Automatic recognition of facial expressions

Dragoş Datcu

November 2004

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Mediamatics: Man-Machine Interaction

Automatic recognition of facial expressions

Master's Thesis in Media & Knowledge Engineering

Man-Machine Interaction Group
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

Dragos Datcu
1138758

November 2004

Man-Machine Interaction Group
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

Members of the Supervising Committee
drs. dr. L.J.M. Rothkrantz
prof. dr. H. Koppelaar
prof. dr. ir. E.J.H. Kerckhoffs
ir. F. Ververs

Abstract

Automatic recognition of facial expressions

Dragos Datcu, student number: 1138758
Delft, November 2004

Man-Machine Interaction Group

Faculty of Electrical Engineering, Mathematics,
and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Members of the Supervising Committee

drs. dr. L.J.M. Rothkrantz
prof. dr. H. Koppelaar
prof. dr. ir. E.J.H. Kerckhoffs
ir. F. Ververs

The study of human facial expressions is one of the most challenging domains in pattern research community. Each facial expression is generated by non-rigid object deformations and these deformations are person dependent. The goal of this MSc. project is to design and implement a system for automatic recognition of human facial expression in video streams. The results of the project are of a great importance for a broad area of applications that relate to both research and applied topics.

As possible approaches on those topics, the following may be mentioned: automatic surveillance systems, the classification and retrieval of image and video databases, customer-friendly interfaces, smart environment human computer interaction and research in the field of computer assisted human emotion analyses. Some interesting implementations in the field of computed assisted emotion analysis concern experimental and interdisciplinary psychiatry. Automatic recognition of facial expressions is a process primarily based on analysis of permanent and transient features of the face, which can be only assessed with errors of some degree.

The expression recognition model is oriented on the specification of Facial Action Coding System (FACS) of Ekman and Friesen. The hard constraints on the scene processing and recording conditions set a limited robustness to the analysis. A probabilistic oriented framework is used in order to manage the uncertainties and lack of information. The support for the specific processing involved was given through a multimodal data fusion platform.

The Bayesian network is used to encode the dependencies among the variables. The temporal dependencies are to be extracted to make the system be able to properly select the right expression of emotion. In this way, the system is able to overcome the performance of the previous approaches that dealt only with prototypic facial expression.

Acknowledgements

The author would like to express his gratitude to his supervisor, Professor Drs. Dr. Leon Rothkrantz for facilitating the integration in the research field, for the trust and for all his support in making the project work come to an end.

Additionally, he thanks all his friends he met during his stay in Delft for all the constructive ideas, excellent advices and nice moments spent together. Special appreciation goes to the community of Romanian students at TUDelft.

Dragos Datcu

September, 2004

Table of content

List of figures

List of tables

INTRODUCTION

The study of human facial expressions is one of the most challenging domains in pattern research community. Each facial expression is generated by non-rigid object deformations and these deformations are person-dependent. The goal of the project was to design and implement a system for automatic recognition of human facial expression in video streams. The results of the project are of a great importance for a broad area of applications that relate to both research and applied topics. As possible approaches on those topics, the following may be presented: automatic surveillance systems, the classification and retrieval of image and video databases, customer friendly interfaces, smart environment human computer interaction and research in the field of computer assisted human emotion analyses. Some interesting implementations in the field of computed assisted emotion analysis concern experimental and interdisciplinary psychiatry.

Automatic recognition of facial expressions is a process primarily based on analysis of permanent and transient features of the face, which can be only assessed with errors of some degree. The expression recognition model is oriented on the specification of Facial Action Coding System (FACS) of Ekman and Friesen [Ekman, Friesen 1978]. The hard constraints on the scene processing and recording conditions set a limited robustness to the analysis. In order to manage the uncertainties and lack of information, we set a probabilistic oriented framework up. Other approaches based on Artificial Neuronal Networks have also been conducted as distinct experiments.

The support for the specific processing involved was given through a multimodal data fusion platform. In the Department of Knowledge Based Systems at T.U.Delft there has been a project based on a long-term research running on the development of a software workbench. It is called Artificial Intelligence aided Digital Processing Toolkit (A.I.D.P.T.) [Datcu, Rothkrantz 2004] and presents native capabilities for real-time signal and information processing and for fusion of data acquired from hardware equipments. The workbench also includes support for the Kalman filter based mechanism used for tracking the location of the eyes in the scene. The knowledge of the system relied on the data taken from the Cohn-Kanade AU-Coded Facial Expression Database [Kanade et al. 2000]. Some processing was done so as to extract the useful information. More than that, since the original database contained only one image having the AU code set for each display, additional coding had to be done. The Bayesian network is used to encode the dependencies among the variables. The temporal dependencies were extracted to make the system be able to properly select the right emotional expression. In this way, the system is able to overcome the performance of the previous approaches that dealt only with prototypic facial expression [Pantic, Rothkrantz 2003]. The causal relationships track the changes occurred in each facial feature and store the information regarding the variability of the data.

The typical problems of expression recognition have been tackled many times through distinct methods in the past. [Wang et al. 2003] proposed a combination of a Bayesian probabilistic model and Gabor filter. [Cohen et. all 2003] introduced a Tree- Augmented-Naive Bayes (TAN) classifier for learning the feature dependencies. A common approach was based on neural networks. [de Jongh, Rothkrantz 2004] used a neural network approach for developing an online Facial Expression Dictionary as a first step in the creation of an online Nonverbal Dictionary. [Bartlett et. all 2003] used a subset of Gabor filters selected with Adaboost and trained the Support Vector Machines on the outputs.

The next section describes the literature survey in the field of facial expression recognition. Subsequently, different models for the current emotion analysis system will be presented in a separate section. The next section presents the experimental results of the developed system and the final section gives some discussions on the current work and proposes possible improvements.

Project goal

The current project aims at the realization of a uni-modal, fully automatic human emotion recognition system based on the analysis of facial expressions from still pictures. The use of different machine learning techniques is to be investigated for testing against the performance achieved in the current context of the project.

The facial expression recognition system consists of a set of processing components that perform processing on the input video sequence. All the components are organized on distinct processing layers that interact for extracting the subtlety of human emotion and paralinguistic communication at different stages of the analysis.

The main component of the system stands for a framework that handles the communication with the supported processing components. Each component is designed in such a way so as to comply with the communication rules of the framework through well defined interfaces. For experimental purposes, multiple components having the same processing goal can be managed in the same time and parallel processing is possible through a multithreading framework implementation.

The following steps for building a facial expression recognition system are taken into consideration and detailed in the thesis:

- A throughout literature survey that aims at describing the recent endeavors of the research community on the realization of facial expression recognition systems.
- The presentation of the model, including the techniques, algorithms and adaptations made for an efficient determination of an automatic facial expression recognizer.
- The implementation of the models described and the tools, programming languages and strategies used for integrating all the data processing components into a single automatic system.
- The presentation of experimental setups for conducting a comprehensive series of tests on the algorithms detailed in the thesis. The presentation also includes the results of the tests that show the performance achieved by the models designed.

- The discussions on the current approach and the performance achieved by testing the facial expression recognition models.

All the processing steps are described in the current report. The final part contains the experiments run by using different functional approaches.

LITERATURE SURVEY

The recognition of facial expressions implies finding solutions to three distinct types of problems. The first one relates to detection of faces in the image. Once the face location is known, the second problem is the detection of the salient features within the facial areas. The final analysis consists in using any classification model and the extracted facial features for identifying the correct facial expression. For each of the processing steps described, there have been developed lots of methods to tackle the issues and specific requirements. Depending on the method used, the facial feature detection stage involves global or local analysis.

The internal representation of the human face can be either 2D or 3D. In the case of global analysis, the connection with certain facial expressions is made through features determined by processing the entire face. The efficiency of methods as Artificial Neural Networks or Principal Component Analysis is greatly affected by head rotation and special procedures are needed to compensate the effects of that. On the other hand, local analysis performs encoding of some specific feature points and uses them for recognition. The method is actually used in the current paper. However, other approaches have been also performed at this layer. One method for the analysis is the internal representation of facial expressions based on collections of Action Units (AU) as defined in Facial Action Coding System (FACS) [Ekman W.V.Friesen, 1978] [Bartlett et al., 2004]. It is one of the most efficient and commonly used methodologies to handle facial expressions.

The use of fiducial points on the face with the geometric positions and multi-scale and multi-orientation Gabor wavelet coefficients have been investigated by [Zhang, 1998; Zhang, 1999]. The papers describe the integration within an architecture based on a two-layer perceptron. According to the results reported by the author, the Gabor wavelet coefficients show much better results for facial expression recognition, when compared to geometric positions.

In the paper [Fellenz et al., 1999] the authors compare the performance and the generalizing capabilities of several low-dimensional representations for facial expression recognition in static pictures, for three separate facial expressions plus the neutral class. Three algorithms are presented: the template-based determined by computing the average face for each emotion class and then performing matching of one sample to the templates, the multi-layered perceptron trained with the back-propagation of error algorithm and a neural algorithm that uses six odd-symmetric and six even-symmetric Gabor features computed from the face image. According the authors, the template-based approach presented 75% correct classification while the generalization achieved only 50%, the multilayered perceptron has 40% to 80% correct recognition, depending on test the data set. The third approach did not provide an increase in the performance of the facial expression recognition.

Several research works study the facial dynamics of recognition of facial expressions. The work of [Yacoob and Davis, 1994] uses optical flow to identify the direction of rigid and non-rigid motions shown by facial expressions. The results range from 80% for sadness to 94% for surprise emotion on a set of 46 image sequences recorded from 30 subjects, for six facial expressions.

Some attempts to automatically detect the salient facial features implied computing descriptors such as scale-normalized Gaussian derivatives at each pixel of the facial image and performing some linear-combinations on their values. It was found that a single cluster of Gaussian derivative responses leads to a high robustness of detection given the pose, illumination and identity [Gourier et al., 2004]. A representation based on topological labels is proposed [Yin et al., 2004]. It assumes that the facial expression is dependent on the change of facial texture and that its variation is reflected by the modification of the facial topographical deformation. The classification is done by comparing facial features with those of the neutral face in terms of the topographic facial surface and the expressive regions. Some approaches firstly model the facial features and then use the parameters as data for further analysis such as expression recognition. The system proposed by [Moriyama et al., 2004] is based on a 2D generative eye model that implements encoding of the motion and fine structures of the eye and is used for tracking the eye motion in a sequence. As concerning the classification methods, various algorithms have been developed, adapted and used during time [Pantic and Rothkrantz, 2000].

Neural networks have been used for face detection and facial expression recognition [Stathopoulou and Tsihrintzis, 2004] [deJong and Rothkrantz, 2004]. The second reference directs to a system called Facial Expression Dictionary (FED) [deJong and Rothkrantz, 2004] that was a first attempt to create an online nonverbal dictionary.

The work of [Padgett et al., 1996] present an algorithm based on an ensemble of simple feed-forward neural networks capable of identifying six different basic emotions. The initial data set used for training their system included 97 samples of 6 male and 6 female subject, considered to portray only unique emotions. The overall performance rate of the approach was reported to be 86% on novel face images. The authors use the algorithm for analyzing sequences of images showing the transition between two distinct facial expressions. The sequences of images were generated based on morph models.

[Schweiger et al., 2004] proposed a neural architecture for temporal emotion recognition. The features used for classification were selected by using optical flow algorithm in specific bounding boxes on the face. Separate Fuzzy ARTMAP neural networks for each emotional class were trained using incremental learning. The authors conducted experiments for testing the performance of their algorithms using Cohn-Kanade database.

Other classifiers included Bayesian Belief Networks (BBN) [Datcu and Rothkrantz, 2004], Expert Systems [Pantic and Rothkrantz, 2000] or Support Vector Machines (SVM) [Bartlett et al., 2004]. Other approaches have been oriented on the analysis of data gathered from distinct multi-modal channels. They combined multiple methods for processing and applied fusion techniques to get to the recognition stage [Fox and Reilly, 2004].

The work of [Bourel et al., 2001] presents an approach for recognition of facial expressions from video in conditions of occlusion by using a localized representation of facial expressions and on data fusion.

In [Wang and Tang, 2003] the authors proposed a combination of a Bayesian probabilistic model and Gabor filter. [Cohen et al., 2003] introduced a Tree-Augmented-Naive Bayes (TAN) classifier for learning the feature dependencies.

The system presented in [Bartlett et al., 2003] is able to automatically detect frontal faces in video and to code the faces according to the six basic emotions plus the neutral. For the detection of faces, the authors use a cascade of feature detectors based on boosting techniques. The face detector outputs image patches to the facial expression recognizer. A bank of SVM classifiers make use of Gabor-based features that are computed from the image patches. Each emotion class is handled by a distinct SVM classifier. The paper presents the results for the algorithms that involved linear and RBF kernels for being used by the SVM classifiers. Adaboost algorithm was used for select the relevant set of features from an initial set of 92160 features.

[Jun et al., 2000] propose a system for the recognition of facial expressions based on Independent Component Analysis - ICA algorithm and Linear Discriminant Analysis - LDA. The algorithm implies the use of ICA for obtaining a set of independent basis images of the face image and the use of LDA for selecting features obtained by ICA. The authors provide the results for each experimental setup for the use of the two methods. They report the highest recognition ration for the case of using LDA and ICA together (95.6%).

[Feng et al., 2000] proposes a sub-band approach in using Principal Component Analysis – PCA. In comparison with the traditional use of PCA namely for the whole facial image, the method described in the paper gives better recognition accuracy. Additionally, the method achieves a reduction of the computational load in the cases the image database is large, with more than 256 training images. The facial expression accuracy ranges from 45.9% in the case of using 4X4 wavelet transform to 84.5% for a size of 16X16. The accuracy of the recognition is improved by 6%, from 78.7% in case of using the traditional approach to 84.5% using sub-band 10. The analysis are carried for wavelet transform ranging from sub-band 1 to sub-band 16 and full size original image.

MODEL

Material and Method

Contraction of facial muscles produces changes in the data collected through different visual channels from the input video sequence. Each channel is assigned to a distinct facial feature.

The preliminary image processing component has the goal to detect the variances that occurs in the appearance of permanent and transient facial features. That is done by tracking the position of specific points on the face surface. The method used implies the detection of the position of the eyes in infra red illumination condition.

The coordinates of the eyes in the current frame are used as constrains for the further detection of the coordinates of the other facial features. The procedure does not require assuming a static head or any initial calibration. By including a Kalman filter based enhancement mechanism, the eye tracker can perform robust and accurate calibration estimation.

The recognition can be performed by using information related to the relative position of the detected local feature points. The accuracy of the recognizer can be improved by including also information concerning the temporal behavior of the feature points.

Data preparation

Starting from the image database, we processed each image and obtained the set points according to an enhanced model that was initially based of 30 points according to Kobayashi & Hara model [Kobayashi, Hara 1972] Figure 1. The analysis was semiautomatic.

A new transformation was involved then to get the key points as described in Figure 2. The coordinates of the last set of points were used for computing the values of the parameters presented in Table 2. The preprocessing tasks implied some additional requirements to be satisfied. First, for each image a new coordinate system was set. The origin of the new coordinate system was set to the nose top of the individual.

The value of a new parameter called base was computed to measure the distance between the eyes of the person in the image. The next processing was the rotation of all the points in the image with respect to the center of the new coordinate system.

The result was the frontal face with correction to the facial inclination. The final step of preprocessing was related to scale all the distances so as to be invariant to the size of the image. Eventually a set of 15 values for each of the image was obtained as the result of preprocessing stage. The parameters were computed by taking both the variance observed in the frame at the time of analysis and the temporal variance. Each of the last three parameters was quantified so as to express a linear behavior with respect to the range of facial expressions analyzed.

The technique used was Principal Component Analysis oriented pattern recognition for each of the three facial areas. Principal Components Analysis (PCA) is a procedure which rotates the image data such that maximum variability is projected onto the axes. Essentially, a set of correlated variables, associated to the characteristics of the chin, forehead and nasolabial area, are transformed into a set of uncorrelated variables which are ordered by reducing variability. The uncorrelated variables are linear combinations of the original variables, and the last of these variables can be removed with minimum loss of real data. The technique was first applied by Turk and Pentland for face imaging [Turk and Pentland 1991].

The PCA processing is run separately for each area and three sets of eigenvectors are available as part of the knowledge of the system. Moreover, the labeled patterns associated with each area are stored (Figure 3).

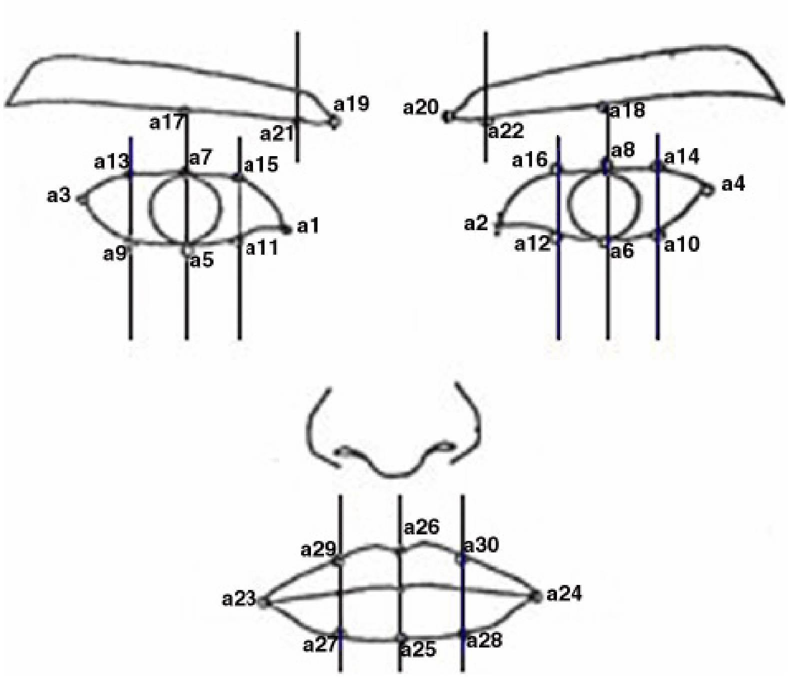


Figure 1. Kobayashi & Hara 30 FCPs model

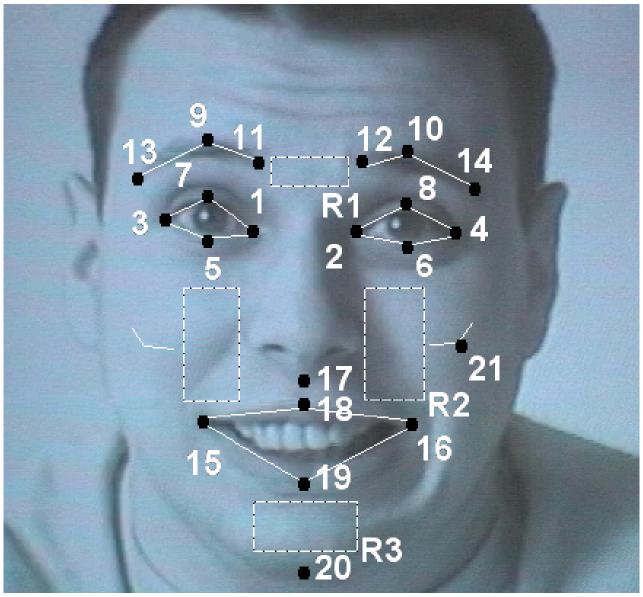


Figure 2. Facial characteristic points model

The computation of the eigenvectors was done offline as a preliminary step of the process. For each input image, the first processing stage extracts the image data according to the three areas. Each data image is projected through the eigenvectors and the pattern with the minimum error is searched.

The label of the extracted pattern is then fed to the quantification function for obtaining the characteristic output value of each image area. Each value is further set as evidence in the probabilistic BBN.

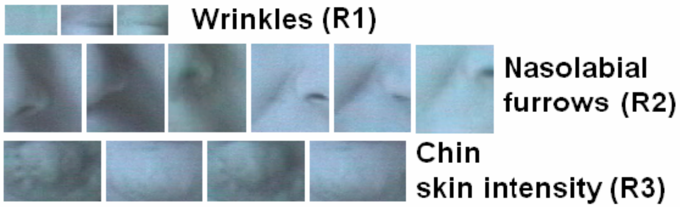
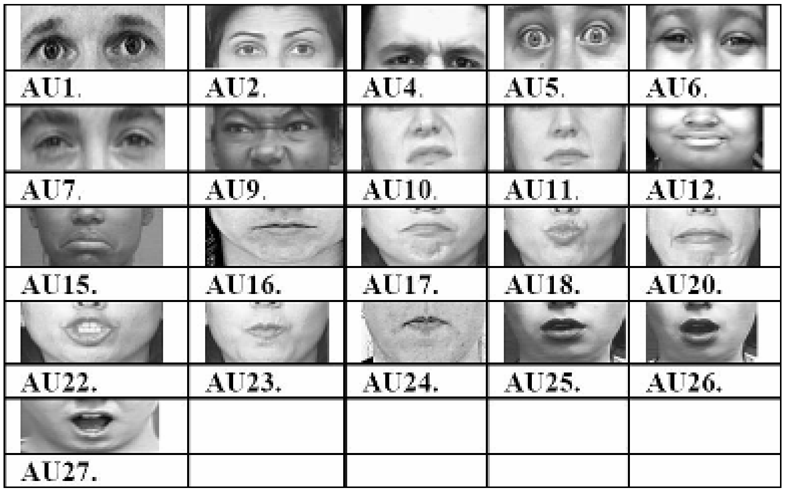


Figure 3. Examples of patterns used in PCA recognition

Model parameters

The Bayesian Belief Network encodes the knowledge of the existent phenomena that triggers changes in the aspect of the face. The model does include several layers for the detection of distinct aspects of the transformation. The lowest level is that of primary parameter layer. It contains a set of parameters that keeps track of the changes concerning the facial key points. Those parameters may be classified as static and dynamic. The static parameters handle the local geometry of the current frame. The dynamic parameters encode the behavior of the key points in the transition from one frame to another. By combining the two sorts of information, the system gets a high efficiency of expression recognition. An alternative is that the base used for computing the variation of the dynamic parameters is determined as a previous tendency over a limited past time. Each parameter on the lowest layer of the BBN has a given number of states. The purpose of the states is to map any continuous value of the parameter to a discrete class. The number of states has a direct influence on the efficiency of recognition. The number of states for the low-level parameters does not influence the time required for obtaining the final results. It is still possible to have a real time implementation even when the number of states is high.

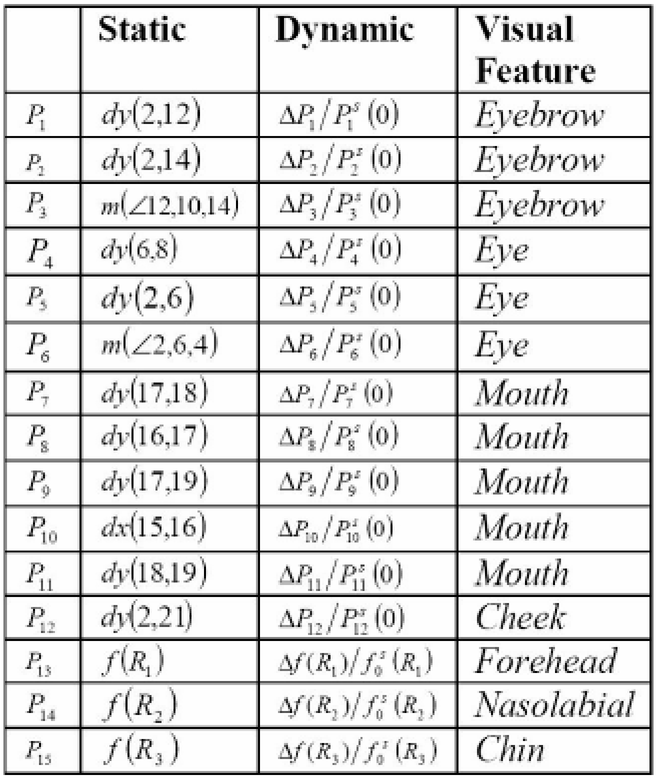
Table 1. The used set of Action Units



AU1.	AU2.	AU4.	AU5.	AU6.
AU7.	AU9.	AU10.	AU11.	AU12.
AU15.	AU16.	AU17.	AU18.	AU20.
AU22.	AU23.	AU24.	AU25.	AU26.
AU27.				

The only additional time is that of processing done for computing the conditioned probability tables for each BBN parameter, but the task is run off-line.

Table 2. The set of visual feature parameters



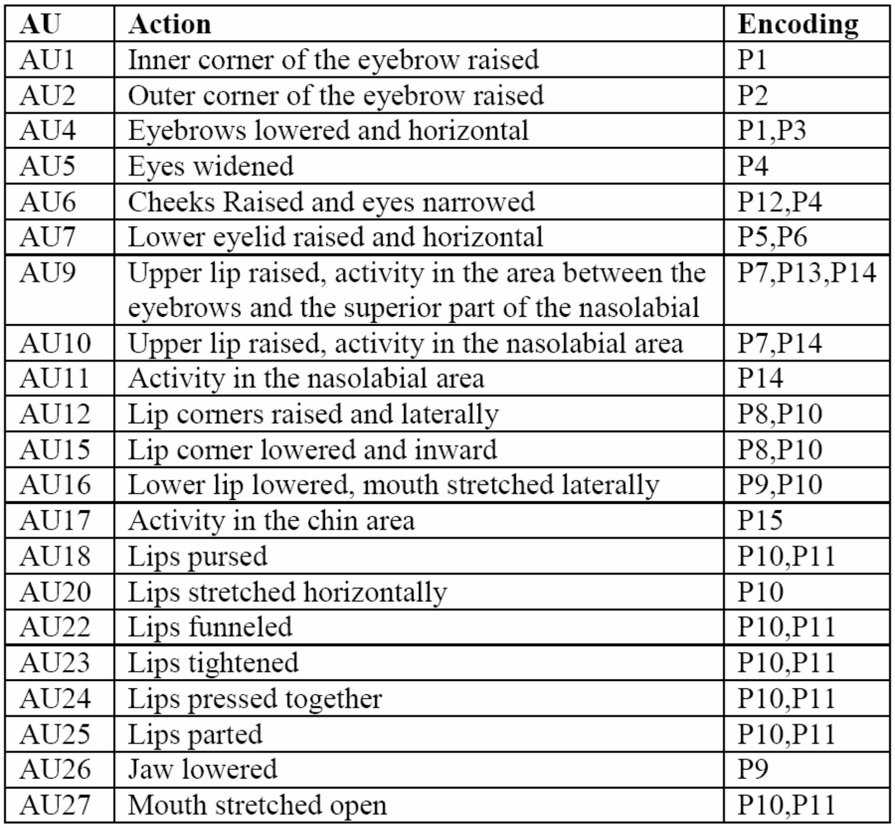
	Static	Dynamic	Visual Feature
P_1	$dy(2,12)$	$\Delta P_1 / P_1^s(0)$	*Eyebrow*
P_2	$dy(2,14)$	$\Delta P_2 / P_2^s(0)$	*Eyebrow*
P_3	$m(\angle 12,10,14)$	$\Delta P_3 / P_3^s(0)$	*Eyebrow*
P_4	$dy(6,8)$	$\Delta P_4 / P_4^s(0)$	*Eye*
P_5	$dy(2,6)$	$\Delta P_5 / P_5^s(0)$	*Eye*
P_6	$m(\angle 2,6,4)$	$\Delta P_6 / P_6^s(0)$	*Eye*
P_7	$dy(17,18)$	$\Delta P_7 / P_7^s(0)$	*Mouth*
P_8	$dy(16,17)$	$\Delta P_8 / P_8^s(0)$	*Mouth*
P_9	$dy(17,19)$	$\Delta P_9 / P_9^s(0)$	*Mouth*
P_{10}	$dx(15,16)$	$\Delta P_{10} / P_{10}^s(0)$	*Mouth*
P_{11}	$dy(18,19)$	$\Delta P_{11} / P_{11}^s(0)$	*Mouth*
P_{12}	$dy(2,21)$	$\Delta P_{12} / P_{12}^s(0)$	*Cheek*
P_{13}	$f(R_1)$	$\Delta f(R_1) / f_0^s(R_1)$	*Forehead*
P_{14}	$f(R_2)$	$\Delta f(R_2) / f_0^s(R_2)$	*Nasolabial*
P_{15}	$f(R_3)$	$\Delta f(R_3) / f_0^s(R_3)$	*Chin*

The action units examined in the project include facial muscle movements such as inner eyebrow raise, eye widening, and so forth, which combine to form facial expressions. Although prior methods have obtained high recognition rates for recognizing facial action units, these methods either use manually pre-processed image sequences or require human specification of facial features; thus, they have exploited substantial human intervention. According to the method used, each facial expression is described as a combination of existent Action Units (AU). One AU represents a specific facial display. Among 44 AUs contained in FACS, 12 describe contractions of specific facial muscles in the upper part of the face and 18 in the lower part. Table 1 presents the set of AUs that is managed by the current recognition system. An important characteristic of the AUs is that they may act differently in given combinations.

According to the behavioral side of each AU, there are additive and non-additive combinations. In that way, the result of one non-additive combination may be related to a facial expression that is not expressed by the constituent AUs taken separately. In the case of the current project, the AU sets related to each expression are split into two classes that specify the importance of the emotional load of each AU in the class. By means of that, there are primary and secondary AUs. The AUs being part of the same class are additive. The system performs recognition of one expression as computing the probability associated with the detection of one or more AUs from both classes.

The probability of one expression increases, as the probabilities of detected primary AUs get higher. In the same way, the presence of some AUs from a secondary class results in solving the uncertainty problem in the case of the dependent expression but at a lower level.

Table 3. The dependency between AUs and intermediate parameters



AU	Action	Encoding
AU1	Inner corner of the eyebrow raised	P1
AU2	Outer corner of the eyebrow raised	P2
AU4	Eyebrows lowered and horizontal	P1,P3
AU5	Eyes widened	P4
AU6	Cheeks Raised and eyes narrowed	P12,P4
AU7	Lower eyelid raised and horizontal	P5,P6
AU9	Upper lip raised, activity in the area between the eyebrows and the superior part of the nasolabial	P7,P13,P14
AU10	Upper lip raised, activity in the nasolabial area	P7,P14
AU11	Activity in the nasolabial area	P14
AU12	Lip corners raised and laterally	P8,P10
AU15	Lip corner lowered and inward	P8,P10
AU16	Lower lip lowered, mouth stretched laterally	P9,P10
AU17	Activity in the chin area	P15
AU18	Lips pursed	P10,P11
AU20	Lips stretched horizontally	P10
AU22	Lips funneled	P10,P11
AU23	Lips tightened	P10,P11
AU24	Lips pressed together	P10,P11
AU25	Lips parted	P10,P11
AU26	Jaw lowered	P9
AU27	Mouth stretched open	P10,P11

The conditioned probability tables for each node of the Bayesian Belief Network were filled in by computing statistics over the database. The Cohn-Kanade AU-Coded Facial Expression Database contains approximately 2000 image sequences from 200 subjects ranged in age from 18 to 30 years. Sixty-five percent were female, 15 percent were

African-American and three percent were Asian or Latino. All the images analyzed were frontal face pictures. The original database contained sequences of the subjects performing 23 facial displays including single action units and combinations. Six of the displays were based on prototypic emotions (joy, surprise, anger, fear, disgust and sadness).

IR eye tracking module

The system was designed as a fully automatic recognizer of facial expressions at the testing session. Moreover, all the computations have to take place in a real-time manner. The only manual work involved was at the stage of building the system's knowledge database.

In order to make the system automatic for real experiments, the estimation of the eye position was made by using low level processing techniques on the input video signal.

By illuminating the eye with infra red leds, a dark and a bright pupil image is obtained for the position of the eyes. The detection component searches for the pixels in the image having the best matching rank to the characteristics of the area.

1. Convert the input image associated to the current frame in the sequence to gray-level.
2. Apply Sobel edge detection on the input image.
3. Filter the points in terms of the gray levels. Take as candidates all the points having white ink concentration above the threshold.
4. For every candidate point compute the median and variation of the points in the next area with respect to the ink concentration. The searched area is as it is illustrated in image.
5. Remove all the candidates that have median and variation of neighborhood pixels above threshold values from the list.
6. Take the first two candidates that have the higher value of the white ink concentration from the list.

