IDENTITY AND EXPRESSION RECOGNITION ON LOW DIMENSIONAL MANIFOLDS

Pedro Martins, Jorge Batista

Institute of Systems and Robotics, Dep. of Electrical Engineering and Computers
University of Coimbra - Portugal
{pedromartins,batista}@isr.uc.pt

ABSTRACT

A solution for identity and facial expression recognition is proposed using a two stage classifier approach using low dimensional representation of the geometry of the face. Face geometry is extracted from input images using Active Appearance Models (AAM) and low dimensional manifolds were then derived using Laplacian EigenMaps (LE) resulting in two types of manifolds, one for model identity and the other for person-specific facial expression. The first stage uses a multiclass Support Vector Machines (SVM) to establish identity across expression changes. The second stage deals with person-specific expression recognition, and is composed by a network of seven Hidden Markov Models (HMM) displaced in parallel, each one specialized on the several facial emotions analysed. The decision was made by the sequence that yielded the highest probability. For evaluation proposes a database was build consisting on 6770 images captured from 4 people exhibiting 7 different emotions. The identity overall recognition rate was 96.8%. Facial expression results are identity dependent, and the most expressive individual achieves 81.2% of overall recognition rate.

1. INTRODUCTION

Facial expression is one of the most powerful, natural and immediate means for humans to share their emotions and intentions. Psychological studies focus on the interpretation on this mean to interact and describe that there are six basic emotions universally recognized [1], namely: joy, sadness, surprise, fear, anger and disgust. An automatic, efficient and accurate facial expression extraction system would thus be a powerfull tool assisting on these studies, allowing also other kinds of applications such as Human Computer Interface (HCI), smart interactive systems, video compression, etc. The proposed identity and facial expression recognition is based on the idea that it is straightforward for a human to capture the emotion and consequently the identity of a mimic, our someone known using makeup. Humans can understand both the identity/expression based only on facial motion. This guidance idea lead to face geometry that could used to recognize the identity and facial expression (focusing on the six basic emotions plus the neutral one). Laplacian EigenMaps (LE) [2] are nonlinear dimension reduction techniques that derive a low dimensional manifold lying in a higher dimensional more complex manifold. Such manifold is derived by embedding image data into a low dimensional space, where a image sequence is then represented as a trajectory in this feature space. Learning a manifold of this nature requires to derive discriminative facial representation from raw images. In this work, face images were represented by a set of 2D sparse feature points extracted using Active Appearance Models (AAM) [3] that is an effective way to locate facial features, modeling both shape and texture from an observed training set, being able to extract relevant face information without background interference. Both the identity and person-specific expression manifolds were learnt in a facial geometric feature space using LE. The recognition is based on a two stage cascade of classifiers. The first stage uses a multiclass Support Vector Machines (SVM) [4] that determines the identity. The second stage deals with the facial expression, being composed by a network of Hidden Markov Models (HMM) [5] displaced in a parallel architecture. For an input image, the AAM fitting framework extracts facial geometric related features and projects-it into the identity manifold. The first SVM stage predicts the identity and the respective person-specific model is loaded to stage two. Here the extracted features are projected into the expression manifold and the HMM based network decides the most likely facial expression. This paper is organised as follows: section 2 gives an introduction to the AAM, the building and fitting of such model. Section 3 briefly describes how to derive low dimensional manifolds using the LE. Section 4 briefly introduces HMM. Section 5 addresses to the architecture of the simultaneous identity and expression recognition system and sections 6 and 7 discusses experimental results.

This work was funded by FCT grant SFRH/BD/45178/2008.

2. ACTIVE APPEARANCE MODELS

Active Appearance Models (AAM) [3] are generative nonlinear parametric models of shape and texture, commonly used to model faces. These adaptive template matching methods, learn offline the variability of shape and texture that is captured from a representative training set, being able to fully describe with photorealistic quality the trained faces as well as unseen.

