Dynamic Facial Expression Recognition Using Laplacian Eigenmaps-based Manifold Learning

Bogdan Raducanu and Fadi Dornaika

***Abstract*— In this paper, we propose an integrated framework for tracking, modelling and recognition of facial expressions. The main contributions are: (i) a view- and texture independent scheme that exploits facial action parameters estimated by an appearance-based 3D face tracker; (ii) the complexity of the non-linear facial expression space is modelled through a manifold, whose structure is learned using Laplacian Eigenmaps. The projected facial expressions are afterwards recognized based on Nearest Neighbor classifier; (iii) with the proposed approach, we developed an application for an AIBO robot, in which it mirrors the perceived facial expression.**

Keywords: facial expression recognition, temporal classifiers, manifold learning, human-robot interaction, AIBO robot

I. INTRODUCTION

In the field of Human-Computer Interaction (HCI), computers will be enabled with perceptual capabilities in order to facilitate the communication protocols between people and machines. In other words, computers will be endowed with natural ways of communication people use in their everyday life. Among them, facial expression represents a powerful mean people use to express their emotions and other aspects related with their social or psychological status.

In the past, a lot of effort was dedicated to recognize facial expression in still images. For this purpose, many techniques have been applied: neural networks [1], Gabor wavelets [2] and Active Appearance Models (AAM) [3]. A very important limitation to this strategy is the fact that still images usually capture the apex of the expression, i.e., the instant at which the indicators of emotion are most marked. In their daily life, people seldom show apex of their facial expression during normal communication.

More recently, attention has been shifted particularly towards dynamic modelling of facial expressions [4], [5], [6]. Dynamical approaches can use shape deformations [7], texture dynamics [8] or a combination of them [9]. In [10], the authors propose a dynamic classifier that is based on building spatio-temporal model for each universal expression derived from Fourier transform. The recognition of unseen expression uses the Hausdorff distance to compute dissimilarity values for classification.

Modelling the variability of facial expressions is a very challenging task. Facial expressions form a class of *objects* with a well-defined structure which suffers elastic deformations. Ideally, an optimal representation would be able to cope with all these complex transformations. This is usually achieved through a manifold learning approach.

The use of linear and non-linear manifolds for facial expression recognition was addressed by many researchers. Most of the proposed manifold learning schemes addressed frame-wise representation of facial textures. In [11], the authors propose a Bayesian approach to modelling temporal transitions of facial expressions represented in a manifold. In [12], the authors propose a Bayesian framework for face recognition from video sequences. They represent face appearances by linear sub-manifolds together with probabilistic transitions. The linear sub-manifolds are obtained via clustering and classical Principal Component Analysis (PCA). In [13], the authors propose a probabilistic video-based facial expression recognition method on manifolds. An enhanced Lipschitz embedding is developed to embed the aligned face appearance in a low dimensional space. A probabilistic model of transition between expressions is learned through training videos in the embedded space.

In this paper we present an integrated framework for dynamic facial expression recognition, consisting of 3 stages. First, a temporal signature extracted from a video sequence will be used as a sample data that encodes facial deformation. We extract facial dynamics by using the 3D face tracker [14] based on Online Appearance Models and a deformable 3D mesh. This face tracker is able to retrieve in real-time the 3D face pose parameters as well as some facial actions needed for recognizing facial expressions. Second, we use the unsupervised non-linear embedding provided by Laplacian Eigenmaps (LE) that preserves local neighborhood information in order to embed temporal signatures on a low-dimension manifold. Third, facial expression recognition is performed on the embedded signatures using classical machine learning techniques: Linear Discriminant Analysis (LDA) with a Nearest Neighbor (NN) classifier. This process is depicted in Figure 1.

What differentiate our work from existing dynamic recognition schemes are the following: 1) expressions can be recognized even in the presence of 3D head motions whereas most of the proposed expression recognition schemes require a frontal view of the face. 2) the recognition is based on shape deformation only, which makes the recognition scheme not depending on the imaging conditions by which the universal expressions are learned. On the other hand, most related works rely on the use of image raw brightness changes. 3) the use of aligned temporal signatures as training

B. Raducanu is with the Computer Vision Center, Edifici "O" - Campus UAB, 08193 Bellaterra (Barcelona), Spain bogdan@cvc.uab.es

F. Dornaika is with IKERBASQUE, Basque Foundation for Science, and the University of the Basque Country, San Sebastian, Spain fadi_dornaika@ehu.es

B. Raducanu is supported by the projects TIN2009-14404-C02-00 and CONSOLIDER-INGENIO 2010 (CSD2007-00018), Ministerio de Educación y Ciencia, Spain.

