

Robust Recovery of Subspace Structures by Low-Rank Representation

Guangcan Liu, *Member, IEEE*, Zhouchen Lin, *Senior Member, IEEE*,
Shuicheng Yan, *Senior Member, IEEE*, Ju Sun, *Student Member, IEEE*,
Yong Yu, and Yi Ma, *Senior Member, IEEE*

Abstract—In this paper, we address the subspace clustering problem. Given a set of data samples (vectors) approximately drawn from a union of multiple subspaces, our goal is to cluster the samples into their respective subspaces and remove possible outliers as well. To this end, we propose a novel objective function named Low-Rank Representation (LRR), which seeks the lowest rank representation among all the candidates that can represent the data samples as linear combinations of the bases in a given dictionary. It is shown that the convex program associated with LRR solves the subspace clustering problem in the following sense: When the data is clean, we prove that LRR exactly recovers the true subspace structures; when the data are contaminated by outliers, we prove that under certain conditions LRR can exactly recover the row space of the original data and detect the outlier as well; for data corrupted by arbitrary sparse errors, LRR can also approximately recover the row space with theoretical guarantees. Since the subspace membership is provably determined by the row space, these further imply that LRR can perform robust subspace clustering and error correction in an efficient and effective way.

Index Terms—Low-rank representation, subspace clustering, segmentation, outlier detection

1 INTRODUCTION

IN pattern analysis and signal processing, an underlying tenet is that the data often contains some type of *structure* that enables intelligent representation and processing. So one usually needs a parametric model to characterize a given set of data. To this end, the well-known (linear) *subspaces* are possibly the most common choice, mainly because they are easy to compute and often effective in real applications. Several types of visual data, such as motion [1],

[2], [3], face [4], and texture [5], have been known to be well characterized by subspaces. Moreover, by applying the concept of reproducing kernel Hilbert space, one can easily extend the linear models to handle nonlinear data. So the subspace methods have been gaining much attention in recent years. For example, the widely used Principal Component Analysis (PCA) method and the recently established matrix completion [6] and recovery [7] methods are essentially based on the hypothesis that the data is approximately drawn from a low-rank subspace. However, a given dataset can seldom be well described by a *single* subspace. A more reasonable model is to consider data as lying near *several* subspaces, namely, the data is considered as samples approximately drawn from a mixture of several low-rank subspaces, as shown in Fig. 1.

The generality and importance of subspaces naturally lead to a challenging problem of subspace segmentation (or clustering), whose goal is to segment (cluster or group) data into clusters with each cluster corresponding to a subspace. Subspace segmentation is an important data clustering problem and arises in numerous research areas, including computer vision [3], [8], [9], image processing [5], [10], and system identification [11]. When the data is clean, i.e., the samples are strictly drawn from the subspaces, several existing methods (e.g., [12], [13], [14]) are able to exactly solve the subspace segmentation problem. So, as pointed out by Rao et al. [3] and Liu et al. [14], the main challenge of subspace segmentation is to handle the *errors* (e.g., noise and corruptions) that possibly exist in data, i.e., to handle the data that may not strictly follow subspace structures. With this viewpoint, in this paper we therefore study the following *robust subspace clustering* [15] problem.

Problem 1.1 (Robust Subspace Clustering). *Given a set of data samples approximately (i.e., the data may contain errors)*

- G. Liu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, the Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana-Champaign, Urbana, IL 61801, and the Department of Electrical and Computer Engineering, National University of Singapore.
E-mail: gutty.liu@gmail.com.
- Z. Lin is with the Key Laboratory of Machine Perception (MOE), School of Electronic Engineering and Computer Science, Peking University, No. 5 Yiheyuan Road, Haidian District, Beijing 100871, China.
E-mail: zlin@pku.edu.cn.
- S. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Block E4, #08-27, Engineering Drive 3, Singapore 117576. E-mail: eleyans@nus.edu.sg.
- J. Sun is with the Department of Electrical Engineering, Columbia University, 1300 S.W. Mudd, 500 West 120th Street, New York, NY 10027. E-mail: jusun@ee.columbia.edu.
- Y. Yu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, No. 800 Dongchuan Road, Minhang District, Shanghai 200240, China. E-mail: yyyu@apex.sjtu.edu.cn.
- Y. Ma is with the Visual Computing Group, Microsoft Research Asia, China, and with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Room 145, 1308 West Main Street, Urbana, IL 61801.

Manuscript received 14 Oct. 2010; revised 8 Sept. 2011; accepted 24 Mar. 2012; published online 4 Apr. 2012.

Recommended for acceptance by T. Jebara.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-10-0786.

Digital Object Identifier no. 10.1109/TPAMI.2012.88.

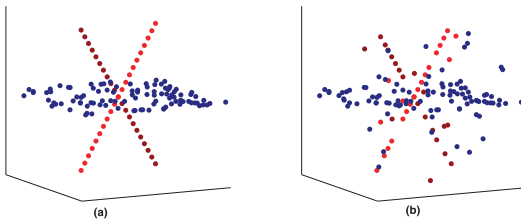


Fig. 1. A mixture of subspaces consisting of a 2D plane and two 1D lines. (a) The samples are strictly drawn from the underlying subspaces. (b) The samples are approximately drawn from the underlying subspaces.

drawn from a union of linear subspaces, correct the possible errors and segment all samples into their respective subspaces simultaneously.

Notice that the word “error” generally refers to the *deviation* between model assumption (i.e., subspaces) and data. It could exhibit as noise [6], missed entries [6], outliers [16], and corruptions [7] in reality. Fig. 2 illustrates three typical types of errors under the context of subspace modeling. In this paper, we shall focus on the sample-specific corruptions (and outliers) shown in Fig. 2c, with mild concerns to the cases of Figs. 2a and 2b. Notice that an outlier is from a different model other than subspaces and is essentially different from a corrupted sample that belongs to the subspaces. We put them into the same category just because they can be handled in the same way, as will be shown in Section 5.2.

To recover the subspace structures from the data containing errors, we propose a novel method termed *Low-Rank Representation* (LRR) [14]. Given a set of data samples, each of which can be represented as a linear combination of the bases in a dictionary, LRR aims at finding the *lowest rank* representation of all data jointly. The computational procedure of LRR is to solve a *nuclear norm* [17] regularized optimization problem, which is convex and can be solved in polynomial time. By choosing a specific dictionary, it is shown that LRR can well solve the subspace clustering problem: When the data is clean, we prove that LRR exactly recovers the *row space* of the data; for the data contaminated by outliers, we prove that under certain conditions LRR can *exactly* recover the row space of the original data and detect the outlier as well; for the data corrupted by arbitrary errors, LRR can also approximately recover the row space with theoretical guarantees. Since the subspace membership is provably determined by the row space (we will discuss this in Section 3.2), these further imply that LRR can perform robust subspace clustering and error correction in an efficient way. In summary, the contributions of this work include:

- We develop a simple yet effective method, termed LRR, which has been used to achieve state-of-the-art performance in several applications such as motion segmentation [4], image segmentation [18], saliency detection [19], and face recognition [4].
- Our work extends the recovery of corrupted data from a single subspace [7] to multiple subspaces. Compared to [20], which requires the bases of subspaces to be known for handling the corrupted

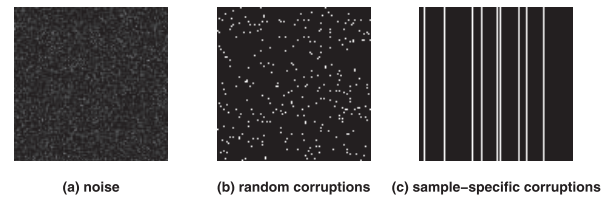


Fig. 2. Illustrating three typical types of errors: (a) noise [6], which indicates the phenomena that the data is slightly perturbed around the subspaces (what we show is a perturbed data matrix whose columns are samples drawn from the subspaces), (b) random corruptions [7], which indicate that a fraction of random entries are grossly corrupted, (c) sample-specific corruptions (and outliers), which indicate the phenomena that a fraction of the data samples (i.e., columns of the data matrix) are far away from the subspaces.

data from multiple subspaces, our method is *autonomous*, i.e., no extra clean data is required.

- Theoretical results for robust recovery are provided. While our analysis shares similar features as previous work in matrix completion [6] and Robust PCA (RPCA) [7], [16], it is considerably more challenging due to the fact that there is a dictionary matrix in LRR.

2 RELATED WORK

In this section, we discuss some existing subspace segmentation methods. In general, existing works can be roughly divided into four main categories: mixture of Gaussian, factorization, algebraic, and spectral-type methods.

In statistical learning, mixed data is typically modeled as a set of independent samples drawn from a mixture of probabilistic distributions. As a single subspace can be well modeled by a (degenerate) Gaussian distribution, it is straightforward to assume that each probabilistic distribution is Gaussian, i.e., adopting a mixture of Gaussian models. Then the problem of segmenting the data is converted to a model estimation problem. The estimation can be performed either by using the Expectation Maximization (EM) algorithm to find a maximum likelihood estimate, as done in [21], or by iteratively finding a min-max estimate, as adopted by K-subspaces [8] and Random Sample Consensus (RANSAC) [10]. These methods are sensitive to errors. So several efforts have been made for improving their robustness, e.g., the Median K-flats [22] for K-subspaces, the work [23] for RANSAC, and [5] use a coding length to characterize a mixture of Gaussian. These refinements may introduce some robustness. Nevertheless, the problem is still not well solved due to the optimization difficulty, which is a bottleneck for these methods.

Factorization-based methods [12] seek to approximate the given data matrix as a product of two matrices such that the support pattern for one of the factors reveals the segmentation of the samples. In order to achieve robustness to noise, these methods modify the formulations by adding extra regularization terms. Nevertheless, such modifications usually lead to non-convex optimization problems which need heuristic algorithms (often based on alternating minimization or EM-style algorithms) to solve. Getting stuck at local minima may undermine their performances, especially when the data is grossly corrupted. It will be shown that LRR can be regarded as a robust generalization of the method in [12] (which is referred to as PCA in this

paper). The formulation of LRR is convex and can be solved in polynomial time.

