A Generalized Framework for Edge-preserving and Structure-preserving Image Smoothing

Wei Liu, Pingping Zhang, Yinjie Lei, Xiaolin Huang, Jie Yang and Michael Ng

Abstract—Image smoothing is a fundamental procedure in applications of both computer vision and graphics. The required smoothing properties can be different or even contradictory among different tasks. Nevertheless, the inherent smoothing nature of one smoothing operator is usually fixed and thus cannot meet the various requirements of different applications. In this paper, we first introduce the truncated Huber penalty function which shows strong flexibility under different parameter settings. A generalized framework is then proposed with the introduced truncated Huber penalty function. When combined with its strong flexibility, our framework is able to achieve diverse smoothing natures where contradictory smoothing behaviors can even be achieved. It can also yield the smoothing behavior that can seldom be achieved by previous methods, and superior performance is thus achieved in challenging cases. These together enable our framework capable of a range of applications and able to outperform the state-of-the-art approaches in several tasks, such as image detail enhancement, clip-art compression artifacts removal, guided depth map restoration, image texture removal, etc. In addition, an efficient numerical solution is provided and its convergence is theoretically guaranteed even the optimization framework is non-convex and non-smooth. A simple yet effective approach is further proposed to reduce the computational cost of our method while maintaining its performance. The effectiveness and superior performance of our approach are validated through comprehensive experiments in a range of applications. Our code is available at 
https://github.com/wliusjtu/Generalized-Smoothing-Framework

Index Terms—Truncated Huber penalty function, edge-preserving image smoothing, structure-preserving image smoothing

1 INTRODUCTION

T

HE key challenge of many tasks in both computer vision and graphics can be attributed to image smoothing. At the same time, the required smoothing properties can vary dramatically for different tasks. In this paper, depending on the required smoothing properties, we roughly classify a large number of applications into four groups.

Applications in the first group require the smoothing operator to smooth out small details while preserving strong edges, and the amplitudes of these strong edges can be reduced but the edges should be neither blurred nor sharpened. Representatives in this group are image detail enhancement and HDR tone mapping. Blurring edges can result in halos while sharpening edges will lead to gradient reversals.

The second group includes tasks like clip-art compression artifacts removal, image abstraction and pencil sketch production. In contrast to the ones in the first group, these tasks require to smooth out small details while sharpening strong edges. This is because edges can be blurred in the compressed clip-art image and they need to be sharpened when the image is recovered, an example is illustrated in Fig. 1(b). Sharper edges can produce better visual quality in image abstraction and pencil sketch. At the same time, the amplitudes of strong edges are not allowed...
to be reduced in these tasks.

Guided image filtering, such as guided depth map upsampling [9], [10], [11] and flash/no flash filtering [12], [13], is categorized into the third group. The structure inconsistency between the guidance image and the target image, which can cause blurring edges and texture copy artifacts in the smoothed image [7], [11], should be properly handled by the specially designed smoothing operator. They also need to sharpen edges in the smoothed image due to the reason that low-quality capture of depth maps and the noise in the no-flash images can lead to blurred edges, see Fig. 1(c) for example.

Tasks in the fourth group require to smooth the image in a scale-aware manner, e.g., image texture removal [8], [14], [15]. This kind of tasks require to smooth out small structures even when they contain strong edges, while large structure should be properly preserved even the edges are weak, Fig. 1(d) shows an example. This is totally different from that in the above three groups where they all aim at preserving strong edges.

Generally, the smoothing procedures in the first to the third groups are usually considered as edge-preserving image smoothing since they try to preserve salient edges, while the smoothing processes in the fourth group are classified as structure-preserving image smoothing because they aim at preserving salient structures.

A diversity of smoothing operators have been proposed for various tasks in the literature. Generally, each of them is designed to meet the requirements of certain applications, and its inherent smoothing nature is usually fixed. Therefore, there is seldom any smoothing operator that can meet all the smoothing requirements of the above four groups. For example, the $L_0$ norm smoothing [5] can sharpen strong edges and is suitable for clip-art compression artifacts removal, however, this will lead to gradient reversals in image detail enhancement and HDR tone mapping. The weighted least squares (WLS) smoothing [11] performs well in image detail enhancement and HDR tone mapping, but it is not capable of sharpening edges. In a higher view, edge-preserving smoothing operators are also not directly applicable to structure-preserving tasks. Thus, designing a smoothing operator that is capable of these different smoothing properties still remains a challenging problem.

