Unsupervised Discovery of Facial Events

Feng Zhou Fernando De la Torre Jeffrey F. Cohn Tomas Simon

CMU-RI-TR-10-10

May 2010

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Automatic facial image analysis has been a long standing research problem in computer vision. A key component in facial image analysis, largely conditioning the success of subsequent algorithms (*e.g*. facial expression recognition), is to define a vocabulary of possible dynamic facial events. To date, that vocabulary has come from the anatomically-based Facial Action Coding System (FACS) or more subjective approaches (i.e. emotion-specified expressions). The aim of this paper is to discover facial events directly from video of naturally occurring facial behavior, without recourse to FACS or other labeling schemes. To discover facial events, we propose a temporal clustering algorithm, Aligned Cluster Analysis (ACA), and a multi-subject correspondence algorithm for matching expressions. We use a variety of video sources: posed facial behavior (Cohn-Kanade database), unscripted facial behavior (RU-FACS database) and some video in infants. Accuracy of (unsupervised) ACA approached that of a supervised version, achieved moderate intersystem agreement with FACS, and proved informative as a visualization/summarization tool.

Contents

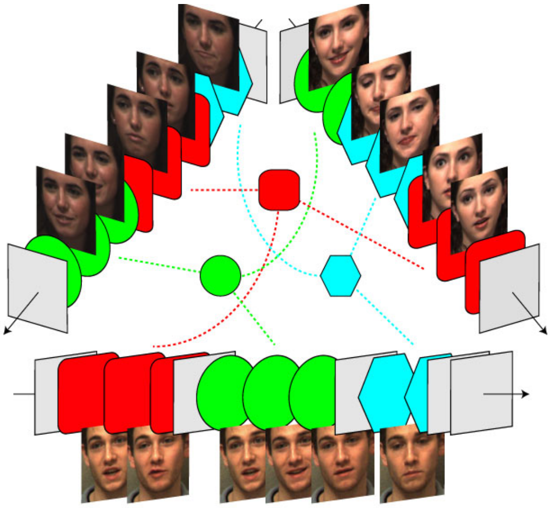


Figure 1: Selected video frames of unposed facial behavior from three participants. Different colors and shapes represent dynamic events discovered by unsupervised learning: smile (green circle) and lip compressor (blue hexagons). Dashed lines indicate correspondences between persons.

1 Introduction

The face is one of the most powerful channels of nonverbal communication. Facial expression provides cues about emotional response, regulates interpersonal behavior, and communicates aspects of psychopathology. While people have believed for centuries that facial expressions can reveal what people are thinking and feeling, it is relatively recently that the face has been studied scientifically for what it can tell us about internal states, social behavior, and psychopathology.

Faces possess their own language. To represent the elemental units of this language, Ekman and Friesen [13] in the 70's proposed the Facial Action Coding System (FACS). FACS segments the visible effects of facial muscle activation into "action units". Each action unit is related to one or more facial muscles. The FACS taxonomy was develop by manually observing graylevel variation between expressions in images and to a lesser extent by recording the electrical activity of underlying facial muscles [5]. Because of its descriptive power, FACS has become the state of the art in manual measurement of facial expression and is widely used in studies of spontaneous facial behavior. In part for these reasons, much effort in automatic facial image analysis seeks to automatically recognize FACS action units [2, 36, 31, 35].

In this paper, we ask whether unsupervised learning can discover useful facial units in video sequences of one or more persons, and whether the discovered facial events correspond to manual coding of FACS action units. We propose extensions of an unsupervised temporal clustering algorithm, Aligned Cluster Analysis (ACA) [45]. ACA is an extension of kernel k-means to cluster multi-dimensional time series. Using this unsupervised learning approach it is possible to find meaningful dynamic clusters of similar facial expressions in one individual and correspondences between facial events across individuals in an unsupervised manner. Fig. (1) illustrates the main idea of the paper. In addition, we show how our algorithms for temporal clustering of facial events can be used for summarization and visualization.

2 Temporal segmentation and clustering of human behavior

This section reviews previous work on temporal clustering and segmentation of facial and human behavior.

With few exceptions, previous work on facial expression or action unit recognition has been supervised in nature (i.e. event categories are defined in advance in labeled training data, see [2, 36, 31, 35] for a review of state-of-the-art algorithms). Little attention has been paid to the problem of unsupervised temporal segmentation or clustering prior to recognition. Essa and Pentland [14] proposed a probabilistic flow-based method to describe facial expressions. Hoey [19] presented a multilevel Bayesian network to learn in a weakly supervised manner the dynamics of facial expression. Bettinger *et al.* [4] used AAM to learn the dynamics of person-specific facial expression models. Zelnik-Manor and Irani [42] proposed a modification of structure-from-motion factorization to temporally segment rigid and non-rigid facial motion. De la Torre *et al.* [10] proposed a geometric-invariant clustering algorithm to decompose a stream of one person's facial behavior into facial gestures. Their approach suggested that unusual facial expressions might be detected through temporal outlier patterns. In summary, previous work in facial expression addresses temporal segmentation of facial expression in a single person. The current work extends previous approaches to unsupervised temporal clustering across individuals.

Outside of the facial expression literature, unsupervised temporal segmentation and clustering of human and animal behavior has been addressed by several groups. Zelnik-Manor and Irani [43] extracted spatio-temporal features at multiple temporal scales to isolate and cluster events. Guerra-Filho and Aloimonos [17] presented a linguistic framework to learn human activity representations. The low level representation of their framework, motion primitives, referred to as kinetemes, were proposed as the foundation for a kinetic language. Yin *et al.* [40] proposed a discriminative feature selection method to discover a set of temporal segments, or units, in American Sign Language. These units could be distinguished with sufficient reliability to improve accuracy in ASL recognition. Wang *et al.* [39] used deformable template matching of shape and context in static images to discover action classes. Turaga *et al.* [37] presented a cascade of dynamical systems to cluster a video sequence into activities. Niebles *et al.* [29] proposed an unsupervised method to learn human action categories. They represented video as a bag-of-words model of space-time interest points. Latent topic models were used to learn their probability distribution, and intermediate topics corresponded to human action categories. Oh *et al.* [30] proposed parametric segmental switching dynamical models to segment honeybees behavior. Related work in temporal segmentation has been done, as well, in the area of data mining [21] and change point detection [18]. Unlike previous approaches, we propose the use of ACA. ACA generalizes kernel k-means to cluster time series, providing a simple yet effective and robust method to cluster multi-dimensional time series with few parameters to tune.

3 Facial feature tracking and image features

Over the last decade, appearance models [6, 27] have become increasingly prominent in computer vision. In the work below, we use AAMs [27] to detect and track facial features, and extract features. Fig. (2a) shows an example of AAM using image data from RU-FACS [2].

