Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ACADEMIC PRESS

Computer Vision and Image Understanding 91 (2003) 160–187

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Facial expression recognition from video sequences: temporal and static modeling

Ira Cohen,[a,*] Nicu Sebe,[b] Ashutosh Garg,[c] Lawrence S. Chen,[d] and Thomas S. Huang[a]

[a] *Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL, USA*

[b] *Faculty of Science, University of Amsterdam, Netherlands*

[c] *IBM Almaden Reasearch Center, USA*

[d] *Imaging Science and Technology Lab, Eastman Kodak Company, USA*

Received 15 February 2002; accepted 11 February 2003

Abstract

The most expressive way humans display emotions is through facial expressions. In this work we report on several advances we have made in building a system for classification of facial expressions from continuous video input. We introduce and test different Bayesian network classifiers for classifying expressions from video, focusing on changes in distribution assumptions, and feature dependency structures. In particular we use Naive–Bayes classifiers and change the distribution from Gaussian to Cauchy, and use Gaussian Tree-Augmented Naive Bayes (TAN) classifiers to learn the dependencies among different facial motion features. We also introduce a facial expression recognition from live video input using temporal cues. We exploit the existing methods and propose a new architecture of hidden Markov models (HMMs) for automatically segmenting and recognizing human facial expression from video sequences. The architecture performs both segmentation and recognition of the facial expressions automatically using a multi-level architecture composed of an HMM layer and a Markov model layer. We explore both person-dependent and person-independent recognition of expressions and compare the different methods.

* Corresponding author.

E-mail addresses: iracohen@ifp.uiuc.edu (I. Cohen), nicu@science.uva.nl (N. Sebe), ashutosh@us.ibm.com (A. Garg), lawrence.chen@kodak.com (L. Chen), huang@ifp.uiuc.edu (T.S. Huang).

1077-3142/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S1077-3142(03)00081-X

1. Introduction

In recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. It is argued that to truly achieve effective human–computer intelligent interaction (HCII), there is a need for the computer to be able to interact naturally with the user, similar to the way human–human interaction takes place. Humans interact with each other mainly through speech, but also through body gestures, to emphasize a certain part of the speech and display of emotions. Emotions are displayed by visual, vocal, and other physiological means. There is a growing amount of evidence showing that emotional skills are part of what is called "intelligence" [16,36]. One of the important way humans display emotions is through facial expressions.

This work describes our real-time automatic facial expression recognition system using video input. Our work focuses on the design of the classifiers used for performing the recognition following extraction of features using our real-time face tracking system. We describe classification schemes in two types of settings: dynamic and 'static' classification.

The 'static' classifiers classify a frame in the video to one of the facial expression categories based on the tracking results of that frame. More specifically, we use Bayesian network classifiers and compare two different models: (1) Naive–Bayes classifiers where the features are assumed to be either Gaussian or Cauchy distributed, and (2) Gaussian Tree-Augmented Naive (TAN) Bayes classifiers. The Gaussian Naive–Bayes classifier is a standard classifier which has been used extensively in many classification problems. We propose changing the assumed distribution of the features from Gaussian to Cauchy because of the ability of Cauchy to account for heavy tail distributions. While Naive–Bayes classifiers are often successful in practice, they use a very strict and often unrealistic assumption, that the features are independent given the class. We propose using the Gaussian TAN classifiers which have the advantage of modeling dependencies between the features without much added complexity compared to the Naive–Bayes classifiers. TAN classifiers have an additional advantage in that the dependencies between the features, modeled as a tree structure, are efficiently learned from data and the resultant tree structure is assured to maximize the likelihood function.

Dynamic classifiers take into account the temporal pattern in displaying facial expression. We first describe the hidden Markov model (HMM) based classifiers for facial expression recognition which have been previously used in recent works [23,29,30]. We further advance this line of research and propose a multi-level HMM classifier, combining the temporal information which allows not only to perform the classification of a video segment to the corresponding facial expression, as in the previous works on HMM based classifiers, but also to automatically segment an arbitrary long video sequence to the different expressions segments without resorting to heuristic methods of segmentation.

An important aspect is that while the 'static' classifiers are easier to train and implement, the dynamic classifiers require more training samples and many more parameters to learn.

The rest of the paper is organized as follows. Section 2 reviews facial expression recognition studies. In Section 3 we briefly describe our real-time face tracking system and the features extracted for classification of facial expressions. Section 4 describes the Bayesian network classifiers used for classifying frames in the video sequence to the different expressions. In Section 5 we describe HMM based classifiers for facial expression recognition from presegmented video sequences and introduce the multi-level HMM classifier for both recognizing facial expression sequences and automatically segmenting the video sequence. We perform experiments for all the described methods using two databases in Section 6. The first is our database of subjects displaying facial expressions. The second is the Cohn–Kanade database [19]. We have concluding remarks in Section 7.

2. Review of facial expression recognition

Since the early 1970s, Paul Ekman and his colleagues [10] have performed extensive studies of human facial expressions. They found evidence to support universality in facial expressions. These "universal facial expressions" are those representing happiness, sadness, anger, fear, surprise, and disgust. They studied facial expressions in different cultures, including preliterate cultures, and found much commonality in the expression and recognition of emotions on the face. However, they observed differences in expressions as well, and proposed that facial expressions are governed by "display rules" in different social contexts. For example, Japanese subjects and American subjects showed similar facial expressions while viewing the same stimulus film. However, in the presence of authorities, the Japanese viewers were more reluctant to show their real expressions. Babies seem to exhibit a wide range of facial expressions without being taught, thus suggesting that these expressions are innate [18].

Ekman and Friesen [11] 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. This process is very time-consuming.

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 [2,9,12,21,23–26,28,30,35,40] has used these "basic expressions" or a subset of them. In [32], Pantic and Rothkrantz provide an in depth review of many of the research done in automatic facial expression recognition in recent years.

The work in computer-assisted quantification of facial expressions did not start until the 1990s. Mase [25] used optical flow (OF) to recognize facial expressions. He was one of the first to use image processing techniques to recognize facial expressions. Lanitis et al. [21] used a flexible shape and appearance model for image coding, person identification, pose recovery, gender recognition, and facial expression

recognition. Black and Yacoob [2] used local parameterized models of image motion to recover non-rigid motion. Once recovered, these parameters were used as inputs to a rule-based classifier to recognize the six basic facial expressions. Yacoob and Davis [41] computed optical flow and used similar rules to classify the six facial expressions. Rosenblum et al. [35] also computed optical flow of regions on the face, then applied a radial basis function network to classify expressions. Essa and Pentland [12] used an optical flow region-based method to recognize expressions. Donato et al. [9] tested different features for recognizing facial AUs and inferring the facial expression in the frame. Otsuka and Ohya [30] first computed optical flow, then computed the 2D Fourier transform coefficients, which were used as feature vectors for a hidden Markov model (HMM) to classify expressions. The trained system was able to recognize one of the six expressions near real-time (about 10 Hz). Furthermore, they used the tracked motions to control the facial expression of an animated Kabuki system [31]. A similar approach, using different features, was used by Lien [23]. Nefian and Hayes [26] proposed an embedded HMM approach for face recognition that uses an efficient set of observation vectors based on the DCT coefficients. Martinez [24] introduced an indexing approach based on the identification of frontal face images under different illumination conditions, facial expressions, and occlusions. A Bayesian approach was used to find the best match between the local observations and the learned local features model and an HMM was employed to achieve good recognition even when the new conditions did not correspond to the conditions previously encountered during the learning phase. Oliver et al. [28] used lower face tracking to extract mouth shape features and used them as inputs to an HMM based facial expression recognition system (recognizing neutral, happy, sad, and an open mouth).

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.

As mentioned in the previous section, the classifiers used can either be 'static' classifiers or dynamic ones. 'Static' classifiers use feature vectors related to a single frame to perform classification (e.g., Neural networks, Bayesian networks, and linear discriminant analysis). Temporal classifiers try to capture the temporal pattern in the sequence of feature vectors related to each frame such as the HMM based methods of [23,28,30].

3. Face tracking and feature extraction

The face tracking we use in our system is based on a system developed by Tao and Huang [39] called the Piecewise Bézier Volume Deformation (PBVD) tracker.