Besides the challenge mentioned above, there are also challenging cases that cannot be well handled by either edge-preserving smoothing or structure-preserving smoothing. Fig. 2 illustrates an example of clip-art compression artifacts removal. In this example, the heavy compression artifacts lead to some small structures with large-amplitude edges. At the same time, both the “black lines” labeled with the red arrows (small structures with strong edges) and the “shades” labeled with the blue arrows (large structures with weak edges) need to be preserved, and the edges should also be sharpened. The challenges of this case are twofold. On the one hand, if we remove the artifacts in an edge-preserving manner, then the weak edges around the “shades” will also be smoothed as their amplitudes are small. Fig. 2(b) illustrates the result of the static/dynamic (SD) filter [2] which is an edge-preserving smoother. The edges of the “black lines” are sharpened in the result, but the edges of the “shades” are blurred while some large-amplitude artifacts still retain in the result. On the other hand, if the artifacts are removed with a structure-preserving smoother, the “black lines” may also be removed as their structures are small. Fig. 2(c) shows the result of relative total variation (RTV) smoothing [8] which is a structure-preserving smoother. Although the “shades” are preserved, some “black lines” are also removed together with the artifacts. Besides, the edges are not sharpened in the result. In contrast, besides the various edge-preserving and structure-preserving smoothing properties mentioned in previous paragraphs, our method is also able to yield simultaneous edge-preserving and structure-preserving smoothing behavior which is seldom achieved by previous approaches. This blended smoothing property can enjoy the advantages of both edge-preserving smoothing and structure-preserving smoothing. Fig. 2(d) shows the result of our method where both the “black lines” and the “shades” are preserved, and the edges are also sharpened. As we will show in Sec. 3.4, this blended simultaneous edge-preserving and structure-preserving smoothing property can act as a promising alternative for handling the tasks in the second group and the third group for better performance.

In this paper, we propose a new smoothing operator. In contrast to most of the smoothing operators in the literature, it can achieve various smoothing behaviors and is able to handle all the challenges mentioned above. The main contributions of this paper are as follows:

1. We introduce the truncated Huber penalty function which has seldom been used in image smoothing. By varying the parameters, it shows strong flexibility.
2. A generalized framework is proposed with the truncated Huber penalty function. When combined with the strong flexibility of the truncated Huber penalty function, our model can achieve various smoothing behaviors. We show that it is able to handle the tasks in the four groups mentioned above. Besides, our model is also able to achieve simultaneous edge-preserving and structure-preserving smoothing property which can yield superior performance over previous methods in challenging cases (e.g., Fig. 2). All these are seldom achieved by previous smoothing operators in the literature. We also show that our method is able to achieve state-of-the-art performance in many tasks.
3. An efficient numerical solution to the proposed optimization framework is provided. Its convergence is theoretically guaranteed even the framework is non-convex and non-smooth. A simple yet effective approach is further proposed to reduce the computational cost of our method while maintaining its performance.

This manuscript is the extension of its conference version [16] with the following differences: (1) We provide more detailed analysis of the introduced penalty function in Sec. 3.1 and the proposed model in Sec. 3.2. The concept of “dilated neighborhood” is further introduced in Sec. 3.3 to reduce the computational cost of our method while maintaining its performance. (3) We show more applications and experimental results in Sec. 4. More quantitative evaluation is also provided to validate the effectiveness of our method.
The rest of this paper are organized as follows: Sec. 2 describes the related work of our method. Sec. 3 is devoted to our approach including the introduction of the truncated Huber penalty function, the definition of our model, its numerical solution and further property analysis of the model. Our method is compared against many state-of-the-art approaches in a range of applications in Sec. 4. We draw the conclusion and summarize the limitations of this paper in Sec. 5.

