# Temporal Segmentation of Facial Behavior

**Fernando De la Torre† Joan Campoy† Zara Ambadar‡ Jeffrey F. Cohn‡**
†, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.
ftorre@cs.cmu.edu jcampoy@andrew.cmu.edu
‡, University of Pittsburgh. Department of Psychology. Pittsburgh, Pennsylvania 15260.
ambadar@pitt.edu jeffcohn@pitt.edu

## Abstract

*Temporal segmentation of facial gestures in spontaneous facial behavior recorded in real-world settings is an important, unsolved, and relatively unexplored problem in facial image analysis. Several issues contribute to the challenge of this task. These include non-frontal pose, moderate to large out-of-plane head motion, large variability in the temporal scale of facial gestures, and the exponential nature of possible facial action combinations. To address these challenges, we propose a two-step approach to temporally segment facial behavior. The first step uses spectral graph techniques to cluster shape and appearance features invariant to some geometric transformations. The second step groups the clusters into temporally coherent facial gestures. We evaluated this method in facial behavior recorded during face-to-face interactions. The video data were originally collected to answer substantive questions in psychology without concern for algorithm development. The method achieved moderate convergent validity with manual FACS (Facial Action Coding System) annotation. Further, when used to preprocess video for manual FACS annotation, the method significantly improves productivity, thus addressing the need for ground-truth data for facial image analysis. Moreover, we were also able to detect unusual facial behavior.*

## 1. Introduction

Temporal segmentation of facial behavior from video is an important unsolved problem in automatic facial image analysis. With few exceptions, previous literature has treated video frames as if they were independent, ignoring their temporal organization. Facial actions have an onset, one or more peaks, and offsets, and the temporal organization of these events is critical to facial expression understanding and perception [2, 5, 6]. For automatic facial image analysis, temporal segmentation is critical to decomposing facial behavior into action units (AUs) and higher-order combinations or expressions [5], to improving recognition performance of facial expression recognizers, and to detecting unusual expressions, among other applications.

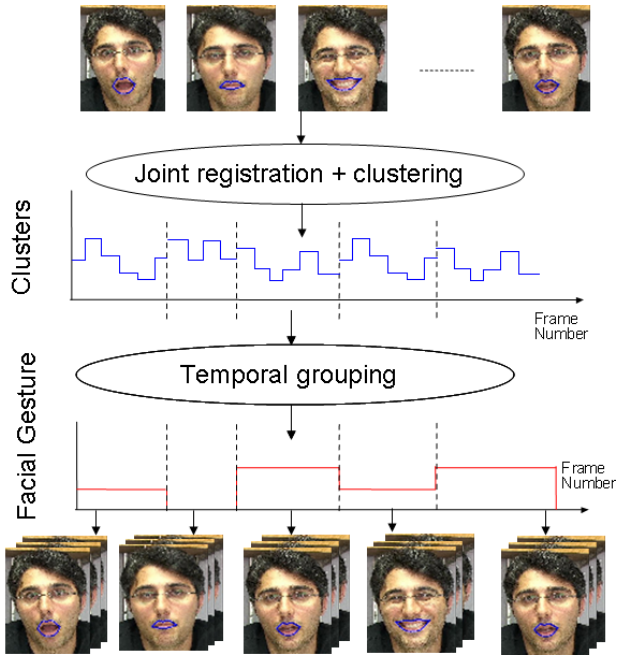


Figure 1. Temporal segmentation of facial gestures.

Several factors make the task of recovering the temporal structure of facial behavior from video a challenging topic, especially when video is obtained in realistic settings characterized by non-frontal pose, moderate out-of-plane head motion, subtle facial actions, large variability in the temporal scale of facial actions (both within and between event classes) and an exponential number of possible facial action combinations. To address these challenges, we propose a two-step approach to temporally segment facial behavior. The first step uses spectral graph techniques to cluster shape and appearance features. The resultant clusters are invariant to some geometric transformations. The second step groups the clusters into temporally coherent facial gestures (fig. 1).

This paper is organized as follows. Section 2 reviews previous work on facial expression recognition and temporal segmentation. Section 3 reviews state of the art in clustering algorithms. Section 4 proposes a method to discover temporal clusters. Section 5 presents experimental results and two novel applications of the method. One detects unusual or rare facial actions; the other increases the efficiency of manual FACS annotation by preprocessing video using automatic segmentation.

