CREATING EMOTIONS
AND FACIAL EXPRESSIONS
FOR EMBODIED AGENTS

The Duy Bui

Dissertation committee:

Prof. dr. ir. A. Nijholt, promoter
Dr. D.K.J. Heylen, assistant-promoter
Prof. dr. ir. W.H.M. Zijm, chairman/secretary
Prof. dr. P.M.G. Apers
Prof. dr. F.M.G. de Jong
Dr. M. Pantic (TU Delft)
Dr. Z. Ruttkay (CWI, Amsterdam)
Prof. dr. E. André (Univ. Augsburg, Germany)
Prof. dr. C. Pelachaud (Univ. de Paris, France)
Prof. dr. S. Marsella (USC Information Sciences Institute, the USA)

Taaluitgeverij Neslia Paniculata. Uitgeverij voor Lezers en Schrijvers van Talige Boeken, Nieuwe Schoolweg 28, 7514 CG Enschede.

CTIT Ph.D. Thesis Series No. 04-63. Centre for Telematics and Information Technology, University of Twente, P.O. Box 217, 7500 AE Enschede.

SIKS Dissertation Series No. 2004-12. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by: PrintPartners Ipskamp, Enschede
Cover designed by: Vu Linh Chi Nguyen

ISBN 90-75296-10-X
ISSN 1381-3617; No. 04-63 (CTIT Ph.D. Thesis Series)

CREATING EMOTIONS
AND FACIAL EXPRESSIONS
FOR EMBODIED AGENTS

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
on the authority of the rector magnificus,
prof. dr. F.A. van Vught,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday, July 1st, at 15.00

by

The Duy Bui

born on May 9th, 1978

in Ha Nam Ninh, Vietnam

This dissertation is approved by the promoter,
prof. dr. ir. A. Nijholt,
and by the assistant-promoter,
dr. D.K.J. Heylen.

to my family

Preface

Three years ago, I started as a PhD student at the Language, Knowledge and Interaction group (TKI) of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, the Netherlands. I worked under the supervision of my promoter, Anton Nijholt, and my daily supervisor, Dirk Heylen. As a Computer Science student, I had no problem with programming or any Computer Science related stuff. However, a big challenge had arrived - reading psychological literature. Psychology, a social subject, is much much different from mathematics or computer science. No more exact concepts, no more precise formulas! Being able to finish this thesis, which is both computer science and psychology related, I have owed people around me much.

Once I was interviewed by UT-News with the question: "what is the best part of living in the Netherlands?" My answer was simply: "the group where I work." Needless to say that I am indebted to the people in the TKI group. They are just more than colleagues, they are friends. Of course, the first person I would like to thank is Anton Nijholt, my promoter. He brought me to the group. He supported me all the way, all the time. It was amazing that he answered my emails from anywhere at anytime, even two o'clock in the morning. His comments on my work were always short but meaningful and useful. The second person I want to mention is Dirk Heylen, my daily supervisor. Can you imagine how hard it is to be a daily supervisor of a PhD student? I went to him everyday, asking questions, discussing ideas and even talking about life. I would like to thank him for sharing his expertise and knowledge, for his enthusiasm and his support. I also want to thank Mannes Poel, who unexpectedly became my "second daily supervisor". He was there all the time whenever I need his help. His criticisms were strict but undoubtedly helpful. I thank Hendri Hondorp for his technical support. I thank Charlotte Bijron and Alice Vissers for their secretarial support as well as their care. I thank Lynn Packwood for helping me translating all the Dutch stuff to English and for correcting my English. I thank my roommate Wojtek Jamroga for chatting, for discussing, and especially for all the help during my stay in the Netherlands. I thank all my other colleagues in the group for their interest and their productive discussion.

I am grateful to Peter Asveld, Maja Pantic and Franciska de Jong for reading through every line in my thesis carefully and giving me extremely valuable and constructive feedback.

I am very honored to have such an international committee with all the well known people: dr. Maja Pantic from TU Delft, dr. Zsófia Ruttkay from Amsterdam, prof. Elisabeth André from Germany, prof. Catherine Pelachaud from France, prof. Stacy Marsella from the USA, prof. Franciska de Jong, prof. Peter Apers and prof. Henk Zijm.

Many thanks go to my former lecturers in Hanoi National University and University of Wollongong. The knowledge they equipped me with is invaluable and is vital for me to finish this thesis. More thanks go to all my Vietnamese friends and my teammates in Drienelo 5 football team for their support. They also keep me entertained in my leisure time.

I want to express my gratitude to my parents in law who always back me up from distance. I deeply thank my grandmother, my parents and my sister for their lifetime care. They seem to be around me wherever I go. Without their encouragements, this thesis would not have been possible. Finally, special thanks are due to my wife, Linh Chi, for her love and her sincerity. She looks after every of my meals, every of my days ... and just everything of mine.

The Duy Bui

Enschede, June 2004.

Contents

1 Introduction 19
- 1.1 Embodied agents 20
- 1.2 This thesis 21
- 1.3 Structure of the Thesis 22

I Facial Expression Generation and Exportation 25

2 A muscle based 3D face model 27
- 2.1 Introduction 27
- 2.2 Previous Work 29
- 2.3 Our face model 42
- 2.4 The muscles that drive the facial animation 46
- 2.5 Experimental result 57
- 2.6 Conclusion 59

3 Exportation of 3D Facial Expressions 63
- 3.1 Introduction 63
- 3.2 Previous work 65
- 3.3 Overview of our approach 68
- 3.4 Radial Basis Functions Networks 68
- 3.5 The Error Function 70
- 3.6 Adjusting the landmarks 73
- 3.7 Experimental result 77
- 3.8 Conclusion 77

4 Combination of facial movements on a 3D talking head 79
- 4.1 Introduction 79
- 4.2 System overview 80
- 4.3 The 3D face model 82
- 4.4 Atomic facial movements 83
- 4.5 Combination of movements in one channel 86
- 4.6 Combination of movements in different channels 91

4.7 Conclusion 92

II Emotions and Facial Expressions 95

5 ParleE: A Computational Implementation of Emotions 97
5.1 Introduction 97
5.2 Psychological research on emotions 101
5.3 Existing computational implementations of emotions 109
5.4 Models of personality 114
5.5 The ParleE implementation 115
5.6 Intensity of emotions 122
5.7 The Planner 129
5.8 Learning components 133
5.9 Illustration 134
5.10 Conclusion 135

6 From Emotions to Emotional Facial Expressions 137
6.1 Introduction 137
6.2 Emotions and Facial Expressions 139
6.3 Computational work on emotional facial expressions 143
6.4 Fuzzy rule-based system 143
6.5 Overview of the system 145
6.6 The fuzzy rule-based systems 148
6.7 Experimental results 152
6.8 Conclusion and future research 154

III Example Embodied Agents 155

7 An application: Obie – A football supporter 157
7.1 Introduction 157
7.2 Embodied agent that can experience and express emotions 158
7.3 The football supporter domain 160
7.4 Illustration 162
7.5 Conclusion 166

8 Conclusion 171

A Facial Action Coding System (FACS) 175

B Detailed description of implemented facial muscles 177

C The football supporter domain 179

D Colored illustrations **183**

Bibliography **207**

Summary **207**

Samenvatting (in Dutch) **211**

Curriculum Vitae **213**

SIKS Dissertation series **214**

Index **221**

List of Figures

2.1 The CANDIDE (left) and Greta (right) face models. 30
2.2 A B-spline surface. 32
2.3 Waters' sphincter muscle model. 38
2.4 Waters' sheet muscle model. 39
2.5 Waters' vector muscle model. 39
2.6 The action of the real **Orbicularis Oris** muscle: (a) pushing out the lips and (b) pressing the lips. 42
2.7 (a) The wire-framed face with muscles; (b) the neutral face; and (c) the effect of the left Zygomatic Major. 43
2.8 The region division on right half of the face model. 44
2.9 Schematic of all implemented facial muscles. 47
2.10 The lips (left) and the control points of the lips (right). 47
2.11 (a) The face mesh without the lips; and (b) the lips are integrated to the whole face mesh. 48
2.12 The effect of a single muscle on the mesh (a and b); The effect of two muscles on the mesh (shown step by step): by simulating parallelism (c,d,e,f); by adding displacement (g,h,i,j). 50
2.13 The effect of three muscles on the mesh: by simulating parallelism (a); by adding displacement (b). 51
2.14 Sadness eyebrows with handling multiple muscle contractions: by simulating parallelism (a); by adding displacement (b). 51
2.15 The zone that contains wrinkles due to the contraction of a vector muscle. 52
2.16 The wrinkle function. 52
2.17 The "unrepresentative vertex normal" problem and its solution. . 53
2.18 The wrinkles due to the muscle contraction. 53
2.19 Illustration of the action of **Orbicularis Oris**: pushing out the lips (a); pressing the lips (b). 55
2.20 The deformation of the lips due to the contraction of **Orbicularis Oris**: neutral position (a); pushing out the lips (b); pressing the lips (c). 56
2.21 The closing of only one eye (a) and both eyes (b). 57
2.22 Eye orientation angles. 58
2.23 Eye visible area. 58

2.24 Sadness on our face model with (a) and without (b) wrinkles and handling multiple muscle contractions. 59
2.25 The face model displays surprise. 59
2.26 The face model displays happiness. 60
2.27 The face model displays sadness. 60

3.1 Overview of the face exportation system. 69
3.2 The source face model (4650 polygons). 71
3.3 Intersections between a face model's surface and a sampling plane. 71
3.4 How sampling points are located on the intersection between a sampling plane and the surface of the target and the deformed face model. 73
3.5 An initial landmark point and a variant of it. 74
3.6 Multi-point crossover. 76
3.7 How the GA process converges: broken line – solutions are replaced by their projection onto the target face model during the GA process; continuous line – projection is used only to calculate the error function and to generate the final solution. 76
3.8 The landmarks on the source face model (left) and the target face model(right). 77
3.9 The deformed face model (4650 polygons) (left) and the target face model (4142 polygons) (right). 78
3.10 Another example of the deformed face model (4650 polygons) (left) and the target face model (7736 polygons) (right). 78
3.11 Facial expressions on the deformed face model: sadness (left) and happiness (right). 78

4.1 Overview of the talking head system. 82
4.2 The addition of two conflicting muscles: the lip pressor (left) and the lip protrusion (middle) results in an unnatural movement (right). 83
4.3 The activity function of a facial movement. 86
4.4 Dominance of a parameter of two lip movements over time (top panel) and the resulting activity of the parameter (bottom panel). Circles in the bottom panel indicate target value for the parameter in the two movements. 87
4.5 Combination of the Jaw Rotation of the two movements in the same channel. 90
4.6 Our talking face display two smiles: a full smile followed by a half smile (frame by frame). 90
4.7 The activity of Zygomatic major and Orbicularis Oris before (top) and after (bottom) applying our combination algorithm. . . 93
4.8 Our talking head utters the sentence "Oh, really? I like it very much." while displaying surprise followed by happiness (frame by frame). 94

5.1 Roseman's model. 106
5.2 The OCC model. 108
5.3 Overview of the ParleE implementation. 117
5.4 Rocket domain for multi-agent planning defined in STRIPS. . . . 129
5.5 Planning graph for the probabilistic rocket problem. 130
5.6 Top-down Dynamic Programming for calculating the probability of achieving a goal. 131
5.7 Possible paths to search for an optimal plan. 132
5.8 The neutral version of the agent (left) expresses facial expressions compared to the optimistic one (middle) and sensitive one (right) after the user eats his bread. 135

6.1 Ekman's view on how emotions and expressions are related (depicted by Fridlund, 1994). 140
6.2 The emotion-to-expression system. 146
6.3 Membership functions for emotion intensity. 148
6.4 Membership functions for muscle contraction level. 149
6.5 Basic emotions: neutral, **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust**, **Surprise** (from left to right). 153
6.6 Increasing **Surprise**. 153
6.7 Blends of **Anger** and **Disgust** (left), **Happiness** and **Surprise** (right). 153

7.1 Overview of an agent that can experience and express emotions. 159
7.2 The view of a soccer field. 161
7.3 A snapshot of the football supporter application. 163
7.4 The graph of Obie's fear emotion and a snapshot of his expression of fear. 164
7.5 The graph of Obie's happiness emotion and a snapshot of his expression of happiness. 165
7.6 Obie's happiness decays to return to the neutral state. 165
7.7 The graph of Obie's surprise emotion and a snapshot of his expression of surprise. 166
7.8 The graph of Obie's sadness emotion at the last goal and a snapshot of his expression of sadness. 167
7.9 A snapshot of Obie's expression of happiness and surprise. 167
7.10 The graph of Obie's happiness emotion with four different personalities: neutral, pessimistic, realistic and sensitive (from top to bottom). 168
7.11 The graph of Obie's sadness emotion with four different personalities: neutral, pessimistic, realistic and sensitive (from top to bottom). 169

D.1 (a) The wire-framed face with muscles; (b) the neutral face; and (c) the effect of the left Zygomatic Major. 184
D.2 The wrinkles due to the muscle contraction. 184

D.3 The face model displays sadness. 185
D.4 Sadness on our face model with (a) and without (b) wrinkles and handling multiple muscle contractions. 185
D.5 The face model displays surprise. 188
D.6 The face model displays happiness. 188
D.7 Basic emotions: neutral, **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust**, **Surprise** (from left to right). 189
D.8 Increasing **Surprise**. 189
D.9 Blends of **Anger** and **Disgust** (left), **Happiness** and **Surprise** (right). 189
D.10 Our talking head utters the sentence “Oh, really? I like it very much.” while displaying surprise followed by happiness (frame by frame). 192

List of Tables

2.1 Implemented muscles in our face model (right side of the face). . 46
2.2 The result of the improvement on the animation (on a Pentium III 800Mhz, 256MB RAM, NVidia GeForce3 video card). 55

4.1 Conflicting muscles in the system (muscles 10, 13, 14, 15, 18, 19 are not included because they do not conflict with any other muscle). 84

5.1 Dimensions of a personality (from (Rousseau, 1996)). 116
5.2 Rules to trigger emotions from OCC model. 120
5.3 ParleE rules to trigger emotions based on the OCC model (I_1: impact of the event to the agent's goal; old P_{goal}: the probability of achieving the goal before the event occurs; new P_{goal}: the probability of achieving the goal after the event occurs; Goal status: unconfirmed, success or fail; I_2: impact of the event to the other agent's goal; LL: the liking level of the agent toward the other agent; Pwn: the praiseworthiness of the event; actor: who causes the event to happen). 121

6.1 Fuzzy rules for emotion **Sadness**, vh:**VeryHigh** h:**High** m:**Medium** l:**Low** vl:**VeryLow** vs:**VerySmall** s:**Small** m:**Medium** b:**Big** vb:**VeryBig** −:no contraction. 149
6.2 Fuzzy rules for emotion **Happiness**. 150
6.3 Fuzzy rules for emotion **Anger**. 150
6.4 Fuzzy rules for emotion **Fear**. 150
6.5 Fuzzy rules for emotion **Disgust**. 151
6.6 Fuzzy rules for emotion **Surprise**. 151
6.7 List of muscles that take part in blend expressions of emotion . . 152

A.1 The Facial Action Coding System (part 1). 175
A.2 The Facial Action Coding System (part 2). 176

Chapter 1

Introduction

> *"First we thought the PC was a calculator. Then we found out how to turn numbers into letters with ASCII – and we thought it was a typewriter. Then we discovered graphics, and we thought it was a television. With the World Wide Web, we've realized it's a brochure."*
>
> – *Douglas Adams*

The combination of research in the field of computer graphics, autonomous agents, and speech and language technology has led to the development of embodied agents. The emerging technology of embodied agents can realize different promising applications including human-like interfaces to improve the interaction between human and computer; simulated virtual characters for different applications such as entertainment, education, and the like; and believable animated characters to increase the interestingness of computer games.

At present, most of the interaction between humans and computers is done via text, mouse and keyboard. Together with the rapid development of computer graphics and speech technology, researchers are increasingly paying attention to making this interaction more adaptive, flexible and human-oriented. One way to do so is through the creation of embodied agents. Embodied agents have been used as interfaces for human computer interaction (Cassell et al., 2000). These interfaces, often referred as 'embodied conversational agents" (ECA), are believed to offer a more natural interaction between user and computer.

Along with the growth of virtual worlds, the technology of embodied agents allows the creation of simulated characters for different purposes such as entertainment and education. The capabilities of speech, facial expression and gesture of embodied agents makes them well suited for entertainment applications. The virtual story teller at the university of Twente is an example of such an embodied agent (Theune et al., 2003). The agent is an embodied, speaking agent to present the generated stories using appropriate prosody and gestures. The agent is located within a replica of a local theater: the Virtual Music Center (VMC). Besides this traditional form of storytelling, extensions to virtual drama are planned for embodied characters to act out the story on the stage of the VMC. There are many other examples of entertaining embodied agents including synthetic characters in the MIT media lab (Blumberg et al., 2002), robot soccer commentators (André et al., 1997).

Embodied agents can serve as mentors that can demonstrate procedures, answer questions, and monitor students' performance (Rickel and Johnson, 1998). Embodied agents can also serve as teammates in some tasks that require several people. An example of these agents is Steve (Soar Training Expert for Virtual Environments) at the University of Southern California (Rickel and Johnson, 1998). Steve was built to support team training in a three-dimensional, interactive, simulated mock-up of a student's work environment when training on real equipment is impractical.

Embodied agents can also be used to improve computer games. Traditional computer games usually contain passive, machinelike characters. Compared to a single human game where a player plays with and against computer characters, games between human and human are still preferable (Sweetser et al., 2003). Playing with and against different human players who play differently, and who react differently is interesting because there is always something new even when the same game is played. Games with only computer characters still fail to achieve this. Embodied agents (or synthetic animated characters) with their own goals, knowledge, and capabilities can be a good answer to this issue (Laird and van Lent, 2000), especially for the types of games which use the computer to create virtual worlds and characters for people to dynamically interact with such as Quake, Everquest or Black and White. Thus, synthetic animated characters are being developed and embedded in many kinds of computer games. They also can potentially lead to completely new kinds of games (Laird and van Lent, 2000).

1.1 Embodied agents

So what are embodied agents? Let us first have a look at the concept of "agent". As defined by Wooldridge (1999):

> "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives."

This includes control systems, software daemons, web spiders, virtual assistants like that annoying Word paperclip, or IRC bots.

Embodied agents are a special kind of agent in the sense that they are represented by animated human or animal bodies, or sometimes just an animated talking head. The common aim is to make embodied agents believable in such a way that their actions and their reactions to the human users can be as lifelike as those of the living creatures they are representing.

The major motivation behind the embodied agents is believability where the simulation of the agent's body is biologically and physically inspired. To quote

Nijholt (2001):

> "Embodiment makes it possible to show facial expressions, body language, lip movements and interaction supporting gestures. It allows a multi-modality to show information and to express motivating nonverbal behaviors, making interaction more natural and robust. In addition, the embodiment provides a point of the focus for interaction. Embodiment allows also, more than just language, the expression of emotional behavior in which personality shows."

An embodied agent will typically contain the following components:

- a talking head which is able to speak, to display lip movements during speech, emotional facial expressions, and conversational signals.
- a body which is able to display gestures.
- a model of mind which may contain models of belief, emotion, behavior planner, motivational states and personality, etc.

Because embodied agents' practical applications are numerous, they are receiving attention from many research communities including computer graphics and artificial intelligence (AI).

1.2 This thesis

In this thesis we consider embodied agents which are represented by an animated talking head. For such embodied agents to be believable, the minds of agents should not be restricted to model reasoning, intelligence and knowledge but also emotions and personality. Furthermore, it is necessary to pay attention not only to the agent's capacities for natural language interaction but also to its non-verbal aspects of expression.

As a matter of fact, both AI and graphics research communities are interested in animated talking heads (Prevost and Pelachaud, 1995). While AI researchers and cognitive scientists have primarily tried to understand and model the motivation behind facial movements, computer graphics researchers have focused on the physical and anatomical aspects of those movements. The combination of research on facial animation in these two communities can result in embodied agents that can interact autonomously, with humans or with each other. The research described in this thesis can be positioned in both computer graphics and AI communities.

There are several issues we deal with in this thesis. First of all, from the computer graphics direction, we deal with the problem of creating a face model and a facial muscle model in such a way that realistic facial expressions can be produced in real-time on a standard computer. In order to do so, human effort is needed for preprocessing the face model and tuning up the parameters of the muscle model. Therefore, we also deal with the issue of reducing human effort when creating facial expressions on a newly created face model based on the data from an existing model. Secondly, from the AI direction, we deal with the problem of combining different facial movements temporally. Thirdly, we deal with the motivation of one kind of facial expression - emotional facial expressions. We propose an implementation of emotions and a mapping from several emotions to facial expressions.

Although the subsystems that we describe in this thesis are built as components of an embodied agent system, they can be used separately in other systems. Each subsystem can also be replaced by a similar subsystem from another system that fulfills the same task.

1.3 Structure of the Thesis

Chapter 2 is based on (Bui et al., 2003a). In this chapter, we present a simple muscle-based 3D face model that can realize the following objective: it is able to produce both realistic facial expressions and real-time animation for standard personal computers. In order to do that, we propose: (i) a face model that allows high quality and realistic facial expressions, which is sufficiently simple in order to keep the animation real-time and is able to assist the muscle model to control the deformations; (ii) a muscle model that produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time.

Chapter 3 is based on (Bui et al., 2003b,c). In this chapter, we present a method of transferring facial animation from a given face model to a newly created one without much human involvement. We use Radial Basis Function (RBF) networks (Broomhead and Lowe, 1988) to deform a source face model to represent a target face model using the specification of corresponding landmarks on the two face models. The landmarks on the source face model are manually specified once and are reused for every target face model. We introduce a novel method to specify and adjust landmarks on the target face model automatically. The adjustment process is done by Genetic Algorithms (GAs). The fitness function used in the GAs expresses the difference between the surface of the deformed face model and the target face model. We also present an algorithm to calculate this function fast. After all the landmarks have been placed in optimal positions, the RBF networks are used to deform the source face model as well as to transfer the muscles on the source face model to the deformed face model.

Chapter 4 is based on (Bui et al., 2004b). In this chapter, we propose a scheme of combining facial movements on a 3D talking head. There are several types of movements, such as conversational signals, emotion display, etc. We call these channels of facial movement. We concentrate on the dynamic aspects of facial movements and the combination of facial expressions in different channels that are responsible for different tasks. First, we concatenate the movements in the same channel to generate smooth transitions between adjacent movements. This combination only applies to individual muscles. The movements from all channels are then combined taking into account the resolution of possible conflicting muscles.

Chapter 5 is based on (Bui et al., 2002; Bui and Jamroga, 2003; Bui and Heylen, 2003). In this chapter, we describe ParleE, a quantitative, flexible and adaptive computational implementation of emotions for an embodied agent situated in a multi-agent environment. ParleE has been inspired by various other implementations of emotions such as Elliott's Affective Reasoner (Elliott, 1992), Velásquez's Cathexis (Velásquez, 1997), El-Nasr and colleagues' FLAME (El-Nasr et al., 2000) and, in particular, by Gratch's Émile (Gratch, 2000). Like some of these and many other implementations, ParleE generates emotions based on Ortony, Clore and Collins' theory of appraisal (Ortony et al., 1988). Nevertheless, ParleE possesses some significant properties of its own. The main novel differences with other systems are (i) the way it uses forward-chaining search within a finite depth to obtain the probability of achieving a goal; (ii) the way it uses models of other agents' plans and goals to predict their behavior and set up expectations about the likelihood of events; and (iii) the way it incorporates personality, individual characteristics and motivational states in the implementation.

Chapter 6 is based on (Bui et al., 2001). In this chapter, we discuss the problem of generating emotional facial expressions from emotions. It has been believed for a long time that there exists a relationship between facial activity and emotional state. The pioneer publication of Darwin (1872/1965), "The expression of the emotions in man and animals", was one of the first extensive studies on this topic. Much recent research has strengthened the idea that emotions are expressed on faces. A typical example is the descriptive work by Ekman and Friesen (1975), which discusses how several emotions as well as their blends are displayed on the face. We want to base our implementation on this work to map emotion representations onto the contraction level of facial muscles. We focus on two aspects of generating emotional facial expressions. First, we want to take into account the continuous changes in expressions of an emotion depending on the intensity by which it is felt. Secondly, we want to find a way to specify combinations of expressions due to more than one emotion, i.e., blends, in accordance with the literature mentioned. We have found that a fuzzy rule-based system is suitable for these requirements because it allows us to incorporate qualitative as well as quantitative information. Fuzzy rules can capture descriptions which are described in natural language as well as vague concepts like "slight sadness", "more intense sadness", etcetera. Moreover, the fuzzy rule-based approach can assure the smooth mapping between emotions and facial expressions. Following Ekman and Friesen (1975), we consider the following six emotions: **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust** and **Surprise**. These are said to be universal in the sense that they are associated consistently with the same facial expressions across different cultures.

Chapter 7 is based on (Bui et al., 2004a). In this chapter, we discuss how the different systems we mentioned earlier can be combined to create an emotional embodied agent. As an example, we introduce Obie, a football (soccer) supporter agent. Obie is watching a football match in which a team, which he supports, is playing. Obie can experience different emotions by appraising events based on his goals, standards, and preferences. Obie can also show his emotions on a 3D talking head. Typical events that occur in a football match are: kick-off, penalty, goal, free kick, etcetera. These events can be obtained by various ways. For a real football match, the events may be extracted directly by translation from visual to verbal representations or translation from a news stream produced by a mediator (e.g., a human commentator) to a textual representation (Jong and Westerveld, 2001; Nijholt et al., 2003). For a robot cup match, the events can be extracted from a team's vision system (Kooij, 2003). The events in a simulation match can be extracted directly from the data of the match.

Part I

Facial Expression Generation and Exportation

Chapter 2

A muscle based 3D face model

> *"A man's face is his autobiography. A woman's face is her work of fiction."*
>
> – *Oscar Wilde*

2.1 Introduction

The human face is very special. It is the crucial part of the body by which one recognizes a person visually (Hager and Ekman, 1996). Among hundreds of familiar faces, we can still recognize a specific face. We also have the ability to detect even a subtle facial movement. This ability develops from our earliest days of childhood, and it equips us with this most basic communicative tool. As concluded by Sigman and Capps (1997) in their "Children with Autism: A Developmental Perspective", infants at about nine months of age start to monitor the facial responses of other people. Normally developing infants at one year of age can start discriminating expression of emotions. Infants continue to refine this ability while they are growing up.

Human facial movements play an important role in face-to-face communication (Argyle, 1990). Lip movements during speech provide a visual hint about what is being said. Cohen and Massaro (1993) have shown that the presence of video containing lip movements in addition to audio significantly improves the phoneme recognition rate compared to the use of audio only. These lip movements are essential for deaf people to understand a conversation. Facial expressions, in the context of nonverbal communication, usually imply the dynamic changes of the face over time. However, as the static picture of the face can also express emotions, the facial expressions can be static. Without context, though, they might be ambiguous. Facial expressions take place continuously during speech. They provide additional commentary to and illustration of the verbal information. Facial expressions are able to express emotions and moods. They can also convey information about the personality or the characteristics of a person. This information can show the hidden inner side of the person, which may not be accessible from the verbal channel.

The problem of modeling the human face and generating facial movements by computers is a significant challenge in the computer graphics research community. There is an emerging demand for high quality realistic facial expressions and real-time animation. Existing approaches generally fail to achieve both objectives. Key-frame animation, parameterized animation and pseudo-muscle-based animation are a simple way to produce animation in real-time but fail to produce realistic facial expressions as they cannot provide a generic way of producing bulges and wrinkles in the skin and handling multiple parameter/muscle interaction. Multi-layer muscle-based animation with several layers of facial structure, on the other hand, produce realistic expressions; however, they require massive computation and cannot obtain real-time animation on standard personal computers.

In this chapter, we present a simple muscle-based 3D face model that can realize the following objective: it is able to produce both realistic facial expressions and real-time animation for standard personal computers. In order to do that, we propose:

- A face model that allows high quality and realistic facial expressions, which is sufficiently simple in order to keep the animation real-time and is able to assist the muscle model to control the deformations.
- A muscle model that produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time.

Section 2.2 thoroughly reviews the existing techniques that have been used to model and animate the face. First, two main techniques to model the face are studied. They are: facial modeling with skins of polygons and facial modeling with parametric surfaces. Their advantages and disadvantages over the rendering and animation processes are discussed. Then we go into the techniques to animate the face. Four main techniques are considered: key-frame animation, parameterized animation, pseudo-muscle-based animation and muscle-based animation. We pay attention to a very successful approach by Waters (1987), on which our work described in this thesis is based. Section 2.3 presents the structure of our face model. We describe how we create the face mesh in order to generate realistic facial expressions while keeping the animation in real-time. The face mesh is also designed to improve the performance of the muscle model. The muscles that control the facial animation are discussed in Section 2.4. In this section, we describe how we extend Waters' muscle model (1987) to handle the combination of multiple muscle actions and to generate bulges and wrinkles. Techniques to improve the performance of the muscle model are also discussed. Next, we present the implementation of the realistic **Orbicularis Oris** (muscle of the mouth), **Orbicularis Oculi** (muscle of the eyes) and the jaw rotation.

Finally, some experimental results are shown in Section 2.5, which contains faces that display surprise, happiness and sadness to illustrate how our work is used to generate realistic facial expressions.

2.2 Previous Work

The aim of 3D facial animation is to manipulate the 3D surface mesh of the face model over time so that at any moment of time the face model has the desired expression. Approaches to 3D facial animation range from simple ones to achieve real-time animation like the CANDIDE face model (Rydfalk, 1987) to more sophisticated ones to achieve photo-quality expressions like Lee et al.'s multi-layer muscle-based face model (1995). CANDIDE, described by Rydfalk (1987), is a parameterized face model that contains only 75 vertices and 100 triangles. It is specifically developed for model-based coding of human faces, which encodes the faces at reduced bit rates in application such as video telephony. Its low number of polygons allows fast reconstruction and animation with moderate computing power. CANDIDE is controlled by global and local Action Units (AUs) (Ekman and Friesen, 1978). The global ones correspond to rotations around three axes. The local Action Units control the mimicry of the face so that different expressions can be obtained. Lee et al.'s face model (1995) is an example of approaches to achieve photo-quality expressions. It consists of several layers: a biological tissue layer, a muscle layer, and an impenetrable skull structure. To model an individual's face, the generic face model is automatically adapted to the data acquired from a laser scanner. The image from the scanner is also processed to produce the texture for the face model. With a physics-based muscle system, realistic facial expressions are generated on the textured face model.

There are two problems to deal with in 3D facial animations. The first problem is the facial modeling, which deals with the representation of the face model itself. The second problem is the facial motion modeling. We will now review techniques to deal with these two problems in detail. Moreover, we will also discuss the methods to model and animate the lips. Among parts of the face, the lips play a very important role in face-to-face communication. They are an important component of expressing emotions. They also participate in the articulation of speech. Because of their role, the lips have received special treatment in modeling facial animation.

2.2.1 Facial modeling

The human face is a complex and flexible three dimensional surface. It usually contains creases and permanent wrinkles. Temporary bulges and wrinkles are generated on the face during facial expressions. The challenge is how to represent such a surface that allows both real-time animation and realistic expressions.

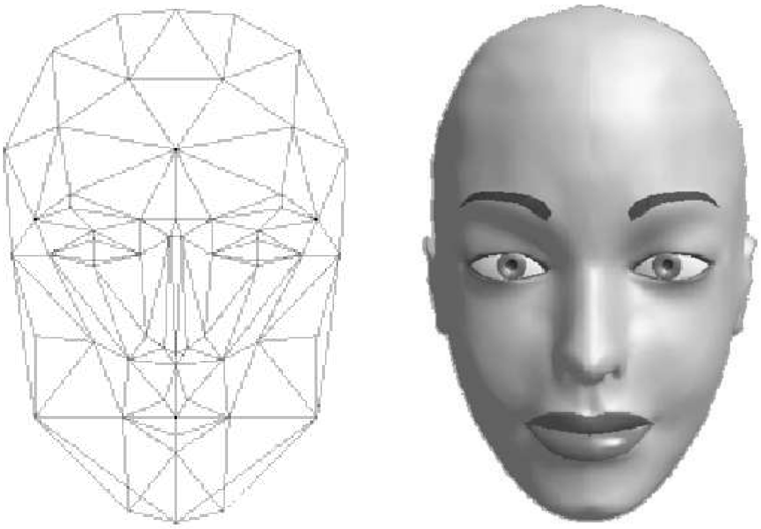


Figure 2.1: The CANDIDE (left) and Greta (right) face models.

Facial modeling with skins of polygons

One possibility is to use a skin of polygons to approximate the surface of the face directly. The face model is then animated by direct manipulation on surface polygon vertices. This approach was first introduced by Gouraud (1971). He sampled the surface of the face at a number of points and connected these points to form a skin of polygons. Using this approach, Parke (1972) created a face model that is one of the premier works in the field of human facial animation. The model contains about 250 polygons defined by about 400 vertices. Since the human face is approximately symmetric, only one side of the face model was manually created. The other side was obtained by mirroring about the plane of symmetry. Parke concentrated on the speed and quality of the rendering process when creating the face model. He minimized the number of polygons consistent with good results to allow faster rendering. To get good smooth shading, he used more polygons in the areas of high curvature (the nose, mouth, around the eyes and the edge of the chin), and less in the area of low curvature (the forehead, cheeks and neck). The polygons in the regions under the eyes, the side of the nose, the edge of the lips and the corner of the mouth, where creases occur on a face, are designed so that their edges coincide with the creases. The polygons in the regions where color boundaries occur on the face, e.g., the lips and eyebrows, are designed so that their edges coincide with these boundaries.

Many other facial animation systems use polygon mesh for representing the skin of the face. Examples include the CANDIDE face model (Rydfalk,

1987) and the Greta face model (Pasquariello and Pelachaud, 2001). The original CANDIDE is a parameterized face mask created by Rydfalk (1987) at the Linköping Image Coding Group, specifically developed for model-based coding of human faces. It contains 75 vertices and 100 triangles, which allow fast reconstruction with moderate computing power. Other improved versions of this face model have been developed in the sequence. Vertices were added to cover the entire frontal head (including hair and teeth) and the shoulders (Candide-2, Welsh, 1991), and to simplify animation by MPEG-4 Facial Animation Parameters (Doenges et al., 1997) (Candide-3 Ahlberg, 2001). Greta (Pasquariello and Pelachaud, 2001) is a face model conforming with MPEG-4 specifications (Doenges et al., 1997). Greta contains around 15000 polygons. Efforts have been made on Greta to give a great level of detail in the most expressive areas of the facial model that are dedicated to the communication of the information and to express the emotions. These areas are the mouth, the eyes, the forehead and the nasolabial furrow. A large number of polygons has been designed and placed in these regions. Moreover, the forehead and the nasolabial furrow were treated specially. The polygons in the forehead region were organized into a regular horizontal grid to generate horizontal wrinkles during the raising of the eyebrows by bump mapping techniques (Moubaraki et al., 1995). The polygons in the nasolabial furrow were arranged so that a separation between the stretched skin near the mouth and the skin in the cheeks is well defined to generate the clear furrow during smile.

Underneath the skin of polygons to model the surface of the face, additional layers such as the dermal fatty layer, the muscle layer and the skull surface are added for simulating physics-based muscle model (Kähler et al., 2001; Lee et al., 1995; Terzopoulos and Waters, 1990).

To model a specific individual, three techniques have been employed. The first one manually builds the face model for an individual with 3D modeling tools such as AutoCAD and 3DMax. This is a very laborious task. The second technique, called photogrammetric measurement, builds the face model from several photographs of the individual's head from different views. Manual specification of the feature points on the photographs is required to find the relation between the photographs and the 3D model. The third technique uses laser scanners to scan real faces and uses the image as the texture mask for the face model. Laser scans provide extremely detailed 3D data as well as the color or grey level of the scanned points. These data can be combined to create amazingly realistic static models.

Approximation of the human face with polygons has several advantages. For polygonal surfaces, the problems of determining whether all or part of an object is within a viewing space (clipping), detecting hidden surfaces, and determining the shading of the visible surfaces have been solved by a number of algorithms (Mahl, 1972; Weiss, 1966). These algorithms are fast and inexpensive when compared to algorithms for surfaces of higher degree; they are also implemented in hardware. Moreover, the development of Gouraud shading (Gouraud, 1971) and Phong shading (Bui, 1975) algorithms gives a continuously curved appearance to a surface made up of polygons. However, if only a small num-

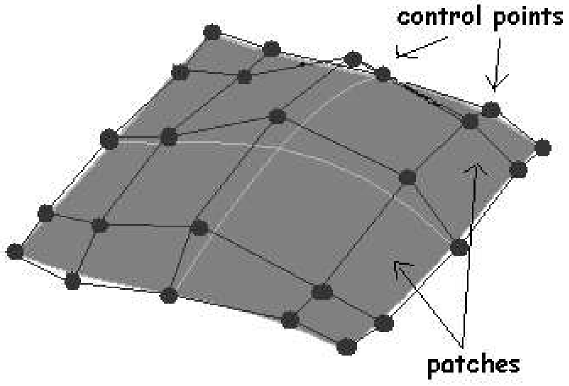


Figure 2.2: A B-spline surface.

ber of polygons are used as in Parke's model (1972) and the CANDIDE face model (Rydfalk, 1987), these algorithms cannot completely hide the flatness of the facial surface.

2.2.2 Facial modeling with parametric surfaces

The human face can be modeled with a parametric surface. The aim is to find a mathematical function that represents the facial surface (Coons, 1967). However, it is very hard to find such a function that allows detailed representation of the face. A more preferable approach is to create a surface by concatenating a collection of parametric patches. These patches have to connect to the neighboring patches at the borders. This is usually called the C^0 continuity property. The smoothness of the whole surface is assessed through the continuity property of higher degrees. The surface possesses C^1 continuity when the patches represented by the first derivative of the original patches are continuous at the borders, C^2 continuity for the second derivative continuity, and so on. The surface with the continuity of higher degree is smoother.

The surface created by these patches is usually determined by a set of points called "control points". The face model is animated by manipulating the control points. These control points are often arranged in a grid of size $(n+1) \times (m+1)$: $W_{i,j}$, $0 \leq i \leq n$, $0 \leq j \leq m$. Points on the surface, $w(u, v)$, are defined as a weighted sum of these control points $W_{i,j}$:

$$w(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_i(u) N_j(v) W_{i,j}$$

where $N_i(u)$ and $Nj(v)$ are basis functions. A basis function represents a curve in two dimensional space. So intuitively, the surface is created by combining many pieces of curve. These curves are selected carefully so that the generated surface is smooth. Splines are usually chosen for their simplicity and property of high order continuity. An example of a surface created by parametric patches is shown in Figure 2.2.

Since the facial skin is a smooth, flexible surface, many systems that use parametric patches chose spline patches to model the face because of splines' continuity and smoothness. One example is Billy created at Pixar (Reeves, 1990). Billy, the baby in the movie "Tin Toy" (Reeves, 1990) was first constructed with triangular Bézier patches but it suffered from wrinkling problems. Bicubic Catmull-Rom spline patches were then used instead of triangular Bézier ones. Wrinkling problems were reduced but not completely eliminated. Billy also still had a problem at the top of his head as well as major problems with the neck. Another example is the Facial Action Control Editor created by Waite (1989). Waite used B-spline patches with a total of 16 × 12 control points. The surface created by B-spline patches is smooth because of the C^2 continuity property of the B-spline patches. The surface created by these patches is often called B-spline surface. Openings for the eyes, nostrils and mouth are created with a technique called geometric trimming. This technique removes unwanted portions of the surface and provides a new mathematical description for the trimmed patch. As may be expected from the number of control points, this is not a very detailed model; there are no eyelids nor are there any eyes.

To model a specific human face with this technique, the face model is adapted to data obtained by the laser scanner. An example of this approach is Hoch et al.'s work (1994). Here, the adaptation process is performed by minimizing the mean square error between the given data points and the surface points with the constraint that the control points are correctly positioned in regions where the respective action units apply. This is to ensure that the control points associated with an action unit result in the deformation of the correct region of skin tissue. Then, the color information obtained from the laser scanner can be used as the texture map for the facial mask.

Modeling the face with parametric patches has the advantage of having to handle less data and producing a more smooth surface compared to polygonal models. However, this approach suffers from the problem of rendering the surfaces. Hidden surface and rendering algorithms exist for quadric surfaces but they tend to be quite inefficient (Mahl, 1972; Weiss, 1966). Similar algorithms can be proposed for surfaces of higher degree, but they would be increasingly expensive to use as the degree of the surface increased. For this reason, surfaces of high degree are then usually approximated by a skin of polygons. This process is called polygonalization; it still requires relatively heavy computations if the sampling resolution is high. The problem is worse when the number of spline control points is large to model correctly every small region on the face. Moreover, because of the smoothness, this approach is not suitable for generating subtle wrinkles on the face. Due to its advantages and disadvantages, this approach is more suitable for modeling a small and smooth region of the face such as the lips.

2.2.3 Facial motion modeling

As discussed in the previous section, the facial surface is animated directly either by manipulating the polygon vertices or indirectly by manipulating the control points. There are four main fundamental techniques to manipulate the facial surface. They are key-frame animation, parameterized animation, pseudo-muscle-based animation, and muscle-based animation.

Key-frame animation

Key-frame animation techniques are widely used as they offer an intuitive approach to facial animation. In these techniques, interpolation functions are used to generate the frames in between key frames. They provide smooth transition between key frames. Key frames in this case are the faces of different expressions.

Key-frame animation techniques can be geometric or parameter based. Geometric based techniques directly modify the positions of the face mesh vertices. These face meshes are required to have the same topology. The position of each vertex in the intermediate face is determined by interpolating between the positions of corresponding vertices in the two key faces. Parameter-based techniques are used with other facial animation techniques such as parameterized animation and muscle-based animation. The parameters can be the parameters in parameterized animation or in the muscle contraction levels. Parameter-based key-frame techniques determine each parameter used to create the intermediate face by interpolating the corresponding parameters used to create the two key faces.

The whole interpolation process is governed by a factor that decides how close the intermediate face is to the first key face compared to the second key face. For example, the face of a half smiled expression is the result of interpolating between the neutral face and fully smiled face with the factor of 0.5. In the case of generating a sequence of animation, the temporal information of the intermediate frame and two key frames is used to calculate this factor.

Key-frame animation techniques can use linear or non-linear interpolations. Linear interpolation is commonly used for simplicity (Pighin et al., 1998). Non-linear interpolations are used to mimic the facial motions more correctly (Parke, 1972; Waters and Levergood, 1993). Because the face is governed by physical laws, its motion is not linear but tends to accelerate and decelerate. Parke (1972) then used a cosine interpolation scheme to approximate these acceleration and deceleration processes. Bilinear interpolation is used when four key frames are involved to generate a greater variety of facial expressions than linear interpolation allows (Parke, 1974).

Key-frame animation techniques are fast and easy to generate primitive facial animations. However, they cannot create an expression that falls outside the bounds of the key frame set. As the range of expressions depends on the number and disparity of key frames, key-frame animation techniques can only generate a very wide range of expressions when there are enough varied key frames. In order to achieve that, an explicit geometric data collection or data generation effort for each key frame and its storage are required. This is a very labor-intensive task. Due to this disadvantage, key-frame animation techniques are only suitable for producing a small set of animations from a few key frames or for combining with other methods to produce intermediate frames.

Parameterized animation

Parameterized animation techniques were first developed by Parke (1974) to overcome some of the limitations and restrictions of key-frame animation techniques. They animate the face with a fairly small set of control parameters. These parameters are hard-wired into particular facial regions to produce local deformations. Deformation determined by a parameter produces a specific facial movement. For example, Hoch et al. (1994) used these techniques for facial movements that correspond to the Action Units in the Facial Action Coding System (Ekman and Friesen, 1978).

The face model used with parameterized animation techniques needs to be specially designed to implement the set of control parameters. This face model is called the parameterized model. One way to create such a model is to observe the human face carefully in order to divide the face into regions. The polygons in these regions then need to be designed in such a way that local deformations can be applied to produce desired facial movements. Contribution and guidance from experienced animators would definitely play an important role in creating a good parameterized face model.

Local deformations can be produced with local region interpolations, geometric transformations and texture mapping techniques. Local region interpolations produce the intermediate shape of the two pre-specified shapes of the local region based on the value of the parameter. The position of each polygon vertex in the local region is determined by interpolating the corresponding vertices of the two pre-specified shapes. Geometric transformations, such as rotation, scaling and position offset, use mathematical functions of the parameter to transform a shape of the local region into another shape. Local deformations can also be produced with surface texture mapping techniques. Surface texture mapping techniques (Oka et al., 1987) manipulate the texture of the facial region to synthesize the facial movements. Let us have a look at an example of the implementation of the eyelid movements to have a better understanding of these techniques. The eyelid movements can be implemented by locally interpolating between the two pre-specified eyelid sub-meshes that correspond to the closed eyelid and the open eyelid. An interpolation factor of 0.5 generates a half closed eyelid. The eyelid movements can also be implemented by the local rotation of the eyelid sub-mesh around a predefined axis. For the eyebrow movements, horizontal wrinkles are visible in the forehead when the eyebrows are raised. Texture mapping techniques can be used to generate these wrinkles. The visibility and position of these wrinkles depend on the value of the eyebrow raising parameter.

Unlike key-frame animation techniques, parameterized animation techniques allow explicit control of specific regions of the face. They provide a large range of facial expressions with relatively low computational costs. However, there are several limitations associated with them. First, the choice of the parameter set depends on the facial mesh topology and, therefore, parameterizations lose the generality when applied to a new facial topology. Second, to date, it is still impossible to create a complete set of control parameters to generate every possible facial movement. Third, it is difficult for parameterized animation techniques to handle the conflicting parameters that affect the same vertices. Therefore, parameterized animation techniques rarely produce natural human expressions when a conflict between parameters occurs. For this reason, parameterized animation techniques are suitable for deforming specific facial regions only; however, this often introduces noticeable motion boundaries. Finally, tedious manual tuning is required to set parameter values, and even after that, unrealistic motions may result.

Pseudo-muscle-based animation

Pseudo-muscle-based techniques deform the facial mesh by simulating real muscle contractions, but ignoring the underlying facial anatomy. These techniques include abstract muscle action (AMA) (Magnenat-Thalmann et al., 1988) and free form deformations (Kalra et al., 1992; Pasquariello and Pelachaud, 2001).

Magnenat-Thalmann et al. (1988) used FACS action units (Ekman and Friesen, 1978) (see Appendix A) as the guide for constructing the AMA procedures. Each AMA procedure works on a specific region of the face, which must be defined when the face is constructed, to approximate the action of a single muscle or a group of closely related muscles. Magnenat-Thalmann et al.'s system contains 30 AMA procedures. An example of AMA procedures is the vertical jaw procedure (mouth opening). The activation of this procedure results in a series of successive small actions controlled by parameters of the procedure. These actions lower the jaw and round up the lip shape. Facial expressions are formed by controlling the AMA procedures in groups. Due to the dependency among the AMA procedures, the ordering of these procedures is important.

Free form deformation (FFD) (Sederberg and Parry, 1986) deforms a volumetric object surrounded by a parallelepiped lattice structure by transforming the lattice vertices (or control points). The points on the object embedded in the original lattice structure are mapped to the deformed lattice over a trivariant Bézier-interpolation. This mapping allows the deformations of the lattice to be automatically passed on to the object. FFDs can produce bumps, dents, bends, twists, or other complex modifications of the model's shape. FFD has been used to simulate the Facial Animation Parameter (FAP) of the MPEG-4 standards (Pasquariello and Pelachaud, 2001). Rational free form deformation (RFFD) incorporates weight factors for each control point, adding another degree of freedom in specifying deformations. Kalra et al. (1992) combine Rational Free Form Deformation (RFFD) with a region based approach to simulate the visual effects of the muscles. First, surface regions are defined corresponding to the anatomical description of the muscle actions. Regions of interest are then defined as the parallelepiped volumes based on the surface regions. By interactively displacing the control points and by changing the weights associated with each control point, the skin deformations are simulated to produce stretching, squashing, expanding, and compressing inside the volume. The deformation of the boundary points lying within the adjoining regions are determined by linear interpolation.

Pseudo-muscle-based techniques do not provide a precise simulation of the facial muscle and the skin behavior. By ignoring the underlying anatomy of the face, they usually fail to model furrows, bulges, and wrinkles in the skin. Finally, not much attention has been paid to correctly handling the interaction of multiple muscles.

Muscle-based animation

Muscle-based techniques use anatomy-based mass-spring or multi-layer models of the skin and muscles to deform the facial mesh. Mass-spring methods propagate muscle forces in an elastic spring mesh that models skin deformation. A layered spring mesh extends a mass-spring structure into three connected mesh layers to model anatomical facial behavior more faithfully.

Platt and Badler (1981) pioneered on muscle modeling by proposing a mass-spring model. The model contains three components: the skin, the bones and muscles. The skin is represented as a set of movable 3D points while the bones is represented as a set of fixed points. The muscles, lying between the skin and the bones, are groups of points with elastic arcs. Facial expression are generated by applying forces to (visco-)elastic meshes through muscle arcs. Platt (1985) later presented a facial model with a 38 regional muscle blocks corresponding to predefined regions of expression of the facial structure. Muscle actions are represented as collections of predefined actions in these regions. An action in one of these regions may or may not cause other changes in other adjacent regions. Regional muscle blocks are interconnected by a spring network. Facial action units (Ekman and Friesen, 1978) are created by applying muscle forces to deform the spring network.

Waters (1987) developed a face model that includes two types of muscles: linear muscles that pull and sphincter muscles that squeeze. Like Platt and Badler, he used a mass-and-spring model for the skin and muscles. However, Waters' muscles have directional (vector) properties. Waters' work will be discussed in more detail later. A finite element method is also used to simulate the (visco-)elastic properties of the skin and to model the facial muscles (Koch et al., 1996; Koch and Bosshard, 1998).

Terzopoulos and Waters (1990) proposed a facial model that models detailed anatomical structure and dynamics of the human face. Their three layers of deformable mesh correspond to skin, fatty tissue, and muscle tied to bone. Elastic spring elements connect each mesh node and each layer. Muscle forces propagate through the mesh systems to create animation. This model achieves great realism, however, simulating volumetric deformations with three-dimensional lattices requires extensive computation.

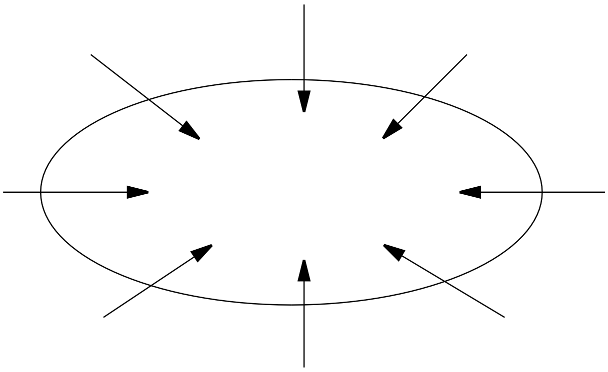


Figure 2.3: Waters' sphincter muscle model.

