

# MULTICLASS SUPPORT VECTOR MACHINES IN PSEUDO-EUCLIDEAN SPACES FOR FACIAL EXPRESSION RECOGNITION

Irene Kotsia<sup>†</sup>, Stefanos Zafeiriou<sup>†</sup>, Nikolaos Nikolaidis<sup>†</sup> and Ioannis Pitas<sup>†</sup>

<sup>†</sup>Aristotle University of Thessaloniki, Department of Informatics  
Thessaloniki, Greece  
email: {ekotsia, dralbert, nikolaid, pitas}@aiia.csd.auth.gr

## ABSTRACT

In this paper, a novel method for the recognition of facial expressions in videos is proposed. The system firsts extracts the deformed Candide facial grid that corresponds to the facial expression depicted in the video sequence. The Hausdorff distance of the deformed grids is then calculated to create a pseudo-Euclidean space. The classification of the sample under examination to one of the 7 possible classes of facial expressions, i.e. anger, disgust, fear, happiness, sadness, surprise and neutral, is performed using multiclass SVMs defined in the previously mentioned pseudo-Euclidean. The experiments were performed using the Cohn-Kanade database and the results show that the above mentioned system can achieve an accuracy of 95.6%.

## 1. INTRODUCTION

In the last two decades, facial expression recognition has attracted the scientific interest due to its vital role in many applications such as human centered interfaces, e.g. virtual reality, video-conferencing, user profiling and customer satisfaction studies for broadcast and web services. Psychologists have defined a set of facial expressions that are thought to be expressed in a similar way all over the world, thus making the facial expression recognition more standard. These facial expressions are anger, disgust, fear, happiness, sadness and surprise [1]. These basic facial expressions in addition with the neutral state are the target of facial expression recognition systems developed nowadays. A survey on automatic facial expression recognition can be found in [2].

In this paper, a novel method for the recognition of facial expressions is proposed. The system firstly tracks the Candide facial grid on the video sequence under examination to obtain the deformed grid that corresponds to the facial ex-

pression depicted. Unlike the method proposed in [3], knowledge of the grid that corresponds to the neutral state is not necessary, as the proposed system requires only the deformed grid obtained from the grid tracking system and does not need to calculate the grid coordinates difference between the neutral and fully expressed image. Thus, the system can take as an input a video sequence starting from any facial expression and classify each frame to one of the seven facial expression classes (6 basic facial expressions plus neutral state). The system achieved an accuracy rate of 95.6% on experiments performed in the Cohn-Kanade database.

## 2. SYSTEM DESCRIPTION

The diagram of the system used for the experiments is shown in Figure 1. The information extraction subsystem consists

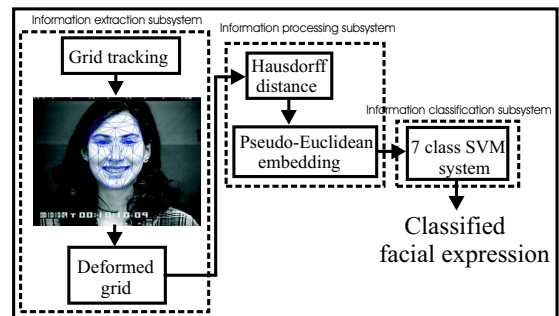


Fig. 1. Flow chart of the proposed system

of the grid tracking system described in [4]. The extracted information, is used as an input to the information processing subsystem, that includes the calculation of the Hausdorff distance and the pseudo-Euclidean embedding part. Finally, the information classification subsystem consists of a 7-class SVMs system that classifies the embedded deformed grid into one of the 7 facial expression classes under examination.

## 3. INFORMATION EXTRACTION SUBSYSTEM

The geometrical information extraction is performed by a grid tracking system, based on deformable models [4], that uses a

This work was supported by the "SIMILAR" European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc) for Ms. Kotsia and by project 03ED849 co-funded by the European Union and the Greek Secretariat of Research and Technology (Hellenic Ministry of Development) of the Operational Program for Competitiveness within the 3rd Community Support Framework for Mr. Zafeiriou.

pyramidal implementation of the well-known Kanade-Lucas-Tomasi (KLT) algorithm. The user has to place manually a number of Candide grid nodes on the corresponding positions of the face depicted at the first frame of the image sequence. The algorithm automatically adjusts the grid to the face and then tracks it through the image sequence, as it evolves through time to reach its highest intensity, thus producing the deformed Candide grid.

