The Affective PDF Reader

Markus Radits
2010-08-05
Subject: Computer Science
Level: Master
Course code: 5DV00E
Acknowledgements

I want to dedicate this work to my parents, Erna and Ewald. Without their support this work definitely would have never been written. I want to thank them for their generosity, for their patience and their big heart. I will never forget!

Another acknowledgement goes to my supervisor Tobias Andersson Gidlund, who was a very helpful guide when writing this paper. I want to thank him for his openness and his efforts to keep my focus on what I wanted to accomplish.

Växjö, 2010
Abstract

The Affective PDF Reader is a PDF Reader combined with affect recognition systems. The aim of the project is to research a way to provide the reader of a PDF with real-time visual feedback while reading the text to influence the reading experience in a positive way. The visual feedback is given in accordance to analyzed emotional states of the person reading the text - this is done by capturing and interpreting affective information with a facial expression recognition system. Further enhancements would also include analysis of voice in the computation as well as gaze tracking software to be able to use the point of gaze when rendering the visualizations.

The idea of the Affective PDF Reader mainly arose in admitting that the way we read text on computers, mostly with frozen and dozed off faces, is somehow an unsatisfactory state or moreover a lonesome process and a poor communication. This work is also inspired by the significant progress and efforts in recognizing emotional states from video and audio signals and the new possibilities that arise from.

The prototype system was providing visualizations of footprints in different shapes and colours which were controlled by captured facial expressions to enrich the textual content with affective information. The experience showed that visual feedback controlled by utterances of facial expressions can bring another dimension to the reading experience if the visual feedback is done in a frugal and non intrusive way and it showed that the evolvement of the users can be enhanced.

Keywords:
Affective Computing, Facial Expression, Affective Feedback, PDF Reader
# Table of content

1. Introduction .................................................................................................................. 1

2. Research Question ....................................................................................................... 2
   2.1 Background .................................................................................................................. 3
   2.2 Constraints ................................................................................................................... 4
   2.3 Outline ......................................................................................................................... 5

3. Theoretical Background on Affective Computing ......................................................... 6
   3.1 A brief History on Affective Computing ...................................................................... 6
   3.2 Emotions .................................................................................................................... 7
     3.2.1 The Functions of Emotions .................................................................................. 8
     3.2.2 Affective Information and measuring Emotions .................................................. 10
     3.2.3 Affective Models ............................................................................................... 12
   3.3 The Affective Loop .................................................................................................... 13
   3.4 Problems in Interpreting Emotions .......................................................................... 15

4. Reference Projects ....................................................................................................... 17
   4.1 eMoto ......................................................................................................................... 17
   4.2 EmoteMail ................................................................................................................... 19
   4.3 FAIM .......................................................................................................................... 20

5. The Affective PDF Reader - Prototype ......................................................................... 22
   5.1 Requirements Analysis ............................................................................................... 22
   5.2 Workflow Diagram ..................................................................................................... 22
   5.3 Facial Expression Recognition .................................................................................... 23
     5.3.1 The Basic Algorithm ............................................................................................ 24
     5.3.2 The Training Process ........................................................................................... 25
     5.3.3 Principle Component Analysis ............................................................................. 25
     5.3.4 Computing the Distance to the neutral Image ..................................................... 26
     5.3.5 Adjustments to Facial Expression Recognition .................................................... 26
     5.3.6 Face Detection ..................................................................................................... 27
     5.3.7 Cropping Face Region .......................................................................................... 27
     5.3.8 Filtering and Light Adjustments .......................................................................... 28
     5.3.9 Applying a Mask .................................................................................................... 29
     5.3.10 The resulting Matlab Application ...................................................................... 30
     5.3.11 User Interface of the Matlab Application ............................................................. 31
   5.4 PDF Reader Application ............................................................................................. 32
     5.4.1 Technologies ......................................................................................................... 32
     5.4.2 Graphical User Interface ....................................................................................... 32
     5.4.3 PDF Reader .......................................................................................................... 33
     5.4.4 UDP Client ............................................................................................................ 33
     5.4.5 Affective Model ..................................................................................................... 33
     5.4.6 The Colour Mapping ............................................................................................. 34
     5.4.7 Visualization Component ...................................................................................... 34
     5.4.8 Footprint Visualization ......................................................................................... 34
     5.4.9 GUI functions ......................................................................................................... 37
     5.4.10 Blurred PDF Text .................................................................................................. 37
   5.5 Performance Issue ....................................................................................................... 38

6. Conclusion ..................................................................................................................... 39

7. Future Work .................................................................................................................. 40
   7.1 Affective Model .......................................................................................................... 40
   7.2 Facial Expression Recognition .................................................................................... 40
   7.3 Gaze Tracking ............................................................................................................ 40
   7.4 Speech Recognition .................................................................................................... 40
7.5 Visualizations ................................................................. 41
7.6 Evaluation ........................................................................ 41
7.7 Redesigning the System .................................................. 41
8. References ........................................................................ 42
9. Figure Index ....................................................................... 46
10. List of Abbreviations ......................................................... 47
11. Tables ............................................................................. 48
1. Introduction

It was in a seminar work in 2009 when I first heard about the scientific area of Affective Computing. It was the first time I read about the need to make computers acting more emotional – that was in a paper written by Rosalind W. Picard in 1995 [1]. In general Affective Computing is considered to include emotion in computation and in human computer interaction (HCI). One of the main arguments for Affective Computing is to enhance the communication between us and our computers or rather to shift it to a new level by using affect recognition tools and systems that are capable of modelling affective interaction.

The imaginable range of applications that could be designed for a better affective interaction is endless. We need to understand that it is important to communicate affective information in HCI. A constantly ignored affective action is considered to be disrespectful in human communication. So if you think about how much time we spend working on computers, why should we not try to enable our software or our machines to respond more emotionally. This would not only lead to applications that fits our needs but to systems that are felt to be enjoyable [2].

As the title of this thesis already implies it is concerned with the reading experience of a digital document. As we know there is a difference in reading text of a digital document compared to reading text of a book. Although we have E-Books and E-Readers we still feel a difference to the old fashioned haptic experience of a real book [3]. But there are undeniable functional advantages when dealing with digital text. Latest advancements in Affective Computing and Image Processing capacitate us with the ability of doing research to improve the reading experience of digital documents by conveying an approach focused on affective interaction.

This thesis project aims in researching a way to enhance the reading experience while reading a PDF document by providing visual feedback in the background of the text. It uses a facial expression recognition system to gather and to interpret data from a webcam to draw conclusions to the emotional state of the reader. The facial expression recognition tool is programmed in Matlab and is able to distinguish between 5 basic emotional categories, like neutral, happy, sad, anger and disgust. Interpreted facial expressions are sent to a Java program – the Affective PDF Reader over UDP for further processing and for visualizing the affective feedback in form of graphical footprints in the background of a PDF. The position of the footprints is either determined by the mouse pointer or randomly. An animator is used to continuously fade out footprints to avoid overloading and perturbing effects. The aim was to design the feedback in a way that is stimulating to the reader and that is perceived to influence the reading experience in a positive way.

The design of the application, the appertaining theoretical aspects of Affective Computing and the results are discussed in this thesis.
2. Research Question

To get a closer idea of the problem it is a good starting point to ask what reading is. “Reading is a highly synesthetic process. There are several stages in the reading process. When we read a word, a visual pattern is perceived which is then decoded with reference to an internal representation of the language. This is analysed syntactically and semantically in the context of a phrase or sentence. In other words: Semiotic signs are read and get a meaning to be transferred into sounds or other media” [4]. Aside getting the informational content of a text, reading is all about synesthesia, which “is the phenomenon in which the stimulation of one sense modality gives rise to a sensation in another sense modality” [4].

The process of reading is a highly complex task and the visual factors in reading play a crucial role in the perception of a text as studies in neuroscience and psychophysics showed. When we read, our eyes are looking at different letters of the same word and combine the perceived images in our brains in a process called fusion. Another interesting aspect is that specific eye movements seem to correlate to the difficulty of the text we are reading. With sophisticated eye tracking equipment it was shown that our eyes make jerky movements when we are focusing on a word, we were not able to read at the first time [5] [6] [7].

It is for sure that physical utterances during reading, for example eye movements, relate to the mental activity of reading. Affective Computing plays a key role in bringing the latest findings in psychophysics and neuroscience to computing to introduce a new dimension in HCI. In ”New Essays on New Media” [8] it is argued that in the coming decades not only the design of digital content will be important but also the design of the delivery systems. We need to stay curious and creative with all the possibilities we are equipped with by latest advancements of algorithms in image processing.

