L_0 Gradient Projection

Shunsuke Ono, *Member, IEEE*

***Abstract*—Minimizing L_0 gradient, the number of the non-zero gradients of an image, together with a quadratic data-fidelity to an input image has been recognized as a powerful edge-preserving filtering method. However, the L_0 gradient minimization has an inherent difficulty: a user-given parameter controlling the degree of flatness does not have a physical meaning since the parameter just balances the relative importance of the L_0 gradient term to the quadratic data-fidelity term. As a result, the setting of the parameter is a troublesome work in the L_0 gradient minimization. To circumvent the difficulty, we propose a new edge-preserving filtering method with a novel use of the L_0 gradient. Our method is formulated as the minimization of the quadratic data-fidelity subject to the hard constraint that the L_0 gradient is less than a user-given parameter α. This strategy is much more intuitive than the L_0 gradient minimization because the parameter α has a clear meaning: the L_0 gradient value of the output image itself, so that one can directly impose a desired degree of flatness by α. We also provide an efficient algorithm based on the so-called alternating direction method of multipliers for computing an approximate solution of the nonconvex problem, where we decompose it into two subproblems and derive closed-form solutions to them. The advantages of our method are demonstrated through extensive experiments.**

***Index Terms*—L_0 gradient, edge-preserving filtering, nonconvex optimization.**

I. Introduction

EDGE-PRESERVING filtering is a fundamental tool in image processing and graphics. Various edge-preserving filtering methods have been proposed in the literature: representative examples include the anisotropic diffusion [1], [2], the bilateral filter [3], the guided filter [4], the domain-transform [5], the weighted least squares [6], the total variation minimization [7], [8], the L_0 gradient minimization [9].

Among these methods, the L_0 gradient minimization is known to have a strong ability of edge-preserving *flattening*, and it has been an active research topic [9]–[19]. In the framework of the L_0 gradient minimization, the filtering is performed by minimizing the sum of an L_0 gradient term and a quadratic data-fidelity term, where the L_0 gradient is defined as the number of the non-zero gradients of the output image, and the quadratic data-fidelity penalizes the L_2 distance between an input image and the output image (the L_1 distance case was studied in [11], [17]). The degree of flatness of the output image is controlled by a user-given parameter λ that balances the relative importance of the L_0 gradient to the quadratic data-fidelity. Since the global optimization is very hard due to the nonconvexity of the L_0 gradient, a number of algorithms that approximately solve the L_0 gradient minimization have been developed [9], [14], [15], [19].

Aside from the difficulty of the optimization, one has to make another effort in the L_0 gradient minimization: the selection of a suitable value of λ. This is because the parameter λ does not have a physical meaning, i.e., it does not directly correspond to the degree of flatness of the output image but it just balances the two terms as said before. Although setting the larger value of λ results in the smaller L_0 gradient value of the output image, the explicit relation between them is unavailable, and thus users cannot specify the L_0 gradient value of the output image in advance. Let us give an example: we show two input images (almost the same resolution) in Fig. 1(a), and the corresponding filtered images generated by the L_0 gradient minimization with λ adjusted so that both images achieve almost the same L_0 gradient value in Fig. 1(b).[1] One sees that λ is quite different for each image, which means that users cannot impose the same degree of flatness for different input images by simply setting the same value of λ. As a result, users have to set λ in an *ad hoc* manner and/or solve the L_0 gradient minimization many times with various λ to achieve a desired degree of flatness.

To circumvent this difficulty, we propose a novel formulation based on the L_0 gradient for edge-preserving filtering, which is the first contribution of the paper. The formulation is defined as a constrained optimization problem: minimizing the quadratic data-fidelity subject to the hard constraint that the L_0 gradient is less than a user-given parameter α. We name this constrained formulation L_0 *gradient projection* because it is analogous to the notion of the metric projection onto a convex set[2] (but our case is nonconvex). In contrast to the L_0 gradient minimization, the parameter controlling the degree of flatness in the L_0 gradient projection, i.e., α, has a very clear meaning: it is the L_0 gradient value of the output image itself. This enables users to directly impose a desired degree of flatness by α, as one can see in Fig. 1(c) where the L_0 gradient values of our results are almost the same as α. Moreover, users can determine α based on the number of the pixels or the L_0 gradient value of an input image. As an example, Fig. 2 shows our filtered images generated with α set to various percentage of the number of the pixels of the input image (left). This kind of setting is useful when we impose the same degree of flatness to input images of different sizes. Consequently, we can say that the parameter setting in our method is much

S. Ono is with Laboratory for Future Interdisciplinary Research of Science and Technology (FIRST), Institute of Innovative Research (IIR), Tokyo Institute of Technology, Yokohama, Kanagawa 226-8503, Japan.
E-mail: ono@isl.titech.ac.jp

Manuscript received xx xxx. 2016; revised xx xxx. 2016; accepted xx xxx. 2016. Date of publication xx xxx. 2016; date of current version xx xxx. 2016. Recommended for acceptance by X. Xxxxx. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. xx.xxxx/TIP.2016.xxxxxxx

[1] We used the algorithm proposed in [19] for optimization.

[2] Given a vector $\bar{\mathbf{x}}$ and a nonempty closed convex set C, the *metric projection onto* C is characterized by the following constrained optimization problem: $\min_{\mathbf{x}} \|\mathbf{x} - \bar{\mathbf{x}}\|$ s.t. $\mathbf{x} \in C$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2651392, IEEE Transactions on Image Processing

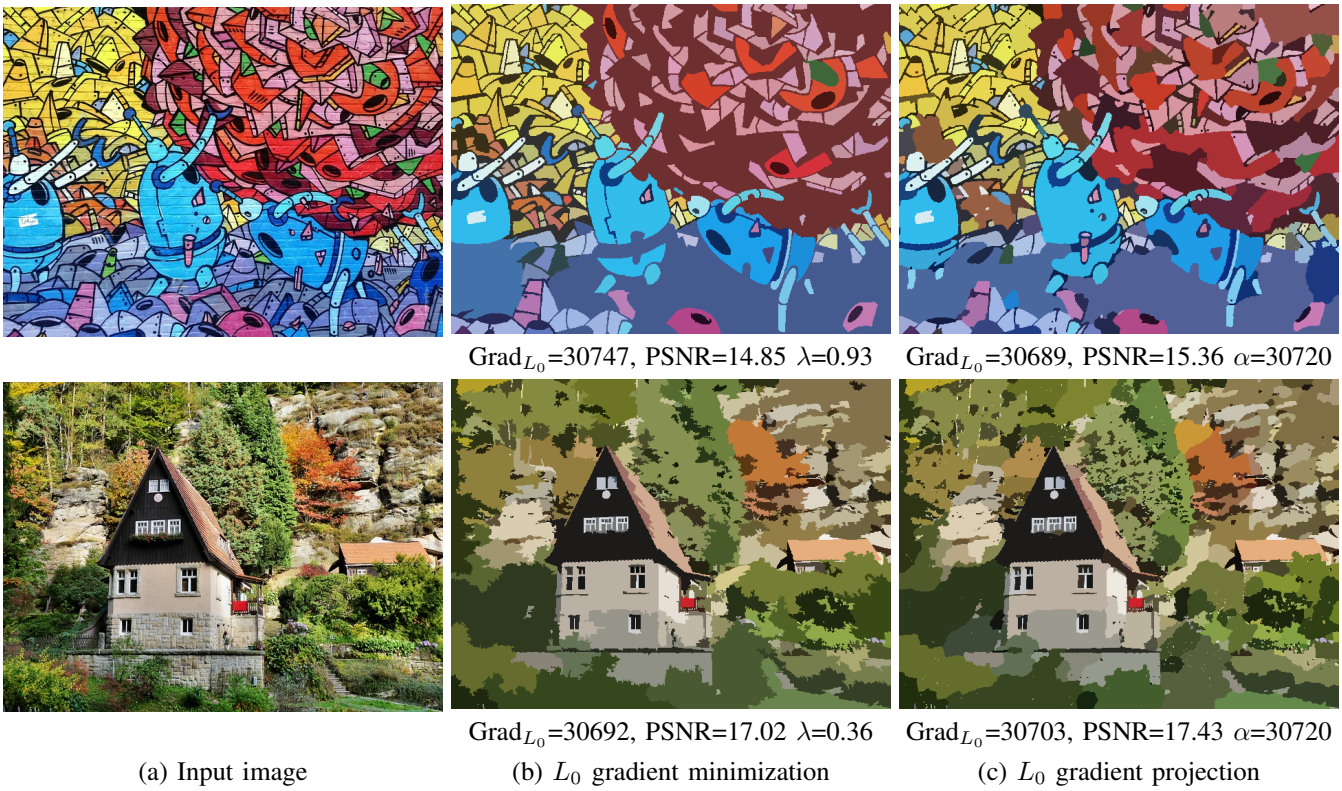


Fig. 1. Comparison of the L_0 gradient minimization and the L_0 gradient projection. We show the L_0 gradient values (Grad$_{L_0}$), PSNR[dB], and the parameters used.

