generally accepted by psychologists working in the field of facial expression are happiness, surprise, sadness, fear, anger and disgust (posed examples of each are given in **figure 2.2**). It should be noted that, given the results of Ekman's New Guinea study, it is not absolutely clear there is a universal distinction between surprise and fear.

Fig 2.2 Posed examples of the six basic expressions



Further evidence for the existence of universals is provided by studies on primates showing that primates can read one another's facial expression [Miller 1971]. **Figure 2.3** summarises one such study that rather elegantly demonstrates the ability of primates to display and understand facial expressions.

Fig 2.3 Recognition of facial expression in rhesus monkeys [Miller 1971]

Step 1. Restrain rhesus monkey in chair. Give monkey electric shock to leg, but provide it with bar that if pressed stops the shock.
Result: Monkey learns to press bar upon receiving shock.

↓

Step 2. Repeat, but this time, several seconds prior to shock treatment turn on a light visible to monkey. If monkey presses bar before shock then shock avoided.
Result: Monkey learns to press bar when sees light, preventing shock.

↓

Step 3. Place two monkeys trained in this way in separate chairs. One is the 'stimulus' monkey and the other the 'responder'. Both monkeys are given the shocks. The 'stimulus' monkey can see the warning light but has no bar to press. The 'responder' monkey can see only the head and face of the 'stimulus' monkey, but is given a bar which, if pressed, prevents both monkeys receiving a shock.
Result: 'Responder' monkey learns to press the lever in response to reading facial expression of fear on 'stimulus' monkey.

↓

Conclusion: Rhesus monkeys can read and correctly interpret the facial expressions of other monkeys.

Before leaving the case for universality of facial expression, it is interesting to highlight the universality found in the sequence of development of facial expression in infants. There is some evidence that children of different cultures begin to show expressions of various emotions at similar ages [Ekman 1979,Ganchrow 1983], with this development summarised in **Table 2.2**. Importantly, this pattern of development is also seen in blind infants, providing vital evidence opposing the suggestion that these expressions are learnt from watching others [Charlesworth 1973]. The universality of this sequence implies a direct unlearned link between particular emotional states and particular facial expressions.

Table 2.2 Development of expression in infants

Period of development	Facial expression
new born	disgust, distress, interest
4-6 weeks	smile
3-4 months	anger
5-7 months	laughter, fear

2.2.2 The Case Against Universality

Many challenges to the theory of universality have been raised. One particularly prominent sceptic is James Russell. In 1994 he reviewed the studies of those such as Ekman's and argued against the soundness of the results and conclusions [Russell 1994]. He questioned the validity of the results obtained, claiming they were seriously affected by, amongst other factors, the method of forcing observers to choose from a selection of emotion words, and by the use of pre-selected posed (rather than natural) facial expression images. He also found that although people were good at attributing smiles to a feeling of happiness, it was not so clear-cut with other facial expressions, particularly in non-Western cultures. He suggested that agreement about the emotional state behind expressions other than a smile may not even exceed chance once methodological artefacts are removed. Ekman later responded to each of these criticisms in turn [Ekman 1999] but Russell remains unconvinced.

In 'The Psychology of Facial Expression' Russell [1997], in conjunction with Fernandez-Dols, came up with three propositions he thought vital to establish if one is to be certain of universality. These were as follows:

1. *The same patterns of facial movement occur in all human groups.*
2. *Observers in different societies attribute the same specific emotions to those universal facial patterns.*
3. *Those same facial patterns are, indeed, manifestations of those very emotions in all human societies.*

He argued that although considerable work has gone into demonstrating propositions 1 and 2, the third proposition has been largely ignored, and more work needs to go into proving this. He maintained that this is particularly important as proposition 3 may not be true even if proposition's 1 and 2 are properly established. He cites as evidence for this the work of Fernandez-Dols & Ruiz-Belda who, in a study of gold medal winning Olympic athletes, found that not all happy people smile, even if ecstatic [Fenandez-Dols 1995]. Instead these researchers concluded that smiles were

limited to social interactions. These conclusions bring us to the next section on the relationship between emotion, expression, and social interaction.

2.3 Expression, Emotion & Social Interaction

Another significant area of discussion in the field of facial expression is the fidelity of the link between facial appearance and the underlying emotional state. In the past it was fairly well accepted that facial expressions were key to understanding peoples' feelings and that facial expression was an involuntary readout of what a person was feeling internally. Recently, however, researchers have claimed that there is no simple direct connection between the expression on a person's face and how that person is feeling inside [Fridlund 1994].

From an evolutionary standpoint, Fridlund has argued that the only way facial displays could have evolved was by others paying attention and behaving appropriately in response to them, and thus the evolution of expressions required grounding in social interaction. Also, no one would have paid attention to these facial displays unless they provided information as to future actions, and that for expressions to have evolved, they must in general have provided information that was consistently advantageous to the expresser. He goes on to suggest that any display of surplus information that could have been damaging would have been repressed or eliminated. Thus, evolution put selective pressure on expressers to display signs readily detectable by others, and on receivers to develop effective ways of detecting these signals.

Fridlund also provides an example of how such co-evolution could have worked [Fridlund 1994]. He says that in the past, if a person intruded on another's territory then a fight would have ensued, possibly leading to the death or severe injury of one or other of the participants. However, if the invader had had advance warning that he was intruding then he could have chosen to retreat and both would have survived. For this to have happened interaction cues would have been necessary, with one person providing the cues, and the other receiving. He goes on to postulate that, should by chance the genotype in one individual have made it prone to bare its teeth just before attacking and a genotype in the other made it responsive to this signal and caused it to

retreat, then both would have survived the encounter, thereby increasing the likelihood that the signalling and receiving genes would spread. Such an event would have occurred not because the signaller wanted to display its feelings or the receiver to know how the signaller felt, but simply because of evolutionary pressure.

Thus, Fridlund concludes that rather than facial expressions being involuntary indicators of our internal state, they are in fact strategic social messages, tools used for negotiation in social encounters, mutually beneficial to both expresser and receiver. He writes that the nature and course of our social interactions have an acute affect on "the kinds of faces we emit, the circumstances under which we emit them, and the ways that we interpret them". In response to the concern of how facial expressions made in private can be tools for negotiation in social encounters, Fridund contends that these occurrences are in fact caused by imaginary instances of social interaction [Fridlund 1991].

2.3.1 Voluntary and Involuntary Expression

From Fridlund's position, as expressions are purely social, there is no need to distinguish between felt, involuntary expression and unfelt, false or deceitful expression. However, those, such as Ekman, who believe facial expressions are involuntary readouts of what we are feeling within, accept that many expressions are in fact fabricated, false expressions. Given the potentially disruptive effects to relationships and cohesive society posed by the ability of a signaller to deceive through false facial expression there has been considerable effort in attempting to characterise the difference between such unfelt displays and genuine facial expressions.

One difference between felt happiness and unfelt happiness has long been known, and was discovered in 1862 by a French Neurologist called Duchenne [Duchenne 1990]. He found that an unfelt smile solely involved the movement of muscles around the mouth, particularly the zygomatic major. However, a felt smile also included the action of the orbicularis oculi muscle, pulling the skin around the eyes towards the eyeball, sometimes leading to the formation of crow's feet around the eye.

It is virtually impossible to tighten the orbicularis oculi muscle intentionally, and conversely it is difficult to prevent tightening when exposed to something pleasurable.

Ekman and Friesen have proposed additional approaches to distinguishing between felt and unfelt happiness expressions [Ekman 1998a]. They claim that spontaneous expressions last between 0.5ms and 4 seconds, so anything outside this range can be considered unfelt. They have also found that when a happiness expression is forced, the action of the zygomatic major muscle is stronger on one side of the face than the other, causing asymmetry in the smile.

In addition to studying the difference between felt and unfelt expressions, Ekman has also discovered small, short, and involuntary changes in the face that give us away when, for example, we are lying. For only a fraction of a second our faces emit these flashes of truth, called microexpressions, giving vital clues as to our true feelings [Ekman 1998b]. Most casual observers don't notice them, but Ekman has been teaching law-enforcement agencies to pick up on them, looking for microexpressions that appear inconsistent with what a suspect is saying. For instance, a potential suspect could be pretending to be overcome with grief following the death of their spouse, but a fleeting expression of joy could give the game away, presenting the law enforcement agents with a potentially important indication that the suspect might be lying.

2.3.2 The Effects of Facial Expression on Emotional Experience & Speech

Interestingly, it has been shown that, as well as facial expressions reflecting our internal state, they themselves can contribute to our experience of emotions. This phenomenon, known as the facial feedback hypothesis, suggests that we receive feedback from our facial expression, thereby increasing the intensity of emotional experience. To test this theory, Kraut carried out a study where the participants were asked to evaluate various odours whilst either posing a smile or a frown [Kraut 1982]. He showed that subjects posing frowns perceived the smells as less pleasant than those posing smiles. To remove any possibility that subjects were influenced by the thought that the researchers wanted them to give happier responses whilst smiling, subsequent studies involved people holding pens in their mouth. By clasping the pen between the

teeth the subject's faces were forced into a smile without explicitly being asked to do so (**figure 2.4**). Results again confirmed the facial feedback hypothesis [Strack 1988].

Figure 2.4 How holding a pen in the mouth can cause activation of smiling muscles

Ingeniously, Zajonc et al [Zajonc 1989] demonstrated facial feedback by use of the phonemes "e", "ah", and the German "ü". When spoken, the phonemes "e" and "ah" activate smiling muscles, whilst "ü" forces the smiling muscles to extend rather than contract as they do when one smiles. The participants were asked to read aloud stories that were balanced in content and emotional tone, but some contained many words with the phoneme "ü", and others none. The participants were then asked to rate how much they liked each story and overall, subjects rated stories containing "ü" words lower than those without.

During this research, the temperature of the brain of each subject was recorded, allowing a mechanism explaining some of the effects of facial feedback to be formulated. It was shown that the temperature of the forehead was significantly increased in those saying "ü", and slightly decreased in those saying "e" and "ah". These changes in temperature were thought to modify the activity of neurochemicals within the brain, thereby altering our mood. The change in brain temperature occurs as, by contracting and extending our facial muscles, we modify the blood flow of

surrounding facial blood vessels, which in turn alters blood flow in the brain, causing a change in brain temperature.

Our facial expression also has acoustic consequences on the tone of our voice, as different facial expressions change the shape of, and thus the acoustic properties of, the mouth cavity. Tartter [Tartter 1980, Tartter 1994] recorded samples of subjects talking normally and then talking with a smile on their face. The subjects were specifically asked not to try to sound happy whilst doing this. Thus, Tartter generated data that differed solely in the facial expression posed whilst speaking, although this methodology has the weakness of instructing subjects to smile. These recordings were then presented to listeners who were asked to identify which example from each pair sounded happier. This, in general, the listeners were able to do.

2.4 Summary

This chapter has described the importance and complexity of the link between human emotion and facial expression, providing a background to consider when developing automated systems for response to the expressions of humans. The psychological research that led to the proposal of the six emotions universally associated with unique facial expressions has been introduced, and these expressions are the very same as those that the system described in this thesis attempts to recognise. The introduction to anatomy provided is also of importance for motion-based expression recognition systems as the movements seen during facial expression are constrained by the muscles that cause them.

3 SURVEY OF AUTOMATED EXPRESSION RECOGNITION

Over the past decade the problem of automated expression recognition has become an active area of research, with some solutions examining dynamic changes in image sequences and others solely using still frame images. Researchers have generally attempted to solve the problem by the use of various optical algorithms flow (see **section 5.1** for definition) [Mase 1991, Yakoob 1993, Rosenblum 1994, Lien 1998, Essa 1997] and to either classify expressions as one of the six key emotions universally associated with unique facial expressions [Yakoob 1993, Rosenblum 1994] or recognise some of the Action Units of the Facial Action Coding System [Bartlett 1999, Lien 1998]. Applications for such systems include sociable robots, improved HCI, automated detection of deceit, and animation of synthetic heads.

This chapter begins, in **section 3.1,** by summarising the problems associated with automated expression recognition and subsequently provides a detailed review of the important pieces of work that have been carried out in this field in recent years. **Section 3.2** gives a brief overview of the history of automated expression recognition. More recent approaches are then described in later sections. **Section 3.3** introduces optical flow-based, **section 3.4** model-based, **section 3.5** feature-based and **section 3.6** image-based approaches. **Section 3.7** then describes systems that combine a number of these techniques, whilst **sections 3.8 & 3.9** summarise this work and discusses the pros and cons of some of the systems. **Section 3.8** also provides a table summarising the performance of the approaches described in this chapter.

3.1 Problems Associated with Automated Expression Recognition

The main problems associated with automated expression recognition relate to localisation of the face, variation in illumination, variation in pose, change in facial features (such as facial hair) and removal of the effects of rigid head motion. Each of these problems will now be addressed in more detail.

- **Face localisation** – prior to expression analysis, it is first necessary to locate the face in the scene and, to prevent the need for manual intervention, this process needs to be automated. However, the problem is often ignored in expression

recognition systems in order to concentrate on the expression analysis side of the work. The difficulty of this task can vary depending on the expression recognition system. For example, some approaches may only need a rough estimate of the location of the face and its facial features, whilst others may need a precise location for all the different parts of the human face prior to expression analysis.

- **Illumination variation** – changes in lighting can have significant effects on the appearance of a facial expression. Variation may occur due to changes in lighting position (eg from above or below) and because of changes in lighting intensity. Illumination may also come from multiple sources rather than from a single location, causing lighting change [Fasel 2003].

- **Pose variation** – changes in pose can also have significant effects on the appearance of a facial expression, with changes possible in both the direction and distance from a subject's face. Systems generally only attempt to recognise expressions from frontal views eg [Black 1997].

- **Human variation** – different people possess different facial features, such as glasses or beards, that may alter facial appearance and occlude features that may be needed for expression analysis. Amongst the human population, there are also huge variations in face size and shape, as well as skin colour and texture, due to differing age, race, and gender [Fasel 2003].

- **Rigid head motion** – one difficulty that may be experienced with motion-based approaches is the separation of non-rigid head motion, caused by expression change, from rigid head motion, caused by movement of the head as a whole. In past work, this problem has been largely ignored and the assumption made that little or no rigid head motion is present eg [Yacoob 1993].

3.2 Early Attempts at Automated Expression Recognition

Arguably the first person to address the problem of automated expression recognition was Suwa in 1978 [Suwa 1978]. Suwa used twenty spots on the face and

tracked their movements in a video sequence. The motion of these spots was then compared with the movements from prototypical facial expression patterns to determine the expression of the face.

Several marker-based approaches have subsequently been developed [Himer et al 1991, Kaiser and Wehrle 1992]. Himer [1991] again used marker movement to classify facial expression, while Kasier and Wehrle [1992] used scale-normalised distances between a number of marker points to recognise not only AUs from the FACS, but also the AU intensities. Kaiser [1992] used neural networks for the classification task and found that performance was comparable to that of a human FACS coding expert. Unfortunately, although marker-based methods can give impressive results, it is necessary to accurately position a number of markers on subjects' faces, and thus their use is limited to carefully set-up experimental conditions

Obviously, for wider applicability it is more desirable to use less intrusive methods that do not require the application of markers to the face. The important early work of Mase & Pentland [1991] addressed this issue, translating 2-dimensional facial motion into a rough estimation of muscle activity. Unlike the works of Suwa [1978] and Himer [1991], the movement of markers was not used, but the motion of the face itself. Dense optical flow was used to estimate the activity of 12 of the 44 facial muscles. A window in the face was allocated for each muscle, with the movement, as determined by optical flow, in each window averaged and then compared to a pre-determined axis of motion along which each muscle expands and contracts. This allowed estimates to be made as to the activity of each muscle, and by using a simplified version of the FACS, it was possible to characterise these activities. A recognition rate of 86% is reported.

3.3 Optical Flow-based Methods

Subsequent to the work of Mase & Pentland [1991], a number of researchers have used optical flow, the calculation of motion fields, for the purposes of facial expression recognition, either on its own [Yacoob 1993, Rosenblum 1994, Black 1997], or in combination with other approaches [Bartlett 1999b, Tian 2001].

Yacoob and Davis [Yacoob 1993] used a correlation-based optical flow technique in combination with a rule-based approach to recognise the six basic emotions universally associated with unique expressions. The systems' rules described the actions of certain facial features when each specific emotion was expressed, so, for example, the actions required for surprise were the raising of the brow and the lowering of the bottom lip, followed by the brow being lowered and the bottom lip rising. The motion of key facial features (right eyebrow, left eyebrow and mouth) was elucidated using an optical flow algorithm. The system then examined the motion output of these features and compared it to the events required for each facial expression, labelling the input as one of the six basic expressions accordingly. The success rate for this approach ranged from 80% to 94% depending on the expression (eg 80% for sadness, 94% for surprise). Yacoob et al tested the system on their own database of 32 subjects expressing 105 emotions.

These researchers [Rosenblum 1994] then developed a more advanced system designed to recognise facial expressions using the same basic methods, but rather than using heuristic rules, they set up radial basis function (RBF) networks (see **appendix 6**) to estimate the stage of an expression from a facial motion description. Once the motion information was obtained, log polar transform was carried out to reduce the effect of scale changes, and the data was then entered into RBF networks. There were separate networks for each expression, with each being subdivided into layers for each facial feature used. The networks were then further divided to enhance sensitivity to the specific motion of the feature for the emotion for which it was set up. The training of the networks used sequences of frontal face images of different human subjects experiencing only the two basic emotions happiness and sadness. This technique was reported to have an 88% success rate at recognising the emotional state of familiar faces, and a 73% success rate at recognising the emotional state of unfamiliar faces.

Black & Yacoob [1997] later extended the work of Yacoob [1993] by using parameterised models for the mouth, eyes, and eyebrows to deal with large-scale head motions, separating rigid from non-rigid head motion. They described facial actions in terms of mid-level and high-level descriptions. The mid-level descriptions characterised

the motion of each feature, whilst the high-level descriptions examined the temporal changes in these mid-level descriptions to recognise expressions. Overall success rates of 93% are quoted for recognition of expressions of 40 subjects recorded by the researchers.

3.4 Model-based Approaches

Essa & Pentland [Essa 1997] also used optical flow estimation, but then coupled it with a model describing facial structure to observe and recognise facial expressions. Rather than using the original FACS, they developed their own representation that they called FACS+, a probabilistic rather than heuristic approach to coding the motion.

The facial model consisted of a mesh with 44 facial muscles, information on where these muscles attach to the skin, and the elastic properties of the skin. The optical flow results were then modified according to this physical model to obtain estimated forces. These estimated forces were then used to classify the facial expressions, and a 98% overall recognition rate was reported.

Unlike much of the other work described here, this system was fairly complete, with automated extraction of the position of the eyes, nose, and lips. Once these feature positions were determined, the face was warped to fit the face mesh, optical flow calculated, before finally estimates were made as to the forces acting. However, the tracking of the motion of the head was successful only if there was not excessive rigid head motion during facial expression.

3.5 Feature-based Approaches

Rather than using facial motion, some systems have used the shape and/or texture of different parts of the human face to recognise expressions in static face images. Lanitis et al [Lanitis 1997] set up a flexible face model to represent both shape and texture information by performing statistical analysis over a training set of face images. Their model described the mean shape of the faces and was able to represent differences caused by changes in facial expression. Interestingly, this work presented a unified system that could not only be used for expression recognition, but also person

identification, gender recognition, and 3D pose recovery. Upon presentation of a novel image to the system, facial features were located by an active shape model (ASM) search (see **appendix 6**) based on a flexible model created during training. Shape and texture information was then extracted at these feature points allowing for expression classification. A 74% success rate is reported from testing over 118 images created by Lanitis and co-workers.

Gargesha & Kuchi [2002] used facial feature information and artificial neural network classifiers to analyse the six basic expressions of static frontal view images. They first found the positions of the eyes, mouth, and eyebrows and then used an Active Contour Model (see **appendix 6**) that moved contour points to regions of each feature that contained edges, areas of maximal/minimal intensity and areas of maximal/minimal curvature. The geometric position of these contours, in addition to inter-feature distances and moments computed from the contour points, were then used to train artificial neural networks. A 73% accuracy rate is given for expression classification on images from the Japanese Female Facial Expression (JAFFE) database [Lyons 1998] using multi-layer perceptrons, and 65% for radial basis function networks. Although these accuracy rates are low, the researchers argue that their results are impressive as the system extracts facial feature positions automatically by use of morphological edge detection (see **appendix 6**).

Matsugu et al [2003] developed a system that automatically detected faces and then analysed smiles using a rule-based algorithm. The face detection system employed convolutional neural networks (see **appendix 6**) trained using back propagation to detect facial features and a rule-based module to process the information provided by the networks. Once a face was found, the expression recognition system used data provided by the face detector and a number of rules to analyse smiles. For example, the system examined cues such as the distance between the endpoint of the eye and the endpoint of the mouth, with a reduction in distance being seen to indicate the raising of the lip. Different weights were assigned to different cues according to their importance, and also modifications made according to the percentage change in distance. By combining the information provided by a number of cues a score was obtained that

could be thresholded to determine whether a smile was present in the image. A recognition rate of 97.6% for smiles was given on a test set of 5600 still images of around ten subjects smiling.

3.6 Image-based Methods

Image-based methods are used to extract features from images without using extensive knowledge regarding the structure of the face. An example of an image-based approach to expression recognition is the work of Padgett and Cottrell [Padgett 1998] who used Principal Components Analysis (PCA) (see **appendix 6**) in combination with neural networks trained using back propagation. They used the principal components from overlapping 32x32 pixel regions around the eyes and mouth of the human face. These were then used as inputs for neural networks.

Fasel & Lüttin [1999] used PCA and Independent Components Analysis (ICA) (see **appendix 6**) to recognise asymmetric AUs from the FACS, as well as determining the intensity of these AUs. First, the background was manually removed from the image to leave just the face, and then facial features were determined by subtraction of a neutral face image from a test image, thereby producing difference images. They then applied PCA and ICA and used nearest neighbour classification (obtaining the closest distance between projection of test image and the ICA/PCA components determined on reference images) to find the AU and intensity present in the test image. ICA was found to perform better than PCA, with an AU recognition rate of 83% achieved on a test set of images. The test images consisted of a single subject showing a single AU. Recognition rates fell to 74% when images of the face showing multiple AUs were included in the test set.

3.7 Systems using a Combination of Methods

Rather than just using, for example, optical flow or shape information, a number of systems have attempted to combine several approaches for the recognition of facial expression. By combining different methods, researchers hope that better results will be achieved than when individual methods are applied alone. Combining methods is

particularly effective if each individual method focuses on a facial feature different to that used by the other methods.

Work into automated recognition of AUs was carried out by Bartlett et al [Bartlett 1999b] who combined motion flow field estimation, PCA and facial feature measurement to produce a hybrid system for measuring upper facial actions. A summary of the three techniques described in this paper is given below:

- PCA - performed on difference images, with variation between each being caused solely by facial dynamics. A back-propagation network was used to classify facial actions given the principal components. An overall recognition rate of 88.6% was obtained when used on test set alone.
- Feature measurement – measured formation of wrinkles and alteration in appearance of sclera (at the outer corner of the eyes). An overall recognition rate of 57% was obtained when used on test set alone.
- Optical flow - used a gradient-based method for estimating optical flow and then classified the flow field by use of a template matching procedure. An overall recognition rate of 84.5% was obtained when used on test set alone.

By combining the results of these three techniques, an overall recognition rate of 92% was achieved for classifying the six upper face AUs. The system was tested on a database of 800 images, with AUs coded mainly by experienced FACS coders.

Lien et al [Lien 1998] developed a system using Hidden Markov Models (HMM) (see **appendix 6**) to automatically recognise AUs in the upper face. They used three different approaches to extract expression information. Facial feature point tracking was used to track eyebrow motion and involved manual marking of 8 facial feature points and tracking by an optical flow algorithm. For detecting motion in the forehead region, dense flow tracking was used to obtain pixel-wise flow. Additionally, line detection was used to take advantage of the formation of furrows and wrinkles during emotional change. HMMs were then used to evaluate the most likely AUs involved. The recognition rates ranged from 80 to 92% depending on which of the

above methods (feature tracking, optical flow, wrinkle formation) were used and the AU involved. The system was tested on the CMU-Pittsburgh AU-Coded Facial Expression Image Database (see **appendix 3**).

Tian et al [Tian 2001] used back propagation neural networks to recognise AUs by tracking and modelling a range of facial features. The system first located an individual's facial features, and then extracted information relating to the movement and shape of the eyes, lips, cheeks, and brows, regarding the formation of furrows in the nose and eye region, and also the distance between the brows. The neural networks were trained to respond to an individual AU, allowing the system to recognise these AUs if they occurred alone or in combination. AU recognition rates of around 96% were reported using the CMU-Pittsburgh AU-Coded Facial Expression Image Database and the database used by Bartlett [1999b].

3.8 Comparison of Facial Expression Approaches

The figures quoted above for the different expression recognition systems were obtained from different facial expression databases and differing quantities of data. Thus direct comparison is not possible. However, it is still thought advantageous to summarise the performance of the different systems to allow some comparison. **Table 3.1** summarises the performance of the different systems where performance figures are provided by the researchers, with the top half of the table describing systems that recognise AUs from the FACS, and the bottom half those systems that recognise some or all or the six emotions with unique facial expressions.

Recognition rates vary from 70% to 98% depending on the approach taken, with different systems having their own strengths and weaknesses. For example, the system giving the best performance for recognising basic emotions is the 3D face model of Essa & Pentland [1997] (NB remember test sets differ between different methods so direct comparison is not accurate), but the system has the disadvantage that 3D face models have heavy computational requirements. Other systems, such as the active shape model system developed by Lanitis [1997] and Lien [1997], require large amounts of time to manually label the positions of features precisely, whilst some optical flow

methods assume there is little or no rigid head motion between frames of the sequence [Yacoob 1993]. Combining a number of different approaches can be an effective way of improving recognition results [Bartlett 1999b], but obviously leads to an increased computational load.

Table 3.1 - Performance summary of different facial expression recognition systems

Author	Number of AU's	Number of expressions	Extraction method	Classification method	Test set size	Recognition Rate (%)
Lien [1998]	3	N/A	Feature point tracking	HMM	-	85
			Optical flow	PCA & HMM	-	93
			Line detection	HMM	-	85
Fasel [1999]	9	N/A	Difference images	ICA and Euclidean distance	45	83
Tian [2001]	16	N/A	Facial component model, canny edge detector	NN	50 seq	96
Bartlett [1999b]	6	N/A	PCA, optical flow, feature measurement	Template matching, NN	800	92
Yacoob [1993]	N/A	6 (all basic emotions)	Optical flow	Heuristic rules	105	80-95
Rosenblum [1994]	N/A	2 (hap. & sad)	Optical flow	RBF NN	34 seq	88
Essa [1997]	N/A	6 (all basic emotions)	Optical flow and 3D face model	Motion template	8 seq	98
Lanitis [1997]	N/A	7 (all basic emotions & neutral)	Appearance model	Mahalanobis distance	300	74
Black [1997]	N/A	6 (all basic emotions)	Motion model	Rule-based	40 seq	93
Gargesha [2002]	N/A	6 (all basic emotions)	Active contour model	NN	10	73
Matsugu [2003]	N/A	1 (happiness)	Convolutional NN	Heuristic rules	5600	97.6

3.9 Summary

To summarise, a range of approaches for automated expression recognition have been attempted in the past with differing degrees of success. However, many assume there is little or no rigid head motion between frames and require manual intervention at crucial stages of the process (eg for facial feature tracking). Other issues not addressed in most of the above work are the problems of pose change, with most assuming frontal views, and the variation seen in expression intensities. Some methods are also extremely expensive computationally, although unfortunately very few of the works reviewed here provide any figures indicating how long the approaches take to process and thus comparison is difficult. The work presented in this thesis requires no manual

intervention at any stage, can handle limited rigid head motion, and can be processed in real-time using current computer systems.

4 FACE TRACKING

Before it is possible to examine the facial expressions of a person, it is first necessary to locate the face in the scene and to follow it as it moves around that scene. This chapter introduces the single face tracker used by the expression recognition system for this purpose. This tracker finds initial face locations using a modified version of the ratio template algorithm [Scassellati 1998], whilst subsequent processing stages reject any false positives. These later stages include a novel ratio-ratios operator that improves recognition rates by examining higher order relationships within the initial ratio template measures, and rapid, simple morphological eye and mouth feature detection. The data from these later stages is fused to allocate each potential face location a face probability score.

The chapter begins in **section 4.1** with a brief review of other face trackers previously cited in the literature. The modified ratio template approach [Scassellati 1998] is then discussed (**section 4.2**), followed in **section 4.3** by a description of how this is integrated with other techniques to form a single face tracker. The characteristics of the completed face tracker are given in **section 4.4**, and examples of images from a tracked sequence in **section 4.5**. **Section 4.6** then describes how the speed of the basic approach is improved for incorporation into the expression recognition system, and a summary given in **section 4.7**.

4.1 Review of Face Detection

The ability of a computer system to accurately and robustly locate a human face in a natural environment is essential for many applications. In addition to recognition of facial expression it is required for tasks such as face recognition and interactive teleconferencing. Previous attempts at solving this problem have either used single frames or whole sequences as input, with these images being either in colour or grey-scale. Attempts to locate faces have been made by examining a range of facial features, such as skin colour and texture, and may involve the use of the whole face or just specific features such as the eyes. This section gives a brief introduction to some of these approaches, with more detailed reviews being provided by Samal et al [1992] and Hjelmas et al [2000].

A number of studies examining face localisation have involved the use of PCA (see **appendix 6**). For example, Mogghaddam and Pentland [Moghaddam 1995] developed a general system for recognising a range of objects, including faces. Separate feature templates for the eyes, mouth and nose were used for training and a 97% success rate is reported for detecting faces.

Colour processing has also been used as a means of detecting faces. Recently this method has been combined with geometrical approaches by Wang and Sung as a means of extracting facial features [Wang 1999]. Skin-colour face segmentation is used here to detect the facial regions. This process involves searching for skin coloured pixels and then grouping them together with the largest area of skin coloured pixels being classified as a face. By solely searching for the largest region of colour, this approach is independent of both scale and viewpoint.

Lam & Li [Lam 1998] proposed the use of eye detection as an effective means of detecting faces from still frontal face images. They firstly used morphological operators (dilation and erosion) (see **appendix 6**) to obtain the valley field of an image. This is an effective method as there is a low intensity region around the eyes. They then found the boundaries of the eyes and searched the resultant image for line intersections meeting their rules for eye corners (distance between, height differences etc) as described in an earlier paper by Lam [Lam 1996]. Any pairs of intersections meeting the required conditions were classified as eyes. A verification procedure is then carried out by use of an eye template.

Kawato and Ohya attempted to use the between-eyes region for detection [Kawato 2000]. They passed a circular filter over images, with this filter maximising its output when positioned directly between the eyes. This relies on the property that, by drawing a circle of the correct radius centred between the eyes, two cycles of bright and dark pixel values are obtained due to the presence of the forehead and nose bridge (bright) and the eyes and brows (dark). However, this procedure cannot work alone and requires further processing to remove the high number of false positives.

Raducanu et al [Raducanu 2001] used morphological multiscale fingerprints (MMF) for face localisation, with these fingerprints preserving the local extremes in the image. They use morphological operators (erosion and dilation) (see **appendix 6**) on images to determine the distance between face-like parts of the image and a mean face pattern. They subsequently compared their new approach with a classic PCA face detection approach and, using ROC curves (see **section 6.6** for description), demonstrated that for all equivalent face detection rates the MMF approach gives fewer false positives than PCA.

Wang and Tan [Wang 2000] used shape information. Pre-processing of the image, using a histogram equalisation, is first carried out to improve contrast and remove noise. Edge detection and linking are subsequently used to produce a binary image. Finally, a deformable elliptical ring template, based on edge information, is applied for matching to the face contour.

Neural networks trained using back propagation have also been used in this field by Rowley et al [Rowley 1998] to detect faces viewed frontally, with the networks using three types of hidden node, each receptive to features of different size. Rather than using a single network, Rowley et al use multiple networks, each trained in a similar manner, and then arbitration between these networks to reject false positives. Although each network tends to detect actual faces at the same place, the small differences in training procedure lead to the networks making different errors, thereby effectively making it possible to distinguish between true matches and false positives. The basic approach takes 383 seconds to process a 320x240 pixel image on a 200MHz machine, although speed-ups are suggested that significantly enhance speed to around 3 seconds on a 200MHz machine.

Yang et al [Yang et al 2000] utilise the SnoW (Sparse Network of Winnows) learning architecture (see **appendix 6**) to detect faces independently of pose (by training with face databases containing large variations in pose). The SnoW architecture is tailored to learn in domains with a large number of potentially unknown

features, and the authors report improved detection rates (94.1%) when compared to approaches using neural networks (90.3) and support vector machines (74.2%).

The face detector of Viola et al [Viola and Jones 2001] uses classifiers in a cascade structure. By combining successively more complex classifiers they rapidly focus attention on regions of interest, thereby achieving much higher frame rates than those obtained by Rowley et al. Efficient classification is also made possible by using an AdaBoost-based learning algorithm (see **appendix 6**) to select a small set of critical features for producing classifiers. A speed 15 times faster than the Rowley detector [Rowley et al 1998] is quoted.

4.2 Ratio Template Algorithm

The tracker used here is based on the ratio template algorithm used in the cognitive robotics Kismet project at MIT [Brazeal 1999]. This project has developed an expressive humanoid robot head that responds to social cues, allowing it to interact face to face with people in a natural and expressive manner [Brazeal 2003]. The robot is able to modify its gaze, the orientation of its head, and also move its facial features (eyelids, eyebrows, lips, and ears) in response to input from 4 cameras situated around the eyes and 2 microphones, one mounted in each ear.

In addition to receiving this input from the outside world, Kismet is also given motivations (drive and emotion) that determine both what action it takes, and when it does so. Kismet's drive and emotion systems are both inspired by theories as to how such motivations work in humans in the hope that Kismet's responses approximately mirror those of a human, making interaction with it seem natural and plausible.

Kismet's 'drives' are to engage with people, engage with toys, and occasionally rest, whilst it is provided with a range of emotional responses, such as disgust and sadness. The drives of Kismet provide a long-term measure of the well being of the robot, whilst the emotion system works on a more rapid time scale. For example, if an undesired stimulus persists in Kismet's surroundings, its emotional response is to become disgusted. This disgust manifests itself as an expression on the robot's face,

allowing those with whom it is interacting to respond appropriately. The emotion system is also linked to a behavioural system that provides Kismet with strategies for carrying out specific tasks such as approach, search, and avoidance. So, in the case of Kismet becoming fearful, as well as reflecting this fear on its face, its avoidance behaviours become engaged, and Kismet moves away from any potentially damaging stimulus.

Obviously, one of the key skills of a social robot is an ability to identify human faces in its surrounding environment, and the ratio template algorithm is used for this purpose. The ratio template algorithm, originally described by Sinha [Sinha 1995], is able to detect frontal views of faces under a range of lighting conditions, although it is ineffective when the subject is illuminated from beneath (interestingly humans are also less able to recognise faces illuminated in this way [Liu 1999]). The method is able to handle limited changes in scale, yaw, pitch, and tilt of the head. The approach's strengths lie in its ease of implementation, its speed and its tolerance to the different illuminations to be expected in an unstructured indoor environment. In addition to these strengths, the main reason for choosing the ratio template algorithm for use in the expression recognition system presented here is that as it uses a spatial face template it provides a rough spatial map of facial features. This is vital for an expression recognition system as it is crucial to know where different facial features are positioned if one is to extract meaningful information as to the motion of these features.

A ratio template consists of a number of regions and a corresponding set of relations between the luminance of these regions. The original template is shown diagrammatically in **figure 4.1a**. Every location in the image is overlaid with the template and the greyscale values of the pixels within each region are averaged. A relation is satisfied if the ratio of the average greyscale value from the first (arrow tail in **figure 4.1a**) to the second (arrow head) exceeds a threshold of 1.1 [Scassellati 1998]. Two types of relation exist, essential (indicated by solid arrows) and confirming (dashed arrows). If at least 10 of the 11 essential and 8 of the 12 confirming relations are satisfied then the location is regarded as a face [Scassellati 1998]. The approach succeeds as certain regions of the face (eg eye) are always darker than others (eg tip of

nose) when illuminated from above. The taking of ratios between these spatial regions gives robustness to changes in overall scene illumination.

4.2.1 Modifications to Ratio Template Algorithm

The version of the ratio template algorithm as implemented by Scassellati [Scassellati 1998] has been modified and extended in two ways. Firstly, shapes of regions in the face template are altered. Secondly, a novel stage is included where ratios of those ratios obtained in the ratio template algorithm are examined to improve selectivity. In the sections that follow these enhancements are discussed and results presented.

4.2.1.1 Golden Ratio Template

The ratio template presented by Scassellati [Scassellati 1998] is apparently hand coded using empirical knowledge of the appearance of faces illuminated from above. A more sensible approach would seem to be to use our knowledge of nature and facial structure to determine the shape of the template. One method of accomplishing this is to alter the template so that its proportions conform more closely to the golden ratio.

The golden ratio, also known as ϕ, has a value of 1.618. It is claimed that the human mind is instinctively attracted to this proportion and the ratio appears regularly in the fields of art, architecture, and music [Green 1995]. However, of more interest is its frequent presence in a range of living organisms. For example, a golden spiral is present in the shell of the nautilus, where it is thought to occur as a direct result of simple growth routines, as well as in the arrangement of seeds in pinecones, where its use optimally packs the seeds. Bodily proportions corresponding to the golden ratio can be seen in the gazelle, some fish, butterflies and moths (where it can also be seen in wing patterns). Proportions corresponding to the golden ratio also occur in many parts of the human body and face [Ghyka 1978]. Examples of ϕ occurring in the human face include:

- face width:head length
- nose bridge-nose tip:nose tip–chin

- nose tip-mouth:mouth-chin
- distance between eye pupils:nose width
- nose bridge-mouth:mouth-chin
- distance between eye pupils:distance between inside of eyes

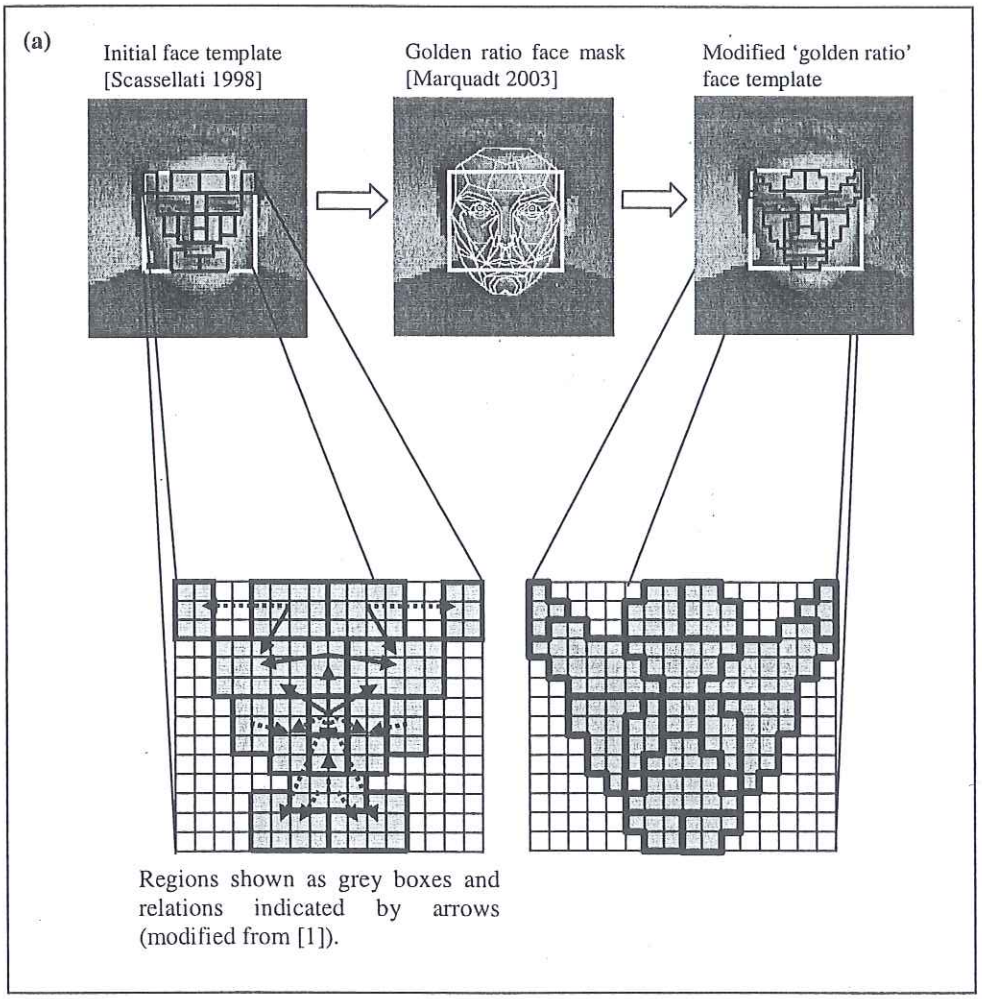
This list is by no means exhaustive, with the golden ratio appearing, not only in frontal views of the face, but also when it is viewed in profile. It can even be found in human teeth [Abdullah 2002]. Therefore, intuitively it seems likely that incorporation of golden ratio proportions into the ratio template algorithm's spatial template so as to make the template correspond to an 'average' human face would enhance the technique.

To help incorporate the golden ratio, a facial mask constructed from golden decagon matrices for use in cosmetic plastic surgery was studied. This mask was constructed using a primary golden decagon matrix to define the facial borders and then a further 42 matrices, differing in size from one another by multiples of ϕ, were positioned at various positions within this matrix in order to form the facial features. The construction of the mask is thus based on ϕ and a short sequence demonstrating its construction is available at [Marquardt 2003]. In addition to its use as a target structure for use in plastic surgery and dentistry to make people appear arguably more attractive, Marquardt also proposes applications for the mask in psychology, where it could be used to understand our perceptions of physical attractiveness, and anthropology, where it could be used in understanding the evolution of how we perceive "humanness" in the human face.

To incorporate the golden ratio proportions into the ratio template algorithm's face template, the Marquardt golden ratio facemask was overlaid by an identically scaled copy of the Scassellati template. Changes were then made to the shape of regions within the template to more closely match those of the Marquardt facemask, thereby incorporating golden ratio proportions. The facial mask and the modified face template are shown in **figure 4.1a**. The number, arrangement and relations between regions are maintained, but the shapes of regions are modified significantly. Rather

than simple rectangular blocks, the regions more closely describe actual facial structure according to shapes within Marquardt's golden ratio mask.

Fig 4.1 (a) From Scassellati template to 'golden ratio' template (b) Exemplar images showing faces correctly identified by 'golden ratio' template that remain undetected or incorrectly detected by Scassellati template



4.2.1.2 'Golden Ratio' Template Results

The modified face template was tested on 359 frontal face images taken from 4 different face databases [Martinez 1998, Georghiades 2001, Belhumeur 1997, Hond 1997], and its efficacy compared to that of the original ratio template. Please note that throughout this work monochrome images are used. The test set contains 137 subjects of differing sex, race, and age, with each face illuminated from a number of different positions. The results of the original vs 'golden ratio' template comparison are tabulated according to lighting position in **table 4.1**, and exemplar images shown in **figure 4.1b**. The original and modified templates correctly identify virtually all faces lit from above and behind. When faces are lit from the side or from the front, however, the 'golden ratio' template fares much better than the original template. The percentage of correct identifications using the 'golden ratio' template is 23% greater with side lighting, and 24% greater with illumination from in front of the subject. The matching rates are fairly low (eg Scassellati template identifies only 22% of side lit images) as the illumination conditions are at the extreme end of those under which the ratio template model can effectively work. Also, all images from each of the different databases were transformed to a single size for this analysis, so it is possible that some faces in the test set were outside the size range under which the template is able to match. It is expected that if the size of these images were scaled appropriately the algorithm would detect these faces.