Generalized Principal Component Analysis (GPCA) [24] presents an algebraic way to model the data drawn from a union of multiple subspaces. This method describes a subspace containing a data point by using the gradient of a polynomial at that point. Then subspace segmentation is made equivalent to fitting the data with polynomials. GPCA can guarantee the success of the segmentation under certain conditions, and it does not impose any restriction on the subspaces. However, this method is sensitive to noise due to the difficulty of estimating the polynomials from real data, which also causes the high computation cost of GPCA. Recently, Robust Algebraic Segmentation (RAS) [25] has been proposed to resolve the robustness issue of GPCA. However, the computation difficulty for fitting polynomials is unfathomably large. So RAS can make sense only when the data dimension is low and the number of subspaces is small.

As a data clustering problem, subspace segmentation can be done by first learning an affinity matrix from the given data and then obtaining the final segmentation results by Spectral Clustering (SC) algorithms such as Normalized Cuts (NCut) [26]. Many existing methods, such as Sparse Subspace Clustering (SSC) [13], Spectral Curvature Clustering (SCC) [27], [28], Spectral Local Best-fit Flats (SLBF) [29], [30], the proposed LRR method, and [2], [31], possess such a spectral nature, so-called spectral-type methods. The main difference among various spectral-type methods is the approach for learning the affinity matrix. Under the assumption that the data is clean and the subspaces are independent, Elhamifar and Vidal [13] show that a solution produced by Sparse Representation (SR) [32] could achieve the so-called ℓ_1 Subspace Detection Property (ℓ_1 -SDP): The within-class affinities are sparse and the between-class affinities are all zeros. In the presence of outliers, it is shown in [15] that the SR method can still obey ℓ_1 -SDP. However, ℓ_1 -SDP may not be sufficient to ensure the success of subspace segmentation [33]. Recently, Lerman and Zhang [34] proved that under certain conditions the multiple subspace structures can be exactly recovered via ℓ_p ($p \leq 1$) minimization. Unfortunately, since the formulation is not convex, it is still unknown how to efficiently obtain the globally optimal solution. In contrast, the formulation of LRR is convex and the corresponding optimization problem can be solved in polynomial time. What is more, even if the data is contaminated by outliers, the proposed LRR method is proven to exactly recover the right row space, which provably determines the subspace segmentation results (we shall discuss this in Section 3.2). In the presence of arbitrary errors (e.g., corruptions, outliers, and noise), LRR is also guaranteed to produce near recovery.

3 PRELIMINARIES AND PROBLEM STATEMENT

3.1 Summary of Main Notations

In this paper, matrices are represented with capital symbols. In particular, I is used to denote the identity matrix, and the entries of matrices are denoted by using $[\cdot]$ with subscripts. For instance, M is a matrix, $[M]_{ij}$ is its (i, j) th entry, $[M]_{i,:}$ is its i th row, and $[M]_{:,j}$ is its j th column. For ease of presentation, the horizontal (respectively, vertical) concatenation of a collection of matrices along row (respectively, column) is denoted by $[M_1, M_2, \dots, M_k]$ (respectively,

$[M_1; M_2; \dots; M_k]$). The block-diagonal matrix formed by a collection of matrices M_1, M_2, \dots, M_k is denoted by

$$\text{diag}(M_1, M_2, \dots, M_k) = \begin{bmatrix} M_1 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & M_k \end{bmatrix}. \quad (1)$$

The only used vector norm is the ℓ_2 norm, denoted by $\|\cdot\|_2$. A variety of norms on matrices will be used. The matrix $\ell_0, \ell_{2,0}, \ell_1, \ell_{2,1}$ norms are defined by $\|M\|_0 = \#\{(i, j) : [M]_{ij} \neq 0\}$, $\|M\|_{2,0} = \#\{i : \|[M]_{:,i}\|_2 \neq 0\}$, $\|M\|_1 = \sum_{i,j} |[M]_{ij}|$, and $\|M\|_{2,1} = \sum_i \|[M]_{:,i}\|_2$, respectively. The matrix ℓ_∞ norm is defined as $\|M\|_\infty = \max_{i,j} |[M]_{ij}|$. The spectral norm of a matrix M is denoted by $\|M\|$, i.e., $\|M\|$ is the largest singular value of M . The Frobenius norm and the nuclear norm (the sum of singular values of a matrix) are denoted by $\|M\|_F$ and $\|M\|_*$, respectively. The euclidean inner product between two matrices is $\langle M, N \rangle = \text{tr}(M^T N)$, where M^T is the transpose of a matrix and $\text{tr}(\cdot)$ is the trace of a matrix.

The supports of a matrix M are the indices of its nonzero entries, i.e., $\{(i, j) : [M]_{ij} \neq 0\}$. Similarly, its column supports are the indices of its nonzero columns. The symbol \mathcal{I} (superscripts, subscripts, etc.) is used to denote the column supports of a matrix, i.e., $\mathcal{I} = \{i : \|[M]_{:,i}\|_2 \neq 0\}$. The corresponding complement set (i.e., zero columns) is \mathcal{I}^c . There are two projection operators associated with \mathcal{I} and \mathcal{I}^c : $\mathcal{P}_{\mathcal{I}}$ and $\mathcal{P}_{\mathcal{I}^c}$. While applying them to a matrix M , the matrix $\mathcal{P}_{\mathcal{I}}(M)$ (respectively, $\mathcal{P}_{\mathcal{I}^c}(M)$) is obtained from M by setting $[M]_{:,i}$ to zero for all $i \notin \mathcal{I}$ (respectively, $i \notin \mathcal{I}^c$).

We also adopt the conventions of using $\text{span}(M)$ to denote the linear space spanned by the columns of a matrix M , using $y \in \text{span}(M)$ to denote that a vector y belongs to the space $\text{span}(M)$, and using $Y \in \text{span}(M)$ to denote that all column vectors of Y belong to $\text{span}(M)$.

Finally, in this paper we use several terminologies, including “block-diagonal matrix,” “union and sum of subspaces,” “independent (and disjoint) subspaces,” “full SVD and skinny SVD,” “pseudo-inverse,” “column space and row space,” and “affinity degree.” These terminologies are defined in the Appendix, which can be found in the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.88>.

3.2 Relations between Segmentation and Row Space

Let X_0 with skinny SVD $U_0 \Sigma_0 V_0^T$ be a collection of data samples *strictly* drawn from a union of multiple subspaces (i.e., X_0 is clean); the subspace membership of the samples is determined by the row space of X_0 . Indeed, as shown in [12], when subspaces are independent, $V_0 V_0^T$ forms a block-diagonal matrix: The (i, j) th entry of $V_0 V_0^T$ can be nonzero only if the i th and j th samples are from the same subspace. Hence, this matrix, termed Shape Interaction Matrix (SIM) [12], has been widely used for subspace segmentation. Previous approaches simply compute the SVD of the data matrix $X = U_X \Sigma_X V_X^T$ and then use $|V_X V_X^T|$ ¹ for subspace segmentation. However, in the presence of outliers and corruptions, V_X can be far away from V_0 and thus the

1. For a matrix M , $|M|$ denotes the matrix with the (i, j) th entry being the absolute value of $[M]_{ij}$.

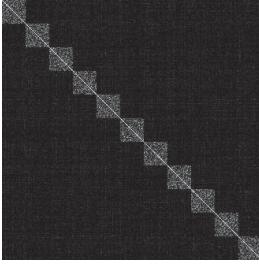


Fig. 3. An example of the matrix $V_0 V_0^T$ computed from dependent subspaces. In this example, we create 11 pairwise disjoint subspaces, each of which is of dimension 20 and draw 20 samples from each subspace without errors. The ambient dimension is 200, which is smaller than the sum of the dimensions of the subspaces. So the subspaces are dependent and $V_0 V_0^T$ is not strictly block-diagonal. Nevertheless, it is simple to see that high segmentation accuracy can be achieved by using the above affinity matrix to do spectral clustering.

segmentation using such approaches is inaccurate. In contrast, we show that LRR can recover $V_0 V_0^T$ even when the data matrix X is contaminated by outliers.

If the subspaces are not independent, $V_0 V_0^T$ may not be strictly block-diagonal. This is indeed well expected since when the subspaces have nonzero (nonempty) intersections, then some samples may belong to multiple subspaces simultaneously. When the subspaces are pairwise disjoint (but not independent), our extensive numerical experiments show that $V_0 V_0^T$ may still be close to be block-diagonal, as exemplified in Fig. 3. Hence, to recover $V_0 V_0^T$ is still of interest to subspace segmentation.

3.3 Problem Statement

Problem 1.1 only roughly describes what we want to study. More precisely, this paper addresses the following problem.

Problem 3.1 (Subspace Recovery). Let $X_0 \in \mathbb{R}^{d \times n}$ with skinny SVD $U_0 \Sigma_0 V_0^T$ store a set of n d -dimensional samples (vectors) strictly drawn from a union of k subspaces $\{\mathcal{S}_i\}_{i=1}^k$ of unknown dimensions (k is unknown either). Given a set of observation vectors X generated by

$$X = X_0 + E_0,$$

the goal is to recover the row space of X_0 or to recover the true SIM $V_0 V_0^T$ as equal.

The recovery of row space can guarantee high segmentation accuracy, as analyzed in Section 3.2. Also, the recovery of row space naturally implies success in error correction. So it is sufficient to set the goal of subspace clustering as the recovery of the row space identified by $V_0 V_0^T$. For ease of exploration, we consider the problem under three assumptions of increasing practicality and difficulty.

Assumption 1. The data is clean, i.e., $E_0 = 0$.

Assumption 2. A fraction of the data samples are grossly corrupted and the others are clean, i.e., E_0 has sparse column supports as shown in Fig. 2c.

Assumption 3. A fraction of the data samples are grossly corrupted and the others are contaminated by small Gaussian noise, i.e., E_0 is characterized by a combination of the models shown in Figs. 2a and 2c.

Unlike [14], the independent assumption on the subspaces is not highlighted in this paper because the analysis in this work focuses on recovering $V_0 V_0^T$ rather than a pursuit of block-diagonal matrix.