Emotion recognition using BBN

The expression recognition is done computing the anterior probabilities for the parameters in the BBN (Figure 4). The procedure starts by setting the probabilities of the parameters on the lowest level according to the values computed at the preprocessing stage. In the case of each parameter, evidence is given for both static and dynamic parameters. Moreover, the evidence is set also for the parameter related to the probability of the anterior facial expression. It contains 6 states, one for each major class of expressions. The aim of the presence of the anterior expression node and that associated with the dynamic component of one given low-level parameter, is to augment the inference process with temporal constrains. The structure of the network integrates parametric layers having different functional tasks. The goal of the layer containing the first AU set and that of the low-level parameters is to detect the presence of some AUs in the current frame. The relation between the set of the low level parameters and the action units is as it is detailed in Table 3.

The dependency of the parameters on AUs was determined on the criteria of influence observed on the initial database. The presence of one AU at this stage does not imply the existence of one facial expression or another. Instead, the goal of the next layer containing the AU nodes and associated dependencies is to determine the probability that one AU presents influence on a given kind of emotion.

The final parametric layer consists of nodes for every emotional class. More than that, there is also one node for the current expression and another one for that previously detected. The top node in the network is that of current expression. It has two states according to the presence and absence of any expression and stands for the final result of analysis. The absence of any expression is seen as a neutral display of the person’s face on the current frame. While performing recognition, the BBN probabilities are updated in a bottom-up manner. As soon as the inference is finished and expressions are detected, the system reads the existence probabilities of all the dependent expression nodes. The most probable expression is that given by the larger value over the expression probability set.

Table 4. The emotion projections of each AU combination

	Primary Aus						Secondary AUs			
Surprise	1+2	5	25	26	27					
Fear	1+2	4	5	7	20	25	6	10	11	
Happy	6+12	12					7	16	25	26
Sadness	15+17	4+17	1				7	24	20+25	
Disgust	4	7+9	17	6+7			23	24	25	
Anger	4	4+7	23	24			17			

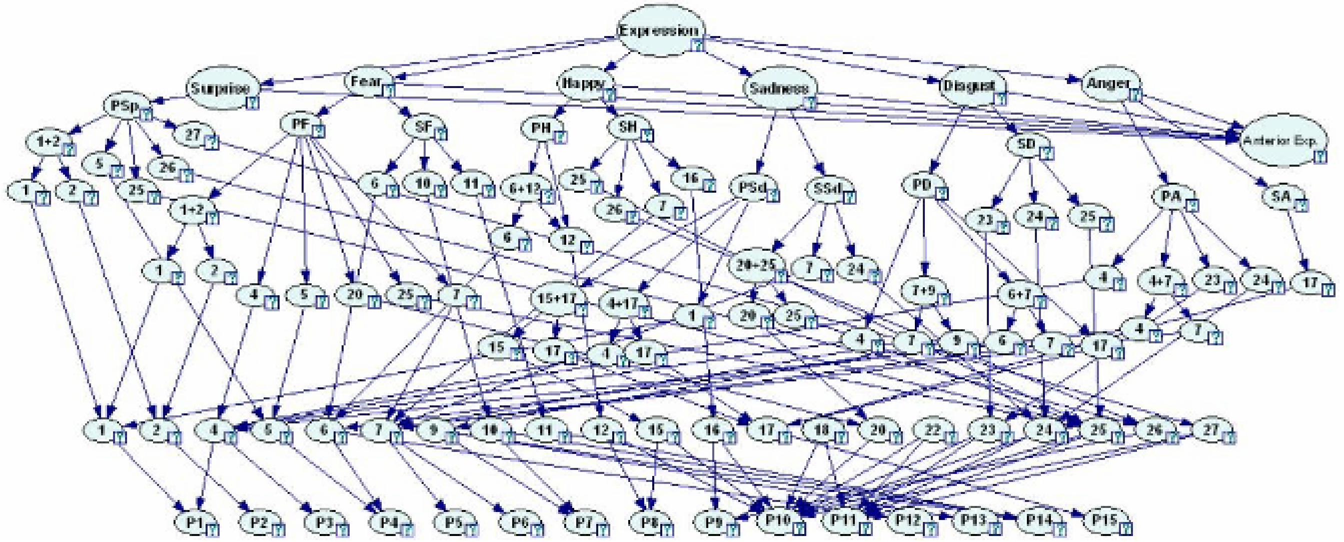


Figure 4. BBN used for facial expression recognition

For one of the conducted experiments, GeNIe Learning Tool was used for doing structure learning based on the data. The tool was configured so as to take into account the next learning rules:

- discovery of causal relationships had to be done only among the parameters representing the Action Units and those representing the model parameters
- causal relationships among AU parameters did not exist
- causal relationships among model parameters did not exist
- all the causal relationships among the parameter representing the emotional expressions and the others, representing the model parameters, were given

The structure of the resulted Bayesian network is presented in Figure 5.
The associated emotion recognition rate was 63.77 %.

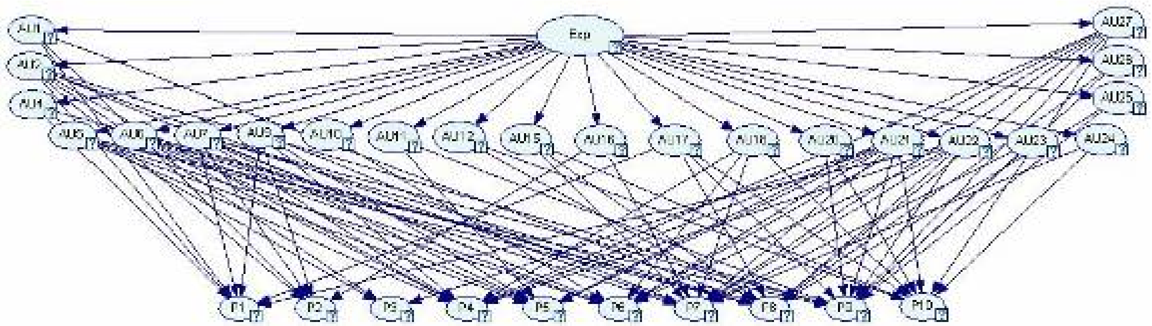


Figure 5. AU classifier discovered structure

The implementation of the model was made using C/C++ programming language. The system consists in a set of applications that run different tasks that range from pixel/image oriented processing to statistics building and inference by updating the probabilities in the BBN model. The support for BBN was based on S.M.I.L.E.

(Structural Modeling, Inference, and Learning Engine), a platform independent library of C++ classes for reasoning in probabilistic models [M.J.Druzdzel 1999]. S.M.I.L.E. is freely available to the community and has been developed at the Decision Systems Laboratory, University of Pittsburgh. The library was included in the AIDPT framework. The implemented probabilistic model is able to perform recognition on six emotional classes and the neutral state. By adding new parameters on the facial expression layer, the expression number on recognition can be easily increased.

Accordingly, new AU dependencies have to be specified for each of the emotional class added. In Figure 7 there is an example of an input image sequence. The result is given by the graphic containing the information related to the probability of the dominant detected facial expression (Figure 6).

In the current project the items related to the development of an automatic system for facial expression recognition in video sequences were discussed. As the implementation, a system was made for capturing images with an infra red camera and for processing them in order to recognize the emotion presented by the person. An enhancement was provided for improving the visual feature detection routine by including a Kalman-based eye tracker. The inference mechanism was based on a probabilistic framework. Other kinds of reasoning mechanisms were used for performing recognition. The Cohn-Kanade AU-Coded Facial Expression Database was used for building the system knowledge. It contains a large sample of varying age, gender and ethnic background and so the robustness to the individual changes in facial features and behavior is high. The BBN model takes care of the variation and degree of uncertainty and gives us an improvement in the quality of recognition. As off now, the results are very promising and show that the new approach presents high efficiency. The further work is focused on the replacement of the feature extraction method at the preprocessing stage. It is more convenient to adapt the system to be working in a regular context without the need of infra red light to be present in order to make the key point detection available.

A BBN model was also created for encoding also temporal behavior of certain parameters.

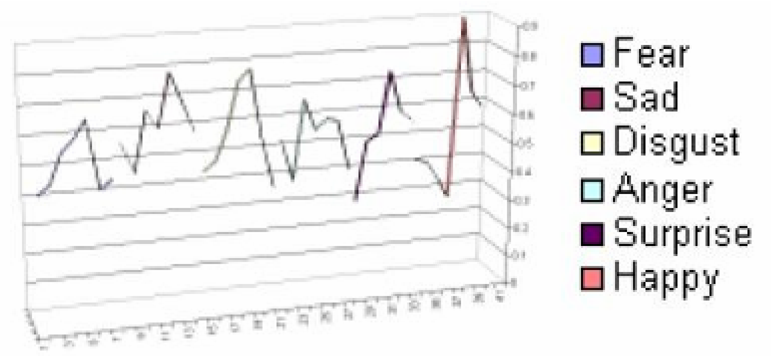


Figure 6. Dominant emotion in the sequence

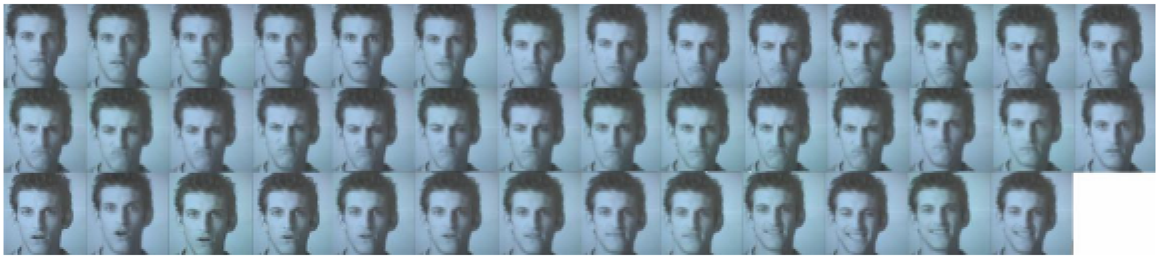


Figure 7. Examples of expression recognition applied on video streams

Bayesian Belief Networks (BBN)

Bayesian networks are knowledge representation formalisms for reasoning under uncertainty. A Bayesian network is mathematically described as a graphical representation of the joint probability distribution for a set of discrete variables.

Each network is a direct acyclic graph encoding assumptions of conditional independence. The nodes are stochastic variables and arcs are dependency between nodes.

For each variable there exists a set of values related to the conditional probability of the parameter given its parents. The joint probability distribution of all variables is then the product of all attached conditional probabilities.

Bayesian networks are statistics techniques, which provide explanation about the inferences and influences among features and classes of a given problem. The goal of the research project was to make use of BBN for recognition of six basic facial expressions. Every expression was analyzed for determining the connections and the nature of different causal parameters. The graphical representation made Bayesian networks a flexible tool for constructing recognition models of causal impact between events. Also, specification of probabilities is focused to very small parts of the model (a variable and its parents).

A particular use of BBN is for handling models that have causal impact of a random nature. In the context of the current project, there have been developed networks to handle the changes of the human face by taking into account local and temporal behavior of associated parameters.

Having constructed the model, it was used to compute effects of information as well as interventions. That is, the state of some variables was fixed, and the posterior probability distributions for the remaining variables were computed.

By using software of Bayesian network models construction, different Bayesian network classifier models could be generated, using the extracted given features in order to verify their behavior and probabilistic influences and used as the input to Bayesian network, some tests were performed in order to build the classifier.

Bayesian networks were designed to encode explicitly encode "deep knowledge" rather than heuristics, to simplify knowledge acquisition, provide a firmer theoretical ground, and foster reusability.

The idea of Bayesian networks is to build a network of causes and effects. Each event, generally speaking, can be certain or uncertain. When there is a new piece of evidence, this is transmitted to the whole network and all the beliefs are updated. The research activity in this field consists of the most efficient way of doing the calculation, using Bayesian inference, graph theory, and numerical approximations.

The BBN mechanisms are close to the natural way of human reasoning, the initial beliefs can be those of experts (avoiding the long training needed to set up, for example, neural networks, unfeasible in practical applications), and they learn by experience as soon as they start to receive evidence.

Bayes Theorem

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

In the formula,

$P(h)$ is prior probability of hypothesis h

$P(D)$ is prior probability of training data D

$P(h \mid D)$is probability of h given D

$P(D \mid h)$ is probability of D given h

Choosing Hypotheses

Generally want the most probable hypothesis given the training data

Maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\left\langle \max_{h \in H} P(h / D) \right\rangle \\ &= \arg\left\langle \max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \right\rangle \\ &= \arg\left\langle \max_{h \in H} P(D \mid h)P(h) \right\rangle \end{aligned}$$

For the case that $P(h_i) = P(h_j)$, a simplification can be further done by choosing the Maximum likelihood (ML) hypothesis:

$$h_{ML} = \arg\left\langle \max_{h_i \in H} P(D / h_i) \right\rangle$$

The Bayesian network is a graphical model that efficiently encodes the joint probability distribution for a given set of variables.

A Bayesian network for a set of variables $X = \{X_1, ..., X_n\}$ consists of a network structure *S* that encodes a set of conditional independence assertions about variables in X , and a set *P* of local probability distributions associated with each variable. Together, these components define the joint probability distribution for X . The network structure *S* is a directed acyclic graph. The nodes in *S* are in one-to-one correspondence with the variables X . The term X_i is used to denote both the variable and the corresponding node, and pa_i to denote the parents of node X_i in *S* as well as the variables corresponding to those parents.

Given the structure *S*, the joint probability distribution for *X* is given by:

Equation 1

$$p(x) = \prod_{i=1}^{n} p(x_i \mid pa_i)$$

The local probability distributions *P* are the distributions corresponding to the terms in the product of Equation 1. Consequently, the pair (*S*;*P*) encodes the joint distribution $p(x)$.

The probabilities encoded by a Bayesian network may be Bayesian or physical. When building Bayesian networks from prior knowledge alone, the probabilities will be Bayesian. When learning these networks from data, the probabilities will be physical (and their values may be uncertain).

Difficulties are not unique to modeling with Bayesian networks, but rather are common to most approaches.

As part of the project several tasks had to be fulfilled, such as:

- correctly identify the goals of modeling (e.g., prediction versus explanation versus exploration)
- identify many possible observations that may be relevant to the problem
- determine what subset of those observations is worthwhile to model
- organize the observations into variables having mutually exclusive and collectively exhaustive states

In the next phase of Bayesian-network construction, a directed acyclic graph was created for encoding assertions of conditional independence. One approach for doing so is based on the following observations. From the chain rule of probability the relation can be written as:

Equation 2

$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_1, ..., x_{i-1})$$

For every X_i there will be some subset $\Pi_i = \{X_1, ..., X_{i-1}\}$ such that X_i and $\{X_1, ..., X_{i-1}\} \setminus \Pi_i$ are conditionally independent given Π_i. That is, for any x,

Equation 3

$$p(x_i \mid x_1, ..., x_{i-1}) = p(x_i \mid \pi_i)$$

Combining the two previous equations, the relation becomes:

Equation 4

$$p(x) = \prod_{i=1}^{n} p(x_i \mid \pi_i)$$

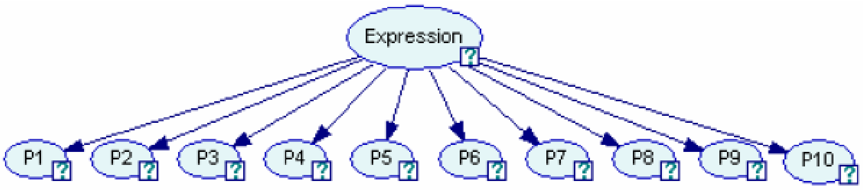


Figure 8. Simple model for facial expression recognition

The variables sets $(\Pi_1, ..., \Pi_n)$ correspond to the Bayesian-network parents $(Pa_1, ..., Pa_n)$, which in turn fully specify the arcs in the network structure *S*.

Consequently, to determine the structure of a Bayesian network, the proper tasks are set to:

- order the variables in a given way
- determine the variables sets that satisfy Equation 3 for $i = 1,...,n$

In the given example, using the ordering $(Expression, P_1, P_2, P_3..., P_{10})$, the conditional independencies are:

Equation 5

$$
\begin{aligned}
&p(P_1 \mid Expression) = p(P_1 \mid Expression)\\
&p(P_2 \mid Expression, P_1) = p(P_2 \mid Expression)\\
&p(P_3 \mid Expression, P_1, P_2) = p(P_3 \mid Expression)\\
&...\\
&p(P_{10} \mid Expression, P_1, P_2,..., P_{10}) = p(P_{10} \mid Expression)
\end{aligned}
$$

The according network topology is as it is represented in Figure 8.

Arcs are drawn from cause to effect. The local probability distribution(s) associated with a node are shown adjacent to the node.

This approach has a serious drawback. If the variable order is carelessly chosen, the resulting network structure may fail to reveal many conditional independencies among the variables. In the worst case, there are $n!$ variable orderings to be explored so as to find the best one. Fortunately, there is another technique for constructing Bayesian networks that does not require an ordering.

The approach is based on two observations:

- people can often readily assert causal relationships among variables
- causal relationships typically correspond to assertions of conditional dependence

In the particular example, to construct a Bayesian network for a given set of variables, the arcs were drawn from cause variables to their immediate effects. In almost all cases, doing so results in a network structure that satisfies the definition Equation 1.

For the experiments 3 and 4, there has been used a tool that is part of the GeNIe version 1.0 for learning the best structure of the BBN with respect to the existent causal relationships. It is called GeNIe Learning Wizard and can be used to automatically learn causal models from data.

In the final step of constructing a Bayesian network, the local probability distributions $p(x_i \mid pa_i)$ were assessed. In the example, where all variables are discrete, one distribution for X_i was assessed for every configuration of Pa_i. Example distributions are shown in Figure 8.

Inference in a Bayesian Network

Once a Bayesian network has been constructed (from prior knowledge, data, or a combination), the next step is to determine various probabilities of interest from the model. In the problem concerning detection of facial expressions, the probability of the existence of happiness expression, given observations of the other variables is to be discovered. This probability is not stored directly in the model, and hence needs to be computed. In general, the computation of a probability of interest given a model is known as probabilistic inference. Because a Bayesian network for X determines a joint probability distribution for X, the Bayesian network was used to compute any probability of interest. For example, from the Bayesian network in Figure 8, the probability of a certain expression given observations of the other variables can be computed as follows:

Equation 6

$$p(Expression \mid P_1, P_2, ..., P_{10}) = \frac{p(Expression, P_1, P_2, ..., P_{10})}{p(P_1, P_2, ..., P_{10})} = \frac{\text{p(Expression}, P_1, P_2, ..., P_{10})}{\underset{\text{Expression'}}{} \text{p(Expression}', P_1, P_2, ..., P_{10})}$$

For problems with many variables, however, this direct approach is not practical. Fortunately, at least when all variables are discrete, the conditional independencies can be exploited encoded in a Bayesian network to make the computation more efficient. In the example, given the conditional independencies in Equation 5, Equation 6 becomes:

Equation 7

$$p(Expression \mid P_1, P_2, ..., P_{10}) = \frac{\text{p(Expression)p(P}_1 \mid \text{Expression)...p(P}_{10} \mid \text{Expression)}}{\underset{\text{Expression'}}{} \text{p(Expression}'\text{)p(P}_1 \mid \text{Expression}'\text{)...p(P}_{10} \mid \text{Expression}'\text{)}}$$

Several researchers have developed probabilistic inference algorithms for Bayesian networks with discrete variables that exploit conditional independence. Pearl (1986) developed a message-passing scheme that updates the probability distributions for each node in a Bayesian network in response to observations of one or more variables. Lauritzen and Spiegelhalter (1988), Jensen et al. (1990), and Dawid (1992) created an algorithm that first transforms the Bayesian network into a tree where each node in the tree corresponds to a subset of variables in X. The algorithm then exploits several mathematical properties of this tree to perform probabilistic inference.
The most commonly used algorithm for discrete variables is that of Lauritzen and Spiegelhalter (1988), Jensen et al (1990), and Dawid (1992). Methods for exact inference in Bayesian networks that encode multivariate-Gaussian or Gaussianmixture distributions have been developed by Shachter and Kenley (1989) and Lauritzen (1992), respectively.

Approximate methods for inference in Bayesian networks with other distributions, such as the generalized linear-regression model, have also been developed (Saul et al., 1996; Jaakkola and Jordan, 1996). For those applications where generic inference methods are impractical, researchers are developing techniques that are custom tailored to particular network topologies (Heckerman 1989; Suermondt and Cooper, 1991; Saul et al., 1996; Jaakkola and Jordan, 1996) or to particular inference queries (Ramamurthi and Agogino, 1988; Shachter et al., 1990; Jensen and Andersen, 1990; Darwiche and Provan, 1996).

Gradient Ascent for Bayes Nets

If w_{ijk} denote one entry in the conditional probability table for variable Y_i in the network, then:

$$w_{ijk} = \mathrm{P}(\mathrm{Y_i} = \mathrm{y_{ij}} \mid \mathrm{Parents}(\mathrm{Y_i}) = \mathrm{thelist\ u_{ik}\ of\ values})$$

Perform gradient ascent by repeatedly performing:

- update all w_{ijk} using training data D

$$w_{ijk} \leftarrow w_{ijk} + \eta \underset{d \in D}{} \frac{P_h(\mathrm{y_{ij}}, \mathrm{u_{ik}} \mid \mathrm{d})}{w_{ijk}}$$

- renormalize the w_{ijk} to assure that:
    - $\sum_{j} w_{ijk} = 1$
    - $0 < w_{ijk} \leq 1$

Learning in Bayes Nets

There are several variants for learning in BBN. The network structure might be known or unknown. In addition to this, the training examples might provide values of all network variables, or just a part. If the structure is known and the variables are partially observable, the learning procedure in BBN is similar to training neural network with hidden units. By using gradient ascent, the network can learn conditional probability tables. The mechanism is converging to the network *h* that locally maximizes $P(D \mid h)$.

Complexity

- The computational complexity is exponential in the size of the loop cut set, as we must generate and propagate a BBN for each combination of states of the loop cut set.
- The identification of the minimal loop cut set of a BBN is NP-hard, but heuristic methods exist to make it feasible.
- The computational complexity is a common problem to all methods moving from polytrees to multiply connected graphs.

Advantages

- Capable of discovering causal relationships
- Has probabilistic semantics for fitting the stochastic nature of both the biological processes & noisy experimentation

Disadvantages

- Can't deal with the continuous data
- In order to deal with temporal expression data, several changes have to be done

For the current project experiments, the class for managing the BBN is represented in Listing 1. The reasoning mechanism is built on the base of the already made SMILE BBN library.

```
//--------------------------
class model
{
DSL_network net;
x_line l[500];
int nl;
int NP;
public:
bool read(char*);
void set_Param(int);
void test(void);
int testOne(int);
};
//--------------------------
```

Listing 1. BBN C++ class

Each data sample in the model is stored in the structure presented in Listing 2.

```
//--------------------------
struct x_line
{
int param[20];
char exp[50];
};
//--------------------------
```

Listing 2. Structure for storing a data sample

The C++ source for the BNN routines, according to conducted experiments, is presented in Appendix.

Principal Component Analysis

The need for the PCA technique came from the fact that it was necessary to have a classification mechanism for handling special areas on the surface of the face. There were three areas of a high importance for the analysis. The first is the area between the eyebrows. For instance, the presence of wrinkles in that area can be associated to the tension in facial muscles 'Corrugator supercilii'/'Depressor supercilii' and so presence of Action Unit 4, for the 'lowered brow' state. The second area is the nasolabial area. There are certain facial muscles whose changes can produce the activation of certain Action Units in the nasolabial area. The Action Unit 6 can be triggered by tension in facial muscle 'Orbicularis oculi, pars orbitalis'. In the same way, tension in facial muscle 'Levator labii superioris alaquae nasi' can activate Action Unit 9 and strength of facial muscle 'Levator labii superioris' can lead to the activation of Action Unit 10.

The last visual area analyzed through the image processing routines is that of the chin.

The tension in the facial muscle 'Mentalis' is associated to the presence of Action Unit 17, 'raised chin' state.

The PCA technique was used to process the relatively large images for the described facial areas. The size of each area was expressed in terms of relative value comparing to the distance between the pupils. That was used for making the process robust to the distance the person stands from the camera, and person-independent.

The analyze was done separately, for each facial area. Every time there was available a set of 485 n-size vectors where n equals the width of the facial area multiplied by the height. In the common case, the size of one sample vector is of the order of few thousand values, one value per pixel. The facial image space was highly redundant and there were large amounts of data to be processed for making the classification of the desired emotions. Principal Components Analysis (PCA) is a statistical procedure which rotates the data such that maximum variability is projected onto the axes. Essentially, a set of correlated variables are transformed into a set of uncorrelated variables which are ordered by reducing variability. The uncorrelated variables are linear combinations of the original variables, and the last of these variables can be removed with minimum loss of real data.

The main use of PCA was to reduce the dimensionality of the data set while retaining as much information as is possible. It computed a compact and optimal description of the facial data set.

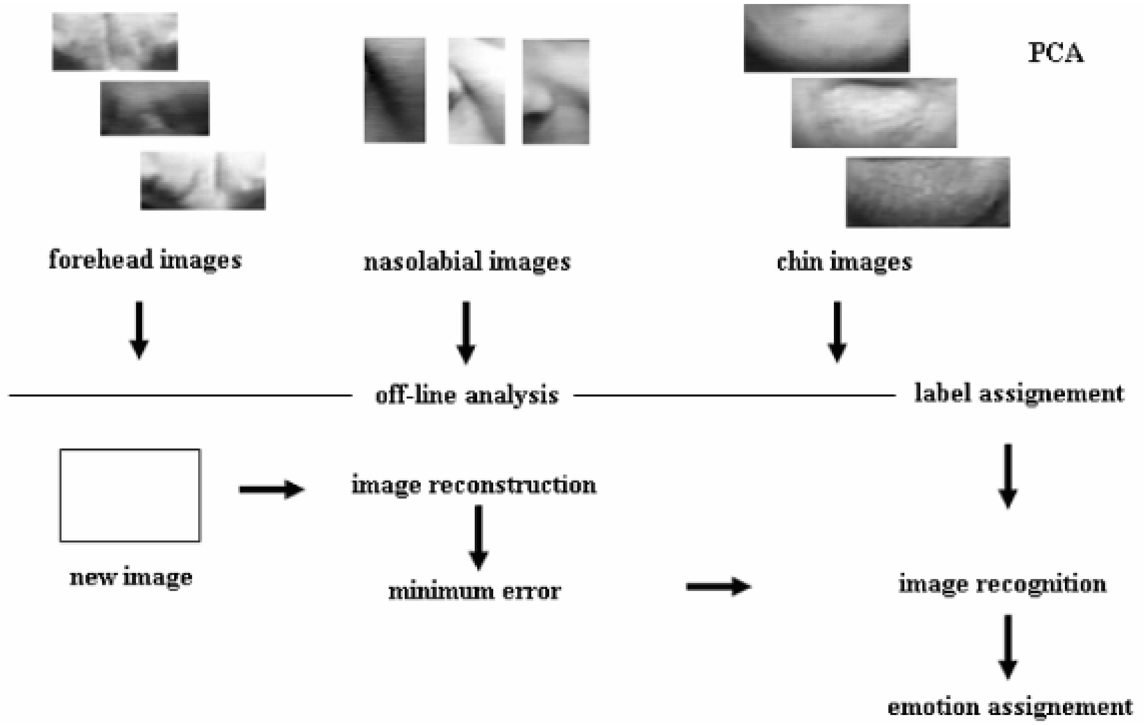


Figure 9. PCA image recognition and emotion assignment

The dimensionality reduction of data was done by analyzing the covariance matrix Σ. The reason, for which the facial data is redundant, is fact that each pixel in a face is highly correlated to the other pixels. The covariance matrix σ_{ij}, for an image set is highly non-diagonal:

Equation 8

$$\sigma_{ij} = X * X^T = \begin{matrix} \sigma^X{}_{11} & \sigma^X{}_{12} & \dots & \sigma^X{}_{1,w*h} \\ \sigma^X{}_{21} & \sigma^X{}_{22} & \dots & \sigma^X{}_{2,w*h} \\ \dots & \dots & \dots & \dots \\ \sigma^X{}_{w*h,1} & \sigma^X{}_{w*h,2} & \dots & \sigma^X{}_{w*h,w*h} \end{matrix}$$

The term σ_{ij} is the covariance between the pixel *i* and the pixel *j*. The relation between the covariance coefficient and the correlation coefficient is:

Equation 9

$$r_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \cdot \sigma_{jj}}}$$

The correlation coefficient is a normalized covariance coefficient. By making the covariance matrix of the new components to be a diagonal matrix, each component becomes uncorrelated to any other. This can be written as:

Equation 10

$$_{Y} = Y * Y^{T} = \begin{matrix} \sigma^{Y}{}_{11} & 0 & \ldots & 0 \\ 0 & \sigma^{Y}{}_{22} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \sigma^{Y}{}_{w*h,w*h} \end{matrix}$$

In the previous relation, **X** is the matrix containing the images of a given facial area and **Y** is the matrix containing the column image vectors.
The form of the diagonal covariance matrix assures the maximum variance for a variable with itself and minimum variance with the others.

The principal components are calculated linearly. If **P** be the transformation matrix, then:

Equation 11

$$Y = P^{T} * X$$
$$X = P * Y$$

The columns of *P* are orthonormal one to each other and:

Equation 12

$$P = P^{-1}$$
$$P^{T} * P = I$$

The constraint that $_Y$ should become a diagonal matrix, gives the mode the *P* matrix has to be computed.

Equation 13

$$
\begin{aligned}
{}_Y &= Y * Y^T = P^T * X * X^T * P \\
&= P^T * {}_X * P
\end{aligned}
$$

That means that $_Y$ is the rotation of $_X$ by *P*. If *P* is the matrix containing the eigenvectors of $_X$, then:

Equation 14

$$
{}_X * P = \Lambda * P
$$

where Λ is the diagonal matrix containing the eigenvalues of $_X$. Further, the relation can be written:

Equation 15

$$
\begin{aligned}
{}_Y &= P^T * \Lambda * P \\
&= \Lambda * P^T * P = \Lambda
\end{aligned}
$$

and $_Y$ is the diagonal matrix containing the eigenvalues of $_X$. Since the diagonal elements of $_Y$ are the variance of the components of the training facial area images in the face space, the eigenvalues of $_X$ are those variances. The maximum number of principal components is the number of variable in the original space. However, in order to reduce the dimension, some principal components should be omitted. Obviously, the dimensionality of the facial area image space is less than the dimensionality of the image space:

Equation 16

$$
\begin{aligned}
&\dim(Y) = column(P) \times column(X) = rank(X * X^T) \times K \\
&\dim(Y) = rank(X * X^T) \times 1
\end{aligned}
$$

The term $rank(X * X^T)$ is generally equal to K and a reduction of dimension has been made. A further reduction of dimension can be made.

At the testing session, an image representing a given type of facial expression is taken. A reconstruction procedure is done for determining which emotional class the image can be associated with. The mechanism is based actually on determining the facial area image that is closer to the new image in the set. The emotional class is that of the image for whom the error is minimum.

PCA mechanism has been also used as direct classifier for the facial emotions. The initial set of data consisted of 10 parameter values for each sample from the database. Each sample has been represented by a label having the description of the facial expression associated to the sample.

Because the data included the values of the parameters and not all the pixels in the image space, the PCA methodology was not used for reducing the dimensionality of the data. Instead the result reflected the rotation on the axes so as to have high efficiency in projecting the input vectors in the axes for a correct classification of facial expression.

The results of the experiment using PCA as direct classifier can be seen in Experiments section.

Artificial Neural Networks

Back-Propagation

The ANN represents learning mechanisms that are inspired from the real world. The structure of such a mathematical abstraction would consist in a set of neurons presenting a certain type of organization and the specific neuronal interconnections.