Lee et al. (1995) presented models of physics-based synthetic skin and muscle layers based on earlier work by Terzopoulos and Waters (1990). The face model consists of three components: a biological tissue layer with non-linear deformation properties, a muscle layer knit together under the skin, and an impenetrable skull structure beneath the muscle layer. The synthetic tissues are modeled as triangular prism elements that are divided into the epidermal surface, the fascia surface, and the skull surface. Spring elements connecting the epidermal and fascia layers simulate skin elasticity. Spring elements that affect muscle forces connect the fascia skull layers. A muscle grid fitting approach (Kähler et al., 2001) was presented to build muscle from a sketched layout of facial muscles. The user first sketches the basic muscle grids row by row. The muscle grids are then refined automatically to fit the geometry. Quadric shaped fibers are inserted to the grids to create the muscles. This process, however, requires the fitting of a generic model of a real human skull to the skin mesh, which is a manual and time-consuming task.

The multi-layer muscle-based models achieve spectacular realism and fidelity; however, tremendous computation is required. Therefore, they are not widely used for real-time animation. Moreover, extensive tuning is needed to model a specific face.

Waters' muscle model

A very successful muscle model was proposed by Waters (Waters, 1987; Parke and Waters, 1996). It includes three types of muscle: vector muscles that pull, sphincter muscles that squeeze and sheet muscles for the **Frontalis**. A sphincter muscle contracts around an imaginary central point to draw the surface surrounding, for instance, the mouth together like the tightening of material at the top of a string bag (see Figure 2.3). A sheet muscle consists of almost-parallel fibers which lie in flat bundles. When it contracts, the skin is not only drawn to one fiber's insertion points but to all fibers' insertion points (see Figure 2.4).

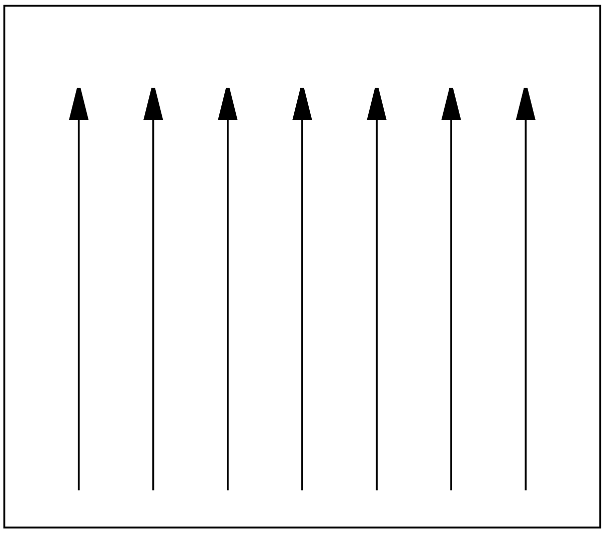


Figure 2.4: Waters' sheet muscle model.

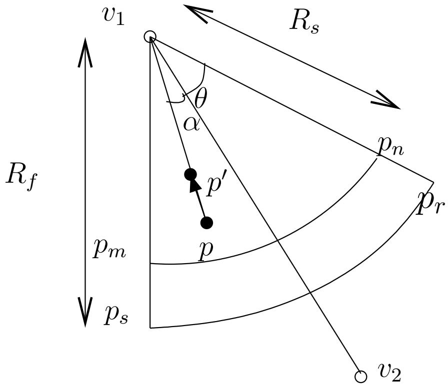


Figure 2.5: Waters' vector muscle model.

Like Platt and Badler, Waters used a mass-and-spring model for the skin and vector muscles. However, Waters' muscles have directional (vector) properties that make them independent of the face topology. Each muscle has a zone of influence, which is illustrated in Figure 2.5.

The muscle is modeled as a vector from v_2 to v_1. R_s and R_f represent the fall-off radius start and finish respectively. The new vertex p' of an arbitrary vertex p located on the mesh within the segment $v_1 p_r p_s$, along the vector (p, v_1), is computed as follows[1]:

$$p' = p + kr\frac{pv_1}{\| pv_1 \|}\cos\alpha$$

where α is the angle between the vector (v_1, v_2) and (v_1, p), D is $||v_1 - p||$, k is a fixed constant representing the elasticity of skin, and r is the radial displacement parameter:

$$r = \begin{cases} \cos\left(1 - \frac{D}{R_s}\right) & \text{for } p \text{ inside sector } (v_1 p_n p_m), \\ \cos\left(\frac{D - R_s}{R_f - R_s}\right) & \text{for } p \text{ inside sector } (p_n p_r p_s p_m). \end{cases}$$

The problem with the model arises when a mesh vertex is under the influence of multiple muscle actions. Waters combined these muscle actions by applying the displacements caused by them on a vertex one by one. When the vertex is shifted out of the zone of influence of adjoining muscle vectors and the contractions become isometric, this approach produces unnatural results, which has been discussed by Wang (1993). Wang took another approach by summing up the displacement and then applying this to the vertex. This approach itself also generates strange results as shown in Figure 2.12(j) and 2.13(b). Both Waters and Wang also tried to eliminate unnatural results by truncating the displacement of a vertex by the vertex maximum displaced distance. This process requires that the maximum displacement of each individual vertex is determined, which is a difficult task. Assigning this distance as the maximum of the displacements generated by each single muscle maximum contraction will not eliminate the problem completely as the vertex is still shifted out of the muscles' zone of influence. Assigning this distance as the minimum of the displacements generated by each single muscle's maximum contraction will incorrectly truncate the displacements when a single muscle contracts. Our solution for this problem will be discussed in Section 2.4.1.

2.2.4 The lips

The lips are a very mobile part of the face. They can move in different directions. They press against each other during an expression of anger. They open and rotate to create funnel shape during the speech for several phonemes such as "ow" in "how". Their corners move outward and upward during smiling, outward and downward during an expression of sadness. The lips are also a very expressive part of the face and they participate in the articulation of speech. Therefore, good modeling of the lip motion and of the lips themselves is a major requirement for a high-quality facial animation system.

---

[1] The formula presented here is for implementation purposes. The explanation of how this formula is derived is out of the scope of this thesis. For a better understanding of this formula, readers should consult (Waters, 1987).

As a part of the face, the lips are normally modeled and deformed in the way the face is modeled. For example, in Parke's face model (1972) or the Greta face model (Pasquariello and Pelachaud, 2001), the lips are represented as a polygonal sub-mesh of the whole face. On the other hands, the lips of Billy (Reeves, 1990) or Waite's face model (1989), are a part of the B-spline surface that represents the whole face.

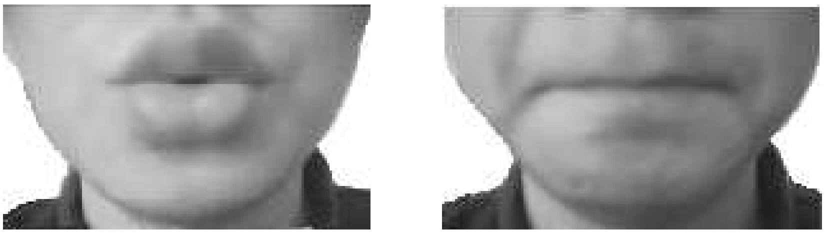
King et al. (2000) separate the lips from the face model and specially model the lips to enhance the realism of facial animation. While the face is still represented as a skin of polygons, the lips are represented by a B-spline surface with a 16×9 control grid. Note that, as the lips are just a small part of the whole face and there are almost no wrinkles on the lips, this approach does not suffer from the disadvantage of B-spline surfaces. Again, the B-spline surface is chosen because it is easy to deform the surface by simply moving control vertices and the surface remains smooth after being distorted. Their lip model has both external and internal lip geometry. The external geometry contains all the red area of the lips. The internal one contains the mucous membrane inside the lips. This internal geometry is important in keeping realism when opening the mouth. It avoids any possible observation of an edge when looking at the lips from the outside. The lips are deformed by a set a parameters. All the parameters except the orbicularis and jaw parameters are mapped to the vector displacements of the control grid vertices. The orbicularis oris constricts the shape of the lips into an oval while also extruding them. The jaw parameters lower the lower lip to open the mouth, move it inward and outward, and skew it to one side or the other. After deformations, the B-spline surface is polygonalized into in a skin of polygons with predefined topology for rendering purpose. The predefined topology are connected to the rest of the facial geometry so that the lips can be rendered as a part of the whole face model.

2.2.5 Summary

Because our main aim is to produce realistic facial expression with detailed wrinkles, we have decided to mainly use skin of polygons for our face model. We follow King et al. (2000) to separate the lips and model them with a B-spline surface because of its advantage when applied to a small and smooth surface.

For deforming our face model, we have to select the appropriate techniques carefully. Although key-frame, parameterized and pseudo-muscle-based techniques produce animation in real-time, they fail to produce realistic facial expressions as they cannot provide a generic way of producing bulges and wrinkles in the skin and handle multiple parameter/muscle interaction. Multi-layer muscle-based approaches with several layers of facial structure, on the other hand, produce realistic expressions; however, they require massive computation and cannot obtain real-time animation in standard personal computers.

We decided to animate our face model mainly based on an extension of Waters' vector muscle model (1987) as it is physics-based and not very computationally costly for real-time animation. The extensions will be discussed in Section 2.4. The actions of the eyelid and jaw rotation are a bit more complex



(a) (b)

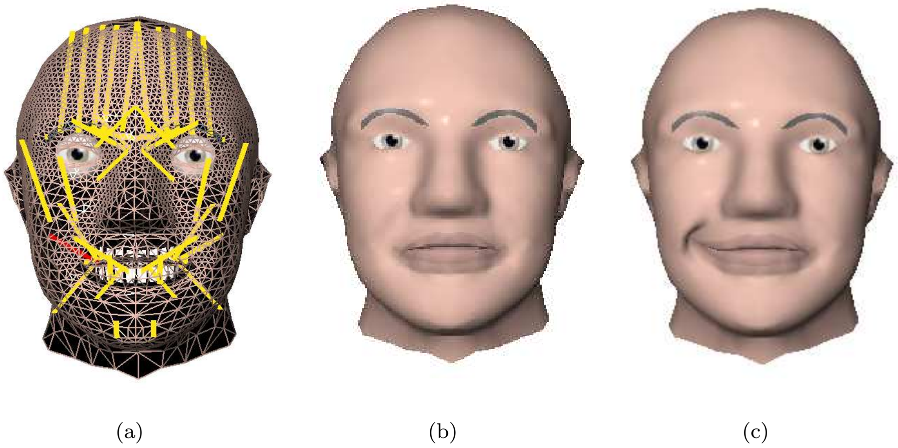
Figure 2.6: The action of the real **Orbicularis Oris** muscle: (a) pushing out the lips and (b) pressing the lips.

than the vector muscles. Their effects look like rotations of specific parts of the face about an axis. Parameterized animation techniques are preferred to the muscle-based techniques in this case. The muscle of the mouth, **Orbicularis Oris**, is too complicated to model with Waters' model of sphincter muscle. As can be seen in Figure 2.3, Waters' sphincter muscle only produces the effect of drawing the lips together to an imaginary central point while the action of the real **Orbicularis Oris** is far more complex (see Figure 2.6). We follow King et al. (2000) and use the parameterized animation techniques to model the action of the **Orbicularis Oris**.

2.3 Our face model

Our face model contains a polygonal face mesh and a B-spline surface of the lips. It is shown in Figure 2.7(b).

The face mesh contains triangular polygons with vertices that are sorted by region. The face mesh is divided into regions to improve the performance of the muscle model. This division helps to prevent visual artifacts generated by the displacement of vertices in the regions that are not affected by a muscle's contraction. Usually, the displacements of polygon vertices by a muscle contraction are controlled by the muscle's zone of influence which has a predefined shape and is independent of the face mesh. However, in some cases, this zone of influence cannot cover exactly the region it should cover in the face model. For example, without specifying the eyelid regions, the polygons in the eyelids can be distorted by **Frontalis** muscles as the eyelids may lie in their zone of influence, which results in unnatural animation. We also use the region division to improve the muscle model. To be able to deform the face model by a particular muscle, the algorithm for the muscle model has to search for all vertices on the face mesh that lie inside the zone of influence of the muscle. The region division is used to reduce this search by skipping vertices that do not lie in the regions



(a) (b) (c)

Figure 2.7: (a) The wire-framed face with muscles; (b) the neutral face; and (c) the effect of the left Zygomatic Major.

that the muscle has effect on. This will be discussed later.

As mentioned before, we chose a simple B-spline surface to represent the lips to ensure their smoothness after being distorted by the muscles. The B-spline surface of the lips is then polygonalized into a predefined topology of vertices and triangles for rendering purpose. These vertices are connected to each other and connected to the rest of the face model in a predefined manner. Then, the lips can be rendered as a part of the whole face model. We will discuss this process later when we describe the lips. The whole face model is rendered with Phong shading (Bui, 1975) based on OpenGL technologies.

2.3.1 The face mesh

Initially, our face mesh data was obtained from a 3D scanner. We processed the data to improve the animation performance while keeping the high quality of the model. This process contains two phases.

In the first phase, we reduced the number of vertices and polygons in some parts of the face. We took the idea from Parke (1972) and Pasquariello and Pelachaud (2001) to have less vertices and polygons in the non-expressive parts. The number of polygons is maintained in the expressive parts, which are the areas around the eyes, the nose, the mouth and the forehead. In the end, the total number of polygons was reduced significantly. Starting with about 30,000 polygons, our final 3D face model contains only 2,480 vertices and 4,744 polygons. The reduction of polygons in other parts increases the animation speed. This approach, however, still preserves the great level of detail in these expressive parts of the face that are involved in sending communication signals and

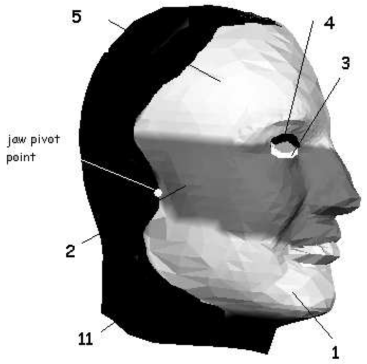


Figure 2.8: The region division on right half of the face model.

emotional expressions.

In the second phase, we divided the face model into regions. Region division has been used in Greta (Pasquariello and Pelachaud, 2001) to limit and control the displacement of polygon vertices induced by facial animation parameters (FAPs). Similarly, we use that technique to control the displacement of polygon vertices produced by muscle contractions. As mentioned before, this helps to prevent visual artifacts generated by the displacement of vertices in the regions that are not affected by a muscle's contraction. The other use of the region division, which is to improve the speed of the muscle model's algorithm, will be discussed in Section 2.4.1.

The human facial muscles are concentrated into six main areas in the face: the left and right parts of the lower, middle and upper face. Based on this distribution, we divide the face model into eleven regions:

- (1) right lower face,
- (2) right middle face,
- (3) right lower eyelid,
- (4) right upper eyelid,
- (5) right upper face,
- (6) left lower face,
- (7) left middle face,
- (8) left lower eyelid,
- (9) left upper eyelid,

- (10) left upper face,
- (11) the non-animated region consisting of the rest of the head.

The regions on the right half of the face model can be seen in Figure 2.8, while the regions on the left half of the face model are in similar positions.

After the face model is divided into regions, in order to improve the speed of the muscle model's algorithm, we have to specify on which regions each muscle has effect. We found that the division scheme we proposed above keeps this job simple. The face model can be divided into smaller regions as in the Greta face model (Pasquariello and Pelachaud, 2001). However, this will harden the job of finding influenced regions of a muscle. Dividing the face model into bigger regions would reduce the number of regions, which reduces the improvement on the speed of the muscle model's algorithm.

We reordered the vertices of the face mesh in the data file by regions. We introduced a simple description file associated with the data file which looks like:

```
Lower face start vertex 0
Lower face end vertex 159
Middle face start vertex 160
Middle face end vertex 484 ...
```

We use this face model as a source face model that can be automatically deformed to represent any newly created face model. We will describe this method in Chapter 3. With this method, we do not need to manually process another face model again when we want to produce animation on that face model. This prevents the approach in this thesis from losing generality when applied to a new face model.

2.3.2 The lip model

We based our lip model on the proposal in (King et al., 2000). Our lip model is a B-spline surface with 24×6 control grid. The lips are shown in Figure 2.10. The control vertices of the lips are shown in Figure 2.10 as well.

To deform the lips, the facial muscles move the control points of the B-spline surface instead of directly altering the vertices of the lip mesh. The control points are then used to re-determine the result of the formula of the B-spline surface.

The B-spline surface is polygonalized to connect to the rest of the face model for rendering purpose. At the preprocessing steps, the polygons that create the original lips are chopped off from the face model. The face model without the lips is shown in Figure 2.11(a). Every time the B-spline surface of the lips is updated, it is sampled by vertices that are pre-arranged in a grid. These vertices are evenly distributed on the B-spline surface. The sampling vertices are connected to create a triangular mesh. This mesh is connected to the rest of

No.	Muscle name	Regions of Influence
1	Right Zygomatic Major	1,2
2	Right Zygomatic Minor	2
3	Right Triangularis	1,2
4	Right Risorius	1,2
5	Right Depressor Labii	1
6	Right Mentalis	1
7	Orbicularis Oris (Lip Funneler)	lip model
8	Orbicularis Oris (Lip Pressor)	lip model
9	Right Frontalis Medialis	5,10
10	Right Frontalis Lateralis	5
11	Right Levator Labii Nasi	2,5
12	Right Levator Labii Superioris	2,5
13	Right Depressor Supercilii	2,5
14	Right Corrugator Supercilii	2,5
15	Right Depressor Glabelle	2,5
16	Levator Palpebrae Superios	4
17	Orbicularis Oculi Palebralis (Eye closing)	3,4
18	Orbicularis Oculi Orbitalis	2,5
19	Masseter (jaw rotation)	1,6

Table 2.1: Implemented muscles in our face model (right side of the face).

the face model. This is shown in Figure 2.11(b). Then, the lips can be rendered as a part of the whole face model.

2.4 The muscles that drive the facial animation

In this section, we will describe the muscles that control the animation of the face model. The muscles are mainly based on Waters' muscle model (Waters, 1987; Parke and Waters, 1996). As mentioned before, Waters has modeled three types of muscle: the vector muscles used for the majority of facial muscles, the sheet muscles for the **Frontalis**, and the sphincter muscles for the **Orbicularis**

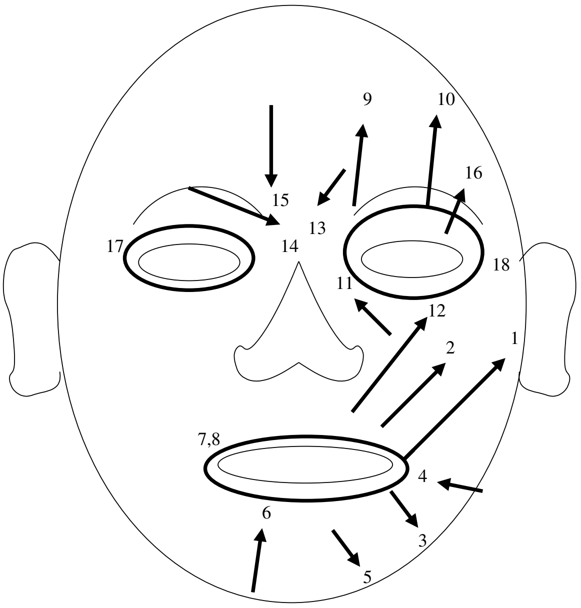


Figure 2.9: Schematic of all implemented facial muscles.

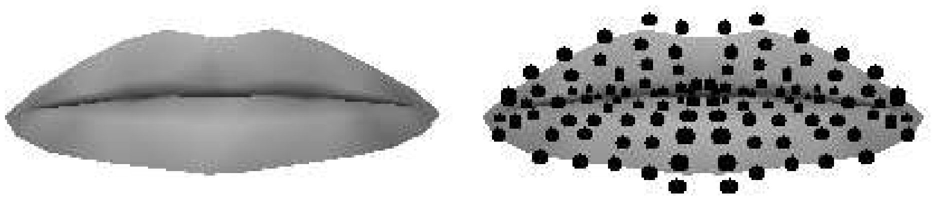


Figure 2.10: The lips (left) and the control points of the lips (right).

Oris.

We have extended the vector muscle model to improve the combination of multiple muscle actions, to generate bulges and wrinkles, and to increase the animation speed (Section 2.4.1). For the **Frontalis**, we use a set of vector muscles instead of the sheet muscle because the forehead is not completely flat. The mouth muscle, **Orbicularis Oris**, is rather complex. As discussed before, modeling this muscle by Waters' sphincter muscle does not give realistic results. We follow King et al. (2000) to model this muscle with a parameterized technique. We will present this in Section 2.4.2. For the eyelid and jaw rotation, the muscle-based techniques are not suitable. Parameterized animation techniques are preferred in this case as the eyelid and jaw rotation do not really produce

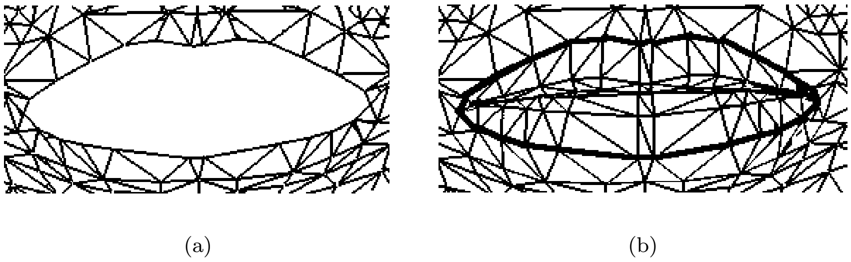

(a) (b)

Figure 2.11: (a) The face mesh without the lips; and (b) the lips are integrated to the whole face mesh.

bulges and wrinkles, which will be described in Section 2.4.3) and 2.4.4 respectively. We have also implemented the eye tracking algorithm proposed in (Parke, 1974) to enable gaze behavior, which will be presented in Section 2.4.5.

Table 2.1 shows all the muscles in the right side of our face model. They are also illustrated in Figure 2.7(a) and Figure 2.9. The detailed description of all these muscles will be presented in Appendix B. An example of the effect of a facial muscle is shown in Figure 2.7(c).

2.4.1 The vector muscles

We have improved the vector muscle model from (Waters, 1987) to eliminate the artifacts created during multiple muscle actions as we mentioned in Section 2.2.3. Next, we add a mechanism to generate wrinkles and bulges to increase the realism of the facial expressions. Finally, we introduce techniques to reduce the computation burden on the muscle model in order to enhance the animation performance.

Combination of multiple muscle actions

Human facial muscles contract in parallel to create facial expressions. We combine muscle contractions in our face model by simulating their parallelism. For a vertex inside a multiple muscles' zone of influence, we repeatedly apply small units δ_c of contraction levels to the vertex until there is no more contraction to apply. Smaller δ_c produces more correct simulation of parallelism; however it needs more computation. By trial and error, we found that the δ_c of 0.2 (the maximum value of a muscle contraction level is 1.0) produces reasonable result while keeping the animation in real-time.

Each time the small unit of contraction is applied, the displacements of the vertex, which are caused by the muscles that have the vertex in their zone of influence, are computed first without being applied. Then, they are summed up and applied to the vertex. As soon as a vertex moves out of the zone of influence of some muscle, it is no longer modified by that muscle.

Our approach with $\delta_c = 0.2$ produces results as in Figure 2.12(f) and 2.13(a). Compared to the result obtained by just adding the displacement vector in Figure 2.12(j) and 2.13(b), our result looks convincingly more realistic and natural. An example of how this method is applied to our face model is shown in Figure 2.14. As can be seen from this figure, an artifact appears between the eyebrows when sadness is expressed if we just add displacements of the vertex.

This approach is also applied to the **Orbicularis Oris** and **Orbicularis Oculi** muscles. Note that the calculation of the muscle distortion is only repeated several times (5 times for $\delta_c = 0.2$), which still keeps our muscle model fast enough for real-time animation on a standard personal computer.

Bulges and wrinkles

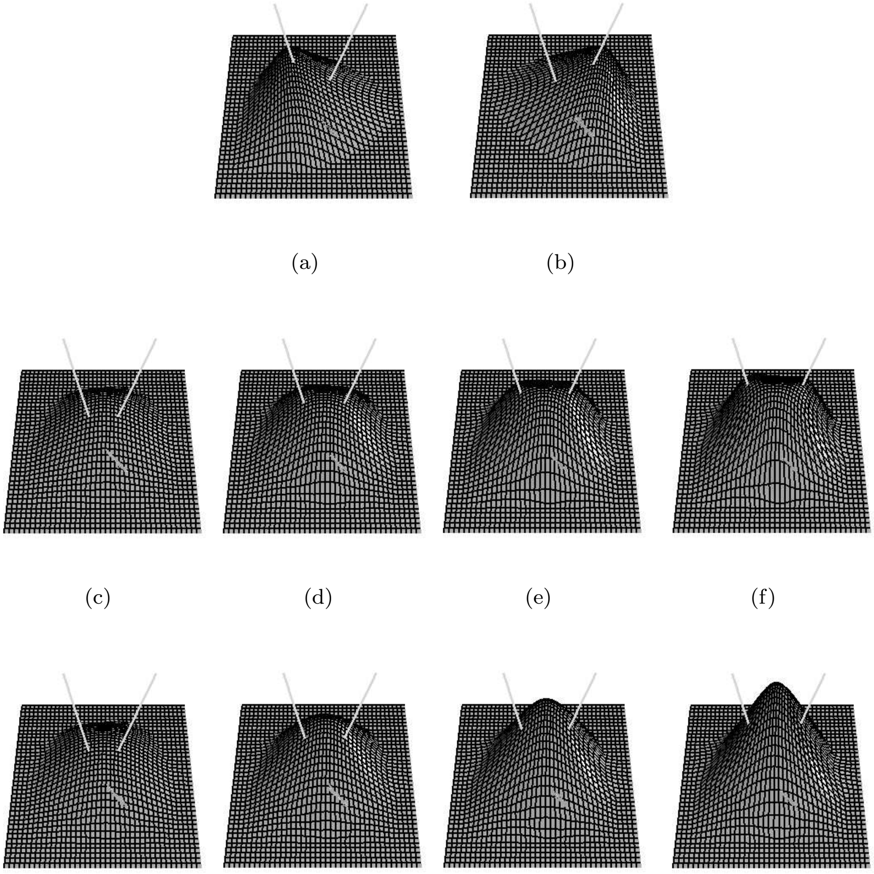
Bulges and wrinkles are created during the contraction of facial muscles. They play an important role in interpreting facial expressions. First, they make the expressions more visible. Second, the intensity of the emotional expressions are represented by the depth of the wrinkles. Third, the wrinkles are used to distinguish between emotional expressions. For example, Ekman and Friesen (1975) have pointed out that there are horizontal wrinkles in the forehead during the expression of surprise, fear and sadness; however, in the expressions of anger, disgust or happiness they are absent.

Pseudo-muscles or parameterization approaches usually fail to create wrinkles because they ignore the underlying anatomy of the face. Wrinkles are easier to generate with physics-based modeling or texture techniques like bump mapping. Wu et al. (1994) use a physics-based model with a simplified version of head anatomy without bones to generate immediate and permanent wrinkles based on the plastic-visco-elastic properties of the facial skin. Viscosity is responsible for temporary wrinkles during muscle contraction while plasticity is for permanent ones. Again, this physics-based technique requires heavy computation; therefore it is not suitable for real-time animation.

Moubaraki et al. (1995) use bump mapping to generate and animate wrinkles without moving the vertices by producing perturbations of the surface normals that alter the shading of a surface. Arbitrary wrinkles can appear on a smooth geometric surface by defining wrinkle functions (Blinn, 1978). This technique generates wrinkles easily by varying wrinkle function parameters. Bump mapping is usually used to generate wrinkles on the forehead or permanent wrinkles, such as in the Greta face model (Paradiso and L'Abbate, 2001). The bump mapping technique is relatively demanding computationally as it requires about twice the computing effort needed for conventional color texture mapping.

Spline segments are also used to model the bulges and wrinkles for spline patches based face model (Viaud and Yahia, 1992). However, this approach requires many more control points in order to represent the wrinkles. This makes the face model much more complex, which is not suitable for real-time animation.

We present in this section a simple approach to create realistic bulges and



(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 2.12: The effect of a single muscle on the mesh (a and b); The effect of two muscles on the mesh (shown step by step): by simulating parallelism (c,d,e,f); by adding displacement (g,h,i,j).

wrinkles that works well with the Waters' muscle model. For the sake of simplicity, we assume that the muscles lie parallel to the facial skin and that the heights of the wrinkles for each muscle are the same. We assign predefined values to the height of the wrinkles and the number of wrinkles (N_w) created by the contraction of each muscle. These values might be computed taking the volume preservation and a model of a skull into consideration.

The wrinkle amplitude is calculated for all vertices that are originally (before applying the displacement caused by the muscle contraction) inside the

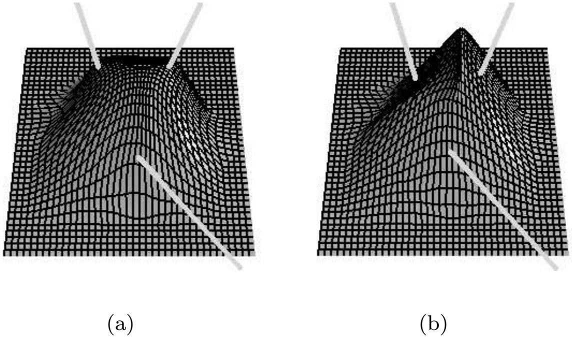

(a) (b)

Figure 2.13: The effect of three muscles on the mesh: by simulating parallelism (a); by adding displacement (b).

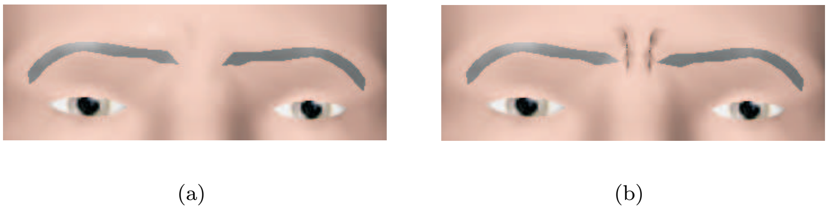

(a) (b)

Figure 2.14: Sadness eyebrows with handling multiple muscle contractions: by simulating parallelism (a); by adding displacement (b).

$p_lp_kp_rv_3p_t$ region (see Figure 2.15), where the distance from p_l and p_k to p_tp_r is:

$$L = \frac{3}{4}|v_1v_3| = \frac{3}{4}R_f \cos\theta.$$

The wrinkle amplitude at a vertex p is a function of the distance l from p to p_tp_r. l is periodically mapped into $[0, 2b)$ with a frequency of N_w:

$$u(l) = l - \lfloor\frac{l}{b}\rfloor b$$

where

$$b = \frac{L}{2N_w}$$

and $\lfloor\cdots\rfloor$ truncates a real number to the biggest integer number that is smaller than it. We use a series of parabolas to represent the wrinkle function (see Figure

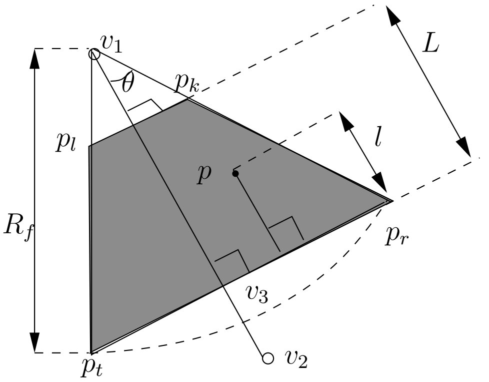


Figure 2.15: The zone that contains wrinkles due to the contraction of a vector muscle.

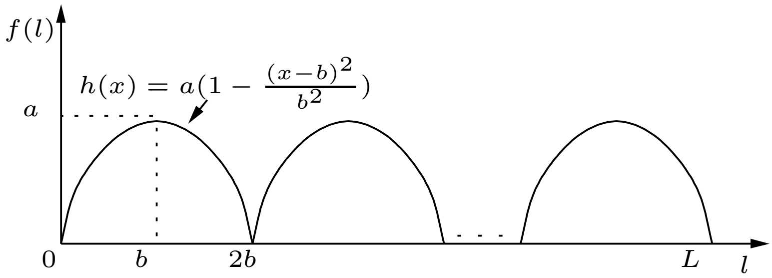


Figure 2.16: The wrinkle function.

2.16), which is described as follows:

$$f(l) = a\left(1 - \frac{(u(l)-b)^2}{b^2}\right),$$

where a is the height of the wrinkles.

The wrinkle amplitudes are applied to the direction of the normal of the vertices after the vertices are displaced by the muscle contractions. For vertices that are inside the zone of influence of multiple muscles, only the maximum wrinkle amplitude caused by these muscle is used.

Note that the wrinkle function is just a parabola function, the values of which are fast to compute in order to maintain the animation in real-time.

Recall from Section 2.2 that interpolated shading algorithms are used to give a continuous curved appearance to a facial surface made up of polygons. However, a defect of these algorithms, named the "unrepresentative vertex normal"

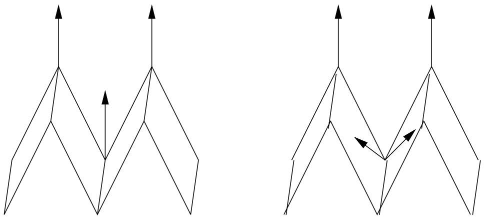
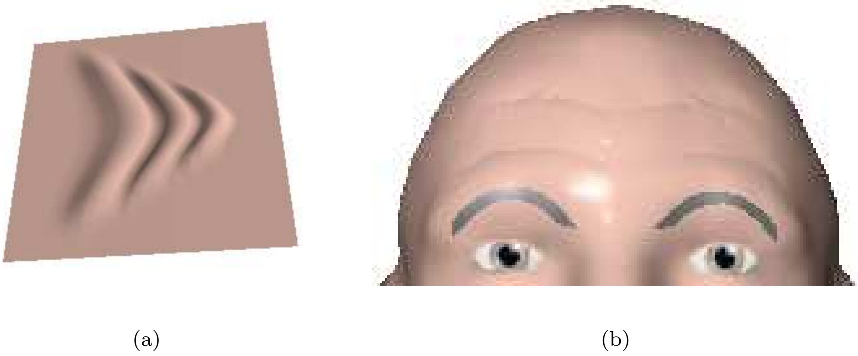


Figure 2.17: The "unrepresentative vertex normal" problem and its solution.



(a) (b)

Figure 2.18: The wrinkles due to the muscle contraction.

defect, prevents the wrinkles from being visible (see Figure 2.17). To make the wrinkles more visibly, we have to eliminate this defect for vertices that lie on the "internal foot" of the wrinkles. These vertices are the ones with the original distance to $p_t p_r$ being:

$2b, 4b, \ldots, L - 2b.$

Instead of using these vertices' average vertex normal, we use the normal of the triangular polygons that contain them.

An example of the wrinkles is shown in Figure 2.18(a). The wrinkles created on the forehead are shown in Figure 2.18(b).

The improvements on the animation performance

The vector muscle model is suitable for 3D face models with a small number of polygons. When the number of polygons increases, enabling more realistic facial expression, the muscle model's heavy computation prevents real-time animation. We have sped up the animation by improving the muscle model with a "cut-off" technique. Originally, the core of the implementation of the muscle model by Waters (Waters, 1996) looks like:

```
for all vertices
    if the vertex is inside the muscle's zone of influence
        then calculate and apply the displacement of the vertex
```

Let us analyze the complexity of the algorithm. For each muscle, the algorithm has to check for each vertex in the mesh if the vertex is inside the muscle's zone of influence. To do this, the algorithm has to calculate the distance from the vertex to the head of the muscle pv_1, and has to calculate the angle pv_1p_m (Figure 2.5). Finally, a cosine function has to be computed. The complexity of the algorithm depends on the number of vertices it has to check to see whether they are inside the zone of influence.

Our 3D face model has been divided into regions (see Section 2.3.1), and with the knowledge of the position of the facial muscles, we know which regions a muscle will have influence on. Each muscle is associated with a flag, indicating which regions of the face model it has effect on. The original implementation is now modified as follows:

```
for all vertices
  if the vertex is inside the region that the muscle has effect on
    if the vertex is inside the muscle's zone of influence
      then calculate and apply the displacement of the vertex
```

This technique first eliminates all the vertices that are outside the region that the muscle has an effect on by a single check of the description file (see Section 2.3.1). As the number of vertices increases, this will hugely reduce the computation of checking lengths and angles. The improvement on our model of 2480 vertices is shown in Table 2.2. We increase the animation speed still further by applying some classic implementation techniques. First of all, the data structure plays an important role in reducing computation time. The vertices are stored separately from the polygons. The polygons contain only links to the vertices. Computation time is reduced because the calculation of the displacement of each vertex does not have to be duplicated for each of the polygons. This also prevents the recalculation of the average vertex normal of every polygon vertex for Phong shading. Second, all the condition checking with division and i-th root mathematical operators are replaced by the ones with multiplication and i-th power ones. The improvement after using all above techniques is also shown in Table 2.2.

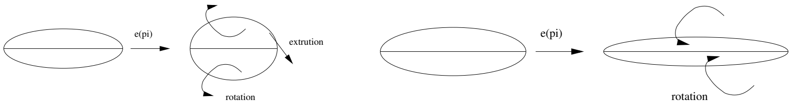
2.4.2 Orbicularis Oris

Physiologically, the **Orbicularis Oris** is not just a simple sphincter muscle but a combination of muscles that can drive the mouth in different directions (Basmajian, 1974). Our parameterization-based implementation for the **Orbicularis Oris** is adopted from (King et al., 2000).

Recall from Section 2.3.2, the lips are represented with a B-spline surface with a 24×6 control grid. In our model, the **Orbicularis Oris** affects only the

	Animation speed (frames per second)
Before improvement	20.5
After muscle model improvement	30.5
After all improvements	35.0

Table 2.2: The result of the improvement on the animation (on a Pentium III 800Mhz, 256MB RAM, NVidia GeForce3 video card).



(a) (b)

Figure 2.19: Illustration of the action of **Orbicularis Oris**: pushing out the lips (a); pressing the lips (b).

lip surface. The surface is deformed by displacing the control points. We model two actions of the **Orbicularis Oris**: pushing out and pressing the lips.

Following King et al. (2000), the displacement of a control point p_i due to the contraction of the muscle **Orbicularis Oris** that pushes out the lips is described as:

$p_i' = o(\theta_i + e(p_i) + x_i)$

where o is the contraction level of the **Orbicularis Oris** muscle; θ is the maximum rotation due to the puckering of the lips, the rotation is 20°, and in opposite directions for the upper and lower lips; x_i is the maximum extrusion from the contraction of the **Orbicularis Oris**; and $e_i(p)$ returns a motion vector for moving the control point p to a point on the ellipse created when contracting the **Orbicularis Oris** (King et al., 2000). This is illustrated in Figure 2.19(a).

The displacement due to the contraction of the muscle **Orbicularis Oris** that presses the lips is described as:

$p_i' = o(\theta_i + e(p_i))$

where o is the contraction level of the **Orbicularis Oris** muscle; θ is the maximum rotation due to the pressing of the lips, the rotation is 20°, and in opposite direction compared to the pushing out of the lip; and $e_i(p)$ returns a motion vector for moving the control point p to a point on the ellipse created when contracting the **Orbicularis Oris**. This is illustrated in Figure 2.19(b).

Figure 2.20 shows the deformation of the lips due to the contraction of **Orbicularis Oris**.

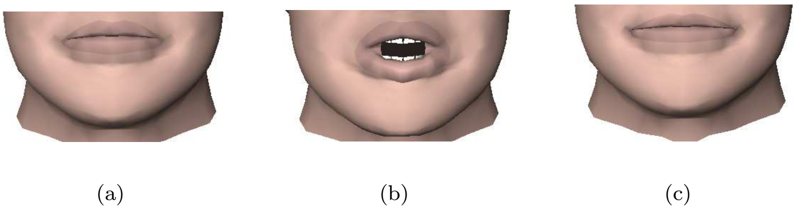


Figure 2.20: The deformation of the lips due to the contraction of **Orbicularis Oris**: neutral position (a); pushing out the lips (b); pressing the lips (c).

2.4.3 Orbicularis Oculi

The **Orbicularis Oculi** contains two parts: the **Pars Palpebralis** that opens and closes the eyelid, and the **Pars Orbitalis** that squeezes the eye.

We have adopted the algorithm from Parke (1974) for the closing and opening of the eyelid. The eyelid is opened and closed by combining a variation of the spherical mapping technique with linear interpolation. The squeezing of the eye is implemented with the sphincter muscle from (Waters, 1987).

We note that the closing of the eyelid and the squeezing of the eye frequently occur together. The squeezing of the eye brings about the the closing of the eyelid. The closing of the eyelid also causes the squeezing of the eye when one tries to close only one of the eyes. We capture this relationship by modifying the contraction level of the **Pars Palpebralis** ($c_{closing}$) and the **Pars Orbitalis** ($c_{squeezing}$) in one eye as follows:

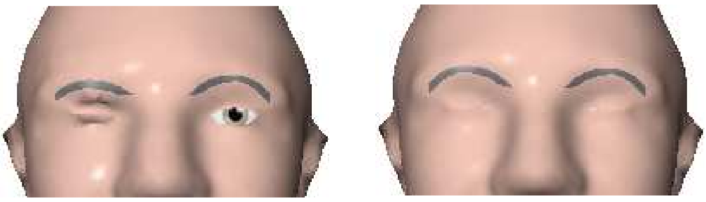
```
if $c_{squeezing} > 0.5c_{closing}$ then
  $c_{closing} = max(1.0, 2 \cdot c_{squeezing})$
else if try to close only one eye then
  $c_{squeezing} = 0.5c_{closing}$
```

The closing of one eye and that of both eyes are illustrated in Figure 2.21.

2.4.4 Jaw rotation

The jaw is opened by rotating the vertices of the lower part of the face model about a jaw pivot axis (Parke, 1974). The axis of rotation is parallel to the X-axis and passes through the jaw pivot point (see Figure 2.8). The vertices located in the lower face region are affected by jaw rotation. The lower lip, lower teeth, and the corner of the mouth rotate with the jaw.

To produce a natural oval-looking mouth, the vertices on the lower lip are rotated by different amounts. The vertices in the middle of the lower lip are rotated by the same amount as the jaw rotation. The amount of rotation decreases



(a) (b)

Figure 2.21: The closing of only one eye (a) and both eyes (b).

when applied to the vertices which are closer to the corners of the mouth. The corners of the mouth are rotated by one third of the jaw rotation.

The upper lip is also affected by the jaw rotation. The vertices on the upper lip are pulled down with different amounts. The amount is zero for vertices in the middle of the upper lip. This amount increases when the vertices are closer to the corners of the mouth.

2.4.5 Eyeball rotation

Gaze behavior is a part of non-verbal skills such as eye contacts and emotional expressions. To enable gaze behavior, we have implemented the eye tracking algorithm proposed in (Parke, 1974). This algorithm has been designed to calculate the orientation angles of the two eyes, Ar, Al and Blr, so that the eyes can track a target (see Figure 2.22). Whenever the position of the tracking target changes, the values of the eye orientation angles are updated. Then, the eyes are updated with the new orientation angles.

The eye movement is independent of the facial muscle movements. Note that both eyes can only track one object (but not two) at a time and an eye can only move in a range constrained by the eye hole, which has an ellipse shape. We have added this constraint to the algorithm so the eyes cannot rotate to impossible positions. This constraint restricts the eyes from tracking objects that are out of the visible area, e.g., positions that are too close to the face or very left/right of the face. This is illustrated in Figure 2.23.

2.5 Experimental result

We have tested our model to display emotional facial expressions. Emotion intensities are converted to the muscle contraction levels by the rules that will be described in Chapter 6. The muscle contraction levels are then applied in the face model to generate facial expressions. The animation speed is relatively high on a Pentium II 800Mhz computer, 35.0 frames per second.

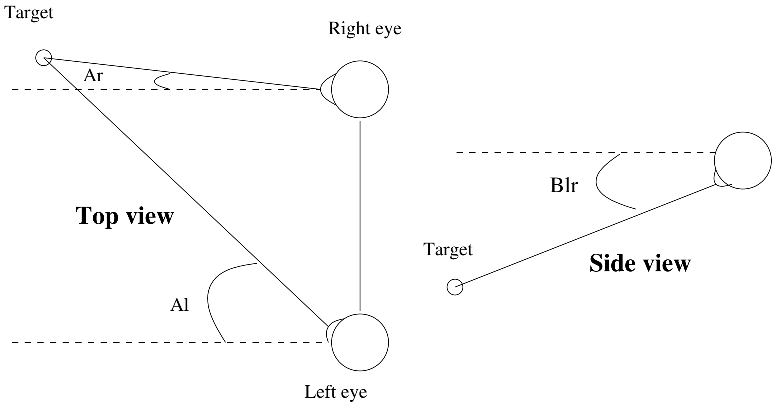


Figure 2.22: Eye orientation angles.

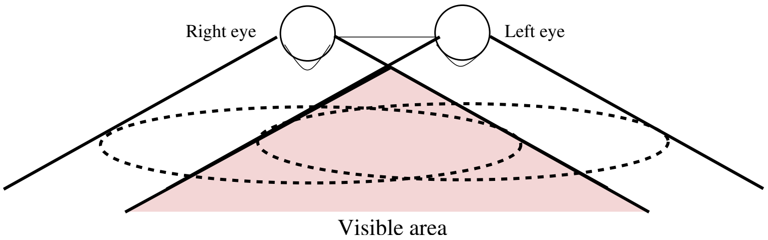


Figure 2.23: Eye visible area.

How the wrinkles and the handling of multiple muscle actions improve the facial expressions is shown in Figure 2.24. Our approach eliminates the artifacts appearing between the two eyebrows. With the visibility of the wrinkles the expression looks more realistic and is easier to recognize.

Some examples of emotional facial expression are shown in Figures 2.25, 2.26 and 2.27. The natural jaw rotation can be seen in the surprised face (Figures 2.25). Also in the surprised face, the combination of the **Frontalis** muscles generates clearly visible wrinkles in the forehead when the eyebrows are raised. The natural nasolabial bulges generated by the **Zygomatic** muscles can be seen in the happy face. The lips look smooth after distortion. Finally, the small bulges and wrinkles in some regions can be seen in the sad face which make it more expressive (Figure 2.27).

The result shows that our model has achieved the aim of keeping it simple and fast while generating natural and realistic facial expressions.

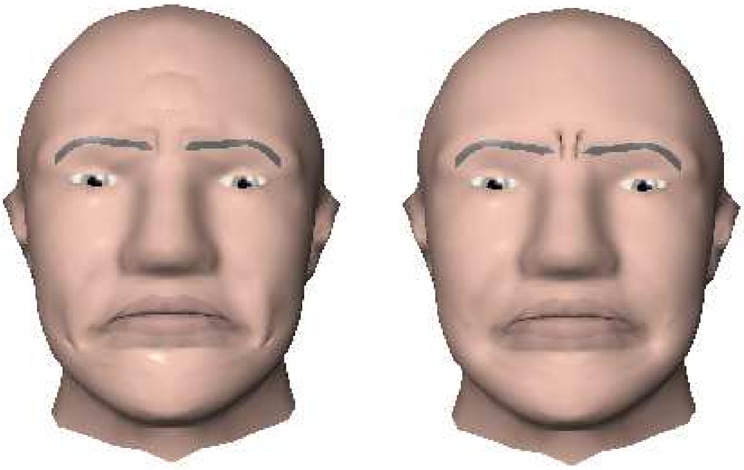

(a) (b)

Figure 2.24: Sadness on our face model with (a) and without (b) wrinkles and handling multiple muscle contractions.

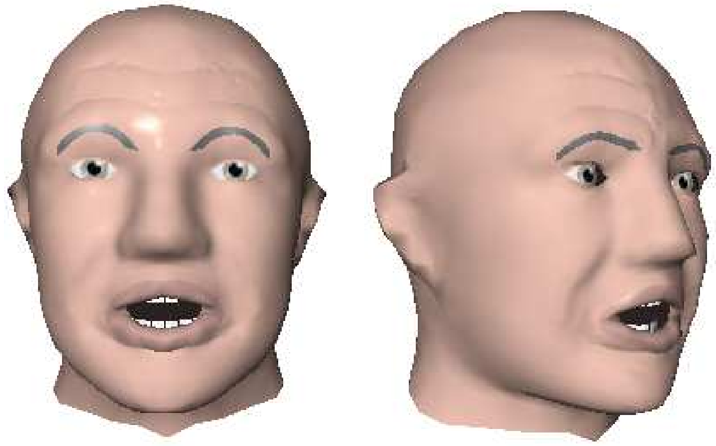


Figure 2.25: The face model displays surprise.

2.6 Conclusion

In this chapter, we have described a simple muscle-based 3D face model that can produce realistic facial expressions in real-time. The face model contains a

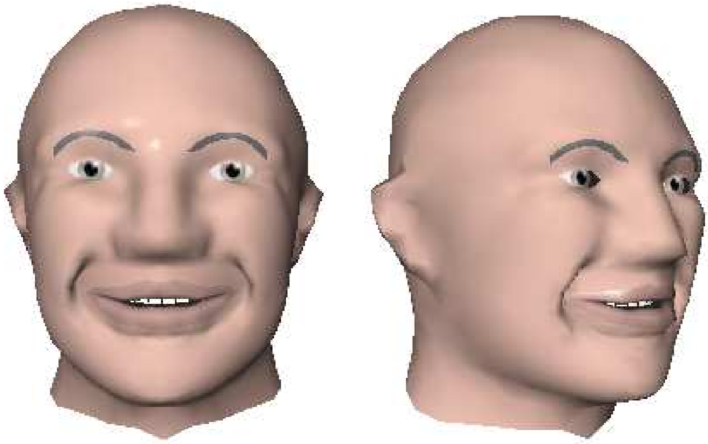


Figure 2.26: The face model displays happiness.

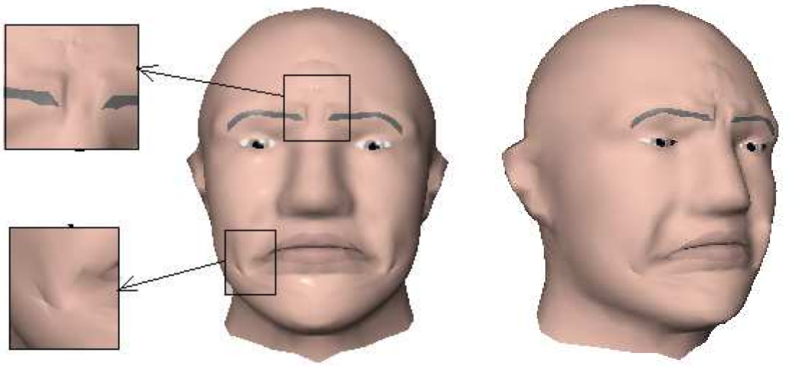


Figure 2.27: The face model displays sadness.

face mesh and a muscle model. The face model allows high quality and realistic facial expressions. In addition, the face model is sufficiently simple in order to keep the animation real-time. The muscle model is an extension of Waters' muscle model. It produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time. We have also presented the implementation of the realistic **Orbicularis Oris** (muscle of the mouth), **Orbicularis Oculi** (muscle of the eyes) and the jaw rotation. By using techniques to improve the performance of the muscle model and maintaining a certain level of simplicity of the whole face model, we have achieved fast animation on a standard personal computer.