## 2. Previous work

There has been substantial effort devoted to automatic facial image analysis over the past decade. Major topics include facial feature tracking, facial expression analysis, and face recognition[36, 26, 22]. Facial expression refers to both emotion-specified expressions (e.g., happy or sad) and anatomically based facial actions [14]. Comprehensive reviews of automatic facial may be found in [30, 36, 22, 33]. Here we briefly review literature most relevant to the current study.

The pioneering work of Black and Yacoob [3] recognizes facial expressions by fitting local parametric motion models to regions of the face and then feeding the resulting parameters to a nearest neighbor classifier for expression recognition. De la Torre et al. [10] use condensation and appearance models to simultaneously track and recognize facial expression. Chang et al. [18] use a low dimensional Leipschitz embedding to build a manifold of shape variation across several people and then use I-condensation to simultaneously track and recognize expressions. Lee and Elgammal [21] use multi-linear models to construct a non-linear manifold that factorizes identity from expression. Littleworth et al. [23] learn an appearance classifier for facial expression recognition. Shape and appearance features are common to most work on this topic. More recently, investigators have proposed use of dynamic features in addition to those of shape and appearance to recognize facial expressions and actions [4, 7, 29]. Dynamics is relevant to temporal segmentation, in which the timing of facial actions (e.g., start, peak, and stop) must be parsed from the stream of behavior.

With few exceptions previous work on expression or action unit recognition is supervised in nature (i.e. there is a training set manually labeled) and little attention has been paid to the problem of unsupervised temporal segmentation prior to recognition. Manual segmentation is feasible for constrained applications, such as in supervised learning. In a pioneering study, Mase and Pentland [27] found that zero crossings in the velocity contour of facial motion are useful for temporal segmentation of visual speech. Recently, Hoey [17] present a multilevel Bayesian network to learn the dynamics of facial expression. Irani and Zelnik [34] propose a modification of structure-from-motion factorization to temporally segment rigid and non-rigid facial motion.

These approaches all assume accurate registration prior to segmentation. Accurate registration of non-rigid facial features, however, is still an open research problem [36]. Especially for 2D image data, factorizing rigid from non-rigid motion is a challenging problem. To solve this problem without recourse to 3D data and modeling, we propose a clustering algorithm that is invariant to specific geometric transformations. This is the first step toward temporal segmentation of facial actions. We then propose an algorithm to group clusters effectively into temporally coherent chunks. We show the benefits of our approach in two novel applications. In one, we detect unusual or rare facial expressions and actions; in the other, we use the method to preprocess video for manual FACS coding. By temporally segmenting facial behavior, we increase the efficiency and reliability of manual FACS annotation.

## 3. Algorithms for clustering

In this section we review the state of the art in clustering algorithms. In this review, we use a new matrix formulation that enlightens the connections between clustering methods.

### 3.1. K-means

Clustering refers to the partition of $n$ data points into $c$ disjoint clusters. Among various approaches to unsupervised clustering, k-means [25, 19] is one of the simplest and most popular. k-means clustering splits a set of $n$ objects into $c$ groups by minimizing the within clusters variation. That is, k-means clustering finds the partition of the data that is a local optimum of the energy function:

$$J(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_n) = \sum_{i=1}^{c} \sum_{j \in C_i} ||\mathbf{d}_j - \boldsymbol{\mu}_i||_2^2 \tag{1}$$

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. 1 can be rewritten in matrix form as:

$$E_1(\mathbf{M}, \mathbf{G}) = ||\mathbf{D} - \mathbf{M}\mathbf{G}^T||_F \tag{2}$$
$$subject\ to\ \mathbf{G}\mathbf{1}_c = \mathbf{1}_n\ \ and\ \ g_{ij} \in \{0, 1\}$$

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 $\mathbf{d}_i$

[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 represent scalars. $diag$ 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.

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, and the columns of $\mathbf{M}$ represent the cluster centroids; $d$ is the dimension of the data. The equivalence between the k-means error function and eq. 2 is valid only 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 others 0, and it minimizes eq. 2. The second step optimizes over $\mathbf{M} = \mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$, which is equivalent to computing the mean of each cluster. Although it can be proven that alternating these two steps will always converge, the k-means algorithm does not necessarily find the optimal configuration over all possible assignments. The algorithm is typically run multiple times from different initial conditions with the best solution chosen. Despite limitations, the algorithm is used fairly frequently because of its ease of implementation and effectiveness.