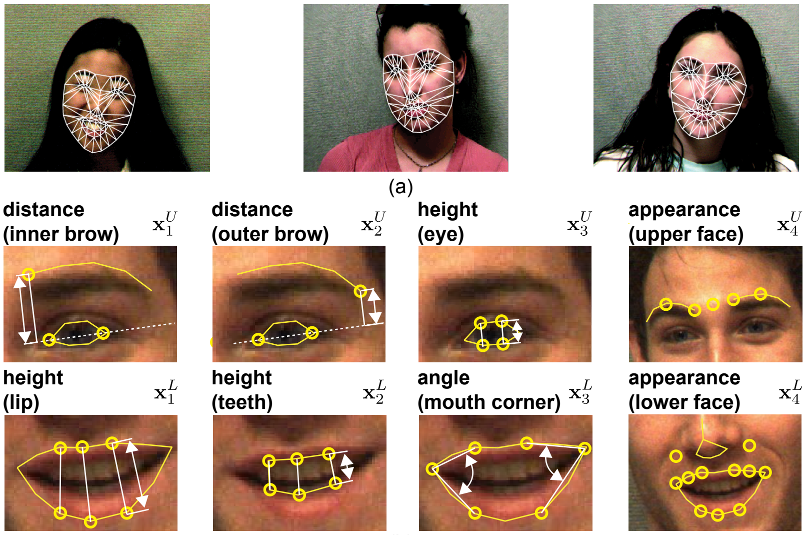
Sixty-six facial features and the related face texture are tracked throughout an image sequence. To register images to a canonical view and face, a normalization step registers each image with respect to an average face. After the normalization step, we build shape and appearance features for the upper and lower face regions. Shape features include, $\mathbf{x}_1^U$ the distance between inner brow and eye, $\mathbf{x}_2^U$ the distance between outer brow and eye, $\mathbf{x}_3^U$ the height of eye, $\mathbf{x}_1^L$ the height of lip, $\mathbf{x}_2^L$ the height of teeth, and $\mathbf{x}_3^L$ the angle of mouth corners. Appearance features are composed of SIFT descriptors computed at points around the outer outline of the mouth (at 11 locations) and on the eyebrows (5 points). The dimensionality of the resulting feature vector is reduced using PCA to retain 95% of the energy, yielding appearance features for the upper ($\mathbf{x}_4^U$) and lower ($\mathbf{x}_4^L$) face. For the task of clustering emotions, features from both face parts were used to obtain a holistic representation of the face. For more precise facial action segmentation, each face part was considered individually. See Fig. (2b) for an illustration of the feature extraction process.

4 Aligned Cluster Analysis (ACA)

This section describes Aligned Cluster Analysis (ACA), an extension of kernel k-means to cluster time series. ACA combines kernel k-means with Dynamic Time Alignment Kernel (DTAK). A preliminary version of ACA was presented at [45].

4.1 Dynamic time alignment kernel (DTAK)

To align time series, a frequent approach is Dynamic Time Warping (DTW). A known drawback of using DTW as a distance is that it fails to satisfy the triangle inequality. To address this issue, Shimodaira *et al.* [34] proposed Dynamic Time Alignment Kernel (DTAK). The DTAK between two sequences, $\mathbf{X} \doteq [\mathbf{x}_1, \cdots, \mathbf{x}_{n_x}] \in \mathbb{R}^{d \times n_x}$ (see


(a)

(b)

distance (inner brow) x_1^U
distance (outer brow) x_2^U
height (eye) x_3^U
appearance (upper face) x_4^U
height (lip) x_1^L
height (teeth) x_2^L
angle (mouth corner) x_3^L
appearance (lower face) x_4^L

Figure 2: Facial features used for temporal clustering. (a) AAM fitting across different subjects. (b) Eight different features extracted from distance between tracked points, height of facial parts, angles for mouth corners, and appearance patches.

notation[1]) and $\mathbf{Y} \doteq [\mathbf{y}_1, \cdots, \mathbf{y}_{n_y}] \in \mathbb{R}^{d \times n_y}$, is defined as:

$$\tau = \max_{\mathbf{Q}} \sum_{c=1}^{l} \frac{1}{n_x + n_y}(q_{1c} - q_{1c-1} + q_{2c} - q_{2c-1})\kappa_{q_{1c}q_{2c}},$$

where $\kappa_{ij}(\mathbf{x}_i, \mathbf{y}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{y}_j)$ represents the kernel similarity between frame $\mathbf{x}_i$ and $\mathbf{y}_j$. Through the paper we used the RBF kernel $\kappa_{ij}(\mathbf{x}_i, \mathbf{y}_j) = \exp(\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{y}_j||_2^2)$, where σ^2 is the average distance to the 10% nearest neighbors. $\mathbf{Q} \in \mathbb{R}^{2 \times l}$ is an integer matrix that contains indexes to the alignment path between $\mathbf{X}$ and $\mathbf{Y}$. For instance, if the c^{th} column of $\mathbf{Q}$ is $[q_{1c}\, q_{2c}]^T$, the q_{1c} frame in $\mathbf{X}$ corresponds to the q_{2c} frame in $\mathbf{Y}$. l is the number of steps needed to align both signals.

DTAK finds the path that maximizes the weighted sum of the similarity between sequences. A more revealing mathematical expression can be achieved by considering a new normalized correspondence matrix $\mathbf{W} \in \mathbb{R}^{n_x \times n_y}$, where $w_{ij} = \frac{1}{n_x+n_y}(q_{1c} - q_{1c-1} + q_{2c} - q_{2c-1})$ if there exist $q_{1c} = i$ and $q_{2c} = j$ for some c, otherwise $w_{ij} = 0$. Then DTAK can be rewritten:

$$\tau(\mathbf{X}, \mathbf{Y}) = tr(\mathbf{K}^T \mathbf{W}) = \psi(\mathbf{X})^T \psi(\mathbf{Y}), \qquad (1)$$

[1] Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$, and all non-bold letters denote scalar variables. $\mathbf{x}_j$ represents the j^{th} column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{X}$. $\mathbf{I}_k \in \mathbb{R}^{k \times k}$ denotes the identity matrix. $||\mathbf{x}||_2^2$ denotes the norm of the vector $\mathbf{x}$. $tr(\mathbf{X}) = \sum_i x_{ii}$ is the trace of the matrix $\mathbf{X}$. $||\mathbf{X}||_F^2 = tr(\mathbf{X}^T\mathbf{X}) = tr(\mathbf{X}\mathbf{X}^T)$ designates the Frobenius norm of a matrix. $\circ$ denotes the Hadamard or point-wise product.

where $\psi(\cdot)$ denotes a mapping of the sequence into a feature space, and $\mathbf{K} \in \mathbb{R}^{n_x \times n_y}$.

4.2 k-means and kernel k-means

Clustering refers to the partition of n data points into k disjoint clusters. Among various approaches to unsupervised clustering, k-means [26, 44] and kernel k-means (KKM) [12, 41] are among the simplest and most popular. k-means and KKM clustering split a set of n objects into k groups by minimizing the within cluster variation. KKM finds the partition of the data that is a local optimum of the following energy function [44, 9]:

$$J_{kkm}(\mathbf{M}, \mathbf{G}) = ||\phi(\mathbf{X}) - \mathbf{M}\mathbf{G}||_F^2, \quad (2)$$

where $\mathbf{X} \in \mathbb{R}^{d \times n}$, $\mathbf{G} \in \mathbb{R}^{k \times n}$ and $\mathbf{M} \in \mathbb{R}^{d \times k}$. $\mathbf{G}$ is an indicator matrix, such that $\sum_c g_{ci} = 1$, $g_{ci} \in \{0, 1\}$ and g_{ci} is 1 if $\mathbf{x}_i$ belongs to class c, n denotes the number of samples. The columns of $\mathbf{X}$ contain the original data points, and the columns of $\mathbf{M}$ represent the cluster centroids; d is the dimension of the kernel mapping. In the case of KKM, d can be infinite dimensional and typically $\mathbf{M}$ cannot be computed explicitly. Substituting the optimal $\mathbf{M} = \phi(\mathbf{X})\mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}$ value, eq. (2) results in:

$$J_{kkm}(\mathbf{G}) = tr\Big(\mathbf{L}\mathbf{K}\Big) \quad \mathbf{L} = \mathbf{I}_n - \mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}. \quad (3)$$

The KKM method typically uses a local search [12] to find a matrix $\mathbf{G}$ that makes $\mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}$ maximally correlated with the sample kernel matrix $\mathbf{K} = \phi(\mathbf{X})^T\phi(\mathbf{X})$.