Facial expression recognition is now a very popular technique to estimate the emotional state of a human. There are different approaches to estimate a facial expression but we still cannot expect these algorithms to provide us with recognition rates that are better than recognition rates of humans. We should not forget that even humans are not able to recognize a facial expression for sure. Some algorithms are doing very well now with recognition rates that are almost equal to recognition rates done by humans as shown in [9].

By capturing and interpreting a facial expression we get affective data we can use to feed our systems to establish new forms of affective loops [10] – in our case while reading a PDF document. This all cuts down to the question:

*Can we influence the reading experience of a PDF in a positive way by building an application that is capable of providing visual feedback that is controlled by affective utterances, to get the users more involved in the reading process?*
2.1 Background

The fact that big parts of society in developed countries are using computers for several hours each day, should lead to the question what happens with us and our communication. It is important to point out the role of research that is done to understand and enhance our communication with computers. When we work with computers it massively influences our behaviours, our ways to think, our ways to talk and hence it has a deep influence on our daily lives. This should be enough reason to put more effort into understanding the way we communicate with computers and to try to reveal the way they change us.

Our behaviour and our communication are driven a lot by underlying emotional states. “We react to an emotional stimulus more quickly than we can even cognitively register the content of the object by means of perceptual processing.”[11]. Understanding emotions is a major key to understand complexity of communication.

Every situation we have passed in our lives leads to specific feelings. No matter if they put us into pleasure, into fear, into anger, compassion, disgust, other categories of feeling, or combinations of it. No matter if we are watching a concert, talking to friends, go shopping, do whatever we do puts us into a different kind of mood. And of course sitting in front of a computer and do whatever we may do, does it too.

Not only in computer science but also in other sciences emotions did not play a big role in research, because they cannot be easily classified and hence were considered to be unscientifically [12]. A very long time the main focus in computer science was more on functional aspects of applications rather than understanding the interface between machines and humans. I do not want to be understood wrong in that case, because the functional aspects in computer science are very important and we heavily rely on advances in technology and their functional aspects. But it is also good to see that social aspects in the context of computer science are now having attention they have never had before.

As the famous Austrian-American psychologist and philosopher Paul Watzlawick stated, it is not possible to not communicate [13]. And that’s completely true. Our communication is not only driven by what we say, but it has a lot to do with our behaviours. I think there’s no need to elaborate on that, because it is commonly agreed.

So working with a computer is always a certain kind of communication but can often be described as poor communication. We are sitting in front of our computers with frozen and dozed off faces most of the time. There is no need to react emotionally because our computers would not respond to it. This turns into a big problem when it comes to human communication. Main focus in information systems is on processing and algorithms. The context in which information has been created and the relation to human souls seems to be completely left out here. I guess that’s why a lot of people have somehow unlearned to transport their emotions in real life face to face communication. That sounds dramatic but I think it is really important to point out that we should design information systems in a way we feel comfortable with and not to force a techno deterministic approach where we humans have to fit to the design of information systems – this would make things not better.

The media can also be considered to be the message as Marshall McLuhan stated [14]. In our context this means that the design of an information system and its capabilities inherently defines the way we perceive information, no matter of what type this information may be. This is where Affective Computing comes in to enrich our communication and our experience in a digital world. Making computers acting more emotional has a great potential in various use cases and applications. A lot of research has been done so far to help on interpreting human behaviour to draw conclusions to emotional states. In general this is a complex task. That’s why Affective Computing is
considered to be an interdisciplinary area that heavily relies on advances in many sciences like psychology, neurology, artificial intelligence, virtual reality, cognitive science and of course advances in computer algorithms.

2.2 Constraints

There were some constraints when doing this work. They mainly arose during proceedings in the practical work where it was necessary to make compromises considering the usage of the facial expression recognition, in displaying the PDF document, the rendering of the background visualizations and computation speed in general. The constraints are summarized as followed:

- **Accuracy of facial expression recognition**

  At the point of writing this document it was not possible to find an open source solution for a facial expression recognition component that was working in real-time and reliably. The compromise was to alter and to tune a facial expression recognition tool in Matlab that originally did not work in real-time. The drawback of this approach is that the facial expression recognition output is not that accurate as wished. But it is sufficient to demonstrate the basic aim of this work. The adjustments made in setting up the facial expression recognition tool are explained later on in this document.

- **Java Client**

  A basic aim of this project is to implement a prototype to demonstrate the principles of the research that has been done. It was the fastest and easiest way to choose Java as platform to program the Affective PDF Reader client, because there are handsome PDF libraries as well as handsome technologies to create visualizations. Of course one drawback when using Java compared to other platforms is computation speed.

- **UDP Streaming**

  Matlab was used as platform for performing the facial expression recognition because it has significant advantages in computation speed and it also has a lot of built in functions to be used in scientific context. So, to get continuous results of the facial expression recognition into the Java client, a UDP communication was set up, with a very simple protocol sending appropriate information from the Matlab face recognition into the Java client of the Affective PDF Reader.
2.3 Outline

This work cuts into the following main parts:

1. Introduction

   Gives an introduction to the area of affective computing and the general aim of this work.

2. Research Question

   Cuts down the research problem to one single question and gives an overview of the research problem.

3. Theoretical Background on Affective Computing

   Gives a short history on Affective Computing and discusses the most important aspects of this scientific area.

4. Reference Projects

   Lists and describes several reference projects that were inspiring for doing this work.

5. The Affective PDF Reader - Prototype

   This section is concerned with the implementation of the prototype system and some accompanying theoretical aspects of the facial recognition component.

6. Future Work

   Discusses ideas to enhance the prototype system and to expand it with affect recognition systems other than facial expression recognition.

7. Conclusion.

   Summarizes the results of the work and draws conclusions.
3. Theoretical Background on Affective Computing

People tend to respond to media in ways that are natural and social, mirroring ordinary human behaviour in social situations [15]. This is a very important note which can be interpreted in a way that our behaviours tend to establish a social connection even when working with machines. To get a better insight on Affective Computing and to understand why it is so important to us I want to give a little information on the history of Affective Computing in the next section.

3.1 A brief History on Affective Computing

There are now an immense number of projects and researchers worldwide dealing with the difficulties to include emotion in human computer interaction, although the scientific area of Affective Computing can be considered to be quite young compared to other sciences. The emergence of this scientific area is commonly dated with Rosalind W. Picard’s paper in 1995[1]. Of course it is not possible to give an overview of all projects on Affective Computing that have been done in the past, but I want to give reference to some papers and projects that I found interesting and that helped me in finding a way through the jungle.

A lot of research has been done and is still going on in the Affective Computing Group at the MIT media laboratory [16]. Till 1997 Picard and here research assistants had their main focus on image and video analysis and not on Affective Computing [17]. Affective Computing as an independent research area arose later on. A lot of first experimental efforts to analyze emotional states of humans were to gather and analyze data from physical measurement. For example from galvanic skin response (GSR) [18] which is a method for measuring the electrical resistance of the skin.

In 1998 a very interesting project was concerned with the measurement of physical data of a conductor during a concert. It was called the “The Conductor's Jacket” [19] where they fabricated a jacket including several sensors to measure muscle activity, GSR, respiration, heart rate and some other data to get an emotional insight of the work of a conductor. Gathered signals were also used as input for manipulating audio signals or audio effects in a live concert situation.

In 1999, Dana Kirsch wrote her master thesis at MIT media laboratory [20] which was concerned with the construction of an emotionally reactive toy. This toy was designed to interpret the way the user played with it and to give affective feedback by either mimicking the user’s mood or by reacting to moods accordingly. The toy was able to perform some basic mimics like opening or closing its mouth. Depending on how the toy was moved and handled conclusions to basic emotional states were drawn (happy, unhappy, very unhappy, very happy and neutral).

In the same year another project aimed in measuring emotional stress when driving a car [21]. This system was able to record the driver’s skin conductance, the respiration, muscle activity and heart activity. A camera was used to capture the driver’s facial expression once per minute. The aim was to develop algorithms used to distinguish between mental stress, physical stress and a relaxed state. Uses cases for such an application would be to determine if the driver should be interrupted by an incoming telephone call or the navigation system, according to analyzed stress.

An important carriage of emotion is our voice. One of the first attempts to analyze audio signals for existence of affective information was in 1999 [22]. This project tried to classify driver’s speech under stress situations with a so called Teager energy operator [23]. This operator was capable to extract the signal energy based on mechanical and physical considerations. The stress was simulated by driving speed and
by a mental task the driver had to solve. One hypothesis was that driving a car while trying to solve a problem produces stress that might reflect in the speech of the driver. They reported significant recognition rates.