2.1. Shape and Texture Models

The shape of an AAM is defined by the vertex locations of a 2D triangulated mesh. Mathematically, the representation used for a single v-point shape is a $2v$ vector given by $\mathbf{s} = (x_1, y_1, \ldots, x_v, y_v)^T$. The AAM training data consists of a set of annotated images with the shape mesh marked (usually by hand). The shapes are then aligned to a common mean shape using a Generalised Procrustes Analysis (GPA), removing location, scale and rotation effects. Principal Components Analysis (PCA) are then applied to the aligned shapes, resulting on the parametric model

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i \tag{1}$$

where the new shapes, $\mathbf{s}$, are synthesised by deforming the mean shape, $\mathbf{s}_0$, using a weighted linear combination of eigenvectors, $\mathbf{s}_i$. n

is the number of eigenvectors that holds a user defined variance, typically 95%. p_i is a vector of shape parameters which represents the weights. Building a texture model, requires warping each training image so that the control points match those of the mean shape, $\mathbf{s}_0$. This texture mapping procedure is performed, using a piece wise affine warp, i.e. partitioning the convex hull of the mean shape by a set of triangles using the Delaunay triangulation. Each pixel inside a triangle is mapped into the correspondent triangle in the mean shape using barycentric coordinates. A texture model is obtained by applying a low-memory PCA on the normalized textures. Defining pixel coordinates as $\mathbf{x} = (x, y)^T$, the appearance of the AAM is an image, A($\mathbf{x}$), defined over the pixels $\mathbf{x} \in \mathbf{s}_0$ such as $A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x}), \quad \mathbf{x} \in \mathbf{s}_0$. The appearance A($\mathbf{x}$) can be expressed as a base appearance $A_0(\mathbf{x})$ plus a linear combination of m appearance images $A_i(\mathbf{x})$ (EigenFaces). The coefficients λ_i are the appearance parameters.

2.2. Model Fitting

Fitting an AAM is usually formulated [6] as minimizing the texture error, in the least square sense, between the model instance A($\mathbf{x}$) and the input backwarped image onto the base mesh $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$,

$$\sum_{\mathbf{x} \in \mathbf{s}_0} \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p})) \right]^2 . \quad (2)$$

In eq. 2 the warp $\mathbf{W}$ is the piecewise affine warp from the base mesh $\mathbf{s}_0$ to the current AAM shape $\mathbf{s}$. Hence, $\mathbf{W}$ is a function of the shape parameters $\mathbf{p}$. Notice that, the shape normalization on the model building process (Procrustes Analysis) the AAM do not model similarity transformations to the target image. Refer to [6] where it is shown how to include it on the warp $\mathbf{W}(\mathbf{x}; \mathbf{p})$.

The Simultaneous Inverse Compositional (SIC) [7] minimize eq. 2 by performing a Gauss-Newtow gradient descent optimization simultaneously on the warp parameters $\mathbf{p}$ and the appearance parameters $\boldsymbol{\lambda}$, with respect to $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$, updating the warp by inverse composition: $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta\mathbf{p})^{-1}$ and the appearance parameters additively: $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$. Denoting, $\mathbf{q} = (\mathbf{p}^T | \boldsymbol{\lambda}^T)^T$, i.e. $\mathbf{q}$ is an $n+m$ dimensional vector containing the warp parameters $\mathbf{p}$ and the appearance $\boldsymbol{\lambda}$, the $m + n$ Steepest Descent images [7] are of the form

$$\mathbf{SD}_{SIC}(\mathbf{x}) = \left(\nabla A \frac{\partial \mathbf{W}}{\partial p_1}, \cdots, \nabla A \frac{\partial \mathbf{W}}{\partial p_n}, A_1(\mathbf{x}), \cdots, A_m(\mathbf{x}) \right) \quad (3)$$

where ∇A is defined as $\nabla A = \nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i$. The parameters update is computed as

$$\Delta\mathbf{q} = -H_{SIC}^{-1} \sum_{\mathbf{x} \in \mathbf{s}_0} \mathbf{SD}_{SIC}^T(\mathbf{x}) E(\mathbf{x}) \quad (4)$$

where H_{SIC} is the Gauss-Newtow approximation of the Hessian given by

$$H_{SIC} = \sum_{\mathbf{x} \in \mathbf{s}_0} \mathbf{SD}_{SIC}^T(\mathbf{x}) \mathbf{SD}_{SIC}(\mathbf{x}), \quad (5)$$