4 LOW-RANK REPRESENTATION FOR MATRIX RECOVERY

In this section, we abstractly present the LRR method for recovering a matrix from corrupted observations. The basic theorems and optimization algorithms will be presented. The specific methods and theories for handling the subspace clustering problem are deferred until Section 5.

4.1 Low-Rank Representation

In order to recover the low-rank matrix X_0 from the given observation matrix X corrupted by errors E_0 ($X = X_0 + E_0$), it is straightforward to consider the following regularized rank minimization problem:

$$\min_{D,E} \text{rank}(D) + \lambda \|E\|_\ell, \quad \text{s.t.} \quad X = D + E, \quad (2)$$

where $\lambda > 0$ is a parameter and $\|\cdot\|_\ell$ indicates certain regularization strategy, such as the squared Frobenius norm (i.e., $\|\cdot\|_F^2$) used for modeling the noise as show in Fig. 2a [6], the ℓ_0 norm adopted by Candès et al. [7] for characterizing the random corruptions as shown in Fig. 2b, and the $\ell_{2,0}$ norm adopted by Liu et al. [14] and Xu et al. [16] for dealing with sample-specific corruptions and outliers. Suppose D^* is a minimizer with respect to the variable D , then it gives a low-rank recovery to the original data X_0 .

The above formulation is adopted by the recently established the Robust PCA method [7], which has been used to achieve the state-of-the-art performance in several applications (e.g., [35]). However, this formulation implicitly assumes that the underlying data structure is a single low-rank subspace. When the data is drawn from a union of multiple subspaces, denoted as $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$, it actually treats the data as being sampled from a single subspace defined by $\mathcal{S} = \sum_{i=1}^k \mathcal{S}_i$. Since the sum $\sum_{i=1}^k \mathcal{S}_i$ can be much larger than the union $\cup_{i=1}^k \mathcal{S}_i$, the specifics of the individual subspaces are not well considered and so the recovery may be inaccurate.

To better handle the mixed data, here we suggest a more general rank minimization problem defined as follows:

$$\min_{Z,E} \text{rank}(Z) + \lambda \|E\|_\ell, \quad \text{s.t.} \quad X = AZ + E, \quad (3)$$

where A is a “dictionary” that linearly spans the data space. We call the minimizer Z^* (with regard to the variable Z) the “lowest rank representation” of data X with respect to a dictionary A . After obtaining an optimal solution (Z^*, E^*) , we could recover the original data by using AZ^* (or $X - E^*$). Since $\text{rank}(AZ^*) \leq \text{rank}(Z^*)$, AZ^* is also a low-rank recovery to the original data X_0 . By setting $A = I$, the formulation (3) falls back to (2). So LRR could be regarded as a generalization of RPCA that essentially uses the standard bases as the dictionary. By choosing an appropriate dictionary A , as we will see, the lowest rank representation can recover the underlying row space so as to reveal the true segmentation of data. So, LRR could handle well the data drawn from a union of multiple subspaces.

4.2 Analysis on the LRR Problem

The optimization problem (3) is difficult to solve due to the discrete nature of the rank function. For ease of exploration, we begin with the “ideal” case that the data is clean. That is, we consider the following rank minimization problem:

$$\min_Z \text{rank}(Z), \quad \text{s.t.} \quad X = AZ. \quad (4)$$

It is easy to see that the solution to (4) may not be unique. As a common practice in rank minimization problems, we replace the rank function with the nuclear norm, resulting in the following convex optimization problem:

$$\min_Z \|Z\|_*, \quad \text{s.t.} \quad X = AZ. \quad (5)$$

We will show that the solution to (5) is also a solution to (4) and this special solution is useful for subspace segmentation.

In the following, we shall show some general properties of the minimizer to problem (5). These general conclusions form the foundations of LRR (the proofs can be found in Appendix, which is available in the online supplemental material).

4.2.1 Uniqueness of the Minimizer

The nuclear norm is convex, but not strongly convex. So it is possible that (5) has multiple optimal solutions. Fortunately, it can be proven that the minimizer to (5) is *always* uniquely defined by a closed form. This is summarized in the following theorem.

Theorem 4.1. Assume $A \neq 0$ and $X = AZ$ have feasible solution(s), i.e., $X \in \text{span}(A)$. Then,

$$Z^* = A^\dagger X \quad (6)$$

is the unique minimizer to (5), where A^\dagger is the pseudo-inverse of A .

From the above theorem, we have the following corollary which shows that (5) is a good surrogate of (4).

Corollary 4.1. Assume $A \neq 0$ and $X = AZ$ have feasible solutions. Let Z^* be the minimizer to (5), then $\text{rank}(Z^*) = \text{rank}(X)$ and Z^* is also a minimal rank solution to (4).

4.2.2 Block-Diagonal Property of the Minimizer

By choosing an appropriate dictionary, the lowest rank representation can reveal the true segmentation results. Namely, when the columns of A and X are exactly sampled from independent subspaces, the minimizer to (5) can reveal the subspace membership among the samples. Let $\{S_1, S_2, \dots, S_k\}$ be a collection of k subspaces, each of which has a rank (dimension) of $r_i > 0$. Also, let $A = [A_1, A_2, \dots, A_k]$ and $X = [X_1, X_2, \dots, X_k]$. Then we have the following theorem.

Theorem 4.2. Without loss of generality, assume that A_i is a collection of m_i samples of the i th subspace S_i , X_i is a collection of n_i samples from S_i , and the sampling of each A_i is sufficient such that $\text{rank}(A_i) = r_i$ (i.e., A_i can be regarded as the bases that span the subspace). If the subspaces are independent, then the minimizer to (5) is block-diagonal:

$$Z^* = \begin{bmatrix} Z_1^* & 0 & 0 & 0 \\ 0 & Z_2^* & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Z_k^* \end{bmatrix},$$

where Z_i^* is an $m_i \times n_i$ coefficient matrix with $\text{rank}(Z_i^*) = \text{rank}(X_i)$, $\forall i$.

Note that the claim of $\text{rank}(Z_i^*) = \text{rank}(X_i)$ guarantees the high within-class homogeneity of Z_i^* since the low-rank properties generally require Z_i^* to be dense. This is different from SR, which is prone to produce a “trivial” solution if $A = X$ because the sparsest representation is an identity matrix in this case. It is also worth noting that the above block-diagonal property does not require the data samples to have been grouped together according to their subspace memberships. There is no loss of generality to assume that the indices of the samples have been rearranged to satisfy the true subspace memberships, because the solution produced by LRR is globally optimal and does not depend on the arrangements of the data samples.

4.3 Recovering Low-Rank Matrices by Convex Optimization

Corollary 4.1 suggests that it is appropriate to use the nuclear norm as a surrogate to replace the rank function in (3). Also, the matrix ℓ_1 and $\ell_{2,1}$ norms are good relaxations of the ℓ_0 and $\ell_{2,0}$ norms, respectively. So we could obtain a low-rank recovery to X_0 by solving the following convex optimization problem:

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad X = AZ + E. \quad (7)$$

Here, the $\ell_{2,1}$ norm is adopted to characterize the error term E since we want to model the sample-specific corruptions (and outliers) as shown in Fig. 2c. For the small Gaussian noise as shown in Fig. 2a, $\|E\|_F^2$ should be chosen; for the random corruptions as shown in Fig. 2b, $\|E\|_1$ is an appropriate choice. After obtaining the minimizer (Z^*, E^*) , we could use AZ^* (or $X - E^*$) to obtain a low-rank recovery to the original data X_0 .

The optimization problem (7) is convex and can be solved by various methods. For efficiency, we adopt in this paper the Augmented Lagrange Multiplier (ALM) [36], [37] method. We first convert (7) to the following equivalent problem:

$$\min_{Z, E, J} \|J\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad X = AZ + E, Z = J.$$

This problem can be solved by the ALM method, which minimizes the following augmented Lagrangian function:

$$\begin{aligned} \mathcal{L} = & \|J\|_* + \lambda \|E\|_{2,1} + \text{tr}(Y_1^T (X - AZ - E)) \\ & + \text{tr}(Y_2^T (Z - J)) + \frac{\mu}{2} (\|X - AZ - E\|_F^2 + \|Z - J\|_F^2). \end{aligned}$$

The above problem is unconstrained. So it can be minimized with respect to J , Z , and E , respectively, by fixing the other variables and then updating the Lagrange multipliers Y_1 and Y_2 , where $\mu > 0$ is a penalty parameter. The inexact ALM method, also called the alternating direction method, is

outlined in Algorithm 1.² Note that although Steps 1 and 3 of the algorithm are convex problems, they both have closed-form solutions. Step 1 is solved via the Singular Value Thresholding (SVT) operator [38], while Step 3 is solved via the following lemma.

Lemma 4.1 ([39]). *Let Q be a given matrix. If the optimal solution to*

$$\min_W \alpha \|W\|_{2,1} + \frac{1}{2} \|W - Q\|_F^2$$

is W^ , then the i th column of W^* is*

$$[W^*]_{:,i} = \begin{cases} \frac{\|[Q]_{:,i}\|_2 - \alpha}{\|[Q]_{:,i}\|_2} Q_{:,i}, & \text{if } \|[Q]_{:,i}\|_2 > \alpha; \\ 0, & \text{otherwise.} \end{cases}$$

4.3.1 Convergence Properties

When the objective function is smooth, the convergence of the exact ALM algorithm has been generally proven in [37]. For inexact ALM, which is a variation of exact ALM, its convergence has also been well studied when the number of blocks is at most two [36], [40]. Up to the present, it is still difficult to *generally* ensure the convergence of inexact ALM with three or more blocks [40]. Since there are three blocks (including Z , J , and E) in Algorithm 1 and the objective function of (7) is not smooth, it would be not easy to prove the convergence in theory.

Algorithm 1. Solving Problem (7) by Inexact ALM

Input: data matrix X , parameter λ .