Our approach is easy to apply to other facial meshes as the muscle representation is independent of the face mesh. Our face model can be used as a source face model that can be automatically deformed to represent any newly created face model. With the method we will describe in chapter 3, we do not have to manually process another face model again when we want to produce animation on that face model.

There are two issues that we think will improve our model in the future. First, the face model still does not have a tongue model. The tongue contributes to the articulation of speech. The tongue is visible when the mouth is opened. Besides the shape of the lips, the position of the tongue is a supplement hint for recognizing what is being said. Therefore, a good tongue model would increase the realism of the facial animation. Second, how texture mapping would affect the quality of facial expressions and the speed of animation should be considered.

Chapter 3

Exportation of 3D Facial Expressions

> *"A computer lets you make more mistakes faster than any invention in human history – with the possible exceptions of handguns and tequila."*
>
> *– Mitch Ratcliffe, "Technology Review" (1992)*

3.1 Introduction

Recall from the previous chapter that the aim of 3D facial animation is to produce facial expressions on a 3D face model. There are four main techniques to achieve this goal. They are key-frame animation (e.g. (Parke, 1972; Waters and Levergood, 1993)), direct parameterized animation (e.g. (Parke, 1974)), pseudo-muscle-based animation (e.g. (Kalra et al., 1992)), and muscle-based animation (e.g. (Waters, 1987; Terzopoulos and Waters, 1990)). One limitation that these techniques have in common is that they cannot reuse available data for animating new face models themselves. Each time a new face model is created for animation, either the animation must be produced from scratch or a process for manually tuning parameters is required. In key-frame animation, key frames have to be recreated for every new face model. In direct parameterized animation, the parameters are manually hard-wired into a group of vertices to produce local deformations. This manual process must be replicated for new face models with different mesh structures. To animate a face model with pseudo-muscle-based techniques, some form of deformation must be manually positioned within the face mesh to imitate the effect created by a muscle. Every new face model is different in size and structure, which requires the new positioning of pseudo-muscle deformation. The same situation happens with Waters' vector muscle-based approach (Waters, 1987), where the muscles have to be placed manually under the surface of the face model. Multi-layer muscle-based animation requires the creation of underlying structures, such as the skull and the muscle layer. For every new face model, these underlying structures have to be recreated. Therefore, muscle contraction values are reusable only when much manual effort has been put into new face models.

The research question studied in this chapter is how to transfer the animation from a given face model to a newly created one. Several approaches have been proposed for this purpose. First, the “expression cloning” by Noh and Neumann (2001) transfers the motion vectors between two face models using the specification of corresponding landmarks on the two models. Second, the work of Mani and Ostermann (2001) transfers the Face Animation Table (FAT) from an MPEG-4 face model to another one. Third, the work of Kähler et al. (2002) transfers the face mesh, the muscles and the skull mesh of a multi-layer muscle-based face model to a new face model. All these approaches, however, require heavy human involvement to specify and adjust the correspondences between the source and the target face model.

In this chapter, we present a method of transferring facial animation without much human involvement. We use Radial Basis Function (RBF) networks (Broomhead and Lowe, 1988) to deform a source face model to represent a target face model using the specification of corresponding landmarks on the two face models. The landmarks on the source face model are manually specified once and are reused for every target face model. We introduce a novel method to specify and adjust landmarks on the target face model automatically. The adjustment process is done by Genetic Algorithms (GAs). The fitness function used in the GAs expresses the difference between the surface of the deformed face model and the target face model. We also present an algorithm to calculate this function fast. After all the landmarks have been placed in optimal positions, the RBF networks are used to deform the source face model as well as to transfer the muscles on the source face model to the deformed face model.

There are several advantages to the use of a deformed face model to represent a new face model over that of the new one itself. First, we can keep a fixed face model with fixed number of vertices and polygons when achieving facial animation for different face models. Second, tags on the fixed face model can be reused, for example tags for the jaw rotation and eyelid rotation. Third, the regions information on the face model, described in previous chapter, can be reused to improve the performance of the vector muscle and to control the animation.

Section 3.2 reviews the existing work in transferring facial animation. Section 3.3 gives an overview of our approach, and describes how corresponding landmarks are used to deform a source face model to represent a target face model and how these landmarks are adjusted to reduce the difference between the deformed and the target face model. The RBF networks we use to deform the face model are presented in Section 3.4. To be able to minimize the difference between the deformed and the target face model, we need a computational error function to assess this difference. We sample the deformed face model resulting in a number of sampling points. The sum of the squares of the distances between these sampling points and their projection onto the target face model is taken as the error function. The sampling method is described in Section 3.5. In that section, we also show how to find the projection of the sampling points fast. Once a method for calculating the difference between two face models has been specified, a Genetic Algorithms process is performed to adjust the landmarks in order to minimize the error function. This process is discussed in detail in Section 3.6. Finally, some experimental results are presented in Section 3.7.

3.2 Previous work

In this section we discuss three existing approaches for transferring animation. They are: the "expression cloning" by Noh and Neumann (2001), which transfers the motion vectors between two face models using the specification of corresponding landmarks on the two models; the work of Mani and Ostermann (2001), which transfers the Face Animation Table (FAT) from an MPEG-4 face model to another one; and the work of Kähler et al. (2002), which transfers the face mesh, the muscles and the skull mesh of a multi-layer muscle-based face model to a new face model.

3.2.1 Expression cloning

Noh and Neumann (2001) proposed a method, which is called "expression cloning", to transfer animation between face models. They reuse deformation data from an existing face model to create animation in a new face model. The deformation data is in the form of vertex motion vectors, which are the displacement vectors of vertices in the face model to create a facial expression. Each expression of the source face model is directly transferred to the target face model using three Radial Basis Function (RBF) networks (Broomhead and Lowe, 1988) followed by a cylindrical projection. The RBF mapping approximately aligns the features of the mapped face, such as eye sockets, nose ridge, lip corners, and chin points, to the target face. The cylindrical projection of the mapped source model onto the target model ensures that all the vertices of the mapped model lie in the target model surface.

The parameters of the three RBF networks are determined by a set of source landmarks and a set of target landmarks. The user has to select these landmarks manually. Some heuristic rules are given to find some of these landmarks automatically. Each RBF network maps a 3D point to a scalar value. In this approach, the three networks are used to map points on source face model to the x, y, and z components of the points on target face model:

$$(x, y, z) \rightarrow (RBF_1(x, y, z), RBF_2(x, y, z), RBF_3(x, y, z))$$

The x, y, and z components of source landmarks and target landmarks are used as the training set for the three networks.

For the cylindrical projection, a vertical line through the centroid of the head is set up as the projection centerline. To cylindrically project a vertex to a surface, a ray perpendicular to the projection centerline is passed through the vertex and intersected with the surface. The first intersection found is used as the projected vertex.

To clone expression, the RBF functions are used to map the vertices of the source face model to the target face mode resulting in a mapped face model. The motion vector of the vertices of the source face model is also mapped. Next, each vertex v of the target face model are projected onto the surface of the mapped face model cylindrically resulting in a vertex v'. Note that because the surface of the mapped face model consists of triangles, v' lies inside some triangle t. Three barycentric coordinates are calculated to determine the relative position of v' with regard to three corners of t. The motion vector of v is determined by the weighted sum, with the barycentric coordinates as coefficients, of the motion vector of these three corners. Next, the magnitude and direction of the motion vector of v are adjusted with regard to the local shape of the target face model. Finally, the motion vector of all vertices of the target face model are applied resulting in the desired facial expression.

"Expression cloning" allows animation created by any animation technique to be re-targeted to new face models. However, it requires that each facial expression has to be created on the source face model before the expression can be cloned on the new face model. Therefore, the cloning process has to be run for every frame of the animation. In addition to the cost to generate each facial expression on the source face model, this cloning process prevents the animation from being in real-time. Moreover, "expression cloning" needs heavy human involvement to specify and adjust the correspondences between the source and the target face model. Although this involvement is reduced by a set of heuristic rules to find some corresponding points automatically, not all the heuristic rules can be applied easily. Some rules are correct for almost every face model. For example, the tip of the nose is the vertex with the highest z value[1]; the top of the head is the vertex with the highest y value; and so on. Some rules can only applied to well behaved face models. For example, the top of the nose is the vertex with the local minimum z-value while searching upward from the tip of the nose, along the ridge of the nose.

3.2.2 Transfer of MPEG-4 Face animation tables

Mani and Ostermann (2001) use a weighted sum of B-splines as a mapping function to clone MPEG-4 Face Animation Tables (FATs) from a source face model to a target face model. B-splines are defined as functions, particularly as cubic polynomials over some specified interval. The weights are calculated based on the value of corresponding landmarks on the source and target face model.

An MPEG-4 face model is animated with a set of facial animation parameters (FAPs). Each FAP is associated with a set of predefined vertices. A FAT contains maximal displacements of these vertices. The FAP and FAT together are used

---

[1] The face model is oriented to look in the positive z-direction; the y-axis goes through the top of the head; and the x-axis goes through the right ear.

to determine the movement of these vertices, which results in a desired facial movement.

The cloning process of FATs is performed as follows. First, the user has to select corresponding landmarks between source and target face models manually. Using these corresponding landmarks, vertices of the target face model are mapped to the source face model resulting in a mapped face model. Then each FAT is applied on the vertices of the mapped face model. The result is inversely mapped to the target face model to create the deformed target face model. The difference between the vertices of the deformed and the original target face model is used to compute the FAT of the target face model.

The quality of the transferred animation completely depends on how many corresponding points the user selects and how the user selects these points. A manual process is also required to separate the vertices on the upper and lower lips to prevent mismanaging them. Moreover, additional manual adjustment of B-spline's weights has to be done to increase the correctness in mapping the MPEG-4 FATs; however this correctness is still not warranted.

3.2.3 Transfer of a multi-layer facial structure

Kähler et al. (2001) proposed a multi-layer face model which contains a face mesh, a muscle layer and a skull mesh. Muscles are created by first manually sketching the basic muscle grids row by row. The muscle grids are then refined automatically to fit the geometry. Quadric shaped fibers are inserted to the grids to create the muscles. The skull mesh is obtained by offsetting each vertex of the face mesh along the negated normal vectors of the vertex.

Following Noh and Neumann (2001), Kähler et al. (2002) use RBF networks (Broomhead and Lowe, 1988) to deform the face mesh, the underlying muscle, and the skull mesh to represent a new face. After manually selecting and adjusting the position of corresponding landmarks on the source face mesh and the target face mesh, the source face mesh is deformed by directly applying the RBF networks to the vertices of the mesh. The landmarks for the skull mesh are obtained by offsetting the landmarks for the face mesh along the negated normal vectors at the landmarks. The RBF networks with the landmarks for the skull mesh are applied to the vertices of the skull mesh. To transfer the muscles to the new face model, the RBF networks with landmarks for the face mesh are applied to the basic muscle grids. These grids are used to compute the muscles for the new face model.

Again, the approach by Kähler et al. (2002) requires the manual specification and adjustment of the landmarks on the target geometry. To ease this "repeated point-and-click" task, Kähler et al. (2002) presented a computer-aided process. This process starts with several manually specified landmarks, which are fixed. Next it approximates the location of some other landmarks based on fixed ones. The user can adjust these landmarks and fix some more landmarks. The process is repeated several times until all the landmarks are specified and placed in the desired positions. The process of specifying 60 landmarks takes about 10-20 minutes in practice.

3.3 Overview of our approach

Like (Kähler et al., 2002), we deform a source face model to represent a target face model using RBF networks (Broomhead and Lowe, 1988), which are trained by the value of corresponding landmarks on the two face models. The landmarks on the source face are fixed, while we search for the landmarks on the target face to minimize the difference between the deformed face model and the target face model. A schematic of our approach can be seen in Figure 3.1.

We use a muscle-based source face, which is presented in the previous chapter. The muscles in the source face model are manually placed in a careful way to create realistic facial expressions. The source face model is also preprocessed to speed up the animation. Once the source face has been deformed to represent a new face model and the muscles have been transferred to the deformed face model, the data of muscle contraction levels can be reused to create facial animation on the deformed face model.

A set of landmarks is predetermined on the source face model. Most landmarks are specified manually except several that can be detected automatically to cover all the features of the face. These landmarks are determined only once and are reused for every new target face model. For the landmarks on the target face model, we first determine several easy-to-detect landmarks, namely the top of the head, the tip of the nose, and so forth. The auto-detected landmarks will stay fixed whereas the rest of the landmarks will be adjusted; for convenience's sake let us call these "non-detected landmarks". We use the auto-detected landmarks and their correspondences on the source face model as the training set for the RBF networks to determine the initial version of the non-detected landmarks on the target face model. A deformed version of the source face model is created by RBF networks using the landmarks on the source and the target face model. The non-detected landmarks on the target face model are then adjusted with Genetic Algorithms to minimize the difference between the deformed and the target face model.

3.4 Radial Basis Functions Networks

RBF networks (Broomhead and Lowe, 1988) are usually applied to approximate a curve or a surface (Bishop, 1995). They can also be used for face model fitting (Ulgen, 1997; Pighin et al., 1998; Enciso et al., 2000; Noh and Neumann, 2001; Kähler et al., 2002). Ulgen (1997) started to exploit RBF networks for face model fitting. Pighin et al. (1998) and Enciso et al. (2000) then used RBF networks to fit a 3D face model to one (or several) 2D image(s). Noh and Neumann (2001) and Kähler et al. (2002) employed RBF networks to map a 3D face model to another one, resulting in transferring facial animation.

A RBF network consists of a weighted linear combination of a number of basic functions. To map a 3D face model to another one, the basic functions and the weights are determined by landmarks on the two face models. Each basic function is a function of the distance from a vertex to one of the source

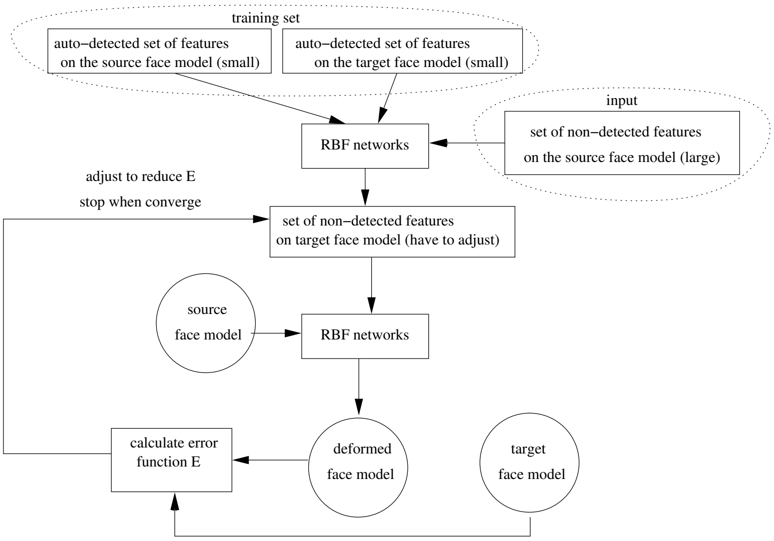


Figure 3.1: Overview of the face exportation system.

landmarks.

We follow Noh and Neumann (2001) to use three RBF networks, which map a vertex on source face model to the x, y, and z components of a vertex on target face model:

$$(x, y, z) \longmapsto (RBF_1(x, y, z), RBF_2(x, y, z), RBF_3(x, y, z))$$

Each RBF_i is given by

$$RBF_i(x, y, z) = \sum_{j=1}^{n} w_{i,j} h_{i,j}(x, y, z)$$

where the $w_{i,j}$ are the weights of the network that need to be determined or learned on the basis of a training set. The x, y, and z components of source landmark points and target landmark points are used as the training sets for the three networks. For the basis functions $h_{i,j}$ we follow the successful approach given by Noh and Neumann (2001):

$$h_{i,j}(v) = \sqrt{\| \; v - \mu_{i,j} \; \|^2 + s_{i,j}^2} \quad , \quad v = (x, y, z)$$

where $\mu_{i,j}$ is called the center of $h_{i,j}$ and $s_{i,j}$ is given by

$$s_{i,j} = min_{k \neq j} \; \| \; \mu_{i,k} - \mu_{i,j} \; \|$$

This choice for $s_{i,j}$, as suggested by Eck (1991), leads to smaller deformations for widely scattered center points and larger deformations for closely located points.

In order to prevent overfitting and improve generalization, a regularization term $\sum_j w_{i,j}^2$ is added to the error term for each RBF_i, cf. (Bishop, 1995).

3.5 The Error Function

The error function assesses the differences between the deformed source face model and the target face model by calculating the distance between them at sampling points. A set of sampling points is first specified automatically on the deformed face model. These sampling points are evenly distributed over the deformed face model. Using the cylindrical projection described in Section 3.2, these sampling points are projected onto the target face model. The distance between the sampling points to their projection is used to determine the difference between the deformed and the target face model. Because there are no facial expressions on the back of the head, we ignore the back of the head to concentrate on the front part of the head. We also ignore the neck when taking the sampling points in order to avoid any unnecessary mis-fitting between a part of a head and another head's neck.

Note that the calculation of the cylindrical projection of a point onto a surface represented by polygons is an expensive process. The reason is that for every polygon of the surface, it has to determine whether the ray, which goes through the point and is perpendicular to the projection centerline, intersects the polygon. Thus, calculating the projection of every sampling point would slow down the search by the Genetic Algorithms which we mentioned before. While other methods to determine the sampling points can be used, the method that we will present here allows the calculation of the projection of sampling points onto the target face model at no cost.

For simplicity, we place all the face models in the same coordinate system, as can be seen in Figure 3.2. The face model is oriented to look in the positive z-direction; the y-axis goes through the top of the head; and the x-axis goes through the right ear.

Let V_{th} denote the top head vertex of the deformed face model. Because RBF networks map landmarks to corresponding ones in a relatively accurate manner and the top head vertex is one of the landmarks, the top head vertex of the deformed face model should relatively match the top head vertex of the target face model. Let l denote the projection centerline that goes through V_{th} and is parallel with the y-axis.

First, we take m sampling planes P_i $(i = 0..m-1)$ which go through l. Each P_i is determined by the angle α between itself and the plane YOZ:

$$\alpha = \frac{i}{m}\pi \quad \text{with} \quad i = 0..m-1.$$

Each plane P_i intersects the surface of the deformed and the target face model at a number of edges as illustrated in Figure 3.3. By filling all the holes, such as the eye and mouth holes, on the deformed and the target face model, these edges form two continuous curves, which can be seen in Figure 3.4:

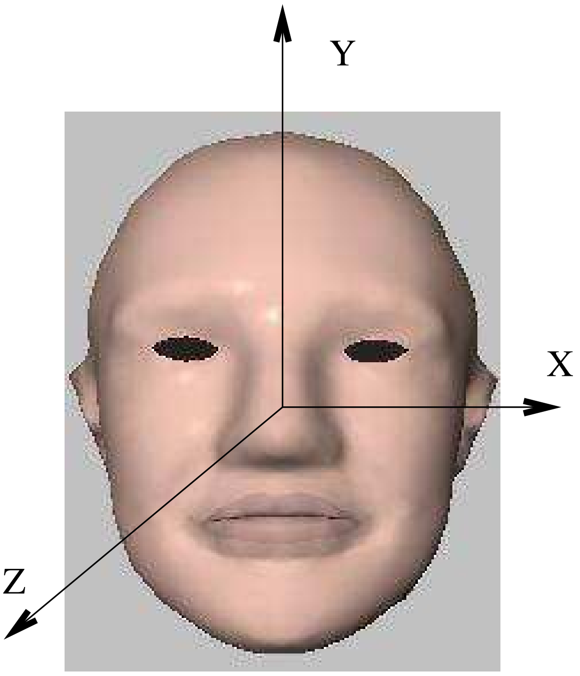


Figure 3.2: The source face model (4650 polygons).

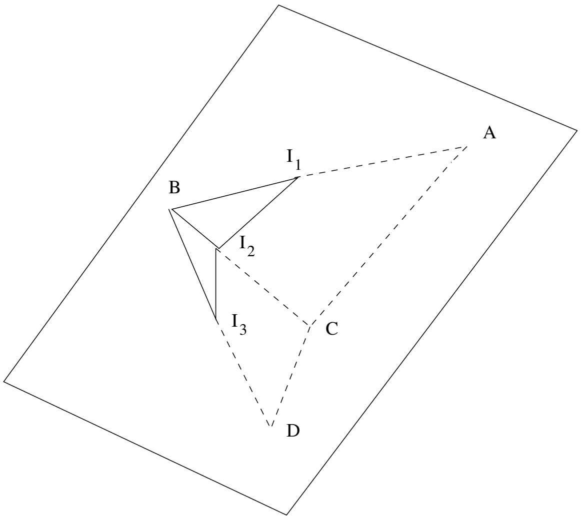


Figure 3.3: Intersections between a face model's surface and a sampling plane.

$$em_{i,1}, em_{i,2}, ..., em_{i,NM_i}$$

and

$$et_{i,1}, et_{i,2}, ..., et_{i,NT_i}$$

Note that both the two curves lie on the same plane P_i. We chop off these curves to have their starting vertices and ending vertices having the same y-coordinate $Y_{starting}$ and Y_{ending}. We give a big penalty to the part of the curves that are outside this range and add this penalty to the error function. We then take n sampling vertices SM_{ij} and ST_{ij} ($j = 0..n-1$) on each of these two curves. SM_{ij} and ST_{ij} have the same y-coordinate, which is determined by:

$$y_{starting} + \frac{j \cdot (y_{ending} - y_{starting})}{n-1}$$

To find a sampling point SM_{ij}, we just have to go through all the vertices $et_1, et_2, ..., et_{NT}$ and do a simple search for pair (et_k, et_{k+1}) that the y-coordinate of SM_{ij} is in the interval of the y-coordinates of et_k and et_{k+1}. It is similar to find a sampling point ST_{ij}.

Because SM_{ij} and ST_{ij} have the same y-coordinate, the line which goes through SM_{ij} and ST_{ij} is perpendicular to the projection centerline l. Therefore, ST_{ij} is the cylindrical projection of SM_{ij} onto the target face model. Thus, after the sampling process is done, the projection of sampling points is immediately determined at no cost. The complexity of the sampling method we present above lies on the determination of the intersection curve between a plane P_i and a face mesh. The calculation for the intersection curve can be painfully slow if we just simply check whether the plane intersects every polygon of the face mesh. We overcome this problem by traversing from the top head vertex down each face mesh through the polygonal structure of the mesh. Starting from a point with highest y-coordinate on the curve, we only check the related polygons to find the next point on the curve. We continue this until we find no more points on the curve.

As the top head vertex lies on the plane P_i, it is the first vertex on the curve. Let us call it the "current edge" on the face mesh that intersects P_i (a point is a special case of an edge, i.e., zero length edge). Next, we only check the intersection between the plane P_i with edges that are connected to the "current edge". The new edge that intersects P_i becomes the new "current edge". We mark all the found edges to prevent from going back to edges that we have already visited. This process stops when P_i does not intersect any more edges that are unmarked and connected to the "current edge".

Once all the sampling points and their projections onto the target face model have been specified, the error function is computed by:

$$E = \sum_{i=1}^{m} \sum_{j=1}^{n} \parallel SM_{ij} - ST_{ij} \parallel^2 + \Delta$$

where Δ is the sum of the penalties we described above when we chop off the intersection curves.

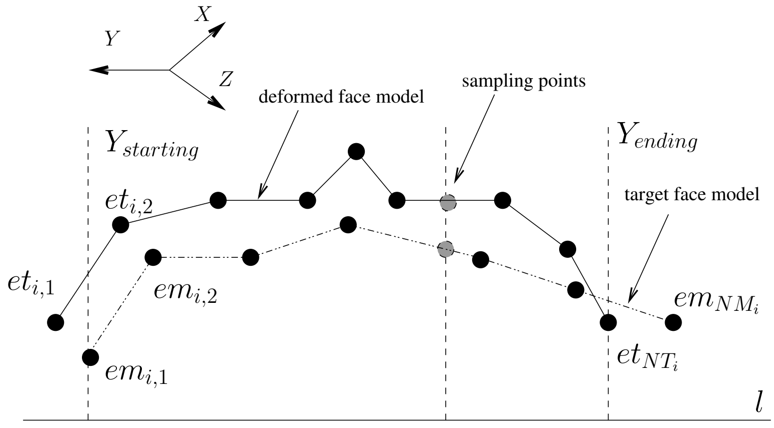


Figure 3.4: How sampling points are located on the intersection between a sampling plane and the surface of the target and the deformed face model.

3.6 Adjusting the landmarks

Recall from Section 3.3 that several landmarks on the target face model can be detected automatically. The rest of the landmarks are "non-detected" ones. We use Genetic Algorithms to adjust the non-detected landmarks on the target face model to minimize the differences between the deformed and the target face model. The Genetic Algorithms search for the optimal variant of the landmarks to reduce the error function described in the previous section.

3.6.1 Genetic Algorithms

Genetic Algorithms (GAs) are search algorithms based on the process of natural evolution and survival of the fittest in the biological world (Goldberg, 1989). While traditional optimization techniques search for an optimal solution from a single point, GAs search from a population of solutions. A GA problem solving approach is, basically, to rate or to rank possible solutions as they are generated, and then use these rankings to guide further exploration for related solutions. This exploration is done through a number of iterations. During each iteration of as GA, a competitive selection is performed to get rid of incompetent solutions. The solutions with high fitness get more chance to be selected to recombine with each other in order to form new promising solutions. Solutions are also mutated by applying random small changes. The way of search in GAs explores the search space biased toward the regions that contain visited good solutions. Pseudo-code for a genetic algorithm looks like:

```
Initialize the population
Evaluate initial population
Repeat
  Perform competitive selection
```

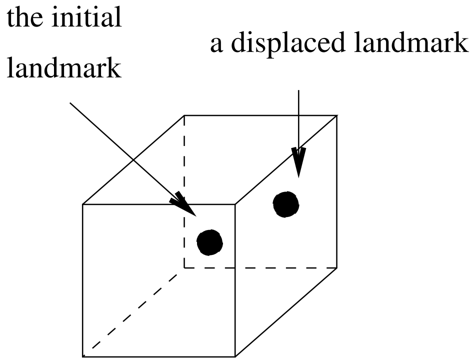


Figure 3.5: An initial landmark point and a variant of it.

```
    Apply genetic operators to generate new solutions
    Evaluate solutions in the population
Until some convergence criteria is satisfied
```

For a more background and a thorough treatment of Genetic Algorithms, the readers may consult (Goldberg, 1989) and (Davis, 1991).

3.6.2 Applying GAs to adjust the landmarks

The GAs process starts with a random set of solutions, which are represented as chromosomes. Because we want to constrain the search to be only around the initial landmarks, each solution of the GAs process is a variant of the landmark points on the target face model, which is the modification of the initial landmark points. In this modification, each variant of the landmark point is any point inside the cube with specified length and the initial landmark point as the center (see Figure 3.5). Solutions from one population are taken and used to form a new population. Then the new solutions are selected according to their fitness. The more suitable they are, the better chances they have to reproduce. In this case, the fitness function is the inverse of the error function that minimizes the difference between the deformed and the target face model.

The Chromosome A chromosome is the concatenation of binary representations of all non-detected landmarks on the target face model. Each landmark is represented in the chromosome by the distance between this landmark and the original landmark.

Each landmark can be described by three coordinates v_1, v_2, v_3. Starting from an initial version of the landmark, various versions of the landmark can be obtained by modifying each coordinate in a specified range. Let $Rmin_1$, $Rmax_1$, $Rmin_2$, $Rmax_2$, $Rmin_3$, $Rmax_3$ be the ranges of value for these coordinates. The coordinate of a landmark can be represented as:

$$(p_1, p_2, p_3)$$

where $0 \leq p_i \leq 1$,

$$p_i = \frac{v_i - Rmin_i}{Rmax_i - Rmin_i} \quad i = 1..3$$

and, consequently, we have

$$v_i = p_i(Rmax_i - Rmin_i) + Rmin_i.$$

We then convert p_i to a binary string. Using 2^n as upper limit[2], we represent p_i as $\lfloor 2^n p_i \rfloor$ in its binary format:

$$c_{i1}, c_{i2}, ..., c_{in} \qquad c_{ij} = 0 \text{ or } 1.$$

We concatenate the binary representation of the three coordinates of all landmark points to form a chromosome.

The fitness function The fitness function is the inverse of the error function described in Section 3.5:

$$\text{fitness(solution)} = \frac{1}{\text{E(deformed\_face(solution),target\_face)}}$$

where deformed_face(solution) is the deformed face model using the solution as the landmarks on the target face model.

Crossover We use multi-point crossover, which is illustrated in Figure 3.6. For multi-point crossover, several crossover positions are chosen at random with no duplicates and sorted in ascending order. Then, the chromosome elements between successive crossover points are exchanged between the two parents to produce two new offsprings. The section between the first element and the first crossover point is not exchanged between individuals. The idea behind multi-point, and indeed many of the variations on the crossover operator, is that parts of the chromosome representation that contribute the most to the performance of a particular individual may not necessarily be contained in adjacent substrings (Booker, 1987). Moreover, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals early in the search; thus making the search more robust (Spears and DeJong, 1991).

Mutation Solutions are mutated by applying random flipping to the component of the chromosomes (0 to 1 and 1 to 0). We start with a mutation rate

---

[2] By trial and error, n ranging between 5 to 10 gives best convergence and best result for the GA.

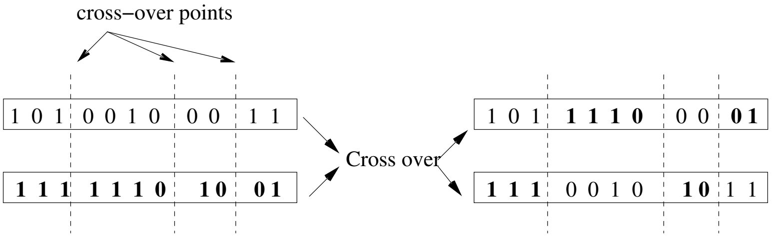


Figure 3.6: Multi-point crossover.

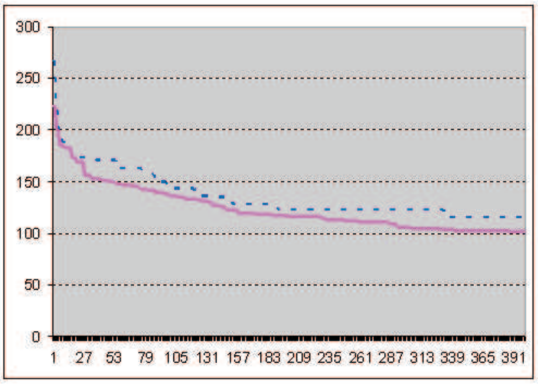


Figure 3.7: How the GA process converges: broken line – solutions are replaced by their projection onto the target face model during the GA process; continuous line – projection is used only to calculate the error function and to generate the final solution.

of 0.3. We increase this mutation rate when the error stays stable, and decrease it when the GA process produces smaller errors (better results). This mutation rate is constrained to be in the range of 0.3 to 0.5.

Additional Operation We have introduced an additional operation to the GA process. This operation projects each solution of the GAs to the target face model's surface using the cylindrical projection as described in Section 3.2. We have tested two approaches to implement this operation. The first approach is to replace each solution of the GA process with its projection. How the GA converges in that case is shown by the broken line in Figure 3.7. The second approach is to use the projection of a solution only to calculate the error function and to generate the final solution. How the GA converges in this case is shown by the continuous line in Figure 3.7. As can be seen from this figure, the first approach causes the error to decrease and the process to converge very fast, whereas the second approach causes the error to decrease and the process to converge slowly. However, the first approach seems to end up at some local minimum and cannot get out of this local minimum. This happens due to the

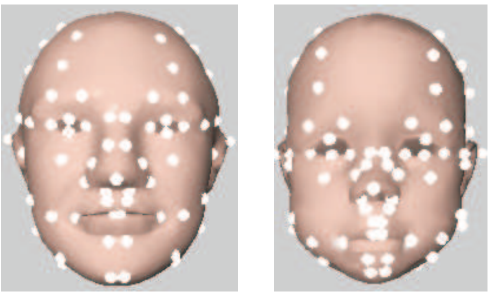


Figure 3.8: The landmarks on the source face model (left) and the target face model(right).

decrease of diversity of GA's solutions by replacing the solutions with their projections.

3.7 Experimental result

Figure 3.8 shows the landmarks on the source face model and the adjusted landmark points on the target face model. As can be seen from this figure, the landmarks on the target face model are adjusted to the right position. The deformed face model generated by RBF networks using these landmarks is shown in Figure 3.9. The deformed face model has the overall shape, forehead and cheek surface, chin shape as the target face model. Eyes, nose and mouth are in correct position. The shape of the deformed face model's lips, however, does not completely match the shape of the target face model's lip. This is because the difference between two pairs of lips is hard to measure even if they look very different. Another example of the exportation's result is shown in Figure 3.10.

After the deformed face model is created, the muscles are also transferred from the source face model to the deformed face model. Using these muscles, we can create facial expressions on the deformed face model, which are shown in Figure 3.11. The muscles, however, are not in the perfect positions in the the deformed face model. Some more fine tuning is still needed.

3.8 Conclusion

In this chapter, we introduced a novel method of automatically finding the training set of RBF networks to deform a source face model to represent a new face model. This was done by specifying and adjusting corresponding landmarks on a target face model automatically. The RBF networks were then used to transfer the muscles on the source face model to the deformed face model. Genetic Algorithms were used to adjust the landmarks on the target face model to minimize the difference between the surface of the deformed and the target

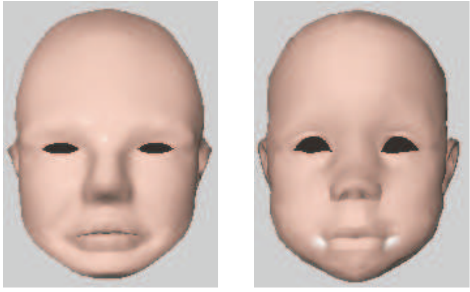


Figure 3.9: The deformed face model (4650 polygons) (left) and the target face model (4142 polygons) (right).

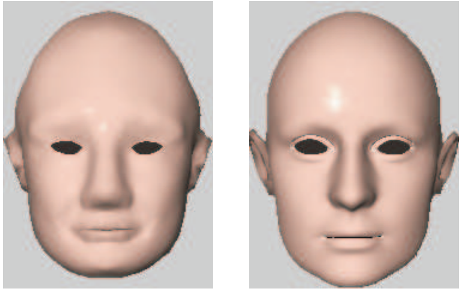


Figure 3.10: Another example of the deformed face model (4650 polygons) (left) and the target face model (7736 polygons) (right).

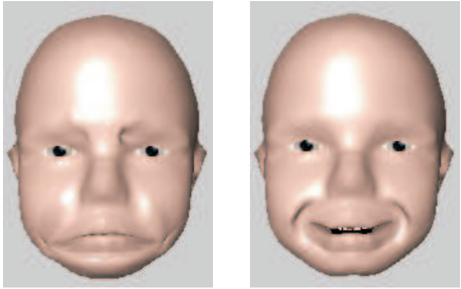


Figure 3.11: Facial expressions on the deformed face model: sadness (left) and happiness (right).

face model. We defined a fitness function to assess the difference between the two models. We also presented an algorithm to calculate this function fast.

Chapter 4

Combination of facial movements on a 3D talking head

"Self-expression must pass into communication for its fulfillment."

– Pearl S. Buck

4.1 Introduction

Facial movements play an important role in interpreting spoken conversations and emotions. They occur continuously during social interactions and, particularly, in conversations. They include lip movements when talking, conversational signals, emotion displays and manipulators to satisfy biological needs. Unfortunately, when and how a movement appears and disappears, and how co-occurrent movements are integrated (co-articulation effects, for instance) are difficult to quantify (Ekman et al., 1993). In addition, the problems of overlaying and blending facial movements in time, and the way felt emotions are expressed in facial activity during speech, have not been studied thoroughly (Latta et al., 2002).

Recall from Chapter 2 that facial animation has received quite a lot of attention in the field of embodied agents. Realistic animation of faces would improve the realism and lifelikeness of the interaction between humans and machines. To create realistic facial animation, many 3D face models have been proposed; cf. Chapter 2. Recently, more and more attention has been paid to facial animation in synchronization with synthesized/natural speech. Many talking faces have been developed. Examples include (Albrecht et al., 2002a), (DeCarlo et al., 2002), (King et al., 2000) and (Pelachaud et al., 1993). These systems combine facial movements by just adding them together without taking into account the resolution of conflicting muscles/parameters temporally. Specifically, significant attention has been paid to visual speech (Cohen and Massaro, 1993; King et al., 2000). Some systems are also able to generate facial expressions as conversational signals during speech (Albrecht et al., 2002b; Pelechaud et al., 1996). However, no appropriate methods have been proposed for integrating all these facial movements.

The activity of human facial muscles is far from simply additive. A typical example would be smiling while speaking. The Zygomatic Major and Minor muscles contract to pull the corner of lip outward, resulting in a smile. The viseme corresponding to the phoneme “@U” in the word “Hello” requires the contraction of the lip funneler Orbicularis Oris, which drives the lips into a tight, pursed shape. However, the activation of the Zygomatic Major and Minor muscles together with the lip funneler Orbicularis Oris would create an unnatural movement. We call these “conflicting” muscles. The activation of a muscle may require the deactivation of other muscles (Ekman and Friesen, 1978). Depending on the priority of the tasks to be performed on the face, appropriate muscles are selected for activation. In most of the cases, the visual speech has higher priority than the smile. The smile may also have higher priority than the visual speech when the subject is too happy to utter the speech naturally.

In this chapter, we propose a scheme of combining facial movements on a 3D talking head. There are several types of movements, such as conversational signals, emotion display, etc. We call these channels of facial movement. We concentrate on the dynamic aspects of facial movements and the combination of facial expressions in different channels that are responsible for different tasks. First, we concatenate the movements in the same channel to generate smooth transitions between adjacent movements. This combination only applies to individual muscles. The movements from all channels are then combined taking into account the resolution of possible conflicting muscles.

Section 4.2 gives an overview of the system. We break facial movements into so-called “atomic” movements. Each “atomic” movement belongs to a specific channel. There are six channels in our system which contain: manipulators, lip movements when talking, conversational signals, emotion displays and emotion emblems, gaze movements, and head movements. We use the 3D face model described in Chapter 2 for the talking head. The concept of “conflicting muscles” on the face is discussed in Section 4.3. In this section, we also present a summary of all conflicting muscles. Atomic movements are described in Section 4.4. This section shows how we define the temporal pattern of an atomic movement. Section 4.5 explains how facial movements inside a channel are combined while Section 4.6 is devoted to the combination of movements from all channels.

4.2 System overview

Our talking face takes as input the text to be pronounced marked up with “atomic” facial movements other than lip movements when talking. We define an “atomic” movement as a group of muscle contractions that share the same function (e.g., conversation signal, emotion display), start time, end time, and onset and offset duration. A simple example of marked-up text looks like this

<PHONEME time="0" text="Oh really? I like it very much." />
<EMOTIONDISPLAY time="0" duration="2.0" Surprise="0.7" />

Each "atomic" facial movement belongs to a specific channel, which contains only non-conflicting movements. Atomic movements within a channel occur sequentially, although they may overlap each other at their beginning and ending. This classification is also based on the function of the movements (Ekman, 1989). It is similar to Pelachaud et al.'s one (Pelachaud et al., 1993). Movements from different channels can happen in parallel and can involve conflicting muscles.

In our system, we distinguish six channels:

- Channel 1 contains **manipulators**, which are movements to satisfy biological requirements of the face. In our system, we consider eye blinking to wet the eyes as manipulators. These movements are random rather than repeated with fixed rate as in (Pelechaud et al., 1996). The random eye blinking is generated based on the algorithm proposed in (Itti et al., 2003).

- Channel 2 contains **lip movements** when talking (represented as viseme segments). Lip movements are generated from the text that is going to be spoken by the talking head. The text is converted to phoneme segments (phoneme with temporal information – starting and ending time) (Smart, 2000). The phonemes are converted to corresponding visemes. Each viseme is equipped with a set of dominance functions of parameters participating in the articulation of the speech segment. We use dominance functions from (DeCarlo et al., 2002) for each viseme segment.

- Channel 3 contains **conversational signals**. These are movements to accentuate or emphasize speech, or to provide feedback from a listener. They can occur on pauses due to hesitation or to signal punctuation marks (such as a comma or an exclamation mark). They are used to improve the interaction between the speaker and the listener. The generation of conversational signals can be done by analyzing the text (Pelechaud et al., 1996) or speech (Albrecht et al., 2002b).

- Channel 4 contains **emotion displays**, which are **emotional expressions** or **emotion emblems**. **Emotional expressions** are movements to express felt emotions of the speaker. On the other hands, **emotion emblems** express emotions that are being mentioned, for instance, a disgust expression when talking about something disgusting. We have proposed a fuzzy rule-based system to generate emotion displays from emotions; see Chapter 6.

- Channel 5 contains **gaze movements** and Channel 6 contains **head movements**. Gaze and head movements are generated to support eye contact or to point to something during conversation. Head movements are also used to replace verbal content (e.g., nodding the head for saying yes). As the eyes and the head do not stay in the same place all the time, we use a noise generating function to create random subtle movements to make the talking head more lively.

An overview of our system can be seen in Figure 4.1. From text input, the text to phoneme module (Smart, 2000) generates phoneme sequences, which are used to synthesize speech (Dutoit et al., 1996). They are also used to generate lip movements when talking.

The system also takes other facial movements as input. Facial movements are then combined in two stages: internal channel combination and cross channel combination. The former concatenates movements in the same channel and creates smooth transitions between them. The latter combines the movements from all channels taking into account the muscle conflicting resolution. The result is displayed on a 3D face model to create the final animation in synchronization with the synthesized speech.

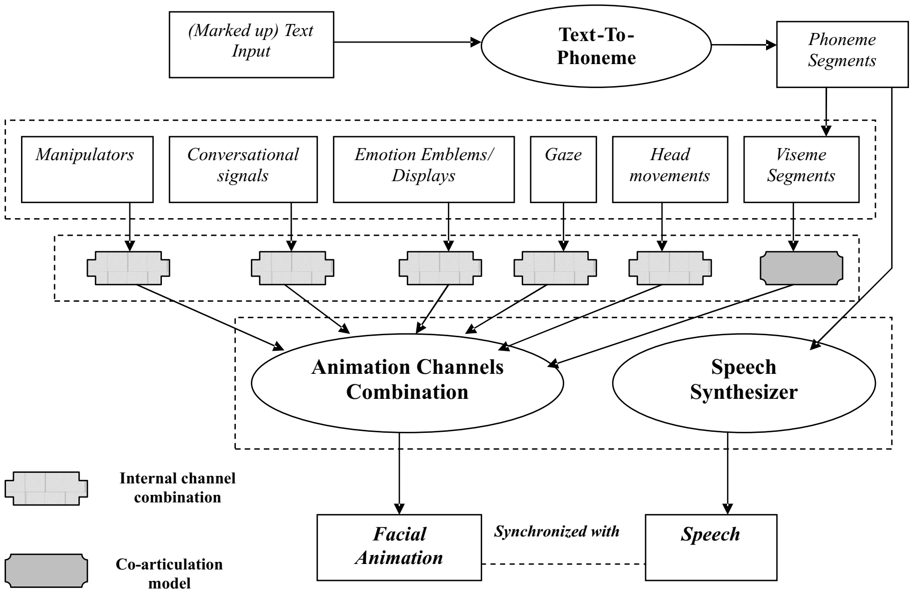


Figure 4.1: Overview of the talking head system.

4.3 The 3D face model

Our 3D face model is described in Chapter 2 in some detail. It is a simple muscle-based 3D face model that can realize both of the following objectives: producing realistic facial expressions and real-time animation on a standard personal computer. The face model, which is sufficiently simple in order to keep the animation real-time, allows high quality and realistic facial expressions. The face is equipped with a muscle system that produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time.

In our face model, there are 35 muscles that are responsible for visual speech (lip movements) and facial expressions. The muscles are shown in Table 2.1. Note that for every muscle in that table, there is a left one and a right one except for the two Orbicularis Oris muscles and the Masseter. We also have parameters for eye and head movements.

We have paid attention to the combination of multiple muscle contractions (cf. Chapter 2). However, in the face, not all combinations are possible. For example, the lip protrusion muscles Levator Labii Superioris, which pull the upper lip apart from the lower lip, and the lip pressor muscle Orbicularis Oris, which presses the lips together, cannot contract together. The simple additive combination of their contraction would result in an unnatural movement, which can be seen in Figure 4.2. Thus, some muscle actions require the deactivation of other muscles. Ekman and Friesen (1978) have discussed possible conflicting movements on the face in term of Action Units (summarized in Wojdel et al., 2003). Based on this information, we have generated a list of all conflicting pairs of muscles in Table 4.1. As can be seen from this table, most pairs of conflicting muscles involve the mouth region. Other ones include the opening and closing of the eyelids, the Frontalis Medialis and the nose wrinkling muscles.

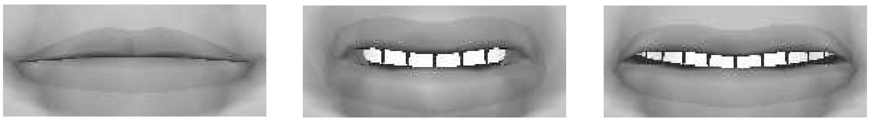


Figure 4.2: The addition of two conflicting muscles: the lip pressor (left) and the lip protrusion (middle) results in an unnatural movement (right).

For convenience, when referring both to facial muscle and parameters for eye and head movements, from now on we will use the same term “parameter”.

4.4 Atomic facial movements

Although facial movements happen continuously, most of them are known from electromyography (EMG) studies to occur in distinct phases (Essa, 1994). The flow of movements can then be broken up into so-called “atomic” movements. Kalra et al. (Kalra, 1993) break each movement into four phases: attack (onset), decay, sustain (apex), and release (offset). We follow Pelechaud et al. (1996) to synthesize facial movements in three phases: onset, apex, and offset.

Basically, each facial movement in our system is defined as a triple:

	01	02	03	04	05	06	07	08	09	11	12	16	17
01			x	x		x	x						
02			x	x		x	x						
03	x	x					x						
04	x	x					x						
05						x		x					
06	x	x			x								
07	x	x	x	x				x					
08					x		x			x	x		
09										x			
11								x	x			x	
12								x					
16										x			x
17												x	

Table 4.1: Conflicting muscles in the system (muscles 10, 13, 14, 15, 18, 19 are not included because they do not conflict with any other muscle).

$$\Gamma^m = ((PV_i^m), Ts^m, Te^m)$$

where the set of parameter values (PV_1^m,PV_2^m,..,PV_n^m) defines the target state of the movement (n is the number of deformation parameters in the face); Ts^m and Te^m are the starting and ending time of the movement, respectively.

Lip movements when talking:

The lip movements when talking are also associated with an apex duration Da^m:

$$\Gamma l^m = ((PV_i^m), Ts^m, Te^m, Da^m)$$

The activity of these lip movements is created based on dominance functions, which are used for producing co-articulation effect (Cohen and Massaro, 1993). This will be discussed in Section 4.5.1.

Other movements:

Other movements are associated with two more temporal values: the **onset duration**, Do^m, determines how long the facial movement takes to appear; the **offset duration**, Dr^m, determines how long the facial movement takes to disappear:

$$\Gamma o^m = ((PV_i^m), Ts^m, Te^m, Do^m, Dr^m)$$

The activity of a parameter involved in the creation of a single facial movement is described as a function of time:

$$F_p^m(t) = \begin{cases} PV_p^m \cdot \phi_+(t - Ts^m, Do^m) & \textbf{if} \quad (Ts^m < t < Ts^m + Do^m) \\ PV_p^m & \textbf{if} \quad (Ts^m + Do^m \leq t \\ & \qquad\qquad \leq Te^m - Dr^m) \\ PV_p^m \cdot \phi_-(t - Te^m + Dr^m, Dr^m) & \textbf{if} \quad (Te^m - Dr^m < t < Te^m) \\ 0 & \textbf{if} \quad (t \leq Ts^m \quad \text{or} \quad t \geq Te^m) \end{cases}$$

where ϕ_+ and ϕ_- are the functions that describe the onset and offset phase of the parameter activity. The activity of a parameter described by the formula above can be explained as follows: at the starting time, it increases to the target value during the onset duration; then it stays stable for a while; finally it starts decreasing during the offset duration resulting in no activity at the ending time.

We used Essa's work (Essa, 1994) on analysis, identification and synthesis of facial expressions to design the temporal pattern of facial movements. Essa used exponential curves to fit the onset and offset portions of each parameter. A function of the form $(e^{bx} - 1)$ is suggested for the onset portion, while a function of the form $(e^{c-dx} - 1)$ is suggested for the offset portion.

Based on the suggested functions by Essa, we derived two functions for the onset and offset portion of a parameter activity. For the onset portion, we want to choose b so that:

$$\phi_+(0, Do^m) = e^{b \cdot 0} - 1 = 0$$
$$\phi_+(Do^m, Do^m) = e^{b \cdot Do^m} - 1 = 1$$

From the second equation, we obtain:

$$e^{b \cdot Do^m} = 2,$$

and so

$$b = \frac{\ln 2}{Do^m}.$$

Replacing b with the obtained value, the derived function to describe the onset portion of a parameter activity is defined as:

$$\phi_+(x, Do^m) = \exp(\frac{\ln 2}{Do^m} x) - 1$$

For the offset portion, we want to choose c and d so that:

$$\phi_-(0, Dr^m) = e^{c - d \cdot 0} - 1 = 1$$
$$\phi_+(Dr^m, Dr^m) = e^{c - d \cdot Dr^m} - 1 = 0$$

From the first equation, we obtain:

$$e^c - 1 = 1,$$
$$e^c = 2,$$

$$c = \ln 2$$

Replacing c with the obtained value to the second equation, we can infer:

$$e^{\ln 2 - d \cdot Dr^m} - 1 = 0,$$

and so

$$d = \frac{\ln 2}{Dr^m}.$$

Replacing b with the obtained value, the derived function to describe the offset portion of a parameter activity is defined as:

$$\phi_{-}(x, Dr^m) = \exp(\ln 2 - \frac{\ln 2}{Dr^m} x) - 1$$

The activity of a muscle in a movement with $7s$ in duration, $2s$ in onset duration, and $3s$ in release duration can be seen in Figure 4.3.

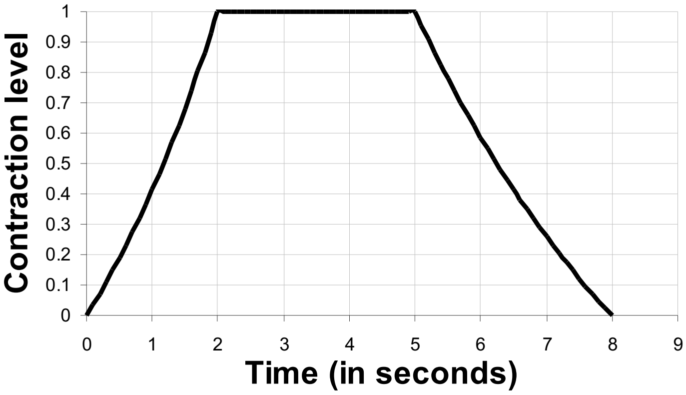


Figure 4.3: The activity function of a facial movement.