The Back Propagation approach for the ANN implies that the learning process takes place on the base of having learning samples for input and output patterns. In the case of learning such a system to model the mechanism of classifying facial expressions, it is required to have a set of input and output sample data. There would be two stages, one for making the system aware of the structure and associations of the data, so that is the training stage. The second step would be that of testing. In the training step, the system would build the internal knowledge, based on the presented patterns to be learned. The knowledge of the system resides in the weights associated to the connections between the neurons.

The training of the ANN is done by presenting the network with the configuration of the input parameters and the output index of facial expression. Both kinds of data are encoded as values in the network neurons. The input parameters are encoded on the neurons grouped in the input layer of the network. In the same way, the emotion index in encoded in the neuron(s) in the output layer.

For the experiments run on the current project, a back-propagation neural network was used as the classifier for the facial expressions. The topology of the network describes three neuron layers. The input layer is set to handle the input data according to the type of each experiment. The data refer to the parameters of the model used for analysis.

Encoding the parameters in the neurons

There are two type of encoding a value in the neurons of the ANN. Since the neurons present values in the ANN, each neuron can be used for storing almost any kind of numeric information. Usually, a neuron is set to handle values within a given interval, i.e. [-0.5, +0.5] or [0, 1]. Since there is a limitation in representing numeric values, any other value that is outside the interval can be mapped to a value in the interval (Figure 10).

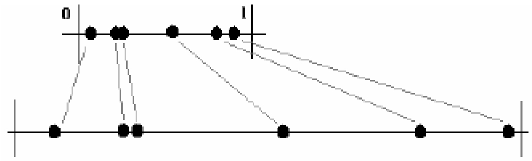
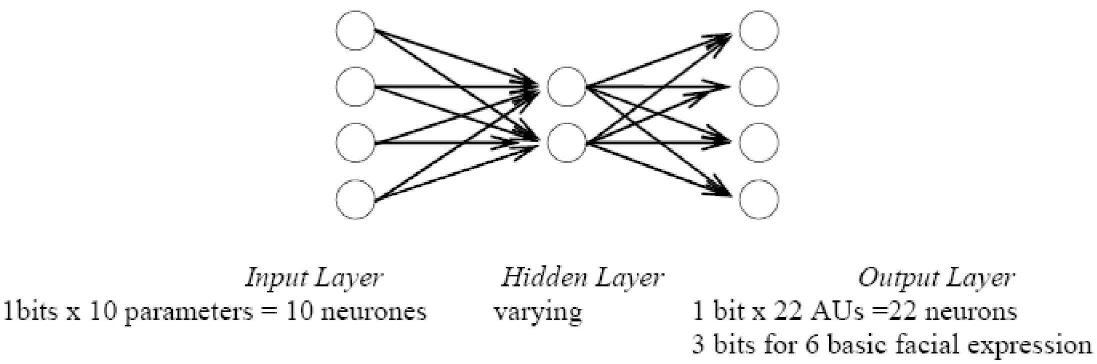


Figure 10. Mapping any value so as to be encoded by a single neuron

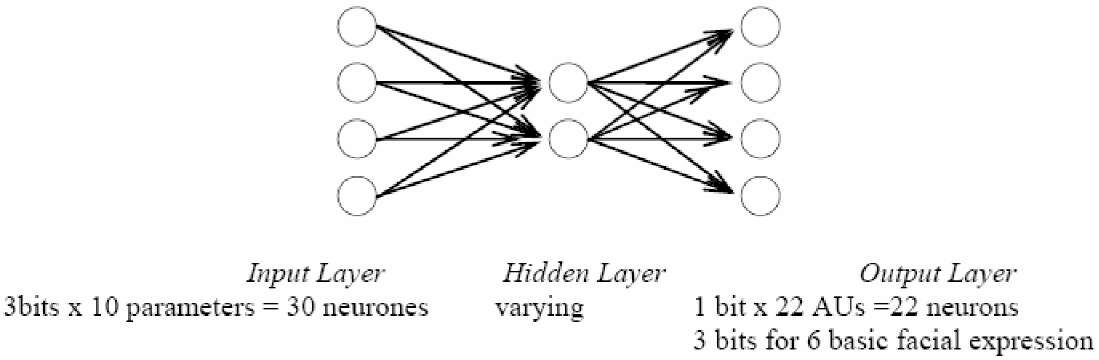
In this case of encoding, the previous process of discretization applied on the value of the model parameters is no longer necessary. Every neural unit can manage to keep any value of one parameter without any other intervention. By using a single neuron for encoding the value of a parameter, the structure of the ANN becomes simpler and the computational effort is less. The network's structure is as presented below:


Input Layer
1bits x 10 parameters = 10 neurones
Hidden Layer
varying
Output Layer
1 bit x 22 AUs =22 neurons
3 bits for 6 basic facial expression

In the case the network is used for recognizing the presence of Action Units, the output layer would require 22 neurons, one for each AU. Moreover, in the case the ANN would be assumed to recognize facial expressions, the output layer would consist only of few bits, enough for encoding an index to be associated to one of the six different basic facial expressions.

The second method of encoding parameter value in an ANN is to use discretization mechanism before and to encode any value by using a small group of neurons.

For a set of 10 parameters, each parameter is encoded as a group of three neurons that is able to encode a value big enough to represent the maximum number of the classes for each discrete parameter. The experiments conducted required a number of 5 or 7 distinct values per parameter. The second layer is that of the hidden neurons. For the experiments there has been used different numbers of the hidden neurons. The output layer manages the encoding of the six basic facial expression classes. It basically contains three output neurons. There has also been conducted some experiments for recognition of the 22 Action Units (AUs) and for each Action Unit there exists a distinct neuron for encoding the state of presence or absence. The general architecture to the ANNs used for experiments is as presented:


Input Layer
3bits x 10 parameters = 30 neurones
Hidden Layer
varying
Output Layer
1 bit x 22 AUs =22 neurons
3 bits for 6 basic facial expression

Knowledge Building in an ANN

When the system learns a new association in the input/output space, a measure is used to give the degree of the improvement or of the distance the ANN is from the point it recognize all the samples without any mistake.

The learning algorithm is based on a gradient descent in error space, the error being defined as:

$$E = \quad {}_{P} \quad E_p$$

The term E_p is the error for one input pattern, and

$$E_p = \frac{1}{2} \; {}_{i} \; (t_i - a_i)^2$$

The weights are adjusted according to the gradient of error

$$\Delta w = -\eta \nabla E$$

The term η is a constant scaling factor defining the step-size.

The weight change for the connection from unit *i* to unit *j*, of this error gradient can be defined as:

$$\Delta w_{ji} = -\eta \nabla_{ji} E = -\frac{\partial E}{\partial w_{ji}}$$

The gradient components can be expressed as follows:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}}$$

The third partial derivative in the previous equation can be easily computed based on the definition of ∂net_j

$$\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} {}_k \quad w_{jk} a_k = {}_k \frac{\partial w_{jk} a_k}{\partial w_{ji}}$$

Using the chain rule the previous relation can be written as:

$$\frac{\partial net_j}{\partial w_{ji}} = {}_k \left(\frac{\partial w_{jk}}{\partial w_{ji}} a_k + w_{jk} \frac{\partial a_k}{\partial w_{ji}}\right)$$

Examining the first partial derivative, it can be noticed that $\frac{\partial w_{jk}}{\partial w_{ji}}$ is zero unless $k = i$. Furthermore, examining the second partial derivative, if w_{jk} is not zero, then there exists a connection from unit *k* to unit *j* which implies that $\frac{\partial a_k}{\partial w_{ji}}$ must be zero because otherwise the network would not be feed-forward and there would be a recurrent connection.

Following the criteria,

$$\frac{\partial net_j}{\partial w_{ji}} = a_i$$

The middle partial derivative is

$$\frac{\partial a_j}{\partial net_j}$$

If $f(net_j)$ is the logistic activation function, then $f(net_j) = \frac{1}{1+e^{-net_j}}$ and:

$$\frac{\partial a_j}{\partial net_j} = f'(net_j) = \frac{d(1+e^{-x})^{-1}}{dnet_j}$$

By solving the previous equation, the result is:

$$\frac{d(1+e^{-x})^{-1}}{dnet_j} = (-1)(1+e^{-x})^{-2}e^{-x}(-1)$$
$$= \frac{e^{-x}}{1+e^{-x}}\frac{1}{1+e^{-x}}$$
$$= \frac{1+e^{-x}-1}{1+e^{-x}}\frac{1}{1+e^{-x}}$$
$$= (\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}})\frac{1}{1+e^{-x}}$$
$$= (1-a_j)a_j$$

That means that:

$$\frac{\partial a_j}{\partial net_j} = (1-a_j)a_j$$

The first derivative of the relation is $\frac{\partial E}{\partial a_j}$ and $E_p = \frac{1}{2}\ {}_{i}\ (t_i - a_i)^2$

The sum is over the output units of the network. There are two cases to be considered for the partial derivative:

- *j* is an output unit,
- *j* is not an output unit.

If *j* is an output unit, the derivative can be computed simply as:

$$\frac{\partial E}{\partial a_j} = \frac{\partial}{\partial a_j}\frac{1}{2}\ {}_{i}\ (t_i - a_i)^2$$
$$= \ {}_{i}\ (t_i - a_i)\frac{\partial(t_i - a_i)}{\partial a_j}$$
$$= (t_i - a_i)(-1)$$
$$= -1(t_i - a_i)$$

In the relation, for the case that a_j is not an output unit, the relation is:

$$\frac{\partial E}{\partial a_j} = \quad {}_i \quad \frac{E}{\partial a_{pk}} \frac{\partial a_{pk}}{\partial net_{pk}} w_{kj}$$

The second term is known and the first term is computed recursively.

Encoding ANN

- A minimizing a cost function is performs for mapping the input to output
- Cost function minimization
  - Weight connection adjustments according to the error between computed and desired output values
  - Usually it is the squared error
    - Squared difference between computed and desired output values across all patterns in the data set
  - other cost functions
    - Entropic cost function
      - White (1988) and Baum & Wilczek (1988)
    - Linear error
      - Alystyne (1988)
    - Minkowski-r back-propagation
      - r_{th} power of the absolute value of the error
      - Hanson & Burr (1988)
      - Alystyne (1988)
  - Weight adjustment procedure is derived by computing the change in the cost function with respect to the change in each weight
  - The derivation is extended so as to find the equation for adapting the connections between the F_A and F_B layers
    - each F_B error is a proportionally weighted sum of the errors produced at the F_C layer

- The basic *vanilla version back-propagation* algorithm minimizes the squared error cost function and uses the three-layer elementary back propagation topology. Also known as the generalized delta rule.

Advantages

- it is capable of storing many more patterns than the number of FA dimensions
- it is able to acquire complex nonlinear mappings

Limitations

- it requires extremely long training time
- offline encoding
- inability to know how to precisely generate any arbitrary mapping procedure

The C++ class that handles the operations related to the ANN is presented in Listing 3.

```
//-----------------------------
class nn
{
model&m;
int ni,nh,no;
float i[NI],h[NH],o[NO],eh[NH],eo[NO],w1[NI][NH],w2[NH][NO];
public:
nn(model&,int,int,int);
void train(void);
void test(void);
void save(char*);
void load(char*);
private:
void randomWeights(void);
float f(float);
float df(float);
void pass();
float trainSample(int);
};
//-----------------------------
```

Listing 3. C++ class to handle the ANN

The class presented offers the possibility to save a defined structure of ANN including the weights and to load an already developed one. The source of the ANN routines is presented in the appendix.

Spatial Filtering

The spatial filtering technique is used for enhancing or improving images by applying filter function or filter operators in the domain of image space (x,y) or spatial frequency (x,h). Spatial filtering methods were applied in the domain of image space and aimed at face image enhancement with so-called enhancement filters. While applied in the domain of spatial frequency they are aimed at reconstruction with reconstruction filters.

Filtering in the Domain of Image Space

In the case of digital image data, spatial filtering in the domain of image space was achieved by local convolution with an n x n matrix operator as follows.

$$y(i,j) = \sum_{k=i-w}^{i+w} \sum_{l=j-w}^{j+w} f(k,l)h(i-k,j-l)$$

In the previous relation, *f* is the input image, *h* is the filter function and *g* is the output image.

The convolution was created by a series of shift-multiply-sum operators with an *nXn* matrix (n: odd number). Because the image data were large, n was selected as 3. The visual processing library used for the project also included convolution routines that used larger matrixes.

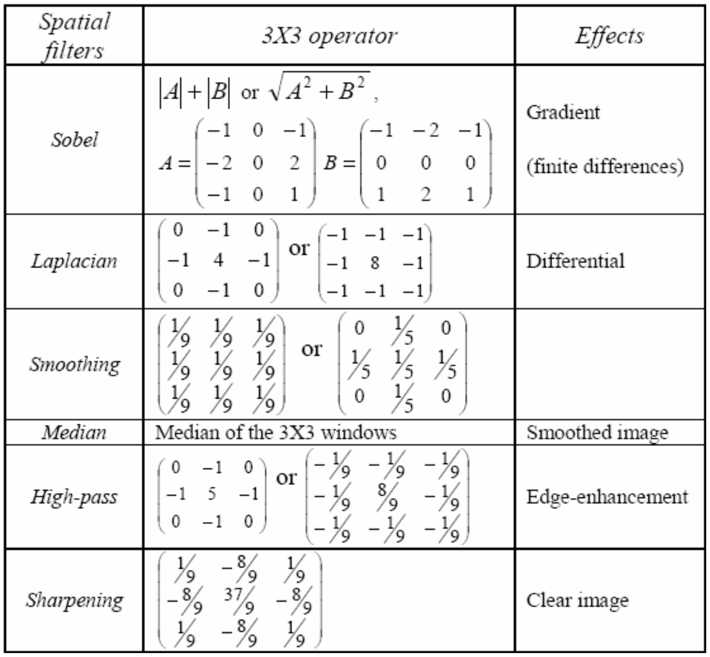
Filtering in the domain of Spatial Frequency

The filtering technique assumes the use of the Fourier transform for converting from image space domain to spatial frequency domain.

$$G(u,v) = F(u,v)H(u,v)$$

In the previous relation, *F* is Fourier transformation of input image and *H* is the filter function. The inverse Fourier transform applied on the filtering of spatial frequency can be used for recovering the initial image. The processing library used on the project included also support for filtering in the spatial frequency domain.

Table 5. 3x3 window enhancement filters



Spatial filters	*3X3 operator*	*Effects*
Sobel	$\lvert A\rvert + \lvert B\rvert$ or $\sqrt{A^2+B^2}$, $A = \begin{pmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$	Gradient (finite differences)
Laplacian	$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$ or $\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$	Differential
Smoothing	$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}$ or $\begin{pmatrix} 0 & 1/5 & 0 \\ 1/5 & 1/5 & 1/5 \\ 0 & 1/5 & 0 \end{pmatrix}$	
Median	Median of the 3X3 windows	Smoothed image
High-pass	$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$ or $\begin{pmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & 8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{pmatrix}$	Edge-enhancement
Sharpening	$\begin{pmatrix} 1/9 & -8/9 & 1/9 \\ -8/9 & 37/9 & -8/9 \\ 1/9 & -8/9 & 1/9 \end{pmatrix}$	Clear image

Low pass filters, **high pass filters**, **band pass filters** are filters with a criterion of frequency control. Low pass filters which output only lower frequency image data, less than a specified threshold, were applied to remove high frequency, noise, in some cases in the images of the initial database, before training. In addition to that, the specified techniques were used also at the testing session of the recognition system. In the same way high pass filter were used for removing stripe noise of low frequency. Some of the filtering routines included in the image processing library are presented in the Table 5. 3x3 window enhancement filters.

Eye tracking

The architecture of the facial expression recognition system integrates two major components. In the case of the real-time analysis applied on video streams, a first module is set to determine the position of the person eyes. The eye detector is based on the characteristic of the eye pupils in infra-red illumination. For the project experiments, an IR-adapted web cam was used as vision sensor (Figure 11).

Given the position of the eyes, the next step is to recover the position of the other visual features as the presence of some wrinkles, furrows and the position of the mouth and eyebrows. The information related to the position of the eyes is used to constrain the mathematical model for the point detection. The second module receives the coordinates of the visual features and uses them to apply recognition of facial expressions according to the given emotional classes.

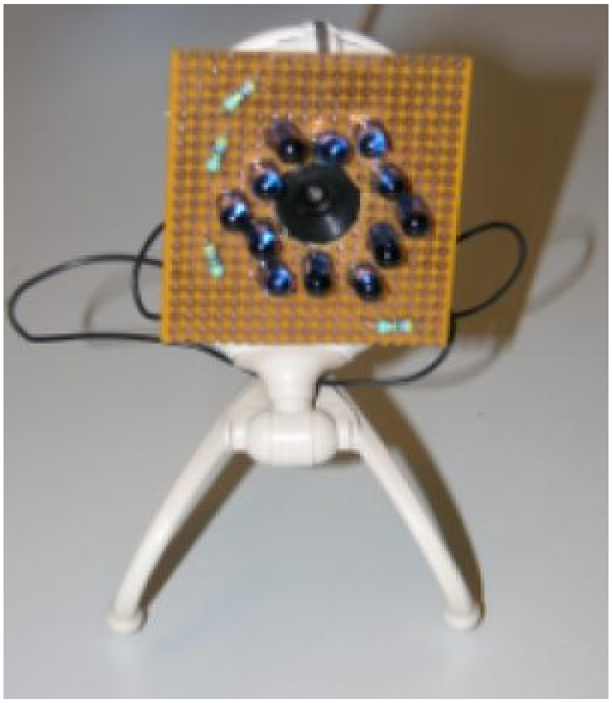


Figure 11. IR-adapted web cam

The enhanced detection of the eyes in the image sequence is accomplished by using a tracking mechanism based on Kalman filter [Almageed et. all 2002]. The eye-tracking module includes some routines for detecting the position of the edge between the pupil and the iris. The process is based on the characteristic of the dark-bright pupil effect in infrared condition (Figure 12).

Figure 12. The dark-bright pupil effect in infrared

However, the eye position locator may not perform well in some contexts as poor illuminated scene or the rotation of the head. The same might happen when the person wears glasses or has the eyes closed. The inconvenience is managed by computing the most probable eye position with Kalman filter. The estimation for the current frame takes into account the information related to the motion of the eyes in the previous frames. The Kalman filter relies on the decomposition of the pursuit eye motion into a deterministic component and a random component. The random component models the estimation error in the time sequence and further corrects the position of the eye. It has a random amplitude, occurrence and duration. The deterministic component concerns the motion parameters related to the position, velocity and acceleration of the eyes in the sequence. The acceleration of the motion is modeled as a Gauss-Markov process. The autocorrelation function is as follows:

Equation 17

$$R(t) = s^{2}e^{-b|t|}$$

The equations of the eye movement are defined according to the equation 18.

Equation 18

$$\begin{matrix}\dot{\mathrm{x}}_1\\ \dot{\mathrm{x}}_2\\ \dot{\mathrm{x}}_3\end{matrix} = \begin{matrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & -\beta\end{matrix} \begin{matrix}\mathrm{x}_1\\ \mathrm{x}_2\\ \mathrm{x}_3\end{matrix} + \begin{matrix}0\\ 0\\ 1\end{matrix} u(t)$$

$$z = \left[\sqrt{2\sigma^2\beta} \quad 0 \quad 0\right] \begin{matrix}\mathrm{x}_1\\ \mathrm{x}_2\\ \mathrm{x}_3\end{matrix}$$

In the model we use, the state vector contains an additional state variable according to the Gauss-Markov process. $u(t)$ is a unity Gaussian white noise. The discrete form of the model for tracking the eyes in the sequence is given in Equation 17. $\phi = \mathrm{e}^{\mathrm{f}\Delta\mathrm{t}}$, w are the process Gaussian white noise and ν is the measurement Gaussian white noise.

Equation 19

$$\begin{aligned} x_k &= \phi_k + w_k \\ z_k &= H_k \cdot x_k + v_k \end{aligned}$$

The Kalman filter method used for tracking the eyes presents a high efficiency by reducing the error of the coordinate estimation task. In addition to that, the process does not require a high processor load and a real time implementation was possible.

IMPLEMENTATION

Facial Feature Database

In the process of preparing the reasoning component to perform a reliable classification of facial expressions, data concerning visual features of the human face had to be available. All the relevant information had to be extracted from the image data and stored in a proper format. The reasoning methods used in the experiments consisted in statistical analysis, as Principal Component Analysis, neuronal networks and probabilistic techniques, as Bayesian Belief Networks. The PCA method was used for deciding what class of emotion can be assigned for some given image structure of certain facial areas, such as for the chin, the forehead and nasolabial areas. The other techniques were used for directly mapping an entrance of parameter values to certain groups of outputs, as Action Units or/and facial expressions. In all the cases, the values of some parameters, according to the chosen model for recognition, were manually computed from the Cohn-Kanade AU-Coded Facial Expression Database. Subjects in the available portion of the database were 100 university students enrolled in introductory psychology classes. They ranged in age from 18 to 30 years. Sixty-five percent were female, 15 percent were African-American, and three percent were Asian or Latino. The observation room was equipped with a chair for the subject and two Panasonic WV3230 cameras, each connected to a Panasonic S-VHS AG-7500 video recorder with a Horita synchronized time-code generator. One of the cameras was located directly in front of the subject, and the other was positioned 30 degrees to the right of the subject.

Only image data from the frontal camera were available at the time. Subjects were instructed by an experimenter to perform a series of 23 facial displays that included single action units (e.g., AU 12, or lip corners pulled obliquely) and combinations of action units (e.g., AU 1+2, or inner and outer brows raised). Subjects began and ended each display from a neutral face. Before performing each display, an experimenter described and modeled the desired display. Six of the displays were based on descriptions of prototypic emotions (i.e., joy, surprise, anger, fear, disgust, and sadness). For the available portion of the database, these six tasks and mouth opening in the absence of other action units were coded by a certified FACS coder. Seventeen percent of the data were comparison coded by a second certified FACS coder. Inter-observer agreement was quantified with coefficient kappa, which is the proportion of agreement above what would be expected to occur by chance (Cohen, 1960; Fleiss, 1981). The mean kappa for inter-observer agreement was 0.86.

Image sequences from neutral to target display were digitized into 640 by 480 or 490 pixel arrays with 8-bit precision for grayscale values. The image format is “png”. Images were labeled using their corresponding VITC.

FACS codes for the final frame in each image sequence were available for the analysis. In some cases the codes have been revised. The final frame of each image sequence was coded using FACS action units (AU), which are reliable descriptions of the subject's expression.

In order to make the task of computing the model parameter values possible, a software application was developed. It offered the possibility to manually plot certain points on each image of the database in an easy manner. The other components of the system automatically computed the values of the parameters so as to be ready for the training step for the neuronal networks or for computing the probabilities table in the case of BBN.

SMILE BBN library

SMILE [Structural Modeling, Inference, and Learning Engine] is a fully platform independent library of C++ classes implementing graphical probabilistic and decision theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. It was designed in a platform independent fashion as an object oriented robust platform. It has releases starting from 1997. The interface is so defined as to provide the developers with different tools for creating, editing, saving and loading of graphical models. The most important feature is related to the ability to use the already defined models for probabilistic reasoning and decision making under uncertainty. The release of

SMILE resides in a dynamic link library. It can be embedded in programs that use graphical probabilistic models as their engines for reasoning. Individual classes of SMILE are accessible from C++ or (as functions) from C programming languages. There also exists an ACTIVEX component as an alternative for embedding the library in the program that is supposed to have access to the SMILE routines. That makes possible for different programs that have been developed under distinct programming languages to still be able to query SMILE functionality.

Primary features

- It is platform independent. There are versions available for Unix/Solaris, Linux and PC
- The SMILE.NET module is available for use with .NET framework. It is compatible with all .NET languages including C# and VB.NET. It may be used for developing web-based applications of Bayesian networks
- It includes a very thorough and complete documentation

GeNIe BBN Toolkit

The GeNIe stands from Graphical Network Interface and is a software package that can be used to intuitively create decision theoretic models using a graphical click-and-drop interface. In addition to the capability to graphically design BBN models, it offers the possibility for testing and performing reasoning. The feature is realized by the integration of SMILE library. The latest version of GeNIe is GeNIe 2.0. It came as an improvement for the previous version, GeNIe 1.0 (1998) and includes new algorithms and techniques based on the various suggestions and requirements from the users.

The great advantage of using GeNIe is that the models can be quickly developed and tested by using an easy graphic interface. Once a model is ready, it can be integrated in a program as support for a backend engine by using SMILE functionality.

Primary Features

- Cross compatibility with other software through the support for other file types (Hugin, Netica, Ergo)
- Support for handling observation costs of nodes
- Support for diagnostic case management
- Supports chance nodes with General, Noisy OR/MAX and Noisy AND distribution

GeNIe Learning Wizard component

The module is part of GeNIe Toolkit version 1.0. It can be used for performing automatic discovery of causal models from data. It includes several learning algorithms, including constraint-based and Bayesian methods for learning structure and parameters. In addition to this it offers support for discrete, continuous and mixed data. The missing data are handled through a variety of special methods. There are several simple and advanced methods included for discretization of data. The user can specify many forms of background knowledge.

FCP Management Application

The Cohn-Kanade AU-Coded Facial Expression Database was used to create the initial knowledge for the system reasoning mechanism. The database consists of a series of image sequences done by 100 university students. The sequences contain facial expressions, according to the specifications given by the experimenter to the students. There are both single and combinations of Action Units. The most important part of the database stands for the set of facial images that were coded by means of Action Units. There are 485 distinct images and each has the correspondent AU sequence.

In order to be useful for the emotion recognition system, the AU coded images has to be analyzed for extracting the location of some given Facial Characteristic Points (FCPs).

A preprocessing step has been involved for preparing each image for being ready for processing. There has been applied some simple procedures to increase the quality of the images.

Each of the points has an exact position on the surface of the face. For making the process of the point extraction easier, a software application has been developed.

The application has a friendly interface and offers the possibility to manually set the full set of 36 FCPs in a graphical manner.

Initially, the image is loaded by choosing an option in the menu or by clicking on a given button in the application's toolbar (📂).

The image has to be stored in a BMP format with 24 bits per pixel. Other image formats are not allowed.

As soon as the image is loaded in the memory, it is shown on the surface of the application. From the current point, the user is assumed to specify the location of each of the 36 FCPs by clicking with the mouse on the certain image location. The first two points are used to specify the FCP of the inner corner of the eyes. The information is further taken for computing the degree of the rotation of the head. The angle is computed by using formula in equation 20.

Equation 20

$$\alpha = \arctan\left(\left|\frac{y_{FCP1} - y_{FCP2}}{x_{FCP1} - x_{FCP2}}\right|\right)$$

The value of the angle is then used for correcting the whole image by rotating it with the computed angle. Following the rotation procedure, the symmetric facial points are horizontally aligned. The point used as image rotation center is computed to be on the segment at the half distance between the eyes. The distance between the eyes represents the new parameter, *base* , whose value is computed using Equation 21.

The rotation point has the coordinates $O\left(\frac{x_{FCP1} + x_{FCP2}}{2}, \frac{y_{FCP1} + y_{FCP2}}{2}\right)$.

Equation 21

$$base = \sqrt{\left(x_{FCP1} - x_{FCP2}\right)^2 + \left(y_{FCP1} - y_{FCP2}\right)^2}$$

The parameter *base* is further used to adjust all the distance parameters between any FCPs for making the recognition process robust to the distance to the camera, and also person-independent.

Given the value of α , all the pixels are rotated on the screen by using Equation 22.

Equation 22

$$x_1 = x_0 + D \cdot \sin(\alpha)$$
$$y_1 = y_0 + D - D \cdot \cos(\alpha)$$

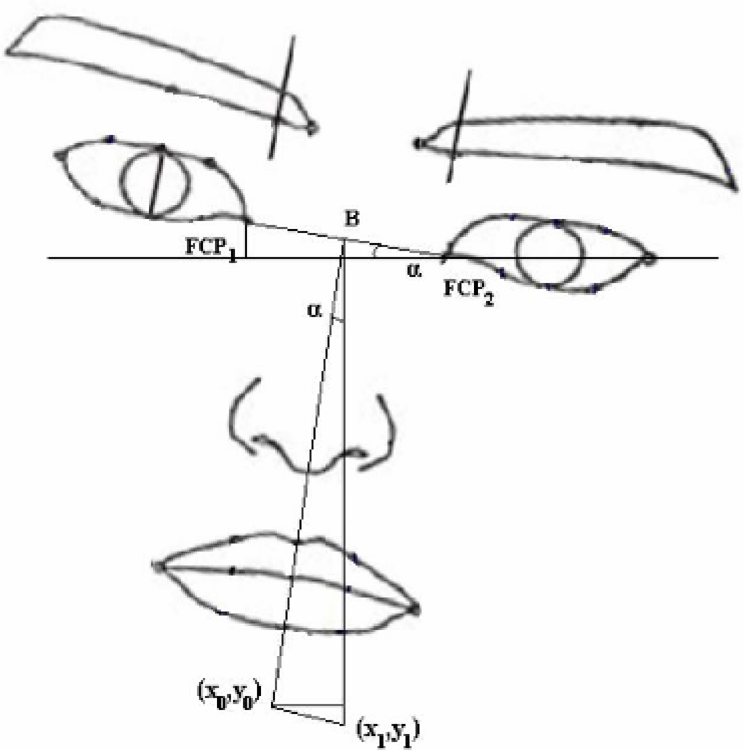


Figure 13. Head rotation in the image

In the relation above, the parameter *D* is the distance of the current point to the center of rotation *B*. The value of *D* is given as in Equation 23.

Equation 23

$$D = \sqrt{(x_0 - x_B)^2 + (y_0 - y_B)^2}$$

As soon as the loaded image is rotated, the user can set all the rest of the points. At the initial stage of working with FCP Management Application, the user is presented with the full loaded image. For setting the FCPs, the application can focus only on a certain area of the image. The user can switch between the two modes of working (🟩 🟪).
By using the application for setting the location of FCPs, the user has the option to zoom in or out (⊕ ⊖) for a better view of the interest area (Figure 14). It is also possible to switch from the modes of showing or not the FCP labels (a1).