One of the advantages of relating the clustering problem to an error function is the easy of deriving bounds. For instance, after eliminating $\mathbf{M}$, eq. 2 can be rewritten as:

$$E_2(\mathbf{G}) = ||\mathbf{D} - \mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T||_F = tr(\mathbf{D}^T\mathbf{D}) - tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}) \geq \sum_{i=c+1}^{min(d,n)} \lambda_i \quad (3)$$

where $\lambda_i$ are the eigenvalues of $\mathbf{D}^T\mathbf{D}$. Minimizing eq. 3 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 $\mathbf{G}$ value that minimizes eq. 3 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 [12, 35], where they show that a lower bound of eq. 3 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 [12] of $\mathbf{D}^T\mathbf{D}$.

### 3.2. Spectral graph clustering

Spectral graph clustering is popular because of its ease of programming and favorable trade-off between accuracy and computational complexity. Recently, [11, 8] pointed out similarities between k-means and standard spectral graph algorithms, such as Normalized Cuts [32], unifying both approaches by means of kernel methods. The kernel is a implicit way of "lifting" the points of a dataset to a higher dimensional space in which they may be linearly separable (assuming that such a mapping can be found). Let us consider a mapping of the original points to a higher dimensional space, $\mathbf{\Gamma} = [\ \boldsymbol{\phi}(\mathbf{d}_1)\ \boldsymbol{\phi}(\mathbf{d}_2)\ \cdots\ \boldsymbol{\phi}(\mathbf{d}_n)\ ]$ where $\boldsymbol{\phi}$ is a high dimensional mapping. The kernelized version of eq. 2 will be [8]:

$$E_3(\mathbf{M}, \mathbf{G}) = ||(\mathbf{\Gamma} - \mathbf{M}\mathbf{G}^T)\mathbf{W}||_F \quad (4)$$

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

$$E_3 \propto -tr((\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{W}\mathbf{W}^T\mathbf{G}) \quad (5)$$

where $\mathbf{\Gamma}^T\mathbf{\Gamma}$ is the standard affinity matrix in Normalized Cuts [32]. After a change of variable $\mathbf{Z} = \mathbf{G}^T\mathbf{W}$, the previous equation can be expressed as $E_3(\mathbf{Z}) \propto -tr((\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{W}^T\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{W}\mathbf{Z}^T)$. Choosing $\mathbf{W} = diag(\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{1}_n)^{-0.5}$ 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).

### 3.3. Invariant clustering

In practice, most spectral graph methods (e.g. Normalized Cuts [32]) compute the eigenvector of the normalized affinity matrix $diag(\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{1}_n)^{-0.5}\mathbf{\Gamma}^T\mathbf{\Gamma}diag(\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{1}_n)^{-0.5}$. Eigenvectors computed in this way provide an embedding better suited for clustering with standard algorithms such as k-means [32]. Given a video with a set of tracked facial features, success of temporal segmentation depends in part on the ability to compute a set of clusters that are invariant to geometric transformations. In this section, we show how this might be accomplished for shape and appearance features.

We assume that facial features have been tracked using Active Appearance Models (AAMs) [28, 9] (see fig. 6). Given the fiduciary points of the AAM we interpolate between points with a spline curve, which results in a non-uniform sampling of the shape (see fig. 2 for an example in the mouth region). Given the interpolated tracked shape, we then estimate the affinity matrix $\mathbf{K} = \mathbf{\Gamma}^T\mathbf{\Gamma}$ by computing all possible pairwise distances between the samples. To compensate for in-plane rigid motion, we remove the similarity transform between all possible pairs of shapes (i.e. set of points tracked by the AAM) at different time instances. That is, given the shape at times 1 and 2, $\mathbf{s}_1$ and $\mathbf{s}_2$, $k_{12} = e^{\frac{-||\mathbf{S}_1 - \mathbf{H}\mathbf{S}_2||_2^2}{2\sigma_s^2}}$, where $\mathbf{H} = \begin{pmatrix} r\cos\alpha & r\sin\alpha & tx \\ -r\sin\alpha & r\cos\alpha & ty \\ 0 & 0 & 1 \end{pmatrix}$ is a matrix with 4 parameters ($x$ and $y$ translation, rotation, scale). $\mathbf{H}$ is optimally computed for each pair of shapes. A more general $\mathbf{H}$ can represent an homography or an affine transformation that can compensate for out-of-plane rotations. It is well known [1] 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.