This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. In the first frame of the image sequence, landmark facial features such as the eye corners and mouth corners are selected interactively.

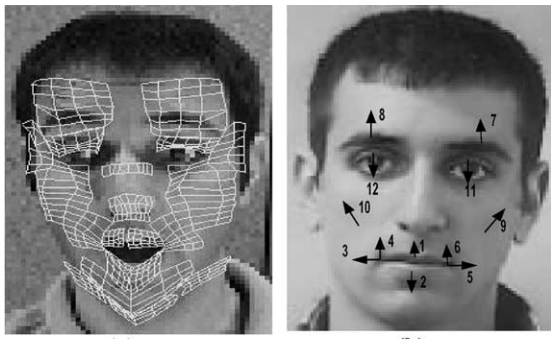

Fig. 1. (a) The wireframe model and (b) the facial motion measurements.

The generic face model is then warped to fit the selected facial features. The face model consists of 16 surface patches embedded in Bézier volumes. The surface patches defined this way are guaranteed to be continuous and smooth. The shape of the mesh can be changed by changing the locations of the control points in the Bézier volume.

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. Fig. 1a shows an example from one frame of the wireframe model overlayed on a face being tracked.

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 Bézier volume control parameters. We refer to these motions vectors as Motion-Units (MUs). Note that they are similar but not equivalent to Ekman's AUs and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The MUs used in the face tracker are shown in Fig. 1b and are described in Table 1.

The MUs are used as the basic features for the classification scheme described in the next sections.

4. Bayesian network classifiers for facial expression recognition

Bayesian networks can represent joint distributions in an intuitive and efficient way; as such, Bayesian networks are naturally suited to classification. We can use

Table 1
Action units used in our face tracker

MU	Description
1	Vertical movement of the center of upper lip
2	Vertical movement of the center of lower lip
3	Horizontal movement of left mouth corner
4	Vertical movement of left mouth corner
5	Horizontal movement of right mouth corner
6	Vertical movement of right mouth corner
7	Vertical movement of right brow
8	Vertical movement of left brow
9	Lifting of right cheek
10	Lifting of left cheek
11	Blinking of right eye
12	Blinking of left eye

a Bayesian network to compute the posterior probability of a set of *labels* given the observable *features*, and then we classify the features with the most probable label.

A Bayesian network classifier represents the dependencies among features and labels by a directed acyclic graph. This graph is the *structure* of the Bayesian network. Typically, Bayesian network classifiers are learned with a fixed structure—the paradigmatic example is the Naive–Bayes classifier. More flexible learning methods allow Bayesian network classifiers to be selected from a small subset of possible structures—for example, the Tree-Augmented Naive–Bayes structures [14]. After a structure is selected, the parameters of the classifier are usually learned using maximum likelihood estimation.

We propose using Bayesian network classifiers for recognizing facial expressions given the tracking results provided by the face tracking algorithm. Our classifiers are 'static' in the sense that their features are tracking results at each point in time.

Given a Bayesian network classifier with parameter set Θ, the optimal classification rule under the maximum likelihood (ML) framework to classify an observed feature vector of n dimensions, $X \in R^n$, to one of $|C|$ class labels, $c \in \{1, \ldots, |C|\}$, is given as:

$$\hat{c} = \underset{c}{\operatorname{argmax}} P(X|c; \Theta). \tag{1}$$

There are two design decisions when building Bayesian network classifiers. The first is to choose the structure of the network, which will determine the dependencies among the variables in the graph. The second is to determine the distribution of the features. The features can be discrete, in which case the distributions are probability mass functions. The features can also be continuous, in which case one typically has to choose a distribution, with the most common being the Gaussian distribution. Both these design decisions determine the parameter set Θ which defines the distribution needed to compute the decision function in Eq. (1). Designing the Bayesian network classifiers for facial expression recognition is the focus of this section.

4.1. Continuous Naive–Bayes: Gaussian and Cauchy Naive–Bayes classifiers

A Naive–Bayes classifier is a probabilistic classifier in which the features are assumed independent given the class. Naive–Bayes classifiers have a surprisingly very good record in many classification problems, although the independence assumption is usually violated in practice. Examples of applications that use the Naive–Bayes classifiers are abundant such as text classification [27] and face pose estimation [1]. Although the Naive–Bayes model does not reflect in many cases the true underlying model generating the data, it is still observed to be successful as a classifier in practice. The reason for the Naive–Bayes model's success as a classifier is attributed to the small number of parameters needed to be estimated, thus offsetting the large modeling bias with a small estimation variance [13]. Recently, Garg and Roth [15] showed using information theoretic arguments additional reasons for the success of Naive–Bayes classifiers. Thus, it is natural to explore the performance of the Naive–Bayes classifier before more complex structures. An example of a Naive–Bayes classifier is given in Fig. 2.

If the features in X are assumed to be independent of each other conditioned upon the class label c (the Naive–Bayes framework), Eq. (1) reduces to:

$$\hat{c} = \underset{c}{\operatorname{argmax}} \prod_{i=1}^{n} P(x_i|c;\Theta). \tag{2}$$

Now the problem is how to model $P(x_i|c;\Theta)$, which is the probability of feature x_i given the class label. In practice, the common assumption is that we have a Gaussian distribution and the ML can be used to obtain the estimate of the parameters (mean and variance). However, Sebe et al. [37] have shown that the Gaussian assumption is often invalid and proposed the Cauchy distribution as an alternative model. Intuitively, this distribution can be thought of as being able to model the heavy tails observed in the empirical distribution. This model is referred to as Cauchy Naive–Bayes.

The difficulty of this model is in estimating the parameters of the Cauchy distribution. For a feature of size N having a Cauchy distribution the likelihood is given by:

$$L(x_i|c;a_i,b_i) = \prod_{d=1}^{N} \left[\frac{b_i}{\pi(b_i^2 + (x_i^d - a_i)^2)} \right], \tag{3}$$

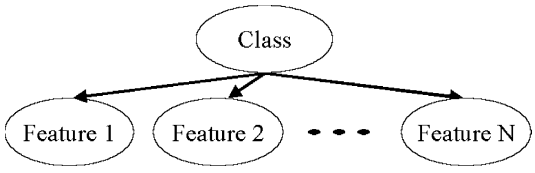

Fig. 2. An example of a Naive–Bayes classifier.

where a_i is the location parameter, b_i is the scale parameter, and $i = 1, \ldots, n$. Note that similar with the Gaussian case we have to estimate only two parameters.

Let $\hat{a}_i$ and $\hat{b}_i$ be the maximum likelihood estimators for a_i and b_i. The logarithm of the likelihood is:

$$\log L = -N\log\pi + N\log\hat{b}_i - \sum_{d=1}^{N} \log(\hat{b}_i^2 + (x_i^d - \hat{a}_i)^2). \tag{4}$$

Hence, the maximum likelihood equations are:

$$\frac{\partial \log L}{\partial a_i} = \sum_{d=1}^{N} \frac{x_i^d - \hat{a}_i}{\hat{b}_i^2 + (x_i^d - \hat{a}_i)^2} = 0, \tag{5}$$

$$\frac{\partial \log L}{\partial b_i} = \sum_{d=1}^{N} \frac{\hat{b}_i^2}{\hat{b}_i^2 + (x_i^d - \hat{a}_i)^2} - \frac{N}{2} = 0. \tag{6}$$

Eqs. (5) and (6) are high order polynomials and therefore a numerical procedure must be used in order to solve them for $\hat{a}$ and $\hat{b}$. For solving these equations we used a Newton–Raphson iterative method with the starting points given by the mean and the variance of the data. We were always able to find unique positive solutions for $\hat{a}$ and $\hat{b}$ which is in accordance with the conjecture stated by Hass et al. [17]. In certain cases, however, the Newton–Raphson iteration diverged, in which cases we selected new starting points.

An interesting problem is determining when to use the Cauchy assumption versus the Gaussian assumption. Our solution is to compute the distribution for the training data and to match this distribution using a Kolmogorov–Smirnov test with the model distribution (Cauchy or Gaussian) estimated using the ML approach described above.

The Naive–Bayes classifier was successful in many applications mainly due to its simplicity. Also, this type of classifier is working well even if there is not too much training data. However, the strong independence assumption may seem unreasonable in our case because the facial motion measurements are highly correlated when humans display emotions. Therefore, when sufficient training data is available we want to learn and to use these dependencies.