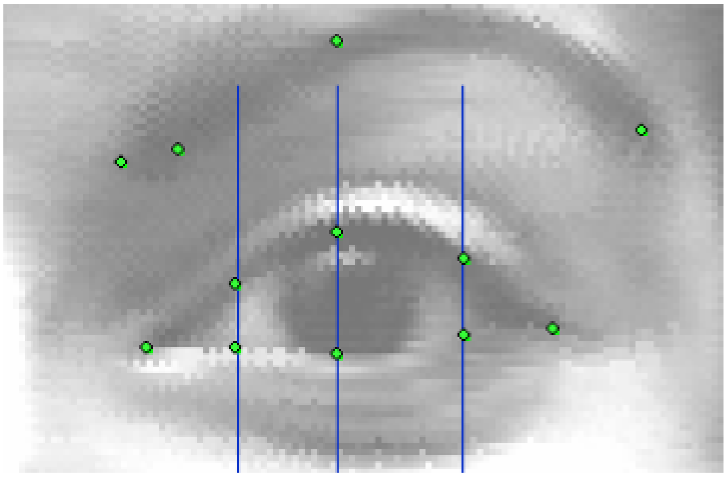
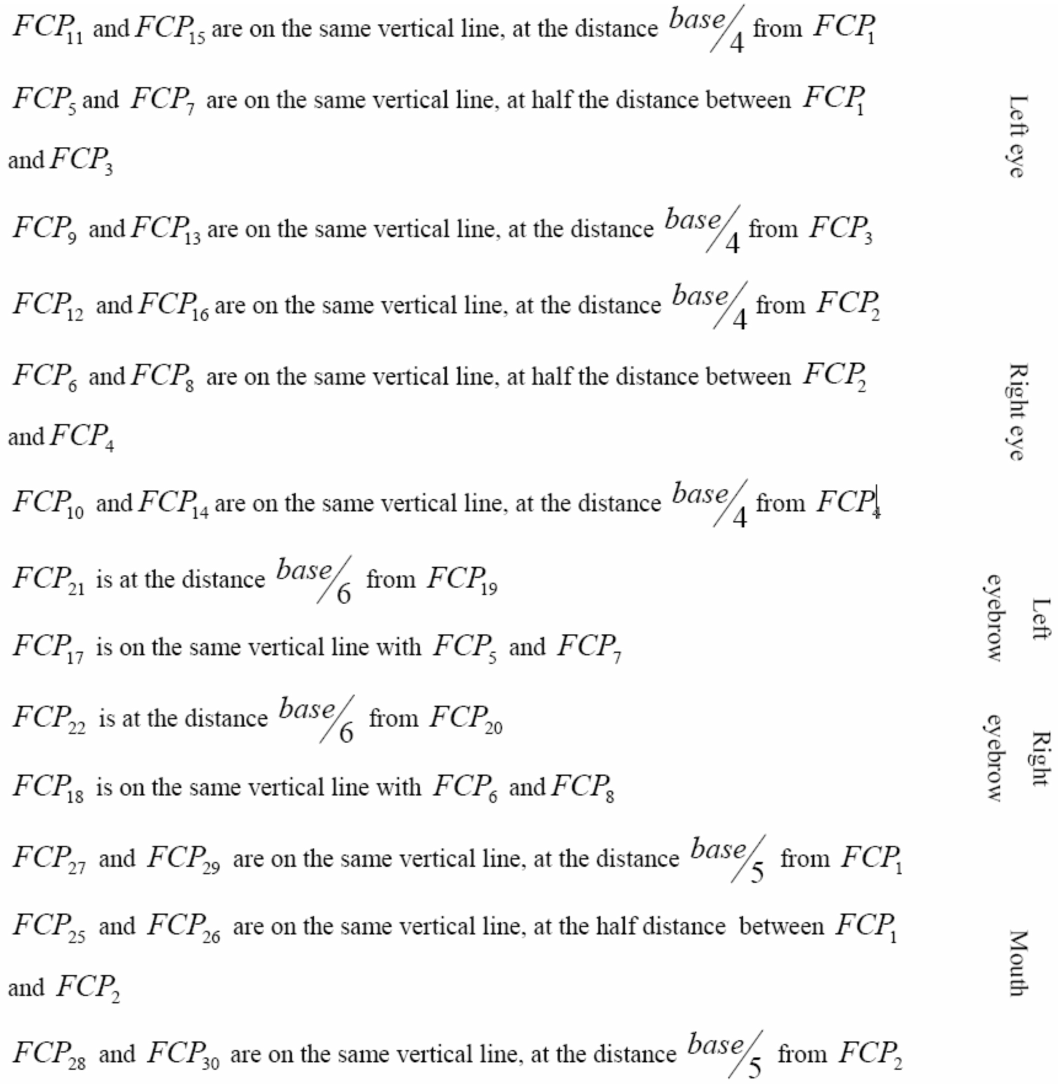


Figure 14. The ZOOM-IN function for FCP labeling

During the process of setting the Facial Characteristic Points on the image, the application does some checking so as to make sure the points are entered correctly. There are also some additional guiding lines drawn on the image (Figure 15). The checking rules followed are focused on the verification, whether the two or more points are on the same vertical line or at a given distance from other mark points, as described in table 6.

Table 6. The set of rules for the uniform FCP annotation scheme



FCP_{11} and FCP_{15} are on the same vertical line, at the distance $base/4$ from FCP_1	Left eye
FCP_5 and FCP_7 are on the same vertical line, at half the distance between FCP_1 and FCP_3	
FCP_9 and FCP_{13} are on the same vertical line, at the distance $base/4$ from FCP_3	
FCP_{12} and FCP_{16} are on the same vertical line, at the distance $base/4$ from FCP_2	Right eye
FCP_6 and FCP_8 are on the same vertical line, at half the distance between FCP_2 and FCP_4	
FCP_{10} and FCP_{14} are on the same vertical line, at the distance $base/4$ from FCP_4	
FCP_{21} is at the distance $base/6$ from FCP_{19}	Left eyebrow
FCP_{17} is on the same vertical line with FCP_5 and FCP_7	
FCP_{22} is at the distance $base/6$ from FCP_{20}	Right eyebrow
FCP_{18} is on the same vertical line with FCP_6 and FCP_8	
FCP_{27} and FCP_{29} are on the same vertical line, at the distance $base/5$ from FCP_1	Mouth
FCP_{25} and FCP_{26} are on the same vertical line, at the half distance between FCP_1 and FCP_2	
FCP_{28} and FCP_{30} are on the same vertical line, at the distance $base/5$ from FCP_2	

Once all the FCPs are set, the application can store the data in a text file on the disk.
The output file can have the same name with that of the original BMP file but with a distinct extension ("KH"), or a name specified by the user. For saving the FCP set, the user has to choose the proper option from the menu or to click on the button in the application toolbox (💾).

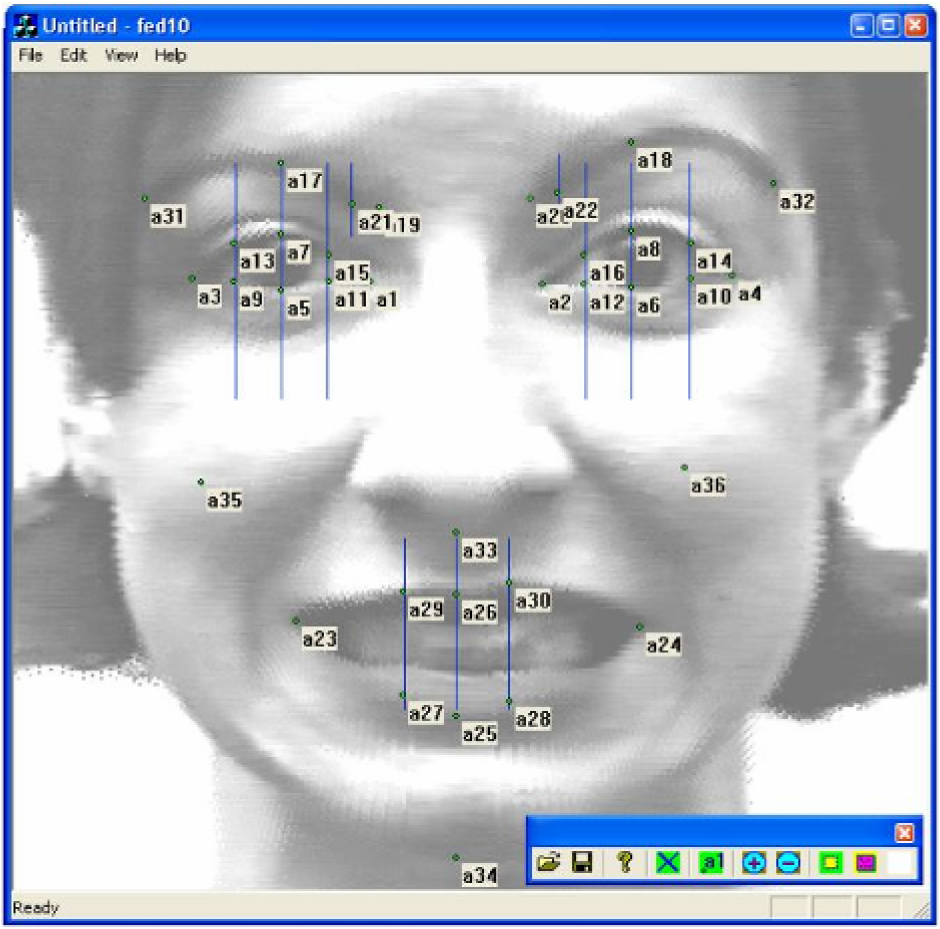


Figure 15. FCP Annotation Application

For the all set of images from the database, there has been obtained an equal number of FCP set files ("KH") (Figure 16).

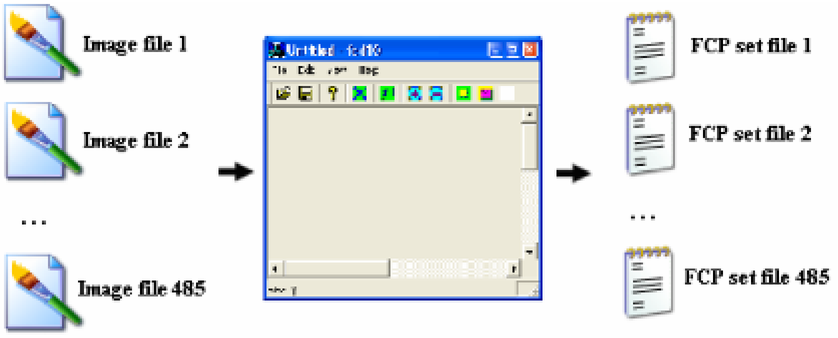


Figure 16. The preprocessing of the data samples that implies FCP anotation

The format of the output text file is given line by line, as:

----

a text string ("K&H.enhanced 36 points")

[image width] [image height]
x_{FCP1} y_{FCP1} x_{FCP2} y_{FCP2}
$p_i : x_{FCPi}, y_{FCPi}$

An example of an output file is given below

```
K&H.enhanced 36 points
640 490
348 227 407 226
p1:349,226
p2:407,227
p3:289,225
p4:471,224
p5:319,229
p6:437,228
p7:319,210
p8:437,209
p9:303,226
p10:457,225
p11:335,226
p12:421,227
p13:303,213
p14:457,213
p15:335,217
p16:421,217
p17:319,186
p18:437,179
p19:352,201
p20:403,198
p21:343,200
p22:412,196
p23:324,341
p24:440,343
p25:378,373
p26:378,332
p27:360,366
p28:396,368
p29:360,331
p30:396,328
p31:273,198
p32:485,193
p33:378,311
p34:378,421
p35:292,294
p36:455,289
```

Parameter Discretization

By using the FCP Management Application, all the images from the initial Cohn-Kanade AU-Coded Facial Expression Database were manually processed and a set of text files including the specification of Facial Characteristic Point locations has been obtained. The Parameter Discretization Application was further used for analyzing all the “KH” files previously created and to gather all the data in a single output text file.

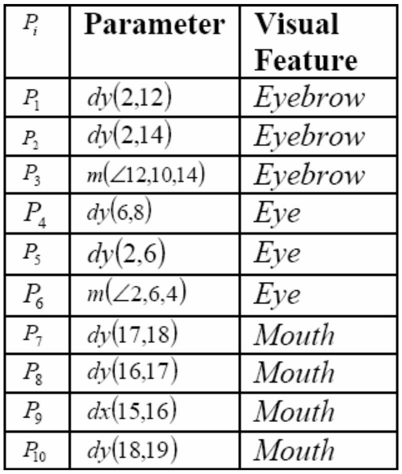
An important task of the application consisted in performing the discretization process for the value of each of the parameters, for all the input samples.

Once executed, the tool recurrently searched for files having the extension "KH" in a specified directory given as call parameter for the console application. For each file found, the application loads the content into the memory by storing the coordinates of the FCPs.

For each sample, it further applies a set of computations in order to determine the value of the parameters (Table 7), given the adopted processing model (Figure 17).

For the conducted experiment, no dynamic parameters were involved, since there were no data concerning the temporal variability available. However, the initial design included a general functional model that consisted also in a set of dynamic characteristics. For the results presented in the project report, there were also no values encoding the behavior of the parameters related to the forehead, chin and nasolabial areas. Instead, an additional set of experiments was run for analyzing the influence of those parameters.

Table 7. The set of parameters and the corresponding facial features



P_i	**Parameter**	**Visual Feature**
P_1	$dy(2,12)$	*Eyebrow*
P_2	$dy(2,14)$	*Eyebrow*
P_3	$m(\angle 12,10,14)$	*Eyebrow*
P_4	$dy(6,8)$	*Eye*
P_5	$dy(2,6)$	*Eye*
P_6	$m(\angle 2,6,4)$	*Eye*
P_7	$dy(17,18)$	*Mouth*
P_8	$dy(16,17)$	*Mouth*
P_9	$dx(15,16)$	*Mouth*
P_{10}	$dy(18,19)$	*Mouth*

The values of the parameters were initially considered real numbers. After the discretization process was finished, all the parameter values consisted in integer values.

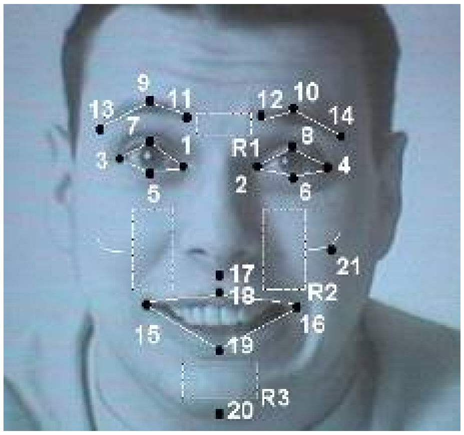


Figure 17. The facial areas involved in the feature extraction process

The discretization process started after all the FCP files were loaded into the memory. For each parameter, the system searched for the minimum and maximum value. Then the values were used to create an interval that includes all the sample values. Given the number of distinct classes to exist following the discretization process, the interval was split in a number of pieces equal to that of the classes (Figure 18).

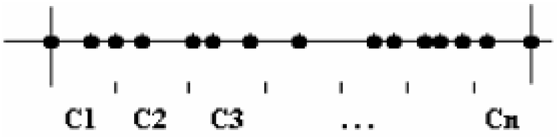


Figure 18. Discretization process

The result of the discretization process is presented in Listing 4.

```
//------------------------------------------------------------------------------------------
1: 486 10 7
2: 10 001 1+2+20+21+25 3 4 3 5 1 1 3 2 3 5
3: 10 002 1+2+5+25+27 4 5 5 7 4 3 3 5 6 2
4: 10 003 4+17 2 4 2 5 3 1 3 2 2 3
5: 10 004 4+7e+17d+23d+24d 2 3 2 3 1 1 3 2 2 2
6: 10 005 4+6+7+9e+16+25 1 3 2 3 1 1 2 3 4 2
7: 10 006 6+12+16c+25 2 3 3 5 2 1 7 1 1 5
8: --------------------------------------------------------------------------
9: 11 001 1+2+25+27 4 6 2 6 3 3 4 6 6 2
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
476: 137 002 12b 5 4 2 3 2 2 3 3 2 3
477: 137 003 25 6 4 1 4 3 3 3 3 3 3
478: 137 004 25+26 5 4 1 4 2 2 2 4 3 3
479: --------------------------------------------------------------------------
480: 138 001 1+4+5+10+20+25+38 4 3 2 4 3 1 3 3 3 5
481: 138 002 5+20+25 3 3 2 5 4 3 3 3 3 4
482: 138 003 25+27 3 3 2 4 2 2 3 4 4 2
```

```
483: 138 004 5+25+27 3 3 3 6 4 2 2 6 6 2
484: 138 005 6+7+12+25 3 3 2 2 1 1 2 1 2 6
485: 138 006 6+7+12Y 3 3 2 4 1 1 2 1 2 6
//------------------------------------------------------------------------------------------
```

Listing 4. The parameter discretization result

The more classes for the discretization process are, the higher the final emotion recognition rate was achieved. It was also found that after a certain value, the additional recognition percent obtained as result of increasing the number of the classes decreases.

The discretization process on parameters was mainly implied by the presence of Bayesian Belief Network reasoning. There was no need for discretization of parameters in the case only neural network computations are run. In that case, for instance, it would have determined a certain number of bits any value of the parameters could be represented and the values could directly be encoded on the neurons in the input layer. Another option would have been to work directly with values of the neurons in a given interval. Any value taken by the parameters could be scaled to the correspondent in the interval and encoded in the proper input neuron.

In the case of Bayesian Belief Networks, the number of classes determines the number of states of each parameter.

In the testing stage of the system, a new set of Facial Characteristic Points is provided by the FCP detection module. Based on the values, a set of parameter values is obtained following the computations. In case there is any value exceeding the limits of the interval, for one of the parameters, the number according to the nearest class is used instead.

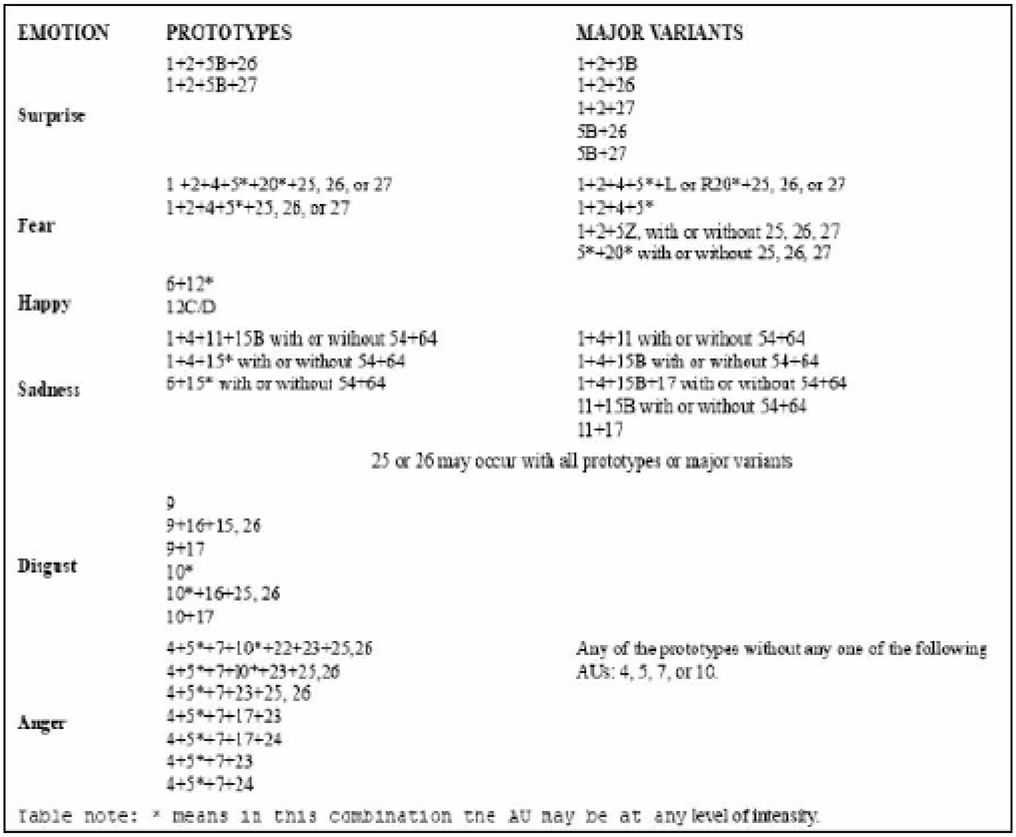
The BBN based reasoning is done by setting the class values of the parameters as evidence in the network and by computing the anterior probabilities of the parameters. Finally, the probabilities according to each emotional class are read from the proper parameter.

Facial Expression Assignment Application

Once the FCPs were specified in the set of 485 images and the 10 parameter values were computed, a software application processed each sequence of FACS for assigning the correct emotion to each face.

The functionality of the tool was based on a set of translation rules for re-labeling the AU coding into emotion prototypes as defined in the Investigator's Guide to the FACS manual, FACS 2002, Ekman, Friesen & Hager (Table 8). A C++ program was written to process a text input file and to translate each FACS sequence of each sample into the correspondent facial emotion.

Table 8. Emotion predictions



EMOTION	PROTOTYPES	MAJOR VARIANTS
Surprise	1+2+5B+26 1+2+5B+27	1+2+5B 1+2+26 1+2+27 5B+26 5B+27
Fear	1 +2+4+5*+20*+25, 26, or 27 1+2+4+5*+25, 26, or 27	1+2+4+5*+L or R20*+25, 26, or 27 1+2+4+5* 1+2+5Z, with or without 25, 26, 27 5*+20* with or without 25, 26, 27
Happy	6+12* 12C/D	
Sadness	1+4+11+15B with or without 54+64 1+4+15* with or without 54+64 6+15* with or without 54+64	1+4+11 with or without 54+64 1+4+15B with or without 54+64 1+4+15B+17 with or without 54+64 11+15B with or without 54+64 11+17
	25 or 26 may occur with all prototypes or major variants	
Disgust	9 9+16+15, 26 9+17 10* 10*+16+25, 26 10+17	
Anger	4+5*+7+10*+22+23+25,26 4+5*+7+10*+23+25,26 4+5*+7+23+25, 26 4+5*+7+17+23 4+5*+7+17+24 4+5*+7+23 4+5*+7+24	Any of the prototypes without any one of the following AUs: 4, 5, 7, or 10.

Table note: * means in this combination the AU may be at any level of intensity.

As output, the application provided also the field associated with the facial expression in addition to the FACS field for each image sample.

The information in the Table 8 was loaded from a text file saved on the disk. The content of the file is according to Listing 5.

```
//---------------------------------------------------------------
Surprise
:prototypes
1+2+5b+26
1+2+5b+27
:variants
1+2+5b
1+2+26
```

```
1+2+27
5b+26
5b+27
Fear
:prototypes
1+2+4+5*+20*+25 // 1+2+4+5*+20*+25,26,27
1+2+4+5*+20*+26
1+2+4+5*+20*+27
1+2+4+5*+25 // 1+2+4+5*+25,26,27
1+2+4+5*+26
1+2+4+5*+27
:variants
1+2+4+5*+l20*+25 // 1+2+4+5*+L or R20*+25,26,27
1+2+4+5*+l20*+26
1+2+4+5*+l20*+27
1+2+4+5*+r20*+25
1+2+4+5*+r20*+26
1+2+4+5*+r20*+27
1+2+4+5*
1+2+5z+25 // 1+2+5z+{25,26,27}
1+2+5z+26
1+2+5z+27
5*+20*+25 // 5*+20*+{25,26,27}
5*+20*+26
5*+20*+27
Happy
:prototypes
6+12*
12c // 12c/d
12d
Sadness
:prototypes
1+4+11+15b // 1+4+11+15b+{54+64}
1+4+11+15b+54+64
1+4+15* // 1+4+15*+{54+64}
1+4+15*+54+64
6+15* // 6+15*+{54+64}
6+15*+54+64
:variants
1+4+11 // 1+4+11+{54+64}
1+4+11+54+64
1+4+15b // 1+4+15b+{54+64}
1+4+15b+54+64
1+4+15b+17 // 1+4+15b+17+{54+64}
1+4+15b+17+54+64
11+15b // 11+15b+{54+64}
11+15b+54+64
11+17
Disgust
:prototypes
9
9+16+15 // 9+16+15,26
9+16+26
9+17
10*
10*+16+25 // 10*+16+25,26
10*+16+26
10+17
Anger
:prototypes
4+5*+7+10*+22+23+25 // 4+5*+7+10*+22+23+25,26
4+5*+7+10*+22+23+26
4+5*+7+10*+23+25 // 4+5*+7+10*+23+25,26
4+5*+7+10*+23+26
4+5*+7+23+25 // 4+5*+7+23+25,26
4+5*+7+23+26
4+5*+7+17+23
4+5*+7+17+24
4+5*+7+23
4+5*+7+24
:variants
//-----------------------------------------------------------------
```

Listing 5. The emotion translation rules grouped in the input file

A small part of the output text file is as it is presented in
Listing 6. The first item on each row stands for the identification number of the subject. The second item identifies the number of the recording sequence. The third item of every row represents the emotional expression and the fourth item is the initial AU combination.

```
//---------------------------------------------------------------
1: 10 001 Fear 1+2+20+21+25
2: 10 002 Surprise 1+2+5+25+27
3: 10 003 Sadness 4+17
4: 10 004 Anger 4+7e+17d+23d+24d
5: 10 005 Disgust 4+6+7+9e+16+25
6: 10 006 Happy 6+12+16c+25
7: ----------------
8: 11 001 Surprise 1+2+25+27
........................
483: 138 004 Surprise 5+25+27
484: 138 005 Happy 6+7+12+25
485: 138 006 Happy 6+7+12Y
//---------------------------------------------------------------
```

Listing 6. The output text file containing also the facial expressions

A second application put the data related to each sample together and outputted the result by using a convenient format. The result is presented in Listing 8.

```
//---------------------------------------------------------------
1: 485 10 7
2: 10 001 1+2+20+21+25 Fear 3 4 3 5 1 1 3 2 3 5
3: 10 002 1+2+5+25+27 Surprise 4 5 5 7 4 3 3 5 6 2
4: 10 003 4+17 Sadness 2 4 2 5 3 1 3 2 2 3
5: 10 004 4+7e+17d+23d+24d Anger 2 3 2 3 1 1 3 2 2 2
6: 10 005 4+6+7+9e+16+25 Disgust 1 3 2 3 1 1 2 3 4 2
7: 10 006 6+12+16c+25 Happy 2 3 3 5 2 1 7 1 1 5
8: -------------------------------------------------------------------------
9: 11 001 1+2+25+27 Surprise 4 6 2 6 3 3 4 6 6 2
........................
477: 137 003 25 Fear 6 4 1 4 3 3 3 3 3 3
478: 137 004 25+26 Surprise 5 4 1 4 2 2 2 4 3 3
479: -------------------------------------------------------------------------
480: 138 001 1+4+5+10+20+25+38 Fear 4 3 2 4 3 1 3 3 3 5
481: 138 002 5+20+25 Fear 3 3 2 5 4 3 3 3 3 4
482: 138 003 25+27 Surprise 3 3 2 4 2 2 3 4 4 2
483: 138 004 5+25+27 Surprise 3 3 3 6 4 2 2 6 6 2
484: 138 005 6+7+12+25 Happy 3 3 2 2 1 1 2 1 2 6
485: 138 006 6+7+12Y Happy 3 3 2 4 1 1 2 1 2 6
//---------------------------------------------------------------
```

Listing 7. Final data extracted from the initial database (version I)

On the first line there are put details on the process that has as result the current file. The first item represents the number of samples included in the analysis. The second item stands for the number of parameters and the third is the number of classes per parameter used for the discretization process. Another format that was more efficient for the next processing steps of the project is that presented in Listing 7.

```
//---------------------------------------------------------------
1: P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 Exp AU1 AU2 AU4 AU5 AU6 AU7 AU9 AU10 AU11 AU12 AU15
AU16 AU17 AU18 AU20 AU21 AU22 AU23 AU24 AU25 AU26 AU27
2: 3 4 3 5 1 1 3 2 3 5 Fear 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
3: 4 5 5 7 4 3 3 5 6 2 Surprise 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
4: 2 4 2 5 3 1 3 2 2 3 Sadness 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
5: 2 3 2 3 1 1 3 2 2 2 Anger 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0
..............................................................
483: 3 3 3 6 4 2 2 6 6 2 Surprise 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
484: 3 3 2 2 1 1 2 1 2 6 Happy 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
485: 3 3 2 4 1 1 2 1 2 6 Happy 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
//---------------------------------------------------------------
```

Listing 8. Final data extracted from the initial database (version II)

For every row, the sequence of Action Units was encoded by using the value "1" for denoting the presence of the AU and "0" for absence. The discretization process for the given example was done on 7 classes per parameter basis. For the conducted experiments, there were also generated training files with 5 and 8 classes per parameter discretization.

CPT Computation Application

The application is used for determining the values in the table of conditioned probabilities for each of the parameters included in the Bayesian Belief Network.

Initially, a BBN model can be defined by using a graphical-oriented application, as GeNIe. The result is a "XDSL" file that exists on the disk. The current application has been developed in C++ language. It parses an already made "XDSL" file containing a Bayes Belief network and analyses it. It also load the data related to the initial samples from the Cohn-Kanade database. For each of the parameters it runs some tasks as shown:

- determines the parent parameters of the current parameter
- analyzes all the states of the parent parameters and determines the all possible combination of the states
- for each state of the current parameter, passes through each of the possible combinations of the parents and:
  - create a query that includes the state of the current parameter
  - add the combination of parent parameters

    - call a specialized routine for computing the probability of existence of current parameter being in current state, given the parent parameters being in the states specified through the current combination
    - fills in the data in the Conditional Probability Table (CPT) of the current parameter
- saves the data on the disk, in a file having the same name as the input file. The difference is that the saved model has the data in parameter CPTs.

In order to be able to query the program for obtaining the probability associated to a certain state of the parameters, a C++ class tackles the data within the chosen parametric model (Listing 9).

```
//---------------------------------------------------
class model
{
public:
file a[1000];
int n;
int NP; //no.of parameters
int NC; //no.of classes
public:
model():n(0){}
bool readDatabase(char*);
void show(void);
float computeP(cere&);
float computeP2(cere2&);
float computeP3(int,char*);
float computeP4(int,char*,bool);
int countExpression(char*);
bool isIncluded(FACS&,FACS&);
void select(query&,model&);
void emptyDatabase(void){n=0;}
int countFields(void){return n;}
int getClass(int,int);
void save(char*);
char* getFieldExpression(int);
private:
float compare(FACS&,FACS&);
void insertField(int,int,FACS&,char*);
};
//---------------------------------------------------
```

Listing 9. C++ class to handle the model data

The class is used to manage all the processing on the data and to store the data related to the sample data from the initial sample database. An instance of the structure “file” stores internally the Action Units sequence, the description of the facial emotion and the value of the parameters. The definition is given in Listing 10.

```
//------------------------
struct file
{
int nsubj;
int nscene;
FACS facs;
char expr[100];
float param[50];
};
//------------------------
```

Listing 10. Data structure to store the information related to one sample

Each Action Unit sequence is handled separately by using a structure that is defined in Listing 11. It also contains some routines for easily converting from a character structure, to efficiently copy data from another similar structure and to check for the existence of some condition.

```
//------------------------
struct FACS
{
int n;
unsigned char a[30][3];
void assign(char*);
void copyFrom(FACS&);
void show(void);
bool Is(int,int);
bool containAU(int);
};
//------------------------
```

Listing 11. The internal structure to handle the AU sequence of one sample

A query is encoded in a structure that is shown in
Listing 12. It can be used for computing the probability of any expression, given the state of existence/absence of the Aus from the set.

```
//------------------------
#define MAXQUERYLEN 50
struct query
{
char expr[100];
bool swexpr;
bool neg_expr;
bool neg[MAXQUERYLEN];
FACS f[MAXQUERYLEN];
int n;

query():n(0),swexpr(false),neg_expr(false){};
bool add(char*e,bool sw_n=true);
void empty(void){n=0;neg_expr=false;swexpr=false;}
};
//------------------------
```

Listing 12. The structure of a query

There is another kind of structure (Listing 13) used for computing the probability of a parameter being in a certain state, given the state of presence/absence of the Action Unit parameters.

```
//-------------------------
struct cere2
{
int AU[10][2];
int n;
int P[2];
cere2():n(0){}
void empty(void){n=0;}
void setP(int k,int clasa){P[0]=k;P[1]=clasa;}
void add(int au,int activ){AU[n][0]=au;AU[n][1]=activ;n++;}
};
//-------------------------
```

Listing 13. A query for computing the CPTs for the first level

The sources of all the routines can be consulted in the appendix section.

Facial expression recognition application

As a practical application for the current project, a system for facial expression recognition has been developed. Basically, it handles a stream of images gathered from a video camera and further applies recognition on very frame. Finally, it provides as result a graph with the variances of the six basic emotions of the individual in a time domain. The figure below offers an overview on the functionality of the designed system.

