Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **18**, No. 1, pp. 267-295

DOI: 10.4208/aamm.OA-2023-0315
February 2026

An Effective ℓ_p-Nonconvex Regularization Method for Image Smoothing

Biao Fang[1], Xiao-Guang Lv[2,*], Guo-Liang Zhu[1,3], Le Jiang[2] and Jia-Qi Mei[1]

[1] *School of Electronic Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu 222005, China*
[2] *School of Science, Jiangsu Ocean University, Lianyungang, Jiangsu 222005, China*
[3] *Ocean College, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China*

Received 7 December 2023; Accepted (in revised version) 11 March 2024

Abstract. Image smoothing techniques are widely used in computer vision and graphics applications. In this paper, we present an ℓ_p-nonconvex regularization method for image smoothing. To induce sparsity prior of the smoothed images more strongly than the ℓ_1 norm regularization, we take the nonconvex arctangent penalty function of the image gradient as the regularization term. To make the proposed model more flexible and effective for different image smoothing applications, we use the ℓ_p norm function as the fidelity term, instead of the ℓ_2 norm function. The powerful majorization-minimization (MM) algorithm is employed for the proposed nonconvex optimization model. The convergence of the resulting MM algorithm is discussed. Comprehensive experiments and comparisons show that the proposed method is effective in various image processing tasks such as texture smoothing, detail enhancement, artifact removal, image denoising, high dynamic range (HDR) tone mapping, edge detection, and image composition.

AMS subject classifications: 94A08, 68U10, 90C26

Key words: Image smoothing, ℓ_p-nonconvex, regularization, majorization-minimization, convergence.

1 Introduction

The purpose of image smoothing is to eliminate unwanted details while maintaining the important edges. As a significant image processing technique, image smoothing is

*Corresponding author.
Emails: biaofang2021@163.com (B. Fang), xiaoguanglv@jou.edu.cn (X. Lv), guoliangzhu2021@163.com (G. Zhu), jiangle@jou.edu.cn (L. Jiang), mjq1333907@163.com (J. Mei)

 ©2026 Global Science Press

widely used in computer vision and graphics applications such as texture smoothing, detail enhancement, artifact removal, image denoising, high dynamic range (HDR) tone mapping, edge detection, and image composition [1–7]. In recent years, image smoothing methods have received considerable attention from the image processing research community.

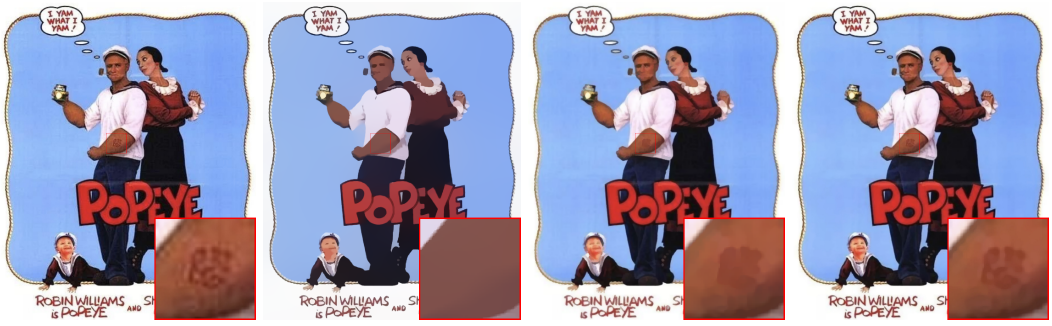
Generally, edge-preserving methods are used to smooth out the small details while preserving or sharpening the strong edges. One of the most classical edge-preserving methods is the bilateral filter (BF) [8–10]. The BF is widely used in various smoothing tasks due to its simplicity and effectiveness in removing noise-like structures. However, as pointed out by He et al. in [11], the BF tends to produce undesirable gradient reversal artifacts at the edges. A guided filter was proposed by He et al. in [11] to alleviate this problem and obtain high-quality images. To reduce the running cost of BF, Zhang et al. in [12] designed a fast weighted median filter that runs more than 100 times faster than BF. In [13], to improve the ability of BF-based filters to extract the multi-scale details, Farbman et al. introduced the weighted least squares (WLS) optimization method. Although the WLS method alleviates many problems, it is insufficient in capturing fine-scale textures. To remedy this deficiency, Subr et al. in [14] explored the local extreme model to extract fine-scale features. This model captures textures based on the assumption that fine-scale details are rapidly oscillating information between minima and maxima. In [15], Gastal et al. considered a domain transformation method for addressing the gradient reversal problem by using one-dimensional space to accelerate two-dimensional edge-aware filtering. In [16], Xu et al. used the ℓ_0 gradient minimization (ℓ_0GM) method to enforce the gradient sparsity of the smoothed images. In [17], Paris et al. designed a local Laplacian filter for edge-aware image smoothing. The critical step is to characterize the edges with a simple threshold on the pixel values so that large-scale edges can be distinguished from small-scale details. In [18], Ham et al. developed a static and dynamic (SD) filter to handle structural inconsistencies between the guidance and input images. To improve the problem of over-sharpening caused by excessive regularization parameter in the ℓ_0-norm method, Liu et al. in [19] adopted the gradient sparsity and surface area (GSSA) minimization scheme. In [20], Liu et al. proposed an effective iterative least squares (ILS) for edge-preserving image smoothing. The implementation of the ILS algorithm is close to real-time due to the use of the Fourier transform during the iterative process. In [21], Zhu et al. built a benchmark for edge-preserving image smoothing from the point of both qualitative and quantitative performance evaluation. The reasonable edge-preserving filter shouldn't blur or over-sharpen the edges that define object boundaries or other important features, while smoothing the areas between these edges. Unfortunately, such a filter does not exist, as it is generally not possible to accurately determine which edges should be retained.

Structure-preserving smoothing methods are usually applied to preserve the salient structures of the restored images. In [22], Xu et al. developed the relative total variation (RTV) measures to accomplish structure extraction from texture images. In [23], Karacan et al. employed a novel approach based on patch-analysis. Specifically, they estimated

the similarity between two image patches within an adaptive filtering framework by considering the region covariance matrix of color, orientation and intensity. Inspired by Karacan et al.'s work, Jeon et al. in [24] investigated a new constraint model for capturing textures in intrinsic image decomposition. In [25], Guen presented the total variational-ℓ_1 (TV-ℓ_1) model for some structure-preserving smoothing tasks such as texture removal and structure extraction. In [26], Zhang et al. studied a rolling guidance filter (RGF) based on scale-space theory to smooth structures at different scales. In [27], Zhu et al. adopted an effective patch-group-based nonlocal mean framework for structure-preserving image smoothing. In [28], Lee et al. introduced a novel adaptive smoothing image gradient model to smooth out the textures of images. In this model, texture components are distinguished from structural edges and smoothly changing shadows based on the interval gradient operator. In [29], Guo et al. proposed a model that jointly considers structure discovery and texture removal (SDT). The principle of SDT is that the structure edge detector and texture smoother are mutually beneficial. In [30], Fan et al. designed an unsupervised learning framework to achieve the leading smoothing results. In fact, the framework imposes different forms of regularization on diverse images through a spatially adaptive ℓ_p flattening criterion. In [31], Zhao et al. utilized two local activity-driven filtering for structure-preserving image smoothing. The local activity-driven anisotropic diffusion with two edge-stopping functions is used to mitigate coding artifacts of depth images, while the local activity-driven relative total variation with the way of the product between gradient and the local activity measurement is applied for achieving image smoothing and scale representation. In [32], Cao et al. proposed a hybrid ℓ_0-H^{-1} optimization model for structure-preserving image smoothing. The model measures the sparsity of scale-aware gradients by the ℓ_0 norm and the structure fidelity by the H^{-1} norm with an embedded Laplace operator. Some deep learning methods have been proposed for structure-preserving image smoothing. We refer the reader to [33, 34] and references therein. These structure-preserving methods explicitly smooth out the textures in the image. However, it fails to maintain the important edge information. It also produces a slight staircasing artifact near edge boundaries, which affects the quality of the processed image.

Besides, a series of edge-preserving and structure-preserving image smoothing methods have also been proposed recently. In [35], Cai et al. presented a novel image smoothing method to eliminate textures at any level while preserving other image content via relativity-of-Gaussian. Experimental results have shown that the technique has the capability of smoothing the input image without distorting edges and structures. In [36], Liu et al. proposed a generalized optimization framework with a truncated Huber (TH) penalty function for edge-preserving and structure-preserving image smoothing. Due to its strong flexibility, the framework is able to achieve diverse smoothing natures. In [37], Cao et al. utilized the relative bilateral filter as a local regularization to capture and suppress weak large-scale textures from salient structures and edges.

In the mentioned literature, we observe that most researchers usually employ the Euclidean distance between the low-level feature vectors associated with each pixel as an



(a) Input (b) ℓ_0GM (c) TV-ℓ_1 (d) Ours

Figure 1: Comparison of smoothing effects by different algorithms. (a) The input image; (b)-(d) The smoothing images by using ℓ_0GM, TV-ℓ_1 and the proposed model respectively.

affinity function. However, this measure of similarity has its own limitations, since it does not consider any higher-level structure. To alleviate this problem, in this paper, we propose a novel smoothing method based on the ℓ_p-nonconvex minimization model. To induce sparsity prior of the smoothed images more strongly than the ℓ_1 norm regularization, we take the nonconvex arctangent penalty function of the image gradient as the regularization term. To make the proposed model more flexible and effective, we use the ℓ_p norm function as the fidelity term, instead of the ℓ_2 norm function. The proposed nonconvex method offers good possibilities for restoring images with homogeneous regions surrounded by neat edges. In Fig. 1, we show an example of the smoothing effect using ℓ_0GM [16], TV-ℓ_1 [25] and our method. Fig. 1(a) is the input image and Figs. 1(b)-(d) are the smoothing results generated by ℓ_0GM, TV-ℓ_1 and our method, respectively. Although the ℓ_0GM is widely used in image smoothing, it inevitably suffers from the over-sharpening effect if the regularization parameter is large. In particular, from the zooming-in area of Fig. 1(b), we observe a severe color shift phenomenon due to over-sharpening. Fig. 1(c) shows that the TV-ℓ_1 method is a significant improvement over the ℓ_0GM approach. However, it does not preserve the image structure at a coarse scale well, as shown in the enlarged part of Fig. 1(c). It is clear from Fig. 1(d) that the proposed method can eliminate the undesired details while preserving the weak edges of prominent structures. Our smoothed image looks more satisfactory than the others.

The main contributions of this paper include the following aspects.

(i) We propose an effective ℓ_p-nonconvex regularization method for the image smoothing problem, which is rarely discussed in image smoothing before. The parameter p controls the edge-preserving ability and contrast of the smoothed image, and different choice of p in the regularization term reveals different smoothing behaviors.

(ii) We employ the powerful majorization-minimization (MM) algorithm to solve the proposed nonconvex optimization model and prove the convergence of the MM algorithm.

(iii) We demonstrate the effectiveness of the proposed method in several image processing tasks, including texture smoothing, detail enhancement, artifact removal, image denoising, HDR tone mapping, edge detection, and image composition.

The rest of this paper is organized as follows. In Section 2, we establish the ℓ_p-nonconvex minimization model. In Section 3, we solve the proposed nonconvex optimization model with the MM algorithm and prove the convergence of the resulting MM algorithm. In Section 4, we provide a comprehensive comparison between our method and the state-of-the-art methods in various applications. In Section 5, we give a short conclusion.

2 Proposed model

In this paper, we investigate the following ℓ_p-nonconvex minimization model for the image smoothing problem:

$$\min_u J(u) = \Phi_{\text{fid}}(u) + \lambda \Psi_{\text{reg}}(u), \tag{2.1}$$

where $\Phi_{\text{fid}}(u)$ represents the fidelity term and $\Psi_{\text{reg}}(u)$ represents the regularization term, $u \in \mathbb{R}^{MN}$ represents the smoothing image of size $M \times N$ to be recovered and $\lambda > 0$ is a regularization parameter which controls the strength of the smoothing. A larger regularization parameter will lead to more smoothing, as the penalty for large weights becomes more significant, while a smaller regularization parameter will result in less smoothing, allowing the optimization model to fit more closely to the original data.