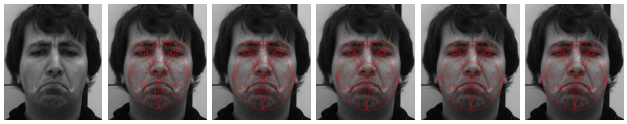
and the error image,$E(\mathbf{x})$, is defined as

$$E(\mathbf{x}) = I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) \right] . \quad (6)$$

The Simultaneous Inverse Compositional, when compared with other fitting approaches, such as the Project-Out [6] or the precomputed numerical estimate [3], work rather slow, since the Steepest Descent images depend on the appearance parameters and they have to re-computed in every iteration. On the other hand, SIC achieves the better fitting accuracy which is desirable for our proposes. Starting with a given estimate for the model, $\mathbf{q}_0$, and a rough estimate of the location of the face (provided by AdaBoost [8] method), an AAM model can be fitted with SIC following the algorithm 1. Figure 1 shows an example of AAM fitting into a target image.

Algorithm 1 Simultaneous Inverse Compositional Image Alignment

1: Evaluate the gradients ∇A_0 and ∇A_i for $i = 1, \cdots, m$
2: Evaluate the Jacobian of the warp $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$
3: **while** MaxIterations reached or $|\Delta\mathbf{q}| < \varepsilon$ **do**
4: Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
5: Compute the error image, $E(\mathbf{x})$, using eq. 6
6: Compute the Steepest Descent images, $\mathbf{SD}(\mathbf{x})$, using eq. 3
7: Compute the Hessian matrix, H, eq. 5
8: Compute the parameters updates, $\Delta\mathbf{q}$, with eq. 4
9: Inverse Compose the Warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta\mathbf{p})^{-1}$
10: Update the appearance parameters $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$
11: **end while**



(a) Input I (b) 1^{st} (c) 2^{nd} (d) 5^{th} (e) 12^{th} (f) Final

Fig. 1. AAM fitting.

3. LAPLACIAN EIGENMAPS

Laplacian EigenMaps (LE) [2] are nonlinear dimension reduction techniques that derive a low dimensional manifold lying in a higher dimensional more complex manifold. The LE builds a graph that incorporates neighborhood information of the dataset and using the notion of the Laplacian of the graph, computes a low dimensional representation that optimally preserves local neighborhood information. Given k feature points $\mathbf{x}_1, \cdots, \mathbf{x}_k \in \Re^l$,a weighted graph with k nodes is build, one for each point, with a set of edges connecting neighboring points. The embedding map is found by computing the eigenvectors of the graph Laplacian [2]. See algorithm 2 where this method is described. Finding such embedding map, Φ, requires tuning n nearest neighbors for graph building and select the number of dimensions, m, where the input features were projected into.

Algorithm 2 Laplacian EigenMaps

- Build the Adjacency Graph:
Nodes i and j (or j and i) are connected by an edge to the n nearest neighbors.
- Choosing the weights W_{ij}: (if i and j are connected by an edge) then $W_{ij} = 1$
- Build EigenMaps:
Compute eigenvalues and eigenvectors for the generalized eigenvector problem

$$L\mathbf{f} = \lambda D\mathbf{f} \quad (7)$$

where $D_{ii} = \sum_j W_{ji}$ is a diagonal weight matrix and $L = D - W$ is the Laplacian matrix. Let $\mathbf{f}_0, \cdots, \mathbf{f}_{k-1}$ be the solutions of eq. 7 order by eigenvalues $\lambda_0 = 0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$). Leaving out the eigenvector $\mathbf{f}_0$ corresponding to eigenvalue 0, the embedding m-dimensional Euclidian space is given by $\Phi = [\mathbf{f}_1 | \mathbf{f}_2 | \cdots | \mathbf{f}_m]$.

