Structure-Preserving Sparse Decomposition for Facial Expression Analysis

Sima Taheri, *Student Member, IEEE,* Qiang Qiu, *Student Member, IEEE,* and Rama Chellappa, *Fellow, IEEE*

***Abstract*—Although facial expressions can be decomposed in terms of action units (AUs) as suggested by the Facial Action Coding System (FACS), there have been only a few attempts that recognize expression using AUs and their composition rules. In this paper, we propose a dictionary-based approach for facial expression analysis by decomposing expressions in terms of AUs. First, we construct an AU-dictionary using domain experts' knowledge of AUs. To incorporate the high-level knowledge regarding expression decomposition and AUs, we then perform structure-preserving sparse coding by imposing two layers of grouping over AU-dictionary atoms as well as over the test image matrix columns. We use the computed sparse code matrix for each expressive face to perform expression decomposition and recognition. Since domain experts' knowledge may not always be available for constructing an AU-dictionary, we also propose a structure-preserving dictionary learning algorithm which we use to learn a structured dictionary as well as divide expressive faces into several semantic regions. Experimental results on publicly available expression datasets demonstrate the effectiveness of the proposed approach for facial expression analysis.**

***Index Terms*—Facial Expression, Action Units, Sparse Decomposition, group sparsity, structure preserving**

I. Introduction

Some emotions motivate human actions and others enrich the meaning of human communications [1]. Therefore, understanding the users' emotions is a fundamental requirement of human-computer interaction systems (HCI). Facial expressions are important means of detecting several emotions. Following the work of Ekman *et al.* [2], many studies have focused on the analysis and recognition of facial expressions. The goal of facial expression analysis is to create systems that can automatically analyze facial feature changes and map them to facial expressions. This has been an active research topic for several years and has attracted the interest of many computer vision researchers and behavioral scientists, with applications in behavioral sciences, security, animation and human-computer interaction.

Facial expressions are combinations of a set of action units (AUs) introduced in the Facial Action Coding System (FACS) [2]. Action units are the smallest visibly discriminable muscle actions that combine to perform expressions and FACS is a human-observer-based system designed to code these subtle changes in facial features [3]. Such changes happen locally in the face and result in both local appearance changes and shape deformations. Previous research studies on expression analysis indicate the importance of proper modeling of such local deformations for automatic expression analysis. AUs are suitable as mid-level representations in automatic facial expression analysis systems as they reduce the dimensionality of the problem [4], [5]. However, there have been only a few attempts that exploit the domain experts' knowledge on AU composition rules and expression decompositions for designing systems to analyze and recognize expressions.

In this paper, we propose a dictionary-based approach for facial expression analysis including expression decomposition, classification and synthesis. Using the domain experts' knowledge on various AUs and how local facial regions are affected by these AUs, we first learn an AU-dictionary, D. This dictionary, as shown in Fig. 1, consists of AU-blocks, i.e., dictionary atoms corresponding to each AU, and so it has a particular structure which helps capture the high-level knowledge regarding AUs and their composition rules extracted from FACS. To encode this knowledge as sparse codes while designing the dictionaries, we propose a two-layer approach for grouping the dictionary atoms. The lower layer is the AU-layer which groups dictionary atoms corresponding to each AU. The top layer is called the expression-layer which uses the high-level knowledge to group different AUs that are composed to form a particular expression (e.g. Sad, Happy, Angry). This two-layer approach suggests a multi-layer structure-preserving sparse coding problem. The sparse code matrix, X, approximates an expressive face, Y, using this AU-dictionary.

As shown in Fig. 1, the test face is represented using a matrix of features, Y, which is referred to as the image matrix. Therefore, we are dealing with a multi-layer as well as multi-variable (columns of Y) sparse coding problem. The image matrix also has a structure in which local descriptors (columns) corresponding to each AU region on the face are grouped together and in the top layer all the columns are grouped together to represent a particular expression. In order to preserve this structure we define two grouping layers for this image matrix as well. Then by employing a multi-layer, multi-variable group sparse coding algorithm, we minimize a proper objective function which imposes these groupings (for both dictionary and image matrix) into the sparse code matrix X. This is effective for expression classification as well as decomposing an unknown expression. We can also synthesize new expressions through valid composition of AU-blocks of the dictionary.

Sima Taheri is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail:taheri@umiacs.umd.edu).

Qiang Qiu is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: qiu@cs.umd.edu).

Rama Chellappa is with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: rama@umiacs.umd.edu).

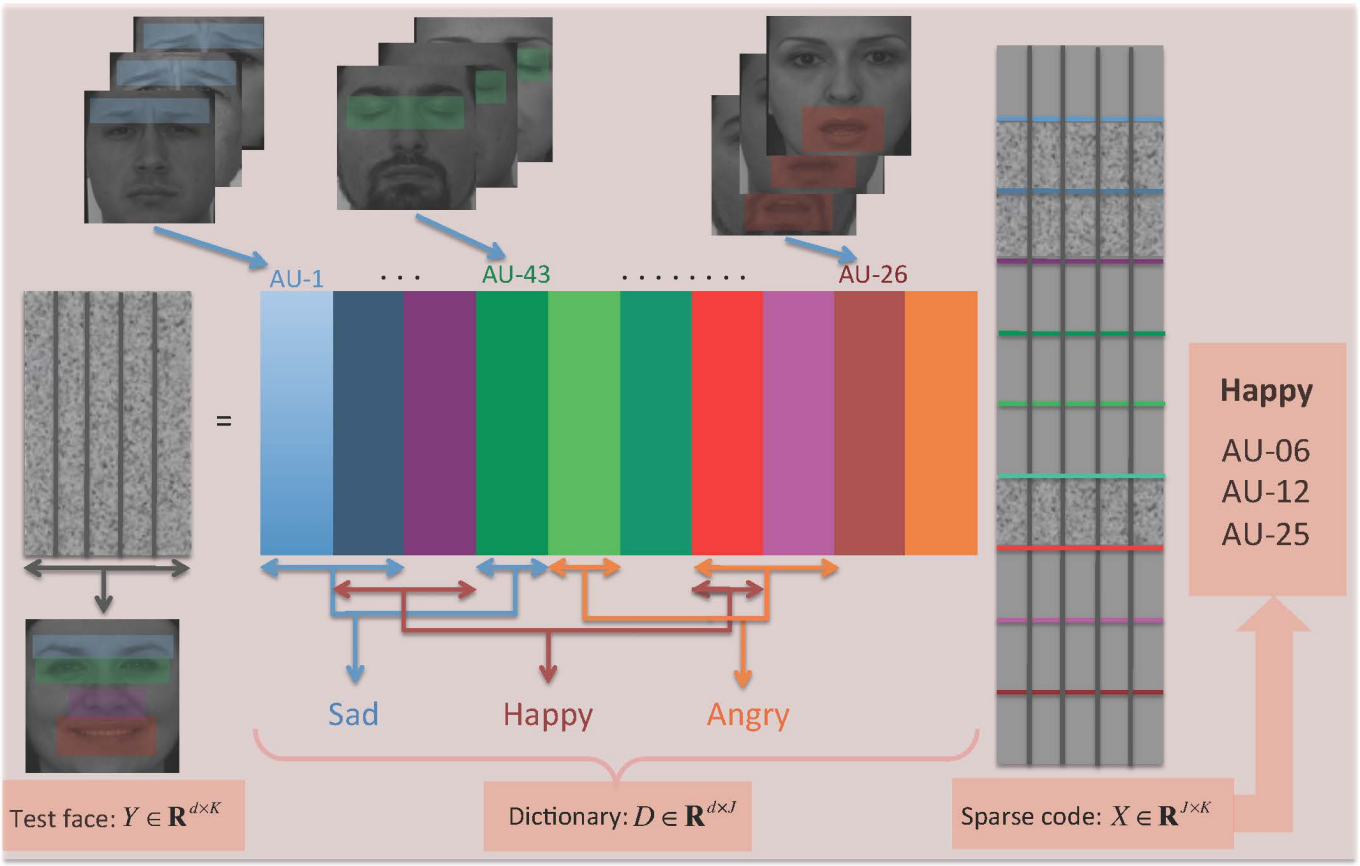


Fig. 1. Summary of the proposed algorithm. The AU-dictionary D is constructed from various blocks of AU atoms. The test face is represented using a matrix of features (image matrix Y) and it is decomposed using the AU-dictionary and coded as a structure-preserving sparse code matrix X ($Y = DX$). This representation enables expressive face classification as well as decomposition into its constituent AUs.

