IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 16, 2020, accepted July 11, 2020, date of publication July 17, 2020, date of current version July 29, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3009971*

# Nonconvex Sparse Representation With Slowly Vanishing Gradient Regularizers

**EUNWOO KIM**[1], **(Member, IEEE), MINSIK LEE**[2], **(Member, IEEE), AND SONGHWAI OH**[3], **(Member, IEEE)**
[1]School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea
[2]Division of Electrical Engineering, Hanyang University, Ansan 15588, South Korea
[3]Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, South Korea

Corresponding author: Songhwai Oh (songhwai@snu.ac.kr)

This work was supported in part by the Chung-Ang University Research Grants, in 2019, and in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01371, Development of Brain-Inspired AI with Human-Like Intelligence).

**ABSTRACT** Sparse representation has been widely used over the past decade in computer vision and signal processing to model a wide range of natural phenomena. For computational convenience and robustness against noises, the optimization problem for sparse representation is often relaxed using convex or nonconvex surrogates instead of using the $l_0$-norm, the ideal sparsity penalty function. In this paper, we pose the following question for nonconvex sparsity-promoting surrogates: What is a good sparsity surrogate for general nonconvex systems? As an answer to this question, we suggest that the difficulty of handling the $l_0$-norm does not only come from the nonconvexity but also from its gradient being zero or not well-defined. Accordingly, we propose desirable criteria to be a good nonconvex surrogate and suggest a corresponding family of surrogates. The proposed family of surrogates allows a simple regularizer, which enables efficient computation. The proposed surrogate embraces the benefits of both $l_0$- and $l_1$-norms, and most importantly, its gradient vanishes slowly, which allows stable optimization. We apply the proposed surrogate to well-known sparse representation problems and benchmark datasets to demonstrate its robustness and efficiency.

**INDEX TERMS** Sparse representation, nonconvex sparsity measure, slowly vanishing gradient.

## I. INTRODUCTION

Recently, sparse representation of signals has been one of the most successful models in many fields including signal processing, machine learning, and computer vision. Sparse representation has shown to be a powerful tool for high-dimensional data such as images [1], [2], where the goal is to represent or compress cumbersome data using a few representative samples. A simple sparse representation problem (for a noiseless scenario) can be described as follows:

$$\min_{\alpha} \ \|\alpha\|_0, \quad \text{s.t. } x = D\alpha, \tag{1}$$

where $\|\alpha\|_0 = \#\{i : \alpha_i \neq 0, \ \forall i\}$ is the $l_0$-norm, $x \in \mathbb{R}^m$ is an observation data, $D \in \mathbb{R}^{m\times p}$ is an overcomplete dictionary ($m \ll p$), and $\alpha \in \mathbb{R}^p$ is the coefficient vector to be estimated. Typical applications of sparse representation include face recognition [3], image restoration [4], and super-resolution [5], to name a few.

Behind the successful outcomes, many efforts have been made for learning sparse representation efficiently [1], [3], [6]–[16], since solving a sparse representation problem using the $l_0$-norm has two main drawbacks: (1) the computational intractability of a combinatorial search and (2) its noise sensitivity due to the nature of the $l_0$ ball. One of the most popular algorithms to estimate sparse signals is the orthogonal matching pursuit (OMP) [6], which finds the best matching projection based on an overcomplete dictionary. However, the greedy pursuit method can find a sub-optimal solution and even can fail to find a reasonable solution. Even worse, there can be a computational issue when the size of the dictionary is large.

There is little doubt that the recent popularity of the sparse representation is attributed to the attempt that the $l_0$-norm is relaxed to its convex counterpart, i.e., the $l_1$-norm [17]. In many cases, the use of the $l_1$-norm turns the problem into convex optimization, which can be efficiently solved with theoretical guarantees. Especially, some analyses have shown that the $l_1$-norm-based problems can exactly recover the best

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Remagnino.

sparse solution under certain conditions [2], [18], making a strong justification for the use of the $l_1$-norm. Accordingly, the $l_1$-norm has been extensively utilized in many problems under different forms, and many efficient methods, including the basis pursuit denoising (BPDN) methods, such as FISTA [19], have been proposed to solve $l_1$-norm minimization problems.

Obviously, the $l_1$-norm relaxation is beneficial when the relaxed problem or system indeed becomes convex. However, some problems are inherently nonconvex and, for those problems, replacing the sparsity surrogate to a convex one does not necessarily make the overall problem convex. Some well-known examples of such problems are: matrix factorization [20] and rank-constrained subspace learning [21]–[23]. For these problems, using the $l_1$-norm will not bear as much significance as the previous examples. In fact, for general problems aside from some special (convex) cases mentioned above, the constant slope of the $l_1$-norm, which is also known as a biased penalty function[1] [8], can over-penalize the values of nonzero elements unlike the $l_0$-norm and make the solution deviate from the desired solution [8], [9], [12], [13]. This constant slope is the one that makes the $l_1$-norm a convex surrogate, which is not really necessary for the nonconvex settings discussed here. Note that there is a tighter convex approximation to the $l_0$-norm [24], but it also has a constant gradient along each direction.

Motivated by the fact that the $l_1$-norm may not be the best choice for nonconvex problems, we ask the following question: What is a good nonconvex sparsity surrogate if it is not possible to transform the problem to a convex one? It is quick to notice that the $l_0$-norm is still difficult to handle in nonconvex problems. It is worth mentioning that the nonconvex nature is not the only difficulty of the $l_0$-norm, but *its gradient either being zero (for the most parts) or not being well-defined is the true culprit*, which necessitates a combinatorial optimization approach. If we can find an approximation of the $l_0$-norm that has a sufficiently large region with nontrivial gradients, it would be much easier to apply conventional optimization techniques.

As prior works, there have been attempts to use nonconvex smooth (or possibly nonsmooth) approximations of the $l_0$-norm [7]–[10], [12], [25], [26]. We will discuss the theoretical relevance and difference of the proposed surrogate compared to other nonconvex alternatives in Section III-B.

### A. PAPER CONTRIBUTIONS

In this paper, we analyze possible approximations of the $l_0$-norm. We first propose desirable criteria to be a good nonconvex surrogate and present a representative family of curves, termed *slowly vanishing gradient* (SVG) surrogates, that is a solution of a differential equation. The proposed surrogate avoids existing issues of rapidly vanishing gradient of other well-known surrogates which can hinder the optimization progress. We also show that there is a trade-off between the values and the vanishing speed of their gradients. Locally, the surrogate follows the $l_1$-norm to reduce the chance of numerous local optima without losing the ability of promoting sparsity. Globally, it follows the $l_0$-norm to reduce penalty on large-values, but it still possesses slowly vanishing gradients to help drawing the solution of an optimization algorithm to sparse points. Moreover, it has an efficient proximity operator for the surrogate. The proposed surrogate function is applied to various applications such as low-rank approximation (LRA), sparse coding with dictionary learning (SC), and sparse subspace clustering (SSC) problems, to demonstrate its adequacy and experimental results confirm that our proposal performs favorably against those of other well-known sparsity surrogates.

### B. NOTATIONS

An observation matrix is denoted by $X \in \mathbb{R}^{m\times n}$, where each column corresponds to a data sample in $\mathbb{R}^m$. We denote matrices, vectors, and scalars by bold letters in upper case, bold letters in lower case, and letters in lower case, respectively, unless stated otherwise. Spaces and subspaces are denoted by bold italic letters in upper case. Throughout this paper, we use $\|A\|_q$ to denote matrix norms of a matrix $A$, with $q = 1$ for the matrix $l_1$-norm, $\sum_{ij} |a_{ij}|$, and $q = F$ for the Frobenius-norm, $\sqrt{\sum_{ij} |a_{ij}|^2}$. We denote the projection operator by $\mathcal{P}(\cdot)$ and the support set of a matrix $A$ by $\Omega_A$. rank($A$) denotes the rank of $A$ and $|\cdot|$ denotes the absolute value operation of a scalar. Diagonal elements in a matrix $A$ is denoted by diag($A$).

## II. ANALYSIS ON THE $l_0$-NORM APPROXIMATION

### A. DESIRABLE CRITERIA FOR A NONCONVEX SURROGATE

In this section, we will mainly discuss a sparse representation problem whose cost function consists of a data term and a regularizer. As explained earlier, if the problem itself (data term) has a nonconvex structure, then the convexity of the sparsity surrogate (regularizer) is not absolutely necessary. In this case, the constant slope of the $l_1$-norm will not necessarily make the problem convex but over-penalize nonzero values in the input, which makes the solution deviate from the desired solution, especially when the problem assumes the presence of noises. Hence, we might be interested in finding a good nonconvex surrogate for such general nonconvex problems. Prior works support the superiority of nonconvex sparsity-promoting surrogates [9], [12], [26]–[31].

If the nonconvexity of the $l_0$-norm is not a problem, then the only difficulty in handling it is that its value only changes around zero (or we can imagine that its shape appears as if it gives an extremely local gradient at the origin). This is highly undesirable from the perspective of conventional optimization. Since the derivative of the $l_0$-norm is zero for nonzero inputs and undefined at zero, there can be undesirable effects for finding sparse solutions and discovering a good local optimal solution.