Observe that $\mathbf{K}$ is symmetric but there is no guarantee that it is positive definite, which could cause degenerate so-

lutions. However, in our experimental results we encounter no problems of this type.

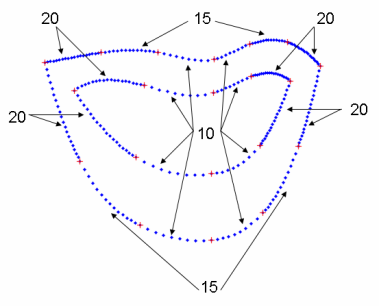


Figure 2. Number of samples in each segment.

Shape features alone are unlikely to capture differences between subtle facial gestures. For instance, we can have two completely different gestures with the same shape (see fig. 3 bottom). To compensate for this effect, we incorporate appearance features. The appearance features are extracted by a geometric invariant histogram recently introduced in [13]. We can decouple the effects of registration in the appearance representation since the histogram proposed in [13] is invariant to perspective transformations (see fig. 3). The final affinity matrix is $k_{ij} = e^{\frac{-||\mathbf{S}_i - \mathbf{H}_{ij}\mathbf{S}_j||_2^2}{2\sigma_s^2}} * e^{\frac{-||\mathbf{h}_i - \mathbf{h}_j||_2^2}{2\sigma_a^2}}$, where $\mathbf{h}_i$ is the invariant histogram of $i^{th}$ sample (in a given region) and $\sigma_a$ is the standard deviation of the appearance invariant histogram.

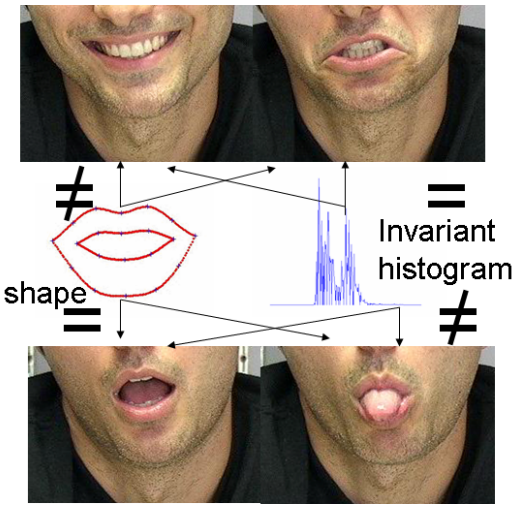


Figure 3. Features used for temporal segmentation.

## 4. Discovering temporal clusters

Once the facial features have been clustered into coherent shape/appearance clusters, the goal is to group them into a set of dynamic facial gestures (sets of consecutive clusters that occur more than p times, where p is a user specified criterion). In this section, we propose a simple but effective method to search for temporal coherent clusters.

### 4.1. Removing temporal redundancy

In a first step, we automatically detect all neutral expressions (i.e. AU 0 in FACS) [5] because they are usually the most common facial "cluster" and are useful in many recognition tasks. To detect subtle facial actions, for instance, it is necessary to compute the difference between a neutral and target image [24]. The algorithm to detect AU0 works as follows: first, we compute the 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. For state 0, representing no-change, the probability is given by $\frac{1}{1+e^{-\beta x}}$ and similarly for the other state $\frac{1}{1+e^{-\beta(x+\tau)}}$ ). $x$ is the normalized error and $\beta, \tau$ are parameters computed from the error histogram. To find a maximum a *posteriori* solution, the standard Viterbi algorithm (dynamic programming) is executed. This identifies a set of static facial expressions, for which there is no movement for two or more frames. This set includes AU 0 as well as other action units. In the next step, we separate AU 0 from other AUs by performing spectral clustering of the shape/appearance features. The cluster that has an average mean aperture of the mouth smaller than a threshold and contains the larger number of samples is classified as AU0. Fig 4 illustrates the process.