Initialize: $Z = J = 0, E = 0, Y_1 = 0, Y_2 = 0, \mu = 10^{-6}, \mu_{max} = 10^6, \rho = 1.1$, and $\varepsilon = 10^{-8}$.

while not converged **do**

1. fix the others and update J by

$$J = \arg \min_{\frac{1}{\mu} \|J\|_* + \frac{1}{2} \|J - (Z + Y_2/\mu)\|_F^2}.$$

2. fix the others and update Z by

$$Z = (I + A^T A)^{-1} (A^T (X - E) + J + (A^T Y_1 - Y_2)/\mu).$$

3. fix the others and update E by

$$E = \arg \min_{\frac{\lambda}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - (X - AZ + Y_1/\mu)\|_F^2}.$$

4. update the multipliers

$$Y_1 = Y_1 + \mu(X - AZ - E),$$

$$Y_2 = Y_2 + \mu(Z - J).$$

5. update the parameter μ by $\mu = \min(\rho\mu, \mu_{max})$.

6. check the convergence conditions:

$$\|X - AZ - E\|_\infty < \varepsilon \text{ and } \|Z - J\|_\infty < \varepsilon.$$

end while

Fortunately, there actually exist some guarantees for ensuring the convergence of Algorithm 1. According to the theoretical results in [41], two conditions are *sufficient* (but may not necessary) for Algorithm 1 to converge: The first condition is that the dictionary matrix A is of full column rank; the second one is that the optimality gap produced in

2. To solve the problem $\min_{Z,E} \|Z\|_* + \lambda \|E\|_1$, s.t. $X = AZ + E$, one only needs to replace Step 3 of Algorithm 1 by $E = \arg \min_{\frac{\lambda}{\mu} \|E\|_1 + \frac{1}{2} \|E - (X - AZ + Y_1/\mu)\|_F^2}$, which is solved by using the shrinkage operator [36]. Also, please note here that the setting of $\varepsilon = 10^{-8}$ is based on the assumption that the values in X has been normalized within the range of $[0, 1]$.

each iteration step is monotonically decreasing, namely, the error

$$\epsilon_k = \left\| (Z_k, J_k) - \arg \min_{Z,J} \mathcal{L} \right\|_F^2$$

is monotonically decreasing, where Z_k (respectively, J_k) denotes the solution produced at the k th iteration, $\arg \min_{Z,J} \mathcal{L}$ indicates the “ideal” solution obtained by minimizing the Lagrangian function \mathcal{L} with respect to both Z and J simultaneously. The first condition is easy to obey since (7) can be converted into an equivalent problem where the full column rank condition is always satisfied (we will show this in the next subsection). For the monotonically decreasing condition, although it is not easy to *strictly* prove it, the convexity of the Lagrangian function could guarantee its validity to some extent [41]. So, it could be well expected that Algorithm 1 has good convergence properties. Moreover, inexact ALM is known to *generally* perform well in reality, as illustrated in [40].

That μ should be upper bounded (Step 5 of Algorithm 1) is required by the traditional theory of the alternating direction method in order to guarantee the convergence of the algorithm. So we also adopt this convention. Nevertheless, please note that the upper boundedness may not be necessary for some particular problems, e.g., the RPCA problem as analyzed in [36].

4.3.2 Computational Complexity

For ease of analysis, we assume that the sizes of both A and X are $d \times n$ in the following. The major computation of Algorithm 1 is Step 1, which requires computing the SVD of an $n \times n$ matrix. So it will be time consuming if n is large, i.e., the number of data samples is large. Fortunately, the computational cost of LRR can be easily reduced by the following theorem, which follows from Theorem 4.1.

Theorem 4.3. *For any optimal solution (Z^*, E^*) to the LRR problem (7), we have that*

$$Z^* \in \text{span}(A^T).$$

The above theorem concludes that the optimal solution Z^* (with respect to the variable Z) to (7) always lies within the subspace spanned by the rows of A . This means that Z^* can be factorized into $Z^* = P^* \tilde{Z}^*$, where P^* can be computed in advance by orthogonalizing the columns of A^T . Hence, (7) can be equivalently transformed into a simpler problem by replacing Z with $P^* \tilde{Z}$:

$$\min_{\tilde{Z}, E} \|\tilde{Z}\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t. } X = B\tilde{Z} + E,$$

where $B = AP^*$. After obtaining a solution (\tilde{Z}^*, E^*) to the above problem, the optimal solution to (7) is recovered by $(P^* \tilde{Z}^*, E^*)$. Since the number of rows of \tilde{Z} is at most r_A (the rank of A), the above problem can be solved with a complexity of $O(dnr_A + nr_A^2 + r_A^3)$ by using Algorithm 1. So LRR is quite scalable for large-size (n is large) datasets, provided that a low-rank dictionary A has been obtained. While using $A = X$, the computational complexity is at most $O(d^2n + d^3)$ (assuming $d \leq n$). This is also fast provided that the data dimension d is not high.

While considering the cost of orthogonalization and the number of iterations needed to converge, the complexity of Algorithm 1 is

$$O(d^2n) + O(n_s(dnr_A + nr_A^2 + r_A^3)),$$

where n_s is the number of iterations. The iteration number n_s depends on the choice of ρ : n_s is smaller while ρ is larger, and vice versa. Although larger ρ does produce higher efficiency, it has the risk of losing optimality to use large ρ [36]. In our experiments, we always set $\rho = 1.1$. Under this setting, the iteration number usually locates within the range of 50-300.

5 SUBSPACE RECOVERY BY LRR

In this section, we utilize LRR to address Problem 3.1, which is to recover the original row space from a set of corrupted observations. Both theoretical and experimental results will be presented.

5.1 Exactness to Clean Data

When there are no errors in data, i.e., $X = X_0$ and $E_0 = 0$, it is simple to show that the row space (identified by $V_0V_0^T$) of X_0 is exactly recovered by solving the following nuclear norm minimization problem:

$$\min_Z \|Z\|_*, \quad \text{s.t.} \quad X = XZ, \quad (8)$$

which is to choose the data matrix X itself as the dictionary in (5). By Theorem 4.1, we have the following theorem, which has also been proven by Wei and Lin [42].

Theorem 5.1. *Suppose the skinny SVD of X is $U\Sigma V^T$, then the minimizer to (8) is uniquely defined by*

$$Z^* = VV^T.$$

This naturally implies that Z^ exactly recovers $V_0V_0^T$ when X is clean (i.e., $E_0 = 0$).*

The above theorem reveals the connection between LRR and the method in [12], which is a counterpart of PCA (referred to as “PCA” for simplicity). Nevertheless, it is well known that PCA is fragile to the presence of outliers. In contrast, it can be proven in theory that LRR exactly recovers the row space of X_0 from the data contaminated by outliers, as will be shown in the next section.

5.2 Robustness to Outliers and Sample-Specific Corruptions

Assumption 2 is to imagine that a fraction of the data samples are away from the underlying subspaces. This implies that the error term E_0 has sparse column supports. So, the $\ell_{2,1}$ norm is appropriate for characterizing E_0 . By choosing $A = X$ in (7), we have the following convex optimization problem:

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad X = XZ + E. \quad (9)$$

The above formulation “seems” questionable because the data matrix (which itself can contain errors) is used as the dictionary for error correction. Nevertheless, as shown in

the following two subsections, $A = X$ is indeed a good choice for several particular problems.³

5.2.1 Exactness to Outliers

When an observed data sample is far away from the underlying subspaces, a typical regime is that this sample is from a different model other than subspaces, a so-called *outlier*.⁴ In this case, the data matrix X contains two parts, one part consists of authentic samples (denoted by X_0) strictly drawn from the underlying subspaces and the other part consists of outliers (denoted as E_0) that are not subspace members. To precisely describe this setting, we need to impose an additional constraint on X_0 , that is,

$$\mathcal{P}_{\mathcal{I}_0}(X_0) = 0, \quad (10)$$

where \mathcal{I}_0 is the indices of the outliers (i.e., the column supports of E_0). Furthermore, we use n to denote the total number of data samples in X , $\gamma \triangleq |\mathcal{I}_0|/n$ the fraction of outliers, and r_0 the rank of X_0 . With these notations, we have the following theorem, which states that LRR can exactly recover the row space of X_0 and identify the indices of outliers as well.

Theorem 5.2 ([43]). *There exists $\gamma^* > 0$ such that LRR with parameter $\lambda = 3/(7\|X\|\sqrt{\gamma^*n})$ strictly succeeds, as long as $\gamma \leq \gamma^*$. Here, the success is in the sense that any minimizer (Z^*, E^*) to (9) can produce*

$$U^*(U^*)^T = V_0V_0^T \text{ and } \mathcal{I}^* = \mathcal{I}_0, \quad (11)$$

where U^ is the column space of Z^* and \mathcal{I}^* is the column supports of E^* .*

There are several important notices in the above theorem. First, although the objective function (9) is not strongly convex and multiple minimizers may exist, it is proven that *any* minimizer is effective for subspace recovery. Second, the coefficient matrix Z^* itself does not recover $V_0V_0^T$ (notice that Z^* is usually asymmetric except $E^* = 0$), and it is the column space of Z^* that recovers the row space of X_0 . Third, the performance of LRR is measured by the value of γ^* (the larger, the better), which depends on some data properties such as the incoherence and the extrinsic rank r_0 (γ^* is larger when r_0 is lower). For more details, please refer to [43].

Fig. 4 shows some experimental results which verify the conclusions of Theorem 5.2. Notice that the parameter setting $\lambda = 3/(7\|X\|\sqrt{\gamma^*n})$ is based on the condition $\gamma \leq \gamma^*$ (i.e., the outlier fraction is smaller than a certain threshold), which is just a sufficient (but not necessary) condition for ensuring the success of LRR. So, in practice (even for synthetic examples), where $\gamma > \gamma^*$ it is possible that other values of λ achieve better performances.

5.2.2 Robustness to Sample-Specific Corruptions

For the phenomenon that an observed sample is away from the subspaces, another regime is that this sample is an

3. Note that this does not deny the importance of learning the dictionary. Indeed, the choice of dictionary is a very important aspect in LRR. We leave this as future work.