4.2. Beyond the Naive–Bayes assumption: finding dependencies among features using a Gaussian TAN classifier

The goal of this section is to provide a way to search for a structure that captures the dependencies among the features. Of course, to attempt to find all the dependencies is an NP-complete problem. So, we restrict ourselves to a smaller class of structures called the Tree-Augmented-Naive Bayes (TAN) classifiers. TAN classifiers have been introduced by Friedman et al. [14] and are represented as Bayesian networks. The joint probability distribution is factored to a collection of conditional probability distributions of each node in the graph.

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 (see Fig. 3). Friedman et al. [14] proposed using the TAN model as a classifier, to enhance the performance over the simple Naive–Bayes classifier. TAN models are more complicated than the Naive–Bayes, but are not fully connected graphs. The existence of an efficient algorithm to compute the best TAN model makes it a good candidate in the search for a better structure over the simple NB.

Learning the TAN classifier is more complicated. In this case, we do not fix the structure of the Bayesian network, but try to find the TAN structure that maximizes the likelihood function given the training data out of all possible TAN structures.

In general, searching for the best structure has no efficient solution, however, searching for the best TAN structure does have one. The method is using the modified Chow–Liu algorithm [5] for constructing tree augmented Bayesian networks [14]. The algorithm finds the tree structure among the features that maximizes the likelihood of the data by computation of the pairwise class conditional mutual information among the features and building a maximum weighted spanning tree using the pairwise mutual information as the weights of the arcs in the tree. The problem of finding a maximum weighted spanning is defined as finding the set of arcs connecting the features such that the resultant graph is a tree and the sum of the weights of the arcs is maximized. There have been several algorithms proposed for building a maximum weighted spanning tree [7] and in our implementation we use the Kruskal's algorithm described in Fig. 4.

The five steps of the TAN algorithm are described in Fig. 5. This procedure ensures to find the TAN model that maximizes the likelihood of the data we have. The algorithm is computed in polynomial time ($O(n^2 \log N)$, with N being the number of instances and n the number of features).

The learning algorithm for the TAN classifier as proposed by Friedman et al. [14] relies on computations of the class conditional mutual information of discrete features. In our problem the features are continuous, and computation of the mutual information for a general distribution is very complicated. However, if we assume that the features are Gaussian, computation of the conditional mutual information is feasible and is given by (see Appendix A for more details):

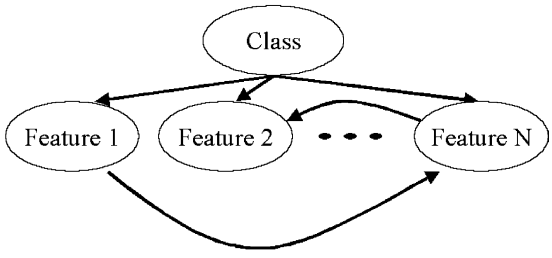

Fig. 3. An example of a TAN classifier.

Consider an undirected graph with n vertices and m edges, where each edge (u, v) connecting the vertices u and v, has an associated positive weight $w_{(u,v)}$. To construct the maximum weighted spanning tree graph follow the following steps:

1. Create an empty set of edges called *spanningTree*.
2. For each vertex v in the graph, create a set containing v.
3. Sort all edges in the graph using the weights in the edges from highest to lowest.
4. In order of the sorted edges, for each edge (u, v), if the set that contains u is different from the set that contains v:
   - Put the edge (u, v) in *spanningTree*.
   - Make u and v belong to the same set (union of sets).
5. *spanningTree* contains all the edges in the maximum weighted spanning tree.

Fig. 4. Kruskal's Maximum Weighted Spanning Tree algorithm.

1. Compute the class conditional pair-wise mutual information between each pair of features, (X_i, X_j) for all $i, j \in \{1, ..., n\}$,

$$I_P(X_i, X_j|C) = \sum_{X_i, X_j, C} P(x_i, x_j, c) \log \frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)}, i \neq j.$$

2. Build a complete undirected graph in which each vertex is a variable, and the weight of each edge is the mutual information computed in Step 1.
3. Build a maximum weighted spanning tree (MWST) (see Figure 4).
4. Transform the undirected MWST of Step 3 to a directed graph by choosing a root node and pointing the arrows of all edges away from the root.
5. Make the class node the parent of all the feature nodes in the directed graph of Step 4.

Fig. 5. TAN learning algorithm.

$$I(X_i, X_j|C) = -\frac{1}{2} \sum_{c=1}^{|C|} P(C = c) \log(1 - \rho^2_{(ij)|c}), \tag{7}$$

where $\rho_{(ij)|c}$ is the correlation coefficient between X_i and X_j given the class label c. We replace the expression for the mutual information in Step 1 of the TAN algorithm with the expression in Eq. (7), to find the maximum likelihood Gaussian-TAN classifier.

The full joint distribution of the Gaussian-TAN model can be written as:

$$p(c, x_1, x_2, \ldots, x_n) = p(c) \prod_{i=1}^{n} p(x_i|pa_{x_i}, c), \tag{8}$$

where pa_{x_i} is the feature that is the additional parent of feature x_i. pa_{x_i} is empty for the root feature in the directed tree graph of Step 4 in Fig. 5.

Using the Gaussian assumption, the pdf's of the distribution in the product above are:

$$p(X_i = x_i | pa_{x_i}, C = c) = N_c(\mu_{x_i} + a \cdot pa_{x_i}, \sigma_{x_i}^2 \cdot (1 - \rho^2)), \tag{9}$$

where $N_c(\mu, \sigma^2)$ refers to the Gaussian distribution with mean and variance given that the class is c, $\mu_{x_i}, \sigma_{x_i}^2$ are the mean and variance of the feature x_i,

$$\rho = \frac{\mathrm{COV}(x_i, pa_{x_i})}{\sigma_{x_i}\sigma_{pa_{x_i}}}$$

is the correlation coefficient between x_i and pa_{x_i}, and

$$a = \frac{\mathrm{COV}(x_i, pa_{x_i})}{\sigma_{pa_{x_i}}^2}.$$

For further details on the derivation of the parameters see the Appendix A.

After learning the structure, the Gaussian-TAN classifier's added complexity compared to the Naive-Bayes classifier is small: there are $|C| \cdot (n - 1)$ extra parameters to estimate (the covariances between features and their parents). For learning the structure, all pairwise mutual information are estimated using the estimates for the covariances.

For facial expression recognition, the learned TAN structure can provide additional insight on the interaction between facial features in determining facial expressions. Fig. 6 shows a learned tree structure of the features (our Motion Units) learned using our database of subjects displaying different facial expressions (more details on the experiments are in Section 6). The arrows are from parents to children MUs. From the tree structure we see that the TAN learning algorithm produced a structure in which the bottom half of the face is almost disjoint from the top portion, except for a weak link between MU 4 and MU 11.

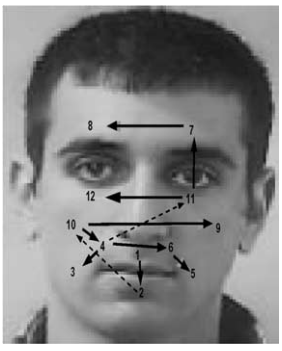

Fig. 6. The learned TAN structure for the facial features. Dashed lines represent links that are relatively weaker than the others.

5. The dynamic approach: facial expression recognition using multi-level HMMs

As discussed in Section 1, the second approach to perform classification of video sequences to facial expression is the dynamic approach. The dynamic approach uses classifiers that can use temporal information to discriminate different expressions. The logic behind using the temporal information is that expressions have a unique temporal pattern. When recognizing expressions from video, using the temporal information can lead to more robust and accurate classification results compared to methods that are 'static.'

The method we propose automatically segments the video to the different facial expression sequences, using a multi-level HMM structure. The first level of the architecture is comprised of independent HMMs related to the different emotions. This level of HMMs is very similar to the one used in [23,29,30] who used the likelihood of a given sequence in a ML classifier to classify a given video sequence. Instead of classifying using the output of each HMM, we use the state sequence of the HMMs as the input of the higher-level Markov model. This is meant to segment the video sequence, which is the main problem facing the previous works using HMMs for expression recognition. Moreover, this also increases the discrimination between the classes since it tries to find not only the probability of each the sequence displaying one emotion, but also the probability of the sequence displaying one emotion and not displaying all the other emotions at the same time.