Learning an AU-dictionary requires a dataset of subjects performing various AUs which is not always available. On the other hand, since the definition of AU is an ambiguous semantic description in FACS [6], it is hard to define proper structures for AUs. Hence, we further propose a structure-preserving dictionary learning algorithm to jointly learn several semantic structures on expressive faces and their corresponding dictionary atoms (structure-blocks). For this purpose, we introduce an appropriate objective function and propose a greedy algorithm to optimize that. We then use the learned dictionary (concatenation of structure-blocks) for expression classification in a similar way as discussed for the AU-dictionary. We evaluate the proposed algorithm on two publicly available datasets and demonstrate the effectiveness of algorithms for expression decomposition and classification. We also illustrate some preliminary examples of expression synthesis using our generative algorithm.

II. Related Works

Many previous approaches for expression analysis have proposed discriminative classifiers for AUs and/or universal emotions [7], [8], [9], [10], [11], [12], [6], [13]. Among these, Littlewort *et al.* [7] presented the Computer Expression Recognition Toolbox (CERT) which is a software tool for fully automatic real-time facial expression recognition. CERT can automatically code the intensity of 19 different facial actions from FACS and 6 different universal facial expressions. Although some of these approaches show very promising recognition rates on emotions/AUs, they do not benefit from the connection among AUs and emotions provided in FACS as well as they are pure discriminative classifiers. For surveys on recent developments in universal emotions and AU recognition, we refer the readers to [1], [4].

A few algorithms that employed the knowledge presented in FACS regarding AUs and emotions for expression analysis are reviewed here. Tong *et al.* [9] systematically combined prior knowledge about facial expressions and AUs with image measurements through a dynamic Bayesian network (DBN) to achieve accurate, robust, and consistent facial expression analysis. They modeled the relationship between AUs and the local facial components as well as the relationships among AUs themselves using a complex graphical model and used that to infer facial expressions.

Yang *et al.* [6] interpreted facial expressions by learning some compositional features based on local appearance features around AU areas. They avoided AU detection, and tried to interpret facial expression by learning these compositional appearance features. They showed the consistency of the built compositional features with respect to the interpretation of FACS.

There are some generative approaches for expression analysis in which new expressions are recognized as compositions

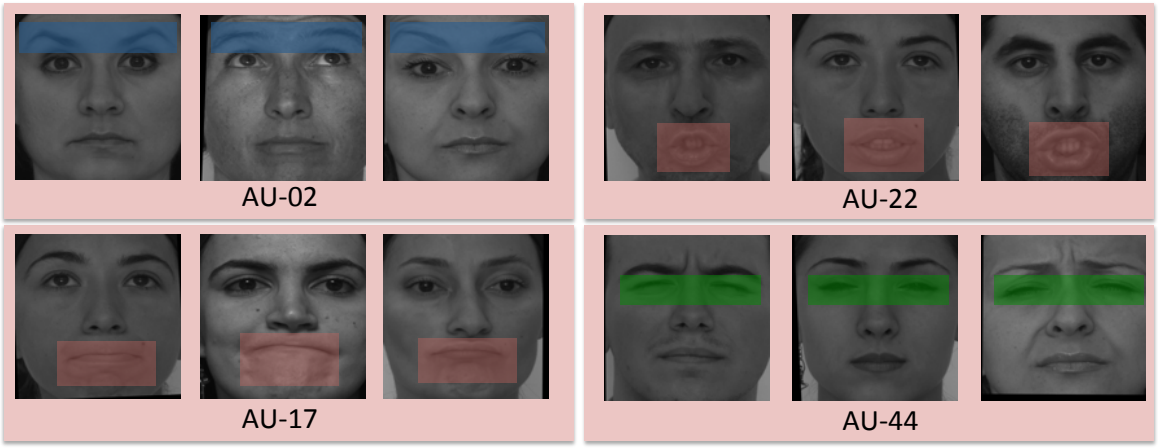


Fig. 2. Some samples of various AUs and different subjects performing them from the Bosphorus dataset. The regions that each AU affects are illustrated on the faces.

of simpler/basic expressions. Active shape models (ASM) and active appearance models (AAM) [14] are used to approximate deformable face models using linear subspace analysis. However, linear subspace methods are inadequate to represent the underlying structure of real data and so nonlinear manifold learning approaches are proposed. Liao *et al.* [15] decomposed each expression using a basis of eight one-dimensional manifolds each learned offline from sequences of labeled universal expressions. They applied tensor voting to learn these nonlinear deformation manifolds and showed results for both expression recognition and synthesis.

By observing that images of all possible facial deformations of an individual make a smooth manifold embedded in a high dimensional image space, Chang *et al.* [16] proposed a probabilistic video-based facial expression recognition method on manifolds. They represented a complete expression sequence as a path on the expression manifold and used a probabilistic model to perform expression recognition as well as synthesize image sequences. Taheri *et al.* [17] modeled AUs as geodesic pathways on the Grassmann manifold. This representation enables expression models to be generalized across view changes. Moreover, it enables the decomposition of an expression into constituent AUs, synthesis of new expressions and expression mapping between different subjects.

These studies have limitations in terms of the expressions they recognize or features they use. Most of them are pure discriminative classifiers or only use the facial shape information and so can only synthesize the shape sequences. But they all suggest that it is important to propose a generative expression analysis system that models AUs and employ them for expression analysis by incorporating domain experts' knowledge provided by FACS regarding expression decomposition and AU composition rules.

Dictionary learning and sparse coding have been effective for robustly modeling data with some level of noise and intra-class variations. Algorithms for data-driven learning of domain-specific overcomplete dictionaries are widely employed for reconstruction and recognition applications [18], [19], [20], [21]. Local variations in the appearance of the faces due to various expressions can also be modeled using a set of dictionary atoms. But facial expressions are structured actions (e.g. deformations corresponding to each AU occur at a particular local region of the face) and maintaining this structure is important for expression analysis. Yu *et al.* [22] propose a computationally efficient MAP-EM algorithm for structure-preserving dictionary learning. This approach regularizes the sparse estimation by assuming dependency on the selection of active atoms. In this work, we also need to learn a structure-preserving dictionary for modeling facial expressions and AUs.

Contribution: Our main contributions in this paper are:

- We learn an AU-dictionary by defining proper semantic regions on the face.
- We incorporate the high-level knowledge from FACS regarding AUs and their composition rules as a two-layer grouping over dictionary atoms. We also impose a similar two-layer grouping over the test image matrix.
- We extend the single multi-layer group sparse coding algorithm proposed in [23] to the multi-layer, multi-variable group sparse coding case.
- We further propose a structure-preserving dictionary learning algorithm to replace the AU-dictionary.

Outline: The rest of the paper is organized as follows. Section III discusses the structured AU dictionary, our approach to generate it and the groupings we impose over the dictionary atoms. Then in section IV we present the multi-layer multi-variable group sparse coding, the objective function and the algorithm to optimize it. A structure-preserving dictionary learning algorithm is then proposed in section VI. Experimental results are presented in section VII.