4.5 Combination of movements in one channel

We combine movements in one channel by modulating the activity of each muscle involved in the movements in that channel, to create transition effects between movements. Recall from Section 4.2 that movements in one channel occur sequentially. However, they can be specified to overlap each other. When there are two overlapping movements, there will be a transition from the preceding one to the following one.

We use the dominance model (Cohen and Massaro, 1993) to create the co-articulation effect of lip movements when talking. Co-articulation is the blending effect that surrounding phonemes have on the current phonemes.

For the combination of movements from other channels, we propose an algorithm in Section 4.5.2 to produce smooth transition between movements.

4.5.1 Combination of lip movements when talking

Cohen and Massaro (1993) have adapted Löfqvist's gestural production model (1990) to drive their synthetic visual speech. In this model, a lip movement corresponding to a speech segment is represented as a viseme segment. It has dominance over the vocal articulators that increase and decrease over time during articulation. This dominance function determines how close the lips come to reaching their target value of the viseme. Adjacent movements will have overlapping dominance functions which lead to a blending over time of the articulatory commands related to these movements. Each movement has not only a single dominance function but rather a set of such functions, one for each parameter.

The general form for dominance suggested by Cohen and Massaro is given by the negative exponential function:

$$\Delta(\tau) = e^{-\theta\tau^c}$$

In this function, dominance falls off according to the time distance τ from the segment center, to the power c modified by the rate θ. Different dominance functions can overlap for a given movement. The weighted average of all the co-occurrent dominance functions produce the final lip shape.

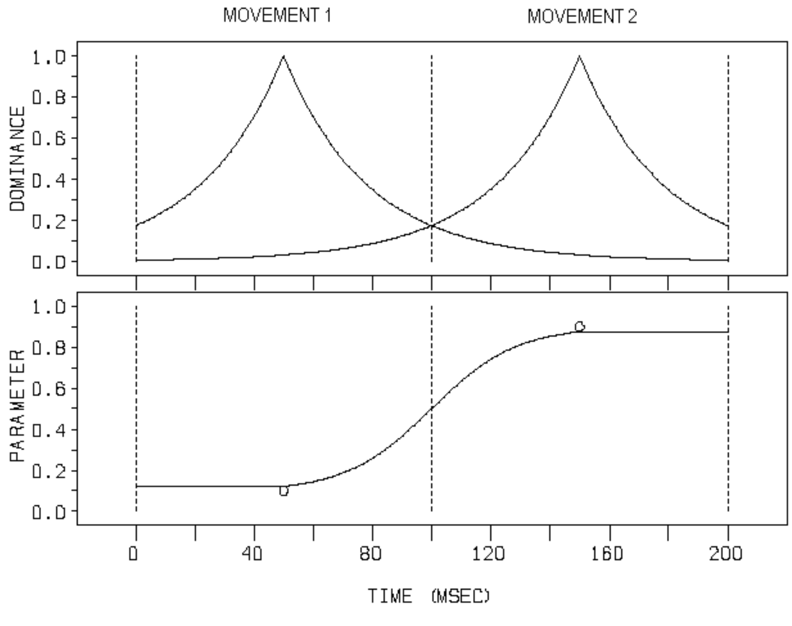


Figure 4.4: Dominance of a parameter of two lip movements over time (top panel) and the resulting activity of the parameter (bottom panel). Circles in the bottom panel indicate target value for the parameter in the two movements.

That general form is then expanded to:

$$\Delta_p^m(\tau) = \alpha_p^m e^{-\theta^m_{\leftarrow p}|\tau^c|} \quad , \quad \text{if} \quad \tau \geq 0,$$

for the case of time prior to the center of movement Γl^m. Quantity $\Delta_p^m(\tau)$ is the dominance of parameter p of lip movement Γl^m. The value of α_p^m gives the magnitude of the dominance function of parameter p of lip movement Γl^m, and $\theta^m_{\leftarrow p}$ represents the rate on the anticipatory side. Similarly, the dominance in the temporal range following the center of a unit is given by:

$$\Delta_p^m(\tau) = \alpha_p^m e^{-\theta^m_{\rightarrow p}|\tau^c|} \quad , \quad \text{if} \quad \tau < 0.$$

In both cases, the temporal distance τ from the peak of the dominance function is given by:

$$\tau = Tc_p^m + To_p^m - t$$

where t is the running time, To_p^m is the time offset from the center of movement Γl^m for the peak of dominance for parameter p, and

$$Tc_p^m = Ts^m + \frac{Te^m - Ts^m}{2}$$

is the time of the center of movement Γl^m. Using these dominance functions, we can combine the target values PV_p for each unit over time according to the weighted average:

$$F_p(t) = \frac{\sum_{m=1}^{N} \Delta_p^m(t) \cdot PV_p^m}{\sum_{m=1}^{N} \Delta_p^m(t)}$$

where N is the number of lip movements in an utterance. An example of dominance functions and the activity of a parameter after the co-articulation effect is shown in Figure 4.4. For this example, $\theta_{\leftarrow p} = \theta_{\rightarrow p} = 0.35$, $c = 1$, both movements last $100ms$, and the target value for the parameter in the two movements are 0.1 and 0.9 respectively. As can be seen from this figure, a gradual transition occurs between the two targets.

4.5.2 Other movements

We propose here an algorithm for the concatenation of facial movements other than lip movements. When there are two overlapping movements, we create the transition from the preceding movement to the following one based on their activity.

Consider two subsequent movements:

$$\Gamma o^1 = ((PV_i^1), Ts^1, Te^1, Do^1, Dr^1)$$

and

$$\Gamma o^2 = ((PV_i^2), Ts^2, Te^2, Do^2, Dr^2).$$

The two movements must occur sequentially, which requires:

$$Ts^2 > Ts^1$$

and

$$Te^2 > Te^1.$$

They are overlapping when:

$$Ts^1 < Ts^2 < Te^1$$

The activity of the combined movement follows the first movement until time Ts^2. Next, it increases/decreases to reach the target of the second movement and then follows the normal second movement. The activity of parameter p of the combined movement is described as follows:

(1) If the activity of parameter p of the first movement at the starting time of the second movement Ts^2, is less than the target value of parameter p of the second movement PV_p^2, which means $F_p^1(Ts^2) < PV_p^2$, then start increasing the combined activity of parameter p until it reaches the target value of parameter p of the second movement:

$$F_p^{12}(t) = \begin{cases} F_p^1(t) & \textbf{if} \quad t \leq Ts^2 \\ F_p^1(Ts^2) + \phi_+(t - Ts^2, Do^2) & \textbf{if} \quad Ts^2 < t < Ts^2 + \xi \\ PV_p^2 & \textbf{if} \quad Ts^2 + \xi \leq t \leq Ts^2 + Do^2 \\ F_p^2(t) & \textbf{if} \quad t > Ts^2 + Do^2 \end{cases}$$

where ξ is determined by $\phi_+(\xi, Do^2) = PV_p^2 - F_p^1(Ts^2)$,

(2) If the activity of parameter p of the first movement at the starting time of the second movement, Ts^2, is greater than or equal to the target value of parameter p of the second movement, PV_p^2, which means $F_p^1(Ts^2) \geq PV_p^2$, then start decreasing the combined activity of parameter p until it reaches the target value of parameter p of the second movement:

$$F_p^{12}(t) = \begin{cases} F_p^1(t) & \textbf{if} \quad t \leq Ts^2 \\ F_p^1(Ts^2) - PV_p^2 + \phi_+(t - Ts^2, Dr^1) & \textbf{if} \quad Ts^2 < t < Ts^2 + \xi \\ PV_p^2 & \textbf{if} \quad Ts^2 + \xi \leq t \leq Ts^2 + Do^2 \\ F_p^2(t) & \textbf{if} \quad t > Ts^2 + Do^2 \end{cases}$$

where ξ is determined by $\phi_-(\xi, Dr^1) = F_p^1(Ts^2) - PV_p^2$.

The combined movement of Γo^1 and Γo^2 is then further combined with the next movements in the same manner to create the final facial movement of the channel. An example of combining the Zygomatic Major of two movements in the same channel is shown in Figure 4.5. The Zygomatic Major's activity of the combined movement follows the first movement until time 3, when there is a stimulus to the second movement. It then releases to the target value of the Zygomatic Major in the second movement (0.5), followed by the second movement.

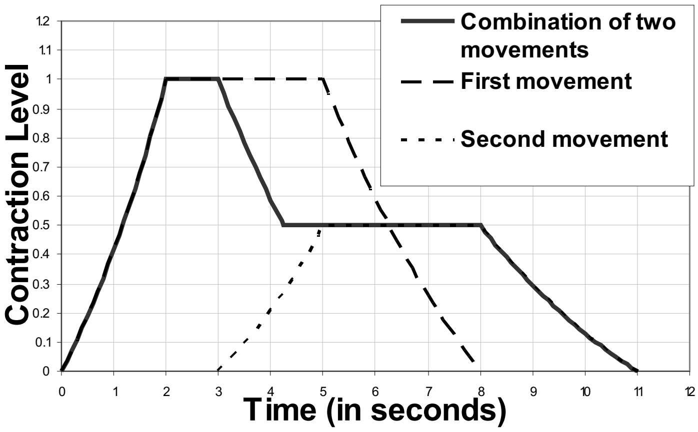


Figure 4.5: Combination of the Jaw Rotation of the two movements in the same channel.

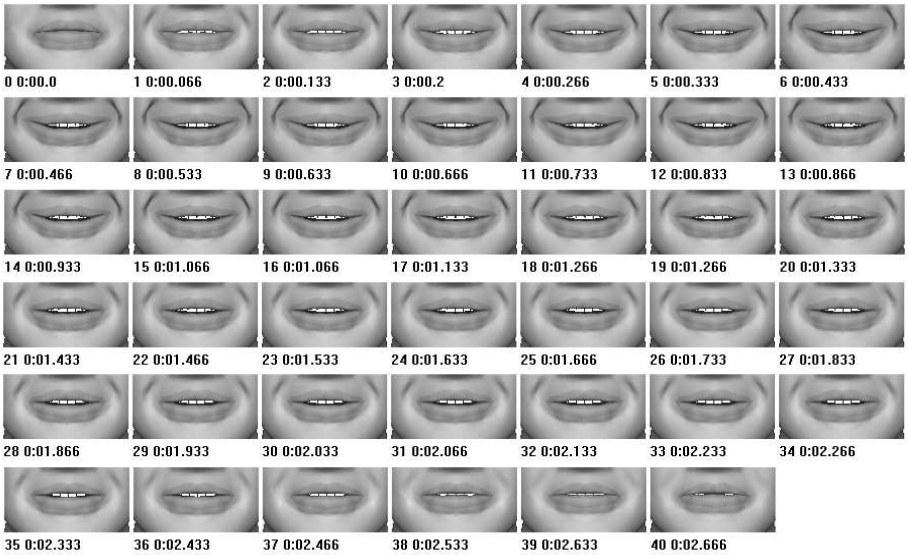


Figure 4.6: Our talking face display two smiles: a full smile followed by a half smile (frame by frame).

Figure 4.6 illustrates the combination of two subsequent, overlapping smiles. The first smile with intensity 1.0 starts at time 0 and lasts 2 seconds. The second smile with intensity 0.5 starts 1 second later and lasts 2 seconds. The combination of these two smiles result in a transition from a full smile to a half smile, which happens from frame 15 to frame 20. It is a full smile before frame 15, which follows the first smile, and half smile from frame 20, which follows the second smile.

4.6 Combination of movements in different channels

In this section we consider the combination of facial movements from different channels. To combine them, we first resolve the conflicts between the parameters involved in different movements. Then, the activities of parameters are combined by taking the maximum values.

At a certain moment in time, when there is a conflict between parameters in different animation channels, the parameters involved in the movement with higher priority will dominate the ones with lower priority. The activity of that muscle around that time is also affected so that the parameter cannot activate or release too fast.

In order to do so, for a parameter p at time t, we first determine if there is an activated conflicting parameter involved in a higher priority movement at that time t. If so, the activity of parameter p has to be deactivated. Next, we find the last time $T_{\leftarrow}$ before t when the activity of parameter p has to be deactivated because there exists an activated conflicting parameter involved in a higher priority movement. To prevent parameter p from activating too fast from time $T_{\leftarrow}$ to time t, the activity of parameter p at time t cannot exceed the highest possible level a parameter can reach within a duration of $t - T_{\leftarrow}$, that is:

$$\Psi_{\leftarrow}(t) = \phi_{+}(t - T_{\leftarrow}, Dmin_{+})$$

where $Dmin_{+}$ is the minimum onset duration for a parameter to activate completely. Finally, we find the next time $T_{\rightarrow}$ after t when the activity of parameter p has to be deactivated because there exists an activated conflicting parameter involved in a higher priority movement. To prevent parameter p from releasing too fast from time t to time $T_{\rightarrow}$, the activity of parameter p at time t cannot exceed the highest possible level a parameter can reach in order to release completely within a duration of $T_{\rightarrow} - t$, that is:

$$\Psi_{\rightarrow}(t) = \phi_{-}(Dmin_{-} - (T_{\rightarrow} - t), Dmin_{-})$$

where $Dmin_{-}$ is the minimum offset duration for a parameter to release completely. After all constraints for the activity of parameter p at time t have been determined, the modified activity $aF_p(t, c)$ of the paramter p at time t in channel c is defined as:

- $aF_p(t,c) = 0$ if there is an activated conflicting parameter involved in a higher priority movement at that time.
- $aF_p(t,c) = \min(F_p(t,c), \Psi_{\leftarrow}(t), \Psi_{\rightarrow}(t))$ if there is no activated conflicted parameter involved in a higher priority movement at that time.

The final activity of a parameter is obtained by taking the maximum value of this parameter from all channels c:

$$F_p(t) = \max_c(aF_p(t,c))$$

Figure 4.7 illustrates the effect of this algorithm. It shows the Orbicularis Oris muscle involved in speech. The Orbicularis Oris muscle conflicts with the Zygomatic major muscle and has higher priority. When the Orbicularis Oris is activated (at time 3), the Zygomatic major is inhibited. The activity of the Zygomatic major before that time is adjusted; so it does not release too fast, which creates an unnatural movement.

Figure 4.8 shows (frame by frame) our face model uttering the sentence “Oh, really? I like it very much”. Two emotion displays happen during the utterance. The first one is a full surprise display, which starts at time 0 and lasts 2 seconds. The second one is a full happiness display, which starts at 1 second later and lasts 3 seconds.

The figure shows the smooth and natural combination of visual speech and emotion displays. First, the transition from the surprise display to the happiness is shown from frame 16 to frame 23. The happiness display is then combined with the visual speech for “I like it” as there are no conflicting parameters in these movements. The conflict starts at frame 34, where the pronunciation of phoneme “v” in the word “very” requires the activation of the lip funneler Orbicularis Oris. This muscle conflicts with the smiling muscles Zygomatic Major and Minor. Because the visual speech movement has higher priority than the emotion display, the activation of the lip funneler Orbicularis Oris results in the deactivation of the Zygomatic Major and Minor. This deactivation, however, starts earlier from frame 30; so they do not release too fast to prevent unnatural movement. After finishing the sentence, the Zygomatic Major and Minor gradually reach the full contraction (frame 47).

4.7 Conclusion

In this chapter, we have discussed the problem of combining facial movements on a 3D talking head. We concatenated the movements in the same channel to generate smooth transitions between adjacent movements. Then the movements from all channels were combined taking into account the resolution of possible conflicting muscles. The activity of all muscles has been adjusted so that the muscles do not contract or release too fast. We have succeeded in creating natural facial animations of a talking head that utters while displaying other facial

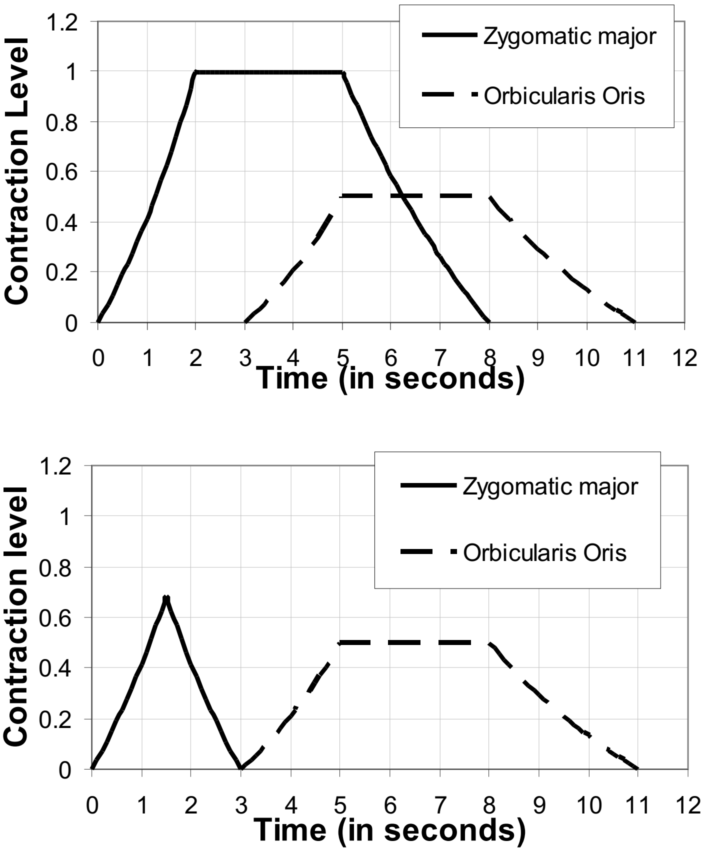


Figure 4.7: The activity of Zygomatic major and Orbicularis Oris before (top) and after (bottom) applying our combination algorithm.

movements such as conversational signals, manipulators and emotion displays as well.

With respect to the dynamics of the face, we believe that it would be interesting to test other functions that simulate the onset and offset portion of a muscle's activity. Although this is not very important for most of the facial movements, it may affect the believability of felt emotion displays. Finding the appropriate values of the onset and offset durations for felt emotion displays also plays an important role in making these displays natural, which is also an issue that needs to be investigated.

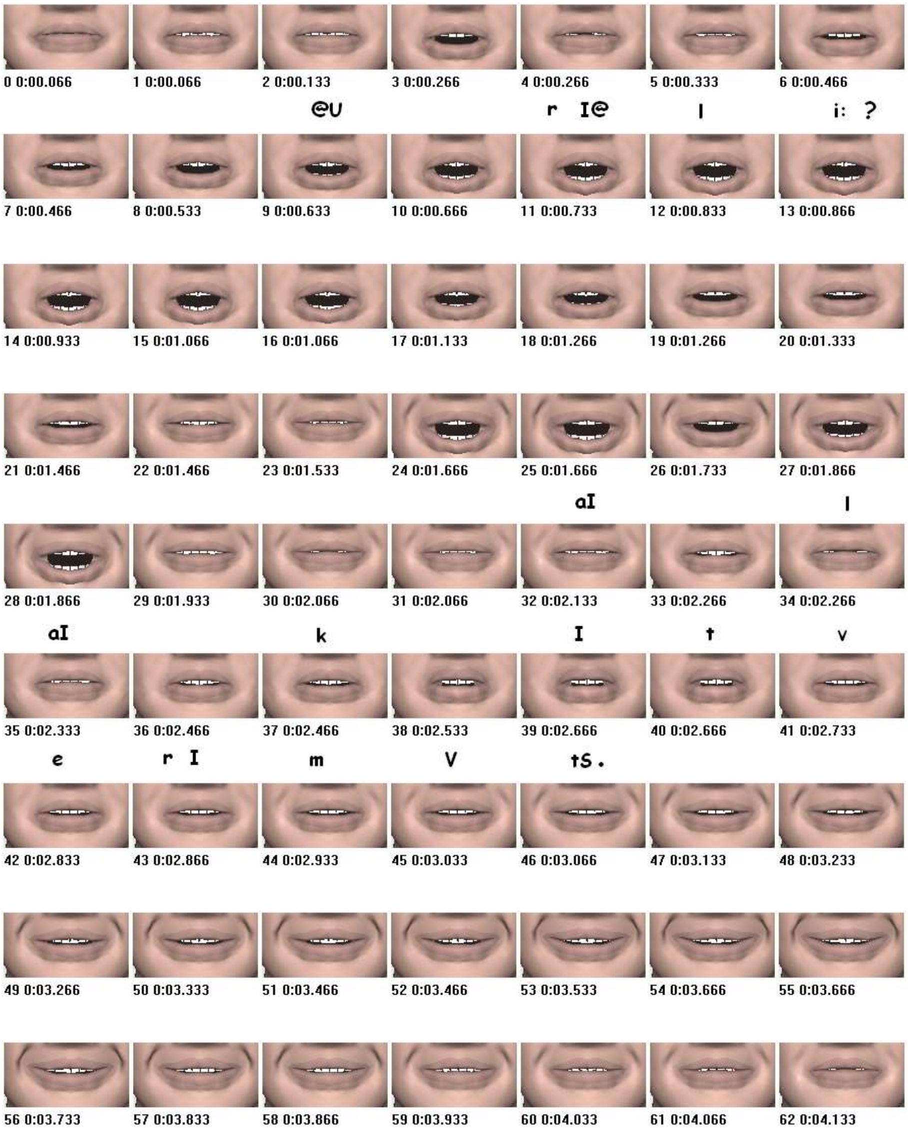


Figure 4.8: Our talking head utters the sentence “Oh, really? I like it very much.” while displaying surprise followed by happiness (frame by frame).

Part II

Emotions and Facial Expressions

Chapter 5

ParleE: A Computational Implementation of Emotions

> *"Anyone can become angry - that is easy. But to be angry with the right person, to the right degree, at the right time, for the right purpose, and in the right way; this is not easy."*
>
> – *Aristotle*

5.1 Introduction

Emotions have been studied for a long time. They have been viewed as obstacles to human cognitive functions. LeDoux (1996) shows in his "The Emotional Brain" that many scientists concluded that certain emotions make it impossible for us to think.

Recent findings show that emotions play an important role in human cognitive functions. According to many scientists including Galernter (1994), Forgas and Moylan (1987), and Damasio (1994), emotions affect creativity, evaluative judgement, rational decision making, communication, and other cognitive processes. This has been summarized by Picard (1997) in her "Affective Computing".

Galernter (1994) demonstrated in his "The Muse in the Machine" that emotions may affect creative thinking. Emotions can reduce one's concentration and make his thoughts become more random. Subsequently, this randomness can result in creative output. This sometimes happened to well-known painters, artists, writers and to many other people whose work requires creativity.

The study of Forgas and Moylan (1987) reveals the effect of emotions on evaluative judgement. This phenomenon has been discovered when researchers carried out interviews with different groups of people on some specific topic such as politics, crimes, future events and life satisfaction. These interviews were held outside a movie theater with people who just saw one or several movies and other people who did not. People who had seen the films surprisingly gave answers that have been deeply influenced by emotional tones of the film whereas no such effect was shown in the answers of those who had not entered the movie theater.

Confirming earlier results, Damasio (1994) has found a vital relationship between emotion and rational thinking. From his research, presented in his "Descartes' Error", it follows that different levels of emotion can lead to different levels of rationality, in particular, a low level of emotion can have a negative impact on decision making. This is his conclusion of his studies on a group of unusual patients with too little emotion. The nervous systems of these patients have been partly defected in such a way that they cannot experience emotion properly. Although these people appear to have normal intelligence and rationalness they all failed to make proper decisions in real life, regardless of whether it was an investment decision or a social one. They could not be aware of the link between bad feelings and hazardous choices; therefore they failed to make rational and efficient decisions. All these studies eventually imply that the lack of emotion can constrain the ability to be rational when it comes to decision making.

Emotions also play an essential role in communication (Picard, 1997). The expression of emotions indicates whether the communication should be continued, finished, or turned in another direction. For example, when people are being criticized and no emotions are shown on their face, it is the signal for the communication to be continued. When a look of sadness or anger appears on their face, then the criticism is understood and it is time to finish the communication.

Recognizing the importance of emotions to human cognitive functions, Picard (1997) concluded that if we want computers to be genuinely intelligent, to adapt to us, and to interact naturally with us, then they will need the ability to recognize and express emotions, to model emotions, and to show what has come to be called "emotional intelligence" (Goleman, 1995).

As early as the 1930s, traditional character animators, in particular from Disney, have incorporated emotion into animated characters to make audiences "believe in characters, whose adventures and misfortunes make people laugh - and even cry" (Thomas and Johnston, 1981). The animators believe that emotion, appropriately timed and clearly expressed, is one of the keys to creating the quality of animated films. According to Thomas and Johnston (1981),

> "from the earliest days, it has been the portrayal of emotions that has given the Disney characters the illusion of life."

In the areas of computational synthetic agents, emotions have received much attention for their influences in creating believable characters, e.g. (Elliott, 1992), (Bates, 1994). Elliott (1992) incorporates his implementation of emotion, the Affective Reasoner, into a world of some taxi drivers. These taxi drivers can experience twenty-four emotions in response to each other and to the situations that arise in the simulated world. They can reason about the emotions of other drivers. They can also express their emotional state as a series of actions, which are in turn represented as new events in the world. Taxi drivers have personality. They responded differently to different situations. For example, when one taxi driver is given a small tip, he may get angry while another one may just consider it is part of the job in the same situation. Both may also get angry but one is rude to his passenger while another may just smile and pretend that he does not care. All this, emotion and personality, has made these taxi drivers lifelike. Bates (1994) argued that the way a character feels about what happens in the world makes us care about that character; “if the character does not react emotionally to events, if they don’t care, then neither will we; the emotionless character is lifeless, as a machine.” Bates was the head of the Oz project in which real-time, interactive, self-animating creatures were situated in a small simulated world. These characters, which were based on the principles of traditional character animation, were equipped with emotions to make them believable.

So how to provide characters with emotions? According to Thomas and Johnston (1981), the emotional state of the character must be clearly defined and well displayed. A computational agent can display emotional expression without any real representation of emotions within itself in order to improve the interaction between computers and human beings. However, this provides no mechanism to keep consistency within the displayed emotions, making them less believable and the agent less comprehensible. A more preferable way is to use modelling techniques both to represent and to generate emotions. In this chapter, we discuss the problem of implementing emotions on computers. There are several things to keep in mind when we implement emotions computationally.

- Each application has its own domain. An implementation of emotions certainly has to work on the domain information. However, in order to be effectively reused for different applications, the implementation itself needs to be flexible and domain independent. In order to achieve this, the domain information should not be hard-coded into the implementation. A fixed rule-based system is not suitable here because a large part of the set of rules often must be rewritten and re-verified when extending or changing the application domain. This would cost huge manual efforts when building up an emotion implementation for each application. A more desired approach in this case should only take the domain information as the input. The implementation itself does not have to change when applied to a new domain.

- Emotions are dynamic. They occur at different levels of intensity and at different moments in time. There is a difference between “a little bit sad” and “very sad” in their role of influencing behavior, facial expression (Ekman and Friesen, 1975) and other emotions (Velásquez, 1997). Therefore, a believable agent also needs emotions with intensity. The implementation of its emotions should handle this. Moreover, without being affected, the emotions do not maintain forever (Ekman, 1984). Consequently, there should be some mechanism for the emotion intensities to decay over time.

- The environment around an agent dynamically changes over time. The mind of the agent should also learn to reflect the environment. Its emotion component is not an exception. El-Nasr et al. (2000) have demonstrated that learning components of an emotion implementation produce significant improvement in the believability of the agent's behavior. They carried out an experiment with two versions of an emotional pet, one that implements learning components and the other that does not. The users were asked to answer a questionnaire after being shown these two versions of the pet. The results from the questionnaire shows that the behavior of the first one is more believable.
- Different people experience emotions in different ways even in the same situation. A pessimistic student may feel sad when getting a bad mark for an assignment while an optimistic one almost does not feel anything at all. A single person in different states may also experience emotion differently. People are more likely to get angry when they are tired. It would be nice to have an implementation that can let different agents experience emotions differently in different states. Personality plays an important role in differentiating people (Nye and Brower, 1996). It is an important feature that can enhance believability of animated characters (Thomas and Johnston, 1981). Motivation states are internal states that drive the subject to particular actions and also affect emotional states (Bolles and Fanselow, 1980). These observations suggest that an implementation of emotion should incorporate personality and motivational states.

In this chapter, we describe ParleE, a quantitative, flexible and adaptive computational implementation of emotions for an embodied agent situated in a multi-agent environment. ParleE has been inspired by various other implementations of emotions such as Elliott's Affective Reasoner (Elliott, 1992), Velásquez's Cathexis (Velásquez, 1997), El-Nasr et al's FLAME (El-Nasr et al., 2000) and, in particular, by Gratch's Émile (Gratch, 2000). Like some of these and many other implementations, ParleE generates emotions based on Ortony et al.'s theory of appraisal (Ortony et al., 1988).

Nevertheless, ParleE possesses some significant properties of its own. The main novel differences with other systems have to do with (i) the way it uses forward-chaining search within a finite depth to obtain the probability of achieving a goal; (ii) the way it uses models of other agents' plans and goals to predict their behavior and set up expectations about the likelihood of events; and (iii) the way it incorporates personality, individual characteristics and motivational states in the implementation.

We first review psychological research on emotion in Section 5.2. In this section, we will have a look at four different perspectives on emotion: the *Darwinian*, the *Jamesian*, the *social constructivist*, and the *cognitive*. Also in this section, we discuss in more details two cognitive theories of emotion, namely Roseman's theory (Roseman, 1984; Roseman et al., 1990, 1996) and Ortony et al.'s theory (1988), that are used by many computer scientists to actually implement emotions on computers. In Section 5.3, we give an overview of existing computational implementations of emotions with focus on four implementations that inspired ParleE, i.e. Cathexis (Velásquez, 1997), Affective Reasoner (Elliott, 1992), FLAME (El-Nasr et al., 2000) and Émile (Gratch, 2000). We finish the review on existing work with discussing models of personality in Section 5.4. Section 5.5 discusses the ParleE implementation. An overview of the implementation is presented first followed by the detailed description of each component of the implementation. Section 5.6 shows how to calculate the intensity of triggered emotions. We first present several variables that contribute to the intensity of emotion. Next, formulas to calculate the values of an event's impact on emotions are presented. These values are used to update the emotional state taking into account the decay process. We also present a way to determine the effect of personality on the intensity of emotions. ParleE is based on a planner and several learning components. The planner we used in this implementation is discussed in Section 5.7. It is a probabilistic planner which gives the probability for an agent to achieve a goal at a specific state. It incorporates models of other agents in order to predict the behavior of other agents. Learning components are discussed in Section 5.8. Finally, an illustration of the implementation is presented in Section 5.9.

5.2 Psychological research on emotions

5.2.1 Views on emotion

Predate psychological studies on emotion seemed to be so diverse that it was very hard to find some central points of view (perspectives) on emotions. Nowadays, as argued by Cornelius (1996) in his "The Science of Emotion", most recent research on emotions demonstrates a small number of traditions only. More precisely, he describes four major theoretical traditions in psychology to define, study, and explain emotions. These are the *Darwinian*, the *Jamesian*, the *social constructivist*, and the *cognitive* perspectives on emotions. Each of these perspectives has its own set of assumptions to represent a different way of thinking about emotions. The *Darwinian* perspective, initiated by Darwin (1872/1965), argues that emotions are universal and have adaptive functions; the *Jamesian* perspective, initiated by James (1884), considers emotions as bodily responses; the social constructivist perspective, first presented by Averill (1980), judges emotions as social constructions to serve social purposes; and the cognitive perspective, first employed by Arnold (1960), believes that emotions are based on cognitive appraisal.

There is a degree of overlap among these perspectives and not all research on emotion solely follow just a single perspective. Some work deals with two or three of the perspectives. For example, the work of Ekman (1984) follows both the *Darwinian* and *Jamesian* perspectives to understand emotions.

The Darwinian perspective

To paraphrase Cornelius (1996), the Darwinian perspective focuses on the function of emotions in the context of evolution by natural selection. Psychologists who work within this perspective focus on the emotional displays of humans and other animals. Since humans share part of their evolutionary history with other mammals, especially with other primates, their emotions and expressions of emotions should show some similarities. Darwin has claimed that the behavioral mechanisms that we consider "emotional expressions" evolved not for the purpose of expressing emotion, but for other purposes; they are considered "emotional" because they accompany other actions associated with strong emotions. The disgust face a person makes is similar to the face a dog makes when they smell or taste some unpleasant food because both are originally associated with the action of spitting out the food.

People from many different cultures can recognize the facial expressions of a small number of emotions. The studies of Ekman (1972) and of Izard (1994) shows that there are "constants across culture" in certain simple facial expressions of emotion. These studies imply that this universality is part of human evolution. The face one makes is similar to the faces other humans make when they are angry because such faces are important communication tools during our species' long history.

Some psychologists in this perspective also examine the adaptive functions evident in other aspects of emotion. Plutchik's "psychoevolutionary" theory of emotion (1984) points out the importance of emotional behavior in the survival of all animal life. Plutchik considers emotions as the adaptations to life's contingencies. He proposes several basic behavior patterns that may be found in all organism such as *incorporation* (intake of food) or *protection* (avoiding danger). These behavior patterns are mapped into emotional words. For example, fear/terror is associated with the pattern *protection*. The emotions that emerge from this mapping form basic emotions that all organisms share. Sharing the Darwinian perspective, Frijda's theory of "action tendencies" (1986) considers emotions as the awareness of action tendencies. The action tendencies are closely linked to the way a person perceives or "appraises" the environment. The evolutionary prototype theory of emotion presented by Shaver et al. (1992) starts with a similar assumption to Plutchik's one that all human beings share a set of prototypical reactions to the environment. Following Frijda that emotions are "action tendencies" that follow from a person's "appraisal" of the environment, Shaver and his colleagues argue that there should be a small number of basic emotions recognized by all cultures.

The Jamesian perspective

The work of James (1884) insists that the experience of bodily changes primarily originates the experience of emotion. James' theory of emotion is often called the "James-Lange" theory because of a similar theory by Lange (1922) shortly after the publication of James' theory. This theory is the starting point for many other theories of emotions. James considers three types of bodily change: expressive behavior such as crying and smiling, instrumental acts such as running and cowering, and physiological changes such as trembling. Modern approaches of this perspective consider "visceral" changes and expressive behavior as bodily changes. "Visceral" changes are the increases in the sympathetic nervous system, a branch of the autonomic nervous system (ANS). These changes express as activity and the effects of such activity on the heart, stomach and other organs innervated by the sympathetic nervous system (Grings and Dawson, 1978; Shields and Stern, 1979). Expressive behavior are changes in posture and facial expression (Izard, 1990; Laird and Bresler, 1990).

Several studies on emotions from Jamesian perspective including that of Levenson (1992) and Ekman et al. (1983) have demonstrated that a small number of emotions such as fear, anger, sadness, and happiness are believed to be differentiated by various patterns of autonomic activity. The results from the study within the Jamesian perspective also suggest that the autonomic nervous system feedback helps to determine the intensity of the emotion experienced. Research by Allport (1924), Izard (1981) and other researchers shows that feedback from the face may also be used to determine the intensity of emotion and to differentiate emotions.

The social constructivist perspective

In contrast to Darwin's and James' assumptions that emotions are mainly biological phenomena, the social constructivists believe that emotions are culture-associated and can only be analyzed by looking at different social levels. Averill (1980) defines an emotion as

> "a transitory social role (a socially constituted syndrome) that includes an individual's appraisal of the situation and that is interpreted as a passion rather than as an action."

For example, if one is humiliated by one of his good friends in the presence of several other people, he is expected to become angry if he grew up in the United States or another Western country; however, if he grew up in Japan, he might simply smile at the friend who angered him.

Research within the social constructivist perspective is part of a larger social constructivist program in psychology (Gergen, 1985), sociology (Coulter, 1989) and other related disciplines. The aim is to discover the socially constructed nature of many phenomenons such as emotions, gender and even science. According to Cornelius (1996), the main reasoning of social constructivists is that the experience and expression of emotions are dependent on learned conventions or rules. These conventions or rules are different in different cultures. Based on that reasoning, social constructivists have found important cross-cultural and historical differences in the way emotions are conceptualized and experienced. Social constructivist-oriented research also claims that the expression of emotions shows its cross-cultural variability. This is often questioned by those researchers following the *Darwinian* and *Jamesian* perspectives, who believe there is a degree of universality in the expression of emotions. The evidence of this variability or universality, however, is still controversial.

The cognitive perspective

The cognitive perspective disclosed the role of cognition in the experience of emotion by focusing on the relationship between emotions and the way an individual appraises events in the environment. Emotions are considered as responses to the meaning of events with regard to the individual's goals and motivations.

In 1960, Arnold (1960) started this perspective of research on emotions by first questioning the validity of James' famous formulation:

BODILY CHANGES = EMOTION.

If emotion is considered to be bodily changes, then which process initiates the bodily changes in the first place. Arnold argued that the emotions actually are initiated by an individual's appraisal of his or her circumstances. Appraisal refers to the process of judging the personal significance for good or ill of an event. Arnold recognized that a person's past experience and his or her goals are important aspects of the way that person appraises a situation. According to Arnold, the sequence for thinking about emotions should be:

PERCEPTION - APPRAISAL - EMOTION.

On the other hand, Arnold still believed Darwin's idea that emotions serve survival-related purposes and James' idea that every emotion has its own distinct pattern of bodily activity.

Shortly after the publication of Arnold's cognitive theory of emotion, Speisman et al. (1964) did a series of studies to conform to what Arnold's theory would have predicted, namely, that the character of a person's emotional response to an event depends on how he or she appraises the event. The result of these studies are then replicated and extended in a follow-up study by Lazarus and Alfert (1964). Lazarus started his cognitive-motivational-relational theory in (Lazarus et al., 1970) and fully developed it recently (Lazarus, 1991). The main idea of this theory, which is supported by most of the recent cognitive approaches to emotion, is that emotions are post-cognitive. This idea was attacked by Zajonc (1980) with the argument that some emotions can occur when no cognitive activity has taken place. Both Lazarus and Zajonc may be right as their focuses are different: Lazarus focuses on complex emotions while Zajonc focuses on very simple emotions.

There are now many researchers working within this perspective. Examples include the work by Mandler (1975) which points out the role of the "cognitive interpretation" of arousal - the activities if the autonomic nervous system, and the perception of arousal; Oatley and Johnson-Laird's "communicative" theory of emotion (1987) that emphasizes that the central functions of emotions within the human cognitive system are to carry out communication; and so on. Among the cognitive theories of emotion, many studies (Frijda, 1986; Lazarus, 1991; Oatley and Johnson-Laird, 1987; Ortony et al., 1988; Roseman, 1984) are concerned with the specification of a cognitive structure associated with emotions. The results of these research efforts have not only some theoretical importance but also practical significance for many fields such as psychotherapy and artificial intelligence.

Summary

The four different views on emotions help us to further understand the nature of emotion. Our knowledge of emotions has been advanced considerably by the research within the evolutionary perspective, especially the one on facial expressions by Ekman et al. and Izard et al. Much evidence has been presented to show that human emotions are part of our evolutionary heritage. Without contrasting to the evolutionary perspective, researchers who follow the Jamesian perspective have demonstrated the relationship between emotions and bodily changes - ANS changes in particular. The research within the cognitive perspective has shown how emotions depend on the way human appraise events. Finally, the social constructivist perspective illustrates how experience and expression of emotions depend on rules that are learned from social relationships.

Our implementation of emotions presented in this chapter is based on the works within the cognitive perspective. However, we do not claim for the validation of these works.

5.2.2 Cognitive models of emotion

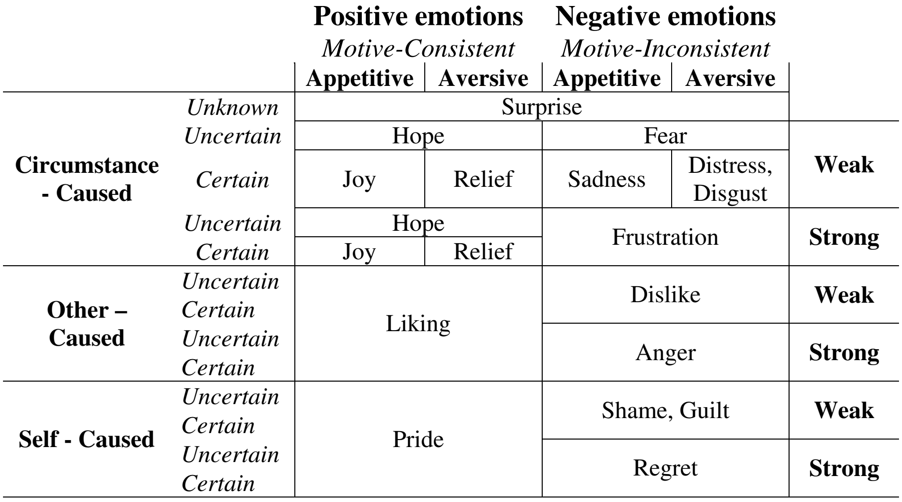
We will now discuss in more details two cognitive theories of emotion, namely Roseman's theory (Roseman, 1984; Roseman et al., 1990, 1996) and Ortony et al.'s theory (Ortony et al., 1988), that are used by many computer scientists, e.g. (Elliott, 1992; El-Nasr et al., 2000; Gratch, 2000), to actually implement emotions by means of computers. Both models fit within the cognitive view on emotions.

Roseman's model of emotion

Roseman et al. specify a cognitive structure associated with emotions based on the assessment of an event that gives rise to seventeen emotions (Roseman, 1984; Roseman et al., 1990, 1996). This structure is illustrated in Figure 5.1.

Roseman's model has several dimensions. One of the dimensions is used to divide events into motive-consistent, and motive-inconsistent events. Motive-consistent events are consistent with one of the subject's goals whereas a motive-inconsistent event threatens one of the subject's goals. The event can be motivated by the desire to obtain a reward (appetitive motive) or to avoid punishment (aversive motive). For example, an event that is appetitive motive-consistent elicits joy; an aversive motive-consistent event produces relief. Events are also categorized by "agency", that is an event can be caused by circumstance, other or self. Unexpected and circumstance-caused event produces surprise. Other circumstance caused events elicit emotions such as hope, joy and sadness. The probability of the outcome of the event and the potential an individual has to control the situation are used to further sort out emotions in this category. A motive-consistent event with uncertain outcome produces hope whereas the one with certain outcome produces joy or relief. If the individual has low potential of controlling the situation, a motive-inconsistent event produces fear, sadness or distress. If the individual has high potential of controlling the situation, a motive-inconsistent event produces frustration and disgust. When the event is caused by others, emotions are felt toward others, namely liking-love, dislike, anger and contempt. The emotions felt toward self are pride, regret, guilt and shame.

Roseman's model received positive feedback from the AI society because of its simple structure which can be translated quickly into rules to define which appraisal triggers which emotion. However, as admitted by Roseman et al. (1996), the model is difficult to prove empirically. It also has a problem of dealing with a situation in which one person makes two different appraisals. If two states of the same dimension are present at the same time, Roseman's model cannot clearly predict what emotion will be experienced.



		Positive emotions *Motive-Consistent*		**Negative emotions** *Motive-Inconsistent*		
		Appetitive	**Aversive**	**Appetitive**	**Aversive**	
Circumstance - Caused	*Unknown*	Surprise				
	Uncertain	Hope		Fear		**Weak**
	Certain	Joy	Relief	Sadness	Distress, Disgust	
	Uncertain	Hope		Frustration		**Strong**
	Certain	Joy	Relief			
Other – Caused	*Uncertain*	Liking		Dislike		**Weak**
	Certain					
	Uncertain			Anger		**Strong**
	Certain					
Self - Caused	*Uncertain*	Pride		Shame, Guilt		**Weak**
	Certain					
	Uncertain			Regret		**Strong**
	Certain					

Figure 5.1: Roseman's model.

Ortony et al.'s appraisal model of emotion

Ortony et al. (1988) developed a model of emotion named OCC, another event-appraisal model that is similar to Roseman's one but they divide the concerns of an agent into **goals**, **standards**, and **preferences**.

Goals are divided into three types: *active goals*, i.e., goals that are states of the world that agents want to bring about and can directly plan to make happen such as getting a job; *interest goals*, i.e. goals that are also states of the world that agents want to become reality but in general they cannot take direct action to accomplish them such as seeing a supported soccer team win; and *replenishment goals*, i.e., goals that are periodically reactivated based on the time since last fulfillment such as attaining food, sleeping, etc. **Standards** represent how people should behave. They are triggered not only when something relevant happens to an agent but also when something relevant happens to anyone. **Preferences** track the likes and dislikes of an agent.

Emotions in the OCC model are generated by three types of subjective appraisals: (1) the appraisal of the pleasingness of the events with respect to the agent or another agent's goals; (2) the appraisal of the approval of the actions of the agent or another agent with respect to a set of standards for behavior; and (3) the appraisal of the liking of objects with respect to the attitudes of the agent. There are also some emotions that are caused by combinations of other emotions. For example, joy is produced by the occurrence of a desirable event for self; gloating is experienced by an individual when he or she is pleased about an event presumed to be undesirable for someone else; pride appears when one approves his or her own action; and dislike is associated with an unappealing object. The full cognitive structure associated with emotions in the OCC model is presented in Figure 5.2.

To calculate the intensity of emotions, some global and local variables that can potentially affect the process by which an emotion is triggered are defined. Examples of local variables are the likelihood of an event to occur, the effort to achieve some goal, the possible realization of a goal, etc. Examples of global variables are sense of reality, unexpectedness, etc.

The OCC model of emotions provides a clearer view on emotions compared to Roseman's model. In Roseman's model, events are appraised only according to goals. This way, attitude and standard related emotions such as like/dislike, anger are not defined in a reasonable way. The OCC model treats these emotions more appropriately. We therefore base ParleE on the OCC model. However, because there is no definition for the emotion surprise in the OCC model, we have to adopt it from Roseman's model.

5.2.3 Emotions, moods and motivational states

When talking about emotions, it is also worthwhile mentioning moods and motivational states and the differences between them. According to Sloman (2002), moods are related to, but different from, emotions. Moods tend to last longer than emotions. Oatley (1992) refers moods as emotional states that last for more than a few hours, especially when the subject is unaware of how the state started. From the cognitive perspective, Ortony et al. (1988) considers emotions as "changes in action readiness, and other cognitive changes accompany them". Moods are also based on exactly the same kinds of readiness. However,

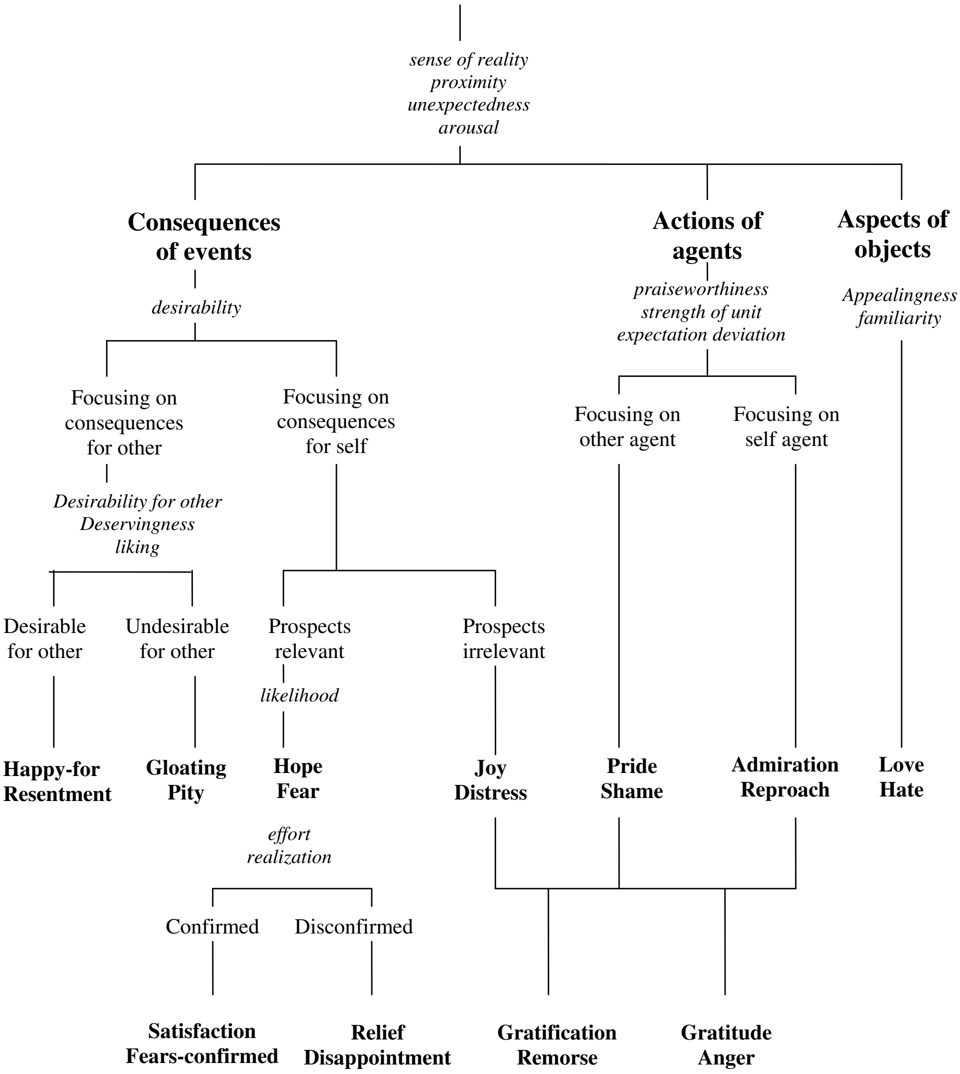


Figure 5.2: The OCC model.

moods are continuing “background states” that are almost not affected by external stimuli. Following Oatley (1992), when modeling emotions in Cathexis, Velásquez (1996) considers a mood as an emotion type with intensity lower than an activation threshold. In Cathexis, emotions have effects on other emotions and other mental processes. When an emotion type is in the state of mood, it almost does not influence other processes. Differently, in FLAME, El-Nasr et al. (2000) just consider two kind of moods: positive/good mood and negative/bad mood. They use moods to influence the learning process of simulated characters.

Motivational states are any internal states that tend to drive the subject to take actions to fulfill important needs (Bolles and Fanselow, 1980). These needs are similar to the concept of replenishment goals in Ortony et al.'s model of emotion (1988). For example, the subject is driven to get food when he/she is hungry. Motivational states are believed to have effect on emotions (Bolles and Fanselow, 1980), e.g., pain may inhibit fear; hunger may inhibit happiness; and so on.

5.3 Existing computational implementations of emotions

The advent of many emotions theories, especially the theories by Ortony et al. (1988) and Roseman (1984), lead the way to many computational implementations of emotions. These implementations are in many forms: rule-based systems such as the Oz project (Reilly and Bates, 1992) and the Affective Reasoner (Elliott, 1992); fuzzy rule-based systems such as FLAME (El-Nasr et al., 2000); distributed systems such as Cathexis (Velásquez, 1997); connectionist systems such as (Lisetti, 1999; Inoue et al., 1996; Bozinovski and Bozinovska, 1999); Dynamic Belief Networks such as (Rosis et al., 2003); plan based systems such as Émile (Gratch, 2000). None of these systems, however, fully provide a flexible, domain independent and adaptive way of implement emotions with intensity, which also incorporates personality and motivational states.

We will now study in more detail some implementations that inspired ParleE.