5.1. Hidden Markov models

Hidden Markov models have been widely used for many classification and modeling problems. Perhaps the most common application of HMM is in speech recognition [34]. One of the main advantages of HMMs is their ability to model non-stationary signals or events. Dynamic programming methods allow one to align the signals so as to account for the non-stationarity. However, the main disadvantage of this approach is that it is very time-consuming since all of the stored sequences are used to find the best match. The HMM finds an implicit time warping in a probabilistic parametric fashion. It uses the transition probabilities between the hidden states and learns the conditional probabilities of the observations given the state of the model. In the case of emotion expression, the signal is represented by the measurements of the facial motion. This signal is non-stationary in nature, since an expression can be displayed at varying rates and with varying intensities even for the same individual.

An HMM is given by the following set of parameters:

$$\lambda = (A, B, \pi),$$

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leqslant i, j \leqslant N,$$

$$B = \{b_j(O_t)\} = P(O_t | q_t = S_j), \quad 1 \leqslant j \leqslant N,$$

$$\pi_j = P(q_1 = S_j),$$

where A is the state transition probability matrix, B is the observation probability distribution, and π is the initial state distribution. The number of states of the HMM is given by N. It should be noted that the observations (O_t) can be either discrete or continuous, and can be vectors. In the discrete case, B becomes a matrix of probability entries (Conditional Probability Table), and in the continuous case, B will be given by the parameters of the probability distribution function of the observations (normally chosen to be the Gaussian distribution or a mixture of Gaussians). Given an HMM, there are three basic problems of interest. The first is how to efficiently compute the probability of the observations given the model. This problem is related to classification in the sense that it gives a measure of how well a certain model describes an observation sequence. The second is how to find the corresponding state sequence in some optimal way, given a set of observations and the model. This will become an important part of the algorithm to recognize the expressions from live input and will be described later in this paper. The third is how to learn the parameters of the model λ given the set of observations so as to maximize the probability of observations given the model. This problem relates to the learning phase of the HMMs which describe each facial expression sequence. A comprehensive tutorial on HMMs is given by Rabiner [33].

5.2. Expression recognition using emotion-specific HMMs

Since the display of a certain facial expression in video is represented by a temporal sequence of facial motions it is natural to model each expression using an HMM trained for that particular type of expression. There will be six such HMMs, one for each expression: {*happy*(1), *angry*(2), *surprise*(3), *disgust*(4), *fear*(5), *sad*(6)}. There are several choices of model structure that can be used. The two main models are the left-to-right model and the ergodic model. In the left-to-right model, the probability of going back to the previous state is set to zero, and therefore the model will always start from a certain state and end up in an 'exiting' state. In the ergodic model, every state can be reached from any other state in a finite number of time steps. In [30], Otsuka and Ohya used left-to-right models with three states to model each type of facial expression. The advantage of using this model lies in the fact that it seems natural to model a sequential event with a model that also starts from a fixed starting state and always reaches an end state. It also involves fewer parameters and therefore is easier to train. However, it reduces the degrees of freedom the model has to try to account for the observation sequence. There has been no study to indicate that the facial expression sequence is indeed modeled well by the left-to-right model. On the other hand, using the ergodic HMM allows more freedom for the model to account for the observation sequences, and in fact, for an infinite amount of training data it can be shown that the ergodic model will reduce to the left-to-right model, if that is indeed the true model. In this work, both types of models were tested with various numbers of states in an attempt to study the best structure that can model facial expressions.

The observation vector O_t for the HMM represents continuous motion of the facial action units. Therefore, B is represented by the probability density functions

(pdf) of the observation vector at time t given the state of the model. The Gaussian distribution is chosen to represent these pdfs, i.e.,

$$B = \{b_i(O_t)\} \sim N(\mu_j, \Sigma_j), \quad 1 \leqslant j \leqslant N, \tag{10}$$

where μ_j and Σ_j are the mean vector and full covariance matrix, respectively.

The parameters of the model of emotion-expression specific HMM are learned using the well-known Baum–Welch reestimation formulas (see [22] for details of the algorithm). For learning, hand labeled sequences of each of the facial expressions are used as ground truth sequences, and the Baum algorithm is used to derive the maximum likelihood (ML) estimation of the model parameters (λ).

Parameter learning is followed by the construction of a ML classifier. Given an observation sequence O_t, where $t \in (1, T)$, the probability of the observation given each of the six models $P(O_t|\lambda_j)$ is computed using the forward–backward procedure [33]. The sequence is classified as the emotion corresponding to the model that yielded the highest probability, i.e.,

$$c^* = \underset{1 \leqslant c \leqslant 6}{\operatorname{argmax}}[P(O|\lambda_c)]. \tag{11}$$

5.3. Automatic segmentation and recognition of emotions using multi-level HMM

The main problem with the approach taken in the previous section is that it works on isolated facial expression sequences or on presegmented sequences of the expressions from the video. In reality, this segmentation is not available, and therefore there is a need to find an automatic way of segmenting the sequences. Concatenation of the HMMs representing phonemes in conjunction with the use of grammar has been used in many systems for continuous speech recognition [34]. Dynamic programming for continuous speech has also been proposed in different researches. It is not very straightforward to try and apply these methods to the emotion recognition problem since there is no clear notion of language in displaying emotions. Otsuka and Ohya [30] used a heuristic method based on changes in the motion of several regions of the face to decide that an expression sequence is beginning and ending. After detecting the boundaries, the sequence is classified to one of the emotions using the emotion-specific HMM. This method is prone to errors because of the sensitivity of the classifier to the segmentation result. Although the result of the HMMs are independent of each other, if we assume that they model realistically the motion of the facial features related to each emotion, the combination of the state sequence of the six HMMs together can provide very useful information and enhance the discrimination between the different classes. Since we will use a left-to-right model (with return), the changing of the state sequence can have a physical attribute attached to it (such as opening and closing of mouth when smiling), and therefore there we can gain useful information from looking at the state sequence and using it to discriminate between the emotions at each point in time.

To solve the segmentation problem and enhance the discrimination between the classes, a different kind of architecture is needed. Fig. 7 shows the proposed architecture for automatic segmentation and recognition of the displayed expression at each

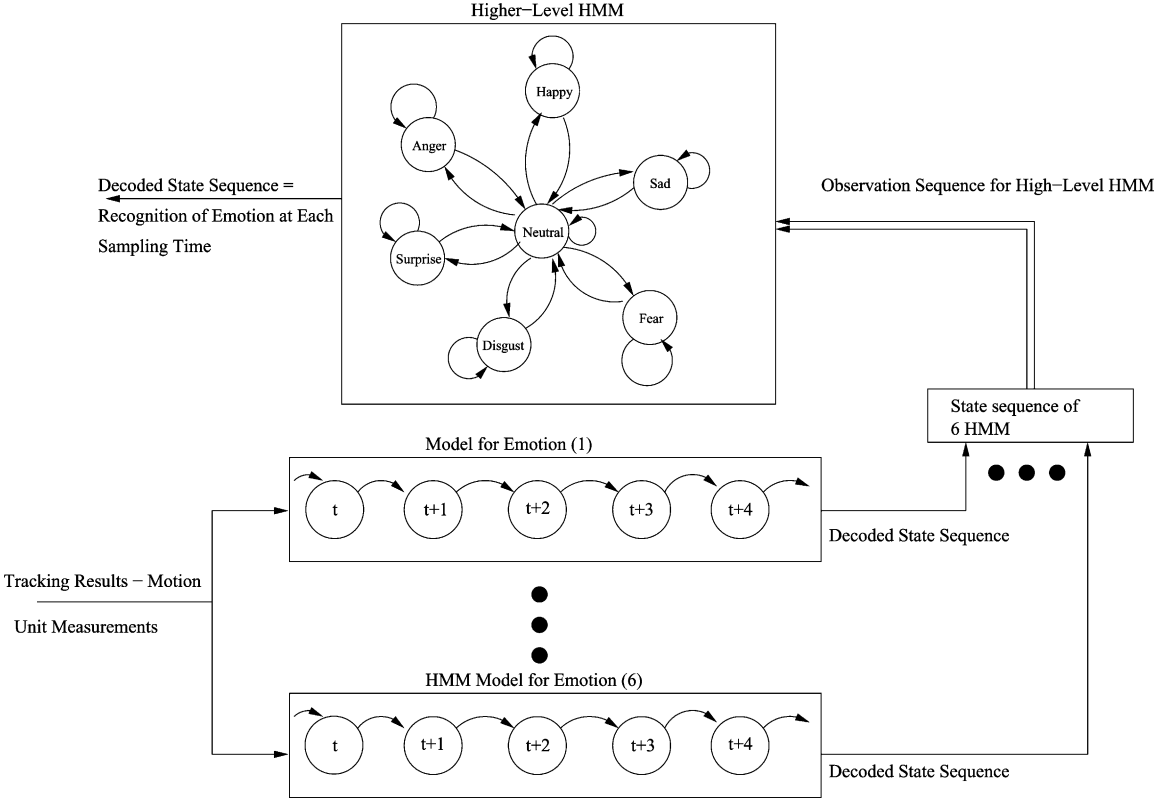

Fig. 7. Multi-level HMM architecture for automatic segmentation and recognition of emotion.

