Communication Error Detection using Facial Expressions

by

Sy Bor Wang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author ....
Department of Electrical Engineering and Computer Science
August 29, 2008

Certified by ....
Trevor J. Darrell
Associate Professor
Thesis Supervisor

Certified by ....
David Demirdjian
Research Scientist
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Communication Error Detection using Facial Expressions

by

Sy Bor Wang

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Automatic detection of communication errors in conversational systems typically rely only on acoustic cues. However, perceptual studies have indicated that speakers do exhibit visual communication error cues passively during the system's conversational turn. In this thesis, we introduce novel algorithms for face and body gesture recognition and present the first automatic system for detecting communication errors using facial expressions during the system's turn. This is useful as it detects communication problems before the user speaks a reply. To detect communication problems accurately and efficiently we develop novel extensions to hidden-state discriminative methods. We also present results that show when human subjects become aware that the conversational system is capable of receiving visual input, they become more communicative visually yet naturally.

Thesis Supervisor: Trevor J. Darrell
Title: Associate Professor

Thesis Supervisor: David Demirdjian
Title: Research Scientist

Acknowledgments

There are many people I would like to thank over here, this thesis would not be possible without the guidance, moral support and help of all these people.

First, I would like to thank my thesis advisors, Trevor Darrell and David Demirdjian. David has been a wonderful advisor and has offered me many good advice in research and in life. I am very grateful to his nurturing, kind and patient attitude towards me throughout my last five years in the Vision Interface group as he took the time to advise me, even though it was not part of his duty. He has been a great mentor. I would like to thank Trevor for being a great co-advisor as well. He offered very good ideas, had great positive energy and was always very encouraging. Throughout the years he has taught me a lot about writing and doing research. I would like to thank him for all his generosity with funding me all these years and the extra effort to fund me this summer.

Stefanie Shattuck-Hufnagel and Jim Glass have been wonderful thesis committee members. I am grateful to Stefanie for all her advice over the years about conducting my experiments. Despite only being a thesis committee member, she has spent a lot of time advising me in this thesis about conducting experiments and the use of audio features throughout the last few years. Jim was a wonderful and kind advisor when I was completing my masters in his research group. They also read my thesis in detail and provided very insightful feedback.

This thesis of mine has been highly collaborative and many members of the Vision Interface Group had a hand in making it complete. Ariadna Quattoni, Mario Christodias, Tom Yeh, Louis Phillppe Morency, Kevin Wilson and Kate Saenko were wonderful paper readers who often provided invaluable feedback, assisted me in data collection, and provided good research advice. Other members of CSAIL who provided a lot of help and ideas were Mike Siracusa, Kin Tieu, Biswajit Bose, Gerald Dalley, Javier Carreras, Thomas Yeo and Wanmei Ou. Other friends who were there when times were hard were Alex Park, Han Pang Chiu, Huankiat Seh, Tony Quek and Ulas Ziyan. Several other friends I would to thank for assisting me in data collection include: Xiaogang Wang, Jenny Yuen, Kevin Chen, Emily Whiting, Yeuhi Abe, Paul Green, Clare Poynton, Emily Fox, Zi Tec Chong, Jonathan Quek, Biliana Kaneva, Trevor Ng, Trina Kok, Shuyin Sim, Wuisiew Tan, Racquel Urtasun, Valerie Gordeski, Tony Eng, Yidong Chong and Jiamin Chen. Special thanks to Maysoon as she was always available to read my papers, thesis and correct my grammar.

There are several other members of the MIT community I would like to thank as well. Jae Lim provided me with funding when I was in the midst of switching research groups, and I became his teaching assistant for 6.003 for two semesters. He had a lot of wisdom in dealing with difficult life situations and I was grateful I was there at the right time when I needed such advice. Marilyn Pierce in the graduate department of EECS was also very helpful and she was constantly looking out for my well-being here in MIT.

I would also like to thank my research advisors back in Carnegie Mellon where I was an undergraduate. I never thought I would be here in MIT doing a phd without the kind and nurturing guidance of my undergraduate research advisor, Tsuhan Chen and Ed Schlesinger. Tsuhan was very inspirational and sparked my interest in image processing and computer vision interfaces.

Last but not least, I would like to thank my family for all their patience and help throughout all these years of my education. The words here will not do justice to the amount of help and guidance they have provided me. Dad was very motivational and encouraged me to be the best that I can be, while Mum was very caring and constantly watched over my health and well being. My two sisters were behind me all these times too when I was studying, from providing me with four wheels to giving invaluable advice when I am facing difficult situations. I wouldn't be here without the support of all of them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many spoken dialogue systems have difficulty detecting the occurrence of communication errors. By our definition, a communication error occurs when the conversational system misinterprets the user and makes an erroneous reply. Considerable research has been invested in monitoring audio cues to detect such communication errors. Various researchers have shown that human users change their speaking style when they encounter a problem with a conversational system [59]. For example, users tend to speak slower or louder when speech recognition errors occur. These problems motivated the monitoring of prosodic aspects of a speaker's utterances. Different studies using prosodic features to detect communication errors automatically have achieved varying results. For example, Oviatt et al. [59] showed that hyper-articulation in speakers is highly indicative of system errors. However, Ang et al. [1] conducted a separate study and contradicted Oviatt's findings. Another group of researchers demonstrated the use of speech recognition confidence scores [3, 22] to detect errors robustly. Hirschberg et al. [41], showed otherwise and that a combination of prosody features and the confidence scores attained better accuracy.

A recent perceptual study [5] indicated that conversational error detection can be improved significantly if visual modalities are taken into account when the speaker is listening. This study showed that human observers performed better at recognizing communication errors when they were given the visual footage when the speaker is listening to the system's response as compared to when only audio recordings were provided. This insight motivates us to detect communication errors automatically by using the non-verbal facial

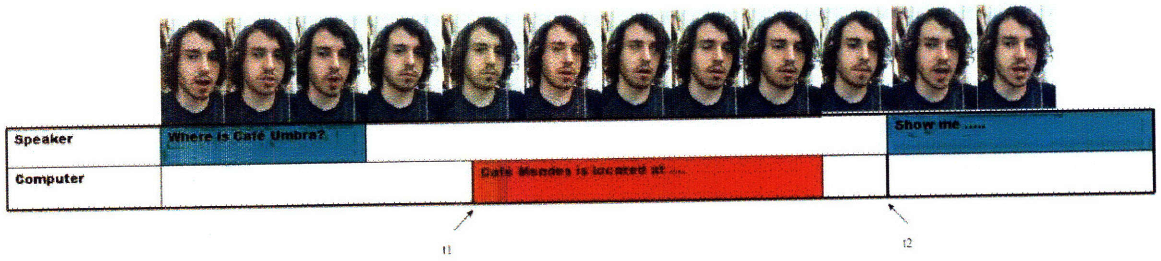


Figure 1-1: Diagram illustrating a speaker encountering a communication error. Notice that during the system's conversational turn (from time $t1$ to time $t2$), where it is giving an erroneous reply, the speaker twitched his eyes naturally to signal an error. This suggests that visual analysis during the system turn may be useful.

expressions of the speakers when the system is making its reply. Figure 1-1 illustrates the reaction of a user experiencing a speech recognition error from a conversational system. Notice that when the system is giving its reply, the co-occurring facial expressions can be indicative of the communication state. In most existing work, when a system makes an erroneous reply, errors are detected only during the speaker's next turn in the conversation, but the above insight shows that we can possibly detect errors within the same turn. Systems could potentially benefit from this early feedback to improve the quality of the conversation.

Figure 1-2 shows several other facial expressions that co-occurred with communication errors in various datasets we have collected (which we will describe in detail in Chapters 4.3 and 5). From this figure, different negative-valence emotions typically occur with a communication error e.g. frustration, disgust, confusion etc. It is difficult to precisely determine the emotions that are involved in communication error, so we have taken the approach of detecting error conditions directly from perceptual signals, rather than via an intermediate emotion representation. Note that we are not claiming that we can detect communication errors in all scenarios; rather, the detection attains high accuracy during specific types of replies by the system. An instance where our mechanism may break down could be where no speech recognition error occurred, but the speaker did not like the path the course of the conversation was taking and he/she may roll his/her eyes. Clearly the distinction between error and frustration is a subtle one; as described in more detail in later chapters we presume ground truth is available for training, marking with boolean labels which dialog turns resulted in a communication error.

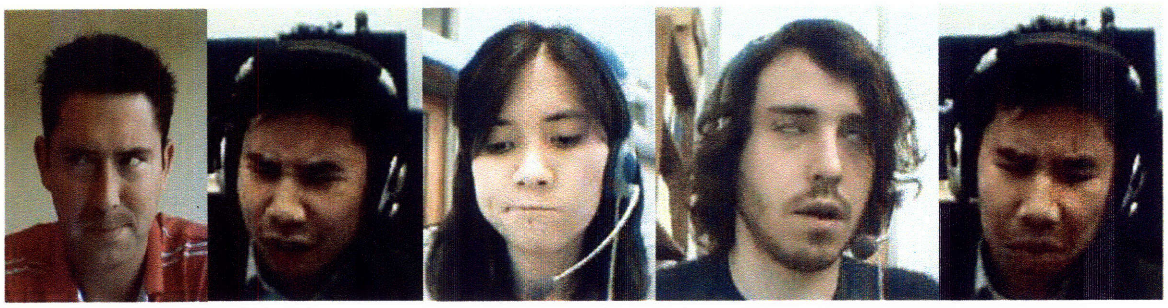


Figure 1-2: Images of different subjects during a communication error. Notice that different emotions are conveyed e.g. frustration, confusion, disgust etc, but they all occur during a communication error. Our work will not detect these emotions, but instead focus on detecting communication errors using these indicative visual cues.

Detecting communication errors robustly, faces many challenges. First, there is a large variety of expressions that are indicative of an error, e.g. from subtle expressions like a stare or a frown, to highly distinctive ones like a pout or a sulk. The model needs to learn this high variability of expressions and recognize them as the same error state. Second, while a single image frame of the expression may not be distinct, the sequential dynamics of these frames and the motion patterns of some expressions become highly indicative of an error, e.g. a head shake, where the speaker typically begins with his head in a frontal pose, shifts his gaze to the left/right and reverses the direction after. This implies that the model should incorporate a structure that learns these sequence dynamics accurately. Lastly, in the sequential dynamics of these expressions, communication errors and non-errors may share a common structure. For example, a neutral, frontal face usually precedes or follows a pouting expression, which conveys a communication error. This neutral, frontal face also occurs frequently when the system reply is correct. Hence, learning a common structure between different error states could potentially be useful. Given these challenges and constraints, we will argue that a hidden state model is the most appropriate solution. Hidden state models can capture the within class structure *and* the between structure by the use of hidden states.

We begin by presenting a model which shares features between gesture classes in Hidden Markov Models (HMMs), and then proceed to explore the use of discriminative hidden state models, including Hidden Conditional Random Fields and Latent Dynamic Conditional Random Fields. At the end of the thesis, we develop a novel voting Latent Dynamic Conditional Random Fields method for the segmented gesture classification task which is significantly easier to train than HCRFs due to the use of an *a priori* partitioned latent space; this method was also empirically more accurate in most cases we tested, perhaps due to better convergence at the expense of latent space optimality.

1.1 Contributions

In this thesis, we develop a communication error detector by analyzing the speaker's facial expressions.

The key contributions of this thesis are:

- The first automatic system for detecting communication errors using facial expressions during the system's turn.
- Demonstrating that Gaussian Mixture Models-Linear Discriminant Analysis (GMM-LDA) and Hidden Conditional Random Fields(HCRF) are useful models for visual gesture recognition.
- Developing a voting Latent Dynamic Conditional Random Fields algorithm for efficient segmented gesture recognition.

1.2 Thesis Roadmap

The rest of this thesis will describe the various contributions in detail. In the next chapter, I will review related work. Before addressing visual conversational error detection, I will present several preliminary methods I have shown to be useful for visual gesture recognition and which motivated my later results: in Chapter **??**, I will describe my experiments applying Gaussian Mixture Models (GMMs) and Linear Discriminant Analysis (LDA) for arm gesture recognition and in Chapter 3.2 I will present experiments demonstrating Hidden Conditional Random Fields to be a useful model for arm and head gesture recognition. After describing gesture recognition experiments, Chapter 4 describes our approach in detecting communication errors and the techniques I have used for face and head tracking. In the following chapter, I will describe using Hidden Conditional Random Fields for communication error detection. Chapter 5 will introduce voting Latent Dynamic Random Fields and how we use it for detecting communication errors. I conclude in Chapter 6 by discussing future work and limitations of our results so far.

Chapter 2

Related Work

In this chapter, I frame the problem setting for my research and overview related work. Many factors need to be considered: namely, the audio cues, the visual cues, the simultaneous audiovisual cues and the turn of the conversation (system or speaker) etc. I will divide related work generally into three categories, namely, audiovisual emotion recognition (audiovisual), system error detection (audio) and facial expression analysis (visual).

2.1 Audio cues

There has been considerable research activity in the domain of detecting communication problems using audio cues. All of these are focused on the audio cues produced during the speaker's turn. Oviatt [59] showed that there is a pattern of hyper-articulation in speakers when they are trying to correct system errors. The speaker's hyperarticulation subsequently leads to even worse recognition performance, due to deviations from trained acoustic and language models [72]. This work demonstrated that prosody features and statistics like pause durations, average pitch, pitch range etc were useful, but no machine learning method was used to learn and detect these errors automatically. Apart from prosodic features, another possible approach was to use recognition confidence scores to reject any unreliable recognizer hypotheses, either caused by hyperarticulation, or due to other reasons such as out-of-vocabulary words, noise interference, etc. Several works [3, 22] have focused on using these scores, and proposed schemes that work mainly at the frame, phoneme, or word level. For instance, word-level confidence annotation assigns a reliability tag to each word token in the decoder hypothesis. Typically a two-class annotation scheme is used, which marks the word instance correct or incorrect. However, this type of model is not always sufficient for dialog systems. Hirschberg et al. [41] subsequently developed an automated system that used a combination of prosody features and speech recognizer confidence scores to detect errors. In this system, a rule based classification model was used for error learning and detection. Mixed results, however, were attained during evaluation using different datasets. In the first dataset, the speech recordings were manually segmented, and using only prosody to predict system errors, outperformed using automatic speech recognition (ASR) confidence scores, whereas in the second dataset, the combination of ASR confidences scores and prosody scores produced better results.

This suggests that additional higher level features could be incorporated to improve error detection. Hirschberg et al. [42] used a combination of prosody features and other lexical features available to the speech recognizer such as dialog history etc. to detect speaker corrections. This combination of different audio features detected corrections more accurately than traditional methods for ASR rejection. However, in this work, experiments were evaluated on a speaker-dependent model. A more recent work by Skantze et. al [73] also affirmed that a combination of word confidence scores, contextual and lexical features improve error detection. Although the detection is automatic, it suffers from high dependence on confidence scores of specific content words, which is not sufficient for dialog systems.

A related issue regards detecting the emotional state of a user, because humans sometimes extend their inter-personal behavior into their interaction with computers. For example, users might express their satisfaction verbally or prosodically when a system works well for them; on the other hand, they tend to get frustrated and angry when a system keeps making errors. Thus, it is useful for a dialogue system to detect user emotion and react properly. Ang et al. [1] collected a naturally occurring dataset consisting of speakers engaged in an air travel reservation task that itself gives rise to emotions. This work used a combination of prosodic, langauge model and ASR features. Results show that a prosodic model can predict whether an utterance is neutral versus annoyed or frustrated with an accuracy on par with that of human interlabeler agreement. Accuracy increases when discriminating frustration from other utterances, and when using only those utterances on which labelers originally agreed. However, human interlabeler agreement in this dataset was only 71% and the speakers utterances were very short e.g. yes or no, which may not be sufficient in representing the variability of words a speaker will utter under varying emotional states. The "frustrated" vs else experiment also involved very little data. At the same time, this work suggested that hyper-articulation may not be a good predictor of annoyance or frustration, which is a conflict with other previous work [59]. In general, research on emotion detection in speech has not found a unified set of features that is indicative of errors consistently in different experimental settings, and obtaining objective ground truth remains challenging. Finding good acoustic cues for various emotional modes remains an interesting research problem. It is also desirable to use data from real interactions between a user and a computer, rather than data in simulated or acted mood, to carry out the experiments. In real interactions, the speaker's visual facial expressions can provide important cues to indicate communication errors.

2.2 Visual cues

Most of the vision community working on tracking face or head features do not focus their attention on detecting communication errors in conversational systems. Instead, their attention is turned towards facial expression analysis. There are two main streams in the current research, namely facial affect (emotion) detection and facial action unit detection. Facial affect detection aims to draw high level classifications e.g. emotional states, personality etc., from a given display of facial expressions. This can be achieved by passing the extracted facial features e.g. from an active appearance model (AAM), directly into a machine learning algorithm like Support Vector Machines (SVMs) for direct evaluation [2]. Research involving Facial Action units (AUs), on the other hand, attempt to provide a quantitative description of the face, by means of assigning intermediate labels, while the inference or classification using these labels is left to another high order decision maker. For example, a brow furrow can be judged as anger in an affect detection approach and as a facial movement that lowers and pulls the eyebrows closer together in an facial action unit approach. The most commonly used method for manual labeling facial actions is Facial Action Coding System (FACS) proposed by Ekman et al. [27]. FACS is a comprehensive and anatomically based system that is used to measure all visually discernible facial movements in terms of facial Action Units.

There is some controversy over the need to use FACS for facial expression analysis. Many surveys claimed [63, 97, 21] that action units are independent of interpretation, hence they can be used for any high-level decision making process including recognition of basic emotions according to Emotional FACS (EMFACS) rules [1]. The labeling for the facial action units has been a manual process, which requires a trained human observer, although recently, there has been a considerable amount of work on automated estimation of FAC primitives. In addition, in existing systems that do not use FACS [2], facial features extracted from images are also independent of interpretation, and the higher order inference is left to machine learning algorithms. Lastly, the approach of using FACs is not tested by the developmental psychology community. Does a baby infant decompose his parent's facial expressions into action units before making the inference about his parent's emotional state? Or does the baby simply use the overall appearance (texture and shape) of the face to make the inference? To the best of our knowledge, there has been no such study. On the other hand, for psychology studies, a clear, unambiguous approach is required for human coders to discover or establish the relationship between facial behavior with cognitive state, and FACS by far has been the most objective method available.

As for interpretation or classification categories, most of the existing efforts classify expressions into the six basic emotions (fear, sadness, happiness, anger, disgust, surprise). These classification categories are proposed by Ekman [26] and discrete emotion theorists, who suggest that these six emotions are universally displayed and have a marked reference representation in our affective lives. There are a few efforts to detect non-basic affective states from deliberately displayed facial expressions. For example, Gu and Ji and Ji et al. [35, 38] used facial action units as features to a Dynamic Bayesian Network(DBN) to detect fatigue. Yeasin et. al [91] implemented a two-stage classification approach that first recognizes six universal facial expressions, and later computes the levels of interest by

[1]http://face-and-emotion.com/dataface/general/homepage.jsp

mapping the facial expressions into a 3-D affect space.

The following works mentioned in this paragraph are closest to our thesis as they all use visual analysis of a user's frustration or disagreement in an interactive setting. El Kaliouby et al. [29], proposed an automated DBN system that uses head motion and facial action units to classify the following mental states: agreement, disagreement, interest, concentration, uncertainty, thought, but the dataset consisted of emotions performed on demand and were not necessarily reflective of realistic conversation. Teeters et. al [78] extended El Kaliouby's work and presented a wearable camera system for real time classification of the same mental states in a human-to-human social interaction setting as a self-reflection tool. Communication errors could possibly have been inferred from the detection of several mental states, but to our knowledge, there was no attempt to learn such an association and no analysis on how the conversation state could be correlated to the mental state. Kapoor et al. [46] used a Gaussian process classification technique to detect user frustration and included several other modalities like skin conductance, mouse pressure as feature inputs. In a separate work, Kapoor and Picard [45] used a mixture of Gaussian processes for multi-sensory fusion to detect interest in children. Both works were not directly evaluated in a conversational dialog setting.

In all these works, with the exception of [45, 46], the facial expressions are deliberate or exaggerated displays. Studies have shown that people do different things with their faces when posing versus during a spontaneous experience. Ekman et al. [28] found that directed facial action tasks differ in timing from spontaneously occurring behavior. Cohn et al. [19] showed that spontaneous smiles were of smaller amplitude and had a larger and more consistent relation between amplitude and duration than deliberate smiles. Valstar et al. [81] developed a multimodal approach to distinguish between posed and spontaneous smiles and attained a 94% accuracy. Tian et. al. [77] found that many people are able to raise their outer brows spontaneously while leaving their inner brows at rest but few can perform this action voluntarily. Valstar et al. [80] showed that the temporal dynamics of brow actions like velocity, duration, and order of occurrence of brow actions are highly relevant parameters for distinguishing posed from spontaneous brow actions. Littlewort et al. [50] found that the lower eye brow action unit was the key feature in distinguishing between faked pain from spontaneous pain expressions. Given these recent spate of findings, there has been a growing effort on automatic analysis of spontaneous facial expressions. Some of them study automatic recognition of Action Units rather than emotions from spontaneous facial displays. For example, Valstar et al. [81] and Cohn et al. [19] analyzed the muscle activity of orbicularis oculi (action unit 6), zygomaticus major (action unit 12), depressor anguli oris (action unit 15), and mentalis (action unit 17) for distinguishing posed smiles from spontaneous ones.

In general, the extracted visual features can be divided into two types: 2D spatio-temporal features or 3D face models. For 2D spatio-temporal facial features, the extracted features are typically either geometric features such as the shapes of the facial components (eyes, mouth, etc.) and the location of facial salient points (corners of the eyes, mouth, etc.) or appearance features representing the facial texture including wrinkles, bulges, and furrows. For geometric-feature based methods, Chang et al. [15] used a shape model that consisted of 58 facial landmarks. The shape model is based on manifold embedding and probabilistic modeling in the embedded space. In his system, however, all of the learning/recognition is offline and far from real time. There is no rescue strategy when the system fails, which is necessary in a practical system. Pantic and her colleagues [64, 81] extracted a set of facial characteristic points around the mouth, eyes, eyebrows, nose, and chin. They use a point-based model composed of two 2D facial views, the frontal and the side view. Multiple feature detectors are applied to each prominent facial feature, and the best of the detected features are selected. This system cannot deal with minor inaccuracies of the extracted facial data and it deals merely with images of faces without facial hair or glasses. For appearance-feature based methods, Bartlett et al. [8] and Guo and Dyer [36] used Gabor wavelets as inputs to their classifier for Action Unit recognition. Although recognition rates are high, Gabor wavelets are inefficient in memory usage and slow due to the high redundancy of the Gabor representation. A more efficient representation is proposed by Whitehill and Omlin [89] who used boosted Haar features.

A few approaches to automatic facial expression analysis are based on 3D face models. Huang and his colleagues [18, 71, 88, 95] used features extracted by a 3D face tracker called Piecewise Bezier Volume Deformation Tracker [76]. This model calculates correspondences based on the geometry using a 3D deformable/morphable model. While the recovered low dimensional model can be used effectively in classification, the low degree of freedom in this model is not a sufficient representation of spontaneous facial expressions, since many distinct characteristics lie in the subtle details such as the wrinkles and the furrows that are generated by highly local skin deformations. Cohn et al. [20] focused on analysis of brow action units and head movement based on a cylindrical head model [90]. Chang et al. [15] and Yin et al. [83, 92] used 3D expression data for facial expression recognition. In general, the advantage of 3D face models is that they may yield view-independent facial expression recognition, which is important for spontaneous facial expression recognition because the subject can be recorded in less controlled, real-world settings. However, the trade off would be the the lower degree of freedom in the model.

For a more detailed survey of machine analysis of facial expressions and the state of the art methods, readers are referred to two comprehensive reviews by Fasel et al. [30] and Pantic et al. [63].

2.3 Audiovisual cues

A comprehensive survey of using audiovisual cues for emotion recognition can be found in [23, 97]. The most recent and relevant work was presented by Zeng et al. [94], who used a voting method to combine the facial expression and prosody-based emotion classifiers to improve affect recognition. However, due to the higher frame rates coming from the audio classifier, the voting mechanism is biased more to audio cues than visual cues. In addition, their facial expression tracker requires initialization from a neutral expression. Chen [16] fused the scores of facial feature and prosody classifiers occurring sequentially in time. In this work, however, a pure facial expression is assumed to precede and follow a spoken sentence, which is not a realistic assumption and automatic segmentation of this facial expression is poorly defined. Yoshitomi et al. [93], used a weighted sum of neural network classifiers to combine audio and visual features. Song et al. [74] proposed a tripled hidden Markov Model (HMM) to perform audio visual emotion recognition. The key advantage of this model was allowing asynchrony of the audio and visual observation sequences while preserving their natural correlation over time. All these audiovisual methods however, were evaluated on facial expressions or emotions that are deliberately produced upon request.

There are several recent efforts on detecting spontaneous affect audiovisually under various data collection scenarios. Caridakis et al. [13] used data collected in Wizard of OZ scenarios. In this system, as visual input, facial animation parameters (FAPs) were extracted from key facial feature points e.g. eyes, nostrils, mouth etc, per frame, while segmented-based prosodic features were extracted as audio input. Both features are passed into a Recurrent Neural Network(RNN) for evaluation into four categories. Zeng et al. [96], used the data collected in a psychological research interview (Adult Attachment Interview). This work proposed an Adaboost multi-stream hidden Markov model (AMHMM) to integrate audio and visual affective information. The sample size of the datasets collected for evaluation by Caridakis et al. [13] and Zeng et al. [96] is very small, containing four [13] or two subjects [96] respectively.