Table 4.1 Results achieved using Scassellati and 'golden ratio' face templates under different lighting conditions

Lighting Position	Percentage faces correctly located	
	Scassellati face template	'Golden ratio' template
Behind	94%	93%
Front	29%	53%
Side	22%	45%

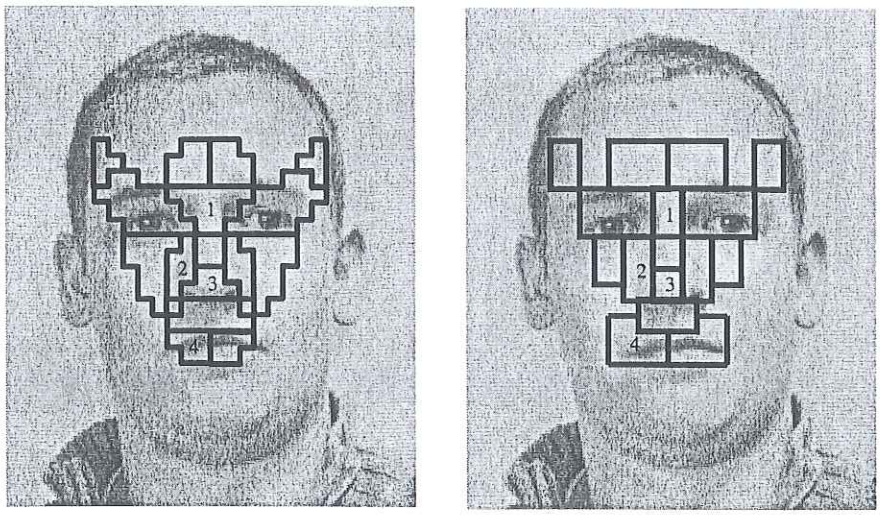
The results demonstrate that the 'golden ratio' template improves tolerance to different illuminations. It is thought that this is because the modified 'golden ratio' template more accurately describes the structure of the human face, and hence how light reflects from the surfaces forming it. Although regions of the original template correspond to identifiable regions of the face such as the eye, nose tip, nose bridge etc, because only rectangles are used they do not accurately resemble the shape. The 'golden ratio' template addresses this shortcoming. For example, compare the regions

labelled 1-4 in the two face templates shown in **figure 4.2**. By inspection, it is evident that the parts of the face to which the template regions map are not simply rectangular. The lips are fuller in the middle, the bridge of the nose branches out at the top and there are nostrils at the bottom of the nose. The modified face template more accurately models the shape of these features than the original face template, and this is reflected in the improved performance. Overall, the original face template correctly locates 47% of faces in the test set, whilst the modified template locates significantly more at 62%. (Chi-squared tests show the probability of the result if both templates were equally effective at locating faces as being <0.001).

4.2.1.3 Higher Order Information: Ratio-Ratios

In a cluttered environment, the ratio template model can still pick up a number of false positives that need to be rejected. An effective way we have found for determining which of the matches is correct is to take ratios of those ratios already calculated by the ratio template algorithm.

Figure 4.2 Improved modelling of human face structure by modified 'golden ratio' template



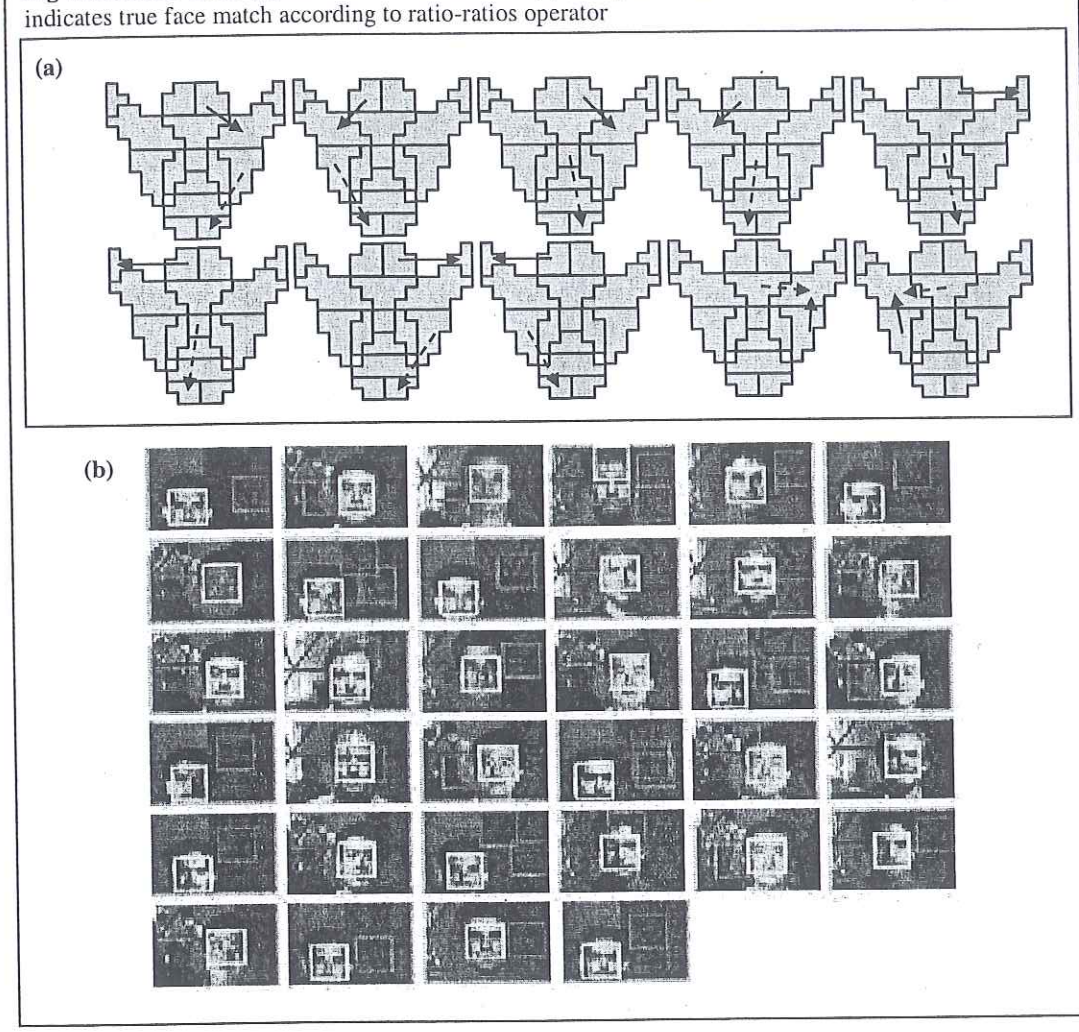
The original ratio template algorithm takes ratios of averaged luminance and then either labels the ratios as matches or not, according to simple thresholds. However, the absolute values of the ratios themselves contain useful information that

can be used to reject false positives. This approach is particularly appealing for a real-time system, as the original ratios are already computed, so there is minimal overhead in taking further ratios of these values.

To determine which characteristic set of higher order ratios to use, two sets of calibration images were produced, one of face and the other of non-face images (consisting of over 100 images). The ratio template algorithm was applied to these images, and where faces were located according to the ratio template algorithm, the absolute values of each ratio were recorded. Each individual ratio was subsequently averaged for the face and non-face sets separately.

Comparison between these two sets of averaged ratios made it possible to pick out differences between face and non-face matches. For example, the average value of the left eye:left forehead ratio in the face set was found to be significantly higher than the value of the left cheek:left chin ratio. This was found not to hold for the averaged ratio values obtained from the non-face set. Thus, the taking of a ratio to contrast the left eye:left forehead ratio and left cheek:left chin ratio provides useful information as to whether a match is a true face or just a false positive. A set of 10 ratios of ratios was determined based on empirical examination, with a ratio of ratios relation being satisfied if the ratio of the first ratio selected to that of the second exceeded 1. These ratios of ratios are shown in **figure 4.3a**.

Fig 4.3 Ratio of ratios (a) the ratios of ratios taken (b)exemplar results from test set, white square indicates true face match according to ratio-ratios operator



4.2.1.4 Ratio-Ratio Results

This approach was tested on 110 face images taken from [Martinez 1998, Georghiades 2001, Hond 1997, Turk 1991, Hancock], none of which were used in the initial calibration process. Once again subjects of various sex, race, and age were used and pictures were taken under a range of illumination conditions. The test set was limited to 110 as the test required one or more false ratio template model positives in each image. This was not the case in the majority of database images (NB all images with false positives were included in the testing procedure). There is a mean of 1.46 false positives per image, and the location achieving the highest number of valid ratio-

ratios in each image is declared a face. Incorporation of this additional test on the ratios allows correct distinction between correct and incorrect face matches in 83% of images tested. Exemplar images from the test set are in **figure 4.3b**.

4.3 Improving System Robustness Using Data Fusion

This section describes additional measures used by the face tracker to help distinguish between true and false face matches. By taking multiple measures the robustness of the system is improved.

4.3.1 Morphological Eye Filtering

Once the spatial template has identified the best candidate locations in the scene, we can use the facial model to look for further corroborative evidence of a face. Thus, the next stage of the face tracking system employs some basic eye and mouth feature detection at the appropriate spatial locations within the template. The approach taken here is a simplified version of the eye detection procedure developed by Lam & Li [Lam 1998] who use morphological operators to find potential eye locations. The image is dilated, eroded, and the result subtracted by the original image. This approach finds valley fields in the image, and hence the eyes and mouth, as these regions are of a low intensity.

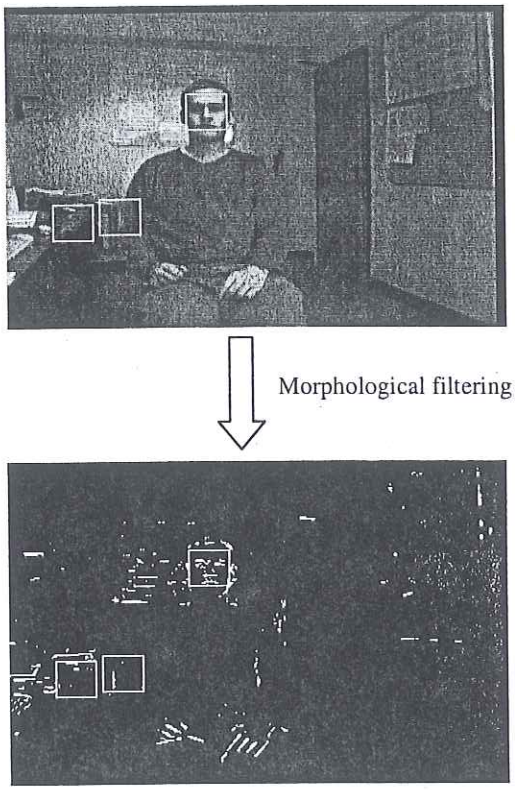
Lam & Li [Lam 1998] then search for eye corners in the valley fields, but this technique is not used here. Rather, the subtracted image is binarised and a search made for blobs whose centre of gravity is located where the eyes and mouth should be, and whose size lies within the bounds expected for an eye or mouth. The ratio template algorithm provides a rough spatial map of feature locations and the eye/mouth search occurs at all places in the image where the ratio template algorithm determines the presence of a face. The process is demonstrated in **figure 4.4**.

4.3.2 Matching Density

Upon identifying an actual face in the image, the ratio-template algorithm tends to give a number of local positive matches in the face's position. This is because positioning of the ratio-template need not be exactly centred on the face. It was found

empirically that false positives are more likely to produce only a single local match. Thus, positions with low numbers of matches in their local area are less likely to be a true face and we use this heuristic to reduce our belief that these locations are genuine faces.

Fig 4.4 Morphological filtering



4.3.3 Combining System Measures

The above techniques are combined as a cascade into a hybrid face tracking architecture, and by combining these stages it is possible to accurately determine where the face is located in the image. Given all results from the different stages a decision must be made as to which is the actual face in the image. To help make an informed decision, results were collated for the ratio-ratios operator, matching density, and eye/mouth detection for a range of faces under different illumination conditions. This

collection of data made it possible to determine the probability that a specific combination of these measures is in fact a genuine face.

Figure 4.5 shows the results obtained from the test set for face and non-face matches. **Figure 4.5a** presents the ratio-ratios results, and shows non-faces often achieve only 2 or 3 ratio-ratio matches, while genuine faces achieve 8 or 9. **Figure 4.5b** shows that over 80% of non-faces are determined by the morphological eye/mouth detection to have no eyes or mouth, compared to just 11% of actual faces used. The matching density results (**figure 4.5c**) demonstrate the increased number of matches obtained for faces when compared to non-faces. Independently, each measure can give a reasonable indication as to whether a match is correct, but in combination, the approaches become a powerful tool for face identification.

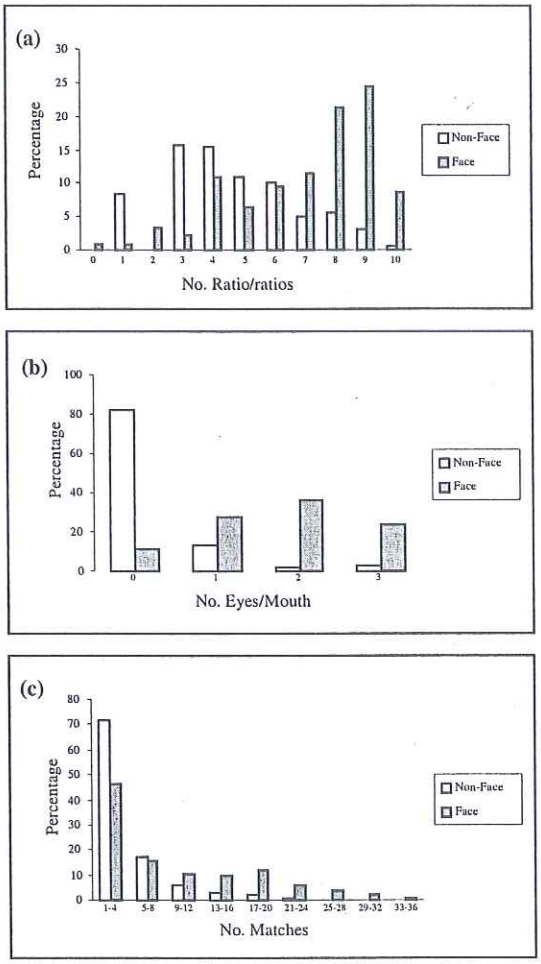
The full system operates on a live video feed. The ratio template algorithm is used to determine an initial set of potential face locations. Each location is then allocated a face probability according to the ratio-ratios, heuristics, and morphological eye filtering results. These probabilities have been determined by running the system on face and non-face image sets and using the results to calculate the probability that any single ratio-ratios, heuristics, and eye detection combination is a true face. These probabilities are given in **appendix 7**.

The system then makes use of several other techniques to improve robustness. We would expect that a face would normally exhibit some feature motion, so we determine pixel change between video frames at each candidate face location, and if this change is above the background level, we increase the face probability for that location.

Inertia is introduced into the system and examines the spatio-temporal continuity of faces in the scene, providing extra stability once a face has been identified. Inertia operates by modifying the face probability of each potential face location according to the face match history of that position. Checks are made to establish the proportion of past frames that the face tracker has labelled each potential a

face. Locations that have frequently contained a face have their face probability increased by the inertia system.

Fig 4.5 Results of (a) ratio-ratios operator, (b) matching density, & (c) eye/mouth detection when used on face & non-face ratio template algorithmic matches



Both the motion and inertia weightings were determined empirically, but are easily modified according to the environment in which the tracker is being used. For example, in a highly dynamic scene it would be desirable to lower the motion and inertia weightings as motion would be a less significant indicator of facial location, and faces would change position more frequently.

Face probabilities are averaged over recent frames to give extra continuity, and some occlusion handling is included. Face probability averaging simply averages the newly determined face probability at a location with the previous face probability of that position. The occlusion handling takes the form of maintaining a match for a couple of frames even if the ratio template algorithm no longer finds a face at that location. This lost match could be due to brief occlusion, and the occlusion handling means that if a match is lost, the history (inertia, average face score) of that match is not wiped. All stages are combined as a cascade and at the bottom of the cascade each candidate face location has an individual face probability value associated with it. A 'winner takes all' strategy determines the location with the highest value to be the face. The final face position is then labelled as the average location of the matches in that local area (remember that the ratio template algorithm tends to give a number of local positive matches in the face location). This averaging is useful as it helps find more accurately the true face position, something that is important for the expression recognition system to which it is connected. The hybrid face tracker is summarised diagrammatically in **figure 4.6**.

Parallels can be drawn between the stages of the face tracker and biological face detection systems. It is known that there are cells in the human cortex that fire specifically when a face in a particular pose is present in their receptive field [Tovee 1996], with the ratio template performing the same function. Also, the advantages of, and biological plausibility for, the taking of luminance ratios have been previously identified [McOwan 1999]. The alterations made to the ratio template are inspired by the golden ratio, while the use of eye/mouth detection is supported by studies into human face detection strategies [Tovee 1996, Hietanen 1999]. Face detecting cells respond to the relative positions of features within a face and any change to this arrangement of facial features reduces the cells' response [Tovee 1996]. There have also been suggestions that humans possess specialised detectors for eyes [Hietanen 1999]. The morphological eye/mouth detection mimics these effects by checking for the presence of eyes, and checking they and the mouth are in the correct relative positions.

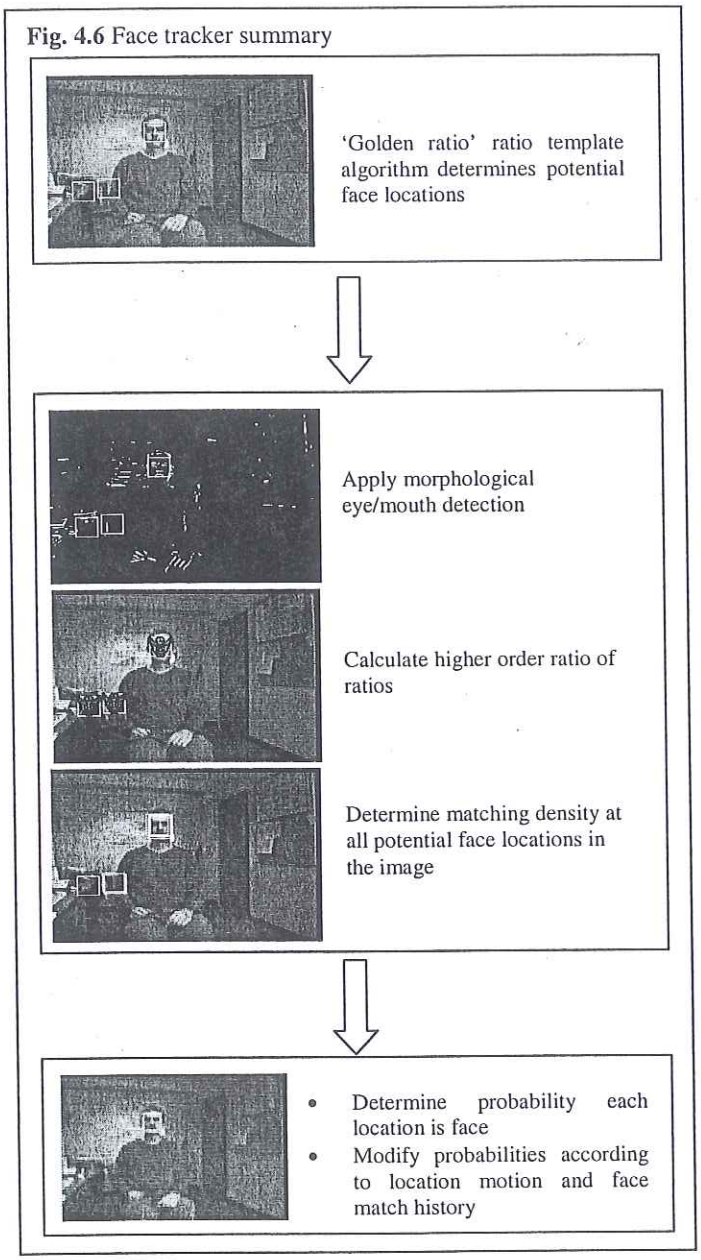


Fig. 4.6 Face tracker summary

4.4 Characterising Face Tracker

A number of experiments were carried out to characterise the conditions under which the face tracker can operate effectively. Several experiments were conducted to ascertain the effects on detection rates of changing illumination condition, head scale and head pose. Correct detection is deemed to have occurred if the tracker says the

centre of the face is located anywhere within 3 pixels of the centre of the face as labelled manually.

4.4.1 Illumination

To demonstrate the ratio template algorithm's ability to handle a range of different lighting conditions, the approach was tested on images taken from the Yale Face Database [Georghiades 2001], which contains frontal view images of 10 different people taken under 64 different illumination conditions. The images from the database were reduced to a size optimal for the ratio template algorithm and results obtained for the approach when exposed to each different illumination condition. A contour map indicating the different recognition rates under differing illumination conditions is given in **figure 4.7**. A positive azimuth indicates illumination to the right of the subject and a negative one to the left, whilst a positive elevation indicates above the horizon, and negative below.

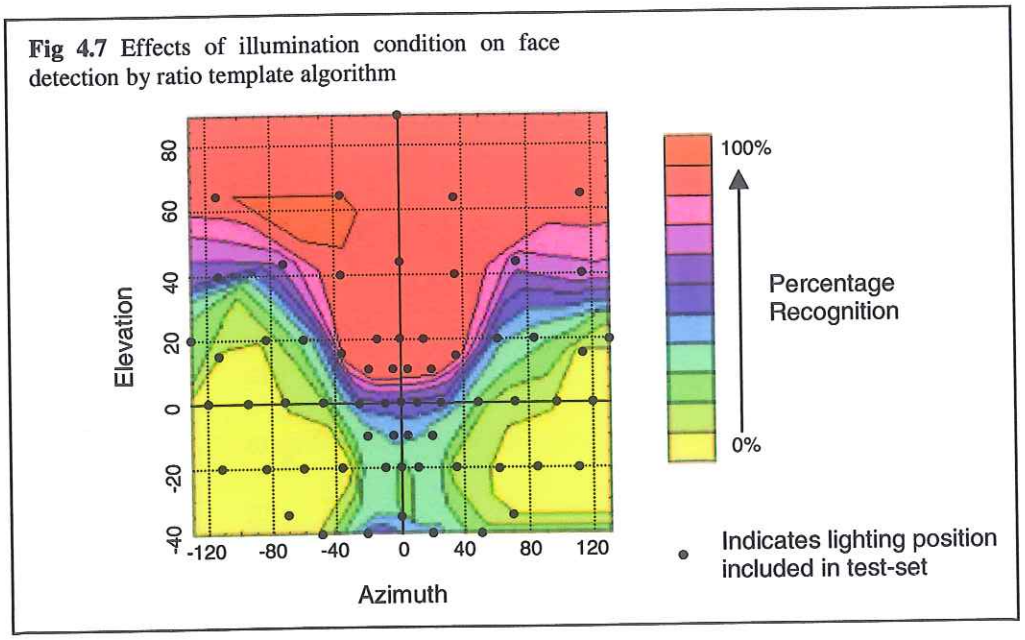


Fig 4.7 Effects of illumination condition on face detection by ratio template algorithm

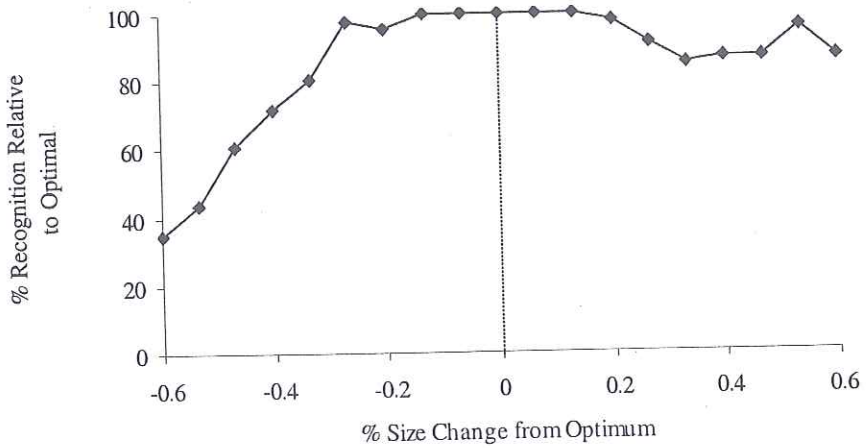
As can be seen, the approach works optimally when lighting is from above and in front of the face. It fails to correctly locate faces at all when illumination is from below the horizon and also when lighting is to the left or right and with low elevation. The

isolated patch in **figure 4.7** where 100% successful recognition can be seen is an artefact of the small data set used (only ten images used for each illumination condition).

4.4.2 Face Scale

To investigate the effects of changes in head scale, images were taken from the Essex face database [Hond 1997] (chosen as the faces in this database are consistent in size and in lighting condition and are easily identified by the face tracker) and the sizes adjusted (larger and smaller) away from the optimum. The recognition rates were then recorded. The results (**figure 4.8**) show that the ratio template algorithm is effectively able to cope with changes in size ± 20 % without significant degradation in performance. The performance then drops away rapidly when the image size is reduced by more than 30%. When the size is increased above 20% of the optimum performance begins to drop away before starting to improve again. This improvement in detection at larger scales results from the spatial template of the ratio template algorithm also having an affinity for locating the bridge of the nose when at an appropriate scale.

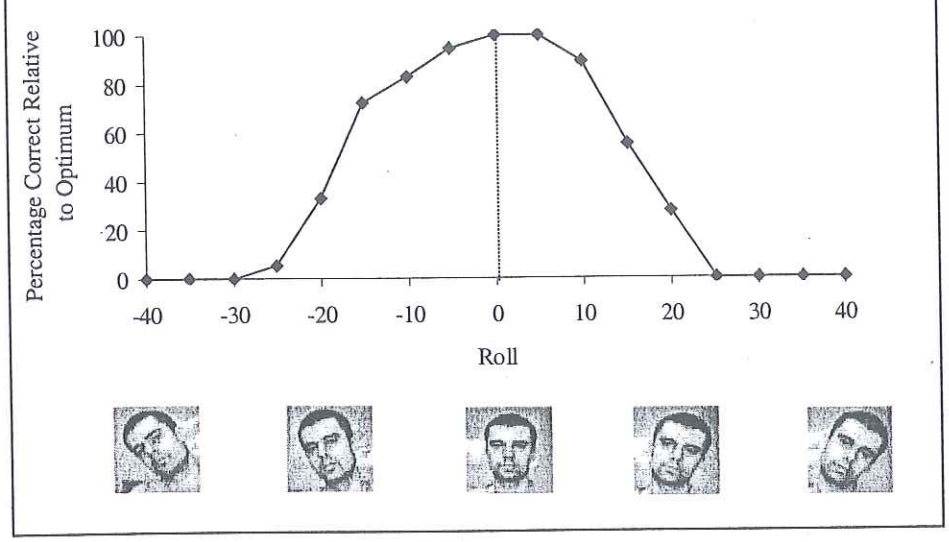
Fig 4.8 Tolerance to scale change of ratio template algorithm



4.4.3 Roll

Images for determining the tolerance of the tracker to head roll were obtained in our laboratory. Under ambient lighting conditions sequences of subjects were recorded as they slowly rolled their heads from side to side. Images at 5° intervals of roll were then extracted from these sequences for each subject. There were 9 subjects in the test

Fig 4.9 Effects of Roll on Face Detection by RatioTemplate Algorithm



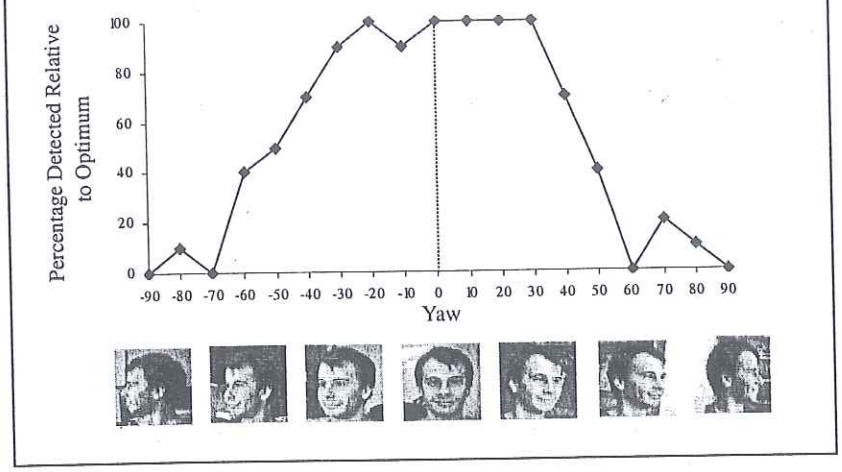
set, with two images of each subject used in each pose. The results (**figure 4.9**) show the tracker is able to handle a limited amount of head roll (± 10° from vertical performance remains above 80% of optimal). Beyond these bounds performance rapidly drops away as expected.

4.4.4. Yaw

Once again images for determining the effects of yaw were taken in our laboratory. To keep illumination constant, subjects were asked to sit still whilst a camera moved around them to fixed locations and images taken. Images of 9 subjects were taken and results (**figure 4.10**) show that the tracker is able to effectively handle

changes in yaw of around 30° (performance remains above 80% of optimum) in either direction from the fully frontal viewpoint.

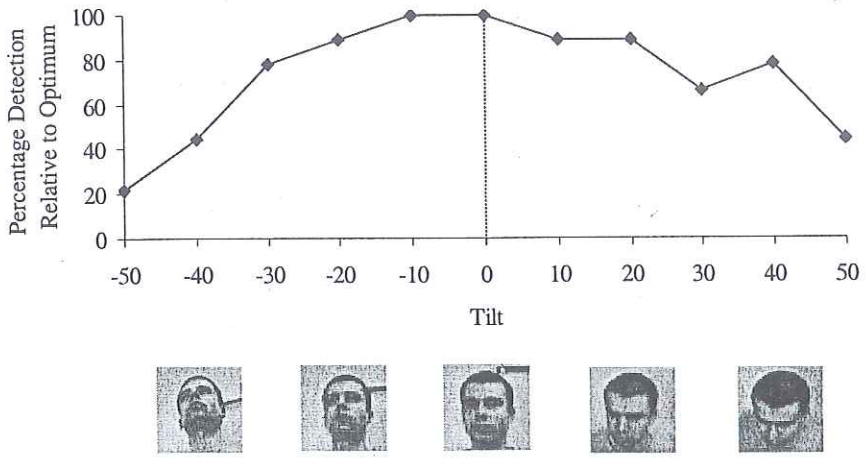
Fig 4.10 Effect of Changing Yaw on Face Detection



4.4.5 Tilt

To determine the effect of tilt on performance, images were again obtained in our laboratory. Pictures were taken under ambient room lighting, and subjects were asked to stand against a wall. The height of the camera was altered to a number of pre-determined points. Results can be seen in **figure 4.11**. Results show the approach is tolerant to limited changes in tilt, with detection remaining at above 80% of optimum with a tilt of ± 20°. The full conditions under which the face tracker can operate are summarised later in **table 4.2**.

Fig 4.11 Effects of Tilt on Face Detection by Ratio Template Algorithm



4.4.6 System Speed

The face tracker runs at ~6fps on a 192x123 and 2.5fps on a 384x247 image using a 450MHz Pentium III machine with Matrox Genesis DSP boards. Such frame rates are possible as the components added to the ratio-template algorithm are easily processed. The ratio-ratios operator and the matching density use data provided by the ratio-template algorithm itself. Image motion can quickly be determined, and the morphological eye/mouth detection, although more lengthy, takes relatively little time to process on the Matrox hardware (see **appendix 2**).

4.5 Example Sequence for Full Face Tracker System

Exemplar frames from a typical recorded sequence are shown in **figure 4.12**, and the complete movie is to be found in a file named "*FaceTracker.raw*" on the compact disk attached to the back of the thesis. The sequence is 276 frames long and the images are 384x247 pixels in size. The modified ratio template algorithm alone locates the face in 93% of images in the sequence where a face is visible. However, a large number of false identifications are also present (see bottom panel of **figure 4.12**). The hybrid face tracker successfully removes these false positives and correctly locates only the actual face in 89% of images in the sequence where a face is visible. A large proportion of

frames where the hybrid system fails to identify the face when the ratio-template model succeeds are in the section of the sequence where the camera is panned to the right. The system has not currently been designed to handle such an event as the applications for which this face tracker, and hence the expression recognition system as a whole, has been designed use a single fixed camera.

4.6 Speed-Ups for Incorporation into Real-Time Expression Recognition System

The basic speed of the face tracker is ~2.5fps on a 384x247 image. However, significant speed-ups are desirable for use in the expression recognition system in order to free more time for the determination and processing of motion information. Thus strategies are employed to further speed up the face tracker.

Once a face is located in the image, the processing separates into two threads. The first thread uses the basic ratio template algorithm alone to continue to search the location in the image currently deemed to be the face. Thus any movement of the face is tracked by the ratio template algorithm and a lock is maintained on the face.

The second thread uses the complete face tracker as described in this chapter, but spreads its search of the entire scene over 8 frames of processing, carrying out a single check for new objects that are more face-like than the one currently locked onto. This search over the entire scene occurs whilst 8 frames are grabbed, the current face tracked by the basic ratio template algorithm and the facial expressions classified.

This approach not only increases the frame rate to ~14fps on a 384x247 image, but also serves to improve stability, as the face tracker can change location a maximum of once every 8 frames. This approach thus leads to the tracker operating at speeds 5 times that of the basic tracker and plays an important role in allowing the completed system to run at a frame rate of 4fps.

4.7 Summary of Face Tracker

To summarise, the ratio template algorithm has been extended and its tolerance to changes in illumination improved by changing its spatial face model to incorporate the

golden ratio. This has been combined with additional processing stages to remove false positives. The basic tracker runs at a speed of ~2.5fps on a 384x247 image, but speed-ups have been included that enhance the speed to ~14fps. The conditions under which the tracker can effectively operate are summarised in **table 4.2** below.

Table 4.2 Summary of conditions under which face tracker can operate

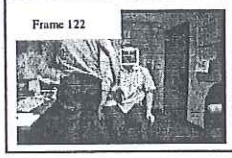
Operating condition	Tolerance
Illumination	Lighting from above. Performance gradually degrades as lighting source moves to left or right of subjects face
Scale	± 20 % from optimal scale
Roll	Head ± 10° from vertical
Yaw	Head ± 30° from frontal view around horizontal plane
Tilt	Head ± 20° from frontal view around vertical plane

The completed system has the advantage that it is very simple to implement and the operation of the ratio template algorithm is intuitively easy to understand, with regions of the spatial face mask corresponding to easily identifiable facial regions in a frontal view. It also runs at high frame rates, with further speed-ups easily included. Unlike the systems described elsewhere [Viola 2001, Rowley 1998, Yang 2000] it requires no pre-training with data taken from large face databases. Importantly for the work presented here, this face tracker also provides a rough spatial map of facial feature locations, providing vital information to the expression recognising components of the work.

Fig. 4.12 Example frames from face tracking sequence

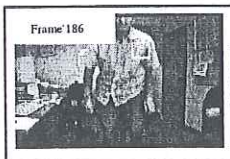
Frame 1: Face tracker correctly identifies face in the cluttered scene. The face is lit from above and in front.

Frames 80-111: Tracker continues to track face moving round scene. Ratio-template algorithm is running at a single scale, but images demonstrate the reasonably large range in size over which it can track.

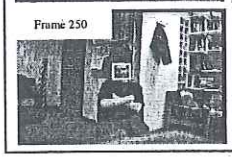


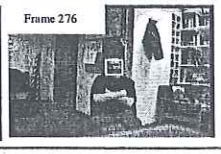
Frames 120-147: A 2nd figure enters the scene, occluding the face for a couple of frames. 2nd figure then moves around in background.

Face tracker stays locked on face despite obstruction due to occlusion handling. The figure moving in background does not cause incorrect identification.

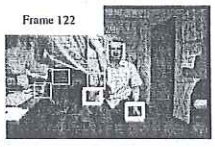


Frames 184-218: 1st figure leaves the scene and camera pans to a 3rd figure lit from above and to the left. Face initially not detected while camera is moving.





Frames 226-250: Tracker locks on face, loses the lock for a couple of frames and then returns to face where remains till frame 276, the last in the sequence.



Frames 1-250: Examples of raw data obtained from ratio template algorithm alone showing large number of false positives in scene. These example frames are representative of the number of false positives present throughout the sequence.

5 MOTION DETECTION

Once a face has been located in the scene by the face tracker, an optical flow algorithm is used to determine facial motion. Motion information is used for the purposes of expression recognition as, first, expressions are inherently dynamic events, and, second, by using motion the task is simplified as it ignores variations in the texture of different people's faces. Hence, the facial motion patterns seen when each of the basic emotions is expressed are similar, independent of who is expressing the emotion. Interestingly, facial motion alone has already been shown to be a useful cue in the field of human face recognition, for example in determining gender [Hill & Johnston 2001].

In the system described here, the multi-channel gradient model (MCGM) is employed to determine facial optical flow. This chapter first introduces other optical flow algorithms in **section 5.1,** and then briefly outlines the MCGM in **section 5.2**. The incorporation of the MCGM into the expression recognising architecture is discussed in **section 5.3** and a demonstration of the differences in flow output from the model when exposed to different facial expressions is given in **section 5.4**. The key points of this chapter are summarised in **section 5.5**.

5.1 Optical Flow Algorithms

Accurate measurement of image velocity is a vital component of many problems in computer vision, with applications including object tracking, surveillance, and object recognition [Barron 1994]. The process involves using the intensity values of sequential still images to obtain 2-D motion fields. However, as intensity values are solely used, optical flow techniques can only ever provide an estimation of actual motion. This is because, in a region of constant intensity, motion cannot be detected and an infinite number of potential solutions exist. For example, consider a rotating sphere with no texture. If the sphere is rotated on a stationary point at its centre, no change in image intensity is observed. Nonetheless, the sphere is still moving. Another problem with optical flow techniques is that it describes 3-D motion using a 2-D motion field.

Many approaches to obtain 2-D motion fields from image sequences have been proposed in the past and generally fit into one of four categories. These are the differential, region-based matching, energy-based, and phase-based approaches. This short review summarises each approach in turn and then briefly describes a prominent example of each. More detailed surveys, with comparisons made between the approaches, can be found [Barron et al 1994, Liu et al 1998].

5.1.1 Differential Approach

Differential techniques use spatio-temporal derivatives of either image intensity or filtered versions of the image to determine optical flow. Speed is calculated by examining ratios of outputs from spatio-temporal filters. A problem with this approach is that the ratio can become infinite when the denominator of this ratio becomes zero (at peaks and troughs in the image), and thus velocity is undefined. This problem can be overcome by introducing conditioning factors on the ratio denominator or by using derivatives of increasing order and then combining them such that the denominator never becomes zero.

An example of a differential method is the approach developed by Lucas and Kanade [1981], who pre-smoothed image data and then computed 1st order derivatives only. Barron et al [Barron et al 1994] carried out extensive comparisons between different optical flow algorithms and found that the Lucas approach gave accurate results (2nd best performance of all algorithms tested).

5.1.2 Region-based Matching

Region-based matching approaches, as used in MPEG-4 standard [Mukherjee 1998], use tiles from previous images and attempts to fit these tiles to subsequent image frames. The best match achieved is the most likely actual tile displacement and, by measuring this displacement, optical flow can be obtained. However, Barron and co-workers [Barron et al 1994] concluded that matching-based techniques provide less accurate velocity estimates than other techniques. Nevertheless, they do have the advantage of requiring only 2 or 3 frame long image sequences to determine motion, unlike differential approaches.

5.1.3 Energy-based Approach

This template approach based on Fourier representation exploits the fact that 'all non-zero power associated with a translating 2-D pattern lies on a plane through the origin in frequency space' [Barron et al 1994]. It involves the application of a number of velocity tuned filters to the image sequence, each tuned to a particular oriented Fourier spectrum component, and, according to the outputs of these filters, it is possible to obtain image velocity.

One such example is the technique developed by Heeger [Heeger 1987]. This approach uses 12 Gabor-energy spatial filters. Each filter is applied at several different scales and is tuned to a different spatial orientation and temporal frequency. By comparing the actual and predicted responses of these filters it is possible to estimate image motion. This approach has been found to be less reliable than most of the other techniques tried [Barron et al 1994].

5.1.4 Phase-based Approach

Similar to energy-based approaches, phase-based approaches involve the application of a number of velocity tuned filters. However, velocity is defined here in terms of the phase behaviour of filter outputs. Fleet and Jepson developed the first phase-based approach [Fleet & Jepson 1990]. This approach was the most accurate of those tested by other workers [Barron et al 1994], but had the disadvantage of having a particularly high computational load. Another problem with this technique is that a high response from a filter can be caused not only by the presence of motion to which the filter is sensitive, but also by high image contrast.

5.2 The Multi-Channel Gradient Model

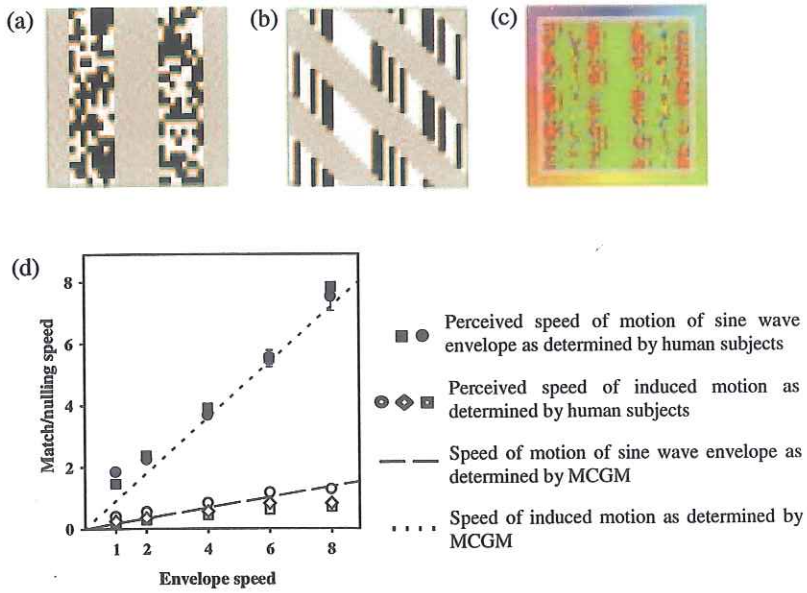
The MCGM is a differential technique based on a model of the human cortical motion pathway. It operates in 3 dimensions (two spatial, and one temporal), recovering the dense velocity field of the image at all locations. The model involves the application of a range of spatial and temporal differential filters to the image, with appropriate ratios being taken to recover speed and direction (a detailed description of

the approach can be found in the paper of Johnston et al [1999]). The taking of these ratios means the MCGM is robust to changes in scene luminance, removing problems associated with using Fourier energy or template matching methods of recovering optical flow. Interestingly, there is evidence to show that the MCGM is a biologically plausible model of the human cortical motion pathway, as it has been shown to correctly predict a number of motion-based optical illusions.