[1] Throughout this paper, we use the term *penalty functions* and *surrogates* interchangeably.

In order to find a surrogate which has least undesirable effects and can also be handled efficiently, we might consider smooth approximations of the $l_0$-norm [9], [10], [13]. However, there can be infinitely many such approximations and we need some criteria for finding a good surrogate. Below are basic assumptions to be a good candidate:

*Assumption 1:* We pose the following criteria on the surrogate[2] $\phi(x)$ (defined on $-\infty < x < \infty$) we are looking for:

1) *Symmetry*: The sign of an input does not matter but the magnitude, hence, we assume $\phi(x) = \phi(-x)$.
2) *Continuity at* $x = 0$: In order to avoid a jump at $x = 0$, we assume $\phi'(0^+) = \phi'(0^-) = \phi(0)$.
3) *Asymptotic convergence*: Assume $\phi(0) = 0$. Then, $\phi(x)$ satisfies $\lim_{x\to\infty} \phi(x) = 1$. This prevents $\phi(x)$ from penalizing large nonzero inputs equally as small ones, and makes it closer to the $l_0$-norm.
4) *Monotonicity*: In order for $\phi$ to be a valid surrogate, we assume $\phi'(x) > 0$ for $x > 0$ where $\phi'(x)$ is the derivative of $\phi(x)$ at $x$, i.e., $\phi$ is a monotonically increasing function on $x > 0$.
5) *Smoothness (Monotonicity of gradient)*: Assume $\phi$ is twice continuously differentiable when $x \neq 0$. There can be some choices of $\phi$ that $\phi'(x)$ goes up and down, but this behavior is unnecessary and will overcomplicate $\phi(x)$. Hence, we assume $\phi''(x) < 0$ for $x > 0$, i.e., the gradient decreases monotonically for $x > 0$.
6) *Finite nonzero derivative around* $x = 0$: Let us define $\phi'(0^+) = \lim_{x\to 0^+} \phi'(x)$. Then, $\phi'(0^+)$ should be a finite nonzero value to promote sparsity, i.e., $0 < \phi'(0^+) = b < \infty$. In many examples, $b$ will be chosen as $b = 1$ for ease of explanation.

*Remark 1:* We give more details for the last criterion. Note that a local optimum of a cost function, consisting of a data term and a continuous and symmetric regularizer term, exists at a sparse point when the sum of both (generalized) subgradients [32] of the two terms becomes 0. First, $\phi'(0^+)$ should be nonzero to promote sparsity. If $\phi'(0) = 0$, due to the symmetry assumption, then the derivatives of a cost function depend only on the data term. However, if $\phi'(0^+)$ is nondifferential, the chance of an optimal point can increase since a subgradient of the regularizer may make one of the subgradients of the entire cost function zero. In short, a nondifferentiable regularizer increases the chance of zero subgradient of a cost function at sparse points more than a differentiable regularizer. Second, $\phi'(0^+)$ should be finite to avoid a sub-differential containing infinite large number of subgradients ($-\infty \leq \phi'(0^+) \leq \infty$), since this sub-differential always makes the sub-differential of the entire cost function contain zero, thus it can create too many local optima at sparse points. In other words, we would like to avoid a sparse point becoming a local optimum if the data term at the point has a steep slope.

Aside from the above criteria, we have another criterion on the choice of $\phi$. As discussed before, the gradient either being 0 or not being well-defined is what makes the optimization difficult for the $l_0$-norm. Thus, we aim to find a surrogate that has an opposite characteristic: $\phi(x)$ whose gradient is as large as possible across the entire interval. Because of the fifth criterion, this is equivalent to finding $\phi(x)$ that has *slowly vanishing gradients*. If $\phi'(x)$ decreases slowly, then the effect of the sparsity surrogate can spread across a large region to help drawing the solution to sparse points. This can be viewed as mimicking the constant slope of the $l_1$-norm under the above criteria. Hence, we might try to find $\phi(x)$ with the most slowly decreasing gradient. However, due to the third criterion, the "total amount" of gradient is finite, i.e.,

$$\int_{0^+}^{\infty} \phi'(x)dx = 1. \tag{2}$$

This means that we have to divide this finite value for $0 < x < \infty$.

### B. A REPRESENTATIVE FAMILY OF SURROGATES

To analyze the situation discussed above more closely, we present two extreme examples among the possible family of surrogates that satisfy the above criteria. Because of the first criterion, we can assume $\phi(x) = y(|x|)$ for some function $y$ on $\mathbb{R}^+$.

First, let us see an example that is a smooth relaxation of the $l_0$-norm, but its gradient vanishes exponentially fast in a relatively local region. An easy example is

$$y = 1 - e^{-x}, \tag{3}$$

which satisfies $y(0) = 0, y(\infty) = 1$, $y'(0^+) = 1$, and all the above criteria. Its derivative is $y'(x) = e^{-x}$, which means that the gradient vanishes exponentially. Hence, this surrogate will quickly become negligible except the local region near $x = 0$.

As an opposite example, let us consider a case, in which the gradient vanishes very slowly;

$$y = 1 - \frac{1}{(1 + \frac{x}{a})^a}, \tag{4}$$

with very small $a > 0$. Its derivative is

$$y'(x) = \frac{1}{(1 + \frac{x}{a})^{1+a}}, \tag{5}$$

and this also satisfies $y(0) = 0, y(\infty) = 1$, $y'(0^+) = 1$, and all of the above criteria. Here, since $a$ is very small, $y'(x)$ is close to a reciprocal function $\frac{1}{1+\frac{x}{a}}$. Integrating $\frac{1}{1+\frac{x}{a}}$ for $0 \leq x < \infty$ does not converge, hence, this can be seen as an extreme example with very slowly vanishing gradients. However, $\frac{1}{(1+\frac{x}{a})^{1+a}}$ is very close to 0 for most of $x$, which is a natural consequence of spreading a finite value

[2] For ease of explanation, we sometimes deal with a scalar function throughout the paper due to the separability of the surrogate, even though this paper is about the sparsity-promoting penalty. An extension to a vector case is straightforward.

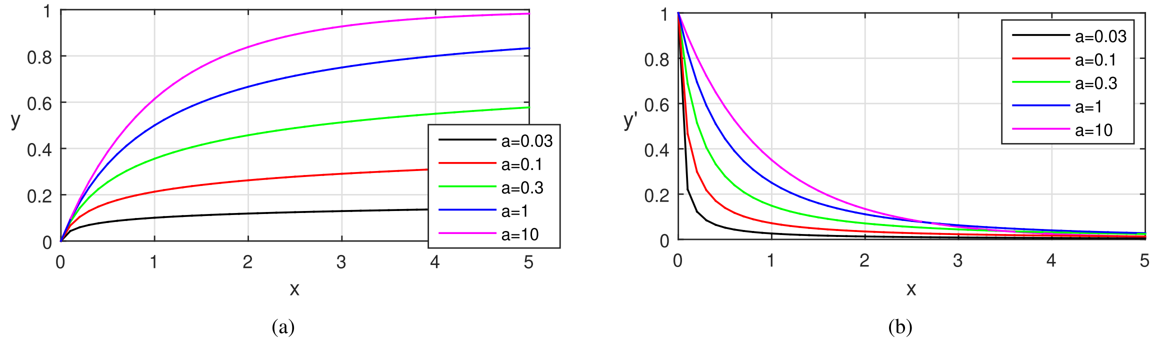


**FIGURE 1.** Graphical illustration of a family of representative curves (a) $y$ and (b) their derivatives $y'$ for different choices of $a$.

($\int_0^\infty y'(x)dx = 1$) to a broad interval. Indeed, we can verify that

$$\lim_{a\to 0} \frac{1}{(1+\frac{x}{a})^{1+a}} = 0 \text{ if } x \neq 0 \tag{6}$$

and the function itself approaches to zero, i.e.,

$$\lim_{a\to 0} 1 - \frac{1}{(1+\frac{x}{a})^{a}} = 0. \tag{7}$$

Note that the previous example can be viewed as an opposite extreme in this sense as $\lim_{a\to\infty} \frac{1}{(1+\frac{x}{a})^{1+a}} = e^{-x}$. Therefore, there is a tradeoff between the spread (vanishing speed) of gradients and their actual values. Some example curves of $y$ and $y'$ for various values of $a$ are illustrated in Figure 1.