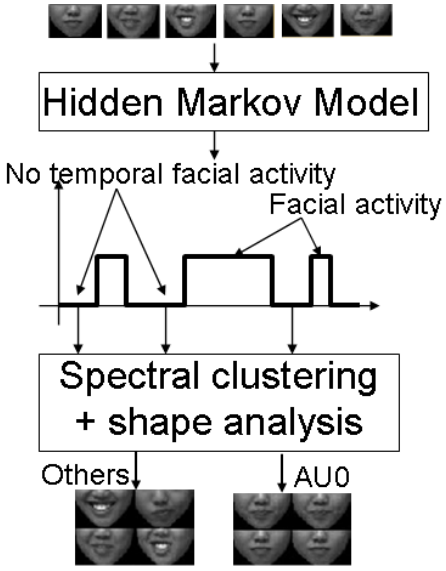


Figure 4. Process to automatically detect AU0.

The second step in discovering temporal clusters is to achieve temporal invariance to the speed of the facial ges-

ture. Towards this end, we first remove all the consecutive frames that belong to the same cluster. In this way only the changes in cluster state are preserved. Once this process is completed, the video is reduced in length to about $10-20\%$ its original size. See fig. 8.a and 8.b .

### 4.2. 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. The algorithm starts selecting long patterns (usually $8-9$ consecutive clusters) as templates. Then, it computes normalized correlation of each of the templates with the sequence. All the instances that have normalized correlation of 1 (i.e. same pattern as the template) are removed from the sequence. If the data is too noisy, smaller thresholds than 1 can be imposed. After that, the algorithm selects smaller templates (typically one cluster less), searches again for all instances of normalized correlation 1, and proceeds this way until all the frames have been searched.

Fig. 5 shows how the algorithm works on synthetic data containing three temporal clusters of length 4 (fig. 5.a and 5.b). The algorithm automatically discovers the 3 temporal clusters.

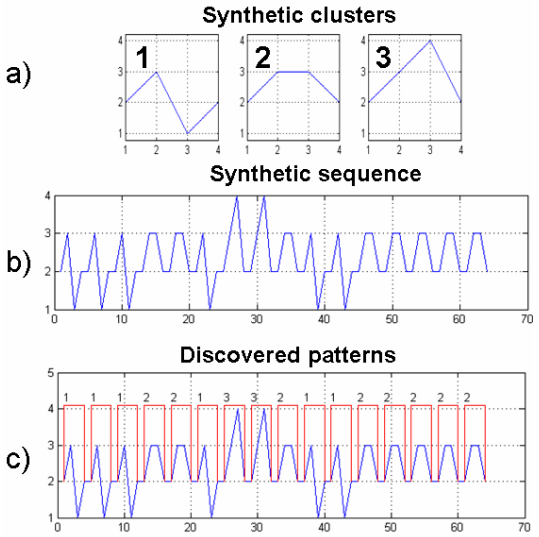


Figure 5. a) 3 synthetic clusters b) Synthetic sequence c) Temporal clusters found by our algorithm.

## 5. Experiments

We evaluated the algorithm two ways. One, we tested its ability to temporally segment facial gestures and identify ones that occur rarely. Two, we used it to preprocess video of spontaneous facial expression intended for FACS annotation.

### 5.1. Temporal segmentation of mouth events

In this experiment, we have recorded a video sequence in which the subject spontaneously made five different facial gestures (sad, sticking out the tongue, speaking, smiling, and neutral). We use person-specific Active Appearance Models [28, 9] to track the non-rigid/rigid motion in the sequence (see fig. 6).

Figure 6. AAM tracking across several frames.

After using AAM to track the video sequence, we use the algorithm proposed in Sections 4.1 and 4.2 to identify clusters (see fig. 7 for AU0) and remove temporal redundancy from the video sequence. By eliminating consecutive frames that have the same cluster label, sequence length is reduced to $20\%$ of the original length (see fig. 8.a and 8.b). Then, the temporal segmentation algorithm discovers the facial gestures shown in 8.c. Observe that there are some time windows that remain unclassified. These windows correspond to gestures lasting only a single frame or ones that are unusual or infrequent.

Figure 7. Examples of detected AU0.