Useful appendages in detecting and sensing frustration were discussed in 2001 [24]. Several sensors were proposed to analyze the frustration of a user by consideration of mouse movements, skin conductivity, yelling and by providing haptic interfaces where users were able to express their frustration. One of the main conclusions was that participants liked having devices to communicate frustration and to see a change in the provided feedback after detecting frustration.

One of the first fully automated facial action recognition systems was presented in 2003 [25]. The System was able to analyze fundamental blocks of facial expressions enumerated in Paul Ekman’s Facial Action Coding System (FACS) [26]. The Facial Action Coding System (FACS) is used to categorize facial behaviours based on the muscles that produce them [27]. By using so called Action Units (AU) it is possible to manually code nearly any anatomically possible facial expression.

A face was decomposed into 32 Action Units and their movements in time were captured. They reported a recognition rate for different action units of 69%.

An impressive project was presented in 2009 at Machine Perception Laboratory, University of California [28]. A lot of media were paying attention. A robotic head, better known as the “Einstein Robot” was taught to self guide in learning realistic facial expressions. Under the robotic head’s skin there were working 31 servo motors to map different kind of emotional facial expressions. The robot was able to learn facial expressions coded as FACS within one hour by moving his servo motors in random order. Every time the software was recognizing a similarity to a given facial expression the actual state of the servo motors was mapped to a facial expression. And there are a lot of more interesting projects, but it would be beyond the scope of this work to mention them all. Ten years after the initial paper of Picard was presented, in October 2005 the First International Conference on Affective Computing & Intelligent Interaction took place in Beijing [29] and the results were published subsequently [30].

The Second International Conference on Affective Computing & Intelligent Interaction was held 2007 in Lisbon and again a collection of topic related papers and projects was released subsequently [31].

Affective Computing grew more and more popular finding its way not only in academic research but also in economy – for example in the gaming industry. Also a lot of companies exist now trying to focus on different kind of affective technologies.

3.2 Emotions

As this master thesis is concerned with emotional aspects of computing, the basic functions of emotions should be explained. Emotions play a major role in decision making. This is an unconscious process most of the time. Emotions guide us in situations where it would be impossible to traverse all different kind of paths in an endless decision tree. Emotions make us deciding within seconds. Of course this can lead to making a wrong decision in a certain context. The ability to act on instinct seems to raise uncertainty in our decisions. “Why should computers have emotions?”, Rosalind W. Picard asks provoking [1] by pointing out that emotions make our decisions more uncertainty and that it seems not useful to make computers more affective when keeping in mind which sensible areas of our lives they are controlling (like cars, nuclear power plants, and so on). It could make you think that an affective computer would be dangerous. Of course this is an exaggeration.
But we need to understand the role of emotions before pleading for affective machines. There are a lot of models and definitions out there trying to describe emotions. This is not an easy task. The role of emotions is by far not researched fully. So for us as computer scientists it is important to stay curious and to pick out useful information from latest research results in psychology and other sciences. At this point I completely agree with Petra Sundström – she says: “Theories of emotions are not easily applicable to computer science. They should be treated as an inspiration and not as directly implementable models.” [32]

It would not make a lot of sense to delve into Affective Computing without understand the very basic functional aspects of emotions and the way they guide us. Emotions make us communicate on a very subconscious level. They do good work for us when we are more concerned with conscious aspects of a daily life. That’s why we do not have direct control of our emotions. If so, it would be possible to reprogram our very basic behaviour within seconds. Most of the time, this is not desirable, because our subconscious brain is very well equipped to guide us in surviving in this world. Of course there are situations where we would be glad to reprogram ourselves just by pressing a button. For example when we want to get rid of some annoying behaviour. There are though a lot of ways how to manipulate emotions. And several occupational categories are quite good in it – like actors or politicians for example [1].

Our basic emotional system may be controlled by a reward – punishment system [33]. Our genes may define goals rather than prescribing particular behavioural patterns. By this way it is ensured that we have a lot of freedom in achieving our goals. To achieve a goal our brain is performing a huge amount of “if” statements – like: “If I do this the response would be like that”. And so on. By that way we are able to find an appropriate way for our strategies.

A general distinction of positive and negative emotions can be described as followed: “Positive emotions generally serve to bring the organism in contact with potentially beneficial resources - food, water, territory, mating or other social opportunities. Negative emotions serve to protect the organism from danger - mainly to ensure fight-or-flight responses, or other appropriate defensive strategies such as submissive behaviour or withdrawal, protection of territory or kin, and avoidance of pain.” [33]

When we try to make a first classification into positive and negative feelings we should not forget that emotional categories are in general not easy to find. Emotions in general are rich and complex. And sometimes emotional interference is that heavy that we do not even know what is going on with ourselves.

3.2.1 The Functions of Emotions

In order to develop sustainable affective applications it is important to know about the functions of emotions. The following list of functional aspects of emotions is taken from the book ”Who Needs Emotions?: The Brain Meets the Robot” by Jean-Marc Fellous [33]. The explanations to every aspect are summarized.

1. **Elicitation of Autonomic Responses.**
   
   Means functions like the change in heart rate or the release of adrenalin.

2. **Flexibility of behavioural responses to reinforcing stimuli.**
   
   Behaviour is specified by reward and punishment. We have the choice on acting in different ways to obtain a reward or a punishment in relation to associated
costs. The argument is that goals are not specified by actions but by genes which have then a powerful way to influence behaviour.


   Emotions are motivating in a way to avoid noxious stimuli. The motivation is triggered by the genes with their specific goals.

4. *Communication*

   We are communicating emotional states by behaviours like facial expression, or body movements. Monkeys for example are making open-mouth threats to signalize that they are willing to compete for some recourse.

5. *Social Bonding*

   This includes emotions that are responsible for attaching parents to their children and parents to each other, because it is more likely that children will survive if both parents are caring.

6. *The current mood state can affect the cognitive evaluation of events or memories.*

   Backprojections from parts of the brain involved in emotions can influence our emotional states.

7. *Emotions may facilitate the storage of memories*

   Emotions facilitate the storage of memory of particular episodes (episodic memories). This also includes a mechanism where the current emotional state affects recalled memories. And it is likely that emotions guide the brain in building different representations of the world.

8. *Establish a persistent and continuing motivation and direction of behaviour.*

   This is useful for achieving a particular goal.

9. *Trigger of recall of memories*

   The recall of memories can be influenced by different moods.

This model of the function of emotions can help us to develop sustainable affective applications. Furthermore it has been shown that positive feelings enhance our capabilities in creative problem solving and that it enables a broader form of thinking whereas negative emotions in general seem to foster a more spontaneous reasoning [34].

Arguing for affective applications means to distinguish between what “is right” and what “feels right” and I guess this is one main aspect where computer scientists could have problems when designing for affective applications because computer science traditionally deals with functional problems and hence it is more concerned on what is
considered “to be right” and not what “feels right”. But especially in HCI what “is right” never automatically leads to a conclusion that something may “feel right” too.

Hence designing for affective applications means to design for uncertainty, for stolidity, for desire, for social alternatives, means to design for certain degrees of freedom that is not functionally required – it means to design for human beings!

3.2.2 Affective Information and measuring Emotions

What is considered to be affective information? As we want to build affective applications and information systems we need to decide which information is serving as input for affective systems. But does input from sensors already contain any affective information? In general it is a complex task to draw conclusions to affective states from captured data. No matter how accurate these data may be. Even for humans it is sometimes difficult to interpret the mood of another person, although our sensorium seems to be well equipped for that kind of task. So what are the possibilities to explore and to measure emotions?

We benefit a lot from the fact that emotional incidents reflect in physical utterances. Picard calls the process from emotional states expressing in physical behaviour, sentic modulation [35]. So facial expression, modulation in voice, gestures and postures have their roots in emotional states – but not all physical utterance is sentic modulation. Picard also mentions a quite new form of measuring the outcome of a sentic modulation – this is done by measuring changes in bioelectric and biochemical states by implants.

We are not able to measure emotions directly by capturing sentic modulation but with this knowledge we are able to build more accurate models to draw conclusions from captured physical utterances. A very interesting anecdote on sentic modulation can be found in [1]: “The neurology literature also indicates that emotions travel their own special path to the motor system. If the neurologist asks a patient who is paralyzed on one side to smile, then only one side of the patient's mouth raises. But when the neurologist cracks a funny joke, then a natural two-sided smile appears [36].”