For the fidelity term, we adopt the following ℓ_p-norm function

$$\Phi_{\text{fid}}(u) = \frac{1}{p}\|u-f\|_p^p = \frac{1}{p}\sum_{i=1}^{MN} \varphi_p((u-f)_i) \tag{2.2}$$

with

$$\varphi_p(x) = |x|^p, \tag{2.3}$$

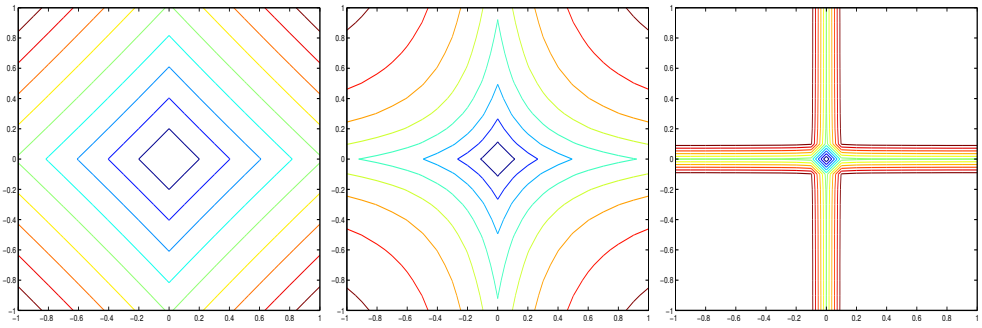
where $f \in \mathbb{R}^{MN}$ represents the input image of size $M \times N$ and $p \in (0,1]$. Clearly, the ℓ_p-norm with $0 < p < 1$ is nonsmooth and nonconvex, the ℓ_p-norm with $p = 1$ is nonsmooth and convex.

For the regularization term, we choose the nonconvex and nonsmooth arctangent penalty function. The arctangent penalty function can induce sparsity more strongly than the total variation regularization [38]. More precisely, the arctangent penalty term is given by

$$\Psi_{\text{reg}}(u) = \sum_{i=1}^{MN} s_i(g) \cdot \psi_a(\|\nabla_i u\|_2), \tag{2.4}$$

where

$$s_i(g) = e^{-\frac{\|\nabla_i g\|_2^2}{2\sigma^2}} \tag{2.5}$$



(a) $a=0.02$ (b) $a=2$ (c) $a=200$

Figure 2: Level lines of the arctangent penalty function with different a-values. (a)-(c) The parameter a is set to be 0.02, 2 and 200 respectively.

with a guidance image g and a positive range parameter σ, and

$$\psi_a(x)=\frac{3}{\pi}\left(\operatorname{atan}\left(\frac{1+2a|x|}{\sqrt{3}}\right)-\frac{\pi}{6}\right) \tag{2.6}$$

with a given positive concavity parameter a [39]. The parameter a in (2.4) is used to control the sparsity of the image gradient. Here

$$\|\nabla_i g\|_2^2=(\nabla_i^h g)^2+(\nabla_i^v g)^2 \quad \text{and} \quad \|\nabla_i u\|_2=\sqrt{(\nabla_i^h u)^2+(\nabla_i^v u)^2},$$

where ∇_i^h and ∇_i^v correspond to the linear operators of the first-order differences along the horizontal and vertical directions at pixel i, respectively. When the guidance image is a zero matrix, the proposed model becomes a non-guided method in [40]. This manuscript can be regarded as the extension of its conference version.

In Fig. 2, we plot the level lines of the arctangent penalty function $\psi_a(x)$ with different a-values. Obviously, when a is small, the penalty function has a similar result as the ℓ_1-norm. Increasing a can obtain an approximation of the ℓ_0-norm.

It is obvious that the proposed model (2.1) with the fidelity term (2.2) and the regularization term (2.4) is coercive for any $u\in\mathbb{R}^{MN}$. Since the objective function $J(u)$ in (2.1) is continuous and bounded below, the optimal set

$$\hat{U}\doteq\left\{\hat{u}:\ J(\hat{u})=\inf_{u\in\mathbb{R}^{MN}}J(u)\right\}$$

is nonempty for any input image f.

3 Numerical algorithm

Many algorithms such as orthogonal matching pursuit [41], iterative hard thresholding [42], ℓ_1 greedy [43], parallel iteration [44, 45] and MM [46] have been devised to

compute the sparse solutions of nonconvex optimization problems. In this paper, we employ the powerful MM method to solve the proposed nonconvex model. The MM method for minimizing (2.1) has the form

$$u^{(t+1)} = \arg\min_u \mathrm{Q}(u;u^{(t)}), \tag{3.1}$$

where $\mathrm{Q}(u;w)$ is an upper bound function of $J(u)$. Usually, the MM method requires that each function $\mathrm{Q}(u;w)$ satisfies the following properties [46,47]:

(i) $\mathrm{Q}(u;w)$ is quadratic in u;

(ii) $\mathrm{Q}(u;w) \geq J(u)$ for all $u \in \mathbb{R}^{MN}$;

(iii) $\mathrm{Q}(w;w) = J(w)$ and $\partial_u \mathrm{Q}(w;w) = J'(w)$, where ∂_u represents the first-order partial derivative with respect to the first variable u;

(iv) $\partial_{uu}\mathrm{Q}(u;w)$ is independent of u, where ∂_{uu} represents the second-order partial derivative with respect to the first variable u.

According to properties (iii) and (iv), $J(u)$ must be a continuously differentiable function. For this reason, we need to modify the fidelity term and the regularization term separately to make the function $J(u)$ continuously differentiable. For the fidelity term, a well-known smoothed version of the function φ_p is given by

$$\varphi_{p,\beta}(x) = \left(x^2+\beta\right)^{\frac{p}{2}} \quad \text{for } p \in (0,1], \tag{3.2}$$

where β is a small positive parameter. For the regularization term, a smoothed version of the function ψ_a is given by

$$\psi_{a,\gamma}(x) = \frac{3}{\pi}\left(\operatorname{atan}\left(\frac{1+2a\left(x^2+\gamma\right)^{\frac{1}{2}}}{\sqrt{3}}\right) - \frac{\pi}{6}\right), \tag{3.3}$$

where γ is a small positive parameter.

For the use of the MM method, we replace the original nonsmooth ℓ_p-nonconvex minimization problem (2.1) with the following smoothed minimization problem

$$\min_u J_{\beta,\gamma}(u) = \frac{1}{p}\sum_{i=1}^{MN} \varphi_{p,\beta}((u-f)_i) + \lambda \sum_{i=1}^{MN} s_i(g)\cdot \psi_{a,\gamma}(\|\nabla_i u\|_2). \tag{3.4}$$

For the MM method, it generally includes two main steps: a majorization step which generates an upper bound function of the objective function, and a minimization step which determines a minimizer of this upper bound function.

3.1 Majorization step

Let $\phi(x)$ be a continuously differentiable function, we can construct a quadratic form of the upper bound function $\mathrm{q}(x;z)$ of $\phi(x)$ as

$$\begin{aligned}\mathrm{q}(x;z)&=\frac{\phi'(z)}{2}\left(\frac{x^2}{z}-z\right)+\phi(z)\\&=\frac{1}{2}\cdot\frac{\phi'(z)}{z}\cdot x^2+C,\end{aligned}\tag{3.5}$$

where C represents a constant independent of x [48].

Using (3.5), we get the upper bound function of $\varphi_{p,\beta}((u-f)_i)$ at the point $u^{(t)}$ as

$$\begin{aligned}\mathrm{q}((u-f)_i;(u^{(t)}-f)_i)&=\frac{1}{2}\cdot\frac{\varphi'_{p,\beta}((u^{(t)}-f)_i)}{(u^{(t)}-f)_i}\cdot(u-f)_i^2+C\\&=\frac{p}{2}\cdot\varsigma_i^{(t)}\cdot(u-f)_i^2+C,\end{aligned}\tag{3.6}$$

where

$$\varsigma_i^{(t)}=((u^{(t)}-f)_i^2+\beta)^{\frac{p-2}{2}}.$$

Similarly, we obtain the upper bound function of $\psi_{a,\gamma}(\|\nabla_i u\|_2)$ at the point $u^{(t)}$ as

$$\begin{aligned}\mathrm{q}(\|\nabla_i u\|_2;\|\nabla_i u^{(t)}\|_2)&=\frac{1}{2}\cdot\frac{\psi'_{a,\gamma}(\|\nabla_i u^{(t)}\|_2)}{\|\nabla_i u^{(t)}\|_2}\cdot\|\nabla_i u\|_2^2+C\\&=\frac{1}{2}\cdot\varpi_i^{(t)}\cdot\|\nabla_i u\|_2^2+C,\end{aligned}\tag{3.7}$$

where

$$\varpi_i^{(t)}=\frac{3\sqrt{3}a}{2\pi((\|\nabla_i u^{(t)}\|_2^2+\gamma)^{\frac{1}{2}}+a(\|\nabla_i u^{(t)}\|_2^2+\gamma)+a^2(\|\nabla_i u^{(t)}\|_2^2+\gamma)^{\frac{3}{2}})}.$$

With the notations

$$\nabla=\begin{bmatrix}\nabla^h\\ \nabla^v\end{bmatrix},\quad \nabla^h u=[\nabla_1^h u,\nabla_2^h u,\cdots,\nabla_{MN}^h u]^{\mathrm{T}}\quad\text{and}\quad \nabla^v u=[\nabla_1^v u,\nabla_2^v u,\cdots,\nabla_{MN}^v u]^{\mathrm{T}},$$

we obtain the upper bound function $\mathrm{Q}_{\beta,\gamma}(u;u^{(t)})$ for $J_{\beta,\gamma}(u)$ at the point $u^{(t)}$ as

$$\begin{aligned}\mathrm{Q}_{\beta,\gamma}(u;u^{(t)})&=\frac{1}{p}\sum_{i=1}^{MN}\mathrm{q}((u-f)_i;(u^{(t)}-f)_i)+\lambda\sum_{i=1}^{MN}s_i(g)\cdot\mathrm{q}\left(\|\nabla_i u\|_2;\|\nabla_i u^{(t)}\|_2\right)\\&=\frac{1}{2}(u-f)^{\mathrm{T}}\Lambda^{(t)}(u-f)+\frac{\lambda}{2}\left(\begin{bmatrix}\nabla^h\\ \nabla^v\end{bmatrix}u\right)^{\mathrm{T}}\underbrace{\begin{bmatrix}\Gamma^{(t)}&0\\0&\Gamma^{(t)}\end{bmatrix}}_{W^{(t)}}\left(\begin{bmatrix}\nabla^h\\ \nabla^v\end{bmatrix}u\right)+C\\&=\frac{1}{2}(u-f)^{\mathrm{T}}\Lambda^{(t)}(u-f)+\frac{\lambda}{2}u^{\mathrm{T}}\nabla^{\mathrm{T}}W^{(t)}\nabla u+C,\end{aligned}\tag{3.8}$$

where

$$\Lambda^{(t)}=\operatorname{diag}\left(\varsigma_i^{(t)}\right) \quad \text{and} \quad \Gamma^{(t)}=\operatorname{diag}\left(s_i(g)\cdot\varpi_i^{(t)}\right)$$

are the diagonal $MN\times MN$ matrix, and C is all terms independent of u.

3.2 Minimization step

According to (3.1), $u^{(t+1)}$ is the minimizer of $\mathrm{Q}_{\beta,\gamma}(u;u^{(t)})$. Since (3.8) is a quadratic function, u can be solved by the linear problem

$$\left(\Lambda^{(t)}+\lambda\nabla^{\mathrm{T}}W^{(t)}\nabla\right)u^{(t+1)}=\Lambda^{(t)}f. \tag{3.9}$$

With the Sherman-Morrison-Woodbury formula [49], the solution of the linear problem (3.9) can be rewritten as

$$u^{(t+1)}=f-\left(\Lambda^{(t)}\right)^{-1}\nabla^{\mathrm{T}}\left(\frac{1}{\lambda}\left(W^{(t)}\right)^{-1}+\nabla\left(\Lambda^{(t)}\right)^{-1}\nabla^{\mathrm{T}}\right)^{-1}\nabla f. \tag{3.10}$$

This iteration can be implemented at a low computational cost, since the resulting matrix in (3.10) is tridiagonal.