4. HIDDEN MARKOV MODELS

Hidden Markov Models (HMM) [5] have been widely used for many classication and modeling problems. One of the main advantages of

HMMs is their ability to model nonstationary signals or events. It uses the transition probabilities between the hidden states and learns the conditional probabilities of the observations given the state of the model. An HMM is given by the following set of parameters [5]:

$$A_{i,j} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq h \quad (8)$$

$$B = b_j(O_t) = P(O_t | q_t = S_j), 1 \leq j \leq h \quad (9)$$

$$\pi_i = P(q_1 = S_i) \quad (10)$$

where A is the state transition probability matrix, B is the observation probability distribution, and π is the initial probabilities. The number of (hidden) states of the HMM is given by h. Observations (O_t) at time t can be either discrete or continuous. In our case, the continuous case (the signal are measurements of the facial motion), will be given by the parameters of the probability distribution function of the observations (normally chosen to be the Gaussian distributions). $B \sim N(\mu_i, \Sigma_i), 1 \leq i \leq h$ where μ_i and Σ_i are the mean the full covariance matrix respectively.

5. IDENTITY AND FACIAL EXPRESSION RECOGNITION

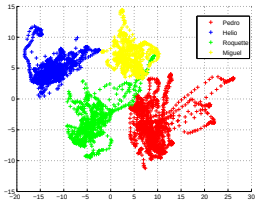
The proposed solution models both identity and facial expression in independent low dimensional manifolds, building person-specific expression models. The different manifolds were derived from embedding image data into a low dimensional subspace using Laplacian EigenMaps (LE) [2]. Learning these manifolds requires a discriminative facial representation from images, that is provided by the AAM fitting framework, see figure 1, where face images are represented by a set of sparse 2D feature point. As discriminatory features, insted of (x, y) feature points, there were used AAM related geometric features, i.e. regarding eq. 1 the shape parameters, $\mathbf{p}$, provide the same geometric information but using less dimensional features ($n << 2v$). Face normalization is done by only selecting the shape parameters that model deformation (ignoring the 4 similarity parameters, refer to [6]). Both identity,Φ_I, and person-specific expression manifolds Φ_{Ei} (with respect to subject i) were then learnt in a facial geometric feature space, consequently, a image sequence from a test subject describing a facial emotion (starting from the neutral expression, then exhibiting an emotion into a maximum of expressivity and returning back to the neutral) is represented as a trajectory in the learnt manifold. See figure 3. These manifolds were build using LE representations for the shape parameters (which are related to face geometry). This approach maps the shape parameters, $\mathbf{p}$, into a less dimensional space, i.e. $\mathbf{p}' = \Phi^T \mathbf{p}$, where the mapped features in our experiments acquire a huge discrimination power. As mentioned two kinds of LE manifolds were derived: The first type of manifold (lets call it identity manifold Φ_I) was built using data from all individuals, see figure 2; the second type (the expression manifold Φ_{Ei}) uses data only from a single individual, emphasising the differences in individual facial motion of the different expressions, see figure 3. This system holds an identity manifold and expression manifold for each of the individuals in the training set. For recognition proposes, an approach with two stage cascade of classifiers was used. The first stage deals with identity recognition (across expression changes) where a multiclass Support Vector Machines (SVM) [4] was trained with the identity manifold resulted data. The person-specific expression recognition, due the temporal dependence during the evolution of a facial emotion, is performed on the second stage using Hidden Markov Models (HMM) [5]. Seven HMM displaced in a parallel architecture were trained, each one specialized on the analysed expressions. Input observation sequences (expression manifold projected features) fed each one of the HMM and the final decision was based on the sequence that yielded the highest (forward-backward) probability.