Accuracy of the clustering approach was confirmed by visual inspection. The results of the video can be downloaded from $www.cs.cmu.edu/\tilde{}ftorre/seg\_facial\_behavior.avi$. Fig. 9 shows one frame of the output video resulting from finding the temporal clusters in the video sequence. Each frame of the video contains three columns, the first column shows the original image fitted with a person-specific AAM [28] model. The second column represents a prototype of each of the clusters found by the algorithm. The third column shows all facial gestures found in the video. In each frame, the cluster and temporal

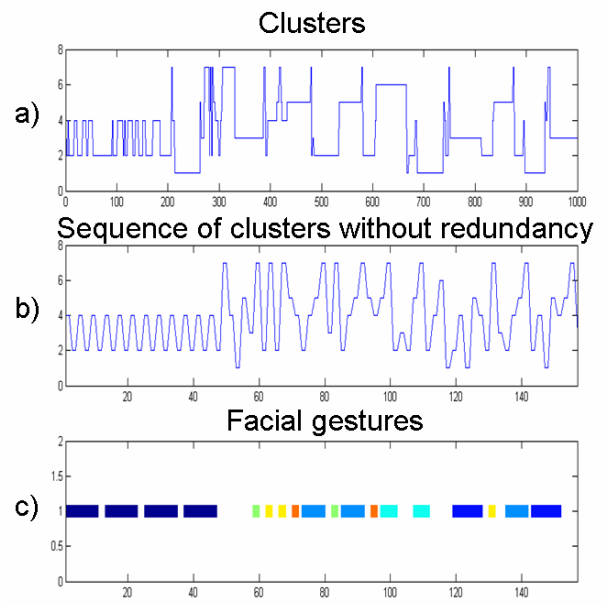


Figure 8. a) Original sequence of clusters. b) Sequence of clusters with just the transitions. c) Discovered facial gestures.

gesture that corresponds to the image is highlighted.

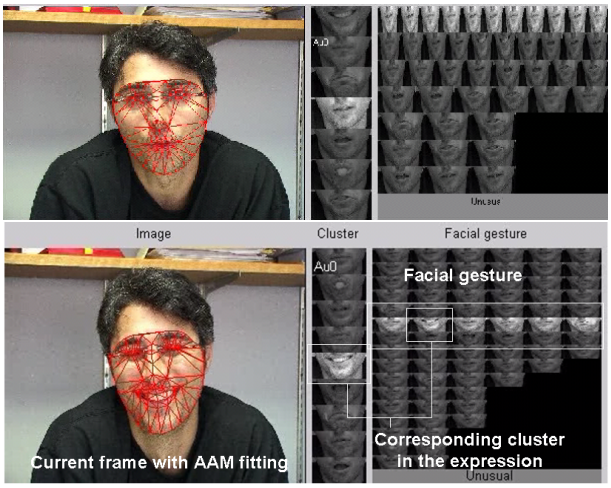


Figure 9. Frame of the output video.

## 5.2. Computer-Assisted system to increase speed and reliability of manual FACS coding

FACS (Facial Action Coding System [14]) coding is the state of the art in manual measurement of facial action [5]. FACS coding, however, is labor intensive and difficult to standardize. A goal of automated FACS coding [6] is to eliminate the need for manual coding and realize automatic recognition and analysis of facial actions. Completing the necessary FACS coding for training and testing algorithms has been a rate-limiter. Manual FACS coding remains expensive and slow. The speed, efficiency, and quality control of FACS coding can be increased dramatically by making use of the temporal segmentation proposed in this paper to preprocess video streams for human coders.

### 5.2.1 Current Approaches to FACS coding

Currently, FACS coders typically proceed in either single or multiple passes through the video. When a single-pass procedure is used, they view the video and code the occurrences of all target action units in each frame. As they proceed, they may easily have to remain alert to as many as 20, 30, or more action units simultaneously. They must be generalists and watch for all possible action units. Coding proceeds slowly (because all AUs must be considered), and quality suffers because similar action units occurring at different frames cannot be reviewed together. Instances of any given AU may be separated by long periods of coding other AUs, which interferes with the ability to visually recall past occurrences. When coders proceed in multiple passes, quality improves because only a subset of AUs is coded in any given pass. The coder is then a specialist, looks only for those few AU and benefits from memory of previous occurrences in the same subject. This process is inefficient, however, because the coder must view potentially long expanses of video that do not include any of the target AUs. Visual memory becomes impaired as the time between coding of the same or related AUs stretches out.

The inefficiency in both approaches is not inherent to FACS. It is inherent to the failure of technology to make coders more productive by providing them with relatively homogeneous video to process. In this section, we show how our dynamic clustering segments video likely to contain similar action units. FACS coders will no longer need to code all possible AUs in one pass, which compromises quality and efficiency; nor will they need to code in multiple passes, which wastes their attention during uneventful segments and challenges visual memory, albeit less than in the single pass case. We will present the FACS coder with preprocessed video segments that are likely to contain the target AU, resulting in a faster and more reliable FACS coding.

