SIMULTANEOUS REGISTRATION AND CLUSTERING FOR TEMPORAL SEGMENTATION OF FACIAL GESTURES FROM VIDEO

Fernando De la Torre, Joan Campoy, Jeffrey F. Cohn and Takeo Kanade
Robotics Institute, Carnegie Mellon University, Pittsburgh, USA

Keywords: Facial expression analysis, Clustering, Facial Gesture, Learning, Temporal segmentation.

Abstract: Temporal segmentation of facial gestures from video sequences is an important unsolved problem for automatic facial analysis. Recovering temporal gesture structure from a set of 2D facial features tracked points is a challenging problem because of the difficulty of factorizing rigid and non-rigid motion and the large variability in the temporal scale of the facial gestures. In this paper, we propose a two step approach for temporal segmentation of facial gestures. The first step consist on clustering shape and appearance features into a number of clusters and the second step involves temporally grouping these clusters.
Results on clustering largely depend on the registration process. To improve the clustering/registration, we propose a Parameterized Cluster Analysis (PaCA) method that jointly performs registration and clustering. Besides the joint clustering/registration, PaCA solves the rounding off problem of existing spectral graph methods for clustering. After the clustering is performed, we group sets of clusters into facial gestures. Several toy and real examples show the benefits of our approach for temporal facial gesture segmentation.

1 INTRODUCTION

Temporal segmentation of facial gestures from video sequences is an important unsolved problem towards automatic facial interpretation. Recovering temporal gesture structure from a set of 2D facial features tracked points is a challenging problem because of the difficulty of factorizing rigid and non-rigid motion and the variability of temporal scales for different facial gestures. This problem is particulary hard if the sequence contains subtle expression changes and strong pose changes (most real interesting video sequences). In this paper, we propose a two step approach to temporal segmentation of facial gestures. The first step groups the shape and appearance features of facial features into a given number of clusters. The second step finds the temporal grouping of these clusters (see fig. 1).

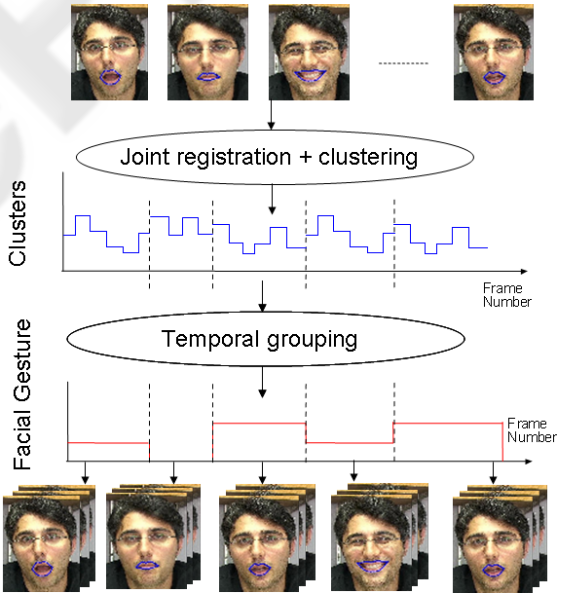

Figure 1: Temporal segmentation of facial gestures.

A key for the success of the clustering relies on the registration step. If the tracker do not explicitly track with a 3D model is usually hard to decouple rigid and non-rigid motion. In this paper, we propose Parameterized Cluster Analysis (PaCA), that jointly performs registration and clustering. Once the clustering is done, we propose a simple but effective way of discovering temporal structure in the set of clusters. Additionally, a new matrix formulation for clustering is introduced that enlightens connections between clustering methods.