4. Precisely, we define an outlier as a data vector that is independent of the samples drawn from the subspaces [43].

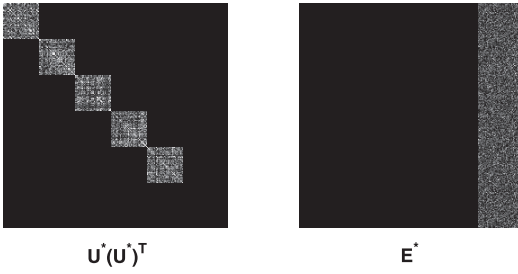


Fig. 4. An example of the matrices $U^*(U^*)^T$ and E^* computed from the data contaminated by outliers. In a similar way as [14], we create five pairwise disjoint subspaces, each of which is of dimension 4, and draw 40 samples (with ambient dimension 200) from each subspace. Then, 50 outliers are randomly generated from $\mathcal{N}(0, s)$, where the standard deviation s is set to be three times as large as the averaged magnitudes of the samples. By choosing $0.16 \leq \lambda \leq 0.34$, LRR produces a solution (Z^*, E^*) with the column space of Z^* exactly recovering the row space of X_0 , and the column supports of E^* exactly identifying the indices of the outliers.

authentic subspace member, but grossly corrupted. Usually, such corruptions only happen on a small fraction of data samples, so-called “sample-specific” corruptions. The modeling of sample-specific corruptions is the same as outliers, because in both cases E_0 has sparse column supports. So the formulation (9) is still applicable. However, the setting (10) is no longer valid, and thus LRR may not exactly recover the row space $V_0 V_0^T$ in this case. Empirically, the conclusion of $\mathcal{I}^* = \mathcal{I}_0$ still holds [14], which means that the column supports of E^* can identify the indices of the corrupted samples.

While both outliers and sample-specific corruptions⁵ are handled in the same way, a question is how to deal with the cases where the authentic samples are heavily corrupted to have similar properties as the outliers. If a sample is heavily corrupted so as to be independent from the underlying subspaces, it will be treated as an outlier in LRR, as illustrated in Fig. 5. This is a reasonable manipulation. For example, it is appropriate to treat a face image as a nonface outlier if the image has been corrupted to be look like something else.

5.3 Robustness in the Presence of Noise, Outliers, and Sample-Specific Corruptions

When there is noise in the data, the column supports of E_0 are not strictly sparse. Nevertheless, the formulation (9) is still applicable, because the $\ell_{2,1}$ norm (which is relaxed from the $\ell_{2,0}$ norm) can handle the signals that approximately have sparse column supports well. Since all observations may be contaminated, it is unlikely in theory that the row space $V_0 V_0^T$ can be exactly recovered. So we target near recovery in this case. By the triangle inequality of matrix norms, the following theorem can be simply proven without any assumptions.

Theorem 5.3. *Let the size of X be $d \times n$, and the rank of X_0 be r_0 . For any minimizer (Z^*, E^*) to (9) with $\lambda > 0$, we have*

$$\|Z^* - V_0 V_0^T\|_F \leq \min(d, n) + r_0.$$

5. Unlike outlier, a corrupted sample is unnecessary to be independent of the clean samples.



Fig. 5. Two examples of the matrix $U^*(U^*)^T$ computed from the data corrupted by sample-specific corruptions. (a) The magnitudes of the corruptions are set to be about 0.7 times as large as the samples. Considering $|U^*(U^*)^T|$ as an affinity matrix, the average affinity degree of the corrupted samples is about 40, which means that the corrupted samples can be projected back onto their respective subspaces. (b) The magnitudes of the corruptions are set to be about 3.5 times as large as the samples. The affinity degrees of the corrupted samples are all zero, which means that the corrupted samples are treated as outliers. In these experiments, the data samples are generated in the same way as in Fig. 4. Then, 10 percent samples are randomly chosen to be corrupted by adding errors of Gaussian distribution. For each experiment, the parameter λ is carefully determined such that the column supports of E^* identify the indices of the corrupted samples.

Fig. 6 demonstrates the performance of LRR, in the presence of noise, outliers, and sample-specific corruptions. It can be seen that the results produced by LRR are quite promising.

One may have noticed that the bound given in the above theorem is somewhat loose. To obtain a more accurate bound in theory, one needs to relax the equality constraint of (9) into:

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad \|X - XZ - E\|_F \leq \xi,$$

where ξ is a parameter for characterizing the amount of the dense noise (Fig. 2a) possibly existing in data. The above problem can be solved by ALM, in a similar procedure as Algorithm 1. However, the above formulation needs to invoke another parameter ξ , and thus we do not further explore it in this paper.

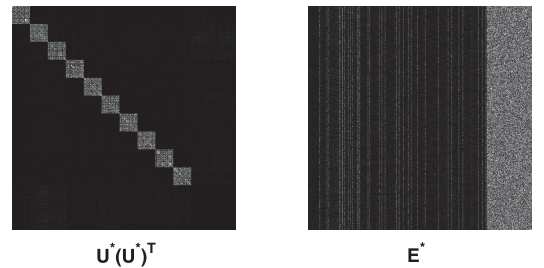


Fig. 6. An example of the matrices $U^*(U^*)^T$ and E^* computed from the data contaminated by noise, outliers, and sample-specific corruptions. In this experiment, first we create 10 pairwise disjoint subspaces (each of which is of dimension 4) and draw 40 samples (with ambient dimension 2,000) from each subspace. Second, we randomly choose 10 percent samples to be grossly corrupted by large errors. The remaining 90 percent samples are slightly corrupted by small errors. Finally, as in Fig. 4, 100 outliers are randomly generated. The total amount of errors (including noise, sample-specific corruptions, and outliers) is given by $\|E_0\|_F / \|X_0\|_F = 0.63$. By setting $\lambda = 0.3$, $U^*(U^*)^T$ approximately recovers $V_0 V_0^T$ with error $\|U^*(U^*)^T - V_0 V_0^T\|_F / \|V_0 V_0^T\|_F = 0.17$, and the column supports of E^* accurately identify the indices of the outliers and corrupted samples. In contrast, the recover error produced by PCA is 0.66 and that by the RPCA method (using the best parameters) introduced in [16] is 0.23.

5.4 Algorithms for Subspace Segmentation, Model Estimation, and Outlier Detection

5.4.1 Segmentation with Given Subspace Number

After obtaining (Z^*, E^*) by solving (9), the matrix $U^*(U^*)^T$ that identifies the column space of Z^* is useful for subspace segmentation. Letting the skinny SVD of Z^* be $U^*\Sigma^*(V^*)^T$, we define an affinity matrix W as follows:

$$[W]_{ij} = ([\tilde{U}\tilde{U}^T]_{ij})^2, \quad (12)$$

where \tilde{U} is formed by $U^*(\Sigma^*)^{\frac{1}{2}}$ with normalized rows. Here, for obtaining better performance on corrupted data, we assign each column of U^* a weight by multiplying $(\Sigma^*)^{\frac{1}{2}}$. Notice that when the data is clean, $\Sigma^* = \mathbf{I}$ and thus this technique does not take any effects. The technical detail of using $(\cdot)^2$ is to ensure that the values of the affinity matrix W are positive (note that the matrix $\tilde{U}\tilde{U}^T$ can have negative values). Finally, we could use the spectral clustering algorithms such as Normalized Cuts [26] to segment the data samples into a given number k of clusters. Algorithm 2 summarizes the whole procedure of performing segmentation by LRR.

Algorithm 2. Subspace Segmentation

Input: data matrix X , number k of subspaces.

1. obtain the minimizer Z^* to problem (9).
2. compute the skinny SVD $Z^* = U^*\Sigma^*(V^*)^T$.
3. construct an affinity matrix W by (12).
4. use W to perform NCut and segment the data samples into k clusters.

5.4.2 Estimating the Subspace Number k

Although it is generally challenging to estimate the number of subspaces (i.e., number of clusters), it is possible to resolve this model estimation problem due to the block-diagonal structure of the affinity matrix produced by specific algorithms [13], [44], [45]. While a strictly block-diagonal affinity matrix W is obtained, the subspace number k can be found by first computing the normalized Laplacian (denoted as L) matrix of W and then counting the number of zero singular values of L . While the obtained affinity matrix is just near block-diagonal (this is the case in reality), one could predict the subspace number as the number of singular values smaller than a threshold. Here, we suggest a soft thresholding approach that outputs the estimated subspace number \hat{k} by

$$\hat{k} = n - \text{int}\left(\sum_{i=1}^n f_{\tau}(\sigma_i)\right). \quad (13)$$

Here, n is the total number of data samples, $\{\sigma_i\}_{i=1}^n$ are the singular values of the Laplacian matrix L , $\text{int}(\cdot)$ is the function that outputs the nearest integer of a real number, and $f_{\tau}(\cdot)$ is a soft thresholding operator defined as

$$f_{\tau}(\sigma) = \begin{cases} 1, & \text{if } \sigma \geq \tau, \\ \log_2(1 + \frac{\sigma^2}{\tau^2}), & \text{otherwise,} \end{cases}$$

where $0 < \tau < 1$ is a parameter. Algorithm 3 summarizes the whole procedure of estimating the subspace number based on LRR.

Algorithm 3. Estimating the Subspace Number k

Input: data matrix X .

1. compute the affinity matrix W in the same way as in Algorithm 2.
2. compute the Laplacian matrix $L = \mathbf{I} - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, where $D = \text{diag}(\sum_j [W]_{1j}, \dots, \sum_j [W]_{nj})$.
3. estimate the subspace number by (13).

5.4.3 Outlier Detection

As shown in Theorem 5.2, the minimizer E^* (with respect to the variable E) can be used to detect the outliers that possibly exist in data. This can be simply done by finding the nonzero columns of E^* , when all or a fraction of data samples are clean (i.e., Assumptions 1 and 2). For the cases where the learned E^* only approximately has sparse column supports, one could use thresholding strategy, that is, the i th data vector of X is judged to be outlier if and only if

$$\| [E^*]_{:,i} \|_2 > \delta, \quad (14)$$

where $\delta > 0$ is a parameter.

Since the affinity degrees of the outliers are zero or close to being zero (see Figs. 4 and 6), the possible outliers can be also removed by discarding the data samples whose affinity degrees are smaller than a certain threshold. Such a strategy is commonly used in spectral-type methods [13], [34]. Generally, the underlying principle of this strategy is essential the same as (14). Comparing to the strategy of characterizing the outliers by affinity degrees, there is an advantage of using E^* to indicate outliers, that is, formulation (9) can be easily extended to include more priors, e.g., the multiple visual features as done in [18] and [19].