Figure 5.1 provides an example of this correct prediction of optical illusions. Humans see not only the rightwards motion of the square wave in the image sequence shown in **figure 5.1(a)**, but also perceive motion in the opposite (leftward) direction at the borders of the wave. Interestingly, a similar result is seen in the output of the MCGM (**figure 5.1(c)**). In the high and low contrast regions, the MCGM indicates movement to the right (green regions), but at the border of the wave it finds movement to the left (red regions).

To confirm this result, psychophysical studies were carried out to determine if the speeds of the rightward and leftward components of motion in this sequence corresponded between humans and the MCGM (**figure 5.1(d)**). Two experiments were carried out, both using the same sequence of a sine wave passing over a binary static background of noise.

Fig 5.1 Prediction of optical illusion by MCGM



(a) Single frame of modulated noise stimulus sequence where a square wave moves rightwards across a static binary background of noise (b) a space-time plot of the sequence, with time increasing downwards. Plot shows grey square wave moving to right and static background noise (c) Motion output of MCGM. Direction of motion indicated by position of corresponding colours in colour wheel bordering the image (eg green indicates movement to the right) (d) Psychophysical results for stimulus of a sine wave moving over a static binary background of noise. The x-axis indicates the changing speed of the wave (or envelope) across the image, and the y-axis indicates perceived speeds.

The first experiment determined the speed at which humans perceived the sine wave to be moving. The methodology involved asking the subjects to match the speed of the sine wave across the noisy background to that of a sine wave moving across a neutral background. The second experiment determined the speed at which humans perceived the induced motion in the opposite direction to be moving by 'nulling' this induced motion. This 'nulling' was carried out by changing the speed of the noisy background (rather than having a static background) until the subjects no longer perceived any leftwards motion at the sine wave boundary. By examining **figure 5.1d** it is clear that in both cases (for motion of sine wave and for induced motion) the figures

obtained from human subjects closely correspond to those obtained from the MCGM, although the model slightly overestimates the induced effects at high speeds.

Importantly for the work presented here, a real-time version of the MCGM has been implemented in hardware on a machine with Matrox Genesis DSP boards [Dale 2002]. This version of the model is greatly cut down, using up to 3rd order derivatives (rather than 6th order derivatives as used by full version) and 8 orientations (rather than 24). However, it still recovers a robust measure of optical flow and is ideal for the purpose of incorporation into this real-time expression recognition system. It runs at a speed of ~18fps on a 64 x 41 image.

5.3 Incorporation of MCGM into real-time expression recognition system

The MCGM is gated by the face tracker described in **chapter 4**. The tracker provides a location (most probably the face) in the scene and turns on the MCGM. Image motion is determined solely at the face location and its immediate surrounds. The region of motion computed provides a border around the face so that reasonable horizontal and vertical rigid head movements are possible. If only the motion of the face is determined with no border, rigid head motions would immediately move the face outside the region of the image where the optical flow algorithm is operating. This is undesirable as the MCGM must process several frames of information before giving correct motion output. The border size is easily modified but of course has an effect on computational speed. **Figure 5.2** summarises the gating approach.

Fig 5.2 Gating of MCGM by face tracker

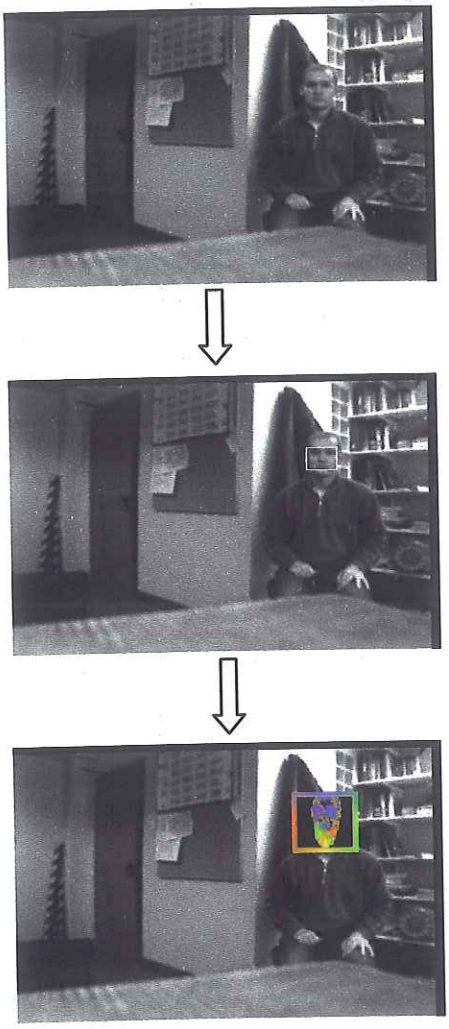


Image feed provided to real-time expression recognition system

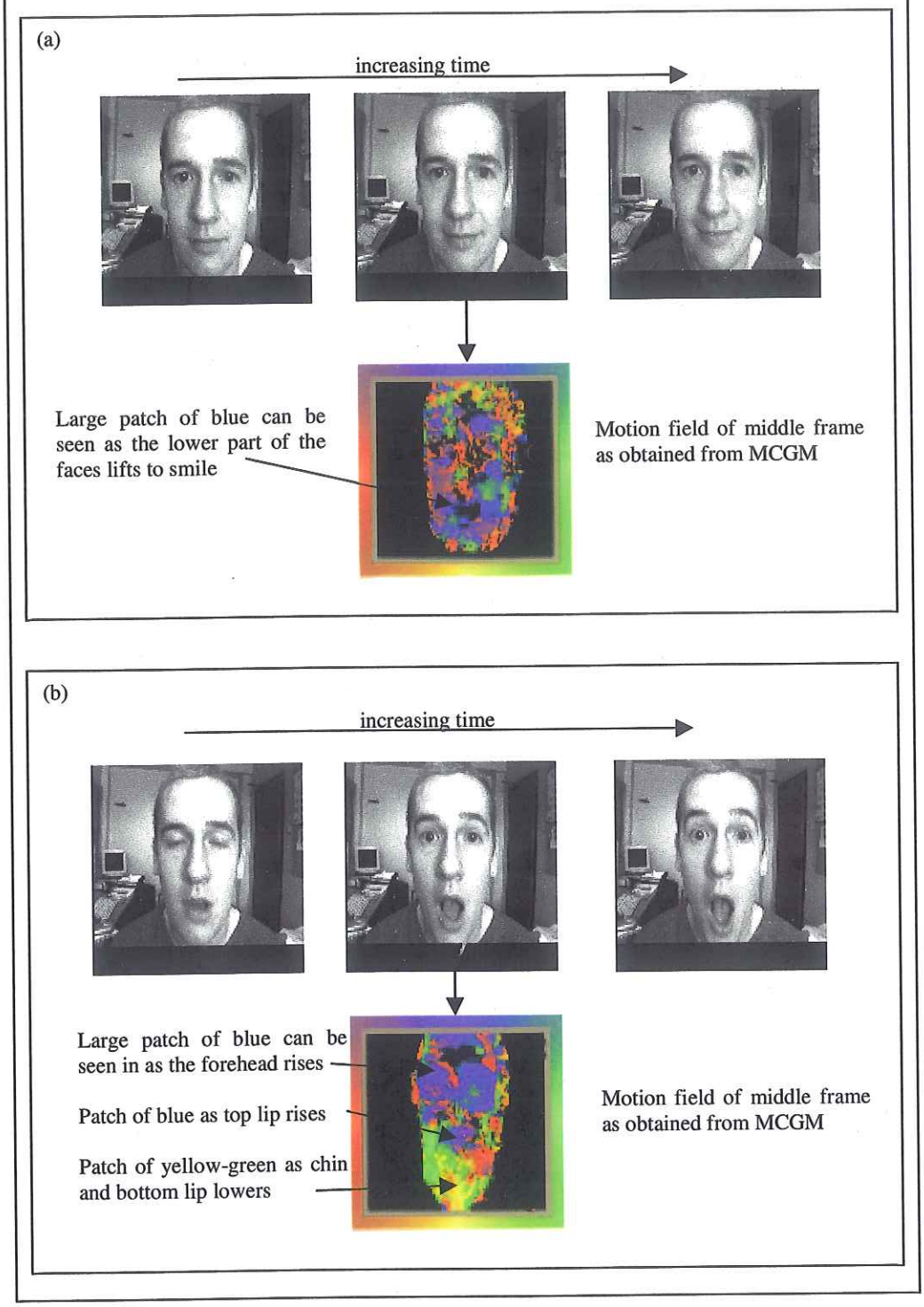
Face tracker determines location in image where face is most probably located

Tracker gates MCGM, turning it on at the location in the image where face is positioned. Note the extra motion information calculated bordering the face, allowing limited movement of head vertically and horizontally

5.4 Model Output

The MCGM provides velocity, that is both speed and direction information, and, as facial expression involves the movement of different facial parts in different directions at different speeds, the output provided by the model will vary depending on the expression to which it is exposed. To demonstrate these differences, it was decided to run the MCGM on two separate sequences, the first sequence of someone changing from a neutral expression to a happy one, and the second from a neutral expression to a surprised one. The results are shown in **figure 5.3**.

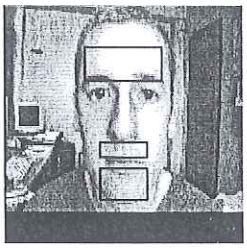
Fig 5.3 Output of the MCGM when exposed to (a) Happiness expression (b) Surprise expression

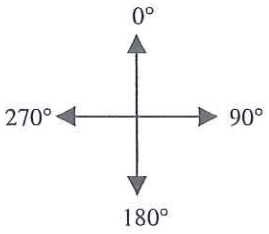


To obtain actual figures for these differences, the direction of motion, as obtained by the MCGM, was averaged over three regions of the face when exposed to the expression of happiness and surprise. The results are given in **figure 5.4** and show that the directions of motion of the top lip and forehead regions are fairly similar for a smile and for expression of surprise. However, the motion of the bottom lip/chin region is completely different, with it moving upwards at the onset of a smile and downwards at the onset of an expression of surprise.

Fig 5.4 Difference in direction of motion when happiness and surprise are expressed (a) The regions over which motion information are averaged (b) the direction of motion in these regions. 0° indicates upwards motion, 90° rightwards motion, 180° downwards motion, and 270° leftwards motion

(a)





(b)

Facial Region	Direction of movement	
	Happiness	Surprise
Forehead	38°	6°
Top lip	11°	33°
Bottom lip/chin	0°	170°

In addition to direction information, the MCGM also provides speed information not given here that would also allow distinction between different expressions, even if regions move in similar directions as they move at different speeds. Thus, it is thought that provision of the motion information provided by the MCGM to pre-trained classifiers (such as multi-layer perceptrons and Support Vector Machines) would be sufficient to distinguish between a range of different facial expressions.

5.5 Summary

This chapter has described approaches to recovering optic flow and specifically the multi-channel gradient model used by the expression recognition system presented in this thesis to determine facial motion and shown how its operation is gated by the

outputs of the system's face tracker. It has then been demonstrated that velocity field outputs from the MCGM change according to the facial expression to which it is exposed, thereby showing that it is a useful tool for distinguishing between human facial expression types.

6 CANDIDATE EXPRESSION RECOGNITION APPROACHES

Once facial motion has been determined, it is necessary to place the motion signatures into the correct class of facial expression (either a non-expression or one of the six basic emotions). To do this we need, first, to decide on what approach to use to carry out the classification procedure and, second, to find a way of representing motion data to make the distinction between classes as easy as possible. The two approaches chosen for comparison for the classification procedure are multi-layer perceptrons (MLPs) trained using the back propagation algorithm and Support Vector Machines. MLPs have been used previously for expression recognition by Bartlett [1999b], whilst SVMs have been used by Dumas [2001]. Subsequent chapters compare and contrast the relative performance of these two approaches and search for the best data representation strategy. However, first it is necessary to introduce these two techniques and give some preliminary information regarding general data representation strategies.

Thus, this chapter provides some background information relating to MLPs trained using the back propagation algorithm (**section 6.1**) and Support Vector Machines (**section 6.2**). **Section 6.3** introduces some of the general approaches used in this work to pre-process motion data prior to input into the classifiers. **Sections 6.4** & **6.5** describe the specifics of the MLP and SVM classifiers used. **Section 6.6** then describes the methods taken to evaluate classifier performance before, finally, a summary is provided in **section 6.7**.

6.1 Multi-Layer Perceptrons and the Back Propagation Algorithm

Artificial Intelligence (AI) has been defined by Luger and Stubblefield [1993] as "*the branch of computer science concerned with the automation of intelligent behaviour*". AI systems can be thought of as falling into one of four categories, those that think in a human-like way, act in a human-like way, think rationally, or act rationally [Russell & Norvig 1995]. The two main approaches to Artificial Intelligence are the symbolic and connectionist approaches.

The symbolic approach is a top-down approach, seeking to reproduce intelligence by analysing cognition without considering the structure of the brain. It

models problems using symbols, manipulating these symbols using a set of logical rules to reach conclusions. The symbolic approach is used, for example, to develop expert systems that capture human expert's knowledge to make decisions about problems that occur in the real world [Shortliffe 1976].

The connectionist approach is bottom-up, using neurophysiology as its inspiration to model the functioning of the brain. The connectionist approach combines the outputs of a number of simple, interacting processing units to model a problem. Rather than knowledge being coded explicitly by symbols (as in the symbolic approach), the connectionist approach codes the knowledge implicitly.

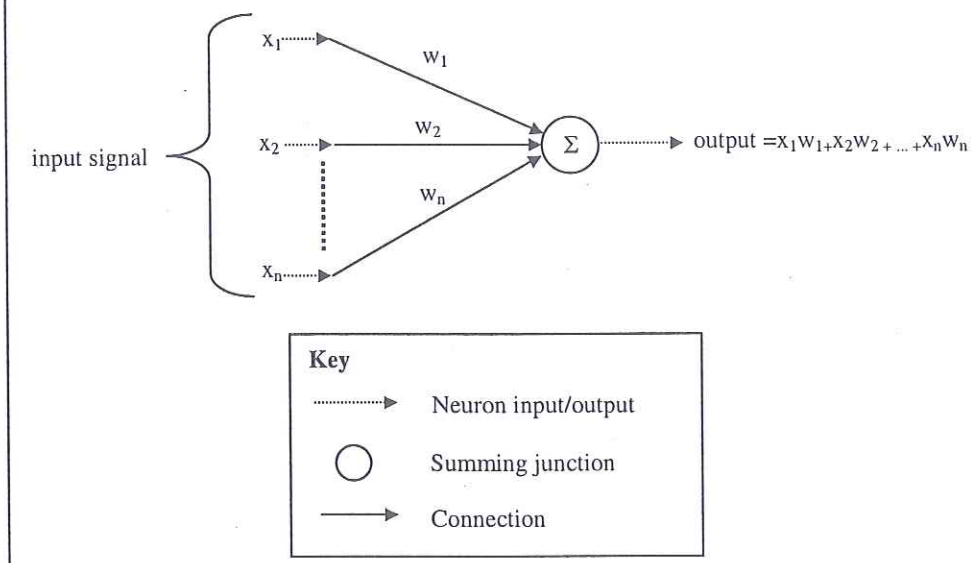
Artificial neural networks (ANNs) are used by the connectionist approach and their development was inspired by the human brain in that, unlike normal digital computers, ANNs process information in a parallel manner [Wasserman 1989]. They are built up of simple processing units with connections between these units, and acquire knowledge through a learning process that allows them to store this knowledge effectively. Once set up, they can cope with minor variation in inputs (to a higher or lesser extent depending on the situation), a vital characteristic for their use in pattern recognition tasks where data may be noisy or where the data provided during the learning process is incomplete.

A number of different neural network types, architectures, and training methods have been developed over the past fifty years. However, the introduction presented here concentrates on a single approach used for the work in this thesis, the multi-layer perceptron (MLP) trained using back propagation. Nonetheless, many of the issues addressed in subsequent sections are relevant for other types of ANN. Before discussing the structure of MLPs, it is first necessary to introduce the processing unit of a MLP, called the artificial neuron.

6.1.1 The Artificial Neuron

The artificial neuron is the information-processing unit that is integral to the functioning of a neural network, and is designed to mimic a simple model of the biological neuron. A diagram of the artificial neuron is provided in **figure 6.1**. The inputs are labelled x_1 to x_n and are introduced to the artificial neuron. Each input is multiplied by an associated weight (ie x_1 is multiplied by w_1). All the weighted inputs are then summed to produce the output of the artificial neuron. The weights are analogous to a biological neuron's synaptic connections, and the summation to its cell body.

Figure 6.1 – The artificial neuron

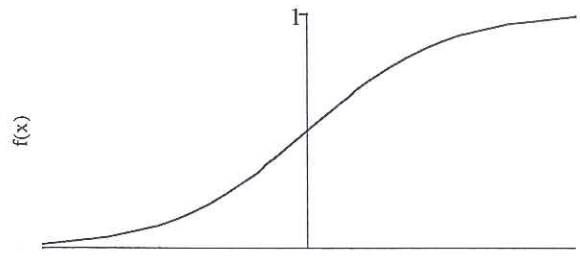


The pioneering work of McCulloch and Pitts [McCulloch & Pitts 1943, Pitts & McCulloch 1947] showed that by combining a sufficient number of these units and correctly setting the weights it was possible to compute any computable function. They later used this model of the artificial neuron for the purposes of pattern recognition [McCulloch & Pitts 1947], joining a number of neurons into a single layer. A threshold was used such that, if the output was greater than the threshold, the output of the neuron was labelled 1, and 0 if not. Systems that use the type of neuron shown in **figure 6.1** are called perceptrons.

6.1.2 Activation Functions

The thresholding function used by McCulloch & Pitts is an example of an activation function. Activation functions are used by artificial neurons to limit outputs to a specified range. However, rather than thresholding, non-linear sigmoidal activation

Figure 6.2 – Logistic function



functions are far more commonly used by today's perceptrons. Sigmoidal functions are S-shaped, allowing networks to handle both small and large input values. The sigmoid is centred around zero, giving high gain for small signals that need amplifying, and lower gain for largely positive or negative values, thereby preventing saturation. An example sigmoidal function, known as the logistic function, is given in **figure 6.2**, where x is the weighted sum, and f(x) is the modified perceptron output. This function compresses the output of a perceptron between the range 0 and 1.

In addition to non-linear activation functions, one other feature often used in perceptron architectures is bias [Haykin 1999]. Bias is included to offset the origin of the logistic activation function, thereby speeding up the training process. The weighted sum of the input vector to a perceptron (see **figure 6.1**) may not necessarily fall on the optimal part of the activation function. By using bias one can shift the weighted sum to a better part of the curve, and speed up training. Bias consists of a constant input to each perceptron of 1, and this input is weighted as with other inputs. During the training process the weighting for each individual perceptron's bias is learnt.

6.1.3 Single Layer Perceptrons

The power of artificial neurons comes from their combination into single multi-perceptron architectures. The simplest way to combine perceptrons is into a single-layer, as done by McCulloch & Pitts [1947]. An example single layer perceptron is given in **figure 6.3**.

The input vector arrives at the input end (left side) of the network and propagates through the network node by node until it finally arrives at the output end (right side) of the network, giving the network response. More specifically, it operates in the following manner:

1) The network is provided with an input vector, as indicated by the dashed arrows at left side of network.
2) Each element of the input vector then passes along the indicated connections (solid arrow) to each node. These connections are weighted, so the value of each input is multiplied by the weight of each connection.
3) The weighted inputs at each perceptron are then summed.
4) The weighted sum of each perceptron is modified by the activation function and a value is output.

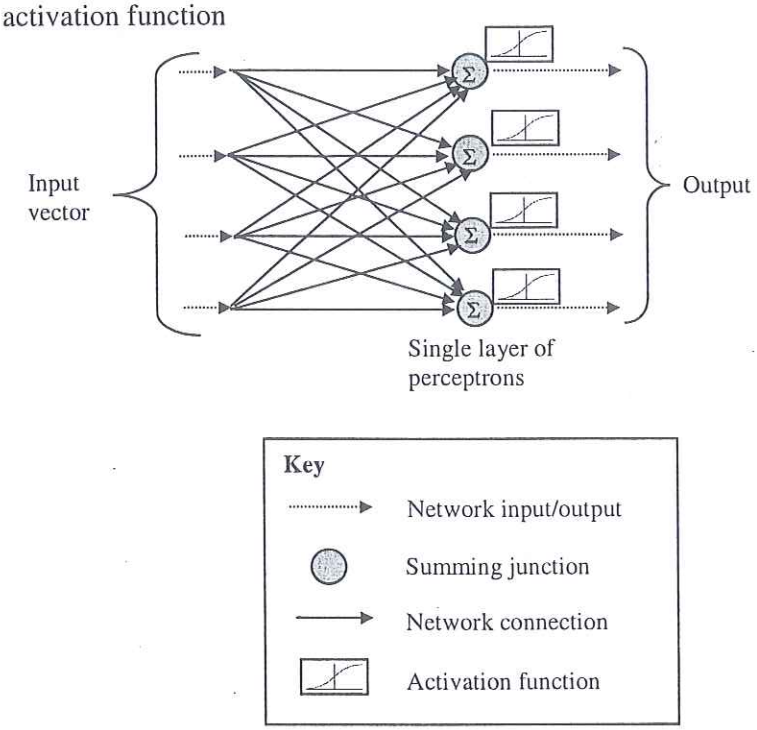
6.1.4 Learning

It is the ability of perceptron architectures, such as the one shown in **figure 6.3** (and more generally neural networks), to learn an input/output mapping that makes them so useful. They can alter their behaviour in accordance with the situation to which they are exposed such that, given a set of inputs, they can produce consistent responses.

By correctly setting the weights of each connection it is possible to use perceptron architectures for a range of classification and regression tasks. However, the early work of McCulloch & Pitts [McCulloch & Pitts 1943, Pitts & McCulloch 1947] necessitated the manual setting of connection weights. The use of perceptrons for complicated tasks, where thousands of connections could be involved, was therefore

unfeasible. Thus, the development of algorithms allowing automatic setting of weights was vital if the use of perceptrons was to become widespread.

Figure 6.3 – Architecture of a single-layer perceptron using a sigmoidal activation function



Several years later, a number of approaches allowing perceptron weights to be learnt were developed [Wasserman 1989]. These algorithms operated by the application of training example vectors to the perceptrons, and the subsequent modification of the weights, such that the outputs gradually converged to those desired. There are two main types of training algorithm, supervised and unsupervised.

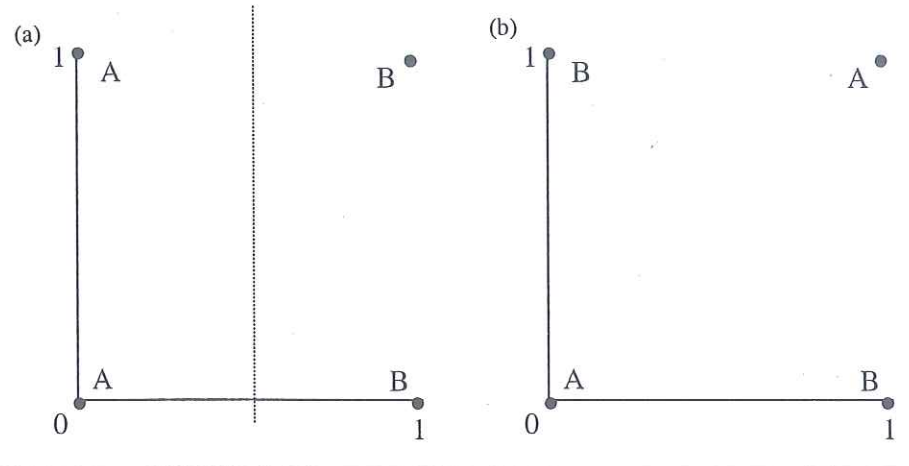
Training of MLPs is, in general, supervised, with supervised training requiring the application of an input vector in conjunction with a target vector representing the desired output. Such pairs are often called training pairs. At each training iteration the network response to the input vector is compared to the target vector and adjustments made to the weights to reduce any error.

On the other hand, unsupervised learning does not use target vectors and thus groups similar input vectors solely into classes. In general, networks trained in this manner must have their outputs examined subsequent to training so that the outputs can be understood. Biologically, unsupervised training methods are of importance as they demonstrate how learning can take place in the absence of a teacher. Many of the early unsupervised learning algorithms were based on ideas proposed by Hebb [1961]. The fundamental concept behind these algorithms was that, if two nodes on either side of a connection became activated simultaneously when an input vector was applied, then the strength (ie weight) of that connection was increased. Conversely, if the two nodes were activated separately, the strength of the connection was reduced. Thus, some paths became strengthened in response to certain input vectors.

6.1.5 Linear Separability

With the development of Hebbian and many other learning algorithms (eg [Widrow 1960]) there existed for a period of time considerable excitement about the potential applications for single layer perceptrons. However, researchers soon found that single layer perceptrons failed at what were seen as relatively simple tasks. Minsky and Papert [1969] investigated these failures and found that, crucially, single layer perceptrons were unable to distinguish between two classes that were not linearly separable. The classic example of a non-linearly separable problem is the XOR function (**figure 6.4**).

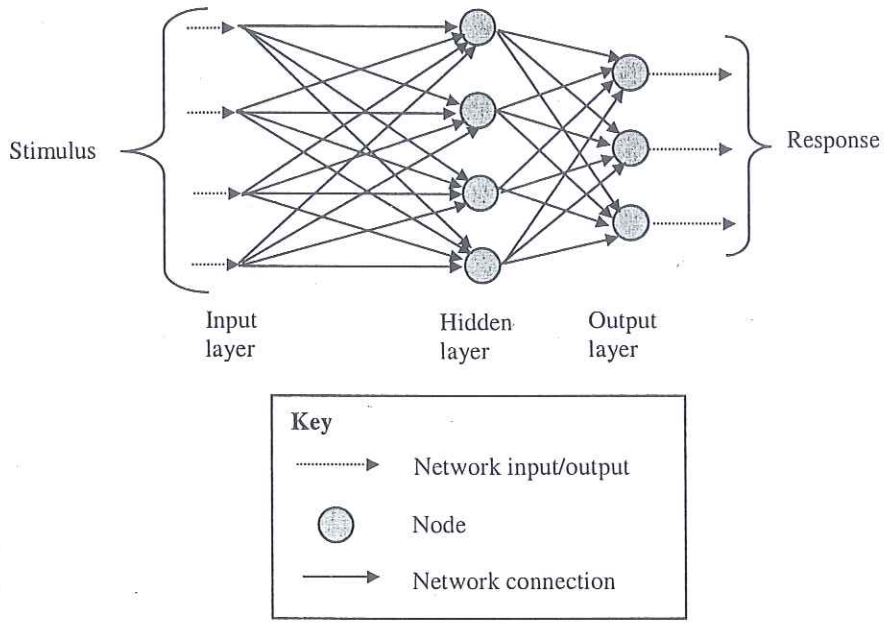
Figure 6.4 – Linear separability and perceptron classification (a) A linearly separable problem. The two classes A & B can be separated by a single straight line (decision boundary indicated by dotted line) (b) The XOR Problem. Two inputs are provided and each can be either a 1 or a 0. A response should be given only if one of the two inputs is a 1 (class B). If neither or both inputs are a 1 then no response is given (class A). The XOR problem is linearly inseparable, as it is not possible to separate the two classes, A & B, with a single straight line.



6.1.6 Multi-Layer Perceptrons

The basic architecture of a Multi-Layer Perceptron (MLP) is given in **figure 6.5**, showing a fully connected MLP consisting of two processing layers, the output layer and an additional processing layer in front of this, called the hidden layer (NB MLPs can include not just one hidden layer but multiple hidden layers). By inclusion of additional stages of processing, it is possible to solve non-linearly separable problems. Unfortunately, although early researchers understood the potential power of MLPs, for a number of years there was no effective training algorithm that could properly adjust the weights in all layers of the architecture. This led to a loss of interest in the field until 1986's back propagation algorithm, developed by Rumelhart, Hinton, and Williams [1986].

Figure 6.5 – Architecture of a multi-layer perceptron



6.1.7 Back Propagation Algorithm

The back propagation algorithm is an approach commonly used to train MLPs. Its presentation to a large audience in 1986 by Rumelhart et al [1986] led to an increased interest in MLPs, although it was subsequently found that the earlier works of Parker [1982] and Werbos [1974] had already described the approach.

MLPs trained using back propagation use sigmoidal activation functions (usually the logistic function shown in **figure 6.2**) and there are two reasons for this. The first is that sigmoidal activation functions are non-linear, and this non-linearity allows the network to perform non-linear tasks (MLPs using a linear activation function possess no greater representational power than single layer perceptrons [Wasserman 1989]). Secondly, with differentiation being fundamental to the back propagation algorithm, the easy differentiation of sigmoidal activations is advantageous.

Training with back propagation is supervised and requires the iterative application of training pairs and subsequent adjustment of weights so that the output of

the network to each input vector becomes closer to that desired. The training process proceeds as follows:

- Initialise MLP with small random weights (to prevent network saturation).
- Apply an input vector from a training pair in the training set.
- Calculate output of MLP.
- Determine error (ie how far is the actual output away from the desired output).
- Modify network weights according to an adaption rate and the estimated contribution of each individual weight to the network error.
- Repeat until error has reached an acceptable level.

Appendix 1 gives details of the operation of the back propagation training algorithm, also providing a simple worked example. MLPs trained using back propagation have been used for a range of problems, such as signature recognition [Everitt & McOwan 2003], character recognition [Yamakawa & Matsumoto 1998], mass identification in mammograms [Bovis et al 2000] and modelling of insect camouflage approach strategies [Anderson & McOwan 2003].

6.1.8 Problems with MLPs trained using back propagation

There are four main problems associated with MLPs trained using back propagation:

- The back propagation training process is inefficient (and therefore time consuming). Training speed can be enhanced with approaches such as conjugate gradient descent [Haykin 1999] (see **appendix 6**).
- It is hard to decide when to terminate the training process.
- Hand-crafting of structure and size is usually required to obtain good performance.
- The back propagation training algorithm can get stuck at local minima.

The speed of the training process is not addressed further here as the training process is carried out offline for the expression classification task and obtaining improved training rates is not of significance. However, the other issues are now discussed in greater detail.

6.1.8.1 Training Termination

One commonly used strategy to terminate training is to wait for the gradient of the rate of error improvement on the training set to become very small. However, if one trains the MLP for too long, over-fitting of data can sometimes be seen [Haykin 1999]. Over-fitting of data (**figure 6.6a**) occurs when a MLPs outputs become too closely tuned to the training data (which could be noisy) and thus the MLP no longer generalises effectively ie gives accurate outputs to the training examples, but performs poorly on novel examples.

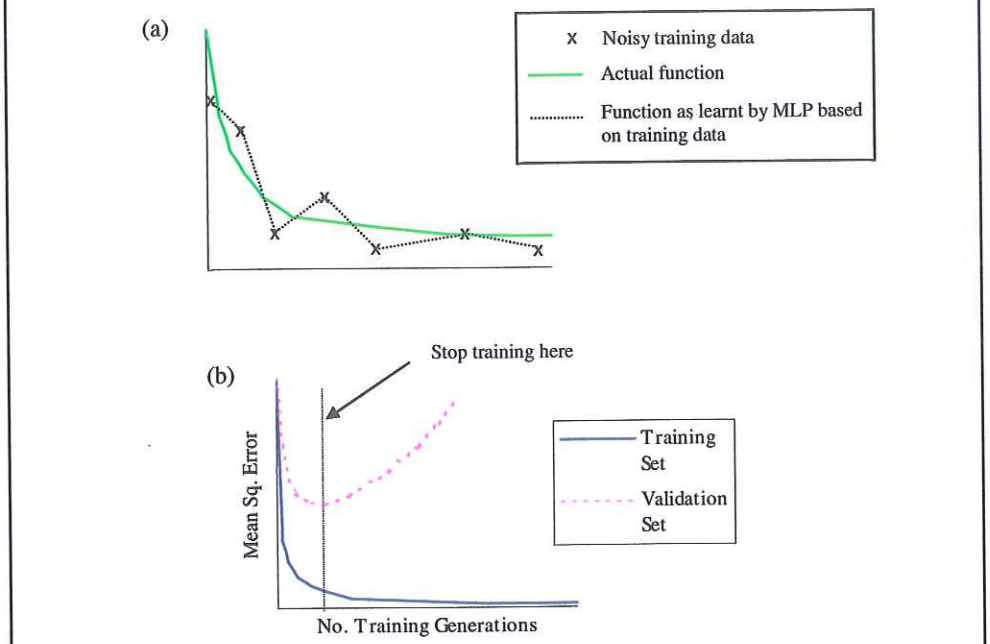
Use of cross-validation can be used to help overcome this problem. In cross-validation, the original training set is split into two parts, thereby creating an additional subset of data called the validation set. However, the validation set is not presented as a training example and therefore has no affect on connection weights in the MLP.

Rather, training is periodically interrupted and the validation set presented to the MLP. The error of the MLP in response to the validation set can then be determined. However, this error is not propagated back through the network to modify weights, but is instead used to test the ability of the MLPs to generalise. If the MLP performs well on a validation set (remember it has not participated in the learning of weights) it is reasonable to assume that the MLP would perform well on other novel examples, and thus has generalised effectively. Thus, once the error to this validation set is minimised, the training process is terminated (**figure 6.6b**).

6.1.8.2 Hand-Crafting

Cross-validation does not solve all problems associated with over-fitting, and thus a certain level of hand-crafting (eg modifying the size of the MLP) is required. This is due to the bias-variance dilemma.

Figure 6.6 – Overfitting of data (a) Due to noisy data, the MLP fails to effectively generalise to the solution (b) As training proceeds, although the error on the actual training set continues to lower, the error on the validation set eventually increases as MLP fits too closely to training data, losing the ability to generalise. Hence, stop training when error on validation set reaches its minimum



The training process of a MLP attempts to minimise the test (or expected) error (in that one attempts to make the MLP model the target function as accurately as possible) so that the lowest possible error is obtained when input examples are presented. The test error can be thought of as arising from two sources, the bias and the variance. The bias of a network is the average error of the MLP compared to the target function, whilst the variance is the amount that the MLP is sensitive to the training data. In general, the more connections that are present in a MLP the more sensitive it is to the training data.

An overly simplistic (ie small) MLP has a low variance but a high bias as it is not sufficiently complex to approximate the function accurately. An overly complex (ie big) MLP will have a low bias but a high variance as the MLP will be particularly

sensitive to the training data (ie if a different training set were used it would be likely that a completely different solution would be reached). Thus the bias-variance dilemma arises from the fact that a MLP with a low bias will have a high variance and vice versa. For MLPs, and neural networks in general, it is important to find the optimum balance between bias and variance.

By adjusting the size of a MLP, it is possible to adjust the complexity of the network, and thus the balance between bias and variance. This can be done manually by training a number of MLPs to find the optimal size. Alternatively, one can start with a large MLP and use network pruning techniques to remove or weaken certain connections in an orderly manner during the training process, in effect reducing the size.

6.1.8.3 Local Minima

Back propagation uses gradient descent to adjust the MLP weights, thereby following the error surface towards its minima. This can cause problems with complex error surfaces that are non-convex in shape and have multiple local minima as there is no guarantee that back propagation will find the global minimum. Instead it can get stuck in a locally optimal solution. If this occurs and performance is not satisfactory, it is necessary to retrain the MLP, although there is no guarantee that a global minimum will ever be reached [Wasserman 1989].

6.2 Support Vector Machines

As previously discussed, MLPs trained with back propagation have problems such as getting stuck in sub-optimal solutions and the bias-variance dilemma. An alternative approach commonly used for classification tasks are Support Vector Machines (SVMs). Support Vector Machines are an approach to pattern classification, with modifications having been developed to allow their use for regression. They are based on statistical learning theory and the ideas of structural risk minimisation that directly address the issue of the bias-variance. Developed by Vapnik [1995], they have been applied to problem domains such as speaker identification [Schmidt 1996], text categorisation [Joachims 1997], face detection [Ng & Gong 1999], face authentication [Smeraldi 1999], and identification of junk e-mails [Woitaszek 2003]. In addition to their basis in statistical learning theory, SVMs are attractive due to the fact that they are easy to analyse theoretically, unlike neural network based approaches that produce more complex models. This simple analysis is possible as SVMs correspond to a linear approach in a higher dimensional space.

This introduction to SVMs first describes, in **section 6.2.1,** the statistical learning theory underlying SVMs. **Section 6.2.2** then tells how this theory is applied in SVMs for solving linearly separable problems. This is followed in **section 6.2.3** with a description of how the approach is expanded for the solving of non-linear problems. A detailed introduction to SVMs has been written by Burges [1998].

6.2.1 Statistical Learning Theory

With learning machines such as SVMs, the objective is to train them so that they either accurately model a target function (for regression) or correctly classify novel examples (for classification). However, there is a trade-off between learning machines that can model more complex functions but may over-fit data (and so not generalise well), and more simple machines that, although they don't over-fit data, cannot model complex functions.

This problem has been addressed in statistical learning theory, which has shown that, given a finite set of training data, the best generalisation performance is achieved

when an effective balance is struck between the average error achieved on the training set and the capacity of the machine [Burges 1998]. The capacity of a learning machine in this case is defined as its ability to learn any training set without error.

These findings have been formulated by Vapnik [1995] such that an estimation of the test error (also called expected risk) of a learning machine can be obtained, and thus how well it is likely to generalise. The test error of a learning machine is given by:

$$Test\ Error \leq Training\ Error + \sqrt{\frac{h(log(2r/h)+1)-log(\eta/4)}{r}} \qquad (6.1)$$

where r is the number of training examples, h is a measure of machine capacity, and the probability that the value is correct is given by 1- η . The second term of this equation thus addresses the issue of matching the complexity of the machine to the quantity of training data. Please note that this only gives an upper bound to the actual error, not the error itself, and that there is a probability, η, associated with this upper bound.

The capacity of a machine, h, is measured in terms of Vapnik-Chervonekis (VC) dimension. The VC dimension is defined as the maximum number of points that can be correctly classified by a machine for all possible labellings of that set of points (NB a set of n points has 2n possible labellings).

Given a set of different learning machines, one can use **equation 6.1** to choose a machine with lowest test error. Overly complex machines that have over-fit data are penalised in **equation 6.1** by a high VC dimension. Conversely, overly simple machines unable to model the function are penalised in **equation 6.1** by a high training error. The machine giving lowest actual error is, in general, a machine with a good balance between capacity (VC dimension) and training error (also called empirical risk), and this machine would also be most likely to have generalised best. With this in mind, the method of structural risk minimisation (SRM) operates as follows:

- Minimise the training error for a set of pattern classifiers.
- Identify the classifier with the lowest test error according to each classifiers training error and capacity.

6.2.2 Linear SVMs

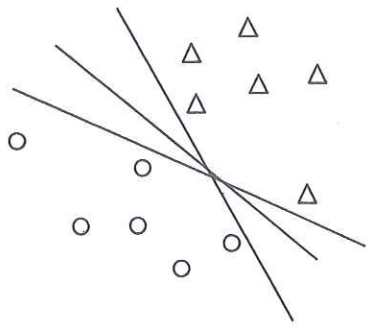
Consider a classification problem that is linearly separable. In such a case, it is likely that there would be a number of linear functions that could effectively separate the two classes, each forming a different hyperplane in the feature space (**figure 6.7a**). However, only one of these hyperplanes is considered the optimal hyperplane, defined as the hyperplane with a margin that maximally separates the two classes (**figure 6.7b**). SVMs operate by searching for this optimal hyperplane, and this they do by solving a constrained quadratic optimisation problem.

By solving this problem, one ends up with examples from each class on the border of the margin of separation (filled circles in **figure 6.7b**), and these examples are called support vectors. They are called support vectors as they alone participate in the definition of the optimal hyperplane, with all other examples playing no part in defining the hyperplane. Thus, all examples that are not support vectors can be removed or moved (as long as they do not cross the border of the margin of separation) without affecting the position of the optimal hyperplane.

So how does this tie in with the previously introduced statistical learning theory? Well, one possible approach to minimising the actual error is to hold the training error term of **equation 6.1** at zero and then attempt to minimise the 2^{nd} term. In the linearly separable case shown in **figure 6.7**, the two classes are completely separated by the hyperplane, thereby holding the training error (empirical risk) at 0. To achieve good generalisation capability one must then minimise the VC dimension. By only ever selecting the optimal hyperplane (with maximal separation), it reduces the number of possible hyperplanes and thus the capacity in **equation 6.1** (often called the confidence term). Therefore, SVMs directly address the principles of structural risk minimisation.

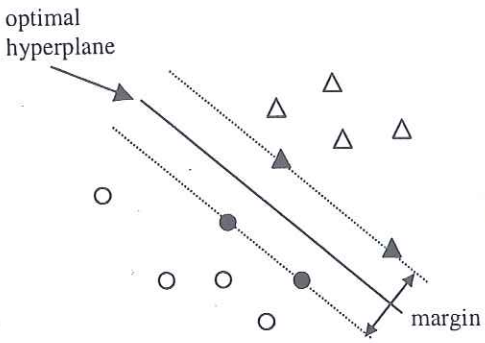
Figure 6.7 (a) Separation of two linearly separable classes by multiple hyperplanes (b) optimal hyperplane chosen by SVMs

(a)



Given two linearly separable classes (indicated by circles and triangles) often a number of hyperplanes can be found that separate the two classes

(b)



Only one of the hyperplanes is considered optimal. The optimal hyperplane is the one that optimally separates the two classes (ie has the greatest margin).

NB Filled shapes indicate support vectors

Importantly, however, the inputs to SVMs are often likely to be noisy or the data not separable. In such a case complete separation is either not possible (in the non-separable case) or may lead to overfitting (in the noisy case). Thus, better generalisation may be achieved if the confidence term is further reduced, even though this necessitates allowing training set errors that lead to an increased empirical risk.

A technique included in the formulation of SVMs makes this possible by allowing for some classification errors on the training set. Additionally, adjustments are possible, by modifying a single variable, allowing one to control the trade off between the complexity of the learning machine and the frequency of error on the training set. By increasing this variable, higher penalties are assigned to training errors, and thus changes are made to the weighting between the empirical risk and confidence terms of **equation 6.1**.

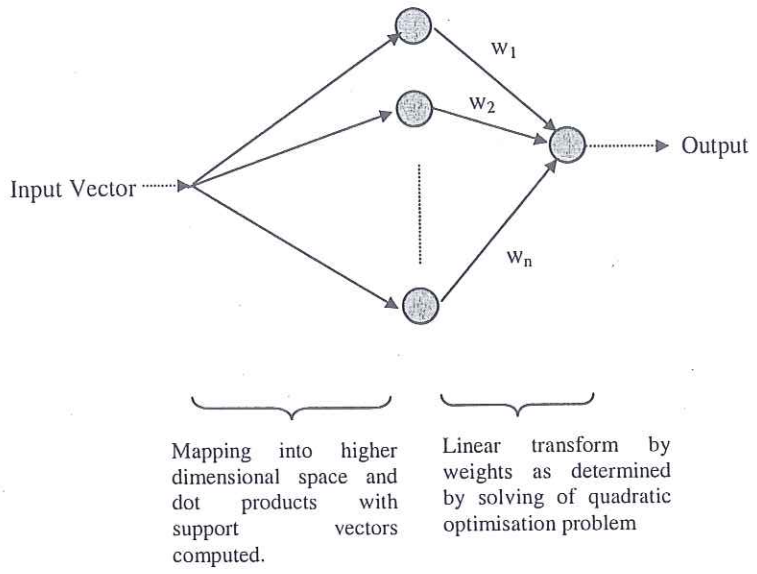
6.2.3 Non-Linear SVMs

So far only linear SVMs have been discussed, but many problems are non-linear. SVMs address this problem by mapping data non-linearly into a higher dimensional feature space where they are linearly separable. One problem with this is that performing calculations in the feature space can be extremely expensive computationally. However, this problem is solved in SVMs by carrying out the mapping using kernel functions. In effect, a kernel function of the input space is equivalent to the dot product in a higher dimensional space, and thus it is not necessary to be explicit about the transformation into the higher dimensional space, making it computationally feasible [Burges 1998]. Commonly used kernel functions include polynomial, radial basis, and sigmoidal functions. By using different kernel functions, SVMs can in effect mimic other types of networks (ie a SVM using a sigmoidal kernel mimics a MLP).