In addition to the extreme examples, there are infinitely many functions that satisfy our criteria. However, the details of curve shapes do not matter much because local differences between two curves does not bear a significant meaning for general problems. Hence, it suffices to choose a representative family of curves that has a nice interpretation and includes various rates of gradient vanishment, in order to narrow down our choices. In fact, the previous examples are good candidates, since they are solutions to the following differential equation that has an elegant meaning:

$$(1-y)^{1+\frac{1}{a}} = \epsilon y', \quad y(0) = 0, \tag{8}$$

where $a > 0$ and $\epsilon > 0$ are parameters. It is worth noting that $(1-y)$ on the left side is the difference between the $l_0$-norm and $y$, thus, the decreasing speed of $(1-y)$ is identical to the rate of asymptotic convergence (criterion 3). Therefore, this equation describes the rate of gradient vanishment in terms of the rate of asymptotic convergence. This can be transformed into a Bernoulli equation, and the solution is given as

$$y(x) = 1 - \frac{1}{(1+\frac{x}{a\epsilon})^{a}}, \tag{9}$$

which satisfies $y'(0^+) = \frac{1}{\epsilon}$, $y(0) = 0$, and $y(\infty) = 1$ for $a > 0$. We call the corresponding penalty functions satisfying (9) as a family of *slowly vanishing gradient* (SVG) surrogates. As a special case of the family of SVG surrogates when $\epsilon = 1$ and $a \to \infty$, the solution leads to (3).

## III. PROPOSED NONCONVEX SPARSITY SURROGATE

### A. CHOOSING A SIMPLE ONE AMONG THE SVG FAMILY

As explained in the previous section, there is a tradeoff between the vanishing speed and the actual value of the gradient. Thus, we can, at best, choose a good compromise between them. Since there is no clear winner between the curves in our SVG family, it is better to choose the simplest one among the reasonable choices. Accordingly, we constrained $a$ to be an integer, and find one that gives the slowest decreasing rate of gradient, which is $a = 1$. As a result, we have $y(x) = 1 - \frac{\epsilon}{x+\epsilon} = \frac{x}{x+\epsilon}$. Based on this function, our proposed sparsity surrogate[3] is given as follows:

$$\|\boldsymbol{\alpha}\|_{\text{SVG}}^{\epsilon} = \sum_i \frac{|\alpha_i|}{|\alpha_i| + \epsilon}, \tag{10}$$

where $\epsilon > 0$ is a weighting parameter that determines the slope at $\alpha_i = 0^+$. The following proposition shows the pointwise convergence of the proposed surrogate to the conventional norms.

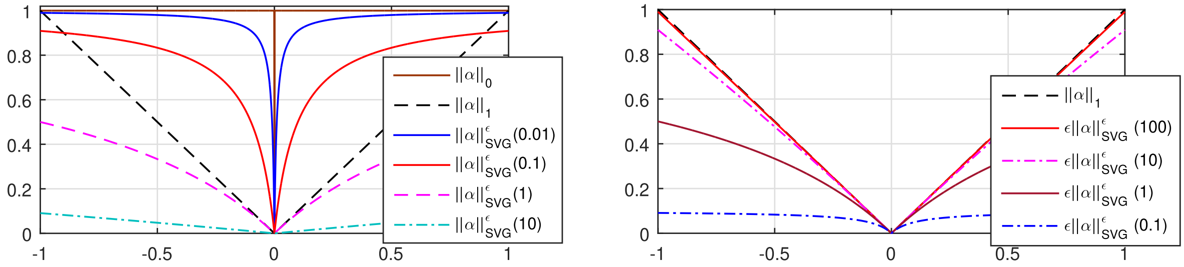
*Proposition 1:* SVG approximates the $l_0$- and $l_1$-norms:

1) $\|\boldsymbol{\alpha}\|_{\text{SVG}}^{\epsilon} \le \|\boldsymbol{\alpha}\|_0 \ \forall\epsilon$ and $\|\boldsymbol{\alpha}\|_{\text{SVG}}^{\epsilon} \to \|\boldsymbol{\alpha}\|_0$ if $\epsilon \to 0$.
2) $\epsilon\|\boldsymbol{\alpha}\|_{\text{SVG}}^{\epsilon} \le \|\boldsymbol{\alpha}\|_1 \ \forall\epsilon$ and $\epsilon\|\boldsymbol{\alpha}\|_{\text{SVG}}^{\epsilon} \to \|\boldsymbol{\alpha}\|_1$ if $\epsilon \to \infty$.

Note that the above properties still hold for the proposed SVG family based on (9). The proof is included in Appendix A. Some example curves of SVG are illustrated in Figure 2 to visualize these properties.

Another nice property of SVG is that it possesses a simple proximity operator. Recently, there have been remarkable theoretical progresses on convergence analysis for the sparse optimization techniques, and nonconvex versions for the accelerated proximal gradient method (nAPG) [33] and the alternating directional method of multipliers (nADMM) [34] have been proposed to solve sparse optimization problems efficiently in nonconvex settings. Hence, even though SVG is nonconvex, having a simple proximity operator is still a good advantage to incorporate the above methods for efficient nonconvex programming.

[3] We just denote the surrogate as SVG in that it is one of our SVG family.



(a)

(b)

**FIGURE 2. Graphical illustration of SVG of a vector $\alpha$ with respect to various values of $\epsilon$ (a) compared to the $l_0$-norm, and (b) to the $l_1$-norm. (·) denotes the value of $\epsilon$.**

The proximity operator for SVG is defined as follows:

$$\mathrm{prox}^{\epsilon}_{\mathrm{SVG},\lambda}(\boldsymbol{x}) = \min_{\boldsymbol{u}} \lambda\|\boldsymbol{u}\|^{\epsilon}_{\mathrm{SVG}} + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|^2. \qquad (11)$$

Note that this equation is separable, and we can solve it for each element of $\boldsymbol{u}$. Since SVG is a symmetric function for each element, an element of the solution vector $\hat{\boldsymbol{u}}$ will either be of the same sign with the corresponding element of $\boldsymbol{x}$ or be zero. Let us assume that the sign of $x_i$, the $i$th element of $\boldsymbol{x}$, is positive without loss of generality. Then, one of the positive solutions of the following cubic equation

$$\begin{aligned}(u_i+\epsilon)^2\left(\frac{\lambda u_i}{u_i+\epsilon}+\frac{1}{2}(x_i-u_i)^2\right)'\\ = \lambda\epsilon + (u_i-x_i)(u_i+\epsilon)^2 \triangleq g(u_i) = 0 \qquad (12)\end{aligned}$$

or zero will be the optimal point of $u_i$. Note that the coefficient of the third-order term of $g(u_i)$ is positive, as well as the value of $g(-\epsilon) = \lambda\epsilon > 0$. This indicates that $g(u_i)$ has at least one root for $u_i < 0$, i.e., there can be at most two roots for $u_i \geq 0$. If there is no root or a double root for $u_i \geq 0$, $g(u_i)$ is nonnegative for $u_i \geq 0$, i.e., the cost function is monotonically increasing for positive $u_i$, and the optimal point will be 0. If there are two distinct roots, then the solution with a larger value is a local minimum, so either this solution or zero will be the optimal point. In conclusion, the optimal $\hat{u}_i$ is either the largest positive root of (12) or zero, and we can compare the costs of these two points to find the final solution. This analysis will relieve the computational complexity when solving the third-order equation.

### B. RELATIONSHIP WITH OTHER SPARSITY SURROGATES

There are many nonconvex sparsity surrogates, such as smoothly clipped absolute deviation (SCAD) [8], minimax concave penalty (MCP) [12], and Capped-$l_1$ (CapL1) penalty [26], which have been proposed to approximate the $l_0$-norm. A comprehensive study on the nonconvex sparsity surrogate can be found in [27], [35]. In [8], authors advocate a nonconvex surrogate that has three desired properties: unbiasedness, sparsity, and continuity. More general properties to be a good nonconvex surrogate are described in [35] (see Assumption 1). Note that the proposed family of surrogates satisfies the properties and further extends them by introducing an important new criterion; slowly vanishing gradients. In addition, our surrogate can provide theoretical guarantees as shown in Section III-C. The above mentioned surrogates do not satisfy the criterion of slowly vanishing gradients, since they have large *flat regions* (gradient zero or quickly converging gradient). This may increase the chance of local optima if some local optima of a loss function (data term) are located at the plateau of the surrogate functions (regularizers). Our aim is to mitigate this effect.

Unlike the above surrogates, there is another line of surrogates [7], [9], [10] as an alternative to the $l_0$-norm, which gives a constantly inclinatory curve analogous to the proposed surrogate. A typical example is the $l_q$-norm ($0 < q < 1$) [7]. However, there is no analysis about the $l_q$-norm analogous to ours. Even worse, the $l_q$-norm is known to be difficult to solve due to the $q$-th power. Whereas, ours enjoys a simple proximity operator and handles the raised issues efficiently. Analogous to the $l_q$-norm penalty, the log-sum penalty (LSP) [9] gives a non-flat curve, but it does not give the satisfying performance compared to the proposed penalty as shown in Section IV-A. There has been another attempt to use a smooth approximation of the $l_0$-norm (SL0) based on an exponential function in [10], but no analysis was provided for justifying such a choice. Furthermore, our analysis shows that the approximation based on an exponential function has fast vanishing gradients, which is more prone to local optima, and thus this approximation does not give satisfactory performance as shown in Section IV-D.