In “Evaluating affective interactions: Alternatives to asking what users feel” [34] it is proposed to classify the evaluation of affect into body measures and task measures. I have added the interactional approach [12] to our possibilities of the evaluation of affect:

- **Body Measures of Affect**

  Body Measures of affect means to draw conclusions to affective states by capturing data with sensors. The sensors can have different forms: This includes web cams, microphones, EEG [37], GSR on so on. There a numerous possibilities to capture data from physical utterances. A partial set of body measures taken from [34] is shown in Table 3.1.
<table>
<thead>
<tr>
<th>Modality</th>
<th>Sensor</th>
<th>Is it socially communicated?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facial Activity</td>
<td>Video</td>
<td>Yes</td>
<td>Facial expressions can differ significantly from genuinely felt feelings</td>
</tr>
<tr>
<td>IR Video</td>
<td>Highlights pupils &amp; specularities</td>
<td></td>
<td>Usually works better than ordinary video when head moves (better eye detection)</td>
</tr>
<tr>
<td>Thermal Video</td>
<td></td>
<td>No</td>
<td>Being explored to detect stress and other changes related to deception and frustration</td>
</tr>
<tr>
<td>Posture Activity</td>
<td>Force sensitive resistors</td>
<td>Yes, but not as Pressure</td>
<td>Good results discriminating level of interest in students in computer learning interactions</td>
</tr>
<tr>
<td>Hand Tension &amp; Activity</td>
<td>Force sensitive resistors or Sentograph</td>
<td>Varies; depends on gesture</td>
<td>Can be sensed from handling of mouse, steering wheel, etc., and pressure has been shown to be higher during a frustrating task</td>
</tr>
<tr>
<td>Gestural Activity</td>
<td>Electromyogram electrodes</td>
<td>Visibility varies</td>
<td>Shown for expression sensing in conducting music; other gestures largely unexplored w.r.t. expression</td>
</tr>
<tr>
<td>Vocal Expression</td>
<td>Microphone</td>
<td>Yes</td>
<td>Most methods are great for discriminating arousal but not for valence; limited to times when user is speaking</td>
</tr>
<tr>
<td>Language and choice of words</td>
<td>Text analysis tools</td>
<td>Yes</td>
<td>Can be used with interfaces requiring textual input; promising for valence; trivial to sense scripted dialogue moves</td>
</tr>
<tr>
<td>Electrodermal Activity (a.k.a Galvanic Skin Response)</td>
<td>Electrodes (can also be clothing snaps, metallic fabric, etc.)</td>
<td>No; except perhaps sweaty palm</td>
<td>Good at detecting changes in arousal but doesn’t distinguish positive/negative, and can also be triggered by non-affective changes</td>
</tr>
</tbody>
</table>

Table 3.1: Body Measures.

- **Task Measures of Affect**

Affective states do influence our behaviour in a certain way. This comprises that we would do a sequence of tasks in a certain way that is influenced by our feelings. In “Situated Cognition and the Wisdom of Feelings: Cognitive Tuning” [38], Norbert Schwarz a professor of social psychology at University of Michigan brings this approach to the point:

“If happy moods increase, and sad moods decrease, our tendency to rely on the ‘usual routines’, we may expect that individuals in a happy mood are more
likely to rely on an applicable script than individuals in a sad mood. Empirically, this is the case.”
So there is evidence that we tend to solve a problem in a certain way according to our emotional states.

- Interactive and Interpretative Approach

The interactive and the interpretative approach to explore or to communicate emotions is claimed by Boehner, De Paula, Dourish and Sengers [12]. They argue that our feelings are not pre-existing facts but developed from interaction and conversation where emotions are constructed and experienced by individuals through their culture. This means that emotions cannot considered being another kind of data that can be transferred from one place to another. And it is important to note that we cannot capture emotions directly from sensors. Emotions as a social product need to be experienced by humans to be understood. This approach points out that our feelings are more than just a sequence of biological facts. We, as computer scientists should take this theory seriously although it doesn’t help us directly in modelling affective applications. But what we can learn is to respect the fact that we cannot transfer emotions in a way we pass information within information systems.

3.2.3 Affective Models

The difficulty is to find a mathematical model that is capable of describing the emotional state of humans in a way that is holistic and accurate and provides mechanisms for a flexible changeover from one state to another as well as mechanisms to model overlapping emotional states. When designing affective applications we need to impose an affective model - without an affective model it would not be possible to relate captured data to emotions.

Currently affective models do need enormous simplifications to cope with the complexity of emotions. An affective model could have the form of a Hidden Markov Model [39] like applied in a playmate robot for children [40]. This model was able to achieve the transfer of several affective states under some basic restrictions and enabled the robot to behave and to communicate like a human child. Figure 3.1 shows the concept of transferring affective states by a Hidden Markov Model that was applied with the playmate robot. Curves A and B depict the transfer to a stimulating state by the influence of some outside stimulation. Under certain conditions an outside stimulation causes an affective state to stay within a range of equilibrium, as curve F states. After an outside stimulation, a certain affective state transfers to another affective state according to the stimulating state, as depicted by curve C. Curve E describes the transfer from an affective state to a quiet state when an outside stimulation is lacking for a certain period of time [40].
When designing an affective model it is helpful to have knowledge about some basic affective models in psychology. Another example of an affective model would be the PAD model (Pleasure-Arousal-Dominance) that was used to describe an approach for developing multimodal affective interfaces [41]. “The PAD model measures emotional tendencies and response along three dimensions: pleasure-displeasure, corresponding to cognitive evaluative judgements; arousal-nonarousal to levels of alertness and physical activity; and dominance-submissiveness to the feeling of control and influence over others and surroundings.” [41]. The PAD model can be visualized as affective model space where a vector represents the different valances of emotional characteristics as shown in Figure 3.2.

3.3 The Affective Loop

The Affective Loop is a design technique used in HCI that tries to research the user's emotional experience when working with an application. The aim is to find appropriate affective feedback responding to a user's input. This also includes feedback that is
designed to influence the affective state of the user but is not considered to be affective [42]. The Affective Loop can be described as giving the user the chance to express his feelings. These affective utterances in turn need to be captured in some way. There are endless imaginable possibilities for capturing emotional states of a person. This of course depends on technical possibilities and accuracy of affective models.

Aside of directly capturing facial expression or voice this can be accomplished by providing the user with different choices in interaction that map to different kind of affective states. For example by choosing a colour or a landscape. According to the users reaction we are able to return different kinds of affective feedback like showing a colour, an avatar or playing some sound. This in turn affects the user’s emotional experience and provokes another reaction. This is what produces the Affective Loop. It is like playing a game by making several assumptions and by offering a feedback system with certain degrees of freedom.

One can argue that human computer interaction was always concerned with user’s feedback in general and with establishing feedback systems that make sense for the users and that keep its attraction. But the common principles in HCI have their focus more on the functional aspects of interaction rather than researching the emotional experience in it. It was more about understanding the goals of the users, to understand the tasks he is working on and to search for the best interactional solution on that. Well, to design for reliable interaction it is mandatory to foresee the user’s reaction and to respond accordingly to support the user skills in a good manner. But the novice with the Affective Loop is to explicitly design for emotional experience in HCI. To explore the affective contents of an interaction implies to design for better acceptance and for a better experience in the communication between humans and computers. The creative potential in this approach opens doors for completely new interfaces with less technique deterministic side effects on human behaviour.

The Design of an Affective Loop can follow these principles taken from [32]:

1. “The user first expresses her emotions through some physical interaction involving the body, for example, through gestures or manipulations of an artifact”

2. “The system (or another user through the system) then responds through generating affective expression, using for example, colours, animations, and haptics”

3. “This in turn affects the user (both mind and body) making the user respond and step-by-step feel more and more involved with the system”

This is just a very basic guide of designing applications for an Affective Loop. It requires a lot of creativity and experiments to design for affective vibrancy in a specific case. It is about making assumptions how the user will feel after being provided with a certain kind of feedback and how the feedback should change in time to make the communication to be felt right. This leads to a system where the user feels the certain kind of freedom the application is giving to express his or herself.

A very important principle in the design of affective feedback is to preserve ambiguity to make interpretations by the user endless. It is important that we feel free in making decisions and that our opinion originates from ourselves. We do not want our
computers to act as guardians for our opinion making. Good affective feedback is a fine line between respecting and understanding our feelings and the important possibility to surprise. Without unforeseeable results an Affective Loop would become a boring endless loop. Fortunately we are able to surprise. Most of the time our computers are not. But this has a drowsy effect on our communication with computers.