4.3 ACA objective function

Given a sequence $\mathbf{X} \doteq [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ with n samples, ACA decomposes $\mathbf{X}$ into m disjointed segments, each of which corresponds to one of k classes. The i^{th} segment, $\mathbf{Z}_i \doteq [\mathbf{x}_{s_i}, \cdots, \mathbf{x}_{s_{i+1}-1}] \doteq \mathbf{X}_{[s_i, s_{i+1})} \in \mathbb{R}^{d \times n_i}$, is composed of samples that begin at position s_i and end at $s_{i+1} - 1$. The length of the segment is constrained as $n_i = s_{i+1} - s_i \leq n_{\max}$. $n_{\max}$ is the maximum length of the segment that controls the temporal granularity of the factorization. An indicator matrix $\mathbf{G} \in \{0, 1\}^{k \times m}$ assigns each segment to a class; $g_{ci} = 1$ if $\mathbf{Z}_i$ belongs to class c.

ACA combines kernel k-means with the DTAK to achieve temporal clustering by minimizing:

$$J_{aca}(\mathbf{G}, \mathbf{M}, \mathbf{s}) = ||[\psi(\mathbf{Z}_1) \ \cdots \ \psi(\mathbf{Z}_m)] - \mathbf{M}\mathbf{G}||_F^2. \quad (4)$$

The difference between KKM and ACA is the introduction of the variable $\mathbf{s}$ that determines the start and end of each segment $\mathbf{Z}_i(\mathbf{s})$. $\psi(\cdot)$ is a mapping such that, $\tau_{ij} = \psi(\mathbf{Z}_i)^T\psi(\mathbf{Z}_j) = tr(\mathbf{K}_{ij}^T\mathbf{W}_{ij})$ is the DTAK. Observe that there are two kernel matrices, $\mathbf{T} \in \mathbb{R}^{m \times m}$ is the kernel segment matrix and $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the kernel sample matrix (kernel between samples). $\mathbf{T} \in \mathbb{R}^{m \times m}$ can be expressed re-arranging the $m \times m$ blocks of $\mathbf{W}_{ij} \in \mathbb{R}^{n_i \times n_j}$ into a global correspondence matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, that is:

$$\mathbf{T} = [\tau_{ij}]_{m \times m} = [tr(\mathbf{K}_{ij}^T\mathbf{W}_{ij})]_{m \times m} = \mathbf{H}(\mathbf{K} \circ \mathbf{W})\mathbf{H}^T,$$

where $\mathbf{H} \in \{0, 1\}^{m \times n}$ is the segment-sample indicator matrix; $h_{ij} = 1$ if j^{th} sample belong to i^{th} segment. Unfortunately, DTAK is not a strictly positive definite kernel

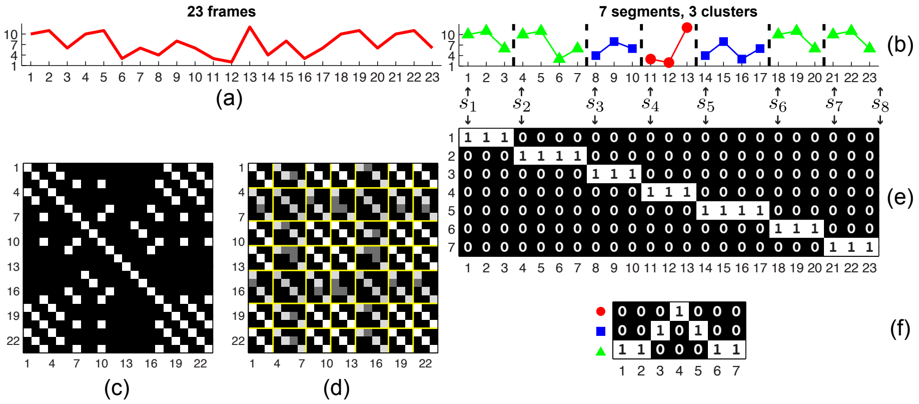


Figure 3: Example of temporal clustering. (a) 1-D sequence. (b) Results of temporal clustering. (c) Self-similarity matrix ($\mathbf{K}$). (d) Correspondence matrix ($\mathbf{W}$). (e) Frame-segment indicator matrix ($\mathbf{H}$). (f) Segment-class indicator matrix ($\mathbf{G}$).

[8]. Thus, we add a scaled identity matrix to $\mathbf{K}$; that is, $\mathbf{K} \leftarrow \mathbf{K} + \sigma \mathbf{I}_n$, were σ is chosen to be the absolute value of the smallest eigenvalue of $\mathbf{T}$ if it has negative eigenvalues.

After substituting the optimal value of $\mathbf{M}$ in eq. (4), a more enlightened form of J_{aca} can be rewritten as:

$$J_{aca}(\mathbf{G}, \mathbf{s}) = tr\Big((\mathbf{L} \circ \mathbf{W})\mathbf{K}\Big), \tag{5}$$

where $\mathbf{L} = \mathbf{I}_n - \mathbf{H}^T\mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}\mathbf{H}$. Recall $\mathbf{H}$ depends on $\mathbf{s}$. Fig. (3) illustrates the matrices $\mathbf{K}$, $\mathbf{H}$, $\mathbf{W}$ and $\mathbf{G}$ in a synthetic example of temporal clustering. Consider the special case when, $m = n$ and $\mathbf{H} = \mathbf{I}_n$; that is, each frame is treated as a segment. In this case, DTAK would be a kernel between two frames, *i.e.*, $\mathbf{W} = \mathbf{1}_n\mathbf{1}_n^T$ and ACA is equivalent to kernel k-means, eq. (3).

Optimizing ACA is a non-convex problem. We use a coordinate descent strategy that alternates between optimizing $\mathbf{G}$ and $\mathbf{s}$ while implicitly computing $\mathbf{M}$. Given a sequence $\mathbf{X}$ of length n, the number of possible segmentations is exponential, which typically renders a brute-force search infeasible. We adopt a dynamic programming (DP) based algorithm that has a complexity $O(n^2 n_{max})$ to exhaustively examine all the possible segmentations.

We rewrite eq. (4) as a sum of the following distances:

$$J_{aca}(\mathbf{G}, \mathbf{s}) = \sum_{c=1}^{k}\sum_{i=1}^{m} g_{ci}\, dist_{\psi}^2(\mathbf{Z}_i, \mathbf{m}_c) = \sum_{i=1}^{m} dist_{\psi}^2(\mathbf{Z}_i, \mathbf{m}_{c_i^*}) \tag{6}$$

where c_i^* denotes the label of the closest cluster for segment $\mathbf{Z}_i$, *i.e.*, $g_{c_i^* i} = 1$. Observe that the solution $\mathbf{G}$ is determined once $\mathbf{s}$ is known. To further leverage this relationship between $\mathbf{G}$ and $\mathbf{s}$, we introduce an auxiliary function, $J : [1, n] \to \mathbb{R}$,

$$J(v) = \min_{\mathbf{G},\mathbf{s}} J_{aca}(\mathbf{G}, \mathbf{s})|_{\mathbf{X}_{[1,v]}} \tag{7}$$

to relate the minimum energy directly with the tail position v of the subsequence $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_v]$. We can further justify that J satisfies the principle of optimality [3], *i.e.*,

$$J(v) = \min_{1<i\leq v} \left(J(i-1) + \min_{\mathbf{G},\mathbf{s}} J_{aca}(\mathbf{G}, \mathbf{s})|_{\mathbf{X}_{[i,v]}} \right) \tag{8}$$

which implies that the optimal decomposition of the subsequence $\mathbf{X}_{[1,v]}$ is achieved only when the segmentations on both sides $\mathbf{X}_{[1,i-1]}$ and $\mathbf{X}_{[i,v]}$ are optimal and their sum is minimal. Although the number of possible ways to decompose sequence $\mathbf{X}$ is exponential in n, dynamic programming [3] offers an efficient approach to minimize J by using Bellman's equation,

$$J(v) = \min_{v-n_{\max}<i\leq v} \left(J(i-1) + \min_{\mathbf{g}} \sum_{c=1}^{k} g_c dist_{\psi}^2(\mathbf{X}_{[i,v]}, \dot{\mathbf{m}}_c) \right) \tag{9}$$

where $dist_{\psi}^2(\mathbf{X}_{[i,v]}, \dot{\mathbf{m}}_c)$ is the squared distance between the segment $\mathbf{X}_{[i,v]}$ and the center of class c:

$$dist_{\psi}^2(\mathbf{X}_{[i,v]}, \dot{\mathbf{m}}_c) = \tau(\mathbf{X}_{[i,v]}, \mathbf{X}_{[i,v]}) - \frac{2}{\dot{m}_c} \sum_{j=1}^{\dot{m}} \dot{g}_{cj} \tau(\mathbf{X}_{[i,v]}, \dot{\mathbf{Z}}_j) + \frac{1}{\dot{m}_c^2} \sum_{j_1,j_2=1}^{\dot{m}} \dot{g}_{cj_1} \dot{g}_{cj_2} \tau(\dot{\mathbf{Z}}_{j_1}, \dot{\mathbf{Z}}_{j_2})$$