2 RELATED WORK
Image smoothing has been a well-studied research filed. Tremendous smoothing operators have been proposed in recent decades. In terms of edge-preserving smoothing, bilateral filter (BLF) [17] is the early work that has been used in various tasks such as image detail enhancement [2], HDR tone mapping [18], etc. However, it is prone to produce results with gradient reversals and halos [1]. Gastal et al. [19] proposed adaptive manifold filter (AMF) as an alternative of BLF to handle high-dimension data such RGB color images. They further introduced domain transform filter (DTF) [20] for fast image processing. Similar to BLF, these approaches also share similar problems of resulting in gradient reversals and halos. Guided filter (GF) [3] can produce results free of gradient reversals but halos can still exist. The WLS smoothing [1] forms image smoothing as a global optimization problem. It has been one of the well-known milestones due to its superior performance in handling the gradient reversals and halos. However, as noted by Hessel et al. [21], the WLS smoothing is prone to result in compartmentalization artifacts. In contrast, our method provides a promising alternative that can properly eliminate the compartmentalization artifacts as well as gradient reversals and halos. The $L_0$ norm smoothing proposed by Xu et al. [5] is able to smooth out weak edges while sharpening strong edges, which can be applied to the tasks in the second group. Its shortage is that low-amplitude structures in the image can also be eliminated. To handle the structure inconsistency problem, Shen et al. [22] proposed to perform mutual-structure joint filtering. They also explored the relation between the guidance image and the target image via optimizing a scale map [23], however, additional processing was adopted for structure inconsistency handling. Ham et al. [2] proposed to handle the structure inconsistency by combining a static guidance weight with a Welsch’s penalty regularized smoothness term, which led to a static/dynamic (SD) filter. Gu et al. [23] presented a weighted analysis representation model for guided depth map enhancement. They also proposed to smooth images by layer decomposition, and different sparse representation models were adopted for different layers [26].

In terms of structure-preserving smoothing, Zhang et al. [14] proposed to smooth structures of different scales with a rolling guidance filter (RGF). Cho et al. [15] modified the original BLF with local patch-based analysis of texture features and obtained a bilateral texture filter (BTF) for image texture removal. Karacan et al. [27] proposed to smooth image textures by making use of region covariances that captured local structure and textural information. Xu et al. [8] adopted the relative total variation (RTV) as a prior to regularize the texture smoothing procedure. Chen et al. [28] proved that the TV-$L_1$ model [28], [29] could smooth images in a scale-aware manner, which is ideal for structure-preserving smoothing, e.g., image texture removal [30], [31].

Most of the approaches mentioned above are limited to a few applications because their inherent smoothing natures are usually fixed. In contrast, our method proposed in this paper can have strong flexibility in achieving various smoothing behaviors, which enables wider applications of our method than most of them. In addition, our method is also able to achieve the smoothing behavior that is seldom achieved by previous approaches. This makes our method able to better handle the challenges in many tasks. We show that our method can show superior performance over these methods in several applications that they are specially designed for.

In recent years, a large majority of deep learning based approaches have also been proposed [32], [33], [34], [35], [36], [37]. These methods adopt neural network architectures to learn the smoothing behaviors of different existing filters. However, their main drawback is that different models need to be trained separately for different filters. Even for a given filter, different models also need to be trained separately for different parameter settings of the filter. Thus, the deep learning based methods are not easy to tune the parameters. Chen et al. [32] and Fan et al. [38] tried to make their methods tunable of the parameters, however, their methods only explored the tunability of one parameter while the other parameters need to be fixed.

3 OUR APPROACH
The first challenge raised in Sec. 1 is that we need a switch which can control whether to sharpen edges or not. This can be achieved by adopting a proper penalty function to
regularize the output. However, a penalty function with such kind of “switch” is seldom introduced in the literature. A widely utilized one to sharpen edges is the Welsch’s penalty function [24] which was also adopted in the recent proposed SD filter [7] to produce sharp edges. The shortage of the Welsch’s penalty function is that it is close to \( L_2 \) norm when the input is small, as illustrated in Fig. 3(a). This means it cannot properly preserve weak edges. An example is shown in Fig. 4(b) where the weak edges labeled with the red arrows are largely smoothed. The gradient \( L_0 \) norm smoothing proposed by Xu et al. [5] is another well-known method that is able to sharpen salient edges. However, as they utilize a series of truncated \( L_2 \) norms to gradually approximate the \( L_0 \) norm, weak edges can seldom be preserved in their results, as shown in Fig. 4(c).