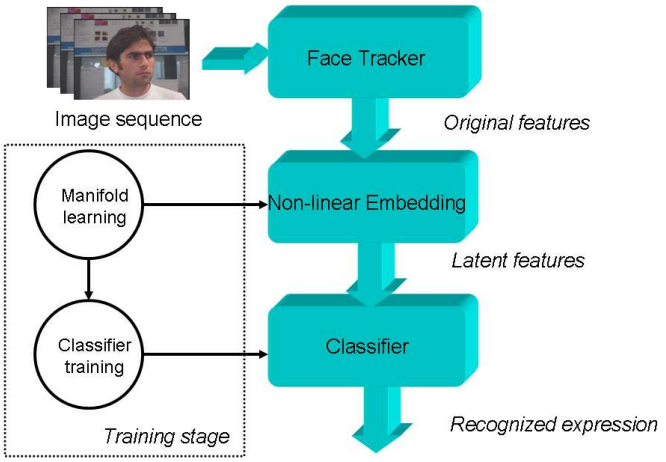


Fig. 1. Integrated framework for dynamic facial expression recognition.

examples can link our proposed method to all classical machine learning approaches.

The rest of the paper is organized as follows. Section II describes the extraction of temporal signatures associated with universal expressions. Section III reviews the Laplacian Eigenmaps embedding. In section IV we present some experimental results as well as an application for the AIBO robot. Finally, in section V we draw our conclusions.

II. FROM VIDEOS TO FACIAL DYNAMICS AND EXPRESSIONS

The objective of this work is to recognize facial expressions in continuous videos using data-driven machine learning algorithms. Therefore, encoding the displayed universal expressions is a crucial step. Extracting facial dynamics associated with facial muscle deformations from video sequences is a challenging task. This task is made more difficult if the subject's head moves in 3D space. The recognition of facial expressions with significant head motion is required by many applications such as human computer interaction and computer graphics animation [15], [16], [17] as well as training of social robots [18], [19].

A. Modelling Faces

In our work, we use a common 3D deformable face model—the *Candide* model [20] (See Figure 2). Despite the simplicity of this 3D wireframe model, it can be used to extract a subset of 3D facial dynamics in real time using one single camera. The 3D shape of this wireframe model is directly recorded in coordinate form. It is given by the coordinates of the 3D vertices $\mathbf{P}_i, i = 1, \ldots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$; the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ is written as:

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\,\boldsymbol{\tau_a} \quad (1)$$

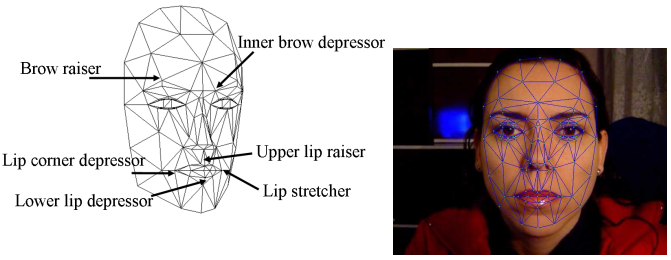
where $\mathbf{g}_s$ is the static shape of the model, $\boldsymbol{\tau_a}$ the animation control vector, and the columns of $\mathbf{A}$ are the Animation Units. In this study, we use six modes for the facial Animation Units (AUs) matrix $\mathbf{A}$. We have chosen the following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer, outer eyebrow raiser (see Figure 2**.(a)**). These AUs are enough to cover most common facial animations. Moreover, they are essential for conveying emotions.