III. Action Unit Dictionary

AUs are the basic components of each expression and they are usually different for various expressions. Therefore, breaking an expression into a set of AUs is an important step toward facial expression analysis. Facial action coding proposed in FACS serves only as an initial step. However, the next step for modeling these AUs and exploiting them for expression analysis is particularly difficult due to the

AU	Name	AU	Name
1	Inner Brow Raiser	17	Chin Raiser
2	Outer Brow Raiser	20	Lip Stretcher
4	Brow Lowerer	22	Lip Funneler
5	Upper Lip Raiser	23	Lip Tightener
6	Cheek Raiser	24	Lip Pressor
7	Lid Tightener	25	Lips Part
9	Nose Wrinkler	26	Jaw Drop
12	Lip Corner Puller	27	Mouth Stretch
15	Lip Corner Depressor	28	Lip Suck
16	Lower Lip Depressor	43	Eyes Closed

Emotion	AUs
Happy	6+12+25
Sad	1+4+15
Surprise	1+2+5+26
Fear	1+2+4+5+20+26
Angry	4+5+7+23
Disgust	9+15+16

Fig. 3. FACS action units and their compositions. As the right table shows, combinations of different AUs generate universal facial emotions. [3]

ambiguous semantic nature of AUs. In this section, we propose a dictionary learning framework for facial AUs.

A. Modeling Action Units

Each AU determines the deformation of its corresponding facial components, as shown in Fig. 2 and Fig. 3. Figure 2 shows faces of a few subjects performing different AUs. As can be seen, each AU acts in a local area of the face while keeping other parts unchanged. This motivates using local features extracted from expressive faces to model each AU. Moreover, Fig. 2 shows that there is a large degree of inter-subject variations in performing various AUs. These individual differences in expressiveness relates to the degree of facial plasticity, morphology, frequency of intense expression, and overall rate of expression [3]. These intra-subject variations should also be considered while modeling various AUs. As discussed earlier, dictionary learning is effective for modeling such data with large degree of inter-class variations.

While dictionary learning using all local descriptors extracted from faces can be effective for expression analysis, it does not preserve the structure of facial deformations. For example, a deformation in a particular local area of the face (e.g. mouth) can be reconstructed using the same set of dictionary atoms used for representing a deformation in different parts of the face (e.g. brow). The coherence of such a dictionary limits its descriptive and discriminative power and the large degree of freedom in choosing dictionary atoms may become a source of instability in decomposition [22].

Therefore, it is important to preserve the structure of deformations while modeling AUs. To this end, we learn a dictionary per AU using local features extracted from AU semantic regions on faces performing that AU. These semantic regions are defined as regions on the face in which local deformations corresponding to various AUs occur. Such regions can be subjectively defined by looking at various faces performing particular AUs. Figure 2 illustrates some examples of these semantic regions we use for corresponding AUs. After defining these regions, we apply the well-known K-SVD algorithm [18] to learn data-driven AU dictionary blocks, i.e., atoms corresponding to each AU. There are also some extensions proposed for the K-SVD algorithm, [19], [20], to learn dictionaries that are compact, discriminative as well as reconstructive. We can apply these extensions to learn dictionaries for the AUs acting on the same semantic region of the face in order to learn dictionaries that are discriminative for those AUs.

Finally, we have several blocks of dictionaries (AU-blocks) which can be combined to generate an AU-dictionary, as illustrated in Fig. 1. This dictionary has a structure indicated by its AU-blocks. This structure is later imposed as a constraint for structure-preserving sparse coding (section IV). Details of feature representation and extraction are discussed in section III-C.

B. Composing Action Units

There are some subsets of AUs that usually co-occur on the face to generate meaningful facial emotions. The number of these subsets is much smaller than all possible combinations of AUs. These subsets of AUs are especially well-known for universal expressions. For example, it is known that a happy face is usually a combination of AUs-{6+12+25}. Figure 3 shows some combinations of AUs that generate universal facial emotions. After learning the AU-dictionary, we should define these high-level groupings for AU-blocks in the dictionary. Such a grouping is particularly useful for expression classification into one of universal facial emotion classes as well as expression decomposition.

It should be noted that while most of the expressions can be reconstructed as linear combinations of additive AUs, there are some non-additive AU combinations as well. An example of an additive AU combination is smiling with mouth open, which can be coded as AU-{12+25}, AU-{12+26}, or AU-{12+27} depending on the degree of lip parting [3]. However, non-additive combinations usually affect the same area of the face where the outcome of their simultaneous occurrence is different from the effect of each of the constituent AU. An example is AU-{12+15}, which often occurs during embarrassment. Although AU-12 raises the cheeks and lip corners, its action

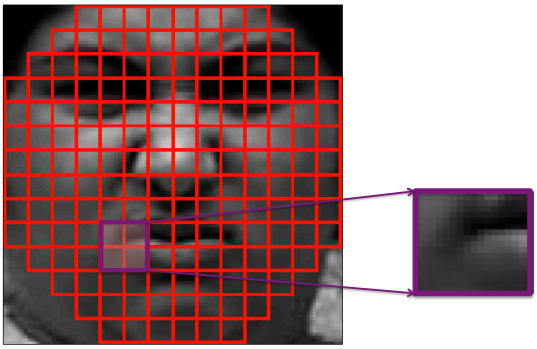


Fig. 4. Local overlapping patches are placed on the face image and local descriptors are extracted from each patch.

on lip corners is modified by the downward action of AU-15 [3]. Although such non-additive combinations usually occur sequentially, the simultaneous occurrence of them imposes further constraints. Since the number of such non-additive AUs is limited, this issue can be addressed by adding these non-additive combinations as new AUs to the dictionary.

C. Feature Extraction

Feature representation plays an important role in facial expression recognition. The features used for this purpose generally fall into two categories, geometric features [17], [16], [15] and appearance features [6], [24], [9]. In this work we use proper appearance features to model local appearance deformations.

Selection of appropriate features is critical in facial expression analysis. Popular local appearance descriptors include Gabor filter, Haar-like features, SIFT and Local Binary Patterns (LBP). While all these features are powerful for describing local appearances, the SIFT features are effective in describing the edges and finer appearance features. Since deformations corresponding to facial expressions are mainly in the form of lines and wrinkles, we chose SIFT features for our experiments.

To extract the local features, we divide each face image into overlapping patches so that they cover all the AUs' semantic regions (Fig. 4). We extract SIFT features at three scales from the center of all the patches, denoted as $\{P_1, P_2, ...P_K\}$, where K is the number of image patches. We choose the local patches to be of size $(n/8 \times n/8)$ where the size of the face images are $(n \times n)$ (after aligning and resizing). The amount of overlaps between patches may vary and in the experiments presented in this paper it is set to $n/16$.

IV. Structure-Preserving Sparse Coding

In the previous section we discussed how to learn an AU-dictionary and defined some compositional rules for grouping AUs to form various emotions. The particular structure in the dictionary and the AUs grouping can be imposed as constraints for structure-preserving sparse coding. In this section, we discuss the formulation of our multi-layer, multi-variable group sparse coding and present methods for predicting the class label of an expressive face and decomposing an expression into its constituents using sparse codes.

A. Problem Formulation

Having the structured AU-dictionary, an expressive face can be decomposed into combinations of some AUs. Our goal is to impose proper grouping-based constraints on this decomposition so that a sparse subset of AU-blocks in the dictionary is used for expression reconstruction and the subset is among the valid compositions of AUs that we discussed in section III-B. Such grouping-based constraints encode the high-level knowledge regarding expression formations on the face.

Dictionary grouping: There are two layers of grouping constraints to be imposed on the dictionary atoms. In the lower layer, to emphasize that the dictionary atoms used for reconstructing the test face should come from a sparse set of AUs, we impose the sparsity constraint on the AU-blocks in the dictionary and force the number of blocks with non-zero coefficients to be as few as possible. On the top layer, we impose AUs co-occurrence information through valid AUs composition (or expression-layer grouping) by forcing the groups having non-zero coefficients for their AU-blocks to be as few as possible. We refer to these two layers of grouping constraints as the *dictionary-AU-layer* and the *dictionary-expression-layer*.

