FACIAL EXPRESSION RECOGNITION: A FULLY INTEGRATED APPROACH

Roberto Valenti, Nicu Sebe, Theo Gevers

Faculty of Science, University of Amsterdam, The Netherlands
{rvalenti,nicu,gevers}@science.uva.nl

SUBMITTED TO VMDL07 AND ICIAP07

ABSTRACT

The most expressive way humans display emotions is through facial expressions. Humans detect and interpret faces and facial expressions in a scene with little or no effort. Still, development of an automated system that accomplishes this task is rather difficult. There are several related problems: detection of an image segment as a face, facial features extraction and tracking, extraction of the facial expression information, and classification of the expression (e.g., in emotion categories). In this paper, we present our fully integrated system which performs these operations accurately and in real time and represents a major step forward in our aim of achieving a human-like interaction between the man and machine.

1. INTRODUCTION

In the recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. This emerging field has been a research interest for scientists from several different scholastic tracks, i.e., computer science, engineering, psychology, and neuroscience. These studies focus not only on improving computer interfaces, but also on improving the actions the computer takes based on feedback from the user. Feedback from the user has traditionally been given through the keyboard and mouse. Other devices have also been developed for more application specific interfaces, such as joysticks, trackballs, datagloves, and touch screens. The rapid advance of technology in recent years has made computers cheaper and more powerful, and has made the use of microphones and PC-cameras affordable and easily available. The microphones and cameras enable the computer to “see” and “hear,” and to use this information to act. A good example of this is the “Smart-Kiosk” [9].

Psychologists and engineers alike have tried to analyze facial expressions in an attempt to understand and categorize these expressions. This knowledge can be for example used to teach computers to recognize human emotions from video images acquired from built-in cameras. In some applications, it may not be necessary for computers to recognize emotions. For example, the computer inside an automatic teller machine or an airplane probably does not need to recognize emotions. However, in applications where computers take on a social role such as an “instructor,” “helper,” or even “companion,” it may enhance their functionality to be able to recognize users’ emotions. In her book, Picard [17] suggested several applications where it is beneficial for computers to recognize human emotions. For example, knowing the user’s emotions, the computer can become a more effective tutor. Synthetic speech with emotions in the voice would sound more pleasing than a monotonous voice. Computer “agents” could learn the user’s preferences through the users’ emotions. Another application is to help the human users monitor their stress level. In clinical settings, recognizing a person’s inability to express certain facial expressions may help diagnose early psychological disorders.

There is a vast body of literature on emotions. The multifaceted nature prevents a comprehensive review, we will review only what is essential in supporting this work. Recent discoveries suggest that emotions are intricately linked to other functions such as attention, perception, memory, decision making, and learning. This suggests that it may be beneficial for computers to recognize the human user’s emotions and other related cognitive states and expressions.

Ekman and Friesen [7] developed the Facial Action Coding System (FACS) to code facial expressions where movements on the face are described by a set of action units (AUs). Each AU has some related muscular basis. This system of coding facial expressions is done manually by following a set of prescribed rules. The inputs are still images of facial expressions, often at the peak of the expression. Ekman’s work inspired many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis and recognition has used the “basic expressions” (i.e., happiness, surprise, fear, disgust, sad, and anger) or a subset of them. The two recent surveys in the area [16, 8] provide an in depth review of the existing approaches towards automatic facial expression recognition. These methods are similar in that they first extract some features from the images, then these features are used as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted from the video images and in the classifiers used to distinguish between the different emotions.

This paper presents our real time facial expression recognition system [11] which uses a facial features detector and a model based non-rigid face tracking algorithm to extract motion features that serve as input to a Bayesian network classifier used for recognizing the different facial expressions.

2. FACIAL EXPRESSION RECOGNITION SYSTEM

Our real time facial expression recognition system is composed of facial feature detector and a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to a Bayesian network classifier. We describe these components in the following sections. A snap shot of the system, with the face tracking and recognition result is shown in Figure 4.