### 3.1 Truncated Huber Penalty Function

Beside there is seldom a “switchable” penalty function, the current widely used penalty functions for sharpening edges do not work well to preserve weak edges as mentioned above. To overcome these shortages, we introduce the truncated Huber penalty function which is defined as:

\[
    h_T(x) = \begin{cases} 
        h(x), & |x| \leq b \\
        b - \frac{a}{2}, & |x| > b 
    \end{cases} \text{ s.t. } a \leq b, 
\]

where \( a, b \) are constants. \( h(\cdot) \) is the Huber penalty function [29] defined as:

\[
    h(x) = \begin{cases} 
        \frac{1}{2} x^2, & |x| < a \\
        |x| - \frac{a}{2}, & |x| \geq a 
    \end{cases}
\]

\( h_T(\cdot) \) and \( h(\cdot) \) are plotted in Fig. 3(a) with \( a = \epsilon \) where \( \epsilon \) is a sufficient small value (e.g., \( \epsilon = 10^{-3} \)).

The truncation in Eq. (1) enables \( h_T(\cdot) \) to sharpen edges. This can be better understood through the concept of edge stopping functions. The edge stopping function of a penalty function \( \rho(x) \) is defined as \( \varphi(x) = \frac{\rho'(x)}{x} \) where \( \rho'(x) \) is the derivative of \( \rho(x) \) with respect to \( x \). A larger value of \( \varphi(x) \) means that \( \rho(x) \) has larger penalty on the input \( x \) and vice versa. This concept is not directly applicable to \( h_T(x) \) because it is not differentiable at \( x = b \). However, if we define the value of \( h_T'(x) \) as its left limit and the value of \( h_T'(b) \) as its right limit, then we have the edge stopping function of \( h_T(x) \) as:

\[
    \varphi_T(x) = \frac{h_T'(x)}{x} = \begin{cases} 
        \frac{1}{x}, & a \leq |x| \leq b \\
        0, & |x| > b 
    \end{cases} 
\]

Eq. (3) indicates that the parameter \( b \) in \( h_T(x) \) can act as a threshold: for the input \( x \) larger than \( b \), it will directly not be penalized while the input smaller than \( b \) remains penalized. \( \varphi_T(x) \) is illustrated in Fig. 3(b). If we use \( h_T(x) \) to regularize the gradients of the output image, its penalty behaviour can enable the salient edges of the input image to be sharpened. This property also shares a similar mechanism with the Welsch’s penalty function. The edge stopping function of the Welsch’s penalty function rapidly reduces (at an exponential rate) to the values close to zero when the input increases, as shown in Fig. 3(b). Similarly, the value of the edge stopping function of \( h_T(\cdot) \) directly decreases to zero when the input is larger than \( b \).

Based on the above analysis, we first show that \( h_T(\cdot) \) in Eq. (1) is a switchable penalty function where the value of \( b \) can properly control whether to sharpen edges or not. Given the input intensity values are within \([0, I_m]\), then the amplitude of any edge will fall in \([0, I_m]\). If we set \( b > I_m \), \( h_T(\cdot) \) will be actually the same as \( h(\cdot) \) because the second condition in Eq. (1) can never be met, then \( h_T(\cdot) \) will be an edge-preserving penalty function that does not sharpen edges. Conversely, when we set \( b < I_m \), the truncation in
$h_T(\cdot)$ will be activated. This can lead to penalty on weak edges without penalizing strong edges, and the strong edges are thus sharpened. In this way, $b$ can act as a switch to decide whether $h_T(\cdot)$ can sharpen strong edges or not. Besides the switchable property mentioned above, $h_T(\cdot)$ can also properly preserve weak edges if we set $a = \epsilon$ to a sufficient small value. This is because $h_T(\cdot)$ will be close to the $L_1$ norm for the small input when $a$ is sufficient small, and $L_1$ norm shows good edge-preserving property for weak edges. We show examples in Fig. 3(d) and (e) where we use $h_T(\cdot)$ with $a = \epsilon$ for regularization, and the weak edges are better preserved than that in Fig. 3(b) and (c). In addition, we set $b > I_m$ in Fig. 3(d) where the salient edges are not sharpened, and we set $b < I_m$ in Fig. 3(e) where the salient edges are sharpened, see the comparison between the regions labeled with the blue arrow and the green arrow.