So why should our computers not be able to surprise us in a way that is beneficial in HCI? Normally when your computer surprises you it means nothing good. The expected way our computers surprise us is when they are not working the way they should. But emotions are also nourished by human to human conversation that is able to surprise. With the ability to surprise the communication is shifting to the next level. This does not mean that a computer’s feedback should vary arbitrarily but it should include the possibility to make a user astonishing. What if your computer would be able to make a joke like “Sorry, there was an error – all your personal data is lost”? After some seconds where your face turns pale and you start to think about good antivirus software it would relief you with: “It was just a joke – take it easy”. The computer would also be able to interpret your facial expression and hence know if you laughed or if you were shocked and would be able to learn your kind of humour. Our curiosity in live is also driven by an implicit wish to be surprised, while searching for whatever we may do. So why should we not think about endowing our computers with this ability?

This motivation leads to a scenario-based development of affective models. An affective model could be a system of rules that directs emotional feedback with all its dependencies in an Affective Loop. The sophistication of such a model could vary from a very basic one to an artificial neuronal network with the ability to learn.

3.4 Problems in Interpreting Emotions

Some people argue that measuring emotions by taking discrete values in time has not at all to do with the emotional insight of a human and that the subject of interpreting emotions is too complex to be done by computers. This seems to be true if we consider the richness of our emotional life. Analysis of emotions is normally carried out through verbal communication which is not a trivial task.

In “How Emotion is made and measured” [12] the authors argue for a dynamic, culturally mediated, and socially constructed model of Affective Computing. This means that the purpose of Affective Computing should not be to directly use discrete emotional values in information systems but to “understand, interpret, and experience emotions in its full complexity and ambiguity.” It is not possible to measure a personal or cultural emotional component that has evolved over time. The main argument is that emotions are constructed by social and cultural conditions and that it needs an interactional approach to explore them. This is a legitimate argument and it is important to notice that we are far away from modelling realistic humanic emotional behaviour.

When we capture data in order for affective analysis, we only have captured physical utterances. It is a gamble to draw conclusions from these utterances to an emotional insight of a human. One way to deal with this problem is to design for an Affective Loop which is discussed in 3.3 of this document. But efforts in comprising and ”measuring” emotions should be perpetuated. Although we know such a little of the complex construction of emotional conditions I am sure that future affective models will become better and that they will be able to understand our emotional insights in a new dimension.

When seeing affect as social constructed it seems nearly impossible to establish a standardized model of affective behaviour. But when our emotional reactions to a certain impulse are that different how can we design for an affective model that is
capable of comprising these differences? There is no clear answer on that question but we can try to satisfy the complexity of our emotional world by providing a certain degree of freedom within affective applications. This ensures that there is not just one standard way for users to react but to react in a more creative and natural way. By designing the feedback – response cycle in affective applications as free as possible, it will be able to form a more fertile relationship between us and our computers.
4. Reference Projects

This chapter gives reference to some works that were inspiring for this thesis. They have in common that they deal with affective information while reading text (Email, SMS communication and real – time messaging systems).

4.1 eMoto

The eMoto project [43] was a project at the Swedish Institute of Computer Science. The project website can be found at [44]. They were building a messaging service that was capable of sending and receiving affective text messages. This was done by including a certain type of background into a text message. The shapes, colours and animations were built of so called affective gestures. And these gestures were recorded by an accelerometer and a pressures sensor. The basic gestures they were using are shown in Figure 4.1.

![Figure 4.1: Basic gestures of the eMoto project[43].](image)

To map affective gestures to a graphical representation, a dimensional model of emotions (Russell 1980) was used [43]. The resulting colour map where users could express themselves is shown in Figure 4.2.
Resulting affective text messages on a mobile phone are shown in Figure 4.3, Figure 4.4 and Figure 4.5.
By simply adding another dimension (a background image) to SMS communication it is possible to influence personal communication in a way that helps us to interpret the meaning of a message. That was inspiring for the Affective PDF Reader.

4.2 EmoteMail

Another interesting project trying to enhance text messages with affective or contextual information is called EmoteMail [45]. EmoteMail is an email client that captures some affective information when writing an email. It captures the facial expression of the writer and the speed of writing. This contextual information should help the reader to “decode the tone of the mail” [45].

The captured facial expressions are shown as static images beside every paragraph. By that they want to display the fluctuation of emotions when writing the text. Figure 4.6 shows a screenshot of the EmoteMail prototype.
EmoteMail is another approach of including contextual and affective information into a text message.

4.3 FAIM

The FAIM (Facial Affect in Instant Messaging) project [47] is about communication with instant messaging systems. They wanted to enrich communication of real – time messaging systems by introducing an animated character. A so called facial affect analyzer was responsible for detecting facial feature points of the writer. Figure 4.7 shows a set of feature points that are used to determine the affective state of the user’s facial expression. An animator was responsible for mapping detected facial expressions onto the virtual character that was displayed at the receiver. Figure 4.8 is a schematic diagram of the FAIM workflow. The facial affect analyzer interprets the facial expression of a user gathering data from a video input source and sends the result to the affective state manager. The affective state manager recognizes a change in affect and informs the animator module, which is responsible for mapping the detected facial expression onto the virtual character [47].

![Figure 4.7: Screenshot of the facial affect analyzer in FAIM [47].](image-url)
Figure 4.8: Workflow of the FAIM system [47].
5. The Affective PDF Reader - Prototype

By implementing a prototype of the Affective PDF Reader, the basic concept and the idea of this work should be demonstrated. The user sits in front of the computer and reads the text of a PDF. According to the facial expressions the user makes during reading the PDF, visualizations are rendered in the background of the text. This chapter is concerned with all aspects of the practical work within this thesis project, with a focus on explaining the functionality of the facial expression recognition tool and some accompanying mathematical background.

5.1 Requirements Analysis

This section describes the essential basic software functionalities and components needed to make up the prototype of the Affective PDF Reader.

- A webcam is required which supports a resolution of at least 800 x 600 pixel.
- A facial expression recognition component is required to interpret the continuous facial expression of a user. The functionality of the facial expression recognition component is explained in detail in section 5.3.
- Matlab 7 is required to be able to run the facial expression recognition tool.
- Interpreted facial expressions are sent to a Java Client over UDP to control the behavior of visualizations. The separate Java Client is used to simplify development of the PDF reader and the visualization framework.
- Eclipse IDE 3.5.2 was used to develop the Java Client.
- A PDF library is required to display PDF documents in the Java Client. In this case the PDF Renderer [48] library for Java was used, because it is very easy to use.
- A user interface was needed to control the behavior of several functions, to display the PDF and to provide feedback. In this case SWT was used because it has some performance merits compared to a SWING GUI.
- The program was developed and runs under a Windows 7/64 bit machine with 4 GB of RAM. Processor architecture was Intel Core i3.

5.2 Workflow Diagram

Figure 5.1 represents a workflow diagram of the Affective PDF Reader. Data from a video input source are processed in a Matlab client, which is responsible for implementing the facial expression recognition component. Interpreted results are sent to a Java Client over UDP. The Java Client then takes the UDP input to control the behavior of the visualizations and the render the PDF content.
5.3 Facial Expression Recognition

Although there are many facial expression recognition tools out there it was not possible to find a tool that from the very first beginning was fitting the needs. The requirements were, that the tool needs to be open source and that it works in real – time. Although I found two very good facial expression recognition tools on the internet, I could not find a tool that was doing the job for free and in real – time. An implementation I really was impressed of was the demo application provided by the Fraunhofer institute [49] that was built from their SHORE library which is a software library for sophisticated object detection. Unfortunately this library is proprietary but at the point of writing this document I could not find any better solution on facial expression recognition. There is a demo application [50] if you want to give it a try.

Fortunately I found a tool programmed in Matlab which is doing facial expression recognition by performing principal component analysis [51]. But not in real – time, so there were some adjustments needed to get it to work with the Affective PDF Reader. The facial expression recognition tool as well as the basic algorithm this tool is using is explained in the next sections.
5.3.1 The Basic Algorithm

The facial expression recognition in the Affective PDF Reader was done in Matlab and is mainly derived from precedent work from Iftekhar Tanveer [52], a graduate student of the Department of Electrical and Computer Engineering in University of Memphis. He developed a basic facial recognition system in Matlab that is able to track changes in facial expression by performing principle component analysis (PCA) to reduce complexity. His program can be downloaded from Mathworks/Matlab forum [51].