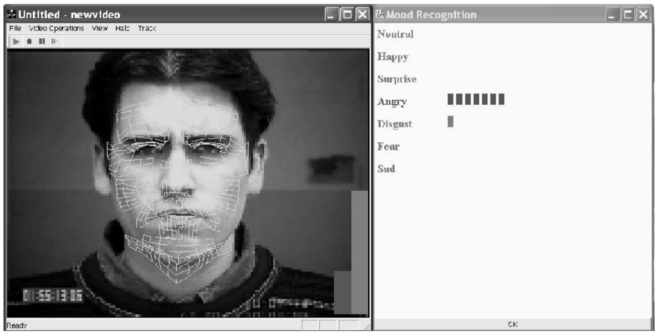


Fig. 1. A snap shot of our realtime facial expression recognition system. On the right side is a wireframe model overlayed on a face being tracked. On the left side the correct expression, Angry, is detected.

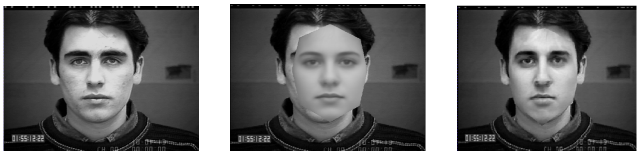
2.1. Facial Feature Detection

For facial feature detection we consider the idea of using the knowledge of a face detector inside an active appearance model [4] (AAM), by employing what we call a 'virtual structuring element' (VSE), which limits the possible settings of the AAM in an appearance-driven manner. We propose this visual artifact as a good solution for the background linking problems and respective generalization problems of basic AAMs.

The main idea of using an AAM approach is to learn the possible variations of facial features exclusively on a probabilistic and statistical basis of the existing observations (i.e., which relation holds in all the previously seen instances of facial features). This can be defined as a combination of shapes and appearances. At the basis of AAM search is the idea to treat the fitting procedure of a combined shape-appearance model as an optimization problem in trying to minimize the difference vector between the image $\mathbf{I}$ and the generated model $\mathbf{M}$ of shape and appearance: $\delta\mathbf{I} = \mathbf{I} - \mathbf{M}$. Cootes et al. [4] observed that each search corresponds to a similar class of problems where the initial and the final model parameters are the same. This class can be learned offline (when we create the model) saving high-dimensional computations during the search phase.

Learning the class of problems means that we have to assume a relation $\mathbf{R}$ between the current error image $\delta\mathbf{I}$ and the needed adjustments in the model parameters m. The common assumption is to use a linear relation: $\delta m = \mathbf{R}\delta\mathbf{I}$. Despite the fact that more accurate models were proposed [14], the assumption of linearity was shown to be sufficiently accurate to obtain good results. To find $\mathbf{R}$ we can conduct a series of experiments on the training set, where the optimal parameters m are known. Each experiment consists of displacing a set of parameters by a know amount and in measuring the difference between the generated model and the image under it. Note that when we displace the model from its optimal position and we calculate the error image $\delta\mathbf{I}$, the image will surely contain parts of the background.

What remains to discuss is an iterative optimization procedure that uses the found predictions. The first step is to initialize the mean model in an initial position and the parameters within the reach of the parameter prediction range (which depends on the perturbation used during training). Iteratively, a sample of the image under the initialization is taken and compared with the model instance. The differences between the two appearances are used to predict the set of parameters that would perhaps improve the similarity. In case a prediction fails to improve the similarity, it is possible to damp or amplify the prediction several times and maintain the one with the best result. For an overview of some possible variations to the original AAMs algorithm refer to [5]. An example of the AAM search is shown in Fig. 2 where a model is fitted to a previously unseen face.



(a) Unseen face (b) Initialization (c) Converged model

Fig. 2. Results of an AAM search on an unseen face

One of the main drawbacks of the AAM is coming from its very basic concept: when the algorithm learns how to solve the optimization offline, the perturbation applied to the model inevitably takes parts of the background into account. This means that instead of learning how to generally solve the class of problems, the algorithm actually learns how to solve it only for the same or similar background. This makes AMMs domain-specific, that is, the AAM trained for a shape in a predefined environment has difficulties when used on the same shape immersed in a different environment. Since we always need to perturbate the model and to take into account the background, an often used idea is to constrain the shape deformation within predefined boundaries. Note that a shape constraint does not adjust the deformation, but will only limit it when it is found to be invalid.

To overcome these deficiencies of AAMs, we propose a novel method to visually integrate the information obtained by a face detector inside the AAM. This method is based on the observation that an object with a specific and recognizable feature would ease the successful alignment of its model.