Image matrix grouping: As mentioned before, an expressive face is represented as a set of local descriptors extracted from overlapping patches on the face. So we have a multi-variable (multi-column) representation for each test face. However, the sparse representation for each of the local descriptors is not independent from others. Hence, we define two layers of grouping constraints on the image matrix. On the top layer, we force the same sparsity pattern for all the local descriptors by grouping them together in one group. This means that we want all these local descriptors to have the same expression label. In the lower layer, we want the descriptors extracted from the same AU semantic regions on the face to have similar sparsity patterns. This implies that these descriptors are reconstructed using the atoms in the same dictionary-AU-layer. We refer to these two layers of grouping on the image matrix as the *test-AU-layer* and the *test-expression-layer*. An illustration of these different layers of grouping constraints and the sparse code matrix is depicted in Fig. 1.

The AU-dictionary consists of N blocks for N action units, $D = \{D_1, D_2, ..., D_N\}$, where the i^{th} block has J_i dictionary atoms. Therefore, the AU-dictionary, D, is a matrix of size $d \times J$ where d is the dimensionality of local descriptors extracted from a patch on the face and $J = \sum_i J_i$. The dictionary-AU-layer is formed by grouping atoms in each D_i and the dictionary-expression-layer is formed by grouping some valid subsets of $\{D_i\}$'s as discussed in section III-B. We express these two grouping layers using a set of indices, $G_{dic} = \{g_1, g_2, ..., g_{|G_{dic}|}\}$, where $g_i \subset \{1, ..., J\}$ includes indices of those dictionary atoms grouped together in either

AU- or expression-layer. $|G_{dic}|$ is the total number of such groupings which is a summation of the number of groups in two layers.

Image matrix Y is a $d \times K$ matrix where K indicates the number of local descriptors extracted from each expressive face (number of patches). We also define a set of indices, $G_{tst} = \{g_1, g_2, ..., g_{|G_{tst}|}\}$ where $g_i \subset \{1, ..., K\}$ includes indices of grouped dictionary atoms in either the AU- or the expression-layer. $|G_{tst}|$ is the total number of such groupings which is the summation of number of groups in two layers. The matrix of sparse codes, $X \in \mathbb{R}^{J \times K}$ should be computed such that the grouping structures indicated by G_{dic} and G_{tst} are satisfied. Therefore, we formulate the objective function of multi-layer, multi-variable structure-preserving sparse coding as follows:

$$\begin{aligned} \min_{X \in \mathbb{R}^{J \times K}} f(X) = \min_{X} \quad & \frac{1}{2}\|Y - DX\|_F^2 \\ & + \gamma_{dic} \sum_{k=1}^{K} \sum_{g \in G_{dic}} \omega_g^{dic} \|X_g^{(k)}\|_2 \\ & + \gamma_{tst} \sum_{j=1}^{J} \sum_{g \in G_{tst}} \omega_g^{tst} \|X_g^{(j)}\|_2 + \lambda \|X\|_1 \end{aligned} \tag{1}$$

Here γ_{dic}, γ_{tst}, ω_g^{dic}, ω_g^{tst} and λ are weights on different layers and different groups. $X^{(k)}$ and $X^{(j)}$ are the k^{th} column and the j^{th} row of matrix X, respectively, $X_g^{(k)}$ is a part of the column $X^{(k)}$ indicated by the indices in $g \in G_{dic}$ and $X_g^{(j)}$ is a part of the row $X^{(j)}$ indicated by the indices in $g \in G_{tst}$. Also $\|X\|_1$ is the L^1-norm of the matrix X which is defined as the L^1-norm of the vector formed by concatenating all the columns of the matrix. This L^1-norm penalty encourages the solution to be generally sparse, and λ is the regularization parameter that controls the sparsity level. For the $X_g^{(k)}$ and $X_g^{(j)}$ we use the L^2-norm to encode the sparse codes within each group as a unit. We adopt the Proximal Gradient method proposed recently in [23] to optimize this objective function and extend it to the multi-layer, multi-variable sparse coding case, as discussed in the next section.

V. Objective Optimization

In this section we discuss an extension of the Proximal Gradient method [23] to optimize the objective function (1). This objective function consists of three terms as follows,

$$\min_{X \in \mathbb{R}^{J*K}} f(X) = \min_{X} g(X) + \Omega(X) + \lambda \|X\|_1 \tag{2}$$

where $g(X) = \frac{1}{2}\|Y - DX\|_F^2$ is the squared-error loss and $\Omega(X)$ is called the structured-sparsity-inducing penalty [23]. The main challenge in optimizing this objective function arises from the overlapping group structure in the non-smooth penalty term $\Omega(X)$. The overlaps among $\{X_g^{(k)}\}_{g \in G_{dic}}$ and $\{X_g^{(j)}\}_{g \in G_{tst}}$ make the block coordinate descent methods [25], [26] which are commonly used for the problem with non-overlapping groups (group Lasso) not applicable. The most widely adopted method for addressing this problem is to formulate it as a second-order cone programming (SOCP) and solve it by the interior method (IPM) [27]. But this approach is computationally prohibitive even for problems of moderate size. Very recently, Chen *et al.* [23], [28] proposed the Proximal Gradient method for estimating regression parameters with the overlapping group structure encoded in the structured-sparsity-inducing norm. They showed that using the dual norm, the non-separable structured-sparsity-inducing penalty $\Omega(X)$ can be approximated using a smooth function such that its gradient can easily be calculated. The approximation problem can then be solved by the first-order proximal gradient method: fast iterative shrinkage-thresholding algorithm (FISTA) [29]. In this paper, we adopt this method and extend it to the multi-layer, multi-variate group sparse coding in order to optimize (1).

The non-smooth penalty term $\Omega(X)$ can be formulated as

$$\begin{aligned} \Omega(X) \quad &= \quad \gamma_{dic} \sum_{k=1}^{K} \sum_{g \in G_{dic}} \omega_g^{dic} \|X_g^{(k)}\|_2 \\ &+ \quad \gamma_{tst} \sum_{j=1}^{J} \sum_{g \in G_{tst}} \omega_g^{tst} \|X_g^{(j)}\|_2 \\ &= \quad \max_{A_1, A_2} \quad \langle C_1 X, A_1 \rangle \quad + \quad \langle C_2 X^T, A_2 \rangle \end{aligned} \tag{3}$$

where A_1 and A_2 are auxiliary matrices associated with $X_g^{(k)}$ and $X_g^{(j)}$ respectively. The matrix A_1 is of size $\sum_{g \in G_{dic}} |g| \times K$ and its k^{th} column is defined as $\alpha^k = [{\alpha_{g_1}^k}^T, ..., {\alpha_{g_{|G_{dic}|}}^k}^T]^T$ with the domain $\mathcal{Q} \equiv \{\alpha^k | \|\alpha_g^k\|_2 \leq 1, \forall g \in G_{dic}\}$, where $\mathcal{Q}$ is the Cartesian product of unit balls in Euclidean space and thus a closed and convex set. A_2 is also a matrix of size $\sum_{g \in G_{tst}} |g| \times J$ and its j^{th} column is defined as $\alpha^j = [{\alpha_{g_1}^j}^T, ..., {\alpha_{g_{|G_{tst}|}}^j}^T]^T$ with the similar domain as defined before. There are also two highly sparse matrices, C_1 and C_2, which help separating the overlapping groups in X. In the matrix $C_1 \in \mathbb{R}^{\sum_{g \in G_{dic}} |g| \times J}$, the rows are indexed by all pairs of $(i, g) \in \{(i, g) | i \in g, i \in \{1, ..., J\}\}$, the columns are indexed by $j \in \{1, ..., J\}$, and each element is given as:

$$C_{(i,g),j}^{(1)} = \begin{cases} \gamma_{dic} \omega_g^{dic} & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}$$

and similarly the elements of $C_2 \in \mathbb{R}^{\sum_{g \in G_{tst}} |g| \times K}$ are also defined (replacing $\gamma_{dic}\omega_g^{dic}$ with $\gamma_{tst}\omega_g^{tst}$).