5.3.1 Cathexis model

Cathexis has been proposed by Velásquez (1997) to generate emotions and focus on their influence on the behavior of synthetic agents. It differentiates emotions from a similar kind of effective phenomena, namely moods. Moods are explained as low tonic levels of arousal within the emotional system while emotions are explained as higher levels of arousal in the same system. Cathexis also introduces the concept of "temperament", which determines how easy/difficult an individual would experience different emotions. Cathexis models emotions, moods and temperaments as a network of systems comparable to Minsky's proto-specialist (Minsky, 1988). Two thresholds are associated with each system. The first threshold, denoted by α, controls the activation of the associated emotion type and differentiates between moods and emotions. Emotion types with intensity level lower than their α threshold are considered as moods, and do not have as much influence as emotions on an agent's behavior. Emotion types with intensity level higher than their α threshold are considered emotions. The second threshold, denoted by ω, controls the saturation of the associated emotion type: it is the upper limit for intensity of that emotion type.

These proto-specialists run in parallel and constantly update the intensity of the emotions associated with them. They also influence each other. When the intensity of the emotions associated with these proto-specialists exceeds the α threshold, they may inhibit or excite other proto-specialists. For example, Fear inhibits Happiness, Happiness and Sadness inhibit each other, and so on.

Each proto-specialist is equipped with sensor detecting internal (neurotransmitters, brain temperature, etc) and external (events from environment) stimuli to check for the existence of the appropriate conditions that would elicit the emotion represented by that proto-specialist. The sensors in Cathexis are divided into four categories: Neural, Sensorimotor, Motivational and Cognitive. Neural sensors detect stimuli from neurotransmitters, brain temperature, and other neuroactive agents. Sensorimotor sensors detect stimuli from sensorimotor processes, such as facial expressions and body posture. Motivational sensors receive inputs from all motivations, which include drives (e.g., Thirst and Hunger), emotions (e.g., Sadness and Happiness), and pain regulation. Cognitive sensors receive inputs from all kind of cognitions, such as appraisal and interpretation of events, beliefs and desires, memory and so on. The cognitive elicitors in Cathexis are implemented based on Roseman's revised theory of emotions presented in (Roseman et al., 1990). Input from these sensors changes the intensity of the emotion type. To calculate the intensity of emotion e at time t, the intensity of all emotions at time $t-1$ and the values of emotion e's elicitors are used. The value of an elicitor of emotion e is derived from the value and weight of the conditions that contribute to the activation of the elicitor. These values and weights have to be predefined for each application. The huge effort to correctly determine these values and weights for each particular application makes this model inflexible. Parallel to the update of emotional intensity from stimuli, there is a decay process going on, which eventually brings the emotional intensity to a state of rest.

Cathexis models six basic emotions: anger, fear, distress/sadness, enjoyment/happiness, disgust and surprise. The special property of Cathexis is that it models not only cognitive elicitors but also non-cognitive elicitors. However, many values to calculate the emotional elicitors are hard-coded inside Cathexis, which makes the system inflexible. Moreover, there are no adaptive components inside the system to take into account the dynamical nature of the environment. Personality is not mentioned in Cathexis at all.

5.3.2 Elliott's Affective Reasoner

Elliott's Affective Reasoner (Elliott, 1992) is a computational adaptation of the OCC model (Ortony et al., 1988). This implementation assesses the relationship between events and an agent's disposition which is described by its goals, social standards, and preferences. The relationship is characterized in terms of a set of features called *emotion-eliciting conditions*.

Each agent includes a representation of the self and identity of other agents involved in the situation. During the simulation, each agent judges an event according to the pleasantness and status of the event by matching the event to knowledge structures called "construal frames". These frames first assess whether the event is relevant to the agent. If so, they extract several features of the event to evaluate the significance of the event to the agent's emotion. These features are used as the input of a set of domain-dependent rules to generate emotions. The evaluation of the event is based on several conditions:

- **Self**: with respect to which agent is the event assessed?
- **Desire-self**: is the event considered to be *desirable* for the agent?
- **Status**: what is the status of an expectation? Is that *unconfirmed*, *confirmed*, or *disconfirmed*?
- **Evaluation**: does the event uphold or violate the social standard?
- **Responsible agent**: which agent is responsible for the upholding or violation of the social standard?
- **Desire-other**: is the event desirable for other agents?
- **Other**: to which other agents is the "**desire-other**" assessed?
- **Pleased**: what is the reaction of the agent to "desire-other" status of other agents?
- **Appealingness**: does the event contain an attractive or repulsive object?

Elliott proposed in the Affective Reasoner rules for triggering emotions with the input being the assessment of events based on the above conditions. For example, when the event is *confirmed* and *desirable* for **self**, joy is triggered. A domain-dependent example of this rule in the Affective Reasoner looks like: "if the supported soccer team scores a goal then trigger joy". The Affective Reasoner can generate twenty-four different emotions. These emotions are expressed through a set of predefined emotional expressions.

The Affective Reasoner also considers social interaction in modeling emotions. Agents' own knowledge of emotions and actions is used to understand the other agents emotional states, emotional expressions, and their actions, which can potentially enhance the interaction process.

Elliott's Affective Reasoner does not address the following issues. It is not a quantitative implementation as it does not consider the intensities of emotions. Besides, there is no conflicting emotion resolution, adaptation, or integration of personality and motivational states in this implementation. It is also limited by the use of domain-specific rules to appraise events.

5.3.3 FLAME

El-Nasr et al's FLAME model (El-Nasr et al., 2000) is a computational implementation of emotions which uses fuzzy logic rules to map assessments of the impact of events on goals into emotional intensities. FLAME consists of three components: an emotional component, a learning component and a decision-making component.

The agent in FLAME first perceives external events from the environment. Each perceived event is evaluated by the emotional component. The emotional component first determines which goal is affected by the event. Then it determines the desirability of the event to the agent's goal. Fuzzy rules are used for this purpose. After being evaluated, the desirability is used by the appraisal process to update the emotion state of the agent. FLAME uses a combination of Ortony et al.'s model (Ortony et al., 1988) and Roseman's model (Roseman, 1984) to trigger emotions. The triggered emotion or mixture of emotions is filtered to produce a coherent emotional state. To filter emotions and resolve conflicting emotions, FLAME incorporates an emotion filtering component, which takes into account motivational states and mood. The filter process is based on the work of Bolles and Fanselow (1980). When a motivational state reaches a sufficient level, it may inhibit emotional process to perform actions that fulfill the need associated with the motivational state. In FLAME, while motivational states directly inhibit the emotional process, mood participates in the inhibition process between emotions. When an agent is in the negative mood, negative emotions tend to inhibit positive emotions; when an agent is in the positive mood, positive emotions tend to inhibit negative emotions.

FLAME can generate emotions with intensity. The intensity of emotions also decays eventually. FLAME calculates the intensity of triggered emotions from the value of expectation and desirability of an event based on the formulas proposed by Price et al. (1985). Almost all of the formulas are in the form of a weighted sum of the values of expectation and desirability. We do not think that this is an appropriate way of calculating the intensity of emotion. Let us have a look at an example: the formula to calculate the intensity of Sadness,

$$Sad = (2 \cdot \text{expectation}^2) - \text{desirability}$$

Suppose that the agent drops a one-cent coin into the river. In this case, one cent is very small to the total amount of money the agent has. The event that the agent loses the coin is an expected event with the expectation value close to 1.0 and with the value of desirability close to 0.0. According to the formula above, its occurrence would trigger a sadness with very high intensity. However, in this case, the intensity of sadness should be close to zero as the event does not harm the agent's goal that much. The formulas presented in FLAME are also inappropriate in the way they use expectation of an event to calculate intensity of emotions such as joy and sadness. According to these formulas, the highly expected events would cause emotions such as joy and sadness with high intensity. Actually, the intensity of emotions such as joy should be higher when the desirable event is highly unexpected.

To increase the adaptation in modelling emotions, FLAME incorporates a learning component . This learning component learns patterns of events, associations between objects and emotions, and the impact of events. The patterns of events are learned with a probabilistic approach based on the conditional probability an event happens provided several previous events happen in a specific order. The associations between objects and emotions are learned with classical conditioning technique. Every time an event that contains an object occurs, using the count and the intensity of emotion triggered by the event, FLAME learns the intensity of the emotion associated with the object. This type of learning provides a type of expectation triggered by objects rather than events. Impact of event is learned by the accumulation of observations about many transitions of environment states, along with occasional rewards.

In order to incorporate emotion into the decision making process, FLAME uses another fuzzy rule-based system to determine the behavior according to the situation, the emotion and mood of the agent.

The events and the agent's goals in FLAME have to be hard coded in the rules. Every time the domain is extended or changed, the rules have to be changed and verified. This makes FLAME a domain-dependent and inflexible approach. Moreover, FLAME does not provide a way of calculating the impact of an event on an agent's goal. Instead, it uses a predefined reward value for the user's action's impact on an agent's goal. Personality is not incorporated in this model either.

5.3.4 Gratch's Émile

Gratch's Émile (Gratch, 2000) uses classical planning methods to appraise the emotional significance of events as they relate to plans and goals, to model and predict the emotional state of other agents, and to alter behavior accordingly. The core of classical planning methods is to detect and resolve threats that prevent an agent from obtaining his/her goals. Social standards in Émile are defined as domain-specific constraints of behavior such as "thou shalt not kill".

Émile is inspired by Elliott's construal theory implemented in his Affective Reasoner (Elliott, 1992). However, Émile's classical planning view of an agent's mental state shows advantages over the rule-based system in the Affective Reasoner. First, planning techniques reduce the difficulties in representing future expectations of the considered agents by obtaining these expectations from the plan representation. This is a more efficient domain-independent solution to manage expectation, which improves the domain-specific problem inherent in a rule-based system. Second, the plan-based implementation provides an easy way to calculate the value of variables contributing to the intensity of triggered emotions. Rule-based systems face the problem of determining these values.

In the implementation of Émile, Gratch has proposed a simple way of calculating the probability of a goal being achieved, the probability of threats to a goal and its importance. He also proposed formulas to calculate the intensity of triggered emotions. The formulas are in the form of a product between the importance of a goal and the probability of achieving that goal or the probability of threats to that goal. However, because there is no event expectation in Émile, these formulas do not contain event expectation either.

The limitation of this model is that it leaves out the value of an event's unexpectedness when calculating the intensity of event-based emotions like joy and distress. Moreover, Émile's threat detection approach would mistreat the event that is both establisher and threat to the agent's goal. For example, for an agent having a goal of watching TV, the preconditions (or subgoals in Émile) are the TV is in the living room and the TV is not broken. However, the TV currently is broken although it is in the living room. An available plan to achieve the top-level goal are "bring the TV to the shop to have it fixed", and then "bring the TV back". In Émile, the action of "bring the TV to the shop" is to satisfy the second condition/subgoal that the "the TV is not broken", but it is considered as a threat to the first condition/subgoal that "the TV is in the living room". Émile would generate sadness for the event "the TV is in the shop", which is not logically sensible. Our approach, which uses a probabilistic planning algorithm with heuristic searching, does not encounter this problem. We do not appraise an event by considering it as an establisher or a threat to the agent's goal but by assessing how that event changes the probability of achieving the agent's goal.

Émile also does not pay attention to the way motivational states and personality affect emotions.

5.4 Models of personality

Personality plays an important role in differentiating people (Nye and Brower, 1996). It distinguishes people in the way they experience emotion and in the way they behave (Ortony, 2002). Personality is a crucial feature that can enhance the believability of computational characters (Thomas and Johnston, 1981). Loyall and Bates (1997), for example, vary the generated text for a conversational agent to reflect personality of the agent.

According to Trappl and Petta (1997), personality models, at least simple ones initially, have to be developed in order to give synthetic agents some autonomy. Especially when building believable emotional agents, one needs not only to incorporate appropriate models of emotion in response to the situation but also to incorporate personality as an individual factor that orients emotional reactions and responses (Ortony, 2002). For example, Rosis et al. (2003) incorporate personality into an agent named Greta to modify the way Greta feels and shows emotions. In this agent, personality is viewed as weights that people put on different goals. Since emotions are considered to relate to goals, de Rosis et al. assess the relationship between personality and emotion through goals.

The question is how we can incorporate all personality traits into computational characters. Constructing personalities trait by trait is only effective if the number of implemented traits is small. Fortunately, there is much empirical evidence that traits can be grouped together and can be portrayed by a small number of factors. This evidence dates back to the 1930s. Pioneered by Allport and Odbert (1936), much of the research in linguistics has been established to identify trait-descriptive words in the English language, which has been the starting point of language-based personality trait research for the last sixty years. However, this research was seriously flawed. The work of Cattell (1946) is a typical example. Fiske (1949) was the first one who presented evidence of the flaws existing in Cattell's work. Fiske then proposed five factors that accounted for the variance in personality trait descriptors. Tupes and Christal (1961) first made use of Allport and Odbert's work. Based on the work of Cattell and Fiske, Tupes and Christal completely established the five factors. Unfortunately, the result they presented in an unknown Air Force publication was not read either by the psychology or academic communities. Learning from the work of Tupes and Christal (1961), Norman (1963) published a strengthened five-factor structure for trait taxonomy, which was known as "Norman's Big Five"[1]. Personality traits that are likely to happen together are categorized into Big Five dimensions. The most popular categorization is the one developed by McCrae and Costa (1987); Costa and McCrae (1992), which uses the five dimensions: extroverted (as opposed to introverted), likeable (as opposed to antagonistic), conscientious (as opposed to negligent), openness (as opposed to new experiences) and neuroticism (as opposed to emotional stability). According to McCrae and Costa, people who are extroverted are likely to be sociable, warm and talkative; people who are likeable are likely to be forgiving, good-natured, and softhearted; and so on.

The "Big Five" classification of personality traits, however, just shows how traits can occur together. It does not provide a view on the impact of personality on a character's emotional experience and behavior. A more structural way of categorizing personality traits was presented by Rousseau (1996), which provides an excellent way of incorporating personality into computational characters. It also provides "a sufficiently rich structure based on convention architecture of an intelligent agent" (Rousseau, 1996). Personalities are classified according to different processes that an agent can perform: perceiving, reasoning, learning, deciding, acting, interacting, revealing, and feeling emotions. Each process is considered at two levels: the natural inclination that the agent has to perform the process, and the main aspect that the agent focuses on while performing the process. For example, an agent who focuses more on undesirable effects is described as pessimistic while the one who focuses more on desirable effects is described as optimistic; an agent with low level inclination of feeling emotions is considered emotionless while the one with high level of inclination of feeling emotions is considered sensitive; and so on. A summary of the dimensions of a personality can be seen in Table 5.1. Rousseau (1996) used this structure of personality to simulate agents in the Virtual Theater projects (Hayes-Roth and van Gent, 1997) with different personality. He used a rule-based system to capture the personality's influence to the behavior of agents. We base on Rousseau's model to incorporate personality into ParleE because we find this model convenient and easy to implement and to assess the influence of personality on other processes, e.g. emotion.

5.5 The ParleE implementation

In ParleE, we consider agents in a [Markov Decision Processes]-style framework, in which the state of the world is known but actions are probabilistic. When

---

[1] Appropriately, this should be Tupes and Christal's Big Five.

Process	Inclination Level	Illustrated word(s)
Perceiving	Low High	Absentminded Alert
Reasoning	Low High	Silly Rational
Learning	Low High	Incurious Curious
Deciding	Low High	Insecure Self-confident
Acting	Low High	Passive Zealous
Interacting	Low High	Introverted Extroverted
Revealing	Low High	Secretive Open
Feeling emotions	Low High	Emotionless Sensitive

Process	Focus Aspect	Illustrated word(s)
Perceiving	Expectations Reality	Imaginative Realistic
Reasoning	Undesirable effects Desirable effects	Pessimistic Optimistic
Learning	What is learned only What is known only	Gullible Intolerant
Deciding	First reaction Good decision	Impulsive Thoughtful
Acting	Anything besides the task Result of the task	Indifferent Perfectionist
Interacting	Addressee as a threat Addressee as a help	Hostile Friendly
Revealing	Lie Truth	Dishonest Honest
Feeling emotions	Self Others	Selfish Unselfish

Table 5.1: Dimensions of a personality (from (Rousseau, 1996)).

there are more than one agent situated, the actions which other agents are going to perform are also unknown. Using the information from an agent's planner and a model of the other agent, ParleE calculates intensities of emotion based on the change in probability of achieving the agent's goal. This is derived from the view of Oatley and Johnson-Laird (1987) and Reilly (1996). When exporting to a new application, ParleE reuses the agent's existing planner and information about the new world. Like Émile (Gratch, 2000), it provides a domain-independent and flexible way of generating emotions. And therefore, ParleE just fits in any application with the same framework. However, rather than Émile's threat detection approach, ParleE uses forward-chaining search within a finite depth to obtain the probability of achieving a goal. By doing this, Parle solves the problem of mistreating the event that is both establisher and threat to the agent's goal (see Section 5.3.4). ParleE also provides a generic way of incorporating personality, individual characteristics and motivational states. In ParleE, personality traits are categorized based on different processes that an intelligent agent could perform such as perceiving, reasoning, feeling emotion and so on (Rousseau, 1996). This is discussed in Section 5.5.3. Individual characteristics determine how easily the agent experience certain emotions, which will be described in Section 5.5.2.

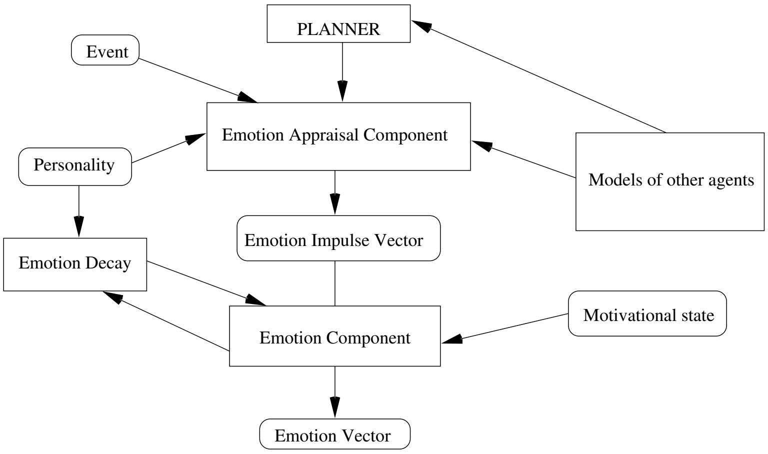


Figure 5.3: Overview of the ParleE implementation.

When an event happens, an Emotion Impulse Vector (EIV) is generated by appraising the event using the rules proposed by the OCC appraisal theory (Ortony et al., 1988) based on the agent's goals, plans and standards. An EIV contains the values of the event's impact on emotions. The EIV is then used to update the Emotion State Vector (ESV), which contains values representing intensities of emotion. This will be discussed in Section 5.6.4. An overview of the system can be seen in Figure 5.3.

The Planner produces an optimal plan to achieve the agent's goal. The planner calculates the probability of achieving that goal by searching through a number of possible states within a finite depth. The planner also incorporates

models of other agents to predict the behavior of other agents and to obtain the probability of an event to happen. These models are also used to generate emotions about the fortunes of others, named **Desire-other** emotions. The probabilities generated by the planner is then used by the Emotion Appraisal Component to calculate the EIV.

Actually, any planner that can produce the same output as described above would fit into ParleE. This allows ParleE to update with the new development within the area of probabilistic planning. This also allows other systems equipped with similar planners to incorporate ParleE. Our current planner will be discussed in Section 5.7.

The Emotion Appraisal Component takes the event, personality, plan, and models of other agents as inputs to produce the EIV as output. This component will be explained in Section 5.5.1.

The Emotion Component takes the EIV and motivational states as input and produces the updated emotion vector as the output. This component also cooperates with the Emotion Decay component to produce decayed emotions. This component is described in Section 5.5.2.

The Emotion Decay Component calculates how emotions decay taking into account the personality. The decay function is discussed in Section 5.6.3.

5.5.1 Emotion Appraisal Component

The Emotion Appraisal Component is in charge of generating emotions when an event occurs. This is done based on Ortony et al.'s appraisal theory (Ortony et al., 1988). Emotions are generated by assessing events relating to goals, expectations and standards, which is summarized in Table 5.2.

We base the emotion appraisal component on the probabilistic planning algorithm described above and on learning. The planning algorithm is to find the probability of achieving an agent's goal and to assess the significance of an event to the agent's goal. We take the difference between the probability of achieving the agent's goal before and after the event happens as the effect of the event on the goal. The learning component increases the adaptiveness of the planning algorithm. It is also used to relate the events to standards.

Ortony et al. (1988) have suggested rules to implement the OCC model in computers; these rules look like:

if DESIRE$(p, e, t) > 0$
then JOY-POTENTIAL$(p, e, t) := f($DESIRE$(p, e, t),\ I_g(p, t, e))$

where DESIRE(p, e, t) returns the desirability of an event e at time t to person p, $I_g(p, t, e)$ returns the combined effects of the global intensity variables.

We follow the suggested rules by Ortony et al.'s to trigger emotions in ParleE. We use the value of several variables to determine which emotions will arise from the occurrence of an event. These variables are: the value of the event's impact on the goal, the probability of achieving the agent's goal before and after the event happens, the goal status (unconfirmed/success/fail), the event's impact on the other agent's goal, the liking level toward the other agent, the praiseworthiness/blameworthiness of the action and who performed the action. These variables will be discussed in more detail later.

Table 5.3 summarizes the computation rules that we use to trigger emotions. These rules are elaborated as follows:

```
if the Impact of the Event on the Agent's Goal is positive then
  generate Joy
else
  generate Distress
...
```

There is no emotion **Surprise** in the OCC model (Ortony et al., 1988). We adopted the idea from Roseman's model (Roseman, 1984) that **Surprise** is associated with unexpectedness. In ParleE, **Surprise** arises only when a highly unexpected event occurs. We evaluate this unexpectedness by checking whether the probability of the event happening is smaller than 0.5.

5.5.2 The Emotion Component

The Emotion Component contains a representation of emotions and is responsible for updating the emotional state. Emotions are represented as a vector of intensities for every emotion type. Each emotion is associated with two thresholds, which will be described below.

As in Cathexis (Velásquez, 1997), we distinguish between moods and emotions by the level of arousal. We adopt Cathexis's concept of two thresholds associated with each emotion type. The first threshold, denoted by α, controls the activation of the emotion type and differentiates between moods and emotions. Emotion types with intensity level lower than their α threshold are considered as moods, and do not have as much influence as emotions on an agent's behavior. Emotion types with intensity level higher than their α threshold are considered emotions. The idea of an activation threshold is also consistent with the view of Ortony et al. (1988) in implementing emotions. The second threshold, denoted by ω, controls the saturation of the emotion type — the upper limit for intensity of the emotion type. This is to ensure that the intensity of each emotion type does not increase forever but has a limit. In our implementation, the two thresholds of each emotion type are dynamic in two ways: they vary when the physical state of the agent changes; and they vary for different agents (We consider this as the emotional characteristics of agents). This allows each agent in different physical states to experience emotion differently, and it allows different agents with different individual characteristics to experience emotion differently as well.

Emotions	**Rules**
Joy	Pleased about a desirable event
Distress	Displeased about an undesirable event
Happy-for	Pleased about an event presumed to be desirable for someone else
Sorry-for	Displeased about an event presumed to be undesirable for someone else
Resentment	Displeased about an event presumed to be desirable for someone else
Gloating	Pleased about an event presumed to be undesirable for someone else
Hope	Pleased about the prospect of a desirable event
Fear	Displeased about the prospect of an undesirable event
Satisfaction	Pleased about the confirmation of the prospect of a desirable event
Fears-confirmed	Displeased about the confirmation of the prospect of an undesirable event
Relief	Pleased about the disconfirmation of the prospect of an undesirable event
Disappointment	Displeased about the disconfirmation of the prospect of a desirable event
Pride	Approving of one's own praiseworthy action
Self-reproach	Disapproving of one's own blameworthy action
Appreciation	Approving of someone else's praiseworthy action
Reproach	Disapproving of someone else's blameworthy action
Anger	Distress + Reproach
Gratitude	Joy + Appreciation
Gratification	Joy + Pride
Remorse	Distress + Self-reproach

Table 5.2: Rules to trigger emotions from OCC model.

5.5.3 Personality

We extend Rousseau's qualitative model of personality (Rousseau, 1996) to a quantitative model. Recall that Rousseau describes personality with sixteen dimensions (see Table 5.1). A personality in ParleE is represented as a vector in 16-dimensional space:

$$Personality = (p_1, p_2, ..., p_{16})$$

where $0.0 \leq p_i \leq 1.0$

We now illustrate how this vector models an agent's personality by explaining

Emotions	I_1	old P_{goal}	new P_{goal}	Goal status	I_2	LL	Pwn	actor
Joy	+							
Distress	−							
Happy−for					+	+		
Sorry−for					−	+		
Resentment					+	−		
Gloating					−	−		
Hope			+	unconf.				
Fear			−	unconf.				
Satisfaction		+		success				
Fears−confirmed		−		fail				
Relief		−		success				
Disappointment		+		fail				
Pride							+	self
Self−reproach							−	self
Appreciation							+	other
Reproach							−	other

Table 5.3: ParleE rules to trigger emotions based on the OCC model (I_1: impact of the event to the agent's goal; old P_{goal}: the probability of achieving the goal before the event occurs; new P_{goal}: the probability of achieving the goal after the event occurs; Goal status: unconfirmed, success or fail; I_2: impact of the event to the other agent's goal; LL: the liking level of the agent toward the other agent; Pwn: the praiseworthiness of the event; actor: who causes the event to happen).

some of its components. The first component of the personality vector represents the *Inclination Level* of *Perceiving* (we symbolize it as *perceivingLevel* for convenience). The lower the value of this component is, the more absentminded the agent. The higher the value of this component is, the more alert the agent. The second component represents the *Focus Aspect* of *Perceiving* (we symbolize it as *perceivingFocus*). When the value of this component is low, the agent focuses more on expectation (the agent is more imaginative). The agent focuses more on reality (the agent is more realistic) when the value of this component is high. Other components of the personality vector are interpreted and symbolized in the same way.

The values of some components of the personality vector are used to modify emotion intensities by influencing the values of the emotion variables. Several components of the personality vector are used to decide the learning rate and how the agent displays emotions.

5.5.4 Motivational states

The human body is complicated. Its physical state can be described by a number of variables such as heart beat, body temperature, level of thirst and so on. As in FLAME (El-Nasr et al., 2000), we consider four variables to describe our agent's physical state. They are level of hunger, fatigue, thirst, and pain. These variables can also be considered as motivational states because their high values will generate a goal for the agent, viz. the agent wants to reduce these values. This type of goal is called *replenishment goals* in the OCC model (Ortony et al., 1988).

In ParleE, the motivational state of an agent changes during the actions of the agent. The level of hunger, fatigue and thirst increases every time the agent has performed an action; the level of pain increases when the agent is hit. The level of hunger decreases when the agent eats; the level of thirst decreases when the agent drinks; and the levels of fatigue and pain decrease when the agent takes a rest.

Motivational states affect the emotions of agents in ParleE by changing the α threshold for each emotion. When the level of fatigue gets higher, the α thresholds for negative emotions decreases (when the agent is tired, his/her negative emotions seem to be easily activated) and the α thresholds for positive emotions increases. High levels of pain tend to decrease the α threshold for emotion fear, while high levels of hunger and thirst tend to increase the α thresholds for all emotions.

5.6 Intensity of emotions

To compute the intensities of emotions, Ortony et al. (1988) have proposed many intensity variables including *desirability*, *praiseworthiness*, *appealingness*, and *unexpectedness*. Our ParleE implementation uses six main variables: *impact of an event on goal* and *goal importance* (to compute the *desirability*), *probability of achieving a goal* and *probability of an event occurring* (these two probabilities are equivalent to the *unexpectedness*), *praiseworthiness* (value of an action), and *liking* (between agents). These variables are used to calculate the Emotion Impulse Vector (EIV) (cf. Section 5.5) rather than the emotion vector. For each emotion type, we first calculate the value of the impulse with regard to each goal. The sum of all impulses for all goals is taken as the final impulse for that emotion type. The impulse vector is then used to update the decayed emotion vector.

We will now discuss these emotional variables, followed by the EIV, the decay function and the integration of the EIV into the emotion vector.

5.6.1 Emotion variables

Goal importance is a major factor for determining the intensity of goal-related emotions. It is denoted as $Import(goal)$. In ParleE, we only consider top-level goals rather than any subgoals that arise in the plans developed to achieve top-level goals. Thus, if an event happens that affects the goal but does not make the goal succeed or fail, we consider that it partially affects the goal. Because of this approach, we are only concerned with the intrinsic importance of the goal (as defined in (Gratch, 2000) as "the reward (utility) an agent receives from achieving the goal"). We do not consider a goal's extrinsic worth (how a goal promotes other goals) as we do not have subgoals in ParleE. For the intrinsic importance of the goal, we assign predefined values as in (Gratch, 2000).

The probability of achieving a goal, denoted by $P(goal)$, is adopted from the planner, which is described in Section 5.5. This planner computes the optimal probability of a goal success from a state of the world. We take that value as the probability of achieving the goal. The advantage of a plan-based emotion model over other approaches is that the planning algorithm allows a generic (domain-independent) way of calculating this probability.

The probability of an event occurring, which is denoted by $P(event)$, is defined in ParleE as follows:

– If the agent is waiting for the result of his/her own action then $P(event)$ is the probability of an outcome of an action (cf. Section 5.7). This probability is first assigned some predefined value. It is then updated by learning from what actually happens, which will be discussed in Section 5.8.

– If the agent is waiting for another agent's action then $P(event)$ is the probability of another agent's action at the current world state s times the probability of an outcome of that action. The probability of another agent's action is calculated from this agent's model of another agent (cf. Section 5.7).

The impact of an event on a goal in this model is the difference between the probability of achieving the goal before and after the event happens (denoted by $Impact(event, goal)$). Thus, if the value of impact is negative, that is if the probability decreases after the event happens, then the event is undesirable for this goal; if the value of impact is positive, that is if the probability increases after the event happens, then the event is desirable for this goal.

$$Impact(event, goal) = P_{after}(goal) - P_{before}(goal)$$

Praiseworthiness is used to evaluate how the agent's action meets standards of behaviors. We use the idea of FLAME (El-Nasr et al., 2000) about learning values of actions to form the standards in appraising emotions. The variable is denoted by $Praiseworthiness(action)$. Another agent can provide feedback on the agent's action. In ParleE, the feedback is a point which ranges from -1.0 to 1.0. The average feedback point for an action over the time is taken as the value of *Praiseworthiness*(*action*).

Liking, denoted by *LikingLevel*(*anotherAgent*), is a variable that contributes to the intensity of **Desire-other** emotions. In ParleE, the agent maintains a dynamic liking level toward each another agent. Each liking level ranges from −1.0 to 1.0. Starting with a neutral attitude toward another agent (the value of liking level is 0.0), the agent's liking level changes when another agent has performed an action that affected the agent's goal. The agent's liking level increases if it is a desirable event and decreases if it is undesirable. The agent's liking level toward another agent is updated as follow when the preceding event is caused by another agent:

$$\mathit{Desirability}(\mathit{event}) = \sum_{\mathit{goal}} \mathit{Import}(\mathit{goal}) \cdot \mathit{Impact}(\mathit{event}, \mathit{goal})$$

$$\mathit{LikingLevel}_{t+1} = \max(-1.0, \min(1.0, \mathit{LikingLevel}_t + 0.1 \cdot \mathit{Desirability}(\mathit{event})))$$

5.6.2 The Emotion Impulse Vector

We now present formulas to calculate the intensity of emotion impulses. These formulas are based on the ones presented by Gratch in Émile (Gratch, 2000) and Elnasr et al. in FLAME (El-Nasr et al., 2000). In Émile, however, the event expectation is not used in calculating the intensity of emotions. In FLAME, there is no clear distinction between the expectation of a goal succeeding and an event happening. Our formulas presented below will take into account these issues.

Hope arises from the belief that a goal is going to succeed and it is important that the goal does not fail. In ParleE, when the probability of achieving the goal is higher than 0.5, the goal is believed to succeed. On the contrary, **Fear** arises from the belief that a goal is going to fail (probability of achieving the goal is lower than 0.5). Therefore, the probabilities of the goal succeeding directly relate to the intensity of **Hope** and **Fear**. In addition, the probability of achieving a more important goal seems to have more influence on the intensity of **Hope** and **Fear**. We follow the formulas proposed by Gratch in Émile (Gratch, 2000) to define the intensity of an emotion as the product of the value of variables that contribute to that emotion. To capture those characteristics of **Hope** and **Fear**, we propose the following formulas:

$$\mathit{Hope} = \sum_{\mathit{goal}} \mathit{Import}(\mathit{goal}) \cdot (P(\mathit{goal}) - 0.5)$$

$$\mathit{Fear} = \sum_{\mathit{goal}} \mathit{Import}(\mathit{goal}) \cdot (0.5 - P(\mathit{goal}))$$

Happiness arises when a goal succeeds or becomes more likely to succeed (the value of the impact from the event on the goal is positive). **Sadness** arises when the goal is more likely to fail (the value of the impact from the event on the goal is negative). The value of the impact of an event and the importance of the goal are the two factors contributing to the intensity of **Happiness** and **Sadness**. Moreover, the unexpectedness of the event also plays a role in determining the intensity of the two emotions, and it is likely to have different effect on **Happiness** and **Sadness** (Price et al., 1985; El-Nasr et al., 2000). Following the approach in FLAME (El-Nasr et al., 2000), we raise the unexpectedness of the event by a power of 0.5 and 2 in the formulas calculating the intensity of **Happiness** and **Sadness** respectively. The unexpectedness of the event is determined by $1-P(event)$. The intensities of **Happiness** and **Sadness** impulses are modelled as follows:

$$Happiness = \sum_{goal} Impact(event, goal) \cdot Import(goal) \cdot (1 - P(event))^{0.5}$$

$$Sadness = \sum_{goal} Impact(event, goal) \cdot Import(goal) \cdot (1 - P(event))^{2}$$

Anger arises when another agent performs some action that is undesirable for the agent. The intensity of **Anger** is also defined using the value of the impact of the event, the importance of the goal and the unexpectedness of the event:

$$Anger = \sum_{goal} Impact(event, goal) \cdot Import(goal) \cdot (1 - P(event))^{0.5}$$

The intensity of **Gratitude**, **Gratification** and **Remorse** is defined in a similar way.

There is no emotion **Surprise** in the OCC model (Ortony et al., 1988). We adopted the idea from Roseman's model (Roseman, 1984) that **Surprise** is associated with unexpectedness. In ParleE, **Surprise** arises only when a highly unexpected event occurs. We evaluate this unexpectedness by checking the condition $P(event) < 0.5$. The intensity of the **Surprise** impulse is determined by:

$$Surprise = 0.5 - P(event)$$

The **Happy_for** emotion arises when an agent that is liked is happy (when the value of the liking level is positive). We consider this emotion as an example of **Desire-other** emotions. The intensity of **Happy_for** is determined by the agent's liking level toward another agent and the intensity of that other agent's **Happiness** (derived from the models of other agents):

$$\mathit{Happy\_for} = \mathit{LikingLevel}(\mathit{anotherAgent}) \cdot (\mathit{anotherAgent's\_Happiness})$$

The intensity of **Sorry_for**, **Resentment** and **Gloating** is determined in a similar way.

Pride and **Shame** are two emotions related to standards of behavior. The value of these depends on the praiseworthiness of the performed action (if the praiseworthiness is positive, then **Pride** arises; if it is negative, then **Shame** arises):

$$\mathit{Pride} = \mathit{Praiseworthiness(action)}$$

$$\mathit{Shame} = -\mathit{Praiseworthiness(action)}$$

The intensity of **Appreciation** and **Reproach** is calculated in a similar way.

Relief arises when there is a disconfirmation of the fear that a goal is going to fail. We therefore determine the intensity of **Relief** based on the intensity of **Fear** from last time:

$$\mathit{Relief} = \mathit{previous\_Fear}$$

The intensity of **Satisfaction**, **Fear-confirmed** and **Disappointment** is determined in a similar way.

5.6.3 Decay function

It is important to derive a reasonable decay function for emotions over time. It seems that emotions decay slower when their intensity levels are lower. With the definition of mood above, it is reasonable that moods persist much longer than high intensity emotions. Moreover, negative emotions tend to differently compared to positive emotions. Personality also influences how emotions decay. Taking these considerations into account, we propose a decay function as follows:

$$\Psi(E_{i,t+1}) = \Phi(E_i, \mathit{personality}) \cdot E_{i,t}$$

where $E_{i,t}$ is the intensity of an emotion E_i at time t, $\Phi(E, \mathit{personality})$ is a function of emotions that generates different decay rates for each emotion with regard to the agent's personality. In ParleE, the *Inclination Level* of the agent's *feeling emotions* (see Table 5.1 and Section 5.5.3) affects the function

$\Phi(E, \mathit{personality})$. This *Inclination Level* of the agent's *feeling emotions* determines if the agent is more emotionless or more sensitive. The more sensitive the agent is, the slower their emotions decay. The decay rate for an emotion E_i is modified as follows:

$$\Phi(E_i, \mathit{personality}) = \mathit{decayFactor}(E_i) \cdot \mathit{feelingLevel}(personality)$$

5.6.4 Updating emotions

To update emotions over time, one has to consider the previous emotion states, the decay function, and the values of emotion impulses.

The intensity of emotion E_i is the decayed value of the intensity at previous time plus the reduced value of emotion impulse I_i. As the value of emotion impulse I_i is affected by the intensities of emotions at previous time, it is proportionally reduced by the sum of effect values from all emotions to emotion E_i. For example, the presenting level of happiness may reduce the effect of an undesirable event. Finally, the intensity of emotion E_i is limited to the upper threshold ω_i by the function min:

$$E_{i,t+1} = \min(\omega_i, \Psi(E_{i,t}) + I_{i,t+1} \cdot (1 - \sum_j \frac{S_{j,t}}{\omega_j} \cdot M_{j,i}))$$

$$S_{j,t} = \begin{cases} 0 & \text{if} \quad E_{j,t} < \alpha_j \\ E_{j,t} & \text{if} \quad E_{j,t} \geq \alpha_j \end{cases}$$

where $E_{i,t}$ is the intensity of emotion E_i at time t, ω_i is the upper threshold for the intensity of emotion E_i , $\Psi(E_{i,t})$ is the decay function as described above, $I_{i,t+1}$ is the intensity of emotion impulse for emotion E_i at time $t+1$, $M_{j,i}$ is the effect factor from emotion E_j to emotion E_i. To assure that the effect of an impulse on an emotion has as minimum 0 and as maximum the intensity of the impulse itself, we introduce the following constraint on $M_{j,i}$:

for all i

$$0 \leq \sum_j M_{j,i} \leq 1.$$

Since for all j, we have

$$0 \leq E_{j,t} \leq \omega_j$$

$$0 \leq S_{j,t} \leq \omega_j$$

holds as well, which implies

$$0 \leq \frac{S_{j,t}}{\omega_j} \leq 1$$

Consequently,

$$0 \leq \frac{S_{j,t}}{\omega_j} \cdot M_{j,i} \leq M_{j,i}$$

and therefore

$$0 \leq \sum_j \frac{S_{j,t}}{\omega_j} \cdot M_{j,i} \leq 1$$

Hence we obtain that

$$0 \leq E_{i,t+1} \leq \omega_i$$

and

$$0 \leq E_{i,t+1} \leq \Psi(E_{i,t}) + I_{i,t+1}$$

5.6.5 How personality affects intensity of emotions

We now present formulas that show how personality influences the way an agent experiences emotion.

If the agent focuses more on expectations when perceiving (the value of *perceivingFocus* is close to 0.0), then the value of the expectation variables $P(goal)$ and $P(event)$ have higher influences on emotion intensities:

$$\mathit{newExpectation} = \mathit{Expectation}^{(\mathit{perceivingFocus}+0.5)}$$

If the agent focuses more on undesirable effects (the value of *reasoningFocus* is close to 0.0), the impact variable, when negative, will have more influence on the intensities of negative emotions. If the agent focuses more on desirable effects, then the impact variable, when positive, will have more influence on the intensities of positive emotions:

$$\mathit{newImpact} = \begin{cases} -(-\mathit{impact})^{(\mathit{reasoningFocus}+0.5)} & \text{if} \quad \mathit{impact} < 0 \\ \mathit{impact}^{(1.5-\mathit{reasoningFocus})} & \text{if} \quad \mathit{impact} \geq 0 \end{cases}$$

The agent with a lower inclination level of feeling emotions (less sensitive) will have lower intensity of emotions for the same event compared to the agent with a higher inclination level of feeling emotions (more sensitive):

$$\mathit{newImpulse} = \mathit{impulse}^{(1.5-\mathit{feelingLevel})}$$

The agent that focuses more on others when feeling emotions (the value of *feelingFocus* is closed to 0.0) will have higher intensities for **Desire-other** emotions than the agent who focuses more on self when feeling emotions:

$$\mathit{newDesire\text{-}other\_impulse} = \mathit{Desire\text{-}other\_impulse} \cdot \mathit{feelingFocus}$$

The value of *revealingLevel* is used in the emotion displaying component and the value of *learningFocus* influences the learning rate.

```
(operator LOAD
  (params (<object> CARGO) (<rocket> ROCKET) (<place> PLACE) (<agent> AGENT))
  (preconds (at <rocket> <place>) (at <object> <place>)
    (at <agent> <place>) (is-acting <agent>))
  (effects (0.9 (in <object> <rocket>) (del at <object> <place>))
    (0.1) ))

(operator UNLOAD
  (params (<object> CARGO) (<rocket> ROCKET) (<place> PLACE) (<agent> AGENT))
  (preconds (at <rocket> <place>) (in <object> <rocket>)
    (in <agent> <rocket>) (is-acting <agent>))
  (effects (0.9 (at <object> <place>) (del in <object> <rocket>))
    (0.1) ))

(operator MOVE
  (params (<rocket> ROCKET) (<from> PLACE) (<to> PLACE) (<agent> AGENT))
  (preconds (has-fuel <rocket>) (at <rocket> <from>)
    (in <agent> rocket) (is-acting <agent>))
  (effects (0.8 (at <rocket> <to>) (del has-fuel <rocket>) (del at <rocket> <from>))
    (0.2) ))
```

Figure 5.4: Rocket domain for multi-agent planning defined in STRIPS.

5.7 The Planner

We extend Blum and Langford's Probabilistic GraphPlan algorithm (Blum and Langford, 1998) to support planning in a multi-agent environment. Probabilistic GraphPlan uses a graph structure to solve STRIPS-style planning problems. It allows the calculation of a plan's probability of success from a specific state of the world. We then take the change in a goal's probability of success when an event happens as the impact of the event on the goal.

5.7.1 STRIPS-style planning

STRIPS (Fikes and Nilsson, 1971) is widely used by planning algorithms although it is not ideal. In STRIPS domains, the state of the world is a set of true facts. A *goal* is a set of facts that we want to be true. An *operator* represents a legal action that may be performed. An operator has conjunctive preconditions, a list of facts to be added and a list of facts to be deleted. Blum and Langford extend STRIPS to have probabilistic actions. Each operator then has conjunctive preconditions and a set of possible outcomes. Each outcome is defined as a set of add and delete effects, each set having an associated probability. For example, a cook action might require that there is food, and with 80% probability of having a good meal, and 20% probability of having a bad meal. An example of the rocket domain can be seen in Figure 5.4.

5.7.2 Blum and Langford's planning

In a probabilistic planning algorithm, the aim is to find a plan with the optimal probability of achieving the goal. Blum and Langford's Probabilistic Graphplan planning algorithm (Blum and Langford, 1998) compiles a planning problem into a polynomial-sized, directed graph which is broken up into levels. This graph is called *planning graph*. The planning graph for the rocket domain in Figure 5.4 is presented in Figure 5.5.

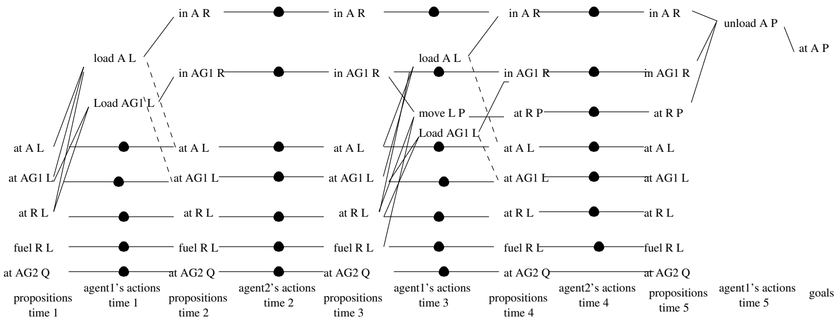


Figure 5.5: Planning graph for the probabilistic rocket problem.

Each node in the first level of the planning graph represents a proposition in the initial conditions. Each node in the next level represents an action that might be performed at time 1. The preconditions for this action must be present in the initial conditions. The next level of the graph contains all propositions that might be true at time 2. These propositions are the add-effects of all actions in the previous action level including *no-ops*. A *no-op* action simply propagates the proposition forward in time. Nodes in the next level represent actions that might be performed at time 2, and so on.

The algorithm proposed by Blum and Langford uses Top-Down Dynamic Programming to calculate the probability of achieving the goal. This algorithm is described in Figure 5.6. It traverses down from the current node to every possible successive node within a finite depth to find the optimal path.

5.7.3 Model of other agents

To support planning in a multi-agent environment, we incorporate models of other agents to predict their behavior. Each model of another agent contains goals and a planner. We assume that the goals of the other agent are known. Based on the information about their goals, their planner is set up. This planner is then used to predict their plan. The probability that the other agent performs an action is derived from the predicted plan. To trigger emotions about the other agent's fortunes, the desirability of an event to their goals is also derived from the predicted plan.

We use the approach proposed by Sen and Weiss (1999) to represent agents that model the behavior of other agents. They propose three type of agents: 0-level agents, 1-level agents and 2-level agents. In our implementation, we use $d + 1$ types of agents: 0-level, 1-level, ..., d-level agents, d is an integer with $d \geq 0$. A d-level agent is defined as follows:

```
DPsolve(state $s$, time $t$): Compute $value(s,t)$= probability of achieving the goal within the time window starting from state$s$ at time step $t$.

1. if $t = t_{max}$ then return 0 if the goal is not attained, otherwise return 1.
2. If already-visited($s$,$t$), then return the previously-computed value.
3. For each possible action $a$.
  (a) For each possible state $s'$ that could return from taking action $a$ in state $s$, recrusively call DPsolve($s'$,$t+1$).
  (b) Let $valueA(s,a,t) = prob(s',s,a) \cdot value(s',t+1)$, where $prob(s',s,a)$ is the probability that state $s'$ is the successive state of state $s$ if action $a$ is taken.
  (b) Let $value(s,t)$=$max_a valueA(s,a,t)$. Return this quantity, after first storing it in case we ever visit $s$ at time $t$ in a later recursive call.
```

Figure 5.6: Top-down Dynamic Programming for calculating the probability of achieving a goal.

- A d-level agent is an agent that models other agents as $(d-1)$-level agents.
- A $(d-1)$-level agent is an agent that models other agents as $(d-2)$-level agents.

  ...

- A 2-level agent is an agent that models other agents as 1-level agents. That is, this agent views other agents as agents which are modeling other agents as 0-level agents.

  ...

- A 0-level agent does not model the behavior of other agents.

d-level agents search for their optimal plan within the depth of d. 0-level agents search for their optimal plan within the depth of 0, which means that they just check if their goals are present. This approach is similar to the M^* algorithm proposed by Carmel and Markovitch (1996). Their algorithm has been designed to model an opponent in a two player zero-sum game. It maximizes the player's payoff by searching up to a depth of d, while assuming that the opponent is minimizing the player's payoff by searching up to a depth of $d-1$. That is to assume that the opponent is assuming that the player is maximizing their payoff by searching up to a depth of $d-2$ and so on.

5.7.4 Our planning algorithm

Our implementation supports planning with multiple goals. Each goal is associated with three values: the priority of the goal, the importance of the goal, and the deadline for the goal to be obtained. The goal with the highest priority will be planned first. The level of importance is used to determined how much the obtainment of the goal contributes to the trigger of emotions. Finally, the goal must be obtained before the deadline to be considered successful.

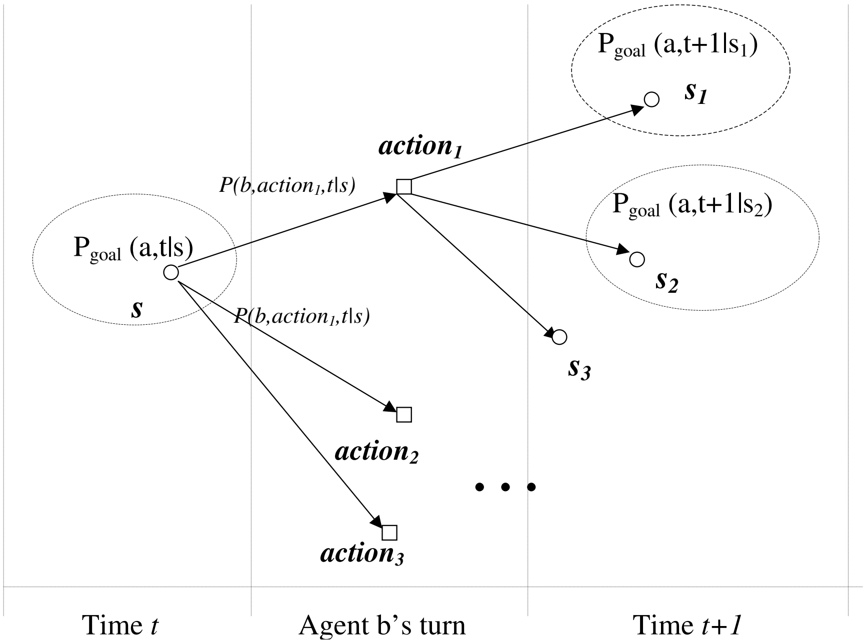


Figure 5.7: Possible paths to search for an optimal plan.

In our extension of the planning algorithm proposed by Blum and Langford (1998), the possible paths that we will follow to find the optimal plan is illustrated in Figure 5.7. We denote the probability for an agent a to achieve his/her goal at time t and the state of the world s as $P^d_{goal}(a,t|s)$ by searching through a depth of d.

If $d = 0$, $P^0_{goal}(a,t|s)$ is calculated as follows:

$$P^0_{goal}(a,t|s) = \begin{cases} 0 & \text{if the goal is not present} \\ 1 & \text{if the goal is present} \end{cases}$$

If $d > 0$, $P^d_{goal}(a,t|s)$ is calculated as follows, provided it is agent b's turn to act:

$$P^d_{goal}(a,t|s) = \sum_{i=1}^{n} P^{d-1}(b, action_i, t|s) \cdot P^d_{goal}(a,t|s, action_i)$$

where $P^{d-1}(b, action_i, t|s)$ is the probability of an agent b to perform an action i at time t and the state of the world s assuming that agent b is searching for his/her optimal plan within the depth of $d-1$; $P^{d}_{goal}(a, t|s, action_i)$ is the probability for an agent a to achieve his/her goal at time t and the state of the world s provided action i is performed by searching through a depth of d. $P^{d}_{goal}(a, t|s, action_i)$ is calculated as follows:

$$P^{d}_{goal}(a, t|s, action_i) = \sum_{s'} P(s'|action_i, s) \cdot P^{d-1}_{goal}(a, t+1|s')$$

where s' is a possible state of the world with probability $P(s'|action_i, s)$ if action i is performed at the state of the world s.