While preparing this manuscript, we became aware of that our proposal, as a special case ($a = 1$) of the SVG family, leads to the same surrogate proposed by Geman and Yang [25] over two decades ago. However, it is important to note that there are clear differences between their and our studies. First, the specific choice for approximating the $l_0$-norm is not justified in [25] because its focus is an image reconstruction problem. Second, we provide detailed analyses and design motivation for the SVG surrogate. Furthermore, the optimization approach in [25] is outdated, while we provide efficient algorithms based on a proximity operator for the proposed surrogate. Lastly, we show comprehensive experimental results compared to existing nonconvex surrogates using many well-known examples in the literature.

Overall, our motivation and analysis give a new insight from the optimization perspective for nonconvex sparsity surrogates and the proposed one provides superior performance compared to the existing surrogates of the $l_0$-norm as described in Section IV.

### C. CONNECTION TO EXISTING THEORY

In this section, we provide an analysis about connection to existing theory. To this end, we first describe the following well-studied assumption:

*Assumption 2 ( [35]):* We consider a scalar variable $x$ for simplicity and define a regularizer as $\phi_\lambda : \mathbb{R} \to \mathbb{R}$.

1) The function $\phi_\lambda$ satisfies $\phi_\lambda(0) = 0$ and is symmetric around zero (i.e., $\phi_\lambda(x) = \phi_\lambda(-x)$ for all $x \in \mathbb{R}$).
2) On the nonnegative real line, $\phi_\lambda$ is nondecreasing.
3) For $x > 0$, the function $x \mapsto \frac{\phi_\lambda(x)}{x}$ is nonincreasing.
4) A surrogate function $\phi_\lambda$ is differentiable for all $x \neq 0$ and subdifferentiable at $x = 0$, with $\lim_{x\to 0^+} \phi'_\lambda(x) = \lambda L$.
5) There exist $\mu > 0$ such that $\rho_{\lambda,\mu}(x) \triangleq \phi_\lambda(x) + \frac{\mu}{2}x^2$ is convex.

Now, we show that our representative family of surrogates satisfying the criteria in Assumption 1 meets Assumption 2.

*Proposition 2:* The representative family of surrogates $\phi_\lambda$ designed by our criteria with the parameters $\epsilon$ and $a$ satisfies the conditions of Assumption 2 with $L = \frac{1}{\epsilon}$ and $\mu = \frac{(a+1)\lambda}{a\epsilon^2}$.

*Corollary 1:* The proposed SVG surrogate given in (10) satisfies the conditions of Assumption 2 with $L = \frac{1}{\epsilon}$ and $\mu = \frac{2\lambda}{\epsilon^2}$.

The proof of Proposition 2 is included in Appendix B.

### D. LEARNING SPARSE REPRESENTATION WITH SVG

The proposed surrogate can be applied to various sparse representation problems that the $l_0$-norm and $l_1$-norm are applied. In this section, we focus on three important problems including low-rank approximation (LRA) [20], sparse coding (SC) [1], and sparse subspace clustering (SSC) [36].

#### 1) SVG FOR MODELING SPARSE ERRORS IN LRA

Sparse representation has been widely used in many applications to filter out outliers in data. One of the most popular applications is the low-rank approximation (LRA) of a matrix under the existence of outliers, and the $l_1$-norm is usually used to model the sparse outliers [2], [14], [37]. We consider an LRA problem that the rank is explicitly specified, such as structure reconstruction [38] and photometric stereo [18], to name a few. In this case, it becomes a nonconvex problem. For the problem, we apply SVG for modeling sparse errors denoted by $\boldsymbol{E}$, whose problem formulation (LRA-SVG) is constructed as

$$\min_{\boldsymbol{E},\boldsymbol{M}} \|\mathcal{P}_{\Omega_X}(\boldsymbol{E})\|_{\text{SVG}}^{\epsilon}, \quad \text{s.t.}\ \boldsymbol{E}=\boldsymbol{X}-\boldsymbol{M},\ \text{rank}(\boldsymbol{M})\leq r. \tag{13}$$

This problem can be efficiently solved using the nADMM framework [34] as discussed before. The derivation of LRA-SVG is included in Appendix C.

#### 2) SVG FOR SPARSE CODING

The proposed surrogate can be applied to another well-known nonconvex sparse representation problem, sparse coding (SC) with dictionary learning [1], [39], which is basically a matrix factorization problem. Here, SVG is used to enforce the sparsity of the encodings in this case. The problem formulation of SC for observation vectors $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$, where $n$ is the number of samples, based on SVG (SC-SVG) can be given as follows:

$$\min_{\boldsymbol{D},\boldsymbol{\alpha_1},\ldots,\boldsymbol{\alpha_n}} \frac{1}{2}\sum_i^n \|\boldsymbol{x_i} - \boldsymbol{D}\boldsymbol{\alpha_i}\|_2^2 + \lambda\|\boldsymbol{\alpha_i}\|_{\text{SVG}}^{\epsilon}, \tag{14}$$

where $\boldsymbol{D}$ and $\boldsymbol{\alpha_i}$ are an overcomplete dictionary and the $i$th sparse coefficient vector corresponding to $\boldsymbol{x_i}$, respectively. This problem is solved in an alternating fashion based on the proximal gradient method.

#### 3) SVG FOR SSC

Subspace clustering is a problem to find the cluster memberships of data points based on an assumption that a point can be represented by a linear combination of other points in the same cluster. Note that this problem can be efficiently solved based on convex optimization, nevertheless we apply SVG to this problem, in order to verify the capability of the proposed surrogate in general problems. We apply SVG to the well-known sparse subspace clustering (SSC) [36], where the corresponding formulation (SSC-SVG) under noisy scenario is given as follows:

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_F^2 + \lambda\|\boldsymbol{Z}\|_{\text{SVG}}^{\epsilon}, \quad \text{s.t. diag}(\boldsymbol{Z}) = 0, \tag{15}$$

where $\boldsymbol{Z}$ is an affinity matrix to reveal cluster membership. This problem can be efficiently solved by nAPG under the nonmonotone update framework [33]. Especially, we incorporate the nonmonotone update framework [33] to accelerate the convergence of the algorithm.

Note that initial values of optimization variables for the proposed algorithm are set to zero, based on empirical observations that the algorithm is not sensitive to initial values.

## IV. EXPERIMENTAL RESULTS

In this section, we report numerical results of the sparse representation algorithms based on SVG: LRA-SVG, SC-SVG, and SSC-SVG. We compare these algorithms with other state-of-the-art algorithms[4]: RPCA [18], ALADM [37], and LRA-L1 (an $l_1$-norm version of LRA-SVG) for low-rank approximation problems, KSVD [1] and SC [39] for sparse coding problems, and LRR [40], SSC-BP [36], SSC-OMP [41], and SSC-SL0 (SSC based on smoothed $l_0$-norm [10]) for subspace clustering tasks, face clustering and motion segmentation. We also compare with other well-known nonconvex sparsity surrogates, SCAD [8], MCP [12], CapL1 [26], and LSP [9], in order to demonstrate the superiority of the proposed nonconvex surrogate for problems

[4] In order to compare the proposed method with various algorithms, we report results also for convex algorithms based on the $l_1$-norm.

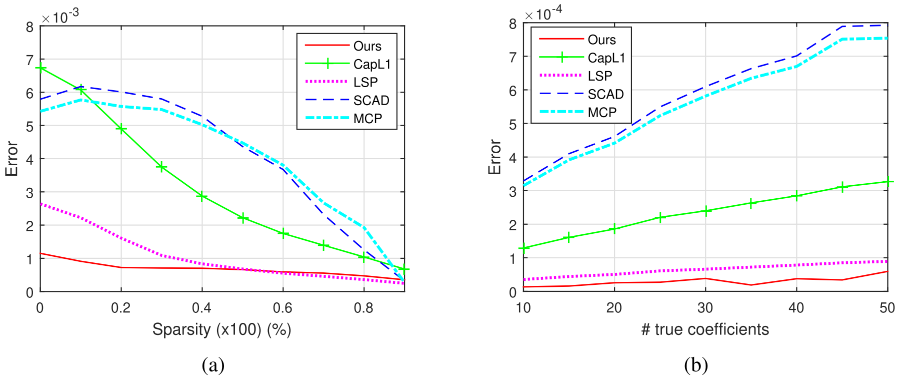


**FIGURE 3. Average performances on synthetic examples. (a) Reconstruction errors w.r.t. the sparsity. (b) Coefficient errors w.r.t. different sparsity numbers for nonnegative sparse coding.**

described above. For the compared algorithms, we used the codes provided by the authors, unless stated otherwise, and each algorithm's parameters are tuned to yield the best performance for each dataset. For low-rank approximation and sparse coding problems, we compute the reconstruction error as

$$\frac{\|\boldsymbol{W} \odot (\boldsymbol{M}^{GT} - \boldsymbol{M})\|_1}{\|\boldsymbol{W}\|_1}, \tag{16}$$

where $\boldsymbol{M}^{GT}$ and $\boldsymbol{M}$ are the ground-truth and reconstructed matrices or vectors, respectively, $\boldsymbol{W}$ is a weight matrix concerning missing entries, and $\odot$ is the Hadamard product operator. For subspace clustering, we compute the accuracy by the Hungarian method [42],

$$\frac{1}{n}\sum_{i=1}^{n} \delta(\boldsymbol{p}_i, map(\boldsymbol{q}_i)), \tag{17}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ are the $i$-th ground-truth and obtained cluster labels, respectively, $\delta(a, b)$ is the Kronecker delta function, and $map(\cdot)$ is a mapping function to permute the obtained labels to match with the ground-truth labels, which is computed by the Kuhn-Munkres algorithm [42]. We set the parameter $\epsilon$ of SVG to 3 for all experiments except the motion segmentation problem where $\epsilon$ is set to 0.07. We also provide settings for $\lambda$ in the following experiments. All experiments were performed using MATLAB environment on a desktop computer with 24GB RAM and a 3.4GHz quad-core CPU.