The smooth approximation of $\Omega(X)$ is formulated as follows,

$$f_\mu(X) = \max_{A_1, A_2} \big(\langle C_1 X, A_1 \rangle + \langle C_2 X^T, A_2 \rangle - \mu(d(A_1) + d(A_2)) \big) \tag{4}$$

where μ is the positive smoothness parameter which controls the degree of approximation and $d(A_{(.)}) = \frac{1}{2}\|A_{(.)}\|_F^2$. Chen *et al.* [23] proved that for any $\mu > 0$, $f_\mu(X)$ in a convex and continuously-differentiable function in X, and the gradient of $f_\mu(X)$ takes the following form:

$$\nabla f_\mu(X) = C_1^T A_1^* + A_2^{*T} C_2$$

where A_1^* and A_2^* are optimal solutions to (4). The paper [23] also provides the closed-form equations for these optimal solutions. The equations presented in [23] can be easily extended to our problem. Given the smooth approximation

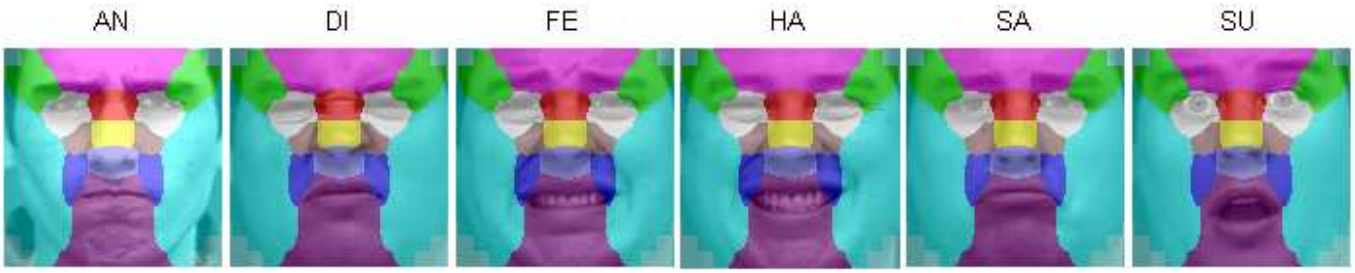


Fig. 5. Learned structures for the universal expressions from the CK+ dataset (best viewed in color). AN: Angry, DI: Disgust, FE: Fear, HA: Happy, SA: Sad, SU: Surprise.

of the non-smooth structured-sparsity-inducing penalties, the fast iterative shrinkage-thresholding algorithm (FISTA) can be applied to minimize the objective function in (2). Readers are referred to [23], [28] for more details on this optimization technique and the smoothing proximal gradient algorithm for structured sparse coding.

A. Expression Decomposition and Classification

Using the structure-preserved sparse code matrix X, we can decompose an expressive face into its constituent AUs. Then the magnitude of each AU (the L2-norm of the corresponding block in the sparse code matrix X) can be an indication of the intensity of that AU in the face. This decomposition is not limited to faces with universal emotions and can be performed for any expressive face. However since FACS specifically has the information regarding AUs composition rules for the universal emotions, we can enforce these particular groupings into the dictionary-expression-layer and employ that for universal expression recognition.

Using the information presented in Fig. 3, right table, we form the grouping indices for the dictionary-expression-layer. Now having a test face with universal emotion, we can predict the class label for this face based on the reconstruction error in the dictionary-expression-layer such that we assign the test image to the class with the minimum residual error computed as:

$$c^* = \arg\min_c \|Y - D\delta_c(X)\|_F \tag{5}$$

where $\delta_c(X)$ is obtained by setting all the coefficients in X except those in the c^{th} dictionary-expression-layer to be zero.

VI. Structure-Preserving Dictionary Learning

Constructing the AU-dictionary needs expert-level knowledge regarding AUs and the regions they affect on the face as well as a dataset with subjects performing various AUs for data-driven dictionary learning. Such information and dataset may not always be available. Therefore, it is necessary to have an automatic approach for structure-preserving dictionary learning for facial expression analysis.

To this end, we propose a structure-preserving dictionary learning algorithm for facial expressions. The goal is to jointly estimate some semantic structures on different expressive faces and their corresponding dictionary atoms. Our approach is motivated by Yu *et al.* [22] which proposed a computationally efficient MAP-EM algorithm for structure-preserving dictionary learning. With the goal of preserving the image directional regularity, they defined an initial set of dictionary bases using directional PCAs. Then using the EM framework, patches from the input image are clustered based on their residual errors over the initial dictionary bases and the dictionary bases are updated. This process converges after some iterations and finally the directional structures on the image as well as their dictionary bases are obtained.

We model a face as a collection of local subspaces so that each subspace element is well reconstructed using that subspace basis and its approximation via other subspaces results in a large residual error. So the goal is to find these subspace structures over various expressive faces. In other words, we want to find some clusters $\{S_i\}$ over the facial patches and their corresponding dictionary atoms $\{D_i\}$ so that the final clusters (or their corresponding subspace representations) are as separate as possible. This can be achieved by maximizing the summation over cross-residual errors which we define as the error in representing each cluster's descriptors using other cluster's dictionary atoms. The objective function can be formulated as follows,

$$\max_{\{S_i\},\{D_i\}} \sum_i \sum_{j \neq i} \|Y(S_i) - D_j X_{ij}\|_F^2 \tag{6}$$

where $Y(S_i)$ is a matrix with the columns of all the descriptors extracted from the patches at cluster S_i, and X_{ij} is the matrix of sparse coefficients resulting from decomposition of $Y(S_i)$ on dictionary D_j where $j \neq i$. This problem can be solved by a greedy algorithm based on a bottom-up pair-wise merging procedure. The algorithm starts with some initial clustering of input patches and then in order to maximize (6), at each step we greedily merge two clusters with minimum cross-residual costs.

We initialize the clusters at each of the patch locations, p, on expressive faces with a same expression label, e, and learn the initial dictionary atoms, $D_{e,p}$, using patch descriptors at each initial cluster, $S_{e,p}$. This means that if we have K local patches at each face image and E different expression labels, we start with a $K \times E$ initial set of dictionary blocks. The reason to incorporate the expression labels for structure initialization is that the patches at the same locations but on different expressive faces do not necessarily encode the same semantic knowledge. For example, the patches corresponding to the lip corners are in different locations for Happy and Surprise faces.

As mentioned earlier, we adopt a greedy bottom-up procedure to merge pairs of clusters. At each iteration of the

Fig. 6. Some example of expressive faces in (first row) CK+ and (second row) Bosphorus datasets after the preprocessing step is completed.

algorithm (before the stopping criterion is met), the cross-residual error between each pair of clusters, C_{rc}, is calculated and then the two clusters with the minimum merging cost, $C_{\tilde{r}\tilde{c}}$, are merged to form a new cluster,

$$\begin{aligned}(\tilde{r},\tilde{c}) &= \arg\min_{r,c} C_{rc} \\ &= \arg\min_{r,c} \big(\|Y(S_r) - D_c X_{rc}\|_F^2 \\ &+ \|Y(S_c) - D_r X_{cr}\|_F^2\big)\end{aligned} \tag{7}$$

The dictionary for the new cluster is updated using the K-SVD algorithm [18]. The procedure stops when the minimum merging cost of two clusters goes above a threshold. A summary of this algorithm is presented in Algorithm 1. Here $\{y^e_{:,p}\}$ is the feature pool extracted from the p^{th} patch location in all the images with expression label e. Figure 5 illustrates the result of applying this algorithm on expressive faces with universal emotions from the CK+ dataset. As it can be seen, the learned structures are almost similar for different expression classes. Now in order to construct the structure-preserving dictionary, we learn the dictionary blocks for the structures on each of the expression classes. Then the final dictionary is formed by putting these dictionary blocks together. In this case the two-layer grouping is formed in the lower layer by grouping the dictionary atoms corresponding to each structure and in the top layer by grouping the structures corresponding to each expression class. We employ this dictionary for the universal expression recognition.