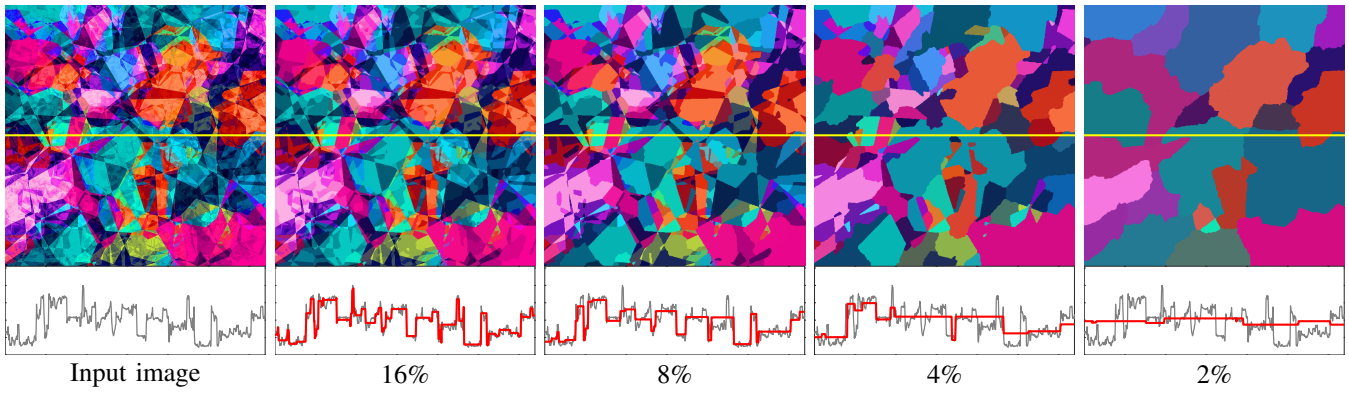


Fig. 2. Examples of our filtered images generated with α set to various percentages of the number of the pixels of the input image. We also plot the 1D scanlines from their luminance channels.

more intuitive than that in the L_0 gradient minimization. Such advantages of constrained formulation over unconstrained one have been addressed in the literature of image restoration based on convex optimization [20]–[26].

The next question would be how to solve it, because in addition to the nonconvexity of the L_0 gradient, constrained optimization is harder to solve in general than unconstrained counterpart. The second contribution of the paper is to provide an efficient algorithm to compute an approximate solution of the L_0 gradient projection. Our algorithm is in the framework of the so-called alternating direction method of multipliers (ADMM) [27]–[29], which is a generic convex optimization algorithm but it has been shown to be effective even for nonconvex optimization, e.g., [29]–[35]. In our algorithm, the L_0 gradient projection is decomposed into two subproblems, and they are solved alternately with a dual variable update. Although one of the subproblems is still a constrained nonconvex optimization problem, we prove that it has a closed-form solution. As a result, the computational cost of each iteration turns out to be $\mathcal{O}(N \log N)$ time (N is the number of pixels). Moreover, we empirically show that at each step of our algorithm, the gap between the L_0 gradient value of the current variable and α decreases monotonically. Hence, our algorithm can compute an approximate solution of the L_0 gradient projection efficiently.

Despite the fact that our algorithm is established for solving

the L_0 gradient projection (not for the L_0 gradient minimization), interestingly, the output image of our algorithm achieves a smaller value of the quadratic data-fidelity (higher PSNR) than those of existing L_0 gradient minimization algorithms, under the condition that their L_0 gradient values are the same (see Fig. 1). This indicates that our algorithm is also effective in terms of the L_0 gradient minimization.

The remainder of the paper is organized as follows. Section II introduces several mathematical ingredients. In Section III, we establish the L_0 gradient projection framework and briefly discuss the relation to existing L_0 gradient minimization algorithms. The advantages of the L_0 gradient projection are demonstrated through extensive experiments in Section IV. Section V concludes the paper.

The preliminary version of this work, without mathematical details, deeper discussion, comparison to existing methods, or new applications, has appeared in conference proceedings [36].

II. Preliminaries

Throughout the paper, let $\mathbb{R}$ be the set of real numbers. We shall use boldface lowercase and capital to represent vectors and matrices, respectively. We denote the transpose of a vector/matrix by $(\cdot)^\top$, the Euclidean norm (the ℓ_2 norm) of a vector by $\|\cdot\|$, and the zero vector of appropriate size by $\mathbf{0}$. For notational convenience, we treat an RGB color image in $\mathbb{R}^{N_v \times N_h \times 3}$ as a vector in $\mathbb{R}^{3N}$ ($N := N_v N_h$ is the number of the pixels) by stacking its columns on top of one another, in the order of the R, G and B channels. Clearly, both treatments are essentially the same.

A. L_0 Gradient

For a color image $X \in \mathbb{R}^{N_v \times N_h \times 3}$, the L_0 *gradient* of X [9] is defined by

$$\mathrm{Grad}_{L_0}(X) := \sum_{i=1}^{N_v}\sum_{j=1}^{N_h} F\left(\sum_{c=1}^{3}\left(|X_{i+1,j,c} - X_{i,j,c}| + |X_{i,j+1,c} - X_{i,j,c}|\right)\right), \tag{1}$$

where $X_{i,j,c}$ denotes the cth channel component of the pixel at (i,j), and $F(x) := 1$, if $x \neq 0$; $F(x) := 0$, otherwise. Note that we define $|X_{i+1,j,c} - X_{i,j,c}| := 0$ for $i+1 > N_v$, and $|X_{i,j+1,c} - X_{i,j,c}| := 0$ for $j+1 > N_h$. In a nutshell, the function Grad_{L_0} counts the number of pixels whose vertical and/or horizontal difference is nonzero, i.e., it quantifies the degree of flatness of X.

B. Alternating Direction Method of Multipliers (ADMM)

The *Alternating direction method of multipliers* (ADMM) [27]–[29] is an algorithm that can solve convex optimization problems of the form:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}) \;\text{ subject to }\; \mathbf{y} = \mathbf{L}\mathbf{x}, \tag{2}$$

where the functions f and g are usually assumed to be proper lower semicontinuous convex, and the matrix $\mathbf{L}$ is assumed to have full-column rank. For arbitrarily chosen $\mathbf{y}^{(0)}, \mathbf{z}^{(0)}$ and $\gamma > 0$, ADMM iterates the following steps:

$$\left\lfloor \begin{array}{l} \mathbf{x}^{(n+1)} = \underset{\mathbf{x}}{\mathrm{argmin}}\, f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{y}^{(n)} - \mathbf{L}\mathbf{x} - \mathbf{z}^{(n)}\|^2 \\ \mathbf{y}^{(n+1)} = \underset{\mathbf{y}}{\mathrm{argmin}}\, g(\mathbf{y}) + \frac{1}{2\gamma}\|\mathbf{y} - \mathbf{L}\mathbf{x}^{(n+1)} - \mathbf{z}^{(n)}\|^2 \\ \mathbf{z}^{(n+1)} = \mathbf{z}^{(n)} + \mathbf{L}\mathbf{x}^{(n+1)} - \mathbf{y}^{(n+1)}. \end{array} \right. \tag{3}$$

Although ADMM was originally developed for convex optimization, it has been reported that ADMM works well in practice for nonconvex optimization in various applications [29]–[35], which motivates us to use ADMM in our framework.

III. Proposed Method

A. Formulation of L_0 Gradient Projection

For a given input image $\bar{\mathbf{u}} \in \mathbb{R}^{3N}$, we newly formulate edge-preserving filtering to $\bar{\mathbf{u}}$ as a constrained optimization problem involving the L_0 gradient, named the L_0 *gradient projection*, as follows:

$$\text{Find } \mathbf{u}^\star \in \underset{\mathbf{u}\in\mathbb{R}^{3N}}{\mathrm{argmin}} \frac{1}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2 \text{ subject to } \mathrm{Grad}_{L_0}(\mathbf{u}) \leq \alpha, \tag{4}$$

where α is a user-given parameter that is the least upper bound of $\mathrm{Grad}_{L_0}(\mathbf{u}^\star)$, i.e., a desired degree of flatness imposed on the output image $\mathbf{u}^\star$. The cost function of Prob. (4) is only the quadratic data-fidelity that penalizes the L_2 distance between the input and output images. If we set $\alpha \geq \mathrm{Grad}_{L_0}(\bar{\mathbf{u}})$, then the optimal solution of Prob. (4) equals to the input image $\bar{\mathbf{u}}$ itself, i.e., $\mathbf{u}^\star = \bar{\mathbf{u}}$, which is trivial. Otherwise, the optimal solution(s) of Prob. (4) must be as close to $\bar{\mathbf{u}}$ as possible, so that $\mathbf{u}^\star$ would satisfy $\mathrm{Grad}_{L_0}(\mathbf{u}^\star) = \alpha$. Hence, we can say that this problem formulation characterizes the best approximation of the input image $\bar{\mathbf{u}}$ within a user-given degree of flatness α.