In equation (1), the 3D shape is expressed in a local coordinate system. However, one should relate the 3D coordinates to the image coordinate system. To this end, we adopt the weak perspective projection model. We neglect the perspective effects since the depth variation of the face can be considered as small compared to its absolute depth. Thus, the state of the 3D wireframe model is given by the 3D face pose parameters (three rotations and three translations) and the internal face animation control vector $\boldsymbol{\tau_a}$. This is given by the 12-dimensional vector $\mathbf{b}$:

$$\mathbf{b} = [\theta_x, \ \theta_y, \ \theta_z, \ t_x, \ t_y, \ t_z, \ \boldsymbol{\tau_a}^T]^T \quad (2)$$

Note that if only the aspect ratio of the camera is known, then the component t_z is replaced by a scale factor having the same mapping role between 3D and 2D. In this case, the state vector is given by (s denotes the scale factor):

$$\mathbf{b} = [\theta_x, \ \theta_y, \ \theta_z, \ t_x, \ t_y, \ s, \ \boldsymbol{\tau_a}^T]^T \quad (3)$$



(a) **(b)**

Fig. 2. **(a)** *Candide* model. **(b)** *Candide* model adapted to an input facial image.

B. Simultaneous Face And Facial Action Tracking

In order to recover the facial expression one has to compute the facial actions encoded by the vector $\boldsymbol{\tau_a}$ which encapsulates the facial deformation. Since our recognition scheme is view-independent these facial actions together with the 3D head pose should be simultaneously estimated. In other words, the objective is to compute the state vector $\mathbf{b}$ for every video frame.

For this purpose, we use the tracker based on Online Appearance Models [14]. This appearance-based tracker aims at computing the 3D head pose and the facial actions, i.e. the vector $\mathbf{b}$, by minimizing a distance between the incoming warped frame and the current *shape-free* appearance of the face. This optimization is carried out using a gradient descent

method. The statistics of the *shape-free* appearance as well as the gradient matrix are updated every frame. This scheme leads to a fast and robust tracking algorithm.

C. Representing Dynamic Universal Expressions By Features

In order to learn the spatio-temporal structures of the facial actions associated with facial expressions, we have used a simple supervised learning scheme that consists in two stages. In the first stage, training video sequences depicting different universal facial expressions are tracked using the appearance-based face tracker. The retrieved facial actions $\boldsymbol{\tau}_{\mathbf{a}}$ are represented by time series. In other words, an example (expression going from neutral to apex) is encoded by a sequence of facial actions $\boldsymbol{\tau}_{\mathbf{a}(1)}, \ldots, \boldsymbol{\tau}_{\mathbf{a}(T)}$. One can note that this temporal sequence (trajectory) can be considered as a compact representation of the spatio-temporal facial structure that one expects to observe whenever the face undergoes a given universal expression. In the second stage, since we are using example based classifiers all examples should have the same dimension. To this end, all facial action sequences are aligned in the time domain using the Dynamic Time Warping (DTW) technique [21]. Dynamic Time Warping is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. Thus, a given example (universal expression) is represented by a feature vector obtained by concatenating the vectors $\boldsymbol{\tau}_{\mathbf{a}}(t)$ belonging to the aligned temporal sequence.

More precisely, video sequences have been picked up from the CMU database [22]. These sequences depict five frontal view universal expressions (surprise, sadness, joy, disgust and anger). Each expression is performed by 20 different subjects, starting from the neutral one. Altogether we select 35 video sequences composed of around 15 to 20 frames each, that is, the average duration of each sequence is about half a second. The training video sequences have an interesting property: all performed expressions go from the neutral expression to a high magnitude expression by going through a moderate magnitude around the middle of the sequence. In the final stage of the learning all training trajectories are aligned in the time domain using the Dynamic Time Warping technique by fixing a nominal duration for a facial expression. In our experiments, this nominal duration is set to 18 frames. This choice was guided by many observations that show that a complete expression can be displayed in 15-20 frames assuming that the video rate is 30 fps.

Finally, a training video sequence associated with a universal expression is represented by a feature vector $\mathbf{y}$ corresponding to the second half of the aligned trajectory (only nine frames are used). This feature vector $\mathbf{y}$ is given by

$$(\boldsymbol{\tau}_{\mathbf{a}(10)}^T, \boldsymbol{\tau}_{\mathbf{a}(11)}^T, \boldsymbol{\tau}_{\mathbf{a}(12)}^T, \boldsymbol{\tau}_{\mathbf{a}(13)}^T, \boldsymbol{\tau}_{\mathbf{a}(14)}^T, \boldsymbol{\tau}_{\mathbf{a}(15)}^T, \boldsymbol{\tau}_{\mathbf{a}(16)}^T, \boldsymbol{\tau}_{\mathbf{a}(17)}^T, \boldsymbol{\tau}_{\mathbf{a}(18)}^T)^T$$

Thus, the dimension of this feature vector is 54. Figure 3 shows nine frames encoding a temporal signature of a joy expression.