In summary, we obtain the following algorithm for image smoothing.

Algorithm 3.1 Image smoothing via the ℓ_p-nonconvex model.

Input: an input image f, a guidance image g, λ, a, p, β and γ.
Output: a smoothing image u.

Initialization: $u^{(0)}$
for $t=1,2,\cdots,$ **do**
(1). construct $\Lambda^{(t)}$ with $\varsigma_i^{(t)}$;
(2). construct $\Gamma^{(t)}$ and $W^{(t)}$ with $\varpi_i^{(t)}$ and $s_i(g)$;
(3). compute $u^{(t+1)}$ by (3.10);
end for

In the following, we analyze the convergence of Algorithm 3.1.

Theorem 3.1. *Let the sequence $\{u^{(t)}\}$ be generated by Algorithm* 3.1*, then the sequence $\{J_{\beta,\gamma}(u^{(t)})\}$ is monotonically decreasing and convergent.*

Proof. It is easy to show that

$$J_{\beta,\gamma}(u^{(t)})=\mathrm{Q}_{\beta,\gamma}(u^{(t)};u^{(t)})\geq\mathrm{Q}_{\beta,\gamma}(u^{(t+1)};u^{(t)})\geq J_{\beta,\gamma}(u^{(t+1)}). \tag{3.11}$$

The first equality and the last inequality come from the properties of the upper bound function. The second inequality holds, since $u^{(t+1)}$ is the minimizer of $\mathrm{Q}_{\beta,\gamma}(u;u^{(t)})$. From (3.4), we know that $\{J_{\beta,\gamma}(u^{(t)})\}$ is bounded from below by zero. Therefore, the sequence $\{J_{\beta,\gamma}(u^{(t)})\}$ is monotonically decreasing and convergent. □

Theorem 3.2. *Let the sequence* $\{u^{(t)}\}$ *be generated by Algorithm* 3.1. *Then, for any initial solution* $u^{(0)} \in \mathbb{R}^{MN}$, *we have*

$$\lim_{t\to\infty} \left\| u^{(t)} - u^{(t+1)} \right\|_2 = 0.$$

Proof. Since $Q_{\beta,\gamma}(u;u^{(t)})$ is quadratic with respect to u, we take Taylor expansion of $Q_{\beta,\gamma}(u;u^{(t)})$ at $u^{(t+1)}$ as

$$\begin{aligned} Q_{\beta,\gamma}(u;u^{(t)}) =& Q_{\beta,\gamma}(u^{(t+1)};u^{(t)}) + \left\langle \partial_u Q_{\beta,\gamma}(u^{(t+1)};u^{(t)}), u-u^{(t+1)} \right\rangle \\ &+ \frac{1}{2}(u-u^{(t+1)})^{\mathrm{T}} \partial_{uu} Q_{\beta,\gamma}(u^{(t+1)};u^{(t)})(u-u^{(t+1)}). \end{aligned} \tag{3.12}$$

It is straightforward to get $\partial_u Q_{\beta,\gamma}(u^{(t+1)};u^{(t)}) = 0$, since $u^{(t+1)}$ is the minimizer of $Q_{\beta,\gamma}(u;u^{(t)})$. According to (3.8), we have

$$\partial_{uu} Q(u^{(t+1)};u^{(t)}) = \Lambda^{(t)} + \lambda \nabla^{\mathrm{T}} W^{(t)} \nabla.$$

Obviously, the matrix $\partial_{xx} Q(u^{(t+1)};u^{(t)})$ is symmetric positive definite with the smallest eigenvalue τ more than zero. By taking $u = u^{(t)}$ in (3.12), we get

$$Q_{\beta,\gamma}(u^{(t)};u^{(t)}) \geq Q_{\beta,\gamma}(u^{(t+1)};u^{(t)}) + \frac{\tau}{2} \left\| u^{(t)} - u^{(t+1)} \right\|_2^2. \tag{3.13}$$

Together with (3.11), it then follows that

$$\begin{aligned} J_{\beta,\gamma}(u^{(t)}) - J_{\beta,\gamma}(u^{(t+1)}) \geq& Q_{\beta,\gamma}(u^{(t)};u^{(t)}) - Q_{\beta,\gamma}(u^{(t+1)};u^{(t)}) \\ \geq& \frac{\tau}{2} \left\| u^{(t)} - u^{(t+1)} \right\|_2^2. \end{aligned} \tag{3.14}$$

We immediately obtain that

$$\begin{aligned} \sum_{t=0}^{\infty} \left\| u^{(t)} - u^{(t+1)} \right\|_2^2 \leq& \frac{2}{\tau} \sum_{t=0}^{\infty} (J_{\beta,\gamma}(u^{(t)}) - J_{\beta,\gamma}(u^{(t+1)})) \\ =& \frac{2}{\tau} (J_{\beta,\gamma}(u^{(0)}) - J^*_{\beta,\gamma}), \end{aligned} \tag{3.15}$$

where $J^*_{\beta,\gamma}$ denotes the limit point of the convergent sequence $\{J_{\beta,\gamma}(u^{(t)})\}$. Therefore, we obtain that

$$\lim_{t\to\infty} \left\| u^{(t)} - u^{(t+1)} \right\|_2 = 0.$$

This completes the proof. □

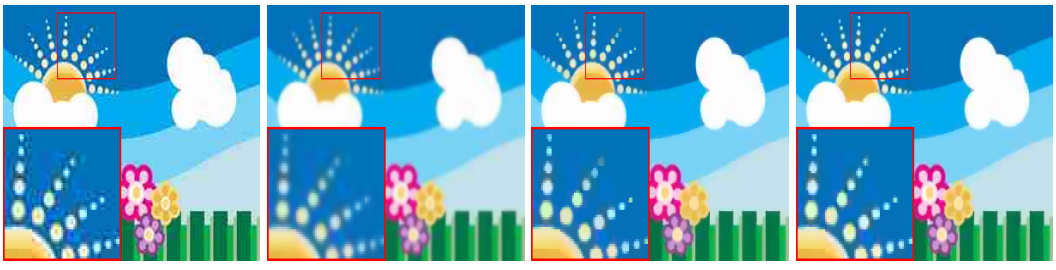
4 Experiments and comparisons

In the section, we compare the proposed image smoothing approach with the highly related methods, including BF [9], WLS [13], ℓ_0GM [16], SD [18], GSSA [19], ILS [20], RTV [22], TV-ℓ_1 [25], RGF [26], ℓ_0-H^{-1} [32] and TH [36]. Note that for the SD, GSSA, ILS, RTV, TV-ℓ_1, and TH methods, we follow the parameter setting suggested by the authors in the original papers. For the remaining comparative methods, we carefully tune the parameters to obtain the best results. All of the numerical experiments were carried out in Matlab R2021a. The results were obtained by running the Matlab codes on an Intel(R) Core(TM) i7-8700 CPU (3.70 GHz, 3.70 GHz) computer with RAM of 16 GB.

In the tests, the guidance image g is obtained by the BF method with the empirical parameter. In (2.5), we choose the range parameter $\sigma=0.5$ to control the sensitivity to edges. Both β and γ in $J_{\beta,\gamma}(u)$ are set to be 10^{-4} in the proposed method. The L-curve, discrepancy principle and generalized cross-validation methods are commonly used techniques for selecting the regularization parameter λ in image processing [50,51]. In this paper, we empirically choose the value of the regularization parameter λ for satisfactory smoothing performance.

4.1 Effect of guidance image g

We first compare the difference of the smoothing behaviors without or with a guidance image. In Fig. 3, we show the image smoothing results with different methods. Fig. 3(a) depicts a degraded image containing severe compression artifacts. Fig. 3(b) displays a smoothing image produced by the BF method. Figs. 3(c) and (d) show the smoothing results obtained by the proposed approach without and with the guidance image, respectively. For the proposed method, $p=0.8$, $\lambda=3$ and $a=3$ are selected in Figs. 3(c) and (d). We take the filtered image of the BF method as a guidance image in Fig. 3(d). It is clear from Fig. 3 that the restored result by the proposed method with the guidance image is better than the restored results by the BF method and the proposed method without



(a) Input (b) BF (c) Without guidance (d) With guidance

Figure 3: Effect of guidance image. (a) The input image; (b)-(d) The smoothing images by using BF, the proposed model without and with the guidance image respectively.

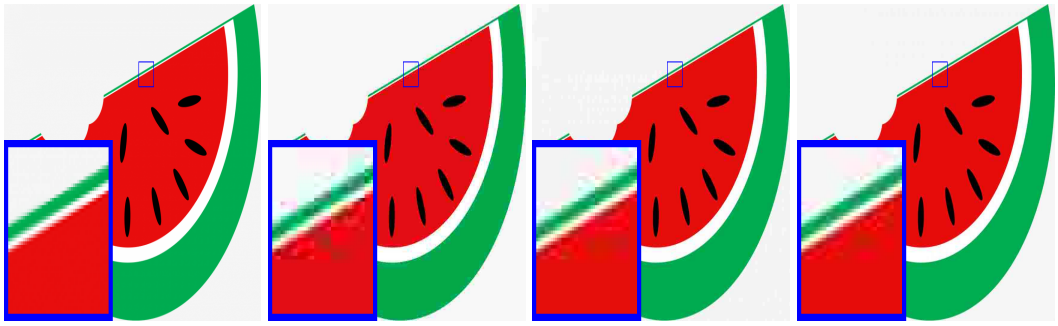
the guidance image.

4.2 Convergence behavior

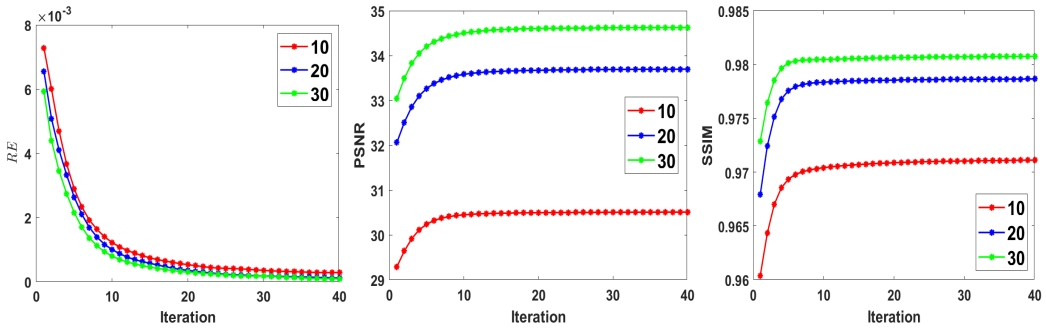
To show the convergence behavior of the proposed ℓ_p-nonconvex regularization method for image smoothing, we report the relative error (RE), peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) versus each iteration for the artifact removal problem. The RE between two successive iterations is defined by

$$RE = \frac{\left\| u^{(t)} - u^{(t-1)} \right\|_2}{\left\| u^{(t-1)} \right\|_2}.$$

In Fig. 4, we plot the numerical convergence results of RE, PSNR and SSIM for different quality factors. The degraded images in Figs. 4(b)-(d) are generated by compressing the original image in Fig. 4(a) with a quality factor of 10, 20 and 30, respectively. In Figs. 4(e)-(g), we show the RE, PSNR and SSIM results by the proposed method. It can be seen



(a) Input (b) Quality factor 10 (c) Quality factor 20 (d) Quality factor 30



(e) RE (f) PSNR (g) SSIM

Figure 4: Convergence behavior. (a) The input image; (b)-(d) The degraded images with a compression quality factor of 10, 20 and 30 respectively; (e)-(g) The RE, PSNR and SSIM values are given by the proposed method respectively.

from Figs. 4(e)-(g) that the proposed method is convergent within 20 iterations. In the following experiments, the iterations for the proposed algorithm are terminated when the maximum number of outer iterations $Maxit=20$ has been carried out or $RE<10^{-3}$.