Remark 1 (Comparison with L_0 gradient minimization)**.** The L_0 gradient minimization [9] is formulated as follows:

$$\text{Find } \mathbf{u}^\star_{\min} \in \underset{\mathbf{u}\in\mathbb{R}^{3N}}{\mathrm{argmin}} \lambda \mathrm{Grad}_{L_0}(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2, \tag{5}$$

where $\lambda > 0$ is a user-given parameter that balances the two terms in (5). As addressed in Sec. I, the parameter λ does not directly correspond to the degree of flatness of $\mathbf{u}^\star_{\min}$. Consequently, one cannot specify $\mathrm{Grad}_{L_0}(\mathbf{u}^\star_{\min})$ in advance, and $\mathrm{Grad}_{L_0}(\mathbf{u}^\star_{\min})$ varies depending on $\bar{\mathbf{u}}$ even if λ is fixed at a certain value, as observed in Fig. 1(b). On the other hand, the parameter α in Prob. (4) forces $\mathbf{u}^\star$ to satisfy $\mathrm{Grad}_{L_0}(\mathbf{u}^\star) = \alpha$, so that one can always obtain a filtered image of a desired degree of flatness, independent of $\bar{\mathbf{u}}$, as shown in Fig. 1(c). In addition, one can determine α based on the information on $\bar{\mathbf{u}}$, such as a certain percentage of N or $\mathrm{Grad}_{L_0}(\bar{\mathbf{u}})$. From the above facts, we claim that the parameter setting in our method is intuitive and easy.

B. Optimization of L_0 Gradient Projection

We propose an efficient algorithm for solving the L_0 gradient projection defined in (4) based on ADMM. Solving

Prob. (4) is essentially a challenging task due to the nonconvexity of the L_0 gradient. Moreover, constrained optimization like Prob. (4) is more difficult in general than unconstrained counterpart like Prob. (5). We resolve the difficulty as follows: we first split Prob. (4) into certain subproblems and then derive closed-form solutions to them.

1) Reformulation of Prob. (4): To establish the algorithm, we first reformulate Prob. (4) into an ADMM-applicable form defined in (2).

To this end, we start with introducing another expression of the L_0 gradient. Let $\mathbf{x} \in \mathbb{R}^{3N}$ be a color image, and $\mathbf{D} \in \mathbb{R}^{6N\times 3N}$ be a discrete difference operator with periodic boundary that maps the RGB channels of a color image to their vertical and horizontal discrete differences. We also define the mixed $L_{1,0}$ pseudo-norm as follows.

Definition 1 (Mixed $L_{1,0}$ pseudo-norm)**.** Let $\mathbf{y}$ be a vector of $\mathbb{R}^M$, and let $\mathcal{G}_1, \ldots, \mathcal{G}_K$ $(1 \le K \le M)$ be index sets such that

- Each $\mathcal{G}_k$ is a subset of $\{1, \ldots, M\}$.
- $\mathcal{G}_k \cap \mathcal{G}_l = \emptyset$ for any $k \neq l$.
- $\cup_{k=1}^{K} \mathcal{G}_k = \{1, \ldots, M\}$.

Suppose that $\mathbf{y}_{\mathcal{G}_k}$ $(k \in \{1, \ldots, K\})$ denotes a subvector of $\mathbf{y}$ with the entries specified by $\mathcal{G}_k$. Then, the mixed $L_{1,0}$ pseudo-norm of $\mathbf{y}$ is defined as

$$\|\mathbf{y}\|_{1,0}^{\mathcal{G}} := \|(\|\mathbf{y}_{\mathcal{G}_1}\|_1, \cdots, \|\mathbf{y}_{\mathcal{G}_K}\|_1)\|_0, \tag{6}$$

where $\|\cdot\|_1$ is the L_1 norm defined as the sum of the absolute values of all the entries of $(\cdot)$, and $\|\cdot\|_0$ is the L_0 pseudo-norm defined as the number of the non-zero entries of $(\cdot)$.

In short, the mixed $L_{1,0}$ pseudo-norm counts the number of subvectors whose L_1 norms are nonzero. Then, another expression of the L_0 gradient is given by

$$\mathrm{Grad}_{L_0}(X) = \|\mathbf{B}\mathbf{D}\mathbf{x}\|_{1,0}^{\mathcal{G}'}, \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{6N\times 6N}$ is a diagonal matrix with binary entries (0 or 1) that forces discrete differences between opposite boundaries (due to the periodic boundary condition of $\mathbf{D}$) to be zero. Here, the number of the subvectors equals to N, i.e., $\mathcal{G}'_1, \ldots, \mathcal{G}'_N$, and each $\mathcal{G}'_k$ contains the indices corresponding to the vertical and horizontal differences at the kth pixel (to avoid confusion, we use $\mathcal{G}'$ to represent the specific index sets for the L_0 gradient, instead of $\mathcal{G}$).

Using the expression in (7) and introducing an auxiliary variable $\mathbf{v} := \mathbf{D}\mathbf{u}$, we can reformulate Prob. (4) into

$$\begin{aligned} &\text{Find } \mathbf{u}^\star \in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^{3N}} \frac{1}{2}\|\mathbf{u}-\bar{\mathbf{u}}\|^2 \\ &\text{subject to } \|\mathbf{B}\mathbf{v}\|_{1,0}^{\mathcal{G}'} \le \alpha \text{ and } \mathbf{v} = \mathbf{D}\mathbf{u}. \end{aligned} \tag{8}$$

Next, we define the indicator function of the inequality constraint on the mixed $L_{1,0}$ pseudo-norm composed with the operator $\mathbf{B}$ as

$$\iota_{\{\|\mathbf{B}\cdot\|_{1,0}^{\mathcal{G}}\le\alpha\}}(\mathbf{y}) := \begin{cases} 0, & \|\mathbf{B}\mathbf{y}\|_{1,0}^{\mathcal{G}} \le \alpha, \\ \infty, & \text{otherwise.} \end{cases} \tag{9}$$

Then, Prob. (8) is further reformulated as follows:

$$\begin{aligned} \text{Find } (\mathbf{u}^\star, \mathbf{v}^\star) \in \operatorname*{argmin}_{\substack{\mathbf{u}\in\mathbb{R}^{3N} \\ \mathbf{v}\in\mathbb{R}^{6N}}} &\frac{1}{2}\|\mathbf{u}-\bar{\mathbf{u}}\|^2 + \iota_{\{\|\mathbf{B}\cdot\|_{1,0}^{\mathcal{G}'}\le\alpha\}}(\mathbf{v}) \\ &\text{subject to } \mathbf{v} = \mathbf{D}\mathbf{u}, \end{aligned} \tag{10}$$

where $\mathbf{v}^\star = \mathbf{D}\mathbf{u}^\star$.

2) Algorithm: Now, by letting

$$f(\mathbf{u}) := \frac{1}{2}\|\mathbf{u}-\bar{\mathbf{u}}\|^2,\ g(\mathbf{v}) := \iota_{\{\|\mathbf{B}\cdot\|_{1,0}^{\mathcal{G}'}\le\alpha\}}(\mathbf{v}),\ \text{and } \mathbf{L} := \mathbf{D},$$

one can see that Prob. (10) is identical to Prob. (2) Thus, we can apply ADMM to Prob. (10), which yields an abstract version of our algorithm: for arbitrarily chosen $\mathbf{v}^{(0)}, \mathbf{w}^{(0)}$ and $\gamma > 0$, the algorithm iterates

$$\left\lfloor \begin{aligned} &\mathbf{u}^{(n+1)} = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^{3N}} \tfrac{1}{2}\|\mathbf{u}-\bar{\mathbf{u}}\|^2 + \tfrac{1}{2\gamma}\|\mathbf{v}^{(n)} - \mathbf{D}\mathbf{u} - \mathbf{w}^{(n)}\|^2 \\ &\mathbf{v}^{(n+1)} = \\ &\operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^{6N}} \iota_{\{\|\mathbf{B}\cdot\|_{1,0}^{\mathcal{G}'}\le\alpha\}}(\mathbf{v}) + \tfrac{1}{2\gamma}\|\mathbf{v} - \mathbf{D}\mathbf{u}^{(n+1)} - \mathbf{w}^{(n)}\|^2 \\ &\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} + \mathbf{D}\mathbf{u}^{(n+1)} - \mathbf{v}^{(n+1)}. \end{aligned} \right. \tag{11}$$

In what follows, we derive closed-form solutions to the subproblems in (11), in order to describe the detailed procedures of our algorithm.