Since faces have many highly relevant features, erroneously located ones could lead the optimization process to converge to local minima. The novel idea is to add a virtual artifact in each of the appearances in the training and the test sets, that would inherently prohibit some deformations. We call this artifact a **virtual structuring element** (or **VSE**) since it adds structure in the data that was not available otherwise. In our specific case, this element adds visual information about the position of the face. If we assume that the face detector successfully detects a face, we can use that information to build this artifact. As the face detector we use the one proposed by Viola and Jones [19].

After experimenting with different VSEs, we propose the following guideline to choose a good VSE. We should choose a VSE that: (1) Is big enough to steer the optimization process; (2) Does not create additional uncertainty by covering relevant features (e.g., the eyes or nose); (3) Scales accordingly to the dimension of the detected face; and (4) Completely or partially removes the high variance areas in the model (e.g., background) with uniform ones.

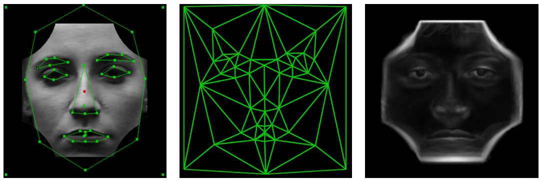


Fig. 3. The effect of a virtual structuring element to the annotation, appearance, and variance (white indicates a larger variance)

In the used VSE, a black frame with width equal to 20% of the size of the detected face is built around the face itself. Besides the regular markers that capture the facial features (see Fig. 3 and [11] for details) four new markers are added in the corners to stretch the convex hull of the shape to take in consideration the structuring element. Around each of those four points, a black circle with the radius of one third of the size of the face is added. The resulting annotation, shape, and appearance variance are displayed in Fig. 3. Note that in the variance map the initialization variance of the face detector is automatically included in the model (i.e., the thick white border delimitating the VSE).

This virtual structuring element visually passes information between the face detection and the AAM. The obtained facial features are used in our system as inputs into the next block dealing with face and facial feature tracking.

2.2. Face and Facial Feature Tracking

The face tracking we use in our system is based on a system developed by Tao and Huang [18] called the piecewise Bezier volume deformation (PBVD) tracker. The face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed (see Fig. 1). A generic face model is warped to fit the detected facial features. The face model consists of 16 surface patches embedded in Bezier volumes. The surface patches defined this way are guaranteed to be continuous and smooth.

Given a set of $n+1$ control points $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n$, the corresponding Bezier curve (or Bernstein-Bezier curve) is given by

$$\mathbf{x}(u) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(u) = \sum_{i=0}^{n} \mathbf{b}_i \binom{n}{i} u^i (1-u)^{n-i} \qquad (1)$$

where the shape of the curve is controlled by the control points $\mathbf{b}_i$ and $u \in [0,1]$. As the control points are moved, a new shape is obtained according to the Bernstein polynomials $B_i^n(u)$ in Eq. (1). The displacement of a point on the curve can be described in terms of linear combinations of displacements of the control points.

The Bezier volume is a straight-forward extension of the Bezier curve and is defined by Eq. (2) written in matrix form

$$\mathbf{V} = \mathbf{B}\mathbf{D}, \qquad (2)$$

where $\mathbf{V}$ is the displacement of the mesh nodes, $\mathbf{D}$ is a matrix whose columns are the control point displacement vectors of the Bezier volume, and $\mathbf{B}$ is the mapping in terms of Bernstein polynomials. In other words, the change in the shape of the face model can be described in terms of the deformations in $\mathbf{D}$.

Once the model is constructed and fitted, head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be tracked. First the 2D image motions are measured using template matching between frames at different resolutions. Image templates from the previous frame and from the very first frame are both used for more robust tracking. The measured 2D image motions are modeled as projections of the true 3D motions onto the image plane. From the 2D motions of many points on the mesh, the 3D motion can be estimated by solving an overdetermined system of equations of the projective motions in the least squared sense.

The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face, defined in terms of the Bezier volume control parameters. We refer to these motions vectors as motion-units (MU's). Note that they are similar but not equivalent to Ekman's AU's, and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The MU's used in the face tracker are shown in Figure 4 (a).