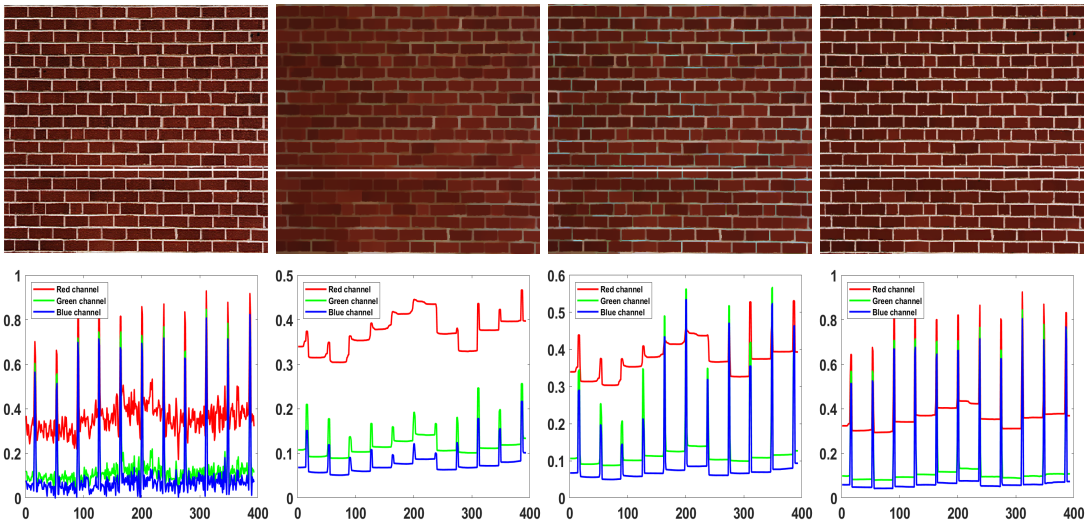
4.3 Effect of parameters p and a

We compare the effects of the smoothing behavior using the proposed method for different values of p and a. In Fig. 5, we show the smoothing results for different p-values. Fig. 5(a) shows the input image and Figs. 5(b)-(d) display the smoothing results using the proposed ℓ_p-nonconvex model with $p=0.8$, 1, 2, respectively. It is noted that the discussion and results about the nonconvex regularization methods with $p=2$ can be found in [52,53]. As discussed in [54], when $p=0.8$ and 1, the smoothing images in Figs. 5(b)-(c) are composed of partly of constant patches and partly of pixels that fit data samples exactly. From Figs. 5(b)-(d), we observe that $p=0.8$, 1, 2 achieve effective piecewise constant visual restoration results with edge preservation. As expected, the differences between neighboring pixels in homogeneous regions are smaller than a small threshold, while they are larger than a large threshold at edges. It is not difficult to see from Fig. 5 that the proposed model maintains the local contrast and edges of the smoothing image precisely when p is less than or equal to 1, while the proposed model lays stress on a neat classification of differences as belonging to smooth regions or to edges when $p=2$, as seen in the circles in the zoomed-in region of Figs. 5(b)-(d). So when $p\leq 1$, the proposed method has the potential to exhibit a better performance in edge and contrast preservation.

In Fig. 6, we show the smoothing results for different a-values. The input image and its 262^{nd} row in RGB channels are given in Fig. 6(a). We display the smoothing results and the one-dimensional representation of their RGB channels for $a=0.1$, 1 and 10 in Figs. 6(b)-(d), respectively. As can be seen from Fig. 6(b), smaller a tends to generate piecewise smoothing results but with lower pixel values (similar to the ℓ_1-norm). From Figs. 6(c)-(d), we observe that the larger a yields more piecewise constant results (similar to the ℓ_0-norm). In this case, the proposed model effectively removes low-amplitude

(a) Input (b) $p=0.8$ (c) $p=1$ (d) $p=2$

Figure 5: Smoothing results for different p-values. (a) The input image; (b)-(d) The parameter p is set to be 0.8, 1 and 2 respectively.



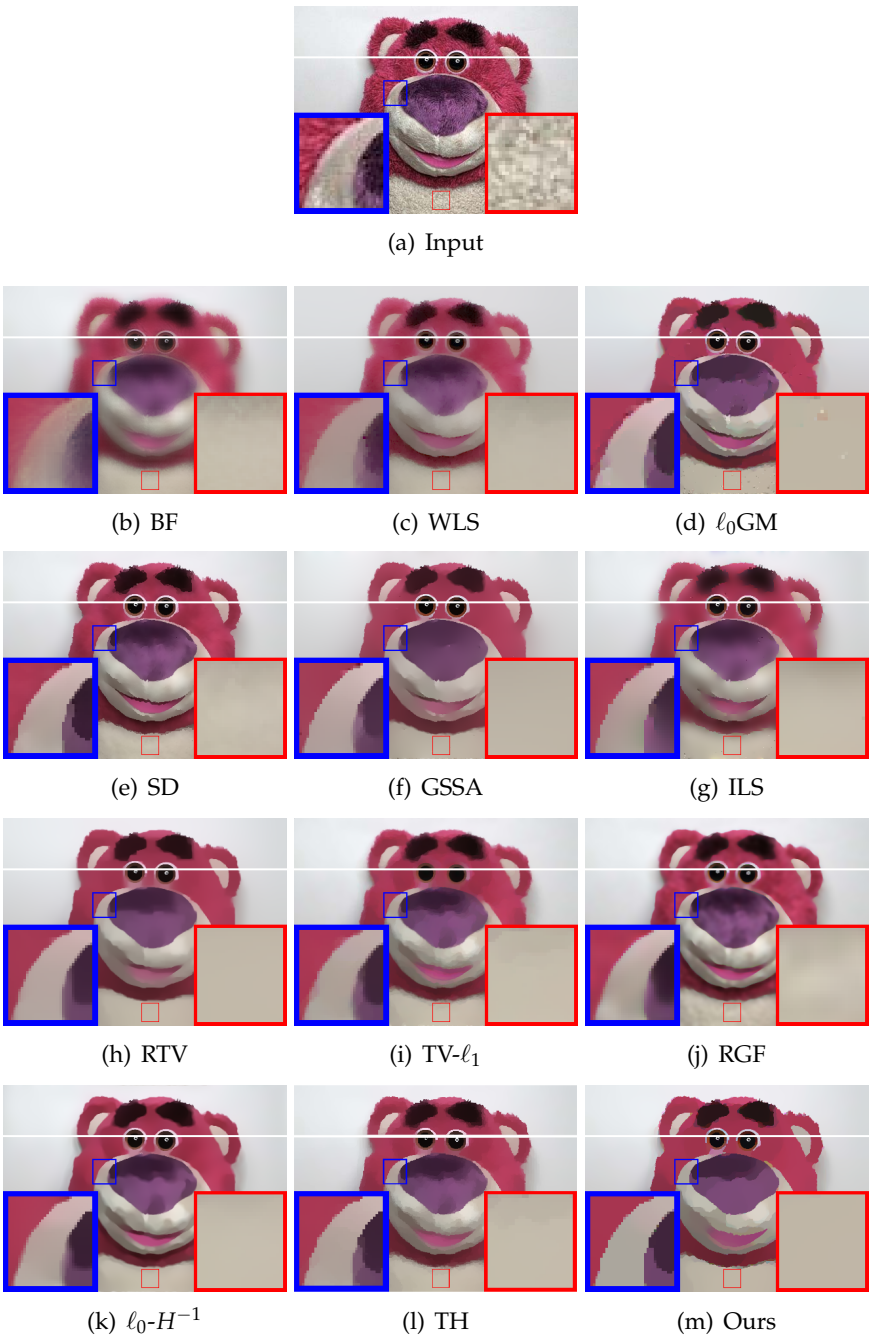
(a) Input (b) $a=0.1$ (c) $a=1$ (d) $a=10$

Figure 6: Smoothing results for different a-values. (a) The input image and its 262^{nd} row in RGB channels; (b)-(d) The smoothing images and the one-dimensional representation of their RGB channels for $a=0.1$, 1 and 10 respectively.

structure and sharpens the edges. In the proposed model, the choice of a-value depends on what kind of smoothing properties we desire. In the following tests, the parameters of the proposed method are carefully adjusted to obtain the best visual results.

4.4 Texture smoothing

Many human-created artworks contain plenty of textures, such as graffiti and fluff toys. The task of texture smoothing is to simultaneously extract important structures and smooth out meaningless textures. We employ the BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH methods and the proposed method to solve the texture smoothing problem. Fig. 7 depicts the smoothing results generated by the competing methods. For the proposed method, the satisfactory result is achieved by $p=1$, $\lambda=7.5$ and $a=7.5$. The original image is displayed in Fig. 7(a). It contains many fluff textures and noise-like structures. The smoothing images obtained by the competing methods are shown in Figs. 7(b)-(m), respectively. It is easy to see from Fig. 7(b) that the BF method yields a blurry restored image and doesn't preserve the significant edges. The WLS method eliminates most of the texture successfully, but the smoothed image still looks blurred, as shown in Fig. 7(c). As displayed in Fig. 7(d), the ℓ_0GM method succeeds in preserving the boundaries of the object. However, some noise-like structures and color artifacts can be observed in the enlarged parts, possibly due to the limitations of the ℓ_0 regularization used in the model. We observe from the zoomed-in regions of Fig. 7(e) that the SD



(a) Input

(b) BF (c) WLS (d) ℓ_0GM

(e) SD (f) GSSA (g) ILS

(h) RTV (i) TV-ℓ_1 (j) RGF

(k) ℓ_0-H^{-1} (l) TH (m) Ours

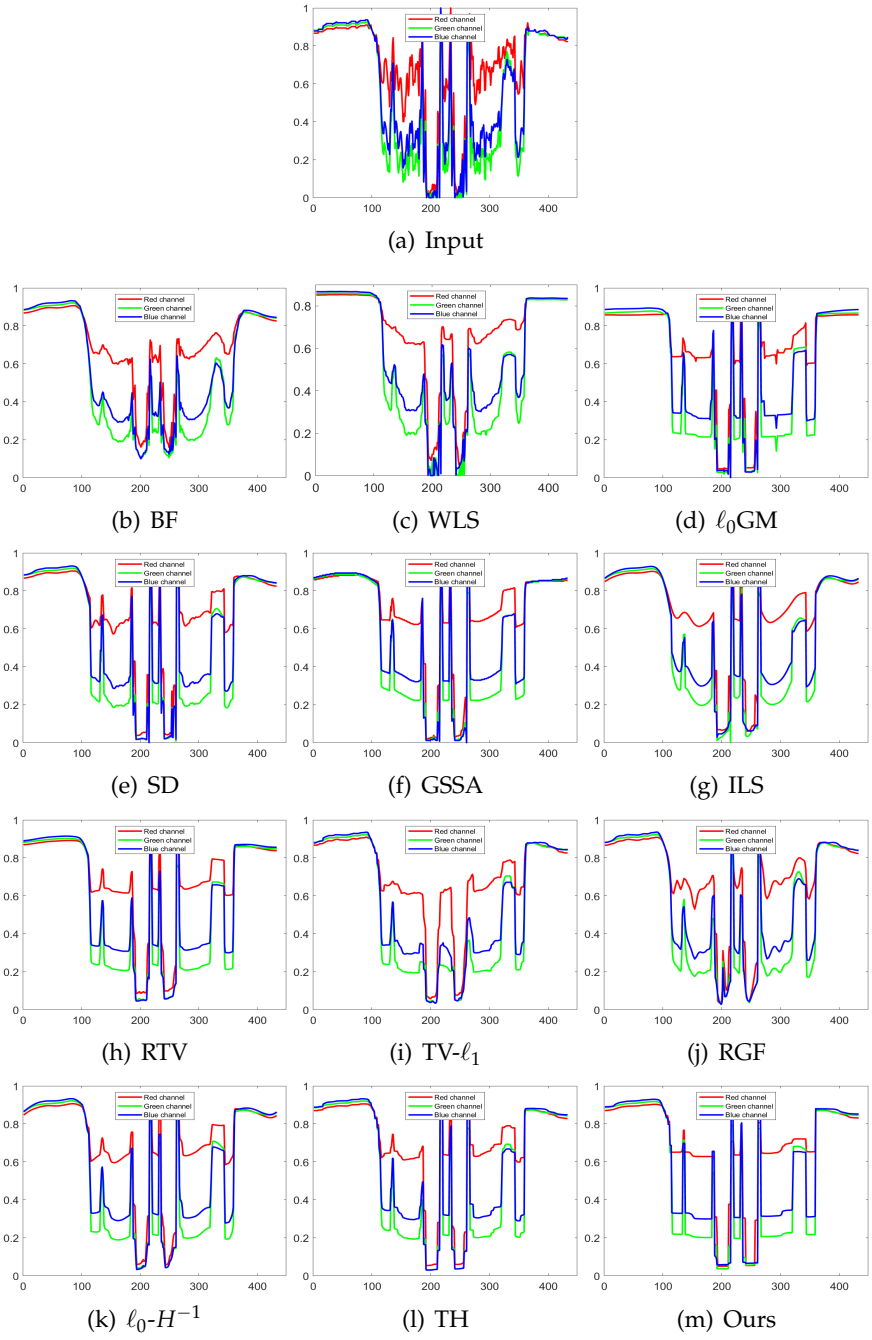
Figure 7: Comparison of texture smoothing. (a) The input image; (b)-(m) The smoothing images by using BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and the proposed model respectively.