Since the first subproblem, the update of $\mathbf{u}^{(n)}$, is a strictly convex quadratic minimization, it boils down to solving the system of linear equations, yielding

$$\mathbf{u}^{(n+1)} = (\mathbf{I}+\gamma^{-1}\mathbf{D}^\top\mathbf{D})^{-1}(\bar{\mathbf{u}}+\gamma^{-1}\mathbf{D}^\top(\mathbf{v}^{(n)}-\mathbf{w}^{(n)})). \tag{12}$$

Since the boundary condition of $\mathbf{D}$ is periodic, the matrix inversion in (12) can be computed efficiently via the 2D fast Fourier transform (2DFFT), i.e., the inversion in Eq. (12) can be calculated as

$$\mathbf{u}^{(n+1)} = \mathbf{F}^*(\mathbf{I} + \gamma^{-1}\mathbf{\Lambda})^{-1}\mathbf{F}(\bar{\mathbf{u}} + \gamma^{-1}\mathbf{D}^\top(\mathbf{v}^{(n)} - \mathbf{w}^{(n)})), \tag{13}$$

where $\mathbf{F}$ and $\mathbf{F}^*$ are the 2D discrete Fourier transform matrix and its inverse, respectively, and $\mathbf{\Lambda}$ is the diagonal matrix with its entries being the Fourier-transformed Laplacian filter kernel. Thanks to the structure that $\mathbf{I} + \gamma^{-1}\mathbf{\Lambda}$ is a diagonal matrix, its inversion is reduced to entry-wise division.

By noticing the definition of the indicator function in (9), the second subproblem, the update of $\mathbf{v}^{(n)}$, can be rewritten as follows:

$$\begin{aligned} \mathbf{v}^{(n+1)} = \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^{6N}} &\|\mathbf{v} - \mathbf{D}\mathbf{u}^{(n+1)} - \mathbf{w}^{(n)}\|^2 \\ &\text{subject to } \|\mathbf{B}\mathbf{v}\|_{1,0}^{\mathcal{G}'} \le \alpha, \end{aligned} \tag{14}$$

where the weight $\frac{1}{2\gamma}$ is removed since it is irrelevant to the optimization. Prob. (14) might appear to be difficult, but fortunately its optimal solution can be computed in a closed form, given by the following result.

Proposition 1 (Projection onto mixed $L_{1,0}$ pseudo-norm ball with binary mask)**.** *Let $\mathbf{z}$ be a vector of $\mathbb{R}^M$, let α be a nonnegative integer, let $\mathbf{S} \in \mathbb{R}^{M\times M}$ be a diagonal matrix with its diagonal entries being binary, and let $\mathcal{G}_1, \ldots, \mathcal{G}_K$*

$(1 \leq K \leq M)$ be index sets satisfying the conditions in Definition. 1. *Without loss of generality, we can assume that* $\mathbf{Sz} = (\mathbf{z}_{\mathcal{G}_1}^\top \cdots \mathbf{z}_{\mathcal{G}_K}^\top)^\top$. *In addition, we denote the subvectors* $\mathbf{z}_{\mathcal{G}_1}, \ldots, \mathbf{z}_{\mathcal{G}_K}$ *sorted in descending order in terms of their* L_2 *norms by* $\mathbf{z}_{\mathcal{G}_{(1)}}, \ldots, \mathbf{z}_{\mathcal{G}_{(K)}}$, *i.e,* $\|\mathbf{z}_{\mathcal{G}_{(1)}}\| \geq \|\mathbf{z}_{\mathcal{G}_{(2)}}\| \geq \cdots \geq \|\mathbf{z}_{\mathcal{G}_{(K)}}\|$. *Consider the problem:*

$$\text{Find } \mathbf{y}^\star \in \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^M} \|\mathbf{y}-\mathbf{z}\|^2 \text{ subject to } \|\mathbf{Sy}\|_{1,0}^{\mathcal{G}} \leq \alpha. \quad (15)$$

Then, one of the optimal solutions of the problem is given by

$$\mathbf{y}^\star = \begin{cases} \mathbf{z}, & \text{if } \|\mathbf{Sz}\|_{1,0}^{\mathcal{G}} \leq \alpha, \\ (\tilde{\mathbf{z}}_{\mathcal{G}_1}^\top \cdots \tilde{\mathbf{z}}_{\mathcal{G}_K}^\top)^\top + (\mathbf{I}-\mathbf{S})\mathbf{z}, & \text{if } \|\mathbf{Sz}\|_{1,0}^{\mathcal{G}} > \alpha, \end{cases} \quad (16)$$

where

$$\tilde{\mathbf{z}}_{\mathcal{G}_k} := \begin{cases} \mathbf{z}_{\mathcal{G}_k}, & \text{if } k \in \{(1), \ldots, (\alpha)\}, \\ \mathbf{0}, & \text{if } k \in \{(\alpha+1), \ldots, (K)\}. \end{cases} \quad (17)$$

Proof: Since the case of $\|\mathbf{Sz}\|_{1,0}^{\mathcal{G}} \leq \alpha$ is trivial, we consider the case of $\|\mathbf{Sz}\|_{1,0}^{\mathcal{G}} > \alpha$. To satisfy the inequality constraint $\|\mathbf{Sy}\|_{1,0}^{\mathcal{G}} \leq \alpha$ in Prob. (15), at least $K-\alpha$ subvectors of $\mathbf{Sy}^\star$ must be exactly zero vectors $\mathbf{0}$ from the definition of $\|\cdot\|_{1,0}^{\mathcal{G}}$ in (6). Meanwhile, any change in $\mathbf{y}^\star$ from $\mathbf{z}$ increases the value of $\|\mathbf{y}-\mathbf{z}\|^2$. From these facts, we see that the kth subvector of the $\mathbf{S}$-selected part of the optimal solution, denoted by $\mathbf{y}_{\mathcal{G}_k}^\star$ where $\mathbf{Sy}^\star = (\mathbf{y}_{\mathcal{G}_1}^{\star\top} \cdots \mathbf{y}_{\mathcal{G}_K}^{\star\top})^\top$, must consist of $\mathbf{z}_{\mathcal{G}_k}$ or $\mathbf{0}$. Incidentally, from the conditions on $\mathcal{G}_1, \ldots, \mathcal{G}_K$ in Definition 1, the cost function can be expressed as

$$\begin{aligned} \|\mathbf{y}-\mathbf{z}\|^2 &= \|\mathbf{S}(\mathbf{y}-\mathbf{z})\|^2 + \|(\mathbf{I}-\mathbf{S})(\mathbf{y}-\mathbf{z})\|^2 \\ &= \sum_{k=1}^{K} \|\mathbf{y}_{\mathcal{G}_k} - \mathbf{z}_{\mathcal{G}_k}\|^2 + \|(\mathbf{I}-\mathbf{S})(\mathbf{y}-\mathbf{z})\|^2, \end{aligned} \quad (18)$$

implying two things: (i) If we set $\mathbf{y}_{\mathcal{G}_k} = \mathbf{0}$ then the cost is increased by $\|\mathbf{z}_{\mathcal{G}_k}\|^2$, and (ii) $(\mathbf{I}-\mathbf{S})\mathbf{y}$ must be the same as $(\mathbf{I}-\mathbf{S})\mathbf{z}$. Hence, from the fact that $\|\mathbf{z}_{\mathcal{G}_{(1)}}\| \geq \|\mathbf{z}_{\mathcal{G}_{(2)}}\| \geq \cdots \geq \|\mathbf{z}_{\mathcal{G}_{(K)}}\|$, we can conclude that setting $\mathbf{y}_{\mathcal{G}_{(1)}} = \mathbf{z}_{\mathcal{G}_{(1)}}, \ldots, \mathbf{y}_{\mathcal{G}_{(\alpha)}} = \mathbf{z}_{\mathcal{G}_{(\alpha)}}$ and $\mathbf{y}_{\mathcal{G}_{(\alpha+1)}} = \mathbf{0}, \ldots, \mathbf{y}_{\mathcal{G}_{(K)}} = \mathbf{0}$ minimizes the cost function $\|\mathbf{y}-\mathbf{z}\|^2$ subject to the inequality constraint $\|\mathbf{Sy}\|_{1,0}^{\mathcal{G}} \leq \alpha$. □