### 5.2.2 Grouping facial expressions

We use subject 19 from the DS107 database [16]. The DS107 is a deception scenario in which 20 young adults must convince an interviewer of their honesty whether or not they are guilty of having taken a sum of money. In the observational scenario, subjects entered a room in which there was or was not a check for a specified amount (typically $100). Subjects were instructed that they could take the check if they wished and then would be interrogated about their actions. The subject's task then was to convince

| Subject | Accuracy | # of clusters | # of frames |
|---|---|---|---|
| 19 | 68% | 21 | 558 |

Table 1. Clustering Accuracy.

the interrogator that they had not taken the check whether or not they had. We have tracked the facial features of subject 19 with AAMs [28, 9]. We remove AU0 by the procedure describe in section 4. After the preliminary processing we have 558 frames that have been manually labeled into 21 AUs by a certified FACS coder. This manual FACS coding provides ground truth for the analysis.

From the shape and appearance data, we compute the affinity matrix $\mathbf{K}$ with shape and appearance information and compute the first 20 eigenvectors. We run 50 iterations of k-means in the embedded space and keep the solution with smallest error. To compute the accuracy of the results for a $c$ cluster case with the ground truth, we compute a $c-$by$-c$ confusion matrix $\mathbf{C}$, where each entry $c_{ij}$ is the number of samples in cluster $i$, which belong to class $j$. It is difficult to compute the accuracy by only using the confusion matrix $\mathbf{C}$ because we do not know which cluster matches which class. An optimal way to solve for the correspondence [20] is to compute the following maximization problem:

$$\max\ tr(\mathbf{CP}) \mid \mathbf{P}\ is\ a\ permutation\ matrix \qquad (6)$$

and the accuracy is obtained by dividing the results for the number of data points to be clustered. To solve eq. 6, we use the classical Hungarian algorithm [20]. Table 1 shows the accuracy results. The clustering approach achieved 68% agreement with manual annotation, which is comparable to the inter-observer agreement of manual coding (70%).

It is interesting to notice that the clustering results depend on the shape and appearance parameters, i.e. $\sigma_s$ and $\sigma_a$. Figure 10 shows the accuracy as a function of these two parameters, and we can observe that it is stable over a large range of values.

## 6. Conclusions and Future work

In this paper we have presented a method for temporal segmentation of facial behavior and illustrate its usefulness in two novel applications. The method is invariant to geometric transformations, which is critical in real-world settings in which head motion is common. The method clusters similar facial actions, identifies unusual actions, and could be used to increase the reliability and efficiency of manual FACS annotation.

The current implementation is for the mouth region. This is the most challenging region in that the degrees of freedom of facial motion are largest in this region. The densest concentration of facial muscles is in the mouth region and the range of motion includes horizontal, lateral, and oblique

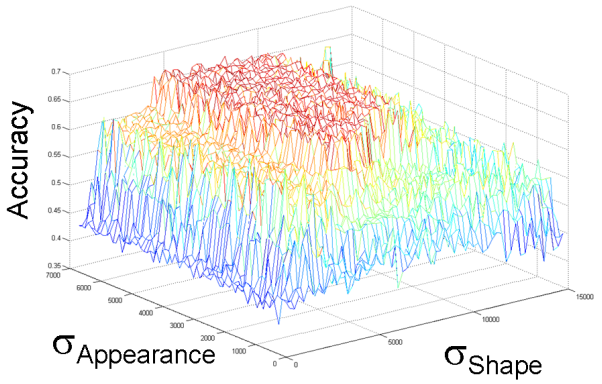


Figure 10. Accuracy variation versus $\sigma_a$ and $\sigma_s$.

[15]. Also, because of higher concentration of contralateral innervation in the lower face, the potential for asymmetric actions is much greater than for the rest of the face [31]. To be useful, a system must include all facial regions. Current work expands clustering to include eye, midface, and brow features.

**Acknowledgements** This work was partially supported by National Institute of Justice award 2005-IJ-CX-K067 and National Institute of Health Grant R01 MH 051435. Thanks to Iain Matthews for assistance with the AAM code and insightful comments about tracking. Thanks to Tomas Simon and the anonymous reviewers for helpful comments.

## References

[1] G. Adiv. 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, July 1985.