6.2.4 Structure of SVMs

Thus far the architecture of SVMs has not been discussed. They can in fact be thought of as having a structure similar to a multi-layer perceptron with a single hidden layer. The inputs are first mapped non-linearly into a higher dimensional feature space where dot products are computed (using the support vectors only). Thus the number of nodes in the hidden layer of a SVM is automatically determined by the number of support vectors (in contrast to MLPs where handcrafting of size may be necessary). These processes are carried out in a single step by the kernel functions. The transformed values are then modified by a linear function to determine the output of the classifier. This linear combination involves the use of weights found by the solving of the constrained quadratic optimisation problem that finds the optimal hyperplane. The structure of a SVM is summarised in **figure 6.8**.

Figure 6.8 - SVM architecture



A broad introduction to SVM and MLP classification has now been provided. The remainder of this chapter talks more specifically about the classifiers used in this work (**sections 6.4** & **6.5**) and the data used to train them (**section 6.3**).

6.3 Expression Data & Data Representation

The MLPs and SVMs used in this work were trained and tested using the CMU-Pittsburgh AU-coded facial expression database (see **appendix 3**) that provided 253 examples of the six basic facial expressions. This expression set consisted of 57 sequences of the expression of happiness, 49 of sadness, 55 of surprise, 30 of disgust, 33 of anger, and 29 of fear. 70% of this set was used for training/validation of classifiers, with the remaining 30% used for testing. The training/validation set of 176 examples was then further divided into 140 examples for training and 36 for validation (a 80% to 20% split). The validation set was used in MLP training to determine when training should be terminated, whilst the validation set was used by the SVMs to determine optimal kernel parameters (see **section 8.5**).

Before the examples included in the CMU-Pittsburgh AU-coded facial expression database could be used for training/testing of classifiers, it was necessary to process the sequences such that they were in the format required by the system. This processing took the following form:

- Reduce frame rate of sequences from the 40fps of the original CMU-Pittsburgh AU-Coded database to the 4fps at which the completed system runs.

- Reduce the size of the faces in the images to the scale detectable by the face tracker and to that used by the expression recognition system (approximately 45x55 pixels in size). It should be noted that all images in the database were reduced to the same size, so some variation in head size is present due to normal human differences and the differing poses people adopted in the seats when filmed.

- Label the start frame of each sequence where facial expression begins.

The sequences were then used to generate motion pattern examples for each of the six basic emotions. In this system, the approach taken to generate data for the classifiers was fully automated. The face tracker described in **chapter 4** was used to obtain face locations and, due to its use of the ratio template algorithm with a spatial face template, provided a rough map of face feature locations. The MCGM was then used to obtain the facial motions according to this map. Thus, facial motion data was generated for training of classifiers without the need for manual labelling of facial feature locations, unlike many other systems developed in the past eg [Bartlett 1999b].

The facial expression sequence was processed by the face tracker and MCGM to generate motion data for use in classifier training and testing, with four consecutive frames of motion data being used to represent each example of a facial expression. As the overall system frame rate was 4fps these four frames represented one second of

facial motion and consisted only of the start phase of a facial expression (ie from neutral face to expressive face).

Prior to entry of this motion data into the classifiers used by the expression recognition system, two general approaches were taken to condense and modify the data. These were the averaging of motion data and the taking of ratios of averaged motion:

- **Motion Averaging** - Rather than inputting the raw optical flow output directly into the classifiers, the motion (speed and direction) data generated by the MCGM is condensed into a more efficient form. Such condensation of data reduces the amount of information entered into the classifiers making the classification problem more simple. In terms of MLPs, a smaller size can therefore be used to solve the problem. The smaller the network size, the faster the output can be obtained, and the better for use in real-time systems. Different approaches to motion averaging are used in this work and are discussed as they are used in subsequent chapters. Region averaging also helps give robustness to the small inaccuracies in localisation of facial features by the face tracker. The face tracker only gives a rough map of feature positions and so can sometimes be 1 or 2 pixels away from being perfectly centred on the face. By averaging motion over large numbers of pixels it is thought the effects of the localisation inaccuracies are reduced.

- **Ratios of averaged motion** - A problem that has to be addressed to permit effective expression recognition is the removal of the effects of overall head motion. When someone moves the head as a whole whilst expressing an emotion, a dramatic change is seen in the optical flow output. For example, the optical flow of someone smiling whilst moving the head downwards would look very different from that of someone smiling whilst raising the head. One approach used to cancel this motion out was proposed by Lien [1998b] who subtracted the motion from data points in the face that are not deformed by facial expression. However, a different approach is taken here

facial motion and consisted only of the start phase of a facial expression (ie from neutral face to expressive face).

Prior to entry of this motion data into the classifiers used by the expression recognition system, two general approaches were taken to condense and modify the data. These were the averaging of motion data and the taking of ratios of averaged motion:

- **Motion Averaging** - Rather than inputting the raw optical flow output directly into the classifiers, the motion (speed and direction) data generated by the MCGM is condensed into a more efficient form. Such condensation of data reduces the amount of information entered into the classifiers making the classification problem more simple. In terms of MLPs, a smaller size can therefore be used to solve the problem. The smaller the network size, the faster the output can be obtained, and the better for use in real-time systems. Different approaches to motion averaging are used in this work and are discussed as they are used in subsequent chapters. Region averaging also helps give robustness to the small inaccuracies in localisation of facial features by the face tracker. The face tracker only gives a rough map of feature positions and so can sometimes be 1 or 2 pixels away from being perfectly centred on the face. By averaging motion over large numbers of pixels it is thought the effects of the localisation inaccuracies are reduced.

- **Ratios of averaged motion** - A problem that has to be addressed to permit effective expression recognition is the removal of the effects of overall head motion. When someone moves the head as a whole whilst expressing an emotion, a dramatic change is seen in the optical flow output. For example, the optical flow of someone smiling whilst moving the head downwards would look very different from that of someone smiling whilst raising the head. One approach used to cancel this motion out was proposed by Lien [1998b] who subtracted the motion from data points in the face that are not deformed by facial expression. However, a different approach is taken here

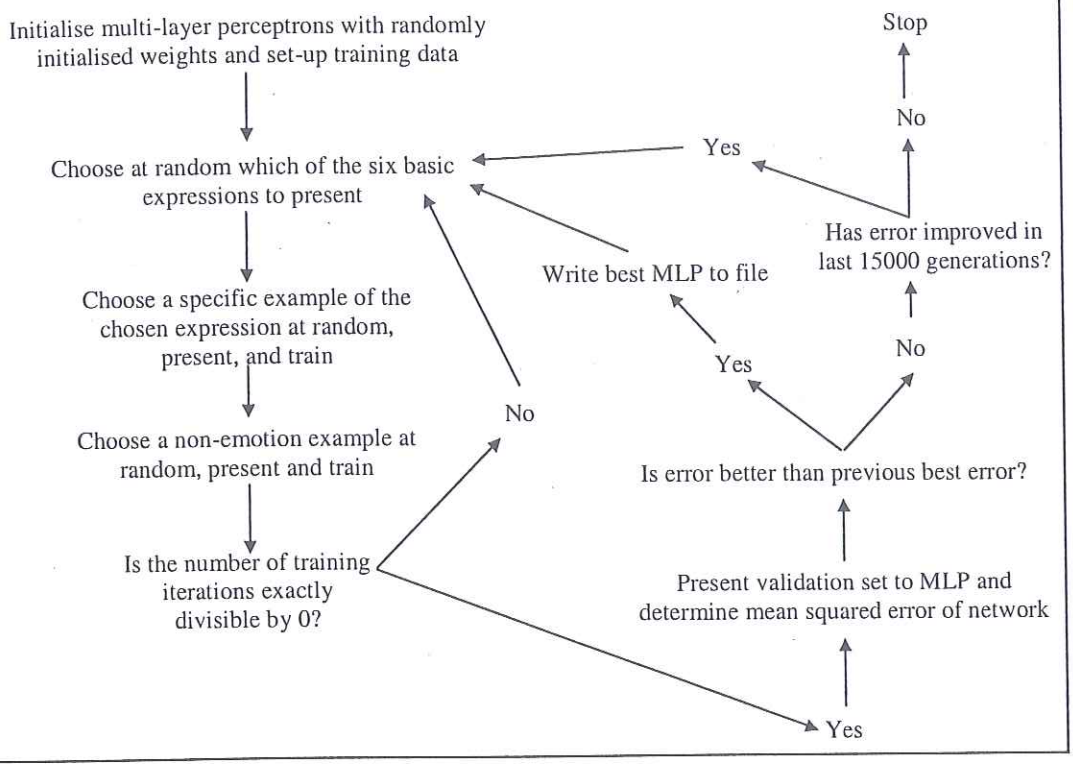
to remove the effect of rigid head motion, by the taking of motion ratios e.g. ratio of left cheek motion to left side of chin. By taking ratios it is possible to determine how different facial parts are moving relative to one another, independently of how the head is moving globally. **Appendix 5** summarises this approach. Please note that as well as cancelling out rigid head motion, the taking of ratios may also help compress the motion information into a more condensed form (a single direction value, as determined by the motion ratio calculation, is calculated with information regarding the speed and direction of movement of 2 separate regions).

6.4 MLPs for Expression Recognition

The MLPs used in this work were fully connected, used the logistic activation function and were trained using standard back propagation. Momentum (see **appendix 1**) and bias were also used. Although there were more examples of happiness than for, say, anger the training procedure for MLPs randomly selected examples from the training set such that there was an equal probability of any one type of facial expression example being presented at any one training iteration. The training strategy taken for the MLPs in this work is given in **figure 6.9**.

The trained MLPs were then tested on a test set of 77 expression examples and a 1440 frame long sequence of non-expression examples (see **section 7.1** in next chapter for description of the non-expression sequence). These test examples consisted of sequences and faces not seen before in the training or validation sets and therefore were an effective means of testing for generalisation. ROC curves (discussed later in **section 6.6**) were then used to evaluate performance.

Figure 6.9 – Training Procedure of MLPs using Back Propagation



6.5 SVMs for Expression Recognition

For the work presented here, SVM^{light} Version 5.00, an implementation of Vapnik's Support Vector Machine [Vapnik 1995], was chosen for the purposes of SVM classification [Joachims 1999a]. SVM^{light} is a binary classifier and was chosen as it can handle problems involving many thousands of support vectors, converges fast, and has minimal memory requirements. It has already been used by a range of researchers to efficiently solve a host of problems, such as text classification [Joachims 1999b], patient monitoring [Morik 1999], and protein-protein folding prediction [Bock & Gough 2001].

SVM^{light} supports the use of several basic kernels (linear, polynomial, radial basis function, sigmoidal) and also allows the users to define their own kernels if so required. The default kernels are defined as follows:

- *Linear:* $K(x_i,x_j) = x_i x_j$

- *Polynomial:* $K(x_i,x_j) = (ax_ix_j+b)^c$
- *Radial Basis Function:* $K(x_i,x_j) = exp(-a||x_i-x_j||^2)$, $a>0$
- *Sigmoidal:* $K(x_i,x_j) = tanh(ax_ix_j+b)$

Where a, b, & c are all adjustable kernel parameters.

Another feature of SVM^{light} is that it allows one to train the SVMs with cost models. Thus SVM^{light} can effectively handle unbalanced numbers of positive and negative examples by adjusting the relative weightings of training errors on positive and negative examples. For example, if a data set consists of 10 positive examples and 1000 negative examples, it is desirable to make modifications such that individual training errors in the positive training set are weighted more highly than those in the negative. SVM^{light} is available for download at *svmlight.joachims.org*.

The models produced by SVM^{light} are binary classifiers and thus it was necessary here, where we have six facial expressions to classify, to merge the outputs of multiple different binary classifiers. Two commonly used approaches for multi-class SVM classification are the "1-against-all" and "1-against-1" strategies.

For a problem consisting of k classes, the "one-against-all" approach (**figure 6.10a**) requires k SVM models. As the name implies, each binary classifier uses a single class as the positive training example, and all other classes as the negative training examples. By training k models and obtaining the output from each, the winning class is the model that outputs the greatest value from the decision function.

The "one-against-one" approach (**figure 6.10b**) requires k(k-1)/2 classifiers (where k is the number of classes), with each classifier training data from two of the different classes. In effect there is a classifier comparing each class with every other class. So when classifying the input data, the data is entered into each SVM in turn and a vote is cast for the winning class of each SVM. The class with the highest number of votes at the end is labelled the winner. If two classes have identical numbers of votes, the output of the binary classifier comparing the two classes determines the winner. A comparison of different methods for multiclass SVMs is provided by Hsu & Lin [2002].

Both the "1-against-all" and "1-against-1" strategies are used in work presented later in the thesis and the SVMs were trained and tested using sets of data identical to those used by the MLPs.

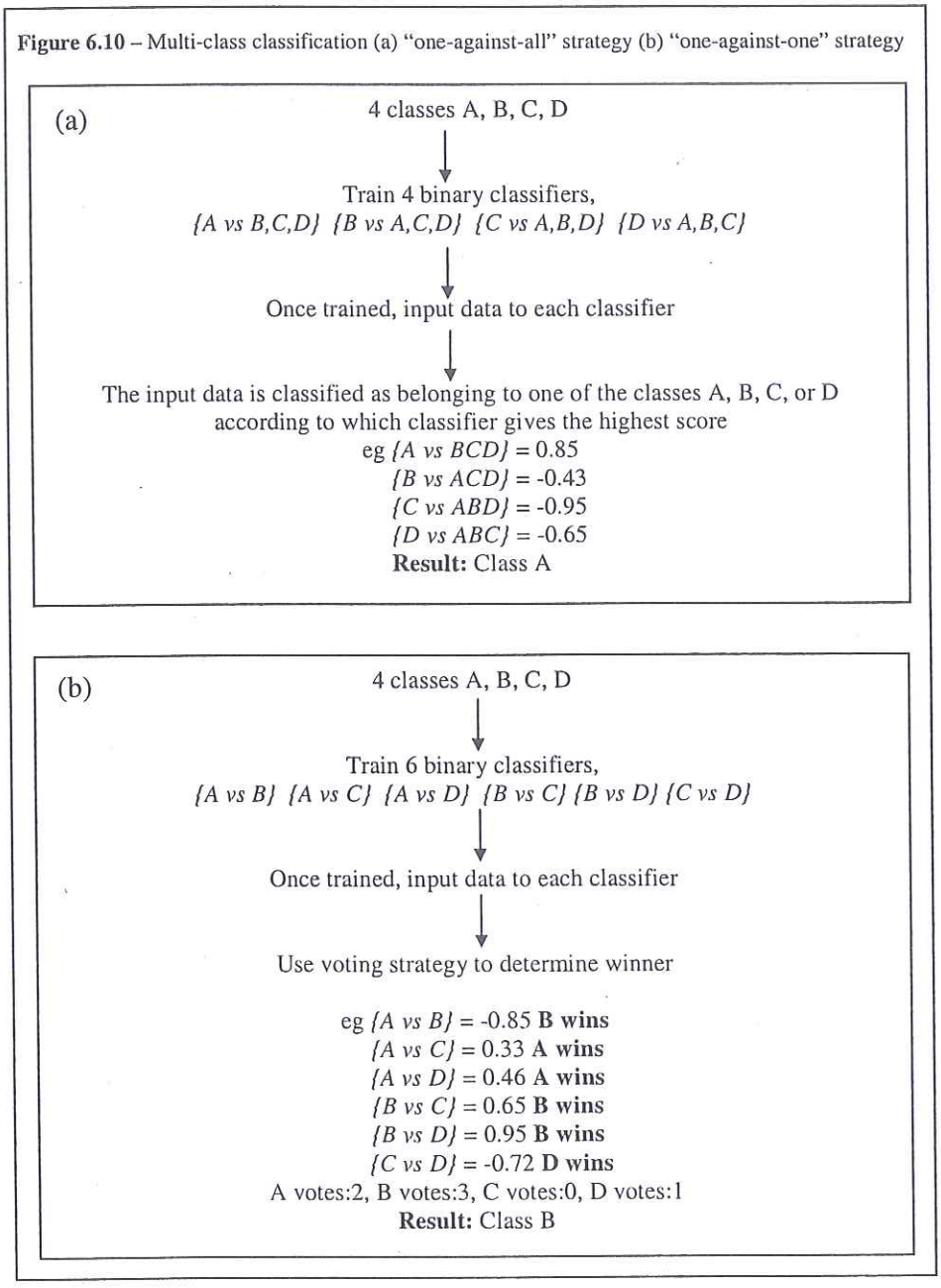


Figure 6.10 – Multi-class classification (a) "one-against-all" strategy (b) "one-against-one" strategy

6.6 Presentation of Results

As the system not only has to recognise facial expressions, but also recognise when a motion sequence is not a facial expression, thresholding needs to be introduced so that a classifier output above a certain level is classified as an expression, whilst below this level it is classified as a non-expression. However, the level at which the threshold is set can have a dramatic role in the performance of the expression recognition system.

Both the MLP and SVM approaches output a value for each expression following introduction of input data. Adjustment of the value at which the outputs are thresholded (as either an expression or not) alters the probability that the system will fire. For instance, consider a classifier giving the following response to an input representing an expression of surprise:

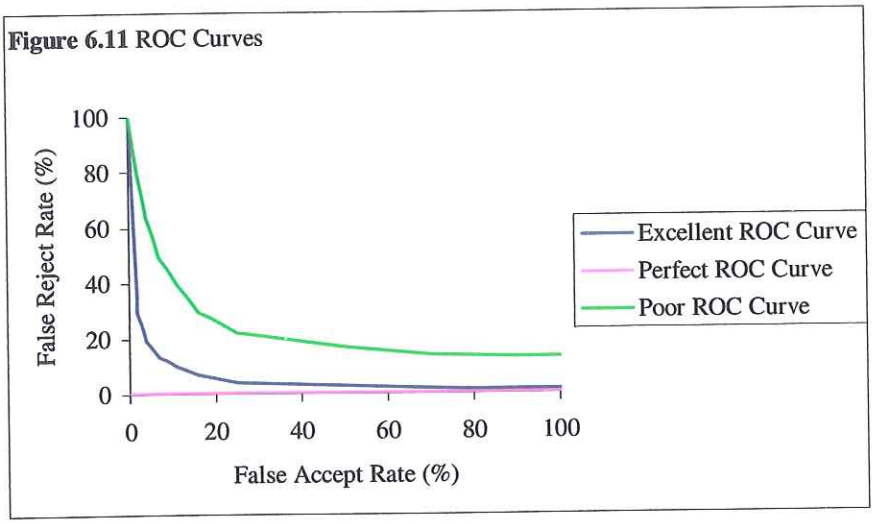
Happiness: 0.05, Surprise: 0.84, Sadness: 0.12, Disgust: 0.02, Fear: 0.01, Anger: 0.07

If the threshold were set at 0.80 then the system would correctly label the sequence as surprise, whereas if the threshold were set at 0.90 then the system would incorrectly label the sequence as not an expression.

Thus, a low threshold causes the system to fire more frequently, but has the undesirable attribute that it would be more likely to fire when a non-expression example is input. Alternatively, a high threshold causes the system to fire less frequently. The probability of firing when a non-expression example is input is thus reduced, but at the cost of making the system more likely not to fire when an actual expression is seen. Additionally, different classifiers are affected by thresholds in different ways so, for instance, a threshold of 0.4 may be suitable for one classifier, whilst 0.7 could be suitable for another.

Therefore, it is important to characterise the system independently of the threshold. For this purpose, it has been decided to use Receiver Operating Characteristics (ROC) graphs to characterise the system [Fawcett 2003]. These graphs plot false reject rates (FRR) against false accept rates (FAR). The FRR is the percentage of positive examples (actual expressions) incorrectly classified as negative examples (non-expressions), whilst the FAR is the percentage of negative examples (non expressions) incorrectly classified as positive examples (actual expressions). This is a useful approach as it allows one to compare two classifiers at all FRRs and FARs (ie all threshold levels at once). Example ROC curves are given in **figure 6.11**.

Figure 6.11 ROC Curves



The perfect ROC curve is shown in pink. This curve suggests that, no matter what thresholds are used in the classifier, it would never classify a non-expression example as an expression, whilst it never rejects an actual expression example and says it is not an expression. However, realistically such a result is not achievable. Rather, it is possible to improve the FRR (NB the lower the FRR the better) but at a cost of degrading the FAR (NB the higher the FAR the worse as it means more non-expression examples are classified as expressions). The blue curve in **figure 6.11** is an example of the type of ROC curve a good classifier would have, with low FRRs achieved at relatively low FARs. A poor classifier (green curve) on the other hand would never achieve low FRRs and also the FRR would decrease at a lower rate as the FAR increases.

In addition to the use of the 2-dimensional representation of performance provided by a ROC curve, it was decided to reduce this depiction to a single figure indicative of classifier performance. A commonly used strategy of calculating the area under the ROC curve (AUC) was therefore adopted [Bradley 1997]. However, a slight modification was made to the usual approach, as it was thought that the values of importance lay where the FAR was below 5%. Performance of the classifiers at levels of FAR greater than 5% was not of interest. At such levels the classifiers would be firing at a rate higher than once every 20 frames, regardless of whether any facial emotions were being expressed. The AUC value was calculated using the values of FRR at 20 evenly spaced intervals (every 0.25%) between FAR values of 0.25% and 5% and averaging them. The lower the resultant figure, the better the performance of the classifier.

Finally, as well as the ROC curves and their corresponding AUC values, one final value has been provided to help characterise each classifier. This value is the absolute recognition rate, defined as the percentage of expressions correctly classified by the classifiers, independent of thresholds (in effect the maximal recognition rate). A response is labelled a correct classification if the SVM or MLP node giving the strongest response to an input signal matches the class of the input signal (ie if the happiness node of a MLP gives the strongest response to a happiness input signal then this is a correct classification).

6.7 Summary

This chapter has provided a broad introduction to the MLP and SVM approaches used in this work for classification, also describing some of the problems associated with them. It has provided an introduction to, and justifications for, the use of the data representations used for expression classification in the following two chapters. Finally it has described how classifier performance will be evaluated to select those classifiers performing the best in the subsequent chapters.

7 MULTI-LAYER PERCEPTRON EXPRESSION CLASSIFICATION

The first approach taken to classifying facial expression involved the use of multi-layer perceptrons trained using back propagation. This chapter describes the empirical process followed to find the best data representation and MLP architecture for expression classification. Results are presented and discussed at each stage of this process. **Section 7.1** demonstrates the importance of the inclusion of non-expression examples in the training set, and this is followed in **section 7.2** by a comparison between performance achieved when data is normalised prior to entry into MLPs and when raw, unnormalised data is used. Changes to the representation of the motion data prior to input into the networks are then examined in **sections 7.3, 7.4 & 7.5**. Finally, **section 7.6** studies the effects of modifying network architecture. Thus, the experiments in this chapter investigate the problem of expression recognition in the following order:

- Firstly a preliminary investigation is carried out to determine what data should be included in the training set and whether data normalisation is of importance.
- Then an attempt is made to improve performance on this training data by empirically modifying the motion averaging and ratio taking strategies.
- Finally, once the best data representation is found, performance is optimised for that data representation by modifying MLP architecture.

Once an optimal level of performance using MLPs is determined by the experiments in this chapter, it will be possible to compare and contrast with the optimal performance of SVMs (see **chapter 8**). Such comparison is necessary as the MLP and SVM approaches differ significantly from one another and thus it is unclear apriori which technique should be used. This comparison will then provide the necessary evidence to allow a choice to be made as to the best expression recognition approach to be used in the completed real-time expression recognition system.

7.1 Non-Expression Examples

For incorporation of the system into real-time applications, it is not only vital to label an input motion signal as the correct facial expression, but also important to know when the input data is not indicative of a facial expression, instead being caused by normal rigid head motion.

Therefore, rather than solely training the MLPs with different examples of facial expressions, a sequence of non-facial expressions was also included in the training set. This non-expression sequence was recorded with natural indoor illumination using the following procedure:

1) *Set up computer with XC-ST70 CCD monochrome camera positioned on top of monitor.*
2) *Ask subject to sit in the seat provided in front of the computer, with no restrictions being placed on the positioning of the seat (Head approximately ~60cm from camera).*
3) *Ask subject to use computer for a period of 2 minutes. The subjects were allowed to use the computer in any fashion they liked.*
4) *Subjects were recorded at a rate of 4fps, the speed at which the completed system runs.*
5) *Repeat for another nine subjects. Please note that subjects ranged from 21 to 40 in age, were European or Asian in race and were of mixed sex.*
6) *Process sequences offline using face tracker and optical flow algorithm to obtain raw head motion.*

The non-expression data obtained thus consisted of both rigid head motions and non-rigid head motions that did not fall into one of the six basic expression categories. By recording the subjects using the computer in a non-constrained manner these motions were representative of head motions seen when an individual used a computer naturally. The sequences of 7 of the 10 subjects were used for training/validation and the remaining 3 for testing.

To demonstrate the importance of including non-expression examples in the training set, an experiment was carried out to compare performance when non-expression examples were and were not included in the training set. Two identically structured MLPs were trained using identical procedures and with motion data processed in identical ways (NB the initial size of the MLPs was determined by some preliminary empirical investigation). The only difference between the two MLPs was the inclusion of non-expression examples in the training set for one of them. Absolute recognition rates and area under curve (AUC) values are given in **table 7.1**, whilst **figure 7.1** provides a ROC curve for performance comparison. **Figure 7.1** also summarises the approach taken to data representation and MLP architecture. The initial approach employs the regions of the spatial face template used by the ratio template algorithm for motion averaging. However, two regions were added in the chin region. This was necessary as the motion of the chin plays an important role in facial expression (eg chin drops in expression of surprise) but was not represented in the original spatial face template of the ratio template algorithm. The 18 motion ratios taken were determined empirically.

Table 7.1 Non-expression examples

Input data type	Generations to train	Absolute recognition rate %	AUC
Non-expression data not included	13300	54.55	90.6
Non-expression data included	9100	54.55	87.7

The best absolute recognition rates for both classifiers were identical at 54.55%, showing that the performance of the MLPs in correctly classifying facial expressions when non-expression examples are included is not degraded. However, upon examination of the ROC curve in **figure 7.1b**, it is evident that the inclusion of non-expression examples has led to improved FRRs at all equivalent FARs. The ROC curve of the classifier not trained with non-expression examples has an AUC of 90.6 whilst the AUC for the non-expression data classifier is 87.7. This result could easily be predicted, and is explained in **figure 7.2**. Training time was increased when non-expression examples were included but the performance benefits make this worth it. In all subsequent experiments, non-expression examples were included as part of the training and validation sets.

Figure 7.1 – Inclusion of non-expression example in training set for MLPs (a) Summarises network architecture and representation of data to be entered into networks. These are identical for both MLPs trained in this experiment (b) ROC curve showing effect on inclusion on FAR and FRR

(a)

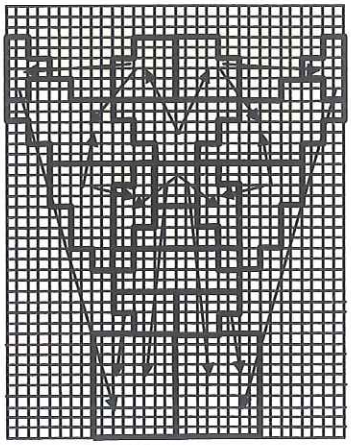
Network Size: 72, 200, 100, 6

Momentum: 0.5

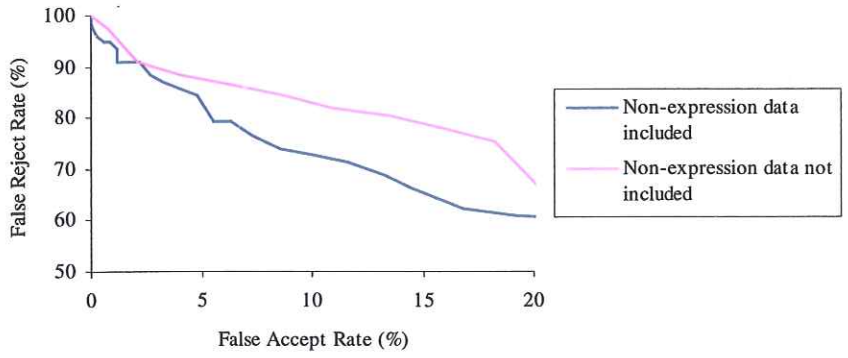
No. Regions: 18

No Ratios: 18

Data Type: Raw motion direction information in range 0-360



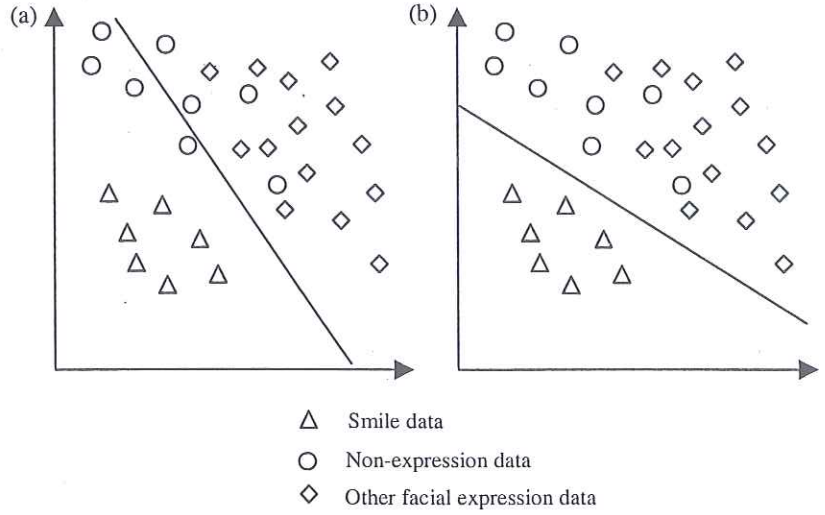
(b)



7.2 Data Normalisation

In general, prior to entry of an input vector into a MLP, it is desirable to normalise the values. The MLPs used in this work make use of the logistic activation function (**figure 6.3**). This function is centred at zero and thus is especially sensitive around this point. Therefore, it is helpful to reposition input data to the range over which this activation function is sensitive. **Figure 7.3** demonstrates the importance of data normalisation.

Figure 7.2 – Effect of including non-expression data in training set (a) Non-expression data not included in training set. MLP effectively separates the two classes included in training set, smile data (triangle) and other facial expression data (diamond). However, it misclassifies some non-expression data points (circles), with some of these data points on the wrong side of the decision boundary (b) Non-expression data included in training set. MLP effectively separates smile data from both other facial expression data and non-expression data. The non-expression data is included, resulting in a change in position of the decision boundary.



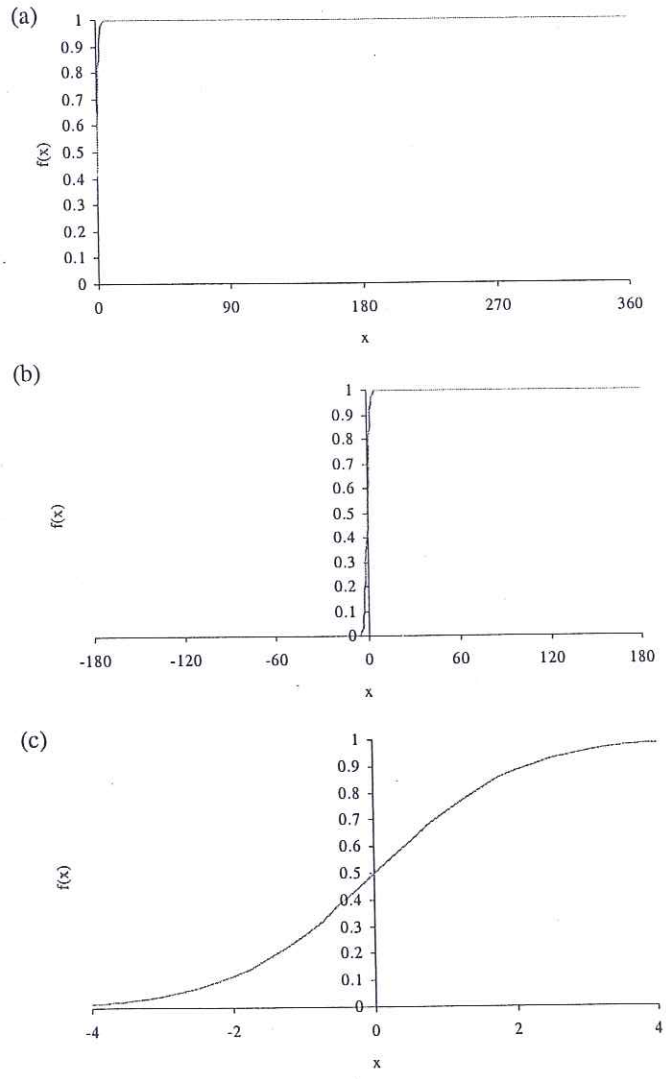
The experiment described here was set up to test the hypothesis that normalisation of data would lead to improved recognition rates and better ROC curves. The architecture of the MLPs and the initial pre-processing of motion data were as in the previous experiment (**figure 7.1a**). However, this time the data was normalised prior to entry into the MLPs using an approach called standard deviation normalisation [Demuth 1998]. The inputs to each node were modified according to:

$$v'(i) = \frac{v(i) - mean(v)}{sd(v)}$$

where v is the feature, i is an instance of that feature and sd is the standard deviation.

By subtracting the mean value, the inputs for each node are centred around zero, and, by dividing by the standard deviation, the inputs are scaled to a standard deviation of 1.

Figure 7.3 – Importance of data normalisation (a) Initial motion direction data provided by MCGM is over range 0 – 360. When input into MLP, only half of the activation functions sigmoid is used. Thus (b) reposition data to range –180 to 180, thereby using whole of sigmoid. However, large number of data points positioned where activation function is insensitive eg change from an input of 60 to 180 leads to a tiny change in output of activation function. So (c) rescale data to a range over which the sigmoid is sensitive.

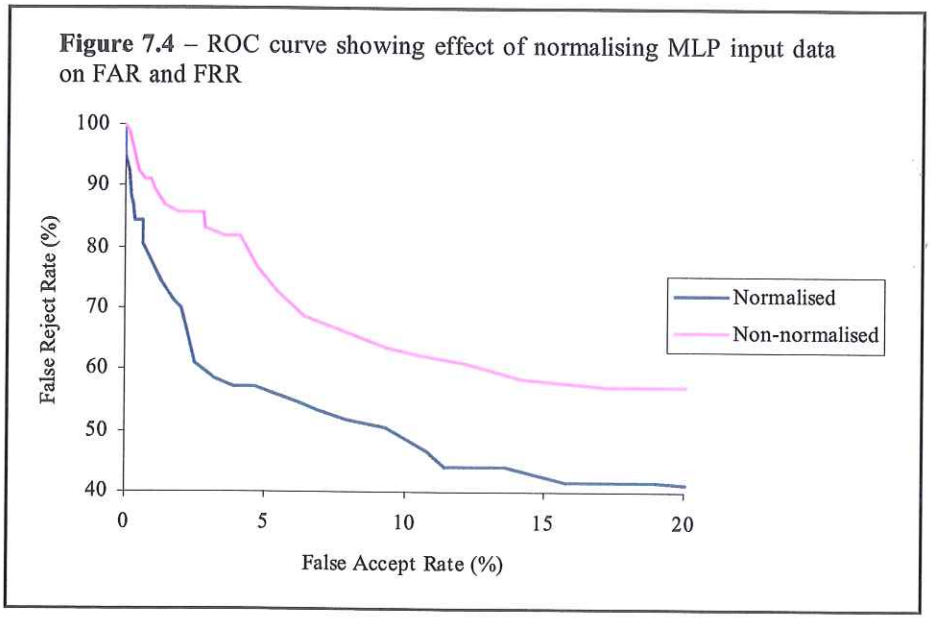


Runs of training were simulated to obtain a set of inputs indicative of that to which the MLPs were exposed during the training process (100000 trial generations of training were used to generate this set). The input data for each input node was then normalised according to the values obtained during the simulation. **Table 7.2** provides absolute recognition rates for the two approaches (normalised inputs and non-normalised inputs), whilst **figure 7.4** provides the relevant ROC curve.

Table 7.2 – Data normalisation

Input data type	Generations to train	Absolute recognition rate %	AUC
Non-normalised	13300	54.55	87.7
Normalised	19800	66.23	61.6

Figure 7.4 – ROC curve showing effect of normalising MLP input data on FAR and FRR



The results show a higher absolute recognition rate (~10% higher) and a much improved ROC curve when inputs are normalised prior to input into the MLPs. The FRR is lower at each equivalent FAR for the normalised MLP, with an AUC of 65.8, compared to the AUC value of 87.7 obtained with the raw non-normalised data. Thus, in all subsequent MLP experiments, data inputs were normalised prior to training of the networks.

The previous two sections examined what data should be included in the MLP training sets and the role of data normalisation. With the training set fixed and in the knowledge that data should be normalised to improve performance, it was then possible to investigate different motion representation strategies. The effect of using different ratios of motion is discussed first.

7.3 Asymmetry and Number of Ratios

There is a body of data suggesting emotions are more intensely expressed on the left-side of the face, although some research has suggested this to be the case only for negative emotions with no difference being seen when subjects smile [Sackheim et al 1979, Schwartz 1979].

Nonetheless, superficially at least, facial expressions are symmetrical and so it should be possible to accurately characterise an expression using only one side of the face. Thus it was considered that it might be advantageous if asymmetrical motion ratios were used for the expression recognition system, in effect reducing data by preventing the use of the same information twice.

By using asymmetrical ratios the motion data can be condensed. For example, when a person smiles, the ratio of the left mouth corner to the left side of the forehead would contain similar information to the ratio of the right mouth corner to the right side of the forehead. Thus, it should theoretically be advantageous to take only one of these ratios and then take a completely unrelated second ratio, for example a chin to right cheek ratio.

7.3.1 Asymmetry

An experiment was carried out to test this theory. Two different but entirely symmetrical sets of ratios were empirically determined and MLPs trained using these sets. The two sets were then merged into a single asymmetrical set and used to train a separate MLP. In all cases network architectures were kept the same, as were the number of inputs to the MLPs. The two symmetrical ratio sets consisted of 18 ratios (**figure 7.5a**). The asymmetrical set also consisted of 18 ratios. However, 9 of these

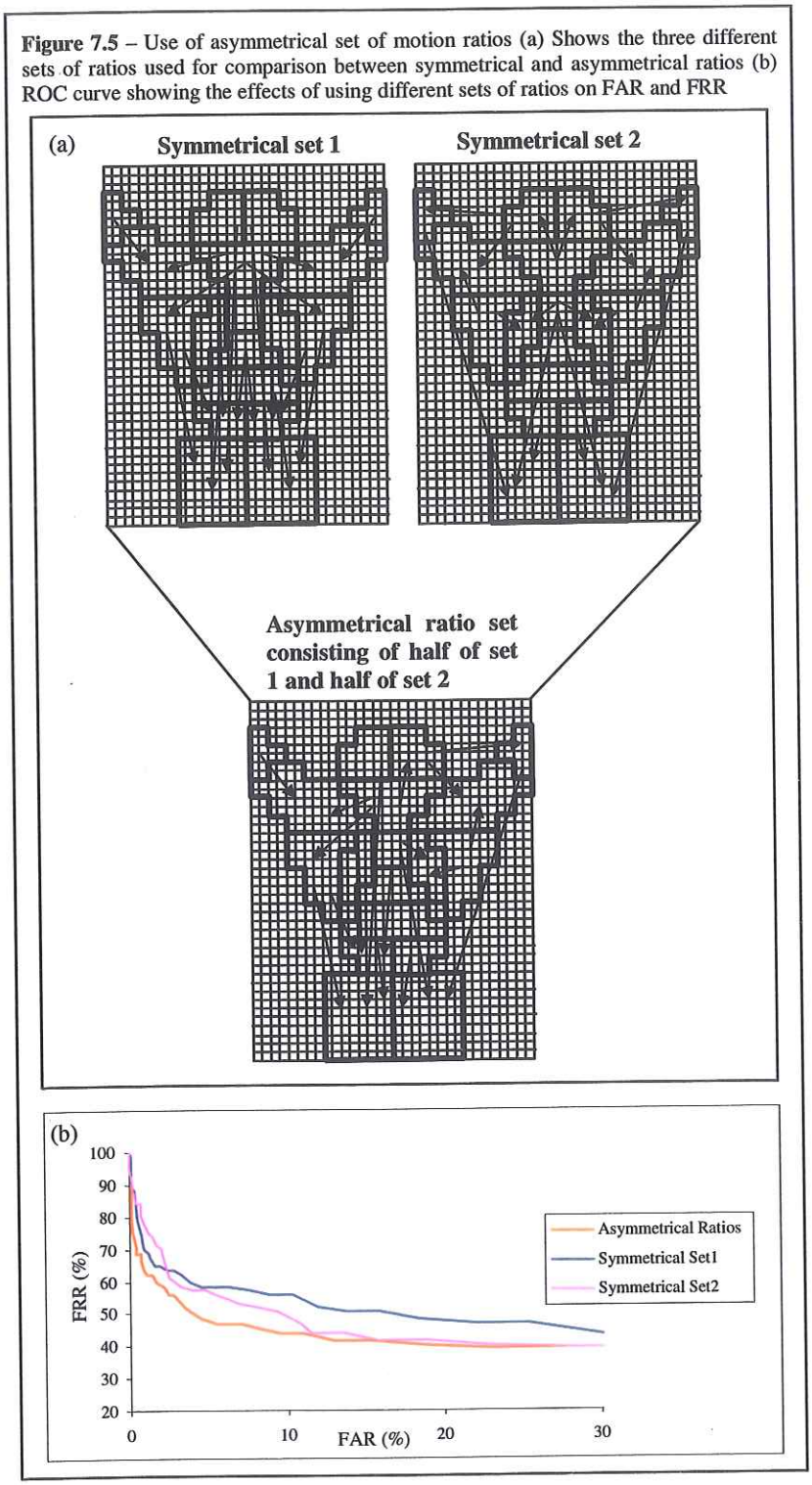


Figure 7.5 – Use of asymmetrical set of motion ratios (a) Shows the three different sets of ratios used for comparison between symmetrical and asymmetrical ratios (b) ROC curve showing the effects of using different sets of ratios on FAR and FRR

ratios were taken from symmetrical set one and 9 from symmetrical set two (**figure 7.5a** shows the resultant asymmetrical set). Absolute recognition rates are given in **table 7.3**

and a description of the ratio sets and a ROC curve is provided in **figure 7.5**.

Table 7.3 – Asymmetrical ratios

Input data type	Generations to train	Absolute recognition rate %	AUC
Symmetrical set 1	24000	63.64	64.4
Symmetrical set 2	11900	66.23	61.6
Asymmetrical set	12900	63.64	54.4

The results show fairly consistent absolute recognition rates of ~65%. However, the ROC curve appears better for the asymmetric ratio set and this is confirmed by the AUC score of 54.4, statistically lower than the AUC scores for the two symmetrical sets. This confirms the hypothesis made previously that use of asymmetric ratios can be more effective than using symmetric ratios, as it prevents duplication of information.