When $v = n$, $J(n)$ is the optimal cost of the segmentation that we seek. The inner values, $i_v^*, \mathbf{g}_v^* = \arg\min_{i,\mathbf{g}} J(v)$, are the head position and label for the last segment respectively that lead to the minima. Equation (9) unifies kernel k-means and segment-based ACA clustering based on the length constraint $n_{\max}$. If $n_{\max} = 1$, each segment consists of one single frame, and (9) is equivalent to kernel k-means.

Fig. 4 illustrates the procedure for optimizing ACA. Given a n-length sequence $\mathbf{X}$ with an initial segmentation (Fig. 4a), ACA applies the following forward-backward algorithm temporally cluster the sequence (Fig. 4b-c):

- **Forward step**. Scan from the beginning ($v = 1$) of the sequence to its end ($v = n$). For each v, $J(v)$ is computed according to (9), as well as the optimal head position i_v^* and label $\mathbf{g}_v^*$.

- **Backward step**. Trace back from the end of sequence ($v = n$). Cut off the segment whose head $s = i_v^*$ and indicator vector $\mathbf{g} = \mathbf{g}_v^*$ could be indexed from the stored records. Repeat this operation on the left part of the sequence ($v = i_v^* - 1$).

These steps are repeated until $J(n)$ converges (Fig. 4d).

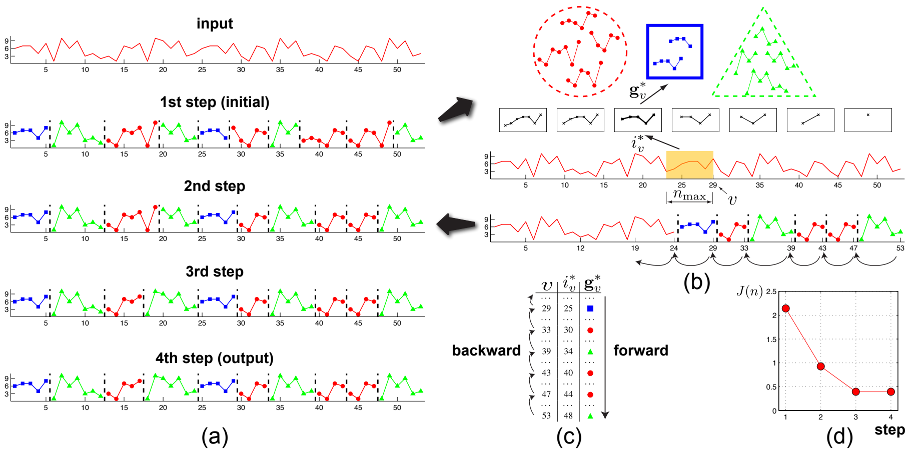


Figure 4: Coordinate-descent optimization for ACA. (a) Optimization of 1-D sequence converged in 4 steps. (b) DP-based search of segmentation between 1^{st} step and 2^{nd} step. (c) Data structure used in DP-based search. (d) Objective of ACA in each step.

5 Supervised ACA (SACA)

Clustering algorithms are generally used in an unsupervised fashion. In real application domains, it is often the case that the experimenter possesses some background knowledge (about the domain or the data set) that could be useful in clustering the data. This sections shows two extensions supervised extensions of ACA one at frame-level and one at a segment-level.

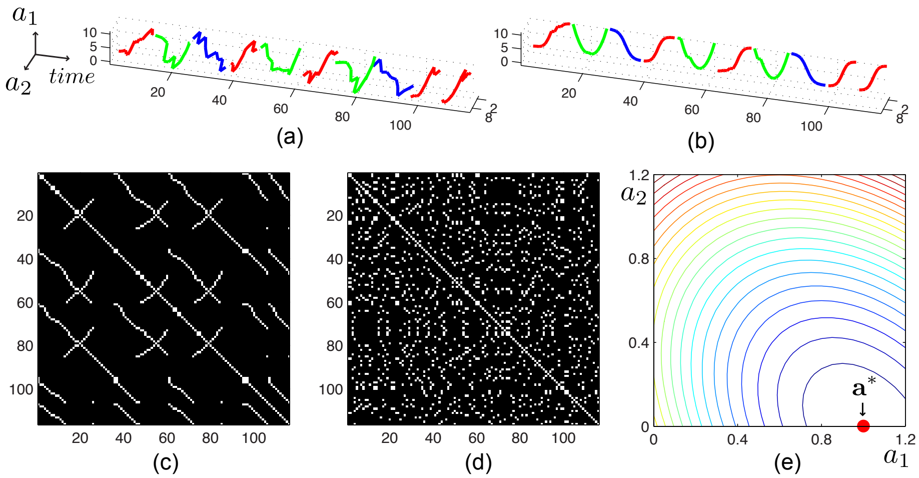
5.1 Frame-based supervised ACA

The success of kernel machines largely depends on the choice of the kernel parameters and the functional form of the kernel. As in previous work on multiple kernel learning [11, 7, 25, 24, 23, 1, 15], we consider the frame kernel as a positive combination of multiple kernels, that is:

$$\mathbf{K}(\mathbf{a}) = \sum_{l=1}^{d} a_l \mathbf{K}_l, \quad \text{s.t.} \quad \mathbf{a} \geq \mathbf{0}_d \tag{10}$$

where the set $\{\mathbf{K}_1, \cdots, \mathbf{K}_d\}$ is given and the a_l's are to be optimized. We call this frame based supervised ACA and through the paper will be referred as supervised ACA.

In the ideal case [11, 7], if two samples belong to the same class, the kernel function outputs a similarity of 1 and 0 otherwise. In the case of temporal segmentation, the label of the i^{th} frame is given by $\mathbf{G}\mathbf{h}_i$. Assuming that all labels ($\mathbf{G}$, $\mathbf{H}$, $\mathbf{W}$) are known, we minimize the distance between the ideal kernel matrix and the parameterized one,



(a) (b) (c) (d) (e)

Figure 5: Learning the kernel for ACA. (a) Original 2-D sequence. (b) Feature re-reweighting after learning. (c) $\mathbf{K}_1$. (d) $\mathbf{K}_2$. (e) Objective function $J_{learn}(\mathbf{a})$.

that is:

$$J_{learn}(\mathbf{a}) = \|\mathbf{W} \circ \big(\mathbf{F} - \mathbf{K}(\mathbf{a})\big)\|_F^2, \tag{11}$$

where $\mathbf{F} = \mathbf{H}^T\mathbf{G}^T\mathbf{G}\mathbf{H}$, and the correspondence matrix ($\mathbf{W}$) weights individually the pair of frames that have been used in the calculation of DTAK.