The facial expression recognition tool from Tanveer was not designed to work in real-time. To get it to work in real-time with the Affective PDF Reader, several adjustments were needed – they are explained in section 5.3.5. To get a better understanding of the facial expression recognition component the basic functionalities of the algorithm used by Tanveer are explained here.

His tool initially was able to distinguish between 5 basic facial expressions: neutral, happy, sad, anger and disgust. To do so, the algorithm must first be trained to be able to measure the distance from a captured facial expression to the neutral face. Tanveer provided two sets of images. The first one is used to train the algorithm and the second is used to perform the facial expression recognition on. So how is the process of training the algorithm accomplished? This is done by mapping the input images which were given in JPG format to the facial expressions by using a text file of the following form (also called the 'label file'):

Image001.jpg,happy
Image002.jpg,happy
Image003.jpg,happy
Image004.jpg,happy
Image005.jpg, disgust
Image006.jpg, disgust
Image007.jpg, disgust
Image008.jpg, disgust
Image009.jpg, anger
Image010.jpg, anger
Image011.jpg, anger
Image012.jpg, anger
Image013.jpg, sad
Image014.jpg, sad
Image015.jpg, sad
Image016.jpg, sad
Image017.jpg, neutral
Image017.jpg, neutral
Image017.jpg, neutral
Image017.jpg, neutral
...
...

As you can see it is possible to use more than one input image for different kinds of moods. In general this makes the recognition more accurate because it has more reference values to compare to afterwards. The drawback is that fine differences between expressions that are mapped to a certain kind of mood are not recognized as a separate category of a mood. For example if we have two faces with a slightly difference in expression both would be possibly classified as a certain kind of mood.
5.3.2 The Training Process

The training process is done by reading in the image files that are given in the label file. For every image file it is necessary to detect the face first before performing the expression recognition. Tanveer detected his faces by using a skin extractor. So the face region was determined by selecting that kind of pixels that were decided to match a certain kind of skin colour. The face detection used in the original algorithm needs to be done on coloured images therefore. After detection, the image was converted into a greyscale image. In the original algorithm those sections of the image which are not part of the face are then painted black to prevent areas that are not part of the face from further processing. Finally Tanveer removed some noise in his detected faces by using a filter.

The face detection used in the original algorithm turned out to be the bottleneck when trying to use it with the Affective PDF Reader, because this function did not work in real-time. Adjustments that were made to get it to work in real-time are explained in section 5.3.5.

After detection of face regions the principal component analysis was performed. The principal component analysis in our case can be described as transferring an image with high dimensional information into a low dimensional face space that represents the most significant parts of the image. The principal component analysis (PCA) is explained in more details in the next section.

5.3.3 Principle Component Analysis

The information on principle component analysis is mostly taken from a tutorial written by Lindsay I Smith in 2002 [53]. To understand the basic concept of the principle component analysis it is necessary to know about some basic statistics methods like standard deviation [54], variance [55] and covariance [56] which are not covered in this work.

In general standard deviation and variance can only be applied on one dimensional data sets. To determine the statistical coherence of two variables of a multidimensional dataset, covariance is used. And to determine the statistical coherence of several variables of a multidimensional dataset, a covariance matrix is needed. The covariance matrix of a dataset is just about computing the covariance of every possible combination of two variables and putting the results into a matrix.

The PCA is about finding significant patterns in a multidimensional dataset. In order to put more accuracy to the output of the principle component analysis there is need to subtract the mean value from every value in the dataset before computing the covariance matrix. This produces a dataset where the mean is zero [53].

After computing the covariance matrix, the next step is to compute the eigenvectors [57] and eigenvalues [57] of the covariance matrix. The eigenvectors will then show the main amplitudes of the dataset which are considered to be the characteristics of the set. The eigenvector with the highest eigenvalue is the principle component of the dataset, the most significant relation between the data dimensions [53].

The next step is to order the eigenvectors by their relating eigenvalues. This gives a list of eigenvectors ordered by significance. It’s then a matter of choice to proceed with a certain amount of eigenvectors. The more eigenvectors are left out the more information from the original data set is lost. But in many cases it is sufficient to proceed with the most significant (the vector with the highest eigenvalue) part. This is the way PCA reduces the dimension of the dataset.

After ordering and choosing the most significant eigenvectors, they are put into a so called feature vector which is just a matrix of the eigenvectors chosen from the step
before. The last step in PCA is then to multiply the transposed feature vector with the original dataset. This reduces the original dataset in terms that are chosen by the selection of most significant eigenvectors.

That’s a very basic introduction to the PCA algorithm. The PCA in Tanveer’s program is done by the Matlab function ‘princcomp’ used in ‘econ’ mode, which is a tuned version of this function speeding up the computation significantly. The function returns three variables. Description is taken from [58]:

- **COEFF**: is a ”p-by-p matrix, each column containing coefficients for one principal component. The columns are in order of decreasing component variance”.

- **SCORE**: contains the ”principal component scores; that is, the representation of X in the principal component space. Rows of SCORE correspond to observations, columns to components.”

- **latent**: is ”a vector containing the eigenvalues of the covariance matrix of X”.

### 5.3.4 Computing the Distance to the neutral Image

In order to determine the captured facial expression, the results of the PCA need to be compared. But how is this done? Before performing PCA on the training set Tanveer puts all training images into one large array. As already mentioned the mean is subtracted from each value in the array to enhance the accuracy of the results. Finally this array is the input for the PCA. After computation of PCA we receive three variables that have been explained in the last section. Tanveer reduces dimensions by selecting the most significant coefficients from COEFF matrix returned by Matlab function ‘princcomp’. The part taken from COEFF matrix is called the ‘Eigenrange’.

Now to compare another input image with the results from PCA the input image needs to go through the same process as the images in the training process. First the face region is detected, than the image is resized to fit the dimensions of the images of the training set and finally the mean is subtracted from the input image.

But in the subsequent step PCA is not performed on this input image, it gets just multiplied with the most significant coefficients from the COEFF matrix (See section 5.3.3 for explanation on PCA). And finally the Euclidian distances of the values from the matrix that was created by multiplying the input image with the most significant coefficients (the COEFF matrix) are computed with the SCORE variable that was returned by PCA in the training process. This result represents the distances from the input image to every image in the training set. By selecting the lowest distance the input image is assigned to a facial expression.

### 5.3.5 Adjustments to Facial Expression Recognition

As already mentioned several adjustments were necessary to include the functionality of Iftekhar Tanveer’s facial expression recognition tool into the Affective PDF Reader. The first problem was to find a face detection tool that is doing the job of finding face regions from input images reliably and in real-time and to integrate this tool. The face detection tool I used is explained in section 5.3.6. Additional adjustments which were necessary are explained in sections 5.3.7., 5.3.8 and 5.3.9. Furthermore the system should analyze facial expressions in real – time, so the screenshots of the webcam were captured in a loop,
without any timer delay. Although this took a lot of computation power the system was still able to perform the facial expression analysis in real-time.

5.3.6 Face Detection

The Face Detection should deliver useful images as input for the facial expression recognition tool. After some research on the topic I committed to the face detection functionality of the OpenCV library [59]. The face detection of the OpenCV library is an implementation of the algorithm proposed by Paul Viola and Michael J. Jones in 2004 [60]. I found a Matlab implementation in Mathwork forums [61] that is using an OpenCV binary to perform face detection, so I was using this function more like a black box rather than experimenting with it. This Matlab implementation included the OpenCV binary, the haarcascade feature file [62] and the Matlab files where the OpenCV binary gets called with appropriate parameters.

Now what this function does is to find the bonding box of the face region. When the bounding box is found the appropriate vertex points are written to a temporary text file. This was the point where I was able to get the vertex points from the temporary text file to calculate the bounding box of the face region of an input image captured by the webcam of my laptop respectively the input images from the training set.

5.3.7 Cropping Face Region

After getting the bounding box data from the OpenCV facet detection, I was computing a bounding box of fixed size, because the face expression recognition algorithm needs to be provided with images in a fixed size and of course just resizing the images would result into distortion and would finally not work. The computation of a bounding box of fixed size is needed because the output bounding box data of the OpenCV algorithm is adjusting the rectangle according to the distance of the detected face from the camera like shown in Figure 5.2 and Figure 5.3.

Figure 5.2: Bounding box with a near face.
The resulting image after cropping with fixed size looks like:

![Figure 5.4: Cropped image.](image)

The problem with this approach is that the facial expression recognition only works reliably if the faces from the training image set and the captured input images are taken in a constant distance to the webcam.