Each facial expression is modeled as a linear combination of the MU's:

$$\mathbf{V} = \mathbf{B}\left[\mathbf{D}_0\mathbf{D}_1\ldots\mathbf{D}_m\right]\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_m \end{bmatrix} = \mathbf{BDP} \tag{3}$$

where each of the $\mathbf{D}_i$ corresponds to an MU, and the p_i are the corresponding magnitudes (or coefficients) of each deformation. The overall motion of the head and face is

$$\mathbf{R}(\mathbf{V}_0 + \mathbf{BDP}) + \mathbf{T} \tag{4}$$

where $\mathbf{R}$ is the 3D rotation matrix, $\mathbf{T}$ is the 3D translation matrix, and $\mathbf{V}_0$ is the initial face model.

The MU's are used as the basic features for the classification scheme described in the next section.

2.3. Learning the "Structure" of the Facial Features

The use of Bayesian networks as the classifier for recognizing facial expressions has been first suggested by Chen et al. [2], who used Naive Bayes (NB) classifiers and who recognize the facial expressions from the same MUs.

When modeling the described facial motion features, it is very probable that the conditional independence assumption of the Naive Bayes classifier is incorrect. As such, learning the dependencies among the facial motion units could potentially improve classification performance, and could provide insights as to the "structure" of the face, in terms of strong or weak dependencies between the different regions of the face, when subjects display facial expressions. Our first attempt [11] was to learn a Tree-augmented Bayesian Network (TAN) classifier. In the TAN classifier structure the class node has no parents and each feature has as parents the class node and at most one other feature, such that the result is a tree structure for the features. Moreover, with unlabeled data, our previous analysis [3] indicated that learning the structure is even more critical compared to the supervised case. As such, in our system we employ different classification methods, in particular we are interested in using the NB, the TAN and Stochastic Structure Search (SSS) algorithms [3]. We briefly present the SSS algorithm below.

In our approach, instead of trying to estimate the best a-posteriori probability, we try to find the structure that minimizes the probability of classification error directly. The basic idea of this approach is that, since we are interested in finding a structure that performs well as a classifier, it would be natural to design an algorithm that use classification error as the guide for structure learning. Here, we can further leverage on the properties of semi-supervised learning: we know that unlabeled data can indicate incorrect structure through degradation of classification performance, and we also know that classification performance improves with the correct structure. Thus, a structure with higher classification accuracy over another indicates an improvement towards finding the optimal classifier.

To learn the structure using classification error, we must adopt a strategy of searching through the space of all structures in an efficient manner while avoiding local maxima. As we have no simple closed-form expression that relates structure with classification error, it would be difficult to design a gradient descent algorithm or a similar iterative method. Even if we did that, a gradient search algorithm would be likely to find a local minimum because of the size of the search space.

First we define a measure over the space of structures which we want to maximize:

Definition 1 *The* inverse error measure *for structure S' is*

$$inv_e(S') = \frac{\frac{1}{p_{S'}(\hat{c}(X) \neq C)}}{\sum_S \frac{1}{p_S(\hat{c}(X) \neq C)}}, \tag{5}$$

where the summation is over the space of possible structures, X represents the MU's vector, C is the class space, $\hat{c}(X)$ represents the estimated class for the vector X, and $p_S(\hat{c}(\mathbf{X}) \neq C)$ is the probability of error of the best classifier learned with structure S.

We use Metropolis-Hastings sampling [15] to generate samples from the inverse error measure, without having to ever compute it for all possible structures. For constructing the Metropolis-Hastings sampling, we define a neighborhood of a structure as the set of directed acyclic graphs to which we can transit in the next step. Transition is done using a predefined set of possible changes to the structure; at each transition a change consists of a single edge addition, removal, or reversal. We define the acceptance probability of a candidate structure, S^{new}, to replace a previous structure, S^t as follows:

$$\min\left(1, \left(\frac{inv_e(S^{new})}{inv_e(S^t)}\right)^{1/T} \frac{q(S^t|S^{new})}{q(S^{new}|S^t)}\right) = \min\left(1, \left(\frac{p_{S^t}}{p_{S^{new}}}\right)^{1/T} \frac{N_t}{N_{new}}\right) \tag{6}$$

where $q(S'|S)$ is the transition probability from S to S' and N_t and N_{new} are the sizes of the neighborhoods of S^t and S^{new}, respectively; this choice corresponds to equal probability of transition to each member in the neighborhood of a structure. This choice of neighborhood and transition probability creates a Markov chain which is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [13].