We decided to remove in our analysis the first half trajectory (from initial, neutral state to half-apex) since we found them irrelevant for the purposes of the current study. Therefore, a feature vector associated with a given universal expression is encoding a signature of one realization of this expression that goes from a moderate magnitude to the apex.

Fig. 3. Constructing the feature vector (54 components) from nine frames associated with joy expression dynamics.

III. EMBEDDING WITH LAPLACIAN EIGENMAPS

In this paper, we use Laplacian Eigenmap [23] to map temporal signatures into a low-dimensional space. Using the notion of the Laplacian of the graph, this non-supervised algorithm computes a low-dimensional representation of the data set by optimally preserving local neighborhood information in a certain sense. We assume that we have a set of N samples $\{\mathbf{y}_i\}_{i=1}^N \subset \mathbb{R}^D$. Define a neighborhood graph on these data, such as a K-nearest-neighbor or ϵ-ball graph, or a full mesh, and weigh each edge $\mathbf{y}_i \sim \mathbf{y}_j$ by a symmetric affinity function $w_{ij} = K(\mathbf{y}_i; \mathbf{y}_j)$, typically Gaussian:

$$w_{ij} = \exp(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{2\,\sigma^2}). \tag{4}$$

We seek latent points $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^L$ that minimizes $\frac{1}{2}\sum_{i,j} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2$, which discourages placing far apart latent points that correspond to similar observed points. If $\mathbf{W}$ denotes the symmetric affinity matrix and $\mathbf{D}$ is the diagonal weight matrix, whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, then the Laplacian matrix is given $\mathbf{L} = \mathbf{D} - \mathbf{W}$. It can be shown that the objective function can also be written as:

$$\frac{1}{2}\sum_{i,j} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = tr(\mathbf{Z}^T\,\mathbf{L}\,\mathbf{Z}) \tag{5}$$

where $\mathbf{Z} = [\mathbf{x}_1^T; \ldots; \mathbf{x}_N^T]$ is the $N \times L$ embedding matrix. The i^{th} row of the matrix $\mathbf{Z}$ provides the vector $\mathbf{x}_i$—the embedding coordinates of the sample $\mathbf{y}_i$.

The embedding matrix $\mathbf{Z}$ is the solution of the optimization problem:

$$\min_{\mathbf{Z}} tr(\mathbf{Z}^T \mathbf{L}\, \mathbf{Z}) s.t. \mathbf{Z}^T \mathbf{D}\, \mathbf{Z} = \mathbf{I}, \mathbf{Z}^T \mathbf{L}\, \mathbf{e} = \mathbf{0} \qquad (6)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{e} = (1, \ldots, 1)^T$. The first constraint eliminates the trivial solution $\mathbf{Z} = \mathbf{0}$ (by setting an arbitrary scale) and the second constraint eliminates the trivial solution $\mathbf{e}$ (all samples are mapped to the same point). Standard methods show that the embedding matrix is provided by the matrix of eigenvectors corresponding to the smallest eigenvalues of the generalized eigenvector problem,

$$\mathbf{L}\, \mathbf{z} = \lambda \mathbf{D}\, \mathbf{z} \qquad (7)$$

Let the column vectors $\mathbf{z}_0, \ldots, \mathbf{z}_{N-1}$ be the solutions of (7), ordered according to their eigenvalues, $\lambda_0 = 0, \ldots, \lambda_{N-1}$. The eigenvector corresponding to eigenvalue 0 is left out and only the next eigenvectors for embedding are used.

The embedding of the original samples is given by the row vectors of the embedding matrix $\mathbf{Z}$, that is,

$$\mathbf{y}_i \longrightarrow \mathbf{x}_i = (z_1(i), \ldots, z_L(i))^T \qquad (8)$$

where $L < N$ is the dimension of the new space.