### 5.3.8 Filtering and Light Adjustments

Filtering and light adjustments make the facial expression recognition more robust against lightning issues and against noise. I first applied a Gaussian filter on the image. Matlab provides the built-in function ‘fspecial’ for that purpose. After filtering the image I did some light compensation by using Matlab function ‘imadjust’. With this function it is possible to raise or to lower the value of certain pixels by providing a threshold. So in the end darker parts of the image are getting brighter and parts of the image that are very bright are getting darker as shown in Figure 5.5 and Figure 5.6.
5.3.9 Applying a Mask

After light compensation and after filtering the image I applied a mask as shown in Figure 5.7, onto the image to prevent not important parts of the image from further processing. I noticed this little trick in a paper called "Mood detection: Implementing a facial expression recognition system" [63]. The image with the applied mask is the input for the facial expression recognition tool. A masked image is shown in Figure 5.8.
The final Matlab application has 5 basic functionalities:

- **Assignment of Facial Expressions**

Before starting the recognition process it is mandatory to have a set of images serving as the training data. To facilitate this process I built in a function to make snapshots of a facial expression and to assign it a different kind of mood. Such an assignment is shown in Figure 5.9.
- **Training of Algorithm**

  The training of the algorithm is done with the provided set of training images. The training process is described in detail in section 5.3.2.

- **Face Detection**

  The face detection needs to be activated before starting to analyse facial expressions. The functionality of the face detection is explained in detail in section 5.3.6.

- **Mood Detection**

  The mood detection makes use of some basic statistical methods and algorithms which is explained in detail in section 5.3.1.

- **Start UDP Server**

  To send the continuous states of the user’s facial expression to the Java client that is responsible for displaying the PDF content and the background visualizations, a UDP server is set up. Currently the computed distance to the neutral image and the detected mood are sent over UDP.

### 5.3.11 User Interface of the Matlab Application

![Screenshot of the Matlab application user interface.](image)

**Figure 5.10:** Screenshot of the Matlab application user interface.
Description for Figure 5.10:

1. Button for taking snapshots
2. Button to enable face detection
3. Button to enable mood detection
4. Button to train the algorithm
5. Button for starting the UDP communication
6. Video feedback screen
7. Feedback box to display textual content.

5.4 PDF Reader Application

The Application to read a PDF and to render the background visualization is done with Java. The Java application was developed with Eclipse and can be divided into the following main parts:

- Graphical User Interface (GUI)
- PDF Reader
- UDP Client
- Affective Model
- Visualization Component

5.4.1 Technologies

The implementation was done on a 64 bit Windows machine. The following technologies, tools and libraries were used:

- JAVA
- SWT [64]
- PDF Renderer [48]
- Java.net
- Eclipse 3.5.2

5.4.2 Graphical User Interface

The graphical user interface (GUI) was developed with Jigloo [65] which is an interface builder and provides a WYSIWYG editor for creating a GUI in Java. This was very handsome in designing the prototype. A first version of the prototype GUI is shown in Figure 5.11.
5.4.3 PDF Reader

The PDF component in the Java client is programmed with the open source PDF Renderer [48] library, which displays PDF content by using Java2D [66]. It just took a while to display the output of the PDF Renderer with SWT, because there were only code examples how display a page in conjunction with a Java SWING GUI which works a bit different compared to SWT. Finally the trick was to convert the AWT BufferedImage output of the PDF Renderer into a SWT ImageData object which was rendered into a SWT canvas with the possibility to zoom and to scroll.

5.4.4 UDP Client

The UDP client is necessary to communicate with the Matlab facial expression recognition application. To do so the Java program creates a UDP Socket and waits for incoming messages on the port defined by the UDP Server of the Matlab application. The protocol of this UDP messages is very simple – it is just a comma separated string of two key value pairs. The first key value pair is the computed distance from the neutral facial expression and the second is the determined mood. These two values are then used for further processing in the visualization component in a way that is defined by the Affective Model.

5.4.5 Affective Model

The affective model is used for modelling the emotional feedback to the user. All assumptions that are made about the user’s emotional state and all assumptions that are made to establish an Affective Loop reflect into the affective model. The affective model can be described to hold the state of the affective feedback and it knows when to transfer from one state to another after an action is triggered. An action in our case would be that the detected mood from the facial expression recognition component has changed. The affective model also keeps track of mood changes in time. For example if
the reader’s facial expression is mapped to the neutral image for a specific period – the affective model assumes that the reader is bored and starts to blur the text of the PDF to force the user to act with another facial expression. Establishing an Affective Model is a creative process and it is experimental work to establish an Affective Loop as it was discussed in section 3.3.

5.4.6 The Colour Mapping

The Affective Model holds also the state of the colour mapping. The colour mapping is used to assign a certain kind of a mood to a certain kind of colour. The colour mapping can be changed in real – time while the background visualization is done. By that way it is possible to personalize the appearance of the visualization a bit more. Changeover of colour is done from the Affective PDF Reader’s user interface through a colour picker as shown in Figure 5.12.

5.4.7 Visualization Component

The visualization component is responsible for rendering the footprint visualization in the background of the PDF according to the data it gets from the affective model and the state from the current colour mapping. Also mouse movements and the position of the mouse pointer are considered in the rendering process to place footprints on screen. The following section takes a closer look on the footprint visualizations of the Affective PDF Reader.

5.4.8 Footprint Visualization

The footprint visualization should establish an Affective Loop by rendering footsteps in the background of a PDF in different sizes and in different colours, according to the data from the affective model (colour mapping, detected mood, mouse movements). The footprint visualization is designed as followed:

It creates a footprint in the background of the PDF when the Affective Model has detected a mood change. The colour of the footprint is set accordingly to the state of the colour mapping. The position on the screen is assigned randomly or by the mouse pointer. A foot print has an initial size. A new footprint on the screen gets larger in time till the affective model has recognized a mood change. Then the scaling of the latest
footprint is stopped and it is set to fade mode, where it gets slowly faded out. At the same time a new footprint is created accordingly to the changed state of the affective model. I guess this is explained best in a few screenshots.

![Figure 5.13: Screenshot of the footprint visualization. Number 1 shows a footprint that has been rendered earlier and has nearly faded out. Number 2 shows a footprint that was rendered lately and represents one of the latest mood changes.](image)

As already mentioned a rendered footprint gets bigger according to the time that elapses till the next mood change is recognized. This is shown in Figure 5.14.
Figure 5.14: Scaling of footprints. A footprint gets bigger the longer its mapped mood is detected (Number 1). Number 2 shows a footstep with a detected mood that did not last that long.

A resulting combination of a PDF with concurrent rendering of footprints looks like Figure 5.15.

Figure 5.15: Combination of a PDF and the rendered footprint visualization with the footprint placed randomly.

Another possibility of placing footprints is to use the position of the mouse pointer. At the time of writing this paper there was no open source gaze tracking software to
find that worked reliably. It would have been a great method for determining the position of footprints. Figure 5.16 shows a screenshot of footprints that are placed by moving the mouse pointer along the text that is read. By that way it is possible to relate captured facial expressions to the text.

5.4.9 GUI functions

Functions that are available through the GUI of the prototype are listed here:

- Possibility to change the size of rendered footprints with a slider.
- Possibility to zoom and scroll the PDF with a slider.
- Enabling/Disabling footprint visualization via button.
- Switching between random positioning of footprints and positioning via mouse pointer.
- Generation of fake moods for testing purpose.
- Changing the colour mapping of detected moods via colour picker.
- Loading and Saving of PDF documents with the current states of footprints.

5.4.10 Blurred PDF Text

The idea of blurring the text of a PDF is to involve the user more and to boost a more expressive reading experience. But how is this accomplished? If the affective model tracks a neutral facial expression and no changes in emotional states are detected in a specific period, then the affective model triggers a method in the render process to...
slightly blur the text of the PDF to force the user to make another facial expression to get back the unblurred PDF and to get involved again. After facial expression has changed the text is rendered for reading again. This is shown in Figure 5.17.

![Figure 5.17: Screenshot of a PDF after it got blurred. This happens when the detected mood is neutral for a certain period. The aim of this is to enhance the user involvement.](image)

5.5 Performance Issue

As already mentioned the performance of the whole system during development was a challenging issue. The facial expression recognition with the Matlab client was consuming a lot of resources but it worked fluently if done in standalone mode without starting the Java client. When starting the Java client concurrently to the Matlab client there is an aggravation in performance which utters in noticeable delays when operating with the Java client user interface. Although computation speed in general is an issue the system was still able to demonstrate the basic aim of this work.