VII. Experimental Results

We conducted experiments using two publicly available datasets. The first is the Bosphorus dataset [30] that is composed of a selected subset of AUs as well as the six universal emotion categories: Anger, Disgust, Fear, Happy, Sadness and Surprise, for 105 subjects. For each subject, the neutral face and the face in the apex of various AUs and emotions are presented. Some AUs or emotions are not available for some subjects. The second dataset is the Extended Cohn-Kanade dataset (CK+) [12] which consists of 593 sequences from 123 subjects. The image sequences incorporate the onset (neutral face) to peak formation of facial expressions. Only 327 out of 593 sequences have emotion labels from each of the six universal emotion categories. Again some emotion sequences are not available for some subjects.

Data: features from all images
Result: final structures and dictionaries: $\{\{S_s\},\{D_s\}\}$
initialization: $S_{e,p} \leftarrow \{y^e_{:,p}\}$,
$D_{e,p} \leftarrow$ K-SVD$\big(Y(S_{e,p})\big)$;
while *stopping criterion has not been met* **do**
  Cost-Matrix =
  $\{C_{rc} = \|Y(S_r) - D_c X_{rc}\|_F^2 + \|Y(S_c) - D_r X_{cr}\|_F^2\}$;
  $(\tilde{r},\tilde{c}) = \arg\min_{r,c}$ Cost-Matrix;
  Update Step:
    $S_{new} \leftarrow$ merge$\{S_{\tilde{r}} \& S_{\tilde{c}}\}$;
    $D_{new} \leftarrow$ K-SVD$\{Y(S_{new})\}$;
    Update Cost-Matrix;
  if $C_{\tilde{r}\tilde{c}} >$ *cost-threshold* **then**
    stopping criterion is met;
  end
end

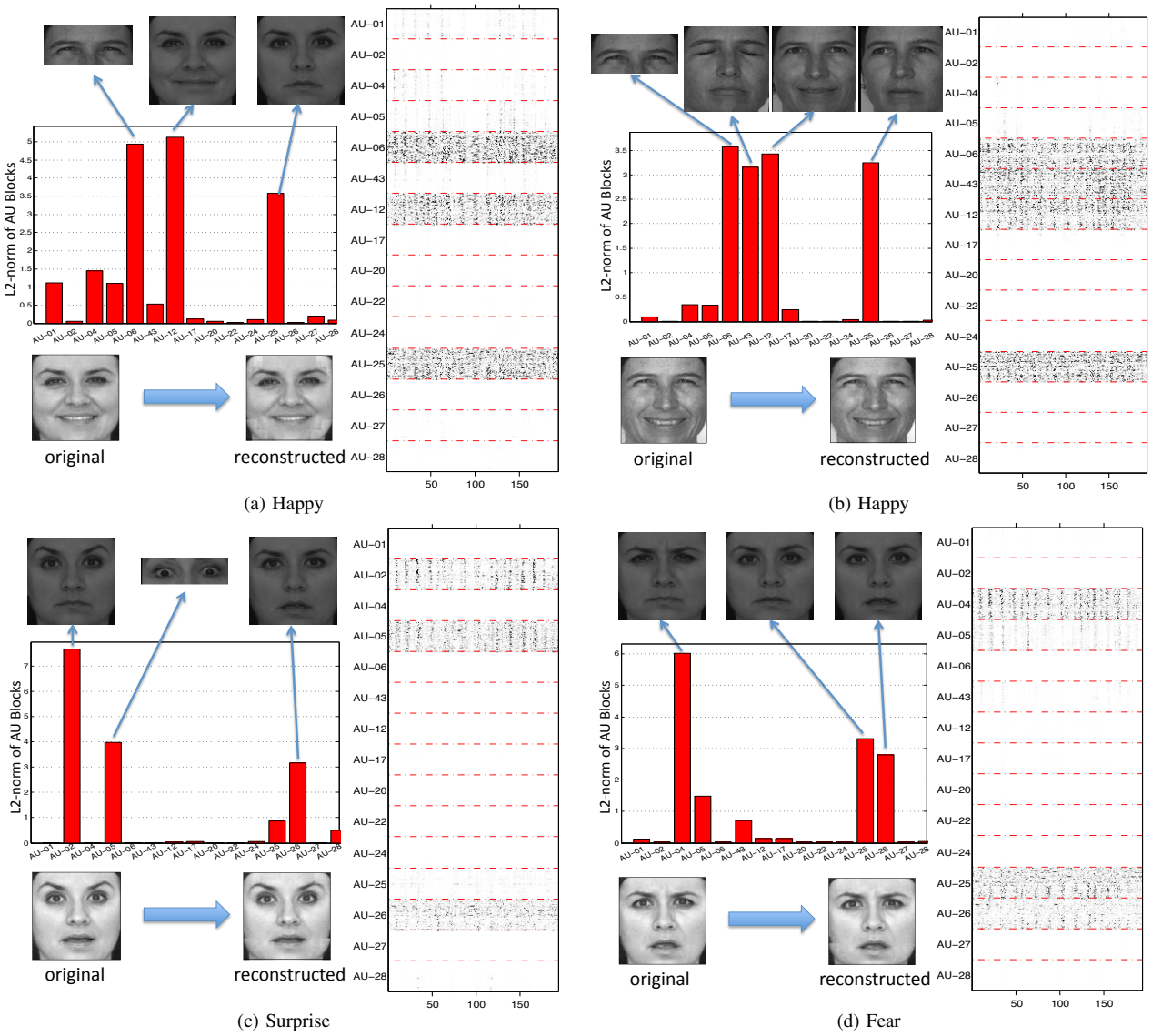
Algorithm 1: Summary of the algorithm for structure-preserving dictionary learning.

In the preprocessing step for the CK+ dataset, we first detect and crop faces at the apex of sequences. Then for both datasets, face images are resized to 128×128 and using the coordinates of eye corners and nose tip (provided by the datasets) the faces are properly aligned. Figure 6 shows some examples of faces from both datasets after the preprocessing step is completed.

A. Parameters Setting

Multi-layer, multi-variate group sparse coding has several parameters and it is important to assign appropriate values to them for better performance of the algorithm. We follow the weighting strategy proposed by Chen *et al.* [23] and also adopted by Gao *et al.* [31]. In this strategy the weight for each group is proportional to the square root of the length of the group, so $\omega_g^{dic} = \sqrt{|g|}$ $(g \in G_{dic})$ and $\omega_g^{tst} = \sqrt{|g|}$ $(g \in G_{tst})$. Then we set $\gamma_{dic} = \gamma_{tst} = \lambda = \theta$. In this way there is only one parameter, θ, in the whole objective function. In our experiments we set $\theta = 10^{-3}$.