To optimize J_{learn} with respect to $\mathbf{a}$, we rewrite eq. (11) as a quadratic programming problem: $J_{learn}(\mathbf{a}) = \mathbf{a}^T\mathbf{Z}\mathbf{a} - 2\mathbf{f}^T\mathbf{a} + c$, where $z_{ij} = \mathrm{Tr}((\mathbf{W}\circ\mathbf{K}_i)^T(\mathbf{W}\circ\mathbf{K}_j))$, $f_i = tr((\mathbf{W}\circ\mathbf{F})^T(\mathbf{W}\circ\mathbf{K}_i))$ and c is a constant. We use the CVX toolbox [16] to solve this problem.

Fig. 5 shows a synthetic example for learning the kernels in temporal segmentation. Suppose that a 2-D sequence $\mathbf{X} \in \mathbb{R}^{2\times n}$ has been generated with meaningful segments in the first dimension and with random noisy feature in the second dimension. After minimizing J_{learn} with respect to the weights for the two kernel matrix computed from each dimension, we obtain $\mathbf{a}^* = [.9979, .0096]^T$ which assigns lower weight to the second dimension than to the first one.

5.2 Segment-based supervised ACA

This sections shows a supervised extension of ACA at a segment level.

We will denote with a dot the variables that are known. Given $\dot{m}$ segments, $\dot{\mathbf{Z}}_1, \cdots, \dot{\mathbf{Z}}_{\dot{m}}$, with known labels, $\dot{\mathbf{G}} \in \{0,1\}^{k\times\dot{m}}$, the original objective function of ACA can be reformulated as

$$J_{saca}(\mathbf{G}, \mathbf{s}) = \|[\psi(\mathbf{Z}_1)\ \cdots\ \psi(\mathbf{Z}_m)] - \dot{\mathbf{M}}\mathbf{G}\|_F^2. \tag{12}$$

where the cluster center, $\dot{\mathbf{M}}$, is defined by the given segments. Observe that unlike ACA that optimizes over $\mathbf{M}$, now $\dot{\mathbf{M}}$ is known. Given a new sequence $\mathbf{Z}$, supervised

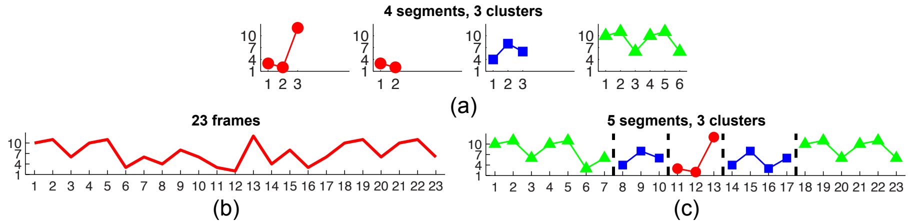


Figure 6: Example of segment-based supervised ACA. (a) Predefined clusters. (b) 1-D sequence. (c) Results of temporal segmentation.

ACA finds the segmentation of the signal $\mathbf{Z}$ that minimizes Eq. 12. The globally optimal segmentation can be found by optimizing the Bellman's equation (eq. 9).

Fig. 6 shows an example of segmenting a time series with known cluster centers. In Fig. 6(a), four instances of three clusters are depicted. Given a sequence with 23 frames (Fig. 6(b)), SACA returns a segmentation with 5 segments (Fig. 6(c)). Notice that the 3^{rd} cluster found by SACA and shown in green in Fig. 6(c) is different from the previous ones in Fig. 3(b) found by unsupervised ACA. This is because SACA optimizes Eq. 12 with respect to the predefined clusters shown in Fig. 6(a).

6 Experiments

This section reports experimental results for unsupervised temporal segmentation of facial behavior and compares them with emotion and FACS labels in two scenarios: first for individual subjects and then for sets of subjects. The ACA code is available online at http://humansensing.cs.cmu.edu/projects/acaCode.html.

6.1 Data sources

We use a variety of video sources: posed facial behavior from the Cohn-Kanade database [20], unscripted facial behavior from the RU-FACS database [2], and infants observed with their mothers [28]. The databases are:

- **Cohn-Kanade (CK) database**: The database contains a recording of posed facial behavior for 100 adults. With a few exceptions, all are between 18 and 30 years of age. There are small changes in pose and illumination, all expressions are brief (about 20 frames on average), begin at neutral, proceed to a target expression, and are well differentiated relative to unposed facial behavior in a naturalistic context (e.g., RU-FACS). Peak expressions for each sequences are AU- and emotion-labeled. The latter were used in the experiment reported below. The emotion labels were surprise, sadness, anger, fear and joy.

- **RU-FACS database**: The RU-FACS database [2] consists of digitized video and manual FACS of 34 young adults. They were recorded during an interview of approximately 2 minutes duration in which they lied or told the truth in response to an interviewer's questions. Pose orientation was mostly frontal with small to moderate out-of-plane head motion. Image data from five subjects could not be analyzed due to image artifacts. Thus, image data from 29 subjects was used.

- **Infant social behavior**: Image data were from a three-minute face-to-face interaction of a 6-month-old infant with her mother [28]. The infant was seated across from her mother. Mean head orientation was frontal but large changes in head orientation were common.

6.2 Facial event discovery for individual subjects

This section describes two experiments in facial event discovery on one individual. The first experiment compares the clustering of ACA with that of a baseline unsupervised approach (KKM), a supervised version of ACA (ACA+learn), and FACS labeling. The second experiment uses ACA to summarize the facial behavior of an infant.

6.2.1 Individual subjects in RU-FACS

We compared performance of ACA, supervised ACA (ACA+learn), and KKM. Features were 8, as described in section 3. For ACA+learn, 10 sets of 19 subjects were randomly selected to learn ACA weights ($\mathbf{a}$). For unsupervised ACA and KKM 10 sets of 10 subjects were used.

Ten subjects were randomly selected for each realization of unsupervised ACA, ACA+learn, and KKM. The initial clustering is provided k-means (best of 20 random initializations). Because the number and frequency of action units varied among subjects, and to investigate the stability of the clustering w.r.t. the number of clusters, between 7 $\sim$ 10 clusters were selected for the lower face and 4 $\sim$ 7 for the upper face. The clustering results are the average over all clusters. The length constraint was set to be $n_{\max} = 80$. Accuracy is computed using the confusion matrix:

$$\mathbf{C}(c_1, c_2) = \sum_{i=1}^{m_{alg}} \sum_{j=1}^{m_{truth}} g_{c_1 i}^{alg} g_{c_2 j}^{truth} |\mathbf{Z}_i^{alg} \cap \mathbf{Z}_j^{truth}| \qquad (13)$$

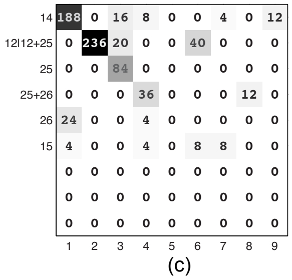
where $\mathbf{Z}_i^{alg}$ is the i^{th} segment returned by ACA (or KKM), and $\mathbf{Z}_j^{truth}$ is the j^{th} segment of the ground-truth data. $C(c_1, c_2)$ represents the total number of frames on the segment c_1 that are shared by the segment c_2 in ground truth. $g_{c_1 i}^{alg}$ is a binary value that indicates whether the i^{th} segment is classified as the c_1 temporal cluster of ACA. $|\mathbf{Z}_i^{alg} \cap \mathbf{Z}_j^{truth}|$ denotes the number of frames that the segment $\mathbf{Z}_i^{alg}$ and $\mathbf{Z}_j^{truth}$ share. The Hungarian algorithm is applied to find the optimum solution for the cluster correspondence problem. Empty rows or columns are inserted if the number of clusters is different from the ground truth. The accuracy is computed as: $\frac{1}{tr(\mathbf{C}\mathbf{1}_{k \times k})} \max_{\mathbf{P}} tr(\mathbf{CP})$ where $\mathbf{P}$ is the permutation matrix computed by the Hungarian algorithm. Due to the possible occurrence of multiple AUs in the same frame, we consider AU combinations as distinct temporal clusters. We consider AUs with a minimum duration of 10 video frames. Any frames for which no AUs occurred were omitted.