Note that all the above techniques for visual analysis of emotion that occur during the speakers turn, could be applied to communication error detection during the system's turn. The key difference lies in the features used. Typically, affect recognition in audiovisual analysis like Zeng et al. [94, 96] and Chen et al. [16] during the speaker's turn, will suffer from poorer discrimination of facial expression among different affect. This is because movements of facial features are related to both affective states and the co-occurring content of speech. Hence, during the speaker's turn, features that represent the facial movements around the mouth may not be very useful. On the other hand, for the task of communication error detection, during the system's turn, the subject is listening in silence most of the time, and all facial motion features can be used. These facial motion features are typically very subtle, occur at very small regions of the face, and happen almost instantaneously. However, the 3D-models used in [94, 96, 20, 71] have a low degree of freedom and may not be sufficient in representing the high degree of freedom of motions in the face. Likewise, the features used in [45, 29, 78], only track the lips, the eyes and the eyebrows; other regions around the face e.g. cheeks (see Figure 4-2), which we will show in Chapter 4.2, are also very useful indicators of communication errors.

We believe our work [87] is the first one to detect communication errors visually during

Table 2.1: Breakdown of related work using visual cues.

Work	Types of Features	Classification
Chang et al. [15]	Active shape model(ASM)	GMM in embedded space
Kapoor et al. [45]	Various	Mixture of Gaussian processes
Wang et al. [87](and this thesis)	Optical Flow	Hidden state models
Zeng et al. [94]	PBVD	Naive Bayes
Caridakis [13]	FAP	Recurrent Neural Networks
Chen et al. [16]	PBVD	SNoW
Bartlett et al. [7]	Gabor filters	Adaboost
Whitehill and Omlin [89]	Haar features	Adaboost
Lucey et al. [53]	Active Appearance Model(AAM)	SVM

PBVD: Piecewise Bezier Volume Deformation
GMM: Gaussian Mixture Model
FAP: Facial Animation Points
SNoW: Sparse Network of Winnows

the conversational system's reply. Our work is motivated by a perceptual study conducted by Barkhuysen et al. [5], which showed that human observers performed better at detecting errors when they only view the visual footage of the speaker listening to the response of the system than when they only use audio cues. Previous works only use the speaker's audio and/or visual features in the next conversational turn to detect errors, but this psychological insight shows that we could detect the errors earlier within the current turn. In addition, we are only concerned with a binary detection of communication errors, without the need for labeling the specific emotions. As such, our labeling process is very simple and unambiguous. A speaker makes a query, and the system gives a reply. If the reply is incorrect, a communication error has occurred. A table summarizing related work in conversational systems is shown in Table 2.2 and a table summarizing related work using visual cues is shown in Table 2.1.

Table 2.2: Breakdown of related work specific to conversational systems. Note that this thesis is not limited to visual cues only.

Work	Emotion / Communication detection?	Input	Turn of conversation
Oviatt et al. [59] Shriberg et al. [72] Hirschberg et al. [42] Ang et al. [1] Bansal et al. [3]	Communication	Audio only	Speaker
Wang et al. [87] (and this thesis)	Communication	Mostly Visual	System
Kapoor et al. [45]	Emotion	Visual+others	Not conversational, learning setting
Teeters et. al [78]	Emotion	Visual only	Not conversational, social setting
El Kaliouby et al. [29]	Emotion	Visual only	Not conversational, pre-recorded enactment
Zeng et al. [94, 96] Caridakis [13] Chen et al. [16]	Emotion	Audiovisual	Speaker

Chapter 3

Gesture Recognition with discriminative models

In this chapter, we will describe two techniques we have explored to improve gesture recognition that predated our work on communication error detection. Communication error detection is a type of gesture recognition problem and the algorithms from our gesture recognition analysis led to the development of the algorithm for error detection that we will show later. In our introduction, we defined our problem as detecting a communication error given a pre-segmented sequence of observations. This problem can also be framed as making two possible interpretations, error or no error, from a given sequence. Such a formulation is analogous to a "two class" segmented gesture recognition problem (as illustrated in Figure 3-1). Readers who are only interested in communication error detection can skip this chapter and proceed directly to Chapter 4.

The key intuition driving the design of the algorithms is that many human gestures classes, while distinct at specific key frames, still share some common features or states. For example, Figure 3-2 shows two subjects articulating two different arm gesture sequences. The top image sequence is a gesture telling the computer screen to shrink laterally, while the bottom sequence is a gesture telling the system to display the next screen. If we just compare frame a and frame d, we can see two distinctive body poses, which tells the two gestures apart. However, if we only look at frames b,c and e, the body poses look very similar. This suggests that a model which allows different gesture classes to share

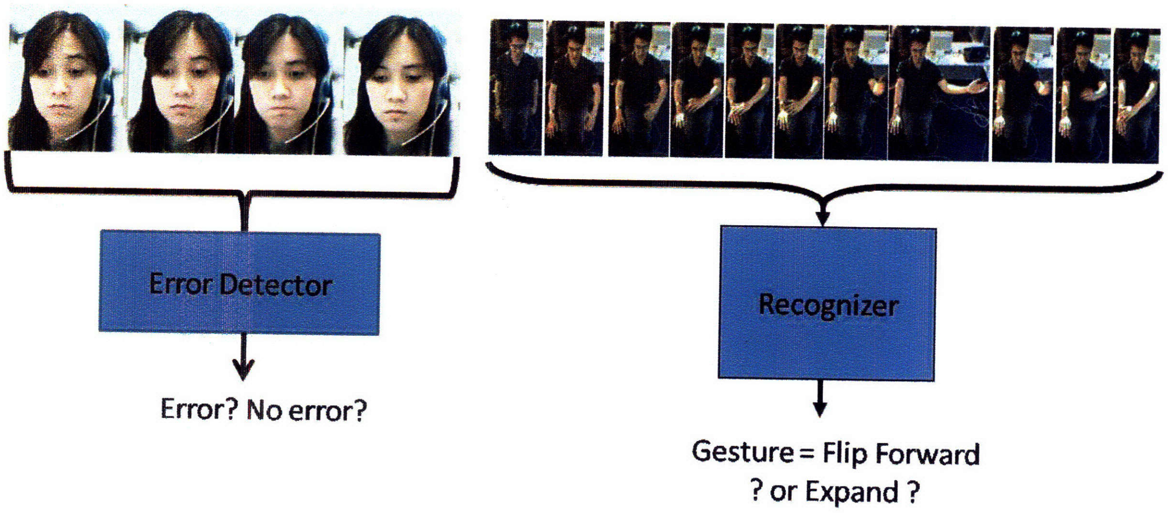


Figure 3-1: Diagram illustrating the similarity between error detection (system diagram on the left) and gesture recognition (system diagram on the right).

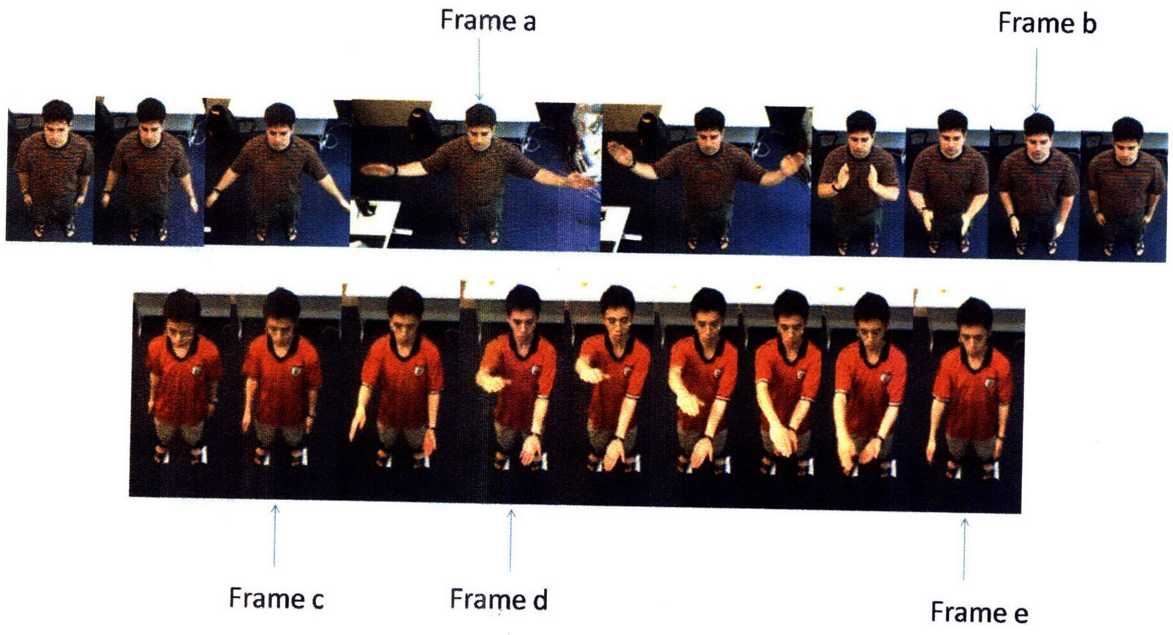


Figure 3-2: Two gesture sequences displayed from frame to frame. The top sequence is a gesture commanding the system to shrink the screen in size, while the bottom sequence is a gesture asking the system to switch to the next screen. The body poses in frames a and d are very different from one another, however, the body poses in frames b, c and e are very similar. This suggests that models that allow shared features could be useful.

common body poses will be useful. At the same time, we can tell the two gestures in Figure 3-2 apart from the sequential dynamics between image frames. This suggests that a model should incorporate a structure that learns the pose transitions. A hidden state model becomes the most appropriate because the hidden states have the dual purpose of allowing the sharing of features between classes, and of learning the sequential structure of the gestures. To share features, common poses in difference gesture classes can be assigned the same hidden state label. In practice, fully observable models are not as successful because the observations do not provide a complete description of gesture sequences and to allow for sharing of features between classes, fully observable state labels will be required. We begin with discriminative vector quantizing features in HMMs, which provide evidence that sharing features is useful.

3.1 Discriminative Vector Quantization

One of the most common approaches for gesture recognition is based on the Hidden Markov Model (HMMs) [69]. Typically, a separate HMM model is trained per gesture class, and during evaluation, a test sequence is passed into each HMM model, and the model with the highest scoring likelihood is the interpreted class of the test sequence. The drawback with this approach is that the HMM models are trained independently of one another, without allowing the features to share a common subspace before training or evaluation. To overcome this, we have shown [84] how Linear Discriminant Analysis (LDA), a technique used in feature separation of phonemes in speech recognition [43], can improve the performance of an arm gesture recognition system.[1]

We propose modeling body gestures using elementary units, called "gestemes" [44]. Gestemes are estimated in the following way:

An input feature vector, f_n is a vector of K consecutive poses.

$$f_n = (\Pi_n, \Pi_{n+1}, ..., , \Pi_{n+K-1})^T$$

[1]The material in this section was previously published in [84] S. Wang and D. Demirdjian "Inferring body pose using speech content". In Proc. ICMI, October 2005, Trento, Italy.

Such a feature vector incorporates information about the pose and motion dynamics. We assume that there are B gestemes that model all the features in our L classes of gestures. A Gaussian Mixture Model containing B centers was estimated using a set of training feature vectors. All the Gaussians are assumed to have full covariances. Parameters for the B Gaussians are determined from the set of training feature vectors $f_n, 1 \leq n \leq N$, using the EM algorithm to maximize a global likelihood function given below:

$$L = \sum_{n=1}^{N} log \sum_{i=1}^{B} \delta_i p(f_n|i)$$

where $p(f_n|i)$ is a single component of the mixture of Gaussians, and δ_i is the ith component's mixing proportion.

Once this likelihood is maximized, each Gaussian cluster is assigned a unique label. The probabilities of each training feature belonging to each cluster, $p(i|f_n)$ are evaluated to determine the most likely cluster they belong to.

$$L_{f_n} = \arg\max_i p(i|f_n)$$

where L_{f_n} is the best scoring Gaussian's label assigned to training feature f_n. After this classification of the feature vectors into gesteme clusters, a projection matrix, that separates these feature vectors in an optimal way, is estimated by running LDA on these labelled training features.

Suppose there are B labels, then the within class expected covariance is:

$$S_w = \sum_{i=1}^{B} p_i \cdot \Lambda_i$$

where p_i is the prior probability and Λ_i is the covariance of Gaussian i respectively.

The mean of the whole mixture of Gaussians is computed as

$$u_{overall} = \sum_{i=1}^{B} p_i \cdot u_i$$

where u_i is the mean of each Gaussian component.

The between class variance is then computed as

$$S_b = \sum (u_j - u_{overall}) \times (u_j - u_{overall})^T$$

Assuming a class-independent transform, the optimizing criterion, $J(\boldsymbol{w})$ is computed as

$$J(\boldsymbol{w}) = (S_w)^{-1} \times S_b$$

The projection matrix, $\mathbf{\Gamma}$, is found as the eigenvector matrix of $J(\boldsymbol{w})$ and the projected feature vector, $f_n^{'}$ becomes

$$f_n^{'} = \mathbf{\Gamma}^T f_n$$

These projected feature vectors, which are more distinct between classes, and less dispersed within each class, are then used for training the HMM to classify the features by the gesture classes.

Appendix 3.5 will describe the dataset we have used and the experiments conducted for evaluating this method. Our results confirm that discriminative vector quantization is a useful technique. By combining GMM and LDA together, we show that the error rates for the gesture recognition decreased significantly. However, the Hidden Markov Model has the unrealistic assumption that observations are conditionally independent. At the same time, the gestures are trained in separate models; a single discriminative model that is jointly trained on all gesture classes can potentially learn a better partition between the gesture classes. With this insight, we proceed to the next chapter and apply Hidden Conditional Random Fields to allow sharing of the features using hidden states.

3.2 Hidden Conditional Random Fields (HCRFs)

In Section 3.1, shared features were passed into Hidden Markov Models(HMMs) for evaluation. However, these models assume that observations are conditionally independent.

This restriction makes it difficult or impossible to accommodate long-range dependencies among observations or multiple overlapping features of the observations. To overcome this constraint, we turn our attention to hidden conditional random fields (HCRFs).

Conditional random fields use an exponential distribution to model the entire sequence given the observation sequence [12, 24, 49]. This avoids the independence assumption between observations, and allows non-local dependencies between state and observations. A Markov assumption may still be enforced in the state sequence, allowing inference to be performed efficiently using dynamic programming. CRFs assign a label for each observation (e.g., each time point in a sequence), and they neither capture hidden states nor directly provide a way to estimate the conditional probability of a class label for an entire sequence.

We propose a model for gesture recognition which incorporates hidden state variables in a discriminative multi-class random field model, extending previous models for spatial CRFs [67] into the temporal domain[2]. By allowing a classification model with hidden states, no a-priori segmentation into substructures is needed, and labels at individual observations are optimally combined to form a class conditional estimate.

This hidden state conditional random field model can be used either as a gesture class detector, where a single class is discriminatively trained against all other gestures, or as a multi-way gesture classifier, where discriminative models for multiple gestures are simultaneously trained. The latter approach has the potential to share useful hidden state structures across the different classification tasks, allowing higher recognition rates.

3.3 HCRFs: A Review

We will review HCRFs as described in [67]. We wish to learn a mapping of observations x to class labels $y \in \mathcal{Y}$, where x is a vector of m local observations, $\mathbf{x} = \{x_1, x_2, \dots x_m\}$, and each local observation x_j is represented by a feature vector $\phi(x_j) \in \Re^d$.

An HCRF models the conditional probability of a class label given a set of observations

[2]The material in this section was previously published in [85] S. Wang, A. Quattoni, L. Morency, D. Demirdjian and T. Darrell, "Hidden Conditional Random Fields for Gesture Recognition". In *CVPR* 2006, New York City, USA.

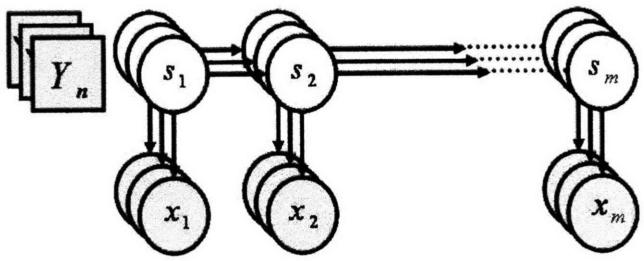


Figure 3-3: HMM model. Suppose there are n gesture classes we need to classify. n HMM models are trained, and during evaluation a test sequence of m observations is passed into each HMM model. The highest scoring model is the interpreted gesture class.

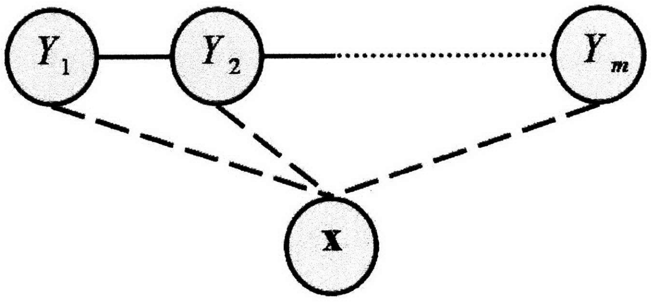


Figure 3-4: CRF Model. A single exponential model is trained on all gesture classes. Given an observation sequence $\mathbf{x}$, which is a vector of m frames, a class label Y_i is assigned per frame. During evaluation, the highest scoring sequence of labels Y where Y is a vector of m labels is assigned.

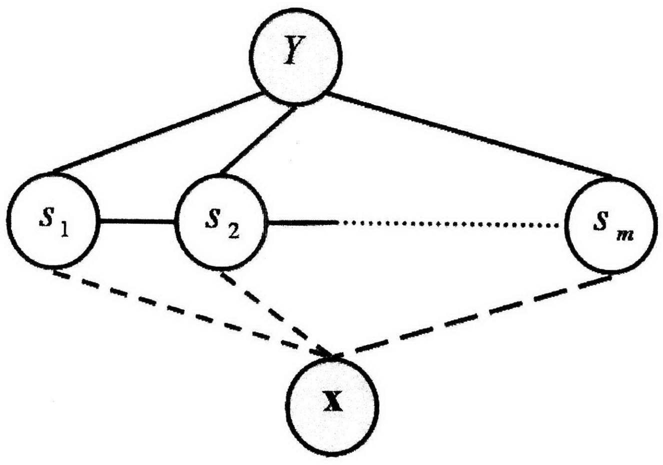


Figure 3-5: HCRF Model. A single exponential model like the Conditional Random Field model, trained on all gesture classes. However, an intermediate level of hidden states is included. During training, the distribution of hidden states is learned per class. During evaluation, a single class label Y is assigned.

by:

$$P(y \mid \mathbf{x}, \theta) = \sum_{\mathbf{s}} P(y, \mathbf{s} \mid \mathbf{x}, \theta) = \frac{\sum_{\mathbf{s}} e^{\Psi(y,\mathbf{s},\mathbf{x};\theta)}}{\sum_{y' \in \mathcal{Y}, \mathbf{s} \in \mathcal{S}^m} e^{\Psi(y',\mathbf{s},\mathbf{x};\theta)}} \tag{3.1}$$

where $\mathbf{s} = \{s_1, s_2, ..., s_m\}$, each $s_i \in S$ captures the unique underlying structure of each class and S is the set of hidden states in the model. If we assume that $\mathbf{s}$ is observed and that there is a single class label y then the conditional probability of $\mathbf{s}$ given $\mathbf{x}$ becomes a regular CRF. The potential function $\Psi(y, \mathbf{s}, \mathbf{x}; \theta) \in \Re$, parameterized by θ, measures the compatibility between a label, a set of observations and a configuration of the hidden states.

Following previous work on CRFs [49, 12], we use the following objective function in training the parameters:

$$L(\theta) = \sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i, \theta) - \frac{1}{2\sigma^2}||\theta||^2 \tag{3.2}$$

where n is the total number of training sequences. The first term in Eq. 3.2 is the log-likelihood of the data; the second term is the log of a Gaussian prior with variance σ^2, i.e., $P(\theta) \sim \exp\left(\frac{1}{2\sigma^2}||\theta||^2\right)$. We use gradient ascent to search for the optimal parameter values, $\theta^* = \arg\max_\theta L(\theta)$. For our experiments we used a Quasi-Newton optimization technique [55].

3.4 HCRFs for Gesture Recognition

HCRFs—discriminative models that contain hidden states—are well-suited to the problem of gesture recognition. Quattoni [67] developed a discriminative hidden state approach where the underlying graphical model captured spatial dependencies between hidden object parts. In this work, we modify the original HCRF approach to model sequences where the underlying graphical model captures temporal dependencies across frames, and to incorporate long range dependencies.

Our goal is to distinguish between different gesture classes. To achieve this goal, we learn a state distribution among the different gesture classes in a discriminative manner. Generative models can require a considerable number of observations for certain gesture classes. In addition, generative models may not learn a shared common structure among gesture classes nor uncover the distinctive configuration that sets one gesture class uniquely against others. For example, the flip-back gesture used in the arm gesture experiments (see Figure 3-6) consists of four parts: 1) lifting one arm up, 2) lifting the other arm up, 3) crossing one arm over the other and 4) returning both arms to their starting position. We could use the fact that when we observe the joints in a particular configuration (see FB illustration in Figure 3-6) we can predict with certainty the flip-back gesture. Therefore, we would expect that this gesture would be easier to learn with a discriminative model. We would also like a model that incorporates long range dependencies (i.e., that the state at time t can depend on observations that happened earlier or later in the sequence.) An HCRF can learn a discriminative state distribution and can be easily extended to incorporate long range dependencies.

To incorporate long range dependencies, we modify the potential function Ψ in Equation 1 to include a window parameter ω that defines the amount of past and future history to be used when predicting the state at time t. Here, $\Psi(y, \mathbf{s}, \mathrm{x}; \theta, \omega) \in \Re$ is defined as a potential function parameterized by θ and ω.

$$\begin{aligned}\Psi(y, \mathbf{s}, \mathbf{x}; \theta, \omega) &= \sum_{j=1}^{n} \varphi(\mathbf{x}, j, \omega) \cdot \theta_s[s_j] + \sum_{j=1}^{n} \theta_y[y, s_j] \\ &+ \sum_{(j,k) \in E} \theta_e[y, s_j, s_k] \end{aligned} \tag{3.3}$$

The graph E is a chain where each node corresponds to a hidden state variable at time t; $\varphi(\mathbf{x}, j, \omega)$ is a vector that can include any feature of the observation sequence for a specific window size ω. (i.e. for window size ω, observations from $t - \omega$ to $t + \omega$ are used to compute the features.)

The parameter vector θ is made up of three components: $\theta = [\theta_e \ \theta_y \ \theta_s]$. We use the notation $\theta_s[s_j]$ to refer to the parameters θ_s that correspond to state $s_j \in S$. Similarly, $\theta_y[y, s_j]$ stands for parameters that correspond to class y and state s_j and $\theta_e[y, s_j, s_k]$ refers to parameters that correspond to class y and the pair of states s_j and s_k.

The inner product $\varphi(\mathbf{x}, j, \omega) \cdot \theta_s[s_j]$ can be interpreted as a measure of the compatibility between the observation sequence and the state at time j at window size ω. Each parameter $\theta_y[y, s_j]$ can be interpreted as a measure of the compatibility between a hidden state k and a gesture y. Finally, each parameter $\theta_e[y, s_j, s_k]$ measures the compatibility between pairs of consecutive states j and k and the gesture y.

Given a new test sequence $\mathbf{x}$, and parameter values θ^* learned from training examples, we will take the label for the sequence to be:

$$\arg\max_{y \in \mathcal{Y}} P(y \mid \mathbf{x}, \omega, \theta^*). \tag{3.4}$$

Since E is a chain, there are exact methods for inference and parameter estimation as both the objective function and its gradient can be written in terms of marginal distributions over the hidden state variables. These distributions can be computed using belief propagation [65].

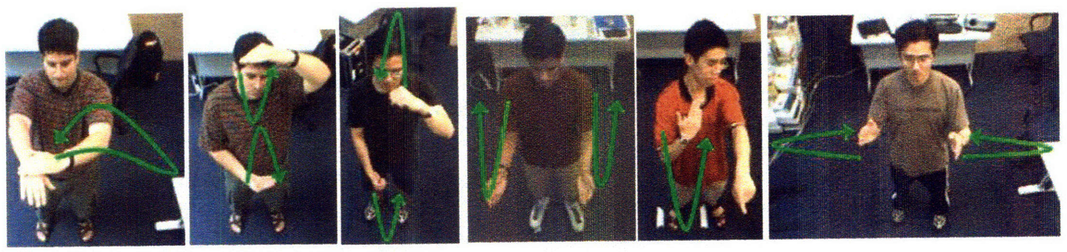


Figure 3-6: Illustrations of the six gesture classes for the experiments. Below each image is the abbreviation for the gesture class. These gesture classes are: FB - Flip Back, SV - Shrink Vertically, EV - Expand Vertically, DB - Double Back, PB - Point and Back, EH - Expand Horizontally. The green arrows are the motion trajectory of the fingertip and the numbers next to the arrows symbolize the order of these arrows. A detailed description of these gestures appears in Section 3.6.

3.5 Gesture recognition using discriminative vector quantization: datasets, experiments and results

In this section, we will describe the datasets, experiments and results using discriminative vector quantization described in Section 3.1.

3.5.1 Arm Gesture Dataset

We labeled eleven gestures for an HMM-based recognizer to classify using discriminative vector quantization described in Section 3.1. Gesture data was collected and gathered from thirteen users. These users were asked to perform the gestures in front of a stereo camera. From each image frame, a 3D cylindrical body model, consisting of a head, torso, arms and forearms was estimated using a stereo-tracking algorithm [25]. The estimated body pose sequences of these gestures were collected.

Half of these samples were used for training the HMM recognizer and for linear discriminant analysis, while the other half was used for testing. Examples of these gestures are shown in Figure 3-7.