Summarizing, the system has a feature extracting mechanism and a two stage cascade classifiers trained with embedded manifold data. For an input image, the AAM fitting framework extracts the normalized shape parameters, $\mathbf{p}$. These parameters are projected into the identity manifold $\mathbf{p}' = \Phi_I^T \mathbf{p}$, and the first SVM stage predict the identity i for the projected parameters $\mathbf{p}'$. The second stage loads the expression manifold, Φ_{Ei}, for the predicted identity. This stage consists on a network of seven HMM displaced in a parallel architecture. The input features are projected into the expression manifold, $\mathbf{p}'' = \Phi_{E_i}^T \mathbf{p}$ and the predicted expression is the one whose HMM model generated sequence yielded the highest probability.

6. EXPERIMENTAL RESULTS

For the purpose of this work, a Facial Dynamics Database was built. It consists of 4 individuals, in a frontal position, showing 7 different facial expressions, namely: neutral expression, happiness, sadness, surprise, anger, fear and disgust. All facial emotions were taken by starting and ending on the neutral expression. Each individual repeated all facial emotions four times. The dataset is formed by a total of 6770 images (640×480). The AAM model was build using a total of 28 images (7 images for each of the 4 person). Since the AAM will be used to fit every frame of the captured database, it should held as much shape variation as possible. The training images were then composed by the most expressive images of the 7 emotions (from a random repetition sequence). These training images were hand annotated using $v = 58$ landmarks. Training the model holding 95% of shape and appearance variance produces an AAM with $n = 18$ shape parameters and $m = 29$ EigenFaces. All the 6770 frames of the Facial Dynamics Database were then fitted using the AAM model, retrieving the shape parameters, $\mathbf{p}$, for each frame. Two main schemes were used for the manifold building: setting data for identity and setting the data for the expressions of each individual. A total of 5 manifolds were constructed (one identity manifold plus four individual-specific expression manifolds). These LE manifolds were build with both the number of adjacency graph neighbours, and the number of dimensions where the input features were projected into, found by cross-validation. Figure 2 and 3 shows the manifolds produced for the identity and expressions respectively. Regarding figure 3 it is noticed that person 1 (figure 3-top-right) is the most expressive and all facial emotions start and end from the neutral expression. This explains the high concentration of projected points over the neutral cluster. On the first stage, a multiclass SVM was trained with the input features of the identity manifold. The SVM classication was achieved using one-against-all voting scheme with a Gaussian Radial Basis Function (RBF) kernel. The kernel parameters and the missclassification penalty, were found also by cross-validation. Each individual-specific expression models in the second stage is composed by a network of seven HMM models displaced in a parallel architecture. These HMM models are specialized in each of the seven expressions. Regarding h as the of number hidden states from a given HMM, h Gaussians pdf were fitted on the low dimensional data of the respective expression using K-means. Then Baumk-Welch re-estimation was used to improve the parameters $(\pi_i, A_i, \mu_i, \Sigma_i)$ estimates. The optimal number of states, h, was found by cross-validation analysing the likelihood outputs on the re-estimation process. The final decision of the HMM network is made by evaluating the highest forward-backward probability on the sequence path provided by the Viterbi algorithm from all of the seven HMM. To evaluate the performance of the system the dataset

was divided into 4 fold for cross validation F1, F2, F3 and F4, that matches to the 4 repetitions of all expressions that each subject has made. The results shown are confusion matrices that were obtained from the cross-validation of the 4 folds (i.e. test F1, train F2,F3,F4; test F2, train F1,F3,F4; ...). Identity and expression models were evaluated independently. Figure 2-right displays results for the identity recognition and table 1 shows results for the expression models confusion matrices for each person in the dataset. Notice that, due the HMM based recognition, the results in table 1 could be misleading. Table 1 shows classification for each of the 6770 frames, but when the observations don't have length enough the HMM don't produce reliable results misclassifying many frames (that happends during the start of an emotion when no previous information is available). For this reason the HMM network decision at the end of each observation sequence (each full expression) is also shown (in blue at table 1).