7.3.2 Intensity of Expression on Each Side of Face

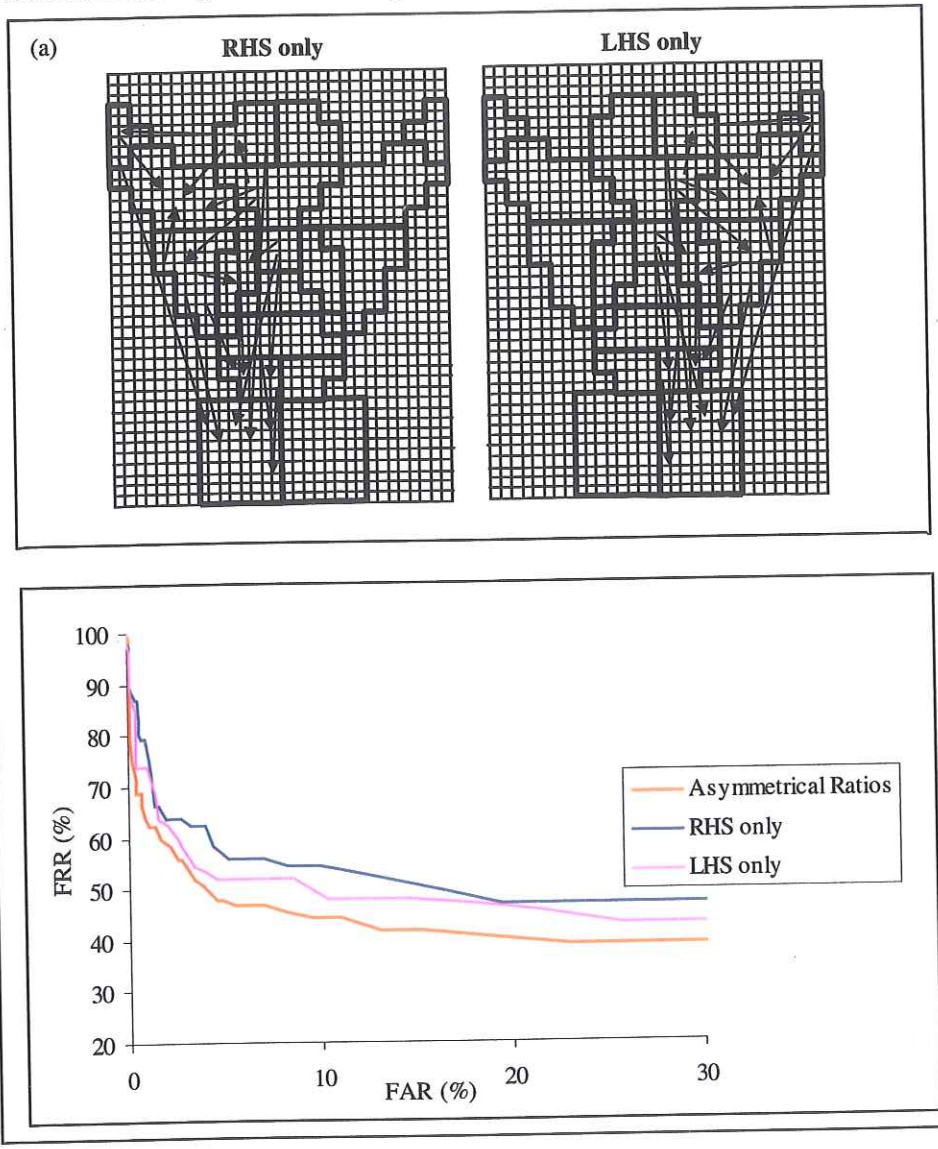
As previously discussed, research has suggested that some expressions are more intensely expressed on the left side of the face than on the right side of the face. It was thus decided to test whether using ratios solely on the left side of the face would be more effective than using an asymmetrical set of ratios for both sides of the face. Three sets of ratios were used in this experiment, the asymmetrical set of eighteen used previously and two sets using the same basic ratios but with these ratios all being moved to either the left side of the face or the right side of the face. The ratio sets used and results achieved are given in **figure 7.4** and **table 7.6**.

The results seem to confirm the finding that the left side of the face is more expressive than the right side, with slightly improved performance when left hand side ratios are used alone than when right hand side ratios are used alone (AUC of 58.1 compared to 63.3 and slightly higher absolute recognition rate). However, performance is still not up to that achieved when the eighteen ratios are split evenly between the two sides of the face. Thus, one can conclude that although the left side may be more expressive, the right side still contains important information and information from both sides of the face should therefore be used.

Table 7.4 – Full to half face comparison

Input data type	Generations to train	Absolute recognition rate %	AUC
LHS only	15100	58.44	58.1
RHS only	16000	57.14	63.3
Asymmetrical set	12900	63.63	54.4

Figure 7.6 – Comparison between asymmetrical motion ratios over whole face and ratios on half the face only (a) Shows the sets of ratios used on one side of face only. Same ratios as for asymmetrical set but moved all to one side. Asymmetric set is as in **figure 7.4** (b) ROC curve showing the effects of using these different sets of ratios on FAR and FRR



7.3.3 Asymmetry and Number of Ratios

Two issues have yet to be addressed:

1) Although a preliminary experiment has shown that using asymmetrical ratios is more effective than using the same number of symmetrical ratios, the use of all possible symmetrical ratios might be even better ie all 18 ratios from symmetrical set 1 & all 18 ratios from symmetrical set 2 (shown in ***figure 7.5a****) forming a new set of 36 ratios.*

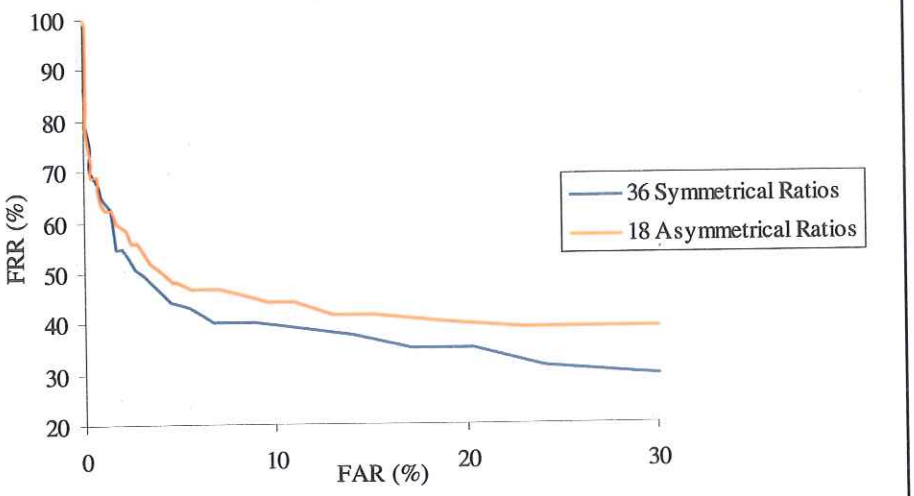
2) Thus far only 18 motion ratios have been used. This number of ratios was decided upon empirically, but there is no reason to believe that 18 is the optimal number of ratios to take.

To address issue 1) an experiment was carried out to compare the performance of the 18 asymmetrical ratio set and all 36 ratios from symmetrical sets 1 and 2 (**figure 7.5a**). This experiment would also indicate whether a higher number of motion ratios would give improved performance. The results are given in **figure 7.7** and **table 7.5**.

Table 7.5 – Symmetrical-asymmetrical comparison

Input data type	Generations to train	Absolute recognition rate %	AUC
All 36 ratios	15300	74.02	50.7
Asymmetrical set	12900	63.64	54.4

Figure 7.7 – ROC curve comparing results achieved when 36 symmetrical ratios are used and those achieved when an asymmetrical subset of 18 is used



The absolute recognition rate for the set of 36 ratios is over 10% higher than that achieved with the asymmetric set, and the AUC is better. It can therefore be concluded that the data representation using 36 ratios is better than the asymmetric set of 18. Thus, although asymmetric ratios can be used to condense information (see AUC in **table 7.3**), the performance when such sets are used is notably worse than when a full set of data is used.

The above experiment has also shown that 36 ratios is more effective than 18. Other experiments were thus carried out to see if an even greater number of ratios would improve performance further (results not shown here). However, this was found not to be the case, with performance staying virtually identical to that achieved by 36. This was thought to be because the information provided by 36 motion ratios adequately contains all the information that can be extracted from the averaged motion signal of the 18 regions of the ratio template algorithm's spatial face template. Addition of extra ratios beyond this number has no measurable effect, and just increases complexity.

7.3.4 Asymmetry Conclusions

The conclusions that can be drawn from the experiments carried out in this section may be summarised as follows:

1) Asymmetrical ratios can be used to condense motion data, working more efficiently than an equal number of symmetrical ratios when a small number of ratios are used. This could be useful in a real-time system where processing constraints have made it impossible to use a full set of ratios (NB this is not the case for the real-time system presented here).

2) Asymmetric ratios do not work as well as the equivalent symmetrical set that is double the size. Thus, a full symmetrical set should be used if processing constraints allow.

3) Expression recognition performance is not improved by adding more than 36 motion ratios. Thus it can be assumed that this number of ratios includes all the direction information that can be extracted from the raw averaged motions of 18 regions.

4) The motion of the left side of the face is marginally more useful for recognition of expression than the right side of the face.

7.4. Importance of Ratios

Having determined an optimal set of 36 ratios, it was decided to test whether the ratio taking approach degrades classification performance relative to the performance of a classifier using the raw averaged motion information.

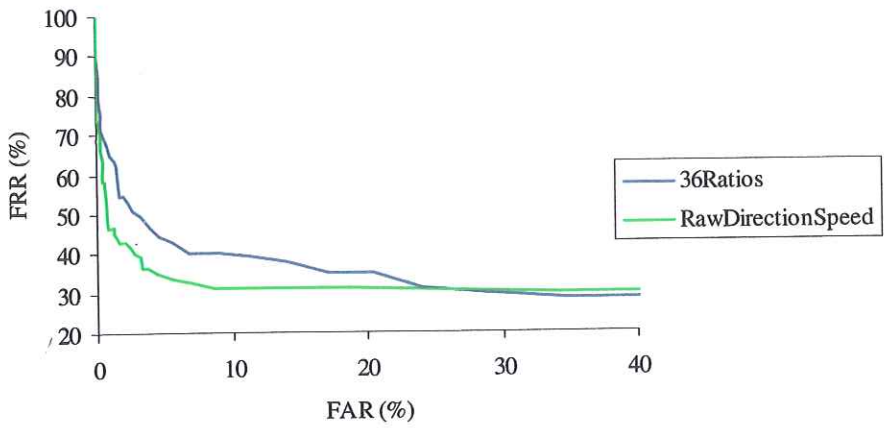
As discussed previously, motion ratios are used in this work primarily to cancel out rigid head motions. In theory, by taking enough ratios it should be possible to include all the information that is present in the raw averaged speed and direction data. However, to test this an experiment was carried out comparing performance of MLPs provided with the raw averaged data and data that had been modified using the ratio approach. The results are provided in **figure 7.8** and **table 7.6**. It should be noted that the test sequences taken from the CMU-Pittsburgh AU-Coded Database contain little or no rigid head motion. This experiment was thus carried out solely to demonstrate that

taking ratios did not degrade performance in a situation where no rigid head motion was present.

Table 7.6 – Effect of using motion ratios

Input data type	Generations to train	Absolute recognition rate %	AUC
36 Ratios	15300	74.02	50.7
Raw Averaged Motion	22600	71.43	41.5

Figure 7.8 – Effect of taking ratios on classifier performance



As can be seen, the performance of the classifier using ratios of motion data was significantly worse than that achieved by the classifiers using the original averaged motion data. The AUC value was 41.5 for the classifier using the raw data compared with 50.7 for the classifier using motion ratios.

Both classifiers were provided with the same number of inputs (144), 36 for each of the four frames frame of motion input. The 36 inputs for the classifier using raw averaged data consisted of the 18 speed and 18 direction values obtained from averaging motion over the 18 regions of the face template. To keep the number of inputs the same for the ratio-based approach, the 36 inputs for the classifier using ratios of averaged motion consisted of the direction values only. It was thought that speed information could be omitted in the ratio-based case as the direction value of the ratios

was dependent on both the speed and direction of movement of the constituent regions. However, the result presented above show this not to be the case.

Although in the case where two regions move in different directions the speed plays an important part in the overall direction of the motion ratio, when two regions are moving in the same direction the speed has no effect on the final direction of the motion ratio. Hence, speed has a varying role to play in the direction value of a motion ratio calculation depending on the relative direction of motions of the two constituent regions. So, by not using the speed information provided by the ratios, some information can be lost, and thus the performance of the ratio-based approach using 36 input values based on direction alone is degraded relative to the approach using 36 inputs based on the raw speed and direction values of averaged motion. It was therefore decided to also provide speed information to the classifiers using motion ratios, thereby doubling the number of MLP inputs. The results are given in **figure 7.9** and **table 7.7** and compared to the results obtained when the raw averaged motion data is used.

The ROC curves of **figure 7.9** demonstrate that inclusion of speed information allows the ratio approach to work as effectively as the approach using raw averaged motion data (AUC of 42.2 compared to 41.5). However, these experiments have demonstrated that there is no potential for using motion ratios to help compress the motion data (as proposed in **section 6.3**). Rather, the opposite is true, as, for the dataset used with no rigid head motion, to achieve identical performance to that obtained by the raw averaged data, it is necessary to provide classifiers with twice as many inputs. Nonetheless, motion ratios are used throughout the rest of this work due to the robustness they provide to rigid head motion

Table 7.7 - Effect of using speed and direction information

Input data type	Generations to train	Absolute recognition rate %	AUC
36 Ratios Direction Only	15300	74.02	50.7
36 Ratios Direction & Speed	41400	71.43	42.2
Raw Averaged Motion	22600	71.43	41.5

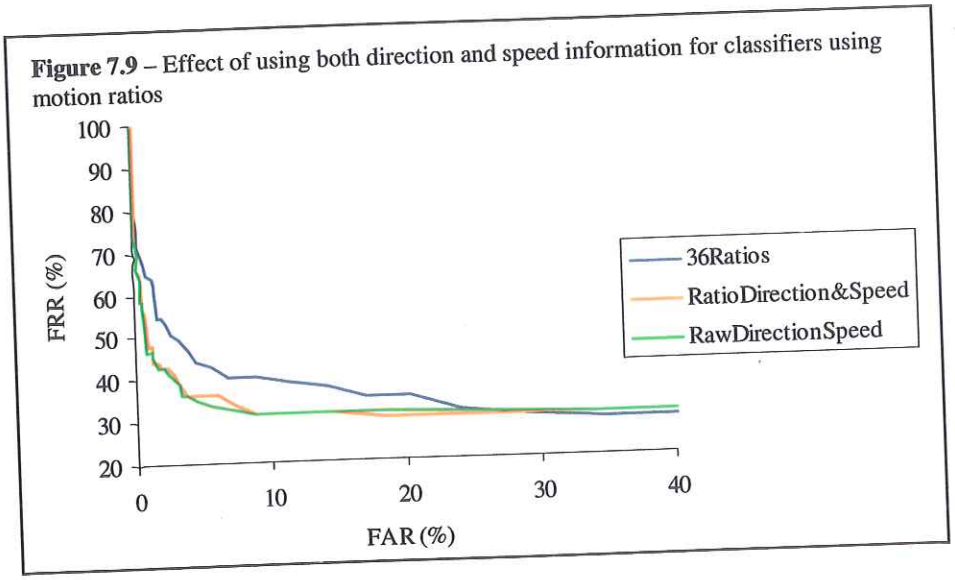


Figure 7.9 – Effect of using both direction and speed information for classifiers using motion ratios

7.5 Modifying Regions for Averaging

Thus far, the regions selected for averaging motion have been based solely on those regions used by the spatial template employed by the system's face tracker. However, the choice of regions over which to do the spatial averaging is important, as otherwise important data relating to salient face emotional movement can be lost. Ideally one wants to choose regions for averaging such that the face is separated into a set of individual parts moving in a concerted manner when different emotions are expressed. It was thought that the use of co-articulation regions would be ideal for this purpose.

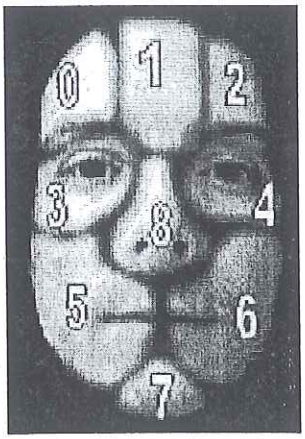
7.5.1 Co-Articulation Regions

The human face consists of a number of muscle groups, each able to contract independently (see **figure 2.1**). Contraction of these muscles causes localised motion of continuous areas of skin on the face, with the region of skin affected in this way being called the muscle's region of influence (ROI). Rather than each muscle group having it's own separate ROI, they in fact overlap one another. Therefore, if two muscle groups with overlapping ROI contract together, the changes seen on the skin surface are due to a combination of the affects of the contracting muscles.

The work of Fidaleo and Neumann [2002] takes advantage of this property by modelling the changes on the skin surface as a set of 9 contiguous regions of skin deformation called co-articulation regions. This approach was used to control the animation of 2D cartoon characters [Fidaleo 2002], and the regions produced, along with the muscles groups responsible for each regions motion, are given in **figure 7.10**.

Figure 7.10 – Co-articulation regions (a) the regions of the nine co-articulation regions (CR), taken from [Fidaleo 2002] (b) the muscles that influence the motion in each co-articulation region

(a)



(b)

CR 0 – *Frontalis L*

CR 1 – *Frontalis C, Corrugator*

CR 2 – Frontalis R

CR 3 – *Orbicularis Occuli L, Levator Palpebrae L*

CR 4 – *Orbicularis Occuli R, Levator Palpebrae R*

CR 5 – *Zygomatic Major L, Risorius L, Triangularis L*

CR 6 – *Zygomatic MajorR, Risorius R, Triangularis R*

CR 7 – *Mentalis*

CR 8 – *Levator Nasi*

It was thought that the use of these co-articulation regions could enhance performance for the expression recognition task, as it describes parts of the face that move together when expressions are made. Thus, a new set of regions was developed based on the co-articulation regions of **figure 7.10**. **Figure 7.11** shows the new regions set up for averaging based on the work of Fidaleo.

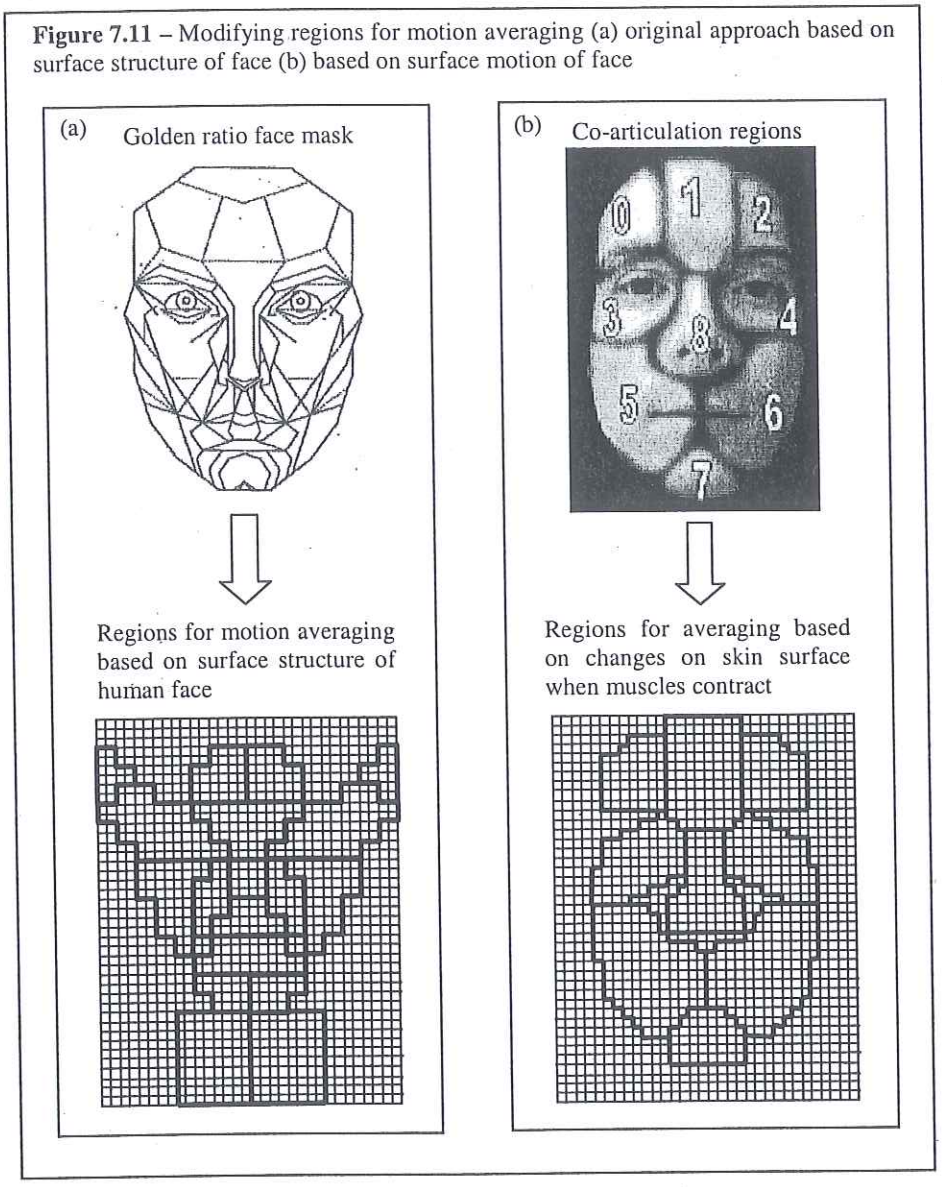


Figure 7.11 – Modifying regions for motion averaging (a) original approach based on surface structure of face (b) based on surface motion of face

An experiment was carried out to compare the results achieved by the new set of face averaging regions with those achieved when the face tracker's facial template was used. Information that has been learnt in previous sections is carried forward, so non-expression data is included in the training sets, data is normalised, and symmetrical ratio sets used.

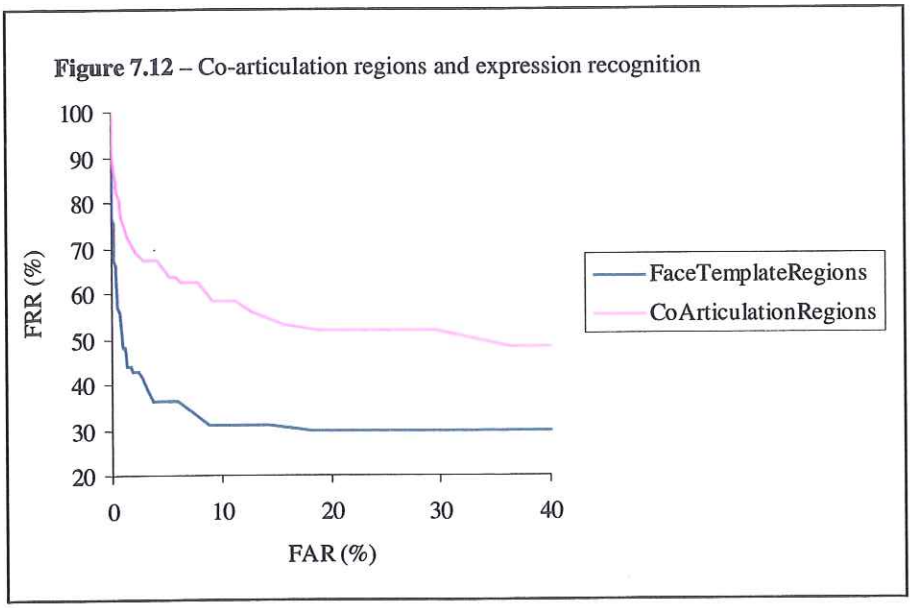
Twenty-seven symmetrical ratios were chosen for the new co-articulation based regions. This set of 27 ratios linked every region to every other region without crossing

boundaries from one side of the face to the other, and thus is the maximum number of ratios possible with a set consisting of nine regions. MLPs were then trained using this new data representation and results are given in **table 7.8** and **figure 7.12**.

Table 7.8 – Co-articulation regions

Input data type	Generations to train	Absolute recognition rate %	AUC
Face Tracker Template with 36 ratios	41400	71.43	42.2
Co-articulation regions with 27 ratios	27500	53.24	70.5

Figure 7.12 – Co-articulation regions and expression recognition



The results show that performance is much worse when the co-articulation regions are used, with an AUC of 70.5 (compared with 42.2) and an absolute recognition rate ~20% below the previous best figure.

7.5.2 Extension to Co-articulation Regions

One reason why averaging using the co-articulation regions was a relative failure was the lack of information they provided about the motion of the mouth, with the regions 5 & 6 (see **figure 7.11**) each covering the whole cheek and mouth regions. Therefore three mouth regions were added as shown in **figure 7.13**, with these mouth regions being identical to the mouth regions used by the face detectors facial template.

A set of 36 symmetrical ratios was then chosen empirically (**figure 7.13**), MLPs trained, and results obtained. These results can be found in **figure 7.14** and **table 7.9**.

Figure 7.13 Co-articulation based averaging with added regions in mouth area and the symmetrical set of 36 ratios

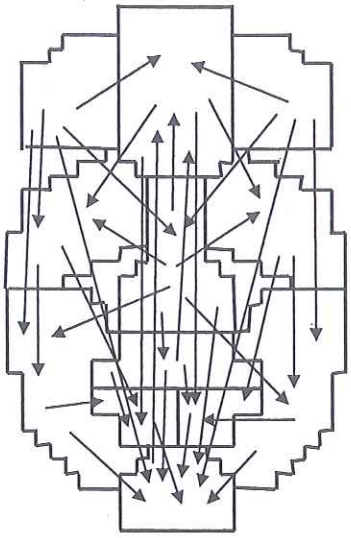
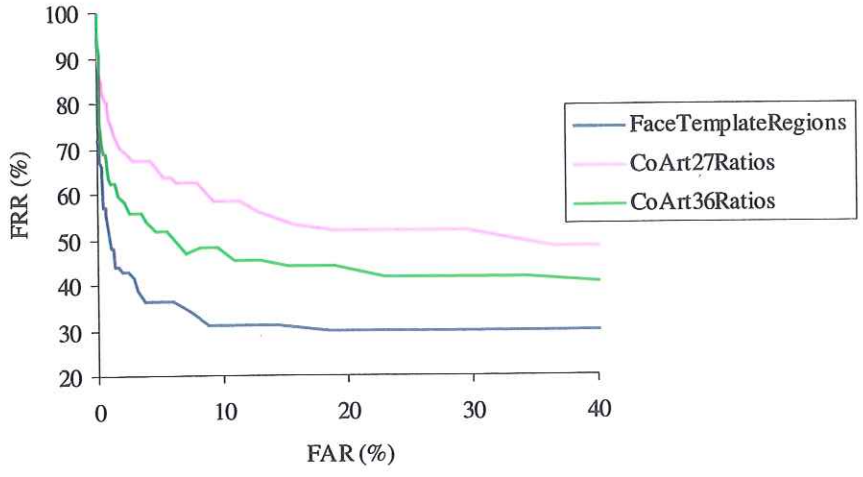


Table 7.9 – Extension to co-articulation regions

Input Data Type	Generations to train	Absolute recognition rate %	AUC
Face tracker template with 36 ratios	41400	71.43	42.2
Co-articulation regions with 27 ratios	27500	53.24	70.5
Co-art regions with mouth parts and 36 ratios	31200	63.64	58.0

Figure 7.14 – Effects of adding mouth regions to original co-articulation regions



Although addition of the specific mouth regions improved performance of the co-articulation approach (from an AUC of 70.5 to 58.0), it is still significantly worse than that achieved when averaging according to the regions of the spatial face template. It is thought that this is because, although the face template does not directly correspond to the muscular structure of the face, it does still model locations of the surface structure of the face, for which the musculature is partially responsible. Also, there are 18 regions in this template compared to the 12 regions of the extended co-articulation regions. Thus there is greater detail included in the ratio template algorithm's spatial face template with 18 averaged direction and speed values provided rather than with just 12. These factors probably account for the decreased performance achieved when averaging is based on co-articulation regions. From these experiments it can therefore be concluded that for the expression recognition task presented here, use of the ratio template algorithms golden ratio face template is more effective than the use of co-articulation regions.

7.6 Network Size & Architecture

As discussed in **section 6.1.8.2**, MLPs are sensitive to their architectural size, with performance being degraded when they are either too large or too small. It was thus decided to use the best data representation found thus far (36 ratios of averaged motion using ratio template algorithm's spatial template) and use different network sizes and architectures to see whether results could be further improved or, if not, to at least find the minimum sized network that provided the desired results. It should be remembered that the current MLP architecture of 2 hidden layers with a size of 288x200x100x6 was determined by some preliminary experiments before the current data representation was introduced.

Two experiments were carried out, the first to find the optimal size using a single MLP for expression recognition and the other to try using six single expression recognising MLPs (ie a separate MLP to recognise each of the 6 basic expressions). This was necessary not only to see if such an approach improved results, but also to allow fair comparison with the SVMs generated by SVM^{light} (**chapter 8**), as SVM^{light} generates only binary classifiers.

7.6.1 Size Using a Single Expression Recognising MLP

A number of differently MLPs were trained, some with a single hidden layer and others with 2 hidden layers. The results achieved when a single hidden layer is used are presented in **figure 7.15** and **table 7.10**, whilst for MLPs with two hidden layers the results are presented in **figure 7.16** and **table 7.11.**

Table 7.10 – Size and single MLP with one hidden layer

MLP Size	Generations to train	Absolute recognition rate %	AUC
288x50x6	46300	70.13	43.5
288x100x6	35100	81.82	37.5
288x200x6	20300	74.02	45.5
288x400x6	20600	70.13	39.4
288x800x6	29200	53.24	57.6
288x1600x6	10700	36.01	72.1

Figure 7.15 – Effect of size on MLPs with a single hidden layer

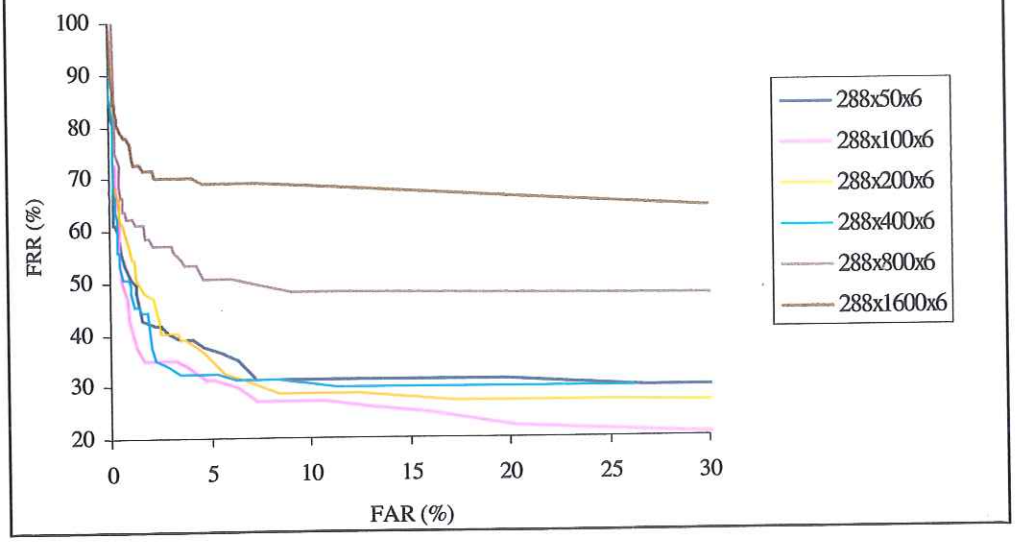


Table 7.11 – Size and single MLP with two hidden layers

MLP Size	Generations to train	Absolute recognition rate %	AUC
288x25x20x6	115200	68.83	44.5
288x50x40x6	61700	70.13	51.5
288x100x80x6	50900	67.53	38.9
288x200x160x6	66400	68.83	44.9
288x400x320x6	31200	70.13	44.3
288x500x400x6	41400	71.43	42.2
288x800x640x6	45000	70.13	53.0

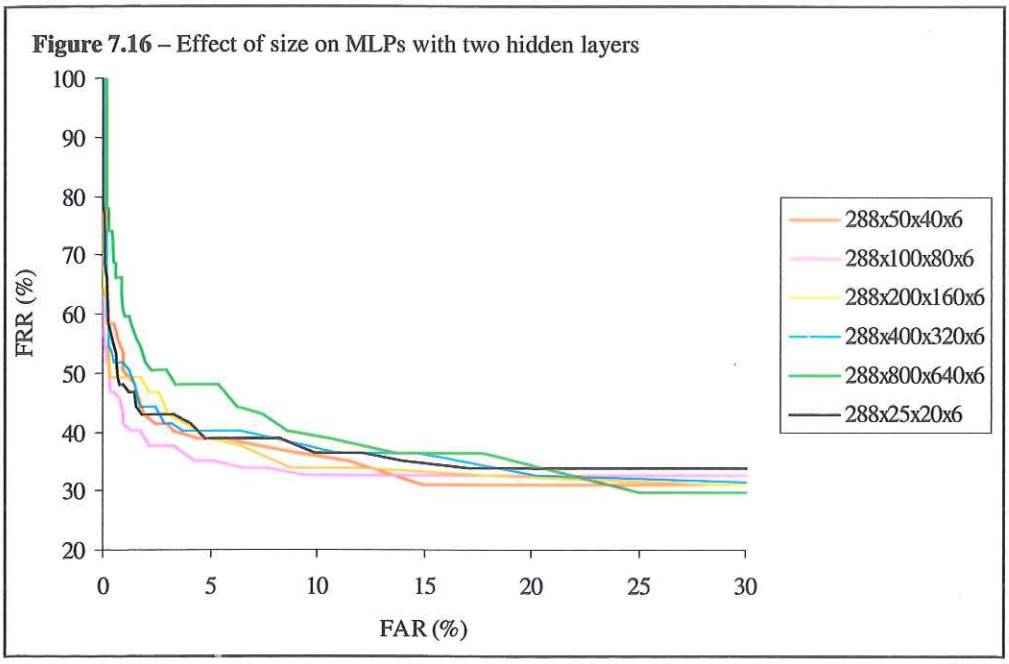


Figure 7.16 – Effect of size on MLPs with two hidden layers

The results show that MLPs with a single hidden layer give similar performance to those with two hidden layers. The best performance of all is obtained by a MLP with a single hidden layer of 100 nodes, giving an absolute recognition rate of 81.82% and an AUC of 37.5. The optimal MLP with two hidden layers has size 288x100x80x6, an AUC of 38.9 and an absolute recognition rate of 67.53%. The reason performance drops away rapidly when the MLPs get large, even though a validation set is in use, is thought to be because it is easier for a large MLP to give good performance on both the training and validations sets without having generalised properly. The smallest MLPs perform worse than the optimally sized MLPs as they are too simple to model the problem.

It is evident that MLPs with two hidden layers are less sensitive to network size change as, upon inspection, it is clear the ROC curves produced for each differently sized MLP are similar when compared to the ROC curves of the MLPs with a single hidden layer. Further discussion of issues relating to size and performance can be found in Haykin [1999].

7.6.2 Size Using Multiple Expression Recognising MLPs

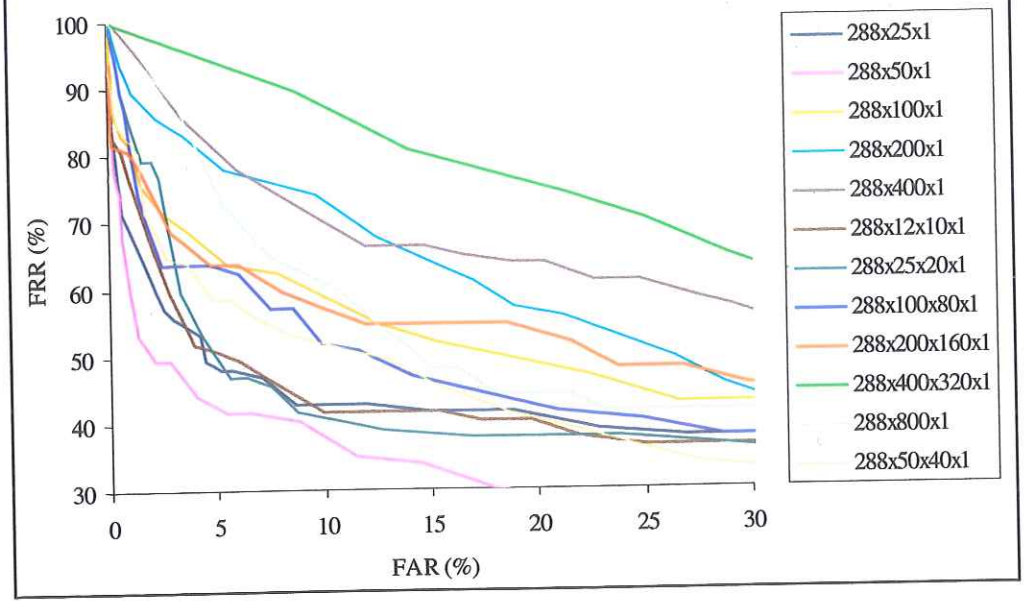
To determine whether performance could be improved, and to allow direct comparison to the SVM approach, rather than using a single MLP, six MLPs each trained for a single specific expression were used ie six binary classifiers. The approach taken was identical to that used when one expression recognising MLP was trained, but each MLP had a single output node and one expression to which it was trained to respond positively. Results from the six binary classifiers were merged as before, with the MLP giving the strongest response (rather than the node of the MLPs recognising all six expressions) labelled the winner. The results achieved are given in **figure 7.17** and **table 7.12**.

Table 7.12 - Size and individual MLPs

MLP Size	Generations to train	Absolute recognition rate %	AUC
288x25x1	163300	75.32	59.6
288x50x1	18200	79.22	51.7
288x100x1	37600	77.92	73.7
288x200x1	14600	72.72	87.4
288x400x1	76700	64.93	89.7
288x800x1	38800	70.13	92.8
288x12x10x1	49800	70.13	64.0
288x25x20x1	143400	68.83	71.0
288x50x40x1	18700	75.32	69.6
288x100x80x1	150100	76.62	70.2
288x200x160x1	38900	67.53	73.0
288x400x320x1	145100	58.44	96.7

The results show that the performance is degraded when using six expression recognising MLPs as compared to a single multi-class MLP. Absolute recognition rates are similar, but the AUC values are considerably higher. The best AUC value achieved is 51.7, compared to 37.5 achieved when a single multi-class classifier is used. Thus it can be concluded that use of multi-class MLP classifiers is better for the expression recognition problem described here than use of binary classifiers.

Figure 7.17 – Individual MLPs and the effects of size



7.7 Summary

The work in this chapter has shown that, of the approaches attempted, the best for expression recognition using MLPs:

- Includes non-expression data in the training set
- Normalises input values prior to entry into classifiers
- Takes symmetrical ratios of averaged motion to cancel out rigid head motion
- Uses both the speed and direction values from these ratios
- Averages motion data over regions of the spatial face template used by the ratio template algorithm
- Uses a single multi-class MLP of size 288x100x6 for the classification task

The best empirically determined approach for expression recognition using MLPs is summarised in **figure 7.18**. This provides a target level of performance to

try to improve upon in the following chapter where the use of SVMs for expression recognition is investigated.

Figure 7.18 – Summary of approach giving optimal expression recognition performance using MLPs trained using back propagation

Network Size: 288,100, 6

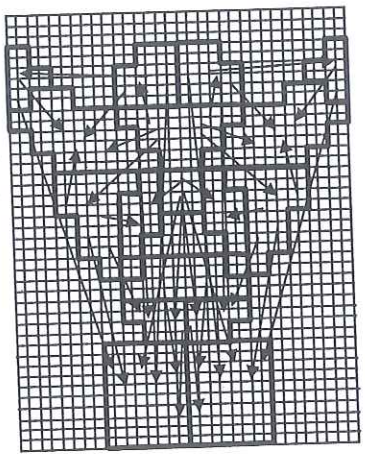
Regions: Regions of ratio template algorithm's face template

No Ratios: 36

Data Type: Normalised speed and direction information

Absolute Recognition Rate: 81.82%

AUC Value: 37.5



8 SUPPORT VECTOR MACHINE EXPRESSION CLASSIFICATION

The second approach considered for classifying facial expressions involved the use of Support Vector Machines (SVMs). As in the previous chapter, a number of ROC curves are presented demonstrating the performance of SVMs under a range of different conditions. As described in **chapter 6**, SVMs take a different approach to classification from MLPs and by finding the optimal SVM expression recognition performance it will be possible to compare and contrast their use with the MLP approach.

Firstly, **section 8.1** presents the results achieved when the best data representation found in the previous chapter is used with a basic SVM. This is followed in **section 8.2** by a demonstration of the importance of data normalisation. **Section 8.3** then presents the effects of using cost models (see **section 6.5**), followed in **section 8.4** by a comparison of two strategies for merging the outputs of the binary SVMs. **Section 8.5** studies the effects of using different kernels and different kernel parameters before the introduction, in **section 8.6,** of a novel optimisation approach taken to learning regions for motion averaging and the ratios to be taken. The MLP and SVM approaches are then compared and contrasted in **section 8.7**, before a final summary is provided in **section 8.8**.

8.1 Initial Results

The first experiment to test the efficacy of the SVM approach was based on the best motion data representation found in the previous chapter (averaging over ratio template algorithm's face template and 36 symmetric ratios between these regions). It was decided to use a second order polynomial kernel, with all other variables set at their defaults by the SVM^{light} application. Outputs of the binary classifiers are merged using the "1-against-all" approach. The variables were set at their defaults by the SVM^{light} application so as to provide a first 'best guess' performance to benchmark against in subsequent sections. Results are presented, with initial comparison to the best MLP results, in **table 8.1** and **figure 8.1**.