time instance. The motion features are continuously used as input to the six emotion-specific HMMs. The state sequence of each of the HMMs is decoded and used as the observation vector for the high-level Markov model. The high-level Markov model consists of seven states, one for each of the six emotions and one for *neutral*. The *neutral* state is necessary as for the large portion of time, there is no display of emotion on a person's face. In this implementation of the system, the transitions between emotions are imposed to pass through the *neutral* state since our training data consists of facial expression sequences that always go through the *neutral* state.

It is possible (although less likely) for a person to go from one expression to another without passing through a neutral expression, as has been reported in [30]. Handling such cases is done by slightly modifying the high-level HMM of Fig. 7. We simply have to set the transition probabilities of passing from all states to all states to values higher than zero (which appears as arcs between the different states of the expressions in the high-level HMM).

The recognition of the expression is done by decoding the state that the high-level Markov model is in at each point in time since the state represents the displayed emotion.

The training procedure of the system is as follows:

- Train the emotion-specific HMMs using a hand segmented sequence as described in the previous section.
- Feed all six HMMs with the continuous (labeled) facial expression sequence. Each expression sequence contains several instances of each facial expression with *neutral* instances separating the emotions.
- Obtain the state sequence of each HMM to form the six-dimensional observation vector of the higher-level Markov model, i.e., $O_t^h = [q_t^{(1)}, \ldots, q_t^{(6)}]^{\mathrm{T}}$, where $q_t^{(i)}$ is the state of the ith emotion-specific HMM. The decoding of the state sequence is done using the Viterbi algorithm [33].
- Learn the probability observation matrix for each state of the high-level Markov model using $P(q_j^{(i)}|S_k) = \{\text{expected frequency of model } i \text{ being in state } j \text{ given that the true state was } k\}$, and

$$B^{(h)} = \{b_k(O_t^h)\} = \left\{ \prod_{i=1}^{6} (P(q_j^{(i)}|S_k) \right\}, \tag{12}$$

  where $j \in (1, \text{Number of States for Lower-Level HMM})$.
- Compute the transition probability $A = \{a_{kl}\}$ of the high-level HMM using the frequency of transiting from each of the six emotion classes to the *neutral* state in the training sequences and from the *neutral* state to the other emotion states. For notation, the *neutral* state is numbered 7 and the other states are numbered as in the previous section. All the transition probabilities could also be set using expert knowledge.
- Set the initial probability of the high-level Markov model to be 1 for the *neutral* state and 0 for all other states. This forces the model to always start at the *neutral* state and assumes that a person will display a *neutral* expression in the beginning of any video sequence. This assumption is made just for simplicity of the testing.

The steps followed during the testing phase are very similar to the ones followed during training. The face tracking sequence is used as input into the lower-level HMMs and a decoded state sequence is obtained using the Viterbi algorithm. The decoded lower-level state sequence O_t^h is used as input to the higher-level HMM and the observation probabilities are computed using Eq. (12). Note that in this way of computing the probability, it is assumed that the state sequences of the lower-level HMMs are independent given the true labeling of the sequence. This assumption is reasonable since the HMMs are trained independently and on different training sequences. In addition, without this assumption, the size of B will be enormous, since it will have to account for all possible combinations of states of the six lower-level HMMs, and it would require a huge amount of training data.

Using the Viterbi algorithm again for the high-level Markov model, a most likely state sequence is produced. The state that the HMM was in at time t corresponds to the expressed emotion in the video sequence at time t. To make the classification result robust to undesired fast changes, a smoothing of the state sequence is done by preserving the actual classification result if the HMM did not stay in a particular state for more than T times, where T can vary between 1 and 15 samples (assuming a 30-Hz sampling rate). The introduction of the smoothing factor T will cause a delay in the decision of the system, but of no more than T sample times.

6. Experiments

In order to test the algorithms described in the previous sections we use two different databases, a database collected by us and the Cohn–Kanade [19] AU code facial expression database.

The first is a database we collected of subjects that were instructed to display facial expressions corresponding to the six types of emotions. The data collection method is described in detail in [4]. 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, and always coming back to a neutral state between each emotion sequence. We imposed the restriction of coming back to the neutral state after each emotion for the sake of simplicity in labeling the sequence. However, as mentioned in the previous section our system is also able to deal with the situation where a person can go from one expression to another without passing through a neutral expression.

The video was used as input to the face tracking algorithm described in Section 3. The sampling rate was 30 Hz, and a typical emotion sequence is about 70 samples long ($\sim$2 s). Fig. 8 shows one frame of each emotion for each subject.

The data were collected in an open recording scenario, where the person was asked to display the expression corresponding to the emotion being induced. This is of course not the ideal way of collecting emotion data. The ideal way would be using a hidden recording, inducing the emotion through events in the normal environment of the subject, not in a studio. The main problem with collecting the data this way is the impracticality of it and the ethical issue of hidden recording.

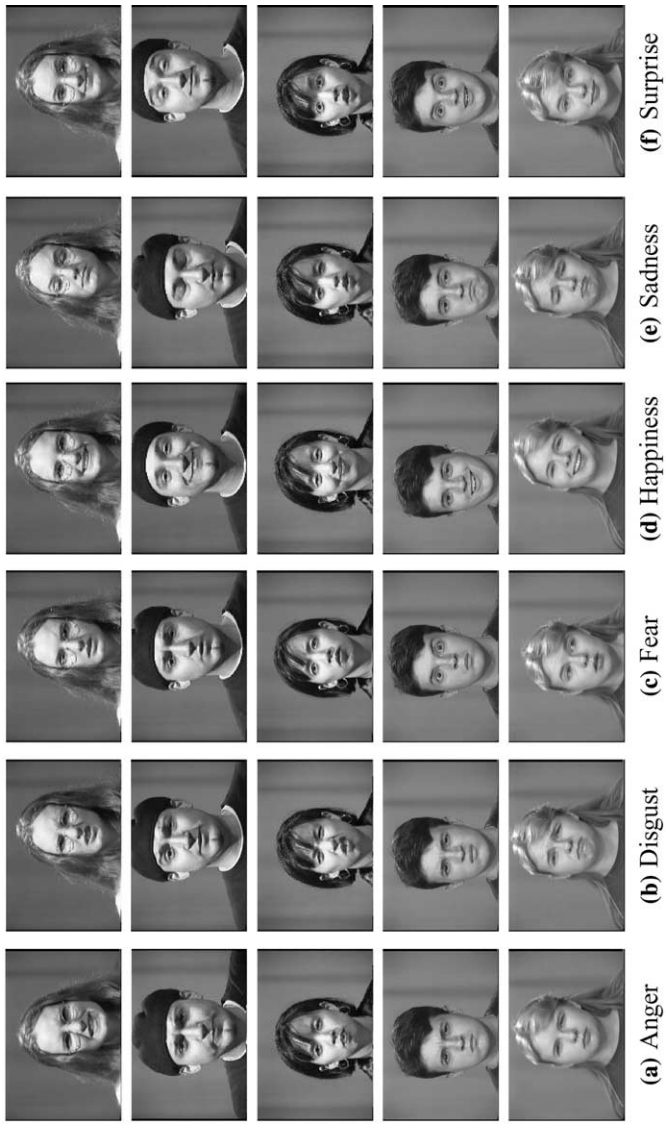

Fig. 8. Examples of images from the video sequences used in the experiment.