T is used as a temperature factor in the acceptance probability. Roughly speaking, T close to 1 would allow acceptance of more structures with higher probability of error than previous structures. T close to 0 mostly allows acceptance of structures that improve probability of error. A fixed T amounts to changing the distribution being sampled by the MCMC, while a decreasing T is a simulated annealing run, aimed at finding the maximum of the inverse error measures. The rate of decrease of the temperature determines the rate of convergence. Asymptotically in the number of data, a logarithmic decrease of T guarantees convergence to a global maximum with probability that tends to one [10].

The SSS algorithm, with a logarithmic cooling schedule T, can find a structure that is close to minimum probability of error. We estimate the classification error of a given structure using the labeled training data. Therefore, to avoid overfitting, we add a multiplicative penalty term derived from the Vapnik-Chervonenkis (VC) bound on the empirical classification error. This penalty term penalizes complex classifiers thus keeping the balance between bias and variance (for more details we refer the reader to [3]).

3. EXPERIMENTAL ANALYSIS

In the following experiments we compare the different approaches mentioned above (i.e., Naive Bayes (NB), TAN, and SSS) for facial expression recognition. We present here only experiments where all the data are labeled. For a detalied investigation on the effect of using both labeled and unlabeled data we direct the interested reader to our previous work reported in [3]. We first perform person dependent tests, also comparing different assumptions on the Bayesian Network structure. Then we perform person independent tests, showing that the best performance is attained with the SSS algorithm.

We use two different databases. The first database was collected by Chen and Huang [2] and is a database of subjects that were instructed to display facial expressions corresponding to the six types of emotions. All the tests of the algorithms are performed on a set of five people, each one displaying six sequences of each one of the six emotions, starting and ending at the Neutral expression. The video sampling rate was 30 Hz, and a typical emotion sequence is about 70 samples long ($\sim$2s). The second database is the Cohn-Kanade database [12]. For each subject there is at most one sequence per expression with an average of 8 frames for each expression.

We measure the accuracy with respect to the classification result of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including neutral). This manual labeling can introduce some ‘noise’ in our classification because the boundary between Neutral and the expression of a sequence is not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral. A different labeling scheme is to label only some of the frames that are around the peak of the expression leaving many frames in between unlabeled. We did not take this approach because a real-time classification system would not have this information available to it.

3.1. Person-dependent Tests

A person-dependent test is first tried. We test for person dependent results only using the Chen-Huang database since the Cohn-Kanade database has only one sequence per expression per subject, making it impossible to do person dependent tests. We train classifiers for each subject, leaving out some of the subject’s data for testing. Table 1 shows the recognition rate of each subject and the (weighted) average recognition rate of the classifiers for all five subjects.

Table 1. Person-dependent facial expression recognition accuracies (in %).

Subject	NB	TAN	SSS
1	89.56	92.48	92.95
2	87.77	91.22	90.31
3	85.10	89.62	90.74
4	87.03	91.49	91.77
5	77.87	89.36	87.25
Average	85.58	90.92	90.48

We compare the results of three structure assumptions. Compared to the Naive Bayes classifier (where facial features are assumed to be independent), learning dependencies, either using TAN or using SSS, significantly improves the classification performance. We do not see any significant differences between SSS and TAN classifiers because the SSS algorithm could not search for structures much more complicated than the TAN, due to the limited size training set (higher complexity classifiers would overfit the training data).

3.2. Person-independent Tests

We perform person independent tests by partitioning the data such that the sequences of some subjects are used as the test sequences and the sequences of the remaining subjects are used as training sequences. Table 2 shows the recognition rate of the test for all classifier. The classifier learned with the SSS algorithm outperforms both the NB and TAN classifiers. One can also note that the results in this case are considerably lower that the ones obtained in the person-dependent case.

Table 2. Recognition rate (%) for person-independent test.

