Linear-Time Subspace Clustering via Bipartite Graph Modeling

Amir Adler, Michael Elad, *Fellow, IEEE*, and Yacov Hel-Or

***Abstract*— We present a linear-time subspace clustering approach that combines sparse representations and bipartite graph modeling. The signals are modeled as drawn from a union of low-dimensional subspaces, and each signal is represented by a sparse combination of basis elements, termed *atoms*, which form the columns of a dictionary matrix. The sparse representation coefficients are arranged in a sparse affinity matrix, which defines a bipartite graph of two disjoint sets: 1) atoms and 2) signals. Subspace clustering is obtained by applying low-complexity spectral bipartite graph clustering that exploits the small number of atoms for complexity reduction. The complexity of the proposed approach is linear in the number of signals, thus it can rapidly cluster very large data collections. Performance evaluation of face clustering and temporal video segmentation demonstrates comparable clustering accuracies to state-of-the-art at a significantly lower computational load.**

***Index Terms*— Bipartite graph, dictionary, face clustering, sparse representation, subspace clustering, temporal video segmentation.**

I. Introduction

DIMENSIONALITY reduction is a powerful tool for processing high-dimensional data, such as video, image, audio, and biomedical signals. The simplest of such techniques is probably principal component analysis (PCA) that models the data as spanned by a single low-dimensional subspace; however, in many cases, a *union-of-subspaces* model can more accurately represent the data: for example, Vidal *et al.* [1] proposed to generalize PCA to identify multiple subspaces for computer vision applications, Ho *et al.* [2] proposed to generalize k-means to cluster facial images, and Lu and Do [3] proposed efficient sampling techniques for practical signal types that emerge from a union-of-subspaces model. Subspace clustering is the problem of clustering a collection of signals drawn from a union of subspaces, according to their spanning subspaces. Subspace clustering algorithms can be divided into four approaches: 1) statistical; 2) algebraic; 3) iterative; and 4) spectral clustering based; see [4] for a review. State-of-the-art approaches, such as sparse subspace clustering (SSC) [5], [6], low-rank representation (LRR) [7], [8], and low-rank subspace clustering (LR-SC) [9], are spectral-clustering based. These methods provide excellent performance, however, their complexity limits the size of the data sets to $\approx 10^4$ signals. K-subspaces [2] is a generalization of the K-means algorithm to subspace clustering that can handle large data sets. However, it requires explicit knowledge of the dimensions of all subspaces and its performance is inferior compared with state-of-the-art. In this paper, we address the problem of applying subspace clustering to very large data collections. This problem is important due to the following reasons.

1) Existing subspace clustering tasks are required to handle the ever-increasing amounts of data, such as image and video streams.
2) Subspace-clustering-based solutions could be applied to applications that traditionally could not employ subspace clustering and require large data processing.

In the following, we formulate the subspace clustering problem, review previous works based on sparse and low-rank modeling, and highlight the properties of our approach.

A. Problem Formulation

Let $\mathbf{Y} \in \mathbb{R}^{N \times L}$ be a collection of L signals $\{\mathbf{y}_l \in \mathbb{R}^N\}_{l=1}^L$, drawn from a union of $K > 1$ linear subspaces $\{\mathbf{S}_i\}_{i=1}^K$. The bases of the subspaces are $\{\mathbf{B}_i \in \mathbb{R}^{N \times d_i}\}_{i=1}^K$ and $\{d_i\}_{i=1}^K$ are their dimensions. The task of subspace clustering is to cluster the signals according to their subspaces. The number of subspaces K is either assumed known or estimated during the clustering process. The difficulty of the problem depends on the following parameters.

1) *Subspaces Separation:* The subspaces may be independent (as defined in Appendix A) or disjoint, or some of them may have a nontrivial intersection, which is considered as the most difficult case.
2) *Signal Quality:* The collection of signals Y may be corrupted by noise, missing entries, or outliers, thus distorting the true subspaces structure.
3) *Model Accuracy:* The union-of-subspaces model is often only an approximation of a more complex and unknown data generation model, and the magnitude of the error it induces affects the overall performance.

B. Prior Art: Sparse and Low Rank Modeling

SSC and LRR reveal the relations among signals by finding a self-expressive representation matrix $\mathbf{W} \in \mathbb{R}^{L \times L}$ such that

Manuscript received November 5, 2013; revised May 30, 2014 and October 22, 2014; accepted October 27, 2014. Date of publication February 20, 2015; date of current version September 16, 2015.

A. Adler and M. Elad are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: adleram@cs.technion.ac.il; elad@cs.technion.ac.il).

Y. Hel-Or is with the Department of Computer Science, Interdisciplinary Center Herzlia, Herzlia 46150, Israel (e-mail: toky@idc.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2374631

$\mathbf{Y} \simeq \mathbf{YW}$ and obtain subspace clustering by applying spectral clustering [10] to the graph induced by $\mathbf{W}$. SSC forces $\mathbf{W}$ to be sparse by solving the following set of optimization problems, for the case of signals contaminated by noise with standard deviation ε in [5, Sec. 3.3]:

$$\min_{\mathbf{w}_i} \|\mathbf{w}_i\|_1 \quad \text{s.t. } \|\mathbf{Y}_{\hat{i}}\mathbf{w}_i - \mathbf{y}_i\|_2 \le \varepsilon \qquad (\text{for } i = 1, \ldots, L) \quad (1)$$

where $\mathbf{w}_i \in \mathbb{R}^{L-1}$ is the sparse representation vector, $\mathbf{y}_i$ is the ith signal, and $\mathbf{Y}_{\hat{i}}$ is the signal matrix $\mathbf{Y}$ excluding the ith signal. By inserting a zero at the ith entry of $\mathbf{w}_i$ and augmenting the dimension of $\mathbf{w}_i$ to L, the vector $\hat{\mathbf{w}}_i \in \mathbb{R}^L$ is obtained, which defines the ith column of $\mathbf{W} \in \mathbb{R}^{L\times L}$, such that $\text{diag}(\mathbf{W}) = 0$. For the case of signals with sparse outlying entries, SSC forces $\mathbf{W}$ to be sparse by solving the following optimization problem:

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_1 + \lambda \|\mathbf{E}\|_1 \quad \text{s.t. } \mathbf{Y} = \mathbf{YW} + \mathbf{E} \text{ and } \text{diag}(\mathbf{W}) = 0 \quad (2)$$

where E is a sparse matrix representing the sparse errors in the data and $\lambda > 0$. LRR forces $\mathbf{W}$ to be low rank by minimizing its nuclear norm (sum of singular values) and solves the following optimization problem for clustering signals contaminated by noise and outliers:

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_* + \lambda \|\mathbf{E}\|_{2,1} \quad \text{s.t. } \mathbf{Y} = \mathbf{YW} + \mathbf{E}. \quad (3)$$

SSC was reported to outperform agglomerative lossy compression [11] and RANdom SAample Consensus, [12], whereas LRR was reported to outperform local subspace affinity [13] and generalized-PCA [1]. LRR and SSC provide excellent performances; however, they are restricted to relatively moderate-sized data sets due the following reasons.

1) Polynomial complexity affinity calculation—SSC solves L sparse coding problems with a dictionary of $L - 1$ columns, leading to an approximate complexity of $O(L^2)$. The complexity of LRR is higher as its augmented Lagrangian-based solution involves repeated Singular Value Decomposition (SVD), computations of an $L \times L$ matrix during the convergence to W, leading to complexity of $O(L^3)$ multiplied by the number of iterations (which can exceed 100).
2) Polynomial complexity spectral clustering—Both LRR and SSC require eigenvalue decomposition (EVD) of an $L \times L$ Laplacian matrix, leading to polynomial complexity of the spectral clustering stage.[1] In addition, the memory space required to store the entries of the graph Laplacian is $O(L^2)$, which becomes prohibitively large for $L \gg 1$.

In addition, whenever the entire data set is contaminated by noise, both LRR and SSC suffer from degraded performance since each signal in Y is represented by a linear combination of other *noisy* signals. LR-SC [9] provides closed-form solutions for noisy data and iterative algorithms for data with outliers. LR-SC provides solutions for noisy data by introducing the clean data matrix Q and solving relaxations of the following problem:

$$\min_{\mathbf{W},\mathbf{E},\mathbf{Q}} \|\mathbf{W}\|_* + \lambda \|\mathbf{E}\|_F \quad \text{s.t. } \mathbf{Q} = \mathbf{QW} \text{ and } \mathbf{Y} = \mathbf{Q} + \mathbf{E}. \quad (4)$$

Note that the computational load of the spectral clustering stage remains the same as that of LRR and SSC since the dimensions of the affinity matrix remains $L \times L$. The clustering accuracy of LR-SC was reported as comparable with SSC and LRR, while better than agglomerative lossy compression [11], local subspace affinity [13], and shape interaction matrix [14]. A fast version of LR-SC was proposed in [15] that achieves a complexity of $O(L \log(L))$ by utilizing a partial SVD to approximate the solution of LR-SC and by employing locality sensitive hashing to construct a sparse affinity matrix. A scalable version of SSC was proposed in [16] that employs four main steps: sampling, clustering, coding, and classifying. The first two steps select a small number of data points as in-sample data and perform SSC over it. The latter steps encode out-of-sample data as a linear combination of in-sample data and assign the out-of-sample data points to the subspace clusters by classification.

Sprechmann and Sapiro [17] proposed a Lloyd-type algorithm that alternates between learning a set of K subdictionaries (for K data classes) and an assignment of each signal to the best subdictionary that represents it. This method differs from the proposed approach in three aspects.

1) It employs either standard spectral clustering or recursive graph clustering for the subdictionaries' initialization stage.
2) It performs jointly the task of dictionary learning and clustering.
3) The number of atoms used in [17] is an order of magnitude higher than that used in the proposed approach.

The complexity of this approach is approximately an order of magnitude higher than that of the proposed approach, mostly due to the signals' classification stage that occurs in each stage of the subdictionaries learning. Probabilistic Sparse Subsapce Clustering (PSSC) [18] proposed a dictionary-based approach, which employs a probabilistic mixture model to compute signals likelihoods and obtains subspace clustering using a maximum-likelihood rule. PSSC treats the sparse representations matrix as a cooccurrence matrix of atoms and signals, and decomposes the cooccurrence matrix into the product of two probability matrices: the first is the joint probabilities of atoms and subspaces and the second is the likelihood probabilities of each signal for the given each subspace. The key differences between the proposed approach and PSSC are the following.