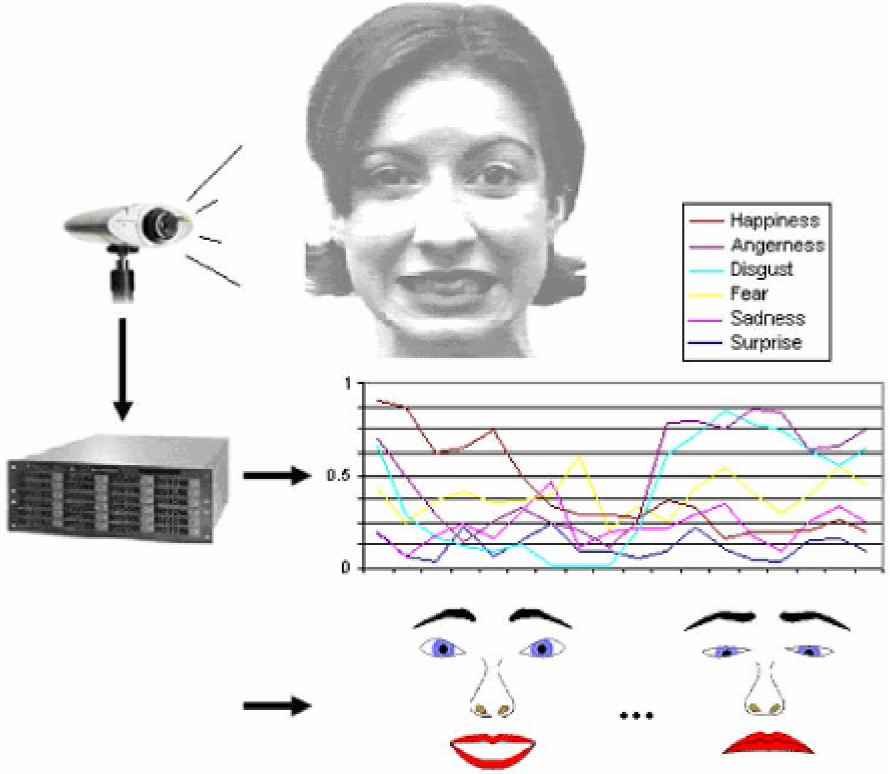


Figure 19. System functionality

The initial design idea specified the working of the system as for regular conditions. In order to have the system working in a common environment on video capturing, it was required to have a reliable module for feature detection from the input video signal. Because such a module was not available (a feature extraction module existed, but had poor detection rate) at the moment of developing the system, a minor change was made. A new module for extracting the visual features from the images was built. It relied on the infra red effect on the pupils as for primarily detecting the location of the eyes. As soon as the data related to the eye location was available, the system could use it as constraints in the process of detecting the other visual features. The feature detection module represents the first processing component applied on the input video signal. The data provided by that module are used for determining the probabilities of the each emotional class to be associated with the individual's current face.

The architecture of the system includes two software applications. One application is assumed to manage the task of data capturing from the video camera. It acts like a client-side application in a network environment. The second application has the goal of performing classification of facial expressions, based on the images received from the first application. The reasoning application was designed as a server-side application. The connectivity among the present components is as it is presented in the figure.

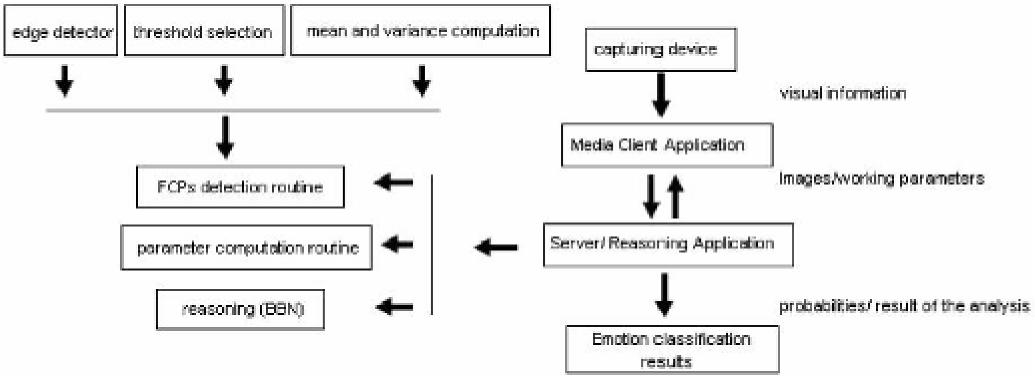

edge detector
threshold selection
mean and variance computation
capturing device
visual information
Media Client Application
Images/working parameters
FCPs detection routine
parameter computation routine
Server/ Reasoning Application
reasoning (BBN)
probabilities/ result of the analysis
Emotion classification results

Figure 20. The design of the system

The two applications send data to each-other through a TCP/IP connection. The media client acts like a bridge between the capturing device and the emotion classification application. It only sends captured images to the other side of the network. The server application can send back parameters concerning the rate of capturing, the frame size or parameters related to the image, as the contrast, brightness, etc.

The server application receives the images, one after another and put them on a processing queue. A thread is assumed to take the first image in the queue and to pass it to a sequence of processing modules. The first module is that of eye location recovery. Based on the detection result, the positions of all the other facial characteristic points are extracted. The next module is that of computing the values of some parameters according to the used model for recognition. The procedure implies analyzes of distances and angles among given facial characteristic points. The actual recognition module consists in a Bayesian Belief Network that already contains the proper values for all the associated probability tables for the nodes. The reasoning is done by setting as evidence the states for all the model parameters according to the discretization algorithm and by computing the anterior probabilities for the parent parameters. The parameter that encodes the Expression node contains six distinct states, one for each basic emotion class. By updating the probabilities, the system is able to provide the user with the emotional load for every expression.

Eye Detection Module

In order to detect the location of the eyes in the current frame, the specialized module passes the input image through a set of simple processing tasks. The initial image looks like that in Figure 21.

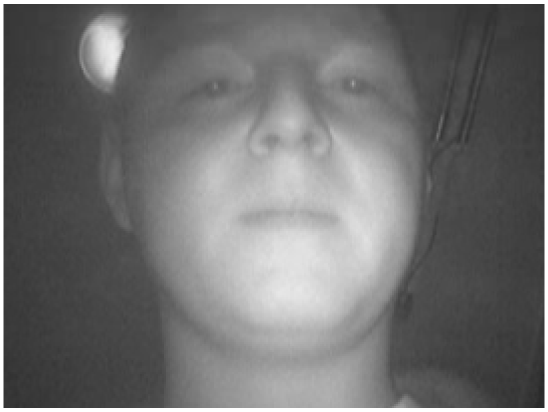


Figure 21. Initial IR image

The eye detector makes use of the property that the reflection of the infra red light on the eye is visible as a bright small round surrounded by a dark area. The routine is supposed to search for pixels that have that property. Finally the eye locations are chosen as the first two in a recovered candidate list.

First a Sobel-based edge detector is applied for detecting the contours that exists in the image. The reason of applying an edge detector is that the searched items contain a clear transition area in the pixels' color. The result is as in Figure 22.

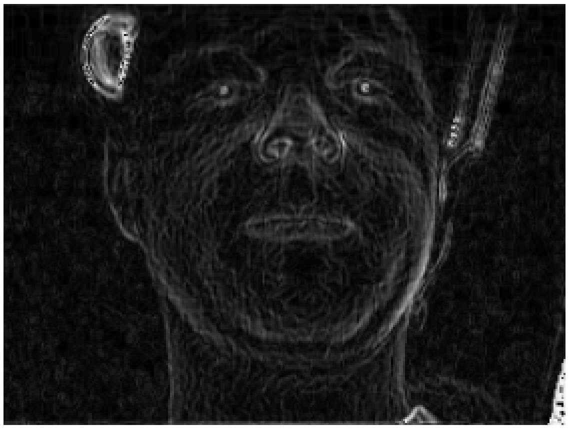


Figure 22. Sobel edge detector applied on the initial image

Because of the fact that the eye areas contain a sharp transition from the inner bright round to the surrounding black area, in the next stage only the pixels with high intensity are analyzed to be chosen as candidates for the eye positions. For removing the unwanted pixels, a threshold step is applied on the pixel map of the image. The threshold is so chosen as to let leave all the possible pixels for denoting the position of the eyes intact. It also has to be low enough for removing as many pixels as possible. The result is as in Figure 23.

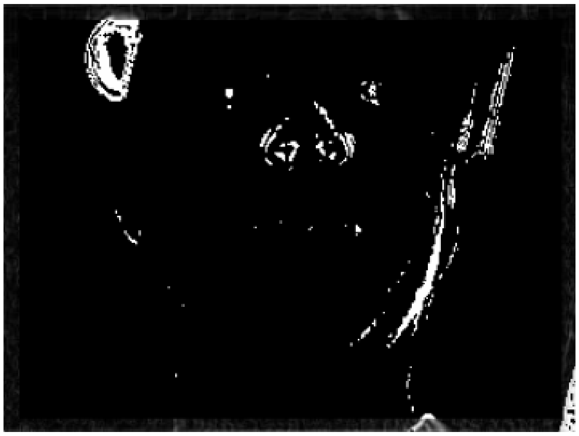


Figure 23. Threshold applied on the image

At the moment there are still a lot of pixels that are to be considered as candidates for the eye locations. The next step is to separately analyze each of those pixels through a given procedure that computes the mean and the variation of the pixels' intensity in a surrounding area. The searched area has to be far enough to the location of the pixel for not taking into account also the pixels with high probability to be part of the same are on the eye surface. The algorithm computes the values of mean and variance for the pixels presented as in Figure 24.

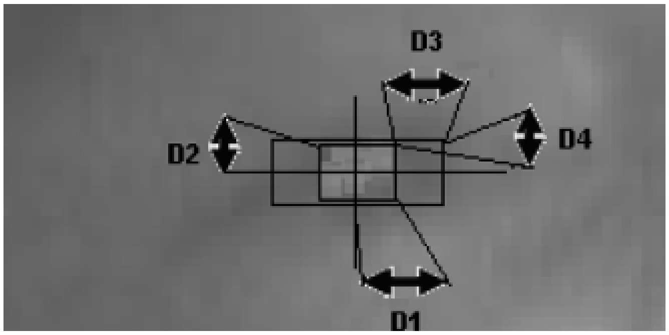


Figure 24. The eye-area searched

The values defining the area to be analyzed are parameters of the module. When the previous procedure is finished for all the candidate pixels, a new threshold procedure for the mean and variance is applied. The threshold for the variance has the goal to remove all the pixels whose surrounding area is not compact with respect to the intensity. The threshold for the mean is assumed to remove the candidate pixels whose intensity of all surrounding pixels is high. The procedure is not applied on the image resulting from the previous processing steps, but on the original image.

After the selection procedure only a few pixels remained that comply with the encoded specifications. One way of finding the position of the eyes is to take the pixels that have the highest intensity far enough from each-other from the remaining candidates queue.

The last two steps can be replaced with a simple back-propagation neural network for learning the function that selects only the proper pixels for eye location based on the value of mean and variance of the surrounding area (Figure 25).

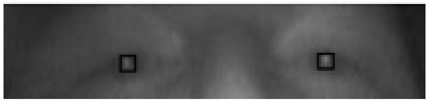


Figure 25. The eyes area found

The major limitation of the algorithm is that it is not robust enough. There are several cases when the results are not as expected. For instance when the eyes are closed or almost closed it obviously does not work. It creates a candidate list for the position of the eyes and further detects the most appropriate two pixels to the known eye area requirements. This can be avoided by setting a condition that finally detected pixels must have the intensity above a given value. The other way would be to create a structure for the alternative previous neural network to encode also the intensity of the candidate pixel on the input neural layer.

Another limitation is that some calibration has to be done before the actual working session. It consists in adjusting all the parameters related to the searched areas and pixel/area characteristics.

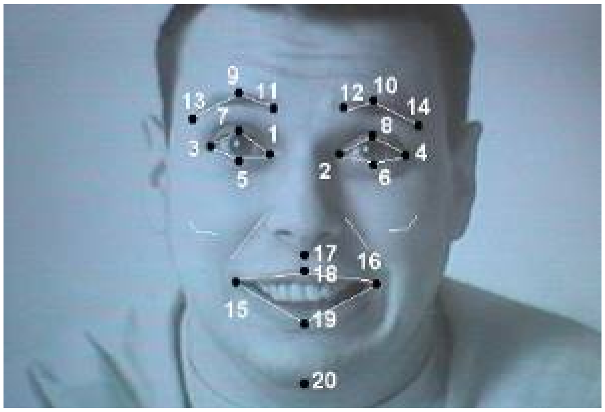


Figure 26. Model 's characteristic points

So far only the positions of the pupils are detected. For the detection of the characteristic points of each eye (Figure 26/ P1, P3, P5, P7, P2, P4, P6, P8), some processing has to be done. A new parameter is computed as the distance between the pupils. That is used for scaling all the distances so as to make the detection person independent.

The analysis starts again from the Sobel edge detector point. It further includes a threshold pixel removing. Given the position of the pupils for both eyes, the procedure is constrained to search for the pixels only in a given area around the pupil. Within the searched area, all pixels are analyzed through the procedure of computing the mean and variance as previously described. Based on the two values, some pixels are removed from the image.

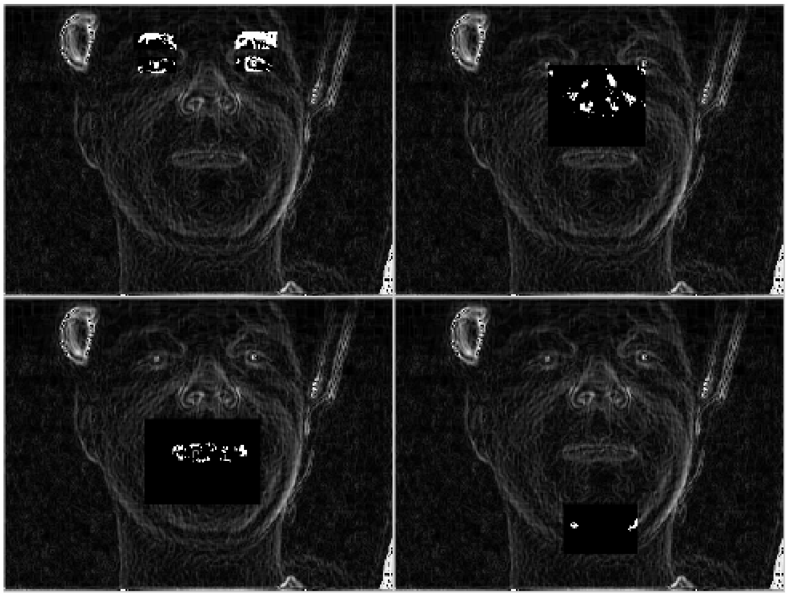


Figure 27. Characteristic points area

In the case of detecting the FCP of the left eye, the P1 is considered to be at the first high intensity pixel from the left side of the characteristic area. In the same way P3 is the right most pixel having a high intensity. The locations of the upper most and lower most points of the left eye are computed in the same manner. The same procedure is followed for the right eye.

In case of the eyebrows, P11 and P13 are the left/right most points in the area to be analyzed and P9 is the upper most one. For detecting the location of the nose point P17, the searching area is defined as starting from the coordinates of the pupils to just above the area of the mouth. The point is found to be the first point with high intensity in the search from the lowest line to the upper most in the area. The chin point P20 is found in the same way as P17. For detecting the points in the mouth area the same first pixel removing procedure is done. The characteristic points are considered to be as the left/right/up/down most bright points in the mouth area (Figure 27). The result of all the detection steps is represented in Figure 28.

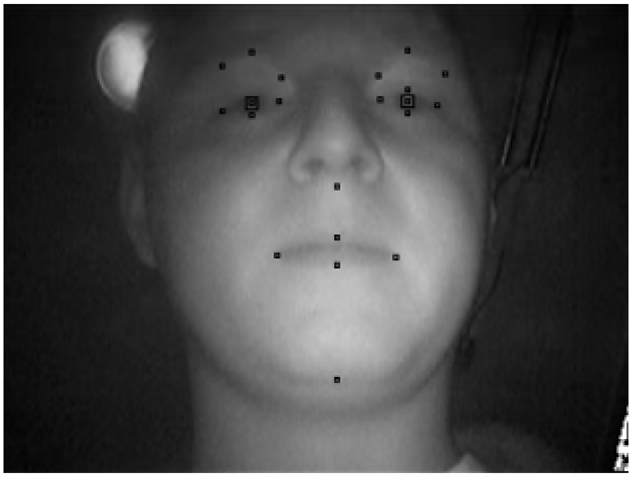


Figure 28. FCP detection

The efficiency in detection of FCPs is strictly related to the efficiency of pupil position recovering. In order to make the pupil detection routine more reliable, an enhancement has been done based on Kalman Filter. The mechanism is supposed to do a tracking of pupils in the time dimension by involving a permanent estimation of the parameters (as position, velocity, acceleration, etc.). The situations when the eyes are closed are now correctly processed by the visual feature detection module. The output of the system can be a simple graph showing the variation of the emotional load in time or on the form of a graphical response. The result of the processing may also include an emotional load, similar to that of the input signal, or different, according to a secondary reasoning mechanism.

Face representational model

The facial expression recognition system handles the input video stream and performs analysis on the existent frontal face. In addition to the set of degree values related to the detected expressions, the system can also output a graphical face model.

The result may be seen as a feedback of the system to the given facial expression of the person whose face is analyzed and it may be different of that. One direct application of the chosen architecture may be in a further design of systems that perceive and interact with humans by using natural communication channels. In the current approach the result is directly associated to the expression of the input face (Figure 29). Given the parameters from the expression recognition module, the system computes the shape of different visual features and generates a 2D graphical face model.

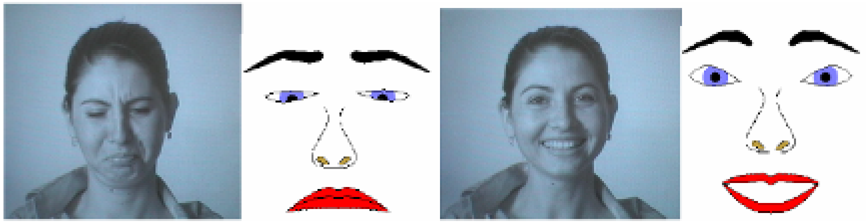


Figure 29. The response of the system

The geometrical shape of each visual feature follows certain rules that aim to set the outlook to convey the appropriate emotional meaning. Each feature is reconstructed using circles and simple polynomial functions as lines, parabola parts and cubic functions. A five-pixel window is used to smooth peaks so as to provide shapes with a more realistic appearance. The eye upper and lower lid was approximated with the same cubic function. The eyebrow’s thickness above and below the middle line was calculated from three segments as a parabola, a straight line and a quarter of a circle as the inner corner. A thickness function was added and subtracted to and from the middle line of the eyebrow. The shape of the mouth varies strongly as emotion changes from sadness to happiness or disgust. The manipulation of the face for setting a certain expression implies to mix different emotions. Each emotion has a percentage value by which they contribute to the face general expression. The new control set values for the visual features are computed by the difference of each emotion control set and the neutral face control set, and make a linear combination of the resulting six vectors.

TESTING AND RESULTS

The following steps have been taken into account for training the models for the facial expression recognition system:

- Obtaining the (Cohn-Kanade) database for building the system's knowledge
- Conversion of data base images from 'png' to 'bmp' format, 24 bits/pixel
- Increasing the quality of the images through some enhancement procedures (light, removing strips, applying filters, etc.)
- Extracting some Facial Characteristic Points (FCPs) by using a special tool (FCP Management Application)
- Computing the value of some parameters according to a given model. Applying a discretization procedure by using a special application (Parameter Discretization Application)
- Determining the facial expression for each of the samples in the database by analyzing the sequence of Action Units (AUs). The tool used to process the files in the database was Facial Expression Assignement Application.
- Using different kind of reasoning mechanisms for emotion recognition. The training step took into account the data provided from the previous steps. Bayesian Belief Networks (BBN) and back-propagation Artificial Neuronal Networks (ANN) were the main modalities for recognition.
- Principal Component Analysis technique was used as an enhancement procedure for the emotion recognition.

The steps that imply testing the recognition models are:

- Capture the video signal containing the facial expression
- Detecting the Facial Characteristic Points automatically
- Computing the value of the model parameters
- Using the parameter values for emotion detection

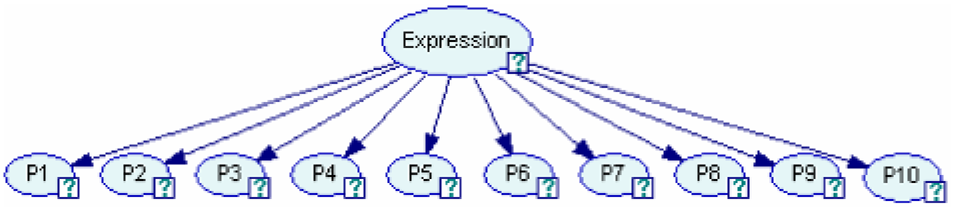
BBN experiment 1

“Detection of facial expressions from low-level parameters”

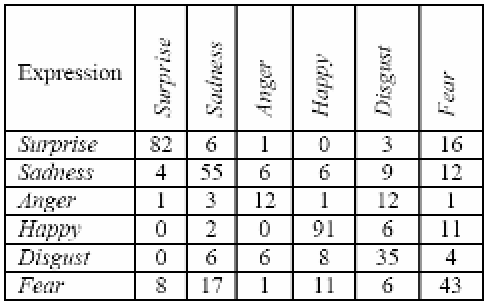
Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 10 size vector
- The network contains 10 parameters, each parameter has 5 states

The topology of the network

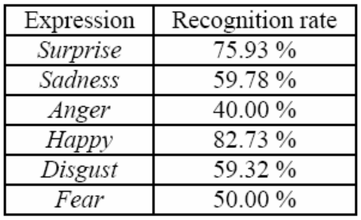


Recognition results. Confusion Matrix.



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	82	6	1	0	3	16
Sadness	4	55	6	6	9	12
Anger	1	3	12	1	12	1
Happy	0	2	0	91	6	11
Disgust	0	6	6	8	35	4
Fear	8	17	1	11	6	43

General recognition rate is 65.57%



Expression	Recognition rate
Surprise	75.93 %
Sadness	59.78 %
Anger	40.00 %
Happy	82.73 %
Disgust	59.32 %
Fear	50.00 %

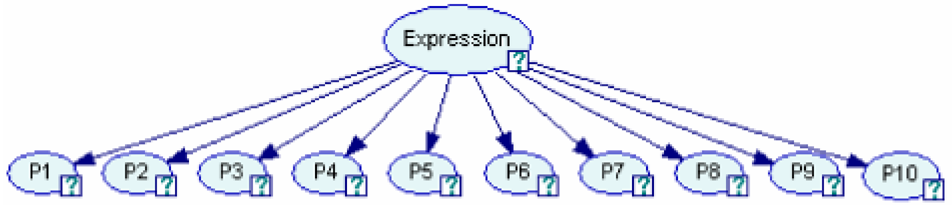
BBN experiment 2

"Detection of facial expressions from low-level parameters"

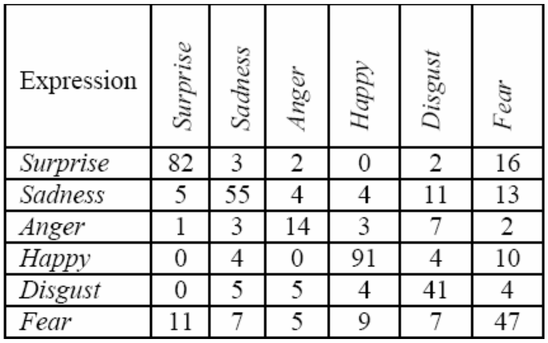
Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 10 size vector
- The network contains 10 parameters, each parameter has 8 states

The topology of the network

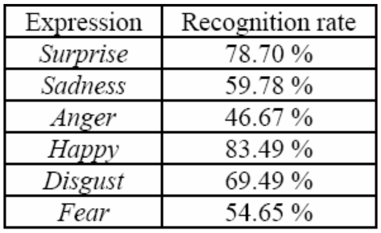


Recognition results. Confusion Matrix.



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	82	3	2	0	2	16
Sadness	5	55	4	4	11	13
Anger	1	3	14	3	7	2
Happy	0	4	0	91	4	10
Disgust	0	5	5	4	41	4
Fear	11	7	5	9	7	47

General recognition rate is 68.80%



Expression	Recognition rate
Surprise	78.70 %
Sadness	59.78 %
Anger	46.67 %
Happy	83.49 %
Disgust	69.49 %
Fear	54.65 %

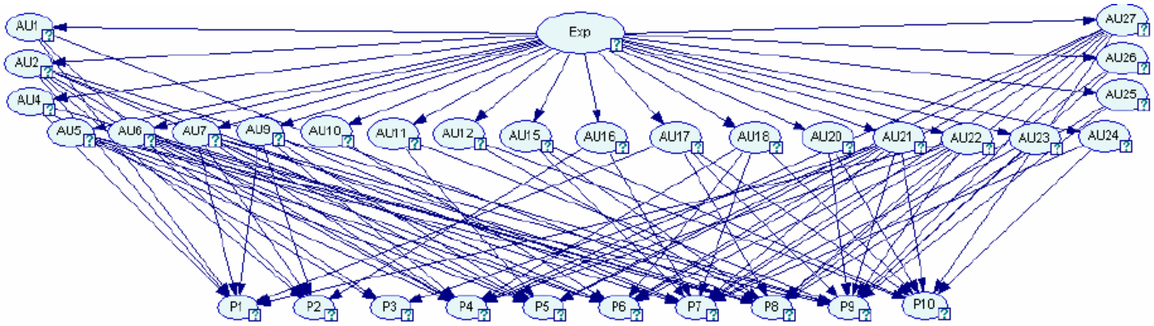
BBN experiment 3

“Facial expression recognition starting form the value of parameters”

Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 22+10 size vector containing all the analyzed Action Units and the 10 parameter values
- The network contains one parameter for expression recognition + 22 parameters for encoding Aus, each parameter has 2 states (present/absent) + 10 parameters for dealing with the value of the model analyzed parameters, 5 states

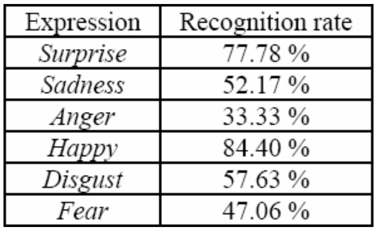
The topology of the network



Recognition results. Confusion Matrix.

Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	84	3	1	0	3	17
Sadness	3	48	6	6	7	22
Anger	0	7	10	1	10	2
Happy	0	2	0	92	6	9
Disgust	0	7	5	8	34	5
Fear	13	14	1	10	7	40

General recognition rate is 63.77%



Expression	Recognition rate
Surprise	77.78 %
Sadness	52.17 %
Anger	33.33 %
Happy	84.40 %
Disgust	57.63 %
Fear	47.06 %

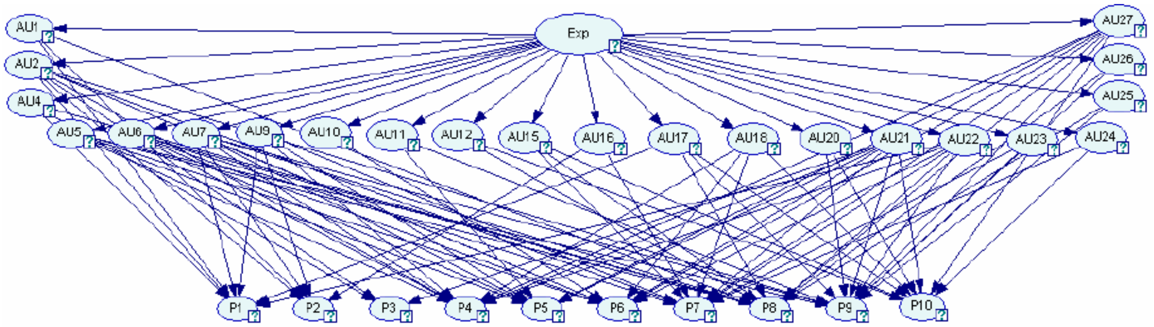
BBN experiment 4

“Facial expression recognition starting form the value of parameters”

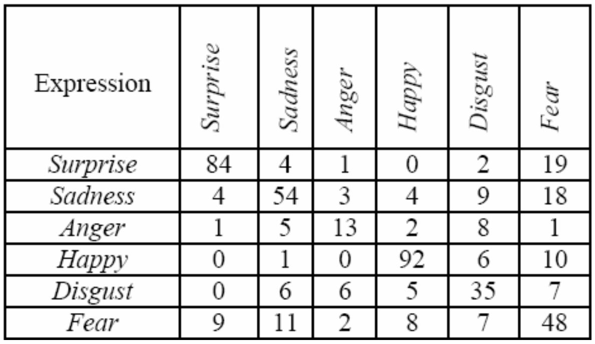
Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples.
- Each sample is a 22+10 size vector containing all the analyzed Action Units and the 10 parameter values.
- The network contains one parameter for expression recognition + 22 parameters for encoding
- Aus , each parameter has 2 states (present/absent) + 10 parameters for dealing with the value of the model analyzed parameters, <u>8 states</u>
- Each of the 10 model parameters takes values in interval [1..8].

The topology of the network

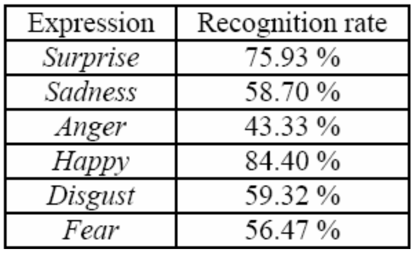


Recognition results. Confusion Matrix.



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	84	4	1	0	2	19
Sadness	4	54	3	4	9	18
Anger	1	5	13	2	8	1
Happy	0	1	0	92	6	10
Disgust	0	6	6	5	35	7
Fear	9	11	2	8	7	48

General recognition rate is 67.08%



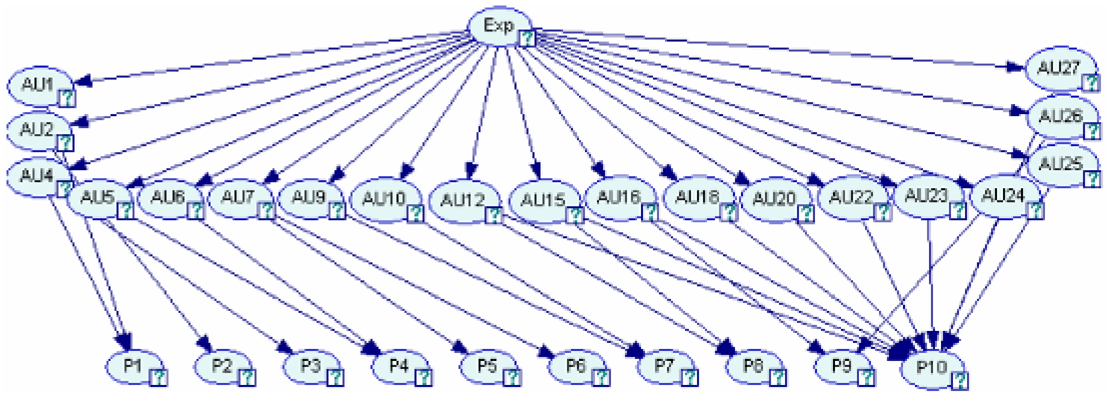
Expression	Recognition rate
Surprise	75.93 %
Sadness	58.70 %
Anger	43.33 %
Happy	84.40 %
Disgust	59.32 %
Fear	56.47 %

BBN experiment 5

Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 10 size vector
- The network contains 10 parameters

The topology of the network



Recognition results. Confusion Matrix.