	Indv1	Indv2	Indv3	Indv4
Indv1	**98.11**	0.09	1.79	0
Indv2	1.32	**98.67**	0	0
Indv3	2.93	0.29	**94.50**	2.27
Indv4	1.29	0.13	2.32	**96.25**

Overall recognition rate = 96.88%

Fig. 2. Left - Identity manifold learnt with geometric AAM related features for 4 persons. Right - Identity model confusion matrix.

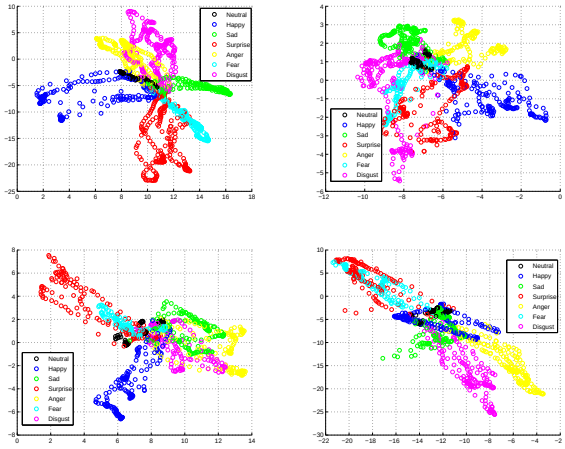


Fig. 3. Low dimensional manifolds learnt with geometric AAM related features for 4 persons exhibiting 7 expressions several turns each. Top-right, top-left, bottom-right and bottom-right figures represent the expression models for person 1, 2 ,3 and 4 respectively.

7. CONCLUSIONS

Human Identity and facial expression recognition were achieved using a two stage classifier approach using low dimensional representation of the geometry of the face. Facial geometry related features were extracted using the Active Appearance Models and low dimensional manifolds for identity and person-specific expression (Φ_I and Φ_{E_i}) were derived using LE. For an input image, the AAM fitting framework extracts the normalized shape parameters, $\mathbf{p}$, and the first SVM stage predicts the identity for the projected parameters $\mathbf{p}'$. The second stage is composed by a network of seven Hidden Markov Models displaced in parallel, each one specialized on the several facial emotions analysed. The normalized shape parameters are projected into the expression manifold of the predicted individual, $\mathbf{p}'' = \Phi_{E_i}^T \mathbf{p}$, and the predicted expression is the one whose HMM generated sequence yielded the highest probability. For evaluation proposes a database was build having 6770 images captured from 4 people exhibiting 7 different emotions. Our 4 fold cross-validation results show that the system is able to recognize an overall 96.8% in the identity. Since it was used person-specific expression models, the facial expression is independent on each individual. In our dataset the most expressive individual achieves an overall recognition rate of 81.2% and the less expressive 71.3%.

Table 1. Expression model confusion matrices for each one of the individuals (total of 6770 images).

Person 1	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**58.40** (3)	0 (0)	3.05 (0)	0 (0)	16.03 (1)	0 (0)	22.52 (0)
Happ	1.25 (0)	**95.00** (4)	0 (0)	0 (0)	3.75 (0)	0 (0)	0 (0)
Sad	0.59 (0)	0 (0)	**97.92** (4)	1.47 (0)	0 (0)	0 (0)	0 (0)
Surp	0 (0)	0 (0)	0.66 (0)	**99.34** (4)	0 (0)	0 (0)	0 (0)
Ang	0 (0)	2.69 (0)	5.09 (0)	0.59 (0)	**87.72** (4)	1.20 (0)	2.69 (0)
Fear	0 (0)	0 (0)	0 (0)	29.37 (1)	2.23 (0)	**68.40** (3)	0 (0)
Disg	0 (0)	0 (0)	4.14 (0)	0.95 (0)	32.80 (1)	0 (0)	**62.10** (3)