De la Torre F., Campoy J., F. Cohn J. and Kanade T. (2007).
SIMULTANEOUS REGISTRATION AND CLUSTERING FOR TEMPORAL SEGMENTATION OF FACIAL GESTURES FROM VIDEO.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 110-115

2 PREVIOUS WORK

There has been quite substantial research efforts devoted to facial expression analysis over the past few years. Most of the work focus on tracking (Zhao and Chellappa, 2006), facial expression recognition or action units recognition (Lucey et al., 2006; Cohn et al., 2006). Little attention has been paid to the problem of temporal segmentation of facial gestures that can greatly benefit the recognition process. Exception is the pioneering work of Mase and Pentland (Mase and Pentland, 1990) that shows how zeros of the velocity of the facial motion parameters were found to be useful for the temporal segmentation and its applications to lip reading. Recently, Joey (Hoey, 2001) present a Multilevel Bayesian network for learning the dynamics of facial expression. In related work, Irani and Zelnik (Zelnik-Manor and Irani, 2004) propose a modification of factorization algorithms for structure from motion to provide temporal clustering of non-rigid motion.

Most previous work assumes an accurate registration process before the segmentation step. Accurate registration of the non-rigid facial features is still an open research problem (Zhao and Chellappa, 2006), in particular decoupling rigid and non-rigid motion from 2D. Unlike previous research, in this paper we propose an algorithm that jointly performs registration and clustering as a first step toward temporal segmentation of facial gestures. Moreover, we develop a simple but effective way to group these clusters into temporally coherent chunks.

3 MATRIX FORMULATION FOR CLUSTERING

In this section we review the state of the art in clustering algorithms using a new matrix formulation that enlightens the connection between several clustering methods and suggests new optimization schemes for spectral clustering.

3.1 K-means

K-means (MacQueen, 1967; Jain, 1988) is one of the simplest and most popular unsupervised learning algorithms to solve the clustering problem. Clustering refers to the partition of n data points into c disjoint clusters. k-means clustering splits a set of n objects into c groups by maximizing the between-clusters variation relative to within-cluster variation. That is, k-means clustering finds the partition of the data that is a local optimum of the following energy function: $J(\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_n) = \sum_{i=1}^{c}\sum_{j\in C_i}||\mathbf{d}_j - \boldsymbol{\mu}_i||_2^2$ where $\mathbf{d}_j$ (see notation [1]) is a vector representing the j^{th} data point and $\boldsymbol{\mu}_i$ is the geometric centroid of the data points for class i. The optimization criteria in previous eq. can be rewritten in matrix form as:

$$E_1(\mathbf{M},\mathbf{G}) = ||\mathbf{D}-\mathbf{M}\mathbf{G}^T||_F \qquad (1)$$
$$subject\ to\ \mathbf{G}\mathbf{1}_c = \mathbf{1}_n\ \ and\ \ g_{ij}\in\{0,1\}$$

where $\mathbf{G}\in\Re^{n\times c}$ and $\mathbf{M}\in\Re^{d\times c}$. $\mathbf{G}$ is a dummy indicator matrix, such that $\sum_j g_{ij}=1$, $g_{ij}\in\{0,1\}$ and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class C_j, c denotes the number of classes and n the number of samples. The columns of $\mathbf{D}\in\Re^{d\times n}$ contain the original data points, d is the dimension of the data. Recall that the equivalence between the k-means error function and eq. 1 is only valid if $\mathbf{G}$ strictly satisfies the constraints.

The k-means algorithm performs coordinate descent in $E_1(\mathbf{M},\mathbf{G})$. Given the actual value of the means $\mathbf{M}$, the first step finds for each data point $\mathbf{d}_j$, the $\mathbf{g}^j$ such that one of the columns is one and the rest 0 and minimizes eq. 1. The second step optimizes over $\mathbf{M}=\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$, equivalent to compute the mean of each cluster. Although it can be proven that alternating these two steps will always terminate, the k-means algorithm does not necessarily find the optimal configuration over all possible assignments. It typically runs multiple times and the best solution is chosen. Despite these limitations, the algorithm is used fairly frequently as a result of its ease of implementation and effectiveness.