IV. EXPERIMENTAL RESULTS AND APPLICATION

A. Tests on the CMU Database

In order to test our approach, we used a subset from the CMU facial expression database [22], containing 20 persons who are displaying 5 expressions: surprise, sadness, joy, disgust and anger. For dynamical facial expression recognition evaluation, we used the truncated trajectories, that is, the temporal sequence containing 9 frames, with the first frame representing a *subtle* facial expression and the last one corresponding to the apex state of the facial expression (similar to those depicted in figure 3). We decided to remove in our analysis the first few frames (from initial, *neutral state* to half-apex) since we found them irrelevant for the purposes of the current study.

Once the original trajectory vectors (temporal signatures) are embedded on the LE space, we further refine the data representation for recognition by using a Linear Discriminant Analysis (LDA). While LE is capable of recovering the intrinsic low-dimensional space, however, it may not be optimal for recognition. For our evaluation, we adopted a 10-fold cross-validation strategy: 90% of the samples are used for training and 10% for test. We chose as classifier the K-nearest neighbor.

In figure 4, we depicted the representation of the first three components of the data embedded on the LE space.

We manually set the parameter K, representing the neighborhood's size in the graph. Table I depicts the recognition rate as a function of K when the first 10 dimensions in LE space have been used. As can be seen, the recognition rate may slightly vary. The scale of the Gaussian kernel $2\sigma^2$ (4) has been automatically estimated once the size of the neighborhood, K, has been fixed. We computed this scale as the average distance to the K-nearest neighbors (over all training examples). This is one of the possibilities of estimating automatically this parameter, as suggested in [23].

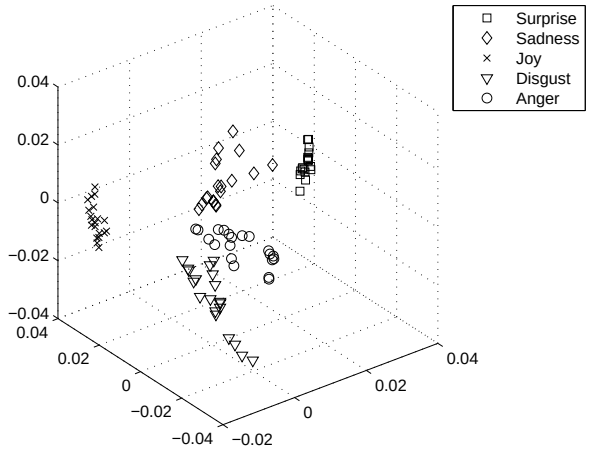


Fig. 4. The projection of the first three components of the original data on LE space.

K = 7	K = 9	K = 13	K =17	K =23
88.57	94.28	91.42	94.28	97.14

TABLE I

RECOGNITION RATE AS A FUNCTION OF THE NEIGHBORHOOD'S SIZE OF THE GRAPH.

In order to assess the performance of the LE embedding, we also perform the tests using the linear embedding provided by PCA (Principal Component Analysis). Thus we compared the proposed LE+LDA scheme for recognition with PCA+LDA. In table II, we represented the recognition accuracy for several values of the embedding dimensionality. For classification, we used the Nearest Neighbor with K=1, 3 and 5. A more elaborated comparison between the schemes LE+LDA and PCA+LDA is depicted in figure 5. It can be appreciated that when the dimensionality of the embedded space is smaller than 20, the recognition rate is higher when the samples are projected on the LE space than on PCA. The fact that LE embedding offers the best results on low dimensionality and its performance degrades when the dimensionality increases is not surprising. A possible explanation for this situation is given in [24]: when the number of dimensions increases, PCA will discard less and less information. At the same time, LE will start overfitting, a problem to which it is much more sensitive than PCA because of its nonlinear nature.

In other words, LE offers a more powerful compression of the original data than PCA. This is a very important result especially for the case when the data lie in very high dimensionality space (like hyperspectral images) and we are interested in a significant dimensionality reduction without any relevant loss of intrinsic information.