We use a probabilistic approach to select an agent's action to avoid local minima. For an agent a, an action i is selected with probability:

$$P^{d}(a, action_i, t|s) = \begin{cases} \frac{P^{d}_{goal}(a,t|s,action_i)}{\mathcal{P}} & \text{if} \quad \mathcal{P} > 0 \\ \frac{1}{n} & \text{if} \quad \mathcal{P} = 0 \end{cases}$$

$$\mathcal{P} = \sum_{j=1}^{n} P^{d}_{goal}(a, t|s, action_j)$$

5.8 Learning components

There are two learning components in the system to make the agent more adaptive. The agent gradually learns the likelihood of an action's outcome and the values of actions (standard of behavior).

The first component is used to learn the likelihood of outcomes of an action. Initially, we assign some pre-determined values for the probability of each outcome. We also assign a presumed value for the number of previous observations. This number of previous observations is determined by the learning rate. Thus, the higher the value of this number of counts, the longer it takes to learn new likelihood. We then update these probabilities each time the action is performed through new observation of the actual outcome. Suppose probabilities of outcomes of an action are P_1,P_2,...,P_m. The current number of observations is n. The actual outcome of the action this time is i. Then the probabilities are updated as follows:

$$newP_j = \begin{cases} \frac{P_j \cdot n+1}{n+1} & \text{if} \quad j = i \\ \frac{P_j \cdot n}{n+1} & \text{if} \quad j \neq i \end{cases}$$

$$new\_n = n + 1$$

The second component is to learn values of actions. For each action that the agent performed, the user can choose to feedback with a point from −1.0 (very bad feedback) to 1.0 (very good feedback). The average point over time is used as the value of the action.

5.9 Illustration

We now use a simple domain to illustrate ParleE. A more sophisticated application will be discussed in Chapter 7.

The domain we use here is constructed as follows. The agent and the user live in a house and share a car. Whenever the agent is hungry, a goal of feeding himself with food is initiated. Several actions are available for the agent to achieve the goal: "go shopping", "buy food" and "unload the food from the car". The user can help the agent or can perform some other actions that may prevent the agent from achieving his goal, such as "drive the car to work". The interaction between the agent and the user looks like this:

```
FACT: (in-car bread car1) (at-home car1)
    Choose an action: (−1 to exit)
        0: No Action
        1: (unload-food bread car1)
        2: (go-shopping car1)
        3: (go-working car1)
    You choose: 0
Agent did: (unload-food bread car1)
    case 1 (with probability of 100 %) happened; effects: (in-fridge bread)
FACT: (at-home car1) (in-fridge bread)
```

During the interactions, the agent's emotions are displayed in the 3D face (see Figure 5.8). This shows how the agent is capable of expressing his emotions in response to an event.

Figure 5.8 shows three versions of the agent with different personalities: a neutral, an optimistic and a sensitive one. The scenario was as follows: "the agent goes to the shop and buys bread. He brings the bread home. The user eats his bread." The neutral version of the agent gets angry after the user eats his bread. As the optimistic agent tends to ignore this negative event, he is still happy with what has happened before. The sensitive agent gets very angry with the user. This shows that agents with different personalities respond differently to the very same event.

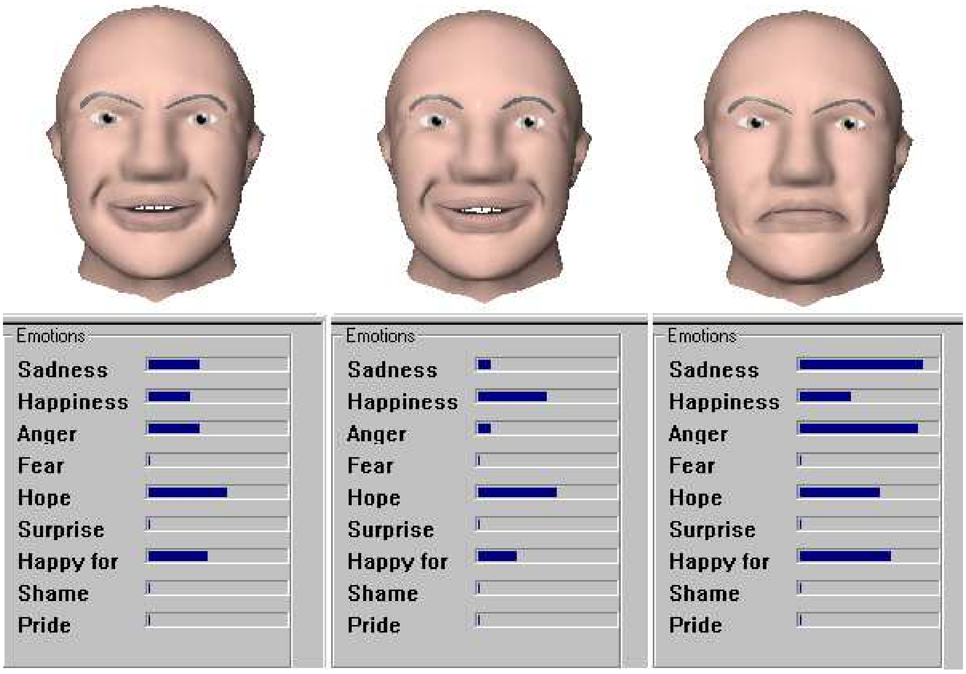


Figure 5.8: The neutral version of the agent (left) expresses facial expressions compared to the optimistic one (middle) and sensitive one (right) after the user eats his bread.

5.10 Conclusion

In this chapter, we describe ParleE, a quantitative, flexible and adaptive computational implementation of emotions for an embodied agent situated in a multi-agent environment. It is an implementation in which event appraisal is based on learning and a probabilistic planning algorithm. ParleE is based on a number of existing implementations of emotions. Nevertheless, ParleE has some significant properties of its own. The main novelties have to do with the way it uses forward-chaining search within a finite depth to obtain the probability of achieving a goal; the way it uses models of other agents' plans and goals to predict their behavior and set up expectations about the likelihood of events; and the way it incorporates personality, individual characteristics and motivational states in the implementation.

There are so many theories of emotion that it is hard for a computer scientist to choose an appropriate one to follow when implementing emotions. Moreover, there are continuous debates among the proponents of different views on emotions about the correctness of these theories. Therefore, the "ease-to-implement" is one of the main criteria for a computer scientist to select an appropriate theory of emotion. Two psychological models of emotion, Roseman's one (Roseman, 1984; Roseman et al., 1990, 1996) and that by Ortony, Clore and Collins (OCC) (Ortony et al., 1988), are used most by computer scientists because they are easy to implement on computers. Both models fit within the *cognitive* view on emotions. Like many of these and other implementations, ParleE generates emotions based on the OCC model. The OCC model of emotions provides a clearer view on emotions than Roseman's model. In Roseman's model, events are appraised only according to goals. This way, attitude and standard related emotions such as like/dislike, anger are not defined in a reasonable way. The OCC model treats these emotions more appropriately by appraising events based on goal, standard and preference. As the OCC model of emotions, like other theories of emotion, is a matter of dispute among psychologists, we do not claim for the trueness of generated emotions in ParleE.

As a direction for future research, fuzzy logic can be implemented in ParleE to eliminate the hard border between negative and positive, between hope and fear, and so on. Love and hate are the two emotions from the OCC model that are not implemented in ParleE. Therefore, it is useful to find a suitable representation of an agent's appealingness to objects in order to trigger these two emotions. Finally, a more sophisticated belief component would improve the believability of the implementation.

Chapter 6

From Emotions to Emotional Facial Expressions

> *"The face the index of a feeling mind."*
>
> – *George Crabbe, "Tales of the Hall"*

6.1 Introduction

Recall from Chapter 4 that facial movements play an important role in interpreting spoken conversation and emotions. They occur continuously during social interaction and, particularly, in conversations. They include lip movements during speech, conversational signals, emotion displays (emotional facial expressions) and manipulators to satisfy biological needs (Ekman, 1989). Lip movements participate in the articulation of speech. Conversational signals are movements to accentuate or emphasize speech, or to provide feedback from a listener. They can occur on pauses due to hesitation or to signal punctuation marks (such as a comma or an exclamation mark). They are used to help the interaction between the speaker and the listener. Emotion displays/emblems are movements to express emotions that are being referenced or currently being experienced by the speaker. Manipulators are movements to satisfy biological requirements of the face, e.g., blinking.

In this chapter, we discuss the problem of generating emotional facial expressions from emotions. It has been believed for a long time that there exists a relationship between facial activity and emotional state. The pioneer publication of Darwin (1872/1965), "The expression of the emotions in man and animals", has reinforced this belief. Much recent research has strengthened the idea that emotions are expressed on faces. A typical example is the descriptive work by

Ekman and Friesen (1975), which discusses how several emotions as well as their blends are displayed on the face. We want to base on this work to map emotion representations onto the contraction level of facial muscles. We focus on two aspects of generating emotional facial expressions. First, we want to take into account the continuous changes in expressions of an emotion depending on the intensity by which it is felt. Secondly, we want to find a way to specify combinations of expressions due to more than one emotion, i.e., blends, in accordance with the literature mentioned. We have found that a fuzzy rule-based system is suitable for these requirements because it allows us to incorporate qualitative as well as quantitative information. Fuzzy rules can capture descriptions which are described in natural language as well as vague concepts like "slight sadness", "more intense sadness", etc. Moreover, the fuzzy rule-based approach can assure the smooth mapping between emotions and facial expressions.

Following Ekman and Friesen (1975), we consider the following six emotions: **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust** and **Surprise**. These are said to be universal in the sense that they are associated consistently with the same facial expressions across different cultures. Ekman and Friesen also describe in detail what the expressions for these emotions and certain blends look like. Emotion feelings may differ in intensity. In (Ekman and Friesen, 1975) it is pointed out how for each of the basic emotions the expression can differ depending on the intensity of the emotion. It is therefore important for us to build our system on a representation that takes intensities into account. We have used their descriptions as the basis for our fuzzy rules.

In Section 6.2 we review psychological work on the relationship between emotion and facial activity. Most of the research on this relationship follows one of three main views: the *basic emotions view*, the *cognitive view* and the *dimensions view*. While the *basic emotions view* assumes that there is a small and limited set of emotions that can be distinguished discretely from one another by emotional facial expressions, the *cognitive view* and the *dimensions view* argue that the facial activity depends on many factors behind the affective process rather than discrete emotions. We ourselves find the *basic emotions view* most useful in simulating facial expressions from emotions. In Section 6.3 we review existing work on simulating emotional facial expressions on computers. Section 6.4 explains how a fuzzy-rule based system works. This section starts with the introduction of fuzzy sets and fuzzy logic. Based on fuzzy sets and fuzzy logic, a set of fuzzy "if-then" rules can be used to simulate various natural processes. In Section 6.5 we give an overview of the complete system that we have implemented. Our system consists of two fuzzy rule-based systems in order to convert from emotion intensities to facial contraction levels, which are used to generate emotional expressions on a 3D face model. We then discuss both fuzzy rule-based systems in Section 6.6 in more detail. Finally, some experimental results are presented in Section 6.7.

6.2 Emotions and Facial Expressions

Recall from Chapter 5 that there are four main perspectives in research on emotions: the *Darwinian*, the *Jamesian*, the *cognitive*, and the *social constructivist.* Similarly, there are different views on the relationship between emotions and facial activity. The most popular one among them is the *basic emotions view*, which assumes that there is a small set of emotions that can be distinguished discretely from one another by facial expressions. We have based our system described in this chapter on this view. However, the studies within the *basic emotions view* have also been attacked most by researchers who take other views.

6.2.1 The basic emotions view

According to Kappas (2003), the proponents of the *basic emotions view* (e.g., Tomkins, 1962, 1963; Ekman and Friesen, 1975; Ekman, 1982; Izard, 1971, 1997) assume that there is a small set of basic emotions that can be distinguished discretely from one another by facial expressions. For example, when people are happy they smile, and when they are angry they frown. Opinions differ on what it means for an emotion to be called basic. Russell and Fernández-Dols (1997) summarize this discussion as follows:

> "Each basic emotion is genetically determined universal and discrete. Each is a highly coherent pattern consisting of characteristic facial behavior, distinctive conscious experience (a feeling), physiological underpinnings, and other characteristic expressive and instrumental actions."

Russell and Fernández-Dols (1997) also summarize other assumptions and implications of the *basic emotions view.* Some important ones among them are:

- There exists a prototypical, innate, and universal expression pattern for each of the basic emotions. The expression pattern, unless hidden or masked, occurs when the basic emotion occurs. Therefore, without being successfully masked or inhibited, the emotional state of an individual is observable from the face.
- "Any state lacking its own facial signal is not a basic emotion."
- "All emotions other than the basic ones are subcategories or mixtures (patterns, blends, combinations) of the basic emotions."

In supporting the *basic emotions view*, there is considerable evidence (see Ekman, 1982) showing that distinct prototypical facial signals corresponding to the six basic emotions of happiness, sadness, surprise, disgust, anger and fear can be reliably recognized across a variety of cultures.

Taking the *basic emotions view* as a central point, Ekman proposed a neurocultural model (Ekman, 1972, 1973, 1977) to explain the communalities and

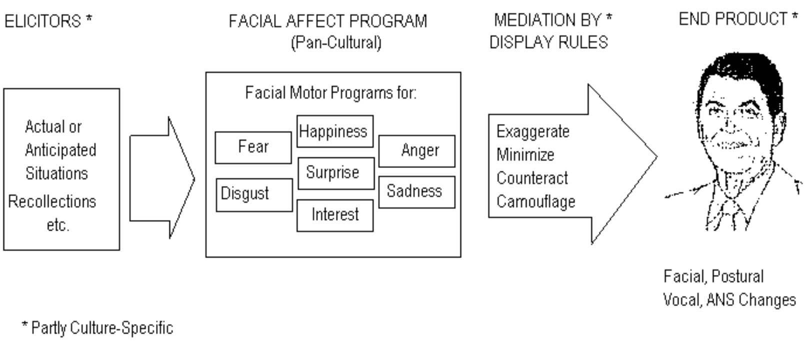


Figure 6.1: Ekman's view on how emotions and expressions are related (depicted by Fridlund, 1994).

variations in human facial displays. This model is depicted by Fridlund (1994) as in Figure 6.1. According to Fridlund,

> "in this model, everyday faces result from innate, prototypical facial expressions that read out emotional state, but which can be modified by socialization."

In the other words, the "facial affect program" of the model establishes an universal and innate correspondence between the emotion state and facial patterns. Blended emotions could occur which may result in the blending in facial patterns. Before being displayed on the face, the facial pattern may be mediated by so-called "cultural display rules". According to Fridlund (1994), these rules

> "can block or attenuate a facial expression of emotion, or force a 'masking' expression that obscures even supplants an emotional state."

Fridlund has also criticized this model in two points. First, there is no criteria in the model for determining when the facial affect program is activated, when a blend of emotions is being elicited, and when a cultural display rule is operative. Second, the model left the relation of facial display to emotion ambiguous. While the discovery of basic emotions are based on the findings of universal faces, the "cultural display rules" make this discovery ambiguous unless there are clear criteria for determining when the cultural display rules are not applied.

The *basic emotions view* itself has also received many attacks from researchers from other views. Particularly, Russell (1997), from the *dimensions view*, has presented several reasons to express his disagreement with the *basic emotions view*, e.g. with respect to the concept of discrete emotions, the universality of expression of basic emotions, and so on. However, these arguments themselves are not unquestionable. For example, with respect to the concept of discrete emotions, Russell argued that he had not seen any evidence about that; about the ability of children to discriminate different patterns of facial movement, Russell believed that it is based on pleasure and arousal rather than discrete emotions. Probably, the strongest, and most convincing, argument is the one claiming that without context, the facial expressions are ambiguous. This argument is also supported by Fernández-Dols and Carroll (1997). Therefore, Russell and Fernández-Dols (1997) maintain that the findings within the *basic emotions view* have not yet allowed researchers to fully understand facial expressions.

6.2.2 Other views on the relationship between emotions and facial activity

The *cognitive view* on emotional facial expressions was proposed by researchers working on emotions from the *cognitive* perspective such as Arnold (1960) and Scherer (1992). As discussed in Chapter 5, the *cognitive* perspective assumes that emotions are triggered by a cognitive evaluation/appraisal process of an individual's situation. Different from the *basic emotions view*'s assumption that facial actions are produced in patterns because a specific emotion has been triggered, promoters of appraisal theory assume that the outcomes of the appraisals are associated with changes in activity in many systems of the body, including the face. For example, the pattern of frowning is generated when something undesirable happens that prevent us from reaching the goal (Scherer, 2001; Smith, 1989).

The *dimensions view* was proposed by researchers who believe that emotional states are fundamentally differentiated on a small number of dimensions, such as valence and activation, and that facial activity is linked to these dimensions (e.g., (Russell, 1997)). The *dimensions view* assumes that the basic dimensions of an underlying emotional state are reflected in facial behavior. For example, Russell (1997) argues that in some situations, facial behavior simply changes toward a pleasant or an unpleasant state rather than due to discrete emotions such as sadness and happiness. This view is more similar to the appraisal view.

6.2.3 Summary

There are everlasting controversies among three views on emotional facial expressions. However, although each view on emotional facial expressions has its own predictions, it is not impossible that there are common ideas among them. According to Ortony et al. (1988), the relationship of the appraisal components to basic dimensions and to discrete emotions seems to be complicated but highly plausible. For example, Arnold (1960) proposed the valence dimension in the *cognitive view*, which is clearly related to the basic valence dimension found in all dimensional approaches.

It seems that no researchers from any view can provide evidence to fully defend their hypothesis. Nevertheless, the psychological studies from these views have a significant effect on our understanding of the link between emotion state and facial activity. These studies also play a very important role in the task of simulating and recognizing emotional facial expressions on computers. According to Kappas (2003), the *basic emotions view* is most useful in the context of diagnosing emotions from facial actions. Compared to research within the *cognitive view* and the *dimensions view*, research within the *basic emotions view* provides more empirical evidence on the relationship between emotion and facial activity. Moreover, the predictions of the *basic emotions view* are usually so clear to confirm or reject. In comparison, many predictions of the *appraisal view* and the *dimensions view* are not specific enough. In our opinion, the results from research within the *basic emotions view* are most useful in simulating the relationship between emotion and facial activity. Because there are also facial movements that relate to non-emotional processes such as visual speech and conversational signals, theories from the *basic emotions view* may not be the best choice for recognizing emotions computationally.

The rule-based system we present in this chapter is based on a collection of theories of emotion and facial expression from the *basic emotions view*, especially the work of Ekman and Friesen (1975) and Izard (1971, 1997). In this chapter we consider the following six emotions: **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust** and **Surprise**. These are said to be universal in the sense that they are associated consistently with the same facial expressions across different cultures (Ekman and Friesen, 1975). In their book, Ekman and Friesen also describe in detail what the expressions for these emotions and certain blends look like. We, however, do not claim that these six emotions are the only emotions that can be expressed on the human face. As long as there is more evidence on facial expressions of other emotions, rules can be added to reflect this evidence. Moreover, there are emotions like shame, shyness, and guilt that have also been considered to be basic in some sense; however these are not consistently associated with a particular facial expression (Izard, 1997). They may be expressed through gaze, behavior, posture, head or body movement (Izard, 1991).

Although our system described in this chapter is based on the *basic emotions view*, it does not have to use all the assumptions and implications presented by that view (work on recognition of emotions would depend more on a view). For example, because we just want to display the emotional state of our agents to 3D faces, we do not have to ensure the universality of the simulated facial expressions. In addition, we do not have to ensure that the simulated facial expressions are context-independent. Actually, we also allow other types of facial expressions (cf. Chapter 4), which makes the simulated emotional expressions context-dependent. For example, the act of "raising eyebrows" could be the simulated expression of surprise or a simulated conversational signal. However, we think that different types of facial expression happen in different contexts with different temporal patterns, e.g., a simulated conversational signal usually lasts much shorter than a simulated emotional expression (Casell, 2000).

Emotion feelings may differ in intensity. In (Ekman and Friesen, 1975) it is pointed out how for each of the basic emotions the expression can differ depending on the intensity of the emotion. It is therefore important for us to build our system on a representation that takes intensities into account. The human face is also able to show a combination (blend) of emotions at the same time. Ekman and Friesen (1975) describe which blends of the basic emotions occur and what these blends look like universally. We have used their descriptions as the basis for our fuzzy rules.

6.3 Computational work on emotional facial expressions

Earlier work on computational models of emotion and facial expression includes the directed improvisation system of Hayes-Roth and van Gent (1997), which makes an emotion-based selection among animation and audio sequences. Perlin and Goldberg's Improv animation system (Perlin and Goldberg, 1996; Perlin, 1997) layers small animations under the control of scripts and state variables including mood and personality. Stern et al. (1998) developed an animated pets system named Petz, with facial expressions, postures, and vocalizations corresponding to each personality profile and internal emotion state of the character. Magnenat-Thalmann et al. (1995) developed animated faces with emotional facial expressions to improve face-to-face communication between real and virtual humans. In most of this work, our concerns with modelling intensity as well as blends figure less prominently.

Blends of emotions are often defined in terms of graphics algorithms combining single emotion expressions (using interpolation for instance, Hendrix et al., 2000; Paradiso and L'Abbate, 2001; Pighin et al., 1998) instead of relying on the empirical rules described in the literature. Hendrix et al. (2000) use interpolation to display the intensity of emotions. For expression of blend of emotions, the basic emotions are arranged on an Emotion Disc with the neutral face in the center and maximal expressions of emotions on the perimeter. Each position on the Emotion Disc corresponds to an expression obtained by interpolation between the predefined expressions positioned on the disc. This method also does not rely on the empirical literature as the emotion intensity may be represented differently in different facial regions. Beside using interpolation, Pighin et al. (1998) also use regional blending to create blends of expressions. However, this method creates uncorrelated facial regions which do not appear in the human face. Moreover, they need to collect photographs of expressions of each basic emotion using cameras at different positions in order to generate blends of expressions.

6.4 Fuzzy rule-based system

6.4.1 Fuzzy sets

Human beings often need to deal with input that is not in precise or numerical form. Inspired by that observation, Zadeh (1965) developed a fuzzy set theory that allows concepts that do not have well-defined sharp boundaries. In contrast to the classical set theory in which an object can only be a full member or a full non-member of a set, an object in fuzzy set theory can possess a partial membership of a fuzzy set. A so-called “membership value” is used to measure the degree to which an object belongs to a fuzzy set. This value is a real number between 0 and 1. A fuzzy set is characterized by a so-called “membership function” that maps objects to their membership values. Note that an ordinary (or crisp) set is a very special case of a fuzzy set in the sense that now the membership values are restricted to the two-element set $0, 1$ which is obviously a proper subset of the real closed interval $[0, 1]$. A typical example of a membership function is shown in Figure 6.3.

A fuzzy proposition of the form “if x is A” is partially satisfied if the object x (usually crisp value x) is partial membership of the fuzzy set A. Based on that, fuzzy logic was developed to deal with fuzzy “if-then” rules where the “if” condition of the rules is a boolean combination of fuzzy propositions. When the “if” condition is partially satisfied, the conclusion of a fuzzy rule is drawn based on the degree to which the condition is satisfied.

6.4.2 Fuzzy rule-based system

A fuzzy rule-based system (FRBS) consists of a set of fuzzy rules to capture the relationship between the inputs and the outputs. Each rule of the system looks like:

$$\begin{array}{ll}\textbf{if } (x_1 \ is \ I_{1,k}) \textbf{ and } (x_2 \ is \ I_{2,k}) \textbf{ and } ... \textbf{ and } (x_n \ is \ I_{n,k}) & \\ \textbf{then } y \ is \ O_k & (k = 1..m)\end{array}$$

where m is the number of fuzzy rules of the system; x_1, x_2, ..., x_n represent inputs and $I_{1,k}$, $I_{2,k}$, ..., $I_{n,k}$ their corresponding fuzzy sets with respective membership functions $\mu_{I_{i,k}}$; y is the output and O_k its corresponding fuzzy set with membership function μ_{O_k}.

When crisp values are given as input for a FRBS, they are converted to membership values of corresponding fuzzy sets used by the system. Because a crisp value can be partial membership of several fuzzy sets, it activates a number of fuzzy rules to some degree. The extent to which the k-th rule is activated is calculated as:

$$\alpha_k = \mu_{I_{1,k}}(x_1) \quad \wedge \quad \mu_{I_{2,k}}(x_2) \quad \wedge \quad ... \quad \wedge \quad \mu_{I_{n,k}}(x_n)$$

where $\mu_{I_{i,k}}(x_i)$ is the membership value of x_i in the $I_{i,k}$ fuzzy set. The inferencing membership value of the output y is calculated as:

$$\mu_{O_k}^{inf} = \alpha_k \quad \wedge \quad \mu_{O_k}(y)$$

The minimum operator min is usually used as a $\wedge$ operator for both cases. The total membership value of the output y is calculated by the compositional rule of inference as:

$$\mu_O^{total}(y) = \mu_{O_1}^{inf}(y) \quad \vee \quad \mu_{O_2}^{inf}(y) \quad \vee \quad ... \quad \vee \quad \mu_{O_m}^{inf}(y)$$

The maximum operator *max* is usually used as an $\vee$ operator. The total membership value of the output y is then defuzzified, which maps the membership value from the fuzzy domain back into the crisp domain. The objective is to obtain a single crisp output that best represents the inferred membership value. Some of the defuzzification methods, which are practically important, are Center of Area (COA), Center of Maximum (COM), and Mean of Maximum (MOM) (Shaw, 1998). In the research presented in this chapter, we use the COA method to obtain the final crisp value. The COA method computes the centroid of the composite area representing the output fuzzy term:

$$y_{final} = \frac{\int \mu_O^{total}(y) \cdot y \cdot dy}{\int \mu_O^{total}(y) \cdot dy}$$

A fuzzy rule-based system provides an easy way to capture rules described in natural language such as “if the speed is low then the fuel consumption is low”. The use of fuzzy sets is very suitable for dealing with naturally vague concepts such as small, high, etc. In modeling the relationship between emotion and facial activity, the fuzzy rule approach allows us to incorporate qualitative descriptions as above with quantitative information (emotion intensity and contraction level). Moreover, we still have a comprehensible rule-based system in which the logical descriptions are visibly encoded. We would miss out on that when using other models like neural networks, for instance.

6.5 Overview of the system

Our system maps a representation of the emotional state to a vector of facial muscle contraction levels which is used to combine with other facial movements to control the facial animations on a 3D face; cf. Chapter 4. The system, as shown in Figure 6.2, consists of five components:

1. The input is an Emotion State Vector (ESV), resulting from the system described in the previous chapter. This is a vector of intensity of the six emotions we mentioned earlier, each of which is represented by a real number:

   $$ESV = (e_1, e_2, ..., e_6)$$

   where $0 \leq e_i \leq 1$

2. The output is a Facial Muscle Contraction Vector (FMCV):

   $$FMCV = (m_1, m_2, ..., m_{19})$$

   where $0 \leq m_i \leq 1$. This is a vector of contraction level of 19 muscles in the right side of our face model which are shown in Table 2.1.

3. The Expression Mode Selection determines whether a single emotion or blend of two emotions will be expressed in the 3D face model.
4. In the Single Expression Mode muscle FRBS, contraction levels from a single emotion intensity are produced.
5. In the Blend Expression Mode FRBS, muscle contraction levels from two emotion intensity values are produced.

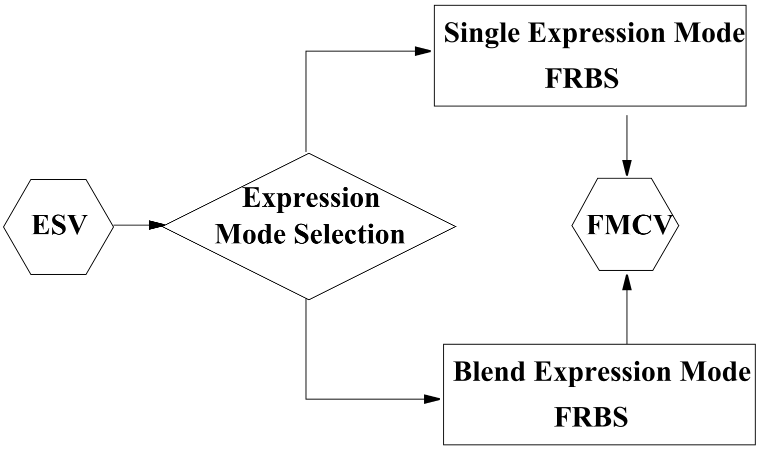


Figure 6.2: The emotion-to-expression system.

FRBS The core of our system is formed by two FRBSs, as described in Section 6.4.2. Two collections of fuzzy if-then rules are used to capture the relationship between the ESV and the FMCV. During fuzzy inference, all the rules that are in one of the collections are fired and combined to obtain the output conclusion for each output variable. The fuzzy conclusion is then defuzzified resulting in a final crisp output.

The fuzzy if-then rules for both single and blend expressions are based on Ekman and Friesen's summary of the facial movements to express basic emotions (Ekman and Friesen, 1975). We used several other information sources to map the descriptions of faces from Ekman and Friesen onto the values for the muscle contraction levels that generate the expressions. These sources were the Facial Action Coding System (FACS)(Ekman and Friesen, 1978), which will be presented in Appendix A, and the book and tutorial by, respectively, Parke and Waters (1996), and Prevost and Pelachaud (1995). Also our own observations on emotion expression in human faces have played a role. We will discuss these rules in more detail in Section 6.6.

As can be seen from the system, the FRBS is actually broken up into three components: the Expression Mode Selection, the Single Expression Mode FRBS and the Blend Expression Mode FRBS. The expression of an emotion in a blend may differ in important ways from the expression of the emotion occurring on its own. Typically, for a single emotion expression several regions of the face are involved whereas in blends some of these regions may be used for another emotion. Therefore we do not want the single expression rules to fire when blends occur. It might be possible to build a system with just a single collection of fuzzy rules. However, this will complicate the statement of the rules considerably.

The emotional state vector, ESV, represents the emotional state of the agent. The human face cannot display all the combinations of emotion intensities that can be felt at the same time universally and unambiguously. In our system, only two emotions can be displayed at the same time. The reason is that we have not found any clear evidence about the expression of more than two emotions on the face.The mapping between emotional state and facial display is not direct for other reasons as well. Several factors may be involved in real persons to decide for an emotion that is felt whether or not it will be displayed. There may be cultural rules for instance that inhibit showing certain emotions (c.f, Section 6.2.1). An Expression Mode Selection module can mediate between the emotion state as it is felt and the rules for representing the emotions to be displayed. In our current implementation we select either the single or the blend expression mode based on the intensities of the emotions felt. The Single Expression Mode is selected when a single emotion has to be expressed. This is the case when only the difference between the two emotions with highest intensity are larger than 0.5. In this case, the Single Expression Mode Fuzzy Rule Based System (FRBS) will be used, and the input of the FRBS is the single emotion with highest intensity. When the Blend Expression Mode FRBS is used, the input of the FRBS is the pair of emotions with highest intensity (in the case that more than two emotions have the same highest intensity, two emotions will be randomly selected to express). We certainly do not claim psychological realism here.

FMCV The muscle contraction levels, to which the rules give rise, are used to combine with other facial movements to manipulate the 3D face. Our 3D face model with 19 muscles is presented in Chapter 2.

The introduction of the FMCV enables the combination of an agent's facial emotion expressions with other facial movements, as described in Chapter 4. The system is also designed to take into account future expansions. The use of the ESV and the Expression Mode Selection allows the integration of the agent's intention and personality into the model without changing the fuzzy rules for expressing emotions. This can be done by distinguishing the real ESV as felt from something like a "to-display" ESV. This "to-display" ESV does not represent the agent's real emotion state but it equals the emotion state the agent wants to express. For example, with a strong personality, the agent may display a fake smile to mask sadness by increasing the intensity of happiness in the "to-display" ESV.

6.6 The fuzzy rule-based systems

The subsystems "single expression mode" and "blend expression mode" are both implemented using fuzzy rules. Both subsystems must convert an emotion state to a contraction level for the facial muscles taking into account the intensity of the emotions. In the literature on facial expressions of emotions qualitative descriptions like "surprise then lift eyebrows" can be found. There are also some quantitative descriptions which are presented more in natural language rather than in a precise number form such as "if the level of sadness is low, then draw the eyebrows together; while if the level of sadness is high, then draw the eyebrows together and draw the corners of the lips down". In order to take intensities as well as vague concepts like "low" and "high" into account, these (logical) rules have been transformed into fuzzy rules.

6.6.1 Fuzzy sets

We model the intensity of each emotion by five fuzzy sets (Figure 6.3): **VeryLow**, **Low**, **Medium**, **High**, and **VeryHigh**. The contraction level of each muscle is described by again five fuzzy sets (Figure 6.4): **VerySmall**, **Small**, **Medium**, **Big**, and **VeryBig**. The exact form of the membership functions and the support of each membership function are experimentally determined.

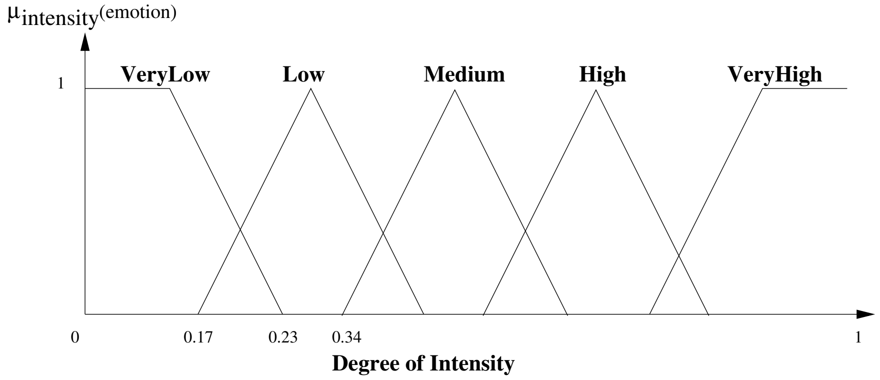


Figure 6.3: Membership functions for emotion intensity.

6.6.2 The Single Expression Mode FRBS

The rules in the single-expression mode FRBS take on the following form:

if Sadness is **VeryLow then**
muscle 9's contraction level is **VerySmall**
muscle 13's contraction level is **VerySmall**
muscle 14's contraction level is **VerySmall**

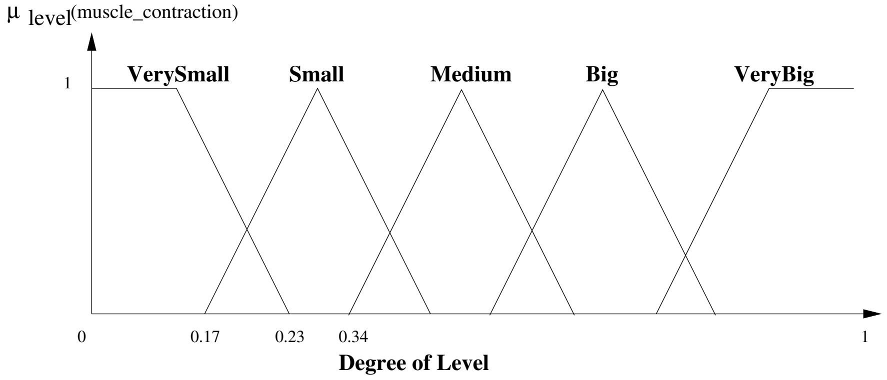


Figure 6.4: Membership functions for muscle contraction level.

muscle 15's contraction level is **VerySmall**
muscle 18's contraction level is **VerySmall**

The sample rule above encodes the information presented in the first row of Table 6.1. Note that the relation between the emotion intensities and the muscle contraction level is not so straightforward that we can use a simple mapping. The name and position of the muscles can be seen in Table 2.1 and Figure 2.7(a) in Chapter 2. All the rules for other single emotions are represented in Table 6.2, 6.3, 6.4, 6.5 and 6.6.

Emotion Intensity	m9,m13,m14, m15,m18	m17	m3
vl	vs	–	–
l	s	vs	–
m	m	s	–
h	b	m	m
vh	vb	m	b

Table 6.1: Fuzzy rules for emotion **Sadness**, vh:**VeryHigh** h:**High** m:**Medium** l:**Low** vl:**VeryLow** vs:**VerySmall** s:**Small** m:**Medium** b:**Big** vb:**VeryBig** −:no contraction.

6.6.3 The Blend Expression Mode FRBS

If the Single Expression Mode is not used, then the Blend Expression Mode is selected automatically. In this mode two emotions are displayed on the face. Normally each of these two emotions is displayed in a separate region of the face.

Emotion Intensity	m1,m2,m18
vl	vs
l	s
m	m
h	b
vh	vb

Table 6.2: Fuzzy rules for emotion **Happiness**.

Emotion Intensity	m4,m8,m10,m13, m14,m15,m17
vl	vs
l	s
m	m
h	b
vh	vb

Table 6.3: Fuzzy rules for emotion **Anger**.

Emotion Intensity	m3,m4,m5, m9,m10,m17	m13	m14,15	m12	m19
vl	vs	vs	vs	vs	–
l	s	vs	vs	s	vs
m	m	m	s	s	vs
h	b	m	m	b	vs
vh	vb	m	m	vb	vs

Table 6.4: Fuzzy rules for emotion **Fear**.

The fuzzy rules for the blend of expressions reflect this fact. The contraction level of a muscle is determined by the intensity of the emotion that will be displayed in the facial region to which this muscle belongs. As the contraction level of each muscle is determined by the intensity of only one of the emotions, there will not be conflict values placing on any muscle's intensity. We will illustrate this with a description of the blend of **Sadness** and **Fear**.

Ekman and Friesen (1975) describe how in a such a blend **Sadness** is expressed in the brows and eyelids while **Fear** is expressed in the mouth. Combining this with the specification of muscle movements in the FACS (see Appendix A), we can define the emotions in muscle terms. **Sadness** is expressed by contracting the muscles Frontalis Medialis(9), Depressor Supercilii(13), Corru-

Emotion Intensity	m6,m11,m12
vl	vs
l	s
m	m
h	b
vh	vb

Table 6.5: Fuzzy rules for emotion **Disgust**.

Emotion Intensity	m9,m10,m16	m19
vl	vs	vs
l	s	vs
m	m	s
h	b	m
vh	vb	m

Table 6.6: Fuzzy rules for emotion **Surprise**.

gator Supercilii(14), Depressor Glabelle(15) and Orbicularis Oculi(17 and 18). **Fear** is expressed by contracting the muscles Triangularis(3), Risorius(4), Depressor Labii(5) and by opening the jaw (19). The contraction level of each of those muscles is then determined by the intensities of **Sadness** and **Fear**. The format of such a rule in our system looks as follows:

if Surprise is **Low** and **Fear** is **Medium then**

muscle 9's contraction level is **Small**
muscle 10's contraction level is **Small**
muscle 16's contraction level is **Small**

muscle 3's contraction level is **Medium**
muscle 4's contraction level is **Medium**
muscle 5's contraction level is **Medium**
muscle 17's contraction level is **Medium**

The rules for blending expressions of emotion are derived from the rules for single expressions and the list of muscles that participate in the blend expressions (see Table 6.7). For example, for the blend expression of **Surprise** and **Fear**, muscle 9, 10 and 16 are contracted to express **Surprise**, while muscle 3, 4, 5 and 17 are contracted to express **Fear**. Then, the contraction levels of muscle 9, 10 and

16 are determined by the rules for the single expression of **Surprise**, while the contraction levels of muscle 3, 4, 5 and 17 are determined by the rules for the single expression of **Fear**.

There are no rules for blend expressions of **Happiness** and **Disgust**, and **Sadness** and **Surprise**. For these expressions, only the emotion with higher intensity is expressed.

Blend expression	First emotion	Second emotion
Surprise + Fear	m9, m10, m16	m3, m4, m5, m17
Surprise + Anger	m19	m13, m14, m15, m17
Surprise + Happiness	m9, m10, m16	m1, m2, m19
Surprise + Disgust	m9, m10, m16	m6, m11, m12
Fear + Anger	m3, m4, m5, m19	m13, m14, m15, m17
Fear + Happiness	m9, m10, m13, m14, m15, m16, m17	m1, m2, m19
Fear+ Sadness	m3, m4, m5, m19	m9, m13, m14, m15, m17, m18
Fear+ Disgust	m9, m10, m13, m14, m15, m16, m17	m6, m11, m12
Anger+ Happiness	m13, m14, m15, m17	m1, m2, m19
Anger + Sadness	m13, m14, m15, m17	m3
Anger + Disgust	m13, m14, m15, m17	m6, m11, m12
Happiness + Sadness	m1, m2, m19	m3, m4, m5, m17, m18
Sadness + Disgust	m3, m4, m5, m17, m18	m6, m11, m12

Table 6.7: List of muscles that take part in blend expressions of emotion

6.7 Experimental results

The expressions of six basic emotions and a neutral face are displayed in Figure 6.5. In Figure 6.6, **Surprise** is shown with increasing intensity. The increasing intensity of **Surprise** can be seen in the increase in the raising of the eyebrows and the increase in the opening of the mouth. Figure 6.7 (left) shows the blend of **Anger** and **Disgust**. It can be seen that **Anger** is represented in the eyebrows and eyelids while **Disgust** is represented in the mouth. Blend of **Happiness** and **Surprise** are shown in Figure 6.7 (right). This is a combination of surprised eyebrows and a happy smiling mouth.

These results also show that the emotions are not only displayed in the main parts of the face like mouth and eyebrows but also in very detailed parts like eyelids and lips. The blends of expression are displayed according to the

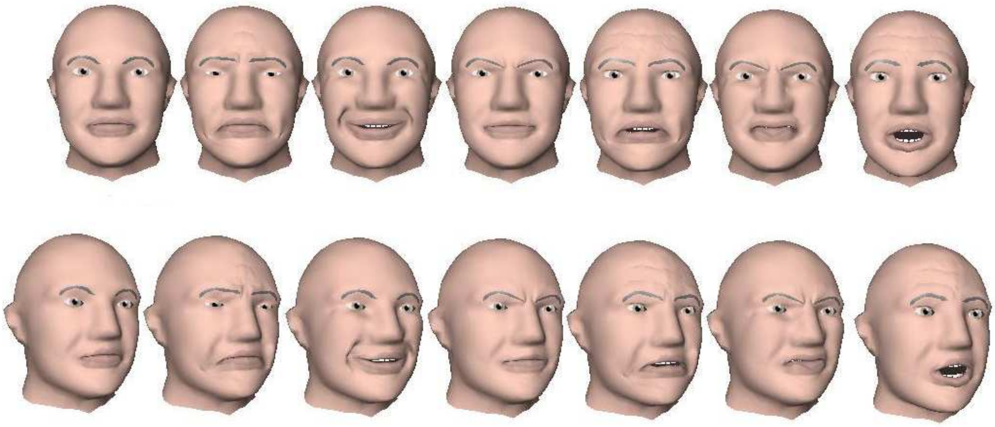


Figure 6.5: Basic emotions: neutral, **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust**, **Surprise** (from left to right).

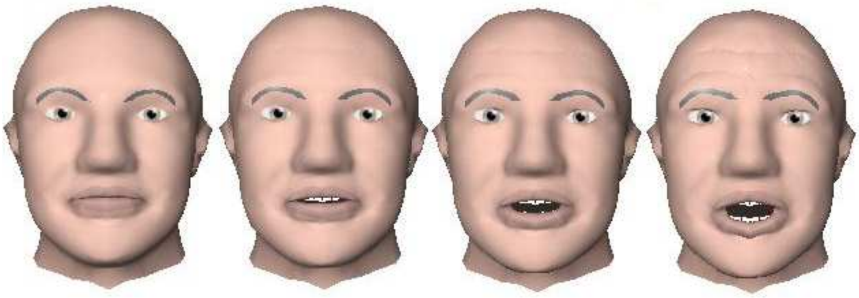


Figure 6.6: Increasing **Surprise**.

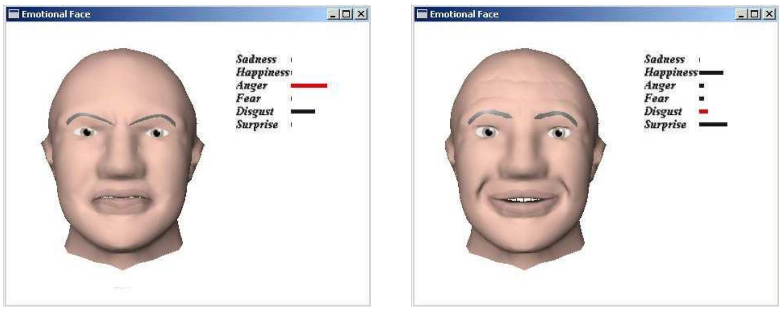


Figure 6.7: Blends of **Anger** and **Disgust** (left), **Happiness** and **Surprise** (right).

rules as described by psychologists instead of being computed by some graphics algorithm that combines values for single emotion expressions (morphing, interpolation). And finally, the quality of the facial expressions is improved by the smooth relationship function between emotion intensities and muscle contraction levels. This smooth relationship function is obtained with fairly simple fuzzy if-then rules rather than with complicated formulas or intensively trained Neural Networks.

6.8 Conclusion and future research

In this chapter, we discussed the problem of generating facial expressions from emotional state. Our main concerns are dealing with emotions with different intensity and blends of emotions. Therefore, we have proposed a fuzzy rule-based system to achieve this aim. We have selected the *basic emotions view* from the psychology literature to base our system on. From this view, it is argued that a small set of basic emotions can be differentiated from each other by emotional facial expressions. Although this view is not without controversies, we have found it most useful in simulating emotional facial expressions. However, we certainly do not claim the psychological realism of our system. The motivation behind the work presented in this chapter is to provide a way to visualize the emotional state of an agent. Therefore, we did not model every human being's possible way of expressing emotion. In the current version of our system, we have modeled the expression of six emotions: **Happiness**, **Sadness**, **Anger**, **Fear**, **Disgust** and **Surprise** because there has been convincing evidence about the universalities in expressing these emotions. With simple fuzzy rules, lifelike emotional facial expressions are generated based on descriptions in the literature. The variations resulting from differences in the intensity of emotions are also successfully displayed. Our system can also be extended to take into account other ways of expressing those six basic emotions as well as the expression of other emotions if there is enough psychological evidence.

The effect of the fuzzy membership function on the manner of expression is one of the issues for future research. A learning component can be added into the system so that we can have slightly different ways of expressing an emotion state for different agents. The expression mode selector should become more complex to take into account other factors besides intensity. Finally, other emotions besides the "universal" ones should be dealt with.

Part III

Example Embodied Agents

Chapter 7

An application: Obie – A football supporter

"Football's not just about scoring goals – it's about winning."

– Alan Shearer

"I'd like to play for an Italian club, like Barcelona."

– Mark Draper

7.1 Introduction

In this chapter we discuss how the several systems which we have described in previous chapters can be combined to create an emotional embodied agent. These systems are: the 3D face model, which is presented in Chapter 2 and is developed into a talking head in Chapter 4; ParleE, the implementation of emotions, to which Chapter 5 is devoted; the fuzzy rule-based system in Chapter 6, which converts emotions into facial expressions. Chapter 3 can be used to create different face models for the agent, although it is not important for creating an emotional agent.

In the present chapter we introduce Obie, a football (soccer) supporter agent. Obie is watching a football match in which a team, which he supports, is playing. Obie can experience different emotions by appraising events based on his goals, standards, and preferences. Obie can also show his emotions on a 3D talking head. We consider three types of football match. The first type consists of the real football matches that take place in stadiums. The second type is the robot football matches where the human players are replaced by physical robots. The third type is the simulation matches where we have 2D or 3D virtual robots and the field only exists on the computer screen. Typical events that occur in a football match are: kick-off, penalty, goal, free-kick, etc. These events can be obtained in various ways. For a real football match, the events may be extracted directly by translation from visual to verbal representations or translation from a news stream produced by a mediator (e.g., a human commentator) to a textual representation (Jong and Westerveld, 2001; Nijholt et al., 2003). For a robot cup match, the events can be extracted from a team's vision system (Kooij, 2003). The events in a simulation match can be extracted directly from the data of the match.

There are several reasons why we choose to model a football supporter. First of all, football is an emotional game. There are many events in the the game that trigger emotions of not only players but also of coaches, supporters, etc. A last-minute goal triggers happiness or relief in some people whereas it triggers sadness, anger or disappointment in other people. Implementing the football supporter's domain gives us the chance to test many emotions as well as blends of emotions. Secondly, because the actions in a football match happen fast, the emotional state of a football supporter also changes fast during the match. This gives the implemented agent a chance to experience many emotions in a short period. Thirdly, a supporter in a football match can experience extreme emotions, which allows us to observe the expressed emotions more easily. Finally, the utterances of a supporter are usually short and simple. Therefore, we do not have to implement a very sophisticated text generation component in order to test the agent's ability to express emotions during speech. Our motivation to implement a football supporter agent is also inspired by two existing projects in our department: the robot soccer project (Seesink et al., 2003) and the football commentary generation project (Jong and Westerveld, 2001; Nijholt et al., 2003).

Section 7.2 discusses how we combine the systems from previous chapters to create an embodied agent that can experience and express emotions. The football supporter's domain is discussed in Section 7.3. Finally, an illustration of the behavior of our supporter agent, Obie, is presented in Section 7.4.

7.2 Embodied agent that can experience and express emotions

By combining several systems that are described in previous chapters, we can create an embodied agent that can experience and express emotions. An overview of the agent can be seen in Figure 7.1. The agent takes events as input. Then it appraises these events according to his/her goals, standards and preferences to generate emotions. Some basic emotions are mapped to facial expressions. Next, facial expressions are displayed on a 3D talking head. The agent's internal state is also shown in the form of charts and graphs as well as the content of the speech which the agent utters. The emotional expressions via the content of the speech is done by a simple mapping between emotions and text, e.g., the emotion fear is mapped with the sentence “Oh no!”.

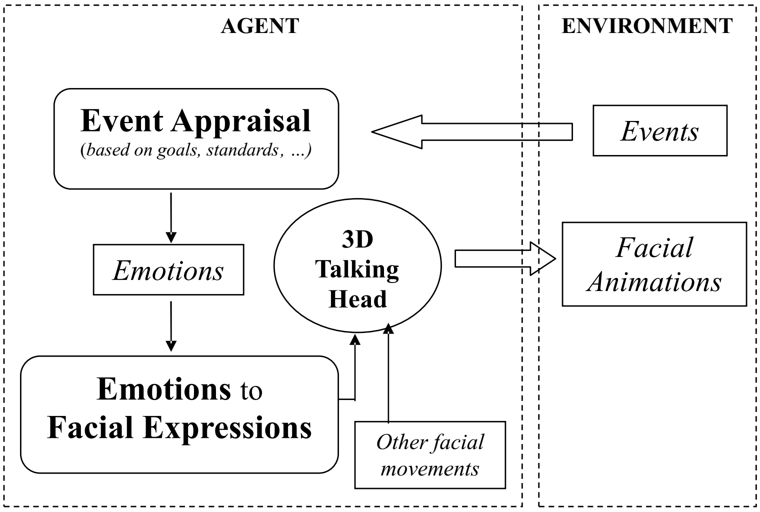


Figure 7.1: Overview of an agent that can experience and express emotions.