1) The sparse representation matrix is utilized by the proposed approach to define a bipartite graph.
2) Subspace clustering is obtained by a spectral bipartite graph clustering approach. PSSC offers linear-time complexity; however, its performance is inferior to the proposed approach in low signal-to-noise ratios (SNRs) and real-life datasets.

[1] Note that a full EVD of the Laplacian has a complexity of $O(L^3)$; however, a complexity of $O(L^2)$ is required for computing only several eigenvectors.

C. Paper Contributions

This paper presents a new spectral clustering-based approach that is built on sparsely representing the given signals using a dictionary, which is either learned or known *a priori*.[2] The matrix of sparse representations is used to construct the affinity matrix of a bipartite graph, which is segmented by a linear-time spectral bipartite clustering algorithm. The contributions of this paper are as follows.

1) *Bipartite Graph Modeling:* A novel solution to the subspace clustering problem is obtained by mapping the sparse representation matrix to an affinity matrix that defines a bipartite graph with two disjoint sets of vertices: dictionary atoms and signals.
2) *Linear-Time Complexity:* The proposed approach exploits the small number of atoms M for complexity reduction, leading to an overall complexity that depends only linearly on the number of signals L.
3) *Theoretical Study:* The conditions for correct clustering of independent subspaces are proved for the cases of minimal and redundant dictionaries.

This paper is organized as follows. Section II overviews sparse representation modeling that forms the core for learning the relations between signals and atoms. Section III presents bipartite graphs and the proposed approach. Section IV provides the performance evaluation of the proposed approach and compares it with leading subspace clustering algorithms.

II. Sparse Representations Modeling

Sparse representations provide a natural model for signals that live in a union of low-dimensional subspaces. This modeling assumes that a signal $\mathbf{y} \in \mathbb{R}^N$ can be described as $\mathbf{y} \simeq D\mathbf{c}$, where $\mathbf{D} \in \mathbb{R}^{N\times M}$ is a *dictionary* matrix and $\mathbf{c} \in \mathbb{R}^M$ is sparse. Therefore, $\mathbf{y}$ is represented by a linear combination of a *few* columns (atoms) of $\mathbf{D}$. The recovery of $\mathbf{c}$ can be cast as an optimization problem

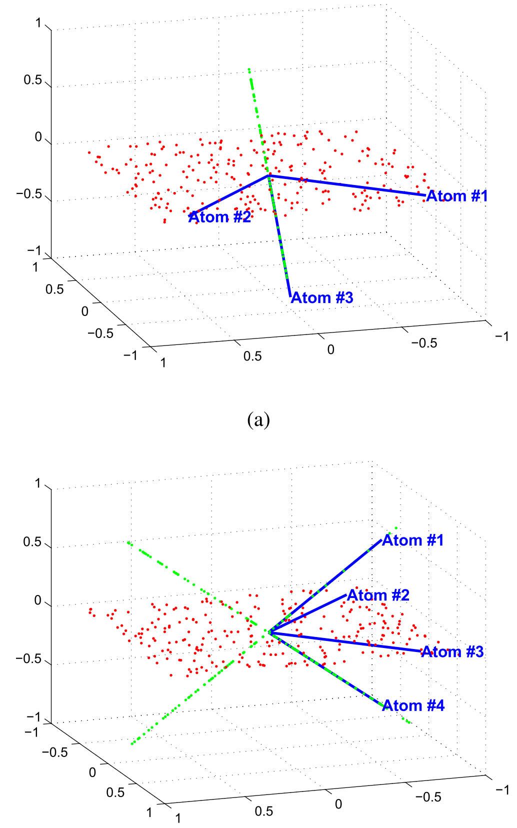
$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{D}\mathbf{c}\|_2 \le \varepsilon \tag{5}$$

for some approximation error threshold ε. The l_0 norm $\|\mathbf{c}\|_0$ counts the nonzeros components of $\mathbf{c}$, leading to a Nondeterministic Polynomial time (NP)-hard problem. Therefore, a direct solution of (5) is infeasible. An approximate solution is given by applying the orthogonal matching pursuit (OMP) algorithm [21], which successively approximates the sparsest solution. The recovery of $\mathbf{c}$ can also be cast by an alternative optimization problem that limits the cardinality of $\mathbf{c}$

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{y}-\mathbf{D}\mathbf{c}\|_2 \quad \text{s.t.} \quad \|\mathbf{c}\|_0 \le T_0 \tag{6}$$

where T_0 is the maximum cardinality. The dictionary $\mathbf{D}$ can be either predefined or learned from the given set of signals (see [22] for a review). For example, the K-SVD algorithm [23] learns a dictionary by solving the following optimization problem:

$$\{\mathbf{D},\mathbf{C}\} = \arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{Y}-\mathbf{D}\mathbf{C}\|_F^2 \quad \text{s.t.} \quad \forall i\ \|\mathbf{c}_i\|_0 \le T_0 \tag{7}$$



(a)

(b)

Fig. 1. Dictionary learning of (a) independent and (b) disjoint subspace bases.

where $\mathbf{Y} \in \mathbb{R}^{N\times L}$ is the signal matrix containing $\mathbf{y}_i$ in its ith column and $\mathbf{C} \in \mathbb{R}^{M\times L}$ is the sparse representation matrix, containing the sparse representation vector $\mathbf{c}_i$ in its ith column. Once the dictionary is learned, each one of the signals $\{\mathbf{y}_i\}_{i=1}^L$ is represented by a linear combination of few atoms. Each combination of atoms defines a low-dimensional subspace; thus, our subspace clustering approach exploits the fact that signals spanned by the same subspace are represented by similar groups of atoms. In the following, we demonstrate this property for signals that are drawn from a union of independent or disjoint subspaces (as defined in Appendix A). Consider data points drawn from a union of two independent subspaces in $\mathbb{R}^3$: a plane and a line, as shown in Fig. 1(a). A dictionary with three atoms was learned from a few hundreds of such points using the K-SVD algorithm, and as shown in Fig. 1(a), the learned atoms span the two subspaces. Next, consider data points drawn from a union of three disjoint subspaces in $\mathbb{R}^3$: a plane and two lines, as shown in Fig. 1(b). A dictionary with four atoms was learned from a few hundreds of such points using the K-SVD algorithm, and as shown in Fig. 1(b), the learned atoms span the three subspaces.

III. Proposed Approach

A. From Bipartite Graphs to Subspace Clustering

The sparse representations matrix C provides an explicit information on the relations between signals and atoms, which

[2] For example, ODCT-based dictionaries are well suited for sparsely representing image patches [19] or audio frames [20].

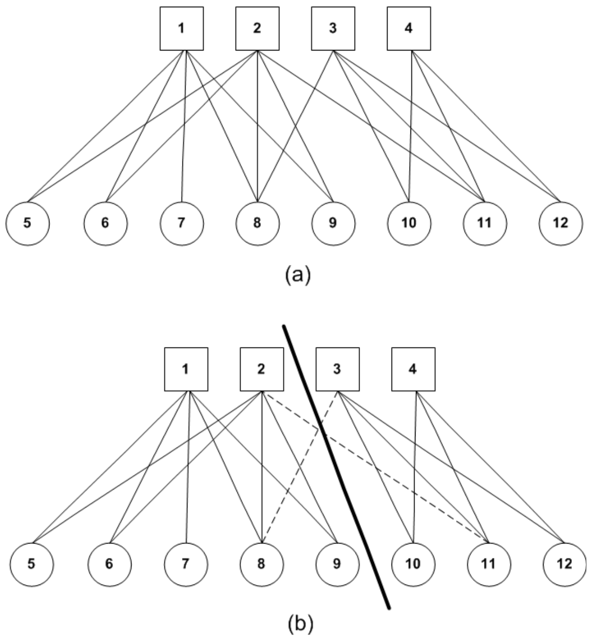


Fig. 2. (a) Bipartite graph consisting of 12 vertices: four atoms (squares) and eight signals (circles). (b) Signals were drawn from a union of two subspaces; however, the sparse coding stage (OMP) produced inseparable groups in the graph. The normalized cut approach attempts to resolve this by grouping together signals with the atoms that are the most significant in the signals' representations. The edges that correspond to the least significant links between atoms to signals are neglected (dashed edges in the figure). The graph partitioning solution is shown by the bold line: the vertices of the first group are {1, 2, 5, 6, 7, 8, 9} and the vertices of the second group are {3, 4, 10, 11, 12}.

we leverage to quantify the latent relations among the signals: the locations of nonzero coefficients in C determine the atoms that represent each signal and their absolute values determine the respective weights of the atoms in each representation. Therefore, subspace clustering can be obtained by a *biclustering* approach: simultaneously grouping signals with the atoms that represent them, such that a cluster label is assigned to every signal and every atom and the labels of the signals provide the subspace clustering result. In cases where a partition into disjoint groups does not exist (as a result of intersecting subspaces, or errors in the sparse coding stage or noise), a possible approach is to group together signals with the most significant atoms that represent them. This *biclustering* problem can be solved by bipartite graph partitioning [24]: let $G = (\mathcal{D}, \mathcal{Y}, E)$ be an undirected bipartite graph consisting of two disjoint sets of vertices: atoms $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_M\}$ and signals $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$, connected by the corresponding set of edges E. An edge between an atom and a signal exists only when the atom is part of the representation of the signal. The two disjoint sets of vertices are enumerated from 1 to $M + L$: the leading M vertices are atoms and the tailing L vertices are signals, as shown in Fig. 2(a). Let $\mathbf{W} = \{w_{ij}\}$ be a nonnegative affinity matrix such that every pair of vertices is assigned a weight w_{ij}. The affinity matrix is defined by

$$\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)} \tag{8}$$

where $\mathbf{A} = |\mathbf{C}|$ (each element A_{ij} equals to $|C_{ij}|$). Note that the structure of $\mathbf{W}$ implies that only signal-atom pairs can be assigned a positive weight (i.e., when the atom is part of the representation of the signal). The matrix $\mathbf{W}$ is used to define the set of edges, such that an edge between the ith and jth vertices exists in the graph only if $w_{ij} > 0$ and the weight of this edge is $e_{ij} = w_{ij}$. Thus, the particular structure of $\mathbf{W}$ imposes only one type of connected components: bipartite components that are composed of at least one atom and one signal. This type of graph modeling differs from the modeling employed by LRR, SSC, and LR-SC since these methods construct a graph with only a single type of vertices (which are signals) and seek for groups of connected signals. In addition, bipartite graph modeling differs from Sprechmann and Sapiro's method [17] that partitions either a graph of atoms or a graph signals (each graph with only a single type of vertices) as an initialization stage of the K subdictionaries learning algorithm.