Generally, $h_T(\cdot)$ can show strong flexibility which enable it to yield different penalty behaviours under different parameter settings. If we set $a = \epsilon, b > I_m$, $h_T(\cdot)$ will be close to the $L_1$ norm in this case, and thus it will be an edge-preserving penalty function that does not sharpen edges. We can also set $a = \epsilon, b < I_m$ to enable $h_T(\cdot)$ to sharpen edges. Similarly, by setting $a = b > I_m$ and $a = b < I_m$, $h_T(\cdot)$ can be easily switched between the $L_2$ norm and the truncated $L_2$ norm. We should note out that the Welch’s penalty function does not enjoy such kind of flexibility. Different cases of $h_T(\cdot)$ and their corresponding edge stopping functions are illustrated in Fig. 3(c) and (d).

### 3.2 Model

Given an input image $f$ and a guidance image $g$, the smoothed output image $u$ is the solution that gives the minimum to the following objective function:

$$E_u(u) = \sum_{i,j \in N_d(i)} \omega_{i,j}^u h_T(u_i - u_j) + \lambda \sum_{i,j \in N_d(i)} \omega_{i,j}^u \omega_{i,j}^h h_T(u_i - u_j), \tag{4}$$

where $h_T$ is defined in Eq. (1); $N_d(i)$ is the $(2d+1) \times (2d+1)$ square patch centered at $i$; $N_w(i)$ is the $(2s+1) \times (2s+1)$ square patch centered at $i$ excluding pixel $i$; $\lambda$ is a parameter that controls the overall smoothing strength. The first term of Eq. (4) is denoted as the data term and the second term of Eq. (4) is denoted as the smoothness term in this paper. To be clear, we adopt $\{a_d, b_d\}$ and $\{a_s, b_s\}$ to denote the parameters of $h_T(\cdot)$ in the data term and the smoothness term, respectively. $\omega_{i,j}^u$ is a Gaussian spatial kernel defined as:

$$\omega_{i,j}^u = \exp \left( \frac{|i - j|^2}{2\sigma^2} \right), \tag{5}$$

we simply fix $\sigma = r_d$ for the data term and $\sigma = r_s$ for the smoothness term in all of our experiments. The guidance weight $\omega_{i,j}^g$ is defined as:

$$\omega_{i,j}^g = \frac{1}{|g_i - g_j|^\alpha + \delta}, \tag{6}$$

where $\alpha$ determines the sensitivity to the edges in $g$ which can be the input image, i.e., $g = f$. $|\cdot|$ represents the absolute value. $\delta$ is a small constant which is set as $\delta = 10^{-3}$.

The adoption of $h_T(\cdot)$ enables our model in Eq. (4) to enjoy strong flexibility. As we will show in Sec. 3.4, under different parameter settings, our model is able to achieve different smoothing behaviors, and it is thus capable of various tasks that require different smoothing properties. In addition, our model can also yield simultaneous edge-preserving and structure-preserving smoothing that is seldom achieved by previous methods, which enables our model to achieve better performance in challenging cases.

### 3.3 Numerical Solution

Our model in Eq. (4) is not only non-convex but also non-smooth, which arises from the adopted $h_T(\cdot)$. Commonly used approaches [42, 43, 44, 45, 46, 47] for solving non-convex optimization problems are thus not applicable. To tackle this problem, we need to first rewrite $h_T(\cdot)$ in a new equivalent form. By defining $\nabla_l^d = u_i - f_j$ and $\nabla_l^g = u_i - u_j$, we have:

$$h_T(\nabla_l^d) = \min_{i,j} \left\{ h(\nabla_l^d - \nabla_l^d) + (b_s - \frac{a_s}{2})|\nabla_l^d|_0 \right\}, \tag{7}$$

where $* \in \{d, s\}$ and $|\nabla_l^d|_0$ is the $L_0$ norm of $\nabla_l^d$. The minimum of the right side of Eq. (7) is obtained at:

$$\nabla_l^g = \left\{ \begin{array}{ll} 0, & |\nabla_l^d|_0 \leq b_s, \\
\nabla_l^g, & |\nabla_l^g|_0 > b_s, \end{array} \right. \quad *, \in \{d, s\}. \tag{8}$$

The detailed proof of Eq. (7) and Eq. (8) is provided in Appendix A. These two equations also theoretically validate our analysis in Sec. 3.1 and Fig. 3(c): we have $|\nabla_l^d|_0 \in [0, I_m]$ if the intensity values are within $[0, I_m]$. Then if $b > I_m$, based on Eq. (7) and Eq. (8), we will always have $h_T(\nabla_l^d) = h(\nabla_l^g)$ which means $h_T(\cdot)$ degrades to $h(\cdot)$.