Example 1. To shed more light into Proposition 1, we give a simple example. Let $\mathbf{z} := (1, 3, -2, 8, 6, -4) \in \mathbb{R}^6$, let $\alpha := 1$, let $\mathbf{S} := \mathrm{diag}(1, 1, 1, 0, 1, 1)$, and let $\mathcal{G}_1 := \{1, 2\}$, $\mathcal{G}_2 := \{3, 4\}$ and $\mathcal{G}_3 := \{5, 6\}$ (thus $K = 3$). Then, $\mathbf{z}_{\mathcal{G}_1} = (1, 3)$, $\mathbf{z}_{\mathcal{G}_2} = (-2, 0)$, and $\mathbf{z}_{\mathcal{G}_3} = (6, -4)$. Since $\|\mathbf{z}_{\mathcal{G}_1}\| = \sqrt{10}$, $\|\mathbf{z}_{\mathcal{G}_2}\| = 2$ and $\|\mathbf{z}_{\mathcal{G}_3}\| = 2\sqrt{13}$, we have $(1) = 3$, $(2) = 1$, and $(3) = 2$, i.e., $\mathbf{z}_{\mathcal{G}_{(1)}} = (6, -4)$, $\mathbf{z}_{\mathcal{G}_{(2)}} = (1, 3)$, and $\mathbf{z}_{\mathcal{G}_{(3)}} = (-2, 0)$. Hence, by recalling $\alpha = 1$, the optimal solution is given by

$$\begin{aligned} \mathbf{y}^\star &= (\mathbf{0}^\top\ \mathbf{0}^\top\ \mathbf{z}_{\mathcal{G}_3}^\top)^\top + (\mathbf{I}-\mathbf{S})\mathbf{z} \\ &= (0, 0, 0, 0, 6, -4) + (0, 0, 0, 8, 0, 0) = (0, 0, 0, 8, 6, -4). \end{aligned}$$

Recalling that $\mathbf{B}$ in (7) is a special case of $\mathbf{S}$, the above result states that a closed-form solution to the second subproblem is available as follows: (i) compute $(1), \ldots, (N)$ by sorting the subvectors of $\mathbf{B}(\mathbf{Du}^{(n+1)} - \mathbf{w}^{(n)})$ in terms of their L_2 norms; (ii) substitute zero vectors for the subvectors specified by $\mathcal{G}'_{(\alpha+1)}, \ldots, \mathcal{G}'_{(N)}$; and then (iii) add $(\mathbf{I}-\mathbf{B})(\mathbf{Du}^{(n+1)} - \mathbf{w}^{(n)})$.

Algorithm 1: Proposed algorithm for the L_0 gradient projection

input : $\bar{\mathbf{u}}, \mathbf{v}^{(0)} = \mathbf{w}^{(0)} = \mathbf{D}\bar{\mathbf{u}}$, $\gamma > 0$, and $0 < \eta < 1$
output: $\mathbf{u}^{(n)}$
while $|\mathrm{Grad}_{L_0}(\mathbf{u}^{(n)}) - \alpha| > \varepsilon$ **do**
- $\mathbf{u}^{(n+1)} = \mathbf{F}^*(\mathbf{I} + \gamma^{-1}\mathbf{\Lambda})^{-1}\mathbf{F}(\bar{\mathbf{u}} + \gamma^{-1}\mathbf{D}^\top(\mathbf{v}^{(n)} - \mathbf{w}^{(n)}))$;
- $\mathbf{v}^{(n+1)} = \mathbf{B}(\mathbf{Du}^{(n+1)} - \mathbf{w}^{(n)})$;
- Compute $(1), \ldots, (N)$ by sorting $\mathbf{v}_{\mathcal{G}'_1}^{(n+1)}, \ldots, \mathbf{v}_{\mathcal{G}'_N}^{(n+1)}$ in descending order in terms of their L_2 norms;
- Set $\mathbf{v}_{\mathcal{G}'_{(\alpha+1)}}^{(n+1)} = \mathbf{0}, \ldots, \mathbf{v}_{\mathcal{G}'_{(N)}}^{(n+1)} = \mathbf{0}$ in $\mathbf{v}^{(n+1)}$;
- $\mathbf{v}^{(n+1)} \leftarrow \mathbf{v}^{(n+1)} + (\mathbf{I}-\mathbf{B})(\mathbf{Du}^{(n+1)} - \mathbf{w}^{(n)})$;
- $\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} + \mathbf{Du}^{(n+1)} - \mathbf{v}^{(n+1)}$;
- $\gamma \leftarrow \eta\gamma$;
- $n \leftarrow n+1$;

Consequently, the detailed procedures of our algorithm are summarized in Algorithm 1. In the algorithm, a scalar η is introduced to gradually decreases the value of γ (we recommend $\eta \in [0.95, 0.99]$), which stabilizes ADMM for nonconvex optimization. This is supported by recent convergence analyses of ADMM for nonconvex cases, e.g., [37], [38], where under appropriate conditions, the sequence generated by ADMM converges to a stationary point with sufficiently small γ. Similar strategies are also employed in existing L_0 gradient minimization algorithms [9], [14], [15], [19].

We will show in Sec. IV that $|\mathrm{Grad}_{L_0}(\mathbf{u}^{(n)}) - \alpha|$ decreases monotonically by the algorithm, so that the L_0 gradient value of the output image is expected to be α, where the stopping criterion ε determines the allowable error from α.

Remark 2 (Computational cost of Algorithm 1)**.** Let us discuss the computational cost of our algorithm. At the update of $\mathbf{u}^{(n+1)}$, we can use 2DFFT to solve the matrix inversion efficiently, and thus the cost is $\mathcal{O}(N \log N)$ time. At the update of $\mathbf{v}^{(n+1)}$, the sorting of the L_2 norms of N subvectors is most expensive, which requires $\mathcal{O}(N \log N)$ time. The cost of the update of $\mathbf{w}^{(n+1)}$ is simply $\mathcal{O}(N)$ time. Hence, the cost of each iteration of Algorithm 1 is $\mathcal{O}(N \log N)$ time.

C. Relation to Existing L_0 Gradient Minimization Algorithms

Before moving on to experiments, we briefly discuss how our algorithm is related to existing L_0 gradient minimization algorithms. First, we stress that the update of $\mathbf{v}$ (Proposition 1) is newly established for our L_0 gradient projection framework, and thus in this sense, our algorithm is essentially different from all existing L_0 gradient minimization algorithms, including the ones discussed below.

The algorithm in [14] uses a coordinate descent approach with variable fusion. In [19], the L_0 gradient minimization is solved by a region-based method, which is a state-of-the-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2651392, IEEE Transactions on Image Processing

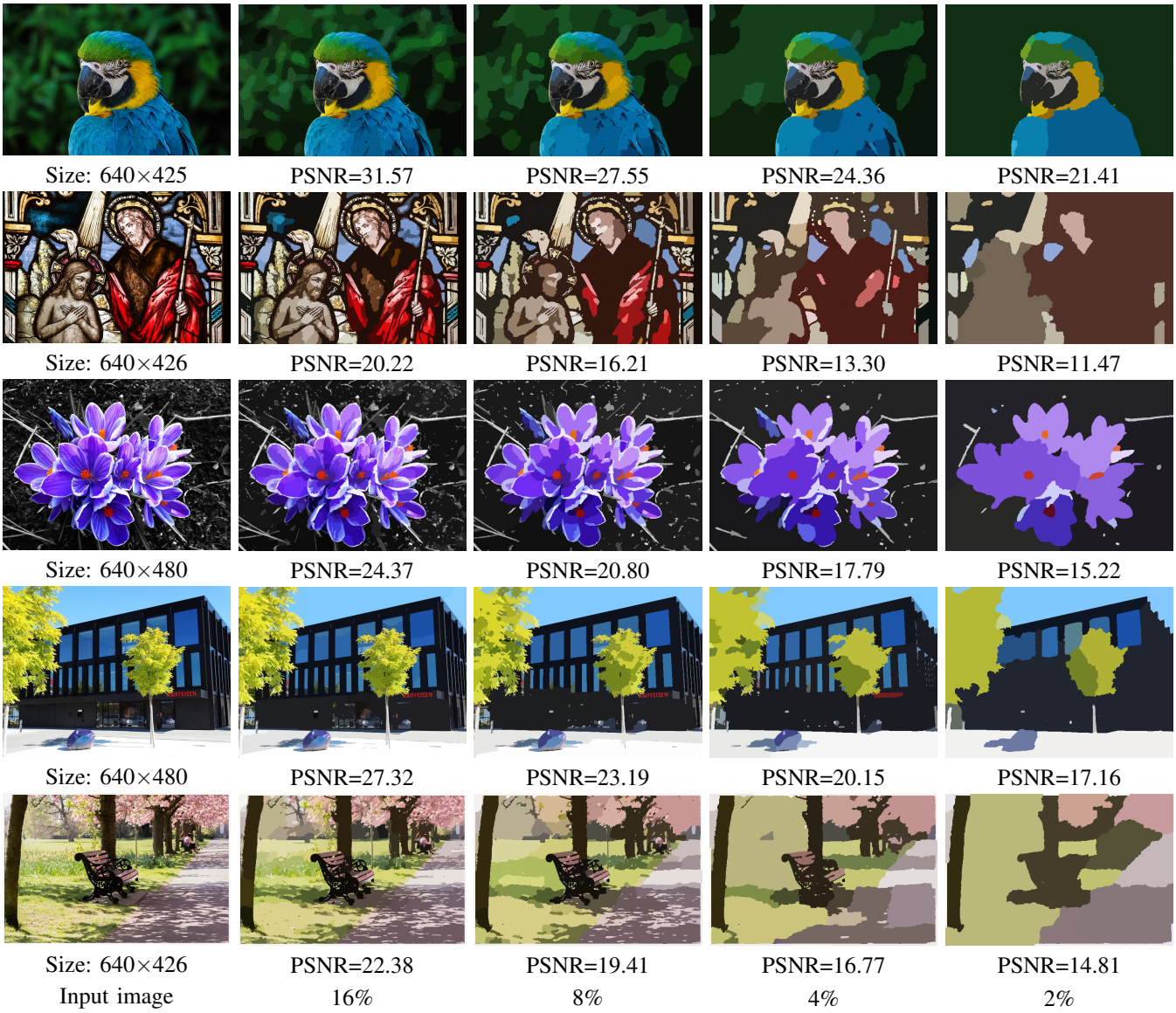


Fig. 3. Input images with their size (the left column), and filtered images generated by L_0 gradient projection with α set to various percentages of the number of pixels (16, 8, 4, and 2%). We show their PSNR[dB] for reference.