### A. EVALUATION FOR NONCONVEX SPARSITY SURROGATES

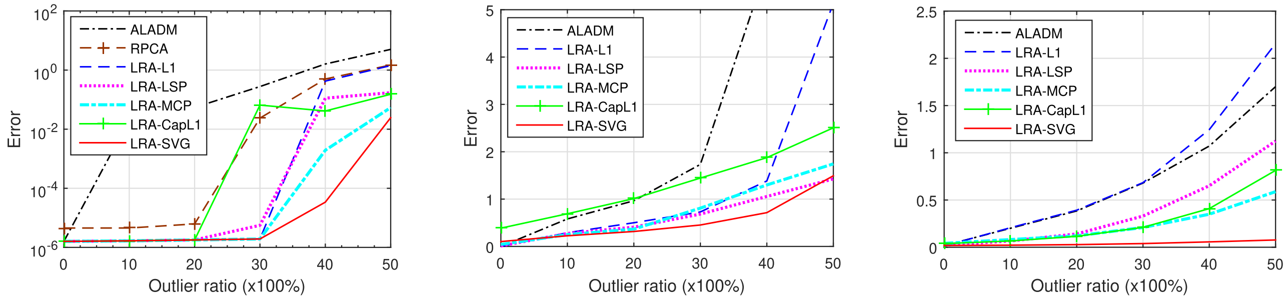
We first evaluate SVG on synthetic examples to compare with other nonconvex sparsity surrogates. We used the codes of other compared penalties provided by the work in [13], which solves the nonconvex optimization problems efficiently with a convergence guarantee. Following the practice in [13], we construct a sparse coding problem to find a sparse vector $\boldsymbol{\alpha}$: $\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \phi(\boldsymbol{\alpha})$, where $\boldsymbol{x} \in \mathbb{R}^m$ is a target vector, $\boldsymbol{D} \in \mathbb{R}^{m\times p}$ is a data/dictionary matrix, and $\phi(\boldsymbol{\alpha})$ is a penalty function. For all experiments in this subsection, we set $m = p = 500$. We made a scenario by varying sparsity $(0 \sim 90\%)$ of a ground-truth vector $\boldsymbol{\alpha}^{GT}$, where lower sparsity means denser representation, and made an observation $\boldsymbol{x}^{GT}$ from the multiplication of $\boldsymbol{D}$ and $\boldsymbol{\alpha}^{GT}$, which are obtained by the standard normal distribution. Based on $\boldsymbol{x}^{GT}$, we made $\boldsymbol{x}$ by adding Gaussian noises from $\mathcal{N}(0, 10^{-2})$. For each setting in the scenario, we performed $k$ independent runs, where $k$ is set to 30. We set $\lambda$ to 0.3. The average reconstruction error is computed as $\frac{1}{k}\sum_{i=1}^{k}\|\boldsymbol{x}_i^{GT} - \boldsymbol{D}_i\boldsymbol{\alpha}_i\|_2$, where $\boldsymbol{x}_i^{GT}$ is the ground-truth vector for the $i$-th scenario.

Average results of the compared surrogates are shown in Figure 3. As shown in Figure 3(a), the proposed surrogate performs better than the other nonconvex surrogates on average. LSP, which represents a similar non-flat curve, gives the similar performance to ours when the sparsity ratio is larger than 40%. SCAD and MCP show the similar but worst performances in this problem. The average computation times (sec) of the surrogates for the reconstruction problem are as follows: 0.15 for CapL1, 0.28 for SCAD, 0.26 for MCP, 0.23 for LSP, and 0.3 for SVG, respectively. In the problem, most of the methods take similar execution times.

We also applied the surrogates to nonnegative sparse coding problems under the same setting as the previous example, except that all coefficients are nonnegative. Figure 3(b) shows the $l_2$ errors between the true coefficient vector and obtained vectors under different numbers of true coefficients in $\boldsymbol{\alpha}_i$. Overall, the results show the similar tendency to the previous experiment, in which the proposed surrogate finds all sparse coefficients with lowest errors.

### B. LOW-RANK APPROXIMATION OF MATRICES

We report the results for low-rank approximation problems using both synthetic and real-world problems. To generate synthetic examples, we made a matrix whose size is $500 \times 500$ and set the rank of the matrix to 10. In the matrix, we added Gaussian noises with $\mathcal{N}(0, 10^{-5})$ and outliers with magnitude of 10 for randomly chosen elements.



(a) Synthetic dataset (b) Shark sequence (c) Static Face dataset

**FIGURE 4. Average performances on low-rank approximation problems in the presence of outliers and missing data.**

The outlier ratio is varied from 0% to 50% to verify the robustness of the proposed method. Here, we compare with MCP, CapL1, and LSP under the same LRA framework to ours. The experimental results for 50 independent trials are described in Figure 4(a). From the figure, we observe that the proposed method withstands higher outlier ratios, whereas other methods fail to find a good solution when the outlier ratio becomes roughly over 30%. The three nonconvex penalty based algorithms perform better than the methods based on the convex $l_1$-norm, on average, but they could not endure as many outliers as ours. Interestingly, the algorithms based on nonconvex penalties bear a lot of outliers relatively compared to the convex penalty based algorithms particularly when the outlier ratio is over 50%. The average computation times (sec) of the algorithms are 0.62 for ALADM, 11.74 for RPCA, 1.76 for LRA-L1, 50.24 for LRA-LSP, 13.77 for LRA-MCP, 13.8 for LRA-CapL1, and 3.16 for LRA-SVG, respectively.

We performed real-world experiments on two problems; nonrigid motion estimation [43] and photometric stereo [38]. For nonrigid motion estimation, we used the Shark sequence (rank 6) [43]. To consider missing environments, we replaced 10% randomly selected entries in the sequence as missing. For photometric stereo, we used Static Face dataset (rank 4) [38] which has 42% missing entries. For these problems, we did not evaluate RPCA because they are rank-constrained problems. Figure 4(b) and 4(c) show the average reconstruction errors of the algorithms for 50 independent runs under various outlier ratios ($0 \sim 50\%$). From the figure, we confirm that the proposed method outperforms the other methods for both problems. Especially, the proposed method is highly robust against corruptions for the Static Face dataset. Most of the nonconvex penalty based algorithms show lower reconstruction error than the $l_1$-norm based algorithms. While LRA-LSP gives competitive results to LRA-SVG for the Shark sequence, it performs poorer than ours for Static Face. The $l_1$-norm approaches, LRA-L1 and ALADM, perform worse than other nonconvex surrogate based algorithms on average for both datasets. The reconstruction results of the three selected algorithms, the proposed method, LRA-LSP, and LRA-L1, for three randomly selected frames of the Shark sequence in the presence of 30% outliers are shown in Figure 5.

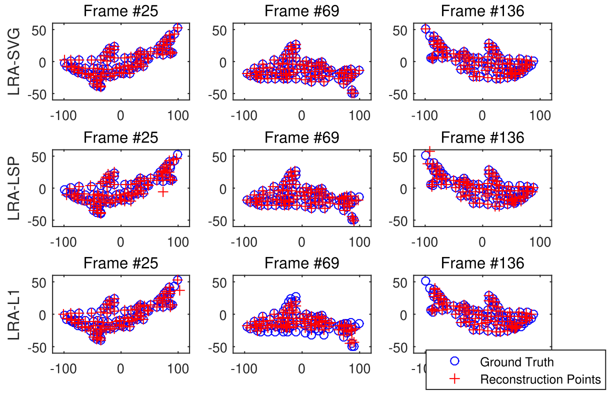


**FIGURE 5. Reconstruction results from the shark sequence by three methods. '○' means the ground truth and '+' means the reconstruction point.**

**FIGURE 6. Test images for the sparse coding problem. From left to right: Barbara, Lena, Boat, and Pappers.**

## C. SPARSE CODING

We conducted experiments for a sparse coding problem (14) based on well-known example images in the literature: Barbara, Lena, Boat, and Peppers, which are shown in Figure 6. Following the practice of [1], we extracted $n$ 64-dimensional word vectors based on $8 \times 8$ local patches for each image, where $n$ is the number of training data which was set to $n = 15,000$. Based on these word vectors, we learned both dictionary and sparse code for each sample. For all tested images, the size of dictionary $\boldsymbol{D}$ was set to 250, i.e., $\boldsymbol{D} \in \mathbb{R}^{64\times 250}$. In each dataset, we added Gaussian noises from $\mathcal{N}(0, 0.3)$. The parameter $\lambda$ for SVG is set to 8 for