A reasonable criterion for partitioning the bipartite graph is the normalized cut [10], which seeks well-separated groups while balancing the size of each group, as shown in Fig. 2(b). Let $\mathcal{V}_1, \mathcal{V}_2$ be a partition of the graph such that $\mathcal{V}_1 = \mathcal{D}_1 \cup \mathcal{Y}_1$ and $\mathcal{V}_2 = \mathcal{D}_2 \cup \mathcal{Y}_2$, where $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}$ and $\mathcal{Y}_1 \cup \mathcal{Y}_2 = \mathcal{Y}$. The normalized cut partition is obtained by minimizing the following expression:

$$N_{\text{cut}}(\mathcal{V}_1, \mathcal{V}_2) = \frac{\text{cut}(\mathcal{V}_1, \mathcal{V}_2)}{\text{weight}(\mathcal{V}_1)} + \frac{\text{cut}(\mathcal{V}_1, \mathcal{V}_2)}{\text{weight}(\mathcal{V}_2)} \tag{9}$$

where $\text{cut}(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i\in\mathcal{V}_1, j\in\mathcal{V}_2} W_{ij}$ quantifies the accumulated edge weights between the groups and weight $(\mathcal{V}) = \sum_{i\in\mathcal{V}} \sum_k W_{ik}$ quantifies the accumulated edge weights within a group. Therefore, we propose to partition the bipartite graph using the normalized cut criterion, and obtain subspace clustering from the signals' cluster labels.

Direct minimization of (9) leads to an NP-hard problem, therefore, spectral clustering [10] is often employed as an approximate solution to this problem. A low-complexity bipartite spectral clustering algorithm was derived in [24] for natural language processing applications. This algorithm is detailed in Appendix B and requires the SVD of an $M \times L$ matrix, which has a complexity of $O(M^2L)$ [25]. Note that in our modeling, the number of atoms is fixed and obeys $M \ll L$, leading to complexity that depends linearly in L (compared with the complexity of the spectral clustering stage of state-of-the-art approaches [6], [8], [9] that is polynomial is L). We leverage the SVD-based algorithm to our problem and incorporate it into the proposed algorithm, as detailed in Algorithm 1. The overall complexity of the proposed approach depends only linearly on L, and is given by $O(q\text{JNML}) + O(q\text{NML}) + O(M^2L) + O(\text{TMKL})$, where the first term is K-SVD complexity (q is the average cardinality of the sparse representations, J the number of iterations, and $L \gg 1$), the second term is OMP complexity, and the third (SVD complexity) and fourth (k-means complexity with T iterations) terms compose the bipartite spectral clustering stage complexity.

B. Theoretical Study

In the following, we provide two theorems that pose conditions for correct segmentation of independent subspaces using

```
Algorithm 1 Subspace Biclustering (SBC)
  Input: data $\mathbf{Y} \in \mathbb{R}^{N \times L}$, # of clusters $K$, # of atoms $M$. 1.
    1) Dictionary Learning: Employ $K$-SVD to learn a dictionary $\mathbf{D} \in \mathbb{R}^{N \times M}$ from $\mathbf{Y}$.
    2) Sparse Coding: Construct the sparse matrix $\mathbf{C} \in \mathbb{R}^{M \times L}$ by the OMP algorithm, such that $\mathbf{Y} \simeq \mathbf{DC}$.
    3) Bi-Clustering:I.
      a) Construct the matrix $\mathbf{A} = |\mathbf{C}|$.
      b) Compute the rank-M SVD of $\overline{\mathbf{A}} = \mathbf{D}_1^{-\frac{1}{2}} \mathbf{A} \mathbf{D}_2^{-\frac{1}{2}}$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ as in equation (11).
      c) Construct the matrix $\mathbf{Z} = \begin{bmatrix} \mathbf{D}_1^{-\frac{1}{2}} \mathbf{U} \\ \mathbf{D}_2^{-\frac{1}{2}} \mathbf{V} \end{bmatrix}$, where $\mathbf{U} = [\mathbf{u}_2 ... \mathbf{u}_K]$ and $\mathbf{V} = [\mathbf{v}_2 ... \mathbf{v}_K]$ as in equation (17). The $M$ leading rows of $\mathbf{Z}$ correspond the atoms and the $L$ tailing rows correspond the signals.
      d) Cluster the rows of $\mathbf{Z}$ using k-means.
  Output: cluster labels for all signals $\hat{k}(y_j)$, $j = 1..L$.
```

the proposed approach. Our analysis proves that given a correct dictionary, OMP will always recover successfully the bipartite affinity matrix.[3] Further segmentation of the bipartite graph using the normalized cut criterion leads to correct subspace clustering. The following theorem addresses the case of a dictionary $\mathbf{D}$ that contains the set of minimal bases for all subspaces.

Theorem 1: Let $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K]$ be a collection of $L = L_1 + L_2 + \cdots + L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^K$. Given a dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N \times d_i}$ spans $\mathbf{S}_i$ and $d_i = \dim(\mathbf{S}_i)$, OMP is guaranteed to recover the correct and unique sparse representation matrix $\mathbf{C}$ such that $\mathbf{Y} = \mathbf{DC}$, and minimization of the normalized cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.

The proof is provided in Appendix C.

We now address the more general case of a redundant dictionary in which the subdictionaries are redundant $\mathbf{D}_i \in \mathbb{R}^{N \times t_i}$ and $t_i > d_i$. This situation is realistic in dictionary learning, whenever the number of allocated atoms is higher than necessary. Note that for a redundant dictionary, there is an infinite number of exact representations for each signal $\mathbf{y}_i \in \mathbf{S}_i$, and OMP is prone to select wrong atoms (that represent subspaces $\mathbf{S}_j \neq \mathbf{S}_i$) during its operation. However, the following theorem proves that the support of the OMP solution is guaranteed to include atoms only from the correct subspace basis (although the accumulated[4] support set might contain atoms that represent other subspaces). Fig. 3 shows this in practice.

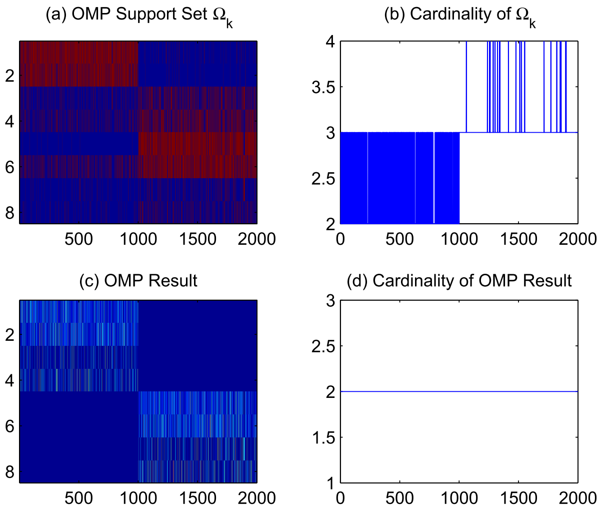


Fig. 3. Sparse representation recovery using OMP with a redundant dictionary and a data collection $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2] \in \mathbb{R}^{4 \times 2000}$, where $\mathbf{Y}_{1,2} \in \mathbb{R}^{4 \times 1000}$ are drawn from two independent subspaces of dimensions two each. A redundant dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2] \in \mathbb{R}^{4 \times 8}$ with four atoms per subspace was used to compute the sparse representation of each data point. (a) Recovered support set often contains atoms of the wrong subspace. (b) Cardinality of the support set often exceeds the correct dimensions of two. Owing to the pseudoinverse in the OMP operation, the wrong coefficients are effectively nulled, thus leading to (c) perfectly correct supports and (d) correct cardinalities.

Theorem 2: Let $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K]$ be a collection of $L = L_1 + L_2 + \cdots + L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^K$. Given a dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N \times t_i}$ spans $\mathbf{S}_i$ and $t_i > \dim(\mathbf{S}_i)$, OMP is guaranteed to recover a correct sparse representation matrix $\mathbf{C}$ such that $\mathbf{Y} = \mathbf{DC}$, $\mathbf{C}$ includes *only* atoms from the correct subspace basis for each signal, and minimization of the normalized cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.

The proof is provided in Appendix C.

The next natural steps in studying the theoretical properties and limitations of our proposed scheme are to explore more general cases of disjoint subspaces instead of independent ones, and also explore to the sensitivity to a wrong dictionary. We choose to leave these important questions for future work.

IV. Performance Evaluation

This section evaluates[5] the performance of the proposed approach for synthetic data clustering, face clustering, and temporal video segmentation. In addition, the performances of SSC, LRR, LR-SC, PSSC, and K-subspaces are compared using code packages that were provided by their authors (the parameters of all methods were optimized for the best performance). The objective of this section is to demonstrate that as long as the collection size L is sufficiently large for training the dictionary, then the clustering accuracy of the proposed approach is comparable with state-of-the-art algorithms. The correct number of clusters was supplied to all algorithms in

[3] Note that this statement is far stronger than a successful OMP conditioned on Restricted Isometry Property [26] or mutual coherence [27] since: 1) we address the case of independent subspaces and 2) our goal is segmentation and not signal recovery.

[4] The accumulated support set is the set of atoms selected by OMP.

[5] The results presented in this paper are reproducible using a MATLAB package that is freely available for distribution.