#### 4. HAUSDORFF DISTANCES FOR PSEUDO-EUCLIDEAN EMBEDDING

A popular measure between sets of points is the Hausdorff distance [5]. In the proposed approach we have adopted this distance in order to measure the similarity between grids.

##### 4.1. Hausdorff distance

Given two finite point sets:  $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_p\}$  and  $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_q\}$  (in our case this set of points is the set of nodes building the Candide facial grid), the Hausdorff distance is defined as:

$$H(\mathcal{A}, \mathcal{B}) = \max\{d(\mathcal{A}, \mathcal{B}), d(\mathcal{B}, \mathcal{A})\} \quad (1)$$

where

$$d(\mathcal{A}, \mathcal{B}) = \sup_{\mathbf{a} \in \mathcal{A}} \inf_{\mathbf{b} \in \mathcal{B}} \|\mathbf{a} - \mathbf{b}\| \quad (2)$$

where  $\|\cdot\|$  represents some underlying norm defined in the space of the two point sets, which is generally required to be an  $L_p$  norm, usually the  $L_2$  or Euclidean norm.

In the proposed method, in order to measure the similarity between facial grids we use a robust alternative of the Hausdorff distance, the so-called mean Hausdorff distance [5]. The mean Hausdorff distance  $d_{MH}(\mathcal{A}, \mathcal{B})$  from  $\mathcal{A}$  to  $\mathcal{B}$  is defined as:

$$d_{MH}(\mathcal{A}, \mathcal{B}) = \frac{1}{N(\mathcal{A})} \sum_{\mathbf{a} \in \mathcal{A}} \min_{\mathbf{b} \in \mathcal{B}} \|\mathbf{a} - \mathbf{b}\| \quad (3)$$

where  $N(\mathcal{A})$  is the number of points in  $\mathcal{A}$ . In the proposed approach we use the Hausdorff distance in (3) in order to create a feature space, using pseudo-Euclidean embedding so as to define later a multiclass SVM classifier in this space.

It should be noted here that in the setup used in this paper, where the same grid (the Candide grid) is tracked in all cases, the correspondences between the grids in the two grid sets (i.e. the nodes of the deformed grids) are known, point  $\alpha_i$  corresponds to point  $\mathbf{b}_i$ . Thus, the sum of Euclidean distances  $\sum_i \|\alpha_i - \mathbf{b}_i\|$  would suffice. However, the use of Hausdorff distance makes the proposed system applicable to other scenarios, e.g. when different grids are used in each use or when part of the grid is not available (e.g. due to image cropping).

##### 4.2. Embedding to pseudo-Euclidean spaces

It can be easily proven that the measure in (3) satisfies the following properties:

- reflectivity i.e.,  $d_{MH}(\mathcal{A}_i, \mathcal{A}_i) = 0$
- positivity i.e.,  $d_{MH}(\mathcal{A}_i, \mathcal{A}_j) > 0$  if  $\mathcal{A}_i \neq \mathcal{A}_j$
- symmetry i.e.,  $d_{MH}(\mathcal{A}_i, \mathcal{A}_j) = d(\mathcal{A}_j, \mathcal{A}_i)$ .

Thus, the mean Hausdorff distance is a proper dissimilarity measure [6]. But this measure is not a metric, since it does not satisfy the triangle inequality. We will use this dissimilarity measure in order to define an embedding in a pseudo-Euclidean space. For more details on pseudo-Euclidean embedding and dissimilarity based pattern recognition, an interested reader can refer to [6] and the references therein.

Let  $\{\mathcal{A}_1, \dots, \mathcal{A}_N\}$  be the set of training facial grid database. The dissimilarity matrix of the training is defined as:

$$[\mathbf{D}]_{i,j} = d_{MH}(\mathcal{A}_i, \mathcal{A}_j). \quad (4)$$

We will use the dissimilarity matrix  $\mathbf{D}$  in order to define an embedding  $\mathbf{X} \in \mathbb{R}^{k \times N}$ , where  $k \leq N$  is the dimensionality of the embedding and the  $i$ -th column of  $\mathbf{X}$ , denoted as  $\mathbf{x}_i$ , corresponds to the feature vector of the facial grid  $\mathcal{A}_i$  in the pseudo-Euclidean space. In order to find the embedding  $\mathbf{X}$ , the matrix  $\mathbf{B}$  is defined as:

$$\mathbf{B} = -\frac{1}{2} \mathbf{J} \mathbf{D} \mathbf{J} \quad (5)$$

where  $\mathbf{J} = \mathbf{I}_{N \times N} - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T \in \mathbb{R}^{N \times N}$  is the centering matrix, where  $\mathbf{I}_{N \times N}$  is the  $N \times N$  identity matrix and  $\mathbf{1}_N$  is the  $N$ -dimensional vector of ones. The matrix  $\mathbf{J}$  projects the data so that the embedding  $\mathbf{X}$  has zero mean. The eigen-decomposition of the matrix  $\mathbf{B}$  will give us the desired embedding. The matrix  $\mathbf{B}$  is positive semi-definite (i.e., it has real and non-negative eigenvalues), if and only if the distance matrix  $\mathbf{D}$  is Euclidean [6]. Therefore, for a non-Euclidean  $\mathbf{D}$  (like the one derived from our dissimilarity measure),  $\mathbf{B}$  has negative eigenvalues. Let  $p$  and  $q$  be the number of positive and negative eigenvalues of matrix  $\mathbf{B}$ . Then, the matrix  $\mathbf{B}$  can be written as:

$$\mathbf{B} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T = \mathbf{Q} |\mathbf{\Lambda}|^{\frac{1}{2}} \begin{bmatrix} \mathbf{M} & \\ & \mathbf{0} \end{bmatrix} |\mathbf{\Lambda}|^{\frac{1}{2}} \mathbf{Q}^T = \mathbf{X}^T \mathbf{M} \mathbf{X} \quad (6)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix with the diagonal consisting of the  $p$  positive and  $q$  negative eigenvalues, which are presented in the following order: first, positive eigenvalues in decreasing order, then negative ones in decreasing magnitude and finally the zero values. The matrix  $\mathbf{Q}$  is an orthogonal matrix of the corresponding eigenvectors. The matrix  $\mathbf{M}$  is equal to  $\mathbf{M} = \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{q \times q} \end{bmatrix}$  where  $\mathbf{I}_{p \times p}$  and  $\mathbf{I}_{q \times q}$  are the identity  $p \times p$  and  $q \times q$  matrices, and  $k = p + q$ . The matrix  $\mathbf{X}$  is the embedding of the set of facial grids in the pseudo-Euclidean space  $\mathbb{R}^k = \mathbb{R}^{(p,q)}$  [6]:

$$\mathbf{X} = |\mathbf{\Lambda}_k|^{\frac{1}{2}} \mathbf{Q}_k^T \quad (7)$$

where  $\Lambda_k$  contains only the non-zero diagonal elements of  $\Lambda$  and  $\mathbf{Q}_k$  is the matrix with the corresponding eigenvectors.

For the dimensions of the embedding  $\mathbf{X}$  that correspond to negative eigenvalues we have chosen to use only the magnitude [6, 7]. This step is preferred when defining the Hessian matrix of the quadratic optimization problem of Support Vector Machines in pseudo-Euclidean spaces [6, 7]. In this case the new embedding is:

$$\mathbf{X}_l = \Delta_l^{\frac{1}{2}} \mathbf{Q}_l^T \quad (8)$$

where  $\Delta_l$  is a diagonal matrix having as diagonal elements the magnitude of the diagonal elements of  $\Lambda_l$ , in descending order. The matrix  $\mathbf{Q}_l$  contains the corresponding eigenvectors. For the dimensionality  $l$  of the new embedding, the following inequality holds:  $l \leq k \leq N$ . The new embedding is purely Euclidean. As already mentioned, the vector  $\mathbf{x}_i^l$ , i.e. the  $i$ -th column of the matrix  $\mathbf{X}_l$  corresponds to the feature vector of the grid  $\mathcal{A}_i$  in the pseudo-Euclidean space.

## 5. MULTICLASS SVMs FOR CLASSIFICATION

### 5.1. Training phase

The new space is purely Euclidean and a multi-class SVM is built now to classify the vectors  $\mathbf{x}_i^l$ . The training data are  $(\mathbf{x}_1^l, l_1), \dots, (\mathbf{x}_N^l, l_N)$  where  $\mathbf{x}_i^l \in \mathbb{R}^L$  are the feature vectors and  $l_j \in \{1, \dots, 7\}$  are the facial expression labels of the feature vectors. The multi-class SVMs problem solves only one optimization problem [8]. It constructs 7 facial expressions rules, where the  $k$ -th function  $\mathbf{w}_k^T \phi(\mathbf{x}_i^l) + b_k$  separates training vectors of the class  $k$  from the rest of the vectors, by minimizing the objective function:

$$\min_{\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}} \quad \frac{1}{2} \sum_{k=1}^7 \mathbf{w}_k^T \mathbf{w}_k + C \sum_{j=1}^N \sum_{k \neq l_j} \xi_j^k \quad (9)$$

subject to the constraints:

$$\begin{aligned} \mathbf{w}_{l_j}^T \phi(\mathbf{x}_i^l) + b_{l_j} &\geq \mathbf{w}_k^T \phi(\mathbf{x}_i^l) + b_k + 2 - \xi_j^k \\ \xi_j^k &\geq 0, \quad j = 1, \dots, N, \quad k \in \{1, \dots, 7\} \setminus l_j. \end{aligned} \quad (10)$$

$\phi$  is the function that maps the deformation vectors to a higher dimensional space, where the data are supposed to be linearly or near linearly separable. It is not necessary to calculate the function  $d$  since all the calculations are performed using the kernel trick [9].  $C$  is the term that penalizes the training errors. The vector  $\mathbf{b} = [b_1 \dots b_7]^T$  is the bias vector and  $\boldsymbol{\xi} = [\xi_1^1, \dots, \xi_i^k, \dots, \xi_N^7]^T$  is the slack variable vector. For the solution of the optimization problem (9) subject to the constraints (10) someone can refer to [8, 9] The solution of (9) subject to (10) provides us with normal vectors  $\mathbf{w}_1, \dots, \mathbf{w}_7$  and with seven bias terms  $b_1, \dots, b_7$ .

### 5.2. Facial grid classification using the trained SVMs

In this Section we will show how features from previously "unseen" facial grids. using the proposed dissimilarity measure which are afterwards classified with the multi-class SVMs system. Let  $\{\mathcal{G}_1, \dots, \mathcal{G}_n\}$  be a set of  $n$  testing facial grids. We create the matrix  $\mathbf{D}_n \in \mathbb{R}^{n \times N}$ , with  $[\mathbf{D}_n]_{i,j} = d_{MH}(\mathcal{G}_i, \mathcal{A}_j)$ . The matrix  $\mathbf{D}_n$  represents the similarity, with respect to Hausdorff distance, between the  $n$  test facial grids and all the training facial grids. The matrix  $\mathbf{B}_n \in \mathbb{R}^{n \times N}$  of inner products that relates all the new (test) facial grids to all facial grids from the training set is then found as follows:

$$\mathbf{B}_n = -\frac{1}{2}(\mathbf{D}_n \mathbf{J} - \mathbf{U} \mathbf{D} \mathbf{J}) \quad (11)$$

where  $\mathbf{J}$  is the centering matrix and  $\mathbf{U} = \frac{1}{N} \mathbf{1}_n \mathbf{1}_N^T \in \mathbb{R}^{n \times N}$ . The embedding  $\mathbf{X}_n \in \mathbb{R}^{l \times n}$  of the test facial grids is defined as:

$$\mathbf{X}_n = \Delta_l^{-\frac{1}{2}} \mathbf{Q}_l^T \mathbf{B}_n^T. \quad (12)$$

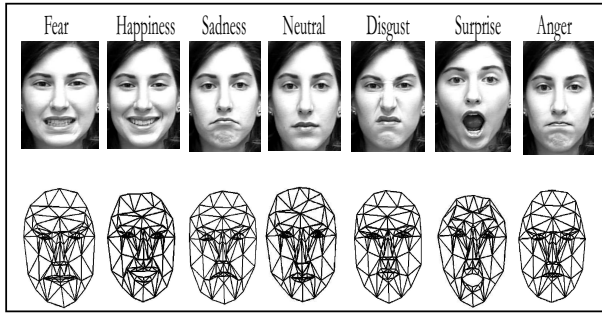
The columns of the matrix  $\mathbf{X}_n$  are the features used for classification. Let  $\mathbf{x}_{i,n} \in \mathbb{R}^l$  be the  $i$ -th column of the matrix  $\mathbf{X}_n$ , i.e. the vector that contains the features of the grid  $\mathcal{G}_i$ . The classification of  $\mathcal{G}_i$  to one of the seven facial expression classes is performed by the decision function:

$$h(\mathcal{G}_i) = \arg \max_{k=1, \dots, 7} (\mathbf{w}_k^T \phi(\mathbf{x}_{i,n}) + b_k), \quad (13)$$

where  $\mathbf{w}_k$  and  $b_k$  have been found during training, as described in Section 5.1.