(a)

Lower Face		Upper Face	
Feature	Weight	Feature	Weight
$\mathbf{x}_1^L$	.11(.02)	$\mathbf{x}_1^U$	.01(.01)
$\mathbf{x}_2^L$	.12(.07)	$\mathbf{x}_2^U$	.04(.02)
$\mathbf{x}_3^L$	.15(.06)	$\mathbf{x}_3^U$	.07(.04)
$\mathbf{x}_4^L$	.97(.03)	$\mathbf{x}_4^U$	1.47(.04)

(b)

Segmentation	Lower Face	Upper Face
ACA + Learn	.704(.116)	.837(.123)
ACA	.687(.161)	.756(.154)
KKM	.434(.086)	.547(.138)



	1	2	3	4	5	6	7	8	9
14	188	0	16	8	0	0	4	0	12
12|12+25	0	236	20	0	0	40	0	0	0
25	0	0	84	0	0	0	0	0	0
25+26	0	0	0	36	0	0	0	12	0
26	24	0	0	4	0	0	0	0	0
15	4	0	0	4	0	8	8	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

(c)

Figure 7: Clustering performance on RU-FACS database. (a) Mean and standard deviation for the feature weights. (b) Temporal clustering accuracy. (b) Confusion matrix for subject S014.

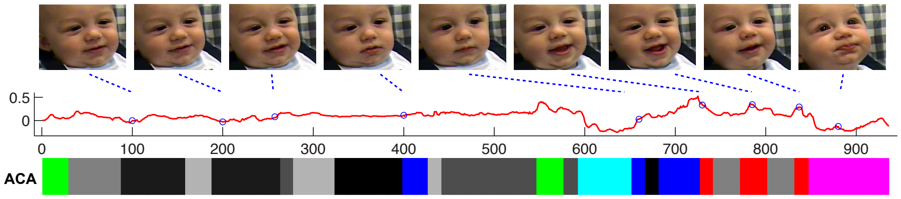


Figure 8: Temporal clustering of infant facial behavior. Each color denotes a unique cluster. Each facial gesture is coded with a different color. Observe how the frames of the same cluster correspond to similar facial expressions.

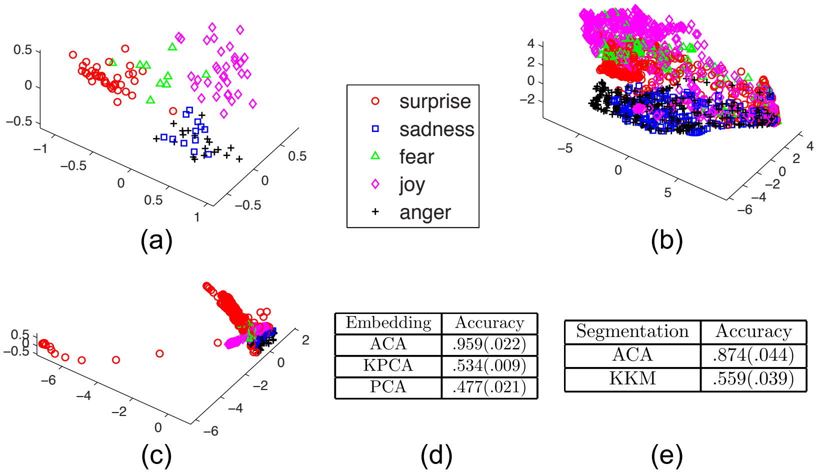
Fig. (7b) shows the mean accuracy and variance of the temporal clustering for unsupervised and supervised (learned weights) versions of ACA and KKM. ACA, KKM, and ACA+learn. ACA was substantially more accurate than KKM and approached the accuracy of ACA+learn. The mean and variance for the weights for all the features in the lower and upper face are shown in Fig. (7a). The weights gave more importance to the appearance features. Fig. (7c) shows a representative lower-face confusion matrix for subject 14.

6.2.2 Infant subject

This experiment shows an application of the proposed techniques to summarize the facial expression of an infant. Infant facial behavior is known to be more temporally complex than that of adults. Fig. (8) shows the results of running unsupervised ACA with 10 clusters on 1000 frames. We used the appearance and shape features for the eyes and mouth. These 10 clusters provide a summarization of the infant's facial events.

6.3 Facial event discovery for sets of subjects

In this section we test the ability of ACA to cluster facial behavior corresponding to different subjects. We first report results for posed facial actions. We then report results



Embedding	Accuracy
ACA	.959(.022)
KPCA	.534(.009)
PCA	.477(.021)

Segmentation	Accuracy
ACA	.874(.044)
KKM	.559(.039)

Figure 9: Clustering of 5 different facial expressions. (a) ACA embedding. (b) Kernel PCA embedding. (c) PCA embedding. (d) Clustering accuracy. (e) Temporal clustering accuracy.

for the more challenging case of unposed, naturally occurring facial behavior in an interview setting.

6.3.1 Sets of subjects in CK

ACA was evaluated in two ways. One, we compared ACA with KKM in the task of temporal clustering of facial behavior. Two, we explored its usefulness as a visualization tool. For both, emotion-labeled sequences were chosen for 30 randomly selected subjects. we used the six shape features that were normalized with respect to the initial frame. A frame kernel was computed as a linear combination of 6 kernels with equal unit weighting.

In the first experiment, we tested the ability of unsupervised ACA to temporally cluster several expressions. First, 30 randomly selected subjects (the number of facial expressions varies across subjects). The number of clusters was five and $n_{max} = 25$. Unsupervised ACA and KKM were initialized with the best solution of k-means (after 20 random initializations). Fig. (9e) shows the mean (and variance) results for 10 realizations. As above, ACA outperformed KKM. See fig. (10) for an example of the temporal clustering result.

In the second experiment ACA was evaluated as a visualization tool. Fig. (9a) shows the ACA embedding of 112 sequences from 30 randomly selected subjects (different expressions). The embedding is done by computing the first three eigenvectors of the kernel segment matrix ($\mathbf{T}$). In this experiment, the kernel segment matrix is computed using the ground-truth data (expression labels). Each point represents a video segment of facial expression. Fig. (9b) and Fig. (9c) represent the embedding computed by kernel PCA and PCA using independent frames (the frames are embedded

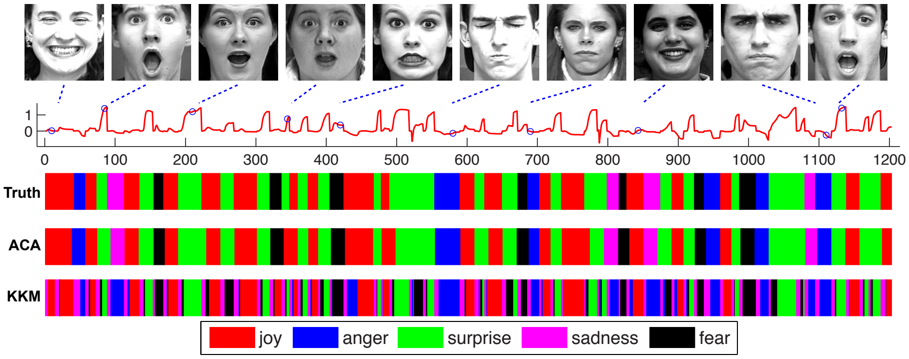


Figure 10: (a) Mouth angle. Blue dots correspond to frames. (b) Manual labels, unsupervised ACA and KKM.

using the first three eigenvectors of the kernel sample matrix $\mathbf{K}$). Because each frame represents a point, visualization of temporal structure is difficult. To test the quality of the embedding for clustering, we randomly generated 10 sets of the facial expression for 30 subjects. For each set the ground-truth label is known and the ”optimal” three dimensional embedding is computed. Then we run KKM to cluster the data into five clusters. The results (mean and variance) of the clustering are shown in Fig. (9d). As expected, the segment embedding provided by ACA achieves higher clustering accuracy than kernel PCA or PCA.