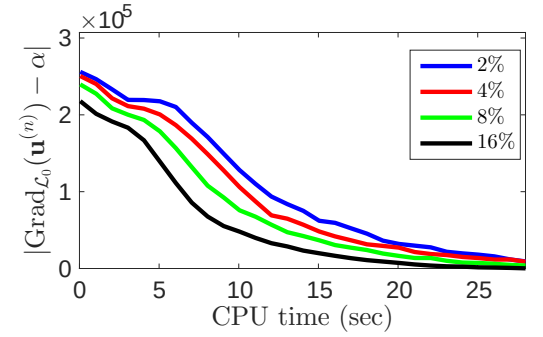
art L_0 gradient minimization algorithm in both terms of efficiency and effectiveness. They employ different optimization techniques from ours.

The original paper of the L_0 gradient minimization [9] adopts a half quadratic optimization approach, which is similar to ADMM without dual variable update but not as much effective as ADMM. The algorithm uses a variable splitting technique similar to ours, i.e., $\mathbf{v} = \mathbf{D}\mathbf{u}$, but it performs a hard-thresholding operation for updating $\mathbf{v}$. On the other hand, at the update of $\mathbf{v}$, our algorithm computes the projection onto the mixed $L_{1,0}$ pseudo-norm ball.

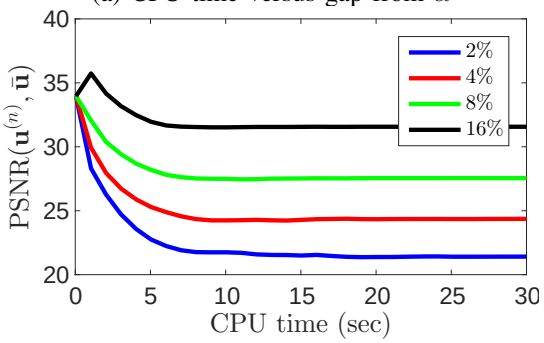
The authors of [15] proposed an ADMM-based algorithm for the L_0 gradient minimization, which is currently the most effective (but relatively slow) for the L_0 gradient minimization, as reported in [19]. Although this algorithm would be most related to our algorithm since both are based on ADMM, this algorithm does not split $\mathbf{D}$, i.e., defining the auxiliary variable not as $\mathbf{v} = \mathbf{D}\mathbf{u}$ but as $\mathbf{v} = \mathbf{u}$ in solving the L_0 gradient minimization (Prob. (5)). As a result, this algorithm requires an inner loop for solving a subproblem via graph cuts, but our algorithm does not. Finally, we note again that this algorithm is also developed for the L_0 gradient minimization and thus is different from our ADMM-based algorithm that uses the computation of the projection onto the mixed $L_{1,0}$ pseudo-norm ball.

IV. Experiments

To illustrate the advantages of our L_0 gradient projection framework, we conducted three experiments. In the first experiment, we investigate the effect of the setting of α to reveal the characteristics of the L_0 gradient projection. We also study the convergence of our algorithm in this experiment. In the second experiment, we compare filtered images and execution time of our method with those of existing L_0 gradient minimization



(a) CPU time versus gap from α



(b) CPU time versus PSNR

Fig. 4. Convergence profiles of our algorithm on computing the filtered images shown in the top row of Fig. 3.

algorithms. In the third experiment, we apply the L_0 gradient projection to two specific problems: clip art JPEG artifact removal and edge extraction.

All the experiments were performed using MATLAB (R2014a, 64bit), on a Windows 10 (64bit) laptop computer with an Intel Core i7 2.6 GHz processor and 8 GB of RAM. We took all test images from a large database of royalty-free images.[3] For the parameters of Algorithm 1, we set $\gamma = 3$, $\eta = 0.97$, and $\varepsilon = 0.0002N$ in all the experiments.

For convenience, we use the well-known (color) peak signal-to-noise ratio (PSNR) as a fidelity metric between two images, instead of the value of the quadratic data-fidelity itself. PSNR[dB] is defined by

$$\mathrm{PSNR}(\mathbf{u}, \bar{\mathbf{u}}) := 10 \log_{10} \frac{3N * 255}{\|\mathbf{u} - \bar{\mathbf{u}}\|^2}, \tag{19}$$

where $\mathbf{u}$ is a target (filtered) image, and $\bar{\mathbf{u}}$ is a reference (input/ground truth) image. The higher PSNR implies the smaller value of the quadratic data-fidelity.

A. Effect of Setting of α in L_0 Gradient Projection

In the first experiment, we study the effect of the setting of α in our method. Specifically, we compare filtered images generated with α set to various percentage of the number of pixels N, where 16%, 8%, 4%, and 2% were examined.

The results are shown in Fig. 3, where for each row, the left is an input image and its size, and the others are the corresponding filtered images and their PSNR (the reference image is the input image). One sees that for every test image, setting α to the same percentage of N (i.e., each column of Fig. 3) generates filtered images of the same degree of flatness, which offers that users set α intuitively in various applications, such as segmentation, character recognition, and color quantization.

In Fig. 4, we give two convergence profiles of Algorithm. 1 on computing the filtered images shown in the top row of Fig. 3. Fig. 4(a) depicts convergence plots of the gap from α, i.e., $|\mathrm{Grad}_{L_0}(\mathbf{u}^{(n)}) - \alpha|$, versus the CPU time. We observe that for every α, the gap decreases monotonically and converges to zero, which guarantees that our algorithm always generates an image satisfying a user-given L_0 gradient value. Fig. 4(b) shows the evolution of PSNR between $\mathbf{u}^{(n)}$ and the input image, i.e., the convergence of the cost function (quadratic data-fidelity), versus the CPU time. One can see that PSNR also converges to a certain value for every α, The reason why PSNR in early iterations is higher than the converged value is that the initial variable $\mathbf{u}^{(0)}$ is set to the input image. These observations demonstrate that our algorithm works well in practice.

B. Comparison with L_0 Gradient Minimization Algorithms

In this second experiment, we compare our results with filtered images generated by existing L_0 gradient minimization algorithms [9], [15], [19].[4] For a fair comparison, we conducted the experiment as follows: for each test image, (i) compute a filtered image by a state-of-the-art L_0 gradient minimization algorithm [19] with certain λ; (ii) measure its L_0 gradient value, denoted by $\alpha_{\min}$; and then (iii) compute a filtered image by our method with $\alpha := \alpha_{\min}$. By this procedure, we can make both filtered images of the same degree of flatness. For the other existing algorithms [9], [15], we adjusted their parameters so that their L_0 gradient values are as close to $\alpha_{\min}$ as possible.

Fig. 5 shows the results, where we also present their L_0 gradient values, PSNR, and execution time (sec). First, we should remark that the region-based algorithm proposed in [19] (implemented in C++) is much faster than the others including our algorithm (implemented in MATLAB), yet our algorithm (Algorithm 1) is second to the region-based algorithm. It is noteworthy that our algorithm achieves the highest PSNR for every test image, under (almost) the same L_0 gradient value (see, each row of Fig. 5). By noticing that the higher PSNR means the smaller value of the quadratic data-fidelity, this observation suggests that our algorithm is more effective in minimizing the sum of the L_0 gradient and the quadratic data-fidelity, i.e., the L_0 gradient minimization, than the existing L_0 gradient minimization algorithms, which is a positive side effect of our algorithm.

C. Applications

As the L_0 gradient minimization, the L_0 gradient projection can be a building block in many applications. In the final

[3]Pixabay https://pixabay.com/en/.