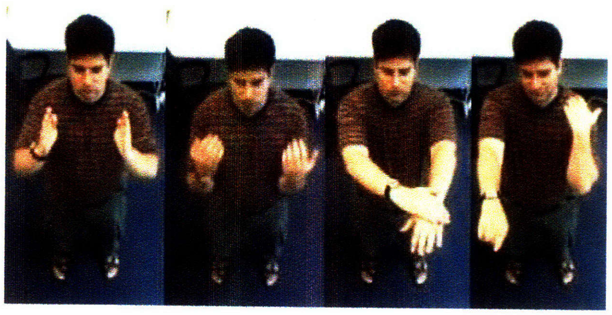


Figure 3-7: Images of a user articulating 4 different gestures.

3.5.2 Experiments and Results

We experimented with different techniques or different types of features, while varying the feature length (the feature length is the number of consecutive poses used to form each feature vector), and our results are plotted in Figure 3-8. In the first technique, we applied LDA to pose features and sent the projected features, $f_n' = \mathbf{\Gamma}^T f_n$ to train the HMM (the plot from this technique is labelled LDA Features in the figure). In the second technique, we simply used pose features, f_n to train the HMM (labelled as pose features in the figure). In the third technique, we used velocity from consecutive poses as features for training the HMM (labelled as velocity features, ν_n). A velocity feature is given as

$$\nu_n = f_n - f_{n-1}$$

In the fourth technique, we simply used both pose and velocity features for training the HMM (labelled as velocity and pose features). A velocity and pose feature is given as

$$\psi_n = \begin{pmatrix} f_n \\ \nu_n \end{pmatrix}$$

In the fifth technique, we applied principal component analysis on the pose features (labelled as PCA). The components that contribute to 98% of the energy were extracted, and the projected pose features were used to train the HMM. In the last technique, we applied kernel principal component analysis (kPCA) on the pose features, and used the projected features to train the HMM. We used a Gaussian kernel for the projection. The Gaussian kernel function $k(x, y)$ is given by:

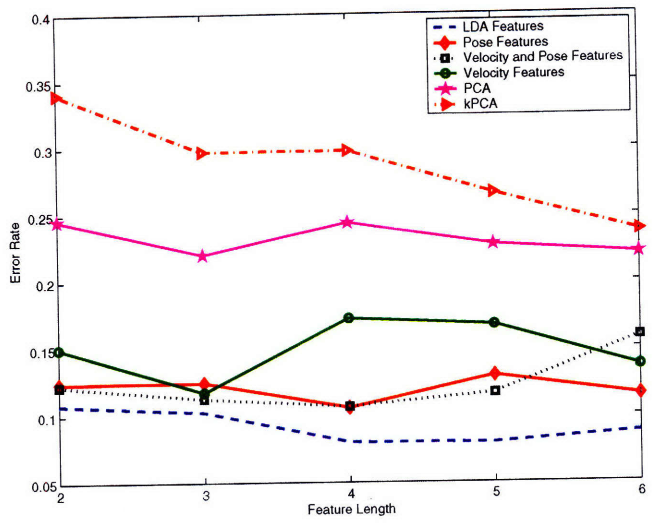


Figure 3-8: Error rates of using different types of features vs their feature length. The graph shows that the LDA-based approach produced lower error rates consistently over different feature lengths.

$$k(x, y) = e^{-\frac{||x-y||^2}{2 \cdot \sigma^2}}$$

and we used a σ value of 10. We did not display results using other kernel functions as they performed significantly worse.

From Figure 3-8, pose features projected into the LDA subspace performed consistently better than the other 5 techniques. The error rate of the kPCA technique did not perform as well as we expected. We are currently investigating this matter.

We trained the HMM 50 times using the various techniques or feature types at a constant feature length of 4, and tabulated the statistics of the error rates. Our results are given in Table 3.1. As a comparison to pose features, the error rate was divided by a factor of 1.5 by applying LDA. The standard deviation of the error rate from our LDA-based approach was also relatively low compared to other techniques, which proves that this approach produced better results consistently.

Our results confirm that sharing features is a useful technique. By combining GMM and LDA together, we show that the error rates for the gesture recognition decreased significantly. With this insight, we proceed to the next section and apply Hidden Conditional Random Fields to allow sharing of the features using hidden states.

	Avg Error Rate, $\mu(\%)$	Std Error Rate, $\sigma(\%)$
Pose Features f_n	12.16	2.05
LDA Features f_n'	7.98	1.68
PCA	26.73	1.84
kPCA	25.09	3.6
Velocity Features ν_n	17.6	13.81
Velocity and Pose Features ψ_n	1.82	2.86

Table 3.1: Statistics of Error Rates of various techniques or feature types using a constant feature length of 4. The standard deviation and the average error rate of the LDA features is smaller than the other feature types. This shows that LDA features will lower the error rate consistently

3.6 Gesture recognition using Hidden Conditional Random Fields: datasets, experiments and results

In this section, we will describe the datasets, experiments and results using HCRFs described in Section 3.2.

3.6.1 Arm Gesture Dataset

We defined six arm gestures for the experiments (see Figure 3-6). In the Expand Horizontally (EH) arm gesture, the user starts with both arms close to the hips, moves both arms laterally apart and retracts back to the resting position. In the Expand Vertically (EV) arm gesture, the arms move vertically apart and return to the resting position. In the Shrink Vertically (SV) gesture, both arms begin from the hips, move vertically together and back to the hips. In the Point and Back (PB) gesture, the user points with one hand and beckons with the other. In the Double Back (DB) gesture, both arms beckon towards the user. Lastly, in the Flip Back (FB) gesture, the user simulates holding a book with one hand while the other hand makes a flipping motion, to mimic flipping the pages of the book.

Users were asked to perform these gestures at a distance of 1.5 meters in front of a stereo camera. We used the Digiclops stereo camera from PointGrey Research Inc to acquire the stereo images. From each image frame, a 3D cylindrical body model, consisting of a head, torso, arms and forearms, was estimated using a stereo-tracking algorithm [25]. Figure 3-9 shows a gesture sequence with the estimated body model superimposed on the user. From

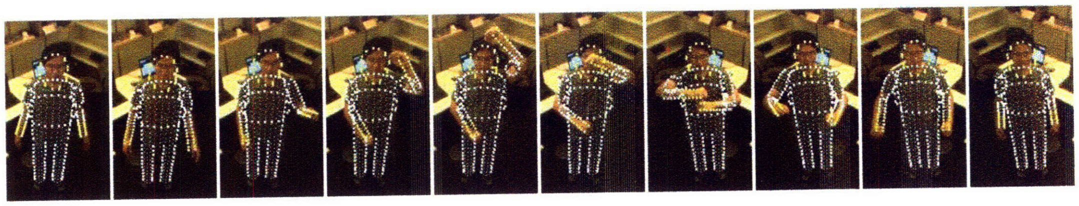


Figure 3-9: Sample image sequence with the estimated body pose superimposed on the user in each frame. The user is articulating the Expand Vertically (EV) gesture.

these body models, both the joint angles and the relative co-ordinates of the joints of the arms are used as observations for our experiments and were manually segmented into six arm gesture classes. Thirteen users were asked to perform these six gestures; an average of 90 gestures per class were collected.

3.6.2 Experiments and Results

We describe the various models used in our experiments before we proceed to discuss the results. Figures 3-3, 3-4 and 3-5 show graphical representations of the HMM model, the CRF model, and the HCRF (multi-class) model used in our experiments.

HMM Model - As a first baseline, we trained a HMM model per class. Each model had four states and used a single Gaussian observation model. During evaluation, test sequences were passed through each of these models, and the model with the highest likelihood was selected as the recognized gesture.

CRF Model - As a second baseline, we trained a single CRF chain model where every gesture class had a corresponding state. In this case, the CRF predicts labels for each frame in a sequence, not the entire sequence. During evaluation, we found the Viterbi path under the CRF model, and assigned the sequence label based on the most frequently occurring gesture label per frame. We ran additional experiments that incorporated different long range dependencies (i.e. using different window sizes ω, as described in Section 4).

HCRF (one-vs-all) Model - For each gesture class, we trained a separate HCRF model to discriminate the gesture class from other classes. Each HCRF was trained using six hidden states. For a given test sequence, we compared the probabilities for each single

Models	Accuracy (%)
HMM $\omega = 0$	84.22
CRF $\omega = 0$	86.03
CRF $\omega = 1$	81.75
HCRF (one-vs-all) $\omega = 0$	87.49
HCRF (multi-class) $\omega = 0$	91.64
HCRF (multi-class) $\omega = 1$	93.81

Table 3.2: Comparisons of recognition performance (percentage accuracy) for body poses estimated from image sequences.

HCRF, and the highest scoring HCRF model is selected as the recognized gesture.

HCRF (multi-class) Model - We trained a single HCRF using twelve hidden states. Test sequences were run with this model and the gesture class with the highest probability was selected as the recognized gesture. We also conducted experiments that incorporated different long range dependencies in the same way as described in the CRF experiments.

For the HMM model, the number of Gaussian mixtures and states were set by minimizing the error on training data, and for hidden state models the number of hidden states was set in a similar fashion.

For the training process, the CRF models for the arm and head gesture dataset took about 200 iterations to train. The HCRF models for the arm and head gesture dataset required 300 and 400 iterations for training respectively.

Table 3.2 summarizes results for the arm gesture recognition experiments. In these experiments the CRF performed better than HMMs at window size zero. At window size one, however, the CRF performance was poorer; this may be due to overfitting when training the CRF model parameters. Both multi-class and one-vs-all HCRFs perform better than HMMs and CRFs. The most significant improvement in performance was obtained when we used a multi-class HCRF, suggesting that it is important to jointly learn the best discriminative structure.

Figure 3-10 shows the distribution of states for different gesture classes learned by the best performing model (multi-class HCRF) in the arm gesture experiment. This graph was obtained by computing the Viterbi path for each sequence (i.e. the most likely assignment for the hidden state variables) and counting the number of times that a given state occurred

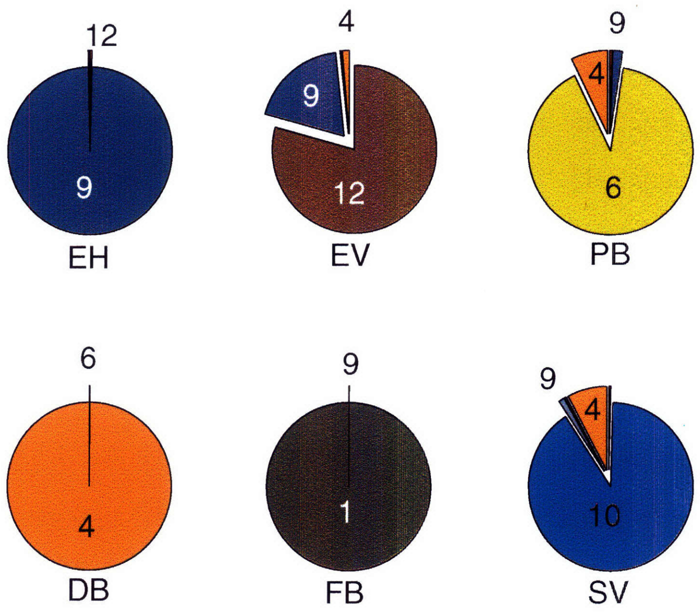


Figure 3-10: Graph showing the distribution of the hidden states for each gesture class. The numbers in each pie represent the hidden state label, and the area enclosed by the number represents the proportion.

Models	Accuracy (%)
HCRF $\omega = 0$	86.44
HCRF $\omega = 1$	96.81
HCRF $\omega = 2$	97.75

Table 3.3: Experiment on 3 arm gesture classes using the multi-class HCRF with different window sizes. The 3 different gesture classes are: EV-Expand Vertically, SV-Shrink Vertically and FB-Flip Back. The gesture recognition accuracy increases as more long range dependencies are incorporated.

among those sequences. As we can see, the model has found a unique distribution of hidden states for each gesture, and there is a significant amount of state sharing among different gesture classes. The state assignment for each image frame of various gesture classes is illustrated in Figure 3-11. Here, we see that body poses that are visually more unique for a gesture class are assigned very distinct hidden states, while body poses common between different gesture classes are assigned the same states. For example, frames of the FB gesture are uniquely assigned a state of one while the SV and DB gesture class have visibly similar frames that share the hidden state four.

The arm gesture results with varying window sizes are shown in Table 3.3. From these results, it is clear that incorporating some amount of contextual dependency is important, since the HCRF performance improved with increasing window size.

In summary, we presented a discriminative hidden-state approach for gesture recognition. Our model combined the two main advantages of previous approaches to gesture recognition: the ability of CRFs to use long range dependencies, and the ability of HMMs to model latent structure. By regarding the sequence label as a random variable we can train a single joint model for all the gestures and share hidden states between them. Our results have shown that HCRFs outperform both CRFs and HMMs for certain gesture recognition tasks. For arm gestures, the multi-class HCRF model outperformed HMMs and CRFs even when long range dependencies were not used, demonstrating the advantages of joint discriminative learning.

With this success in arm and head gestures, we proceed to apply HCRFs to the task communication error detection. For communication error detection, the features used are different. Unlike the arm and head gestures described in this chapter, the head motion and

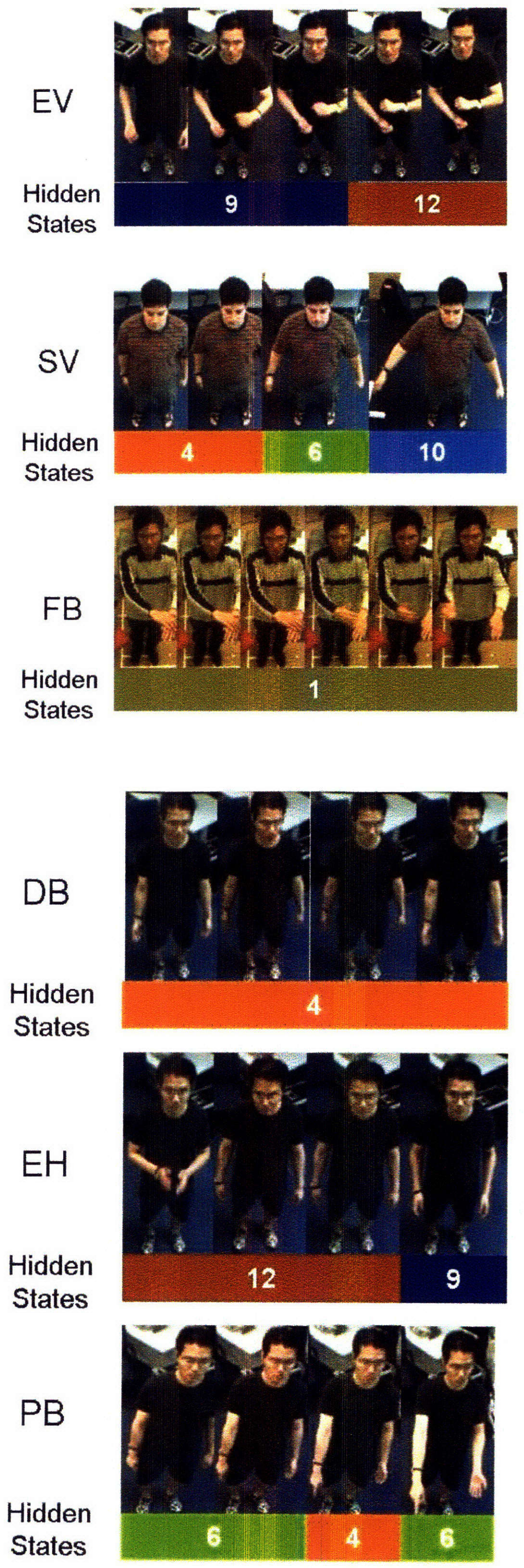


Figure 3-11: Articulation of the six gesture classes. The first few consecutive frames of each gesture class are displayed. Below each frame is the corresponding hidden state assigned by the multi-class HCRF model.

facial features were estimated from monocular images. We describe the estimation of these features in the next chapter.

Chapter 4

Automatic Recognition of visual communication errors: initial results

In this chapter, we will describe our approach to detect visual communication errors and the resources used for extracting the facial motion features and head motion features.

4.1 Approach

As described in the introduction, we propose detecting communication errors visually while the speaker is listening to the system's reply. An approach overview is illustrated in Figure 4-1. The speaker is instructed to achieve a task with the system, e.g. an air/train ticket reservation, and video footage of the speaker's face and head is captured throughout the conversation. During the interaction, video footage of the speaker is segmented per conversation turn. Each segment's boundaries are illustrated in Figure 4-1, where the segment of interest starts from time t_1 and ends at time t_2. In this segment of interest, the system is giving its reply, while the speaker is listening. For each image frame in this video segment, facial motion features and head pose features are estimated. These visual features are subsequently evaluated by various gesture recognition models, described in Chapter **??**, trained for communication error detection.

To train the gesture recognition models, the visual features are manually labeled by a human coder. From the related work section, the reader can see that building a robust

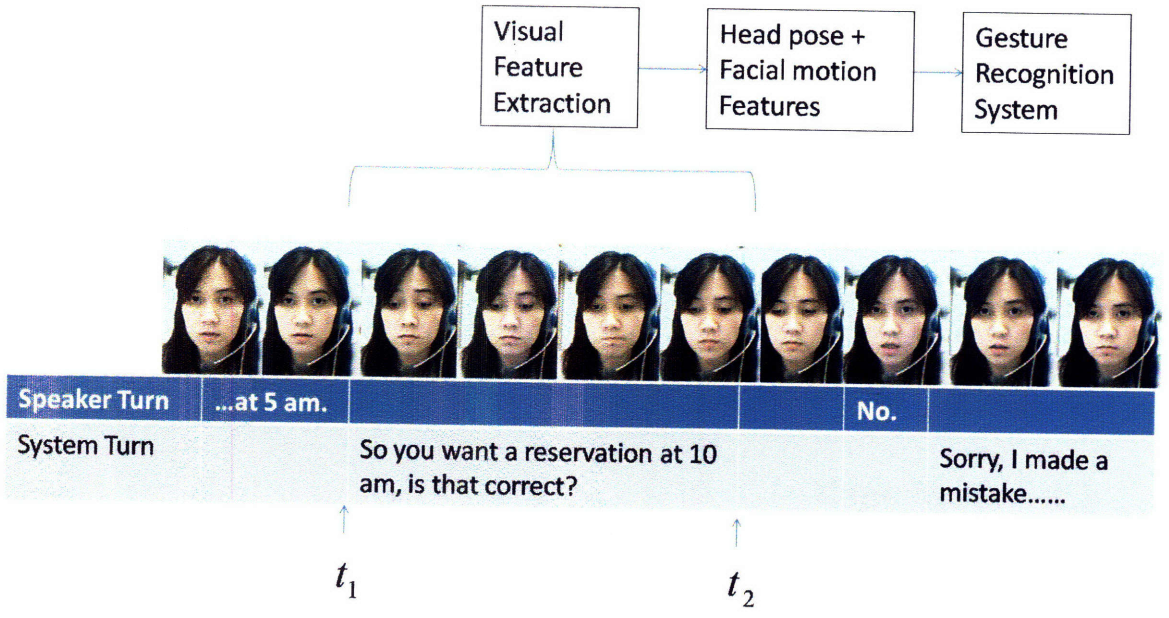

Visual Feature Extraction
Head pose + Facial motion Features
Gesture Recognition System
Speaker Turn
...at 5 am.
No.
System Turn
So you want a reservation at 10 am, is that correct?
Sorry, I made a mistake......
t_1
t_2

Figure 4-1: Diagram illustrating a speaker encountering a communication error. Notice that during the system's conversation turn (from time $t1$ to time $t2$), where it is giving an erroneous reply, the speaker raised her eye brows and spouted to signal an error. We will use this video segment during the system's conversational turn to detect communication errors.

machine learning system for recognition of fine grained affective states is still in its preliminary stages. This is due to the need to have several human coders to label the data, but human coders themselves have considerable labeling variation. In our approach, we avoid this ambiguity in labeling by only defining two coarse states: communication errors and non errors. If the system's reply is not relevant to answering the spoken content in the preceding speaker's turn, the video footage of the speaker during the system's reply is labeled as a communication error. This labeling framework can be verified by the transcript of the conversations and video footage, and our approach avoids subjective labeling of the fine expression variation among multiple human coders.

During the system's turn of the conversation, several different replies can occur. In a reservation system, the system can request more information, verify the last reservation, confirm the reservation, or acknowledge its mistake. When the system is enquiring for more information, we do not expect the speaker's visual expressions to be indicative of an error. Perceptual studies by Barkhuysen et al. [5] have found visual footage to be indicative of communication errors during verification replies. For this thesis, the video footage captured in the following types of replies will be used for evaluation:

- Verification: when a system is trying to verify its interpretation. For example, the system can say

  "So you want to travel from Boston to San Francisco on Monday, is that correct?"

  If the system is incorrect, the video segment during this reply will be labeled as a verification error.

- Confirmation: when the system assumes it has made a correct interpretation. The system can say

  "Your reservation is now confirmed. Goodbye."

  If the system is incorrect, the video segment occurring during this reply will be labeled as a confirmation error.

- Apology: when the system is assumes it had made a mistake and apologizes for it. The system can say

"I am sorry I made a mistake. Let's start over. Which city do you want to travel from?"

If the system is incorrect, the video segment occurring during this reply will be labeled as an apology error.

All apology, confirmation, and verification errors are communication errors.

Once the error is detected, the erroneous reply could be followed by a corrective response. The system could say "Sorry I realize I made a mistake, do you want to start over?" etc. Many corrective strategies exist and this thesis will not explore and examine the different correction strategies. Our focus is on developing a framework for communication error detection using visual cues extracted during the system's turn.

In our framework, head motion is decomposed into two distinct components. The first component consists of the 3D rigid motion, δ, of the head. The second component consists of the local motion, V_f, generated by the non-rigid parts of the face (e.g. mouth, lips, eyes).

4.2 Visual and Audio Feature Extraction

In this section, we will describe the various audio and visual features extracted for our experiments on communication error detection. We rely on the implementation of [56] and [82] to extract the visual features, as described below. Two types of features are extracted, namely facial feature motion and head motion.

4.2.1 Head Motion and Pose Estimation

Head motion is estimated using a 3D model-based tracking algorithm. In our approach, the head of the user is modeled using a 3D mesh, which consists of a set of 3D points $M_i = (X_i, Y_i, Z_i)$.

Following [56], let $\mathbf{R}$ and $\boldsymbol{t}$ be the rotation and translation of the head between time t and $t+1$. In case of small motions, the rotation $\mathbf{R}$ can be parameterized by:

$$\mathbf{R} = \mathbf{I} + \begin{pmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{pmatrix} \tag{4.1}$$

The rigid transformation can then be parameterized by a 6-vector $\delta = (w_x, w_y, w_z, \boldsymbol{t})^\top$. Given the location and orientation of the model (*i.e.* location of the points M_i) at time t, the rigid motion δ of the head is estimated as follows:

Let M_i' be the 3D location of point M_i at time $t+1$. Let m_i and m_i' be the respective projections of M_i and M_i' in the image. Let f and (u_0, v_0) be the focal length and principal point location of the camera respectively. Using the camera as the reference, we can write:

$$m_i = \frac{1}{Z_i}\mathbf{P}M_i \qquad m_i' = \frac{1}{Z_i'}\mathbf{P}M_i' \tag{4.2}$$

where $\mathbf{P}$ is the camera-projection matrix defined as

$$\mathbf{P} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \end{pmatrix} \tag{4.3}$$

If the face moves rigidly then all points M_i move according to the same rigid transformation, *i.e.* $M'_i = \mathbf{R}M_i + \boldsymbol{t}$. Let $\vec{u}_i = m'_i - m_i$ be the optical flow estimated at point m_i. By combining eq. (4.2) and (4.1) for all i's, δ can be found as the solution of a linear system

$$\mathbf{A}\delta = \boldsymbol{b} \tag{4.4}$$

where $\mathbf{A}$ is a matrix and $\boldsymbol{b}$ a vector, whose entries depend on f, (u_0, v_0), M_i and $\vec{u}_i$ (for all i's).

If the head performs a perfectly rigid motion (*i.e.* all points M_i perform the same rigid transformation), eq. (4.4) can be solved exactly using a standard linear least-squares technique. However, in order to account for outliers, i.e., points M_i corresponding to non-rigid parts of the face (e.g., eyes, mouth), a robust estimation algorithm needs to be used. Because of its simplicity and performance, we chose to employ the RANSAC [39] algorithm. The RANSAC algorithm is able to find the dominant rigid motion of the face even if half the points M_i perform some 'outlier' motions.

Our face tracking algorithm is initialized with the Viola-Jones face detector [82]. In order to compensate for drift, we implemented our face motion and pose estimation algorithm in a keyframe-based estimation framework similar to [56]. The size of our head motion features is six dimensions.

4.2.2 Facial Motion Features

Let δ be the dominant rigid motion of the face. The facial motion features are defined as the head motion-compensated optical flow, i.e., the optical flow between the images I_t and I_{t+1} from which the motion δ has been 'subtracted'. The facial motion features correspond to the local non-rigid motion generated by the muscles of the face only (e.g., lips, jaw, eyes) independently from the global head motion.