Table 8.1 - Initial MLP to SVM comparison

Kernel type	Mean no. support vectors per classifier	Absolute recognition rate %	AUC
2nd order polynomial	387.0	54.55	72.9
Multiclass MLP	NA	81.82	37.5
6 Binary MLPs	NA	79.22	51.7

Figure 8.1 – (a) Data representation and SVM parameter summary (b) ROC curve comparing results achieved by best MLP approaches to that achieved by preliminary SVMs

(a)

Kernel Type: Polynomial

Order: 2nd order

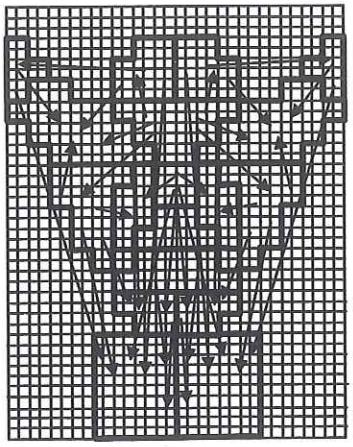
Error/Margin Trade Off: Default

Cost factor: Default

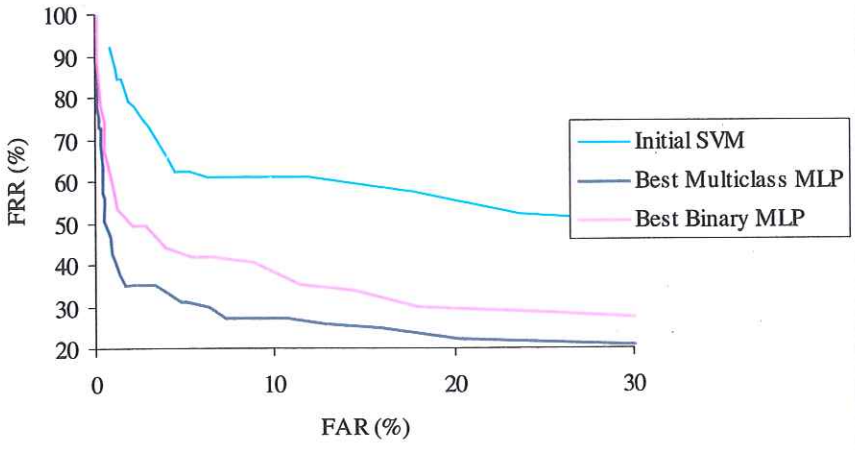
No. Regions: 18

No. Ratios: 36

Merging Strategy: "1-against-all" strategy



(b)



As can be seen from **figure 8.1**, the performance if this initial prototype SVM approach was worse than that of the optimal MLP approach. It was considerably worse than the best MLP result achieved using a single multi-class MLP, with an absolute recognition rate of 54.55% compared to 81.82% and an AUC score of 72.9, compared to 37.5. The performance was also worse than that achieved when only binary classification was used with the MLP approach, with an absolute recognition rate down from the 79.22% achieved with the MLPs. Nonetheless, despite the reduced performance, there are a range of techniques that can be applied to improve this SVM prototype and these are discussed in the remainder of this chapter.

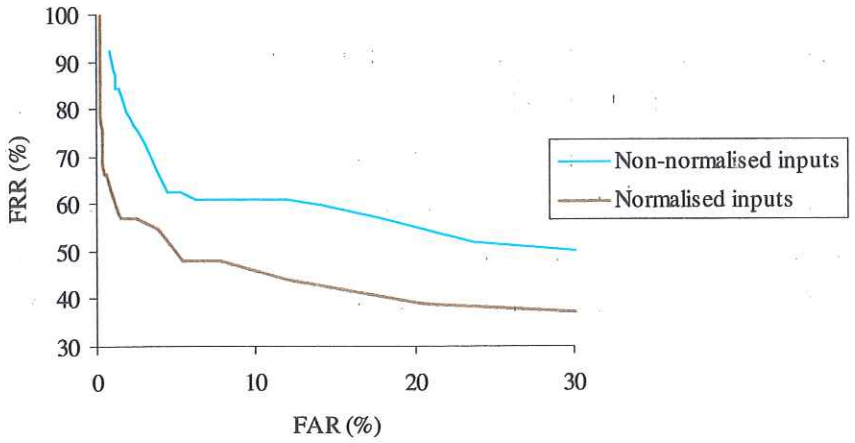
8.2 Normalisation

As with MLPs, normalisation of data for input into SVM is important if one wishes to achieve good results [Gunn 1998]. This helps prevent values in greater numerical ranges dominating those values in the smaller ranges. The experiment described here was set up to test the hypothesis that normalisation of data would lead to improved recognition rates and better ROC curves. The SVM parameters and data representation were identical to those of the previous experiment (**figure 8.1a**). However this time, prior to entry of data, it was normalised, again with standard deviation normalisation [Demuth 1998]. The results are given in **table 8.2** and **figure 8.2**.

Table 8.2 – Data normalisation

Data type	Mean no. support vectors per classifier	Absolute recognition rate %	AUC
Normalised	149.3	68.83	55.0
Not Normalised	387.0	54.55	72.9

Figure 8.2 - ROC curve showing effect of normalising inputs on SVM results



The process of normalisation leads to a much improved recognition rate and ROC curve. The recognition rate improves by more than 14% to 68.83%, and the FRR is improved for all FARs as can be seen in the ROC curve of **figure 8.2** (and as is indicated by a reduction in AUC score from 72.9 to 55.0). In subsequent experiments all data is normalised using the standard deviation based approach.

8.3 Weighting

In previous experiments, the relative weighting of training errors to positive to negative examples has been set at its default level. However, SVM^{light} also allows training with cost models that adjust the relative weightings between errors to positive and negative examples. Cost models are discussed earlier in **section 6.5**.

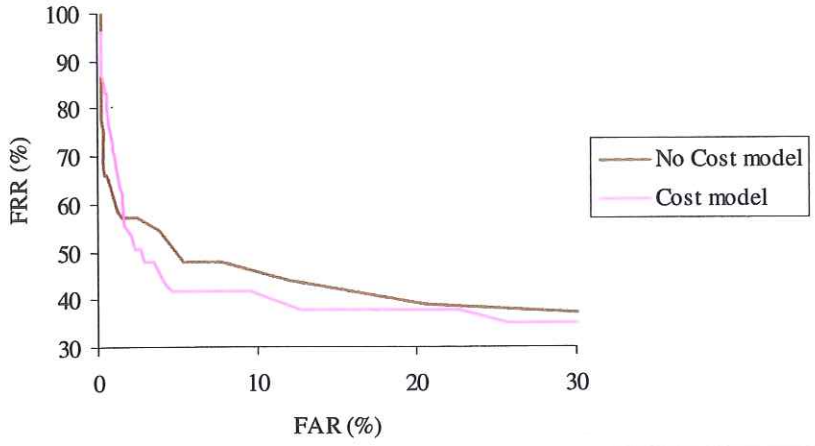
The use of cost models could be of use in the work presented here as there are different proportions of positive to negative examples in the training sets of the six expression recognising classifiers. For example, the training set for the happiness classifier consists of 32 positive examples and 2532 negative (a ratio positive to negative of 0.0126:1), while the training set for the anger classifier consists of 16 positive examples and 2548 negative examples (a ratio positive to negative of 0.0063:1). Thus, errors on

positive examples are not penalised evenly for these two classes. Comparison between performance when cost models are and are not used is given in **figure 8.3** and **table 8.3**.

Table 8.3 – Effects of Cost Models

Data type	Mean no. support vectors per classifier	Absolute recognition rate %	AUC
Normalised	149.3	68.83	55.0
Normalised with Cost	218.2	64.93	51.0

Figure 8.3 - ROC curve showing effect of using Cost models on expression classification performance



The performances of the classifiers trained with and without cost models were fairly similar. The AUC value improved from 55.0 to 51.0 when cost models were used, although the absolute recognition rate dropped from 68.83% to 64.93%. Thus, it is not immediately obvious which strategy should be carried forward for use in subsequent work. It was finally decided to use SVMs trained using cost models in subsequent work as this approach intuitively seems the more sensible. This is because cost models do not weight performance in favour of any facial expression in particular as could be the case with classifiers trained without cost models ie in this situation, classifiers trained without cost models are more likely to achieve higher recognition rates for happiness than for anger, as there is a higher

positive:negative training example ratio, and thus errors to happiness training examples are relatively more heavily penalised than errors to anger training examples. Thus, although use of cost models does not improve performance for this data representation, neither does their use damage performance, therefore cost models are used in subsequent work as they may prove effective for other data representations.

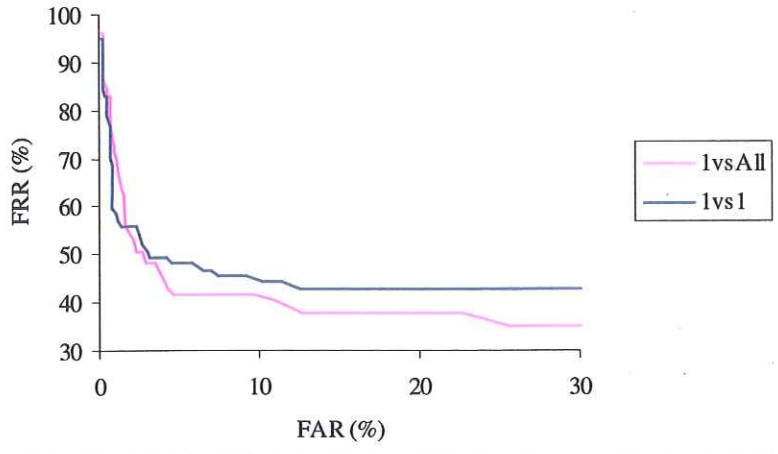
8.4 "1-against-1" vs "1-against-all" Strategies

Two commonly used strategies for merging the outputs of different binary classifiers, "1-against-1" and "1-against-all", were introduced in **section 6.2.6** with the specifics of the classifiers trained in each case provided in **figure 6.10**. It was decided to determine which strategy gave the best performance for the expression recognition task presented here. Results are presented in **table 8.4** and **figure 8.4**.

Table 8.4 – Comparison between different merging strategies

Data type	Mean no. support vectors per classifier	Absolute recognition rate %	AUC
"1-against-All"	218.2	64.93	51.0
"1-against-1"	103.2	57.14	51.6

Figure 8.4 - ROC curve comparing results achieved when "1-against-1" and "1-against-all" strategies are used for merging SVM results



The ROC curves obtained when the two different merging strategies were used were again similar, with AUC values of 51.0 for "1-against-all" and 51.6 for "1-against-1" strategies. However, the absolute recognition rate of 64.93% for the "1-against-all" approach was significantly higher than that achieved by the "1-against-1" strategy (57.14%). Previous work has suggested that for most problems the "1-against-1" approach is better [Hsu 2002], but for the dataset used here it is evident that use of the "1-against-all" approach gives improved performance. Thus the "1-against-all" strategy is used for all subsequent work.

8.5 Kernel Parameters

Thus far, only a 2nd order polynomial kernel with the kernel parameters set at their default levels has been used in conjunction with the SVMs. However, SVM^{light} provides a number of pre-defined kernels, and it was decided to compare results achieved when different kernels were used. All SVM^{light}'s default kernels are used here (linear, polynomial, radial basis function and sigmoidal). However, to compare these kernels properly, it was necessary to determine the value at which each kernel parameter must be set to achieve optimal performance. Therefore, a number of SVMs were trained with different kernel parameters and examined using cross-validation to find the parameters that worked best.

When there was more than one adjustable parameter involved, the search procedure carried out here involved the use of a grid search, where performance of each parameter combination was studied using evenly spaced points in a grid. Mean squared error (MSE) was then used to describe the error of each parameter combination to the validation set used previously with the MLP work. This error was defined as follows:

$$Error = (MSE_{+ve\ examples} + MSE_{-ve\ expression\ examples} + MSE_{nonexpression\ examples})/3$$

NB unlike the MLPs described in ***chapter*** *7, SVM output is not constrained to the range 0-1. Training for the SVMs described here attempts to assign positive examples a score of +1 and negative examples a score of –1. However, output can fall outside this range. Thus, an output >1 for a positive example is defined in this work as having an error of 0, whilst an output <-1 for a negative example*

is also defined as having an error of 0. This is because a classifier giving such outputs for positive and negative examples has effectively separated the two classes.

Once optimal parameters had been determined, SVMs using these parameters were then tested on the usual set of 77 expressions and 1440 frame long sequence of non-expressions. The following sections present results of the parameter searches for the linear (**section 8.5.1**), polynomial (**section 8.5.2**), radial basis function (**section 8.5.3**), and sigmoidal kernels (**section 8.5.4**). **Section 8.5.5** then compares and contrasts the performance of the SVMs using the different kernel types with parameters set optimally according to cross-validation.

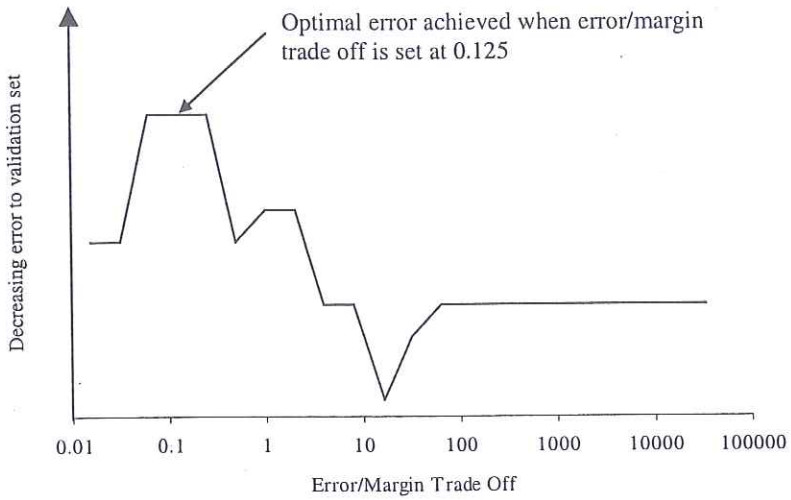
8.5.1 Linear Kernel

The linear kernel is defined as follows:

$$K(x_i, x_j) = x_i x_j$$

Thus, it has no adjustable kernel parameters. However, one non-kernel parameter, easily adjusted using SVM^{light}, is a penalty parameter of the error term, commonly called the error/margin trade off. This parameter, introduced in **section 6.2.2**, modifies the trade off between classifier complexity and frequency of training error. **Figure 8.5** shows the results of changes to this term when the linear kernel is in use. The results show that the optimal value for the error/margin trade off parameter when a linear kernel was used was 0.125 for the training set and data representation used.

Figure 8.5 Effect of changes to Error/Margin Trade Off on recognition performance of SVMs trained with a linear kernel



8.5.2 Polynomial Kernel

The polynomial kernel is defined as follows:

$$K(x_i,x_j) = (ax_ix_j+b)^c,$$

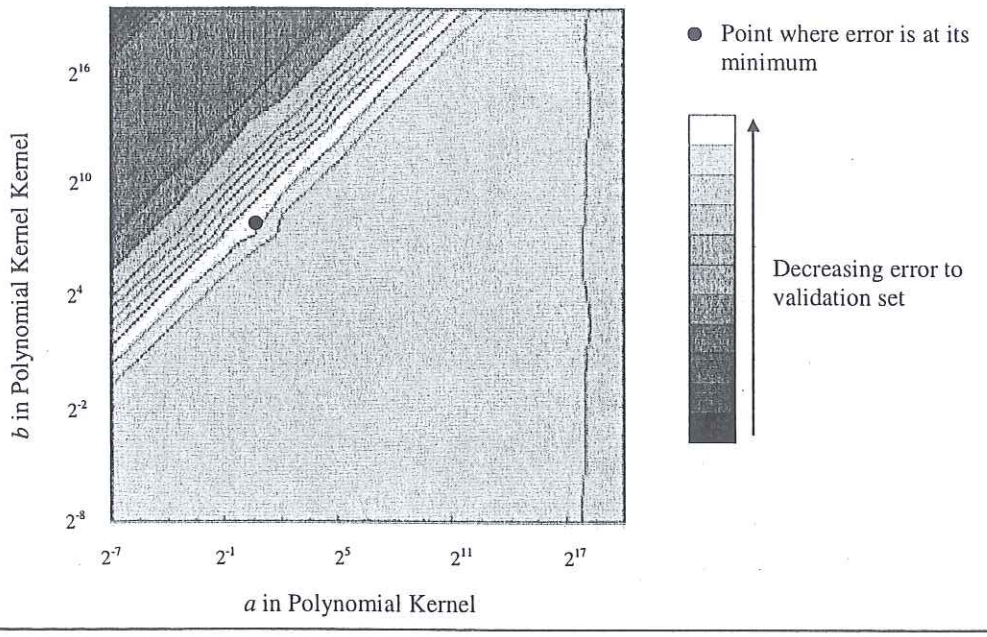
with a and b being adjustable kernel parameters and c describing the order of the polynomial to be used

Thus, for the polynomial kernel it is necessary to find the optimal combination of these three kernel parameters and also the error/margin trade off parameter. A preliminary grid search using widely spaced parameters found that a 2nd order polynomial kernel produced SVMs performing best for the expression classification task. A subsequent grid search was then carried out in a more detailed fashion using only 2nd order polynomial kernels but

modifying the remaining three parameters. The results of this search are given in **figure 8.6** which, for clarity, shows only a 2-dimensional representation of the search where the error/margin trade off parameter was fixed at its optimal value of 2^7. Thus, the figure indicates only the effects of altering parameters *a* and *b* of the polynomial kernel. A valley in the error surface can be seen running diagonally across the search space, with the error to the validation set being at its minimum in this valley. Although error to the validation set was virtually constant in this valley, the error was found to be at its absolute minimum when the kernel parameters were set as follows:

Order (parameter c): 2nd
Parameter a: 2^1
Parameter b: 2^7
Error/margin trade off: 2^7

Figure 8.6 – Grid search for best Polynomial kernel parameters



8.5.3 Radial Basis Function Kernel

The radial basis function kernel (RBF) is defined as follows:

$$K(x_i,x_j) = exp(-a||x_i-x_j||^2),$$

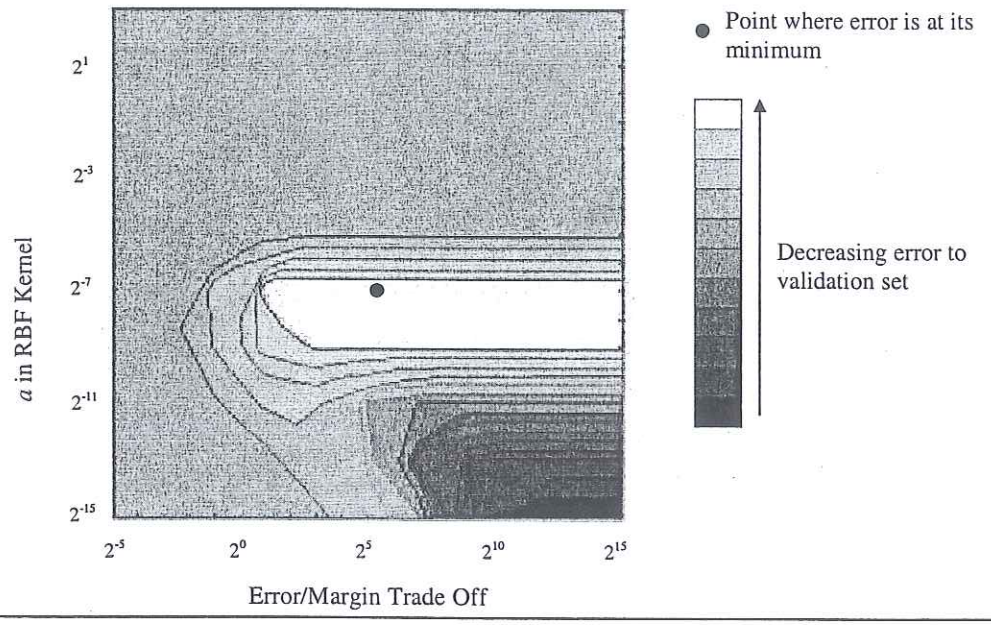
with a being an adjustable kernel parameter

Thus, for the radial basis function kernel, it is necessary to find only the optimum combination of kernel parameter *a* and the error/margin trade off. The results of the grid search are given in **figure 8.7**. A broad valley can be seen where error to the validation set was at or close to its minimum (NB the variation in this valley is so small that it does not show up on the contour graph). The parameters given the absolute lowest error to the validation set were as follows:

Parameter a: 2^{-7}

Error/margin trade off: 2^{5}

Figure 8.7 – Grid search for best RBF kernel parameters



8.5.4 Sigmoidal Kernel

The sigmoidal kernel is defined as follows:

$tanh(ax_ix_j+b)$,
with a and b adjustable kernel parameters

Thus, there are three adjustable parameters, two kernel parameters and the error/margin trade off. Once again a grid search was carried out to determine optimal parameters and these were found to be as follows:

Parameter a: 2^{-20}
Parameter b: 2^{-12}
Error/margin trade off: 2^{11}

No grid is shown here to show the changes in performance as parameters were altered as the error to the validation set was not sensitive to the sigmoidal kernel parameters, with virtually identical errors seen for all parameter combinations.

8.5.5 Comparison of Results

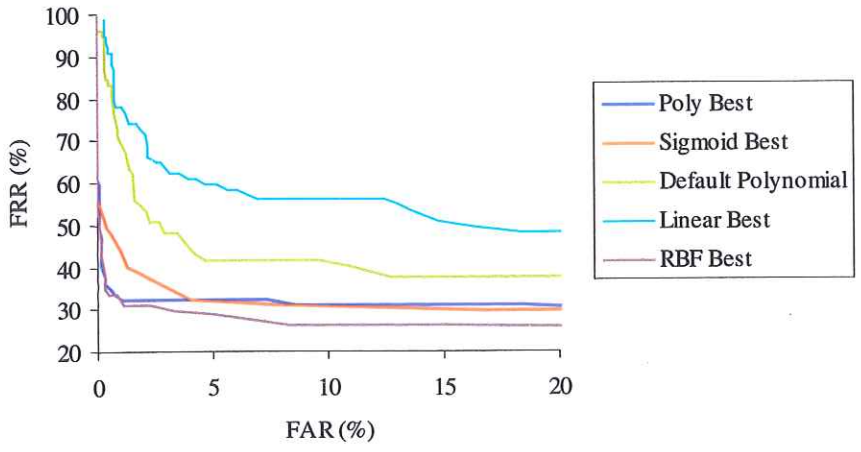
The SVMs trained with the 4 different kernels with parameters set optimally according to cross-validation were then tested on the usual test set. Results are given in **table 8.5** and **figure 8.8**.

Table 8.5 - Comparison between different kernels

Data type	Optimal parameter settings	Error/margin trade off	Mean no. SVs per classifier	Absolute recognition rate %	AUC
Default polynomial	$(1x_ix_j+1)^2$	0	218.2	64.93	51.0
Linear	N/A	2^{-3}	92.7	62.34	65.9
Polynomial	$(2^1x_ix_j+2^7)^2$	2^7	145.8	70.13	32.9
RBF	$(2^{-7}\|x_i-x_j\|^2)$	2^3	418.7	75.32	30.8
Sigmoidal	$\tanh(2^{-20}\ x_i.x_j+2^{-12})$	2^{11}	87.0	71.43	38.4

The optimal kernel was found to be the RBF function kernel, which gave an absolute recognition rate of 75.32% and an AUC value of 30.8. The polynomial kernel gave a similar, but slightly worse performance, whilst the sigmoidal kernel performed worse still, with an AUC of 38.4 compared to the AUC of 30.8 for the RBF kernel. The linear kernel performed worse by a distance for the expression recognition task, with an absolute recognition rate of only 62.34% and an AUC of 65.9, suggesting that the problem is not linearly separable. In addition to performing best, the RBF kernel is also attractive as it has just one adjustable parameter, unlike the polynomial, with 3, and sigmoidal, with 2. This makes searching for optimal kernel parameters much easier. A more detailed introduction to selection of kernel types has been written by Hsu et al [2003].

Figure 8.8 – Results achieved with optimised kernel and training parameters



8.6 Learning Regions & Ratios

So far, the regions for averaging and the ratios subsequently taken have been chosen empirically. However, during the course of the work described in this chapter it was found that, in general, the training time for each SVM was low (often of the order ~5 seconds per SVM on a 1.7GHz machine). Thus, it was thought it could be possible to use an

optimisation approach to learn the regions for averaging and the ratios to take. The motivation behind this is that we want to use the best regions and ratios for expression classification, but so far the selection of these regions and ratios have been based on assumptions as to what would give the best performance. By using optimisation it is possible to find out what is actually the best given the training sets used. The optimisation approach used for this work was simulated annealing due to its simplicity.

8.6.1 Introduction to Simulated Annealing

Simulated annealing is based on the physical annealing processes of solids. The annealing process involves heating a solid in a heat bath until it melts, and then gradually reducing the temperature of the bath until the solid particles arrange themselves into their ground state [Aarts & Korst 1989]. During this cooling process, the particles go from the random arrangement of a liquid to an ordered and highly structured lattice where the energy is minimal. The temperature has to be reduced slowly during the annealing process to allow thermal equilibrium to be reached at each temperature, with the probability of reaching the ground state configuration being reduced if the temperature is lowered too quickly. We can model this annealing process computationally.

Kirkpatrick et al [1982] and Cerny [1985] introduced the concepts of annealing into algorithms for combinatorial optimisation. By substituting the energy of the system with cost and using an algorithm for either accepting or rejecting changes in the system at slowly decreasing computation temperature values, it was possible to solve a range of optimisation problems. Simulated annealing has been applied to problems such as the travelling salesman problem [Cerny 1985], image reconstruction [Sundermann & Lemahieu 1996], and time scheduling [Chen et al 1987].

8.6.2 The Basic Algorithm

An important component of the simulated annealing approach is the selection criteria by which changes are either accepted or rejected. Simulated annealing uses the Metropolis criterion, given by:

$$P = exp\left(-\frac{f(x)-f(y)}{c}\right)$$

where:

f(x) is the cost of a solution x.
f(y) is the cost of a solution y, modified from original solution x.
c is the control parameter, modeling the temperature of the physical system.

Simulated annealing uses the Metropolis criterion when optimising in the following way:

- *if $f(y) \leq f(x)$*

  keep the new solution, y, as the modified solution has a better cost than the original solution

- *otherwise*

  keep the modified solution, y, with a probability, P, given by the Metropolis criterion ie the cost of the new solution is worse than the old solution, but still keep the change with a probability, P.

Therefore, by starting with a high value of c and gradually reducing its value, simulated annealing accepts a higher proportion of detrimental (in terms of cost) changes at the start of the process than at the end. The complete simulated annealing algorithm can be summarised as follows:

1) *Initialise the system:*

   Set c_0, the initial temperature.

   Set L_0, the number of iterations to generate at temperature c_0.

   Set x, the initial solution to the problem.

2) *Generate a new solution, y, from solution x.*
3) *Calculate f(x) and f(y), the costs of the two solutions.*
4) *if $f(y) \leq f(x)$*

   make x equal to y.

else if $exp(-\frac{f(x)-f(y)}{c}) >$ *random number between 0 and 1*

make x equal to y.

5) *If the temperature has not been reduced for* L_k *iterations, reduce the temperature (NB a commonly used decrement function is* $c_{k+1}=\alpha c_k$ *where* α *is a constant close to 1, with typical values lying between 0.8 and 0.99).*
6) *Go back to step 2) and repeat unless stop criterion has been reached (NB a stop criterion usually involves examining the changes in cost over recent generations ie if there has been little or no improvement in cost over n iterations then terminate the algorithm).*

The algorithm described above is the most basic simulated annealing algorithm. A number of variations have been developed to speed up and enhance the results of this approach. For example Fast Simulated Annealing (FSA) [Szu & Hartley 1987] uses a Cauchy-Lorentz visiting distribution, instead of a Gaussian one, resulting in an inverse linear cooling rate. This can help speed up the algorithm. Nonetheless, for the problem presented here, where the search space is fairly limited, it was deemed sufficient to use just the basic simulated annealing algorithm.

8.6.3 Region/Ratio Learning Approach

As the results of the simulated annealing are to be compared to those achieved using the empirically determined regions and ratios used previously, it was decided to constrain the annealed regions/ratios in the same way as done previously. Therefore, as the best approach found thus far used 18 regions and 36 ratios, with regions covering a 32x38 pixel grid, it was decided to limit the numbers and positions of regions and ratios learnt by simulated annealing in the same way.

The region/ratio learning algorithm runs as follows:

1) *Randomly initialise ratios and regions:*
   - *Randomly choose starting x and y locations for 18 regions in a 32x38 pixel grid.*
   - *Make these regions 5 x 5 pixels in size.*

- *Randomly select 36 ratios between these regions.*

2) Initialise simulated annealing system

- *Set initial temperature, c_0, by trialling the system – increase temperature gradually until the number of accepted transitions equals the total number of transitions (ie all changes are accepted).*
- *Set the number of iterations at each temperature, L, at 40 (**NB this is the number of accepted changes, not the total number of attempted changes and the value was chosen empirically**).*
- *Set the cooling constant, α, to 0.97.*

3) Run simulated annealing algorithm

- *Randomly change a region position (move it up/down or left/right 1 pixel) or a ratio (change one of a ratio pair to a different region).*
- *Calculate cost and choose whether to keep change (as described above). Cost is calculated in terms of the mean squared error of the SVMs to a validation set.*
- *Reduce temperature if number of accepted changes at current temperature is greater than L. The new temperature is calculated as $c_{k+1}=\alpha c_k$.*

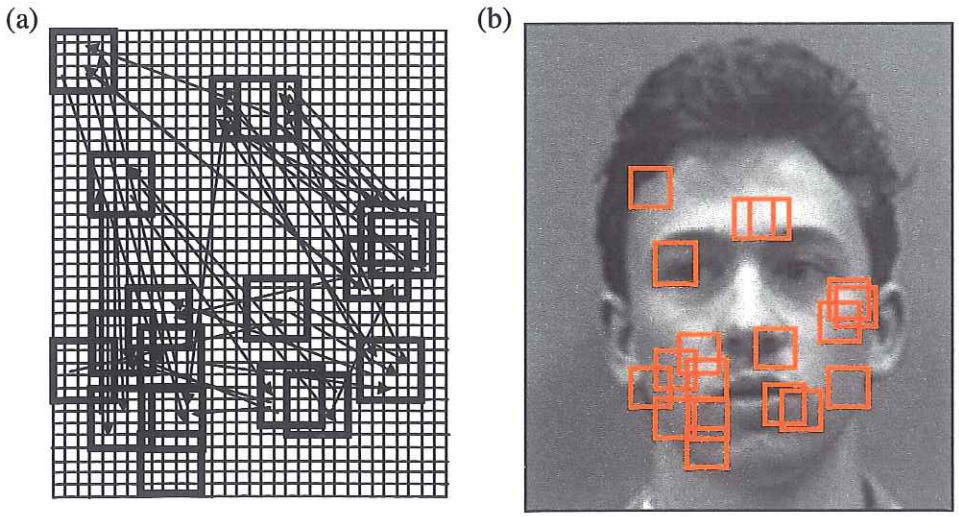
4) Stop when no improvements to cost are made in 300 successive iterations

The SVMs used in this process used RBF kernels and had the parameters set at the level found to be optimal in **section 8.5.3**. There is no guarantee that these parameters are the best for the representations found in this section, but it is not feasible to include kernel parameters into the search space of the simulated annealing algorithm as this would increase the size of the search space significantly. Thus fixing the kernel parameters at the level found to be best in **section 8.5.3** seems sensible.

The regions and ratios found to be the best by the simulated annealing algorithm are given in **figure 8.9**. The results of the optimisation show that the regions of importance for

expression recognition are the mouth corners, eye corners and nose-bridge. Of these areas, the most important region of the face for classification of facial expression appears to be the mouth corners, with a cluster of regions for averaging being positioned here. Only two regions are located at the nose-bridge as it only a small area. However, if one examines the optimised ratios it is evident that a large number involve the nose-bridge, demonstrating its importance. Most of the remaining regions are positioned at the edges of the eye. The motion of the chin and forehead seems to be of little importance.

Figure 8.9 – Regions and ratios learnt by initial simulated annealing approach (a) showing there positions on a 32x38 grid (b) where these regions fall on the human face



A grid search for optimal RBF kernel parameters was then carried out as in **section 8.5.3,** with the representation of input as determined by simulated annealing, and the results are shown in **figure 8.10**. The optimal parameters according to cross validation were found to be:

Parameter a: 2^{-7}

Error/margin trade off: 2^{9}

The results achieved on the test set with these optimal kernel parameters are given in **table 8.6** and **figure 8.11**.

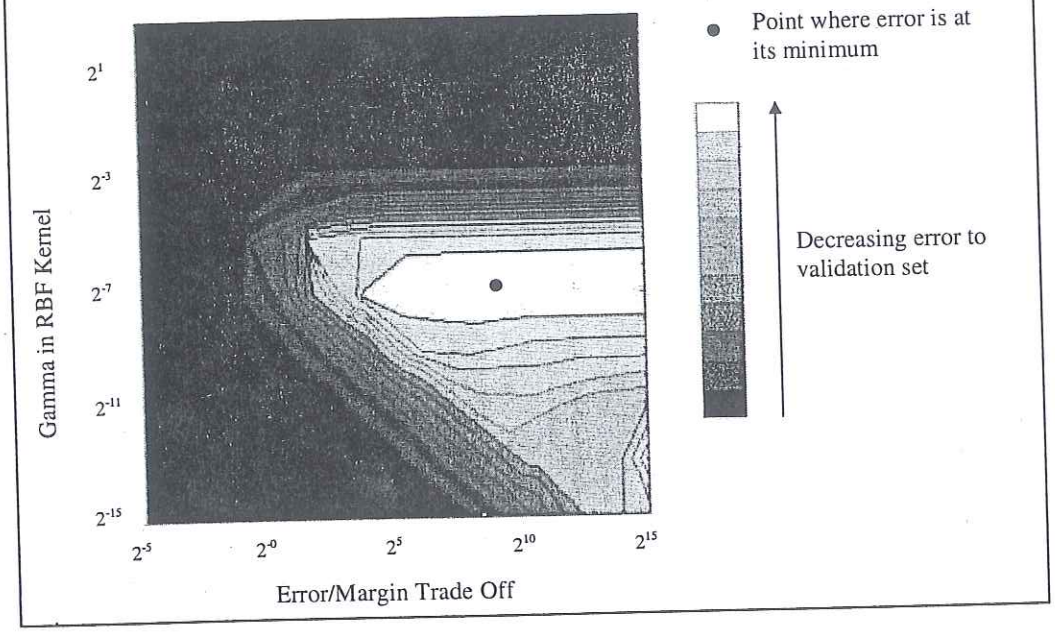


Figure 8.10 – Grid search for best RBF kernel parameter and SVM error/margin trade off parameter for data representation learnt by simulated annealing

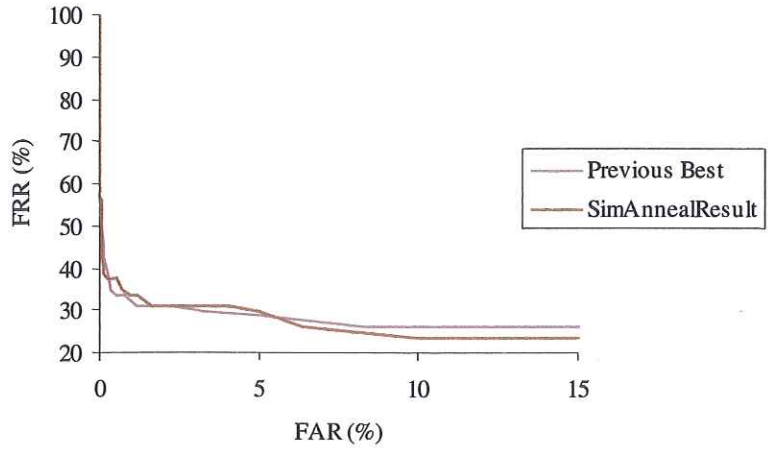
Table 8.6 Effect of using optimised regions

Data type	Mean no. SVs per Classifier	Absolute recognition rate %	AUC
Previous Best	418.7	75.32	30.8
SimAn Regions&Ratios	105.7	80.52	31.8

These results show similar performance when the original representation (using regions of the ratio template algorithm's face template) and the regions and ratios learnt using simulated annealing are used. The AUC values are virtually identical (30.8 and 31.8), although the absolute recognition rate is improved significantly when the simulated annealing approach is used (increased from 75.32% to 80.52%).

The reason that performance was not improved significantly using simulated annealing was thought to be that the simulated annealing search involved several less degrees of freedom than were available to the ratio template algorithm's face template. The regions over which data is averaged in the ratio template algorithm's face template vary in size and are not all square, thereby allowing matching of size and shape of facial components accurately. However, the simulated annealing approach was constrained to square regions of one size only, 5pixels x 5pixels. Further investigation was not carried out into the effects of allowing regions to change size and shape due to processing and memory constraints (the annealing process previously described already takes ~2 days to complete on a 1.7GHz machine).

Figure 8.11 – Comparison between previous best performance and performance when regions and ratios learnt using simulated annealing are used



8.7 Comparison between MLP and SVM Expression Recognition Performance

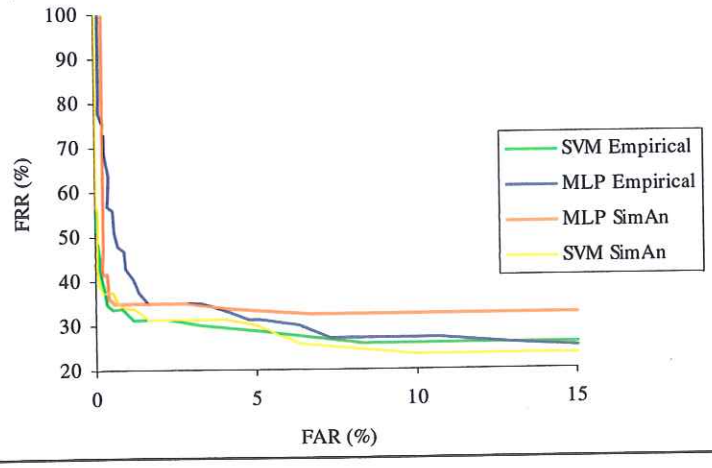
With optimal representations and parameters determined for both the MLP and SVM approaches it is possible to compare and contrast the performances of the two techniques. The performance of the best MLPs and SVMs are given in **table 8.7** and **figure 8.12**. The empirical best shows the best performance when regions from the spatial face template of the ratio template algorithm are used, whilst the learnt best shows the best performance when regions/ratios from simulated annealing are used (NB MLPs were

trained using the regions and ratios learnt by simulated annealing and SVMs to allow for this comparison).

Table 8.7 – Comparison between MLP and SVM approaches

Kernel Type	Absolute recognition rate %	AUC
Empirical SVM	75.32	30.8
Empirical MLP	81.82	37.5
SimAnneal SVM	80.52	31.8
SimAnneal MLP	70.13	34.8

Figure 8.12 – Comparison between expression recognition performance of the best MLP and SVM approaches



The ROC curve and its corresponding AUC value are much improved when SVMs are used for both the empirical and optimised data representations, although the best absolute recognition rate was achieved by the MLPs using the empirical representation (81.82%). From this comparison it can be concluded that the SVMs are the best classifier for use with the final expression recognition system, as it gives the best performance at the low FARs that are required by the system. Also, the best data representation is that provided by the simulated annealing optimisation.

Two other points are also worth making. First, the performance of the SVMs is even more impressive given that they are binary classifiers and for the MLP case performance was reduced significantly when only binary classification was allowed (see **table 8.1**). Second, it is interesting to note that the performance of the SVMs using sigmoidal kernels (see **table 8.5**), and thus mimicking a MLP, was similar to the performance of the MLPs with AUCs of 38.4 and 37.5 respectively. This suggests that the main performance benefit from SVMs came from its ability to use different kernels types.

The advantages of the SVM approach compared with the use of MLPs for the expression recognition task can be summarised as follows:

- **Reduced training time** - training time is much reduced when compared to MLPs. It generally takes a few seconds to train a SVM for this task, compared to anything from ~10 minutes to a number of hours depending on the size of the MLP.
- **Improved ROC curve and AUC value** – the SVMs have managed to separate the expression from the non-expression examples more effectively than the MLPs, resulting in an enhanced ROC curve and AUC score.
- **No handcrafting of size** – Unlike MLPs, there is no need to worry about network size as it is automatically determined by the number of support vectors found.
- **Consistent results** – the SVMs generated when a set of kernel parameters are chosen is identical each time a SVM is trained. Although MLPs give similar performance each time a new network is trained, there is always some slight variation.
- **Speed of obtaining outputs** – the speed at which outputs can be obtained from the trained SVMs is quicker than that from the MLPs. In the final real-time system this frees more time for other tasks such as face tracking.

- **Validation sets** – the basic SVM approach does not require validation sets as used by the MLPs (although they have been used here for the grid searches).

The advantages of the MLP approach over SVMs can be summarised as follows:

- **Improved absolute recognition rate** – the best absolute recognition rate of 81.82% by the MLP approach is higher than that achieved by SVMs.
- **Easier implementation** – the MLP approach is easier to implement than SVMs.

One particularly important advantage that the SVM approach has over MLPs for this task is the speed of training. The two factors affecting expression recognition performance (if the role of the training set is ignored) can be thought of as being the classifiers used and the data representation used. These results show that, although the SVM classifiers perform slightly better than MLP classifiers for the expression recognition task, the results are not significantly different. However, the speed of training of the SVMs has allowed simulated annealing search to be made for an improved data representation, something not feasible with the slower training MLPs. **Figure 8.12** shows that performance can be improved by searching for better data representations. Further work could be carried out in this area, and the SVM approach is much better suited for this task than MLPs.

8.8 Summary

The optimal approach for expression classification found by the studies in the previous two chapters uses regions and ratios learnt by the simulated annealing algorithm and uses SVMs as the classifier. **Figure 8.13** summarises this approach. This representation will be used in the following chapter as part of the completed fully automated real-time expression recognition system to drive the expression-based applications.

Thresholds will be set such that the classifiers give a FAR of 1%. This threshold level has been chosen as, when the FAR is 1%, the FRR is close to its minimum whilst the

FAR is still at an acceptably low level (remember both the FRR and FAR should be as low as possible). If the threshold is reduced then the FAR increases without providing significantly improved FRR rates. Thus the system would fire more often for non-expression inputs without recognising significantly more facial expressions. On the other hand, if the threshold is increased, then the FRR increases rapidly, and thus the system would no longer recognise a number of true facial expressions.

Figure 8.13 – Summary of approach giving optimal expression recognition performance

Classifier Type: SVM using cost models

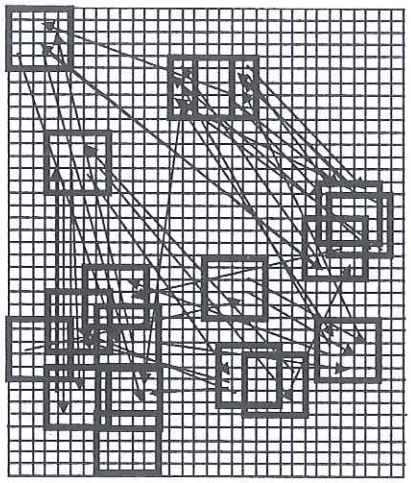
Merging Strategy: "1-against-all"

Kernel Type: RBF

Regions/Ratios: Learnt by simulated annealing optimisation

Absolute Recognition Rate: 80.52%

AUC Value: 31.8



9 APPLICATIONS

In this section we examine the application of the real-time system developed to prototype affective computing systems. **Section 9.1** introduces some applications for automated expression recognition systems in general. **Section 9.2** describes a novel chatroom application, called EmotiChat, and provides an example of the application in use. **Section 9.3** then introduces a piece of empathetic software that attempts to respond to the mood of a computer user with the aid of music. **Section 9.4** briefly discusses the expression recognition systems sensitivity to different views, before finally in **section 9.5** an approach to extending the system beyond the Ekman basic emotions is described.

9.1 Application Domains

An obvious application of an automated expression recognition system is in the field of robotics as, for a socially aware robot, the ability to respond appropriately according to the emotional state of those around is vital. As facial expression is an important cue to emotional state, an automated expression recognition system is essential. However, the use of an automated expression recognition system is not restricted solely to robotics, but ranges from applications specific to facial expression, such as lie detection via recognition of micro-expressions [Ekman 1998b], to applications more general to the field of affective computing, such as socially intelligent software tools. This section describes these and other potential applications.

Use of automated expression recognition as part of a computer tutor would allow it to gauge levels of user interest, confusion, pleasure, or boredom [Picard 1998]. Such an ability would allow it to adapt and make intelligent responses according to the users facial expression. For example, if the user appeared confused the lesson could be slowed down or sections repeated.