[2] Z. Ambadar, J. Schooler, and J. F. Cohn. Deciphering the enigmatic face: The importance of facial dynamics in interpreting subtle facial expressions. *Psychological Science*, 16:403–410, 2005.

[3] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[4] J. F. Cohn. Automated analysis of the configuration and timing of facial expression. In P. Ekman and E. Rosenberg, editors, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS), Oxford University Press Series in Affective Science*, pages 388–392. October 2005.

[5] J. F. Cohn, Z. Ambadar, and P. Ekman. Observer-based measurement of facial expression with the facial action coding system. *J. Coan and J. Allen (Eds). The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science. NY: Oxford.*, 2006.

[6] J. F. Cohn and T. Kanade. *Use of automated facial image analysis for measurement of emotion expression.* The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science., New York: Oxford., 2007.

[7] J. F. Cohn and K. Schmidt. The timing of facial motion in posed and spontaneous smiles. *International Journal of Wavelets, Multiresolution and Information Processing*, 2:1 – 12, March 2004.

[8] F. De la Torre and T. Kanade. Discriminative cluster analysis. In *International Conference on Machine Learning*, volume 148, pages 241 – 248, New York, NY, USA, June 2006. ACM Press.

[9] F. De la Torre, J. Vitrià, P. Radeva, and J. Melenchón. Eigenfiltering for flexible eigentracking. In *International Conference on Pattern Recognition*, pages 1118–1121, 2000.

[10] F. De la Torre, Y. Yacoob, and L. Davis. A probabilisitc framework for rigid and non-rigid appearance based tracking and recognition. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 491–498, 2000.

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

[12] C. Ding and X. He. K-means clustering via principal component analysis. In *International Conference on Machine Learning*, volume 1, pages 225–232, 2004.

[13] J. Domke and Y. Aloimonos. Deformation and viewpoint invariant color histograms. In *British Machine Vision Conference*, pages 11–509, 2006.

[14] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press.*, 1978.

[15] P. Ekman and W. Friesen. Facial action coding system (facs): Manual. In *Consulting Psychologists Press, Palo Alto, CA, USA*, 1978.

[16] M. Frank and P. Ekman. The ability to detect deceit generalizes across different types of high-stakes lies. *Journal of Personality and Social Psychology*, 72(6):1429–1439., 1997.

[17] J. Hoey. Hierarchical unsupervised learning of facial expression categories. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 99–106, 2001.

[18] C. Hu, Y. Chang, R. Feris, and M. Turk. Manifold based analysis of facial expression. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 5*, page 81, Washington, DC, USA, 2004. IEEE Computer Society.

[19] A. K. Jain. *Algorithms For Clustering Data*. Prentice Hall, 1988.

[20] D. E. Knuth. *The Standford GraphBase*. Addison-Wesley Publishing Company, 1993.

[21] C. Lee and A. Elgammal. Facial expression analysis using nonlinear decomposable generative models. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 17–31, 2005.

[22] S. Li and A. Jain. *Handbook of face recognition.* New York: Springer., 2005.

[23] G. Littlewort, M. Bartlett, I. Fasel, J. Chenu, and J. Movellan. Analysis of machine learning methods for real-time recognition of facial expressions from video. In *Computer Vision and Pattern Recognition*, 2004.

[24] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. de la Torre, and J. Cohn. AAM derived face representations for robust facial action recognition. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 155–160, 2006.

[25] J. B. MacQueen. 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, 1967.

[26] A. Martinez. Matching expression variant faces. *Vision Research*, 43(9):1047–1060, 2003.

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

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

[29] M. Pantic and I. Patras. Dynamics of Facial Expression: Recognition of Facial Actions and their Temporal Segments from Face Profile Image Sequences. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(2):433–449, April 2006.

[30] M. Pantic, N. Sebe, J. F. Cohn, and T. Huang. Affective multimodal human-computer interaction. In *ACM International Conference on Multimedia*, pages 669–676, 2005.

[31] W. Rinn. The neuropsychology of facial expression. *Psychological Bulletin*, 95:52–77, 1984.

[32] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[33] Y. Tian, J. F. Cohn, and T. Kanade. *Facial expression analysis. In S. Z. Li and A. K. Jain (Eds.). Handbook of face recognition.* New York, New York: Springer., 2005.

[34] L. Zelnik-Manor and M. Irani. Temporal factorization vs. spatial factorization. In *ECCV*, pages 434–445, 2004.

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

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