Eliminating $\mathbf{M}$, eq. 1 can be rewritten as:

$$E_2(\mathbf{G}) = ||\mathbf{D}-\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T||_F = tr(\mathbf{D}^T\mathbf{D})$$
$$-tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}) \geq \sum_{i=c+1}^{min(d,n)}\lambda_i \qquad (2)$$

where λ_i are the eigenvalues of $\mathbf{D}^T\mathbf{D}$. Minimizing eq. 2 is equivalent to maximizing $tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G})$. Ignoring the special structure of $\mathbf{G}$ and considering the continuous domain, the optimum $\mathbf{G}$ value that optimizes eq. 2 is given by the eigenvectors of the covariance matrix $\mathbf{D}^T\mathbf{D}$ and the error is $E_2=\sum_{i=c+1}^{min(d,n)}\lambda_i$. A similar reasoning has been reported by (Ding and He, 2004; Zha et al., 2001),

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the j column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. All non-bold letters will represent variables of scalar nature. *diag* is an operator that transforms a vector to a diagonal matrix or takes the diagonal of the matrix into a vector. $\circ$ denotes the Hadamard or point-wise product. $\mathbf{1}_k\in\Re^{k\times 1}$ is a vector of ones. $\mathbf{I}_k\in\Re^{k\times k}$ is the identity matrix. $tr(\mathbf{A})=\sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$ and $|\mathbf{A}|$ denotes the determinant. $||\mathbf{A}||_F = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of a matrix.

where they show that a lower bound of eq. 2 is given by the residual eigenvalues. The continuous solution of $\mathbf{G}$ lies in the $c-1$ subspace spanned by the first $c-1$ eigenvectors with highest eigenvalues (Ding and He, 2004) of $\mathbf{D}^T\mathbf{D}$.

3.2 Spectral Clustering

Spectral graph methods for clustering are popular because of ease of programming and because they accomplishes a good trade-off between achieved performance and computational complexity. Recently, (Dhillon et al., 2004; de la Torre and Kanade, 2006) point out the connections between k-means and standard spectral graph algorithms, such as Normalized Cuts (Shi and Malik, 2000), by means of kernel methods. The kernel trick is a standard way of lifting the points of a dataset to a higher dimensional space, where points are more likely to be linearly separable (assuming that the right mapping is found). Let us consider a lifting of the original points to a higher dimensional space, $\Gamma = [\ \phi(\mathbf{d}_1)\ \phi(\mathbf{d}_2)\ \cdots\ \phi(\mathbf{d}_n)\ ]$ where ϕ is a high dimensional mapping. The kernelized version of eq. 1 will be:

$$E_3(\mathbf{M},\mathbf{G}) = ||(\Gamma - \mathbf{M}\mathbf{G}^T)\mathbf{W}||_F \quad (3)$$

where we have introduced a weighting matrix $\mathbf{W}$ for normalization purposes. Eliminating $\mathbf{M} = \Gamma\mathbf{W}\mathbf{W}^T\mathbf{G}(\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{G})^{-1}$, it can be shown that:

$$E_3 \propto -tr((\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{W}^T\Gamma^T\Gamma\mathbf{W}\mathbf{W}^T\mathbf{G}) \quad (4)$$

where $\Gamma^T\Gamma$ is the standard affinity matrix in Normalized Cuts (Shi and Malik, 2000). After a change of variable $\mathbf{Z} = \mathbf{G}^T\mathbf{W}$, the previous equation can be expressed as $E_3(\mathbf{Z}) \propto -tr((\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{W}^T\Gamma^T\Gamma\mathbf{W}\mathbf{Z}^T)$. Choosing $\mathbf{W} = diag(\Gamma^T\Gamma\mathbf{1}_n)^{-0.5}$ the problem is equivalent to solving the Normalized Cuts problem. Observe that this formulation is more general since it allows for arbitrary kernels and weights. Also, observe that the weight matrix could be used to reject the influence of a pair of data points with unknown similarity (i.e. missing data).

4 PARAMETERIZED CLUSTER ANALYSIS

Good registration is critical for segmentation of subtle facial gestures. However, decoupling the rigid and non-rigid motion of the face is a challenging problem even if 3D models are used. In this section, we propose PaCA that jointly performs clustering and registration and alleviates the registration problem for clustering.