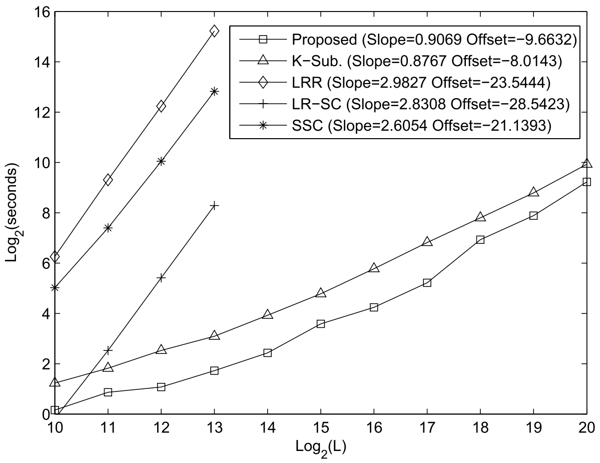


Fig. 4. Computation time versus number of data samples L, for $K = 32$ subspaces, and for data sample dimension $N = 64$ and $M = 64$ learned atoms. The slopes and offsets were estimated using least squares.

every experiment. All experiments were conducted using a computer with Intel i7 Quad Core 2.2 GHz and 8-GB RAM.

A. Computation Time

Computation time comparison of clustering L signals (upto $L = 1\,048\,576$) in $\mathbb{R}^{64}$ is provided in Fig. 4. The reported durations for the proposed approach include a dictionary $D \in \mathbb{R}^{64\times 64}$ learning stage from the L signals if $L < 2^{15}$ or 2^{15} signals if $L \geq 2^{15}$. The results indicate polynomial complexity in L (slope > 2 in logarithmic scale) of state-of-the-art approaches compared with linear complexity (slope ≈ 1 in logarithmic scale) in L of the proposed approach and K-subspaces (PSSC also has linear complexity in L, as discussed in [18]).

B. Synthetic Data Clustering

Clustering accuracy[6] was evaluated for signals contaminated by zero-mean white Gaussian noise, in the SNR range of 5–20 dB. Per each experiment, we generated a set of 800 signals in $\mathbb{R}^{100}$ drawn from a union of eight subspaces of dimensions 10 with equal number of signals per subspace. The bases of all subspaces were chosen as random combinations (nonoverlapping for disjoint subspaces) of the columns of a 100×200 overcomplete discrete cosine transform (ODCT) matrix [23]. The coefficients of each signal were randomly sampled from a Gaussian distribution of zero mean and unit variance. The clustering accuracy results, averaged over 10 noise realizations per SNR point, are presented in Table I. The results of the proposed approach (SBC) are based on a learned dictionary $D \in \mathbb{R}^{100\times 100}$ per every noise realization. The results demonstrate comparable clustering accuracies of the proposed approach and state-of-the-art,[7] and a superior performance compared with K-subspaces. Note that (only) the results of K-subspaces are based on explicit knowledge of the true dimensions ($d = 10$) of all subspaces, as this parameter is required by K-subspaces. For the proposed approach and for PSSC, we employed OMP to approximate the solution of (5) and set the sparse representation target error ε to the noise standard deviation (the target error used for SSC was also close to the noise standard deviation). Fig. 5 compares clustering accuracy of the dictionary-based method PSSC versus the proposed approach in the low SNR range of −2 to +2 dB: it is evident that the proposed approach outperforms PSSC in this range and achieves accuracies above 90% for an SNR of 0 dB and above. The reason for this advantage over PSSC is due to the degraded accuracy in low SNR of the likelihood probabilities estimator employed by PSSC. Fig. 6 shows clustering accuracy versus the number of dictionary atoms per cluster (subspaces dimension = 10): clustering accuracies above 90% are achieved in the range of $M/K = 6$ to $M/K = 16$ for an SNR of 0 dB and above. For an SNR of 6 dB and above, accuracies above 90% are obtained in the range of $M/K = 5$ to $M/K = 25$. For all SNR levels, accuracy deteriorates as M/K drops below five. In the cases of medium to low SNR, it was found that setting the number of atoms to be too high ($M/K > 16$ for 0 dB, $M/K > 20$ for 2 dB, and $M/K > 24$ for 4 dB) degraded[8] the connectivity of the bipartite graph as well as the clustering accuracy. Fig. 7 shows clustering accuracy versus the size L of dictionary training set: close to 100% accuracy is obtained for $L > 800$, namely, at least 10 training samples per atom. Finally, we tested OMP accuracy in terms of selecting correct and wrong atoms: Fig. 8 presents the results for the SNR range of 0–20 dB, using the ground truth dictionary and known support of each subspace. These results indicate that the ratio of correctly selected atoms grows approximately from 35% (0 dB) to 90% (20 dB), and the ratio of mistakenly selected atoms decreases approximately from 15% (0 dB) to 10% (20 dB). Therefore, the bipartite clustering stage achieves high-clustering accuracy even with a partially correct support.

C. Face Clustering

Face clustering is the problem of clustering a collection of facial images according to their human identity. Facial images taken from the same view point and under varying illumination conditions are well approximated as spanned by a subspace of dimension <10 [28], [29], where a unique subspace is associated with each view point and human subject. Subspace clustering was applied successfully to this problem in [2] and [7]. Face clustering accuracy was evaluated using the extended Yale B database [30], which contains 16 128 images of 28 human subjects under nine view points and 64 illumination conditions (per view point). In our experiments, we allocated 10 atoms per human subject (assuming each subspace dimension <10), and to enable efficient dictionary training, we found that a minimum ratio of $L/M > 10$ is required for good clustering results (i.e., at least a hundred

[6] Accuracy was computed by considering all possible permutations and defined by 1 − number of miss-classified signals/total number of signals.

[7] SSC was evaluated using the code that solves (1) with ε = noise standard deviation [5, Sec. 3.3], LRR ($\lambda = 0.15$) was evaluated using the code that solves (3), and LR-SC ($\tau = 0.01$) was evaluated using the code that solves [9, Lemma 1].

[8] This was also verified by increasing L such that $L/M = 10$, as shown in Fig. 7.

TABLE I

CLUSTERING ACCURACY (%) FOR $L = 800$ SIGNALS IN $\mathbb{R}^{100}$ DRAWN FROM EIGHT DISJOINT SUBSPACES WITH DIMENSIONS 10: MEAN, MEDIAN, AND STANDARD DEVIATION WITH RESPECT TO MEAN (σ_{Mean}) AND TO MEDIAN (σ_{Median})

SNR	Parameter	SBC	PSSC	LR-SC	SSC	LRR	K-Sub.
5dB	Mean	**99.90**	97.84	97.64	85.47	89.03	82.96
	Median	**100**	99.75	99.19	82.00	82.13	87.12
	σ_{Mean}	**0.04**	1.80	1.47	2.30	2.82	5.01
	σ_{Median}	**0.05**	1.19	1.55	2.55	3.57	5.18
10dB	Mean	**99.97**	99.77	99.00	85.53	89.29	87.38
	Median	**100**	99.88	99.38	82.13	83.00	92.37
	σ_{Mean}	**0.02**	0.80	0.29	4.26	2.77	5.06
	σ_{Median}	**0.02**	0.90	0.32	4.39	3.41	5.24
15dB	Mean	98.65	**99.87**	99.01	87.42	89.44	97.08
	Median	**100**	99.88	99.19	82.44	83.13	100
	σ_{Mean}	1.25	0.30	**0.25**	2.61	2.73	1.61
	σ_{Median}	1.33	0.30	**0.25**	3.05	3.38	1.86
20dB	Mean	**99.93**	99.79	99.06	89.24	90.90	96.02
	Median	99.94	99.88	99.25	82.50	91.31	**100**
	σ_{Mean}	**0.03**	0.90	0.23	2.78	2.88	1.93
	σ_{Median}	**0.03**	0.90	0.24	3.50	2.88	2.30

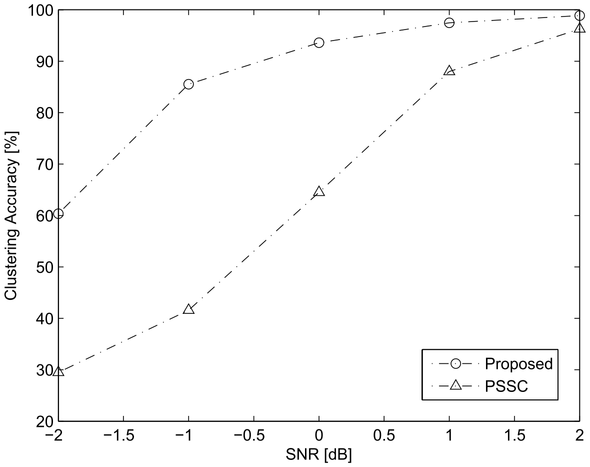


Fig. 5. Clustering accuracy at low SNR of PSSC and the proposed approach (eight disjoint subspaces of dimensions 10 and $L = 800$).

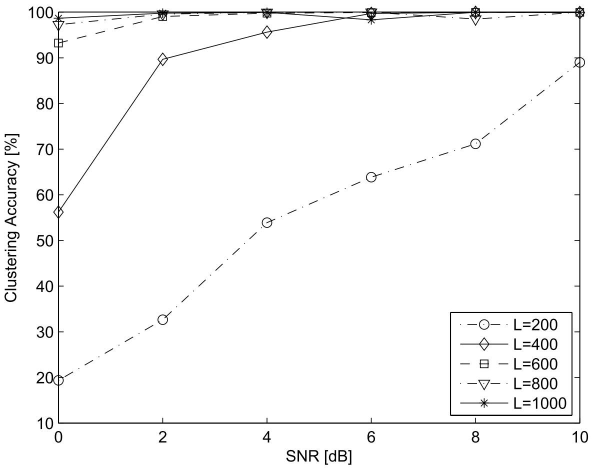


Fig. 7. Clustering accuracy versus the dictionary training set size L (eight disjoint subspaces of dimensions 10 and $M = 100$ atoms).