	NB	TAN	SSS
Chen-Huang Database	71.78	80.31	<u>83.62</u>
Cohn-Kandade Database	77.70	80.40	<u>81.80</u>

It is also informative to look at the structures that were learned from data. Figure 4 (b) and (c) shows two learned tree structure of the features (our Motion Units) one learned using the Cohn-Kanade database and the second from the Chen-Huang database. The arrows are from parents to children MUs. In both tree structures we see that the algorithm produced structures in which the bottom half of the face is almost disjoint from the top portion, except for a link between MU9 and MU8 in the first and a weak link between MU4 and MU11 in the second.

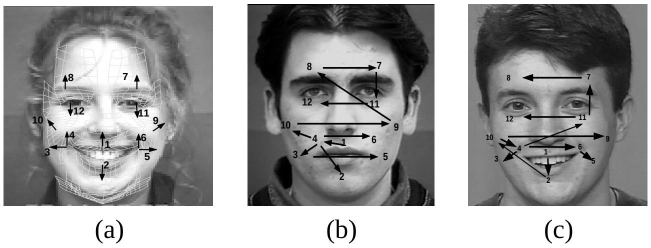


Fig. 4. (a) The facial motion measurements. Two learned TAN structures for the facial features, (b) using the Cohn-Kanade database, (c) using the Chen-Huang database.

4. DISCUSSION

We showed that the TAN and SSS algorithms can be used to enhance the performance of facial expression recognition over the simple Naive Bayes classifier for the person independent and dependent approaches.

Learning the structure of the Bayesian networks also showed that there is a weak dependency between the motion in the lower part of the face and the upper face, an observation that agrees with physiological intuition. The experiments also showed that allowing for learning of the dependencies also enhanced the classification performance with unlabeled data, when using the classification driven SSS algorithm. Such a result suggests that it is not enough to learn first order dependencies between the motion units (as in the case of TAN), rather, more complex dependencies are necessary to truly model the dependencies between facial motions.

Are the recognition rates sufficient for real world use? We think that it depends upon the particular application. In the case of image and video retrieval from large databases, the current recognition rates could aid in finding the right image or video by giving additional options for the queries. For future research, the integration of multiple modalities such as voice analysis and context would be expected to improve the recognition rates and eventually improve the computer's understanding of human emotional states. Voice and gestures are widely believed to play an important role as well [2, 6], and physiological states such as heart beat and skin conductivity are being suggested [1]. People also use context as an indicator of the emotional state of a person. The advantage of using Bayesian network classifiers is that they provide a good framework of fusing different modalities in an intuitive and coherent manner. This work is therefore a first step towards building a more comprehensive system of recognizing human's affective state by computers.

5. REFERENCES

[1] J. Cacioppo and L. Tassinary. Inferring psychological significance from physiological signals. *American Psychologist*, 45:16–28, Jan. 1990.

[2] L. Chen. *Joint Processing of Audio-visual Information for the Recognition of Emotional Expressions in Human-computer Interaction*. PhD thesis, University of Illinois at Urbana-Champaign, Dept. of Electrical Engineering, 2000.

[3] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, and T. Huang. Semi-supervised learning of classifiers: Theory, algorithms, and applications to human-computer interaction. *PAMI*, 26(12):1553–1567, 2004.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[5] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *BMVC*, pages 837–846., 2002.

[6] L. De Silva, T. Miyasato, and R. Natatsu. Facial emotion recognition using multimodal information. In *Proc. IEEE International Conference on Information, Communications, and Signal Processing*, pages 397–401, 1997.

[7] P. Ekman and W. Friesen. *Facial Action Coding System: Investigator's Guide*. Consulting Psychologists Press, 1978.

[8] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36:259–275, 2003.

[9] A. Garg, V. Pavlovic, J. Rehg, and T. Huang. Audio–visual speaker detection using dynamic Bayesian networks. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 374–471, 2000.

[10] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operational Research*, 13:311–329, 1988.

[11] I. Cohen, N. Sebe, A. Garg, L. Chen, T.S.Huang. Facial expression recognition from video sequences: Temporal and static modeling. *CVIU*, 91:160–187, 2003.

[12] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *International Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[13] D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232, 1995.

[14] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[15] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[16] M. Pantic and L. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[17] R. W. Picard. *Affective Computing*. MIT Press, Cambridge, MA, 1997.

[18] H. Tao and T. Huang. Connected vibrations: A modal analysis approach to non-rigid motion tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 735–740, 1998.

[19] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.