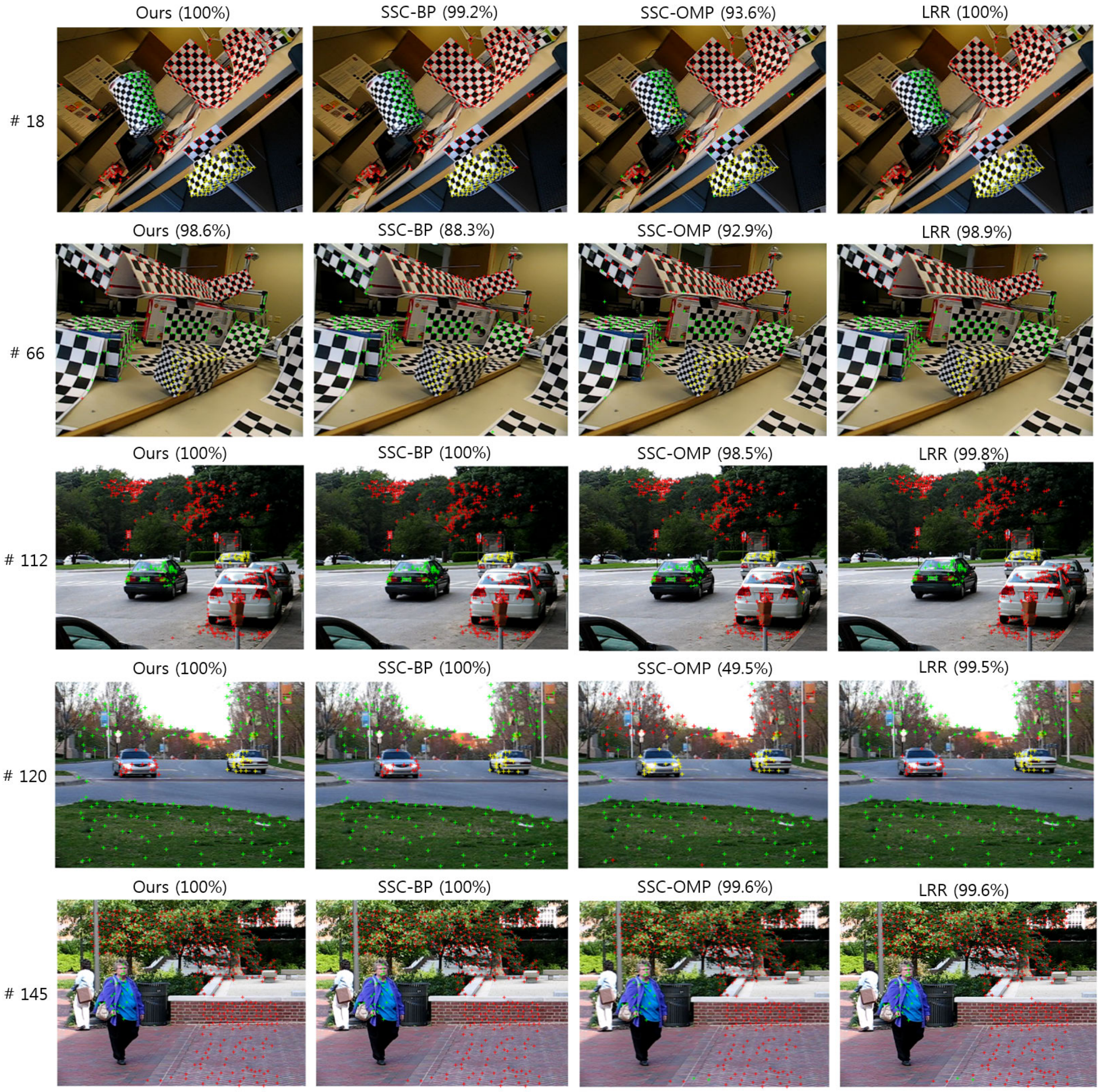


**FIGURE 7.** Motion segmentation results (snapshots) of five randomly chosen video sequences from the Hopkins 155 dataset by four methods: the proposed method, SSC-BP [36], SSC-OMP [41], and LRR [40]. Tracked points are marked by a symbol '+'. Different colors in the mark correspond to independent motion clusters. $(\cdot)$ denotes the segmentation accuracy. Best viewed in color (x2).

**TABLE 1.** Average reconstruction errors ($\times 10^2$) for sparse coding.

| Methods | Barbara | Lena | Boat | Peppers | Average |
|---|---|---|---|---|---|
| KSVD [1] | 2.23 | 1.90 | 2.04 | 2.05 | 2.06 |
| SC [39] | 2.11 | 2.02 | 2.15 | 2.12 | 2.1 |
| SVG (Ours) | **1.48** | **1.86** | **0.79** | **1.1** | **1.31** |

the images. The average reconstruction errors of the tested algorithms are shown in Table 1. In the table, our algorithm gives excellent results for all cases. KSVD, which uses OMP, performs slightly better than SC based on the $l_1$-norm, but it is unsatisfactory compared to ours. This experiment also demonstrates the excellence of the proposed surrogate.

### D. SUBSPACE CLUSTERING

We applied the proposed method, termed SSC-FAN, to two benchmark problems, face clustering [44] and motion segmentation [45], for subspace clustering.

#### 1) FACE CLUSTERING

We evaluated the proposed surrogate on the Extended Yale B database [44] for face clustering. The dataset consists

**TABLE 2.** Performance comparison on clustering accuracy (%) on the Extended Yale B dataset for face clustering.

| No. clusters ($c$) | 2 | 5 | 8 | 10 | Average |
|---|---|---|---|---|---|
| LRR [40] | 96.9 | 89.1 | 87.5 | 80.3 | 88.5 |
| SSC-BP [36] | 94.5 | 93.1 | 88.9 | 70.5 | 86.8 |
| SSC-OMP [41] | 98.4 | **97.8** | 81.1 | 82.9 | 90.5 |
| SSC-SL0 [10] | 98.4 | 75.6 | 66.2 | 53.4 | 73.4 |
| SSC-SVG (Ours) | **99.2** | 97.5 | **92.4** | **88.1** | **94.3** |

**TABLE 3.** Performance comparison with respect to clustering accuracy on the Hopkins 155 dataset for motion segmentation.

| | Mean | Std. | Median | Minimum |
|---|---|---|---|---|
| LRR [40] | 96.53 | 8.04 | 99.72 | **58.19** |
| SSC-BP [36] | 96.47 | 9.12 | **100** | 52.81 |
| SSC-OMP [41] | 87.16 | 14.04 | 93.10 | 46.82 |
| SSC-SL0 [10] | 77.93 | 16.82 | 80.82 | 39.44 |
| SSC-SVG (Ours) | **97.31** | **7.25** | **100** | 58.14 |

of 38 subjects, each of which has 64 frontal face images under illumination changes. We collected the first $c$ subjects, where $c \in \{2, 5, 8, 10\}$, and performed subspace clustering on the images of these subjects. For each problem, we used PCA to project images in $9c$-dimensional subspaces to make an overcomplete dictionary. We set the parameter $\lambda$ to 90. Table 2 shows the clustering accuracy for different numbers of subjects. The proposed method, SSC-SVG, shows better clustering performance than the existing algorithms based on the convex or nonconvex regularizers. SSC-OMP performs better than SSC-BP, SSC-SL0, and LRR on average, but it gives lower accuracy than ours for most scenarios. Especially, its performance collapses considerably when $c > 5$. SSC-SL0 shows the worst performance among the tested algorithms.

### 2) MOTION SEGMENTATION

The goal of motion segmentation task is to segment trajectories of rigidly moving objects based on tracked points along the frames. Since collected trajectories from a rigid motion lie in a low-dimensional subspace, we can solve the motion segmentation as a subspace clustering problem [36]. Hence, we applied SSC-SVG to the well-known benchmark dataset, Hopkins 155 [45], which consists of 155 video sequences with two or three motion clusters. The average size of dimensionality and samples for each sequence are roughly 60 and 296, respectively. In this problem, we set $\lambda = 7 \times 10^{-3}$. Four quantitative measures were used for clustering performance: mean, standard deviation (Std.), minimum, and median, following the work in [36]. The average performance of the algorithms are shown in Table 3. As shown in the table, our proposal outperforms existing algorithms approximating the $l_0$-norm and the dense representation method, LRR. SSC-BP and LRR give the similar performance, but they are unsatisfactory compared to ours, Two algorithms approximating the $l_0$-norm, SSC-OMP and SSC-SL0, show the disappointing results in this problem. Some graphical results on the dataset for four selected methods are illustrated in Figure 7.