The number of atoms in a dictionary block (AU/structure-block) and the sparsity are other parameters needed by the K-SVD [18] algorithm which is used for data-driven dictionary learning. In our experiments, we learn a dictionary of size $J_i = 20$ per structure/AU and set the sparsity to be half of the dictionary size, i.e. 10. Our experiments show that



(a) Happy (b) Happy

(c) Surprise (d) Fear

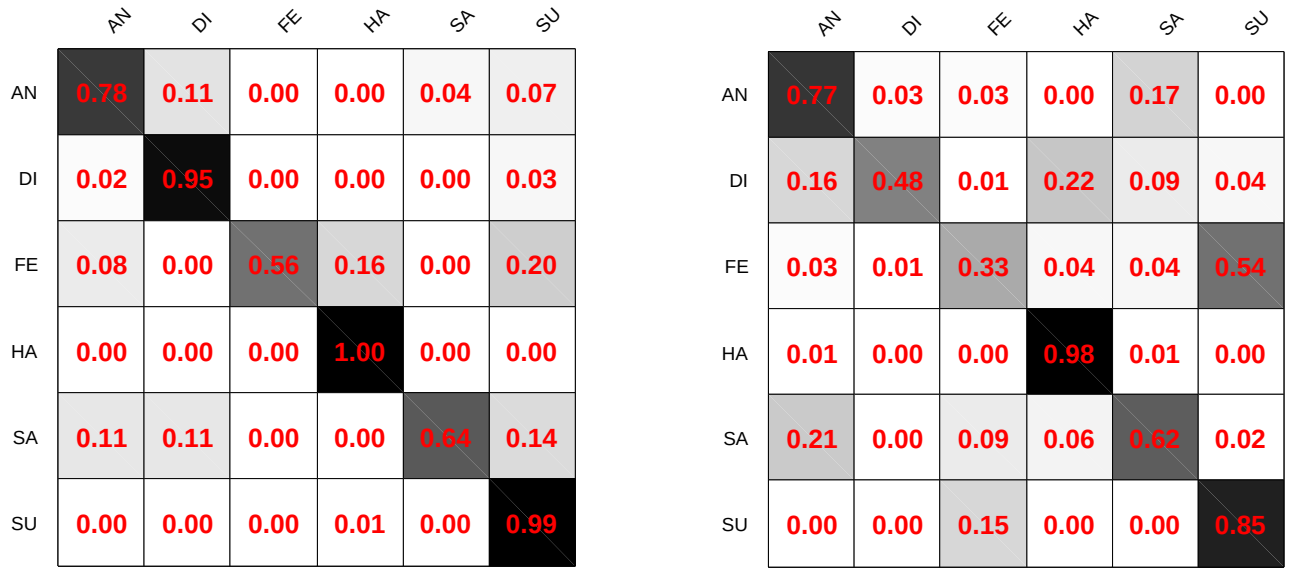
Fig. 7. Expression decomposition into constituent AUs. Each example includes the matrix of sparse codes, X, on the right column, the AU bar plot which shows the L^2-norm of each AU-block in the sparse code matrix, the sample faces with those AUs that have significant magnitude on top of the bar plot, and finally the original and reconstructed expressive faces below the bar graph. The decompositions reveal the correct constituent AUs for each expressive face.

these choices ensure a good trade-off between the accuracy of the representation and the speed of the algorithm. A larger dictionary (up to a point) may slightly improve the results but at the cost of slower convergence of the sparse coding algorithm.

B. Expression Decomposition

As discussed in section V-A, the AU-dictionary can be used to decompose an expressive face into its constituent AUs. In fact the main advantage of modeling AUs for expression analysis is that we are not limited to the six universal expressions, and many other expressions that often occur on the face but do not belong to these universal expressions can also be analyzed by predicting the AUs they are composed of. For this experiment, we learn the AU-dictionary using a selected subset of 15 AUs in the Bosphorus dataset. This subset consists of six upper face AUs (1, 2, 4, 5, 6, and 43) and nine lower face AUs (12, 17, 20, 22, 24, 25, 26, 27 and 28). To learn the AU-dictionary, we first delineate the patches that correspond to each AU semantic region. For simplicity we only define seven regions on the face corresponding to brows, eyes, nose, mouth and forehead. It should be noted that AU-5 and AU-6 are not included in the Bosphorus dataset; however, these two AUs occur in many expressions, so we learn their corresponding dictionary blocks using the happy and surprise samples in the dataset. However, we avoid using the same samples for dictionary training and decomposition testing.

Observations: Figure 7 shows some examples of expression decomposition we performed using expressive samples from the Bosphorus dataset. The figure shows decomposition results for two happy faces of different subjects. Both decompositions predict AUs-{06+12+25} which is in accordance with the information in Fig. 3. However in part (b), AU-43 has also been reported as one of the constituent components of the happy face. This can be due to the extreme smile on the face which makes the eyes look almost closed. Parts (c) and (d) illustrate the decomposition for two expressive faces with Surprise and Fear expressions. The decompositions reveal the correct



(a) CK+ (b) Bosphorus

Fig. 8. Confusion matrix for emotion recognition on the (a) CK+ dataset, (b) Bosphorus dataset. The number of samples per classes for the CK+ dataset is: (AN:45, DI:59, FE:25, HA:69, SA:28, SU:79) and for the Bosphorus dataset is: (AN:71, DI:69, FE:70, HA:105, SA:66, SU:71).

TABLE I
COMPARING THE AVERAGED RECOGNITION RATE USING THREE ALGORITHMS ON CK+.

Algorithms	**Recognition Rates**
Multi-layer multi-variable grouping	88.52%
Multi-layer single grouping	80.2%
Simple Lasso	69.68%

constituent AUs. It should be noted that in these experiments as well as in the expression synthesis experiments, in order to be able to visualize the reconstructed faces we concatenated the SIFT features extracted from each patch with the intensity difference image (the difference between the neutral face and the expressive face in that patch). So to visualize we add the reconstructed intensity difference feature to the neutral face.

C. Expression Recognition

In this section we report the results of expression recognition using the learned structure-preserving dictionaries for both the CK+ and Bosphorus datasets. We adopt the leave-one-subject-out cross-validation configuration to maximize the amount of training and testing data. Figure 8 shows the confusion matrices for both datasets. It should be noted that for the CK+ dataset we only use the apex of each expression sequence for recognition. The average recognition rate on the CK+ dataset is 88.52% and on the Bosphorus dataset is 69.78%. It should be noted that while both the CK+ and the Bosphorus are datasets with posed expressions, but there are main discrepancies in their annotations. The CK+ was first FACS coded manually, then emotion labels were assigned based on FACS rules, while in the Bosphorus the subjects were asked to show the given emotion/AU and hence the emotion labels might not correspond to the given AU combinations. This explains the lower recognition rate on Bosphorus compared to CK+. Also in the Bosphorus dataset, there is a large degree of similarity among faces of some classes (e.g. Fear and Surprise).

To show the importance of incorporating high-level knowledge on grouping AUs, we compare our algorithm with two other algorithms on the CK+ dataset (Table I). In the first algorithm, multi-layer single grouping [23], we removed the grouping information on the image matrix and so the sparse representation for each local descriptor extracted from the test face is obtained independent of others. In the second algorithm both grouping information on the dictionary and image matrix are ignored and a simple LASSO algorithm [34] is applied to learn sparse representation for each local descriptor. As Table I shows, removing each of the high-level grouping information decreases the recognition rate. These results again emphasize the importance of incorporating the high-level information for expression analysis.

Table II compares the average recognition rate of our algorithm with some recent advances in expression recognition for the CK/CK+ datasets. The CK dataset [35] is the old version of CK+ which has fewer subjects and sequences. Most of these algorithms did not follow leave-one-subject-out validation, but instead they divided the dataset randomly into training and testing parts and reported the results on the testing part. As the table shows, our result is comparable to the best reported results. However, it should be noted that as our algorithm is a generative approach, it is expected that its classification performance be lower than fully discriminative methods like Adaboost. But the proposed approach can also

TABLE II
COMPARISON WITH RECENT ADVANCES IN EXPRESSION RECOGNITION ON CK DATASET.