We use our database in two types of experiments. First we performed person-dependent experiments, in which part of the data for each subject was used as training data, and another part as test data. Second, we performed person-independent experiments, in which we used the data of all but one person as training data, and tested on the person that was left out.

For the TAN classifiers we used the dependencies shown in Fig. 6, learned using the algorithm described in Section 4.2. For the HMM-based models, several states were tried (3–12) and both the ergodic and left-to-right with return were tested. The results presented below are of the best configuration (an ergodic model using 11 states), determined using leave-one-out cross-validation over the training set (leaving an expression sequence out for each validation).

The Cohn–Kanade database [19] consists of expression sequences of subjects, starting from a Neutral expression and ending in the peak of the facial expression. There are 104 subjects in the database. Because for some of the subjects not all of the six facial expressions sequences were available to us, we used a subset of 53 subjects, for which at least four of the sequences were available. For each person there are on average 8 frames for each expression, which makes insufficient data to perform person-dependent tests. Also, the fact that each sequence ends in the peak of the facial expression makes the use of our dynamic multi-level HMM classifier impractical since in this case each sequence counts for an incomplete temporal pattern. In these conditions, we only used this database for performing person-independent tests using the 'static' Bayesian network classifiers.

A summary of both databases is presented in Table 2.

For the frame based methods (NB-Gaussian, NB-Cauchy, and TAN), 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. The accuracy for the temporal based methods is measured with respect to the misclassification rate of an expression sequence, not with respect to each frame.

Table 2
Summary of the databases

Database	# of subjects	Overall # of sequences per expression	# of sequences per subject per expression	Average # of frames per expression
Our DB	5	30	6	70
Cohn–Kanade DB	53	53	1	8

In order to show the statistical significance of our results we also present the 95% confidence intervals.

6.1. Results using our database

6.1.1. Person-dependent tests

A person-dependent test is first tried. Tables 3 and 4 show the recognition rate of each subject and the average recognition rate of the classifiers.

The fact that subject 5 was poorly classified can be attributed to the inaccurate tracking result and lack of sufficient variability in displaying the emotions. It can also be seen that the multi-level HMM achieves similar recognition rate (and improves it in some cases) compared to the emotion-specific HMM, even though the input is unsegmented continuous video.

The NB-Cauchy assumption does not give a significant improvement in recognition rate compared to the NB-Gaussian assumption. This is mainly due to the fact that in this case there are not many outliers in the data (we train and test with sequences of the same person in the same environment). However, this may not be the case in a natural setting experiment. Note that only in the case of subject 2 the Gaussian assumption gave better results than the Cauchy assumption. This result can be attributed to the fact that this subject shows the expressions in a more consistent way over time and this counts for fewer outliers in the recorded data. It is also

Table 3
Person-dependent facial expression recognition rates together with their 95% confidence intervals for frame based methods

Subject	NB-Gaussian (%)	NB-Cauchy (%)	TAN (%)
1	80.97	81.69	85.94
2	87.09	84.54	89.39
3	82.5	83.05	86.58
4	77.18	79.25	82.84
5	69.06	71.74	71.78
Average	79.36 ± 0.3	80.05 ± 0.29	83.31 ± 0.27

Table 4
Person-dependent facial expression recognition rates together with their 95% confidence intervals for the emotion-specific HMM and multi-level HMM

Subject	Single HMM (%)	Multi-level HMM (%)
1	82.86	80
2	91.43	85.71
3	80.56	80.56
4	83.33	88.89
5	54.29	77.14
Average	78.49 ± 2.98	82.46 ± 2.76

important to observe that taking into account the dependencies in the features (the TAN model) gives significantly improved results.

In average the best results are obtained by TAN followed by the NB-Cauchy and NB-Gaussian.

The confusion matrix for the TAN classifier is presented in Table 5. The analysis of the confusion between different emotions shows that most of the confusion of the classes is with the Neutral class. This can be attributed to the arbitrary labeling of each frame in the expression sequence. The first and last few frames of each sequence are very close to the Neutral expression and thus are more prone to become confused with it. We also see that most expression do not confuse with Happy.

The confusion matrices for the HMM based classifiers (described in details in [6]) show similar results, with Happy achieving near 100%, and Surprise approximately 90%.

6.1.2. Person-independent tests

In the previous section it was seen that a good recognition rate was achieved when the training sequences were taken from the same subject as the test sequences. A more challenging application is to create a system which is person-independent. In this case, the variation of the data is more significant and we expect that using a Cauchy-based classifier we will obtain significantly better results.

For this test all of the sequences of one subject are used as the test sequences and the sequences of the remaining four subjects are used as training sequences. This test is repeated five times, each time leaving a different person out (leave-one-out cross-validation). Table 6 shows the recognition rate of the test for all classifiers. In this case, the recognition rates are lower compared with the person-dependent results. This means that the confusions between subjects are larger than those within the same subject.

Table 5
Person-dependent confusion matrix using the TAN classifier

Emotion	Neutral	Happy	Anger	Disgust	Fear	Sad	Surprise
Neutral	*79.58*	1.21	3.88	2.71	3.68	5.61	3.29
Happy	1.06	*87.55*	0.71	3.99	2.21	1.71	2.74
Anger	5.18	0	*85.92*	4.14	3.27	1.17	0.30
Disgust	2.48	0.19	1.50	*83.23*	3.68	7.13	1.77
Fear	4.66	0	4.21	2.28	*83.68*	2.13	3.00
Sad	13.61	0.23	1.85	2.61	0.70	*80.97*	0
Surprise	5.17	0.80	0.52	2.45	7.73	1.08	*82.22*

Table 6
Recognition rate for person-independent test together with their 95% confidence intervals

Classifier	NB-Gaussian (%)	NB-Cauchy (%)	TAN (%)	Single HMM (%)	Multi-level HMM (%)
Recognition rate	60.23 ± 0.36	64.77 ± 0.3	66.53 ± 0.28	55.71 ± 3.61	58.63 ± 3.58

The TAN classifier provides the best results. It is important to observe that the Cauchy assumption also yields a larger improvement compared to the Gaussian classifier, due to the capability of the Cauchy distribution to handle outliers. One of the reasons for the misclassifications is the fact that the subjects are very different from each other (three females, two males, and different ethnic backgrounds); hence, they display their emotion differently. Although it appears to contradict the universality of the facial expressions as studied by Ekman and Friesen [11], the results show that for practical automatic emotion recognition, consideration of gender and race play a role in the training of the system.

Table 7 shows the confusion matrix for the TAN classifier. We see that Happy, Fear, and Surprise are detected with high accuracy, and other expressions are greatly confused mostly with Neutral. Here the differences in the intensity of the expressions among the different subjects played a significant role in the confusion among the different expressions.

6.2. Results using the Cohn–Kanade database

For this test we first divided our database in five sets which contain the sequences corresponding to 10 or 11 subjects (three sets with 11 subjects, two sets with 10 subjects). We used the sequences from a set as test sequences and the remaining sequences were used as training sequences. This test was repeated five times, each time leaving a different set out (leave-one-out cross-validation). Table 8 shows the recognition rate of the test for all classifiers. Note that the results obtained with this database are much better than the ones obtained with our database. This is because in this case we have more training data. For training we had available the data from more than 40 different persons. Therefore, the learned model is more accurate and can achieve better classification rates when using the test data.