approach removes the fluff textures without artifacts, but the surface of the restored image is not enough piecewise smoothing. It can be observed from Fig. 7(f) that the GSSA method removes small-scale textures. But the GSSA method blurs the edges, which may be caused by the process of surface area minimization. The result in Fig. 7(g) reveals that the ILS approach can completely remove the fluff textures. Unfortunately, some blocky artifacts are present in the right magnified region. The RTV method is also unsatisfactory in terms of edge-preserving capability, as illustrated in Fig. 7(h). It can be seen from the enlarged areas of Figs. 7(i) and (l) that both TV-ℓ_1 and TH techniques perform well in removing textures. However, the color shift phenomenon and unnatural artifacts can be observed in their results. From Fig. 7(j), we can see that the RGF method can maintain the edges well but cannot smooth out the textures effectively. The ℓ_0-H^{-1} method produces the smoothed image with no visible texture, but lacks in edge preservation, as shown in Fig. 7(k). In contrast to the other methods, the proposed algorithm is able to remove the fluff textures sufficiently while generating neater edges than others, as depicted in Fig. 7(m).

To objectively compare the smoothing capabilities of various algorithms, we extract the 78^{th} row of each smoothed image in RGB channels and display them in Fig. 8. It should be noted that the pixel values have been normalized to the interval [0, 1]. Fig. 8(a) is one-dimensional representation of the RGB channels of the input image. It is easy to see that the observed image has many steep regions. From Fig. 8(b), we notice that the RGB channels have a relatively narrow transition. This phenomenon implies that the output of BF tends to be blurred. As shown in Figs. 8(c)-(e) and (j), the results generated by WLS, ℓ_0GM, SD and RGF have more or less steep areas. Some textures are clearly seen in the smoothed images. We observe from Fig. 8(f) that the GSSA model produces lower pixel values than the others and the entire smoothed image looks strange. From Fig. 8(g), we notice that there are sharp areas in the RGB channels. It suggests that the result of ILS is over-sharpened. It is easy to see from Fig. 8(i) that the difference in pixel values between the RGB channels is small, which indicates the TV-ℓ_1 result with low-contrast. Without zooming in, we see from Figs. 8(h) and (k)-(m) that our RGB channels are more piecewise constant than the other three methods. Combining visual results and RGB channels, the proposed algorithm is superior to other related algorithms in texture smoothing.

4.5 Detail enhancement

An image usually contains a base layer and a detail layer. For detail enhancement, one commonly used method is to boost the detail layer to increase contrast. In Fig. 9, we show a comparison between the proposed method and other related smoothing filters for detail enhancement. The input image is shown in Fig. 9(a). At a glance, we have difficulty in distinguishing details from the petals. The enhanced results by BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and our method are displayed in Figs. 9(b)-(m), respectively. In our method, we choose $p = 0.8$, $\lambda = 5$ and $a = 1.5$ to obtain the enhancement result. In this test, the detail layer is boosted by three. As shown in



(a) Input

(b) BF (c) WLS (d) ℓ_0GM

(e) SD (f) GSSA (g) ILS

(h) RTV (i) TV-ℓ_1 (j) RGF

(k) ℓ_0-H^{-1} (l) TH (m) Ours

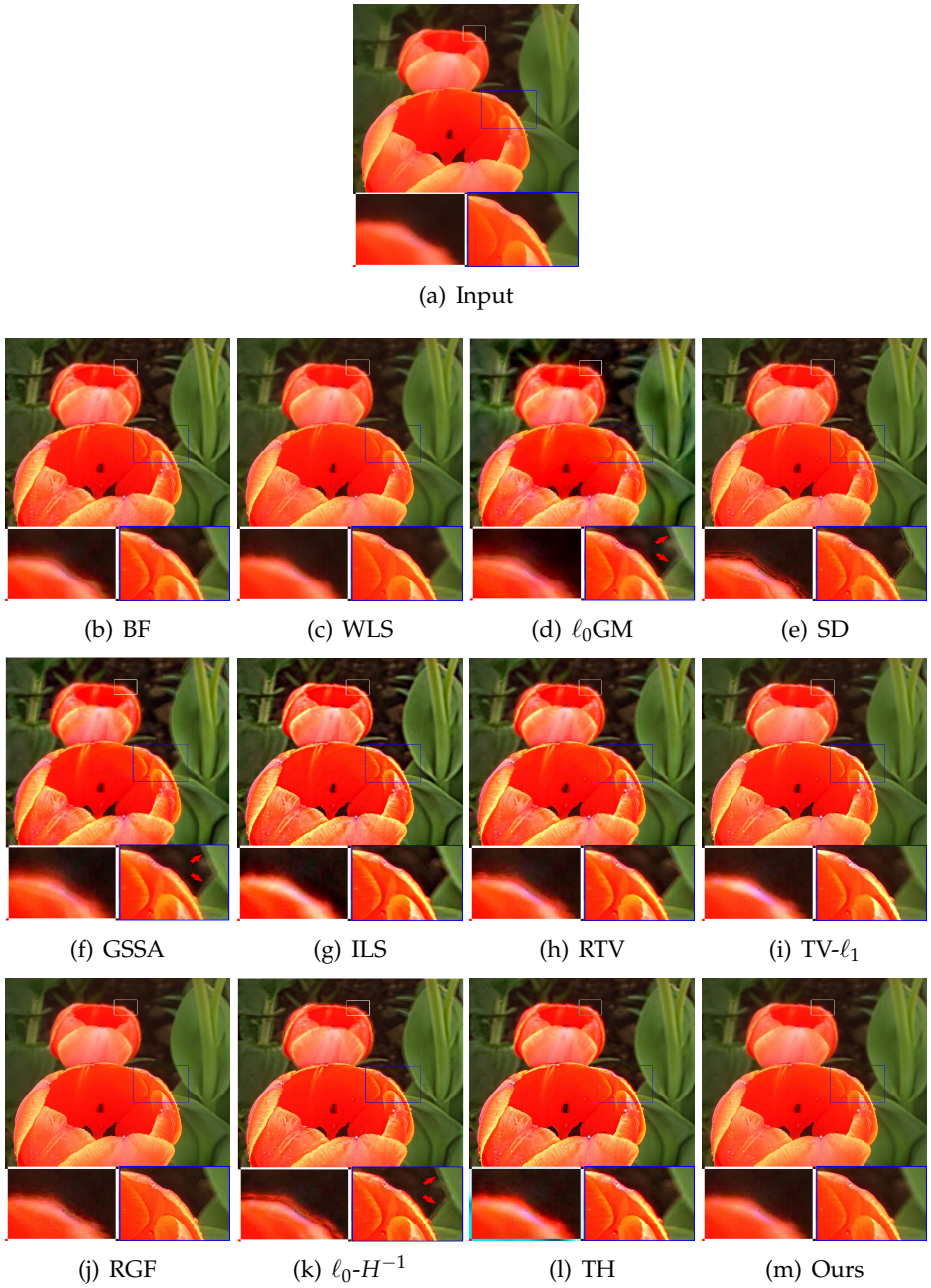
Figure 8: Comparison of the RGB channels of the 78^{th} row of every smoothed image in Fig. 7.

Fig. 9(b), the BF technique leads to slight gradient reversal artifacts at the boundary. We notice from Figs. 9(d) and (f) that the ℓ_0GM and GSSA models relatively alleviate the gradient reversal artifacts. However, the compartmentalization artifacts exist in the left zoom-in patches, and halo artifacts are unexpectedly introduced at the edges of leaves (labeled with the red arrows). It can be seen from Fig. 9(e) that the SD filter produces more visible gradient reversal artifacts than other methods. From Fig. 9(g), we observe that the ILS result has rich details, but suffers from conspicuous gradient reversal artifacts at the edges. We notice from the magnified regions of Figs. 9(h) and (l) that there exist slight gradient reversal artifacts in the RTV and TH results. As can be seen from the enlarged parts of Fig. 9(i), the TV-ℓ_1 method results in a severe color shift phenomenon, which is caused by edge sharpening on the corresponding base layer. From Fig. 9(j), we observe that there are apparent compartmentalization artifacts in the result of RGF. In Fig. 9(k), we display the enhancement result created by ℓ_0-H^{-1}, where both gradient reversals and halos can be observed (labeled with the red arrows). Figs. 9(c) and (m) provide similar results with little visible artifacts. Both WLS and our algorithms provide superior performance in handling gradient reversals and halos. In conclusion, our and WLS output images are visually more attractive than other enhanced images.

4.6 Artifact removal

When an original image is compressed in JPEG format with low-quality, compression artifacts usually appear. In Fig. 10, we show the results of JPEG compression artifacts removal by different methods. Fig. 10(a) is a high-quality input image. The degraded image in Fig. 10(b) is obtained by compressing the original image in Fig. 10(a) with a quality factor of 30. It can be seen that the compressed image exists visible artifacts and color changes at the edges. The artifact removal images by all comparative methods are given in Figs. 10(c)-(n). For the proposed method, $p = 0.8$, $\lambda = 7$ and $a = 8$ are selected for a satisfactory result. We enlarge some details of the restored images for a better visual comparison. From the enlarged part of Fig. 10(c), it is easy to see that the BF method can remove most compression artifacts, but some artifacts still exist. In Fig. 10(d), we notice that the WLS result has some blurred edges and a slight color shift phenomenon, which is due to a trade-off between the ability to remove compression artifacts and maintain edges. As can be seen from Figs. 10(e) and (g)-(h), the ℓ_0GM, GSSA and ILS methods fail to eliminate the artifacts completely. Although the RTV, TV-ℓ_1, RGF and TH methods perform well in suppressing JPEG artifacts, they do not successfully preserve the edges (labeled with the red arrows), as shown in the enlarged parts of Figs. 10(i)-(k) and (m). From Figs. 10(f), (l) and (n), we observe that SD, ℓ_0-H^{-1} and our results are similar in that all simultaneously eliminate visible artifacts and alleviate the color shift problem.

To show the superiority of the proposed method quantitatively, we consider the artifact removal of the compressed images with quality factors of 10, 20, 30, $\cdots$, 90. We use the PSNR and SSIM values between the restored and original images as measures to evaluate the restoration results. The higher the value of SSIM and PSNR are, the better the



(a) Input (b) BF (c) WLS (d) ℓ_0GM (e) SD (f) GSSA (g) ILS (h) RTV (i) TV-ℓ_1 (j) RGF (k) ℓ_0-H^{-1} (l) TH (m) Ours

Figure 9: Comparison of detail enhancement. (a) The input image; (b)-(m) The enhanced images by using BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and the proposed model respectively.