We can then define a new energy function as:

$$E_{ul}(u, l^d, l^g) = \sum_{i,j} \omega_{i,j}^* \left( h(\nabla_l^d - \nabla_l^d) + (b_d - \frac{a_d}{2})|\nabla_l^d|_0 \right) + \lambda \sum_{i,j} \omega_{i,j} \left( h(\nabla_l^g - \nabla_l^g) + (b_s - \frac{a_s}{2})|\nabla_l^g|_0 \right), \tag{9}$$

where $\omega_{i,j} = \omega_{i,j}^u \omega_{i,j}^g$. Based on Eq. (7) and Eq. (8), we then have:

$$E_u(u) = \min_{l^d, l^g} E_{ul}(u, l^d, l^g), \quad *, \in \{d, s\}. \tag{10}$$

Given Eq. (9) as the optimum condition of Eq. (10) with respect to $l^*$, optimizing $E_{ul}(u, l^d, l^g)$ with respect to $u$ only involves the Huber penalty function $h(\cdot)$. The problem can thus be optimized through the half-quadatic (HQ) optimization technique [18, 13]. More specifically, a variable $\mu^*(* \in \{d, s\})$ and a function $\psi(\mu^*)$ with respect to $\mu^*$ exist such that:

$$h(\nabla_l^d - \nabla_l^d) = \min_{\mu^*} \left\{ \mu^*(\nabla_l^d - \nabla_l^d)^2 + \psi(\mu^*) \right\}, *, \in \{d, s\}. \tag{11}$$

where the optimum is yielded at:

$$\mu^* = \left\{ \begin{array}{ll} 1, & |\nabla_l^d - \nabla_l^d| < a_s, \\
\frac{2a_s}{2|\nabla_l^d - \nabla_l^d|}, & |\nabla_l^d - \nabla_l^d| \geq a_s, \end{array} \right. \quad *, \in \{d, s\}. \tag{12}$$

The detailed proof of Eq. (11) and Eq. (12) is provided in Appendix B. Then we can further define a new energy function as:

$$E_{ul}(u, l^d, l^g, \mu^*; \mu^*) = \sum_{i,j} \omega_{i,j} \left( \mu^*(\nabla_l^d - \nabla_l^d)^2 + \psi(\mu^*) \right) + \lambda \sum_{i,j} \omega_{i,j} \left( \mu^*(\nabla_l^g - \nabla_l^g)^2 + \psi(\mu^*) \right), \tag{13}$$
Based on Eq. (11) and Eq. (12), we then have:
\[ E_{ul}(u, l^*) = \min_{\mu^*} E_{ul\mu}(u, l^*, \mu^*), \quad * \in \{d, s\}. \]  
(14)

Given Eq. (12) as the optimum condition of \( \mu^* \) in Eq. (14), optimizing \( E_{ul\mu}(u, l^*, l^*, \mu^*) \) with respect to \( u \) only involves the \( L_2 \) norm penalty function, which has a closed-form solution. However, since the optimum conditions in Eq. (9) and Eq. (12) both involve \( u \), therefore, the final solution \( u \) can only be obtained in an iterative manner. Assuming we have got \( u^k \), then \( (l^k)^* \) and \( (\mu^k)^*(\in \{s, d\}) \) can be updated through Eq. (8) and Eq. (12) with \( u^k \), respectively. Finally, \( u^{k+1} \) is obtained with:
\[ u^{k+1} = \arg\min_u E_{ul\mu}(u, (l^k)^*, (\mu^k)^*), \quad * \in \{d, s\}. \]  
(15)