4.1 Energy Function for PaCA

The key idea of PaCA is to parameterize the shape features Γ in the clustering function. This can be done easily by relating the clustering problem to an error function:

$$E_1(\mathbf{A},\mathbf{G},\mathbf{M}) = ||\Gamma - \mathbf{M}\mathbf{G}^T||_F \quad (5)$$

Unlike previous section, $\Gamma = \phi(\mathbf{T}(\mathbf{A},\mathbf{D})) = [\ \phi(\mathbf{A}_1\mathbf{d}_1)\ \phi(\mathbf{A}_2\mathbf{d}_2)\ \cdots\ \phi(\mathbf{A}_n\mathbf{d}_n)\ ]$ is a parameterized version of the data that accounts for mis-registrations. Each column in $\mathbf{T}(\mathbf{A},\mathbf{D}) \in \mathbf{R}^{d\times n}$, $\mathbf{t}_i$ represents the warped shape $\mathbf{A}_i\mathbf{d}_i$, where $\mathbf{A}_i$ is a linear transformation matrix with the motion parameters. ϕ is a generic mapping, usually to a higher dimensional space. The mapping can be infinite dimensional (e.g. Gaussian kernel).

After optimizing over $\mathbf{M} = \Gamma\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$, eq. 5 is equivalent to:

$$E_2(\mathbf{A},\mathbf{G}) = ||\Gamma(\mathbf{A})(\mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)||_F = tr\big(\Gamma(\mathbf{A})^T\Gamma(\mathbf{A})(\mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)\big) \quad (6)$$

taking into account that $(\mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)$ is a idempotent matrix. $K(\mathbf{A}) = \Gamma(\mathbf{A})^T\Gamma(\mathbf{A})$ is the standard affinity matrix or kernel matrix, where each element in the case of exponential kernel is: $k_{ij} = e^{-\frac{||\mathbf{A}_i\mathbf{D}_i - \mathbf{A}_j\mathbf{D}_j||_F}{2\sigma}}$. Where $\mathbf{A}_i$ is an affinity matrix of 4 (if translation is removed), 6 or 8 parameters. $\mathbf{D}_i \in \mathbf{R}^{(d/2)\times n}$ is a data matrix, such that the first row contains the x-coordinates and the second row contains the y-coordinates.

4.2 Motion Models

In this paper, we will assume that in the video the face of the subject is relatively far away from the camera and that locally the eye region or mouth is a planar surface. It is well known (Adiv, 1985) that the 2D projected motion field of a 3D planar surface can be recovered under orthographic projection ($x = X$ and $y = Y$) by an affine model $\mathbf{f}(\mathbf{x},\mathbf{a})$, parameterized by $\mathbf{a} = [a_1\ a_2\ ...\ a_6]^T$:

$$\mathbf{f}(\mathbf{x},\mathbf{a}) = \begin{bmatrix} a_1 \\ a_4 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \quad (7)$$

where $\mathbf{x}_c = (x_c, y_c)^T$ is the center position of the object.

4.3 Solving the Optimization Problem

Assuming the matrix $\mathbf{A}$ is known, optimizing eq. 6 reduces to:

$$E_5(\mathbf{G}) \propto tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{K}\mathbf{G}) \quad (8)$$

To impose non-negativity constraints in g_{ij}, we parameterize $\mathbf{G}$ as the product of two matrices $\mathbf{G} = \mathbf{V} \circ \mathbf{V}$ (de la Torre and Kanade, 2006) and use a gradient descent strategy to search for an optimum:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \eta_1 \frac{\partial G(\mathbf{V}^n)}{\partial \mathbf{V}} \quad (9)$$
$$\frac{\partial G(\mathbf{V}^n)}{\partial \mathbf{V}} = (\mathbf{I}_c - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)\mathbf{K}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \circ \mathbf{V}$$

The major problem with the update of eq. 9 is to determine the optimal η_1. In our case, η_1 is determined with a line search strategy. To impose $\mathbf{G}\mathbf{1}_c = \mathbf{1}_n$ in each iteration, the $\mathbf{V}$ is normalized to satisfy the constraint. Because eq. 9 is prone to local minima, we start from several random initial points and select the solution with minimum error.