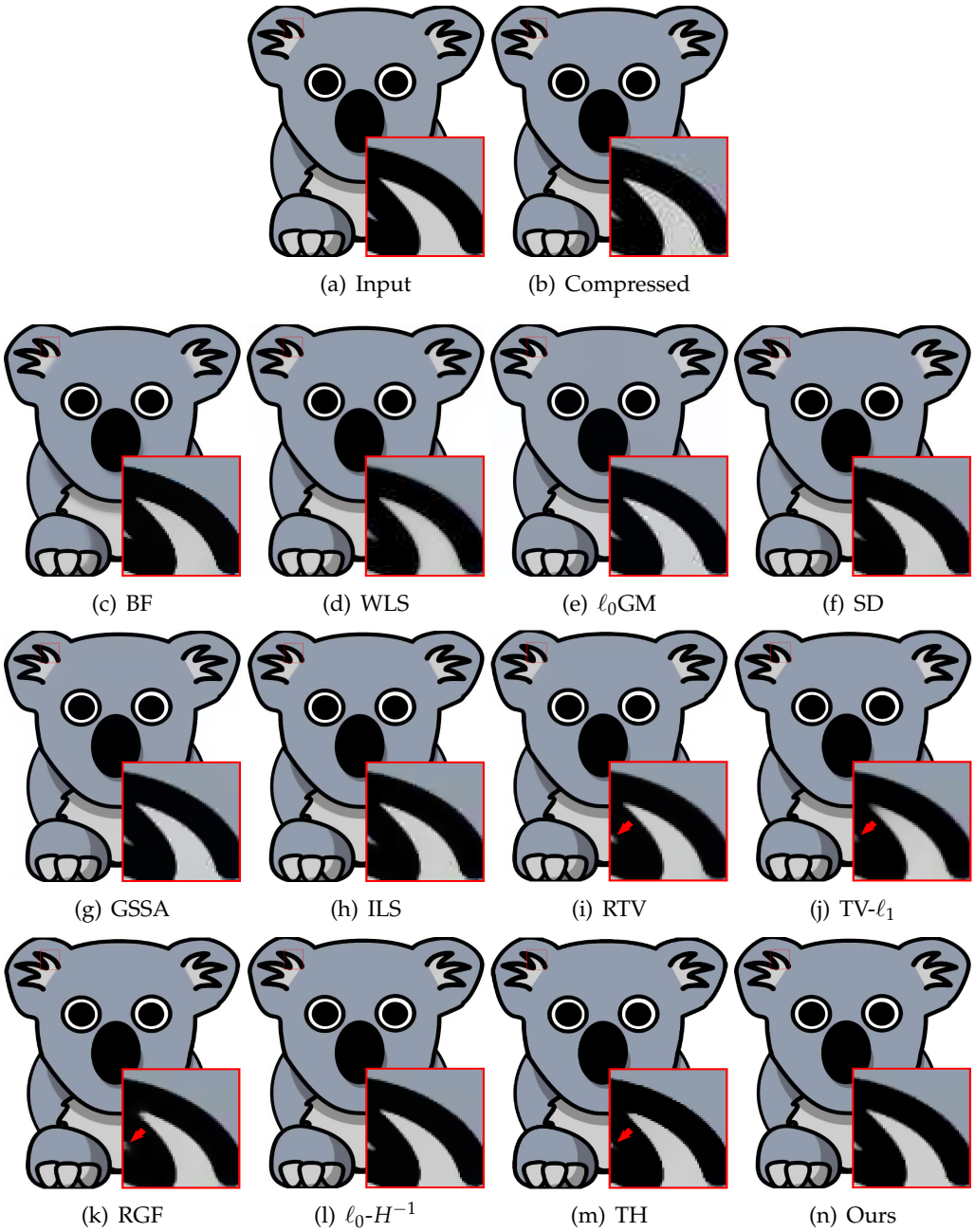


Figure 10: Comparison of JPEG artifact removal. (a) The input image; (b) The degraded image with a compression quality factor of 30; (c)-(n) The smoothing images by using BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and the proposed model respectively.

Table 1: Quantitative comparison of artifact removal.

Methods	BF		WLS		ℓ_0GM		SD		GSSA		ILS		RTV		TV-ℓ_1		RGF		ℓ_0-H^{-1}		TH		Ours	
Compression quality	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	30.97	0.9115	28.14	0.9144	30.65	0.9025	30.14	0.9324	30.84	0.9163	29.98	0.9213	31.14	0.9242	29.71	0.9361	30.76	0.9322	31.05	0.9186	30.88	0.9530	**31.28**	**0.9606**
20	32.35	0.9254	29.83	0.9365	32.71	0.9090	32.45	0.9418	32.45	0.9232	32.48	0.9447	33.02	0.9368	31.88	0.9592	33.06	0.9522	33.49	0.9407	33.31	0.9679	**33.62**	**0.9723**
30	33.57	0.9386	31.10	0.9523	35.03	0.9276	34.35	0.9616	34.35	0.9398	34.81	0.9611	35.06	0.9521	33.76	0.9718	35.21	0.9630	35.76	0.9560	33.38	0.9775	**36.21**	**0.9817**
40	35.29	0.9446	34.29	0.9585	35.53	0.9401	35.83	0.9588	35.83	0.9469	36.04	0.9612	36.91	0.9597	35.45	0.9727	36.36	0.9644	36.17	0.9504	34.21	0.9787	**37.33**	**0.9805**
50	35.86	0.9608	35.39	0.9711	37.06	0.9588	37.05	0.9744	37.05	0.9614	37.49	0.9736	38.05	0.9720	36.57	0.9842	37.38	0.9763	37.84	0.9679	36.50	0.9877	**38.99**	**0.9920**
60	37.23	0.9661	36.15	0.9751	37.64	0.9682	38.39	0.9779	38.39	0.9687	38.56	0.9779	39.30	0.9771	37.17	0.9867	37.87	0.9781	39.66	0.9736	37.00	0.9898	**39.94**	**0.9922**
70	38.78	0.9662	38.58	0.9758	38.95	0.9715	39.91	0.9753	39.91	0.9695	40.46	0.9784	41.42	0.9740	38.36	0.9833	40.04	0.9782	41.85	0.9780	38.68	0.9866	**41.88**	**0.9888**
80	39.24	0.9784	38.80	0.9858	41.02	0.9859	41.42	0.9869	41.42	0.9832	42.45	0.9873	43.67	0.9861	39.58	0.9921	40.34	0.9879	44.61	0.9893	39.84	0.9949	**44.69**	**0.9955**
90	42.00	0.9880	41.48	0.9927	45.42	0.9933	42.78	0.9923	42.78	0.9900	45.96	0.9927	46.72	0.9920	41.13	0.9951	41.16	0.9929	48.27	0.9941	41.52	0.9965	**48.49**	**0.9968**
Average	36.14	0.9533	34.86	0.9625	37.11	0.9508	36.92	0.9668	37.00	0.9554	37.58	0.9665	38.37	0.9638	35.96	0.9757	36.91	0.9695	38.74	0.9632	36.15	0.9814	**39.16**	**0.9845**

quality of restoration will be. In Table 1, we give the PSNR and SSIM values of the artifact removal results via different methods. The best results of PSNR and SSIM are in bold in the table. As can be seen from Table 1, our approach achieves the highest PSNR and SSIM values among these methods, followed by ℓ_0-H^{-1} and TH. Our method performs better than the other methods for artifact removal with respect to PSNR and SSIM.

4.7 Image denoising

The goal of image denoising is to remove noise from the observed noisy image. It is an essential part of image processing. In Fig. 11, we compare the denoising results produced by the competing approaches. In the proposed method, we choose $p = 0.8$, $\lambda = 1.28$ and $a = 0.1$ to achieve satisfactory result. In Fig. 11(a), we show an input image consisting of different colored peppers. The noisy image in Fig. 11(b) is created by adding the salt-and-pepper noise with a ratio of 10% to the input image. The denoised images by the competing methods and our proposed method are shown in Figs. 11(c)-(n), respectively. From Figs. 11(c)-(d) and (h), we notice that these comparative methods are unable to eliminate the salt-and-pepper noise completely, it indicates that these edge-preserving filters are vulnerable to the salt-and-pepper noise. Note that for the ℓ_0GM method, the salt-and-pepper noise cannot be properly removed because the ℓ_0 norm is employed as the regularization term, as shown in Fig. 11(e). As displayed in Figs. 11(f), (i) and (l), the SD, RTV and ℓ_0-H^{-1} approaches remove most of the noise while preserving the structures well. However, these methods all produce oversmoothing results. It can be seen from Fig. 11(g) that the GSSA method performs a smoothing operation, but some noticeable noise can still be seen at the restored image. From Figs. 11(j) and (m)-(n), we observe that the TV-ℓ_1, TH and our method can eliminate salt-and-pepper noise effectively, and the recovered results show a good similarity with the original image in visually.

To quantitatively evaluate the effectiveness of the proposed method in removing salt-and-pepper noise at different noise ratios, we use PSNR and SSIM as metrics to compare the results produced by the twelve methods. In Table 2, we give the PSNR and SSIM values by the comparative methods with noise ratios ranging from 5% to 30%. As shown in Table 2, on average, the proposed method achieves the higher PSNR and SSIM values than BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1} and TH. This shows that our denoising algorithm can achieve better performance for noise removal qualitatively and quantitatively.

(a) Input (b) Noisy

(c) BF (d) WLS (e) ℓ_0GM (f) SD

(g) GSSA (h) ILS (i) RTV (j) TV-ℓ_1

(k) RGF (l) ℓ_0-H^{-1} (m) TH (n) Ours

Figure 11: Comparison of image denoising. (a) The input image; (b) The noisy image with a ratio of 10%; (c)-(n) The smoothing images by using BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and the proposed model respectively.

Table 2: Quantitative comparison of image denoising.