Eq. (15) has a close-form solution as:
\[ u^{k+1} = \left( A^k - 2\lambda \mathcal{W}^k \right)^{-1} (D^k + 2\lambda S^k), \]  
(16)

where \( \mathcal{W}^k \) is an affinity matrix with \( \mathcal{W}_{i,j}^k = \omega_{i,j}(\mu_{i,j}^k)^k \), \( A^k \) is a diagonal matrix with \( A_{i,i}^k = \sum_{j \in N(i)} \omega_{i,j}(\mu_{i,j}^k)^k + 2\lambda \sum_{j \in N(i)} \omega_{i,j}(\mu_{i,j}^k)^k, D^k \) is a vector with \( D_{i,j}^k = \sum_{j \in N(i)} \omega_{i,j}(\mu_{i,j}^k)^k (f_j + (l_{i,j}^k))^k \) and \( S^k \) is also a vector with \( S_{i,j}^k = \sum_{j \in N(i)} \omega_{i,j}(\mu_{i,j}^k)^k (l_{i,j}^k)^k \).

The above optimization procedure monotonically decreases the value of \( E_u(u) \) in each step, and its convergence is theoretically guaranteed. Given \( u^k \) in the \( k \)-th iteration and \( * \in \{d, s\} \), then for any \( u \), we have:
\[ E_u(u) \leq E_{ul}(u, (l^k)^k), \quad E_{ul}(u^k) = E_{ul}(u^k, (l^k)^k). \]  
(17)

\[ \left\{ \begin{array}{l}
E_{ul}(u, (l^k)^k) \leq E_{ul\mu}(u, (l^k)^k, (\mu^k)^k), \\
E_{ul}(u^k, (l^k)^k) = E_{ul\mu}(u^k, (l^k)^k, (\mu^k)^k).
\end{array} \right. \]  
(18)

Given \( (l^k)^k \) has been updated through Eq. (8), Eq. (17) is based on Eq. (10) and Eq. (7). After \( (\mu^k)^k \) has been updated through Eq. (12), Eq. (18) is based on Eq. (14) and Eq. (11). We now have:
\[ E_{ul}(u^{k+1}, (l^k)^k) \leq E_{ul\mu}(u^{k+1}, (l^k)^k, (\mu^k)^k) \]
\[ \leq E_{ul\mu}(u^k, (l^k)^k, (\mu^k)^k) = E_{ul}(u^k, (l^k)^k), \]  
(19)

the first and the second inequalities follow from Eq. (18) and Eq. (15), respectively. We finally have:
\[ E_u(u^{k+1}) \leq E_u(u^{k+1}, (l^k)^k) \leq E_u(u^k, (l^k)^k) = E_u(u^k), \]  
(20)

the first and the second inequalities follow from Eq. (17) and Eq. (19), respectively. Since the value of \( E_u(u) \) is bounded from below, Eq. (20) thus indicates that the convergence of our iterative scheme is theoretically guaranteed.

The above optimization procedure is iteratively performed \( N \) times to get the final smoothed output \( u^N \). \( N \) can vary for different applications, and some tasks do not need to iterate the above procedure until it converges. These will be detailed in Sec. 3.4. In all our experiments, we set \( u^0 = f \), which is able to produce promising results in each application. Our optimization procedure is summarized in Algorithm 1.

### Algorithm 1 A Generalized Framework for Edge-preserving and Structure-preserving Image Smoothing

**Input:** Input image \( f \), guide image \( g \), iteration number \( N \), parameter \( \lambda, \alpha, a_s, b_s, r_s, \)\( u^0 \leftarrow f \), with \( * \in \{d, s\} \)

1. for \( k = 0 : N \) do
   2. With \( u^k \), compute \( \nabla v^k \), update \( (l^k)^k \) according to Eq. (8)
   3. With \( (l^k)^k \), update \( (\mu^k)^k \) according to Eq. (12)
   4. With \( (l^k)^k \) and \( (\mu^k)^k \), solve for \( u^{k+1} \) according to Eq. (15) (or Eq. (16))

**Output:** Smoothed image \( u^N \)

### 3.4 Property Analysis

Under different parameter settings, the strong flexibility of \( h_T(\cdot) \) makes our model able to achieve various smoothing behaviors. First, we show that some classical approaches can be viewed as special cases of our model. For example, by setting \( a_d = b_d > I_m, a_s = c, b_s > I_m, \alpha = 0, r_d = 0, r_s = 1 \), our model is a close approximation of the TV model [49] which is a representative edge-preserving smoother. If we set \( g = f, a_d = b_d = a_s = b_s > I_m, \alpha = 1.2, r_d = 0, r_s = 1 \), our model will be the WLS smoother [11] which performs well in handling gradient reversals and halos in image detail enhancement and HDR tone mapping. With parameters \( a_d = c, b_d > I_m, a_s = c, b_s > I_m, \alpha = 0, r_d = 0, r_s = 1 \), our model is a close approximation of the TV-L1 model [30], [41] which is classical for structure-preserving smoothing.