Assuming $\mathbf{G}$ is known optimizing w.r.t. $\mathbf{A}$ has to minimize: $E_3(\mathbf{A}) = tr(\mathbf{K}(\mathbf{A})\mathbf{F})$ where, $F = (\mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)$. To optimize w.r.t. $\mathbf{A}$ we use a linear time algorithm that uses gradient descent. In the case of exponential kernel,

$$\mathbf{A}^{n+1} = \mathbf{A}^n - \eta_2 \frac{\partial E_2}{\partial \mathbf{A}} \quad (10)$$
$$\frac{\partial E}{\partial \mathbf{A}_i} = -\sum_{j=1}^{n} \frac{f_{ji}}{\sigma} e^{-\frac{\|\mathbf{A}_i\mathbf{D}_i - \mathbf{A}_j\mathbf{D}_j\|_F}{2\sigma}} (\mathbf{A}_i\mathbf{D}_i - \mathbf{A}_j\mathbf{D}_j)\mathbf{D}_j^T$$

As before, η_2 is determined with a line search strategy (Fletcher, 1987).

4.4 Initialization and Clustering Features

Optimizing eq. 6 w.r.t $\mathbf{A}$ and $\mathbf{G}$ is a non-convex problem prone to local minima, that without a good initialization is likely to get stuck into a bad minimum. To give an initial estimate of the matrix $\mathbf{K}$ we compute all possible pairwise affine distances between the set of shape points and with this estimate optimize over $\mathbf{G}$. Observe that at this point $\mathbf{K}$ is symmetric but not necessarily definite positive.

We assume that several facial features of the face have been tracked using Active Appearance Models (AAM) (Matthews and Baker, 2004) (see fig. 6). Once the facial feature points have been tracked, we use PaCA to jointly cluster and register the shape. However, using only the shape as the only feature is not very reliable for capturing subtle facial gestures. For instance, we can have two completely different gestures with the same shape (see fig. 2 bottom). To compensate for this effect, we also incorporate appearance features. The appearance features are extracted by a geometric invariant histogram recently introduced (Domke and Aloimonos, 2006). We can decouple the effects of registration in the appearance representation since the histogram proposed in (Domke and Aloimonos, 2006) is invariant to perspective transformations (see fig. 2). During the clustering process we over-sample the shape feaures to achieve robustness against noise(see fig. 2).

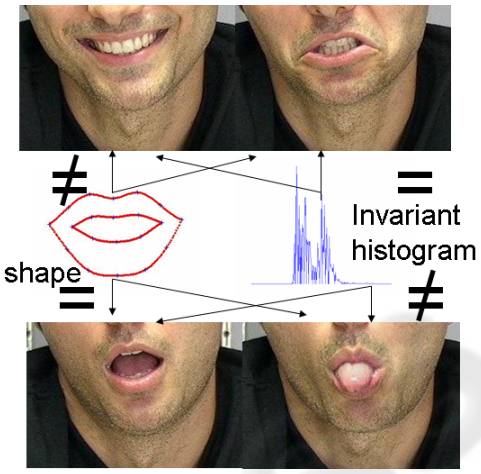

Figure 2: Features used in temporal segmentation.

5 DISCOVERING TEMPORAL CLUSTERS

Once the the facial features have been clustered into coherent shape/appearance clusters, the goal is to group the clusters into facial gestures. In this section, we propose a simple but effective method to search for temporal coherent clusters.