Recognition Methods	Recognition Rates
CERT [7] *	87.21%
Gabor+Adaboost+SVM [32]	93.3%
PGKNMF‡ [33]	83.5%
CAPP⋆ +SVM [12] †	86.48%
RegRankBoost [24]	88%
Combined Features+Adaboost [6]	92.3%
Our approach	88.52%

∗ results on 26 subjects in CK+ but not in CK
‡ PGKNMF = Projected Gradient Kernel Non-negative Matrix Factorization
⋆ CAPP = Canonical Appearance Features
† results on CK+ with leave-one-subject-out validation

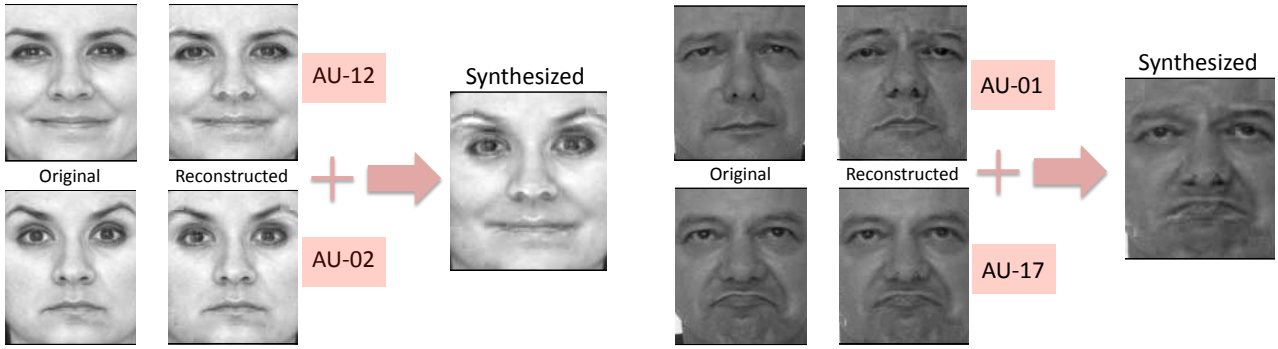


Fig. 9. Expression synthesis using the AU-dictionary.

perform unknown expression decomposition and new expression synthesis which are not possible with pure discriminative algorithms. Moreover, our algorithm provides a general framework for decomposing an action in terms of its constituent basic components. Using a better feature representation for expressive faces as well as a machine learning algorithm for learning over the sparse code matrix, we can expect a higher recognition rate.

D. Expression Synthesis

Combining different AUs can result in new expressions on the face. Using the AU-dictionary, we perform this experiment to generate valid expressions through composition of some AUs. For this purpose, first we need valid models for sparse coefficients corresponding to each AU. We obtain these models for each subject by decomposing the sample of a particular AU onto the AU-dictionary. Then by selecting a subset of these AUs and assigning values to the corresponding coefficients in the sparse matrix we are able to synthesize new expressions. However, it should be noted that this approach does not work for non-additive AUs (discussed in section. III-B). Figure 9 illustrates two examples of expression synthesis.

VIII. CONCLUSION

We presented a dictionary-based approach for facial expression analysis. Using the domain experts' knowledge provided in FACS, we learned an AU-dictionary. We also proposed an automatic algorithm to learn a structure-preserving dictionary without incorporating the experts' knowledge. Then the high-level knowledge regarding the AUs/structures composition is incorporated into this dictionary through a multi-layer grouping of dictionary atoms. Since we are also dealing with a multi-variable problem, we impose appropriate groupings over the test image matrix columns. We employed a multi-layer, multi-variable group sparse coding algorithm to impose these grouping constraints for structure-preserving sparse coding. This enables us to perform expression decomposition into the constituent AUs for a face with any unknown expression. Using the sparse code matrix we can also perform universal emotion recognition. The results indicate the improvements that this high-level knowledge brings to expression analysis. Moreover, since the proposed algorithm is a generative approach, we can also perform new expression synthesis. We show some preliminary results for expression synthesis.

The paper presented a general framework for decomposing an action into its constituent components. We showed the potential of this algorithm for facial expression analysis, including expression decomposition, synthesis and recognition. The proposed algorithm can be generalized to recognition of human actions provided we have a good definition for human action units. This algorithm can be further improved by adding temporal sequence information as a new grouping layer. Incorporating temporal information enables us to predict the intensity of AUs and detect AUs that are combined sequentially to form an expression/action.

REFERENCES

[1] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *TPAMI*, 2009.

[2] P. Ekman and W. Friesen, *The Facial Action Coding System: a technique for the measurement of facial movement*. Consulting Psychologists Press., 1978.

[3] Y. Tian, T. Kanade, and J. F. Cohn, *Facial Expression Recognition*. Springer, 2011.

[4] M. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer, "Meta-analysis of the first facial expression recognition challenge meta-analysis of the first facial expression recognition challenge meta-analysis of the first facial expression recognition challenge," *IEEE Transactions on Systems, Man, and Cybernetics*, 2012.

[5] M. Valstar and M. Pantic, "Biologically vs. logic inspired encoding of facial actions and emotions in video," in *IEEE Int'l. Conf. on Multimedia and Expo*, 2006.

[6] P. Yang, Q. Liu, and M. Dimitris, "Exploring facial expression with compositional features," in *CVPR*, 2010.

[7] G. Littlewort, J. Whitehill, T. Wu, I. R. Fasel, M. G. Frank, J. R. Movellan, and M. S. Bartlett, "The computer expression recognition toolbox (cert)," in *FG*, 2011.

[8] Y. Zhu, F. De la Torre, J. F. Cohn, and Y.-J. Zhang, "Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior," *TAC*, 2011.

[9] Y. Tong, J. Chen, and Q. Ji, "A unified probabilistic framework for spontaneous facial action modeling and understanding," *TPAMI*, 2010.

[10] O. Rudovic, I. Patras, and M. Pantic, "Coupled Gaussian process regression for pose-invariant facial expression recognition," in *ECCV*, 2010.

[11] T. Simon, N. Minh, F. De la Torre, and J. F. Cohn, "Action unit detection with segment-based SVMs," in *CVPR*, 2010.

[12] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *CVPR Workshop*, 2010.

[13] P. Yang, Q. Liu, X. Cui, and D. Metaxas, "Facial expression recognition using encoded dynamic features," *CVPR*, 2008.

[14] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2D+3D active appearance models," in *CVPR*, 2004, pp. 535–542.

[15] W.-K. Liao and G. Medioni, "3D face tracking and expression inference from a 2D sequence using manifold learning," in *CVPR*, 2008.

[16] Y. Chang, C. Hu, and M. Turk, "Probabilistic expression analysis on manifolds," in *CVPR*, 2004.

[17] S. Taheri, P. Turaga, and R. Chellappa, "Towards view-invariant expression analysis using analytic shape manifolds," in *FG*, 2011.

[18] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: Design of dictionaries for sparse representation," *Trans. on Signal Processing*, 2006.

[19] Q. Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in *ICCV*, 2011.

[20] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *CVPR*, 2011.

[21] M. Mahoor, M. Zhou, K. Veon, M. Mavadati, and J. Cohen, "Facial action unit recognition with sparse representation," in *FG*, 2011.

[22] G. Yu, G. Sapiro, and S. M. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity," *CoRR*, 2010.

[23] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "An efficient proximal gradient method for general structured sparse learning," *Journal of Machine Learning*, 2011.

[24] P. Yang, Q. Liu, and D. N. Metaxas, "RankBoost with L1 regularization for facial expression recognition and intensity estimation," in *ICCV*, 2009.

[25] J. H. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, 2010.

[26] L. Meier, S. V. D. Geer, and P. Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society*, 2008.

[27] M. Lobo, L. Vandenberghe, S. Boyd, , and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1998.

[28] X. Chen, Q. Lin, S. Kim, J. Pena, J. G. Carbonell, and E. P. Xing, "An efficient proximal-gradient method for single and multi-task regression with structured sparsity," CMU, Tech. Rep., 2010.

[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[30] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Workshop on Biometrics and Identity Management*, 2008.

[31] S. Gao, L.-T. Chia, and I. W. Tsang, "Multi-layer group sparse coding - for concurrent image classification and annotation," in *CVPR*, 2011.

[32] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: Machine learning and application to spontaneous behavior," in *CVPR*, 2005.

[33] S. Zafeiriou and M. Petrou, "Nonlinear non-negative component analysis algorithms," *TIP*, 2010.

[34] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Royal. Statist.*, 1996.

[35] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *FG*, 2000.