[4]For the implementation of the existing algorithms, we used the source code distributed by the authors of each paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2651392, IEEE Transactions on Image Processing

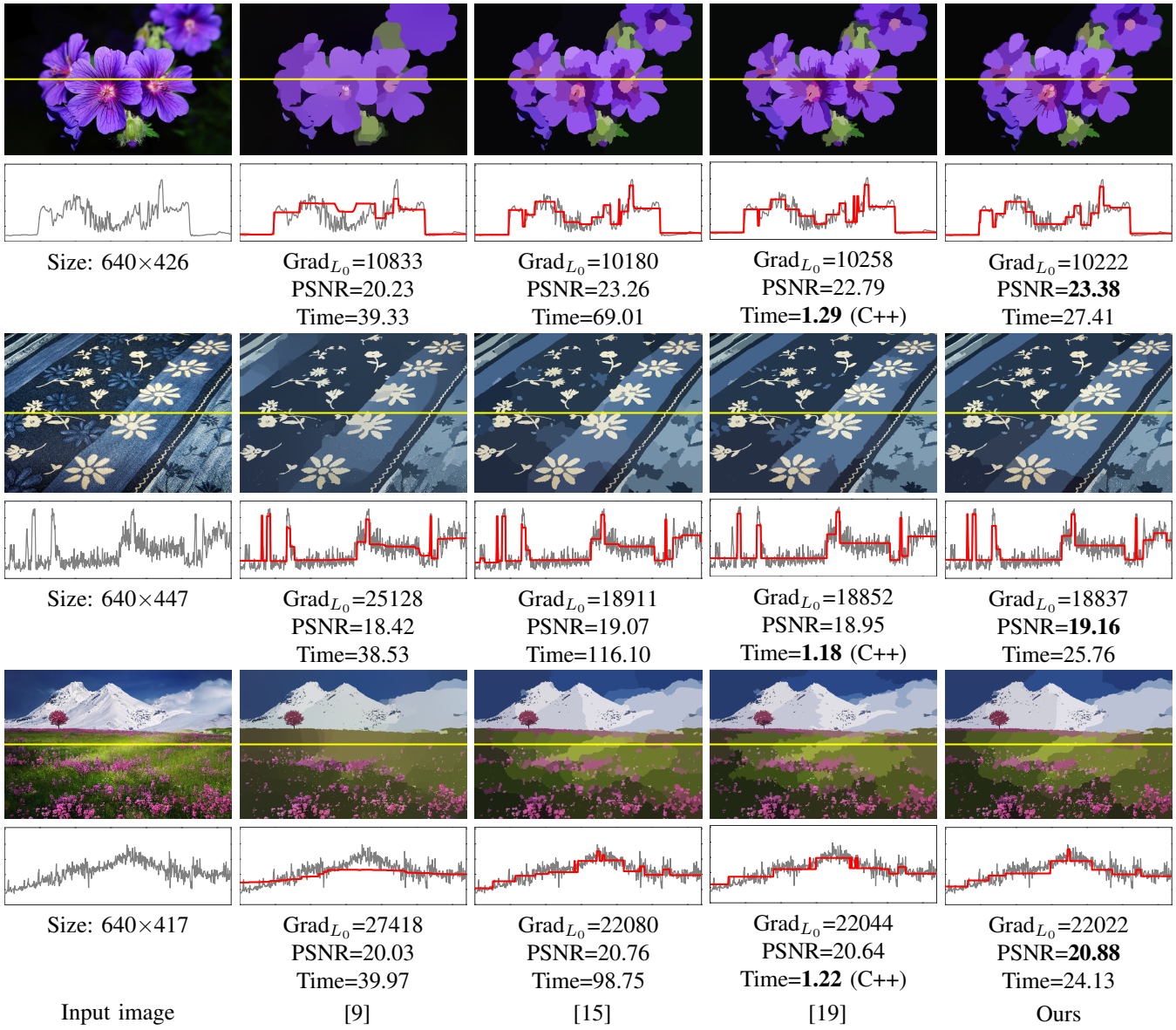


Fig. 5. Comparison of our results and those of existing L_0 minimization algorithms under almost the same L_0 gradient value. For each filtered image, we plot the 1D scanline from its luminance channel, and show the L_0 gradient value, PSNR[dB], and execution time[sec].

experiment, we apply our method to two specific problems: JPEG artifact removal on clip art images and edge extraction, which illustrates the utility of the L_0 gradient projection.

1) Removal of JPEG Artifact in Clip Art Images.: When clip art images, like Fig. 6(a), are compressed by the standard JPEG, typical artifacts, such as block noise and mosquito noise, occur in the compressed images. Since clean clip art images usually consist of flat regions separated by sharp edges, their L_0 gradient values tend to be small. This property is violated by the said artifacts, and thus our method can be used for JPEG artifact removal of clip art images.

We generated JPEG images with severe artifacts, shown in Fig. 6(b) with their PSNR, by compressing the clean clip art images in Fig. 6(a). Then, we applied our method to the JPEG images to obtain artifact-free images, where α was set to 8% of the number of pixels. The results are given in Fig. 6(c) with their PSNR. One can see that blocky artifacts in the JPEG images are removed by our method, resulting in much better visual quality and higher PSNR (NOTE: The PSNR values of JPEG and our results were measured using the ground truth images in Fig. 6(a)).

2) Edge Extraction.: Edge extraction from an image is a fundamental task in many applications. Applying some edge detector to images often generates undesirable results, such as one containing too many discrete edges and/or noise. To stabilize edge detection, the L_0 gradient projection can be used as preprocessing for suppressing details in advance.

We applied the well-known Canny edge detector [39] to original images and filtered images generated by our method, and compare the resulting edge maps. Fig. 7 shows the results on two images, where we set α to 4% and 10% of the number of pixels for the left and right images, respectively. We see

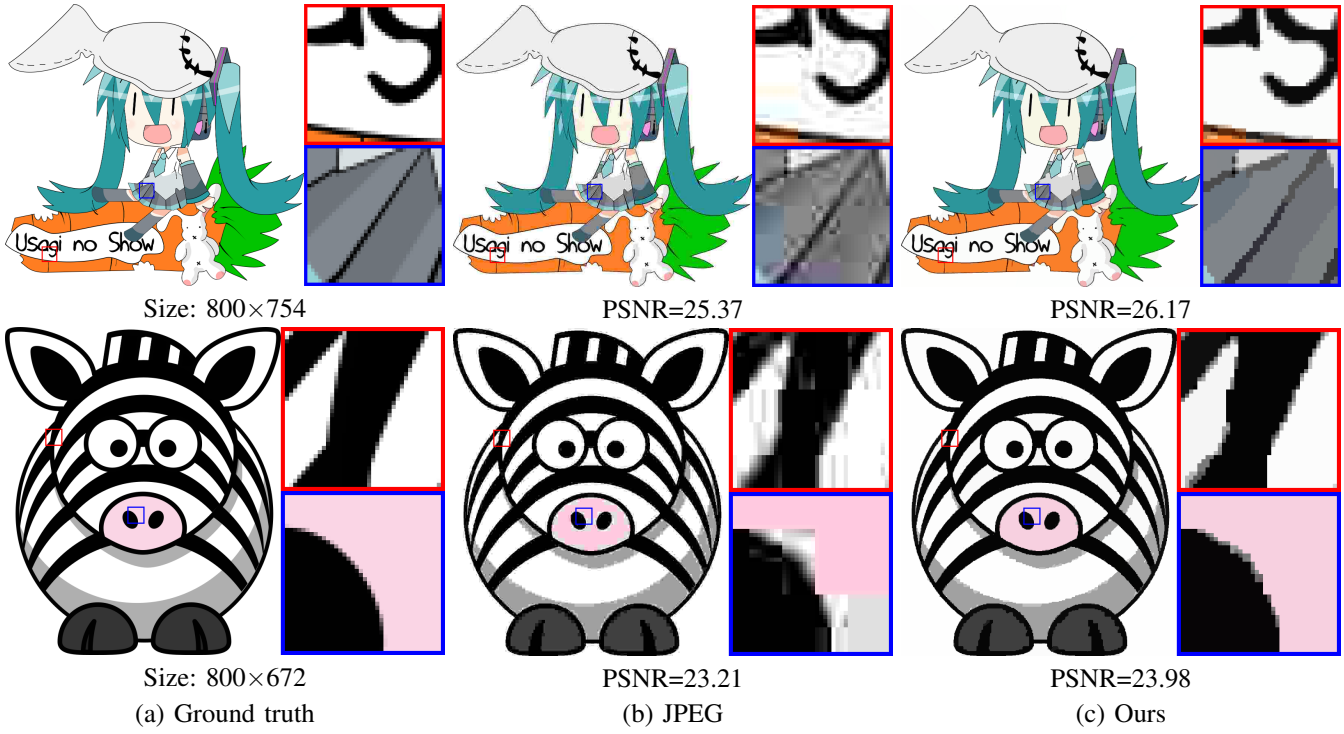


Fig. 6. Results of JPEG artifact removal by our method.

that our method well suppresses details in the original images, which are not needed for edge extraction (e.g., chalk dust). As a result, much better edge detection is achieved.