It needs to be pointed out that the current system is a prototype application. The author proclaims that there can be put a lot of effort into redesigning and reprogramming the whole system, for example by using other platforms than Java as stated in future works in chapter 7. As already mentioned Java was used to ease and to speed up the establishment of the prototype application.
6. Conclusion

It is possible to introduce a new dimension of involvement during reading a PDF when using affect recognition systems, in this case facial expression recognition, to control visualizations in the background of a PDF. The application conveys expressive actions during reading and hence influences the process of reading in a subtle way.

Depending on the sophistication of affect recognition systems it is still mandatory to impose a lot of simplifications when trying to model an affective loop. It takes a lot of effort to support harmonious communication in HCI and it takes much more effort to succeed in a more subconscious approach to support emotional communication between a human and a computer that is felt to be right.

The footprint visualization that was used in this project to give affective feedback was done in a way that is not felt to be perturbing or overloading. This was mainly achieved by rendering footprints partly transparent. It is important that forms of continuous visualizations in the background of a PDF are made with caution to avoid running the risk of interrupting the reading process or influence it in a way that is perceived as annoying by the user. It is believed there is a lot of potential in providing affective feedback in such a use case. As affective technologies will enhance, we have better tools to establish affective models that are able to create affective loops in a more elaborated way.

The research question that was raised in the beginning was asking about the possibilities to influence the reading process in a positive way by using affect recognition systems. Reading is a very subtle, a complex and a creative process. Every stimulus appealing to our sensorial inputs influences the process of how we assemble our very personal association to the textual content, hence how we assemble our picture of the world. The emerging personal experience with the Affective PDF Reader showed that we can influence this process in a positive way that is perceived as stimulating and exciting if we provide visual feedback during reading a text that is controlled by affect recognition systems.
7. Future Work

There are a lot of ideas how to improve and to extend the Affective PDF Reader. They are summarized in this chapter.

7.1 Affective Model

Currently the affective model is able to detect a mood change and to trigger the render mechanism of a footprint. It stores the colour mapping of moods and it is capable of simulating boredom by blurring the text of the PDF if the detected mood stays in neutral time for a specific period. By that way, it forces the user to act after a specific period of inactivity with another facial expression to proceed in reading.

There are endless possibilities to extend the affective model. As it keeps track of all changes in data from sensors and tries to establish an affective loop, it is considered as the central intelligent component for interacting with the system. Any kind of interaction chain is processed here. This includes making assumptions of our emotional behaviours to establish an Affective Loop. It is about making assumptions how the user would react or feel about certain kind of feedback. Having knowledge about psychological models of emotions can help a lot in building more elaborated Affective Models that can be used for computing.

As already mentioned, Markov Models could be used to model statistical changeovers from one state to another after specific kind of actions [40]. This is a very handsome technique to breakdown the complexity of emotional interference. Another great chance is seen in establishing learning mechanisms to adapt the Affective Model more and more according to the knowledge we get about the user’s emotional reactions.

7.2 Facial Expression Recognition

The facial expression recognition component used in this thesis did not work very reliably but sufficient enough to demonstrate the basic purpose of the work. Future enhancements include a redesign of the facial expression recognition component according to latest advancements in algorithms. There are many different approaches in doing facial expression recognition for example like performing principal component analysis or using a so-called active appearance model (AAM) [67]. Reliably facial expression recognition is complex and definitely a topic on its own.

7.3 Gaze Tracking

With reliable gaze tracking software it would be possible to determine the screen area where we are looking at. According to coordinates, we get from gaze trackers we could do visualizations on exactly that area of the screen we are looking at. At the point of writing this document there were some interesting and ongoing projects using special camera systems (like IR cameras) or eye pupil reflections for example, to track the point of gaze [68] [69].

7.4 Speech Recognition

Our voice carries a lot of information on affective states and humans are very good in detecting emotions from speech. A speech recognition system that is capable of detecting emotions by using different kind of features would be a great enhancement to the Affective PDF Reader. It would be possible to get another input source for
generating the affective feedback. There is a lot of research going on in this area and results in speech recognition are quite promising [70].

7.5 Visualizations

Future works also include experiments with different kind of visualizations other than footprints. The question is how we can affect the reading experience in a positive way when designing new visualizations. The activity in finding good visualizations that are felt being right and not intrusive or perturbing is a fine line and a quite experimental task. An idea is to include a 3d render framework to gain more flexibility in visualizations.

7.6 Evaluation

An evaluation process would help to understand the impact the Affective PDF Reader has on users. Different reactions of users should be carried out in a user study with subsequent interviews. This could help to get a better insight to the process of affective interaction and to influence it in a positive way.

7.7 Redesigning the System

To improve the overall performance of the system it is a good idea to reprogram some parts of the system and to redesign the architecture to be able to include other affect recognition systems than facial expression analysis.
8. References


Agrawal N, Cosgriff R, Mudur R. Mood Detection: Implementing a facial expression recognition system.


69 Morimoto CH, Mimica MRM. Eye gaze tracking techniques for interactive applications. Universidade de São Paulo, Brazil: Departamento de Ciência da Computação; 2005.


9. Figure Index

Figure 3.1: Transfer of affective states with a Hidden Markov Model applied in a playmate robot for children[40] ........................................................................................................ 13
Figure 3.2: PAD vector representation [41] ........................................................................................................ 13
Figure 4.1: Basic gestures of the eMoto project[43] .......................................................................................... 17
Figure 4.2: Colour map in eMoto project to express emotions [43] ................................................................. 18
Figure 4.3: Affective text message in eMoto [43] .......................................................................................... 18
Figure 4.4: Affective text message in eMoto [43] .......................................................................................... 18
Figure 4.5: Affective text message in eMoto [43] .......................................................................................... 19
Figure 4.6: Screenshot of EmoteMail prototype [46] ...................................................................................... 19
Figure 4.7: Screenthot of the facial affect analyzer in FAIM [47] ............................................................... 20
Figure 4.8: Workflow of the FAIM system [47] .......................................................................................... 21
Figure 5.1: A workflow diagram of the Affective PDF Reader ................................................................. 23
Figure 5.2: Bounding box with a near face. .................................................................................................. 27
Figure 5.3: Bounding box with a far face...................................................................................................... 28
Figure 5.4: Cropped image .......................................................................................................................... 28
Figure 5.5: Image without filtering and light compensation: ................................................................. 29
Figure 5.6: Image after light compensation and filtering ........................................................................... 29
Figure 5.7: Mask that has been applied to input images. ............................................................................ 29
Figure 5.8: Masked image ............................................................................................................................ 30
Figure 5.9: Assignment of facial expressions to moods ............................................................................... 30
Figure 5.10: Screenshot of the Matlab application user interface .............................................................. 31
Figure 5.11: First draft of the Java client ...................................................................................................... 33
Figure 5.12: Colour picker used to map colours to moods .......................................................................... 34
Figure 5.13: Screenshot of the foot print visualization. Number 1 shows a footprint that has been rendered earlier and has nearly faded out. Number 2 shows a footprint that was rendered lately and represents one of the latest mood changes. .................. 35
Figure 5.14: Scaling of footprints. A footprint gets bigger the longer its mapped mood is detected (Number 1). Number 2 shows a footstep with a detected mood that did not last that long. ......................................................................................... 36
Figure 5.15: Combination of a PDF and the rendered footprint visualization with the footprint placed randomly. ......................................................................................................................... 36
Figure 5.16: Footprint walk with the mouse pointer. By moving the mouse pointer along the text it is possible to relate captured facial expressions to the text that has been read ............................................................................................................................ 37
Figure 5.17: Screenshot of a PDF after it got blurred. This happens when the detected mood is neutral for a certain period. The aim of this is to enhance the user involvement ........................................................................................................... 38
## 10. List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAM</td>
<td>Active Appearance Model</td>
</tr>
<tr>
<td>AU</td>
<td>Action Units</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>FACS</td>
<td>Facial Action Coding System</td>
</tr>
<tr>
<td>FAIM</td>
<td>Facial Affect in Instant Messaging</td>
</tr>
<tr>
<td>GSR</td>
<td>Galvanic Skin Response</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>PAD</td>
<td>Pleasure-Arousal-Dominance</td>
</tr>
<tr>
<td>PCA</td>
<td>Principle Component Analysis</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>What You See Is What You Get</td>
</tr>
</tbody>
</table>
11. Tables

Table 3.1: Body Measures ........................................................................................................11