## V. CONCLUSIONS

We have analyzed desirable criteria to be a good nonconvex sparsity surrogate and presented a corresponding family of surrogates that are a solution of a differential equation, named slowly vanishing gradients (SVG). Among the SVG surrogates, we selected a practical one as a proposed surrogate, which complements both $l_0$- and $l_1$-norms. The penalty possesses a simple proximity operator which allows efficient nonconvex optimizations. The penalty is a good alternative to the $l_0$-norm due to its slowly vanishing gradient property. The proposed surrogate has been tested on various applications in the literature to demonstrate its effectiveness and empirical results have confirmed the superiority of the proposal.

## APPENDIX A
## PROOF OF PROPOSITION 1

Since the proposed surrogate, SVG, is one of our representative family, we prove the properties in Proposition 1 for our family. We redefine the family of curves, called SVGF, as follows:

$$\|\boldsymbol{x}\|^{a,\epsilon}_{SVGF} \triangleq y(\boldsymbol{x}) = 1 - \frac{1}{(1+\frac{|x|}{a\epsilon})^a}, \tag{18}$$

where $a$ and $\epsilon$ are parameters of the family as defined in (9). If $a = 1$, the function in (18) becomes the proposed surrogate.

*Proposition 3:* SVGF satisfies the following properties:

1) $\|\boldsymbol{x}\|^{a,\epsilon}_{\mathrm{SVGF}} \leq \|\boldsymbol{x}\|_0$ $\forall a, \epsilon$ and $\|\boldsymbol{x}\|^{a,\epsilon}_{\mathrm{SVGF}} \to \|\boldsymbol{x}\|_0$ if $\epsilon \to 0$.
2) $\epsilon\|\boldsymbol{x}\|^{a,\epsilon}_{\mathrm{SVGF}} \leq \|\boldsymbol{x}\|_1$ $\forall a, \epsilon$ and $\epsilon\|\boldsymbol{x}\|^{a,\epsilon}_{\mathrm{SVGF}} \to \|\boldsymbol{x}\|_1$ if $\epsilon \to \infty$.

*Proof:* Assume $a$ and $\epsilon$ in $\|\boldsymbol{x}\|^{a,\epsilon}_{SVGF}$ are positive. We simply show the proposition for a scalar case, but its extension to a vector case is straightforward. It is easily checked that $y(x) = 0$ if $x = 0$ and $y(x) \leq 1$ if $x \neq 0$, thus we verify that SVGF is always lower than or equal to the $l_0$-norm for all $x$ regardless of $\epsilon$. If $\epsilon$ goes to zero, $\frac{1}{(1+\frac{|x|}{a\epsilon})^a} \to 0$ when $x \neq 0$. Thus, $y(x) \to 1$ and the asymptotic convergence to the $l_0$-norm holds.

Note that both $y(x)$ and the $l_1$-norm are symmetric around zero and nonnegative (with $y(0) = 0$). Then, $\epsilon y(x)$ is lower than or equal to the $l_1$-norm, since $\epsilon y'(x) = \frac{1}{(1+\frac{x}{a\epsilon})^{a+1}} \leq 1$ for all nonnegative $x$. This also holds for $x < 0$. Finally, in order to show that $\epsilon y(x)$ asymptotically converges to $|x|$ if $\epsilon \to \infty$, we use the following relation:

$$\lim_{\epsilon \to \infty} \epsilon y = \lim_{\beta \to 0} \frac{1}{\beta}\left(1 - \frac{1}{(1+\frac{\beta|x|}{a})^a}\right) \triangleq \lim_{\beta \to 0} \frac{f(\beta)}{g(\beta)}, \tag{19}$$

where

$$f(\beta) = 1 - \frac{1}{(1+\frac{\beta|x|}{a})^a} \quad \text{and} \quad g(\beta) = \beta \triangleq \frac{1}{\epsilon}. \tag{20}$$

Since $\lim_{\beta\to 0} f(\beta) = \lim_{\beta\to 0} g(\beta) = 0$, $g'(\beta) = 1 \neq 0$, and $\lim_{\beta\to 0} \frac{f'(\beta)}{g'(\beta)}$ exists, we have the following results by the L'Hospital's rule:

$$\lim_{\beta\to 0} \frac{f(\beta)}{g(\beta)} = \lim_{\beta\to 0} \frac{f'(\beta)}{g'(\beta)} = \lim_{\beta\to 0} \frac{a\frac{|x|}{a}(1+\frac{\beta|x|}{a})^{-a-1}}{1} = |x|, \tag{21}$$

which completes the proof. ■

## APPENDIX B
## PROOF OF PROPOSITION 2

The first two assumptions in Assumption 2 correspond to some of our criteria: *Symmetry* and *Monotonicity*, respectively. First, it is straightforward to show the symmetry of SVG. By taking a derivative of $\phi_\lambda$ for $x > 0$,

$$\phi'_\lambda = \frac{\lambda}{\epsilon(1+\frac{x}{a\epsilon})^{a+1}} > 0, \tag{22}$$

we can check the nondecreasing nature on the nonnegative real-line. For the third assumption, i.e., $(\frac{\phi_\lambda(x)}{x})' \leq 0$, we can verify it based on the following relation for $x > 0$:

$$(\frac{\phi_\lambda(x)}{x})' \leq 0 \;\Leftrightarrow\; x\phi'_\lambda(x) - \phi_\lambda(x) \leq 0. \tag{23}$$

If $h_\lambda(x) \triangleq x\phi'_\lambda(x) - \phi_\lambda(x)$ is a decreasing function, the third assumption is satisfied. If $h_\lambda(0) \leq 0$ and $h'_\lambda(x) \leq 0$, then $h_\lambda(x) \leq 0$ for $x > 0$. Since we have

$$h_\lambda(0) = 0 \cdot \phi'_\lambda(0) - \phi_\lambda(0) = 0,$$
$$h'_\lambda(x) = \phi'_\lambda(x) + x\phi''_\lambda(x) - \phi'_\lambda(x) = x\phi''_\lambda(x) < 0,$$

from our *Smoothness* criterion, $h_\lambda(x) \leq 0$ is satisfied for $x > 0$, and thus $(\frac{\phi_\lambda(x)}{x})' \leq 0$. For the fourth assumption, we can easily check $\lim_{x\to 0^+} \phi'_\lambda(x) = \frac{\lambda}{\epsilon}$ using the following equation:

$$\phi_{\lambda=1}(x) = 1 - \frac{1}{(1+\frac{x}{a\epsilon})^a}, \tag{24}$$

for $a > 0$, thus we obtain $L = \frac{1}{\epsilon}$. For the last condition, we take the second derivative of $\rho_\lambda(x)$:

$$\rho''_\lambda(x) = \begin{cases} -\frac{(a+1)\lambda}{a\epsilon^2} \cdot \frac{1}{(1+\frac{x}{a\epsilon})^{a+2}} + \mu, & \text{if } x > 0, \\ -\frac{(a+1)\lambda}{a\epsilon^2} + \mu, & \text{if } x = 0, \\ -\frac{(a+1)\lambda}{a\epsilon^2} \cdot \frac{1}{(1+\frac{-x}{a\epsilon})^{a+2}} + \mu, & \text{if } x < 0. \end{cases} \tag{25}$$

Since $\phi''_\lambda(x)$ is lower bounded by $-\frac{(a+1)\lambda}{a\epsilon^2}$, it is true that there exists

$$\mu = \frac{(a+1)\lambda}{a\epsilon^2} > 0 \tag{26}$$

satisfying the convexity of $\rho_{\lambda,\mu}(x)$. From Proposition 2, we directly obtain the result on the proposed surrogate as a special case, i.e., $a = 1$.

## APPENDIX C
## DERIVATIONS OF THE LRA PROBLEMS

For the LRA problem, we apply SVG for modeling sparse errors, whose problem formulation, termed LRA-SVG, is constructed as follows:

$$\min_{\boldsymbol{E},\boldsymbol{M}} \|\mathcal{P}_{\Omega_X}(\boldsymbol{E})\|^{\epsilon}_{\text{SVG}}, \;\text{s.t.}\; \boldsymbol{E} = \boldsymbol{X} - \boldsymbol{M}, \;\; \text{rank}(\boldsymbol{M}) \leq r. \tag{27}$$