In order to estimate the motion-compensated optical flow, we proceed as follows:

Following [86], let $N_i = (X_i^N, Y_i^N, Z_i^N)$ be the 3D coordinates of point M_i undergoing the exact rigid transformation δ, *i.e.* $N_i = \mathbf{R}M_i + \boldsymbol{t}$. Let $n_i = \frac{1}{Z_i^N}\mathbf{P}N_i$ be the projection

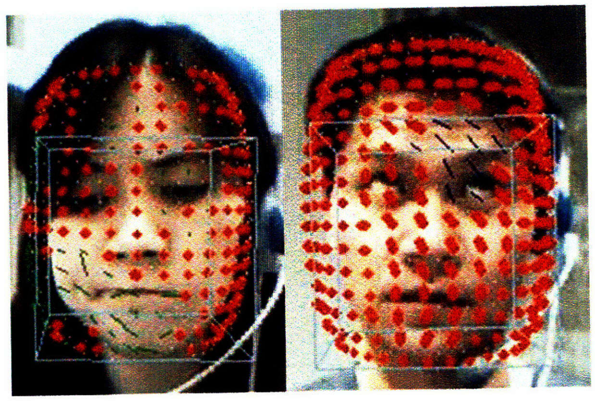


Figure 4-2: Our visual feature motion estimation algorithm consists of estimating the rigid motion component and compensated optical flow on the face surface. In the two images above, the blue box represents the estimated rigid head motion. The estimated optical flow points are represented as red points on the face. The blue or green streaks in certain regions of their faces represent significant flow motion. The regions where the blue or green streaks occurred are also indicative of a communication error.

of N_i in the image. We defined the motion-compensated optical flow $\vec{u}_i^0$ as the discrepancy between the rigid motion-induced flow $\vec{v}_i = n_i - m_i$ and the measured optical flow $\vec{u}_i$.

$$\vec{u}_i^0 = n_i - m_i - \vec{u}_i = \frac{1}{Z_i^N}\mathbf{P}(\mathbf{R}N_i + \boldsymbol{t}) - m_i - \vec{u}_i$$

In our framework, the facial motion features are defined as:

$$V_f = (\vec{u}_1^0, \ldots, \vec{u}_N^0)$$

In each given image, there are 144 motion-compensated optical flow points (in a square 12 x 12 grid) estimated. As a result, the size of the facial motion features is 288 dimensions.

For each video segment of interest, a sequence of facial motion and head motion features are extracted. For each given image frame, the combined facial motion and head motion features created a feature size of 294 dimensions (288 from facial motion features and six from head motion features). We used principal component analysis(PCA) to reduce this high dimensionality and projected the features to a new 10-dimensional subspace. This 10 dimensional subspace was selected as it constituted 90% of the cumulative energy of the eigenvectors.

The sequence of projected features is passed into the various models described in Chapters **??** for training and evaluation. In our initial experiments, we also used audio features, which we describe in the next section.

4.2.3 Audio Feature Extraction

We use three kinds of audio features that were used by [94]: the intensity E, the pitch, F_0, and the first formant frequency, F_1. The audio feature vector A_f is then defined as:

$$A_f = (E, F_0, F_1) \tag{4.5}$$

These features are computed at every 10 msecs using the speech analysis software, PRAAT [10]. The intensity E is computed as: $E = log(\sum_{i=1}^{N}(x[i] - \bar{x})^2)$ where N is the window length and $x[i]$ is the ith sample in that window and $\bar{x}$ is the local average signal value. In our computation (and for the rest of this section) we used a window length of 40 samples. The pitch F_0 is estimated as the reciprocal of the fundamental period as described in [11]. In our experiments, we set the search range of the pitch to be 75 - 1000 Hz. As for the computation of the first formant frequency, F_1, a segment of N samples is extracted for every time step of 1 msec. This segment is multiplied by a Gaussian-like window and the LPC coefficients are computed. This first formant is then extracted using these coefficients by the Burg algorithm described in [17].

In previous work [71] syllable rate was used as an audio feature and the dataset consisted of subjects speaking. However, in our work, our audio data consists of spoken as well as non-spoken words, e.g. exclamations, gasps or humming, which we want to model for automatic problem detection, and our speech recognizer had a lot of difficulty computing an accurate syllable rate. Of the 219 utterances processed by the speech recognizer, 97 utterances have an incorrect number of hypothesized vowel phones. On average, these incorrectly recognized utterances have 2.73 syllables more than the hypothesized ones.

The visual features were extracted during the system's turn, while the audio features were extracted during the speaker's turn. These estimated visual and audio features are subsequently passed into the HCRF model. At the end of the conversation turn, the output scores from the HCRF model in individual streams are combined together by late fusion, which was described in subsection 4.4.1.

We describe the datasets collected and experiments in the next chapter.

4.3 Communication error detection using Hidden Conditional Random Fields

In this chapter, we describe experiments that use Hidden Conditional Random Fields with features in both audio and visual streams to recognize visual communication error detection. Each stream used a Hidden Conditional Random Field as a model for communication error detection. We adopted the approach described previously in Section 4.1, where video footage of the speaker was segmented automatically per conversation turn. For the experiments in this Chapter, the audio stream was segmented in the same fashion. The head motion and facial motion features described in Section 4.2 were extracted from the video segments and passed into a model e.g. an HCRF, for evaluation. We proceed to describe the multimodal fusion model, the audio features and the hidden camera dataset in the next section.

4.4 Hidden camera dataset: experiments and results

For this experiment, we selected HMMs as a baseline model for comparison against the HCRFs. Both models were used to detect communication errors on the individual visual and audio streams. The output scores from these models are subsequently used for multimodal fusion.

4.4.1 Multimodal Fusion Strategies

We combined the results of audio and visual classification within the same conversation turn. We used two common late-fusion strategies as described in [48].

Let the feature input to the j-th classifier, $j = 1, ..., R$ be x_j, and the winning label be h. A uniform prior across all classes was assumed and we used four different combination strategies.

PRODUCT rule: $h = \arg\max_k \prod_{j=1}^{R} P(w_k|x_j)$.

We multiplied the probabilities of the visual feature classifier and the audio feature classifier, and picked the winning class based on the highest scoring multiplication.

SUM rule: $h = \arg\max_{k} \sum_{j=1}^{R} P(w_k|x_j)$.

We added the probabilities of the visual feature classifier and the audio feature classifier, and picked the winning class based on the highest scoring sum.

4.4.2 Audio and Visual Features

For this hidden camera dataset, the visual features were extracted during the system's turn of the conversation. Details of these features were described in Section 4.2. The audio features (also described in Section 4.2) were extracted when the subject was speaking to the system. Note that for a given system's turn, the visual features were extracted, and in the *following speaker's turn*, the audio features were extracted.

4.4.3 Hidden Camera Dataset

We collected an audio-visual database where the facial expressions and the audio cues would correspond to the actual conversational state of the subject. There were several design issues we had to consider to minimize bias of our data collection experiment. These issues were approached in a similar fashion as the database collected for natural facial expressions [71]. First, the subjects cannot know that they are being tested for their communication state. Such knowledge can influence their communication state and invalidate the results. Second, the presence of anyone in the same room as the subject conducting the experiment could bias the subjects behavior (e.g. more inhibited facial expressions or more politely spoken words). Subjects have to conduct the experiment alone.

We set up a conversational kiosk with a hidden camera and two microphones (one hidden, one visible). This conversational kiosk contained a web-based restaurant query speech interface that was developed by Gruenstein et al [34]. This interface consisted of a display showing a Google map with restaurant icons as shown in Figure 4-5. Users communicated with this speech interface via a click-to-talk methodology, i.e. they clicked an icon on the web interface whenever they wanted to speak. Figure 4-4 illustrates a subject sitting in front of the interface. Subjects were told they were supposed to test this system and tell us what they liked or disliked about it at the end of the experiment. They were given a list of restaurants to query for information. To make the queries more challenging, we instructed them to ask for information about French restaurants (the names of which are difficult to pronounce for a non-native speaker) in the area. The subjects had to make the queries in sequential order, and repeat the query in any way they wished (e.g. repeating the same question, or using a different phrase) when the system did not respond correctly. They could only proceed to the next query when the system displayed the correct restaurant information on the display. The purpose of this sequential query was to create a need to solve the communication problem when it occurs. Subjects were also instructed that they could only leave the room once the system successfully answered all their queries.

Here is a sample transcript:

```
 Computer:  "Welcome to the web browser, I can find information
about the restaurants for you.  What can I do for you?"

Speaker:  "Show me restaurants in Cambridge."

Computer:  "Here are the restaurants in Cambridge."(A Google
map of the restaurants is displayed.  A similar snapshot of
this map is exhibited in Figure 4-5.)
```

Apart from the microphone array used by the speech interface, audio and video of the user were recorded throughout the whole experiment. The video was recorded at 15 Hz and the audio at 44kHz. The second microphone was hidden, to disturb the experiment as little as possible while detecting any paralinguistic cues or speech that indicate communication problems in a private setting. At the end of the experiment, we procured agreement for the use of the audio-visual footage from the subjects for our analysis and experiments. A total of 6 subjects performed the study. Figure 4-3 shows two sample images captured from a subject. On average we collected 40 error segments and 23 non-error segments per subject. 90% of this data was used to train the model while the remaining 10% was used for evaluation.

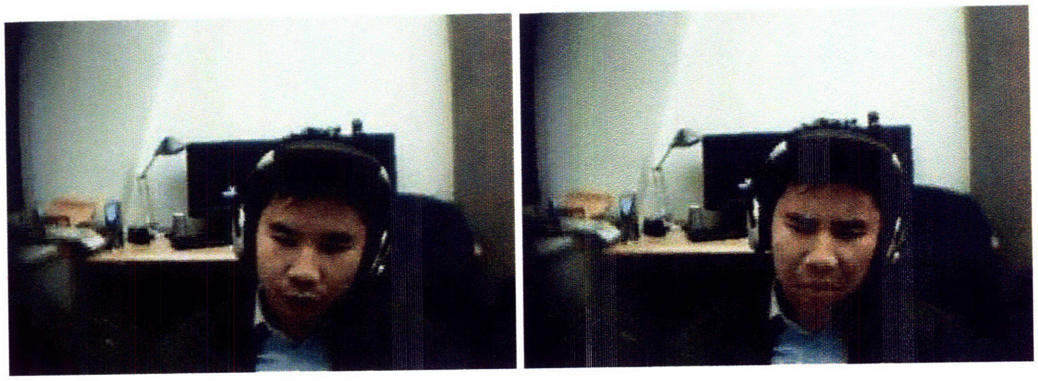


Figure 4-3: Sample images of a subject in the hidden camera dataset. The first image is an example of the subject in a non-error state, and the second image is an example of an error-state. Note that in this setup, the subject has no knowledge of the hidden camera capturing his footage.

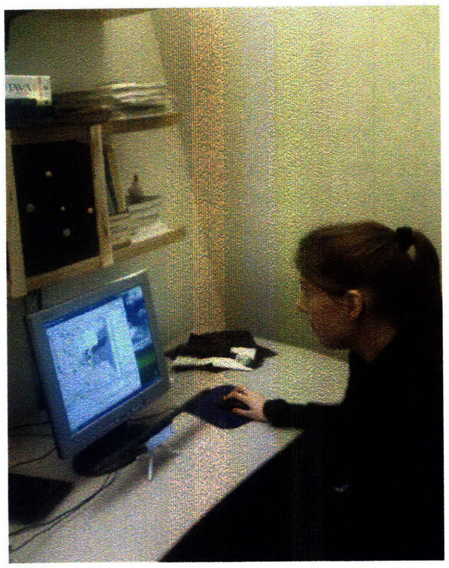
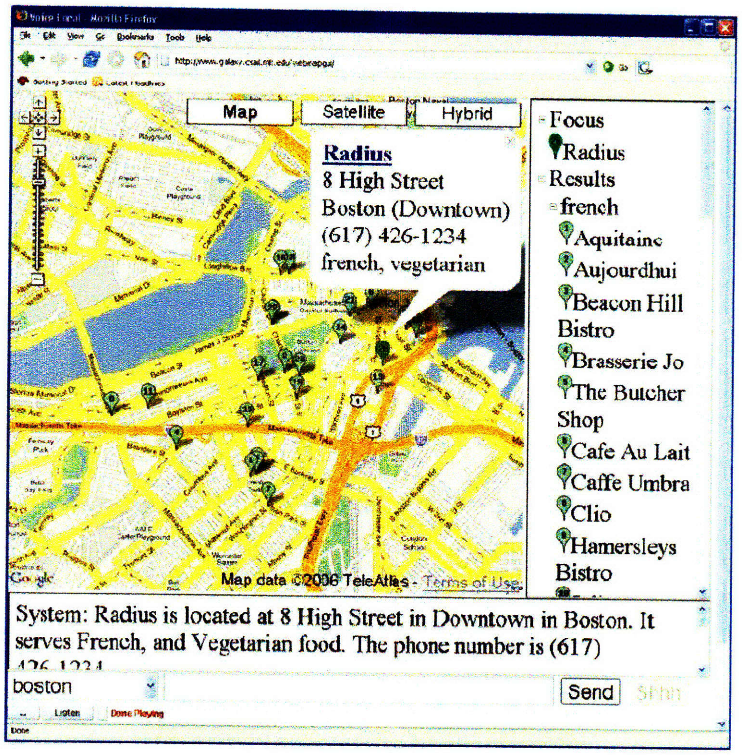


Figure 4-4: Photo of a subject testing the system in the hidden camera dataset. The subject communicates with the system via the microphone array in front of the computer screen. Note that the camera and the second microphone recording the user are not visible in this picture.



Radius
8 High Street
Boston (Downtown)
(617) 426-1234
french, vegetarian

System: Radius is located at 8 High Street in Downtown in Boston. It serves French, and Vegetarian food. The phone number is (617) 426-1234

Figure 4-5: Photo of the restaurant query system. Subjects clicked on a button whenever they wanted to speak to the system. The reply by the system can be heard aurally or visually on the interface.

4.4.4 Visual Features Classification

Facial motion features, V_f, described in Section 4.2 are used as observations for training and testing. We plot the Receiver Operating Characteristics (ROC) curves to display the detection results, using the methodology described in [31]. A detailed explanation on plotting ROC curves for our experiments can be found in Appendix **??**. Figure 4-6 shows the results of the classifiers described in Section **??** using visual features. From this figure, HCRF performs better than HMMs for visual feature classification.

4.4.5 Audio Features Classification

Using audio features, A_f, from Section 4.2 as observations, we trained two classification models, the HMM and the HCRF model. Figure 4-6 shows the ROC curves for the different classifiers. From this figure, both HCRFs and HMMs perform poorly for audio feature classification. We think this is due to two reasons: a noisy environment during data collection (subjects were speaking to a microphone array) and the use of only three acoustic characteristics as our audio features which shows that such features are not very indicative of communication problems.

4.4.6 Audio-Visual Classification

We compared the performance of HMMs and HCRFs in late fusion experiments. Figure 4-7 shows the ROC curve of the combining the various classifiers using the SUM and PRODUCT rule described in Section 4.4.1. Most of the fusion cases perform slightly poorer than those using the single visual classification stream. This poorer fusion performance may be due to the poor audio classification. However, using the SUM rule to combine the HCRF classifiers produced a better result.

4.4.7 ROC curves in this dataset

Certain results shown in this section were previously reported in [86]; note that in [86] convex hull smoothing of ROC plots was not performed (and thus the resulting ROC curves

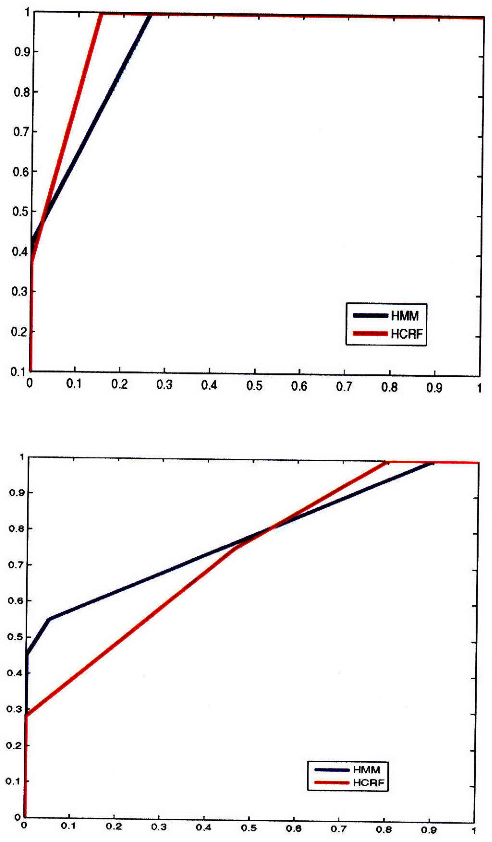
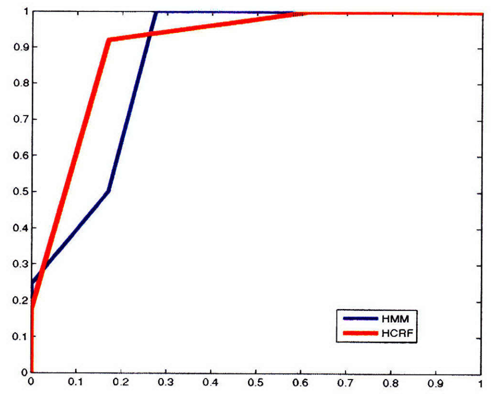


Figure 4-6: (top) ROC curves showing the performance of the different classifiers of visual features. (bottom) ROC curves showing the performance of the different classifiers using audio features. From this figure, HMMS and HCRFs do not perform well on the audio features, while HCRFs perform much better than HMMs on visual features.



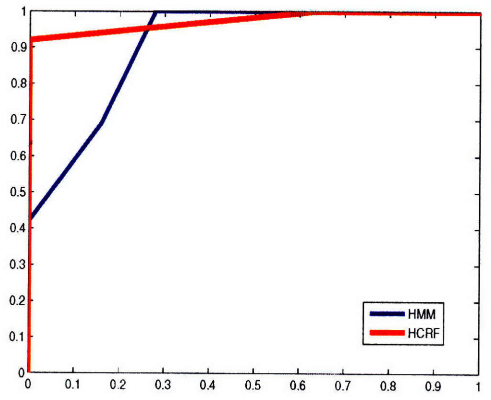


Figure 4-7: (top) ROC curve showing the performance of different classifiers using PRODUCT rule for fusion. (bottom) ROC curve showing the performance of different classifiers using SUM rule for fusion. Both curves show that in most fusion cases, there is a lightly poorer performance in fusion. However, late fusion by using the SUM rule to combine the HCRF classifiers in audio and visual stream improved the performance significantly.

included potentially suboptimal operating points), while in this Chapter we show primary results using convex-hull ROC plots. See Appendix A for details. (Note that certain baseline plots in this chapter are plotted without convex hull smoothing, as described below.)

In these experiments, we evaluated different classification and fusion methods for detecting communication errors in a conversational system. Despite the poor performance of the audio features, the SUM fusion strategy of our HCRF classifiers have improved error detection. In the dataset however, since the subjects were participating in the experiment on a voluntary basis, we question if the facial expressions were truly representative of realistic scenarios. From an interface design point of view, the camera should not be concealed from the subject. We collected another dataset with a different setup to address this issue.

4.5 Motivation dataset : dataset, experiments and results

We collected a new dataset, which we will describe as the "motivation" dataset from on. For this dataset, we only used HMMs as a model for error detection. We want to investigate how effective an automated approach is in detecting communication errors in an unconstrained, natural setting. To do so, we isolate each subject in a room and keep the camera capturing the video footage as unobtrusive as possible. A web-based restaurant query application, which the subjects have to communicate with to retrieve information, is set up on a laptop. This is the same application that was described in the hidden camera dataset. Subjects have to click the "click to talk" button in order to speak to the system. A standard automatic speech recognition system is used.

For each subject, a practice session is conducted before two experiments. In the practice session, the subject is given a walk-through of the system and a guided session by a human expert. In the first experiment, the subject is instructed to perform a series of tasks querying for information about local restaurants. There is no time limit on completing the tasks. In the second experiment, the subject has to perform the same task as in the first experiment. However, to mirror the amount of motivation present in real-life situations, he/she is further informed that there will be a significantly higher monetary reward if the tasks are completed within a time limit. To induce more communication errors, the subject has to query for restaurants that are problematic to pronounce e.g. restaurants with French names etc. On average, subjects completed the first experiment in 15 minutes. For the second experiment, a ten minute time limit for completing the tasks is imposed. Data from a total of seventeen subjects is collected.

This natural dataset is potentially challenging to analyze in many ways. For instance, subjects express themselves at different times during the system's response. Some subjects express their frustration at the end of the system's reply and some express it in the middle of the reply. Most of them are completely unexpressive regardless of the state of the interaction. In addition, many subjects do not stay in full view of the camera as it is not natural

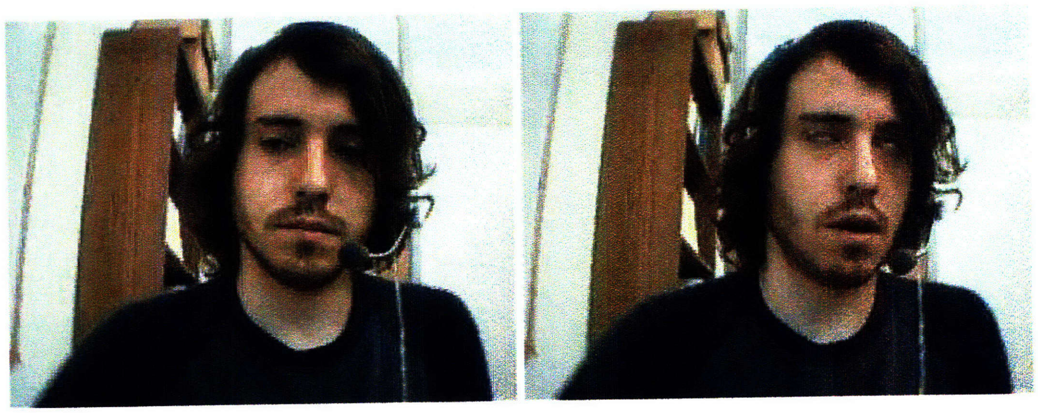


Figure 4-8: Sample images of a subject in the motivation dataset. The first image is an example of the subject in a non-error state, and the second image is an example of an error-state. Note that in this setup, the camera is visible to the subject.

for them to do so when they are not highly conscious of the presence of the camera. Due to unforseen problems in data collection, there are many video sequences where more than 50% of the face is occluded most of the time. These are excluded from our analysis. As a result, thirteen subjects' video footage is used for our experiments. On average, 20 error segments and 15 non-error segments were collected per subject.

Figure 4-8 shows two sample images of a subject interacting with the system.

4.5.1 Experiments and Results

On the motivation dataset, we conduct a six-fold cross validation with randomly generated partitions, making sure our Hidden Markov models are not trained on any data coming from a subject in the test set. In the same manner as described in the experiments on the synthetic dataset, we compute and plot the averaged ROC curve with error bars for communication error detection. This is shown in Figure 4-9. From this figure, however, the detection accuracy is only slightly above chance but is statistically significant (a paired t-test produced $p = 0.013$). To compare against human performance, we conducted a human perception experiment, where video segments evaluated by the HMMs are also evaluated by four human observers. The results of the four human observers are denoted as four green circles in Figure 4-9. The difference in detection accuracy between humans and HMMs is not statistically significant after performing a paired t-test ($p = 0.36$). This

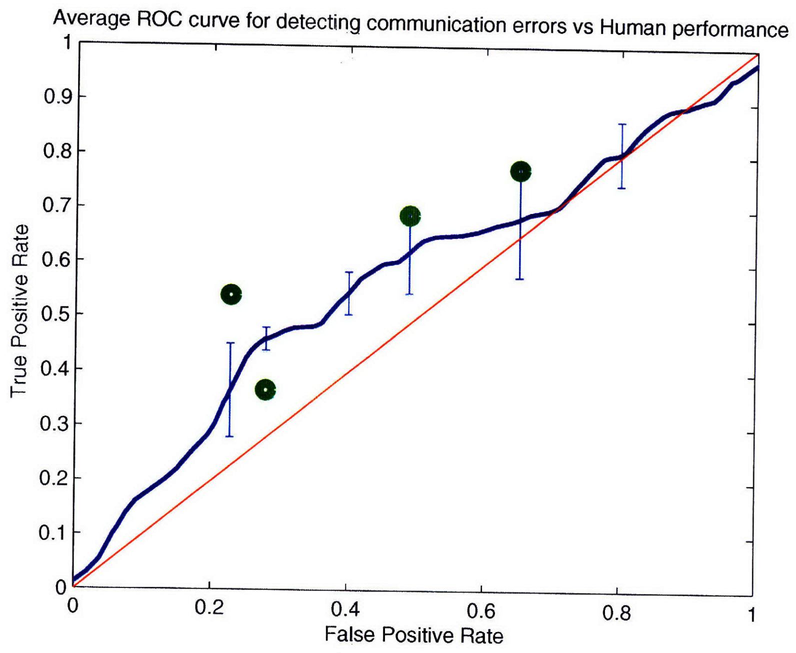


Figure 4-9: Average ROC curve and error bars for detecting communication errors in a natural environment. The green circles represent the performance of four human observers.

shows that the detection accuracy by HMMs is indistinguishable from human performance. Similar performances by humans and machines may indicate that the challenging nature of this dataset outweighs the choice of the visual features or the choice of the classifier. The results also indicate that a motivated setting actually reduces facial expressibility. Note that we actually conducted HCRF experiments as well, but they were not significantly better than HMMs.

We also experimented with a simple fusion of the confidence scores provided by the speech recognizer from [34]in the same conversation turn (i.e. we used the confidence scores that were evaluated by the speech recognizer at the end of the speaker's turn and we used the visual features in the following system's turn). We used the SUM rule described in Section 4.4.1 to combine the speech confidence scores with the HMM visual feature classifier. We computed the average area under the ROC curve for the individual classification and for the fusion classification. For the individual visual and audio streams, the areas were 0.58 and 0.70 respectively, while the area using fusion was 0.71. Our results show that when audio is available, it is more reliable.