Overall recognition rate = 81.27%

Person 2	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**47.32** (2)	1.67 (0)	0 (0)	0 (0)	0 (0)	41.94 (2)	9.06 (0)
Happ	0 (0)	**70.34** (3)	0 (0)	25.85 (1)	0 (0)	3.80 (0)	0 (0)
3 **Sad**	0 (0)	0.35 (0)	**93.43** (4)	0.69 (0)	0 (0)	1.38 (0)	4.15 (0)
Surp	0 (0)	0 (0)	0.38 (0)	**97.32** (4)	0.76 (0)	1.53 (0)	0 (0)
Ang	0 (0)	0 (0)	1.84 (0)	0 (0)	**91.70** (4)	0.92 (0)	5.53 (0)
Fear	0 (0)	1.23 (0)	2.05 (0)	31.14 (2)	0 (0)	**61.47** (2)	4.09 (0)
Disg	15.30 (1)	0.65 (0)	0 (0)	9.12 (0)	1.95 (0)	2.93 (0)	**70.03** (3)

Overall recognition rate = 75.95%

Person 3	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**61.30** (2)	20.10 (1)	0 (0)	0 (0)	18.59 (1)	0 (0)	0 (0)
Happ	0.86 (0)	**96.53** (4)	0 (0)	2.59 (0)	0 (0)	0 (0)	0 (0)
Sad	0.44 (0)	0 (0)	**94.76** (4)	0.87 (0)	2.18 (0)	1.31 (0)	0.44 (0)
Surp	23.40 (1)	0 (0)	0 (0)	**76.06** (3)	0 (0)	0.53 (0)	0 (0)
Ang	0 (0)	2.38 (0)	2.38 (0)	1.90 (0)	**92.86** (4)	0.48 (0)	0 (0)
Fear	21.25 (1)	0 (0)	6.87 (0)	16.87 (1)	1.25 (0)	**46.25** (2)	7.50 (0)
Disg	27.03 (1)	1.35 (1)	16.21 (0)	0 (0)	9.46 (0)	1.35 (0)	**44.60** (2)

Overall recognition rate = 73.20%

Person 4	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**25.00** (1)	8.00 (0)	0 (0)	39.00 (2)	28.00 (1)	0 (0)	0 (0)
Happ	0 (0)	**95.79** (4)	0 (0)	0.47 (0)	0 (0)	0 (0)	3.74 (0)
Sad	1.51 (0)	30.65 (1)	**53.26** (3)	6.53 (0)	8.04 (0)	0 (0)	0 (0)
Surp	1.40 (0)	2.80 (0)	0 (0)	**66.35** (3)	2.80 (0)	26.63 (1)	0 (0)
Ang	0 (0)	0.87 (0)	2.19 (0)	4.82 (0)	**87.72** (4)	0.43 (0)	3.95 (0)
Fear	0 (0)	0 (0)	0 (0)	22.00 (1)	0 (0)	**80.00** (3)	0 (0)
Disg	0 (0)	0 (0)	0 (0)	6.47 (0)	0 (0)	0.49 (0)	**93.03** (4)

Overall recognition rate = 71.30%

8. REFERENCES

[1] *Handbook of Cognition and Emotion*, John Wiley & Sons Ltd, 1999.

[2] Partha Niyogi Mikhail Belkin, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 2003.

[3] G.J. Edwards T.F.Cootes and C.J.Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[4] *The Nature of Statistical Learning Theory*, Springer-Verlag, N.Y., 1995.

[5] L.R. Rabiner, "A tutorial on hidden markov models and selected applications in speech processing," *Proceedings of IEEE*, 1989.

[6] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, 2004.

[7] I.Matthews S. Baker, R.Gross, "Lucas kanade 20 years on: A unifying framework: Part 3," Tech. Rep., Carnegie Mellon University Robotics Institute, 2003.

[8] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascate of simple features," in *Computer Vision and Pattern Recognition*.