5.1 Removing Temporal Redundancy

In a first step, we automatically detect all neutral expressions (i.e. action unit 0-AU0) (Cohn et al., 2006) since is usually the most common facial "cluster" and useful in many recognition tasks. The algorithm to detect the AU0 works as follows. First, we compute a normalized error between the shape/appearance at time t and time $t-1$. A two-state Hidden Markov Model (HMM) is used to temporally segment the time instants that contain appearance/shape changes. The transition probabilities in the HMM are computed using a logistic regression function (i.e. $\frac{1}{1+e^{-\beta x}}$ and $\frac{1}{1+e^{-\beta(x+\tau)}}$), where β, τ are parameters computed from the error histogram. To find a maximum a posteriori solution, the standard Viterbi algorithm (dynamic programming) is executed. In the state representing still configurations of the face there are examples of AU0 and examples of other AU that are static for few frames. In the next step, we separate

these two cases by performing spectral clustering with shape/appearance features. All the clusters such that the average mean aperture of the mouth is smaller that a threshold are classified as AU0. Fig 3 illustrates the process.

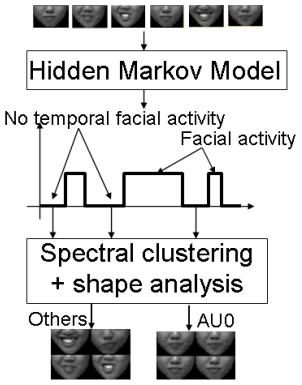

Figure 3: Process to automatically detect AU0.

The second important step towards discovering temporal clusters is to achieve temporal invariance to the speed of the facial gesture. Towards this end, we first remove all the consecutive clusters that are the same and just the changes among consecutive clusters will remain. After this process is done, the video is reduced to about $10-20\%$ its original length.

5.2 Temporal Correlation to Discover Facial Gestures

Once we have simplified the temporal representation of the video sequence, we are ready to find temporal patterns of different lengths in the video sequence. Since we have substantially reduced the amount of temporal data available, we use an exhaustive approach to search over all possible cluster sequences of different lengths and in the sequence to find the same temporal pattern.

The algorithm starts selecting long patterns (usually $8-9$ clusters), and it searches over the whole sequence for peaks of the normalized correlation. All the peaks that have normalized correlation 1 (i.e. is the same pattern) are removed from the sequence, later the rest of the patterns with smaller length are iteratively discovered with the same approach.

Fig. 4 shows how the algorithm works in synthetic data. We have made a sequence with three temporal clusters of length 4 (fig. 4.a and 4.b). The algorithm automatically discovers that there are 3 temporal clusters and correctly identifies them.

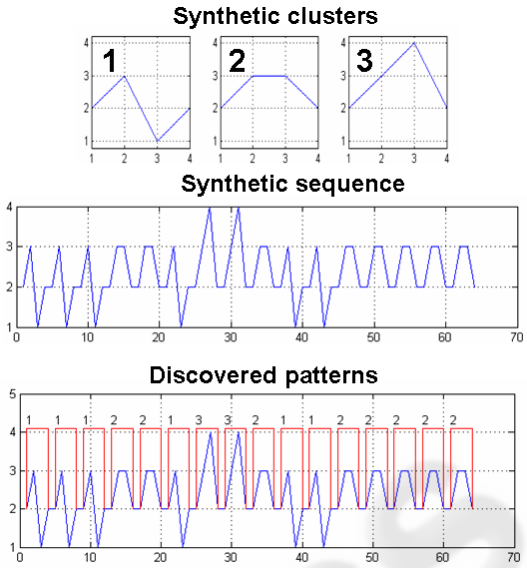

Figure 4: a) 3 synthetic clusters b) synthetic sequence c) Temporal clusters found with our algorithm.

6 EXPERIMENTS

In this section we report preliminary experiments with synthetic and real data.

6.1 Synthetic Data

We have synthetically created three different shape prototypes (fig. 5.b) and perturbed them with 50 random affine transformations (fig. 5.a). After running PaCA we can see the mean of the shape for each cluster in the second row of fig. 5 is correctly recovered. PaCA has correctly clustered the original shapes.