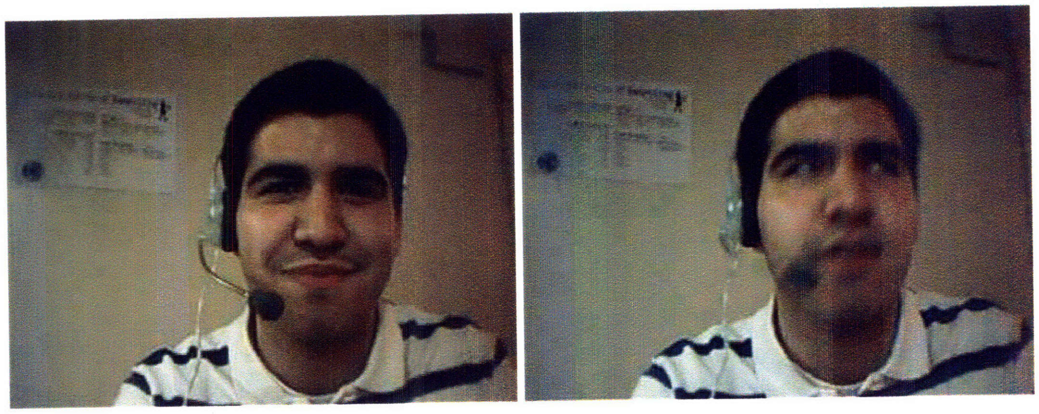


Figure 4-10: Sample images of a subject in the posed expression dataset. The left image is an example of the subject in a non-error state, and the right image is an example of an error-state.

4.6 Posed Expressions : dataset, experiments and results

In this dataset, each subject is asked to pose in front of a camera and demonstrate non-verbal facial expressions that would occur when he/she is listening to the response of a conversational system. For communication errors, the subject presumes the system is giving an incorrect reply. Subjects posed facial expressions depicting varying levels or frustration or confusion. For non-communication errors or neutral situations, each subject presumes the system is giving a correct reply. To vary the intensity of the expressions, scenarios containing different numbers of correct system replies are presented before the subject is asked to articulate his/her facial expression.

Eight posed facial expressions, each lasting on average three seconds, are collected per subject. Data from a total of ten subjects is collected and the facial motion and head pose features are extracted for our experiments. Figure 4-10 shows two sample images collected in this dataset.

4.6.1 Experiments and Results

On this posed expression dataset, the facial motion and head pose features were passed into a HMM classifier for evaluation. We used leave-one-out cross validation (nine subjects' dataset were used for training and the remaining one was used for testing), with randomly

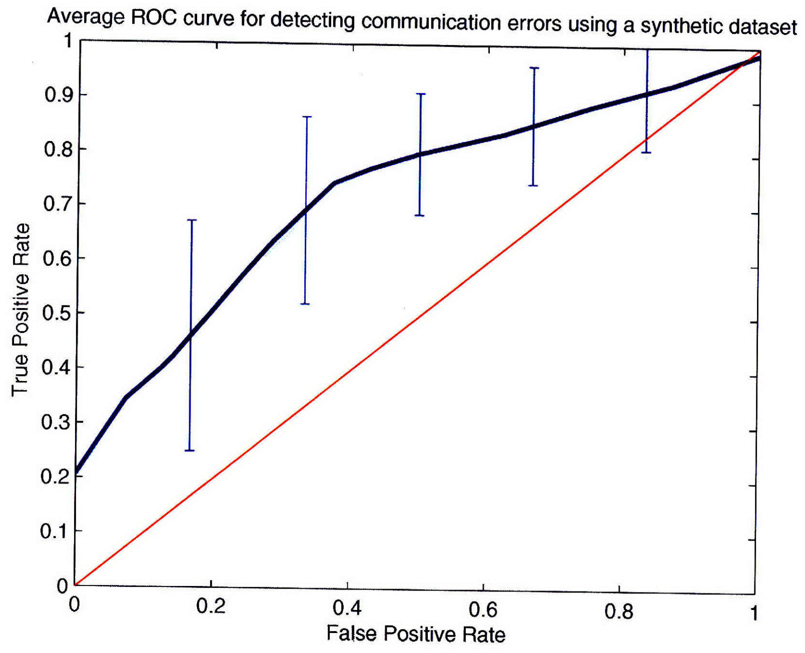


Figure 4-11: Average ROC curve and error bars for detecting communication errors on a staged dataset. The red line, which is used as a reference here, represents detection accuracy at 50%.

generated partitions. For each test set, we plot the ROC curve for communication error detection. We compute the mean and standard deviation of these individual ROC curves and plot an averaged ROC curve with error bars shown in Figure 4-11. From this figure, the detection rate was significantly above chance; however, this performance was evaluated on a staged dataset and the performance on a natural dataset may differ.

We conducted an additional experiment comparing the performance of HMMs in using different visual features. We compared using facial motion features only, head motion features only, and the combination of both. We projected the facial motion features, which was 288 dimensions, to a 10-dimensional subspace using PCA before passing them into a HMM classifier for evaluation and the head motion features, which consisted of six dimensions, were passed directly into a HMM classifier for evaluation. The combined set of facial and head motion features was projected into a ten dimensional subspace before evaluation by a HMM classifier. In the same fashion as the previous experiment, we used leave-one-out cross validation (nine subjects' expressions were used for training and the

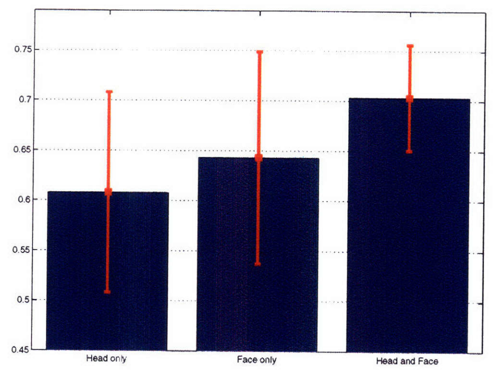


Figure 4-12: Average areas under the ROC curve and the error bars that are using different visual features. From this figure, using the combination of the facial and head motion features performed better.

remaining one was used for testing), with randomly generated partitions. For each test set, we plot the ROC curve for communication error detection. We compute the mean and standard deviation of these ROC curves and plot an averaged area under the ROC curve with error bars in Figure 4-12.

4.7 Preliminary findings

In this chapter, our experiments on the various datasets attained mixed results. In the hidden camera dataset, the visual stream HCRF classifier had a high accuracy of error detection, however, during fusion, only the SUM rule HCRF classifier performed better. In the motivation dataset, the HMM classifier and human perception performance were barely above chance, indicating the challenging nature of this dataset and that most human speakers were not expressive in a motivated setting. HCRF experiments were also conducted on this dataset, but the results were not significantly better than HMMs.

Apart from issues with our experiments, we collected three datasets under various settings to detect communication errors. A key issue that surfaced during these experiments was that the camera was passively monitoring the speaker and the speaker was not aware that camera could be an active visual feedback to the system. Would the speaker become more expressive if he/she is aware of this visual feedback? From a user interface point of view, shouldn't the speaker know that he/she could provide visual feedback? Another variable we did not explore was the effect of avatars, or human-like characters on facial expressivity.

In the next chapter, we will introduce voting Latent Dynamic Random Fields that will prove to perform more consistently, and the dataset we will collect will inform the speaker of the active feedback.

Chapter 5

Efficient Learning of Visual Communication Errors: Voting Latent Dynamic Conditional Random Fields

In this chapter, we will introduce voting Latent Dynamic Conditional Random Fields for communication error detection. We have used HCRFs as a model for communication error detection in previous experiments. Initial experimental results with HCRFs were extremely poor due to the small amount of usable data, and the challenging nature of the motivation dataset described in Section 4.5. HCRFs may seem well-suited to this problem of communication error detection based on our experimental results in the hidden camera dataset, however, in practice, HCRFs tend to overfit and perform poorly when less training data is available. This is due to the need to find optimal values for a high number of free parameters in the HCRF model. In addition, the HCRF is not a convex function due to the potential function representing the binding between the hidden states to the gesture class label. We introduce a new variant of the CRF, which we name as Voting Latent Dynamic Conditional Random Fields(VLDCRFs). Before we can introduce Voting LDCRFs, we will describe LDCRFs in detail in the next section.

5.1 Latent Dynamic Conditional Random Fields (LDCRFs)

The Latent Dynamic Conditional Random Field, developed by Morency et al. [58], is a discriminative model that is originally designed to predict a sequence of labels given a sequence of observations. The LDCRF is a model that combines the strengths of the CRF and the HCRF. Like the CRF, the LDCRF learns the transition dynamics between classes. At the same time, the LDCRF incorporates hidden states like the HCRF, and learns the internal structure of each class. The LDCRF differs from the HCRF by assigning a class label per observation, and does not share the hidden states between classes. In addition, the hidden states are only shared within the same class. By binding a disjoint set of hidden states to each class label, inference on the LDCRF is more efficiently computed using belief propagation during training and testing than the HCRF.

In our task of communication error detection, we want to predict a single label given a pre-segmented sequence of observations but the LDCRF predicts a sequence of labels. We will describe how LDCRFs are applied to gesture recognition in unsegmented sequences, how it has a smaller complexity than HCRFs, and later show an additional voting scheme that transforms this LDCRF model to predict a single label for our purpose.

5.2 LDCRFs for unsegmented gesture recognition

Following [58], in unsegmented gesture recognition, the goal is to predict a label per frame given a sequence of video frames. This means our task is to learn a mapping between a sequence of observations $\mathbf{x} = \{x_1, x_2, ..., x_m\}$ and a sequence of labels $\mathbf{y} = \{y_1, y_2, ..., y_m\}$. Each y_j is a class label for the jth frame of a video sequence and is a member of a set Y of possible class labels. Each frame observation x_j is represented by a feature vector $\phi(x_j) \in \Re^d$. The feature vector in our case, would be the projected facial motion and head motion features. A set of hidden variables $\mathbf{h} = h_1, h_2, ..., h_m$, not observed in the training sequence, are assumed in the model. These hidden variables represent the internal structure of a gesture class. From these assumptions, a latent conditional model is defined as:

$$P(\mathbf{y}|\mathbf{x},\theta) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h},\mathbf{x},\theta)P(\mathbf{h}|\mathbf{x},\theta) \tag{5.1}$$

where θ are the parameters of the model.

Unlike the HCRF, the hidden states are not shared between class labels. Each hidden state belongs to a set $\mathcal{H}_{y_j}$ of hidden states, and each class label y_j has a unique set $\mathcal{H}_{y_j}$ of hidden states. The union of all h_y sets forms the set of all possible hidden states $\mathcal{H}$. By establishing this deterministic mapping between the hidden states H_{y_j} and the class label y_j, $P(\mathbf{y}|\mathbf{h},\mathbf{x},\theta) = 0$ for all sequences which have any $h_j \notin H_{y_j}$. Following [58], Equation 5.1 becomes simplified as:

$$P(\mathbf{y}|\mathbf{x},\theta) = \sum_{\mathbf{h}:\forall h_j \in H_{y_j}} P(\mathbf{h}|\mathbf{x},\theta) \tag{5.2}$$

$P(\mathbf{h}|\mathbf{x},\theta)$ has the same general form as a typical CRF model:

$$P(\mathbf{h}|\mathbf{x},\theta) \propto \exp(\sum_k \theta_k \cdot F_k(\mathbf{h},\mathbf{x})) \tag{5.3}$$

and $\mathbf{F}_k$ is defined as

$$\mathbf{F}_k(\mathbf{h},\mathbf{x}) = \sum_{j=1}^{m} f_k(h_{j-1},h_j,\mathbf{x},j) \tag{5.4}$$

where a feature function $f_k(h_{j-1},h_j,\mathbf{x},j)$ is computed at each observation x_j of the observation sequence $\mathbf{x}$. Each feature function is either a state function $s_k(h_j,\mathbf{x},j)$, which depends on a single hidden variable, or a transition function $t_k(h_{j-1}, h_j, \mathbf{x}, j)$, which depends on pairs of hidden variables.

Following [85, 67, 58], assuming the training set consists of n labeled sequences $(\mathbf{x}_i, \mathbf{y}_i)$, we use the following objective function to learn the parameter θ^*:

$$L(\theta) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i) - \frac{1}{2\sigma^2}||\theta||^2 \tag{5.5}$$

Following [58], the first term in Equation 5.5 is the log-likelihood of the training data. The second term is the log of a Gaussian prior with variance σ^2, i.e., $P(\theta)\ \exp(\frac{1}{2\sigma^2}||\theta||^2)$. We use gradient ascent to search for the optimal parameter values, $\theta^* = \arg\max_\theta L(\theta)$, under this criterion. Further details about training the parameters can be found in [58].

During testing, given a new test sequence $\mathbf{x}$, the most probable label sequence $\mathbf{y}^*$ is estimated as:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \sum_{\mathbf{h}:\forall h_i \in H_{y_i}} P(\mathbf{h}|\mathbf{x}, \theta^*). \tag{5.6}$$

where the parameter values θ^* are learned from training examples. The label y_j of frame j is estimated in two steps. First, we compute the marginal probabilities $P(h_j = a|\mathbf{x}, \theta^*)$ for all possible hidden states $a \in H$. Second, the marginal probabilities are summed according to the disjoint sets of hidden states H_j and the label associated with the optimal set is chosen.

Although both the LDCRF and HDCRF use the same objective function Eq. 3.2 to train their parameters, this function breaks down into a much simpler one, Eq. 5.2, for the LDCRF. This is because specific hidden states are clamped to a class label. For the HCRF, there is no such disjoint set of hidden states associated to a label, and integrating the exponential function $e^{\Psi(y,\mathbf{s},\mathbf{x};\theta,\omega)}$ across all class labels is required (see Eq. 3.2). For example, for the LDCRF model, during training, the gradient of the objective function $L(\theta)$ is computed as:

$$\frac{\delta L(\theta)}{\delta \theta_k} = \sum_{j,a} P(h_j = a|\mathbf{y}, \mathbf{x}, \theta) s_k(j, a, \mathbf{x}) - \sum_{\mathbf{y}',j,a} P(h_j = a, \mathbf{y}'|\mathbf{x}, \theta) s_k(j, a, \mathbf{x}) \tag{5.7}$$

where

$$P(h_j = a|\mathbf{y}, \mathbf{x}, \theta) \quad = \quad \frac{\sum\limits_{\mathbf{h}:h_j=a \wedge \forall h_j \in H_{y_j}} P(\mathbf{h}|\mathbf{x}, \theta)}{\sum\limits_{\mathbf{h}:\forall h_j \in H_{y_j}} P(\mathbf{h}|\mathbf{x}, \theta)} \tag{5.8}$$

Eq. 5.8 is shown in [58] to be computable in $O(m)$ using belief propagation, where m is the number of observations in the sequence. However, the same function is shown in [?] to be computable in $O(|Y|m)$, where Y is the set of possible class labels. Hence, the LDCRF's training complexity is less than the HCRF and the HCRF will tend to overfit faster than the LDCRF whenever the training data size is small. Given these unique properties of the LDCRF, we attempt to fold LDCRF back into the segmented task by using a voting scheme.

5.3 Voting Latent Dynamic Conditional Random Fields for communication error detection

In the LDCRF, the most probable label sequence $\mathbf{y}^* = \{y_1^*, y_2^*, .., y_m^*\}$ is estimated given an observation sequence $x = \{x_1, x_2, ..., x_m\}$. However, in our error detection problem, our sequences are pre-segmented and we only need a single most probable label given an observation sequence. We find the most probable label by finding the majority vote of the most probable label per frame.

Suppose we are given a sequence of labels $\mathbf{y}^* = \{y_1^*, y_2^*, .., y_m^*\}$, then the final assigned label $\mathbf{Y}^*$ is given as:

$$Y = \arg\max_{l} \sum_{i=1}^{m} f(y_i, l)$$

where

$$f(y_i, l) = \begin{cases} 1 & : \quad y_i = l \\ 0 & : \quad y_i \neq l \end{cases}$$

and l is a member of the set of Y possible class labels.

This method makes use of the simpler objective function of the LDCRF for learning the parameters, and associate each label with a disjoint set of hidden states. Unlike the HCRF, estimating the overall class of the segment requires a few steps. First, the label y_j of frame j is estimated by computing all the marginal probabilities $P(h_j = a|\mathbf{x}, \theta^*)$ for all possible hidden states $a \in H$. Then, the marginal probabilities are summed according to the disjoint sets of hidden states H_j and the label associated with the optimal set is chosen for a given frame. Finally, we count the label associated to each frame in the segment, and choose the label with the highest vote count. Now that we have introduced voting LDCRFs, we proceed to the next section to describe the datasets and experiments that use this model.

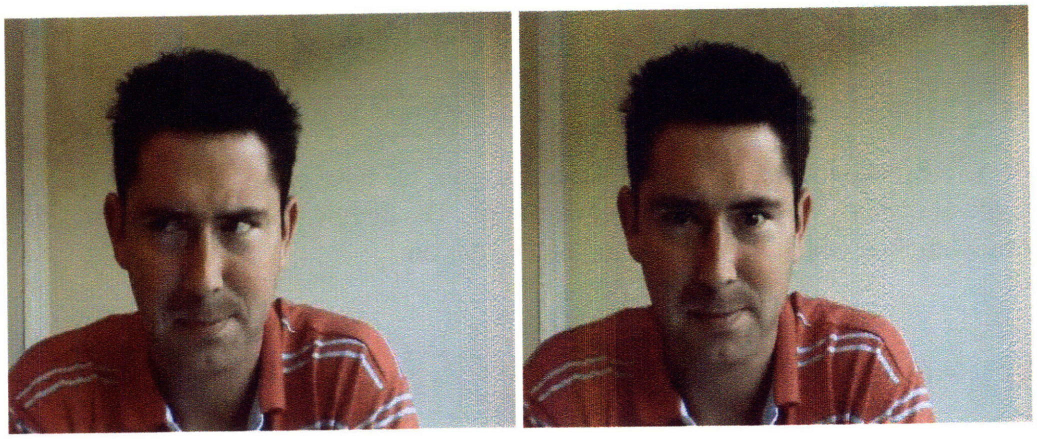


Figure 5-1: Sample images of a subject in the Tilburg dataset. The left image is an example of the subject in an error state, and the right image is an example of a non-error state. Note that in this setup, the camera is visible to the subject.

5.4 Voting LDCRF datasets and experiments

In this section, we describe the datasets we have acquired or collected for evaluating voting LDCRFs and discuss the results.

5.4.1 Tilburg Dataset

For preliminary evaluation of communication error detection in natural settings, we acquired the dataset collected by Barkhuysen et al. [5] for system questions. In this dataset, subjects have made a train ticket reservation and are listening to system verification questions. Nine subjects' video footage during the system's verification response is used for our experiments. Here is an example of a system verification question without errors:

```
User:   Amsterdam
System:   So you want to travel to Amsterdam?
```

And an example verification question with error:

```
User:   Rotterdam
System:   So you want to travel to Amsterdam?
```

More details of this dataset could be found in [5].

	Passive Input	Active Input
No avatar display	Partition 1	Partition 3
Avatar display	Partition 2	Partition 4

Table 5.1: Breakdown of the Active-Avatar dataset, we have two variables: type of input and type of display. This produces four partitions. The number in each entry represents the partition number, which we use as a reference later on.

5.4.2 Active-Avatar Dataset

A subject's facial expressivity could be dependent on internal and external factors. One possible internal factor could be the subject's knowledge of the presence of the camera and knowledge that their expressions are an active input. External factors could be the presence of an avatar as a display which makes the speaker feel more engaged in the conversation or the content of the conversation. In this dataset, we controlled for two variables: active vs passive input and avatar vs no-avatar displays. Active input means the subject is briefed that the computer recognizes his/her facial expressions, and with this knowledge, the subject would feel more inclined to be expressive visually to improve the communication process. On the other hand, the subject may be briefed that the camera is just capturing visual footage passively and not actively recognizing the subject's face, which counts as passive input. The other variable we controlled for was the presence of a visual display. This display is in the form of an avatar while the other option is no display at all. In total, four partitions were created. Table 5.4.2 shows the breakdown of the four partitions and their different attributes. Four Wizard-of-Oz experiments, which represents these four partitions, were conducted. Each subject can only take part in two of the experiments, not all of them. The subjects were randomly assigned the two experiments that will either cue the subject to express himself/herself, or just interact normally with passive observation by the camera. To be clear, using Table 5.4.2 as a reference, a subject can only take part in Parts 1 and 2 *or* Parts 3 and 4. In both experiments, the subject was instructed to make airline reservations with a conversational system. Unknown to the subject, a human was actually manipulating the response of the system. In the first experiment, there was no visual display (Part 1 or 3 in Table 5.4.2). In the second experiment, the subject can choose between three different avatars to interact with on the laptop display (Part 2 or 4 in Table 5.4.2). Figure 5-2 shows the three avatars subjects can choose from for interaction.

Here is a sample transcript of the briefing given by the computer prior to the first experiment:

```
Computer:  "Welcome to the Vision Interface kiosk.  I can
make airline reservations for you.  Which city do you want
to travel from?"

Speaker:  "Boston."
```

Here is a sample transcript of the briefing given by the computer prior to the second experiment:

```
Computer:  "Welcome to the VIP kiosk.  I can make airline
reservations for you.  In anytime during the conversation,
if I give an incorrect reply, you can express your disagreement
using facial expressions in a natural manner.  For example,
you can roll your eyes, frown, shake your head to express
your disagreement.  Do you understand?"

Speaker:  "Yes."

Computer:  "Okie.  Let's begin.  Which city do you want to
travel from?"
```

In both experiments, three types of communication errors occurred: verification, confirmation and apology errors. Here is a sample transcript containing a verification error followed by a confirmation error,

```
Computer:  "So you want to travel from Boston to San Diego
on Monday at 9 am.  Is that correct?"

Speaker:  "No."

Computer:  "Your reservation is now confirmed.  Good bye!"
```

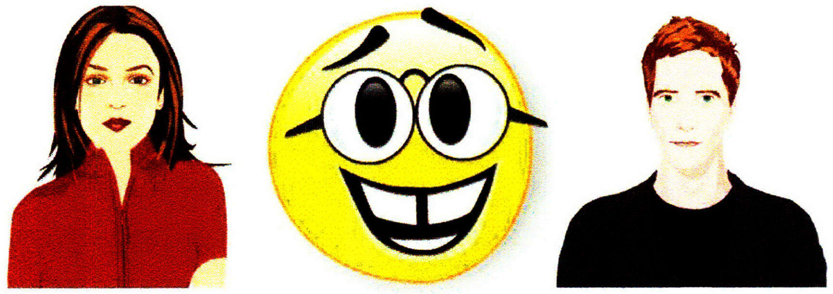


Figure 5-2: Avatars used for visual feedback. In the dataset accompanied with visual feedback, subjects can select from one of these three avatars for interaction.

Here is a sample transcript containing a correct verification followed by an erroneous apology,

```
Computer:  "So you want to travel from Boston to San Diego
on Monday at 9 am.  Is that correct?"

Speaker:  "Yes."

Computer:  "I am sorry I made a mistake.  Let's start over.
Which city do you want to travel from?"
```

Twenty subjects' video footage during the occurrence of these errors were used for our experiments.

Active-Avatar dataset: feature extraction and pre-processing In this dataset, there is a slight difference in the feature pre-processing step that was not done in previous datasets. Preliminary results using facial and head motion features projected onto a ten-dimensional subspace did not produce any meaningful results for all models and we increased the size of the projected subspace to 20 dimensions.

We describe the experiments and results on these datasets in the next section.

Dataset Name	Hidden Camera?	Type of Visual Input	Type of Display	Methods used
Hidden Camera Dataset	Y	Passive	Map display	HMMs, HCRFs
Motivation Dataset	N	Passive	Map display	HMMs, HCRFs
Synthetic Expressions	N	Active	None	HMMs
Tilburg Univ. [5]	N	Passive	None	HMMs, HCRFs, VLDCRFs
Active-Avatar dataset Part 1	N	Passive	None	HMMs, HCRFs, VLDCRFs
Active-Avatar dataset Part 2	N	Passive	Avatar display	HMMs, HCRFs, VLDCRFs
Active-Avatar dataset Part 3	N	Active	None	HMMs, HCRFs, VLDCRFs
Active-Avatar dataset Part 4	N	Active	Avatar display	HMMs, HCRFs, VLDCRFs

Table 5.2: Breakdown of datasets collected for communication error detection

5.4.3 Tilburg Dataset: experiments and results

On this dataset, we conducted several experiments. In the first experiment, we compared the performance of a user-independent model versus human performance. We conducted a nine-fold cross validation and made sure our Hidden Markov models were not trained on any data coming from a subject in the test set. In the same manner as described in the experiments on the staged dataset, we compute and plot the averaged ROC curve with error bars for communication error detection. This is shown in Figure 4-9, where the averaged ROC curve was plotted in a solid blue line. To compare against human performance, we conducted a human perception experiment, where video segments evaluated by the HMMs were also evaluated by four human observers. These four human observers did not view any sample video footage of the subjects before the evaluation. The results of the four human observers were denoted as four green circles in Figure 4-9. The difference in detection accuracy between humans and HMMs was not statistically significant after performing a paired t-test ($p = 0.78$). This showed that the detection accuracy by HMMs was indistinguishable from human performance.