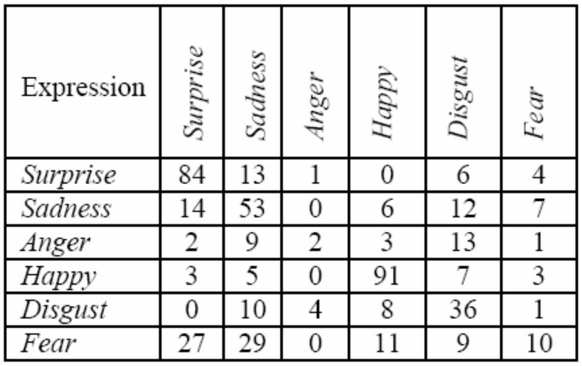
3 states model

Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	85	14	0	1	6	2
Sadness	20	46	0	11	11	4
Anger	1	7	1	5	13	3
Happy	1	5	0	94	9	0
Disgust	3	5	2	10	36	3
Fear	32	25	2	16	9	2

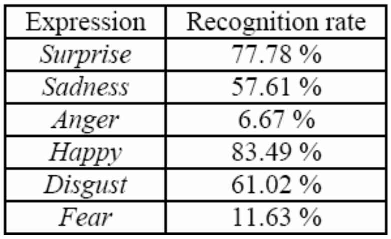
Expression	Recognition rate
Surprise	78.70 %
Sadness	50.00 %
Anger	3.33 %
Happy	86.24 %
Disgust	61.02 %
Fear	2.33 %

General recognition rate is 54.55 %

5 states model



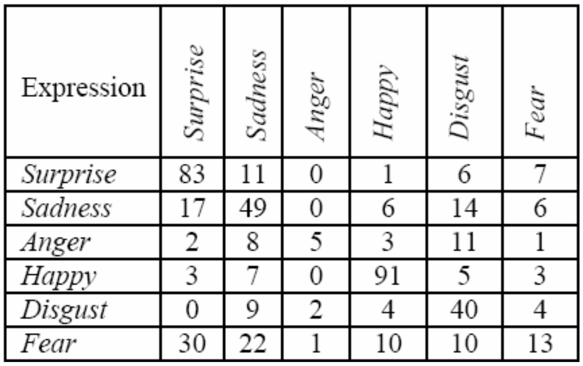
Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	84	13	1	0	6	4
Sadness	14	53	0	6	12	7
Anger	2	9	2	3	13	1
Happy	3	5	0	91	7	3
Disgust	0	10	4	8	36	1
Fear	27	29	0	11	9	10



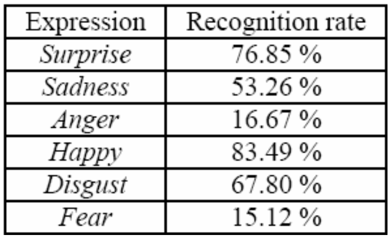
Expression	Recognition rate
Surprise	77.78 %
Sadness	57.61 %
Anger	6.67 %
Happy	83.49 %
Disgust	61.02 %
Fear	11.63 %

General recognition rate is 57.02%

8 states model



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	83	11	0	1	6	7
Sadness	17	49	0	6	14	6
Anger	2	8	5	3	11	1
Happy	3	7	0	91	5	3
Disgust	0	9	2	4	40	4
Fear	30	22	1	10	10	13



Expression	Recognition rate
Surprise	76.85 %
Sadness	53.26 %
Anger	16.67 %
Happy	83.49 %
Disgust	67.80 %
Fear	15.12 %

General recognition rate is 58.06 %

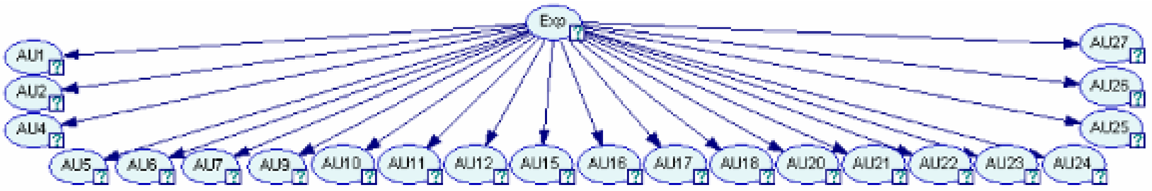
BBN experiment 6

“Recognition of facial expressions from AU combinations”

Details on the analysis

- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 22+1 size vector
- The Expression parameter (Exp) has six states according to the basic emotions to be classified

The topology of the network



Recognition results. Confusion Matrix.

Expressio n	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	107	1	0	0	0	0
Sadness	0	81	1	0	0	10
Anger	0	1	28	0	1	0
Happy	0	0	0	110	0	0
Disgust	0	4	1	0	51	3
Fear	0	2	0	0	0	84

General recognition rate is 95.05 %

Expression	Recognition rate
Surprise	99.07 %
Sadness	88.04 %
Anger	93.33 %
Happy	100.00 %
Disgust	86.44 %
Fear	97.67 %

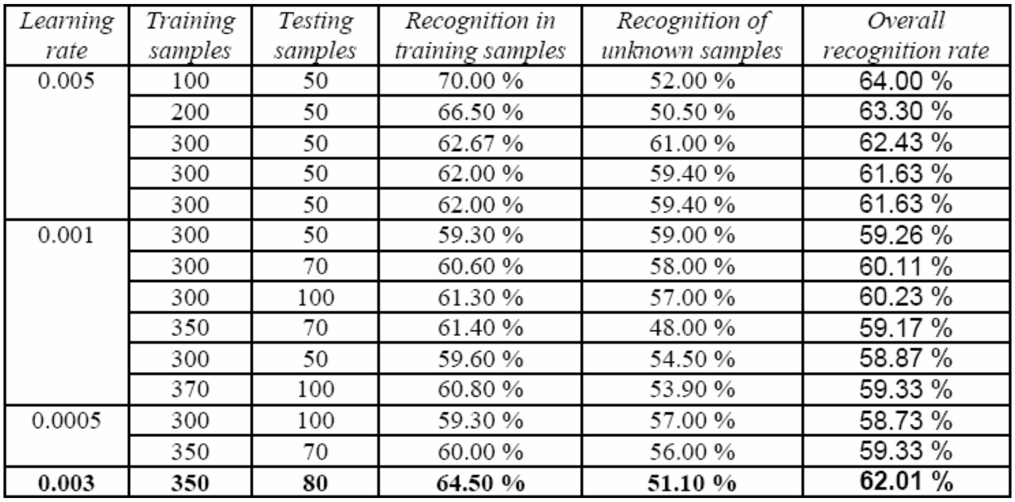
LVQ experiment

"LVQ based facial expression recognition experiments"

Details on the analysis

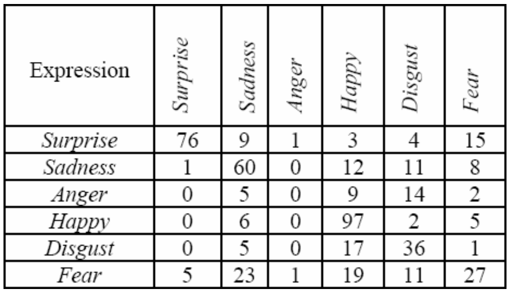
- 485 training samples.
- Each sample is a 10 size vector containing all the analyzed Action Units + one parameter for expression recognition
- Each of the 10 model parameters takes values in interval [1..5].

Recognition results



Learning rate	*Training samples*	*Testing samples*	*Recognition in training samples*	*Recognition of unknown samples*	*Overall recognition rate*
0.005	100	50	70.00 %	52.00 %	64.00 %
	200	50	66.50 %	50.50 %	63.30 %
	300	50	62.67 %	61.00 %	62.43 %
	300	50	62.00 %	59.40 %	61.63 %
	300	50	62.00 %	59.40 %	61.63 %
0.001	300	50	59.30 %	59.00 %	59.26 %
	300	70	60.60 %	58.00 %	60.11 %
	300	100	61.30 %	57.00 %	60.23 %
	350	70	61.40 %	48.00 %	59.17 %
	300	50	59.60 %	54.50 %	58.87 %
	370	100	60.80 %	53.90 %	59.33 %
0.0005	300	100	59.30 %	57.00 %	58.73 %
	350	70	60.00 %	56.00 %	59.33 %
0.003	**350**	**80**	**64.50 %**	**51.10 %**	**62.01 %**

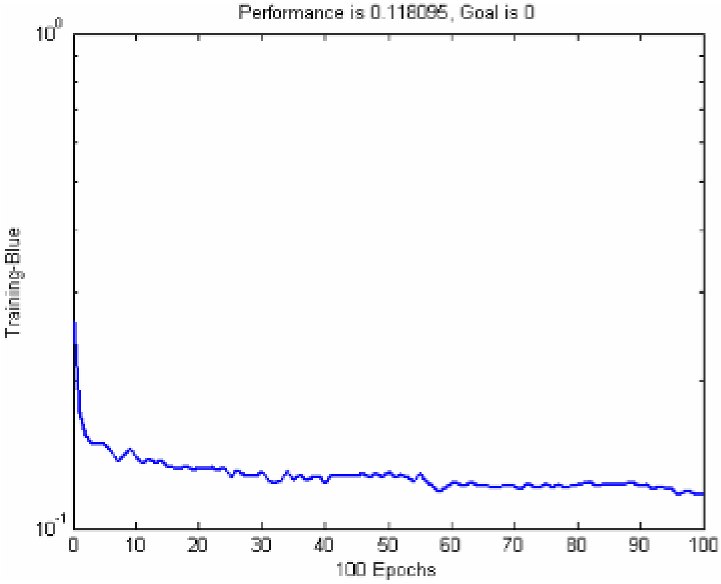
Confusion Matrix. (350 training samples, 80 test samples)



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	76	9	1	3	4	15
Sadness	1	60	0	12	11	8
Anger	0	5	0	9	14	2
Happy	0	6	0	97	2	5
Disgust	0	5	0	17	36	1
Fear	5	23	1	19	11	27

General recognition rate is 61.03 %

Expression	Recognition rate
Surprise	70.37 %
Sadness	65.21 %
Anger	0.00 %
Happy	88.18 %
Disgust	61.01 %
Fear	31.39 %



ANN experiment

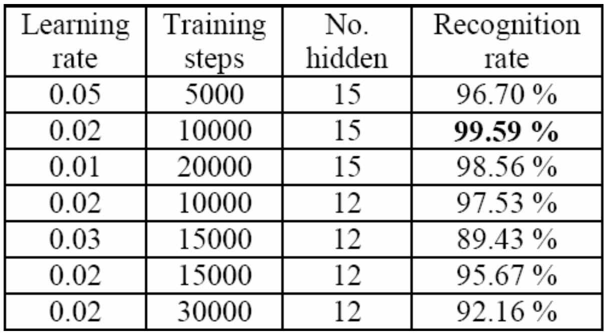
Back Propagation Neural Network ANN experiments

1. Recognition of facial expressions from model parameter values

Details on the analysis

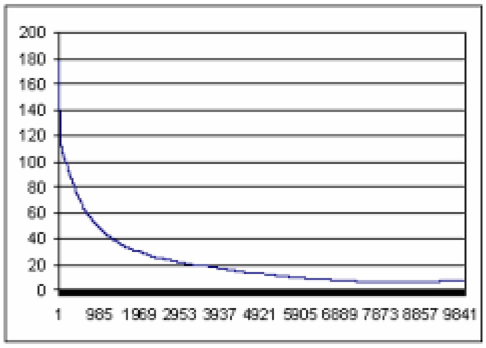
- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 10 size vector
- The network contains 10 parameters, each parameter has 7 states and its value is represented by using 3 input neurons.
- The network has three layers. The first layer contains 30 input neurons. The third layer contains 3 output neurons and gives the possibility to represent the values associated to the 6 basic expression classes.

Recognition results on different network topologies.



Learning rate	Training steps	No. hidden	Recognition rate
0.05	5000	15	96.70 %
0.02	10000	15	**99.59 %**
0.01	20000	15	98.56 %
0.02	10000	12	97.53 %
0.03	15000	12	89.43 %
0.02	15000	12	95.67 %
0.02	30000	12	92.16 %

Learning error graphs.



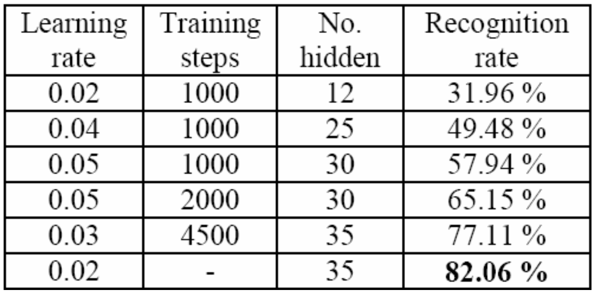
30:15:3, 10000 training steps, 0.02 learning .rate 99.59% facial expression recognition

2. Recognition of Action Units (AUs) from model parameter values

Details on the analysis

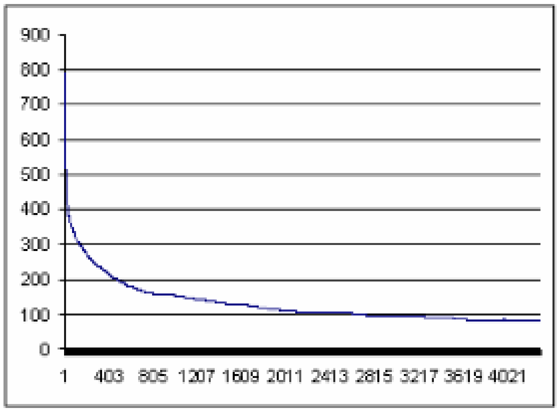
- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 22 size vector
- The network contains 2 parameters; each parameter is represented by using one input neuron.
- The network has three layers. The first layer contains 30 input neurons. The third layer contains 22 output neurons and gives the possibility to encode the presence/absence of all the 22 AUs.

Recognition results on different network topologies.



Learning rate	Training steps	No. hidden	Recognition rate
0.02	1000	12	31.96 %
0.04	1000	25	49.48 %
0.05	1000	30	57.94 %
0.05	2000	30	65.15 %
0.03	4500	35	77.11 %
0.02	-	35	**82.06 %**

Learning error graphs.



30:35:3, 4500 training steps, 0.03 learning .rate 77.11 % AU recognition

PCA experiment

“Principal Component Analysis PCA for Facial Expression Recognition”

Details on the analysis

- 485 facial expression sample vectors with labels
- 10 values per facial expression vector
- Each vector value takes values in interval [1..5]
- Pearson correlation coefficient (normed PCA, variances with 1/n)
- Without axes rotation
- Number of factors associated with non trivial eigenvalues: 10

Results of the processing

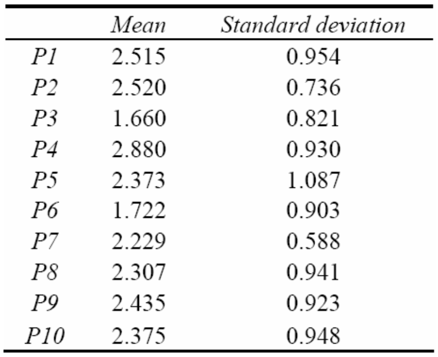
Bartlett's sphericity test:

Chi-square (observed value)	1887.896
Chi-square (critical value)	61.656
DF	45
One-tailed p-value	< 0.0001
Alpha	0.05

Conclusion:

At the level of significance Alpha=0.050 the decision is to reject the null hypothesis of absence of significant correlation between variables. It means that the correlation between variables is significant.

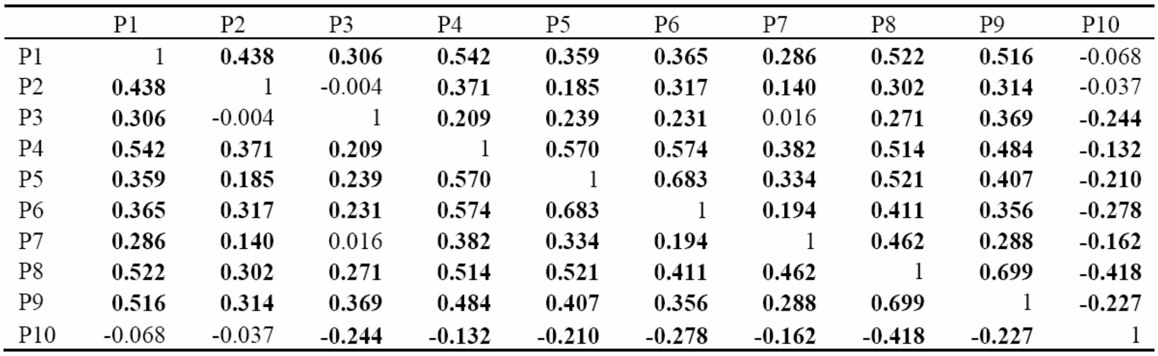
Mean and standard deviation of the columns:



	Mean	*Standard deviation*
P1	2.515	0.954
P2	2.520	0.736
P3	1.660	0.821
P4	2.880	0.930
P5	2.373	1.087
P6	1.722	0.903
P7	2.229	0.588
P8	2.307	0.941
P9	2.435	0.923
P10	2.375	0.948

Correlation matrix:

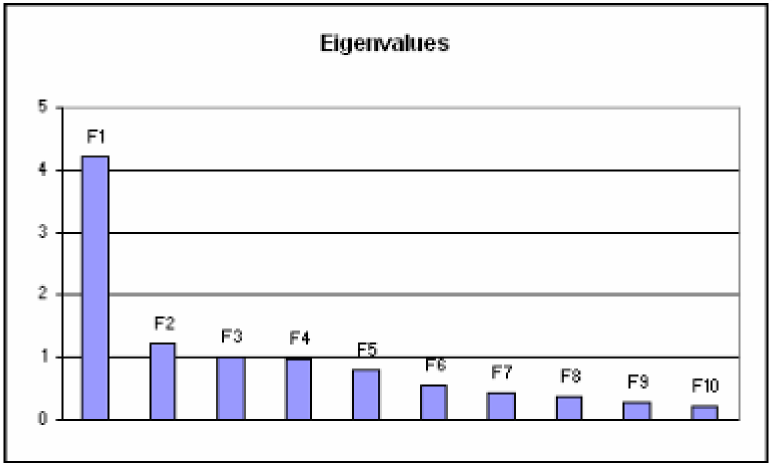
In bold, significant values (except diagonal) at the level of significance alpha=0.050 (two-tailed test)



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	1	**0.438**	**0.306**	**0.542**	**0.359**	**0.365**	**0.286**	**0.522**	**0.516**	-0.068
P2	**0.438**	1	-0.004	**0.371**	**0.185**	**0.317**	**0.140**	**0.302**	**0.314**	-0.037
P3	**0.306**	-0.004	1	**0.209**	**0.239**	**0.231**	0.016	**0.271**	**0.369**	**-0.244**
P4	**0.542**	**0.371**	**0.209**	1	**0.570**	**0.574**	**0.382**	**0.514**	**0.484**	**-0.132**
P5	**0.359**	**0.185**	**0.239**	**0.570**	1	**0.683**	**0.334**	**0.521**	**0.407**	**-0.210**
P6	**0.365**	**0.317**	**0.231**	**0.574**	**0.683**	1	**0.194**	**0.411**	**0.356**	**-0.278**
P7	**0.286**	**0.140**	0.016	**0.382**	**0.334**	**0.194**	1	**0.462**	**0.288**	**-0.162**
P8	**0.522**	**0.302**	**0.271**	**0.514**	**0.521**	**0.411**	**0.462**	1	**0.699**	**-0.418**
P9	**0.516**	**0.314**	**0.369**	**0.484**	**0.407**	**0.356**	**0.288**	**0.699**	1	**-0.227**
P10	-0.068	-0.037	**-0.244**	**-0.132**	**-0.210**	**-0.278**	**-0.162**	**-0.418**	**-0.227**	1

Eigenvalues:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Eigenvalue	4.198	1.207	1.006	0.962	0.805	0.539	0.423	0.371	0.278	0.211
% variance	41.976	12.072	10.065	9.617	8.054	5.393	4.226	3.715	2.776	2.107
Cumulative %	41.976	54.048	64.113	73.730	81.784	87.177	91.403	95.118	97.893	100.000

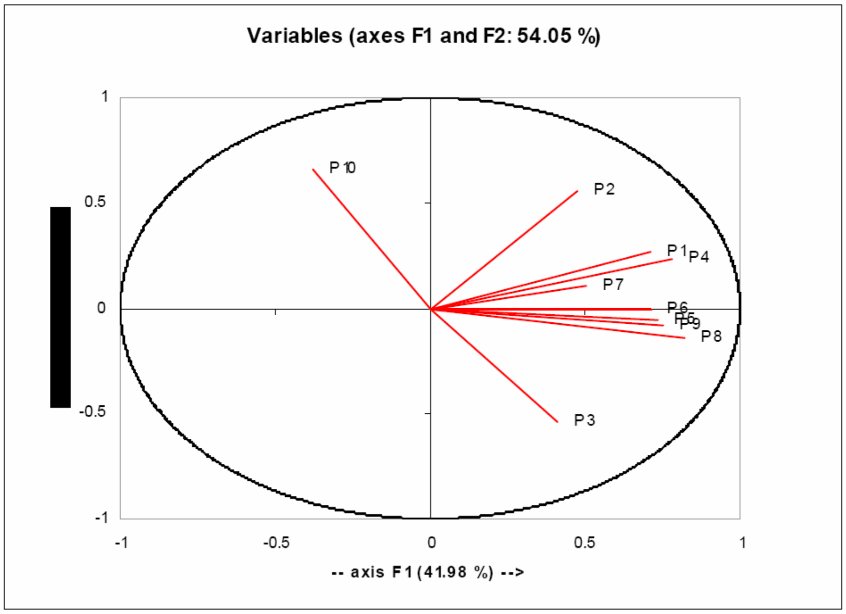


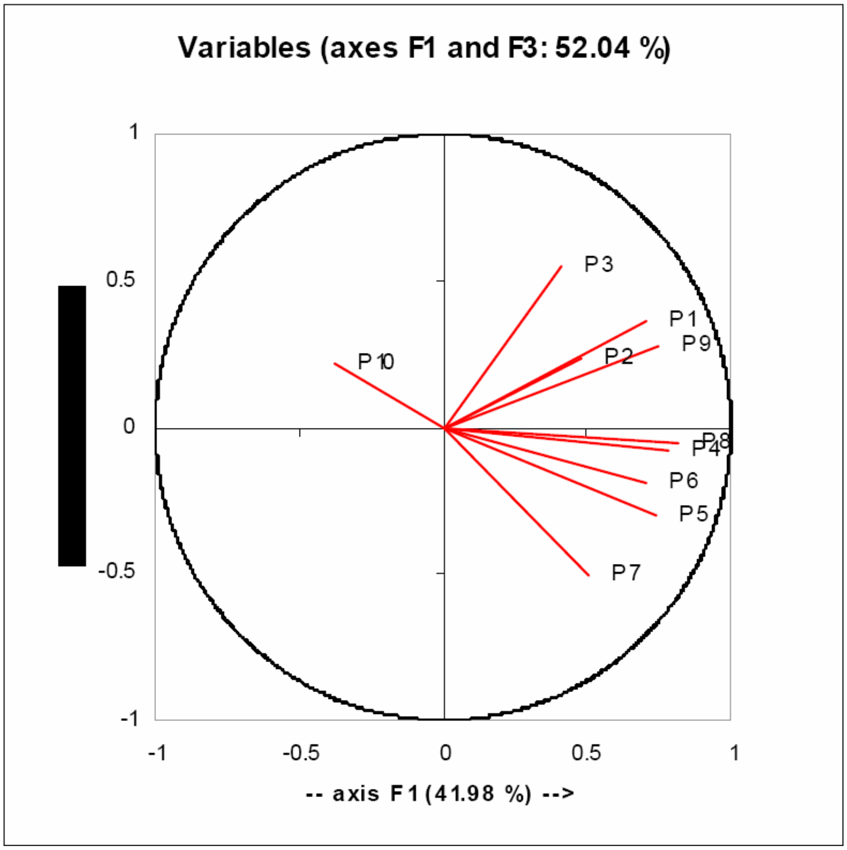
Eigenvectors:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
P1	0.346	0.242	0.362	-0.130	0.101	-0.224	0.612	-0.452	0.101	-0.153
P2	0.233	0.511	0.232	-0.030	-0.592	-0.223	-0.443	-0.003	-0.179	-0.034
P3	0.201	-0.494	0.549	0.121	0.289	-0.426	-0.329	0.063	-0.093	0.117
P4	0.382	0.212	-0.080	0.166	0.150	-0.102	0.328	0.762	-0.230	0.029
P5	0.359	-0.047	-0.300	0.405	0.200	0.173	-0.173	-0.364	-0.494	-0.373
P6	0.346	-0.005	-0.189	0.570	-0.129	-0.033	-0.053	-0.090	0.641	0.284
P7	0.247	0.095	-0.509	-0.504	0.315	-0.440	-0.272	-0.031	0.212	-0.073
P8	0.400	-0.130	-0.054	-0.330	-0.065	0.341	0.025	-0.157	-0.266	0.703
P9	0.367	-0.072	0.274	-0.265	0.060	0.575	-0.210	0.201	0.354	-0.413
P10	-0.186	0.598	0.214	0.133	0.608	0.191	-0.249	-0.066	0.035	0.266

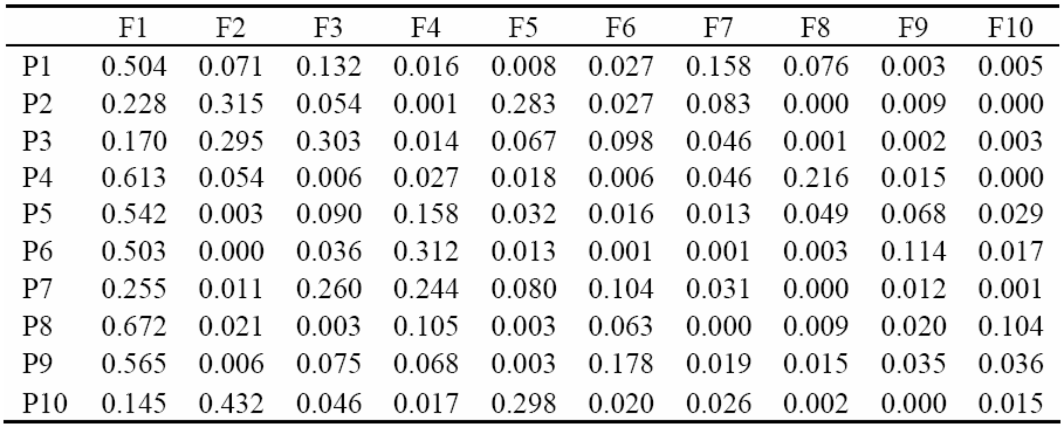
Factor loadings:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
P1	0.710	0.266	0.363	-0.128	0.091	-0.164	0.398	-0.276	0.053	-0.070
P2	0.478	0.561	0.233	-0.030	-0.532	-0.163	-0.288	-0.002	-0.094	-0.016
P3	0.413	-0.543	0.550	0.119	0.259	-0.313	-0.214	0.039	-0.049	0.054
P4	0.783	0.233	-0.081	0.163	0.135	-0.075	0.214	0.464	-0.121	0.013
P5	0.737	-0.051	-0.301	0.397	0.179	0.127	-0.112	-0.222	-0.260	-0.171
P6	0.709	-0.005	-0.189	0.559	-0.116	-0.025	-0.034	-0.055	0.338	0.130
P7	0.505	0.105	-0.510	-0.494	0.282	-0.323	-0.177	-0.019	0.112	-0.033
P8	0.820	-0.143	-0.054	-0.324	-0.058	0.250	0.017	-0.096	-0.140	0.323
P9	0.752	-0.079	0.275	-0.260	0.053	0.422	-0.137	0.123	0.186	-0.190
P10	-0.380	0.657	0.215	0.130	0.546	0.141	-0.162	-0.040	0.018	0.122


Variables (axes F1 and F2: 54.05 %)
1
0.5
0
-0.5
-1
[redacted]
P10
P2
P1
P4
P7
P6
P5
P9
P8
P3
-1
-0.5
0
0.5
1
-- axis F1 (41.98 %) -->

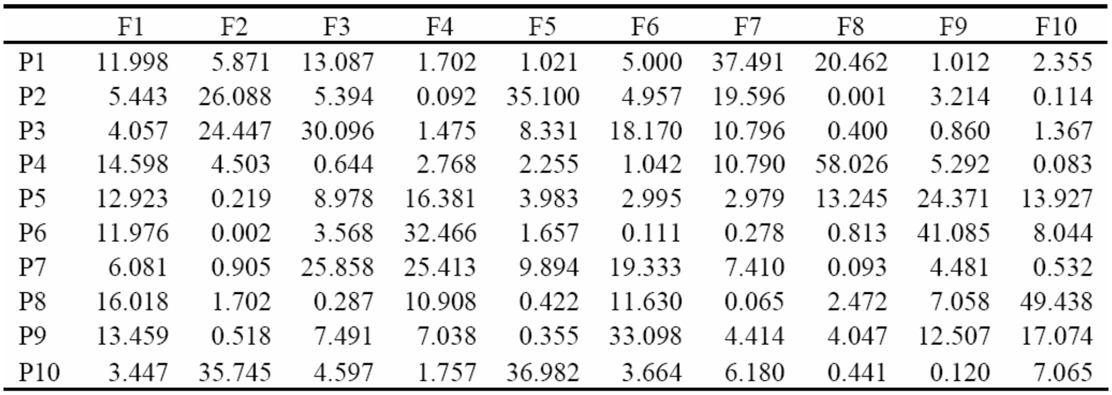

Variables (axes F1 and F3: 52.04 %)
1
0.5
0
-0.5
-1
[redacted]
P3
P1
P9
P2
P10
P8
P4
P6
P5
P7
-1
-0.5
0
0.5
1
-- axis F1 (41.98 %) -->

Squared cosines of the variables:

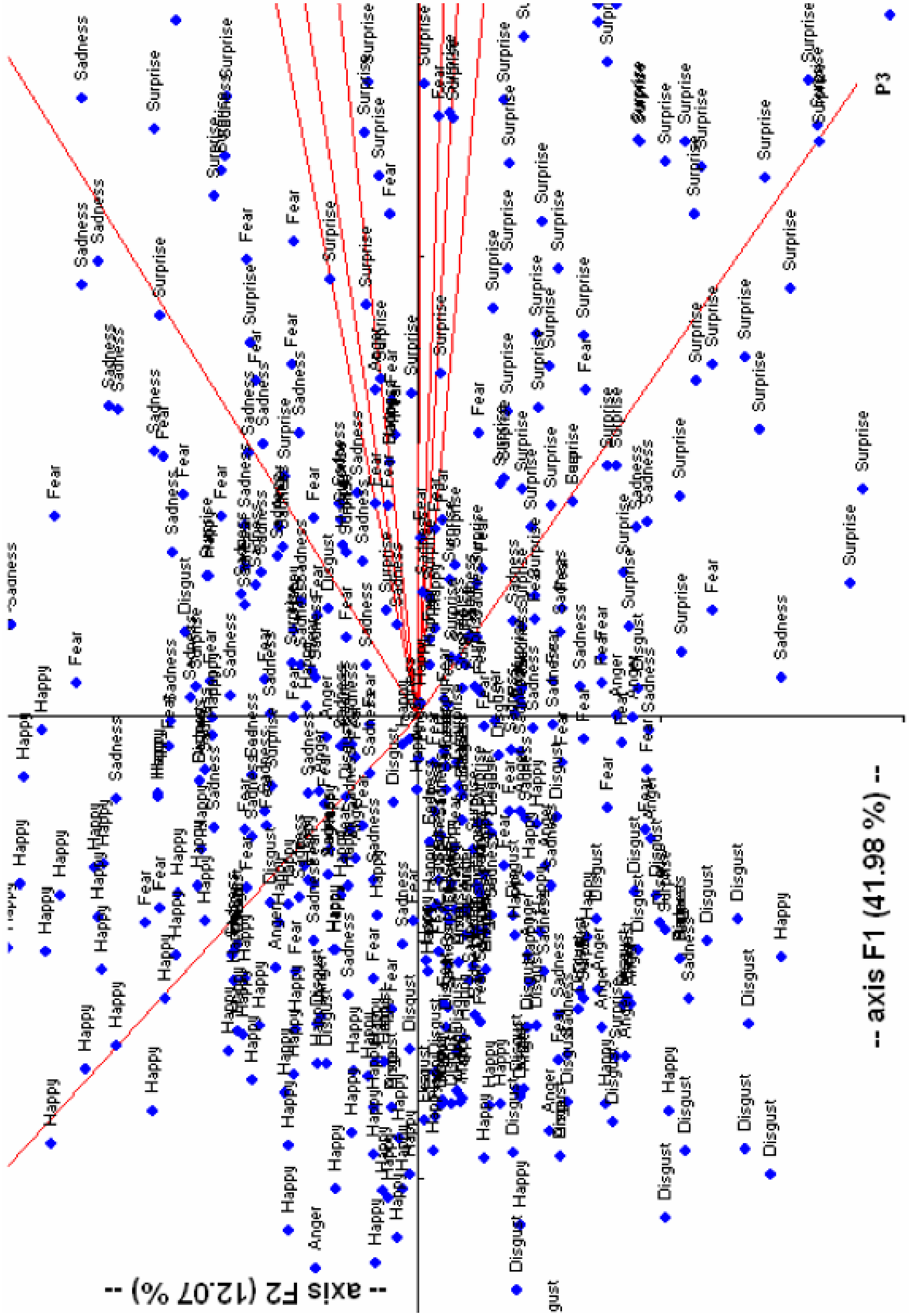


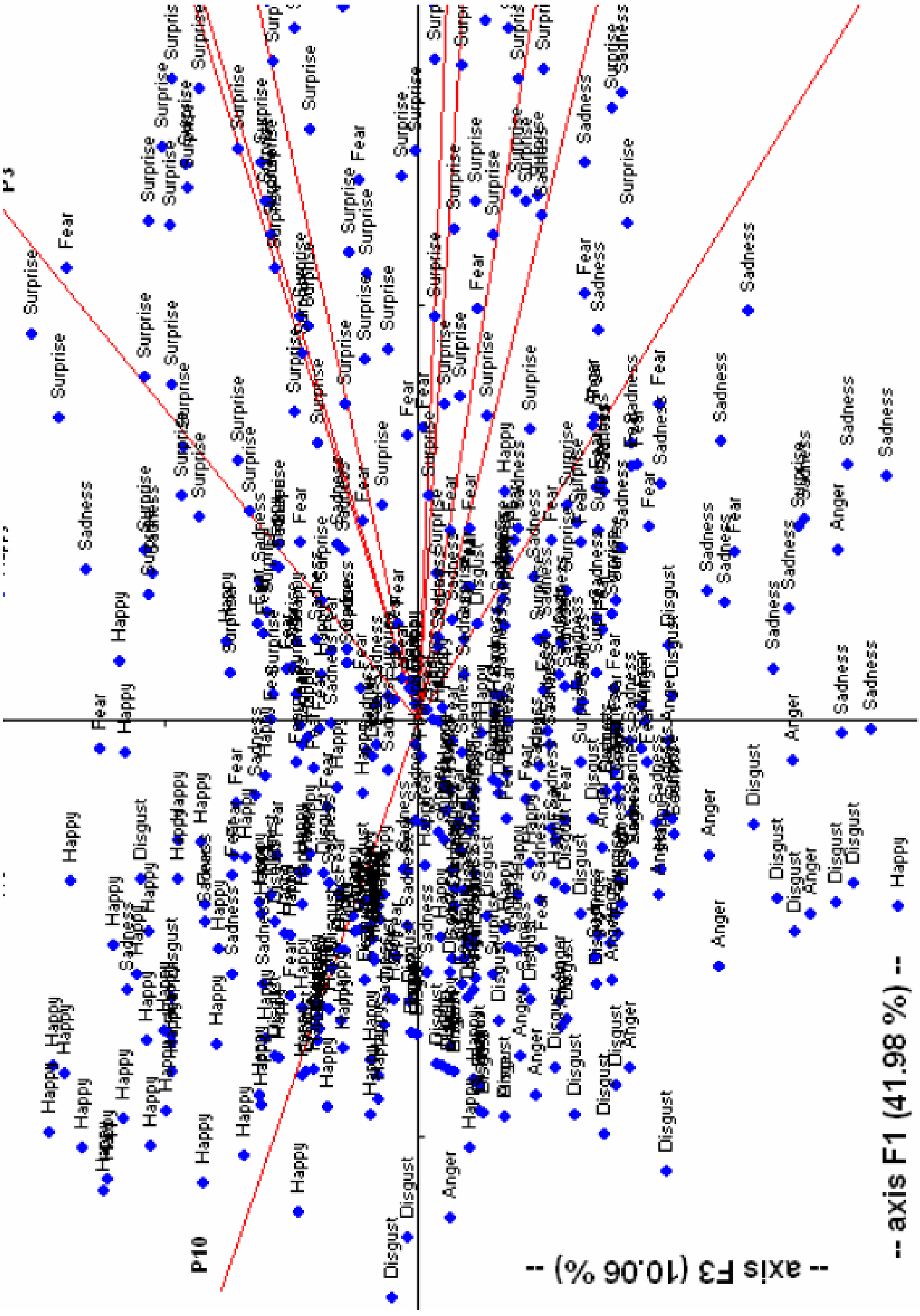
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
P1	0.504	0.071	0.132	0.016	0.008	0.027	0.158	0.076	0.003	0.005
P2	0.228	0.315	0.054	0.001	0.283	0.027	0.083	0.000	0.009	0.000
P3	0.170	0.295	0.303	0.014	0.067	0.098	0.046	0.001	0.002	0.003
P4	0.613	0.054	0.006	0.027	0.018	0.006	0.046	0.216	0.015	0.000
P5	0.542	0.003	0.090	0.158	0.032	0.016	0.013	0.049	0.068	0.029
P6	0.503	0.000	0.036	0.312	0.013	0.001	0.001	0.003	0.114	0.017
P7	0.255	0.011	0.260	0.244	0.080	0.104	0.031	0.000	0.012	0.001
P8	0.672	0.021	0.003	0.105	0.003	0.063	0.000	0.009	0.020	0.104
P9	0.565	0.006	0.075	0.068	0.003	0.178	0.019	0.015	0.035	0.036
P10	0.145	0.432	0.046	0.017	0.298	0.020	0.026	0.002	0.000	0.015

Contributions of the variables (%):



	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
P1	11.998	5.871	13.087	1.702	1.021	5.000	37.491	20.462	1.012	2.355
P2	5.443	26.088	5.394	0.092	35.100	4.957	19.596	0.001	3.214	0.114
P3	4.057	24.447	30.096	1.475	8.331	18.170	10.796	0.400	0.860	1.367
P4	14.598	4.503	0.644	2.768	2.255	1.042	10.790	58.026	5.292	0.083
P5	12.923	0.219	8.978	16.381	3.983	2.995	2.979	13.245	24.371	13.927
P6	11.976	0.002	3.568	32.466	1.657	0.111	0.278	0.813	41.085	8.044
P7	6.081	0.905	25.858	25.413	9.894	19.333	7.410	0.093	4.481	0.532
P8	16.018	1.702	0.287	10.908	0.422	11.630	0.065	2.472	7.058	49.438
P9	13.459	0.518	7.491	7.038	0.355	33.098	4.414	4.047	12.507	17.074
P10	3.447	35.745	4.597	1.757	36.982	3.664	6.180	0.441	0.120	7.065


-- axis F1 (41.98 %) --
-- axis F2 (12.07 %) --
P3
Happy
Sadness
Fear
Surprise
Disgust
Anger

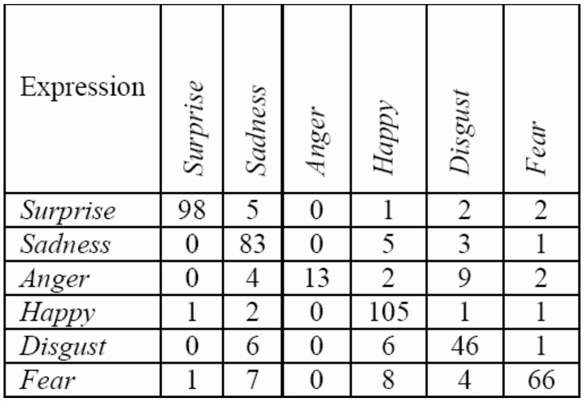

-- axis F1 (41.98 %) --
-- axis F3 (10.06 %) --
P3
P10
Happy
Surprise
Fear
Sadness
Disgust
Anger

PNN experiment

Details on the analysis

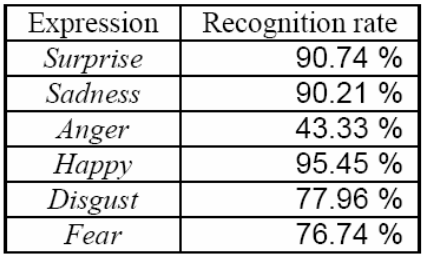
- Conditioned probability tables contain values taken from 485 training samples
- Each sample is a 10 size vector
- The network contains 10 parameters, each parameter has 5 states

Recognition results. Confusion Matrix.



Expression	*Surprise*	*Sadness*	*Anger*	*Happy*	*Disgust*	*Fear*
Surprise	98	5	0	1	2	2
Sadness	0	83	0	5	3	1
Anger	0	4	13	2	9	2
Happy	1	2	0	105	1	1
Disgust	0	6	0	6	46	1
Fear	1	7	0	8	4	66

General recognition rate is 84.76 %



Expression	Recognition rate
Surprise	90.74 %
Sadness	90.21 %
Anger	43.33 %
Happy	95.45 %
Disgust	77.96 %
Fear	76.74 %

CONCLUSION

The human face has attracted attention in the areas such as psychology, computer vision, and computer graphics. Reading and understanding people expression have become one of the most important research areas in recognition of human face. Many computer vision researchers have been working on tracking and recognition of the whole or parts of face. However, the problem of facial expression recognition has not totally been solved yet. The current project addresses the aspects related to the development of an automatic probabilistic recognition system for facial expressions in video streams.

The coding system used for encoding the complex facial expressions is inspired by Ekman's Facial Action Coding System. The description of the facial expressions is given according to sets of atomic Action Units (AU) from the Facial Action Coding System (FACS). Emotions are complex phenomena that involve a number of related subsystems and can be activated by any one (or by several) of them.

In order to make the data ready for the learning stage of the recognition system, a complete set of software applications was developed. The initial image files from the training database were processed for extracting the essential information in a semiautomatic manner. Further the data were transformed so as to fit the requirements of the learning stage for each kind of classifier.

The current project presents a fully automatic method, requiring no such human specification. The system first robustly detects the pupils using an infrared sensitive camera equipped with infrared LEDs. The face analysis component integrates an eye tracking mechanism based on Kalman filter.

For each frame, the pupil positions are used to localize and normalize the other facial regions. The visual feature detection includes PCA oriented recognition for ranking the activity in certain facial areas.

The recognition system consists mainly of two stages: a training stage, where the classifier function is learnt, and a testing stage, where the learnt classifier function classifies new data.

These parameters are used as input to classifiers based on Bayesian Beliaf Networks, neural networks or other classifiers to recognize upper facial action units and all their possible combinations.

The base for the expression recognition engine is supported through a BBN model that also handles the time behavior of the visual features.

On a completely natural dataset with lots of head movements, pose changes and occlusions (Cohn-Kanade AU-coded facial expression database), the new probabilistic framework (based on BBN) achieved a recognition accuracy of 68 %.

Other experiments implied the use of Linear Vector Quantization (LVQ) method, Probabilistic Neural Networks or Back-Prop Neural Networks. The results can be seen in the Experiments section of the report.

There are some items on the design of the project that have not been fully covered yet. Among them, the most important is the inclusion of temporal-based parameters to be used in the recognition process. At the moment of running the experiments there were no data available on the dynamic behavior of the model parameters. However, the dynamic aspects of the parameters constitutes subject to further research in the field of facial expression recognition.

REFERENCES

Almageed, W. A., M. S. Fadali, G. Bebis, ‘A non-intrusive Kalman Filter-Based Tracker for Pursuit Eye Movement’, Proceedings of the 2002 American Control Conference Alaska, 2002

Azarbayejani, A., A. Pentland ‘Recursive estimation of motion, structure, and focal length’ IEEE Trans. PAMI, 17(6), 562-575, June 1995

Baluja, S., D. Pomerleau, ‘Non-intrusive Gaze Tracking Using Artificial Neural Networks’, Technical Report CMU-CS-94-102. Carnegie Mellon University, 1994

Bartlett, M. S., G. Littlewort, I. Fasel, J. R. Movellan ‘Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction’ IEEE Workshop on Face Processing in Video, Washington, 2004

Bartlett, M. S., G. Littlewort, C. Lainscsek, I. Fasel, J. Movellan, ‘Machine learning methods for fully automatic recognition of facial expressions and facial actions’, Proceedings of IEEE SMC, pp. 592–597, 2004

Bartlett, M. S., G. Littlewort, I. Fasel, J. R. Movellan, ‘Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction’, CVPR’03, 2003

Bartlett, M. A., J. C. Hager, P. Ekman, T. Sejnowski, ‘Measuring facial expressions by computer image analysis’, Psychophysiology, 36(2):253–263, March, 1999

Black, M., Y. Yacoob, ‘Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion’, Intel. J. of Computer Vision, 25(1), pp. 23-48, 1997

Black, M., Y. Yacoob, ‘Tracking and recognizing rigid and non-rigid facial motions using local parametric model of image motion’, In Proceedings of the International Conference on Computer Vision, pages 374–381, Cambridge, MA, IEEE Computer Society, 1995

Bourel, F., C. C. Chibelushi, A. A. Low, ‘Recognition of Facial Expressions in conditions of occlusion’, BMVC’01, pp. 213-222, 2001

Brown, R., P. Hwang, ‘Introduction to Random Signals and Applied Kalman Filtering’, 3rd edition, Wiley, 1996

Brunelli, R., T. Poggio, ‘Face recognition: Features vs. templates’, IEEE Trans. Pattern Analysis and Machine Intelligence, 15(10):1042–1053, 1993

Chang, J. Y., J. L. Chen, ‘A facial expression recognition system using neural networks’, IJCNN '99. International Joint Conference on Neural Networks, 1999, Volume: 5, pp. 3511 –3516, 1999

Cohen, I., N. Sebe, A. Garg, M. S.Lew, T. S. Huang ‘Facial expression recognition from video sequences’ Computer Vision and Image Understanding, Volume 91, pp 160 - 187 ISSN: 1077-3142 2003

Cootes, T. F., G. J. Edwards, C. J. Taylor, ‘Active appearance models’, Pattern Analysis and Machine Intelligence, 23(6), June 2001

Covell, M., ‘Eigen-points: control-point location using principal component analyses’ In Proceedings of Conference on Automatic Face and Gesture Recognition, October 1996
T. F. Cootes, G. J. Edwards, and C. J. Taylor. ‘Active appearance models’ Pattern Analysis and Machine Intelligence, 23(6), June 2001

Cowie, R., E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, ‘Emotion recognition in human-computer interaction’, IEEE Signal Processing Magazine, 18(1):33–80, January 2001

Datcu, D., L.J.M. Rothkrantz, ‘A multimodal workbench for automatic surveillance’ Euromedia’04, 2004

Datcu, D., L.J.M. Rothkrantz, ‘Automatic recognition of facial expressions using Bayesian belief networks’, Proceedings of IEEE SMC, pp. 2209–2214, 2004

deJongh, E. J., L.J.M. Rothkrantz ‘FED – an online Facial Expression Dictionary’ Euromedia’04, pp.115–119, April 2004.

Donato, G., M. Bartlett, J. Hager, P. Ekman, T. Sejnowski, ‘Classifying facial actions’, IEEE Pattern Analysis and Machine Intelligence, 21(10):974–989, October 1999

Druzdzel, M. J., ‘GeNIe: A development environment for graphical decision-analytic models’, In Proceedings of the 1999 Annual Symposium of the American Medical Informatics Association (AMIA-1999), page 1206, Washington, D.C., November 6-10, 1999

Ekman, P., W. V. Friesen, ‘Facial Action Coding System: Investigator’s Guide’, Consulting Psychologists Press, 1978

Ekman, P., W. V. Friesen, ‘The Facial Action Coding System: A Technique for Measurement of Facial Movement’, Consulting Psychologists Press, San Francisco, CA, 1978

Essa, A. Pentland, 'Coding, analysis, interpretation and recognition of facial expressions', Pattern Analysis and Machine Intelligence, 7:757–763, July 1997

Fellenz, W. A., J. G. Taylor, N. Tsapatsoulis, S. Kollias, "Comparing Template-based, Feature-based and Supervised Classification of Facial Expressions from Static Images", MLP CSCC'99 Proc., pp. 5331-5336, 1999

Feng, G. C., P. C. Yuen, D. Q. Dai, "Human face recognition using PCA on wavelet subband", Journal of Electronic Imaging -- April 2000 -- Volume 9, Issue 2, pp. 226-233, 2000

Fox, N. A., R. B. Reilly, 'Robust multi-modal person identification with tolerance of facial expression', Proceedings of IEEE SMC, pp. 580–585, 2004

Glenstrup, A., T. Angell-Nielsen, 'Eye Controlled Media, Present and Future State', Technical Report, University of Copenhagen, ttp://www.diku.dk/users/panic/eyegaze/, 1995

Gourier, N., D.Hall, J.L.Crowley, 'Facial feature detection robust to pose, illumination and identity', Proceedings of IEEE SMC 2004, pp. 617–622, 2004

Haro, I. Essa, M. Flickner, 'Detecting and tracking eyes by using their physiological properties', In Proceedings of Conference on Computer Vision and Pattern Recognition, June 2000

Jacob, R., 'Eye Tracking in Advanced Interface Design', In Advanced Interface Design and Virtual Environments, ed. W. Barfield and T. Furness, Oxford University Press, Oxford, 1994

Jun, S., Z. Qing, W. Wenyuan, "A improved facial recognition system method", ISBN: 978-3-540-41180-2, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, vol.1948, pp. 212-221, 2000

Kanade, T., J. Cohn, Y. Tian 'Comprehensive database for facial expression analysis' Proc. IEEE Int'l Conf. Face and Gesture Recognition, pp. 46-53, 2000

Kapoor, A., R. W. Picard. 'Real-time, fully automatic upper facial feature tracking' In Proceedings of Conference on Automatic Face and Gesture Recognition, May 2002

Kobayashi, H., F.Hara, 'Recognition of Six basic facial expression and their strength by neural network', Proceedings, IEEE International Workshop on Robot and Human Communication, pp. 381 –386, 1992

Kobayashi, H., F. Hara, 'Recognition of Mixed Facial Expressions by Neural Network', IEEE International workshop on Robot and Human Communication, 1972

Lien, J., T. Kanade, J. Cohn, C. C. Li, ‘Detection, tracking and classification of action units in facial expression’, Journal of Robotics and Autonomous Systems, 31:131–146, 2000

Lyons, M. J., J. Budynek, S. Akamatsu, ‘Automatic classification of single facial images’, IEEE Trans. Pattern Anal. Machine Intell., vol. 21, no. 12, pp. 1357–1362, 1999

Morimoto, C., D. Koons, A. Amir, M. Flickner, ‘Pupil detection and tracking using multiple light sources’, Technical report, IBM Almaden Research Center, 1998

Moriyama, T., J. Xiao, J. F. Cohn, T. Kanade, ‘Meticulously detailed eye model and its application to analysis of facial image’, Proceedings of IEEE SMC, pp.580–585, 2004

Padgett, C., G. Cottrell, “Representing face images for emotion classification”, In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems, vol. 9, Cambridge, MA, MIT Press, 1997

Padgett, C., G. Cottrell, R. Adolphs, “Categorical perception in facial emotion classification”, In Proceedings of the 18th Annual Conference of the Cognitive Science Society, Hillsdale NJ, Lawerence Erlbaum. 5, 1996

Pantic, M., L. J. M. Rothkrantz, ‘Toward an Affect-Sensitive Multimodal Human-Computer Interaction’ IEEE proceedings vol. 91, no. 9, pp. 1370-1390, 2003

Pantic, M., L. J. M. Rothkrantz, ‘Self-adaptive expert system for facial expression analysis’, IEEE International Conference on Systems, Man and Cybernetics (SMC ’00), pp. 73–79, October 2000

Pantic, M., L .J. M. Rothkrantz, ‘Automatic analysis of facial expressions: the state of the art’, IEEE Trans. PAMI, 22(12), 2000

Pantic, M., L. J. M. Rothkrantz, ‘An expert system for multiple emotional classification of facial expressions’, Proceedings, 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 113-120, 1999

Phillips, P., H. Moon, P. Rauss, S. Rizvi, ‘The FERET september 1996 database and evaluation procedure’ In Proc. First Int’l Conf. on Audio and Video-based Biometric Person Authentication, pages 12–14, Switzerland, 1997

Rosenblum, M., Y. Yacoob, L. S. Davis, ‘Human expression recognition from motion using a radial basis function network architecture’, IEEE Trans. NNs, vol. 7, no. 5, pp. 1121–1137, 1996

Salovey, P., J. D. Mayer, ‘Emotional intelligence’ Imagination, Cognition, and Personality, 9(3): 185-211, 1990

Samal, A., P. Iyengar, 'Automatic recognition and analysis of human faces and facial expression', A survey. Pattern Recognition, 25(1):65–77, 1992

Schweiger, R., P. Bayerl, H. Neumann, "Neural Architecture for Temporal Emotion Classification", ADS 2004, LNAI 2068, Springer-Verlag Berlin Heidelberg, pp. 49-52, 2004

Seeger, M., 'Learning with labeled and unlabeled data', Technical report, Edinburgh University, 2001

Stathopoulou, I. O., G. A. Tsihrintzis, 'An improved neuralnetwork-based face detection and facial expression classification system', Proceedings of IEEE SMC, pp. 666–671, 2004

Tian, Y., T. Kanade, J. F. Cohn. 'Recognizing upper face action units for facial expression analysis' In Proceedings of Conference on Computer Vision and Pattern Recognition, June 2000

Tian, Y., T. Kanade, J. F. Cohn. 'Recognizing action units for facial expression analysis' Pattern Analysis and Machin Intelligence, 23(2), February 2001

Turk, M., A. Pentland, 'Face recognition using eigenfaces", Proc. CVPR, pp. 586-591, 1991

Yacoob Y., L. Davis. 'Computing spatio-temporal representation of human faces' In CVPR, pages 70–75, Seattle, WA, June 1994

Yin, L., J.Loi, W.Xiong, 'Facial expression analysis based on enhanced texture and topographical structure', Proceedings of IEEE SMC, pp. 586–591, 2004

Zhang, Z., 'Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron' International Journal of Pattern Recognition and Artificial Intelligence, 13(6):893–911, 1999

Zhang, Z., M. Lyons. M. Schuster, S. Akamatsu, 'Comparison between geometry based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron', in Proc. IEEE 3rd Int'l Conf. on Automatic Face and Gesture Recognition, Nara, Japan, April 1998

Zhu, Z., Q. Ji, K. Fujimura, K. Lee, 'Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination,' in Proc. Int'l Conf. Pattern Recognition, Aug. 2002

Wang, X., X. Tang, 'Bayesian Face Recognition Using Gabor Features', Proceedings of the 2003 ACM SIGMM Berkley, California 2003

APPENDIX A

The routines for handling the Action Units sequences.

```
//-----------------------------------
.....................
#include "FACS.h"
void FACS::assign(char*s)
{
n=0;
char b[256];
strcpy(b,s);
char*t=strtok(b,"+");
int i,j;
while(t!=NULL)
{
a[n][0]=isalpha(t[0])?tolower(t[0]):0;
a[n][2]=0;
if(isalpha(t[0])) j=1;else j=0;
for(i=j;i<strlen(t);i++)
if(!isdigit(t[i]))
{
if(t[i]==10||t[i]==32) t[i]=0;
a[n][2]=tolower(t[i]);
break;
}
a[n][1]=isalpha(t[0])?atoi(t+1):atoi(t);
n++;
t=strtok(NULL,"+");
}
}
void FACS::copyFrom(FACS&f)
{
n=f.n;
for(int i=0;i<n;i++)
{
a[i][0]=f.a[i][0];
a[i][1]=f.a[i][1];
a[i][2]=f.a[i][2];
}
}
void FACS::show(void)
{
if(!n) return;
for(int i=0;i<n;i++)
{
if(a[i][0]) printf("%c",a[i][0]);
printf("%i ",a[i][1]);
if(a[i][2]) printf("%c ",a[i][2]);
}
printf("\n");
}
bool FACS::Is(int AU,int exist)
{
int sw=0;
for(int i=0;i<n;i++)
if(a[i][1]==AU)
{
sw=1;
break;
}
return (sw==exist);
}
bool FACS::containAU(int au)
{
for(int i=0;i<n;i++)
if(a[i][1]==au) return true;
return false;
}
//-----------------------------------
```

The routines for handling the model sample data.

```
//-----------------------------------
...............
#include "model.h"
bool query::add(char*e,bool sw_n)
{
int i,sw=0;
for(i=0;i<strlen(e);i++) if(isdigit(e[i])) sw=1;
if(sw) // facs
{
if(n==MAXQUERYLEN) return false;
f[n].assign(e);
neg[n]=!sw_n;
n++;
}
else
{
swexpr=true;
strcpy(expr,e);
neg_expr=!sw_n;
}
return true;
}
bool model::readDatabase(char*name)
{
FILE *f=fopen(name,"rt");
if(f==NULL) return false;
fscanf(f,"%i %i %i",&n,&NP,&NC);
char b[1024],t[512];
int i,k;
for(n=0;;)
{
fscanf(f,"%s",t);
if(feof(f)) break;
if(t[0]=='-') continue;
a[n].nsubj=atoi(t);
fscanf(f,"%s",t);
a[n].nscene=atoi(t);
fscanf(f,"%s",t);
a[n].facs.assign(t);
fscanf(f,"%s",t);
strcpy(a[n].expr,t);
for(i=0;i<NP;i++)
{
fscanf(f,"%s",t);
a[n].param[i]=atof(t);
}
//printf("%i %i %s > ",a[n].nsubj,a[n].nscene,a[n].expr);
//for(int j=0;j<NP;j++) printf("%.f ",a[n].param[j]);printf("\n");
n++;
}
fclose(f);
return true;
}
void model::show(void)
{
int i,j;
for(i=0;i<n;i++)
{
printf("%i %i %s | ",a[i].nsubj,a[i].nscene,a[i].expr);
for(j=0;j<NP;j++) printf("%.f ",a[i].param[j]);printf(" facs: ");
a[i].facs.show();
}
}
float model::computeP(cere&c)
{
/*int nt,nk;
int i,j;
bool sw;
nk=0;
nt=0;
for(i=0;i<n;i++)
{
sw=false;
for(j=0;j<a[i].facs.n;j++) if(c.AU==a[i].facs.a[j][1]) sw=true;
sw=(sw==c.swAU);
if(sw)
{
nt++;
for(j=0;j<c.n;j++) if(a[i].param[c.P[j][0]]!=c.P[j][1]) sw=false;
}
if(sw) nk++;
}
return nt?1.*nk/nt:0;*/
int i,j,nt,nk;
bool sw;
nt=nk=0;
for(i=0;i<n;i++)
{
sw=true;
for(j=0;j<c.n;j++) if(a[i].param[c.P[j][0]-1]!=c.P[j][1]) sw=false;
if(sw)
{
nt++;
sw=false;
for(j=0;j<a[i].facs.n;j++) if(c.AU==a[i].facs.a[j][1]) sw=true;
//sw=(sw==c.swAU);
if(sw) nk++;
}
}
return nt?1.*nk/nt:0;
}
float model::computeP2(cere2&c)
{
int i,j,nt=0,nk=0;
bool sw;
for(i=0;i<n;i++)
{
sw=true;
for(j=0;j<c.n;j++) if(a[i].facs.containAU(c.AU[j][0])!=c.AU[j][1]) {sw=false;break;}
if(sw)
{
nt++;
if(a[i].param[c.P[0]]==c.P[1]) nk++;
}
}
return nt?1.*nk/nt:0;
}
float model::computeP3(int AU,char*s)
{
int i,j,nt=0,nk=0;
for(i=0;i<n;i++)
{
if(a[i].facs.containAU(AU))
{
nt++;
if(s[1]==a[i].expr[0] && s[2]==a[i].expr[1]) nk++;
}
}
return nt?1.*nk/nt:0;
}
float model::computeP4(int AU,char*s,bool sw)
{
int i,j,nt=0,nk=0;
for(i=0;i<n;i++)
{
if((strcmp(a[i].expr,s)==0&&sw)||(strcmp(a[i].expr,s)!=0&&!sw))
{
nt++;
if(a[i].facs.containAU(AU)) nk++;
}
}
return nt?1.*nk/nt:0;
}
bool model::isIncluded(FACS&l1,FACS&l2)
{
return (compare(l1,l2)==l1.n);
}
float model::compare(FACS&l1,FACS&l2)
{
int i,j;
float k=0;
for(i=0;i<l1.n;i++)
for(j=0;j<l2.n;j++)
if(l1.a[i][1]==l2.a[j][1]) k+=1.;
return k;
}
int model::countExpression(char*s)
{
int k=0;
for(int i=0;i<n;i++)
if(strcmp(s,a[i].expr)==0) k++;
return k;
}
void model::insertField(int nsubj,int nsce,FACS&f,char*exp)
{
a[n].nsubj=nsubj;
a[n].nscene=nsce;
strcpy(a[n].expr,exp);
a[n].facs.copyFrom(f);
n++;
}
void model::select(query&q,model&d)
{
d.emptyDatabase();
int i,j,k;
bool sw;
for(i=0;i<n;i++)
{
if(q.swexpr)
{
k=strcmp(a[i].expr,q.expr)==0;
if((k && q.neg_expr)||(!k && !q.neg_expr)) continue;
}
sw=true;
for(j=0;j<q.n;j++)
{
k=isIncluded(q.f[j],a[i].facs);
if((k && q.neg[j])||(!k && !q.neg[j])) sw=false;
}
if(sw)
d.insertField(a[i].nsubj,a[i].nscene,a[i].facs,a[i].expr);
}
}
int model::getClass(int i,int j)
{
return a[i].param[j];
}
void model::save(char*name)
```

```
{
FILE*f=fopen(name,"wt");
int i,j;
for(i=0;i<n;i++)
{
fprintf(f,"%i %i %s\t",a[i].nsubj,a[i].nscene,a[i].expr);
for(j=0;j<a[i].facs.n;j++) fprintf(f,"%i ",a[i].facs.a[j][1]);fprintf(f,"\n");
}
fclose(f);
}
char* model::getFieldExpression(int k)
{
return a[k].expr;
}
```

//-----------------------------------

The routines for ANN experiments.

//-----------------------------------

```
..........................
#define LEARNING_RATE .02
#define NR_LEARN 30000
#define NI 30
#define NH 12
#define NO 3
#define D .5
.................................
nn::nn(model&md,int n1,int n2,int n3):m(md),ni(n1),nh(n2),no(n3)
{
}
float nn::f(float x)
{
return (float)((1.0f/(1.0f+exp(-x)))-D);
}
float nn::df(float x)
{
double z=f(x)+D;
return (float)(z*(1.0f-z));
}
void nn::pass()
{
int k,l;
for(k=0;k<nh;k++)
{
h[k]=0;
for(l=0;l<ni;l++) h[k]+=i[l]*w1[l][k];
}
for(k=0;k<no;k++)
{
o[k]=0;
for(l=0;l<nh;l++) o[k]+=f(h[l])*w2[l][k];
o[k]=f(o[k]);
}
}
float nn::trainSample(int ks)
{
int k,l;
/*for(k=0;k<nh;k++) for(l=0;l<no;l++) printf("%f ",w2[k][l]);printf("\n");
for(k=0;k<ni;k++) for(l=0;l<nh;l++) printf("%f ",w1[k][l]);printf("\n");
printf("\n");
getch();*/
for(k=0;k<nh;k++) eh[k]=0;
for(k=0;k<no;k++) eo[k]=0;
for(k=0;k<ni;k++) i[k]=(float)m.l[ks].i[k]-D;
pass();
for(k=0;k<no;k++)
eo[k]=((float)m.l[ks].o[k]-Do[k])*df(o[k]);
for(k=0;k<nh;k++)
{
eh[k]=0;
for(l=0;l<no;l++) eh[k]+=eo[l]*w2[k][l];
eh[k]*=df(h[k]);
}
for(k=0;k<nh;k++) for(l=0;l<no;l++) w2[k][l]+=LEARNING_RATE*eo[l]*h[k];
for(k=0;k<ni;k++) for(l=0;l<nh;l++) w1[k][l]+=LEARNING_RATE*eh[l]*i[k];
float err=0;
for(k=0;k<no;k++) err+=fabs(eo[k]);
return err;
}
void nn::randomWeights(void)
{
int k,l;
for(k=0;k<ni;k++) for(l=0;l<nh;l++) w1[k][l]=0.00001*(float)rand();
for(k=0;k<nh;k++) for(l=0;l<no;l++) w2[k][l]=0.00001*(float)rand();
}
void nn::train(void)
{
float err;
int i=0,k;
randomWeights();
FILE*f=fopen("err.","wt");
do
{
err=0;
for(k=0;k<m.n;k++) err+=trainSample(k);
if(i%100) fprintf(f,"%f\n",err);
printf("[%i] err=%f\n",i+1,err);
i++;
}while(i<NR_LEARN);
fclose(f);
}
void nn::test(void)
{
int l,j,k,n=0;
char t[5];
bool sw;
float h;
int r[6][6];
memset(r,0,36*sizeof(int));
for(l=0;l<m.n;l++)
{
for(k=0;k<ni;k++) i[k]=(float)m.l[l].i[k]-D;
pass();
for(k=0;k<no;k++) printf("%i",m.l[l].o[k]);printf("\n");
j=0;
for(k=0;k<no;k++)
{
printf("%f ",o[k]+D);
h=o[k]+D>.5?1:0;
j+=h*pow(2,no-1-k);
}
r[m.l[l].nexp-1][j-1]++;
if(j==m.l[l].nexp) {printf("*\n");n++;} else printf("\n");
}
for(j=0;j<6;j++)
{
for(k=0;k<6;k++) printf("%4i",r[j][k]);
printf("\n");
}
printf("Recognition rate %.2f %%\n",100.*n/m.n);
}
void nn::save(char*s)
{
int k,l;
FILE*f;
f=fopen(s,"wt");
fprintf(f,"%i %i %i\n",ni,nh,no);
for(k=0;k<ni;k++) for(l=0;l<nh;l++) fprintf(f,"%f ",w1[k][l]);
for(k=0;k<nh;k++) for(l=0;l<no;l++) fprintf(f,"%f ",w2[k][l]);
fclose(f);
}
void nn::load(char*s)
{
int k,l;
FILE*f;
f=fopen(s,"rt");
fscanf(f,"%i %i %i\n",&ni,&nh,&no);
for(k=0;k<ni;k++) for(l=0;l<nh;l++) fscanf(f,"%f",&w1[k][l]);
for(k=0;k<nh;k++) for(l=0;l<no;l++) fscanf(f,"%f",&w2[k][l]);
fclose(f);
}
```

//-----------------------------------

The routines for BNN experiments.