Table 7
Person-independent average confusion matrix using the TAN classifier

Emotion	Neutral	Happy	Anger	Disgust	Fear	Sad	Surprise
Neutral	*76.95*	0.46	3.39	3.78	7.35	6.53	1.50
Happy	3.21	*77.34*	2.77	9.94	0	2.75	3.97
Anger	14.33	0.89	*62.98*	10.60	1.51	9.51	0.14
Disgust	6.63	8.99	7.44	*52.48*	2.20	10.90	11.32
Fear	10.06	0	3.53	0.52	*73.67*	3.41	8.77
Sad	13.98	7.93	5.47	10.66	13.98	*41.26*	6.69
Surprise	4.97	6.83	0.32	7.41	3.95	5.38	*71.11*

Table 8
Recognition rates for Cohn–Kanade database together with their 95% confidence intervals

Classifier	NB-Gaussian (%)	NB-Cauchy (%)	TAN (%)
Recognition rate	67.03 ± 1.33	68.14 ± 1.29	73.22 ± 1.24

Table 9
Person-independent average confusion matrix using the TAN classifier

Emotion	Neutral	Happy	Anger	Disgust	Fear	Sad	Surprise
Neutral	*78.59*	1.03	3.51	8.18	1.85	5.78	1.03
Happy	0	*86.22*	4.91	5.65	3.19	0	0
Anger	2.04	4.76	*66.46*	14.28	5.21	6.09	1.14
Disgust	3.40	1.13	10.90	*62.27*	10.90	9.09	2.27
Fear	1.19	13.57	7.38	7.61	*63.80*	3.80	1.90
Sad	5.55	1.58	13.25	11.19	3.96	*61.26*	3.17
Surprise	0	0	0	0	2.02	4.04	*93.93*

In average the best results were obtained using the TAN followed by NB-Cauchy and NB-Gaussian which is consistent with the results obtained with our database.

The confusion matrix for the TAN classifier is presented in Table 9. In this case, Surprise was detected with over 93% accuracy and Happy with over 86% accuracy. The other expressions are greatly confused with each other.

7. Summary and discussion

In this work, we presented several methods for expression recognition from video. Our intention was to perform an extensive evaluation of our methods using static and dynamic classification.

In the case of 'static' classifiers the idea was to classify each frame of a video to one of the facial expressions categories based on the tracking results of that frame. The classification in this case was done using Bayesian networks classifiers. We showed that there are two design decisions for building such classifiers: (1) determining the distribution of the features and (2) choosing the structure of the network which determines the dependencies among the features.

We first presented Naive–Bayes classifiers which assumed that the features are independent given the class. The common assumption is that we have a Gaussian distribution for the features but we showed that in practice using the Cauchy distribution we obtained improved classification results. The problem with the Naive–Bayes approach is that the independence assumption may be too strong for our application because the facial motion measurements are highly correlated when humans display emotions. Therefore, our next effort was in developing another classifier that took into account these dependencies among features. We used the TAN classifier and showed a method to search for the optimal TAN structure when the features were assumed to be Gaussian. We showed that after learning the structure from data, the Gaussian-TAN classifier added only small complexity to the Naive–Bayes approach and improved significantly the classification results.

A legitimate question here is, ''Is it always possible to learn the TAN structure from the data and use it in classification?'' Provided that there is sufficient training data, the TAN structure indeed can be extracted and used in classification. However, when the data is insufficient the learned structure is unreliable and the use of the

Naive–Bayes classifier is recommended. Note also that in the Naive–Bayes approach one can use a better distribution assumption than the Gaussian (e.g., Cauchy) while in TAN this would be extremely difficult.

In the case of dynamic classifiers the temporal information was used to discriminate different expressions. The idea is that expressions have a unique temporal pattern and recognizing these patterns can lead to improved classification results. We introduced the multi-level HMM architecture and compared it to the straight forward emotion-specific HMM. We showed that comparable results can be achieved with this architecture, although it does not rely on any presegmentation of the video stream.

When one should use a dynamic classifier versus a 'static' classifier? This is a difficult question to ask. It seems, both from intuition and from our results, that dynamic classifiers are more suited for systems that are person-dependent due to their higher sensitivity not only to changes in appearance of expressions among different individuals, but also to the differences in temporal patterns. 'Static' classifiers are easier to train and implement, but when used on a continuous video sequence, they can be unreliable especially for frames that are not at the peak of an expression. Another important aspect is that the dynamic classifiers are more complex, therefore they require more training samples and many more parameters to learn compared with the static approach. A hybrid of classifiers using expression dynamics and static classification is the topic of our future research.

In conclusion, our main contributions are as follows. We applied Bayesian network classifiers to the problem of facial expression recognition and we proposed in the case of Naive–Bayes classifiers the use of Cauchy distribution assumption. Moreover, for the same application we used the TAN classifier and we modified the learning algorithm proposed by Friedman et al. [14] to deal with continuous features. We also proposed the multi-level HMM architecture. We integrated the classifiers and the face tracking system to build a real-time facial expression recognition system.

An important problem in the facial expression analysis field is the lack of agreed upon benchmark datasets and methods for evaluating performance. A well-defined and commonly used database is a necessary prerequisite to compare the performances of different methods in an objective manner. The Cohn–Kanade database is a step in this direction, although there is still a need for an agreement on how to measure performance: frame based classification, sequence based classification, and even the number and names of the classes. The large deviations in the reported performance of different methods surveyed by Pantic and Rothkrantz [32] demonstrate the need to resolve these issues. As a consequence, it is hard to compare our results with the one reported in the literature and assert superiority or inferiority of our methods over others.

Are these 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 [4,8], and physiological states such as heart beat and skin conductivity are being suggested [3]. People also use context as an indicator of the emotional state of a person. This work is just another step on the way toward achieving the goal of building more effective computers that can serve us better.

Acknowledgments

We would like to thank Fabio Cozman for the use of his code implementing the Naive–Bayes and TAN classifiers in the Java language, using the libraries of the JavaBayes system (available at http://www.cs.cmu.edu/~javabayes). We would also like to thank Jilin Tu for his work on the face tracking, Jeffery Cohn for the use of the facial expression database, and Michael Lew for discussions on various parts of this work. This work has been supported in part by the National Science Foundation Grants CDA-96-24396 and IIS-00-85980. The work of Ira Cohen and Ashutosh Garg was supported by Hewlett Packard and IBM fellowships, respectively.

Appendix A. Gaussian-TAN parameters computation

The purpose of this appendix is to complete the derivation of the Gaussian mutual information and parameters of the TAN classifier for purpose of completeness (see Section 4.2). We note that these definitions can be found elsewhere, such as in [20,38].

The mutual information between two continuous random variables, X, Y is given as:

$$I(X,Y) = \iint p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)\mathrm{d}x\,\mathrm{d}y = H(x) + H(y) - H(x,y), \tag{A.1}$$

where $H(\cdot)$ is the differential entropy, analogous to the entropy of discrete variables, defined as:

$$H(Z) = -\int p(z)\log p(z)\mathrm{d}z. \tag{A.2}$$

Here $p(z)$ is the probability density function of Z and the integral is over all dimensions in z.

For a Gaussian random vector Z of N dimensions with covariance matrix Σ, by inserting the Gaussian pdf to Eq. (A.2) and taking the integral, we get that the differential entropy of Z is:

$$H(Z) = \frac{1}{2}\log\left((2\pi e)^N|\Sigma|\right), \tag{A.3}$$

where $|\Sigma|$ is the determinant of Σ.

Suppose now that X and Y are jointly Gaussian. Then,

$$p(X,Y) \sim N\left(\begin{bmatrix}\mu_X\\ \mu_Y\end{bmatrix}, \Sigma_{XY}\right), \tag{A.4}$$

where Σ_{XY} is the covariance matrix given as:

$$\Sigma_{XY} = \begin{bmatrix} \sigma_X^2 & \mathrm{COV}(X,Y) \\ \mathrm{COV}(X,Y) & \sigma_Y^2 \end{bmatrix}. \tag{A.5}$$

Using Eqs. (A.1) and (A.3) we get that the mutual information of X and Y is given by:

$$\begin{aligned} I(X,Y) &= -\frac{1}{2}\log\left(\frac{\sigma_X^2\sigma_Y^2}{\sigma_X^2\sigma_Y^2 - \mathrm{COV}(X,Y)^2}\right) = -\frac{1}{2}\log\left(\frac{1}{1-\frac{\mathrm{COV}(X,Y)^2}{\sigma_X^2\sigma_Y^2}}\right) \\ &= -\frac{1}{2}\log\left(\frac{1}{1-\rho_{XY}^2}\right), \end{aligned} \tag{A.6}$$

where

$$\rho_{XY} = \frac{\mathrm{COV}(X,Y)^2}{\sigma_X^2\sigma_Y^2}$$

is the correlation coefficient between X and Y.