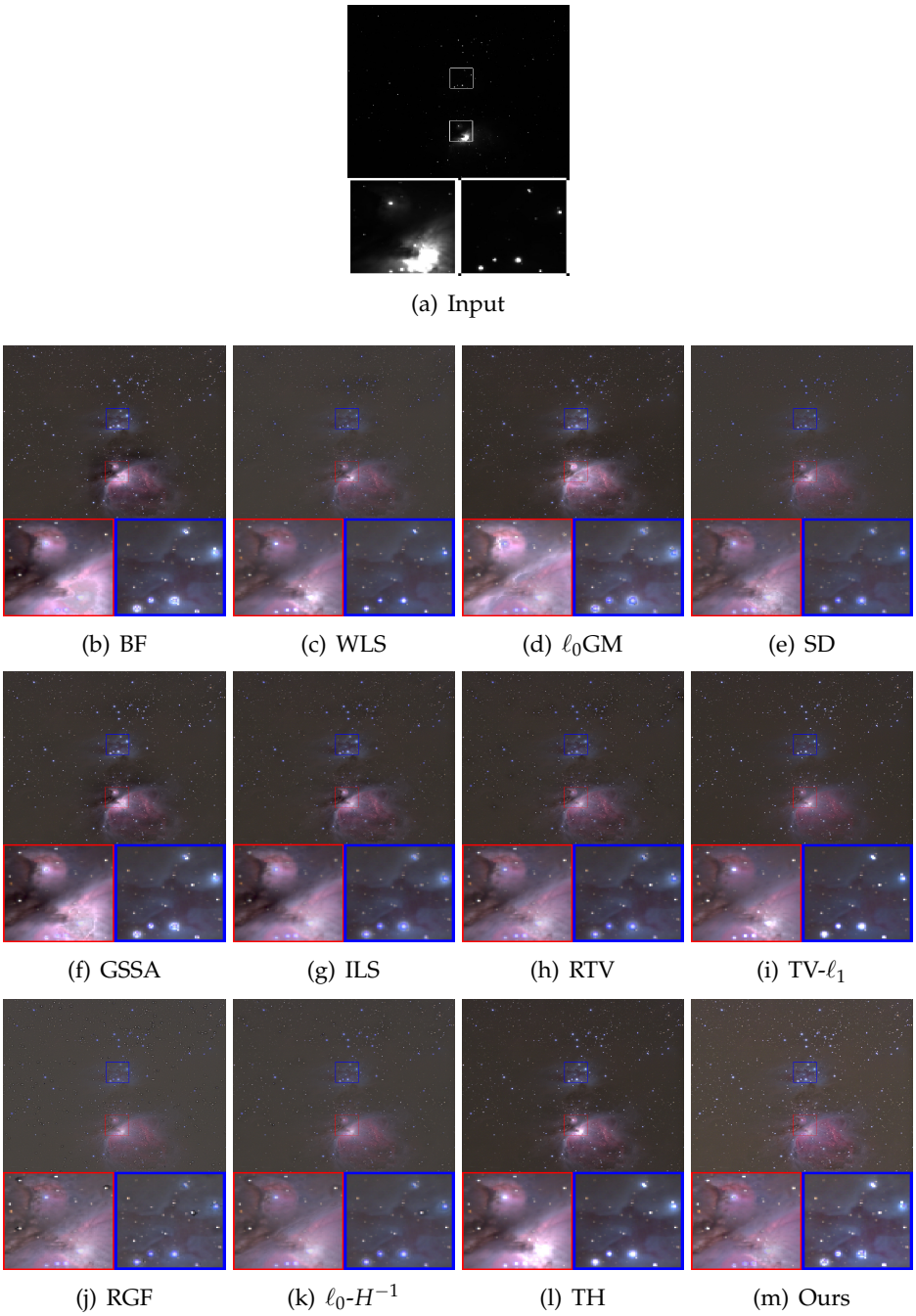
Methods	BF		WLS		ℓ_0GM		SD		GSSA		ILS		RTV		TV-ℓ_1		RGF		ℓ_0-H^{-1}		TH		Ours	
noise ratios	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
5%	27.53	0.9582	26.61	0.9511	17.94	0.7538	26.34	0.9502	20.78	0.8656	26.94	0.9527	27.25	0.9562	34.61	0.9913	28.21	0.9621	27.58	0.9578	30.68	0.9794	**34.92**	**0.9917**
10%	25.57	0.9365	24.85	0.9301	14.93	0.6044	24.66	0.9334	19.82	0.8366	24.93	0.9319	25.27	0.9379	32.97	0.9879	26.11	0.9436	25.92	0.9424	30.32	0.9780	**33.28**	**0.9883**
15%	24.28	0.9206	23.23	0.9045	13.15	0.4996	23.27	0.9147	18.86	0.8109	23.31	0.9095	23.71	0.9178	32.01	0.9852	24.31	0.9220	24.32	0.9233	29.89	0.9761	**32.49**	**0.9864**
20%	22.93	0.8950	21.92	0.8834	11.93	0.4427	22.14	0.8964	18.13	0.7784	22.10	0.8838	22.25	0.8958	31.33	0.9828	22.82	0.8982	22.96	0.9031	29.87	0.9760	**31.83**	**0.9845**
25%	22.02	0.8824	20.71	0.8471	10.96	0.3603	20.94	0.8716	17.42	0.7634	20.93	0.8589	21.18	0.8589	30.61	0.9804	21.47	0.8713	21.68	0.8802	28.83	0.9703	**31.14**	**0.9826**
30%	20.97	0.8593	19.73	0.8252	10.17	0.3093	19.90	0.8453	16.92	0.7243	19.89	0.8316	20.07	0.8457	30.03	0.9781	20.29	0.8411	20.51	0.8518	28.13	0.9658	**30.38**	**0.9797**
Average	23.88	0.9087	22.84	0.8902	13.18	0.4950	22.88	0.9019	18.66	0.7965	23.02	0.8947	23.29	0.9021	31.93	0.9843	23.87	0.9064	23.83	0.9098	29.62	0.9743	**32.34**	**0.9855**

4.8 HDR tone mapping

HDR tone mapping is a technique that reproduces the appearances of HDR images by mapping one set of colors to another on a low dynamic range display. The challenge of this task is to avoid halo artifacts and gradient reversals for the reconstructed image. In Fig. 12, we provide the HDR tone mapping results generated by all competing methods. In our method, we suggest $p=1$, $\lambda=2$ and $a=1$. The original image is given in Fig. 12(a) and the HDR tone mapping images by all methods are displayed in Figs. 12(b)-(m), respectively. From the enlarged parts of Figs. 12(b) and (d), it can be seen that both the BF and ℓ_0GM methods produce strong gradient reversal artifacts near edges. We observe from Fig. 12(c) that the result by WLS contains no halos or gradient reversals. However, the entire image lacks high contrast due to the trade-off between edge-preserving capability and smoothing capability. As shown in Fig. 12(e), the contrast of the SD result is low, and some gradient reversal artifacts can be observed in the highlighted regions. The result of GSSA suffers from annoying gradient reversal artifacts, as illustrated in Fig. 12(f). Compared to the previous edge-preserving algorithms BF and GSSA, the ILS result in Fig. 12(g) shows a better mitigation of the gradient reversal problem, but there are still some artifacts in the magnified regions. The gradient reversal phenomenon occurs in the result produced by the RTV method, as shown in Fig. 12(h). Clearly, we can see no gradient reversals and halo artifacts in the results of TV-ℓ_1 and TH methods. However, the tone mapping results have a slight intensity shift, as shown in the bright spots in the enlarged regions of Figs. 12(i) and (l). From Figs. 12(j) and (k), we notice that more or less visually noisy structures appear on the surfaces of the results produced by RGF and ℓ_0-H^{-1}. The result generated by our method contains neither gradient reversals nor halos, as shown in Fig. 12(m). Compared with the other output images, our tone mapping image looks more attractive and natural.

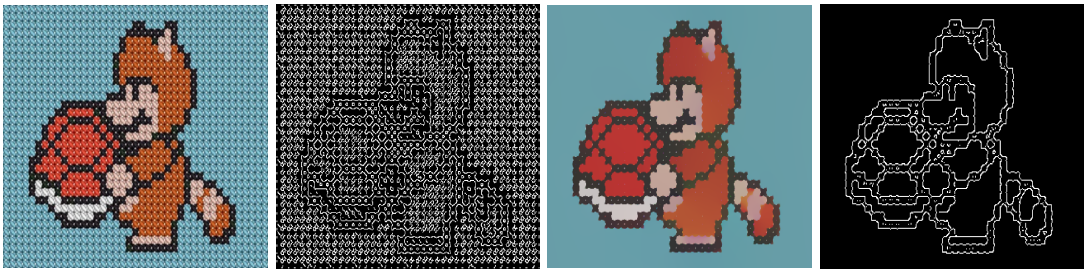
4.9 Edge detection

Detecting edges from the input image is an interesting and important technique in image processing. Since the input image usually contains rich textures, directly extracting edges by an edge detector to the input image is inaccurate. Therefore, texture reduction becomes critical as a pre-processing operation for edge detection. Sometimes, images are affected by vagueness so that the edges of objects are difficult to distinguish. Hence, it is necessary to develop fuzzy image processors, we refer to [55,56] for more details. Among the previously mentioned methods, only the ℓ_0GM and RTV algorithms are used to detect edges with the Canny operator [2] in the original papers, so we only take the two types of methods for comparison. In this task, we adopt $p=0.8$, $\lambda=3.5$ and $a=2$ for the proposed algorithm. In Fig. 13(b), we show the result of edge detection using the Canny operator directly on the "Turtle" image in Fig. 13(a). We see from Fig. 13(b) that some irrelevant edge information appears in the edge map. Clearly, the texture with the high contrast cannot be removed only using the Canny operator. As shown in Fig. 13(d), the edge map

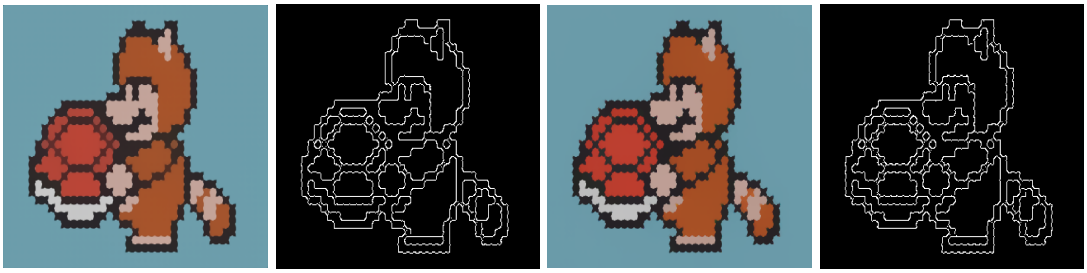


(a) Input (b) BF (c) WLS (d) ℓ_0GM (e) SD (f) GSSA (g) ILS (h) RTV (i) TV-ℓ_1 (j) RGF (k) ℓ_0-H^{-1} (l) TH (m) Ours

Figure 12: Comparison of HDR tone mapping. (a) The input image; (b)-(m) The HDR tone mapping images by using BF, WLS, ℓ_0GM, SD, GSSA, ILS, RTV, TV-ℓ_1, RGF, ℓ_0-H^{-1}, TH and the proposed model respectively.



(a) Input (b) Canny operator on (a) (c) ℓ_0GM (d) Canny operator on (c)



(e) RTV (f) Canny operator on (e) (g) Ours (h) Canny operator on (g)

Figure 13: Comparison of edge detection. (a) The input image; (b) The edge map by using the Canny operator directly on the input image; (c)-(h) The smoothing images and edge maps by using ℓ_0GM, RTV and the proposed model respectively.

of ℓ_0GM is not very accurate since the smoothed image in Fig. 13(c) has false edges. In Figs. 13(e)-(h), we give the RTV and our smoothing results and edge maps, respectively. We observe from these figures that RTV and our method are visually superior to the edge map using the Canny detector directly, since they have removed irrelevant details while maintaining accurate and clean edges. In addition, compared with the RTV method, the proposed algorithm generates a high-contrast smoothing image with cleaner edges.

4.10 Image composition

As well known, graffiti, painting and natural images usually cannot be directly used in image composition. The main reason for this is that the textures of the input image and the target image are incompatible. To mitigate this shortcoming, we first decomposes the input image into a base layer and a detail layer by the proposed method, and then composites the detail layer with the background image to produce a visually plausible result. In Fig. 14, we show an example of image composition. The input image and the background image are given in Figs. 14(a) and (b), respectively. In our method, we choose $p = 1$, $\lambda = 3$ and $a = 1.2$ to obtain the smoothing result. As can be seen from Fig. 14(c), our method can effectively remove the textures and retain the original color information

(a) Input (b) Background (c) Our smoothing (d) Composition

Figure 14: Image composition. (a) The input image; (b) The background image; (c)-(d) The smoothing image and composite image by using the proposed model respectively.

of the face. This makes the composite image in Fig. 14(d) look visually more natural and plausible. It indicates that the proposed method can elegantly achieve the purpose of preserving edges and hence make the composed image more impressive.

5 Conclusions

In this paper, we investigate a novel ℓ_p-nonconvex minimization model for the image smoothing problem. Firstly, we take ℓ_p-norm function with $0<p\leq 1$ as the fidelity term and the arctangent penalty function as the regularization term. Then, we employ the powerful MM method to solve the proposed model and explore the convergence properties of the resulting algorithm. Finally, detailed experiments show that our method is effective in various image processing tasks, including texture smoothing, detail enhancement, artifact removal, image denoising, HDR tone mapping, edge detection, and image composition.

Acknowledgements

This work is supported by NSFC (No. 12171205), Qing Lan project of Jiangsu Province, Hai Yan project, Lianyungang 521 project.

References

[1] PIETRO PERONA AND JITENDRA MALIK, *Scale-space and edge detection using anisotropic diffusion,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7) (1990), pp. 629–639.

[2] PAUL BAO, LEI ZHANG, AND XIAOLIN WU, *Canny edge detection enhancement by scale multiplication,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9) (2005), pp. 1485–1490.