6 EXPERIMENTS

LRR has been used to achieve state-of-the-art performance in several applications such as motion segmentation [4], image segmentation [18], face recognition [4], and saliency detection [19]. In the experiments of this paper, we shall focus on analyzing the essential aspects of LRR under the context of subspace segmentation and outlier detection.

6.1 Experimental Data

6.1.1 Hopkins155

To verify the segmentation performance of LRR, we adopt for experiments the Hopkins155 [46] motion database, which provides an extensive benchmark for testing various subspace segmentation algorithms. In Hopkins155, there are 156 video sequences along with the features extracted and tracked in all the frames. Each sequence is a sole dataset (i.e., data matrix) and so there are in total 156 datasets of different properties, including the number of subspaces, the data dimension, and the number of data samples. Although the outliers in the data have been manually removed and the overall error level is low, some sequences (about 10 sequences) are grossly corrupted and have notable error levels. Table 1 summarizes some information about Hopkins155. For a sequence represented as a data matrix X , its error level is estimated by its rank- r approximation: $\|X - U_r \Sigma_r V_r^T\|_F / \|X\|_F$, where Σ_r contains the largest r singular values of X , and U_r (respectively, V_r)

TABLE 1
Some Information about Hopkins155

	data dimension	# of data samples	# of subspaces	error level
max	201	556	3	0.0130
min	31	39	2	0.0002
mean	59.7	295.7	2.3	0.0009
std.	20.2	140.8	0.5	0.0012

is formed by taking the top r left (respectively, right) singular vectors. Here, we set $r = 4k$ (k is the subspace number of the sequence) due to the fact that the rank of each subspace in motion data is at most 4.

6.1.2 Yale-Caltech

To test LRR's effectiveness in the presence of outliers and corruptions, we create a dataset by combining the Extended Yale Database B [47] and Caltech101 [48]. For Extended Yale Database B, we remove the images pictured under extreme light conditions. Namely, we only use the images with view directions smaller than 45 degrees and light source directions smaller than 60 degrees, resulting in 1,204 authentic samples approximately drawn from a union of 38 low-rank subspaces (each face class corresponds to a subspace). For Caltech101, we only select the classes containing no more than 40 images, resulting in 609 nonface outliers. Fig. 7 shows some examples of this dataset.

6.2 Baselines and Evaluation Metrics

Due to the close connections between PCA and LRR, we choose PCA and RPCA methods as the baselines. Moreover, some previous subspace segmentation methods are also considered.

6.2.1 PCA (i.e., SIM)

The PCA method is widely used for dimension reduction. Actually, it can also be applied to subspace segmentation and outlier detection as follows: First, we use SVD to obtain the rank- r (r is a parameter) approximation of the data matrix X , denoted as $X \approx U_r \Sigma_r V_r^T$; second, we utilize $V_r V_r^T$, which is an estimation of the true SIM $V_0 V_0^T$, for subspace segmentation in a similar way as Algorithm 2 (the only difference is the estimation of SIM); finally, we compute $E_r = X - U_r \Sigma_r V_r^T$ and use E_r to detect outliers according to (14).

6.2.2 Robust PCA

As an improvement over PCA, the robust PCA methods can also do subspace segmentation and outlier detection. In this paper, we consider two RPCA methods introduced in [7], [16], and [42] which are based on minimizing

$$\min_{D, E} \|D\|_* + \lambda \|E\|_\ell, \quad \text{s.t.} \quad X = D + E.$$

In [7], the ℓ_1 norm is used to characterize random corruptions, so referred to as "RPCA₁." In [16] and [42], the $\ell_{2,1}$ norm is adopted for detecting outliers, so referred to as "RPCA_{2,1}." The detailed procedures for subspace segmentation and outlier detection are almost the same as the PCA case above. The only difference is that V_r is formed from the skinny SVD of D^* (not X), which is obtained by solving the above optimization problem. Note here that the value of r is determined by the parameter λ , and thus one only needs to select λ .

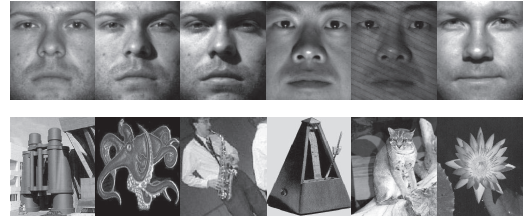


Fig. 7. Examples of the images in the Yale-Caltech dataset.

6.2.3 Sparse Representation

LRR has similar appearance as SR, which has been applied to subspace segmentation [13]. For fair comparison, in this paper, we implement an $\ell_{2,1}$ -norm-based SR method that computes an affinity matrix by minimizing

$$\min_{Z, E} \|Z\|_1 + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad X = XZ + E, [Z]_{ii} = 0.$$

Here, SR needs to enforce $[Z]_{ii} = 0$ to avoid the trivial solution $Z = I$. After obtaining a minimizer (Z^*, E^*) , we use $W = |Z^*| + |(Z^*)^T|$ as the affinity matrix to do subspace segmentation. The procedure of using E^* to perform outlier detection is the same as LRR.

6.2.4 Some other Methods

We also consider for comparison some previous subspace segmentation methods, including Random Sample Consensus [10], Generalized PCA (GPCA) [24], Local Subspace Analysis (LSA) [2], Agglomerative Lossy Compression (ALC) [3], Sparse Subspace Clustering [13], Spectral Clustering [31], Spectral Curvature Clustering [27], Multi Stage Learning (MSL) [49], Locally Linear Manifold Clustering (LLMC) [50], Local Best-fit Flats (LBF) [29], and Spectral LBF (SLBF) [29].

6.2.5 Evaluation Metrics

Segmentation accuracy (error) is used to measure the performance of segmentation. The areas under the Receiver Operator Characteristic (ROC) curve, known as AUC, are used for evaluating the quality of outlier detection. For more details about these two evaluation metrics, please refer to the Appendix, which is available in the online supplemental material.

6.3 Results on Hopkins155

6.3.1 Choosing the Parameter λ

The parameter $\lambda > 0$ is used to balance the effects of the two parts in (9). In general, the choice of this parameter depends on the prior knowledge of the error level of data. When the errors are slight, we should use relatively large λ ; when the errors are heavy, we should set λ to be relatively small.

Fig. 8a shows the evaluation results over all 156 sequences in Hopkins155: While λ ranges from 1 to 6, the segmentation error only varies from 1.69 to 2.81 percent; while λ ranges from 3 to 5, the segmentation error almost remains unchanged, slightly varying from 1.69 to 1.87 percent. This phenomenon is mainly due to two reasons: First, on most sequences (about 80 percent) which are almost clean and easy to segment, LRR could work well by choosing λ arbitrarily, as exemplified in Fig. 8b. Second, there is an "invariance" in LRR. Namely, Theorem 4.3 implies that the

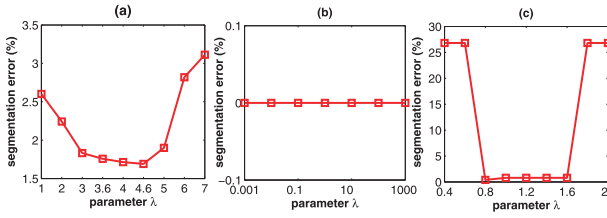


Fig. 8. The influences of the parameter λ of LRR. (a) On all 156 sequences of Hopkins155, the overall segmentation performance is equally good while $3 \leq \lambda \leq 5$. (b) On the 43rd sequence, the segmentation error is always 0 for $0.001 \leq \lambda \leq 1,000$. (c) On the 62nd sequence, the segmentation performance is good only when $0.8 \leq \lambda \leq 1.6$.

minimizer to (9) always satisfies $Z^* \in \text{span}(X^T)$. This implies that the solution of LRR can be partially stable while λ is varying.

The analysis above does not deny the importance of model selection. As shown in Fig. 8c, the parameter λ can largely affect the segmentation performance on some sequences. Actually, if we turn λ to the best for each sequence, the overall error rate is only 0.07 percent. Although this number is achieved in an “impractical” way, it verifies the significance of selecting the parameter λ , especially when the data is corrupted. For the experiments below, we choose $\lambda = 4$ for LRR.

6.3.2 Segmentation Performance

In this section, we show LRR’s performance in subspace segmentation with the subspace number given. For comparison, we also list the results of PCA, RPCA₁, RPCA_{2,1}, and SR (these methods are introduced in Section 6.2). Table 2 illustrates that LRR performs better than PCA and RPCA. Here, the advantages of LRR are mainly due to its methodology. More precisely, LRR *directly* targets on recovering the row space $V_0 V_0^T$, which provably determines the segmentation results. In contrast, PCA and RPCA methods are designed for recovering the column space $U_0 U_0^T$, which is designed for dimension reduction. One may have noticed that RPCA_{2,1} outperforms PCA and RPCA₁. If we use instead the ℓ_1 norm to regularize E in (9), the segmentation error is 2.03 percent ($\lambda = 0.6$, optimally determined). These illustrate that the errors in this database tend to be sample-specific.

Besides the superiorities in segmentation accuracy, another advantage of LRR is that it can work well under a wide range of parameter settings, as shown in Fig. 8, whereas RPCA methods are sensitive to the parameter λ . Taking RPCA_{2,1} for example, it achieves an error rate of 3.26 percent by choosing $\lambda = 0.32$. However, the error rate increases to 4.5 percent at $\lambda = 0.34$ and 3.7 percent at $\lambda = 0.3$.

TABLE 2
Segmentation Results (on Hopkins155)
of PCA, RPCA₁, RPCA_{2,1}, SR, and LRR

segmentation errors (%) over all 156 sequences					
	PCA	RPCA ₁	RPCA _{2,1}	SR	LRR
mean	4.56	4.13	3.26	3.89	1.71
std.	10.80	10.37	9.09	7.70	4.85
max	49.78	45.83	47.15	32.57	33.33
average run time (seconds) per sequence					
	0.2	0.8	0.8	4.2	1.9

TABLE 3
Results (on Hopkins155) of Estimating the Subspace Number

# total	# predicted	prediction rate (%)		absolute error		
156	121	77.6		0.25		
influences of the parameter τ						
parameter τ	0.06	0.07	0.08	0.09	0.10	0.11
prediction rate	66.7	71.2	77.6	75.0	72.4	71.2
absolute error	0.37	0.30	0.25	0.26	0.29	0.30

The efficiency (in terms of running time) of LRR is comparable to PCA and RPCA methods. Theoretically, the computational complexity (with regard to d and n) of LRR is the same as RPCA methods. LRR costs more computational time because its optimization procedure needs more iterations than RPCA to converge.