D. Limitations and Discussions

Although the computational cost of our method described in Remark 2 is not expensive ($\mathcal{O}(N \log N)$ time), our method is slower than the most efficient L_0 gradient minimization algorithm [19] (but faster than the other existing algorithms [9], [15]), as reported in Sec. IV-B. We note that our method was implemented in MATLAB only (and not optimized) for a fair comparison to the algorithms [9], [15], whereas the algorithm [19] was implemented in C++. To compare our method with the algorithm [19] in terms of actual execution time, we also implemented our algorithm in C++, and we observe that the resulting execution times to compute the images in the right column of Fig. 5 are 5.83, 5.45 and 5.14 seconds (from top to bottom), respectively. This indicates that our C++ implementation is about five times faster than the MATLAB counterpart and is not much slower than the C++ implementation of [19]. Hence, we would like to claim that our framework would be preferable in such a situation where users may need to solve the L_0 gradient minimization many times (more than five times) with various λ to achieve a desired degree of flatness.

In some applications, such as segmentation, users may wish to exploit a desired number of *regions*, denoted by r, to determine α. In such a case, although the best value of α is unavailable since it depends on the sizes of the regions (unknown in advance), a relatively good value of α can be set based on r, N_v and N_h (the size of the input image), e.g., $\alpha := \frac{1}{2}(r-1)(N_v + N_h)$. Such a setting is impossible in the L_0 gradient minimization.

V. Concluding Remarks

We have proposed a new edge-preserving filtering method based on the L_0 gradient, named the L_0 gradient projection. In contrast to the L_0 gradient minimization, our L_0 gradient projection framework is very intuitive because one can directly impose a desired L_0 gradient value on the output image.

Our ADMM-based algorithm can compute an approximate solution of the L_0 gradient projection in $\mathcal{O}(N \log N)$ time, where we have established a methodology to handle the hard constraint that the L_0 gradient is less than a user-given parameter α in optimization, by using the projection onto the mixed $L_{1,0}$ pseudo-norm ball with variable splitting. We remark that this technique is not limited to the L_0 gradient projection presented in the current paper, but is also applicable to more complicated formulations that can be solved by ADMM or other proximal splitting methods, which offers another advantage of imposing the L_0 gradient as a hard constraint. For example, let us consider such a situation that one needs to minimize the L_0 gradient term plus two other terms f and g (e.g., considering two input images together in optimization), i.e.,

$$\min_{\mathbf{u}\in\mathbb{R}^{3N}} \lambda_1 \mathrm{Grad}_{L_0}(\mathbf{u}) + \lambda_2 f(\mathbf{u}) + g(\mathbf{u}).$$

In this case, there are two balancing parameters λ_1 and λ_2, and both are interdependent. Thus, the parameter setting becomes much more difficult than the case of a single λ (the standard L_0 gradient minimization formulation). For such cases, imposing

Fig. 7. Results of edge extraction via our method.

the L_0 gradient as a hard constraint would be a particularly good strategy because it can *decouple* the parameters, i.e.,

$$\min_{\mathbf{u}\in\mathbb{R}^{3N}} \mu f(\mathbf{u}) + g(\mathbf{u}) \text{ subject to } \mathrm{Grad}_{L_0}(\mathbf{u}) \leq \alpha,$$

where μ and α can be set independently. The advantage of such strategy is also addressed in the context of hierarchical optimization [40].

We also revealed that our algorithm is more effective than existing L_0 gradient minimization algorithms in the sense of the L_0 gradient minimization. We expect that the L_0 gradient projection framework facilitates the use of L_0 gradient-based image flattening in a wide array of applications.

ACKNOWLEDGMENT

The author is grateful to Keiichiro Shirai for his informative comments on the C++ implementation of our work.

REFERENCES

[1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[2] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, 1998.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 1998, pp. 839–846.

[4] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[5] E. S. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 69:1–69:12, 2011.

[6] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–67:10, 2008.

[7] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[8] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, pp. 89–97, 2004.

[9] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, 2011.

[10] S.-C. Pei, C.-T. Shen, and T.-Y. Lee, "Visual enhancement using constrained L_0 gradient image decomposition for low backlight displays," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 813–816, 2012.

[11] C.-T. Shen, F.-J. Chang, Y.-P. Hung, and S.-C. Pei, "Edge-preserving image decomposition using L1 fidelity with L0 gradient," in *SIGGRAPH Asia 2012 Technical Briefs*, 2012, pp. 6:1–6:4.

[12] L. Xu, S. Zheng, and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 1107–1114.

[13] L. He and S. Schaefer, "Mesh denoising via L_0 minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, 2013.

[14] X. Cheng, M. Zeng, and X. Liu, "Feature-preserving filtering with L_0 gradient minimization," *Computers & Graphics*, vol. 38, pp. 150–157, 2014.

[15] M. Storath, A. Weinmann, and L. Demaret, "Jump-sparse and sparse recovery using potts functionals," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3654–3666, 2014.

[16] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via L_0-regularized intensity and gradient prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2901–2908.

[17] X. Pang, S. Zhang, J. Gu, L. Li, B. Liu, and H. Wang, "Improved L_0 gradient minimization with L_1 fidelity for image smoothing," *PLoS one*, vol. 10, no. 9, 2015, article ID: e0138682.

[18] M. Storath, A. Weinmann, J. Frikel, and M. Unser, "Joint image reconstruction and segmentation using the potts model," *Inverse Problems*, vol. 31, no. 2, 2015, article ID: 025003.

[19] R. Nguyen and M. Brown, "Fast and effective L_0 gradient minimization by region fusion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 208–216.

[20] P. Combettes and J.-C. Pesquet, "Image restoration subject to a total variation constraint," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1213–1222, 2004.

[21] J. M. Fadili and G. Peyré, "Total variation projection with first order schemes," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 657–669, 2011.

[22] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, 2011.

[23] T. Teuber, G. Steidl, and R. H. Chan, "Minimization and parameter estimation for seminorm regularization models with I-divergence constraints," *Inverse Problems*, vol. 29, no. 3, 2013, article ID: 035007.

[24] S. Ono and I. Yamada, "Second-order total generalized variation constraint," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 4938–4942.

[25] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems," *Signal, Image and Video Process.*, vol. 9, no. 8, pp. 1737–1749, 2015.

[26] S. Ono and I. Yamada, "Signal recovery with certain involved convex data-fidelity constraints," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6149–6163, 2015.

[27] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear

variational problems via finite elements approximations," *Comput. Math. Appl.*, vol. 2, pp. 17–40, 1976.

[28] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, pp. 293–318, 1992.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[30] Z. Wen, C. Yang, X. Liu, and S. Marchesini, "Alternating direction methods for classical and ptychographic phase retrieval," *Inverse Problems*, vol. 28, no. 11, 2012, article ID: 115010.

[31] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1701–1709.

[32] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 6201–6205.

[33] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 5135–5143.

[34] B. Zhang, A. Perina, V. Murino, and A. Del Bue, "Sparse representation classification with manifold constraints transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 4557–4565.

[35] S. Ono and I. Yamada, "Color-line regularization for color artifact removal," *IEEE Trans. Comput. Imag.*, vol. 2, no. 3, pp. 204–217, 2016.

[36] S. Ono, "Edge-preserving filtering by projection onto L_0 gradient constraint," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2017, to appear.

[37] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[38] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 2434–2460, 2015.

[39] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[40] S. Ono and I. Yamada, "Hierarchical convex optimization with primal-dual splitting," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 373–388, 2015.

Shunsuke Ono (S'11-M'15) received the B.E. degree in computer science in 2010, and the M.E. and Ph.D. degrees in communications and computer engineering in 2012 and 2014 from the Tokyo Institute of Technology, Tokyo, Japan, respectively.

He is currently an Assistant Professor with Laboratory for Future Interdisciplinary Research of Science and Technology (FIRST), Institute of Innovative Research (IIR), Tokyo Institute of Technology. Since 2016, he has also been a Researcher with Precursory Research for Embryonic Science and Technology (PRESTO), Japan Science and Technology Corporation (JST), Tokyo, Japan. His research interests are in image processing and low-level computer vision, signal processing, and mathematical optimization.

From April 2012 to September 2014, he was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science (JSPS). He has been a Guest Associate Editor of IEICE Transactions on Information and Systems (2016 to present). He received the Excellent Paper Award in 2014 and the Young Researchers' Award in 2013 from the IEICE; the Outstanding Student Journal Paper Award from IEEE Signal Processing Society (SPS) Japan Chapter in 2014; the Yasujiro Niwa Outstanding Paper Award from Tokyo Denki University in 2015; and the TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2016. He is a member of IEICE.