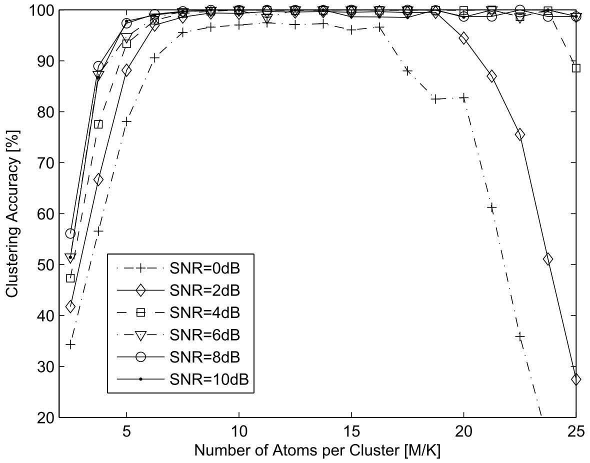


Fig. 6. Clustering accuracy versus the number of dictionary atoms per cluster M/K (eight disjoint subspaces of dimensions 10 and $L = 800$).

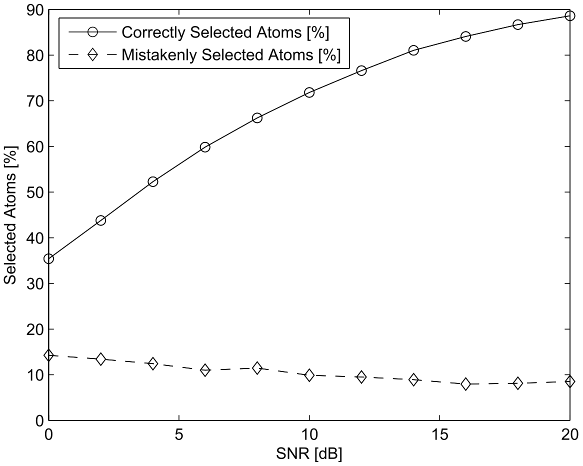


Fig. 8. OMP support estimation accuracy: the ratio (%) of correctly selected atoms and of mistakenly selected atoms.

facial images per subject). Therefore, we generated from the complete collection a subset of 1280 images containing the first 10 human subjects, with 128 images per subject, by merging the fourth and fifth viewpoints which are of similar angles. We further verified that the fourth and fifth view points (of each human subject) can be modeled using a single subspace,

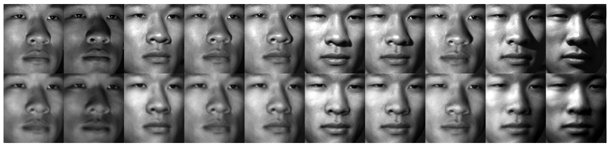


Fig. 9. Reconstruction of facial images from the third merged class of the extended Yale B collection (the five leftmost columns are from the fourth view point and the five rightmost columns are from the fifth view point): the first row displays the original images and the second row displays the reconstructed images from their projections onto the nine leading PCA basis vectors, as obtained from the PCA of the 128 images in the merged class (the union of the fourth and fifth view points).

TABLE II

FACE CLUSTERING ACCURACY (%), AVERAGED OVER 40 DIFFERENT HUMAN SUBJECTS COMBINATIONS PER EACH NUMBER OF CLUSTERS (K)

K =	2	3	4	5	6	7	8
Proposed	92.26	**91.03**	**89.13**	**83.42**	72.15	67.07	64.19
LRR	93.75	85.94	65.47	57.02	51.34	52.86	54.88
LRR-H	91.88	72.36	76.36	74.91	72.49	68.19	66.29
SSC	**95.57**	89.11	85.44	78.98	**73.16**	**72.59**	**73.36**
LR-SC	94.51	80.98	72.86	67.11	59.08	58.82	55.53
PSSC	92.19	82.07	78.26	68.77	61.97	56.96	50.58
K-Subs.	67.60	59.17	50.24	51.34	48.81	48.32	45.35

by reconstructing all 128 images from their projections onto their nine leading PCA basis vectors (obtained by the PCA of each merged class of 128 images). Visual results of this procedure are provided for the third human subject in Fig. 9, demonstrating an excellent quality of the reconstructed images. All images were cropped, resized to 48 × 42 pixels, and column-stacked to vectors in $\mathbb{R}^{2016}$. Clustering accuracy was evaluated for $K = 2, \ldots, 8$ classes by averaging clustering results over 40 different subsets of human subjects, for each value of K, and by choosing 40 different combinations of human subjects out of the 10 classes. Clustering results, provided in Table II, show comparable accuracies of the proposed approach to state-of-the-art[9] and a consistent advantage compared with PSSC [18] and K-subspaces. The parameters of each method were optimized for the best performance and summarized in Table V. Computation times are provided in Table IV, indicating the advantage of the proposed approach over LRR, SSC, and LR-SC. For the proposed approach, we employed OMP to approximate the solution of (6) and set $T_0 = 9$. We have also found that a small accuracy gain of a few percent can be obtained using $A = |C|^p$ $(0 < p < 1)$, which balances edge weights (nonzeros become closer to 1).[10] Clustering accuracy sensitivity of the proposed approach to the number of dictionary atoms was evaluated in the range $M/K = 5$–15, and the results are provided in Table III: the accuracies were high and stable for $M/K = 9, \ldots, 15$ and the choice of $M/K = 5$ provided the lowest accuracies. The highest accuracies for $K = 2, 3, 4$ were obtained using $M/K = 9, 11$; and $M/K = 15$ provided the highest accuracies for $K > 5$.

TABLE III

FACE CLUSTERING ACCURACY (%) VERSUS THE NUMBER OF DICTIONARY ATOMS PER CLUSTER (M/K), AVERAGED OVER 40 DIFFERENT HUMAN SUBJECTS COMBINATIONS PER EACH NUMBER OF CLUSTERS

K =	2	3	4	5	6	7	8
M/K=5	85.85	84.67	82.00	71.08	61.43	53.30	60.23
M/K=7	90.75	88.85	86.85	81.64	67.16	63.71	61.40
M/K=9	92.26	91.03	89.13	83.42	72.15	67.07	64.19
M/K=11	92.60	90.66	88.97	83.79	74.21	67.83	66.15
M/K=13	90.55	90.51	88.21	84.44	74.09	70.48	67.15
M/K=15	89.91	89.34	87.44	83.17	75.15	70.74	67.74

TABLE IV

FACE CLUSTERING COMPUTATION TIME (SECONDS). RESULTS PRESENTED FOR $K = 8$ CLUSTERS ($L = 1024$ FACIAL IMAGES)

Proposed	LRR-H	SSC	LR-SC	PSSC	K-Subs.
7.9	212.4	203.7	50.2	8.1	11.3

TABLE V

FACE CLUSTERING: ALGORITHMS PARAMETERS SETTINGS

K =	2	3	4	5	6	7	8
Proposed, $M =$	20	30	40	50	60	70	80
LRR, λ =	0.25	0.25	0.25	0.3	0.3	0.35	0.35
LR-SC, (τ, γ)=	(5,5)	(6,3)	(6,4)	(6,4)	(6,4)	(6,4)	(6,4)
SSC, (ρ, α)=	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)	(1,10)
K-Subs., $d =$	8	8	8	8	8	8	8

D. Temporal Video Segmentation

Temporal video segmentation is the problem of clustering the frames of a video sequence according to the scene each belongs to (the same scene may repeat several times). By modeling each frame as a point in a high-dimensional linear space, and each scene as spanned by a low-dimensional subspace, temporal video segmentation was successfully solved using subspace clustering in [1]. This work [1] employed GPCA to segment short video sequences of up to 60 frames. In our experiments, we evaluated segmentation accuracy and computational load for two video sequences. The first sequence V_1 contained six scenes and 1190 frames (30 frames/s) of dimensions 360 × 6404 pixels in Red, Green and Blue (RGB) format. The frames of V_1 were converted to gray scale, downsampled to 90 × 160 pixels, and column stacked to vectors in $\mathbb{R}^{14\,400}$. The second sequence V_2 contained three scenes and 12 000 frames (25 frames/s) of dimensions 288 × 512 pixels in RGB format. The frames of V_2 were converted to gray scale, downsampled to 72 × 128 pixels, and column stacked to vectors in $\mathbb{R}^{9216}$. To determine the number of dictionary

[9] State-of-the-art methods were evaluated with sparse outliers support: SSC with the Alternating Direction Method of Multipliers (ADMM)-based version that solves (2), LRR with the version that solves (3), Low Rank Representation version H (LRR-H) that is the same as LRR but with postprocessing of the affinity matrix [8], and LR-SC with the version that solves (4).

[10] Note that the extended Yale B dataset contains many corrupted images, and in our experiments, we found that the dynamic range of the sparse representation coefficients was high. Using $A = |C|^p$ $(0 < p < 1)$, this dynamic range is reduced, thus, balancing edge weights ($p = 0.4$ provided the best results). A similar approach is employed by LRR [8, Sec. 5.4] with $p > 1$ which improves its performance compared with $p = 1$.

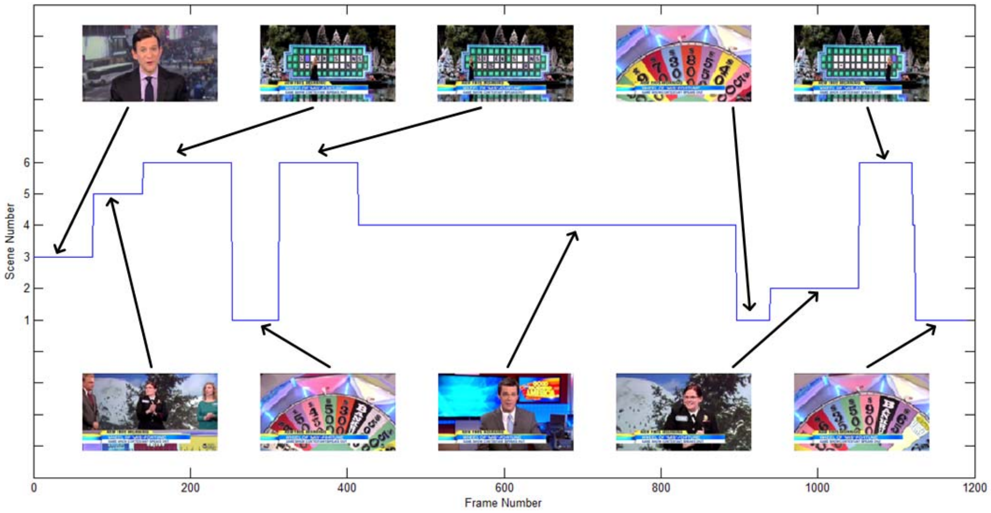


Fig. 10. Temporal video segmentation of V_1 using the proposed approach (98.99% accuracy).