The augmented Lagrangian of (27) is constructed as

$$\mathcal{L}(\boldsymbol{E},\boldsymbol{M},\boldsymbol{\Pi}) = \|\mathcal{P}_{\Omega_X}(\boldsymbol{E})\|^{\epsilon}_{\text{SVG}} + \langle \boldsymbol{\Pi}, \boldsymbol{E} - \boldsymbol{X} + \boldsymbol{M}\rangle + \frac{\gamma}{2}\|\boldsymbol{E} - \boldsymbol{X} + \boldsymbol{M}\|^2_F, \tag{28}$$

such that $\text{rank}(\boldsymbol{M}) \leq r$. Based on (28), we obtain an algorithm using the following steps:

$$\boldsymbol{E}_+ \leftarrow \min_{\boldsymbol{E}} \|\mathcal{P}_{\Omega_X}(\boldsymbol{E})\|^{\epsilon}_{\text{SVG}} + \frac{\gamma}{2}\|\boldsymbol{D} + \frac{\boldsymbol{\Pi}}{\gamma}\|^2_F, \tag{29}$$

$$\check{\boldsymbol{M}} \leftarrow \min_{\boldsymbol{M}} \frac{\gamma}{2}\|\boldsymbol{D} + \frac{\boldsymbol{\Pi}}{\gamma}\|^2_F, \tag{30}$$

$$\boldsymbol{M}_+ \leftarrow \boldsymbol{U}_r \mathcal{S}_{\frac{1}{\gamma}}[\boldsymbol{\Sigma}_r]\boldsymbol{V}_r^T, \tag{31}$$

$$\boldsymbol{\Pi}_+ \leftarrow \boldsymbol{\Pi} + \gamma\boldsymbol{D}, \tag{32}$$

where $\boldsymbol{D} \triangleq \boldsymbol{E} - \boldsymbol{X} + \boldsymbol{M}$, $\boldsymbol{\Pi}$ denotes the Lagrange multiplier, and $\gamma$ is a positive weighting parameter. For (31), we collect $r$ largest singular values and their corresponding singular vectors computed by the singular value decomposition (SVD) on $\check{\boldsymbol{M}}$ obtained from (30), i.e., $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] = svd(\check{\boldsymbol{M}})$. To solve for $\boldsymbol{E}$, we consider the following optimization problem for each element $e_{ij}$ indexed by $\Omega_{\boldsymbol{X}}$:

$$\min_{e_{ij}} \frac{|e_{ij}|}{|e_{ij}| + \epsilon} + \frac{\gamma}{2}(e_{ij} - x_{ij} + m_{ij} + \frac{\pi_{ij}}{\gamma})^2 \tag{33}$$

where $x_{ij}$, $m_{ij}$, and $\pi_{ij}$ are the $(i,j)^{th}$ elements of $\boldsymbol{X}$, $\boldsymbol{M}$, and $\boldsymbol{\Pi}$, respectively. The solution of (33) can be found by an efficient computation for each element separately as explained in Section III-A. For another element $e_{kl}$ indexed by $\overline{\Omega}_{\boldsymbol{X}}$, where $\overline{\Omega}_{\boldsymbol{X}}$ is a complementary support set of $\boldsymbol{X}$, we obtain $e_{kl} \leftarrow x_{kl} - m_{kl} - \frac{\pi_{kl}}{\gamma}$.

For the tested algorithms based on the same ADMM framework, such as LRA-L1, LRA-CapL1, and LRA-MCP, we simply switch the penalty function $\|\cdot\|^{\epsilon}_{\text{SVG}}$ in (27), (28), and (29) to a nonconvex penalty function and solve its corresponding optimization problem. As an example, LRA-L1 considers the following optimization problem in the ADMM framework when solving for the variable $\boldsymbol{E}$:

$$\boldsymbol{E}_+ \leftarrow \min_{\boldsymbol{E}} \|\mathcal{P}_{\Omega_X}(\boldsymbol{E})\|_1 + \frac{\gamma}{2}\|\boldsymbol{D} + \frac{\boldsymbol{\Pi}}{\gamma}\|^2_F, \tag{34}$$

and its solution is computed as follows:

$$\boldsymbol{E}_+ \leftarrow \mathcal{P}_{\Omega_X}(\mathcal{S}_{\frac{1}{\gamma}}(\boldsymbol{Y})) + \mathcal{P}_{\overline{\Omega}_X}(\boldsymbol{Y}), \tag{35}$$

where $\boldsymbol{Y} \triangleq \boldsymbol{X} - \boldsymbol{M} - \frac{\boldsymbol{\Pi}}{\gamma}$ and $\mathcal{S}_\gamma(t) = sign(t)\max(|t| - \gamma, 0)$ is the shrinkage operator [46] for a scalar variable $t$. Other problems based on the nonconvex penalty functions described in the experimental section to solve for $\boldsymbol{E}$ can be solved efficiently by the work in [13].

## REFERENCES

[1] 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.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[4] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[5] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, 1993, pp. 40–44.

[7] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, May 1993.

[8] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[9] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted $l_1$ minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, 2008.

[10] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed $l^0$ norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[11] D. Wipf and S. Nagarajan, "Iterative reweighted $l_1$ and $l_2$ methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[12] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, Apr. 2010.

[13] P. Gong, C. Zhang, Z. Lu, Z. Z. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proc. ICML*, 2013, pp. 37–45.

[14] E. Kim, M. Lee, C.-H. Choi, N. Kwak, and S. Oh, "Efficient $l_1$-norm-based low-rank matrix approximations for large-scale problems using alternating rectified gradient method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 237–251, Feb. 2015.

[15] C. Lu, J. Feng, Z. Lin, and S. Yan, "Nonconvex sparse spectral clustering by alternating direction method of multipliers and its convergence analysis," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[16] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through $l_0$ regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.

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

[20] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, 2003, pp. 1–8.

[21] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2488–2495.

[22] E. Kim, M. Lee, and S. Oh, "Elastic-net regularization of singular values for robust subspace learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 915–923.

[23] E. Kim, M. Lee, and S. Oh, "Robust elastic-net subspace representation," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4245–4259, Sep. 2016.

[24] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the $k$-support norm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1457–1465.

[25] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[26] T. Zhang, "Multi-stage convex relaxation for learning with sparse regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1929–1936.

[27] R. Mazumder, J. H. Friedman, and T. Hastie, "SparseNet: Coordinate descent with nonconvex penalties," *J. Amer. Stat. Assoc.*, vol. 106, no. 495, pp. 1125–1138, Sep. 2011.

[28] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4130–4137.

[29] H. Jiang, D. P. Robinson, R. Vidal, and C. You, "A nonconvex formulation for low rank subspace clustering: Algorithms and convergence analysis," *Comput. Optim. Appl.*, vol. 70, no. 2, pp. 395–418, Jun. 2018.

[30] M. Brbic and I. Kopriva, "$\ell_0$ -Motivated low-rank sparse subspace clustering," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1711–1725, Apr. 2020.

[31] X. Deng, T. Sun, P. Du, and D. Li, "A nonconvex implementation of sparse subspace clustering: Algorithm and convergence analysis," *IEEE Access*, vol. 8, pp. 54741–54750, 2020.

[32] F. H. Clarke, "Generalized gradients and applications," *Trans. Amer. Math. Soc.*, vol. 205, pp. 247–262, Apr. 1975.

[33] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 379–387.

[34] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," 2015, *arXiv:1511.06324*. [Online]. Available: http://arxiv.org/abs/1511.06324

[35] P.-L. Loh and M. J. Wainwright, "Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima," *J. Mach. Learn. Res.*, vol. 16, pp. 559–616, Mar. 2015.

[36] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2790–2797.

[37] Y. Shen, Z. Wen, and Y. Zhang, "Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Softw.*, vol. 29, no. 2, pp. 239–263, Mar. 2014.

[38] A. M. Buchanan and A. W. Fitzgibbon, "Damped Newton algorithms for matrix factorization with missing data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 316–322.

[39] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 801–808.

[40] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. ICML*, 2010, pp. 663–670.

[41] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2487–2517, 2013.

[42] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[43] L. Torresani, A. Hertzmann, and C. Bregler, "Learning non-rigid 3D shape from 2D motion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1555–1562.

[44] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[45] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[46] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011. [Online]. Available: https://arxiv.org/abs/1009.5055

**EUNWOO KIM** (Member, IEEE) received the B.S. degree in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2011, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, in 2013 and 2017, respectively. From 2017 to 2018, he was a Postdoctoral Researcher with the Department of Electrical Engineering and Computer Science, Seoul National University. From 2018 to 2019, he was a Postdoctoral Researcher with the Department of Engineering Science, University of Oxford, Oxford, U.K. He is currently an Assistant Professor with the School of Computer Science and Engineering, Chung-Ang University. His research interests include machine learning, deep learning, subspace representation, and computer vision.

**MINSIK LEE** (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, South Korea, in 2006 and 2012, respectively. From 2012 to 2013, he was a Postdoctoral Researcher with the School of Electrical Engineering and Computer Science, Seoul National University. In 2014, he joined Seoul National University as a BK21 Assistant Professor. He is currently an Associate Professor with Hanyang University, Ansan, South Korea. His research interests include shape and motion analysis, deformable models, computer vision, deep learning, pattern recognition, and their applications.

**SONGHWAI OH** (Member, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1995, 2003, and 2006, respectively. Before his Ph.D. studies, he was a Senior Software Engineer at Synopsys, Inc., Mountain View, CA, USA, and a Microprocessor Design Engineer at Intel Corporation, Santa Clara, CA, USA. In 2007, he was a Postdoctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. From 2007 to 2009, he was an Assistant Professor of electrical engineering and computer science with the School of Engineering, University of California at Merced, Merced, CA, USA. He is currently a Professor with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. His current research interests include robotics, computer vision, cyber-physical systems, and machine learning.