6.3.2 Sets of subjects in RU-FACS

This section tested the ability of ACA to discover dynamic facial events in a more challenging database of naturally occurring facial behavior of multiple people. Several issues contribute to the challenge of this task in the RU-FACS database. These include non-frontal pose, moderate out-of-plane head motion, subject variability and the exponential nature of possible facial action combinations.

To solve this challenging scenarios two strategies are considered: ACA+CAT concatenates all videos and runs unsupervised ACA in the concatenated video sequence. ACA+MDA runs unsupervised ACA independently for each individual and solves for the correspondence of clusters across people using the Multidimensional Assignment Algorithm (MDA) [32].

The MDA problem arises in a variety of topics in computer vision such as Multi-Target (Multi-Sensor) Tracking [33] and Multi-Frame Point Correspondence [38]. A number of approaches have been proposed in past decades to approximate the solution of this classical NP-hard problem by taking advantage of specific constraints. In this paper, we propose a variant based on the well-known Hungarian algorithm [22], which is a polynomial solution for the weighted bipartite matching problem.

Given k types of segments from n subjects, the optimum assignment among subjects is a set of pairwise permutation matrices, $\mathbf{P}^{(ij)} \in \{0,1\}^{k \times k}$, that maximizes the following objective

$$\max_{\mathbf{P}} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{c_i=1}^{k} \sum_{c_j=1}^{k} w_{c_i c_j}^{(ij)} p_{c_i c_j}^{(ij)} \tag{14}$$

$$\text{s.t.} \quad p_{c_i c_j}^{(ij)} = \{0, 1\} \tag{15}$$

$$\sum_{c_i=1}^{k} p_{c_i c_j}^{(ij)} = 1 \text{ and } \sum_{c_j=1}^{k} p_{c_i c_j}^{(ij)} = 1 \tag{16}$$

$$\sum_{i=1}^{n} \sum_{c_i=1}^{k} p_{c_i c_j}^{(ij)} = n \text{ and } \sum_{j=1}^{n} \sum_{c_j=1}^{k} p_{c_i c_j}^{(ij)} = n \tag{17}$$

where $w_{c_i c_j}^{(ij)}$ is the award (or similarity between two temporal segments) received by assigning the same label to the c_i-th segment of i-th subject and c_j-th segment of j-th subject. Notice that we adopt a different objective function (eq. 14) than the general formulation for MDA [32] because in our problem the cost can be simplified. The first and second constraints (eq. 15 and eq. 16 respectively) impose that for each pair of subjects (i and j) is a bipartite matching. Moreover, the label of segments should be globally consistent across the n subjects (eq. 17). At this point, it is important to notice that the algorithm can handle matching two subjects with different number of temporal clusters k_1 and k_2. This can be done by adding extra columns into the matrix $\mathbf{W}^{(ij)}$ with the lowest matching value.

Our approach to solve the problem consists of a greedy approximation that always satisfies the constraints. We call this method Pairwise Approximation-MDA (PA-MDA). We start by solving a sequential bipartite matching for all possible subjects' order. Suppose that there are three subjects, i_1, i_2, i_3, one of the coherent matchings can be found in two steps: (1) matching on subjects i_1 and i_2 and (2) matching subject i_3 with the combination $i \triangleq i_1 \cup i_2$. To combine the subjects, the weights in eq.(14) need to be adjusted as $w_{c_i c_j}^{(ij)} \triangleq w_{c_{i_1} c_j}^{(i_1 j)} + w_{c_{i_2} c_j}^{(i_2 j)}$. For the case of n subjects, the algorithm terminates at a global matching by repeating such a merging process n times. In fact, there are $n!$ possible paths for enumerating all n subjects. By taking advantage of the special path structure, Dynamic Programming is able to complete the enumeration in 2^n steps. The overall complexity is $O(n 2^n k^3)$ instead of the original $O((n!)^k)$ that branch-and-bound searching [32] will incur.

Using the same features described in section 6.2.1, we randomly selected 10 sets of 5 people and report the mean clustering results and variance. For ACA+MDA, we kept the same parameter setting as in the previous segmentation of one subject. The number of clusters in ACA+CAT was set to $14 \sim 17$ for the mouth and $8 \sim 11$ for the eye and the length constraint is the same as before (80). As shown in Fig. (11), ACA+MDA achieved more accurate segmentation than ACA+CAT. Moreover, ACA+MDA scales better for clustering many videos. Recall that ACA+CAT scales quadratically in space and time, which can be a limitation when processing video from many subjects. As expected, the clustering performance is lower than in the case of clustering facial events in a single individual.

Segmentation	Lower Face	Upper face
ACA + MDA	.522(.045)	.688(.087)
ACA + CAT	.493(.064)	.545(.105)
KKM + CAT	.398(.070)	.483(.100)

Figure 11: Temporal clustering across individuals (RU-FACS).

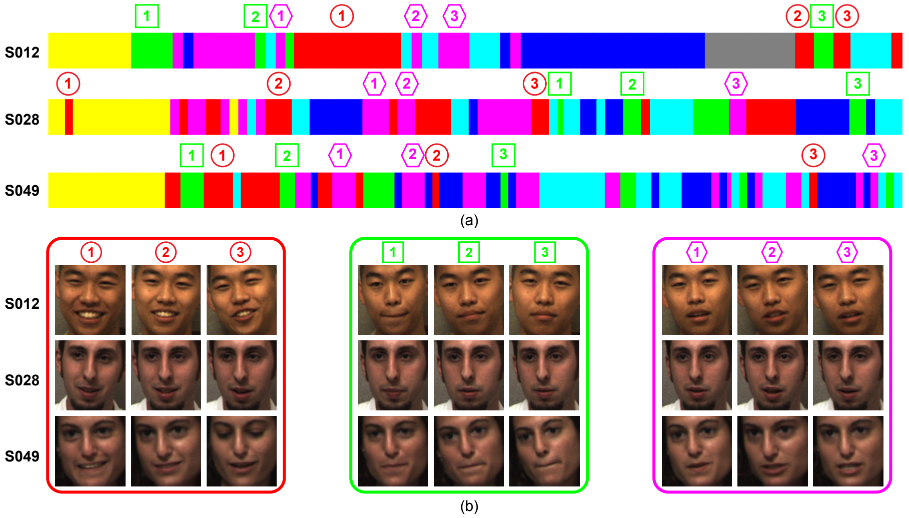


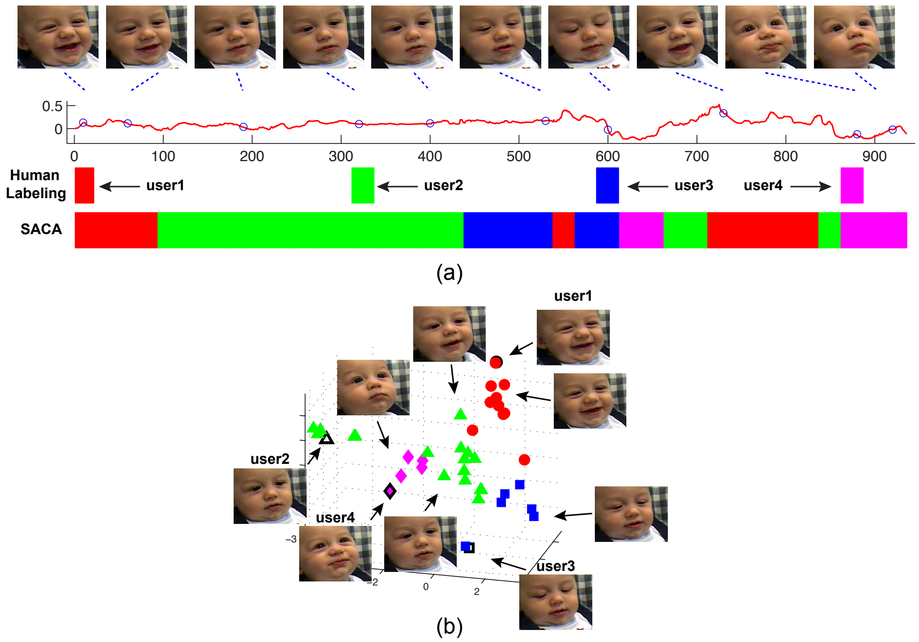
Figure 12: (a) Results obtained by ACA for subjects S012, S028 and S049. (b) Corresponding video frames.