TABLE VI

TEMPORAL VIDEO SEGMENTATION ACCURACY (%): V_1 (1190 FRAMES FROM ABCS TV SHOW WHEEL OF FORTUNE) AND V_2 (12 000 FRAMES FROM ABCS TV SHOW ONE PLUS ONE)

Method	Accuracy (V_1)	Accuracy (V_2)
Proposed	98.82	99.41
PSSC	76.41	88.61
SSC	97.82	N/A
LRR-H	99.16	N/A
LR-SC	98.91	N/A

TABLE VII

TEMPORAL VIDEO SEGMENTATION ACCURACY (%) VERSUS THE NUMBER OF DICTIONARY ATOMS PER CLUSTER (M/K)

Method	Accuracy (V_1)	Accuracy (V_2)
Proposed (M/K=5)	71.28	99.98
Proposed (M/K=7)	99.08	99.18
Proposed (M/K=9)	98.82	99.41
Proposed (M/K=11)	90.01	99.60

TABLE VIII

TEMPORAL VIDEO SEGMENTATION TIME (SECONDS) OF V_1

Proposed	SSC	LRR-H	LR-SC	PSSC
13.5	82.3	2723.5	212.4	13.3

atoms, we computed the PCA basis of several scenes (for each one separately) and found that ~80% of the energy of each scene is represented by its nine leading PCA basis vectors. Therefore, we allocated $9 \times K$ atoms (K is the number of scenes) for the dictionary of each video sequence. The correct segmentation of both sequences was obtained manually, and segmentation accuracy was evaluated using the proposed approach (using $A = |C|$), SSC ($\rho = 1, \alpha = 10$), LRR-H ($\lambda = 0.1$), and LR-SC ($\tau = 0.1$). The parameters of all methods were optimized for the best results,[11] and for SSC, we also projected[12] the column-stacked frames onto their PCA subspace of dimensions nine and segmented the projected frames (excluding this step SSC performance was worse). The results are provided in Table VI, and demonstrate almost perfect segmentation of V_1 (Fig. 10) using all methods, excluding PSSC that segmented the scene between frames 410 and 900 (that contained camera zoom changes) into different scenes. The segmentation of V_2 was possible only with the proposed approach, while LRR, SSC, and LR-RC methods were unable to segment the 12 000 frames due to their complexity. Computation times are provided in Table VIII, indicating the advantage of the proposed approach over LRR, SSC, and LR-SC. The clustering accuracy sensitivity of the proposed approach to the number of dictionary atoms was evaluated in the range $M/K = 5$–11, and the results are provided in Table VII: the accuracies were consistently high for V_2 using $M/K = 5, 7, 9, 11$. For V_1, the accuracies were highest for $M/K = 7, 9, 11$, and decreased for $M/K = 5$.

V. CONCLUSION

Subspace clustering is a powerful tool for processing and analyzing high-dimensional data. This paper presented a low complexity subspace clustering approach that utilizes sparse representations in conjunction with bipartite graph partitioning. By modeling the relations between the signals according to the atoms that represent them, the complexity of the proposed approach depends only linearly in the number of signals. Therefore, it is suitable for clustering very large signal collections. The performance evaluation for synthetic data, face clustering, and temporal video segmentation demonstrate comparable performance[13] with state-of-the-art at

[11] SSC was evaluated with the ADMM-based version without outlier support, LRR-H was evaluated with the version that solves (3) with postprocessing of the affinity matrix, and LR-SC was evaluated with the version that solves [9, Lemma 1].

[12] The ADMM-based SSC code provides the projection option.

[13] The valid intervals of the parameters of the proposed approach are as follows: the dictionary training set size should be an order of magnitude higher than the number of learned atoms (see Fig. 7 for more details). The ratio M/K should be on the same order of the expected subspaces dimensions d_i with a tolerance range of −25% to +50% from the value of d_i.

a significantly lower computational load. We further plan to explore several research directions: 1) extension of the theoretical study to the cases of noiseless data drawn from disjoint subspaces and noisy data drawn from independent subspaces; 2) automatic selection of the number of dictionary atoms; and 3) extension of the proposed approach for data contaminated by outliers and missing entries. Finally, we would like to thank the reviewers of this paper for their constructive comments and guidance.

APPENDIX A
INDEPENDENT AND DISJOINT SUBSPACES

Independent [31] and disjoint subspaces are defined using the sum and the direct sum of a union of subspaces.

Definition 1: The sum of subspaces $\{\mathbf{S}_i\}_{i=1}^K$ is denoted by $\mathbf{V} = \mathbf{S}_1 + \mathbf{S}_2 + \cdots + \mathbf{S}_K$ such that every $\mathbf{v} \in \mathbf{V}$ equals to $\mathbf{v} = \mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_K$ and $\mathbf{s}_i \in \mathbf{S}_i$.

Definition 2: The sum of subspaces $\mathbf{V} = \mathbf{S}_1 + \mathbf{S}_2 + \cdots + \mathbf{S}_K$ is direct if every $\mathbf{v} \in \mathbf{V}$ has a unique representation $\mathbf{v} = \mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_K$, where $\mathbf{s}_i \in \mathbf{S}_i$. The direct sum is denoted by $\mathbf{V} = \mathbf{S}_1 \oplus \mathbf{S}_2 \oplus \ldots \oplus \mathbf{S}_K$.

Given the above definitions, we turn now to define independent and disjoint subspaces.

Definition 3: The subspaces $\{\mathbf{S}_i\}_{i=1}^K$ are independent if their sum is direct. As a consequence, no nonzero vector from any $\mathbf{S}_j$ is a linear combination of vectors from the other subspaces $\mathbf{S}_1, \ldots, \mathbf{S}_{j-1}, \mathbf{S}_{j+1}, \ldots, \mathbf{S}_K$.

Definition 4: The subspaces $\{\mathbf{S}_i\}_{i=1}^K$ are disjoint if $\mathbf{S}_i \cap \mathbf{S}_j = \{0\}$ $\forall i \neq j$. Note that independent subspaces are disjoint; however, disjoint subspaces are not necessarily independent.

APPENDIX B
SPECTRAL BIPARTITE GRAPH CLUSTERING

This appendix provides the derivation of the spectral clustering algorithm for bipartite graphs [24]. Spectral clustering [10] provides an approximate solution to the NP-hard problem of minimizing the normalized cut criterion. This approach requires the solution of the generalized eigenvalue problem $\mathscr{L}\mathbf{z} = \lambda D\mathbf{z}$, where $\mathscr{L} = D - W$ is the Laplacian and D is diagonal such that $D(i,i) = \sum_{k=1}^{M+L} W(i,k)$. In the bipartite case, the affinity matrix is given by

$$\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)}$$

and the Laplacian is given by

$$\mathscr{L} = \begin{bmatrix} \mathbf{D}_1 & -\mathbf{A} \\ -\mathbf{A}^T & \mathbf{D}_2 \end{bmatrix} \in \mathbb{R}^{(M+L)\times(M+L)} \tag{10}$$

where $\mathbf{D}_1 \in \mathbb{R}^{M\times M}$ and $\mathbf{D}_2 \in \mathbb{R}^{L\times L}$ are diagonal such that

$$\mathbf{D}_1(i,i) = \sum_{j=1}^{L} \mathbf{A}(i,j) \quad \text{and} \quad \mathbf{D}_2(j,j) = \sum_{i=1}^{M} \mathbf{A}(i,j). \tag{11}$$

Algorithm 2 Spectral Bipartite Graph Clustering

```
Input: Affinity matrix $\mathbf{W} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}$ and number of clusters $K$. 1)
    1) Compute the SVD of $\overline{\mathbf{A}} = \mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}$.
    2) Construct the matrix $\mathbf{Z} = \begin{bmatrix} \mathbf{D}_1^{-\frac{1}{2}}\mathbf{U} \\ \mathbf{D}_2^{-\frac{1}{2}}\mathbf{V} \end{bmatrix}$, where $\mathbf{U} = [\mathbf{u}_2 ... \mathbf{u}_K]$ and $\mathbf{V} = [\mathbf{v}_2 ... \mathbf{v}_K]$.
    3) Cluster the rows of $\mathbf{Z}$ using the k-means algorithm.
Output: cluster labels for all graph nodes.
```

The generalized eigenvalue problem can be rewritten as

$$\begin{bmatrix} \mathbf{D}_1 & -\mathbf{A} \\ -\mathbf{A}^T & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \tag{12}$$

where $\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}$. Equation (12) can be further expanded as follows:

$$\mathbf{D}_1\mathbf{z}_1 - \mathbf{A}\mathbf{z}_2 = \lambda \mathbf{D}_1\mathbf{z}_1 \tag{13}$$

$$-\mathbf{A}^T\mathbf{z}_1 + \mathbf{D}_2\mathbf{z}_2 = \lambda \mathbf{D}_2\mathbf{z}_2. \tag{14}$$

By setting $\mathbf{u} = \mathbf{D}_1^{1/2}\mathbf{z}_1$ and $\mathbf{v} = \mathbf{D}_2^{1/2}\mathbf{z}_2$, the following are obtained (assuming nonsingularity of $\mathbf{D}_1$ and $\mathbf{D}_2$):

$$\mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_2^{-\frac{1}{2}}\mathbf{v} = (1-\lambda)\mathbf{u} \tag{15}$$

$$\mathbf{D}_2^{-\frac{1}{2}}\mathbf{A}^T\mathbf{D}_1^{-\frac{1}{2}}\mathbf{u} = (1-\lambda)\mathbf{v} \tag{16}$$

which define the SVD equations of $\overline{\mathbf{A}} = \mathbf{D}_1^{-1/2}\mathbf{A}\mathbf{D}_2^{-1/2}$

$$\overline{\mathbf{A}}\mathbf{v}_i = \sigma_i\mathbf{u}_i \quad \text{and} \quad \overline{\mathbf{A}}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i \tag{17}$$

where $\mathbf{v}_i$ is the ith right singular vector, $\mathbf{u}_i$ is the ith left singular vector, and $\sigma_i = 1 - \lambda_i$ is the ith singular value. Therefore, spectral bipartite graph clustering can be obtained from the SVD of $\overline{\mathbf{A}}$, as summarized in Algorithm 2, which has a significant complexity advantage over explicit decomposition of the Laplacian, whenever $M \ll L$, since the complexity of the SVD of $\overline{\mathbf{A}}$ is $O(M^2L)$.