The event appraisal component Events are appraised by ParleE, an implementation of emotions. ParleE is presented in Chapter 5. Note that the input of the agent system is the input of the ParleE implementation. Recall that the core of ParleE is a probabilistic planner, the domain of which is represented in STRIPS (Fikes and Nilsson, 1971). Therefore, the events which the agent takes as input are represented in STRIPS. An event could be:

- the outcome of self's actions,
- the occurrence of other agents' actions and the outcome of their actions.

ParleE allows agents with different personalities, characteristics and in different motivational states to experience emotions differently. ParleE can generate twenty emotions with intensity.

From emotions to facial expressions Six basic emotions are converted to facial expressions by a fuzzy rule-based system, which is discussed in Chapter 6. This system also takes into account the blending of emotions.

The 3D Talking Head A muscle-based 3D face model is presented in Chapter 2. This is a simple face model which could display realistic facial expressions in real-time. Different facial movements on this face model are combined temporally resulting in a 3D talking head, which is described in Chapter 4.

All the systems introduced in Chapters 2-6 are domain-independent. Therefore, our agent can be applied to any application as long as the domain is well established in STRIPS.

7.3 The football supporter domain

In our application, we consider the situation in which a **“for”** and an **“against”** team are playing against each other. Obie is a supporter agent, while other agents are the **“for”** and **“against”** teams. For simplicity, we consider each team as a single agent. The possible actions Obie can do are: watching, cheering, etc. These actions do not affect the match being played. In order to predict the likelihood of events and to appraise events, Obie has knowledge about possible actions that other agents could take to model their plan. Examples of these actions are: tackling when the team does not have the ball, shooting when the team has the ball and the ball is in the other team’s half-field or penalty area. Obie’s objective is the same as the objective of the **“for”** team, which is the **“for”** team winning, whereas the objective of the **“against”** team is the **“against”** team winning.

We implement Obie’s emotions focusing on only one other agent (either **“for”** or **“against”** team). Obie and the other agent perform their actions in turns. Obie’s action is selected based on the planning algorithm, which is given in Chapter 5. The other agent’s action is extracted from a script file that describes the match or is extracted in real-time from a real soccer match or a robot soccer match.

7.3.1 Designing the domain

In our application, an event can inform us about what action is taken, what facts are deleted and what facts are added. Facts in the football supporter domain contains the following information:

- what the current score is,
- where the ball is at this moment,
- which team is controlling the ball,
- whether a kick-off is being taken,
- whether a free-kick is being taken,
- whether a penalty is being taken.

To make the information where the ball is during the match usable, we simply divide the soccer field into four regions, which are displayed in Figure 7.2:

- **“for”** team’s penalty area,
- **“for”** team’s half-field (except the penalty area),
- **“against”** team’s half-field area (except the penalty area),
- **“against”** team’s penalty area.

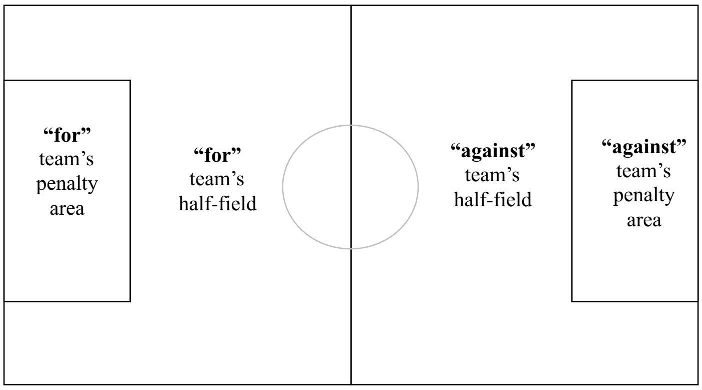


Figure 7.2: The view of a soccer field.

To support a multi-agent environment, a predicate indicating which agent is acting at a certain moment in time is added to the domain actions. The main advantage of this implementation is that we can still use the planning algorithm without modifying the hard code much.

As the STRIPS format does not support quantity, we choose a simple way of representing the possible score rather than extending the STRIPS for quantity. A quantitative STRIPS format would require a big change in the planning algorithm. Our simple way of representing the score is to represent the current difference in the score between the **"for"** team and the **"against"** team:

$-10, \cdots, -3, -2, -1, 0, +1, +2, +3, \cdots, +10$

This way, we would not be able to represent all the numbers, but we think that 10 is reasonable maximum difference between the two teams.

7.3.2 Representing the domain

Possible facts in the domain are represented in STRIPS as follows (a full list of all possible facts will be presented in Appendix C):

- Position of the ball: e.g., (ball-pa team) — the ball is in the penalty area of the **"for"** team.
- Who is controlling the ball: e.g., (ball-control team) — the **"for"** team is controlling the ball.
- Who is acting (for supporting multi-agent planning as described above): e.g., (acting team) — it is the team's turn.
- Match status: e.g., (no-penalty team) — it is not the penalty for the **"for"** team.
- Score: e.g., (score minus-one), (score zero), etc.

After defining the possible facts, the possible actions for a team are:

- Attacking actions (the team is controlling the ball): e.g., *long-shot* — this action is available when the ball is in the other team's half-field.
- Passing actions (the team is controlling the ball): e.g., *pass-from-pa* — pass when the ball is in the team's penalty area.
- Defending actions (the team is not controlling the ball): e.g., *tackle-in-pa* — tackle when the ball is in the team's penalty area.
- Penalty, free-kick, and kick-off related actions: e.g., *defend-freekick-in-hf* — defend free-kick in the team's half-field.

Each action requires the presence of certain facts and produces certain facts. For example, the **"for"** team's action *take-penalty* requires the presence of the fact (penalty team). Each action can have several different outputs. The action *take-penalty* may result in a goal or not. However, this action leads to a goal with a very high probability.

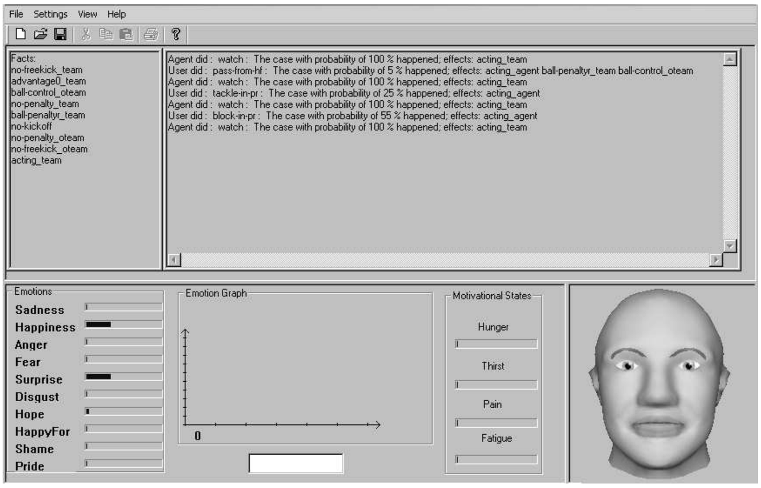
7.4 Illustration

The interface of our football supporter agent, Obie, is shown in Figure 7.3. In the top, there are two windows to show the current state of the world (current facts) and what has happened. In the bottom, there is a chart and a graph to show Obie's emotion state. There is also a 3D face model representing Obie's body.

The football match starts with the score difference zero, the ball is controlled by the **"for"** team in the **"for"** team's half-field. The match lasts for ninety minutes. Each minute there is an event happening due to the actions of the two teams. The events are read from a script file. Obie's purpose is the **"for"** team winning.

As mentioned before, we consider, for simplicity's sake, each team as a single agent and we only deal with one team rather than both teams. We model the actions of the two teams through the action of one single team. For example, the action "defend free-kick" of one team means that the other team is taking the free-kick. Obie has a model of the other agent to derive the likelihood of what will happen next. Every minute, Obie appraises the occurred event based on his goals and standards which triggers some emotions that Obie will experience.

Now we will show how the match progresses and how Obie experiences and expresses his emotions during the match. There are several main events in the match. The **"against"** team opens the score at the 12th minute. The **"for"** team starts attacking back and levels the score at the 30th minute. Ten minutes later, the **"for"** team has a free-kick from their half-field. Amazingly, the free-kick reaches the **"against"** team's penalty area and still in the control of the **"for"** team. Without any hesitation, the **"for"** team makes a shot and leads by



```
Facts:
no-freekick_team
advantage0_team
ball-control_oteam
no-penalty_team
ball-penaltyr_team
no-kickoff
no-penalty_oteam
no-freekick_oteam
acting_team
```

```
Agent did :  watch :  The case with probability of 100 % happened; effects: acting_team
User did :  pass-from-hf :  The case with probability of 5 % happened; effects: acting_agent ball-penaltyr_team ball-control_oteam
Agent did :  watch :  The case with probability of 100 % happened; effects: acting_team
User did :  tackle-in-pr :  The case with probability of 25 % happened; effects: acting_agent
Agent did :  watch :  The case with probability of 100 % happened; effects: acting_team
User did :  block-in-pr :  The case with probability of 55 % happened; effects: acting_agent
Agent did :  watch :  The case with probability of 100 % happened; effects: acting_team
```

Figure 7.3: A snapshot of the football supporter application.

one goal. The match continues without any special event until the 80th minute when the **“against”** team nets a goal, which brings the score back to equal, i.e., the score difference back to zero.

The emotions of Obie occur in mixture. However, we will analyze individual emotions to see clearly how each emotion changes during the match. Figure 7.4 shows the graph of Obie’s fear during the match and Obie’s expression at the beginning of the match. Recall that Obie’s purpose is the **“for”** team winning and the starting score difference is zero. Therefore, Obie has reason to worry at the beginning of the match that the **“for”** team may not win. This fear state ends at the 40th minute when the **“for”** team scores the second goal to lead by one goal. The fear emotion is triggered again at the 80th minute when the **“against”** team levels off the score.

Figure 7.5 shows the graph of Obie’s happiness from the 28th minute to the 46th minute, when there are two goals scored by the **“for”** team. The figure also shows Obie’s expression at the 30th minute when the first goal of the **“for”** team is scored. As can be seen from the graph, Obie’s happiness is increasing when the **“for”** team is attacking and especially after the two goals. After the 46th minute, there are no special events. As a consequence, Obie’s happiness decays to return to the neutral state. This is shown in Figure 7.6.

Figure 7.7 shows the graph of Obie’s surprise and his expression of surprise. As can be seen from the graph, surprise is triggered seldom. For example, at the 40th minute, the **“for”** team takes a long free-kick from their half-field and still receives and controls the ball in the **“against”** team’s penalty area. This situation usually happens with very low probability. Obviously, it triggers Obie’s surprise. After being triggered, Obie’s surprise decays very fast as well.

Figure 7.8 shows the graph of Obie’s sadness during the occurrence of the

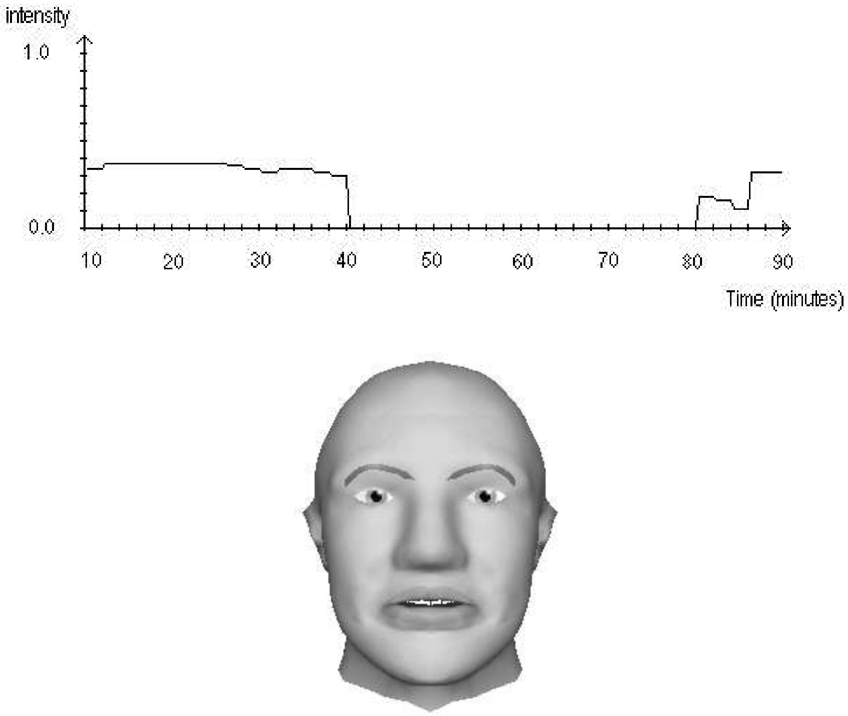


Figure 7.4: The graph of Obie's fear emotion and a snapshot of his expression of fear.

"against" team's second goal and Obie's expression at the 80th minute. The graph also shows that sadness decays very slowly after it has been triggered.

Blends of emotions are shown in Obie's face as well. For example, at the 40th minute, both surprise and happiness are triggered. They both are expressed in Obie's face, which can be seen in Figure 7.9.

We have also tested how agents with different personalities experiences emotions (cf. Chapter 5). Figure 7.10 and 7.11 show the happiness and sadness of Obie with four different personalities: neutral, pessimistic, realistic and sensitive. As can been seen from the figures, compared to the neutral Obie, the emotion state of pessimistic Obie is biased toward negative emotions. Realistic Obie concentrates more on what has happened than what is expected. Sensitive Obie experiences every emotion more intensively than the neutral Obie.

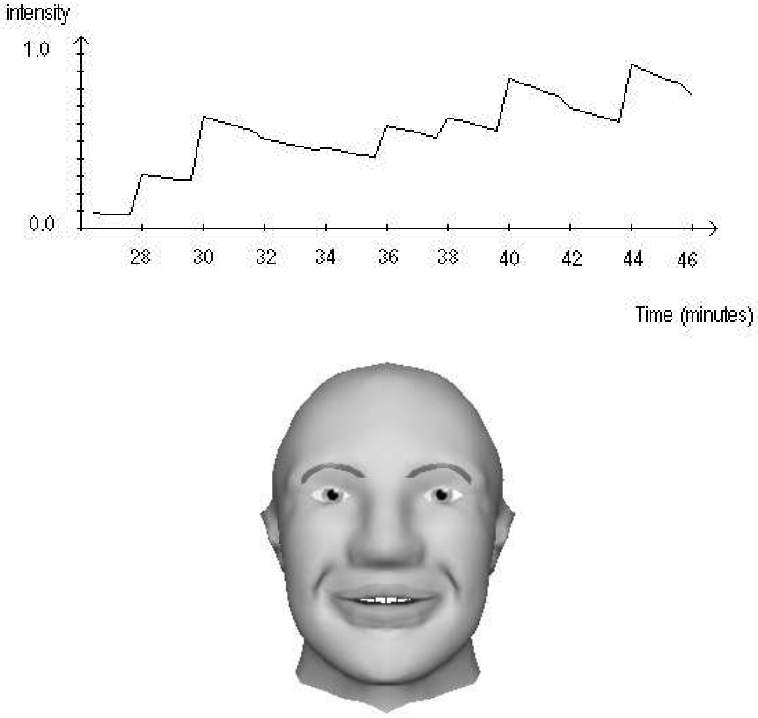


Figure 7.5: The graph of Obie's happiness emotion and a snapshot of his expression of happiness.

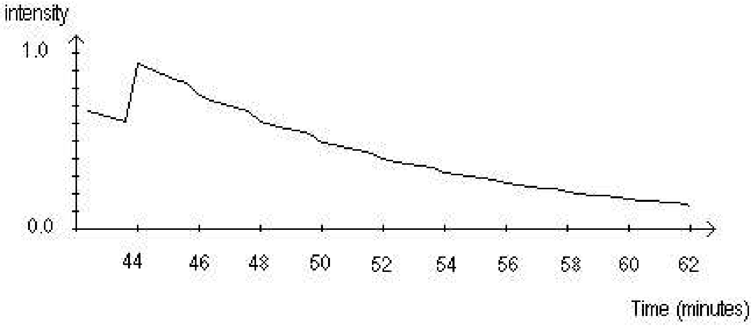


Figure 7.6: Obie's happiness decays to return to the neutral state.

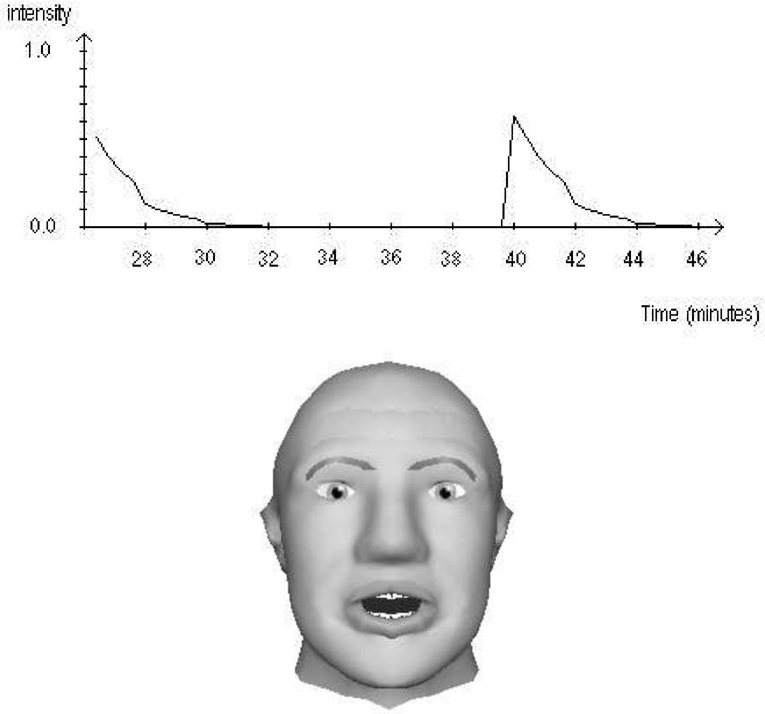


Figure 7.7: The graph of Obie's surprise emotion and a snapshot of his expression of surprise.

7.5 Conclusion

The systems which we have described in previous chapters have been combined to create an emotional embodied agent. In this chapter, we presented Obie, a football supporter agent. Obie has an emotion component which appraises events to trigger emotions. This component is the result of Chapter 5. Obie's emotions are expressed via his utterance or his facial expressions. The expression via utterance is done by a simple mapping from emotions to text fragments. The mapping from emotions to facial expressions was described in Chapter 6. Obie's utterance and facial expressions are presented in a 3D talking head, which is the result of Chapters 2 and 4. We have shown how Obie experiences different emotions during a football match. We have also indicated how Obie with different personalities experiences emotions differently.

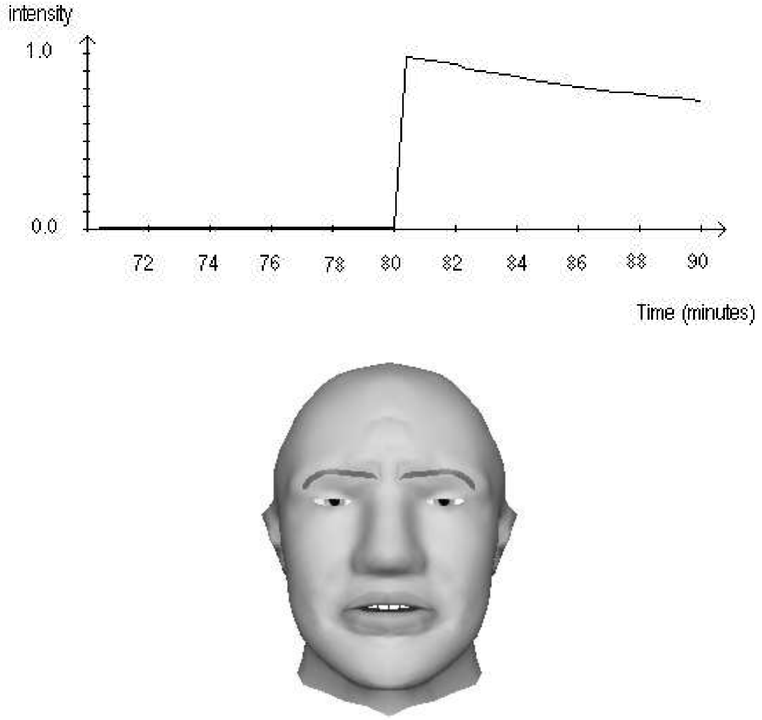


Figure 7.8: The graph of Obie's sadness emotion at the last goal and a snapshot of his expression of sadness.

Figure 7.9: A snapshot of Obie's expression of happiness and surprise.

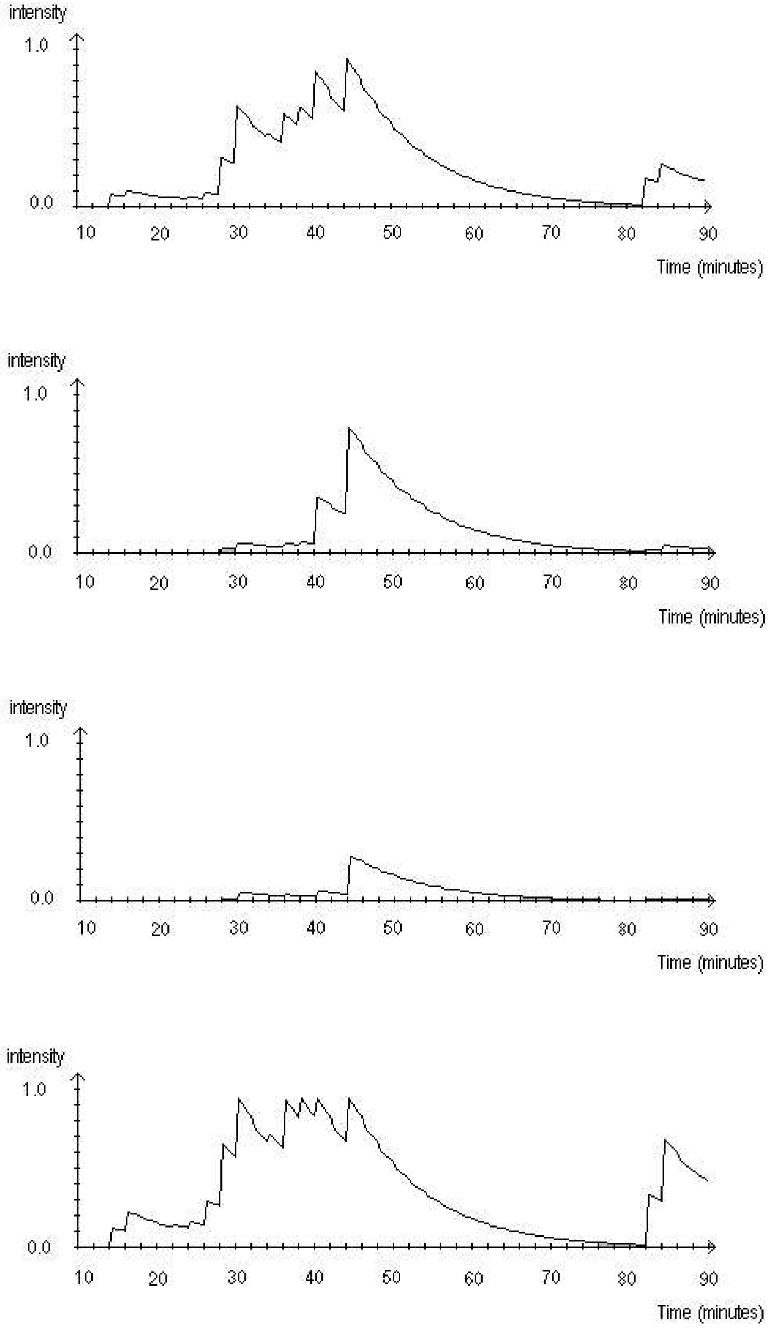


Figure 7.10: The graph of Obie's happiness emotion with four different personalities: neutral, pessimistic, realistic and sensitive (from top to bottom).

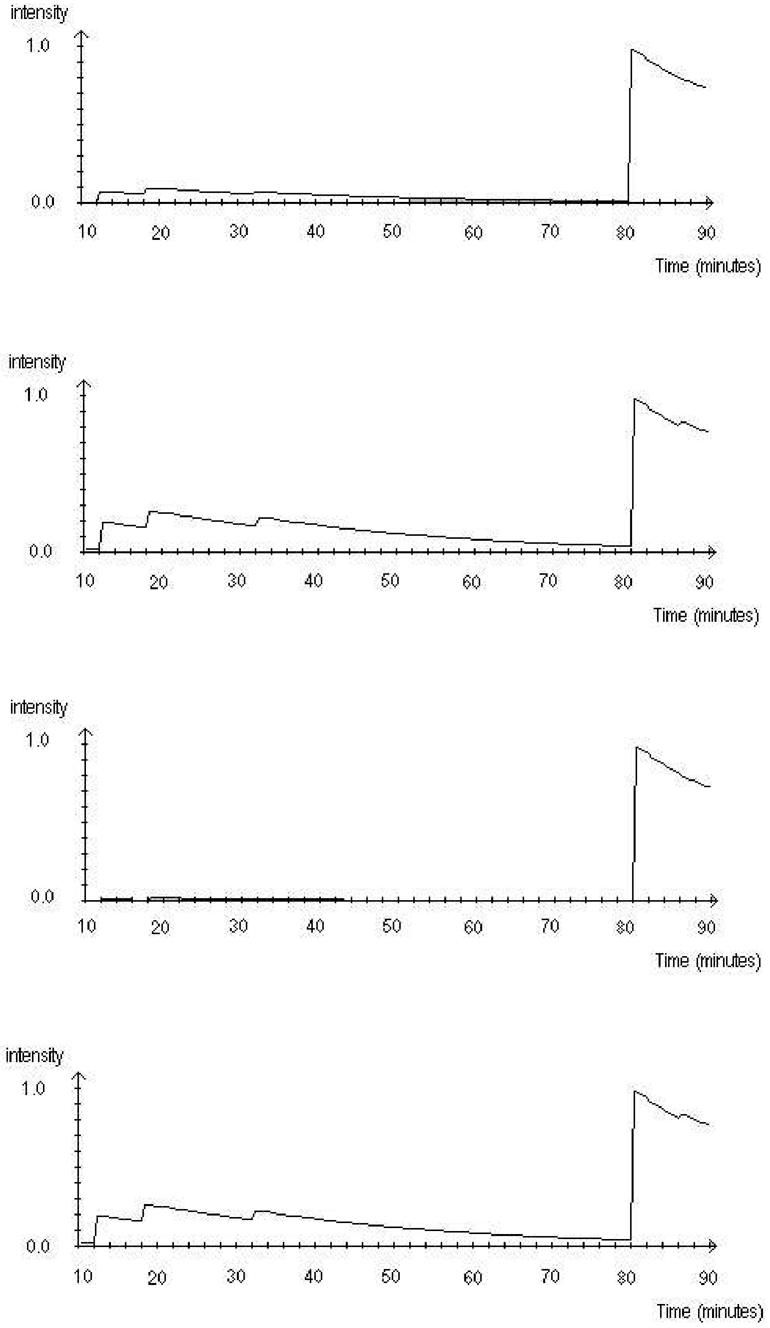


Figure 7.11: The graph of Obie's sadness emotion with four different personalities: neutral, pessimistic, realistic and sensitive (from top to bottom).

Chapter 8

Conclusion

> *"A conclusion is simply the place where you got tired of thinking."*
>
> – *Anonymous*

We introduced in Chapter 2 a simple muscle-based 3D face model that can produce realistic facial expressions in real-time. The face model contains a face mesh and a muscle model. The face model allows high quality and realistic facial expressions. In addition, the face model is sufficiently simple in order to keep the animation real-time. The muscle model is an extension of Waters' muscle model. It produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time. We have also presented the implementation of the realistic **Orbicularis Oris** (muscle of the mouth), **Orbicularis Oculi** (muscle of the eyes) and the jaw rotation. By using techniques to improve the performance of the muscle model and maintaining a certain level of simplicity of the whole face model, we have achieved fast animation on a standard personal computer.

Our approach is easy to apply to other facial meshes as the muscle representation is independent of the face mesh. Our face model can be used as a source face model that can be automatically deformed to represent any newly created face model. With the method we described in Chapter 3, we do not have to manually process another face model again when we want to produce animation on that face model. We introduced a novel method of automatically finding the training set of RBF networks to deform a source face model to represent a new face model. This was done by specifying and adjusting corresponding landmarks on a target face model automatically. The RBF networks were then used to transfer the muscles on the source face model to the deformed face model. Genetic Algorithms were used to adjust the landmarks on the target face model to minimize the difference between the surface of the deformed and the target face model. We defined a fitness function to assess the difference between the two models. We also presented an algorithm to calculate this function fast.

In Chapter 4, we discussed the problem of combining facial movements on a 3D talking head. We concatenated the movements in the same channel to generate smooth transitions between adjacent movements. Then the movements from all channels were combined taking into account the resolution of possible conflicting muscles. The activity of all muscles has been adjusted so that the muscles do not contract or release too fast. We have succeeded in creating natural facial animations of a talking head that can speak while displaying other facial movements such as conversational signals, manipulators and emotion displays as well.

In Chapter 5, we described ParleE, a quantitative, flexible and adaptive computational implementation of emotions for an embodied agent situated in a multi-agent environment. It is an implementation in which event appraisal is based on learning and a probabilistic planning algorithm. ParleE is based on a number of existing implementations of emotions. Nevertheless, ParleE has some significant properties of its own. The main novelties have to do with the way it uses forward-chaining search within a finite depth to obtain the probability of achieving a goal; the way it uses models of other agents' plans and goals to predict their behavior and set up expectations about the likelihood of events; and the way it incorporates personality, individual characteristics and motivational states in the implementation.

There are so many theories of emotion that it is hard for a computer scientist to choose an appropriate one to follow when implementing emotions. Moreover, there are continuous debates among the proponents of different views on emotions about the correctness of these theories. Therefore, the "ease-to-implement" is one of the main criteria for a computer scientist to select an appropriate theory of emotion. Two psychological models of emotion, Roseman's one (Roseman, 1984; Roseman et al., 1990, 1996) and that by Ortony, Clore and Collins (OCC) (Ortony et al., 1988), are used most by computer scientists because they are easy to implement on computers. Both models fit within the *cognitive* view on emotions. Like many of these and other implementations, ParleE generates emotions based on the OCC model. The OCC model of emotions provides a clearer view on emotions than Roseman's model. In Roseman's model, events are appraised only according to goals. This way, attitude and standard related emotions such as like/dislike, anger are not defined in a reasonable way. The OCC model treats these emotions more appropriately by appraising events based on goal, standard and preference. As the OCC model of emotions, like other theories of emotion, is a matter of dispute among psychologists, we do not claim for the trueness of generated emotions in ParleE.

In Chapter 6, we discussed the problem of generating facial expressions from emotional state. Our main concerns are dealing with emotions with different intensity and blends of emotions. Therefore, we have proposed a fuzzy rule-based system to achieve this aim. We have selected the *basic emotions view* from the psychology literature to base our system on. From this view, it is argued that a small set of basic emotions can be differentiated from each other by emotional facial expressions. Although this view is not without controversies, we have found it most useful in simulating emotional facial expressions. However, we certainly do not claim the psychological realism of our system. The motivation behind the work presented in this chapter is to provide a way to visualize the emotional state of an agent. Therefore, we did not model every human being's possible way of expressing emotion. In the current version of our system, we have modeled the expression of six emotions: **Happiness**, **Sadness**, **Anger**, **Fear**, **Disgust** and **Surprise** because there has been convincing evidence about the universalities in expressing these emotions. With simple fuzzy rules, lifelike emotional facial expressions are generated based on descriptions in the literature. The variations resulting from differences in the intensity of emotions are also successfully displayed. Our system can also be extended to take into account other ways of expressing those six basic emotions as well as the expression of other emotions if there is enough psychological evidence.

The systems which we have described in Chapters 2-6 have been combined to create an emotional embodied agent. In Chapter 7, we presented Obie, a football supporter agent. Obie has an emotion component which appraises events to trigger emotions. This component is the result of Chapter 5. Obie's emotions are expressed via his utterance or his facial expressions. The expression via utterance is done by a simple mapping from emotions to text fragments. The mapping from emotions to facial expressions was described in Chapter 6. Obie's utterance and facial expressions are presented in a 3D talking head, which is the result of Chapters 2 and 4. We have shown how Obie experiences different emotions during a football match. We have also indicated how Obie with different personalities experiences emotions differently.

Future direction

There are two issues that we think will improve our face model (Chapter 2) in the future. First, the face model still does not have a tongue model. The tongue contributes to the articulation of speech. The tongue is visible when the mouth is opened. Besides the shape of the lips, the position of the tongue is a supplement hint for recognizing what is being said. Therefore, a good tongue model would increase the realism of the facial animation. Second, how texture mapping would affect the quality of facial expressions and the speed of animation should be considered.

With respect to the dynamics of the face, we believe that it would be interesting to test other functions that simulate the onset and offset portion of a muscle's activity in Chapter 4. Although this is not very important for most of the facial movements, it may affect the believability of felt emotion displays. Finding the appropriate values of the onset and offset durations for felt emotion displays also plays an important role in making these displays natural, which is also an issue that needs to be investigated.

As a direction for future research, fuzzy logic can be implemented in ParleE (Chapter 5) to eliminate the hard border between negative and positive, between hope and fear, and so on. Love and hate are the two emotions from the OCC model that are not implemented in ParleE. Therefore, it is useful to find a suitable representation of an agent's appealingness to objects in order to trigger these two emotions. Finally, a more sophisticated belief component would improve the believability of the implementation.

The effect of the fuzzy membership function on the manner of expression in Chapter 6 is one of the issues for future research. A learning component can be added into the system so that we can have slightly different ways of expressing an emotion state for different agents. The expression mode selector should become more complex to take into account other factors besides intensity. In addition, other emotions besides the "universal" ones should be dealt with.

Finally, the agent described in Chapter 7 should be evaluated systematically.

Appendix A

Facial Action Coding System (FACS)

The FACS was developed by Ekman and Friesen (1978) to identify all possible visually distinguishable facial movements. All single Action Units in the FACS is presented in the following tables.

AU	Description	Facial muscle
1	Inner Brow Raiser	Frontalis, pars medialis
2	Outer Brow Raiser	Frontalis, pars lateralis
4	Brow Lowerer	Corrugator supercilii, Depressor supercilii
5	Upper Lid Raiser	Levator palpebrae superioris
6	Cheek Raiser	Orbicularis oculi, pars orbitalis
7	Lid Tightener	Orbicularis oculi, pars palpebralis
9	Nose Wrinkler	Levator labii superioris alaquae nasi
10	Upper Lip Raiser	Levator labii superioris
11	Nasolabial Deepener	Levator anguli oris (a.k.a. Caninus)
12	Lip Corner Puller	Zygomaticus major
13	Cheek Puffer	Zygomaticus minor
14	Dimpler	Buccinator
15	Lip Corner Depressor	Depressor anguli oris (a.k.a. Triangularis)
16	Lower Lip Depressor	Depressor labii inferioris
17	Chin Raiser	Mentalis
18	Lip Puckerer	Incisivii labii superioris and Incisivii labii inferioris

Table A.1: The Facial Action Coding System (part 1).

AU	Description	Facial muscle
20	Lip stretcher	Risorius w/ platysma
22	Lip Funneler	Orbicularis oris
23	Lip Tightener	Orbicularis oris
24	Lip Pressor	Orbicularis oris
25	Lips part	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris
26	Jaw Drop	Masseter, relaxed Temporalis and internal Pterygoid
27	Mouth Stretch	Pterygoids, Digastric
28	Lip Suck	Orbicularis oris
41	Lid droop**	Relaxation of Levator palpebrae superioris
42	Slit	Orbicularis oculi
43	Eyes Closed	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
44	Squint	Orbicularis oculi, pars palpebralis
45	Blink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
46	Wink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
51	Head turn left	
52	Head turn right	
53	Head up	
54	Head down	
55	Head tilt left	
56	Head tilt right	
57	Head forward	
58	Head back	
61	Eyes turn left	
62	Eyes turn right	
63	Eyes up	
64	Eyes down	

Table A.2: The Facial Action Coding System (part 2).

Appendix B

Detailed description of implemented facial muscles

Here is the detailed description of all implemented facial muscles in Chapter 2. This description is adopted from (Parke and Waters, 1996).

1. Zygomatic Major: is the greater zygomatic muscle which pulls the corners of the mouth outward and upward. The lower part of the nasolabial furrow is deepened while being pulled out and up.

2. Zygomatic Minor: is the lesser zygomatic muscle which pulls the lower part of the nasolabial furrow up and out. It also pulls the corners of the mouth up and out widening the mouth.

3. Triangularis: is the depressor of the corner of the mouth which pulls the corner of the mouth down. The depressor deepens the lower portion of the nasolabial furrow and pulls it downward.

4. Risorius: is the smiling mouth which produces a small depression of dimple in the cheeks where it is attached.

5. Depressor Labii: is the lower lip depressor which pulls the lower lip down and somewhat outward, slightly opening the mouth.

6. Mentalis: is the chin muscle which raises the soft parts of the chin, pressing the lower lip upwards. It emphasizes the chin-lip furrow.

7. Orbicularis Oris (Lip Funneler): is the lip part of the mouth sphincter muscle which constricts the mouth opening. If the red parts of the lips are relaxed, they are pushed out into a funnel shape.

8. Orbicularis Oris (Lip Pressor): is the margin part of the mouth sphincter muscle which constricts the mouth opening. It tightens the red parts of the lips, depresses the upper lip, and raises the lower lip. It can also roll the lips in over the teeth.

9. Frontalis Medialis: is the medical part of the frontalis muscle which raises the inner portion of the eyebrows.

10. Frontalis Lateralis: is the lateral part of the frontalis muscle which raises the outer portion of the eyebrows.

11. Levator Labii Nasi: is the upper lip and the nasal wing levator which pulls up the nasal wings. It raises and deepen the nasolabial furrows and the infraorbital furrow. It may raise the outer part of the upper lip.

12. Levator Labii Superioris: is the upper lip levator which pulls the outer part of the upper lip up and out. It pulls the upper and middle portion of the nasolabial furrow up and out and pushes the inner cheek up toward the eye.

13. Depressor Supercilli: is the eyebrow depressor which depresses the inner portion of the eyebrows.

14. Corrugator Supercilli: is the eyebrow wrinkler which depresses the middle portion of the eyebrows.

15. Depressor Glabelle: depresses the inner portion of the eyebrows together.

16. Levator Palpebrae Superios: raises the upper eyelids.

17. Orbcularis Oculi Palpebralis (Eye closing): is the eyelid part of the sphincter eye muscle which closes the upper eyelid.

18. Orbicular Oculi Orbitalis: is the orbital part of the sphincter eye muscle which squeezes the eyes closed. It can depress the eyebrows and raise the upper part of the cheeks. It also creates folds and furrows around the eyes.

Appendix C

The football supporter domain

Possible facts in the football supporter domain are represented in STRIPS as follows:

- Position of the ball:
    - (ball-penaltyr team) — the ball is in the penalty area of the **for team**.
    - (ball-hfield team) — the ball is in the half field of the **for team**.
    - (ball-hfield oteam) — the ball is in the half field of the **against team**.
    - (ball-penaltyr oteam) — the ball is in the penalty area of the **against team**.
- Who is controlling the ball:
    - (ball-control team) — the **for team** is controlling the ball.
    - (ball-control oteam) — the **against team** is controlling the ball.
- Who is acting (for supporting multi-agent planning as described above):
    - (acting team) — it is the team's turn.
    - (acting agent) — it is the supporter agent's turn.
- Match status:
    - (no-penalty team) — it is not the penalty for the **for team**.
    - (penalty team) — it is the penalty for the **for team**.
    - (no-penalty oteam) — it is not the penalty for the **against team**.
    - (penalty oteam) — it is the penalty for the **against team**.

  - (no-freekick team) — it is not the free-kick for the **for team**.
  - (freekick team) — it is the free-kick for the **for team**.
  - (no-freekick oteam) — it is not the free-kick for the **against team**.
  - (freekick oteam) — it is the free-kick for the **against team**.
  - (no-kickoff) — it is not the kick-off.
  - (kickoff) — it is the kick-off.

- Score:
  - . . .
  - (score minus-one)
  - (score zero)
  - (score one)
  - . . .

After defining the possible facts, the possible actions for a team are:

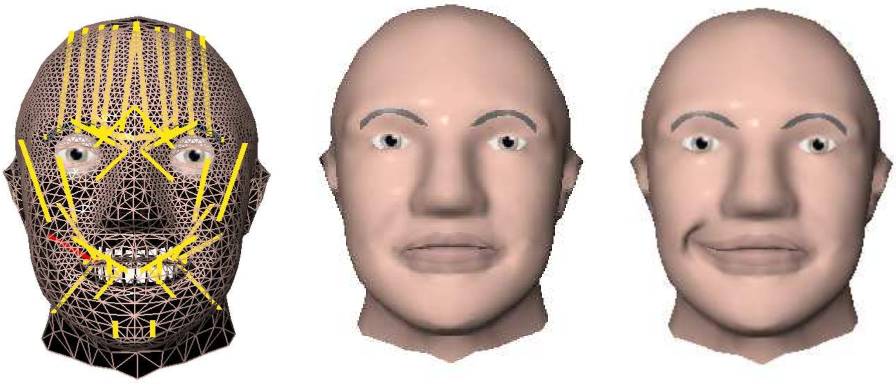
- Attacking actions (the team is controlling the ball):
  - *long-shot* — this action is available when the ball is in the other team's half filed.
  - *shot* — this action is available when the ball is in the other team's penalty area. This action has higher possibility to lead to a goal then the *long-shot* one.
- Passing actions (the team is controlling the ball):
  - *pass-from-pr* — pass when the ball is in the team's penalty area.
  - *pass-from-hf* — pass when the ball is in the team's half field.
  - *pass-from-ohf* — pass when the ball is in the other team's half field.
  - *pass-from-opr* — pass when the ball is in the other team's penalty area.
  - *kick-away-from-pr* — kick the ball away when the ball is in the team's penalty area.
- Defending actions (the team is not controlling the ball):
  - *tackle-in-pr* — tackle when the ball is in the team's penalty area.
  - *tackle-in-hf* — tackle when the ball is in the team's half field.
  - *tackle-in-ohf* — tackle when the ball is in the other team's half field.
  - *tackle-in-opr* — tackle when the ball is in the other team's penalty area.

  - *block-in-pr* — block the ball when the ball is in the team's penalty area.
  - *block-in-hf* — block the ball when the ball is in the team's half field.
  - *block-in-ohf* — block the ball when the ball is in the other team's half field.
  - *block-in-opr* — block the ball when the ball is in the other team's penalty area.

- Penalty, free kick, and kick-off related actions:
  - *defend-penalty* — defend penalty.
  - *defend-freekick-in-hf* — defend free-kick in the team's half field.
  - *defend-freekick-in-ohf* — defend free-kick in the other team's half field.
  - *defend-freekick-in-opr* — defend free-kick in the other team's penalty area.
  - *take-penalty* — take penalty.
  - *take-freekick-in-hf* — take free-kick in the team's half field.
  - *take-freekick-in-ohf* — take free-kick in the other team's half field.
  - *take-freekick-in-pr* — take free-kick in the team's penalty area.
  - *re-kickoff* — start kick-off.

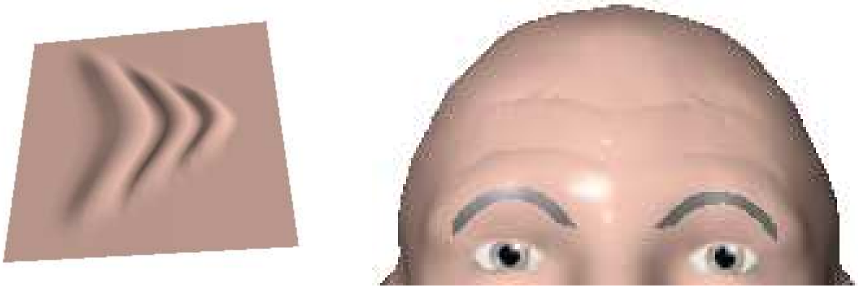
Appendix D

Colored illustrations



(a) (b) (c)

Figure D.1: (a) The wire-framed face with muscles; (b) the neutral face; and (c) the effect of the left Zygomatic Major.



(a) (b)

Figure D.2: The wrinkles due to the muscle contraction.

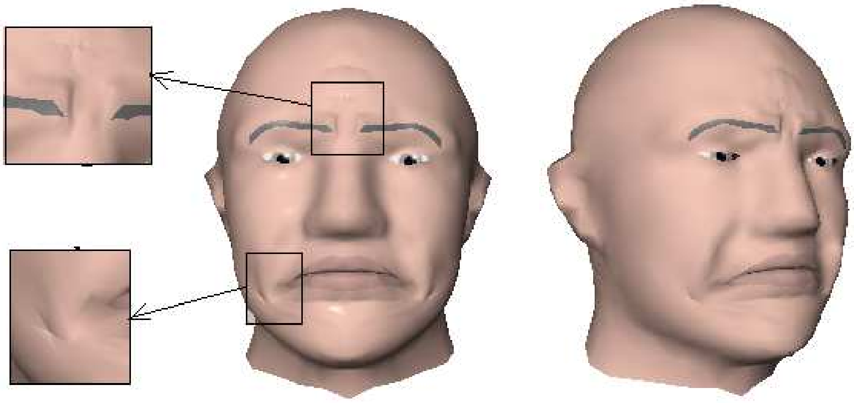
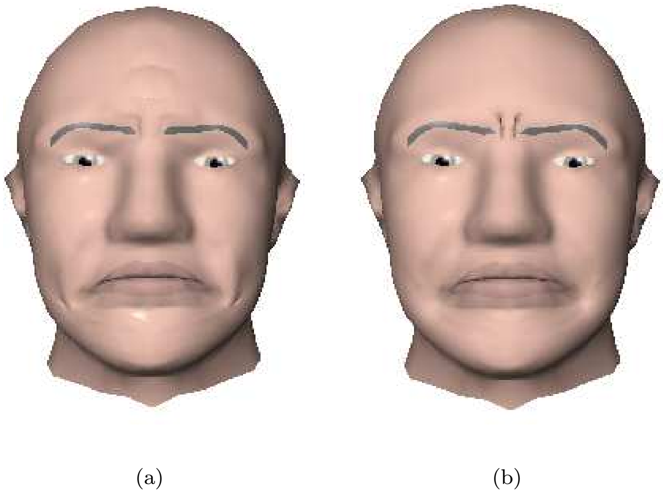


Figure D.3: The face model displays sadness.



(a) (b)

Figure D.4: Sadness on our face model with (a) and without (b) wrinkles and handling multiple muscle contractions.

Colored illustrations

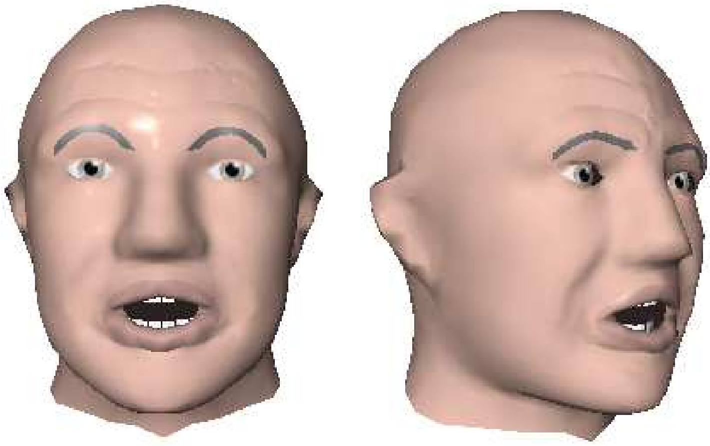


Figure D.5: The face model displays surprise.

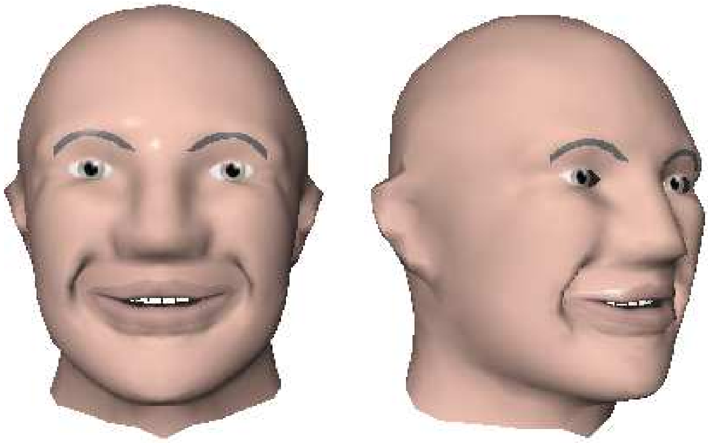


Figure D.6: The face model displays happiness.

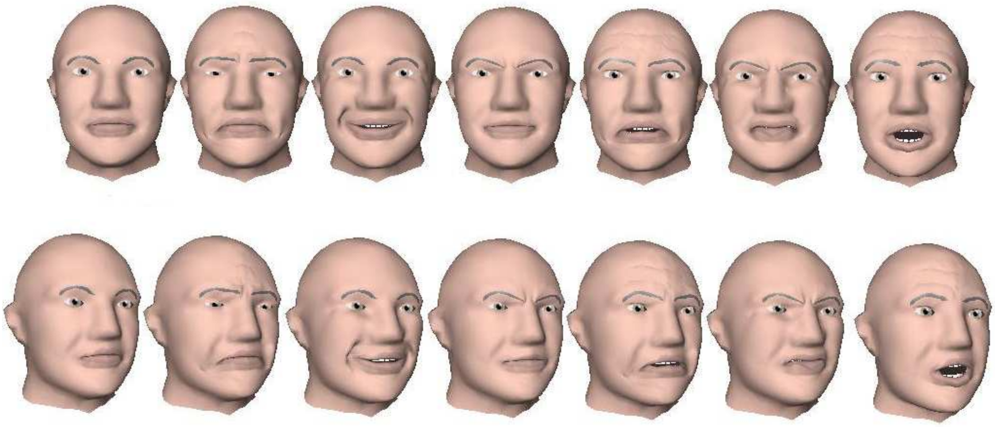


Figure D.7: Basic emotions: neutral, **Sadness**, **Happiness**, **Anger**, **Fear**, **Disgust**, **Surprise** (from left to right).

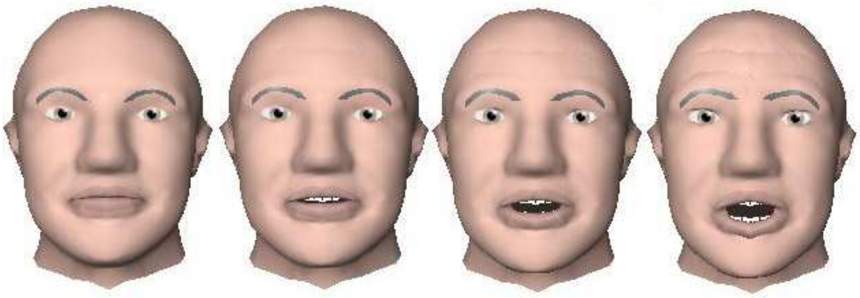


Figure D.8: Increasing **Surprise**.