[3] XIAO-GUANG LV, FANG LI, JUN LIU, AND SHENG-TAI LU, *A patch-based low-rank minimization approach for speckle noise reduction in ultrasound images,* Adv. Appl. Math. Mech., 14(1) (2022), pp. 155–180.
[4] YUPING DUAN, HUIBIN CHANG, WEIMIN HUANG, JIAYIN ZHOU, ZHONGKANG LU, AND CHUNLIN WU, *The l_0 regularized mumford–shah model for bias correction and segmentation of medical images,* IEEE Transactions on Image Processing, 24(11) (2015), pp. 3927–3938.
[5] WEI ZHU, SHI SHU, AND LIZHI CHENG, *An efficient proximity point algorithm for total-variation-based image restoration,* Adv. Appl. Math. Mech., 6(2) (2014), pp. 145–164.
[6] XIAOGUANG LV, LE JIANG, AND JUN LIU, *Deblurring poisson noisy images by total variation with overlapping group sparsity,* Appl. Math. Comput., 289 (2016), pp. 132–148.
[7] CHUNHUA ZHANG, ZHEN GAO, SA YE, AND PENG LI, *Edge detectors based on pauta criterion with application to hybrid compact-weno finite difference scheme,* Adv. Appl. Math. Mech., 15 (2023), pp. 1379–1406.
[8] VOLKER AURICH AND JÖRG WEULE, *Non-linear gaussian filters performing edge preserving diffusion,* in Mustererkennung 1995: Verstehen Akustischer und Visueller Informationen, pages 538–545, 1995.
[9] SYLVAIN PARIS AND FRÉDO DURAND, *A fast approximation of the bilateral filter using a signal processing approach,* International Journal of Computer Vision, 81 (2009), pp. 24–52.
[10] RIYA, BHUPENDRA GUPTA, AND SUBIR SINGH LAMBA, *Structure-aware adaptive bilateral texture filtering,* Digital Signal Processing, 123 (2022), 103386.
[11] KAIMING HE, JIAN SUN, AND XIAOOU TANG, *Guided image filtering,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(6) (2012), pp. 1397–1409.
[12] QI ZHANG, LI XU, AND JIAYA JIA, *100+ times faster weighted median filter (WMF),* in IEEE Conference on Computer Vision and Pattern Recognition, pages 2830–2837, 2014.
[13] ZEEV FARBMAN, RAANAN FATTAL, DANI LISCHINSKI, AND RICHARD SZELISKI, *Edge-preserving decompositions for multi-scale tone and detail manipulation,* ACM Transactions on Graphics, 27(3) (2008), pp. 1–10.
[14] KARTIC SUBR, CYRIL SOLER, AND FRÉDO DURAND, *Edge-preserving multiscale image decomposition based on local extrema,* ACM Transactions on Graphics, 28(5) (2009), pp. 1–9.
[15] EDUARDO SL GASTAL AND MANUEL M. OLIVEIRA, *Domain transform for edge-aware image and video processing,* ACM Transactions on Graphics, 30(4) (2011), pp. 1244–1259.
[16] LI XU, CEWU LU, YI XU, AND JIAYA JIA, *Image smoothing via l_0 gradient minimization,* ACM Transactions on Graphics, 30(6) (2011), pp. 1–12.
[17] SYLVAIN PARIS, SAMUEL W HASINOFF, AND JAN KAUTZ, *Local laplacian filters: Edge-aware image processing with a laplacian pyramid,* ACM Transactions on Graphics, 30(4) (2011), 68.
[18] BUMSUB HAM, MINSU CHO, AND JEAN PONCE, *Robust guided image filtering using nonconvex potentials,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(1) (2018), pp. 192–207.
[19] JUN LIU, MING YAN, JINSHAN ZENG, AND TIEYONG ZENG, *Image smoothing via gradient sparsity and surface area minimization,* in IEEE International Conference on Image Processing, pages 1114–1118, 2019.
[20] WEI LIU, PINGPING ZHANG, XIAOLIN HUANG, JIE YANG, CHUNHUA SHEN, AND IAN REID, *Real-time image smoothing via iterative least squares,* ACM Transactions on Graphics, 39(3) (2020), pp. 1–24.
[21] FEIDA ZHU, ZHETONG LIANG, XIXI JIA, LEI ZHANG, AND YIZHOU YU, *A benchmark for edge-preserving image smoothing,* IEEE Transactions on Image Processing, 28(7) (2019), pp. 3556–3570.

[22] LI XU, QIONG YAN, YANG XIA, AND JIAYA JIA, *Structure extraction from texture via relative total variation,* ACM Transactions on Graphics, 31(6) (2012), pp. 1–10.
[23] LEVENT KARACAN, ERKUT ERDEM, AND AYKUT ERDEM, *Structure-preserving image smoothing via region covariances,* ACM Transactions on Graphics, 32(6) (2013), pp. 1–11.
[24] JUNHO JEON, SUNGHYUN CHO, XIN TONG, AND SEUNGYONG LEE, *Intrinsic image decomposition using structure-texture separation and surface normals,* in European Conference on Computer Vision, pages 218–233, 2014.
[25] VINCENT LE GUEN, *Cartoon + texture image decomposition by the TV-l_1 model,* Image Processing On Line, 4 (2014), pp. 204–219.
[26] QI ZHANG, XIAOYONG SHEN, LI XU, AND JIAYA JIA, *Rolling guidance filter,* in European Conference on Computer Vision, pages 815–830, 2014.
[27] LEI ZHU, CHI-WING FU, YUEMING JIN, MINGQIANG WEI, JING QIN, AND PHENG-ANN HENG, *Non-local sparse and low-rank regularization for structure-preserving image smoothing,* in Computer Graphics Forum, volume 35, pages 217–226, 2016.
[28] HYUNJOON LEE, JUNHO JEON, JUNHO KIM, AND SEUNGYONG LEE, *Structure-texture decomposition of images with interval gradient,* in Computer Graphics Forum, volume 36, pages 262–274, 2017.
[29] XIAOJIE GUO, SIYUAN LI, LIANG LI, AND JIAWAN ZHANG, *Structure-texture decomposition via joint structure discovery and texture smoothing,* in IEEE International Conference on Multimedia and Expo, pages 1–6, 2018.
[30] QINGNAN FAN, JIAOLONG YANG, DAVID WIPF, BAOQUAN CHEN, AND XIN TONG, *Image smoothing via unsupervised learning,* ACM Transactions on Graphics, 37(6) (2018), pp. 1–14.
[31] LIJUN ZHAO, HUIHUI BAI, JIE LIANG, ANHONG WANG, BING ZENG, AND YAO ZHAO, *Local activity-driven structural-preserving filtering for noise removal and image smoothing,* Signal Processing, 157 (2019), pp. 62–72.
[32] WEI CAO, SHIQIAN WU, ZHAOYI LIU, AND SOS AGAIAN, *Scale-aware guided and structure-preserved texture filter,* IEEE Geoscience and Remote Sensing Letters, 19 (2021), pp. 1–5.
[33] KAIYUE LU, SHAODI YOU, AND NICK BARNES, *Deep texture and structure aware filtering network for image smoothing,* in European Conference on Computer Vision, pages 217–233, 2018.
[34] YIDAN FENG, SEN DENG, XUEFENG YAN, XIN YANG, MINGQIANG WEI, AND LIGANG LIU, *Easy2hard: Learning to solve the intractables from a synthetic dataset for structure-preserving image smoothing,* IEEE Transactions on Neural Networks and Learning Systems, 33(12) (2021), pp. 7223–7236.
[35] BOLUN CAI, XIAOFEN XING, AND XIANGMIN XU, *Edge/structure preserving smoothing via relativity-of-gaussian,* in IEEE International Conference on Image Processing, pages 250–254, 2017.
[36] WEI LIU, PINGPING ZHANG, YINJIE LEI, XIAOLIN HUANG, JIE YANG, AND MICHAEL NG, *A generalized framework for edge-preserving and structure-preserving image smoothing,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10) (2022), pp. 6631–6648.
[37] WEI CAO, SHIQIAN WU, JIAXIN WU, ZHAOYI LIU, AND YUWEN LI, *Edge/structure-preserving texture filter via relative bilateral filtering with a conditional constraint,* IEEE Signal Processing Letters, 28 (2021), pp. 1535–1539.
[38] IVAN W. SELESNICK AND İLKER BAYRAM, *Sparse signal estimation by maximally sparse convex optimization,* IEEE Transactions on Signal Processing, 62(5) (2014), pp. 1078–1092.
[39] PO-YU CHEN AND IVAN W SELESNICK, *Group-sparse signal denoising: non-convex regularization, convex optimization,* IEEE Transactions on Signal Processing, 62(13) (2014), pp. 3464–

3478.

[40] GUOLIANG ZHU, XIAOGUANG LV, XUEMAN SUN, AND BIAO FANG, *An l_p-nonconvex regularization method for image smoothing,* in Fourteenth International Conference on Graphics and Image Processing, volume 12705, pages 227–236, 2023.

[41] STÉPHANE MALLAT, A Wavelet Tour of Signal Processing, Academic Press, 1998.

[42] NICK KINGSBURY AND TANYA REEVES, *Redundant representation with complex wavelets: How to achieve sparsity,* in IEEE International Conference on Image Processing, pages I–45, 2003.

[43] LÎUDMILA ANDREEVNA PILIPCHUK, Sparse Linear Systems and Their Applications, 2013.

[44] ZHENGZHENG YAN, DANDAN SHANG, JINGZHI LI, AND RONGLIANG CHEN, *A parallel numerical method for risk assessment of myocardial infarction during liver transplantation: a case study,* Adv. Appl. Math. Mech., 14 (2022), pp. 1225–1245.

[45] ALEXEY SHCHEGLOV, JINGZHI LI, CHAO WANG, ALEXANDER ILIN, AND YE ZHANG, *Reconstructing the absorption function in a quasi-linear sorption dynamic model via an iterative regularizing algorithm,* Adv. Appl. Math. Mech., 16 (2024), pp. 237–252.

[46] GUANGXIN HUANG, ALESSANDRO LANZA, SERENA MORIGI, LOTHAR REICHEL, AND FIORELLA SGALLARI, *Majorization-minimization generalized krylov subspace methods for l_p–l_q optimization applied to image restoration,* BIT Numerical Mathematics, 57(2) (2017), pp. 351–378.

[47] RAYMOND H. CHAN AND HAIXIA LIANG, *Half-quadratic algorithm for l_p–l_q problems with applications to TV–l_1 image restoration and compressive sensing,* in Efficient Algorithms for Global Optimization Methods in Computer Vision, pages 78–103, 2014.

[48] IVAN W. SELESNICK, ANKIT PAREKH, AND İLKER BAYRAM, *Convex 1-D total variation denoising with non-convex regularization,* IEEE Signal Processing Letters, 22(2) (2014), pp. 141–144.

[49] MÁRIO AT FIGUEIREDO, JOSÉ M BIOUCAS-DIAS, AND ROBERT D NOWAK, *Majorization-minimization algorithms for wavelet-based image restoration,* IEEE Transactions on Image processing, 16(12) (2007), pp. 2980–2991.

[50] HUAN PAN, WEI-WEN WEN, AND HUI-MIN ZHU, *A regularization parameter selection model for total variation based image noise removal,* Applied Mathematical Modelling, 68 (2019), pp, 353–367.

[51] CURTIS R. VOGEL, Computational Methods for Inverse Problems, volume 23. SIAM, 2002.

[52] MILA NIKOLOVA, *Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares,* . Multiscale Modeling & Simulation, 4(3) (2005), pp. 960–991.

[53] GUOLIANG ZHU, XIAOGUANG LV, LE JIANG, XUEMAN SUN, AND BIAO FANG, *Nonconvex regularization for convex image smoothing,* Signal Processing, 205 (2023), 108862.

[54] MILA NIKOLOVA, MICHAEL K NG, AND CHI-PAN TAM, *On l_1 data fitting and concave regularization for image recovery,* SIAM Journal on Scientific Computing, 35(1) (2013), pp. A397–A430.

[55] PATRICIA MELIN, CLAUDIA I GONZALEZ, JUAN R CASTRO, OLIVIA MENDOZA, AND OSCAR CASTILLO, *Edge-detection method for image processing based on generalized type-2 fuzzy logic,* IEEE Transactions on Fuzzy Systems, 22(6) (2014), pp. 1515–1525.

[56] SUDEB MAJEE, SUBIT K JAIN, RAJENDRA K RAY, AND ANANTA K MAJEE, *A fuzzy edge detector driven telegraph total variation model for image despeckling,* Inverse Problems and Imaging, 16(2) (2022), pp. 367–396.