Estimating the number of clusters: a simple estimator can be derived using the spectral gap of the Laplacian; following the discussion in [10, Sec. 8.3], which suggests to estimate K such that all eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_K$ are small and λ_{K+1} is relatively large, the relation $\sigma_i = 1 - \lambda_i$ can be utilized to estimate K such that all singular values $\sigma_1, \sigma_2, \ldots, \sigma_K$ are large and σ_{K+1} is relatively small.

APPENDIX C
PROOF OF THEOREMS

The proof of Theorem 1 is composed of two parts: the first part addresses the correctness and uniqueness of the recovery of $\mathbf{C}$ by OMP (as detailed in Algorithm 3), and the second part addresses the correctness of the subspace clustering result by bipartite graph partitioning. The proof relies on the following lemma.

Lemma 1: Let $\mathbf{D} \in \mathbb{R}^{N\times M}$ contain K minimal bases for K independent subspaces, then the null space $\mathcal{N}(\mathbf{D}) = \{0\}$.

Proof: Let $\{\mathbf{S}_i\}_{i=1}^K$ be a collection of K independent subspaces of dimensions $\{d_i\}_{i=1}^K$, respectively, and let $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N\times d_i}$ is a basis of the ith subspace and $\sum_i d_i = M \le N$. Since the subspaces are independent, their sum is direct, and every vector $\mathbf{v}$ in their direct sum has a unique representation $\mathbf{v} = \sum_{i=1}^{K} \mathbf{D}_i \alpha_i$. Equivalently, the solution to the linear system of equations $\mathbf{D}\alpha = \mathbf{v}$ is unique, which leads to $\mathrm{rank}([\mathbf{D}|\mathbf{v}]) = \mathrm{rank}(\mathbf{D}) = M$. Therefore, $\mathbf{D}$ is full rank and $\mathcal{N}(\mathbf{D}) = \{0\}$. ■

Theorem 1: Let $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K]$ be a collection of $L = L_1 + L_2 + \cdots + L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^{K}$. Given a dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N \times d_i}$ spans $\mathbf{S}_i$ and $d_i = \dim(\mathbf{S}_i)$, OMP is guaranteed to recover the correct and unique sparse representation matrix $\mathbf{C}$ such that $\mathbf{Y} = \mathbf{DC}$, and minimization of the normalized cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.

Proof (Part I): The matrix $\mathbf{C}$ is computed column by column using OMP; therefore, correctness is proved for one column $\mathbf{c}_i = \mathbf{x^k}$ that represents a signal $\mathbf{y}_i \neq 0$ from subspace $\mathbf{S}_i$. OMP terminates either if the residual $\mathbf{r^k} = \mathbf{0}$ or the iteration counter $k = K_{\max} = M$. The proof is provided for each possible termination state of OMP.

1) The residual $\mathbf{r}^k = 0$ and the columns of $\mathbf{D}$ selected by the support set $\mathbf{\Omega}^k$ form exactly $\mathbf{D}_i$ ($\mathbf{S}_i = \mathrm{Span}(\mathbf{D}_{\Omega^k})$): in this case, we have $\mathbf{y}_i = \mathbf{Dx^k} = [\mathbf{D_i}\ \mathbf{D_{i^c}}]\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix}$, where $\mathbf{D}_{i^c}$ is equal to $\mathbf{D}$ excluding the ith basis $\mathbf{D}_i$. On the other hand, $\mathbf{y}_i$ has a unique representation using $\mathbf{D}_i$ that is given by $\mathbf{y}_i = \mathbf{D}_i\mathbf{c}_* = [\mathbf{D}_i\ \mathbf{D}_{i^c}]\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$. Therefore, we can write $\mathbf{D}\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix} = \mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which can be rewritten as
$$\mathbf{D}\left(\begin{bmatrix}\mathbf{x_i}\\0\end{bmatrix} - \begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}\right) = 0.$$
Since $\mathcal{N}(\mathbf{D}) = \{0\}$, the only solution to this equation is $\mathbf{x_i} = \mathbf{c}_*$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

2) The residual $\mathbf{r^k} = \mathbf{0}$ and the columns of $\mathbf{D}$ selected by $\mathbf{\Omega}^k$ include $\mathbf{D}_i$ ($\mathbf{S}_i \subset \mathrm{Span}(\mathbf{D}_{\Omega^k})$): in this case, we have $\mathbf{y}_i = \mathbf{Dx^k} = [\mathbf{D_i}\ \mathbf{D_{i^c}}]\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix}$. Using the unique representation of $\mathbf{y}_i$, we obtain $\mathbf{D}\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix} = \mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which can be rewritten as
$$\mathbf{D}\left(\begin{bmatrix}\mathbf{x_i}\\\mathbf{x_{i^c}}\end{bmatrix} - \begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}\right) = 0.$$
Since $\mathcal{N}(\mathbf{D}) = \{0\}$, the only solution to this equation is $\mathbf{x_i} = \mathbf{c}_*$ and $\mathbf{x_{i^c}} = \mathbf{0}$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

3) OMP reached the maximum number of iterations $K_{\max} = M$ and the residual $\mathbf{r^k} \neq \mathbf{0}$: This scenario is impossible as proved in the following. In this case, $\mathbf{x}^k$ is the solution (stage 4 in Algorithm 3) of the convex least squares problem $\arg\min_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{Dx}\|_2$; therefore, the gradient of the least-squares objective equals zero at the global minimum: $\mathbf{D}^T(\mathbf{y}_i - \mathbf{Dx}^k) = 0$. By replacing $\mathbf{y}_i$ with its unique representation $\mathbf{D}\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, we obtain
$$\mathbf{D}^T\mathbf{D}\left(\begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix} - \mathbf{x}^k\right) = 0.$$
Since $\mathrm{rank}(\mathbf{D}^T\mathbf{D}) = M$, then $\mathcal{N}(\mathbf{D}^T\mathbf{D}) = \{0\}$, and the only solution to this equation is $\mathbf{x^k} = \begin{bmatrix}\mathbf{c}_*\\0\end{bmatrix}$, which results in $\mathbf{r^k} = \mathbf{0}$. Therefore, OMP recovers exactly and uniquely the representation of $\mathbf{y}_i$.

Part II: Given the correct recovery of $\mathbf{C}$, the collection $\mathbf{Y}$ is decomposed as follows[14]:
$$\mathbf{Y} = [\mathbf{Y}_1\ \mathbf{Y}_2] = \mathbf{DC} = [\mathbf{D}_1\ \mathbf{D}_2]\begin{bmatrix}\mathbf{C}_1 & 0\\ 0 & \mathbf{C}_2\end{bmatrix}. \tag{18}$$
By defining $\mathbf{A}_1 = |\mathbf{C}_1| \in \mathbb{R}^{d_1 \times L_1}$ and $\mathbf{A}_2 = |\mathbf{C}_2| \in \mathbb{R}^{d_2 \times L_2}$, the affinity matrix is given by
$$\mathbf{W} = \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & \mathbf{A}_1 & 0\\ & & 0 & \mathbf{A}_2\\ \hline \mathbf{A}_1^T & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}}\\ 0 & \mathbf{A}_2^T & & \end{array}\right].$$
The optimal partition is $\mathcal{V}_1 = \{d_1$ *atoms of* $\mathbf{D}_1 \cup L_1$ *signals spanned by* $\mathbf{D}_1\}$ and $\mathcal{V}_2 = \{d_2$ *atoms of* $\mathbf{D}_2 \cup L_2$ *signals spanned by* $\mathbf{D}_2\}$. W.l.o.g., we rearrange the rows and columns of W such that the vertices associated with $\mathcal{V}_1$ are the leading vertices and the vertices associated with $\mathcal{V}_2$ are the tailing vertices. The rearranged affinity is given by
$$\overline{\mathbf{W}} = \left[\begin{array}{cc|cc} 0 & \mathbf{A}_1 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}}\\ \mathbf{A}_1^T & 0 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & 0 & \mathbf{A}_2\\ & & \mathbf{A}_2^T & 0 \end{array}\right].$$
The cut of the optimal partition is given by
$$\mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} \overline{\mathbf{W}}_{ij} = 0 \tag{19}$$
and the weight of each group is given by
$$\mathrm{weight}(\mathcal{V}_{1,2}) = \sum_{i \in \mathcal{V}_{1,2}} \sum_{k} \overline{W}_{ik} = 2S(\mathbf{A}_{1,2}) > 0 \tag{20}$$
where $S(\mathbf{Q}) = \sum_{n,m} \mathbf{Q}_{nm}$ is the sum of matrix entries. Therefore, the normalized cut metric equals zero for the optimal partition. ■

Theorem 2: Let $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K]$ be a collection of $L = L_1 + L_2 + \cdots + L_K$ signals from K independent subspaces of dimensions $\{d_i\}_{i=1}^{K}$. Given a dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ such that $\mathbf{D}_i \in \mathbb{R}^{N \times t_i}$ spans $\mathbf{S}_i$ and $t_i > \dim(\mathbf{S}_i)$, OMP is guaranteed to recover a correct sparse

[14] This part of the theorem is proved for the case of two subspaces, to focus on the essence of the method and avoid cumbersome notations. The extension to more subspaces is as follows: in the case of $K > 2$ clusters, the normalized cut criterion is extended to be the sum of K components (each one is the cut of the kth cluster with the complement of this cluster, divided by the weight of the kth cluster). The extension of the proof to $K > 2$ is enabled by: 1) proving that W can be transformed into a block diagonal matrix (with $K > 2$ blocks) and 2) proving that the cut of each block with the block formed by the union of the other blocks is null.