LE/PCA	K-NN=1	K-NN=3	K-NN=5
5	91.42 / 91.42	88.57 / 88.57	91.42 / 91.42
10	97.14 / 94.285	97.14/ 91.42	97.14 / 91.42
15	91.42 / 85.71	91.42 / 85.71	91.42 / 85.71
20	88.57 / 68.57	88.57 / 65.71	88.57 / 68.57

TABLE II

RECOGNITION RATE AS A FUNCTION OF DIMENSIONALITY OF THE EMBEDDED SPACE: LE VS. PCA.

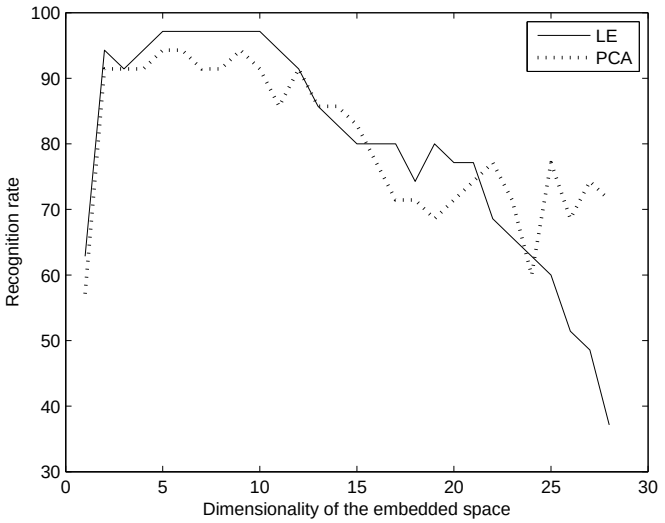


Fig. 5. Recognition rate as a function of LE and PCA dimensionality.

Fig. 6. The experimental setup.

Fig. 7. The figure illustrates the LEDs' configuration for each universal expression.

B. A Human-Robot Interaction Scenario For A Social Robot

In this subsection, we describe a human-robot interaction application based on our proposed approach. The application refers to mimicking the facial expressions of a person perceived by a robot's camera.

Without any loss of generality, we used an AIBO robot for our application. The input to the system is a video stream capturing the user's face (the experimental setup is depicted in figure 6). AIBO's human-like communication system is implemented through a series of *instincts* and *senses*: affection, movement, touch, hearing, sight and balance. AIBO is able to show its emotions through an array of LEDs situated in the frontal part of the head. These are depicted in figure 7, and are shown in correspondence with the six universal expressions. Notice that the blue lights that appear, in certain images, on each part of the head, are blinking LEDs whose meaning is to inform that the robot is remotely controlled[1]. This is a built-in feature and can not be turned off.

In addition to the LEDs' configuration, the robot response contains some small head and body motion. From its concept design, AIBO's affective states are triggered by the Emotion Generator engine. This occurs as a response to its internal state representation, captured through multi-modal interaction (vision, audio and touch). For instance, it can display the 'happiness' feeling when it detects a face (through the vision system) or it hears a voice. But it does not possess a built-in system for vision-based automatic facial expression recognition. For this reason, the application we created for AIBO could be seen as an extension of its pre-defined behaviors. This application is a very simple one, in which the robot is just imitating the expression of a human subject. In other words, we wanted to see its reaction according to the emotional state displayed by a person. Usually, the response of the robot occurs slightly after the apex of the human expression. The results of this application were recorded in a 2 minutes video which can be downloaded from the following address: `http://www.cvc.uab.es/~bogdan/AIBO-emotions.avi`. In order to be able to display simultaneously in the video the correspondence between person's and robot's expressions, we put them side by side. In this case only, we analyzed offline the content of the video and commands with the facial expression code were sent to the robot. Figure 8 illustrates nine recognized facial expressions from a 1600 frame-long video sequence.

[1]The application described in this paper, was built using the Remote Framework (RFW) programming environment (based on C++ libraries), which works on a client-server architecture over a wireless connection between a PC and the AIBO

V. CONCLUSIONS

This paper described an integrated framework for dynamic facial expression recognition. First, we proposed a temporal recognition scheme that classifies a given image in an unseen video into one of the universal facial expression categories using temporal facial deformation. The proposed approach relies on tracked facial actions provided by a real-time

Fig. 8. Person's facial expressions are shown in correspondence with the robot's response.

face tracker. Second, we use the unsupervised non-linear embedding provided by Laplacian Eigenmaps (LE) that preserves local neighborhood information in order to embed temporal signatures on a low-dimension manifold. Third, facial expression recognition is performed on the embedded signatures using classical machine learning techniques.