6.3.3 Performance of Estimating Subspace Number

Since there are 156 sequences in total, this database also provides a good benchmark for evaluating the effectiveness of Algorithm 3, which is to estimate the number of subspaces underlying a collection of data samples. Table 3 shows the results. By choosing $\tau = 0.08$, LRR correctly predicts the true subspace number of 121 sequences. The absolute error (i.e., $|\hat{k} - k|$) averaged over all sequences is 0.25. These results illustrate that it is hopeful to resolve the problem of estimating the subspace number, which is a challenging model estimation problem.

6.3.4 Comparing to State-of-the-Art Methods

Notice that previous methods only report the results for 155 sequences. After discarding the degenerate sequence, the error rate of LRR is 1.59 percent, which is comparable to the state-of-the-art methods, as shown in Table 4. The performance of LRR can be further improved by refining (9), which uses the observed data matrix X itself as the dictionary. When the data is corrupted by dense noise (this is usually true in reality), this certainly is not the best choice. In [51] and [42], a nonconvex formulation is adopted to learn the original data X_0 and its row space $V_0 V_0^T$ simultaneously:

$$\min_{D, Z, E} \|Z\|_* + \lambda \|E\|_1, \quad \text{s.t.} \quad X = D + E, D = DZ,$$

where the unknown variable D is used as the dictionary. This method can achieve an error rate of 1.22 percent. In [4], it is explained that the issues of choosing dictionary can be relieved by considering the unobserved, hidden data. Furthermore, it is deduced that the effects of hidden data can be approximately modeled by the following convex formulation:

TABLE 4
Segmentation Errors (Percent) on Hopkins155
(155 Sequences)

	GPCA	RANSAC	MSL	LSA	LLMC
mean	10.34	9.76	5.06	4.94	4.80
	PCA	LBF	ALC	SCC	SLBF
mean	4.47	3.72	3.37	2.70	1.35
LRR					
	SSC	SC	[51]	[4]	this paper
mean	1.24	1.20	1.22	0.85	1.59

TABLE 5
Segmentation Accuracy (ACC) and AUC Comparison
on the Yale-Caltech Data Set

	PCA	RPCA ₁	RPCA _{2,1}	SR	LRR
ACC (%)	77.15	82.97	83.72	73.17	86.13
AUC	0.9653	0.9819	0.9863	0.9239	0.9927
time (sec.)	0.6	60.8	59.2	383.5	152.6

$$\min_{Z, L, E} \|Z\|_* + \|L\|_* + \lambda \|E\|_1, \quad \text{s.t.} \quad X = XZ + LX + E,$$

which intuitively integrates subspace segmentation and feature extraction into a unified framework. This method can achieve an error rate of 0.85 percent, which outperforms other subspace segmentation algorithms.

While several methods have achieved an error rate below 3 percent on Hopkins155, the subspace segmentation problem is still far from being solved. A long term difficulty is how to solve the model selection problems, e.g., estimating the parameter λ of LRR. Also, it would not be trivial to handle more complicated datasets that contain more noise, outliers, and corruption.

6.4 Results on Yale-Caltech

The goal of this test is to identify 609 nonface outliers and segment the remaining 1,204 face images into 38 clusters. The performance of segmentation and outlier detection is evaluated by segmentation accuracy (ACC) and AUC, respectively. While investigating segmentation performance, the affinity matrix is computed from all images, including both the face images and nonface outliers. However, for the convenience of evaluation, the outliers and the corresponding affinities are removed (according to the ground truth) before using NCut to obtain the segmentation results.

We resize all images into 20×20 pixels and form a data matrix of size $400 \times 1,813$. Table 5 shows the results of PCA, RPCA, SR, and LRR. It can be seen that LRR is better than the PCA and RPCA methods in terms of both subspace segmentation and outlier detection. These experimental results are consistent with Theorem 5.2, which shows that LRR has a stronger guarantee than RPCA methods in performance. Notice that SR is behind the others.⁶ This is because the presence or absence of outliers is unnecessary to notably alert the sparsity of the reconstruction coefficients, and thus it is hard for SR to handle the data contaminated by outliers well.

Fig. 9 shows the performance of LRR while the parameter λ varies from 0.06 to 0.22. Notice that LRR is more sensitive to λ on this dataset than on Hopkins155. This is because the error level of Hopkins155 is quite low (see Table 1), whereas, the Yale-Caltech dataset contains outliers and corrupted images (see Fig. 7).

To visualize LRR's effectiveness in error correction, we create another data matrix with size $8,064 \times 1,813$ by resizing all images into 96×84 . Fig. 10 shows some results produced by LRR. It is worth noting that the "error" term E^* can contain "useful" information, e.g., the eyes and salient objects. Here, the principle is to decompose the data matrix into a low-rank part and a sparse part, with the low-rank

6. The results (for outlier detection) in Table 5 are obtained by using the strategy of (14). While using the strategy of checking the affinity degree, the results produced by SR are even worse, only achieving an AUC of 0.81 by using the best parameters.

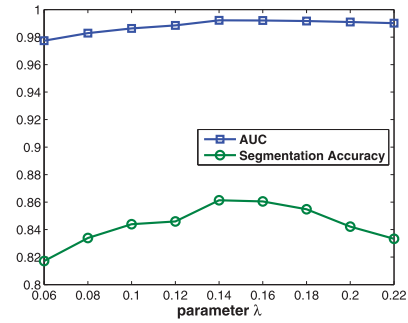


Fig. 9. The influences of the parameter λ of LRR. These results are collected from the Yale-Caltech dataset. All images are resized to 20×20 pixels.

part (XZ^*) corresponding to the principal features of the whole dataset and the sparse part (E^*) corresponding to the rare features which cannot be modeled by low-rank subspaces. This implies that it is possible to use LRR to extract the discriminative features and salient regions, as done in face recognition [4] and saliency detection [19].

7 CONCLUSION AND FUTURE WORK

In this paper, we have proposed low-rank representation to identify the subspace structures from corrupted data. Namely, our goal is to segment the samples into their respective subspaces and correct the possible errors simultaneously. LRR is a generalization of the recently established RPCA methods [7], [16], extending the recovery of corrupted data from single subspace to multiple subspaces. Also, LRR generalizes the approach of Shape Interaction Matrix, giving a way to define an SIM between two different matrices (see Theorem 4.1), and providing a mechanism to recover the true SIM (or row space) from corrupted data. Both theoretical and experimental results show the effectiveness of LRR. However, there still remain several problems for future work.

- It may achieve significant improvements by learning a dictionary A , which partially determines the

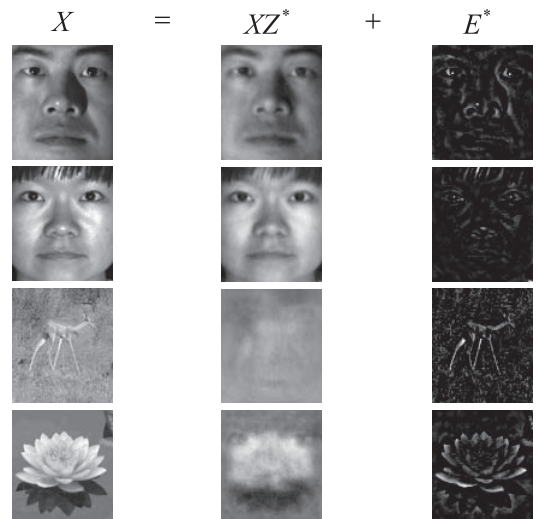


Fig. 10. Some examples of using LRR to correct the errors in the Yale-Caltech dataset. Left: The original data matrix X ; middle: the corrected data XZ^* ; right: the error E^* .

solution of LRR. In order to exactly recover the row space V_0 , Theorem 4.3 illustrates that the dictionary A must satisfy the condition of $V_0 \in \text{span}(A^T)$. When the data is only contaminated by outliers, this condition can be obeyed by simply choosing $A = X$. However, this choice cannot ensure the validity of $V_0 \in \text{span}(A^T)$ while the data contains other types of errors, e.g., dense noise.

- The proofs of Theorem 5.2 are specific to the case of $A = X$. As a future direction, it is interesting to see whether the technique presented can be extended to general dictionary matrices other than X .
- A critical issue in LRR is how to estimate or select the parameter λ . For the data contaminated by various errors such as noise, outliers, and corruptions, the estimation of λ is quite challenging.
- The subspace segmentation should not be the only application of LRR. Actually, it has been successfully used in the applications other than segmentation, e.g., saliency detection [19]. In general, the presented LRR method can be extended to solve various applications well.

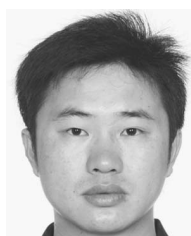
ACKNOWLEDGMENTS

Z. Lin is the corresponding author for this paper. Z. Lin is partly supported by the National Natural Science Foundation of China (Grant Nos. 61272341 and 61231002).