In the TAN classifiers, the class is the parent of all features, and the features are Gaussian given a class label. Thus all the results above apply with an understanding that the distributions are conditioned on the class label (which is omitted for clarity). The class conditional mutual information between the pair X and Y is derived as follows:

$$\begin{aligned} I(X,Y|C) &= \sum_{c=1}^{|C|}\int\int p(x,y,c)\log\left(\frac{p(x,y|c)}{p(x|c)p(y|c)}\right)\mathrm{d}x\,\mathrm{d}y \\ &= \sum_{c=1}^{|C|}\int\int p(c)p(x,y|c)\log\left(\frac{p(x,y|c)}{p(x|c)p(y|c)}\right) \\ &= \sum_{c=1}^{|C|}p(c)I(X,Y|C=c) = -\frac{1}{2}\sum_{c=1}^{|C|}p(c)\log\left(\frac{1}{1-\rho_{XY|c}^2}\right). \end{aligned} \tag{A.7}$$

After finding the TAN structure, suppose that we find that feature X is the parent of Y. Given the class label, X and Y are jointly Gaussian with mean vector and covariance as defined in Eqs. (A.4) and (A.5) (again omitting the conditioning on the class variable for clarity). Since X is the parent of Y, we are interested in finding the parameters of the conditional distribution $p(Y|X)$ as a function of the parameters of the joint distribution. Because X and Y are jointly Gaussian, $Y|X$ is also Gaussian. Using $p(X,Y) = p(X)p(Y|X)$ and the Gaussian pdf, after some manipulations we get:

$$\begin{aligned} p(Y|X) &= \frac{p(X,Y)}{p(X)} = \frac{1}{(2\pi\sigma_Y^2(1-\rho_{XY}^2))^{1/2}}\exp\left(-\frac{(y-\mu_Y-ax)^2}{2\sigma_Y^2(1-\rho_{XY}^2)}\right) \\ &= N\big(\mu_Y + ax, \sigma_Y^2(1-\rho_{XY}^2)\big), \end{aligned} \tag{A.8}$$

where

$$a = \frac{\mathrm{COV}(X, Y)}{\sigma_X^2}.$$

References

[1] S. Baluja, Probabilistic modelling for face orientation discrimination: learning from labeled and unlabeled data, in: Neural Information Processing Systems (NIPS'98), 1998, pp. 854–860.
[2] M.J. Black, Y. Yacoob, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion, in: Proc. Internat. Conf. Computer Vision (ICCV'95), 1995, pp. 374–381.
[3] J.T. Cacioppo, L.G. Tassinary, Inferring psychological significance from physiological signals, Amer. Psychologist 45 (1990) 16–28.
[4] L.S. Chen, Joint processing of audio–visual information for the recognition of emotional expressions in human–computer interaction, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Department of Electrical Engineering, 2000.
[5] C.K. Chow, C.N. Liu, Approximating discrete probability distribution with dependence trees, IEEE Trans. Inform. Theory 14 (1968) 462–467.
[6] I. Cohen, Automatic facial expression recognition from video sequences using temporal information. MS Thesis, University of Illinois at Urbana-Champaign, Department of Electrical Engineering, 2000.
[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.
[8] L.C. De Silva, T. Miyasato, R. Natatsu, Facial emotion recognition using multimodal information. in: Proc. IEEE Internat. Conf. on Information, Communications and Signal Processing (ICICS'97), 1997, pp. 397–401.
[9] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, T.J. Sejnowski, Classifying facial actions, IEEE Trans. Pattern Anal. Machine Intell. 21 (10) (1999) 974–989.
[10] P. Ekman, Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique, Psychol. Bull. 115 (2) (1994) 268–287.
[11] P. Ekman, W.V. Friesen, Facial Action Coding System: Investigator's Guide, Consulting Psychologists Press, Palo Alto, CA, 1978.
[12] I.A. Essa, A.P. Pentland, Coding, analysis, interpretation, and recognition of facial expressions, IEEE Trans. Pattern Anal. Machine Intell. 19 (7) (1997) 757–763.
[13] J.H. Friedman, On bias, variance, 0/1-loss, and the curse-of-dimensionality, Data Mining Knowledge Discovery 1 (1) (1997) 55–77.
[14] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (2) (1997) 131–163.
[15] A. Garg, D. Roth, Understanding probabilistic classifiers, in: Proc. Eur. Conf. on Machine Learning, 2001, pp. 179–191.
[16] D. Goleman, Emotional Intelligence, Bantam Books, New York, 1995.
[17] G. Haas, L. Bain, C. Antle, Inferences for the Cauchy distribution based on maximum likelihood estimators, Biometrika 57 (2) (1970) 403–408.
[18] C.E. Izard, Innate and universal facial expressions: evidence from developmental and cross-cultural research, Psychol. Bull. 115 (2) (1994) 288–299.
[19] T. Kanade, J. Cohn, Y. Tian, Comprehensive database for facial expression analysis, 2000.
[20] R.M. Kay, Entropy and Information Theory, Springer, Berlin, 1990.
[21] A. Lanitis, C.J. Taylor, T.F. Cootes, A unified approach to coding and interpreting face images, in: Proc. 5th Internat. Conf. on Computer Vision (ICCV'95), 1995, pp. 368–373.
[22] S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilitic functions of a Markov process to automatic speech recognition, Bell Lab System Technical J. 62 (4) (1983) 1035–1072.

[23] J. Lien, Automatic recognition of facial expressions using hidden Markov models and estimation of expression intensity. Ph.D. Thesis, Carnegie Mellon University, 1998.
[24] A. Martinez, Face image retrieval using HMMs, in: IEEE Workshop on Content-based Access of Images and Video Libraries, 1999, pp. 35–39.
[25] K. Mase, Recognition of facial expression from optical flow, IEICE Trans. E74 (10) (1991) 3474–3483.
[26] A. Nefian, M. Hayes, Face recognition using an embedded HMM, in: IEEE Conf. on Audio and Video-based Biometric Person Authentication, 1999, pp. 19–24.
[27] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning 39 (2000) 103–134.
[28] N. Oliver, A. Pentland, F. Bérard, LAFTER: a real-time face and lips tracker with facial expression recognition, Pattern Recognition 33 (2000) 1369–1382.
[29] N. Oliver, A. Pentland, F. Bérard, LAFTER: lips and face real time tracker, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'97), 1997, pp. 123–129.
[30] T. Otsuka, J. Ohya, Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences, in: Proc. Internat. Conf. on Image Processing (ICIP'97), 1997, pp. 546–549.
[31] T. Otsuka, J. Ohya, A study of transformation of facial expressions based on expression recognition from temproal image sequences, Technical Report, Institute of Electronic, Information, and Communications Engineers (IEICE), 1997.
[32] M. Pantic, L.J.M. Rothkrantz, Automatic analysis of facial expressions: the state of the art, IEEE Trans. Pattern Anal. Machine Intell. 22 (12) (2000) 1424–1445.
[33] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech processing, Proc. IEEE 77 (2) (1989) 257–286.
[34] L.R. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[35] M. Rosenblum, Y. Yacoob, L.S. Davis, Human expression recognition from motion using a radial basis function network architecture, IEEE Trans. Neural Network 7 (5) (1996) 1121–1138.
[36] P. Salovey, J.D. Mayer, Emotional intelligence, Imagination, Cognition, Personality 9 (3) (1990) 185–211.
[37] N. Sebe, M.S. Lew, D.P. Huijsmans, Toward improved ranking metrics, IEEE Trans. Pattern Anal. Machine Intell. 22 (10) (2000) 1132–1143.
[38] H. Starks, J. Woods, Probability, Random Processes, and Estimation Theory for Engineers, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[39] H. Tao, T.S. Huang, Connected vibrations: a modal analysis approach to non-rigid motion tracking, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'98), 1998, pp. 735–740.
[40] N. Ueki, S. Morishima, H. Yamada, H. Harashima, Expression analysis/synthesis system based on emotion space constructed by multilayered neural network, Systems Comput. Japan 25 (13) (1994) 95–103.
[41] Y. Yacoob, L.S. Davis, Recognizing human facial expressions from long image sequences using optical flow, IEEE Trans. Pattern Anal. Machine Intell. 18 (6) (1996) 636–642.