In the second experiment, we compared various models against the human performance reported in [5]. We trained and tested HMMs, HCRFs, LDCRFs and voting LDCRFs, such that video clips from the same subject were used in training and testing. Note that we made sure the test set consisted of the same clips used for human evaluation in [5]. On average, each subject had fifteen video clips used in the training set and three video clips for the test set. The ROC curves of the various models are plotted in Figure 4-9. From this experiment

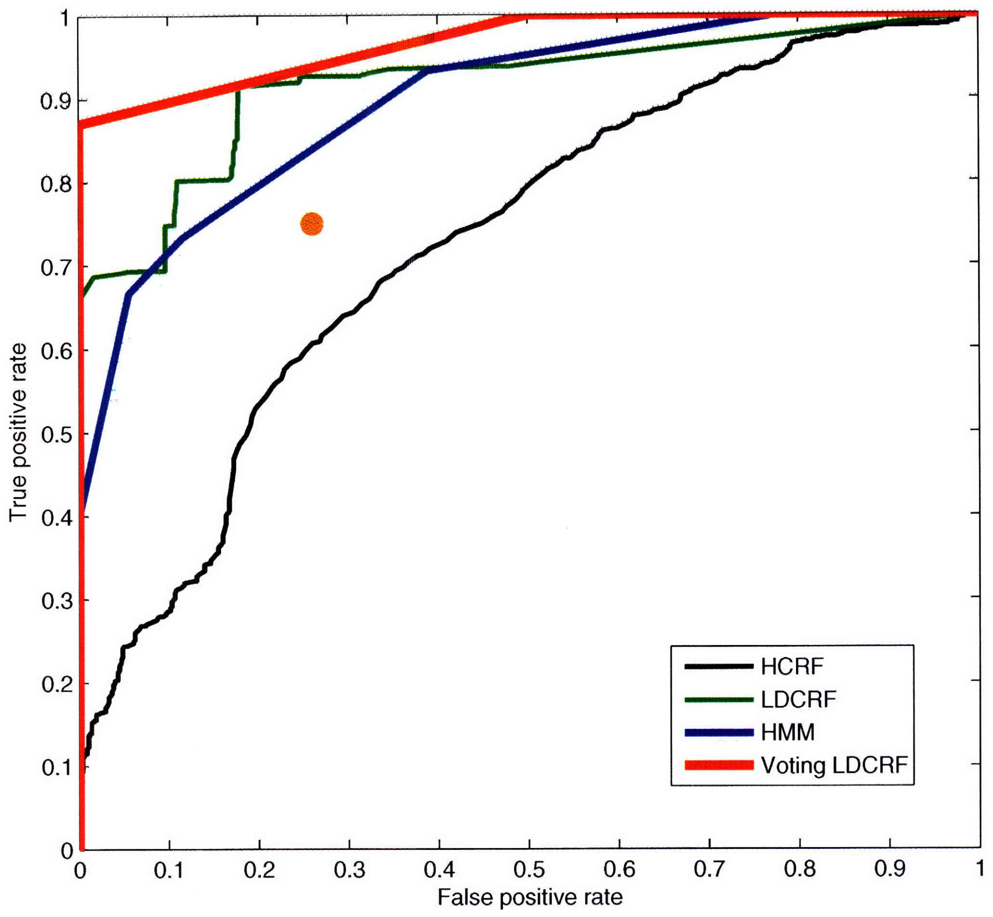


Figure 5-3: ROC curves for detecting communication errors using various algorithms for the second experiment.

onwards, all the ROC curves are plotted using Algorithm 2 described in Appendix A, except for the HCRF and LDCRF curves which are plotted using Algorithm 1. We did not have time to recompute convex-smoothed HCRF and LDCRF curves before submission of this thesis, but the reader can inspect the curves and see that they are significantly inferior whether convex smoothed or not. From this figure, voting LDCRFs performed best among the four models.

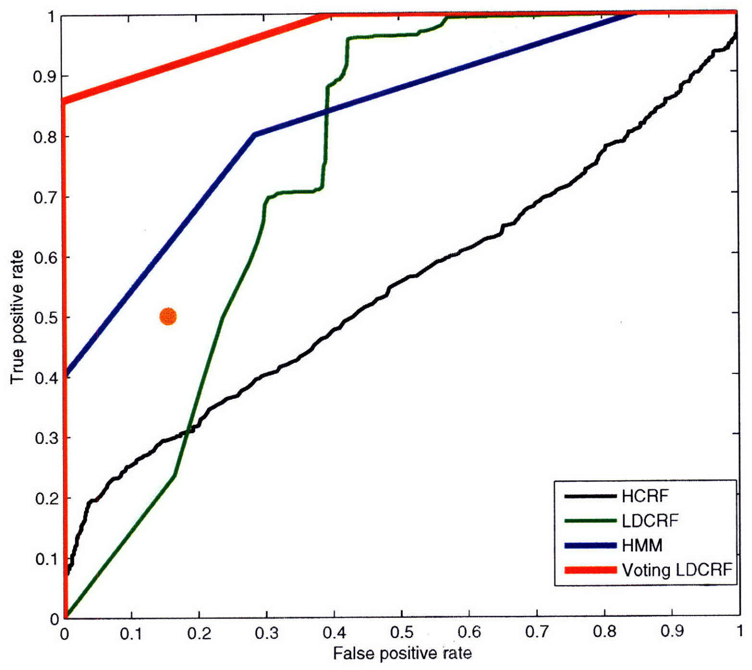


Figure 5-4: ROC curves for detecting communication errors using partition 1.

5.4.4 Active-Avatar Dataset: experiments and results

There are four partitions in this dataset (see Table 5.4.2). In each partition, we compared various models against human performance in the same manner as the second experiment in the Tilburg dataset.

We trained and tested HMMs, HCRFs, LDCRFs and voting LDCRFs, such that video clips from the same subject were used in training and testing. We also made sure the test set consisted of the same clips used for human evaluation. There were four human evaluators. Figures 5-4, 5-5, 5-6 and 5-7 show the ROC curves of various models in partitions 1, 2, 3 and 4 respectively. In all these curves, the average human performance is denoted by an orange star. We further plot an averaged ROC curve over these four experiments, and the averaged ROC curves are shown in Figure 5-8. In all these figures, the voting LDCRFs has a high detection accuracy and performs better consistently than the other models.

As a further analysis of the voting LDCRFs, we examined the voting pattern per frame of a test sequence. This frame-by-frame analysis is displayed in Figure 5-9. The top row

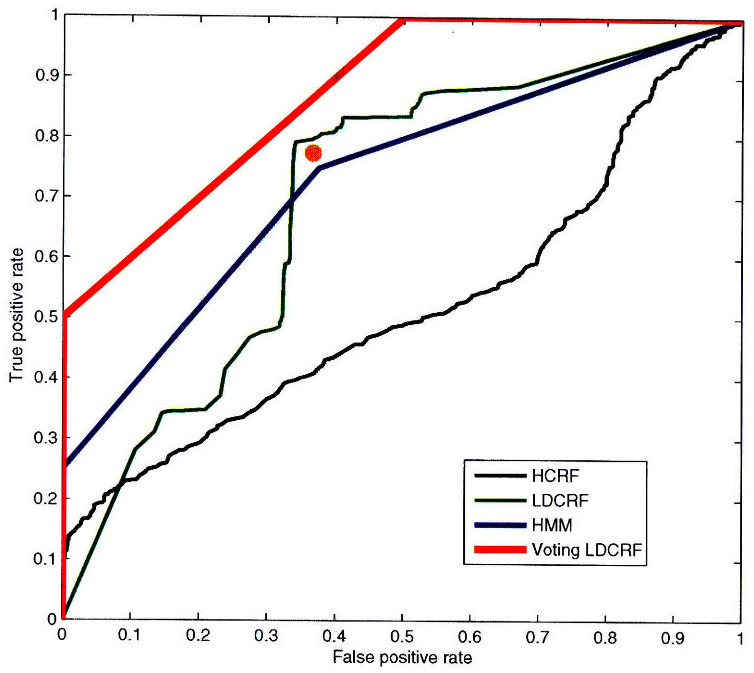


Figure 5-5: ROC curves for detecting communication errors using partition 2.

displays the original frames, the 2nd row displays the corresponding face tracker superimposed on the original frame. The table below the frames shows the cumulative vote count for errors versus non-errors as the sequence progresses from the left to the right. The bottom row shows the words spoken by the system at the same time. These words were part of the reply "So you want to travel from Boston to San Francisco on Monday at 9 am. Is that correct?". From this figure, the cumulative error vote started increasing synchronously with the frames that were more indicative of an error i.e. when the subject rolled up his eyes beginning from the third frame.

In a second experiment, we explored two issues in communication error detection. We questioned if speakers will be more expressive when they have knowledge of active visual input to the system than if the system was just passively monitoring them, regardless of the presence of an avatar. We also explored the performance of the models and speaker expressivity if avatars are on display compared against no display, regardless of the knowledge of the speaker (active visual input or not). We combined the partitions in the first experiment

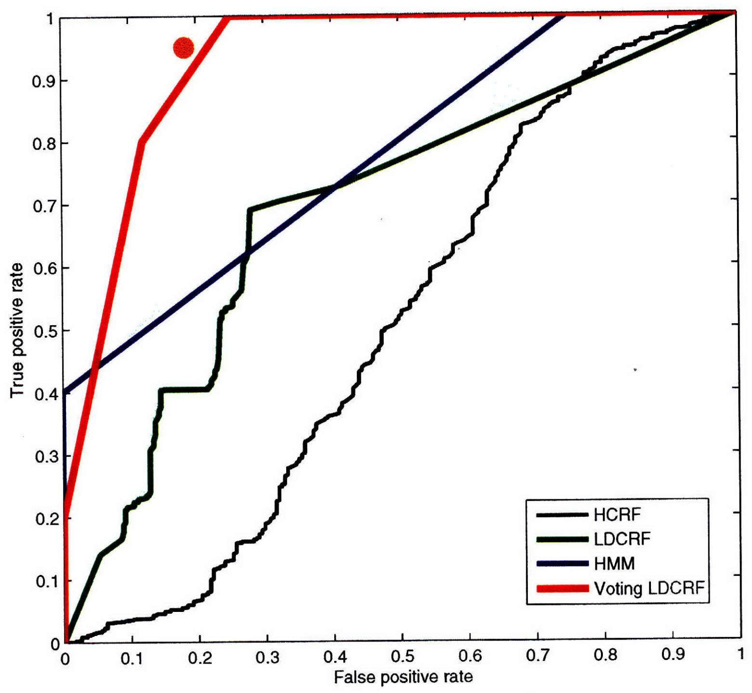


Figure 5-6: ROC curves for detecting communication errors using partition 3.

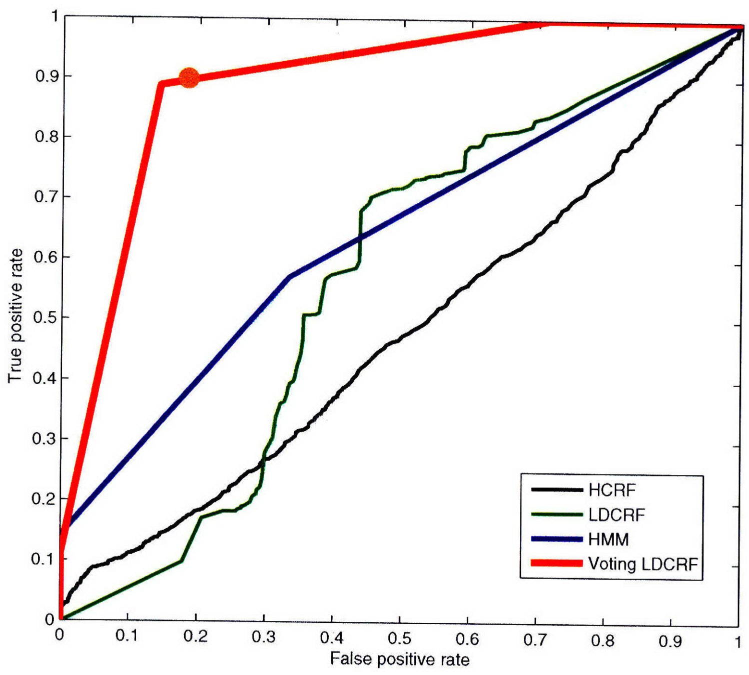


Figure 5-7: ROC curves for detecting communication errors using partition 4.

to form new ones. Here is the breakdown of the new partitions and their constituents:

- Passive partition: combination of partitions 1 and 2
- Active partition: combination of partitions 3 and 4
- No display partition: combination of partitions 1 and 3
- Avatar display partition: combination of partitions 2 and 4

In each of these new partitions, we compared various models against human performance. We trained HMMs, HCRFs, LDCRFs and voting LDCRFs. The training and the testing sets are exactly the same ones in the first experiment. Figures 5-10, 5-11, 5-12 and 5-13 show the ROC curves of various models in the passive, active, no display, and avatar display partitions respectively. The average human performance is denoted by an orange star in each figure. We were expecting the active partition's curves to be much better than the passive partition, and that the avatar partition's curves to be much better than the

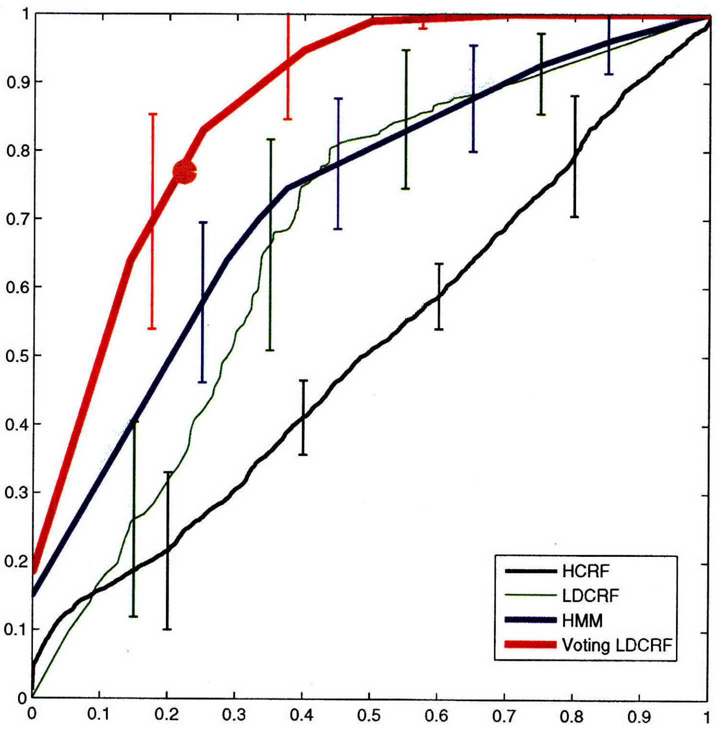
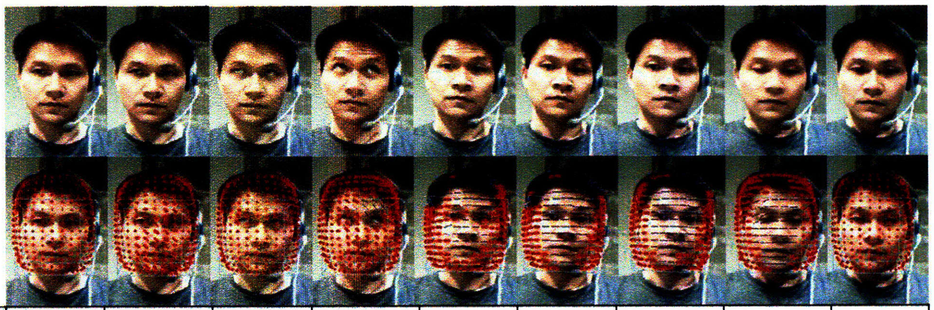


Figure 5-8: Averaged ROC curves for detecting communication errors using all partitions.



No Error cumulative count	46	47	47	47	47	47	47	47	47
Error cumulative count	84	84	85	86	87	88	89	90	91
System Reply	......	on				Mon		....	

Figure 5-9: Diagram showing a breakdown of a portion of an error test sequence into frames. The top row displays the original frames, the 2nd row displays the corresponding face tracker superimposed on the original frame. The table below the frames show the cumulative vote count for errors versus non-errors. The bottom row shows the words spoken by the system at the same time. These words were part of the reply “”So you want to travel from Boston to San Francisco on Monday at 9 am?”

no avatar partition. From the avatar and no avatar figures (Figures 5-12 and 5-13), the difference in performance of the ROC curves is not significant. The performance of the avatar curve was poorer compared to experiments using the individual partition 4, but better than that of partition 2. This implies that knowledge of active visual input has a significant effect on human subjects when they are interacting in the presence of an avatar and that detection models for partitions 1 and 2 should be trained and evaluated separately.

On the other hand, the figures in the passive and active partitions (see Figures 5-10 and 5-11) contradict with our expectations. The voting ldcrf ROC curve in the passive partition performs much better than the same curve in the active partition. One interesting thing to note is that the voting ldcrf performed better than (or is almost as good as) human perception in the passive partition. This could imply several things: first, humans are poor evaluators of passive expressions. Second, the active partition's performance is poorer than the individual partitions 3 and 4 in the first experiment, which implies that the expressions the models learn in partition 3 were different from the ones the models learn in partition 4. This tells us that the avatar has a different effect on facial expressivity on the human

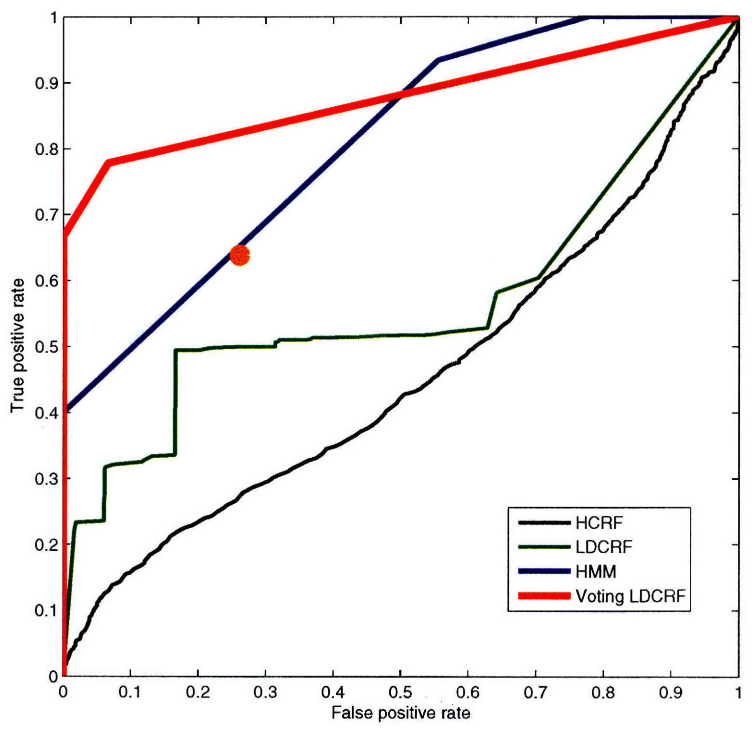


Figure 5-10: ROC curves for detecting communication errors using the passive partition.

subject compared to no avatars, and that our models should not mix the datasets in the two scenarios together during training. Third, in the passive partition, the presence of an avatar does not create significant change in performance of the models, implying that the expressions in partitions 1 and 2 are very similar. From this second experiment, we conclude that when users have active knowledge of visual input, separate models should be trained for interacting with and without an avatar.

From the first and second experiment, we can conclude that voting LDCRFs performed better consistently in all scenarios of communication errors. This conclusion is clearly shown in the plots using partitions 1, 2, 3, 4 and the averaged roc curve of these four partitions (Figures 5-4, 5-5, 5-6, 5-7 and 5-8). This supports our claim that we have devised a scheme for automatic communication error detection.

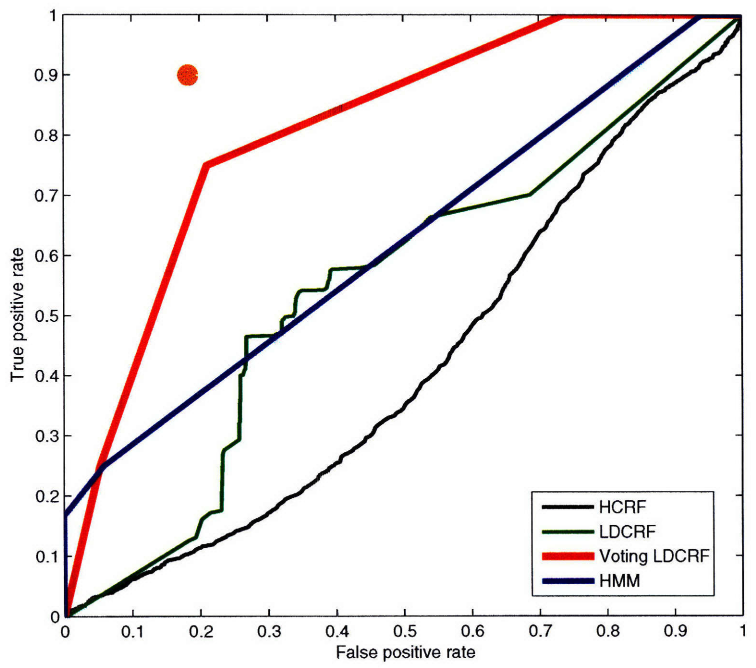


Figure 5-11: ROC curves for detecting communication errors using the active partition.

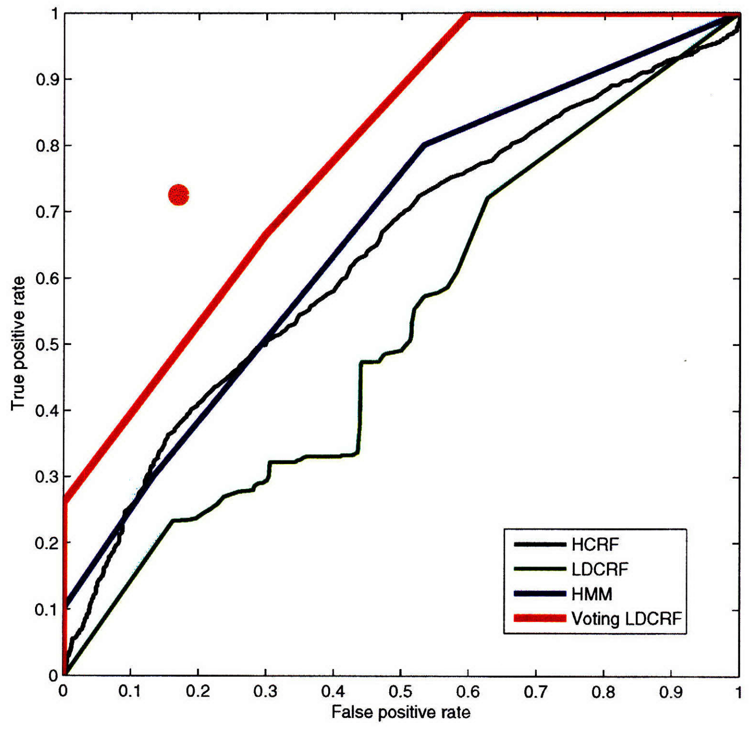


Figure 5-12: ROC curves for detecting communication errors using the no display partition.

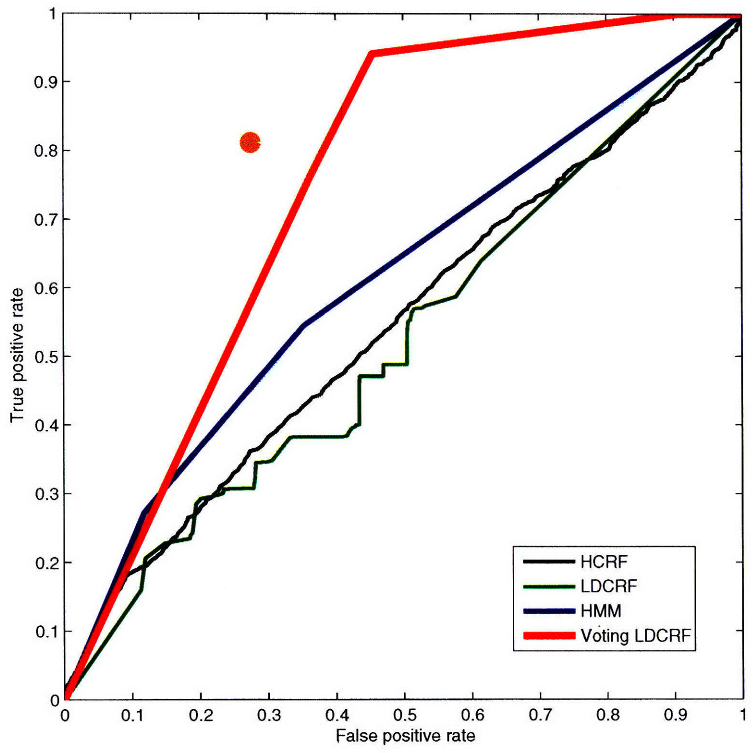


Figure 5-13: ROC curves for detecting communication errors using the avatar display partition.

Chapter 6

Conclusion

In this thesis, we have introduced the first automatic system for detecting communication errors using facial expressions during the system's turn. This is useful as it detects communication problems within the same conversation turn, unlike previous work that detects errors in the next turn or speaker's turn. Our experiments show that humans do exhibit passive, emotional facial expressions e.g. frustration, confusion that can be indicative of a communication error during the system's reply. When human subjects become aware that the conversational system is capable of receiving visual input, they become more communicative visually yet naturally, and our system becomes more successful at detecting the errors. Any conversational system can benefit from this additional cue to change the course of the dialogue and improve user satisfaction.

The success in this automatic error detection system stems from our prior contributions on exploring gesture recognition models. We first developed a discriminative vector quantization model and showed it was useful for visual gesture recognition. Quantizing the features was analogous to sharing features between gesture classes, and it improved the accuracy of arm gesture recognition significantly. The insight gained from sharing features served as a prelude to our next model. By sharing hidden states across all class labels, we demonstrated that Hidden Conditional Random Fields was a useful model for gesture recognition.

Given our success in arm and head gesture recognition, we applied HCRFs to communication error detection. We attained mixed results due to a smaller and more challenging dataset. Subsequently, we embarked on finding a better model, and we introduced voting Latent Dynamic Random Fields for efficient learning and applied it to visual communication error detection during the system's turn.

Although this algorithm is the first one for visual communication error detection during the system's turn, other previous works on visual prosody analysis like Graf et al. [40] during the speaker's turn can also be applied to detecting errors in the system's turn. One key difference would lie in the features used. For tracking visual prosody during the speaker's turn, use of facial motion features around the mouth is difficult due to the large variability of the words that could possibly be spoken. Detecting communication errors during the system's turn can make use of these features instead, since the speaker is listening in silence. At the very same time, most visual prosody methods use a discriminative model like Support Vector Machines to detect the emotion at the frame level. Such models, while discriminative, do not learn the sequential dynamics between frames, nor learn the hidden structures like the voting latent dynamic conditional random field proposed in this thesis. Learning the hidden structure is useful for analysis and future post processing. In this thesis, however, we have not demonstrated the additional advantage of learning this hidden structure.