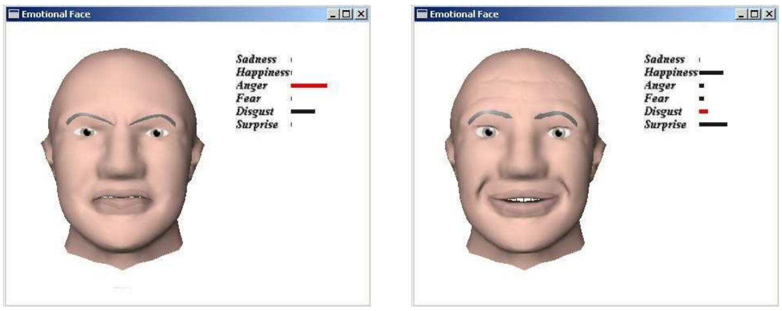


Figure D.9: Blends of **Anger** and **Disgust** (left), **Happiness** and **Surprise** (right).

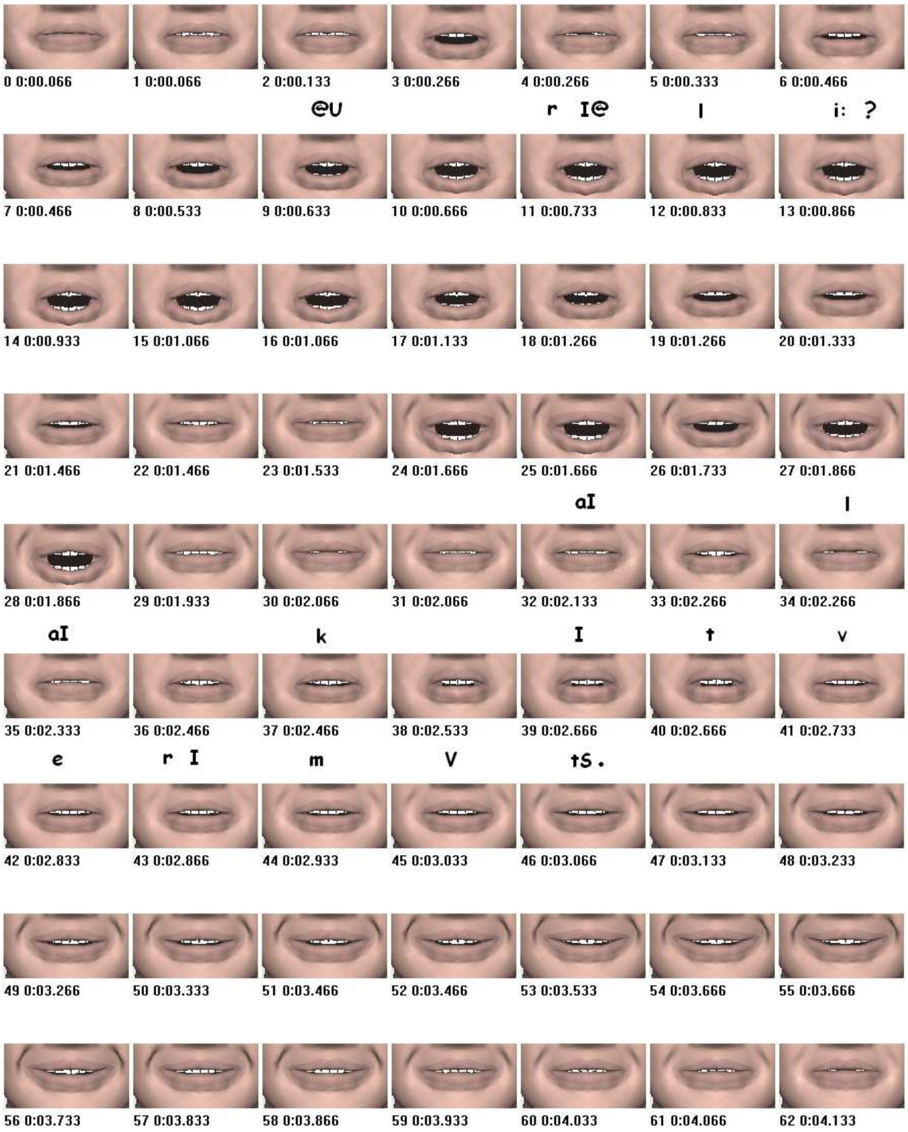


Figure D.10: Our talking head utters the sentence “Oh, really? I like it very much.” while displaying surprise followed by happiness (frame by frame).

Bibliography

Ahlberg, J. (2001). Candide-3 – an updated parameterized face. Technical Report Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linkping University, Sweden.

Albrecht, I., Haber, J., Kähler, K., Schröder, M., and Seidel, H. P. (2002a). May i talk to you? :-) facial animation from text. In *Proceedings Pacific Graphics 2002*, pages 77–86.

Albrecht, I., Haber, J., Schröder, M., and Seidel, H. P. (2002b). Automatic generation of non-verbal facial expressions from speech. In *Proceedings Computer Graphics International (CGI) 2002*, pages 283–293.

Allport, F. H. (1924). *Social psychology.* Houghton Mifflin, Boston.

Allport, G. W. and Odbert, H. S. (1936). Trait names: a psycho-lexical study. *Psychological Monographs*, 47(211).

André, E., Herzog, G., and Rist, T. (1997). Generating multimedia presentations for robocup soccer games. In Kitano, H., editor, *RoboCup '97: Robot Soccer World Cup I*, pages 200–215. Springer-Verlag, New York.

Argyle, M. (1990). *Bodily Communication.* Routledge, London.

Arnold, M. B. (1960). *Emotion and personality: Vol. 1(2). Psychological aspects.* Columbia University Press, New York.

Averill, J. R. (1980). A constructivist view of emotion. *Emotion: Theory, research and experience*, I:305–339.

Basmajian, J. (1974). *Muscles Alive.* The Williams & Wilkins Co., Baltimore.

Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford.

Blinn, J. F. (1978). Simulation of wrinkled surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12 (3), pages 286–292.

Blum, A. and Langford, J. (1998). Probabilistic planning in the graphplan framework. In *AIPS98 Workshop on Planning as Combinatorial Search*, pages 8–12.

Blumberg, B., Downie, M., Ivanov, Y. A., Berlin, M., Johnson, M. P., and Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. In *SIGGRAPH 2002*, pages 417–426.

Bolles, R. and Fanselow, M. (1980). A percuptual defensive recuperative model of fear and pain. *Behav. Brain Scie.*, 3:291–301.

Booker, L. (1987). Improving search in genetic algorithms. In Davis, L., editor, *Genetic Algorithms and Simulated Annealing*, pages 61–73. Pitman Publishing, London.

Bozinovski, S. and Bozinovska, L. (1999). Emotion as internal state evaluation: A connectionist emotion-based learning architecture. In *Proceedings of the Autonomous Agents Workshop on Emotion-Based Agent Architectures (EBAA'99)*.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2.

Bui, T. D. and Heylen, D. (2003). How an emotional agent acts in a multi-agent environment. In *Second Int. Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore.

Bui, T. D., Heylen, D., and Nijholt, A. (2003a). Improvements on a simple muscle-based 3d face for realistic facial expressions. In *16th International Conference on Computer Animation and Social Agents (CASA-2003)*, pages 33–40, Los Alamos, CA. IEEE Computer Society.

Bui, T. D., Heylen, D., and Nijholt, A. (2004a). Building embodied agents that experience and express emotions: A football supporter as an example. In *Proceedings 17th annual conference on Computer Animation and Social Agents (CASA2004)*. Computer Graphics Society. to appear.

Bui, T. D., Heylen, D., and Nijholt, A. (2004b). Combination of facial movements on a 3d talking head. In *Computer Graphics International (CGI 2004)*. IEEE Computer Society. to appear.

Bui, T. D., Heylen, D., Poel, M., and Nijholt, A. (2001). Generation of facial expressions from emotion using a fuzzy rule based system. In *Australian Joint Conference on Artificial Intelligence (AI 2001)*, pages 83–95, Berlin. Lecture Notes in Computer Science, Springer.

Bui, T. D., Heylen, D., Poel, M., and Nijholt, A. (2002). Parlee: An adaptive plan-based event appraisal model of emotions. In *KI 2002: Advances in Artificial Intelligence*, pages 129–143, Berlin. Lecture Notes in Computer Science, Springer.

Bui, T. D., Heylen, D., Poel, M., and Nijholt, A. (2003b). Exporting vector muscles for facial animation. In Butz, A., Krger, A., and Olivier, P., editors, *Proceedings International Symposium on Smart Graphics 2003*, Berlin. Lecture Notes in Computer Science, Springer.

Bui, T. D. and Jamroga, W. (2003). Multi-agent planning with planning graph. In *Proceedings of eunite 2003*, Oulu, Finland.

Bui, T. D., Poel, M., Heylen, D., and Nijholt, A. (2003c). Automatic face morphing for transferring facial animation. In *Proceedings 6th IASTED International Conference on Computers, Graphics, and Imaging (CGIM 2003)*, Anaheim/Calgary/Zurich. ACTA Press.

Bui, T. P. (1975). Illumination for computer generated pictures. *Communications of the ACM*, 18(6).

Carmel, D. and Markovitch, S. (1996). Learning and using opponent models in adversary search. Technical Report CIS9609, Technion, Haifa.

Casell, J. (2000). Nudge nudge wink wink: Elements of face-to-face conversation for embodied conversational agents. In Cassell, J., Sullivan, J., Prevost, S., and Churchill, E., editors, *Embodied Conversational Agents*, pages 1–27. The MIT Press.

Cassell, J., Sullivan, J., Prevost, S., and Churchill, E., editors (2000). *Embodied Conversational Agents*. The MIT Press.

Cattell, R. B. (1946). *The description and measurement of personality*. Harcourt, Brace and World, New York.

Cohen, M. M. and Massaro, D. W. (1993). Modeling coarticulation in synthetic visual speech. In Magnenat Thalmann, N. and Thalmann, D., editors, *Models and Techniques in Computer Animation*, pages 139–156. Springer, Tokyo.

Coons, S. A. (June 1967). Surfaces for computer aided design of space forms. Technical Report Project NAC Report MAC-TR-41, M.I.T, Cambrigdge, MA.

Cornelius, R. (1996). *The Science of Emotion*. Upper Saddle River, NJ.

Costa, P. T. and McCrae, R. R. (1992). *NEO PI-R: Professional Manual*. Psychological Assessment Resources, Odessa, FL.

Coulter, J. (1989). *Mind in action*. Humanities Press International, Atlantic Highlands, NJ.

Damasio, A. (1994). *Descartes' error: Emotion, reason, and the human brain*. G.P. Putnam, New York.

Darwin, C. (1872/1965). *The expression of the emotions in man and animals*. Univerity of Chicago Press, Chicago.

Davis, L. (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

DeCarlo, D. C., Revilla, M. S., and Venditti, J. (2002). Making discourse visible: Coding and animating conversational facial displays. In *Computer Animation 2002.*

Doenges, P., Lavagetto, F., Ostermann, J., Pandzic, I. S., and Petajan, E. (1997). Mpeg-4: Audio/video and synthetic graphics/audio for real-time, interactive media delivery. *Image Communications Journal*, 5(4).

Dutoit, T., Pagel, V., Pierret, N., van der Vreken, O., and Bataille, F. (1996). The mbrola project: Towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In *Proc. ICSLP 96.*

Eck, M. (1991). Interpolation methods for reconstruction of 3d surfaces from sequences of planar slices. *CAD und Computergraphik*, 13(5).

Ekman, P. (1972). Universals and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation 1971*, 19.

Ekman, P. (1973). *Darwin and facial expression: A century of research in review.* Academic Press, New York.

Ekman, P. (1977). Biological and cultural contributions to body and facial movement. In Blacking, J., editor, *The anthropology of the body.* Academic Press, London.

Ekman, P. (1982). *Emotion in the human face.* Cambridge University Press, Cambridge.

Ekman, P. (1984). Expression and the nature of emotion. In Scherer, K. and Ekman, P., editors, *Approaches to Emotion.* Lawrence Erlbaum, Hillsdale, NJ.

Ekman, P. (1989). The argument and evidence about universals in facial expressions of emotion. In Wagner, H. and Manstead, A., editors, *Handbook of Social Psychophysiology.* Wiley, Chichester, New York.

Ekman, P. and Friesen, W. V. (1975). *Unmasking the Face: A Guide To Recognizing Emotions From Facial Clues.* Prentice-Hall, Englewood Cliffs, New Jersey.

Ekman, P. and Friesen, W. V. (1978). *Facial Action Coding System.* Consulting Psychologists Press, Palo Alto, CA.

Ekman, P., Huang, T., Sejnowski, T., and Hager, J., editors (1993). *Final Report to NSF of the Planning Workshop on Facial Expression Understanding.* Human Interaction Laboratory, University of California, San Francisco.

Ekman, P., Levenson, R. W., and Friesen, W. V. (1983). Autonomic nervous system activity distinguishes between emotions. *Science*, 221.

El-Nasr, M. S., Y., J., and Ioerger, T. R. (2000). FLAME-fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3):219–257.

Elliott, C. (1992). *The Affective Reasoner: A Process model of emotions in a multi-agent system.* PhD thesis, Northwestern University, Evanston, IL.

Enciso, R., Li, J., Fidaleo, D., Kim, T.-Y., Noh, J.-Y., and Neumann, U. (2000). Synthesis of 3d faces. In *International Workshop on Digital and Computational Video.*

Essa, I. A. (1994). *Analysis, Interpretation, and Synthesis of Facial Expressions.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Fernández-Dols, J. M. and Carroll, J. M. (1997). Is the meaning perceived in facail expression independent of its context? In Russell, J. A. and Fernández-Dols, J. M., editors, *The Psychology of Facial Expression.* Cambridge University Press, New York, NY.

Fikes, R. and Nilsson, N. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4).

Fiske, D. W. (1949). Consistency of the factorial structures of personality ratings from different sources. *Journal of Abnormal Social Psychology*, 44.

Forgas, J. P. and Moylan, S. (1987). After the movies: the effects of transient mood states on social judgments. *Personality and Social Psychology Bulletin*, 13.

Fridlund, A. J. (1994). *Human facial expression: An evolutionary view.* Academic, New York.

Frijda, N. H. (1986). *The emotions.* Cambridge University Press, Cambridge.

Galernter, D. (1994). *The muse in the machine.* Free Press, New York.

Gergen, K. J. (1985). The social constructionist movement in modern psychology. *American Psychologist*, 40.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA.

Goleman, D. (1995). *Emotional Intelligence.* Bantam Books, New York.

Gouraud, H. (1971). *Computer display of curved surfaces.* PhD thesis, Dept. of Electrical Engineering, University of Utah.

Gratch, J. (2000). Émile: Marshalling passions in training and education. In C. Sierra, M. G. and Rosenschein, J. S., editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 325–332, Barcelona, Catalonia, Spain. ACM Press.

Grings, W. W. and Dawson, M. E. (1978). *Emotions and bodily responses: a psychophysiological approach.* Academic Press, New York.

Hager, J. and Ekman, P. (1996). Essential behavioral science of the face and gesture that computer scientists need to know. In *International Workshop on Automatic Face and Gesture Recognition.*

Hayes-Roth, B. and van Gent, R. (1997). Story-making with improvisational puppets. In Johnson, W. L. and Hayes-Roth, B., editors, *Proceedings of the 1st International Conference on Autonomous Agents*, pages 1–7, New York. ACM Press.

Hendrix, J., Ruttkay, Z., ten Hagen, P., Noot, H., Lelievre, A., and de Ruiter, B. (2000). A facial repertoire for avatars. In *Proceedings of the Workshop 'Interacting Agents'*, Enschede, The Netherlands.

Hoch, M., Fleischmann, G., and Girod, B. (1994). Modeling and animation of facial expressions based on b-splines. *Visual Computer*, 11.

Inoue, K., Kawabata, K., and Kobayashi, H. (1996). On a decision making system with emotion. In *IEEE Int. Workshop on Robot and Human Communication.*

Itti, L., Dhavale, N., and Pighin, F. (2003). Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proc. SPIE 48th Annual International Symposium on Optical Science and Technology.*

Izard, C. (1971). *The face of emotion.* Appleton-Century-Crofts, New York.

Izard, C. (1981). Differential emotions theory and the facial feedback hypothesis activation: Comments on tourangeau and ellsworth's 'the role of facial response in experience of emotion'. *Journal of Personality and Social Psychology*, 40:350–354.

Izard, C. (1990). The substrates and functions of emotion feelings: William james and current emotion theory. *Personality and Social Psychology Bulletin*, 16(4):626–635.

Izard, C. (1991). *The psychology of emotions.* Plenum, New York.

Izard, C. (1994). Innate and universal facial expressions: Evidence from developmental and cross-cultural research. *Psychological Bulletin*, 115(2):288–299.

Izard, C. (1997). Emotions and facial expressions: A perspective from differential emotions theory. In Russell, J. and Fernandez-Dols, J., editors, *The Psychology of Facial Expression.* Maison des Sciences de l'Homme and Cambridge University Press.

James, W. (1884). What is an emotion. *Mind*, 19:188–205.

Jong, F. D. and Westerveld, T. (2001). Mumis: Multimedia indexing and searching. In *Proceedings of the Content-Based Multimedia Indexing workshop (CBMI 2001).*

Kähler, K., Haber, J., and Seidel, H. P. (2001). Geometry-based muscle modeling for facial animation. In Watson, B. and Buchanan, J. W., editors, *Proceedings of Graphics Interface 2001*, pages 37–46.

Kähler, K., Haber, J., Yamauchi, H., and Seidel, H. P. (2002). Head shop: Generating animated head models with anatomical structure. In Spencer, S. N., editor, *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation (SCA-02)*, pages 55–64, New York. ACM Press.

Kalra, P. (1993). *An Interactive Multimodal Facial Animation System.* PhD thesis, University of Geneva and EPFL.

Kalra, P., Mangili, A., Magnenat-Thalmann, N., and Thalmann, D. (1992). Simulation of facial muscle actions based on rational free form deformations. In Kilgour, A. and Kjelldahl, L., editors, *Computer Graphics Forum (EUROGRAPHICS '92 Proceedings)*, volume 11 (3), pages 59–69.

Kappas, A. (2003). What facial activity can and cannot tell us about emotions. In Katsikitis, M., editor, *The human face: Measurement and meaning*, pages 215–234. Kluwer Academic Publishers, Dordrecht.

King, S. A., Parent, R. E., and Olsafsky, B. (2000). An anatomically-based 3d parametric lip model to support facial animation and synchronized speech. In *Proceedings of Deform 2000*, pages 7–19.

Koch, R. M. and Bosshard, A. A. (1998). Emotion editing using finite elements. *Computer Graphics Forum*, 17(3).

Koch, R. M., Gross, M. H., Carls, F. R., von Büren, D. F., Fankhauser, G., and Parish, Y. I. H. (1996). Simulating facial surgery using finite element models. *Computer Graphics*, 30:421–428.

Kooij, N. S. (2003). The development of a vision system for robotic soccer. Master's thesis, Dept. of Computer Sciene, University of Twente, The Netherlands.

Laird, J. D. and Bresler, C. (1990). William james and the mechanisms of emotional experience. *Personality and Social Psychology Bulletin*, 16.

Laird, J. E. and van Lent, M. (2000). Human-level ai's killer application: Interactive computer games. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press / The MIT Press.

Lange, C. G. (1885/1922). The emotions: A psychophysiological study. *The emotions*, pages 33–90.

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

Lazarus, R. S. (1991). Progress on a cognitive-motivational-relational theory ofs emotion. *American Psychologist*, 46.

Lazarus, R. S. and Alfert, E. (1964). Short circuiting of threat by experimentally altering cognitive appraisal. *Journal of Abnormal and Social Psychology*, 69.

Lazarus, R. S., Averill, I. R., and Opton, E. M. (1970). Feeling and emotion. In Arnold, M. B., editor, *Toward a Cognitive Theory of Emotion*, pages 207–232. Academic Press, New York.

LeDoux, J. (1996). *The emotional brain*. Simon & Schuster.

Lee, Y., Terzopoulos, D., and Waters, K. (1995). Realistic face modeling for animation. In Cook, R., editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 55–62. ACM SIGGRAPH, Addison Wesley.

Levenson, R. W. (1992). Autonomic nervous system differences among emotions. *Psychological Science*, 3:23–27.

Lisetti, C. L. (1999). Emotion generation for artificial agents via a hybrid architecture. In *Proceedings of the Autonomous Agents Workshop on Emotion-Based Agent Architectures (EBAA'99)*.

Löfqvist, A. (1990). Speech as audible gestures. *Speech Production and Speech Modeling*, pages 289–322.

Loyall, A. B. and Bates, J. (1997). Personality-rich believable agents that use language. In Johnson, W. L. and Hayes-Roth, B., editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 106–113, Marina del Rey, CA, USA. ACM Press.

Magnenat-Thalmann, N., P, P. K., and Pandzic, I. S. (1995). Direct face-to-face communication between real and virtual humans. *International Journal of Information Technology*, 1(2):145–157.

Magnenat-Thalmann, N., Primeau, E., and Thalmann, D. (1988). Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5):290–297.

Mahl, R. (1972). Visible surface algorithm for quadric patches. *IEEE Transactions on Computers*, C-21(1):1–4.

Mandler, G. (1975). *Mind and Emotion.* Wiley, New York.

Mani, M. V. and Ostermann, J. (2001). Cloning of mpeg-4 face models. In *International Workshop on Very Low Bit rate Video Coding (VLBV01).*

McCrae, R. R. and Costa, P. T. (1987). Validation of the five-factor model of personality across instruments and observers. *Journal of Personality and Social Psychology*, 52:81–90.

Minsky, M. (1988). *The Society of Mind.* Touchstone.

Moubaraki, L., Ohya, J., and Kishino, F. (1995). Realistic 3d facial animation in virtual space teleconferencing. In *4th IEEE International workshop on Robot and Human Communication.*

Nijholt, A. (2001). Towards communicating 3d embodied agents. In *JSAI-Synsophy International Workshop on Social Intelligence Design*, Matsue, Shimane, Japan.

Nijholt, A., op den Akker, H. J. A., and de Jong, F. M. G. (2003). Language interpretation and generation for football commentary. In Miyares, L. R., Alvarez, C. E., and Moreno, A., editors, *ACTAS-1: VIII Symposio Social*, pages 594–599.

Noh, J. and Neumann, U. (2001). Expression cloning. In Fiume, E., editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 277–288. ACM Press / ACM SIGGRAPH.

Norman, W. T. (1963). Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *Journal of Abnormal and Social Psychology*, 66:574–583.

Nye, J. L. and Brower, A. M. (1996). *What's Social About Social Cognition?* Sage Publication, London.

Oatley, K. (1992). *Best Laid Schemes: The Psychology of Emotions.* Cambridge UP, Cambridge.

Oatley, K. and Johnson-Laird, P. N. (1987). Towards a cognitive theory of emotions. *Cognition and Emotion*, 1(1):29–50.

Oka, M., Tsutsui, K., Ohba, A., Karauchi, Y., and Tago, T. (1987). Real-time manipulation of texture-mapped surfaces. *SIGGRAPH*, 87:181–188.

Ortony, A. (2002). Emotions in humans and artifacts. In Trappl, R., Petta, P., and Payr, S., editors, *On making believable emotional agents believable*. MIT Press, Cambridge, MA.

Ortony, A., Clore, G. L., and Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, England.

Paradiso, A. and L'Abbate, M. (2001). A model for the generation and combination of emotional expressions. In Pelachaud, C. and Poggi, I., editors, *Proceedings of the Workshop on Multimodal Communication and Context in Embodied Agents*, Montreal, Canada.

Parke, F. I. (1972). Computer generated animation of faces. Master's thesis, Dept. of Electrical Engineering, Division of Computer Science, University of Utah.

Parke, F. I. (1974). *A parametric model for human faces*. PhD thesis, University of Utah.

Parke, F. I. and Waters, K. (1996). *Computer Facial Animation*. A K Peters. ISBN 1-56881-014-8.

Pasquariello, S. and Pelachaud, C. (2001). Greta: A simple facial animation engine. In *6th Online World Conference on Soft Computing in Industrial Appications, Session on Soft Computing for Intelligent 3D Agents*.

Pelachaud, C., Viaud, M. L., and Yahia, H. (1993). Rule-structured facial animation system. *IJCAI*, 2:1610–1615.

Pelechaud, C., Badler, N. I., and Steedman, M. (1996). Generating facial expressions for speech. *Cognitive Science*, 20:1–46.

Perlin, K. (1997). Layered compositing of facial expression, siggraph'97 technical sketch. Paper Availbale:$< http://mrl.nyu.edu/improv/sig97-sketch/ >$, New York University Media Research Lab.

Perlin, K. and Goldberg, A. (1996). Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, 30(Annual Conference Series):205–216.

Picard, R. (1997). *Affective Computing*. MIT Press, Cambridge, MA.

Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., and Salesin, D. H. (1998). Synthesizing realistic facial expressions from photographs. In Cohen, M., editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 75–84. ACM SIGGRAPH, Addison Wesley.

Platt, S. M. (1985). *A Structural Model of the Human Face*. PhD thesis, University of Pennylvania.

Platt, S. M. and Badler, N. I. (1981). Animating facial expression. *Computer Graphics*, 13(3):245–252.

Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. In Scherer, K. R. and Ekman, P., editors, *Approaches to emotion.* Lawrence Erlbaum, London.

Prevost, S. and Pelachaud, C. (1995). Talking heads: Physical, linguistic and cognitive issues in facial animation. Technical Report Course Notes for Computer Graphics International 1995, Leeds, UK.

Price, D. D., Barrell, J. E., and Barrell, J. J. (1985). A quantitative-experiential analysis of human emotions. *Motivation and Emotion*, 9(1).

Reeves, W. T. (1990). Simple and complex facial animation: Case studies. In *State of the Art in Facial Animation: SIGGRAPH 1990 Course Notes #26*, pages 88–106. 17th International Conference on Computer Graphics and Interactive Techniques.

Reilly, W. S. (1996). Believable social and emotional agents. Technical Report Ph.D. Thesis. Technical Report CMU-CS-96-138, Carnegie Mellon University, Pittsburgh, PA, USA.

Reilly, W. S. and Bates, J. (1992). Building emotional agents. Technical Report CMU-CS-92-143, Carnegie Mellon University, Pittsburgh, PA, USA.

Rickel, J. and Johnson, W. L. (1998). Steve: A pedagogical agent for virtual reality. In *Proceedings of the Second International Conference on Autonomous Agents.*

Roseman, I. (1984). Cognitive determinants of emotions: A structural theory. *Review of Personality and Social Psychology*, 5.

Roseman, I. J., Antoniou, A. A., and Jose, P. E. (1996). Appraisal determinants of emotions: Constructing a more accurate and comprehensive theory. *Cognition and Emotion*, 10.

Roseman, I. J., Jose, P. E., and Spindel, M. S. (1990). Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, 59(5).

Rosis, F. D., Pelachaud, C., Poggi, I., Carofiglio, V., and Carolis, B. D. (2003). From greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. *International Journal of Human-Computer Studies*, 59(1-2).

Rousseau, D. (1996). Personality in computer characters. In *Proceedings of the 1996 AAAI Workshop on Entertainment and AI A-Life*, pages 38–43, Portland, Oregon. AAAI Press.

Russell, J. A. (1997). Reading emotions from and into faces: Resurrecting a dimensional-contextual perspective. In Russell, J. A. and Fernández-Dols, J. M., editors, *The Psychology of Facial Expression*. Cambridge University Press, New York, NY.

Russell, J. A. and Fernández-Dols, J. M. (1997). What does a facial expression mean? In Russell, J. A. and Fernández-Dols, J. M., editors, *The Psychology of Facial Expression*. Cambridge University Press, New York, NY.

Rydfalk, M. (1987). Candide, a parameterized face. Technical Report Report No. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linkping University, Sweden.

Scherer, K. R. (1992). What does facial expression express? In Strongman, K., editor, *International Review of Studies on Emotion*, volume 2. Wiley, Chichester.

Scherer, K. R. (2001). Appraisal considered as a process of multilevel sequential checking. In Scherer, K., Schorr, A., and Johnstone, T., editors, *Appraisal processes in emotion*. Oxford University Press, Oxford.

Sederberg, T. W. and Parry, S. R. (1986). Free-form deformation of solid geometric models. In Evans, D. C. and Athay, R. J., editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20 (4), pages 151–160.

Seesink, R., Dierssen, W., Kooij, N., Schoute, A., Poel, M., Schepers, E., and Verschoor, T. (2003). Fast data sharing within a distributed multithreaded control framework for robot teams. In *FIRA World Congress*.

Sen, S. and Weiss, G. (1999). Learning in multiagent systems. In Weiss, G., editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, pages 259–298. Cambridge, Mass, MIT Press.

Shaver, P. R., Wu, S., and Schwartz, J. C. (1992). Cross-cultural similarities and differences in emotion and its representation: A prototype approach. *Review of Personality & Social Psychology*, 13.

Shaw, I. S. (1998). *Fuzzy Control of Industrial Systems : Theory and Applications*. Kluwer Academic Publishers.

Shields, S. A. and Stern, R. M. (1979). Emotion: The perception of bodily change. In Pliner, P., Blankstein, K. R., and Spigel, I. M., editors, *Perception of emotion in self and others*, pages 85–106. Plenum, New York, NY.

Sigman, M. and Capps, L. (1997). *Children with Autism: A Developmental Perspective*. Harvard University Press, Cambridge, MA.

Sloman, A. (2002). Emotions in humans and artifacts. In Trappl, R., Petta, P., and Payr, S., editors, *How many separately evolved emotional beasties live within us*. MIT Press, Cambridge, MA.

Smart, J. (2000). http://www.anthemion.co.uk/speech.htm.

Smith, C. A. (1989). Dimensions of appraisal and physiological response in emotion. *Journal of Personality and Social Psychology*, 56:339–353.

Spears, W. M. and DeJong, K. A. (1991). An analysis of multi-point crossover. In Rawlings, G. J. E., editor, *Foundations of genetic algorithms*, pages 301–315. Morgan Kaufmann, San Mateo.

Speisman, J. C., Lazarus, R. S., Mordkoff, A., and Davison, L. (1964). Experimental reduction of stress based on ego-defensive theory. *Journal of Abnormal and Social Psychology*, 68:367–380.

Stern, A., Frank, A., , and Resner, B. (1998). Virtual petz: A hybrid approach to creating autonomous, lifelike dogz and catz. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 334–335, New York. ACM Press.

Sweetser, P., Johnson, D., Sweetser, J., and Wiles, J. (2003). Creating engaging artificial characters for games. In *Proceedings of the second international conference on Entertainment computing*, pages 1–8, Pittsburgh, PA, USA. Carnegie Mellon University.

Terzopoulos, D. and Waters, K. (1990). Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation*, 1(2):73–80.

Theune, M., Faas, S., Heylen, D., and Nijholt, A. (2003). The virtual storyteller: Story creation by intelligent agents. In *Proceedings TIDSE 03: Technologies for Interactive Digital Storytelling and Entertainment*, pages 204–215. Fraunhofer IRB Verlag.

Thomas, F. and Johnston, O. (1981). *The Illusion of Life.* Abbeville Press, New York.

Tomkins, S. S. (1962). *Affect, Imagery, Consciousness (Volume 1): The Positive Affects.* Springer, New York.

Tomkins, S. S. (1963). *Affect, Imagery, Consciousness (Volume 1): The Negative Affects.* Springer, New York.

Trappl, R. and Petta, P., editors (1997). *Creating Personalities for Synthetic Actors. Towards Autonomous Personality Agents.* Springer, Berlin, Berlin, Heidelberg, New York.

Tupes, E. C. and Christal, R. E. (1961). Recurrent personality factors based on trait ratings. Technical Report (ASD-TR-61-97), Lackland Air Force Base, TX: Aeronautical Systems Division, Personnel Laboratory.

Ulgen, F. (1997). A step toward universal facial animation via volume morphing. In *6th IEEE International Workshop on Robot and Human communication*, pages 358–363.

Velásquez, J. D. (1996). Cathexis–a computational model for the generation of emotions and their influence in the behavior of autonomous agents. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science.

Velásquez, J. D. (1997). Modeling emotions and other motivations in synthetic agents. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, pages 10–15, Menlo Park. AAAI Press.

Viaud, M. and Yahia, H. (1992). Facial animation with wrinkles. In Hegron, G. and Thalmann, D., editors, *Computer Animation and Simulation '92*, Eurographics, Cambridge, England.

Waite, C. (1989). The facial action control editor, face: A parametric facial expression editor for computer generated animation. Master's thesis, MIT, Media Arts and Sciences.

Wang, C. L. (1993). Langwidere: A hierarchical spline based facial animation system with simulated muscles. Master's thesis, University of Calgary.

Waters, K. (1987). A muscle model for animating three-dimensional facial expression. In Stone, M. C., editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 17–24.

Waters, K. (1996). http://crl.research.compaq.com/publications/ books/ waters/Appendix1/appendix1.html.

Waters, K. and Levergood, T. M. (1993). DECface: An automatic lip-synchronization algorithm for synthetic faces. Technical Report CRL-93-4, Hewlett Packard Laboratories.

Weiss, R. (1966). Be vision, a package of ibm 7090 fortran programs to draw orthographic views of combinations of plane and quadric surfaces. *JACM*, 13:194–204.

Welsh, B. (1991). *Model-based coding of images.* PhD thesis, Essex University.

Wojdel, A., Rothkrantz, L. J. M., and Wojdel, J. C. (2003). Fuzzy-logical implementation of co-occurence rules for combining aus. In *Proceedings of the 6th IASTED International Conference on Computers, Graphics, and Imaging (CGIM 2003).*

Wooldridge, M. (1999). Intelligent agents. In Weiss, G., editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence.* Cambridge, Mass, MIT Press.

Wu, Y., Magnenat-Thalmann, N., and Thalmann, D. (1994). A plastic-visco-elastic model for wrinkles in facial animation and skin. In *Proc. 2nd Pacific Conference on Computer Graphics and Applications.*

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:358–353.

Zajonc, R. (1980). Feeling and thinking: preferences need no inferences. *American Psychologist*, 35:151–175.

Summary

The combination of research in the field of computer graphics, autonomous agents, and speech and language technology has led to the development of embodied agents. The emerging technology of embodied agents can realize different promising applications including human-like interfaces to improve the interaction between human and computer; simulated virtual characters for different application such as entertainment, education, and the like; and believable animated characters to increase the interestingness of computer games.

Embodied agents are a special kind of agent in the sense that they are represented by animated human or animal bodies, or sometimes just an animated talking head. In this thesis we consider embodied agents which are represented by an animated talking head. For such embodied agents to be believable, the minds of agents should not be restricted to model reasoning, intelligence and knowledge but also emotions and personality. Furthermore, it is necessary to pay attention not only to the agent's capacities for natural language interaction but also to its non-verbal aspects of expression.

There are several issues we deal with in this thesis. First of all, from the computer graphics direction, we deal with the problem of creating a face model and a facial muscle model in such a way that realistic facial expressions can be produced in real-time on a standard computer. We propose: (i) a face model that allows high quality and realistic facial expressions, which is sufficiently simple in order to keep the animation real-time and is able to assist the muscle model to control the deformations; (ii) a muscle model that produces realistic deformation of the facial surface, handles multiple muscle interaction correctly and produces bulges and wrinkles in real-time. In order to do so, human effort is needed for preprocessing the face model and tuning up the parameters of the muscle model. Therefore, we also deal with the issue of reducing human effort when creating facial expressions on a newly created face model based on the data from an existing model. We use Radial Basis Function (RBF) networks to deform a source face model to represent a target face model using the specification of corresponding landmarks on the two face models. The landmarks on the source face model are manually specified once and are reused for every target face model. We introduce a novel method to specify and adjust landmarks on the target face model automatically. The adjustment process is done by Genetic Algorithms (GAs). The fitness function used in the GAs expresses the difference between the surface of the deformed face model and the target face model.

Secondly, from the AI direction, we deal with the problem of combining different facial movements temporally. We concentrate on the dynamic aspects of facial movements and the combination of facial expressions in different channels that are responsible for different tasks.

Thirdly, we deal with the motivation of one kind of facial expression - emotional facial expressions. We propose an implementation of emotions and a mapping from several emotions to facial expressions. The implementation of emotions, ParleE, is a quantitative, flexible and adaptive computational implementation of emotions for an embodied agent situated in a multi-agent environment. The mapping from several emotions to facial expressions are based on descriptive work from psychologists. We focus on two aspects of generating emotional facial expressions: (i) the continuous changes in expressions of an emotion depending on the intensity by which it is felt; (ii) the combinations of expressions due to more than one emotion, i.e., blends, in accordance with the literature mentioned.

Finally, we discuss how the several subsystems we mentioned earlier can be combined to create an emotional embodied agent. As an example, we introduce Obie, a football (soccer) supporter agent. Although the subsystems that we describe in this thesis are built as components of an embodied agent system, they can be used separately in other system. Each subsystem can also be replaced by similar subsystem from other system that fulfills the same task.

Samenvatting (in Dutch)

De combinatie van onderzoek op het gebied van computergrafiek, autonome agenten, en spraaken taaltechnologie heeft geleid tot de ontwikkeling van virtuele mensen, meestal in de vorm van pratende hoofden. De nieuwe technologie is geschikt voor verschillende veelbelovende toepassingen met name om de interactie tussen mens en computer te verbeteren. Men kan hierbij denken aan gesimuleerde virtuele karakters voor verschillende toepassingen zoals vermaak, onderwijs, enz. of geloofwaardige geanimeerde karakters als personages in computerspellen.

Voor de geloofwaardigheid van virtuele karakters moeten ze niet alleen beschikken over modellen van redeneren, kennis en intelligente maar ook blijk geven van emoties en een eigen persoonlijkheid. Voorts is het noodzakelijk om niet alleen aandacht te besteden aan de capaciteiten van de agent voor natuurlijke taalinteractie maar ook aan de non-verbale aspecten.

Er zijn verscheidene kwesties die wij in deze thesis hebben behandeld. Eerst en vooral, van uit computergrafiek, behandelen wij het probleem om modellen van het gezicht en onderliggende spieren tot stand te brengen zodanig dat de realistische gezichtsuitdrukkingen in real time op een standaardcomputer kunnen worden veroorzaakt. Wij introduceren: (i) een gezichtsmodel dat hoogstaande en realistische gezichtsuitdrukkingen toestaat, wat voldoende eenvoudig is om de animatie real time te houden en dat geschikt is om door het spiermodel vervormd te worden; (ii) een spiermodel dat correct realistische vervorming van de gezichtsoppervlakte veroorzaakt, veelvoudige spierinteractie correct afhandelt en rimpels in real time veroorzaakten. Dit te doen is normaliter veel handmatige inspanning nodig voor het voorbewerken van het gezichtsmodel en om het afstemmen van de parameters van het spiermodel. Daarom behandelen wij ook de kwestie van het verminderen van menselijke inspanning bij het maken van nieuwe gezichtsmodellen. Wij gebruiken netwerken van de Radial Basis Functions (RBF) om een brongezichtsmodel om te vormen tot een model van het doelgezicht gebruikmakend van de overeenkomstige oriëntatiepunten op de twee gezichtsmodellen. De oriëntatiepunten op het brongezichtsmodel worden eenmalig manueel gespecificeerd en voor elk model van het doelgezicht opnieuw gebruikt. Wij introduceren een nieuwe methode om oriëntatiepunten op het model van het doelgezicht automatisch te specificeren en aan te passen. Het aanpassingsproces wordt gedaan door Genetische Algoritmen (GAs). De geschiktheidsfunctie die in GAs worden gebruikt drukken het verschil tussen de oppervlakte van het vervormde gezichtsmodel en het model van het doelgezicht uit.

Ten tweede, van de AI kvant, behandelen wij het probleem om verschillende gezichtsbewegingen te combineren. Wij concentreren ons op de dynamische aspecten van gezichtsbewegingen en de combinatie gezichtsuitdrukkingen in verschillende kanalen.

Ten derde, behandelen wij de motivatie van één soort gezichtsuitdrukkingen - emotionele gezichtsuitdrukkingen. Wij stellen een implementatie van emoties en een afbeelding van verscheidene emoties op gezichtsuitdrukkingen voor. De implementatie van emoties, ParleE, is een kwantitatieve, flexibele en computerimplementatie van emoties voor een agent die in een multi-agentenmilieu wordt gesitueerd. De afbeelding van verscheidene emoties op gezichtsuitdrukkingen is gebaseerd op het beschrijvende werk van psychologen. Wij concentreren ons op twee aspecten van het produceren van emotionele gezichtsuitdrukkingen: (i) de ononderbroken veranderingen in uitdrukkingen van een emotie afhankelijk van de intensiteit waardoor het wordt gevoeld; (ii) de combinaties van uitdrukkingen toe te schrijven aan meer dan één emotie.

Tot slot bespreken wij hoe de verscheidene subsystemen kunnen worden gecombineerd om een emotionele agent te creëren. Als voorbeeld, introduceren wij Obie, een voetbal supporter. Hoewel de subsystemen die wij in deze thesis beschrijven als componenten van een agentensysteem worden gebouwd, kunnen zij afzonderlijk in ander systeem worden gebruikt. Elk subsysteem kan ook door gelijkaardig subsysteem van andere systeem worden vervangen dat de zelfde taak vervult.

Curriculum Vitae

The Duy Bui was born in the 9th of May, 1978 at Ha Nam Ninh, Vietnam. Afterwards, he moved to Hanoi and grew up there. Since being in school, his favorite subjects have been mathematics and computer science. However, because he was addicted to computer games, he decided to devote his career to computer science. It was the first time when he touched the computer in 1991, when he started programming in GW BASIC. In 1995, he was awarded the top prize for the Vietnamese National Olympiad in Informatics. In 1994 and 1995, he was awarded two bronze medals in the International Olympiad in Informatics (organized in the Netherlands and Hungary respectively). In March 1998, he received the AusAid scholarship from the Australian government to do his undergraduate study at the university of Wollongong, Australia. He got his Bachelor of Computer Science in December 2000. From March 2001 to July 2004, he was AIO (trainee research assistant) and PhD student at the TKI group, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, the Netherlands. This thesis reflects his work carried out at the TKI group.

SIKS DISSERTATION SERIES

1998-1 .... Johan van den Akker (CWI), *DEGAS - An Active, Temporal Database of Autonomous Objects*

1998-2 .... Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*

1998-3 .... Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

1998-4 .... Dennis Breuker (UM), *Memory versus Search in Games*

1998-5 .... E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*

1999-1 .... Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*

1999-2 .... Rob Potharst (EUR), *Classification using decision trees and neural nets*

1999-3 .... Don Beal (UM), *The Nature of Minimax Search*

1999-4 .... Jacques Penders (UM), *The practical Art of Moving Physical Objects*

1999-5 .... Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*

1999-6 .... Niek J.E. Wijngaards (VU), *Re-design of compositional systems*

1999-7 .... David Spelt (UT), *Verification support for object database design*

1999-8 .... Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*

2000-1 .... Frank Niessink (VU), *Perspectives on Improving Software Maintenance*

2000-2 .... Koen Holtman (TUE), *Prototyping of CMS Storage Management*

2000-3 .... Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*

2000-4 .... Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*

2000-5 .... Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*

2000-6 .... Rogier van Eijk (UU), *Programming Languages for Agent Communication*

2000-7 .... Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*

2000-8 .... Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*

2000-9 .... Florian Waas (CWI), *Principles of Probabilistic Query Optimization*

2000-10 .... Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*

2000-11 .... Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*

2001-1 .... Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*

2001-2 .... Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*

2001-3 .... Maarten van Someren (UvA), *Learning as problem solving*

2001-4 .... Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

2001-5 .... Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*

2001-6 .... Martijn van Welie (VU), *Task-based User Interface Design*

2001-7 .... Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*

2001-8 .... Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*

2001-9 .... Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

2001-10 .... Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*

2001-11 .... Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002-01 .... Nico Lassing (VU), *Architecture-Level Modifiability Analysis*

2002-02 .... Roelof van Zwol (UT), *Modelling and searching web-based document collections*

2002-03 .... Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*

2002-04 .... Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*

2002-05 .... Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

2002-06 .... Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*

2002-07 .... Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

2002-08 .... Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

2002-09 .... Willem-Jan van den Heuvel(KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*

2002-10 .... Brian Sheppard (UM), *Towards Perfect Play of Scrabble*

2002-11 .... Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

2002-12 .... Albrecht Schmidt (Uva), *Processing XML in Database Systems*

2002-13 .... Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*

2002-14 .... Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

2002-15 .... Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

2002-16 .... Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*

2002-17 .... Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003-01 .... Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*

2003-02 .... Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*

2003-03 .... Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

2003-04 .... Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*

2003-05 .... Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*

2003-06 .... Boris van Schooten (UT), *Development and specification of virtual environments*

2003-07 .... Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*

2003-08 .... Yongping Ran (UM), *Repair Based Scheduling*

2003-09 .... Rens Kortmann (UM), *The resolution of visually guided behaviour*

2003-10 .... Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

2003-11 .... Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

2003-12 .... Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*

2003-13 .... Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*

2003-14 .... Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

2003-15 .... Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*

2003-16 .... Menzo Windhouwer (CWI), *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*

2003-17 .... David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

2003-18 .... Levente Kocsis (UM), *Learning Search Decisions*

2004-01 .... Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2004-02 .... Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*

2004-03 .... Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

2004-04 .... Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*

2004-05 .... Viara Popova (EUR), *Knowledge discovery and monotonicity*

2004-06 .... Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*

2004-07 .... Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

2004-08 .... Joop Verbeek(UM)), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise*

2004-09 .... Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*

2004-10 .... Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*

2004-11 .... Michel Klein (VU), *Change Management for Distributed Ontologies*

2004-12 .... The Duy Bui (UT)), *Creating emotions and facial expressions for embodied agents*

Index

Ahlberg (2001), 139
Albrecht et al. (2002a), 139
Albrecht et al. (2002b), 139
Allport and Odbert (1936), 139
Allport (1924), 139
Argyle (1990), 139
Arnold (1960), 118, 119, 139
Averill (1980), 139
Basmajian (1974), 139
Bates (1994), 139
Bishop (1995), 139
Blinn (1978), 139
Blum and Langford (1998), 139
Bolles and Fanselow (1980), 139
Booker (1987), 140
Bozinovski and Bozinovska (1999), 140
Bui (1975), 140
Cacioppo et al. (2000), 118, 140
Carmel and Markovitch (1996), 140
Cattell (1946), 140
Cohen and Massaro (1993), 140
Coons (1967), 140
Cornelius (1996), 140
Costa and McCrae (1992), 140
Coulter (1989), 140
Damasio (1994), 140
Darwin (1965), 115, 140
Davis (1991), 140
DeCarlo et al. (2002), 140
Doenges et al. (1997), 141
Dutoit et al. (1996), 141
Ekman and Friesen (1975), 115–117, 120, 124, 126, 141
Ekman and Friesen (1978), 124, 141
Ekman et al. (1983), 141
Ekman et al. (1993), 141
Ekman (1972), 141
Ekman (1982), 117, 141
Ekman (1984), 141
Ekman (1989), 141
El-Nasr et al. (2000), 141
Elliott (1992), 141
Enciso et al. (2000), 141
Essa (1994), 141
Fernández-Dols and Carroll (1997), 117, 142
Fikes and Nilsson (1971), 142
Fiske (1949), 142
Forgas and Moylan (1987), 142
Frijda (1986), 142
Galernter (1994), 142
Gergen (1985), 142
Goldberg (1989), 142
Gouraud (1971), 142
Gratch (2000), 142
Grings and Dawson (1978), 142
Hager and Ekman (1996), 142
Hayes-Roth and van Gent (1997), 121, 142
Hendrix et al. (2000), 121, 142
Hoch et al. (1994), 142
Inoue et al. (1996), 143
Itti et al. (2003), 143
Izard (1971), 117, 120, 143
Izard (1981), 143
Izard (1990), 143
Izard (1991), 120, 143
Izard (1994), 143
Izard (1997), 117, 120, 143
James (1884), 143
Kalra et al. (1992), 143
Kalra (1993), 143
Kappas (2003), 115, 118, 119, 143

King et al. (2000), 144
Koch and Bosshard (1998), 144
Koch et al. (1996), 144
Kähler et al. (2002), 143
Kähler et al. (2001), 143
Laird and Bresler (1990), 144
Lange (1922), 144
Latta et al. (2002), 144
Lazarus and Alfert (1964), 144
Lazarus et al. (1970), 144
Lazarus (1991), 144
LeDoux (1996), 144
Lee et al. (1995), 144
Lee (1990), 124, 144
Levenson (1992), 144
Leventhal and Scherer (1987), 118, 144
Lisetti (1999), 145
Loyall and Bates (1997), 145
Löfqvist (1990), 145
Magnenat-Thalmann et al. (1988), 145
Mahl (1972), 145
Mandler (1975), 145
Mani and Ostermann (2001), 145
McCrae and Costa (1987), 145
Minsky (1988), 145
Moubaraki et al. (1995), 145
Noh and Neumann (2001), 145
Norman (1963), 145
Nye and Brower (1996), 145
Oatley and Johnson-Laird (1987), 145
Oatley (1992), 145
Oka et al. (1987), 146
Ortony et al. (1988), 119, 146
Ortony (2002), 146
Paradiso and L'Abbate (2001), 121, 146
Parke and Waters (1996), 124, 146
Parke (1972), 146
Parke (1974), 146
Pasquariello and Pelachaud (2001), 146
Pelachaud et al. (1993), 146
Pelechaud et al. (1996), 146
Perlin and Goldberg (1996), 121, 146
Perlin (1997), 121, 146
Picard (1997), 146
Pighin et al. (1998), 121, 146
Platt and Badler (1981), 147
Platt (1985), 146
Plutchik (1984), 147
Price et al. (1985), 147
Reeves (1990), 147
Reilly and Bates (1992), 147
Reilly (1996), 147
Roseman et al. (1990), 147
Roseman et al. (1996), 147
Roseman (1984), 147
Rosis et al. (2003), 147
Rousseau (1996), 147
Russell and Fernández-Dols (1997), 117, 147
Russell (1997), 117–119, 147
Rydfalk (1987), 148
Scherer (1992), 118, 148
Scherer (2001), 118, 148
Sederberg and Parry (1986), 148
Sen and Weiss (1999), 148
Shaver et al. (1992), 148
Shields and Stern (1979), 148
Sigman and Capps (1997), 148
Sloman (2002), 148
Smart (2000), 148
Smith and Scott (1997), 118, 148
Spears and DeJong (1991), 148
Speisman et al. (1964), 148
Stern et al. (1998), 121, 149
Terzopoulos and Waters (1990), 149
Thomas and Johnston (1981), 149
Tomkins (1962), 117, 149
Tomkins (1963), 117, 149
Trappl and Petta (1997), 149
Tupes and Christal (1961), 149
Ulgen (1997), 149
Velásquez (1996), 149
Velásquez (1997), 149
Viaud and Yahia (1992), 149
Waite (1989), 149
Wang (1993), 149
Waters and Levergood (1993), 150
Waters (1987), 150

Waters (1996), 150
Weiss (1966), 150
Welsh (1991), 150
Wojdel et al. (2003), 150
Wu et al. (1994), 150
Zadeh (1965), 121, 150
Zajonc (1980), 150