//-----------------------------------

```
bool model::read(char*s)
{
FILE*f=fopen(s,"rt");
if(f==NULL)
{
printf("Input file [%s] not found!\n");
return false;
}
int i,j,NP=10;char b[256];
fgets(b,256,f);
nl=-1;
while(!feof(f))
{
nl++;
for(j=0;j<NP;j++) fscanf(f,"%i",&(l[nl].param[j]));
fscanf(f,"%s",l[nl].exp);
}
fclose(f);
}
int getIndex(char*s)
{
if(strcmp(s,"Surprise")==0) return 1;
if(strcmp(s,"Sadness")==0) return 2;
if(strcmp(s,"Anger")==0) return 3;
if(strcmp(s,"Happy")==0) return 4;
if(strcmp(s,"Disgust")==0) return 5;
if(strcmp(s,"Fear")==0) return 6;
}
char*getExp(int k)
{
switch(k)
{
case 1: return "Surprise";
case 2: return "Sadness";
case 3: return "Anger";
case 4: return "Happy";
case 5: return "Disgust";
case 6: return "Fear";
}
}
//------------------------------------------------------------
void model::set_Param(int k)
{
int i,j,t,x;
char b[20],bt[20];
for(i=0;i<10;i++)
{
strcpy(b,"P");
itoa(i+1,bt,10);
strcat(b,bt);
int id=net.FindNode(b);
t=l[k].param[i];
net.GetNode(id)->Value()->ClearEvidence();
net.GetNode(id)->Value()->SetEvidence(t-1);
}
}
int convert(int k)
{
switch(k)
{
case 0:return 2;
case 1:return 4;
case 2:return 5;
case 3:return 3;
case 4:return 1;
case 5:return 0;
};
}
int model::testOne(int k)
{
set_Param(k);
net.UpdateBeliefs();
int i,m,y;
double r[10];
int id=net.FindNode("Exp");
for(i=0;i<6;i++)
{
DSL_sysCoordinates c(*net.GetNode(id)->Value());
c[0]=i;
c.GoToCurrentPosition();
r[i]=c.UncheckedValue();
}
m=0;
for(i=0;i<6;i++) if(r[m]<r[i]) m=i;
return convert(m);
}
void model::test(void)
{
float r[6][6];
int i,j,k;
for(i=0;i<6;i++)
for(j=0;j<6;j++)
r[i][j]=0;
for(i=0;i<nl;i++)
{
j=getIndex(l[i].exp)-1;
k=testOne(i);
r[j][k]++;
}
//---------------------------------
FILE*f=fopen("rez","wt");
for(i=0;i<6;i++)
{
fprintf(f,"%15s\t",getExp(i+1));
for(j=0;j<6;j++) fprintf(f,"%2i ",(int)r[i][j]);
fprintf(f,"\n");
}
fclose(f);
//---------------------------------
float t;
for(i=0;i<6;i++)
{
t=0;
printf("%15s\t",getExp(i+1));
for(j=0;j<6;j++) {t+=r[i][j];printf("%2i ",(int)r[i][j]);}
printf(" rec.=%.2f%%\n",100*r[i][i]/t);
}
t=0;
for(i=0;i<6;i++) t+=r[i][i];
printf("\nrec. rate=%.2f%%\n",100*t/nl);
}
```

APPENDIX B

Datcu D., Rothkrantz L.J.M., ‘Automatic recognition of facial expressions using Bayesian Belief Networks’, *Proceedings of IEEE SMC 2004*, ISBN 0-7803-8567-5, pp. 2209-2214, October 2004.

Automatic recognition of facial expressions using Bayesian Belief Networks*

D. Datcu Department of Information Technology and Systems T.U.Delft, The Netherlands D.Datcu@ewi.tudelft.nl	**L.J.M. Rothkrantz** Department of Information Technology and Systems T.U.Delft, The Netherlands L.J.M.Rothkrantz@ewi.tudelft.nl

Abstract - *The current paper addresses the aspects related to the development of an automatic probabilistic recognition system for facial expressions in video streams.*

The face analysis component integrates an eye tracking mechanism based on Kalman filter. The visual feature detection includes PCA oriented recognition for ranking the activity in certain facial areas. The description of the facial expressions is given according to sets of atomic Action Units (AU) from the Facial Action Coding System (FACS). The base for the expression recognition engine is supported through a BBN model that also handles the time behavior of the visual features.

Keywords: Facial expression recognition, tracking, pattern recognition.

1. Introduction

The study of human facial expressions is one of the most challenging domains in pattern research community.

Each facial expression is generated by non-rigid object deformations and these deformations are person dependent.

The goal of our project was to design and implement a system for automatic recognition of human facial expression in video streams. The results of the project are of a great importance for a broad area of applications that relate to both research and applied topics.

As possible approaches on those topics, the following may be presented: automatic surveillance systems, the classification and retrieval of image and video databases, customer-friendly interfaces, smart environment human computer interaction and research in the field of computer assisted human emotion analyses. Some interesting implementations in the field of computed assisted emotion analysis concern experimental and interdisciplinary psychiatry. Automatic recognition of facial expressions is a process primarily based on analysis of permanent and transient features of the face, which can be only assessed with errors of some degree. The expression recognition model is oriented on the specification of Facial Action Coding System (FACS) of Ekman and Friesen [6]. The hard constraints on the scene processing and recording conditions set a limited robustness to the analysis. In order to manage the uncertainties and lack of information, we set a probabilistic oriented framework up. The support for the specific processing involved was given through a multimodal data fusion platform. In the Department of Knowledge Based Systems at T.U.Delft there has been a project based on a long-term research running on the development of a software workbench. It is called Artificial Intelligence aided Digital Processing Toolkit (A.I.D.P.T.) [4] and presents native capabilities for real time signal and information processing and for fusion of data acquired from hardware equipments. The workbench also includes support for the Kalman filter based mechanism used for tracking the location of the eyes in the scene. The knowledge of the system relied on the data taken from the Cohn-Kanade AU-Coded Facial Expression Database [8]. Some processing was done so as to extract the useful information. More than that, since the original database contained only one image having the AU code set for each display, additional coding had to be done. The Bayesian network is used to encode the dependencies among the variables. The temporal dependencies were extracted to make the system be able to properly select the right emotional expression. In this way, the system is able to overcome the performance of the previous approaches that dealt only with prototypic facial expression [10]. The causal relationships track the changes occurred in each

* 0-7803-8566-7/04/$20.00 © 2004 IEEE.

facial feature and store the information regarding the variability of the data.

2. Related work

The typical problems of expression recognition have been tackled many times through distinct methods in the past. In [12] the authors proposed a combination of a Bayesian probabilistic model and Gabor filter. [3] introduced a Tree-Augmented-Naive Bayes (TAN) classifier for learning the feature dependencies. A common approach was based on neural networks. [7] used a neural network approach for developing an online Facial Expression Dictionary as a first step in the creation of an online Nonverbal Dictionary. [2] used a subset of Gabor filters selected with Adaboost and trained the Support Vector Machines on the outputs.

3. Eye tracking

The architecture of the facial expression recognition system integrates two major components. In the case of the analysis applied on video streams, a first module is set to determine the position of the person eyes. Given the position of the eyes, the next step is to recover the position of the other visual features as the presence of some wrinkles, furrows and the position of the mouth and eyebrows. The information related to the position of the eyes is used to constrain the mathematical model for the point detection. The second module receives the coordinates of the visual features and uses them to apply recognition of facial expressions according to the given emotional classes. The detection of the eyes in the image sequence is accomplished by using a tracking mechanism based on Kalman filter [1]. The eye-tracking module includes some routines for detecting the position of the edge between the pupil and the iris. The process is based on the characteristic of the dark-bright pupil effect in infrared condition (see Figure 1).

Figure 1. The dark-bright pupil effect in infrared

However, the eye position locator may not perform well in some contexts as poor illuminated scene or the rotation of the head. The same might happen when the person wears glasses or has the eyes closed. The inconvenience is managed by computing the most probable eye position with Kalman filter. The estimation for the current frame takes into account the information related to the motion of the eyes in the previous frames. The Kalman filter relies on the decomposition of the pursuit eye motion into a deterministic component and a random component. The random component models the estimation error in the time sequence and further corrects the position of the eye.

It has a random amplitude, occurrence and duration. The deterministic component concerns the motion parameters related to the position, velocity and acceleration of the eyes in the sequence. The acceleration of the motion is modeled as a Gauss-Markov process. The autocorrelation function is as presented in formula (1):

$$\mathrm{R(t)} = \sigma^2 \mathrm{e}^{-\mathrm{b}|\mathrm{t}|} \quad (1)$$

The equations of the eye movement are defined according to the formula (2). In the model we use, the state vector contains an additional state variable according to the Gauss-Markov process. $u(t)$ is a unity Gaussian white noise.

$$\begin{matrix} \dot{\mathrm{x}}_1 \\ \dot{\mathrm{x}}_2 \\ \dot{\mathrm{x}}_3 \end{matrix} = \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta \end{matrix} \begin{matrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{matrix} + \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} u(t)$$

$$z = \left[\sqrt{2\sigma^2\beta} \quad 0 \quad 0\right] \begin{matrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{matrix} \quad (2)$$

The discrete form of the model for tracking the eyes in the sequence is given in formula (3). $\phi = \mathrm{e}^{\mathrm{f}\Delta\mathrm{t}}$, w are the process Gaussian white noise and n is the measurement Gaussian white noise.

$$\begin{aligned} x_k &= \phi_k + w_k \\ z_k &= H_k \cdot x_k + v_k \end{aligned} \quad (3)$$

The Kalman filter method used for tracking the eyes presents a high efficiency by reducing the error of the coordinate estimation task. In addition to that, the process does not require a high processor load and a real time implementation was possible.

4 Face representational model

The facial expression recognition system handles the input video stream and performs analysis on the existent frontal face. In addition to the set of degree values related to the detected expressions, the system can also output a graphical face model.

The result may be seen as a feedback of the system to the given facial expression of the person whose face is analyzed and it may be different of that. One direct application of the chosen architecture may be in design of systems that perceive and interact with humans by using natural communication channels.

In our approach the result is directly associated to the expression of the input face (see Figure 2). Given the parameters from the expression recognition module, the system computes the shape of different visual features and generates a 2D graphical face model.

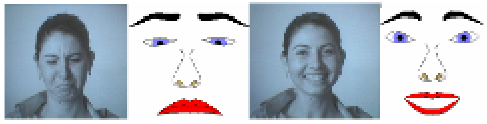


Figure 2. Response of the expression recognition

The geometrical shape of each visual feature follows certain rules that aim to set the outlook to convey the appropriate emotional meaning. Each feature is reconstructed using circles and simple polynomial functions as lines, parabola parts and cubic functions. A five-pixel window is used to smooth peaks so as to provide shapes with a more realistic appearance.

The eye upper and lower lid was approximated with the same cubic function. The eyebrow's thickness above and below the middle line was calculated from three segments as a parabola, a straight line and a quarter of a circle as the inner corner. A thickness function was added and subtracted to and from the middle line of the eyebrow. The shape of the mouth varies strongly as emotion changes from sadness to happiness or disgust.

The manipulation of the face for setting a certain expression implies to mix different emotions. Each emotion has a percentage value by which they contribute to the face general expression.

The new control set values for the visual features are computed by the difference of each emotion control set and the neutral face control set, and make a linear combination of the resulting six vectors.

5 Visual feature acquisition

The objective of the first processing component of the system is to recover the position of some key points on the face surface. The process starts with the stage of eye coordinate detection. Certified FACS coders coded the image data. Starting from the image database, we processed each image and obtained the set of 30 points according to Kobayashi & Hara model [9]. The analysis was semi-automatic.

A new transformation was involved then to get the key points as described in figure 3. The coordinates of the last set of points were used for computing the values of the parameters presented in table 2. The preprocessing tasks implied some additional requirements to be satisfied. First, for each image a new coordinate system was set. The origin of the new coordinate system was set to the nose top of the individual.

The value of a new parameter called *base* was computed to measure the distance between the eyes of the person in the image. The next processing was the rotation of all the points in the image with respect to the center of the new coordinate system. The result was the frontal face with correction to the facial inclination. The final step of preprocessing was related to scale all the distances so as to be invariant to the size of the image.

Eventually a set of 15 values for each of the image was obtained as the result of preprocessing stage. The parameters were computed by taking both the variance observed in the frame at the time of analysis and the temporal variance. Each of the last three parameters was quantified so as to express a linear behavior with respect to the range of facial expressions analyzed.

The technique used was Principal Component Analysis oriented pattern recognition for each of the three facial areas. The technique was first applied by Turk and Pentland for face imaging [11]. The PCA processing is run separately for each area and three sets of eigenvectors are

available as part of the knowledge of the system. Moreover, the labeled patterns associated with each area are stored (see Figure 4).
The computation of the eigenvectors was done offline as a preliminary step of the process. For each input image, the first processing stage extracts the image data according to the three areas. Each data image is projected through the eigenvectors and the pattern with the minimum error is searched.

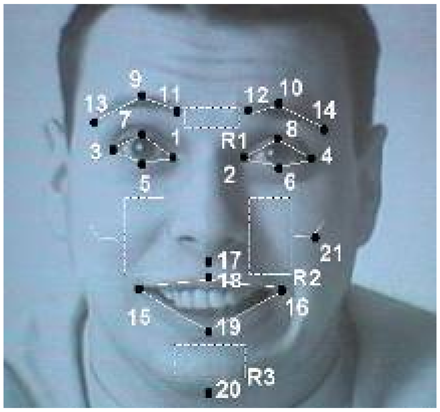


Figure 3. The model facial key points and areas

The label of the extracted pattern is then fed to the quantification function for obtaining the characteristic output value of each image area. Each value is further set as evidence in the probabilistic BBN.

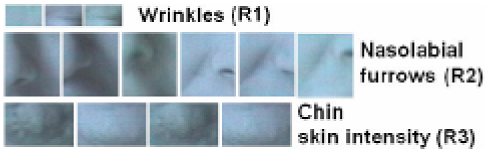


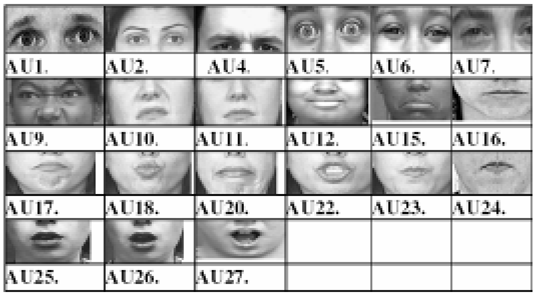
Figure 4. Examples of patterns used in PCA recognition

6 Data preparation

The Bayesian Belief Network encodes the knowledge of the existent phenomena that triggers changes in the aspect of the face. The model does include several layers for the detection of distinct aspects of the transformation. The lowest level is that of primary parameter layer. It contains a set of parameters that keeps track of the changes concerning the facial key points. Those parameters may be classified as static and dynamic. The static parameters handle the local geometry of the current frame. The dynamic parameters encode the behavior of the key points in the transition from one frame to another. By combining the two sorts of information, the system gets a high efficiency of expression recognition. An alternative is that the base used for computing the variation of the dynamic parameters is determined as a previous tendency over a limited past time. Each parameter on the lowest layer of the BBN has a given number of states. The purpose of the states is to map any continuous value of the parameter to a discrete class. The number of states has a direct influence on the efficiency of recognition. The number of states for the low-level parameters does not influence the time required for obtaining the final results. It is still possible to have a real time implementation even when the number of states is high.

The only additional time is that of processing done for computing the conditioned probability tables for each BBN parameter, but the task is run off-line. According to the method used, each facial expression is described as a combination of existent Action Units (AU).

Table 1. The used set of Action Units

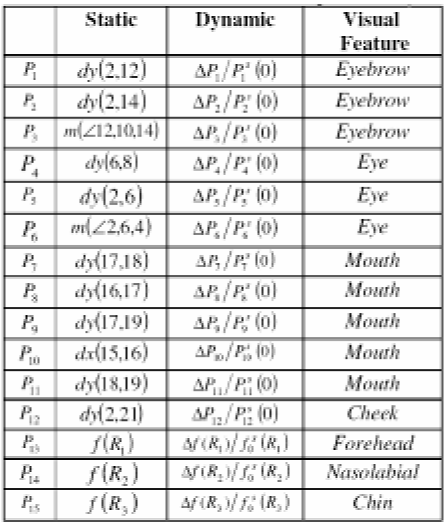


One AU represents a specific facial display. Among 44 AUs contained in FACS, 12 describe contractions of specific facial muscles in the upper part of the face and 18 in the lower part. The table 1 presents the set of AUs that is managed by the current recognition system.

An important characteristic of the AUs is that they may act differently in given combinations. According to the behavioral side of each AU, there are additive and non-additive combinations. In that way, the result of one non-additive combination may be related to a facial expression that is not expressed by the constituent AUs taken separately.

In the case of the current project, the AU sets related to each expression are split into two classes that specify the importance of the emotional load of each AU in the class. By means of that, there are primary and secondary AUs.

Table 2. The set of visual feature parameters



	Static	Dynamic	Visual Feature
P_1	$dy(2,12)$	$\Delta P_1 / P_1^s(0)$	*Eyebrow*
P_2	$dy(2,14)$	$\Delta P_2 / P_2^s(0)$	*Eyebrow*
P_3	$m(\angle 12,10,14)$	$\Delta P_3 / P_3^s(0)$	*Eyebrow*
P_4	$dy(6,8)$	$\Delta P_4 / P_4^s(0)$	*Eye*
P_5	$dy(2,6)$	$\Delta P_5 / P_5^s(0)$	*Eye*
P_6	$m(\angle 2,6,4)$	$\Delta P_6 / P_6^s(0)$	*Eye*
P_7	$dy(17,18)$	$\Delta P_7 / P_7^s(0)$	*Mouth*
P_8	$dy(16,17)$	$\Delta P_8 / P_8^s(0)$	*Mouth*
P_9	$dy(17,19)$	$\Delta P_9 / P_9^s(0)$	*Mouth*
P_{10}	$dx(15,16)$	$\Delta P_{10} / P_{10}^s(0)$	*Mouth*
P_{11}	$dy(18,19)$	$\Delta P_{11} / P_{11}^s(0)$	*Mouth*
P_{12}	$dy(2,21)$	$\Delta P_{12} / P_{12}^s(0)$	*Cheek*
P_{13}	$f(R_1)$	$\Delta f(R_1) / f_0^s(R_1)$	*Forehead*
P_{14}	$f(R_2)$	$\Delta f(R_2) / f_0^s(R_2)$	*Nasolabial*
P_{15}	$f(R_3)$	$\Delta f(R_3) / f_0^s(R_3)$	*Chin*

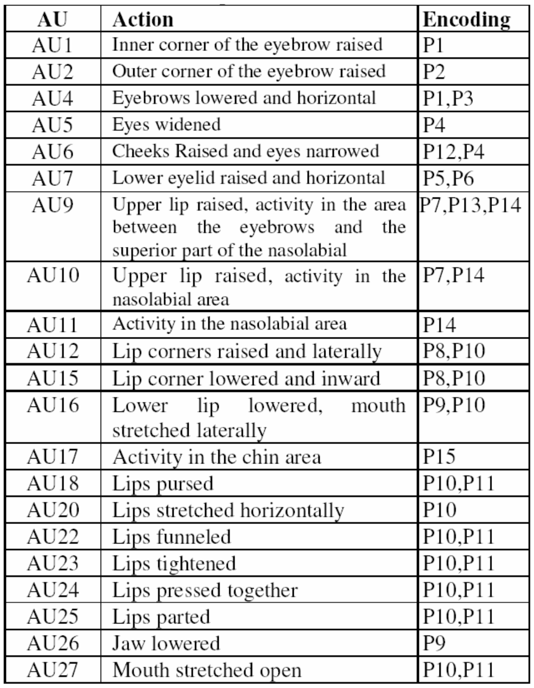
The AUs being part of the same class are additive. The system performs recognition of one expression as computing the probability associated with the detection of one or more AUs from both classes.

The probability of one expression increases, as the probabilities of detected primary AUs get higher. In the same way, the presence of some AUs from a secondary class results in solving the uncertainty problem in the case of the dependent expression but at a lower level.

The conditioned probability tables for each node of the Bayesian Belief Network were filled in by computing statistics over the database. The Cohn-Kanade AU-Coded Facial Expression Database contains approximately 2000 image sequences from 200 subjects ranged in age from 18 to 30 years. Sixty-five percent were female, 15 percent were African-American and three percent were Asian or Latino. All the images analyzed were frontal face pictures.

The original database contained sequences of the subjects performing 23 facial displays including single action units and combinations. Six of the displays were based on prototypic emotions (joy, surprise, anger, fear, disgust and sadness).

Table 3. The dependency between AUs and intermediate parameters



AU	Action	Encoding
AU1	Inner corner of the eyebrow raised	P1
AU2	Outer corner of the eyebrow raised	P2
AU4	Eyebrows lowered and horizontal	P1,P3
AU5	Eyes widened	P4
AU6	Cheeks Raised and eyes narrowed	P12,P4
AU7	Lower eyelid raised and horizontal	P5,P6
AU9	Upper lip raised, activity in the area between the eyebrows and the superior part of the nasolabial	P7,P13,P14
AU10	Upper lip raised, activity in the nasolabial area	P7,P14
AU11	Activity in the nasolabial area	P14
AU12	Lip corners raised and laterally	P8,P10
AU15	Lip corner lowered and inward	P8,P10
AU16	Lower lip lowered, mouth stretched laterally	P9,P10
AU17	Activity in the chin area	P15
AU18	Lips pursed	P10,P11
AU20	Lips stretched horizontally	P10
AU22	Lips funneled	P10,P11
AU23	Lips tightened	P10,P11
AU24	Lips pressed together	P10,P11
AU25	Lips parted	P10,P11
AU26	Jaw lowered	P9
AU27	Mouth stretched open	P10,P11

7 Inference with BBN

The expression recognition is done computing the anterior probabilities for the parameters in the BBN (see Figure 5). The procedure starts by setting the probabilities of the parameters on the lowest level according to the values computed at the preprocessing stage. In the case of each parameter, evidence is given for both static and dynamic parameters. Moreover, the evidence is set also for the parameter related to the probability of the anterior facial expression. It contains 6 states, one for each major class of expressions. The aim of the presence of the anterior expression node and that associated with the dynamic component of one given low-level parameter, is to augment the inference process with temporal constrains.

The structure of the network integrates parametric layers having different functional tasks. The goal of the layer containing the first AU set and that of the low-level parameters is to detect the presence of some AUs in the current frame. The relation between the set of the low-level parameters and the action units is as it is detailed in table 4. The dependency of the parameters on AUs was determined on the criteria of influence observed on the initial database. The presence of one AU at this stage does not imply the existence of one facial expression or another.

Instead, the goal of the next layer containing the AU nodes and associated dependencies is to determine the probability that one AU presents influence on a given kind of emotion. The final parametric layer consists of nodes for every emotional class. More than that, there is also one node for the current expression and another one for that previously detected. The top node in the network is that of current expression. It has two states according to the presence and absence of any expression and stands for the final result of analysis. The absence of any expression is seen as a neutral display of the person's face on the current frame. While performing recognition, the BBN probabilities are updated in a bottom-up manner. As soon as the inference is finished and expressions are detected, the system reads the existence probabilities of all the dependent expression nodes. The most probable expression is that given by the larger value over the expression probability set.

Table 4. The emotion projections of each AU combination

	Primary AUs						Secondary Aus			
Surprise	1+2	5	25	26	27					
Fear	1+2	4	5	7	20	25	6	10	11	
Happy	6+12	12					7	16	25	26
Sadness	15+17	4+17	1				7	24	20+25	
Disgust	4	7+9	17	6+7			23	24	25	
Anger	4	4+7	23	24			17			

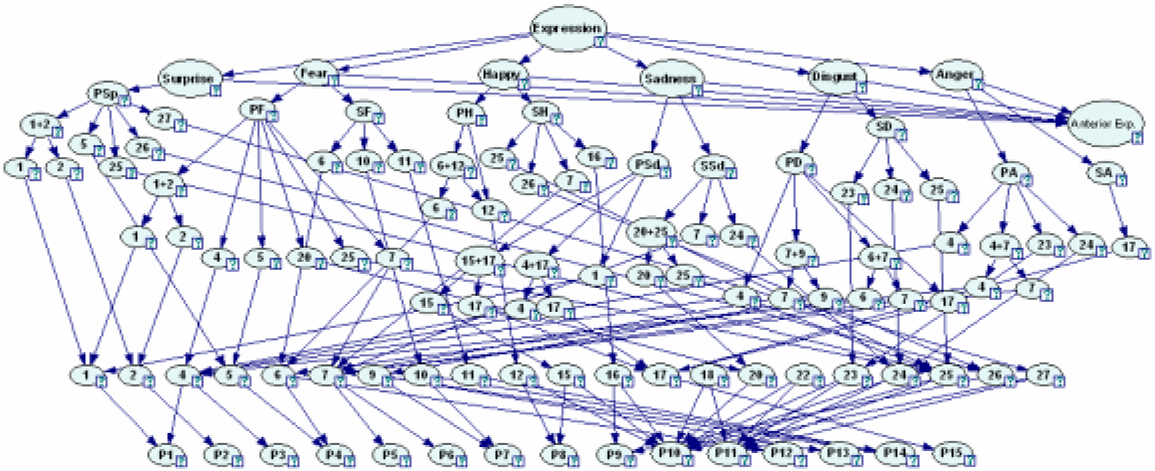


Figure 5. BBN used for facial expression recognition

8 Results

The implementation of the model was made using C/C++ programming language. The system consists in a set of applications that run different tasks that range from pixel/image oriented processing to statistics building and inference by updating the probabilities in the BBN model.

The support for BBN was based on S.M.I.L.E. (Structural Modeling, Inference, and Learning Engine), a platform independent library of C++ classes for reasoning in probabilistic models [5]. S.M.I.L.E. is freely available to the community and has been developed at the Decision Systems Laboratory, University of Pittsburgh. The library was included in the AIDPT framework. The implemented probabilistic model is able to perform recognition on six emotional classes and the neutral state. By adding new parameters on the facial expression layer, the expression number on recognition can be easily increased.

Accordingly, new AU dependencies have to be specified for each of the emotional class added. In figure 7 there is an example of an input video sequence. The recognition result is given in the graphic containing the information related to the probability of the dominant facial expression (see Figure 6).

9 Conclusion

In the current paper we've described the development steps of an automatic system for facial expression recognition in video sequences. The inference mechanism was based on a probabilistic framework. We used the Cohn-

Kanade AU-Coded Facial Expression Database for building the system knowledge. It contains a large sample of varying age, sex and ethnic background and so the robustness to the individual changes in facial features and behavior is high. The BBN model takes care of the variation and degree of uncertainty and gives us an improvement in the quality of recognition. As off now, the results are very promising and show that the new approach presents high efficiency. An important contribution is related to the tracking of the temporal behavior of the analyzed parameters and the temporal expression constrains.

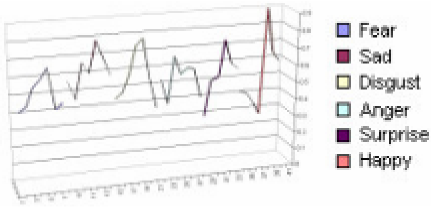


Figure 6. Dominant Emotional expression in sequence

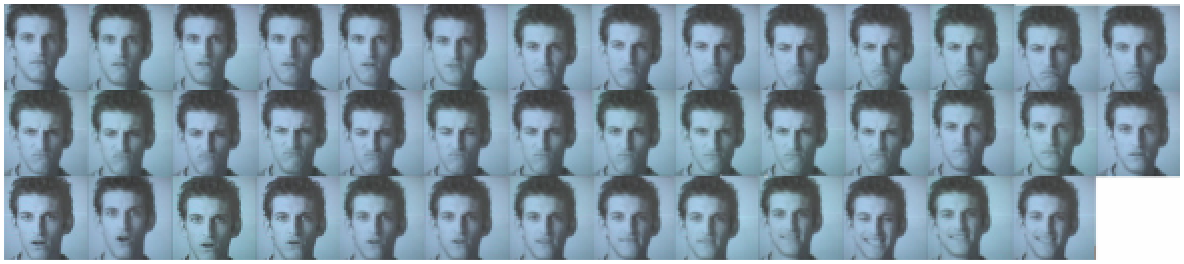


Figure 7. Example of facial expression recognition applied on video streams

References

[1] W. A-Almageed, M. S. Fadali, G. Bebis ‘A nonintrusive Kalman Filter-Based Tracker for Pursuit Eye Movement’ Proceedings of the 2002 American Control Conference Alaska, 2002

[2] M. S. Bartlett, G. Littlewort, I. Fasel, J. R. Movellan ‘Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction’ IEEE Workshop on Face Processing in Video, Washington 2004

[3] I. Cohen, N. Sebe, A. Garg, M. S.Lew, T. S. Huang ‘Facial expression recognition from video sequences’ Computer Vision and Image Understanding, Volume 91, pp 160 - 187 ISSN: 1077-3142 2003

[4] D. Datcu, L. J. M. Rothkrantz ‘A multimodal workbench for automatic surveillance’ Euromedia Int’l Conference 2004

[5] M. J. Druzdzel ‘GeNIe: A development environment for graphical decision-analytic models’. In Proceedings of the 1999 Annual Symposium of the American Medical Informatics Association (AMIA-1999), page 1206, Washington, D.C., November 6-10, 1999

[6] P. Ekman, W. V. Friesen ‘Facial Action Coding System: Investigator’s Guide’ Consulting Psychologists Press, 1978

[7] E. J. de Jongh, L .J. M. Rothkrantz ‘FED – an online Facial Expression Dictionary’ Euromedia Int’l Conference 2004

[8] T. Kanade, J. Cohn, Y. Tian ‘Comprehensive database for facial expression analysis’ Proc. IEEE Int’l Conf. Face and Gesture Recognition, pp. 46-53, 2000

[9] H. Kobayashi and F. Hara. ‘Recognition of Mixed Facial Expressions by Neural Network’ IEEE International workshop on Robot and Human Communication, 381-386, 1972

[10] M. Pantic, L. J. M. Rothkrantz ‘Toward an Affect- Sensitive Multimodal Human-Computer Interaction’ IEEE proceedings vol. 91, no. 9, pp. 1370-1390, 2003

[11] M. Turk, A. Pentland 'Face recognition using eigenfaces, Proc. CVPR, pp. 586-591 (1991)

[12] X. Wang, X. Tang 'Bayesian Face Recognition Using Gabor Features' Proceedings of the 2003 ACM SIGMM Berkley, California 2003