REFERENCES

- [1] W. Gear, "Multibody Grouping from Motion Images," *Int'l J. Computer Vision*, vol. 29, no. 2, pp. 133-150, 1998.
- [2] J. Yan and M. Pollefeys, "A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-Rigid, Degenerate and Non-Degenerate," *Proc. European Conf. Computer Vision*, vol. 4, pp. 94-106, 2006.
- [3] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1832-1845, Oct. 2010.
- [4] G. Liu and S. Yan, "Latent Low-Rank Representation for Subspace Segmentation and Feature Extraction," *Proc. IEEE Int'l Conf. Computer Vision*, 2011.
- [5] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1546-1562, Sept. 2007.
- [6] E. Candès and Y. Plan, "Matrix Completion with Noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925-936, June 2010.
- [7] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *J. ACM*, 2009.
- [8] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering Appearances of Objects under Varying Illumination Conditions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 11-18, 2003.
- [9] G. Liu, Z. Lin, X. Tang, and Y. Yu, "Unsupervised Object Segmentation with a Hybrid Graph Model (HGM)," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 910-924, May 2010.
- [10] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, 1981.
- [11] C. Zhang and R. Bitmead, "Subspace System Identification for Training-Based MIMO Channel Estimation," *Automatica*, vol. 41, no. 9, pp. 1623-1632, 2005.
- [12] J. Costeira and T. Kanade, "A Multibody Factorization Method for Independently Moving Objects," *Int'l J. Computer Vision*, vol. 29, no. 3, pp. 159-179, 1998.
- [13] E. Elhamifar and R. Vidal, "Sparse Subspace Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 2790-2797, 2009.
- [14] G. Liu, Z. Lin, and Y. Yu, "Robust Subspace Segmentation by Low-Rank Representation," *Proc. Int'l Conf. Machine Learning*, pp. 663-670, 2010.
- [15] M. Soltanolkotabi and E. Candès, "A Geometric Analysis of Subspace Clustering with Outliers," *arXiv:1112.4258v2*, 2011.
- [16] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via Outlier Pursuit," *Systems Advances in Neural Information Processing Systems*, vol. 23, pp. 2496-2504, 2010.
- [17] M. Fazel, "Matrix Rank Minimization with Applications," PhD thesis, 2002.
- [18] B. Chen, G. Liu, Z. Huang, and S. Yan, "Multi-Task Low-Rank Affinities Pursuit for Image Segmentation," *Proc. IEEE Int'l Conf. Computer Vision*, 2011.
- [19] C. Lang, G. Liu, J. Yu, and S. Yan, "Saliency Detection by Multi-Task Sparsity Pursuit," *IEEE Trans. Image Processing*, vol. 21, no. 3, pp. 1327-1338, Mar. 2012.
- [20] Y. Eldar and M. Mishali, "Robust Recovery of Signals from a Structured Union of Subspaces," *IEEE Trans. Information Theory*, vol. 55, no. 11, pp. 5302-5316, Nov. 2009.
- [21] A. Gruber and Y. Weiss, "Multibody Factorization with Uncertainty and Missing Data Using the EM Algorithm," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 707-714, 2004.
- [22] T. Zhang, A. Szlam, and G. Lerman, "Median K-Flats for Hybrid Linear Modeling with Many Outliers," *Proc. Workshop Subspace Methods*, 2009.
- [23] A. Yang, S. Rao, and Y. Ma, "Robust Statistical Estimation and Segmentation of Multiple Subspaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.
- [24] Y. Ma, A. Yang, H. Derksen, and R. Fossum, "Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data," *SIAM Rev.*, vol. 50, no. 3, pp. 413-458, 2008.
- [25] S. Rao, A. Yang, S. Sastry, and Y. Ma, "Robust Algebraic Segmentation of Mixed Rigid-Body and Planar Motions in Two Views," *Int'l J. Computer Vision*, vol. 88, no. 3, pp. 425-446, 2010.
- [26] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.
- [27] G. Chen and G. Lerman, "Spectral Curvature Clustering (SCC)," *Int'l J. Computer Vision*, vol. 81, pp. 317-330, 2009.
- [28] G. Chen and G. Lerman, "Foundations of a Multi-Way Spectral Clustering Framework for Hybrid Linear Modeling," *Foundations of Computational Math.*, vol. 9, pp. 517-558, 2009.
- [29] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid Linear Modeling via Local Best-Fit Flats," *arXiv:1010.3460*, 2011.
- [30] E. Arias-Castro, G. Chen, and G. Lerman, "Spectral Clustering Based on Local Linear Approximations," *Electronic J. Statistics*, vol. 5, pp. 1537-1587, 2011.
- [31] F. Lauer and C. Schnörr, "Spectral Clustering of Linear Subspaces for Motion Segmentation," *Proc. IEEE Int'l Conf. Computer Vision*, 2009.
- [32] D. Donoho, "For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ_1 -Norm Solution Is Also the Sparsest Solution," *Comm. Pure and Applied Math.*, vol. 59, pp. 797-829, 2004.
- [33] B. Nasihatkon and R. Hartley, "Graph Connectivity in Sparse Subspace Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2137-2144, 2011.
- [34] G. Lerman and T. Zhang, "Robust Recovery of Multiple Subspaces by Geometric ℓ_p Minimization," *arXiv:1104.3770*, 2011.
- [35] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma, "TILT: Transform Invariant Low-Rank Textures," *Proc. Asian Conf. Computer Vision*, 2010.
- [36] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," UIUC Technical Report UILU-ENG-09-2215, 2009.
- [37] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.
- [38] J. Cai, E. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956-1982, 2010.
- [39] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A Fast Algorithm for Edge-Preserving Variational Multichannel Image Restoration," *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 569-592, 2009.

- [40] Y. Zhang, "Recent Advances in Alternating Direction Methods: Practice and Theory," tutorial, 2010.
- [41] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators," *Math. Programming*, vol. 55, pp. 293-318, 1992.
- [42] S. Wei and Z. Lin, "Analysis and Improvement of Low Rank Representation for Subspace Segmentation," *arXiv:1107.1561*, 2010.
- [43] G. Liu, H. Xu, and S. Yan, "Exact Subspace Segmentation and Outlier Detection by Low-Rank Representation," *Proc. Int'l Conf. Artificial Intelligence and Statistics*, 2012.
- [44] R. Vidal, Y. Ma, and J. Piazzi, "A New GPCA Algorithm for Clustering Subspaces by Fitting, Differentiating and Dividing Polynomials," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 510-517, 2004.
- [45] K. Huang, Y. Ma, and R. Vidal, "Minimum Effective Dimension for Mixtures of Subspaces: A Robust GPCA Algorithm and Its Applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 631-638, 2004.
- [46] R. Tron and R. Vidal, "A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
- [47] K. Lee, J. Ho, and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition Under Variable Lighting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684-698, May 2005.
- [48] F.-F. Li, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 178-188, 2004.
- [49] Y. Sugaya and K. Kanatani, "Multi-Stage Unsupervised Learning for Multi-Body Motion Segmentation," *IEICE Trans. Information Systems*, pp. 1935-1942, 2004.
- [50] A. Goh and R. Vidal, "Segmenting Motions of Different Types by Unsupervised Manifold Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [51] P. Favaro, R. Vidal, and A. Ravichandran, "A Closed Form Solution to Robust Subspace Estimation and Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011.



Guangcan Liu received the bachelor of mathematics from Shanghai Jiao Tong University (SJTU) in 2004, and the PhD degree in computer science and engineering from SJTU in 2010. Between 2006 and 2009, he was a visiting student in the Visual Computing Group, Microsoft Research Asia. From September 2010 to February 2012, he was a research fellow at the National University of Singapore. Currently, he is a postdoctoral research associate at the University of Illinois at Urbana-Champaign. His research interests include machine learning and computer vision. He is a member of the IEEE.



Zhouchen Lin received the PhD degree in applied mathematics from Peking University in 2000. Currently, he is a professor at the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a chair professor at Northeast Normal University. Before March 2012, he was a lead researcher in the Visual Computing Group, Microsoft Research Asia. He was a guest professor at

Shanghai Jiaotong University, Beijing Jiao Tong University, and Southeast University. He was also a guest researcher at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, machine learning, pattern recognition, and numerical computation and optimization. He is an associate editor of *Neurocomputing* and a senior member of the IEEE.



Shuicheng Yan is currently an assistant professor in the Department of Electrical and Computer Engineering at the National University of Singapore, and the founding lead of the Learning and Vision Research Group. His research areas include computer vision, multimedia, and machine learning, and he has authored or coauthored more than 200 technical papers over a wide range of research topics. He is an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, and has been serving as a guest editor for special issues for *TMM* and *CVIU*. He received Best Paper Awards from ACM MM '10, ICME '10, and ICIMCS '09, the winner prize of the classification task in PASCAL VOC '10, an honorable mention prize of the detection task in PASCAL VOC '10, 2010 *TCSVT* Best Associate Editor (BAE) Award, and was a coauthor of the best student papers receiving awards from PREMIA '09 and PREMIA 11. He is senior member of the IEEE.



Ju Sun received the BEng and MEng degrees, both in computer engineering, from the National University of Singapore. Currently, he is working toward the PhD degree in the Electrical Engineering Department, Columbia University, New York. His research thrust lies in deciphering the mystery of visual inference and intelligent recognition in high-level vision, and also discovering provably correct tools and mechanisms for such purposes. He is a student member of the IEEE.



Yong Yu received the master's degree from the Computer Science Department of East China Normal University. He began working at Shanghai Jiao Tong University in 1986. Now he is the vice president of the Department of Computer Science and Engineering, PhD candidate tutor, and the chairman of the E-generation technology research center (SJTU-IBM-HKU). He was the teacher of the course "Computer Graphics and Human-Machine Interface" and the course "Next Generation Web Infrastructure." As the head coach of the SJTU ACM-ICPC team, he and his team won the 2002 and 2005 ACM ICPC Championships. His research interests include semantic web, web mining, information retrieval, and computer vision.



Yi Ma received the bachelor's degrees in automation and applied mathematics from Tsinghua University, Beijing, in 1995, the MS degree in electrical engineering and computer science (EECS) in 1997, the MA degree in mathematics in 2000, and the PhD degree in EECS in 2000, all from the University of California, Berkeley. From 2000 to 2011, he was with the faculty of the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign (UIUC). Currently, he is a principal researcher at Microsoft Research Asia and holds a position of adjunct professor at UIUC. His main research interests include systems theory and computer vision. He was the recipient of the David Marr Best Paper Prize at the International Conference on Computer Vision in 1999 and Honorable Mention for the Longuet-Higgins Best Paper Award at the European Conference on Computer Vision in 2004. He received the CAREER Award from the US National Science Foundation in 2004 and the Young Investigator Program Award from the US Office of Naval Research in 2005. He is a senior member of the IEEE, the IEEE Computer Society, and the ACM.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.