Figure 5: First row: superimposed perturbed shapes for each cluster. Second row: superimposed aligned shapes. Also original prototypes.

6.2 Expression Segmentation

In this experiment, we have recorded a video sequence where the face of the subject is naturally making five different facial gestures (sad, taking out the tongue, speaking, smiling, and neutral). We use AAM

to track the sequence (see fig. 6). After the tracking is done, we automatically detect the AU0 and remove the temporal redundancy in the cluster sequence, reducing the sequence to 20% its original length (see fig. 7.a and 7.b). Later, the temporal segmentation algorithm discovers the facial gestures shown in 7.c. Observe that there are some time windows that are not classified, this windows correspond to one time or unusual facial gestures. We have visually checked the correctness of our approach. The results in video can be downloaded from *www.cs.cmu.edu/ ∼ ftorre/ExpressionSegmentation.avi*.

Figure 6: AAM tracking across several frames.

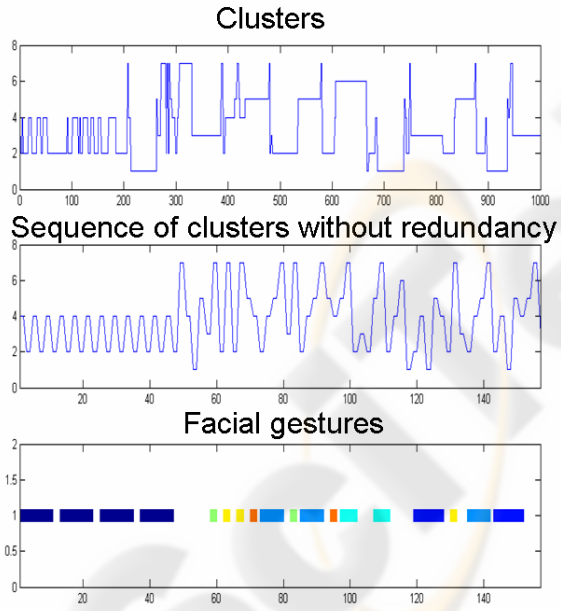

Figure 7: a) Original sequence of clusters. b) Sequence of clusters with just the transitions. c) Discovered facial gestures.

REFERENCES

Adiv, G. (1985). Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401.

Cohn, J., Ambadar, Z., and Ekman, P. (2006). Observer-based measurement of facial expression with the facial action coding system.

de la Torre, F. and Kanade, T. (2006). Discriminative cluster analysis. In *International Conference on Machine Learning*.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). A unified view of kernel k-means, spectral clustering and graph partitioning. In *UTCS Technical Report TR-04-25*.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *International Conference on Machine Learning*, volume 1.

Domke, J. and Aloimonos, Y. (2006). Deformation and viewpoint invariant color histograms. In *BMVC*.

Fletcher, R. (1987). *Practical methods of optimization*. John Wiley and Sons.

Hoey, J. (2001). Hierarchical unsupervised learning of facial expression categories. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on s*.

Jain, A. K. (1988). *Algorithms For Clustering Data*. Prentice Hall.

Lucey, S., Matthews, I., Hu, C., Ambadar, Z., de la Torre, F., and Cohn, J. (2006). AAM derived face representations for robust facial action recognition. In *Proc. International Conference on Automatic Face and Gesture Recognition*.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *5-th Berkeley Symposium on Mathematical Statistics and Probability.Berkeley, University of California Press.*, pages 1:281–297.

Mase, K. and Pentland, A. (1990). Automatic lipreading by computer. (J73-D-II(6)):796–803.

Matthews, I. and Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8).

Zelnik-Manor, L. and Irani, M. (2004). Temporal factorization vs. spatial factorization. In *ECCV*.

Zha, H., Ding, C., Gu, M., He, X., and Simon., H. (2001). Spectral relaxation for k-means clustering. In *Neural Information Processing Systems*, pages 1057–1064.

Zhao, W. and Chellappa, R. (2006). *(Editors). Face Processing: Advanced Modeling and Methods*. Elsevier.