6.1 Limitations

The models we have used can only provide the likelihood of an error at a given image frame and do not provide a breakdown of the face into different regions or components. Finding out which regions or components of the face that are highly indicative of an error will be useful. The facial motion features are optical flow velocities in a 12 by 12 grid around the face, which is highly dependent on finding the correct pose of the head. This optical flow method suffers from some drift. For example, for a given frame, the optical flow velocities that map to a specific region of the face, e.g. eyes, will map to the eye brows several frames later. Nevertheless, our voting LDCRF has managed to detect errors successfully despite such limitations. Our error detection model only makes the inference at the end of the system's turn of the conversation. For practical purposes in real time scenarios, a model that makes the inference in the middle of the system's turn could potentially be more useful.

6.2 Future Work

Future work can be divided into three categories: facial expression tracking, recognition models and dialogue strategies.

Facial expression tracking

There are many challenges confronting facial tracking to this day. Facial expressions are very subtle spatially and temporally. For a camera capturing visual footage at 30 Hz, the significant expression may occur only in a few frames. At the same time, some subjects' expressions only consist of a minute displacement of their facial muscles that are barely visible to the human eye. A face tracker with a low degree of freedom will have difficulty capturing this subtle movement. A possible approach could combine a 3D model based tracker like the Piecewise Bezier Volume Deformation Tracker [76] with appearance-based features like Gabor wavelets [36] or boosted Haar features [89]. Such a tracker does not exist at the time of writing this thesis.

Recognition models

For practical purposes, a model that makes the inference in the middle of the system's turn could potentially be more useful. This could be running a window size of 300 msecs over the video sequence, and making an inference at every window frame. However, this method may suffer from many false positives. To boost the performance, we would find out if there is a significant correlation between the timing of the expression with the co-occurring words spoken by the system. A cursory inspection of the videos show that most subject express the communication error at the end of the spoken sentence, however, subjects that express their error earlier usually do so when a key noun was spoken. This implies two things: a multisensory model that includes parts of speech tags as features or a user-specific model that could improve detection accuracy. Both models will be explored.

Another challenge facing error detection models is the requirement of large amounts of data, which is affirmed by our mixed results in HCRFs. Experiments with the motivation dataset has shown that audio cues can be an additional redundant modality. We could explore adapting the visual and audio classifier to a particular user or noise condition in the same fashion as Christodias et al [9].

Dialogue Strategy

The most interesting component to follow this thesis is the dialogue management strategy that will use this automatic visual error detection mechanism. Visual error detection can improve context resolution if the speaker expresses an error during a co-occurring spoken word. Suppose in an airline reservation system, an error was detected when the destination was spoken, the dialogue manager could either:

1. "barge in" by interrupting the speaker's turn and say "I am sorry I did not get the destination correctly, could you repeat it again?"

2. wait for following speaker's turn to be complete, disregard the content in his/her turn and say "I am sorry I did not get the destination correctly, could you repeat it again?"

On the other hand, if the system cannot detect a strong co-occurrence of the spoken word with the detected error, the dialogue manager could adopt a more generic correction by saying something like:

"I am sorry I made a mistake, which did I get wrong? The destination city,the source city or the time? Please say source, destination or time."

User studies will be conducted to figure out which of these two options or other better strategies should be adopted for improving the conversation.

Another interesting issue to explore is the system design architecture for the automatic "barge-in". Typically, the system has a task/process waiting in the pipeline, which will be turning the microphone on for the speaker's turn. Once the error is detected, the system has to either kill this task waiting in the pipeline or postpone it to the next conversation turn.

Appendix A

Plotting ROC curves

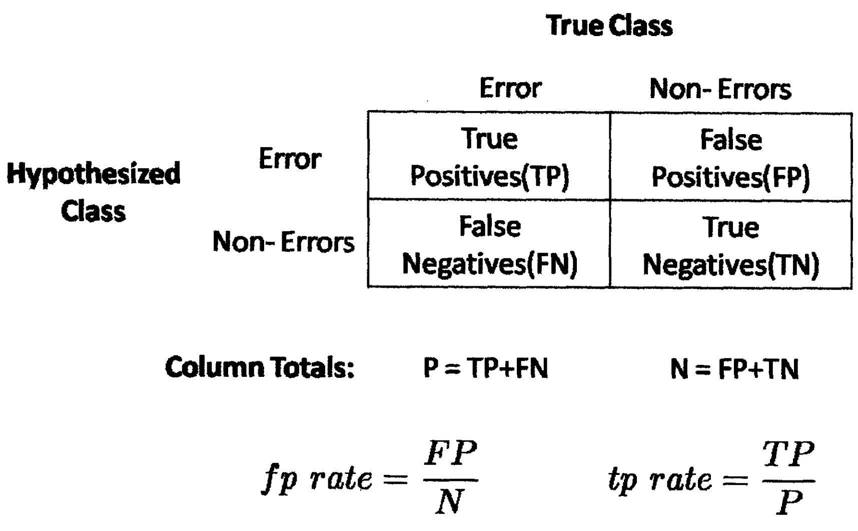
In this section, we will describe two algorithms, enumerated as algorithm 1 and 2, that were used for plotting the Receiver Operating Characteristics (ROC) curve. All the ROC curves in this thesis were plotted based on Algorithm 2. However, in the hidden camera dataset, the curves published in [86] were plotted based on Algorithm 1. We will describe their differences here. Before we do so, we will give a brief introduction to how we plot our ROC curves in this thesis.

A.1 ROC curves: a brief introduction

In communication error detection, for a given test sequence $\mathbf{x}$, our classifier yields a probability score, which represents the degree to which this sequence is a member of the communication error class. Different thresholds may be applied to this score to predict the class membership:

$$P(\mathbf{y} = 1|\mathbf{x}) \;\geq\; \gamma$$

where $\mathbf{y}$ is the class label, and the label assignment 1 represents an error. If the classifier output is above or equal to the threshold γ, the classifier produces an error, else a non-error.



		True Class	
		Error	Non- Errors
Hypothesized Class	Error	True Positives(TP)	False Positives(FP)
	Non- Errors	False Negatives(FN)	True Negatives(TN)
	Column Totals:	P = TP+FN	N = FP+TN

$$fp\ rate = \frac{FP}{N} \qquad tp\ rate = \frac{TP}{P}$$

Figure A-1: Confusion matrix and the performance metrics calculated for this thesis.

Given a classifier, a threshold and a test sequence, there are four possible outcomes:

1. True positive - the test sequence is an error and is classified correctly as an error.
2. True Negative - the test sequence is a non-error and is classified correctly as a non-error.
3. False positive - the test sequence is a non-error and is incorrectly classified as an error.
4. False negative - the test sequence is an error and is incorrectly classified as a non-error.

With a classifier and a set of test sequences, we can construct a confusion matrix illustrated in Figure A-1.

In the same figure, we used two common metrics, the true positive rate and the false positive rate. The true positive rate is estimated as :

$$tp\ \ rate = \frac{No.\ errors\ correctly\ classified}{Total\ no.\ of\ errors\ in\ the\ test\ set}$$

and the false positive rate is estimated as:

$$fp\ \ rate = \frac{No.\ non - errors\ incorrectly\ classified}{Total\ no.\ of\ non - errors\ in\ the\ test\ set}$$

ROC graphs are two-dimensional graphs where the true positive rate is plotted on the y-axis and the false positive rate is plotted on the x-axis. At each given threshold γ in Eqn. A.1, an ($fp\ rate$,$tp\ rate$) pair is produced and plotted as a point on the ROC graph. If we vary the threshold from $-\infty$ to ∞, we will produce a curve through the ROC graph. Algorithm 1, based on this idea, is described below:

Algorithm 1: Conceptual method for creating an ROC curve.

Input: L, the set of test sequences sorted according to their classifier's probability in ascending order, $q(i)$, the classifier's probability that sequence i is an error; *min* and *max*, the smallest and largest values returned by q; *increment*, the smallest difference between any two q values.

```
1  for $\gamma$ = min to max by increment do
2      FP = 0; TP = 0;
3      foreach $i \in L$ do
4          if $q(i) \geq \gamma$ then
5              if $i$ is an error sequence then
6                  $TP = TP + 1$;
7              else
8                  $FP = FP + 1$;
9              end
10         end
11     end
12     Add point $(\frac{FP}{N}, \frac{TP}{P})$ to ROC curve;
13 end
```

γ is increasing from the lowest threshold to the highest one and at each given γ, we compute an $(\frac{FP}{N}, \frac{TP}{P})$ pair to plot on the ROC curve. Algorithm 1, however, suffers from creating concavities in the curve. When the threshold is incremented by one step and only a small test set is available, it is possible that only the number of true positives have changed, resulting only in the change in true positive rate. A portion of the ROC curve would contain an "L" segment. This is evident in the ROC curves published in [86], which are plotted in the left column of Figure A-2. By definition, a classifier is potentially optimal if and only if it lies on the convex hull [4]. Figure A-3 shows the HCRF ROC curve plotted in [86] for visual features classification. The points a, b are located on the convex hull. Each of these points corresponds to a specific probability threshold, thus segment C corresponds to a probability interval. However, segment C is a concave area. Following page three of [32], a concave area means that the ranking obtained from the classifier in this probability interval is worse than random. One way to repair this concavity is by ignoring the probabilistic score calculated by the classifier in this interval, and output a constant score (e.g., the mid-point of the interval). Assuming that ties are broken by assigning a random rank, this would replace the concave region of the ROC curve with the line segment ab. If this procedure is followed for all concavities, this corresponds to constructing the convex hull by discretising the probability scores. More details about removing concavities can be found in [32]. Given this solution, Algorithm 2 is simply the convex hull of the curve generated from Algorithm 1. The ROC curves based on Algorithm 2 are plotted in the right columns of Figure A-2. For the completeness of this thesis, the original unsmoothed ROC curves generated from Algorithm 1 in the Tilburg dataset and the Active-Avatar datasets are plotted here (see Figures A-4, A-5, A-6, A-7, A-8, A-9, A-10, A-11, A-12) with their corresponding smoothed ROC curves.

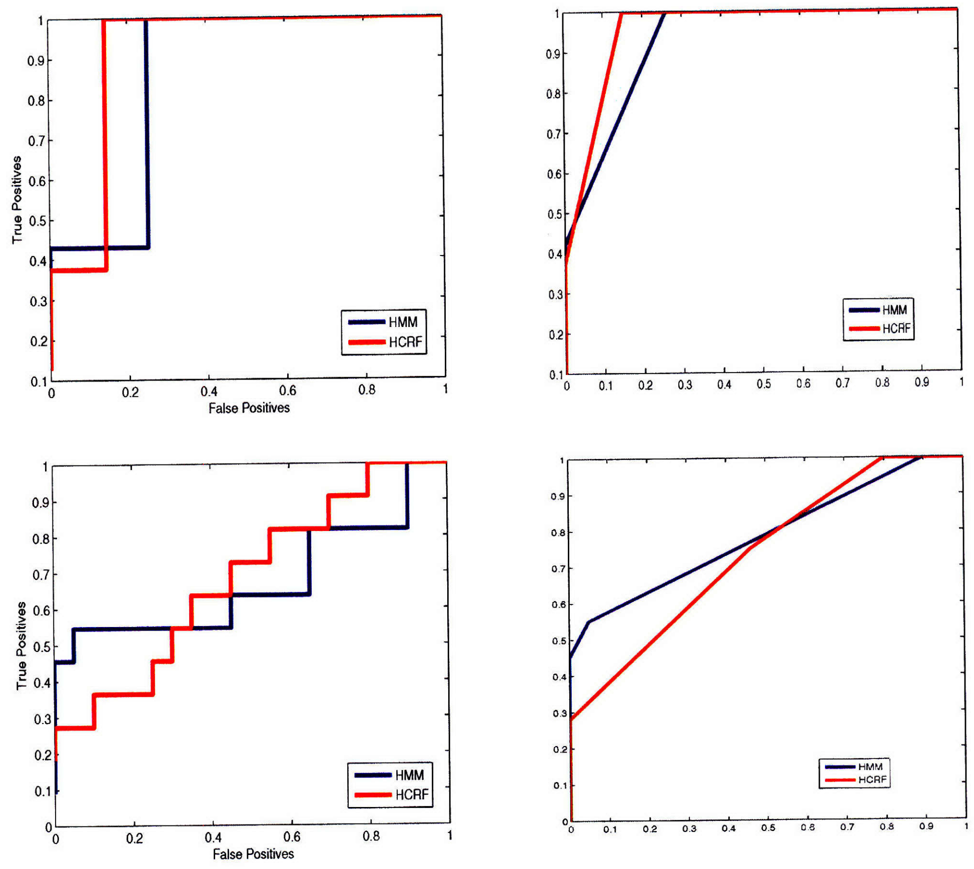


Figure A-2: (top graphs) ROC curves showing the performance of the different classifiers of visual features. The left curve shows the ROC curve plotted using Algorithm 1 and the right shows the same curve plotted using Algorithm 2. (bottom graphs) ROC curves showing the performance of the different classifiers using prosody features. The left curve shows the ROC curve plotted using Algorithm 1 and the right shows the same curve plotted using Algorithm 2. Notice that the graphs in the left column contain many L segments or segments that are made up of sums of multiple step functions.

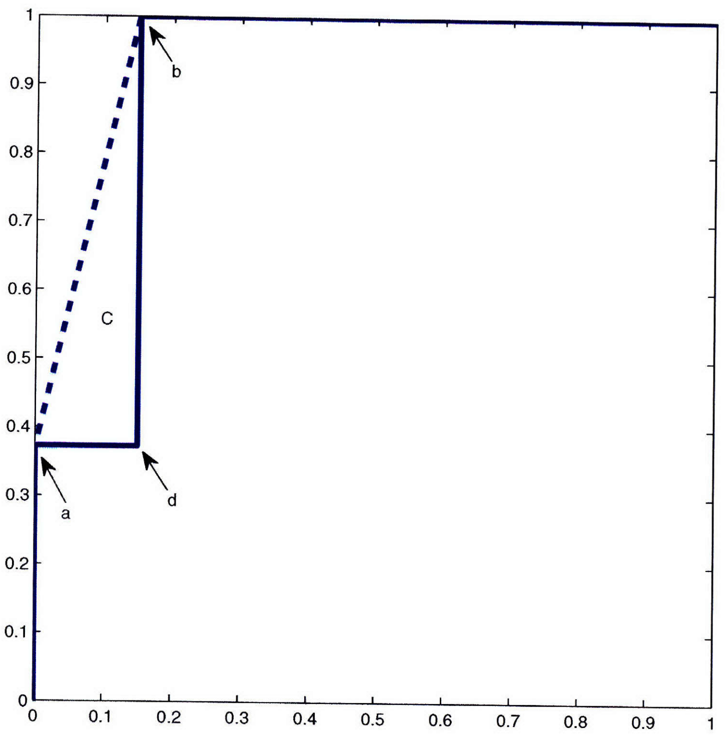


Figure A-3: The HCRF ROC curve from visual features classification and its convex hull.

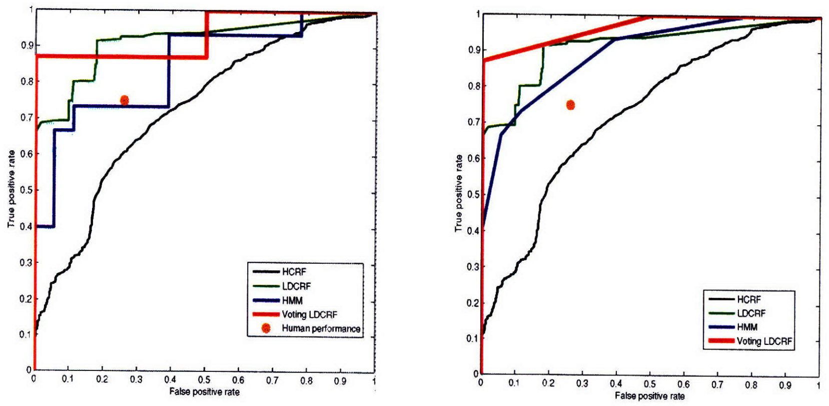


Figure A-4: ROC curves for detecting communication errors using various algorithms for the second experiment of the Tilburg dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

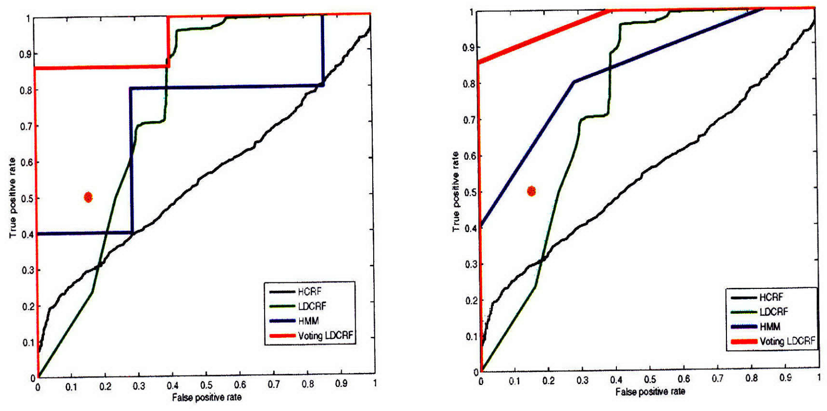


Figure A-5: ROC curves for detecting communication errors using partition 1 of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

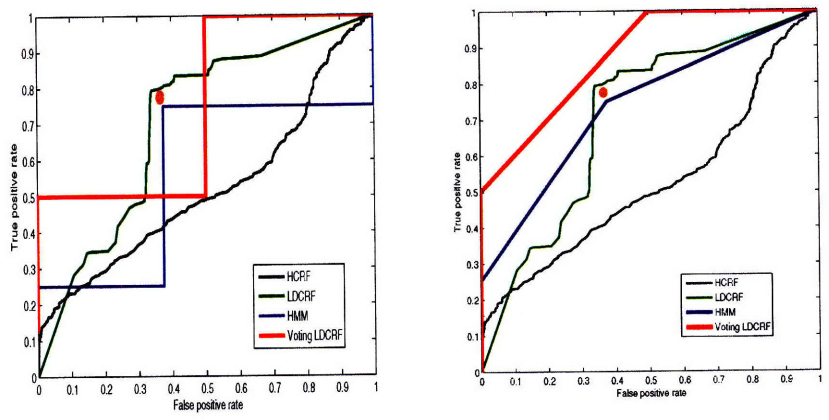


Figure A-6: ROC curves for detecting communication errors using partition 2 of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

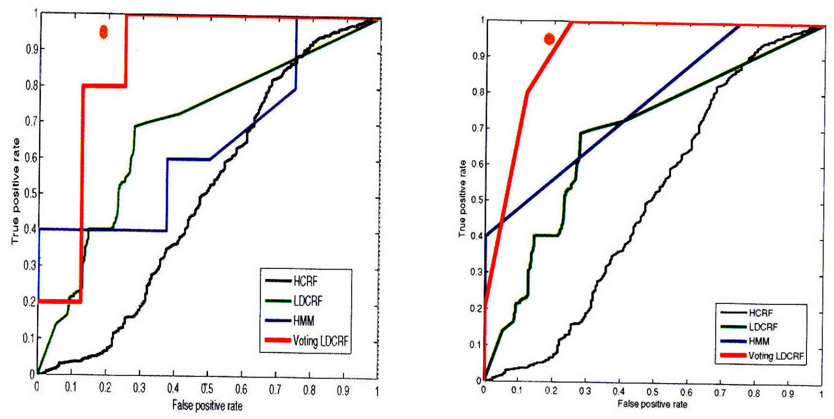


Figure A-7: ROC curves for detecting communication errors using partition 3 of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

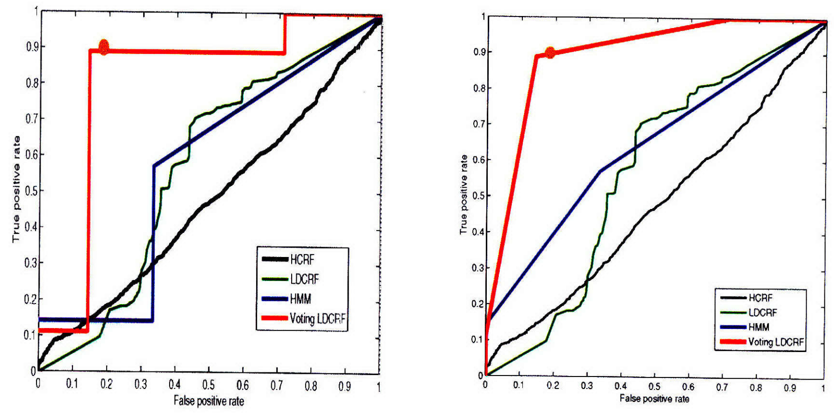


Figure A-8: ROC curves for detecting communication errors using partition 4 of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

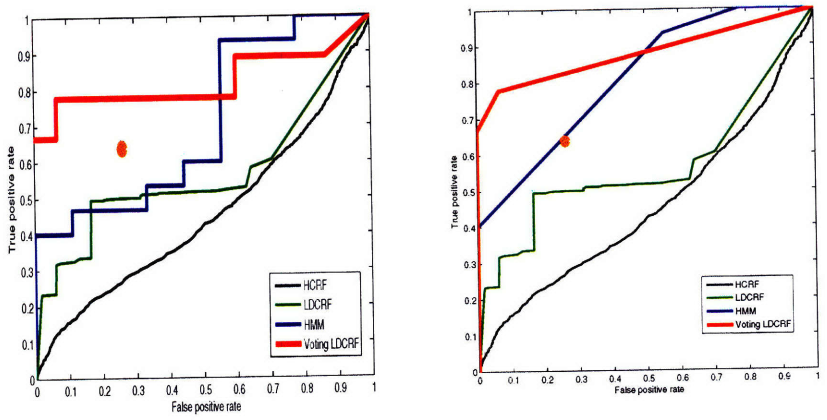


Figure A-9: ROC curves for detecting communication errors using the passive partition of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

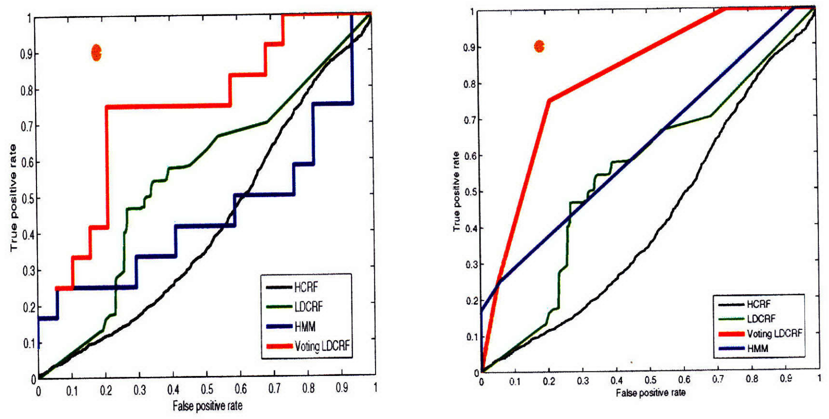


Figure A-10: ROC curves for detecting communication errors using the active partition of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

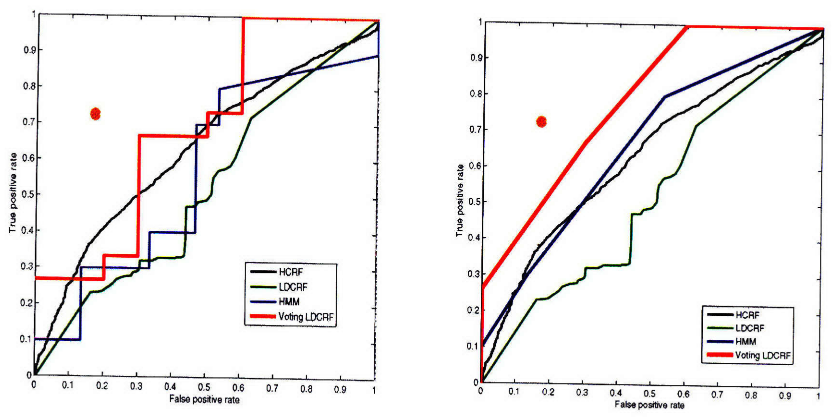


Figure A-11: ROC curves for detecting communication errors using the no display partition of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

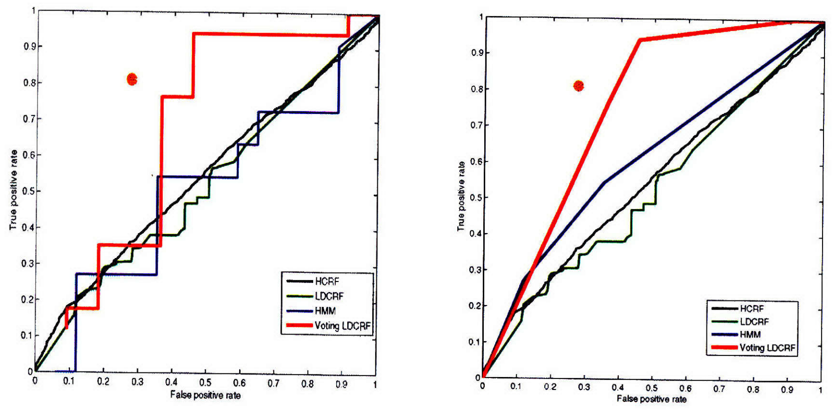


Figure A-12: ROC curves for detecting communication errors using the avatar partition of the Active-Avatar dataset. The raw unsmoothed curve is displayed the left figure while the convex-smoothed version of the same curve is displayed on the right.

Appendix B

Briefing Notes

This section presents the pre-experiment notes subjects had to read before they participated in our experiments described in Chapters 3 and 5.