## 6. EXPERIMENTAL RESULTS

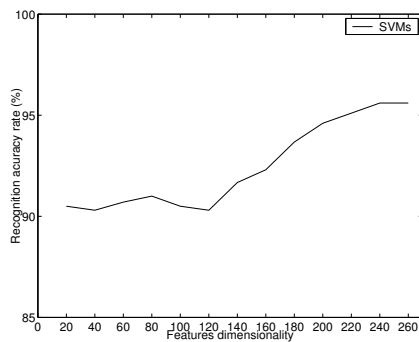
The database used for the facial expression recognition experiments was created using the Cohn-Kanade database [10]. This database is annotated with FAUs. These combinations of FAUs were translated into facial expressions according to [2], in order to define the corresponding ground truth for the facial expressions. In Figure 2, a sample of the grids acquired for one person from the database used for the experiments, is shown. The classifier accuracy was measured using the leave-one-out cross-validation approach described below, in order to make maximal use of the available data and produce averaged classification accuracy results. The image sequences contained in the database are divided into 7 classes, each one corresponding to one of the 7 facial expressions. Each class consists of the same number of fully expressive facial expression grid samples (37 facial grids for every expression). One facial expression sample from each class is used for the test set, while the remaining samples form the training set. During the training procedure the Hausdorff distance matrix  $\mathbf{D}$  of the training samples is calculated. Afterwards, a pseudo-Euclidean embedding is performed and finally the multiclass SVM system is trained.



**Fig. 2.** An example of the grids extracted for a poser from the Cohn-Kanade database.

The seven test samples are firstly projected in the pseudo-Euclidean embedding, as described in Section 5.2, and afterwards classified using (13). Subsequently, the samples forming the test set are incorporated into the current training set and a new set of samples (one for each class) is extracted to form the new test set. The remaining samples create the new training set. This procedure is repeated until all of the samples are used as test sets. The classification accuracy is measured as the mean value of the percentages of the correctly classified facial expressions.

We have experimented with the dimensionality of the pseudo-Euclidean embedding which can be modified by keeping only the  $p$  eigenvectors with the largest eigenvalues (i.e. using a matrix  $\mathbf{X} \in \mathbb{R}^{p \times (36 \times 7)}$ ). Figure 3 depicts the facial expression recognition rate achieved versus the dimensionality of the pseudo-Euclidean space. The accuracy achieved with the



**Fig. 3.** Facial expression recognition rate (7 facial expressions) versus dimensionality of the pseudo-Euclidean embedding in the Cohn-Kanade database

proposed system was equal to 95.6% when SVMs were used with a polynomial kernel of degree equal to 3 for  $p = 260$ .

## 7. CONCLUSIONS

A novel method for the classification of seven facial expressions (i.e. anger, disgust, fear, happiness, sadness, surprise

and neutral) using only facial grids that have been deformed to find the facial characteristics in videos, has been presented. The Hausdorff distance has been exploited in order to create a pseudo-Euclidean space and a multiclass SVM system has been defined in this space to be used for the classification of expression. Experiments showed that the proposed technique achieved an accuracy rate of 95,6% when recognizing seven facial expressions (6 basic facial expressions plus neutral).

## 8. REFERENCES

- [1] P. Ekman and W.V. Friesen, *Emotion in the Human Face*, Prentice Hall, 1975.
- [2] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing*, vol. 18, no. 11, pp. 881–905, August 2000.
- [3] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 172 – 187, 2007.
- [4] S. Krinidis and I. Pitas, "Statistical analysis of facial expressions for facial expression synthesis," *submitted to IEEE Transactions on Multimedia*, 2006.
- [5] Jian-Wei Zhang, Guo-Qiang Han, and Yan Wo, "Image registration based on generalized and mean hausdorff distances," in *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, 18-21 August 2005.
- [6] E. Pekalska, P. Paclik, and R.P.W. Duin, "A generalized kernel approach to dissimilarity-based classification," *Journal of Machine Learning Research*, vol. 2, pp. 175–211, 2001.
- [7] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 482 – 492, 2005.
- [8] J. Weston and C. Watkins, "Multi-class Support Vector Machines," Tech. Rep. Technical report CSD-TR-98-04, 1998.
- [9] V. Vapnik, *Statistical Learning Theory*, J.Wiley, New York, 1998.
- [10] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings of IEEE International Conference on Face and Gesture Recognition*, March 2000, pp. 46–53.