Fig. (12a) shows the results for temporal segmentation achieved by ACA+MDA on subjects S012, S028 and S049. Each color denotes a temporal cluster discovered by ACA. Fig. (12) shows some of the dynamic vocabularies for facial expression analysis discovered by ACA+MDA. The algorithm correctly discovered smiling, silent, and talking as different facial events. Visual inspection of all subjects' data suggests that the vocabulary of facial events is moderately consistent with human evaluation.

6.4 Retrieving similar facial behavior

This section shows the ability of SACA to retrieve similar facial events to the ones defined by the user.

In this experiment the user selected four segments of the video, and the segment-based ACA is able to automatically segment the rest of the sequence in the facial behavior that is similar to the labeled ones. Fig. 13(a) shows the four segments labeled by a user and the results of the segmentation provided by segment-based ACA. Fig. 13(b) shows the embedding of facial events found by ACA, as well as the four predefined events which are denoted with black and bold edges. The 3-D embedding is computed



(a)

(b)

Figure 13: Temporal clustering of infant facial behavior with partial labels. (a) Four instances of different facial events have been provided by humans. Each color denotes a unique cluster. Each facial gesture is coded with a different color. Observe how the frames of the same cluster correspond to similar facial expressions. (b) Embedding of facial events.

by computing the leading eigenvectors of the similarity matrix $\mathbf{T}$. Notice that this is a meaningful semantic embedding, where similar facial events are closer in this embedding. With this embedding, we are able to select those facial events which are most similar to those defined by the user.

7 Conclusions and future work

At present, taxonomies of facial expression are based on FACS or other observer-based schemes. Consequently, approaches to automatic facial expression recognition are dependent on access to corpuses of FACS or similarly labeled video. This is a significant concern, in that recent work suggests that extremely large corpuses of labeled data may be needed to train robust classifiers. This paper raises the question of whether facial actions can be learned directly from video in an unsupervised manner.

We developed a method for temporal clustering of facial behavior that solves for correspondences between dynamic events and has shown promising concurrent validity with manual FACS. In experimental tests using the RU-FACS database, agreement between facial actions identified by unsupervised analysis of face dynamics and FACS approached the level of agreement that has been found between independent FACS coders. These findings suggest that unsupervised learning of facial expression is a promising alternative to supervised learning of FACS-based actions. At least three benefits follow. One is the prospect that automatic facial expression analysis may be freed from its dependence on observer-based labeling. Second, because the current approach is fully empirical, it potentially can identify regularities in video that have not been anticipated by the top-down approaches such as FACS. New discoveries become possible. This becomes especially important as automatic facial expression analysis increasingly develops new metrics, such as system dynamics, not easily captured by observer-based labeling. Three, similar benefits may accrue in other areas of image understanding of human behavior. Recent efforts to develop vocabularies and grammars of human actions [17] depend on advances in unsupervised learning. The current work may contribute to this effort. Current challenges include how best to scale ACA for very large databases and increase accuracy for subtle facial actions. We are especially interested in applications of ACA to detection of anomalous actions and efficient image indexing and retrieval.

Acknowledgements This work was partially supported by National Institute of Health Grant R01 MH 051435. Thanks to Zaid Harchaoui for helpful discussions, and Iain Matthews and Simon Baker for assistance with the AAM code.

References

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[2] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Automatic recognition of facial actions in spontaneous expressions. *J. Multimedia*, 2006.

[3] D. P. Bertsekas. *Dynamic programming and optimal control*. 1995.

[4] F. Bettinger, T. F. Cootes, and C. J. Taylor. Modelling facial behaviours. In *BMVC*, 2002.

[5] J. F. Cohn, Z. Ambadar, and P. Ekman. Observer-based measurement of facial expression with the Facial Action Coding System. *The handbook of emotion elicitation and assessment*, 2007.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998.

[7] N. Cristianini, J. Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *NIPS*, 2002.

[8] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *ICASSP*, 2007.

[9] F. de la Torre. A unification of component analysis methods. In *Handbook of Pattern Recognition and Computer Vision (4th edition)*, October 2009.

[10] F. de la Torre, J. Campoy, Z. Ambadar, and J. Cohn. Temporal segmentation of facial behavior. In *ICCV*, 2007.

[11] F. de la Torre and O. Vinyals. Learning kernel expansions for image classification. In *CVPR*, 2007.

[12] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *PAMI*, 29(11):1944–1957, 2007.

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

[14] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *PAMI*, pages 757–763, 1997.

[15] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

[16] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, 2009.

[17] G. Guerra-Filho and Y. Aloimonos. A language for human action. *IEEE Computer*, 40(5):42–51, 2007.

[18] Z. Harchaoui, F. Bach, and E. Moulines. Kernel change-point analysis. In *NIPS*.

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

[20] T. Kanade, Y. li Tian, and J. F. Cohn. Comprehensive database for facial expression analysis. In *FG*.

[21] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani. An online algorithm for segmenting time series. In *ICDM*, 2001.

[22] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955.

[23] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *ICCV*, pages 1–8, 2007.

[24] J. T. Kwok and I. W. Tsang. Learning with idealized kernels. In *ICML*, pages 400–407, 2003.

[25] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. In *ICML*, pages 323–330, 2002.

[26] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[27] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, pages 135–164, 2004.

[28] D. S. Messinger, M. H. Mahoor, S. M. Chow, and J. F. Cohn. Automated measurement of facial expression in infant-mother interaction: A pilot study. *Infancy*, 14(3):285–305, 2009.

[29] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, (79):299–318, 2008.

[30] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *IJCV*, 77(1-3):103–124, 2008.

[31] M. Pantic and I. Patras. Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Trans. Syst. Man. Cybern. B Cybern.*, 36(2):433–449, 2006.

[32] W. P. Pierskalla. The multidimensional assignment problem. *Oper. Res.*, 16(2):422–431, 1968.

[33] A. B. Poore and N. Rijavec. A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking. *SIAM J. Optim.*, 3:544, 1993.

[34] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In *NIPS*, 2001.

[35] T. Simon, M. H. Nguyen, F. de la Torre, and J. F. Cohn. Action unit detection with segment-based SVMs. In *CVPR*, 2010.

[36] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *PAMI*, 29(10):1683–1699, 2007.

[37] P. Turaga, A. Veeraraghavan, and R. Chellappa. Unsupervised view and rate invariant clustering of video sequences. *CVIU*, (113):353–371, 2009.

[38] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(1):54–72, 2001.

[39] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006.

[40] P. Yin, T. Starner, H. Hamilton, I. Essa, and J. M. Rehg. Learning the basic units in American sign language using discriminative segmental feature selection. In *ICASSP*, 2009.

[41] R. Zass and A. Shashua. A unifying approach to hard and probabilistic clustering. In *ICCV*, 2005.

[42] L. Zelnik-Manor and M. Irani. Temporal factorization vs. spatial factorization. In *ECCV*.

[43] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, 2001.

[44] H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon. Spectral relaxation for k-means clustering. In *NIPS*, pages 1057–1064, 2001.

[45] F. Zhou, F. de la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *FG*, 2008.