B.1 Hidden Camera Dataset

You are asked to try out a restaurant application developed by one of our research groups in MIT. This application provides information about restaurants in the Greater Boston Area. It takes your speech as audio input. Whenever you are about to speak to the system, you will have to:

1. Face the microphone.
2. Click on the "Listen" Button on the bottom right.
3. Once you see the green words "Listening" displayed, you can begin speaking to the system.
4. As you are talking to the system, the words "Recording Started" are visually displayed simultaneously.
5. Once you are done speaking, the words in red "Done Listening" will be appear to signal the end of the recording.

Below is a sequence of questions you will need to ask the system for information about restaurants. *You have to ask the questions in SEQUENTIAL ORDER. You can only proceed to the next question when your current question is answered correctly.* You can rephrase your question to induce the correct response from the system. When you are done with the questions, you may leave the room. The information you need to retrieve are displayed below: Questions:

1. Ask for a display/listing of restaurants in Cambridge, or say something like, "Show me the restaurants in Cambridge."
2. Ask for a display/listing of pizza restaurants in Cambridge.
3. Ask for a display/listing of French restaurants in Boston.
4. Ask for information about the restaurant Les Zygomates.
5. Ask for information about the restaurant Les'palier.

B.2 Motivation Dataset

You are asked to try out a restaurant application developed by one of our research groups in MIT. This application provides information about restaurants in the Greater Boston Area. It takes your speech as audio input. Whenever you are about to speak to the system, you will have to:

1. Face the microphone and the web camera.
2. Click on the "Listen" Button on the bottom right.
3. Once you see the green words "Listening" displayed, you can begin speaking to the system.
4. As you are talking to the system, the words "Recording Started" are visually displayed simultaneously.
5. Once you are done speaking, the words in red "Done Listening" will be appear to signal the end of the recording.

During the whole interaction, please keep your head in front of the camera. You will be given a sequence of questions you will need to ask the system for information about restaurants, and you will have to write the answer on the sheet of paper provided next to the visual display. *You have to ask the questions in SEQUENTIAL ORDER. You can only proceed to the next question when your current question is answered correctly.* You can rephrase your question to induce the correct response from the system. On the right of the display lists the possible words you can say to the system. There will be three sessions of this experiment. The first session is a practice session, followed by two performance based sessions. For each session, once you complete the tasks, you may leave the room.

1st Session: Practice

Retrieve the following information from the system by asking questions, and write your answer using the pen and paper provided next to the computer.

Questions:

1. Ask for a display/listing of restaurants in Cambridge, or say something like, "Show me the restaurants in Cambridge."
2. Ask for a display/listing of pizza restaurants in Cambridge.
3. Ask for a display/listing of French restaurants in Boston.
4. Ask for the phone number of the restaurant Ten Tables.
5. Ask for information about the restaurant .

2nd Session: $10 reward

Retrieve the following information from the system by asking questions, and write your answer using the pen and paper provided next to the computer. Once you complete all the questions, and you may leave the room and you will be awarded $10.

Questions:

1. Ask for a display/listing of restaurants in Cambridge, or say something like, "Show me the restaurants in Cambridge."
2. Ask for a display/listing of pizza restaurants in Cambridge.
3. Ask for a display/listing of French restaurants in Boston.
4. Ask for the phone number of the restaurant Ten Tables.
5. Ask for information about the restaurant .

3rd Session: $10/$50 reward

Retrieve the following information from the system by asking questions, and write your answer using the pen and paper provided next to the computer. *You are given only 10 minutes to retrieve all the information* A timer in front of you will indicate your remaining time. If you complete all the tasks within 10 minutes, you will be awarded $50, else you will only get $10.

Questions:

1. Ask for a display/listing of restaurants in Cambridge, or say something like, "Show me the restaurants in Cambridge."
2. Find the phone number of the restaurant Caffe Umbra.
3. Find the phone number of the restaurant Les Zygomates.
4. Find the phone number of the restaurant Les'palier.
5. Find the nearest subway station near the restaurant King and I.

B.3 Synthetic Expressions

Nonverbal gestures: In the following scenarios, you are required to articulate the non-verbal facial expression of interest. You have to read the paragraphs to have an idea of the expression expected of you. When you are done reading, click on the record button to start recording, wait 2 secs, articulate your expression, wait another 2 secs, then click the same button to stop recording.

Practice Session:

Scene 1:

Scenario - You are interested in this guy/lady. You happen to get this rare chance to speak with him/her along the hallway. You are engaged in a really good conversation with him/her. Suddenly, a friend appeared and obnoxiously interrupted your conversation. You are annoyed but you did not say anything. The guy/lady did not see it, but someone observing you from afar can tell you are bit annoyed.

Express your annoyance without words: (click button and wait 2 sec)—-express non-verbally——- (wait 2 secs, click to stop recording)

Scene 2:

Scenario - You are in a TV game called "Jeopardy". The host asked you a question "What is Victor Zue's favorite color?" This is a $ 400 question, you are not sure of the answer and you did not want to make a random guess as a wrong answer would deduct $ 400. The audience could tell that you are uncertain about your answer from your facial expression.

Express your uncertainty without words: (click button and wait 2 sec) —-express non-verbally——- (wait 2 secs, click to stop recording)

Scene 3:

Scenario - You are in a TV game called "Jeopardy". This is the sudden death round. The host will ask a question and any participant could press the buzzer to answer it. You are very excited as you have a good chance to win this game. You listen intently as host is about to ask his question.

Show that you are listening intently without words: (click button and wait 2 sec) —-

express non-verbally——- (wait 2 secs, click to stop recording)

Nonverbal Positive Expressions

Scene 1: Scenario - You received news that you won $50000 in a lottery. You are in jubilant mood. Everyone could tell you something really good just happened to you.

Express your jubilance without words: (click button and wait 2 sec) —express non-verbally—— (wait 2 secs, click to stop recording)

Scene 2: Scenario - You received an email from your boss congratulating you of a job well done. You are very happy with his/her comments. You feel that your efforts have been reciprocated. You are very satisfied.

Express your satisfaction without words: (click button and wait 2 sec) —express non-verbally—— (wait 2 secs, click to stop recording)

Scene 3: Scenario - You are making an air ticket reservation on the phone with an automated speech recognition system. You always thought that the automated system had many difficulties recognizing your speech correctly and you are always hesitant about using them. Surprisingly, in this conversation, the whole reservation process went very smoothly. There were no errors in recognition of your speech. You were satisfied with the performance of the system.

Express your satisfaction without words: (click button and wait 2 sec) —express non-verbally—— (wait 2 secs, click to stop recording)

Scene 4:

Scenario - You are in a TV game called "Jeopardy". This is the sudden death round. The host will ask a question and any participant could press the buzzer to answer it. You are very excited as you have a good chance to win this game. You listen intently as host is about to ask his question.

Show that you are listening intently without words: (click button and wait 2 sec) —express non-verbally—— (wait 2 secs, click to stop recording)

Nonverbal Negative Expressions

Scene 1: Scenario - You are making an air ticket reservation on the phone with an automated speech recognition system. As you listen intently at the system reply to your query, you hear an incorrect reply. This is the first incorrect reply you have heard. You are annoyed as you have to repeat your query.

Express your annoyance without words: (click button and wait 2 sec) —-express non-verbally—— (wait 2 secs, click to stop recording)

Scene 2: Scenario - You are making an air ticket reservation on the phone with an automated speech recognition system. As you listen intently at the system reply to your query, you hear an incorrect reply. This is the 10th time the system has given you an incorrect reply. You are livid. You are angry beyond words and you wish to throw something at the phone.

Express your anger without words: (click button and wait 2 sec) —-express non-verbally—— (wait 2 secs, click to stop recording)

Scene 3: Scenario - You are in a TV game called "Jeopardy". The host asked you a question "What is Victor Zue's favorite color?" This is a $ 400 question, you are not sure of the answer and you did not want to make a random guess as a wrong answer would deduct $ 400. The audience could tell that you are uncertain about your answer from your facial expression.

Express your uncertainty without words: (click button and wait 2 sec) —-express non-verbally—— (wait 2 secs, click to stop recording)

Scene 4: Scenario - You are in a TV game called "Jeopardy". This is the sudden death round. The host will ask a question and any participant could press the buzzer to answer it. The winner of this round gets to walk away with $ 100000. You are very excited as you think you have a good chance to win this game. The host asked his question. You figured out the answer and quickly pressed the buzzer. Unfortunately some other participant managed to press the buzzer before you by just a split second and that person won the money. You could not hide your disappointment as you knew how close you were in getting the huge prize.

Express your disappointment without words: (click button and wait 2 sec) —-express non-verbally——- (wait 2 secs, click to stop recording)

B.4 Active vs Passive Inputs

You are about to make airline reservations with a conversational system. You will be given the source city, destination city, departure time and the day of your departure. You are supposed to speak to the system to complete the reservation. The system will guide you through to make the reservation. For each task that is successfully completed, you will hear the sentence, "Your reservation is now confirmed. Good bye!"

Here are four reservations you have to make:

1. From Boston to San Francisco on Monday at 9 am.
2. From Boston to San Diego on Monday at 9 am.
3. From Allston to San Francisco on Monday at 9 am.
4. From Boston to San Pablo on Tuesday at 9 am.

You will need to repeat these 4 reservations under two different settings as instructed by the experimenter.

Bibliography

[1] J. Ang, R. Dhillon, A. Krupski, E. Shriberg and A. Stolcke. Prosody-Based Automatic Detection of Annoyance and Frustration in Human-Computer Dialog. In *ICSLP*, 2002.

[2] A.B. Ashraf, S. Lucey, J.F. Cohn, T. Chen, Z. Ambadar, K. Prkachin, P. Solomon and B.J. Theobald. The painful face: pain expression recognition using active appearance models. In *Int'l Conf. on Multimodal Interfaces*, 2007.

[3] D. Bansal and M.K. Ravishankar, New Features for Confidence Annotation In *Proc. of Int'l Conf. on Speech and Language Processing(ICSLP)*, 1998.

[4] C. Barber, D. Dobkin and H. Huhdanpaa. The quickhull algorithm for convex hull. Technical Report GCG53, University of Minnesota, 1996.

[5] P. Barkhuysen, E. Krahmer and M. Swerts. Audiovisual Perception of Communication Problems. In *Speech Prosody*, 2004.

[6] M.S. Bartlett, G. Littlewort, P. Braathen, T.J. Sejnowski and J.R. Movellan. A prototype for automatic recognition of spontaneous facial actions. In *Advances in Neural Information Processing Systems*, Volume15, pages 1271–1278, 2003.

[7] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel and J. Movellan. Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. In *IEEE International Conference on Computer Vision and Pattern Recognition* , pages 568–573, 2005.

[8] M.S. Bartlett, G. Littlewort,M.G. Frank,C. Lainscsek,I. Faseland J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 223–230, 2006.

[9] C. Mario Christodias,K. Saenko,L.-P. Morency and T. Darrell. Co-Adaptation of Audio-Visual Speech and Gesture Classifiers. In *Int. Conf. on Multimodal Interfaces*, 2006.

[10] P. Boersma. Praat, a system for doing phonetics by computer. In *Glot International*, Volume 5, Number 9/10, pages 341–345, 2001.

[11] P. Boersma. Accurate Short-Term Analysis of the Fundamental Frequency and the harmonis-to-noise ratio of a sampled sound. In *IFA*, Chapter 17, 1983.

[12] J. Lafferty, A. McCallum and F. Pereira. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In *Int'l Conf. on Machine Learning*, 2001.

[13] G. Caridakis,L. Malatesta,L. Kessous,N. Amir,A. Paouzaiouand K. Karpouzis. Modeling Naturalistic Affective States via Facial and Vocal Expression Recognition. In *Int. Conf. on Multimodal Interfaces*, pages 146–154, 2006.

[14] J. Cassell. Embodied Conversational Agents: Representation and Intelligence in User Interface. AI Magazine, Winter , 22(3): 67–83, 2001.

[15] Y. Chang,C. Hu,R. FerisandM. Turk. Manifold based analysis of facial expression. In *Proc. Computer Vision and Pattern Recognition*, 2004.

[16] L. Chen and T. S. Huang. Emotional expressions in audiovisual human computer interaction. In *ICME*, 2000.

[17] D. G. Childers. Modern Spectrum Analysis. In *IEEE Press*, pages 252–255, 1978.

[18] I. Cohen, N. Sebe, A. Garg, L. Chen, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. In *CVIU*, volume 91(1-2), pages 160–187, 2003.

[19] J.F. Cohn and K.L. Schmidt. The timing of Facial Motion in Posed and Spontaneous Smiles. In *International Journal of Wavelets, Multiresolution and Information Processing*, 2,pages 1-12, 2004.

[20] J.F. Cohn, L.I. Reed,Z. Ambadar,J. XiaoandT. Moriyama. Automatic Analysis and recognition of brow actions and head motion in spontaneous facial behavior. In *Int. Conf. on Systems, Man and Cybernetics*, 1, pages 610–616, 2004.

[21] J.F. Cohn. Foundations of human computing: Facial expression and emotion. In *Int'l Conf. on Multimodal Interfaces*, 2006.

[22] S. Cox and R. Rose, Confidence Measures for the Switchboard Database. In *ICASSP*, 1996.

[23] R. Cowie,E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J. Taylor. Emotion recognition in human-computer interaction. In *IEEE Signal Processing Magazine*, volume 18, pages 32–80, 2001.

[24] C. Sminchisescu, A. Kanaujia, Z. Li and D. Metaxas. Conditional Models for Contextual Human Motion Recognition. In *Int'l Conf. on Computer Vision*, 2005.

[25] D. Demirdjian and T. Darrell. 3-D Articulated Pose Tracking for Untethered Deictic Reference. In *Int'l Conf. on Multimodal Interfaces*, 2002.

[26] P. Ekman. Emotion in the Human Face. Cambridge University Press,1982.

[27] P. Ekman, W. V. Friesen andJ. C. Hager. Facial Action Coding System. A Human Face. Salt Lake City, Utah, 2002.

[28] P. Ekman and E.L.Rosenberg. What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system. In 2^{nd} edition, Oxford University Press, 2003.

[29] R. El. Kaliouby and P. Robinson. Realtime Inference of complex mental states from facial expression and head gestures. In *Computer Vision and Pattern Recognition Workshop*, Volume3,154, 2004.

[30] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. In *Patt. Recogn.*, volume 36, pages 259–275, 2003.

[31] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. Kluwer Academic Publishers, Netherlands, 2004.

[32] P. Flach and W. Shaoming. Repairing Concavities in ROC Curves. In *Proc. Workshop on Computational Intelligence*, United Kingdom, 2003.

[33] F. Fragopanagos and J.G. Taylor. Emotion recognition in human-computer interaction. In *Neural Networks*, 18:389-405, 2005.

[34] A. Gruenstein, S. Seneff and C. Wang. Scalable and Portable Web-Based Multimodal Dialogue Interaction with Geographical Databases. In *INTERSPEECH*, 2006.

[35] H. Gu and Q. Ji. An Automated Face Reader for Fatigue Detection. In *Int'l Conf. Face and Gesture Recogn.*, 2004.

[36] G. Guo and C.R. Dyer. Learning from examples in the small sample case face expression recognition. In *IEEE Trans. Systems, Man and Cybernetics*-PartB,volume 35,No.3,pages 477–488, 2005.

[37] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. Hidden conditional random fields for phone classification. In *INTERSPEECH*, 2005.

[38] Q. Ji, P. Lan and C. Looney. A probabilistic framework for modeling and real-time monitoring human fatigue. In *IEEE SMC-Part A*, Vol.36, No.5, pages 862-875, 2006.

[39] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24(6), pages 381–395, 1981.

[40] H. P. Graf, E. Cosatto, V. Strom, F. J. Huang. Visual prosody: facial movements accompanying speech. In *Automatic Face and Gesutre Recognition*, 2002.

[41] J. Hirschberg, D. Litman and M. Swerts. Generalizing prosodic prediction of speech recognition errors In *Proc. Int'l Conf. on Speech and Language Processing(ICSLP)*, 2000.

[42] J. Hirschberg, D. Litman and M. Swerts. Identifying User Corrections Automatically in Spoken Dialogue Systems. In *2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, 2001.

[43] X. Huang, A. Acero, and H. Hon. *Spoken Language Processing*. Prentice Hall, Upper Sadle River, New Jersey, 2001.

[44] C. Hundtofte, G. Hager and A. Okamura. Building a task language for segmentation and recognition of user input to cooperative manipulation systems. In *IEEE Virtual Reality Conference (HAPTICS 2002)*, Orlando, Florida, 2002.

[45] A. Kapoor and R.W. Picard. Multimodal affect recognition in learning environment. In *ACM Int'l Conf. on Multimedia*, 677-682, 2005.

[46] A. Kapoor, W. Burleson and R.W. Picard. Automatic prediction of frustration. In *Int'l Journal of Human-Computer Studies.*,volume65(8), pages 724–736, 2007.

[47] K. Karpouzis, G. Caridakis, L. Kessous, N. Amir, A. Raouzaiou, L. Malatesta and S. Kollias. Modeling naturalistic affective states via facial, vocal, and bodily expression recognition. In *Lecture Notes in ArtificialIntelligence*, vol. 4451, pages91-112, 2007.

[48] J. Kittler, M. Hatef, R. Duin and J. Matas. On Combining Classifiers. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pages 226–239, 1998.

[49] S. Kumar and M. Herbert. Discriminative random fields: A framework for contextual interaction in classification. In *Int'l Conf. on Computer Vision*, 2003.

[50] G.C. Littlewort, M.S. Bartlett and K. Lee. Faces of pain: Automated measurement of spontaneous facial expressions of genuine and posed pain. In *Int'l Conf. on Multimodal Interfaces*, 2007.

[51] D. Litman, J. Hirschberg and M. Swerts. Predicting user reactions to system error. In *ACL*, 2001.

[52] D. Litman, J. Hirschberg and M. Swerts. Generalizing prosodic prediction of speech recognition errors. In *Int'l Conf. on Speech and Language Processing(ICSLP)*, 2000.

[53] S. Lucey, A.B. Ashraf and J.F. Cohn. Investigating Spontaneous Facial Action Recognition through AAM Representations of the Face. In *Face Recognition*, I-Tech Education and Publishing, Vienna, Austria, pages 275–286, 2007.

[54] D. Massaro. Speech Perception By Ear and Eye. Lawrence Erlbaum Associates, Hillsdale, NJ,USA, 1987.

[55] Quasi-Newton Optimization Toolbox in MATLAB. http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/f137.html.

[56] L.-P. Morency, A. Rahimi and T. Darrell. Adaptive view-based appearance models. In *Int'l Conf. on Computer Vision and Pattern Recognition*, pages 803–810, 2003.

[57] L.-P. Morency, C. Sidner, C. Lee and T. Darrell. Contextual Recognition of Head Gestures. In *Int'l Conf. on Multimodal Interfaces*, 2005.

[58] L. Morency, A. Quattoni and T. Darrell. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In *CVPR*, 2007.

[59] S. L. Oviatt and R. VanGent. Error Resolution During Multimodal Human-Computer Interaction. In *Speech Communication*, 1998.

[60] P. Pal, A. Iyer and R.E. Yantorno. Emotion detection from infant facial expressions and cries. In *Proc. Intl Conf. Acoustics, Speech and Signal Processing*,2, pages721–724, 2006.

[61] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. In *IEEE Trans. on PAMI*, volume 22(12), pages 1424–1445, 2000.

[62] M. Pantic and L.J.M. Rothkrantz. Expert System for Automatic Analysis of Facial Expression In *Image and Vision Computing Journal*, vol. 18, no. 11, pp. 881-905, 2000.

[63] M. Pantic and M.S. Bartlett. Machine Analysis of Facial Expressions. In *Face Recognition*, K. Delac and M. Grgic, Eds., Vienna, Austria: I-Tech Education and Publishing, 2007.

[64] M. Pantic and I. Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments form face profile imgae sequences. In *IEEE Trans. Systems, Man and Cybernetics*-PartB,volume36,No.2,pages 433–449, 2006.

[65] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988.

[66] S. Petridis and M. Pantic. Audiovisual discrimination between laughter and speech. In *Intl Conf. Acoustics, Speech, and Signal Processing*, 2008.

[67] A. Quattoni and M. Collins and T. Darrell. Conditional random fields for object recognition. In NIPS, 2004.

[68] A. Quattoni, S. Wang, L.-P. Morency and M. Collins and T. Darrell. Hidden-state Conditional Random Fields. In PAMI, 2007.

[69] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–286, 1989.

[70] K. Schmidt, J. F. Cohn and Y. L. Tian. Signal characteristics of spontaneous facial expressions: Automatic movement in solitary and social smiles. In *Biological Psychology*, 2003.

[71] N. Sebe, M.S. Lew, I. Cohen, Y. Sun, T. Gevers and T.S. Huang. Authentic Facial Expression Analysis. In *Int. Conf. on Automatic Face and Gesture Recognition*, 2004.

[72] E. Shriberg, E. Wade and P. Price. Human-machine problem solving using spoken language systems (SLS): factors affecting performance and user satisfaction. In *Proc. of the DARPA Speech and Natural Language Workshop*, 1992.

[73] G. Skantze and J. Edlund. Early error detection on word level. In *Proceedings of ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, 2004.

[74] M. Song, J. Bu, C. Chen and N. Li. Audio-visual based emotion recognition - a new approach. In *CVPR*, 2004.

[75] Q. Summerfield. Some preliminaries to a comprehensive account of audio-visual speech perception. In *Hearing by Eye*, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, pages 3-51, 1987.

[76] H. Tao and T.S. Huang. Explanation based facial motion tracking using a piecewise Bezier volume deformation.mode In *CVPR*,volume 1, pages611–617, 1999.

[77] Y.L. Tian, T. Kanade and J.F. Cohn. Facial expression analysis. In *Handbook of Face Recognition*, S.Z. Li and A.K. Jain (Eds.), Springer, New York, USA, pages 247–276, 2005.

[78] A. Teeters, R.E. Kaliouby and R.W. Picard. Self Cam: Feedback From What Would Be Your Social Partner. In *ACM SIGGRAPH*, Research Posters, pages 138, 2006.

[79] K.R. Thrisson and J. Cassell. Why Put an Agent in a Human Body: The Importance of Communicative Feedback in Human-Humanoid Dialogue. In *Lifelike Computer Characters*, Snowbird, Utah, October 8-11, 1996.

[80] M. Valstar, M. Pantic, Z. Ambadar and J.F. Cohn. Spontaneous vs. Posed Facial Behavior: Automatic Analysis of Brow Actions. In *Int'l Conf. on Multimodal Interfaces*, 2006.

[81] M. Valstar, H. Gunes and M. Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Int'l Conf. on Multimodal Interfaces*, 2007.

[82] P. Viola and M.J. Jones. Robust Real-Time Face Detection In *IJCV*, volume 57(2), pages 137–154, 2004.

[83] J. Wang, L. Yin, X. Wei and Y. Sun. 3D Facial Expression Recognition Based on Primitive Surface Feature Distribution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2:1399-1406, 2006.

[84] S. Wang and D. Demirdjian. Inferring body pose using speech content. In *Int'l Conf. on Multimodal Interfaces*, Trento, Italy, 2005.

[85] S. Wang, A. Quattoni, L.P. Morency, D. Demirdjian and T. Darrell. Hidden Conditional Random Fields for Gesture Recognition In *Proceedings of Computer Vision and Pattern Recognition*, 2006.

[86] S. Wang, D. Demirdjian, H. Kjellstrom and T. Darrell. Multimodal Communication Error Detection For Driver-Car Interaction. In *Int'l Conf. on Informatics, Control, Automation and Robotics*, Angers, France, 2007.

[87] S. Wang, D. Demirdjian and T. Darrell. Communication Error Detection using facial expressions during the System's turn. In *Int'l Conf. on Multimodal Interfaces*, Nagoya, Japan, 2007.

[88] Z. Wen and T.S. Huang. CapturingSubtleFacialMotionsin3DFaceTracking In *Int'l. Conf. on Computer Vision*, pages 1343–1350, 2003.

[89] J. Whitehill and C.W. Omlin. Haar features for FACS AU recognition. In *Int'l. Conf. on Automatic Face and Gesture Recognition*, pages 217–222, 2006.

[90] J. Xiao, T. Moriyama, T. Kanade and J.F. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. In *Int'l. Journal of Imaging Systems and Technology*, Vol.13, No.1, pages 85–94, 2003.

[91] M. Yeasin, B. Bullot and R. Sharma. Recognition of facial expressions and measurement of levels of interest from video. In *IEEE Trans. On Multimedia*, Vol.8, No.3, June, pages 500–507, 2006.

[92] L. Yin, X. Wei, Y. Sun, J. Wang and M.J. Rosato. A 3D facial expression database for facial behavior research. In *Int'l. Conf. on Automatic Face and Gesture Recognition*, pages 211–216, 2006.

[93] Y. Yoshitomi, S. Kim, T. Kawano and T. Kitazoe. Effect of sensor fusion for recognition of emotional states using voice, face image and thermal image of face. In *IEEE International Workshop on Robot and Human Interactive Communication*, 2000.

[94] Z. Zeng, J. Tu, M. Liu, T. Zhang, N. Rizzolo, Z. Zhang, T. S. Huang, D. Roth and S. Levinson. Bimodal HCI-related Affect Recognition. In *Int'l Conf. on Multimodal Interfaces*, 2004.

[95] Z. Zeng, Y. Fu, G.I. Roisman, Z. Wen, Y. Hu and T.S. Huang. Spontaneous Emotional Facial Expression Detection. In *Journal of Multimedia*, 1(5):1-8, 2006.

[96] Z. Zeng,Y. Hu,G.I. Roisman,Z. Wen,Y. FuandT.S. Huang. Audio-visual Spontaneous Emotion Recognition. In *Artificial Intelligence for Human Computing*, Eds: T.S. Huang, A. Nijholt, M. Pantic and A. Pentland, Springer, pages 72–90, 2007.

[97] Z. Zeng, M. Pantic, G.I. Roiseman and T.S. Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2007.