Another potential use would be its inclusion into socially intelligent software tools used in the therapy of autistic children whose skills of social interaction and communication are severely impaired. Many people with autism have been shown to interact very naturally with computer technology, while disliking complex interactions

with people, familiar or not [Ogden 2001]. Therefore, a computer agent allowing autistic children to explore simple social situations without the fear of becoming involved in complex human interactions would be an extremely useful tool. Research on the AURORA Project has already involved the use of mobile robots as social mediators in the field of autism therapy [Dautenhahn 2003].

In the field of behavioural science, highly trained humans are required to assess levels of stress, depression, deceit and so on. Such processes would become much more widely accessible if they could be fully automated [Bartlett 1999]. Additionally, it is known that there is a whole range of facial clues to deceit. Humans have very poor voluntary control over some facial muscles, with different systems controlling spontaneous and forced or posed facial expressions, causing some actions to be missing when an emotion is posed. Also, when a person is lying, certain characteristic movements, called micro-emotions [Ekman 1998b] (see **section 2.3.1**), are seen in specific regions of the face. These micro-emotions are much shorter in duration than normal emotions and could be picked up by an automated system. A final difference between genuine and posed expressions of emotion lies in the dynamics and symmetry of the motion, with genuine emotions being more symmetrical than those that are forced [Bartlett 1999] and smoother in onset. Systems trained to detect human facial expressions could be set up to detect such clues, thereby automatically detecting deceit.

Finally, another potential use of expression recognition lies in consumer behaviour studies. During consumer testing, observers watch consumers to determine how they react to the product, such as food, where products are often manufactured with a range of ingredients, in effect forming a "taste space". As the subjects eat the food, the observers record the reactions of the tasters. Automated expression recognition could also be applied to consumer studies of humans watching films or television programmes, examining their facial responses to certain scenes to see if they find them, for example, humorous or frightening.

Although effective solutions to some of these problems are still some distance away, the system described in this thesis can be applied to simple application domains to demonstrate its uses.

9.2 EmotiChat Application

Emoticons are symbolic abbreviations for emotional state that are used commonly in internet chat rooms, in e-mails, and in mobile phone text messages [Schurter 2003]. The use of emoticons in internet chat rooms demonstrates the key role that signalling emotional state has to play in human interaction. Many chat room users employ a whole range of these emoticons to enhance interaction and give an insight into their current emotional state. Without such information what is meant as a light-hearted joke can easily be misinterpreted, causing the other party to become upset because he or she is unclear as to how the statement was meant. **Table 9.1** provides some example emoticons, although it should be noted that emoticons vary between different cultures (eg (^_^) is used in Japan for a happy person).

Table 9.1 - Emoticons

Emoticon	Meaning
:-)	Happy person
:-(	Sad person
:-D	Laughing person
:-o	Surprised person
;-)	Winking person
}: [	Angry person
:-|	Disgusted person
:'(	Crying person
(:+(	Fearful person

Use of emoticons is so widespread that the Yahoo Messenger chat room application includes a set of animated emoticons that can be inserted into text, whilst MSN Messenger (version 6) allows a person to use a web camera to take pictures and insert them into the text as a replacement for emoticons.

One application visualised for the automated expression recognition system described here is the automatic insertion of emoticons into the text of chatroom users when facial expressions appear on their face during the course of a chatroom

conversation. The system developed here is based on a chatroom application authored by Tauasa Timoteo. The application is called JChat and is available for download from *www.ansurgen.org/board/ttimoteo/jchat/about_jchat.html*. It is a Java servlet using the Java servlet API and has a freely available source code. It runs in any web browser that supports forms and javascript, such as Netscape and Microsoft Internet Explorer. The one weakness of JChat is that it does not automatically search for newly posted messages, and rather waits for users to submit their own messages before looking for new messages. The basic system was modified here to remove this limitation and was then connected to the expression recognition system.

Currently the communication between the expression recognition system and the EmotiChat prototype application takes place via a file, with the expression recognition writing to the file if it identifies a users expression. The EmotiChat application reads from this file when the text is updated and responds appropriately. If an expression is seen in the time frame between one message being sent and the next, an emoticon is automatically inserted into the chatroom text.

An example recording of the application in action is to be found on a compact disk attached to this thesis. The file is entitled "*EmotiChat.mpeg*" and demonstrates the automated insertion of a happy emoticon. The happy emoticon is chosen as the results of a questionnaire provided to computer science students (**appendix 8**) showed that they believed this to be the most important of the expressions to recognise.

The sequence shown in "EmotiChat.mpeg" is a recording of an artificial situation with pre-determined dialogue to show the feasibility of such a system. The training set used by the expression recognition system consists of only posed artificial expressions and it was found that measuring the more subtle and mixed expressions that are seen on users in the real-life use of chatrooms proved challenging. Issues relating to how it may be possible to get around this problem are discussed in more detail in **section 9.5**.

The user is seated in front of the monitor and is reading the text. The expression recognition system is tracking the motions of the face, and when the smile is seen, the system recognises this and triggers the insertion of a happy emoticon into the text of the EmotiChat application. **Figure 9.1** shows a few frames taken from the "*EmotiChat.mpeg*" sequence.

The top panel of **figure 9.1** shows a view of a monitor with the EmotiChat application visible with two users conversing. The second panel then shows a user smiling at a comment made by the other user. The user enters a short response to this, but the EmotiChat application, having recognised the smile on the user's face, inserts, in addition to the user's comment, an extra line saying *":) – Keith just smiled"*.

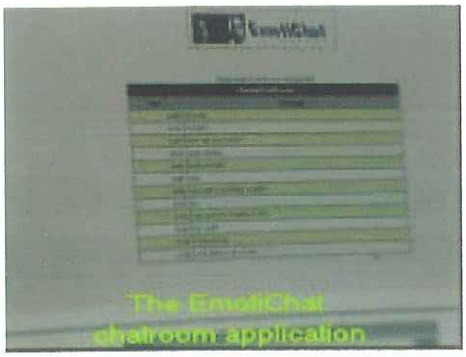
A questionnaire (see **appendix 8**) completed by 100 students shows that in response to the question:

> "Consider a chatroom application that watched your facial expressions and automatically inserted emoticons (eg ɔ) into the text for you when it saw a change in facial expression. Rate on a scale 1-5 how useful (1 – very useful, 5 – of no use) you think such a application would be."

39% gave the system either a usefulness rating of 1 or 2 (mean rating of 2.87). From speaking to those subjects who were critical of the usefulness of such a system it was evident that their main concern was that they did not wish other chatroom users to know about their facial expressions. The results also showed a difference in the responses between non-computer science students and computer science students (chi-squared tests show a probability <0.05), with non-computer scientists giving the system a mean rating of 3.00 and computer science students a mean rating of 2.74.

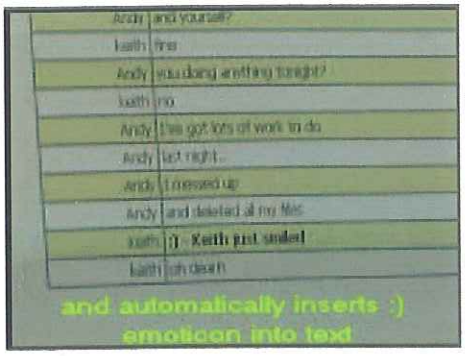
Upon examination of the raw data from the questionnaire it also became evident that the usefulness rating of this application suffered from people's scepticism to affective computing systems in general. Of those who gave the chatroom application a usefulness score of 1 or 2, the mean rating in response to the question:

Figure 9.1 – Example frames taken from the "*EmotiChat.mpeg*" sequence demonstrating the EmotiChat application in use



Two users log into the EmotiChat system, a chatroom application linked to the real-time facial expression recognition system

Whilst chatting, a statement is made and the chatroom user smiles



The real-time expression recognition system correctly classifies the motion signature as a smile and automatically inserts an emoticon into the chatroom text

> "Computing technology that is able to understand and respond to a user's emotions is useful (1 strongly agree, 5 strongly disagree):"

was 2.21, whilst for those who gave the EmotiChat application a usefulness rating of only 4 or 5, the mean score in response to this question was 3.24.

9.3 Music/Web Browsing Application

The second prototype application developed is one that constantly monitors the expressions of a user as they operate a desktop machine, and if a recognisable facial expression is seen it automatically triggers a desktop application. Potential target applications include web browsers and media players.

Such a system could be used in a number of ways. It could either be used to reinforce a user's positive emotions, to attempt to change the mood of a user expressing negative emotion, or simply to show empathy as to a user's emotional state. One specific example could be the application responding to a user's angry face by playing some soothing music.

When asked about the usefulness of such a system, 100 students 36% gave the system a usefulness rating of 1 or 2, where a rating of 1 means very useful and a rating of 5 means of no use (**appendix 8**). As with the EmotiChat application, the mean usefulness rating obtained from computer science students was lower than that of non-computer science students.

An example filming of this application in use is provided on a compact disk attached to this thesis. The short movie is called "*ExpRec.mpeg*". A few frames taken from this sequence are also provided in **figure 9.2**. The top panel shows a view of the monitor where the scene visible to the system's camera can be seen. It shows a room with a person seated and the face tracker locked onto the face, with the facial motion being determined by the real-time implementation of the MCGM. A period of ~20 seconds passes with the user moving their head mainly in a rigid fashion, but with some non-rigid motions. During this time frame the outputs of the classifiers are below the

threshold level required for system firing. The user then smiles and the system correctly identifies the motion signature, labelling it as happiness. The system then triggers the playing of an audio file. Thus, this application responds in a simple manner to a user's facial expression, with the computer demonstrating to the user it has some understanding of their emotional state.

This web-browsing/music application is relatively simple and it would be easy to enhance its usefulness further with the addition of a few basic features. For example, rather than responses to expressions being hard-wired, they could be decided upon by the user themselves when installing the software, or if the application were to be used in a therapeutic domain by those supervising the sessions. Such adaptability would enhance the approach as certain responses may be appropriate for certain people, but not for others.

In addition to the initial set-up of the system being adaptable, it could also be made to be adaptable over time, learning from feedback provided by a user as to what responses are appropriate and when. Such a system could then try to correlate differences in what a user wants with other simple measures that could affect their mood (eg time of day) to make more appropriate responses given time.

Figure 9.2 – Example frames from "ExpRec.mpeg" showing the automated firing of media files by the expression recognition system

Face tracker locks on to user's face and gates MCGM. MCGM determines facial motion and feeds information to expression recognition system

User continues to move face without expressing one of the six basic emotions. System correctly labels these motion signatures as non-expressive

User expresses happiness and this is detected by expression recognition system

System triggers the playing of an audio file by a media player

There is also potential for such a system that triggers desktop applications in response to facial expression to be used by disabled users unable to use normal input devices, such as mice or keyboards. They could thus use facial expression to carry out simple tasks on a computer.

9.4 Tolerance to Yaw & Scale Change

Once the above applications were set-up, it was thought interesting to study the sensitivity of the expression recognition system to changes in viewpoint. Of particular importance was its sensitivity to yaw and scale change.

However, no databases are available that provide facial expression sequences filmed from multiple viewpoints. Thus, obtaining detailed figures (such as those given for the face tracker in **chapter 4**) to characterise the expression recognition system under different conditions is not possible. Nonetheless, by placing subjects in front of the active system and gradually changing the camera viewpoint an investigation was carried out into the sensitivity of the system to yaw and scale change. The results of this study suggest that the expression recognition system can effectively handle changes in yaw ±20° from a fully frontal view and scale changes ±20% from optimum (this equates to a distance range of around one metre in the set-up used for the experiment). For applications involving the use of a desktop computer, where a user's gaze is directed at the monitor and there distance from the monitor is pretty constant, it is thought that such ranges are more than adequate.

9.5 Two-Dimensional Emotion Models

One potential limitation of the system described here is that it recognises only expressions of Ekman's basic emotions. Although this is sufficient for simple interactions, it would be advantageous if it could recognise more varied and/or subtle expressions of emotion, allowing for use in more complex interactions.

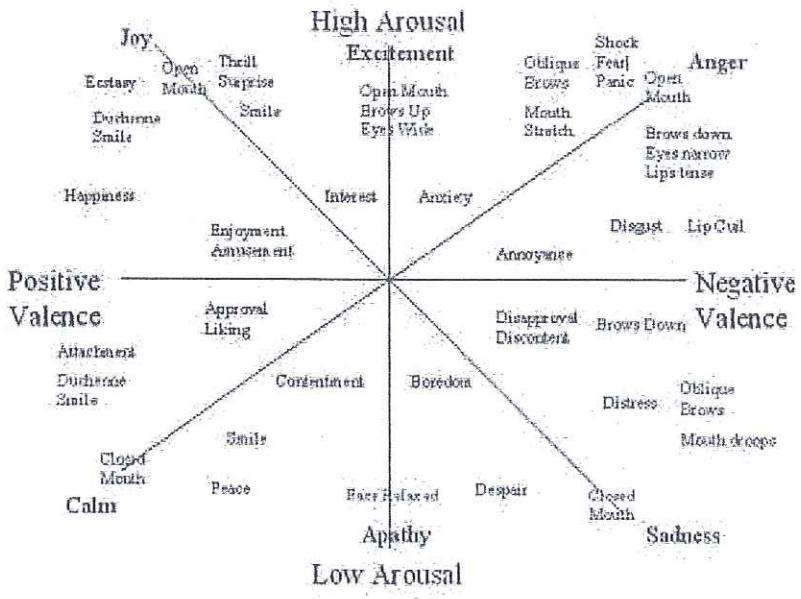
The expression of emotional blends has previously been considered by Cañamero & Fredslund [2001] and Latta et al [2002]. Cañamero & Fredslund produced blends of emotion expression using a LEGO robot, expressing one emotion in the upper part of

the robot's face, and a different emotion in the lower part of the face. The approach taken by Latta et al involved the use of continuous scales, forming an 'emotion-expression space'.

This use of continuous scales is a commonly used alternative to discrete emotions such as the Ekman basic emotions. Frequently used is a 2-dimensional space with the two scales being valence and arousal [Lang 1995]. Valence relates to whether the emotion is positive or negative, whilst arousal relates to the intensity of the emotion. Although more complex, multi-dimensional models [Smith 1985] can be used if finer distinction between emotions is required, the 2-D model is attractive due to its simplicity.

Latta et al [2002] used the 2-D arousal-valence space in the facial expression animation of an avatar. They did this by mapping both discrete emotions and facial expressions onto the 2-dimensional space. The 2-dimensional expression-emotion space used by Latta et al is provided in **figure 9.3**.

Figure 9.3 – Mapping discrete emotions and facial expressions onto the 2-dimensions of affective experience. Taken from Latta et al [2002]



The important point to note here is that, not only are parts of this 2-D space associated with certain distinct emotional states, but also that different parts of the space are associated with different expressions on the face. For example, high arousal and neutral valence is associated with an open mouth, raised brows and wide eyes. Also, the work of Kaiser, Wehrle & Schmidt [1998] has shown that the distribution of AUs from the FACs differ between positive and negative emotions. For example, AUs 6 (cheek raiser) and 12 (lip corner puller) are more commonly seen in the expression of positive emotion.

So why is this link between the emotion map and certain expressions useful to the expression recognition system presented here that is trained to recognise facial expressions as a whole (eg surprised expression) rather than individual components of that expression (eg brows up)? It is because a classifier trained to recognise expression of one of the Ekman emotions (eg surprise) is likely to be more sensitive to one of its component expressive motions (eg raised brow) than a different classifier where such a motion is not present (eg sadness). It is suggested that it may be possible to merge the outputs of all six expression recognising classifiers and map this merged output into an

Figure 9.4 – Responses of the six SVM classifiers to a posed expression of panic.

Classifier Outputs –

Smile:-0.92	Surprise:-0.73	Sadness:-0.88
Disgust:-1.08	Fear:-0.57	Anger:-0.96

In response to this expression of panic the surprise and fear classifiers give the strongest responses. Level of response also varies between the other classifiers.

emotion space such as is provided in **figure 9.3**. This would allow the system to recognise and respond to a greater range of expression types and blends. **Figure 9.4** provides an example of multiple activation of classifiers when exposed to a posed expression of panic. The posed expression was obtained from a drama student asked to

express the emotion panic. In a situation where it is not possible to obtain genuine facial expressions, the use of drama students trained to use their faces expressively seems a good alternative.

Obviously, this mapping approach is non-trivial and is the reason why implementation of such a system is beyond the scope of this thesis. To achieve such a result one would need to:

- Investigate how to effectively merge the outputs of the different classifiers and subsequently map the outputs into an emotion space.
- Obtain example sequences of these different expressions of emotion to learn and then test such a mapping.

It is thought that such an approach could be used to extend the system presented here, allowing it to be used in more complex interactions.

9.6 Summary

Two prototype systems have been presented demonstrating potential applications for the real-time expression recognition system presented here, with results from a questionnaire showing that people believe these applications to be of use. This chapter has also described how this work could be further extended to allow for its use in more complex situations where the recognition of a greater range of expressions is vital.

10 DISCUSSION & SUMMARY

This thesis has introduced a fully automated, real-time expression recognition system able to distinguish between expressions of happiness, sadness, surprise, disgust, fear and anger. Novel applications for such a system have also been introduced. In **section 10.1** the relative strengths and weaknesses of the work presented in this thesis are discussed and contrasted to some previously developed approaches. **Section 10.2** then proposes further work that could be carried out to enhance or expand upon the techniques developed. Finally, a brief summary of the system presented here is given in **section 10.3**.

10.1 Discussion

The optimal absolute recognition rate achieved by the techniques developed in this work was 81.82%. Although this recognition rate is good for a fully automated real-time system, it is lower than that achieved by other researchers using different approaches (see **table 3.1** for recognition rates for these systems). The reasons for this reduced performance may be summarised as follows:

- **Reduced classification rates for fear and anger expressions** – high recognition rates of 80 to 100% were achieved in this work by the classifiers for recognition of the happiness, surprise, disgust and sadness expressions (NB the precise recognition rate varied depending on the exact data representation and classification system used). However, for the expressions of fear and anger the recognition rate was usually around 50% to 60%. It is suggested that this reduced performance is caused by a combination of two factors. The first is that examples of fear and anger expression were relatively few in the CMU-Pittsburgh Expression Database used for training/testing, with 29 and 33 examples respectively. Thus, it was hard for the classifiers to generalise from such a small set of examples (the training sets for fear and anger consisted of only 18 and 16 examples respectively). The second contributing factor is thought to be the difficulty human subjects have in posing the fear and anger expressions. Ask

someone to smile and they will have no problem, whereas ask someone to express fear, even if they are provided with a pre-recorded example, and they will find the task more difficult.

- **Full automation of training and testing phases** – unlike other optical flow work, such as that of Yakoob [1993] and Rosenblum [1994] where boxes bounding facial features are manually positioned at the start of any expression sequence, the classification system presented here is fully automated and relies on the face tracker of **chapter 4** to correctly locate the position of the face for both the training and testing phases. The face tracker provides a rough spatial map of face feature locations but these locations can still be a couple of pixels away from the precise location. This can be significant when one considers that the face size used by the system is only ~35 pixels across. Thus, between different expression examples the positioning of the spatial map used for averaging by the expression classification system varies in position slightly. This reduces performance from that which could be achieved if the locations were precisely labelled manually.

- **Reduced frame rate** – the completed system runs at a frame rate of 4fps. Thus, only four frames of motion information are provided to the classification system for each individual facial expression. The system described here is therefore provided with less motion information than other optical flow-based approaches that are not constrained by the need to operate in real-time.

- **Face size** – to run at a frame rate of 4fps it is only possible to determine the optical flow over a small area. Thus, the size of the template used to characterise the expressions is restricted to 32x38 pixels in size. Therefore only a coarse motion field characterises the movements of the face and hence more subtle motions can be lost. This is likely to be partially responsible for reducing the expression recognition rate.

- **Real-time optical flow algorithm** – the optical flow algorithm used for this work is a real-time implementation of the MCGM. However, the motion output of the real-time version is not as accurate as that of the full MCGM. Hence, minor inaccuracies in motion field output make it more difficult to accurately classify facial expressions.

The majority of the above factors reducing performance have arisen as a direct result of the need here to develop a fully automated, real-time system. However, considering that other systems with higher recognition rates have not had to address such difficulties, it is believed that the performance of the system presented here, with an overall recognition rate of 81.82%, is impressive, particularly when results for the fear and anger expressions are ignored. Also, the technology used here is not the latest available. The Matrox Genesis DSP boards are five years old and the main processor runs at only 450MHz. Thus, using the latest technology it could well be possible to overcome some of the factors hindering performance.

Another issue that must be addressed is that expression classification is constrained to frontal views and a single facial scale. (NB the systems developed by other workers and described in **chapter 3** have also ignored the problems of recognising facial expressions from different viewpoints). However, the main applications for which it is envisaged this system would be used involve the interaction of a human subject with a PC where the user's gaze would almost always be directed at the monitor. Thus, it is acceptable for this system to recognise only expressions from frontal views.

Also, a questionnaire (**appendix 8**) completed by 100 people showed that in response to the question:

"In an interaction when you directly facing a friend, you are more likely to respond to their facial expressions than if you are not directly facing them (1 strongly agree, 5 strongly disagree)."

49% of those questioned strongly agreed, whilst only 4% disagreed and not a single person strongly disagreed (overall mean response of 1.69). These results show that even humans believe frontal views of the face to be important if they are to respond to facial expression.

Regarding the search operating at a single scale, with the user constrained to a seat in front of the computer monitor, a single scale is thought to be sufficient. Remember, although the images from the CMU-Pittsburgh AU-Coded Database were all modified to the same size for training/testing purposes, the heads themselves varied slightly due to normal human variation and pose in the seat where the subjects were filmed. Thus, the performance figures presented in this work have already been affected by small changes in scale of the human head. The preliminary study of **section 9.4** suggests the system has tolerance to scale change ±20% away from optimum and this scale range is sufficient for the situation where subjects are seated in front of a PC. Nonetheless, if further scales were required, the templates used by the face tracker and expression classification systems could easily be doubled, trebled etc in size, but obviously this would require the use of technology with processing power greater than that used here.

This work has also not addressed problems associated with the onset and holding of facial expressions not taking constant periods of time. For instance, one example of a person going from a neutral expression to a happy expression may take 0.8 seconds and another example 1 second. The first example may also show the eyes and mouth of the person 'smiling' at the same time, whereas the second example may show the mouth move before the region around the eyes. Additionally, the expressions could then be held for differing periods before a neutral expression returns to their faces. Thus, an expression recognition system must be able to cope with such changes in dynamics.

However, the work presented here only uses the onset, or start phase of an expression (ie from neutral face to expressive face) and not the end phase (from expression to neutral face). Thus, it operates independently of the length of time an expression is held.

Also, it is thought that the relatively low frame rate of 4fps is helpful in giving the system robustness to changes in expression dynamics, as small changes in the timing of facial movements when expressions are made would be missed due to the low sampling rate.

It is important to emphasise that the timing of the sequences used for training/testing here have not been modified in any way, with the timings solely determined by the period taken by the human subjects themselves to express the facial emotions. Thus, the performance figures presented in this thesis have already been affected by changes in the timings of facial expressions. If, in the future, significant improvements were to be made to the system, it is possible that timing changes could have serious implications for the classification performance of the system. In such a situation the use of time-warping may need to be considered.

Another feature of the system that could be regarded as a weakness is that it requires the use of specialised digital signal processing boards for image processing. In response, it should be pointed out that the technology upon which this work has been implemented is five years old. Already a real-time software version of the MCGM has been developed that operates without the use of specialised imaging boards, and thus it is reasonable to assume that the same would be possible with the system described here, allowing it to run on a normal PC using one of the latest processors available.

The final and, from a human computer interaction viewpoint, perhaps the most important issue is the decision to use the six Ekman basic emotions as the expressions for classification by this system. This can be thought of as a problem as, although all people readily recognise these facial expressions and interpret the underlying emotional state in the same way, apart from happiness these prototypical facial expressions are not often seen in everyday social interaction, with either more subtle expressions or variations of these expression often being used.

For this reason, a large amount of other expression classification research has involved the recognition of Action Units (AUs) from the Facial Action Coding System (FACS). This approach allows facial movements and dynamics to be analysed without making assumptions as to any underlying emotional state. By developing such a system that recognises all AU (and AU combinations) it is hence possible to feed the outputs to a separate emotion interpretation system. Such a system could analyse the expression outputs and then combine it with information in relation to, for example, body pose. Thus, interpretations as to underlying emotional state can be made by use of information accumulated from different sources.

However, the significance of the work carried out in this thesis is the development of a fully automated, real-time system for expression classification and the introduction of applications that respond accordingly. It is not feasible to recognise, analyse and interpret the dynamics of all AUs in real-time, particularly as recognition of some of the more subtle AUs of the FACS would require more detailed information in relation to facial changes than it is possible to obtain at the scale at which this system operates. Additionally, previous research into automated AU recognition that has not been constrained by the need for real-time operation has only ever attempted to recognise a subset of the 44 AUs defined in the FACS.

Thus, the only realistic alternative to recognizing all the AUs of the FACS is the use of the simpler Ekman basic emotions. One reason for choosing them is that, if humans universally interpret these facial expressions in the same way, then they are expressions that are characteristic of the human species as a whole and can be understood by all. Therefore, it is reasonable to expect that an affective computing system should both recognise and interpret the underlying emotional state of someone expressing one of these prototypic expressions as a human would. Also, a weaker but, from the point of view of needing to develop a working system, important reason for choosing the basic emotions is that detailed databases with other facial expressions are not available to the author's knowledge.

Also, in **section 9.5** a description was provided proposing an approach to extending the work presented here such that it may be able to respond to a greater range of expression types than just the six basic emotions. The approach proposed involved the combination of outputs from multiple expression recognising classifiers and subsequent mapping into a 2-dimensional emotion space.

10.2 Further work

Several avenues remain open for further work. These can be summarised as follows:

- **Conversion of the single face tracker into a face detector** – the face tracker described in **chapter 4** assigns each potential face location a face probability score, and then tracks the object with the highest score. Conversion of the tracker into a face detector would require inclusion of a threshold on the face probability score that would allow discrimination between face and non-face matches. Although useful for the expression recognition application described here, it would be preferable to just choose the most face-like location rather than risk failing to detect a face due to the sensitivity of thresholds.

- **Further tracker speed-ups** – in addition to the face tracker speed-ups described in **section 4.6**, frame rate could also be improved by inclusion of a motion pre-filter. Such a pre-filter is used by Scassellati [1998] and allows face search to occur solely at locations where there is motion above the background level. However, this approach has the weakness that a head must move for it to be detected.

- **Separate data representation for different expressions** – for the work described in this thesis the data representation for each expression is identical for any one single approach. However, different facial components do not move in the same way and have varying importance in characterising different facial

expressions. Thus, although the technique used in this work is more quickly processed, it is possible that the introduction of different data representations for each expression could enhance performance further.

- **Improved training and testing sets** – Recognition rates could be improved significantly if the expression database used in this work were to be greater in size. The complete CMU-Pittsburgh AU-Coded Database contains significantly more examples of each facial expression than were made available for the work presented here. By using these, and other sequences, in training the classifiers it is likely that performance would be improved significantly as it would allow the classifiers to generalise more effectively. Also, it would be desirable for these expressions to be spontaneous expressions rather than the posed expressions used in this work

- **Expressions seen when humans interact with technology** – the work presented here attempts to recognise the six emotions universally associated with unique facial expressions. However, there is no reason to believe that all these expressions are made by human subjects whilst interacting with technology. Thus, a study into what expressions are seen when humans interact with technology would be useful in selecting expressions for automated expression recognition. Additionally, it would be interesting to investigate whether this set of expressions is altered and/or people become more expressive facially if they know that technology can understand and respond to facial cues.

- **Role of frame rate** – the system described here operates at a rate of 4 frames per second and this rate is restricted by the available technology. However, the digital processing boards used here are around five years old and the main processor is only 450MHz. Thus, significantly higher frame rates would be possible with the latest technology and it is reasonable to predict that provision

of motion data to the expression recognition system recorded at higher frame rates would help enhance performance.

- **Detailed investigation into efficacy of learning expression features** – some preliminary investigations have been made here into the use of simulated annealing for learning facial regions to average motion data over for the purposes of expression recognition. This approach gave better recognition results then the other empirical techniques applied. Further investigations are therefore merited into learning facial regions when the size and shape of regions are not restricted.

- **Psychology** – during the course of this work some interesting parallels have arisen between the results given by this expression recognition system and the findings of psychological studies of emotion. An example of such a parallel is that the system achieves better recognition performance when provided with data relating to the motion of the left side of the face rather than that of the right and requires motion from both sides of the face to achieve optimal results. These results suggest that expressions are more strongly displayed on the left hand side of the face and that each side of the face carries different information, findings supported by psychological research [Sackheim 1978]. Thus, it is suggested that this work would be of interest and value to psychologists and further work could be undertaken in such a direction. For example, it may be interesting to study in more detail whether, as in humans, the system has difficulties in distinguishing between fear and surprise, and if so to examine the motion signatures (from the MCGM) for these two expressions to look for similarities that make distinction difficult. Another possible avenue would be to examine the correspondences between the regions of importance identified in the human face by the golden ratio face mask and simulated annealing and the regions of the face described by the Action Units of the Facial Action Coding System used in psychological

studies of emotion. The positioning of these Action Units could also be used to guide further changes to the positioning of the regions.

10.3 Summary

Figure 10.1 provides a detailed flowchart summarising the operation of the completed expression recognition system presented that runs in real-time and is fully automated. The system makes use of a novel face tracker that has improved on the ratio template algorithm by modifying the spatial face template to include biological proportions, and has combined it with a number of additional processing stages to determine the most face-like object in the scene, thereby rejecting the numerous false positives given by the ratio template algorithm approach.

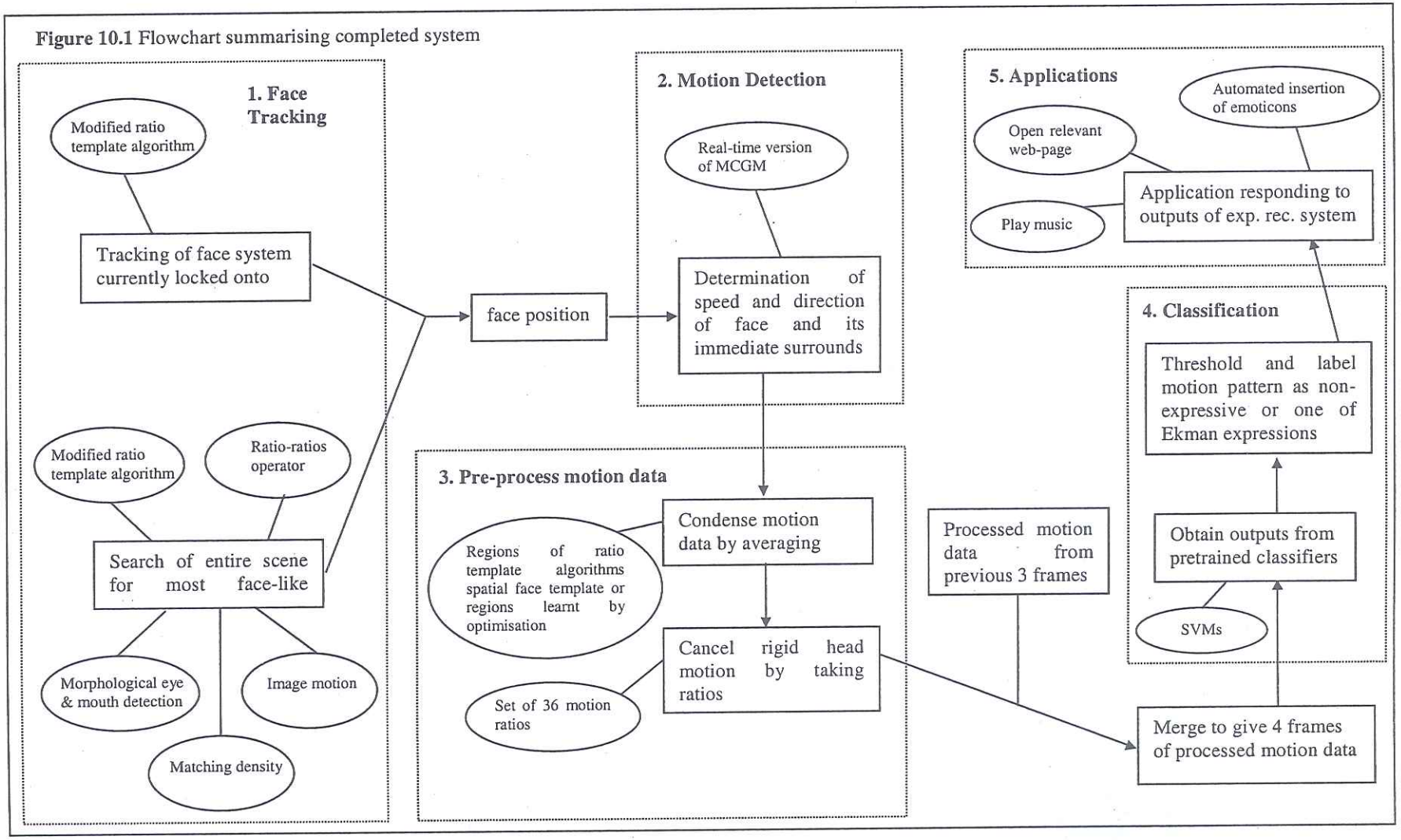
The outputs of the face tracker are then used to gate a real-time version of a MCGM that determines the speed and direction of motion of the face in the scene. This motion data is condensed by averaging over key facial regions and rigid head motion cancelled out by a previously untried technique of taking ratios of averaged motion. As well as using empirical approaches for determining regions for averaging and the ratios to be taken, an optimisation technique (simulated annealing) has been applied in an attempt to further improve the data representation.

The performances of SVMs and MLPs, trained using this motion data, have been compared for the expression recognition task, with SVMs giving slightly better results. Outputs from these classifiers have then been used to drive novel applications. Exemplar applications implemented here include a system for automated inclusion of emoticons into chatroom text and a system that can play music in response to the changing mood of a computer user. An approach to extending the system to allow for the recognition of a greater range of expressions has also been proposed.

The usefulness of systems such as the one described in this thesis is not restricted to the simple prototypical applications previously described. There are a host of other uses in

the fields of affective computing and robotics where these techniques could be used, extended, or combined with other approaches to make interaction with technology both more natural and more rewarding.

Figure 10.1 Flowchart summarising completed system



11 REFERENCES

Aarts E. & Korst J.: Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimisation and Neural Computing. John Wiley & Sons. (1989)

Abdullah M.A.: Inner canthal distance and geometric progression as a predictor of maxillary central incisor width. Journal of Prosthetic Dentistry **88**, 16-20 (2002)

Anderson, A.J. & McOwan, P.W.: Model of a predatory stealth behaviour camouflaging motion. Proc. R. Soc. Lond. B **270**. p489-495 (2003)

Anderson, K. & McOwan, P.W.: Robust Real-Time Face Tracker for Cluttered Environments. Resubmitted with minor corrections to Computer Vision & Image Understanding. (2003)

Anderson, K. & McOwan, P.W.: Changing Faces: The Science of Facial Expression, to appear in Hilla, P.J.: As others see us. Peter Francis Publishers. (2004)

Balakrishnama S. & Ganapathiraju A.: Linear Discriminant Analysis - A Brief Tutorial. http://www.isip.msstate.edu/publications/reports/isip_internal/1998/linear_discrim_analysis/lda_theory_v1.1.pdf (1998)

Barron J.L., Fleet D.J., & Beauchemin S.S.: Performance of Optical Flow Techniques. International Journal of Computer Vision **12**(1), p43-77 (1994)

Bartlett M.S., Donato G., Movellan J.R., Huger J.C., Ekman P., & Sejnowski T.J.: Face Image Analysis for Expression Measurement and Detection of Deceit. Proceedings of the 6th Annual Joint Symposium on Neural Computation. (1999a)

Bartlett M.S., Huger J.C., Ekman P., & Sejnowski T.J.: Measuring Facial Expressions by Computer Image Analysis. Psychophysiology **36**, p253-263. (1999b)

Belhumeur P., Hespanha J., Kriegman D.: Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. PAMI **19**. 711-720 (1997)

Black M. Yacoob Y.: Recognising facial expressions in image sequences using local parameterised models of image motion. International Journal of Computer Vision **25**(1). 23-48 (1997)

Bock J.R. & Gough D.A.: Predicting protein-protein interactions from primary structure. Bioinformatics 17. 455-460 (2001)

Bovis K., Singh., Friedsend J., & Pinder C.: Identification of masses in digital mammograms with MLP and RBF nets. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks **1**. p342-347 (2000)

Bradley A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, **30**(7). 1145-1159 (1997)

Breazeal C., Scassellati B.L.: A context-dependent attention system for a social robot. International Joint Conference on Artificial Intelligence. (1999)

Breazeal C.: Emotion and sociable humanoid robots, International Journal of Human-Computer Studies **59**. (2003) 119-155

Burges C.J.C.: A Tutorial on Support Vectore Machines for Pattern Recognition. Data Mining and Discovery **2**. p121-167 (1998)

Cañamero L.D. & Fredslund J.: I Show You How I Like You-Can You Read it in My Face? IEEE Transactions on Systems, Man and Cybernetics, Part A, **31**(5). p454-459 (2001)

Carlson A.J., Cumby C.M., Rosen J.L. & Roth D.: SNoW User's Guide. UIUC Technical Report. http://l2r.cs.uiuc.edu/~danr/Papers/userguide.ps.gz (1999)

Cerny V.: Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm, Journal of Optimization Theory and Applications **45**. 41-51 (1985)

Charlesworth W.R., Kreutzer M.A.: Facial expression of infants and children. In Ekman P. (editor): Darwin and facial expression. New York Academic Press. 91-168 (1973)

Chen W.M., Wong Y.X., & Ping X.: Flow-shop scheduling by the knowledge of statistical mechanics and annealing. Proceedings of 26th IEEE Conference on Decision and Control. 642-643 (1987)

Cootes T., Taylor C., Cooper D. & Graham J.: Active Shape Models – Their Training and Application. Computer Vision and Image Understanding **61**. p38-59 (1995)

Critchley H.: Emotion and its disorders, British Medical Bulletin **65**. 35-47 (2003)

Dale J.L.: A Real Time Implementation of a Neuromorphic Optic Flow Algorithm. PhD Thesis. University College London. (2002)

Darwin C.: The expression of the emotions in man and animals, New York Philosophical Library. (1872)

Dautenhahn K., Werry I., Salter T., & Te Boekhorst R.: Towards Adaptive Autonomous Robots in Autism Therapy: Varieties of Interactions. IEEE International Symposium on Computational Intelligence in Robotics and Automation. (2003)

DeCoste D & Burl M.C.: Support Vector Machines and Kernel Fisher Discriminants: A Case Study using Electronic Nose Data. Fourth Workshop on Mining Scientific Datasets. (2001)

Demuth H., Beale M.: Neural Network Toolbox Users Guide. The Mathworks Incorporated. (1998)

Duchenne B.: The mechanism of human facial expression or an electro-physiological analysis of the expression of the emotions, Cambridge university press. (1990)

Dumas M.: Emotional expression recognition using Support Vector Machines. citeseer.nj.nec.com/cache/papers/cs/22750/http:zSzzSzwww-cse.ucsd.eduzSzuserszSzelkanzSz254zSzmdumasrep.pdf/dumas01emotional.pdf (2001)

Ekman P., Oster H.: Facial expression of emotion. Annual Review of Psychology, **20**. 527-554 (1979)

Ekman P., Rosenberg E. (editors): What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System, Oxford University Press. (1998a)

Ekman P., Matsumoto D., Friesen W.V.: Facial expression in affective disorders, in Ekman P., Rosenberg E. (editors): What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System, Oxford University Press. (1998b)

Ekman P.: Basic Emotions, in Dalgleish T., Power M. (editors): Handbook of cognition and emotion, John Wiley and Sons Ltd. (1999)

Ekman P.: Facial Expressions, in Dalgleish T., Power M. (editors): Handbook of cognition and emotion, John Wiley and Sons Ltd. (1999)

Elks M.A.: Another look at facial disfigurement. J Rehab. 36-40 (1990)

Essa I.A. & Pentland A.P.: Coding, Analysis, Interpretation, and Recognition of Facial Expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7), p757-763. (1997)

Fasel B. & Lüttin J.: Recognition of asymmetric facial action unit activities and intensities. IDIAP Research Report. (1999)

Fasel B. & Lüttin J.: Automatic facial expression analysis: a survey. Pattern Recognition 36. 259-275 (2003)

Fawcett T.: ROC Graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, Hewlett-Packard Laboratories (2003)

Fernandez R. & Picard R.: Modeling Drivers' Speech under Stress. ISCA Workshop on Speech and Emotions. (2000)

Fernandez-Dols J.M.: Ruiz-Belda M.A.:Are smiles a sign of happiness? Gold medal winners at the Olympic games. Journal of personality and social psychology, **69**. 1113-1119 (1995)

Fidaleo D. and Neumann U.: CoArt: Co-articulation region analysis for control of 2D characters. Proceedings of IEEE Computer Animation 2002. P12-17 (2002)

Fleet D.J. & Jepson A.D.: Computation of Component Image Velocity from Local Phase Information. International Journal Of Comp. Vis. **5**, p77-104 (1990)

Frank M.G. & Ekman P.: the ability to detect deceit generalises across different types of high stake lies. Journal of Personality and Social Psychology **72**. p1429-1439 (1997)

Fridlund A.J.: Human facial expression: An evolutionary view. Academic Press. (1994)

Fridlund A.J.: the sociality of solitary smiles: Effects of an implicit audience. Journal of Personality and Social Psychology **60**. 229-240 (1991)

Ganchrow J.R., Steiner J.E., Daher M.: Neonatal facial expressions in response to different qualities and Intensities of gustatory stimuli. Infant Behaviour and Development **6**. 473-484 (1983)

Gargesha M. & Kuchi P.: Facial expression recognition using artificial neural networks. Artificial Neural Computation Systems. 1-6 (2002)

Georghiades A.S., Belhumeur P.N., Kriegman D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Analysis and Machine Intelligence **23**. 643-660 (2001).

Ghyka M.C.: The geometry of art and life, Dover Publications, 1978.

Green C.D.: All that glitters: a review of psychological research on the aesthetics of the golden section, Perception **24**. 937-968 (1995)

Gunn S.R.: Support Vector Machines for Classification and Regression. University of Southampton Technical Report. www.ecs.soton.ac.uk/~srg/publications/pdf/SVM.pdf (1998)

Hancock P.: Psychological Image Collection at Stirling, http://pics.psych.stir.ac.uk.

Haykin S.: Neural Networks: A Comprehensive Foundation 2nd Edition. Prentice Hall International. (1999)

Hebb D.O.: Organisation of behaviour. New York: Science Editions. (1961)

Hietanen J.K.: Does your gaze direction and head orientation shift my visual attention?, *Neuroreport* **10**, 1999, 3443-3447.

Hill H. & Johnston A.: Categorising sex and identity from the biological motion of faces. Current Biology **11**. p880-885 (2001).