Algorithm 3 Orthogonal Matching Pursuit (OMP)

```
Input: $\mathbf{y}$, $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_M}] \in \mathbb{R}^{\mathbf{N \times M}}$.
Initialize:
    1) Iteration counter $k = 0$.
    2) Maximum number of iterations $K_{\max} = M$.
    3) Support set $\Omega^0 = \emptyset$.
    4) Residual $\mathbf{r^0} = \mathbf{y}$.
Repeat until $\mathbf{r^k} = \mathbf{0}$ or $\mathbf{k} = \mathbf{K_{max}}$
    1) Increment iteration counter $k = k + 1$.
    2) Select atom: find $j = \arg\max_j | < \mathbf{r^{k-1}}, \mathbf{d_j} > |$.
    3) $\Omega^k = \Omega^{k-1} \cup \ j$.
    4) solution $\mathbf{x}^k = \arg\min_{\mathbf{u}} \|\mathbf{y} - \mathbf{Du}\|_2$ s.t. Support$\{\mathbf{u}\} = \Omega^k$.
    5) $\mathbf{r}^k = \mathbf{y} - \mathbf{Dx^k}$
Output: $\mathbf{x^k}$.
```

representations matrix $\mathbf{C}$ such that $\mathbf{Y} = \mathbf{DC}$, $\mathbf{C}$ include *only* atoms from the correct subspace basis for each signal, and minimization of the normalized cut criterion for partitioning the bipartite graph defined by (8) will yield correct subspace clustering.

Proof: The matrix $\mathbf{C}$ is computed column by column using OMP; therefore, correctness is proved for one column $\mathbf{c}_i = \mathbf{x^k}$ that represents a signal $\mathbf{y}_i \neq 0$ from subspace $\mathbf{S}_i$. OMP terminates either if the residual $\mathbf{r^k} = \mathbf{0}$ or the iteration counter $k = K_{\max} = M$. The proof is provided for each possible termination state of OMP.

1) $\mathbf{r^k} = \mathbf{0}$ and $\mathbf{S}_i = \text{Span}(\mathbf{D}_{\Omega^k})$: in this case, we have $\mathbf{y}_i = \mathbf{Dx^k} = [\mathbf{D_i} \ \mathbf{D_{i^c}}]\begin{bmatrix} \mathbf{x}_i \\ 0 \end{bmatrix} = \mathbf{D_i x_i}$, and $\mathbf{x}_i \neq 0$. Therefore, $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.
2) $\mathbf{r^k} = \mathbf{0}$ and $\mathbf{S}_i \subset \text{Span}(\mathbf{D}_{\Omega^k})$: in this case, we have $\mathbf{y}_i = \mathbf{Dx^k} = [\mathbf{D_i} \ \mathbf{D_{i^c}}]\begin{bmatrix} \mathbf{x_i} \\ \mathbf{x_{i^c}} \end{bmatrix}$. On the other hand, $\mathbf{x^k}$ is the solution to the least squares problem 4) of Algorithm 3, which is computed using the pseudoinverse $\mathbf{x}^k = \mathbf{D}^{\dagger}_{\Omega^k}\mathbf{y}_i$. Therefore, this solution is guaranteed to have the smallest l_2-norm among all feasible solutions to the equation $\mathbf{y}_i = \mathbf{Du}$ (such that support$(\mathbf{u}) = \Omega^{\mathbf{k}}$). Since $\mathbf{y}_i \in \mathbf{S}_i$, it can be represented by $\mathbf{y}_i = \mathbf{D}_i\mathbf{c}_* = [\mathbf{D}_i \ \mathbf{D}_{i^c}]\begin{bmatrix} \mathbf{c}_* \\ 0 \end{bmatrix}$, which leads to $\mathbf{D}_i\mathbf{c}_* = \mathbf{D}_i\mathbf{x_i} + \mathbf{D_{i^c}x_{i^c}}$. Note that this equation can be rewritten as[15] $\mathbf{D}_i(\mathbf{c}_* - \mathbf{x_i}) = \mathbf{D_{i^c}x_{i^c}}$, in which the left-hand side is a vector in $\mathbf{S}_i$ and the right-hand side is a vector in $\oplus_{j=1,j\neq i}^{K}\mathbf{S}_j$. The subspaces $\mathbf{S}_i$ and $\oplus_{j=1,j\neq i}^{K}\mathbf{S}_j$ are independent; therefore, their intersection contains only the null vector. The implications of this result are that $\mathbf{D}_{i^c}\mathbf{x_{i^c}} = \mathbf{0}$ and that $\mathbf{x_i}$ is a feasible solution (namely, $\mathbf{y}_i = \mathbf{D}_i\mathbf{x_i}$). Since the pseudoinverse-based solution provides the solution with the smallest l_2-norm, we obtain that
$$\left\|\begin{bmatrix} \mathbf{x_i} \\ 0 \end{bmatrix}\right\|_2 < \left\|\begin{bmatrix} \mathbf{x_i} \\ \mathbf{x_{i^c}} \end{bmatrix}\right\|_2 \quad \forall \ \mathbf{x_{i^c}} \neq \mathbf{0}).$$
Therefore, this solution must lead to $\mathbf{x_{i^c}} = \mathbf{0}$ and thus, $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.
3) OMP reached the maximum number of iterations $K_{\max} = M$: in this case, there is an infinite number of solutions to the equation $\mathbf{y}_i = \mathbf{D}_{\Omega^M}\mathbf{x}^k = \mathbf{Dx}^k = \mathbf{D}_i\mathbf{x_i} + \mathbf{D_{i^c}x_{i^c}}$, such that $\mathbf{D}_{i^c}\mathbf{x_{i^c}} = \mathbf{0}$. Therefore, the minimizer of the convex least squares problem $\arg\min_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{Dx}\|_2$ must reach its global minimum, which is $\mathbf{r^k} = \mathbf{0}$, and following the case 2) above, $\mathbf{y}_i$ is correctly and exclusively represented by atoms that span $\mathbf{S}_i$.

The second part of the theorem follows exactly from part II of Theorem I. Note that the proof of the first part of Theorem II can be also applied to the first part of Theorem I; however, we chose to present the two different approaches: the proof of Theorem I uses null-space properties of the dictionary and covers only the minimal dictionary case, whereas the proof of Theorem II uses pseudoinverse properties and covers the over-complete dictionary case. ■

REFERENCES

[1] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[2] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Madison, WI, USA, Jun. 2003, pp. 11–18.

[3] Y. M. Lu and M. N. Do, "A theory for sampling signals from a union of subspaces," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2334–2345, Jun. 2008.

[4] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[5] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 2790–2797.

[6] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[7] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 663–670.

[8] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[9] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.

[10] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[11] S. R. Rao, R. Tron, Y. Ma, and R. Vidal, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Graz, Austria, May 2006, pp. 94–106.

[14] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[15] X. Zhang, F. Sun, G. Liu, and Y. Ma, "Fast low-rank subspace segmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1293–1297, May 2013.

[15] The following argument relies on [6, Th. 1].

[16] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 430–437.

[17] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 2042–2045.

[18] A. Adler, M. Elad, and Y. Hel-Or, "Probabilistic subspace clustering via sparse representations," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 63–66, Jan. 2013.

[19] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[20] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proc. IEEE*, vol. 98, no. 6, pp. 995–1005, Jun. 2010.

[21] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 1993, pp. 40–44.

[22] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

[23] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[24] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2001, pp. 269–274.

[25] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 1997.

[26] M. A. Davenport and M. B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4395–4401, Sep. 2010.

[27] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[28] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Maui, HI, USA, Jun. 1991, pp. 586–591.

[29] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, Feb. 2003.

[30] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[31] K. M. Hoffman and R. Kunze, *Linear Algebra*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971.

Amir Adler received the B.Sc. (*cum laude*) degree in electrical engineering, the M.Eng. (*cum laude*) degree in electrical engineering, and the Ph.D. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, 1994, 2001, and 2014, respectively.

He was the Director of Signal Processing with GO Networks Inc., Mountain View, CA, USA, from 2003 to 2006, and the Chief Technology Officer with the Wi-Fi Division, NextWave Wireless Inc., San Diego, CA, USA, from 2006 to 2008. His current research interests include sparse and redundant representations with applications for signal and image processing.

Dr. Adler was the recipient of the 2011 Google Europe Doctoral Fellowship in Multimedia.

Michael Elad (F'12) received the B.Sc., M.Sc., and D.Sc. degrees from the Department of Electrical Engineering, Technion—Israel Institute of Technology (Technion), Haifa, Israel, in 1986, 1988, and 1997, respectively.

He has been a faculty member with the Department of Computer Science, Technion, since 2003, where he has also been a Full Professor since 2010. He is involved in signal and image processing, specializing, in particular, on inverse problems, sparse representations, and superresolution.

Prof. Elad received the Technion's Best Lecturer Award six times. He was a recipient of the 2007 Solomon Simon Mani Award for excellence in teaching, the 2008 Henri Taub Prize for academic excellence, and the 2010 Hershel-Rich Prize for innovation. He serves as an Associate Editor of the SIAM *Journal on Imaging Sciences* and *Applied and Computational Harmonic Analysis*. He also serves as a Senior Editor of the IEEE SIGNAL PROCESSING LETTERS.

Yacov Hel-Or received the Ph.D. degree in computer science from the Hebrew University of Jerusalem, Jerusalem, Israel.

He held post-doctoral positions with the Weizmann Institute of Science, Rehovot, Israel, and the NASA Ames Research Center, Mountain View, CA, USA. He served as a Researcher with the Hewlett-Packard Israel Science Center, Technion—Israel Institute of Technology, Haifa, Israel. He has been a faculty member with the School of Computer Science, Interdisciplinary Center Herzlia, Herzlia, Israel, since 1998. His current research interests include computer vision, image processing, computer graphics, and robotics.