In the future, we want to further extend the research reported in this paper by focusing on the out-of-sample case for manifold learning: augmenting the graph Laplacian with new data without recomputing the whole embedding.

REFERENCES

[1] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 97–115, 2001.

[2] M. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan, "Machine learning methods for fully automatic recognition of facial expressions and facial actions," in *Proc. of IEEE. Int'. Conf. on SMC*, vol. I, The Hague, The Netherlands, 2004, pp. 592–597.

[3] J. Sung, S. Lee, and D. Kim, "A real-time facial expression recognition using the staam," in *Proc. of Int'l. Conf. on Pattern Recognition*, vol. I, Hong Kong, PR China, 2006, pp. 275–278.

[4] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang, "Facial expression recognition from video sequences: Temporal and static modeling," *Computer Vision and Image Understanding.*, vol. 91, no. 1-2, pp. 160–187, 2003.

[5] C. Shan, S. Gong, and P. McOwan, "Dynamic facial expression recognition using a bayesian temporal manifold model," in *Proc. of British Machine Vision Conference*, vol. I, Edinburgh, UK, 2006, pp. 297–306.

[6] M. Yeasin, B. Bullot, and R. Sharma, "Recognition of facial expressions and measurement of levels of interest from video," *IEEE Trans. on Multimedia*, vol. 8, no. 3, pp. 500–508, 2006.

[7] F. Dornaika and B. Raducanu, "Inferring facial expressions from videos: Tool and application," *Signal Processing: Image Communication*, vol. 22, no. 9, pp. 769–784, 2007.

[8] P. Yang, Q. Liu, X. Cui, and D. Metaxas, "Facial expression recognition using encoded dynamic features," in *Computer Vision and Pattern Recognition*, 2008.

[9] Y. Cheon and D. Kim, "Natural facial expression recognition using differential-aam and manifold learning," *Pattern Recognition*, vol. 42, pp. 1340–1350, 2009.

[10] T. Xiang, M. Leung, and S. Cho, "Expression recognition using fuzzy spatio-temporal modeling," *Pattern Recognition*, vol. 41, no. 1, pp. 204–216, January 2008.

[11] C. Shan, S. Gong, and P. W. McOwan, "Dynamic facial expression recognition using a bayesian temporal manifold model," in *British Machine Vision Conference*, 2006.

[12] K. Lee, J. Ho, M. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *IEEE Int. Conference on Computer Vision and Pattern Recognition*, 2003.

[13] Y. Chang, C. Hu, and M. Turk, "Probabilistic expression analysis on manifolds," in *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, USA, pp. 520–527.

[14] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107–1124, 2006.

[15] L. Cañamero and P. Gaussier, "Emotion understanding: robots as tools and models," in *Emotional Development: Recent Research Advances*, J. e. a. Nadel, Ed. New York: Oxford University Press, 2005, pp. 235–258.

[16] M. Pantic, "Affective computing," in *Encyclopedia of Multimedia Technology and Networking*, M. e. a. Pagani, Ed. Idea Group Publishing, 2005, vol. I, pp. 8–14.

[17] R. Picard, E. Vyzas, and J. Healy, "Toward machine emotional intelligence: analysis of affective physiological state," *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 23, no. 10, pp. 1175–1191, 2001.

[18] C. Breazeal, "Robot in society: friend or appliance?" in *Proc. of Wksp on Emotion-Based Agent Architectures*, 1999, p. N/A.

[19] ——, "Sociable machines: Expressive social exchange between humans and robots," Ph.D. dissertation, MIT, Cambridge, US, 2000.

[20] J. Ahlberg, *Model-based coding: extraction, coding and evaluation of face model parameters.* Sweden: Ph.D. Thesis, Dept. of Elec. Eng., Linköping Univ., 2002.

[21] M. Müller, *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007.

[22] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000, pp. 46–53.

[23] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[24] D. de Ridder and R. Duin, "Locally linear embedding for classification," in *Technical Report PH-2002-01*, Delft University of Technology, The Netherlands, 2002.