Himer W., Schneider F., Kost G., & Heimann H.: Computer-based Analysis of Facial Action: A New Approach. Journal of Psychophysiology **5**(2), p189-195. (1991)

Hjelmas E., Low B.K.: Face detection: A survey. Computer Vision & Image Understanding **83**. 236-274, doi:10.1006/cviu.2001.0921 (2000)

Hond D., Spacek L.: Distinctive descriptions for face processing. Proceedings of the 8th BMVC, (1997)

Hsu C., Chang C., & Lin C.: A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (2003)

Hsu C.W. & Lin C.J.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks **13**(2). 415-425 (2002)

Hyvärinen A. & Oja E.: Independent Component Analysis: Algorithms and Applications. Neural Networks **13**. p411-430 (2000)

Jain R., Kasturi R., Schunck B.G.: Machine Vision. McGraw-Hill International Editions. (1995)

Joachims T.: Text categorisation with support vector machines. Technical Report No. 23. University of Dortmund.

Joachims T.: Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, (1999a).

Joachims T.: Transductive Inference for Text Classification using Support Vector Machines. International Conference on Machine Learning (1999b)

Johnston A., McOwan P.W.: Benton C.P.: Robust Velocity Computation from a Biologically Motivated Model of Motion Perception. Proceedings of the Royal Society of London **266**. 509-518 (1999)

Kaiser S. & Wehrle T.: Automated coding of facial behavior in human-computer interactions with FACS. Journal of nonverbal behaviour **16**(2). 67-83 (1992)

Katsikitis M. & Pilowsky I.: A study of facial expression in Parkinson's disease using a novel microcomputer-based method. Journal of Neurology, Neurosurgery, and Psychiatry **51**. 362-366 (1988)

Kawato S. & Ohya J.: Real-time Detection of Nodding and Head-shaking by directly Detecting and Tracking the "Between-Eyes". Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition. (2000)

Kirkpatrick S., Gelatt Jr. C.D., & Vecchi M.P.: Optimization by Simulated Annealing. Science **220**. 671-680 (1983)

Kraut R.: Social presence, facial feedback, and emotion. Journal of Personality and Social Psychology **42**. 853-863 (1982)

Lam K. & Yan H.: An Improved Method for locating and Extracting the Eye in Human Face Images. IEEE Proceedings of ICPR 1996. p411-415 (1996)

Lam K. & Li Y.: An Efficient Approach for Facial Feature Detection. Proceedings of ICSP 98, p1100-1103. (1998)

Lang P.J.: The Emotion Probe: Studies of motivation and attention. A study in the Neuroscience of Love and Hate. Lawrence Erlbaum Associates. (1995)

Lanitis A., Taylor C.J., & Cootes T.F.: Automatic Interpretation and Coding of Face Images Using Flexible Models. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7), p743-756. (1997)

Latta C., Alvarado N., Adams S.S., & Burbeck, S.: An expressive system for animating characters or endowing robots with affective displays. Society for Artificial Intelligence and Social Behavior (AISB), 2002 Annual Conference, Symposium on Animating Expressive Characters for Social Interactions. (2002)

Lawrence K., Campbell R., Swettenham J., Terstegge J., Akers R., Coleman M., Skuse D.: Interpreting gaze in Turner syndrome: impaired sensitivity to intention and emotion, but preservation of social cueing, Neuropsychologia **41**. 894-905 (2003)

Lien J.J, Kanade T., Cohn J.F., & Li C.: Automated Facial Expression Recognition Based on FACS Action Units. Third IEEE International Conference on Automatic Face and Gesture Recognition. p390-395 (1998)

Lien J.J.: Automatic recognition of facial expression using Hidden Markov Models and estimation of expression intensity. PhD thesis, CMU. 1998b

Liu C.H., Collin C.A., Burton A.M., & Chaudhuri A.: Lighting direction affects recognition of untextured faces in photographic positive and negative, Vision Research **39**. 4003-4009 (1999)

Liu H., Hong T., Herman M., & Camus T.: accurace vs Efficiency Trade-offs in Optical Flow Algorithms. Computer Vision & Image Understanding **72**(3). p271-286 (1998)

Luger G.F. & Stubblefield W.A.: Artificial Intelligence, structures and strategies for complex problem solving. Benjamin/Cummings Publishing, 2nd edition. (1993)

Lukas B. & Kanade, T.: An iterative image registration technique with an application to stereo vision. Procedures of the DARPA Image Understanding Workshop. 121-130 (1981)

Lyons M. & Akamatsu S., Kamachi M., & Gyoba J.: Coding Facial Expressions with Gabor Wavelets. Proceedings of the Third International Conference on Automatic Face and Gesture Recognition. p200-205 (1998)

Kaiser S., Wehrle T. & Schmidt S.: emotional episodes, facial expressions, and reported feelings in human-computer interactions. Proceedings of the 10th Conference of the International Society for Research on Emotions. p82-86 (1998)

Kass, M., Witkin, A.P., and Terzopoulos, D., Snakes: Active Contour Models, Int. Journal. Computer Vision **1**(4). p321-331 (1988)

Katsikitis M. & Pilowsky I.: a study of facial expression in Parkinson's disease using a novel microcomputer-based method. Journal of Neurology, Neurosurgery, and Psychiatry **51**. p362-366 (1988)

LeCun Y. & Bengio Y.: Convolutional networks for images, speech, and time series, in editor Arbib M.: The Handbook of Brain Theory and Neural Networks. MIT Press (1995)

Marquardt S.R.: MBA website, www.beautyanalysis.com

Martinez A.M., Benavente R.: The AR face database. CVC Technical Report **24**. 1998.

Mase K. & Pentland A.: Recognition of facial expression from optical flow IEICE Trans E **74**(10). 408-410 (1991)

Matsugu M., Mori K., Mitari Y., Kaneda Y.: Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks **16**. 555-559 (2003)

McCulloch W.W. & Pitts W.: A Logical Calculus of the Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics **5**. p115-133 (1943)

Mukherjee D., Deng Y. & Mitra S.K.: A Region-based Video Coder Using Edge Flow Segmentation and Hierarchical Affine Region Matching. Proc. of SPIE, Visual Communications and Image Processing **3309**. p.338-49 (1998)

McOwan P.W., Benton C., Dale J., Johnston A.: A multi-differential neuromorphic approach to motion detection, International Journal of Neural Systems **9**. 429-434 (1999)

Miller R.E.: Experimental studies of communication in the monkey, in Rosenblum L.A. (editor): Primate behaviour: Developments in field and laboratory research. Academic Press. 139-175 (1971)

Minsky M.L. & Papert S.: Perceptrons. MIT Press. (1969)

Moghaddam B. & Pentland A.: Probabilistic Visual Learning for Object Detection. 5th International Conference on Computer Vision. p786-793 (1995)

Morik K., Brockhausen P., & Joachims T.: Combining statistical learning with a knowledge-based approach – a case study ion intensive care monitoring. Proceedings of 16th International Conference on Machine Learning (1999).

Ng J. & Gong S.: Performing multi-view face detection and pose estimation using a composite support vector machine across the view sphere. Proc. IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. p26-27 (1999)

Ogden, B.: Interactive Vision in Robot-Human Interaction. Progression Report, 42-55. (2001)

Oster H., Hegley D., Nagel L.: Adult judgements and fine-grained analysis of ifant facial expressions: Testing the validity of a priori coding formulas. Developmental Psychology **28**. p1115-1131 (1992)

Padgett C. & Cottrell G.W.: A Simple Neural Network Models Categorical Perception of Facial Expressions. Proceedings of the Twentieth Annual Cognitive Science Conference. (1998)

Parker D.B.: Learning Logic. Invention Report. Stanford University (1982)

Picard R.W.: Affective Computing. MIT Press. (1995)

Picard R.W.: Towards Agents that Recognize Emotion. Actes Proceedings IMAGINA. p153-155. (1998)

Pitts W. & McCulloch W.W.: How we Know Universals. Bulletin of Mathematical Biophysics **9**. p127-147 (1947)

Psarrou A., Gong S. &Walter M.: Recognition of human gestures and behaviour. Image and Vision Computing **20**. p349-358 (2002)

Rabiner L.R. & Juang B.H.: An introduction to hidden Markov models. IEEE ASSP Magazine. p4-16 (1986)

Raducanu B., Grana M., Albizuri F.X., & d'Anjou A.: Face Localisation Based on the Morphological Multiscale Fingerprints. Pattern Recognition Letters **22**. p359-371. (2001)

Rosenblum M., Yacoob Y., & Davis L.: Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture. IEEE Workshop on Motion of Non-Rigid and Articulated Objects. (1994)

Rowley H.A., Baluja S., & Kanade T.: Neural Network-Based Face Detection. IEEE Transactions on Pattern Anal. Machine Intelligence **20**(1). (1998)

Rumelhart D.E., Hinton G.E., & Williams R.J.: Learning internal representations by error propagation. Parallel Distributed Processing **1**. p318-362 (1986)

Russell J.A.: Is there universal recognition of emotion from facial expression? A review of the cross-cultural studies, Psychological Bulletin **115**. 102-142 (1994)

Russell J.A., Fernandez-Dols J.M.: What does a facial expression mean, in Russell J.A., Fernandez-Dols J.M. (editors): The psychology of facial expression, Cambridge University Press. (1997)

Russell S. & Norvig P.: Artificial Intelligence: A Modern Approach. Prentice Hall. (1995)

Sackheim H.A., Gur R.C. Saucy M.C.: Emotions are expressed more strongly on the left side of the face. Science **202**. (1978)

Samal A., Iyengar P.A.: Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern Recognition **25**. 65-77 (1992)

Scassellati B.: Eye finding via face detection for a foveated, active vision system. Proceedings of the Fifteenth National Conference on Artificial Intelligence. (1998)

Schmidt M.: Identifying speaker with support vector networks. Interface 1996 Proceedings. (1996)

Schurter R.: Emoticons.
www.geo.unizh.ch/elearning/didactica/Downloads/Kommunikation/Emoticons.pdf
(2003)

Shortliffe E.H.: MYCIN: Computer-Based Medical Consultations. American Elsevier. (1976)

Shwartz G.E. Ahern G.L. Brown S.L.: Lateralized facial muscle response to positive and negative emotional stimuli. Psychophysiology **16**. 561-571 (1979)

Sinha P.: Perceiving and recognising three-dimensional forms, PhD dissertation, M.I.T. (1995)

Sinha V.: The facial nerve and the muscles of facial expression. anatome.ncl.ac.uk/tutorials/facial/index.html (2001)

Smeraldi F., Capdevielle N. & Bigun J.: Face Authentication by retinotopic sampling of the Gabor decomposition and Support Vector Machines. Proceedings of the 2nd International Conference on Audio and Video Based Biometric Person Authentication. p125-129 (1999)

Smith C. A. & Ellsworth, P. C.: Patterns of cognitive appraisal in emotion. Journal of Personality and Social Psychology **48**. 813-838 (1985)

Smith L.I.: A Tutorial on Principal Components Analysis. www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf (2002)

Strack F., Martin L., Stepper S.: Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis. Journal of Personality and Social Psychology **54**. 768-777 (1988)

Sundermann E. & Lemahieu I.: PET Image Reconstruction Using Simulated Annealing. Proc. SPIE Medical Imaging '95 (Image Processing). p378-386 (1995)

Suwa M., Sugie N., & Fjimora K.: A preliminary note on pattern recognition of human emotional expression. Proceedings of the 4th international conference on pattern recognition. p408-410 (1978)

Szu H. &Hartley R.: Fast simulated annealing. Physics Letters A **122**. 157-162 (1987)

Tartter V.C.: Happy talk: Perceptual and acoustic affects of smiling on speech. Perception and Psychophysics **27**(1). 24-27 (1980)

Tartter V.C., Braun D.: Hearing smiles and frowns in normal and whisper registers. JASA **96**(4). 2101-2107 (1994)

Tian Y, Kanade T., & Cohn J.F.: Recognizing Action Units for Facial Expression Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(2). p97-115 (2001)

Tovée M.J.: An introduction to the visual system. Cambridge University Press. (1996)

Turk M., Pentland A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3**. 71-86 (1991)

Vapnik V.N.: The Nature of Statistical Learning Theory. Springer.(1995)

Viola P. and Jones M.J.: Robust real-time object detection, 2nd International Workshop on Theories of Visual Modelling, Learning, Computing, and Sampling. (2001)

Wang J. & Sung E.: Frontal-view Face Detection and Facial Feature Extraction using Color and Morphological Operations. Pattern Recognition Letters **20**, p1053-1068. (1999)

Wang J. & Tan T.: A New Face Detection Method Based on Shape Information. Pattern Recognition Letters **21**(6-7). p463-471 (2000)

Wasserman P.D.: Neural Computing Theory and Practise. Van Nostrand Reinhold. (1989)

Werbos P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences. Masters Thesis, Harvard University. (1974)

Widrow B. & Hoff M.: Adaptive switching circuits. IRE WESCON Convention Record. p96-104 (1960)

Windeatt T. & Ghaderi R.: Adaboost and neural networks. European Symposium on Artificial Neural Networks. p123-128 (1999)

Woitaszek M, Shaaban M, Czernikowski R.: Identifying junk electronic mail in Microsoft outlook with a support vector machine. Proceedings 2003 Symposium on Applications and the Internet. p166-169 (2003)

Yacoob Y. & Davis L., Recognizing Facial Expressions by Spatio-Temporal Analysis. IEEE CVPR. p70-75 (1993)

Yamakawa T; Matsumoto G.: Fuzzy Hough transform, linguistic sets and soft decision MLP for character recognition. Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems. Methodologies for the Conception, Design and Application of Soft Computing **2**. p975-978 (1998)

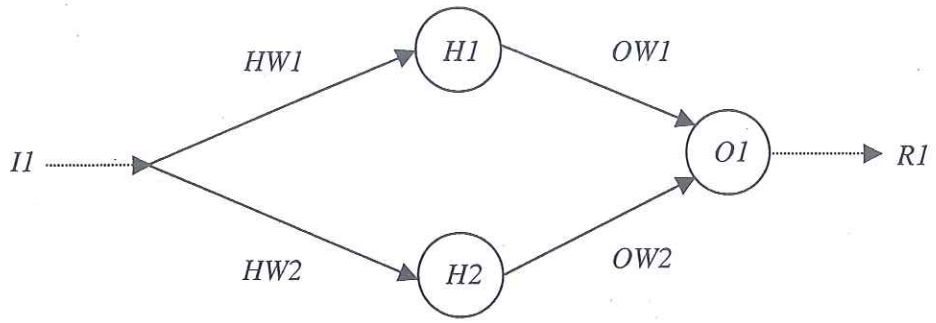
Yang M., Roth D., Ahuja N.: A SnoW-based face detector, NIPS **12**. 855-861 (2000)

Zajonc R.B., Murphy S.T., Inglehart M.: Feeling and facial efference: Implications of the vascular theory of emotions, Psychological Review **96**. 395-416 (1989)

APPENDIX 1 – The Back Propagation Training Algorithm

This appendix provides a worked example of an iteration of the back propagation training algorithm operating on a MLP. For a detailed introduction to the concepts of MLPs and back propagation see **chapter 6**.

Consider the following multi layer perceptron, MLP1, with a single input, I1, two hidden nodes, H1 & H2, and an output node, O1, outputting the response, R1. The logistic activation function is used, and the initial weights, OW1 & OW2, have been chosen at random.



Initial weights: HW1=0.34, HW2=-0.23,OW1=-0.05,OW2=0.19

Learning Rate, L = 0.1

Suppose the training input, I1, is 0.4, and the desired output, D1, is 1.0. Training using back propagation operates in the following order:

1) **Calculate MLP output**:

The output of a node in a MLP is a function of the sum of the weighted inputs. With a logistic activation function this equates to:

(1) *output,* $O_{(n)} = 1/(1+e^{x})$*, where x is:*

(2) $x = \sum_{b=1}^{a} I_b H_{b,}$ *where I is the node input, H is the weight corresponding to that input, and a is the number of node inputs.*

In terms of the specific example, MLP1, one must first calculate the outputs of H1 and H2. As there is only one input for H1 and H2, their outputs are as follows:

$$H1\ output = 1/(1+e^{(HW1xI1)}) = 1/(1+e^{(0.34x0.4)}) = 0.53$$
$$H2\ output = 1/(1+e^{(HW2xI1)}) = 1/(1+e^{(-0.23x0.4)}) = 0.48$$

Using these outputs as inputs for O1 one gets:

$$O1\ output = 1/(1+e^{((OW1xH1)+(OW2xH2))}) = 1/(1+e^{((-0.05x0.53)+(0.19x0.48))}) = 0.516$$

Therefore the MLP1 response, R1, to the input 0.4 is:

$$R1 = O1\ output = 0.516$$

2) Calculate the gradient of the error function in the output layer, O1:

The output of the MLP is a function of the MLP weights, so by modifying these weights it is possible to reduce the error to an input vector. The back propagation training algorithm does this by trying to move to the lowest point on an error surface by determining the gradient of the error function with respect to each weight in the MLP. By moving each weight in the opposite direction to the gradient, one moves the current position on the error surface further down the slope towards the bottom, thereby reducing the error.

Calculation of the gradient is relatively simple, as the derivative of the logistic function can be expressed in terms of the function itself. The gradient of the error function of a node in the output layer with respect to a weight, *Wij*, equates to:

(3) $\delta E(n) / \delta W_{ij}(n) = -(O_j(n) * (D_j(n)-O_j(n)) * (1-O_j(n))) * O_i(n)$, *where* $O_j(n)$ *is the output of a node in the output layer j at time n,* $D_j(n)$ *is the desired output of that node, and* $O_i(n)$ *is the output of the node in the previous layer corresponding to weight* W_{ij}.

NB in this and all subsequent equations, h, i, j refer to consecutive layers of a MLP, moving from input to output layers

In relation to MLP1, the gradient of the error function with respect to the weights OW1 and OW2 at the output layer is:

Gradient with respect to OW1 $= -(R1 * (D1-R1) * (1-R1)) * H1output$
$= -(0.516 * (1-0.516) * (1-0.516)) * 0.53$
$= - 0.064$

Gradient with respect to OW2 $= -(R1 * (D1-R1) * (1-R1)) * H2output$
$= -(0.516 * (1-0.516) * (1-0.516)) * 0.48$
$= -0.058$

3) Calculate Weight Changes for Output Layer:

Make the weight change in the opposite direction to the slope, proportional to the learning rate:

(4) *Weight Change,* $\Delta W_{ij}(n) = -L \times \delta E(n) / \delta W_{ij}(n)$ *, where* $\delta E(n) / \delta W_{ij}(n)$ *is the gradient of the error function with respect to* $W_{ij}(n)$ *and L is the learning rate*

Thus, the weight changes for OW1 and OW2 are:

$$\Delta W \text{ for } OW1 = -0.1 * -0.064 = 0.0064$$
$$\Delta W \text{ for } OW2 = -0.1 * -0.058 = 0.0058$$

4) Make Weight Changes in Output Layer:

Modify weights according to calculated weight change. So the new weight at time n+1 is:

(5) $W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n)$, *where* $W_{ij}(n)$ *is the weight at time n and* $\Delta W_{ij}(n)$ *is the calculated weight change at time n*

Thus for OW1 & OW2:

$$New\ OW1 = OW1 + (\Delta W \text{ for } OW1) = -0.05 + 0.0064 = -0.0436$$
$$New\ OW2 = OW2 + (\Delta W \text{ for } OW2) = 0.19 + 0.0058 = 0.1958$$

5) Calculate the Gradient of the Error Function in the Hidden Layers:

The error signal is then passed backwards through the network a layer at a time, thereby giving back propagation its name.

The gradient of the error function with respect to weights in a hidden layer is:

(6) $\delta E(n) / \delta W_{hi}(n) = O_i(n) * (1- O_i(n)) * O_h(n) * -(\sum_{j=1}^{a} \delta_j(n) * W_{ij}(n))$, *where* $\delta j(n)$ *is:*

$\delta_j(n) = (O_j(n) * (D_j(n)-O_j(n)) * (1-O_j(n)))$, *definitions of variables as in equation (3).*

and $O_i(n)$ is output of a node in the hidden layer i at time n, and $O_h(n)$ is the output of the node in the previous layer corresponding to weight W_{hi}.

Thus, for HW1 and HW2:

Gradient with respect to HW1
*= H1output * (1-H1output) * I1 * -(R1 * (D1-R1) * (1-R1) * OW1)*
*= 0.53 * (1-0.53) * 0.4 * -(0.516 * (1-0.516) * (1-0.516) * -0.05)*
*= 0.53 * 0.47 * 0.4 * 0.006*
= 0.000602

Gradient with respect to HW2
*= H2output * (1-H2output) * I1 * -(R1 * (D1-R1) * (1-R1) * OW2) =*
*= 0.48 * (1-0.48) * 0.4 * -(0.516 * (1-0.516) * (1-0.516) * 0.19) =*
*= 0.48 * 0.52 * 0.4 * -0.0230 =*
= -0.002293

6) **Calculate and make weight changes in hidden layer:**
Using equations (4) and (5):

*New HW1 = HW1 + (-L * Gradient with respect to HW1)*
*= 0.34 + (-0.1 * 0.000602)*
= 0.3399398

*New HW2 = HW2 + (-L * Gradient with respect to HW2)*
*= -0.23 + (-0.1 * -0.002293)*
= -0.2297707

7) **Repeat until mean squared error acceptable:**
This process continues with different training examples until the mean squared error reaches an acceptable level (for discussion of different approaches to ending MLP training see **section 6.1.8.1**). The mean squared error is defined as:

Mean Squared Error = $\sum_{a=1}^{b}(D_a(n) - O_a(n))^2$, *where* $D_a(n)$ *is the desired output node a in the output layer,* $O_a(n)$ *is the actual output of node a in the output layer, and b is the number of nodes in the output layer.*

To demonstrate the effect of the described training iteration, the mean squared error for input I1 prior to the previous round of training was:

$$mean\ squared\ error = (D1\text{-}R1)^2 = (1.0\text{-}0.516)^2 = 0.234256$$

The squared error for input I1 after the round of training becomes:

$$mean\ squared\ error = (1.0\text{-}0.518)^2 = 0.232324$$

given an output of 0.518 by MLP1 with modified weights. As can be seen, the changes made to the weights by the back propagation algorithm have reduced the mean squared error for training example I1.

Momentum

The round of training above demonstrates training of the most basic type of MLP. However, other techniques such as bias (see **section 6.1.2**) and momentum [Rumelhart 1986] are generally included to speed up training and improve the stability of the training process. The use of momentum simply involves adding a term to the weight change calculation of equation (5) such that an additional adjustment is made proportional to the previous change in weight. The weight change when momentum is used is:

$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) + (M*\Delta W_{ij}(n-1))$, *where* $W_{ij}(n)$ *is the weight at time n,* $\Delta W_{ij}(n)$ *is the calculated weight change at time n, M is*

the momentum rate and $\Delta W_{ij}(n-1)$ is the weight change made at the previous time step.

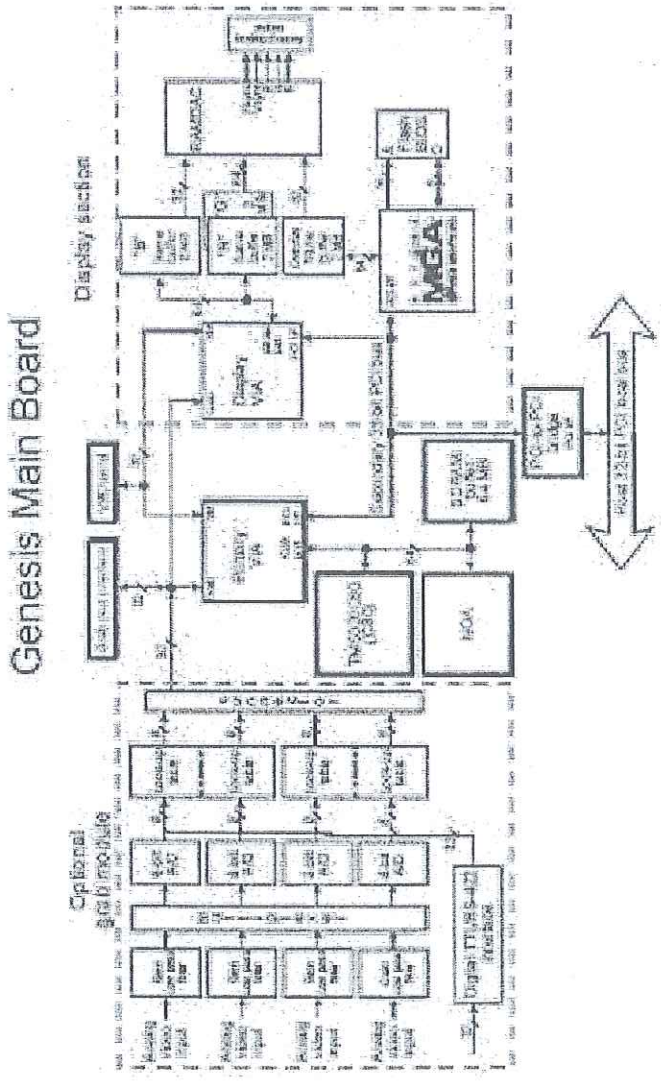
This momentum term not only has a stabilising effect on the training process, but may also stop the learning process getting stuck in a shallow local minimum of the error surface [Haykin 1999].

APPENDIX 2 - Matrox Genesis Imaging Boards

The use of the Matrox Genesis Imaging boards is a vital aspect of the real-time nature of the work presented in this thesis. The Matrox Genesis board is a single slot PCI board, and is designed for the integration of processes for image acquisition, display and processing. Processing is performed by the Texas Instruments TMS320C80 running at 50MHz. It contains 5 programmable processors, consisting of a single RISC master processor and four parallel processors, and is capable of grabbing from virtually any video stream (an XC-ST70 CCD monochrome camera is used in the work presented here). There are also 64 Mbytes of on-board processing memory and any application requiring more can access the host's memory over a PCI bus at high speed. This data transfer takes place at a speed of around 80 Mbytes/sec.

The Matrox Genesis DSP board also has its own C-callable, board-specific library consisting of an extensive range of functions for image processing and other specialised operations. This library has been specifically designed for the development of fast applications. Available functions include template matching and morphological and geometric operations. As well as having a whole range of predefined kernels for carrying out such functions, custom written alternatives may also be incorporated, with only a slight cost in terms of speed. A diagram showing the architecture of the Matrox Genesis main board is given in **figure A2.1** and a detailed performance specification can be found at www.matrox.com/imaging/support/old_products/genesis/b_genesis.pdf.

Figure A2.1 – Architecture of the Matrox Genesis main board. Image taken from the Matrox Genesis Library Command Reference.



APPENDIX 3 - CMU-Pittsburgh AU-Coded Database

To allow effective training and testing of the classifiers used by this system it is necessary to have a reasonably extensive facial expression database with numerous examples of each facial expression. A limited number of these databases have been set-up in the past, and it was decided to use perhaps the most complete of these, the CMU-Pittsburgh AU-Coded database.

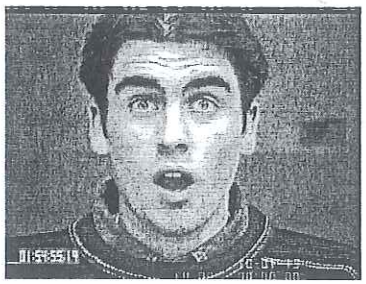
The CMU-Pittsburgh AU-Coded face expression database [Kanade et al 2000] consists of 1917 sequences of 210 adults ranging between the ages of 18 and 50 years old. 69% of subjects are female, 31% male. The majority of subjects are Euro-Americans (81%), with the remainder consisting of mainly Afro-Americans. The sequences have been recorded at a frame rate of 40fps using Panasonic WV3230 cameras connected to a AG-7500 video recorder. The subjects are lit either by ambient room lighting with a single high intensity lamp, or by two high intensity lamps with reflective umbrellas.

Only a portion of the database has been provided by the database's authors for work here. This portion consists of 490 sequences of 97 subjects filmed from directly in front. The subjects are seen to perform a range of facial displays, starting from a neutral expression, and FACS codes produced by certified FACS coders were provided for each of these sequences.

A large number of sequences containing AU combinations representing examples of the six prototypic expressions were extracted for use in training. 57 sequences of the expression of happiness, 49 of sadness, 55 of surprise, 30 of disgust, 33 of anger, and 29 of fear were selected. The size and frame rate of these sequences were reduced to that at which the system runs. An example image of each facial expression taken from this database is provided in **figure A3.1**.

Figure A3.1 – Examples of expressions from The CMU-Pittsburgh AU-Coded face expression database

Happiness



Surprise

Sadness

Disgust

Fear

Anger

APPENDIX 4 – Action Units of Facial Action Coding System (FACS)

AU Number	Descriptor	Muscle/muscles responsible for action
1	Inner brow raiser	Frontalis
2	Outer brow raiser	Frontalis
4	Brow lowerer	Depressor Glabellae, Depressor supercilli, Corrugator
5	Upper lid raiser	Levator palpebrae superioris
6	Cheek raiser	Orbicularis oculi
7	Lid tightener	Orbicularis oculi
9	Nose wrinkler	Levator labii superioris, Alaeque nasi
10	Upper lip raiser	Levator labii superioris, Caput infraorbitalis
11	Nasolabial fold deepener	Zygomatic major
12	Lip corner puller	Zygomatic minor
13	Cheek puffer	Caninus
14	Dimpler	Buccinator
15	Lip corner depressor	Triangularis
16	Lower lip depressor	Depressor labii
17	Chin raiser	Mentalis
18	Lip puckerer	Incisivii labii superioris, Incisivii labii inferioris
19	Tongue out	-
20	Lip stretcher	Risorius
21	Neck tightener	-
22	Lip funneler	Orbicularis oris
23	Lip tightener	Orbicularis oris
24	Lip pressor	Orbicularis oris
25	Lips part	Depressor labii, or Orbicularis oris
26	Jaw drop	Masetter, Temporal & Internal pterygoid relaxed
27	Mouth stretch	Pterygoids, Digastric
28	Lip suck	Orbicularis oris
29	Jaw thrust	-
30	Jaw sideways	-
31	Jaw clencher	-
32	Lip bite	-
33	Cheek blow	-
34	Cheek Puff	-
35	Cheek suck	-
36	Tongue blow	-
37	Lip wipe	-
38	Nostril dilator	-
39	Nostril compressor	-
41	Lip droop	-
42	Slit	-
43	Eyes closed	-
44	Squint	-
45	Blink	-
46	Wink	-

APPENDIX 5 – Motion Ratios

Consider the following situation, with a subject smiling. In such a situation, optical flow outputs to be expected in the mouth region are shown in **figure A5.1a**. Then consider averaging of the motion (**figure A5.1b**), an approach used by this system, which gives motion values for each region, as shown in **table A5.1**.

Figure A5.1 (a) Optical flow outputs around mouth for a smile, lines indicate direction and speed of motion (b) average the motion

Table A5.1 Averaged motion values with a stationary head

Region	Direction	Speed (pixels)
Right cheek	320°	14
Left cheek	45°	15
Chin	5°	7

For heads that are held stationary during expression, these figures would remain fairly constant between examples of different peoples' smiles, and could be effectively used as inputs to a classification system. However, consider a situation where a head moves upwards by 5 pixels during a smile expression. The new values for speed and direction of motion are given in **table A5.2**.

Table A5.2 Averaged motion values with a moving head

Region	Direction	Speed (pixels)
Right cheek	330.2°	18.1
Left cheek	34.2°	18.9
Chin	2.92°	12.0

Notice the effect on the value of both the speed and direction values just by this small amount of rigid head motion. To cancel out this effect it is possible to take ratios of head motion. For instance, consider taking ratios of motion of the right cheek to chin with this rigid head motion.

Right cheek: Direction 330.2°, Speed 18.1

Chin: Direction 2.92°, Speed: 12.0

Use simple trigonometry to determine ratio of these two motions:

*Ratio Dir = atan(((sin(330.2)*18.1)-(sin(2.92)*12.0))/((cos(330.2)*18.1)-(cos(2.92)*12.0)))*

Ratio Dir = 291.3 °

Ratio Speed = $\sqrt{(((sin(330.2)*18.1)-(sin(2.92)*12.0))^2 + ((cos(330.2)*18.1)-(cos(2.92)*12.0))^2)}$

Ratio Speed = 12.0

If the same calculation is performed on the values obtained when there is no rigid head motion:

Right cheek: Direction 320°, Speed 14

Chin: Direction 5°, Speed: 7

Ratio Dir = 291.3 °

Ratio Speed = 12.0

Notice that, by taking ratios of motion, the effects of rigid head movement are cancelled out, and this cancelling works independently of the direction and speed of the rigid head movement. Please note that this approach could also be an effective way of compressing the data. Following the above calculation, a single direction value has encoded inside it information relating to the direction and speeds of two parts of the face (ie what was originally four values is now incorporated into a single value).

APPENDIX 6 - Glossary of Techniques

Active Contour Models - solve an energy minimisation problem and were first proposed by Kass [1988] for use in image segmentation and understanding. They are also commonly referred to as 'snakes' due to the movements seen as they lock onto objects. An initial spline (snake) is placed and then an energy minimisation process is carried out, attempting to minimise the energy of two forces, internal and external. The external forces come from the image structure, whilst the internal forces derive from the physical properties of the model (eg elastic forces).

Active Shape Models – proposed by Cootes [1995], Active Shape Models are similar to Active Contour Models. However, Active Shape Models restrict their movement such that they can only deform in ways found in a training set. Hence, Active Shape Models are specific to the object they attempt to represent.

AdaBoost – is an approach that allows one to combine a number of 'weak' learners to solve more complex problems [Windeatt 1999]. Whilst training a 'weak' learner each example in the training set is assigned a weight. Initially all weights are equal, but in each round of training the learner returns a hypothesis, and all examples miss-classified by that hypothesis have their weights increased. Thus learners are forced to focus on the more difficult examples in the training set.

Each learner is trained using an identical feature set, but on a different training set. The outputs of the 'weak' learners are then combined using a weighting strategy that modifies the weights according to the performance of each 'weak' learner over its training set.

Conjugate Gradient Descent – this approach addresses the issue that back propagation training tends to zigzag across the error surface to the error minimum. Conjugate gradient descent smoothes this movement, thereby speeding up the training process. It does this by using not only the current error gradient but also previous error gradients, reinforcing

weight change according to previously successful movements. The approach is discussed in more detail in Haykin [1999].

Convolutional Networks – are a class of multi-layer perceptron. One problem that multi-layer perceptrons have when applied to image-based applications is that they have no in-built invariance to translation or distortion. Convolutional networks are specifically designed to address this shortcoming, recognising 2-D shapes in images with considerable invariance to translation, scaling and skewing [Haykin 1999]. Convolutional networks achieve this by use of localised receptive fields, shared weights, and sub-sampling. Use of localised receptive fields forces the network to extract local features (such as oriented edges). Sharing weights reduces the number of free variables, improving generalisation performance, and also gives intrinsic insensitivity to translation of the input (as weights are shared over different parts of the image). Sub-sampling means that successive layers perform pattern recognition at steadily larger spatial scales (but at lower resolution), reducing sensitivity to shifts and distortions [LeCun 1995].

Fisher Linear Discriminants Analysis – is a tool for multi-class classification and data reduction. The idea behind linear discriminants is to find a new representation for data that is more efficient for discrimination. It does this by looking for a direction in the initial feature space where the data can be easily separated [DeCoste 2001]. The Fisher Linear Discriminant does this by finding the direction in feature space that results in the optimal ratio between the separation between means of different classes (this should be large) and the scatter around these means (this should be small).

Hidden Markov Models (HMM) – HMMs are a tool for modelling probability distributions over sequences of observations [Rabiner 1986]. They consist of a finite set of states and at each time interval a new state is entered according to a transition probability distribution dependent on the previous state. After each transition is made an observation is output according to a probability distribution dependent on the current state. These probability distributions satisfy the Markov property in that they remain the same

regardless of all previous states. HMMs are called hidden as their outputs are made without the underlying state being revealed.

Once set up, HMMs can be used for recognition tasks. For instance, consider the word recognition task. A set of observations is generated that characterises an unknown word. One can then determine the probability that each of a set of HMMs (each set up to recognise a different word) would generate this set of observations. One can then label the unknown word according to the HMM giving the highest probability.

Independent Components Analysis (ICA) – is a statistical technique for decomposing a complex set of data into components that are maximally independent of each other. It can be seen as an extension to PCA. However, ICA is a more powerful technique, succeeding where PCA can sometimes fail. Like PCA, ICA decorrelates the signals, but it also reduces higher-order statistical dependencies, attempting to make the signals as independent as possible. The directions of the axes are determined by both the second and higher order statistics of the original data.

Morphological Operators – the two main morphological operators are erosion and dilation, and both involve the passing of a structuring element over an image. By using different structuring elements, dilation tends to grow bright areas of the image and erosion tends to shrink bright regions of an image [Jain 1995].

These two operators are commonly combined in binary images to find object boundaries [Lam 1998]. The combination process involves the subtraction of an eroded binary image from a dilated binary image. The reason this approach is more useful than simply subtracting the eroded image from the original binary image is that the dilation operation can help fill any 'holes' in the white areas, resulting in a cleaner boundary. Small 'holes' are filled due to the growing of the white areas that results from the dilation operation.

Principal Components Analysis (PCA) – is used for data compression, dimension reduction and to find patterns in high dimensional data [Smith 2002]. PCA operates by finding a set of principal components, where the first principal component is the direction along which there is most variance over all samples. Subsequent principal components account for as much of the remaining variance as possible. PCA is of use for computer vision as it identifies statistical patterns in the data, and thus measurement of the differences between images along the axes determined by PCA can be a powerful tool for tasks such as face recognition.

Radial Basis Function (RBF) Networks – RBF networks are feed-forward, fully connected and consist of three layers of nodes, an input layer, a hidden layer and an output layer [Haykin 1999]. An input vector arrives at the input layer and propagates to the hidden layer where, unlike MLPs that calculate the inner product of the input and the weights, each node in the hidden layer calculates a radial basis function of the distance between the weight and the input of each connection. This radial basis function is usually Gaussian and transforms the problem non-linearly into a higher dimensional space. The justification behind doing this is that a problem transformed into a high dimensional space is more likely to be linearly separable than in low dimensional space. The nodes of the output layer then calculate a linear summation of their inputs.

One other point of note with RBF networks is that although training is in the end supervised, the initial stage is unsupervised. This preliminary unsupervised stage determines the weights in the hidden layer, thereby defining the centroid of the radial basis function for each hidden node. These weights then become fixed and the weights in the output layer are learnt using supervised training.

Sparse Network of Winnows (SNoW) – is a learning architecture that is usually used for learning large scale learning tasks where the number of features involved may be very high [Carlson 1999]. The SNoW architecture is a multi-class classifier that learns a sparse network of linear functions. The update rule used during training is a variant of the

Winnows update rule. The important feature of this update rule is that the number of examples it needs to learn a linear function grows linearly with the number of relevant features, making it useful for problems where there may be a huge number of features but where only a few of them are relevant.

APPENDIX 7 – Face Probabilities

Eye/Mouth No	Ratio of Ratios No	No Matches	Face Probability
0	0-1	0-8	0.49
0	0-1	9-16	0.69
0	0-1	>16	1.76
0	2-4	0-8	7.06
0	2-4	9-16	9.58
0	2-4	>16	21.58
0	5-7	0-8	40.86
0	5-7	9-16	49.07
0	5-7	>16	71.44
0	8-10	0-8	49.60
0	8-10	9-16	57.86
0	8-10	>16	78.09
1	0-1	0-8	3.76
1	0-1	9-16	5.17
1	0-1	>16	12.41
1	2-4	0-8	37.40
1	2-4	9-16	45.46
1	2-4	>16	68.39
1	5-7	0-8	84.46
1	5-7	9-16	88.34
1	5-7	>16	95.16
1	8-10	0-8	88.56
1	8-10	9-16	91.52
1	8-10	>16	96.56
2	0-1	0-8	8.30
2	0-1	9-16	11.21
2	0-1	>16	24.69
2	2-4	0-8	58.04
2	2-4	9-16	65.86
2	2-4	>16	83.36
2	5-7	0-8	92.03
2	5-7	9-16	94.61
2	5-7	>16	97.85
2	8-10	0-8	94.71
2	8-10	9-16	96.15
2	8-10	>16	98.48
3	0-1	0-8	28.01
3	0-1	9-16	35.18
3	0-1	>16	58.49
3	2-4	0-8	85.60
3	2-4	9-16	89.24
3	2-4	>16	95.56
3	5-7	0-8	98.18
3	5-7	9-16	98.09
3	5-7	>16	99.49
3	8-10	0-8	98.71
3	8-10	9-16	99.08
3	8-10	>16	99.64

APPENDIX 8 – Facial Expression Questionnaire

The questionnaire that follows was provided to 50 computer science students and 50 non-computer science students from Queen Mary College, University of London. **Tables A8.1** & **A8.2** summarise the results of this questionnaire.

1) *Consider a chatroom application that watched your facial expressions and automatically inserted emoticons (eg ☺) into the text for you when it saw a change in facial expression. Rate on a scale 1-5 how useful (1 – very useful, 5 – of no use) you think such a application would be:*

1 2 3 4 5

2) *Place in order of importance (1 most important, 6 least important) what expressions you think the system should recognise in this chatroom application:*

Happiness .....
Sadness .....
Disgust .....
Surprise .....
Fear .....
Anger .....

3) Consider a desktop application that responds to your facial expressions by playing music relevant to you mood or by opening an appropriate web page. Rate on a scale 1-5 how useful (1 – very useful, 5 – of no use) you think such a application would be:

1 2 3 4 5

4) Does someone's facial expression indicate their underlying emotional state (1 strongly agree, 5 strongly disagree):

1 2 3 4 5

5) In an interaction with a friend, how important are their facial expressions to the interaction (1 very important, 5 no importance)?

1 2 3 4 5

6) In an interaction when you are directly facing a friend, facial expression improves the quality of the interaction (1 strongly agree, 5 strongly disagree):

1 2 3 4 5

7) In an interaction when you directly facing a friend, you are more likely to respond to their facial expressions than if you are not directly facing them (1 strongly agree, 5 strongly disagree):

1 2 3 4 5

8) Good interaction does not require direct face to face interaction (1 strongly agree, 5 strongly disagree):

1 2 3 4 5

9) Computing technology that is able to understand and respond to a users emotions is useful (1 strongly agree, 5 strongly disagree):

1 2 3 4 5

10) You are happy to use technology that is able to understand and respond to your emotions (1 strongly agree, 5 strongly disagree)):

1 2 3 4 5

Table A8.1 – Questionnaire results for 50 computer science students

Question No.	% subjects ranking 1	% subjects ranking 2	% subjects ranking 3	% subjects ranking 4	% subjects ranking 5	Mean Rating
1	18	22	30	28	2	2.74
3	14	28	24	20	14	2.92
4	22	38	24	14	2	2.36
5	38	42	8	10	2	1.96
6	32	42	12	8	6	2.14
7	46	26	20	8	0	1.90
8	4	16	50	20	10	3.16
9	18	38	26	14	4	2.48
10	10	38	32	12	8	2.70

From responses to question (2) the final ranking, from most important to least important, for automated recognition by a chatroom application was: happiness, sadness, surprise, anger, disgust, fear.

Table A8.2 – Questionnaire results for 50 non-computer science students

Question No.	% subjects ranking 1	% subjects ranking 2	% subjects ranking 3	% subjects ranking 4	% subjects ranking 5	Mean Rating
1	10	28	28	20	14	3.00
3	12	18	30	22	18	3.16
4	18	30	36	8	8	2.58
5	54	34	8	2	2	1.64
6	52	40	6	0	2	1.60
7	52	48	0	0	0	1.48
8	4	12	20	46	18	3.62
9	8	32	34	16	10	2.88
10	18	36	22	10	14	2.66

From responses to question (2) the final ranking, from most important to least important, for automated recognition by a chatroom application was: happiness, sadness, surprise, anger, disgust, fear.