#### 3.4.1 Tasks in the Fourth Group

Besides the special cases mentioned above, our model can also achieve other smoothing behaviors under different parameter settings and better results can be further produced. To simplify the analysis in the following subsections, we first start with the tasks in the fourth group which require structure-preserving smoothing. For these tasks, the parameters are set as \( a_d = c, b_d > I_m, a_s = c, b_s > I_m, r_d = r_s = 1 \), \( \alpha = 0.5, g = f \). In this case, our model has the following two advantages: first, the setting \( a_d = c, b_d > I_m, a_s = c, b_s > I_m \) enables our model to have the structure-preserving property similar to that of the TV-L1 model; second, the guidance weight with \( \alpha = 0.5, g = f \) can make our model to produce sharper edges in the results than the TV-L1 model. We illustrate this with 1D smoothing results in Fig. 5(a) and (b). Fig. 6(b) and (c) further show a comparison of image texture removal results. As shown in the figures, both the TV-L1 model and our model can properly remove the small textures, however, edges in our result are much sharper than that in the result of the TV-L1 model. We fix \( r_d = r_s = 1 \). The value of \( \lambda \) depends on the texture size. Larger \( \lambda \) can lead larger structures to be removed. The iteration number is set as \( N = 10 \). Our model under this parameter setting is denoted as the SP-1 mode of our method in Tab. 1.

#### 3.4.2 Tasks in the First Group

When dealing with image detail enhancement and HDR tone mapping in the first group, one way is to set the parameters so that our model can perform WLS smoothing
TABLE 1
Parameter settings for different tasks. \( \epsilon \) refers to a small constant, e.g., \( \epsilon = 10^{-3} \); \( I_m \) denotes the maximum intensity value of the input image \( f \); \( g = f \) means the guidance image is the same as the input image to be smoothed; \( g = g \) denotes that the guidance image is different from the input image to be smoothed. \( SP \) and \( EP \) are short for structure-preserving and edge-preserving, respectively.

| mode       | \( g \) | \( \alpha \) | \( a_d \) | \( b_d \) | \( a_s \) | \( b_s \) | \( r_d \) | \( r_s \) | \( N \) | Applications          | Properties                     |
|------------|--------|-------------|----------|----------|----------|----------|--------|--------|-------|-----------------------|-----------------------------|
| SP-1       | \( b \) | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | 1        | 1      | 1      |       | structure-preserving   | tasks in the first group    |
| SP-2       | \( b \) | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | 0        | 1      | 1      |       | structure-preserving   | tasks in the second group   |
| EP-1       | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | 0      | 1      | 1      | edge-preserving, do not sharpen edges | tasks in the first group |
| EP-2       | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | 0      | 1      | 1      | edge-preserving, sharpen edges | tasks in the second and third group |
| EP\&SP     | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | \( \epsilon \) | \( > I_m \) | 1      | 1      | 10     | simultaneous edge-preserving and structure-preserving, sharpen edges | tasks in the second and third group |

Fig. 5. 1D signal with structures of different scales and amplitudes. Smoothing result of (a) TV-L1 smoothing, (b) our method of the SP-1 mode, (c) WLS (our method of the EP-1 mode), (d) our method of the SP-2 mode, (e) SD filter [7], (f) our method of the EP-2 mode and (g) our method of the EP\&SP mode.

Fig. 6. Comparison of image texture removal results. (a) Input image. Result of (b) TV-L1 smoothing [31], (e) our method of the SP-1 mode.

, which is denoted the EP-1 mode of our method in Tab. 1. The WLS smoothing is a strong baseline in the literature that performs well in handling gradient reversals and halos. However, it can also produce intensity shift in the results [4, 50]. On the one hand, the intensity shift can be adopted to enhance the image contrast, which is appealing in image detail enhancement, as shown in the first row of Fig. 7(b) and (d). On the other hand, the intensity shift can lead to compartmentalization artifacts in some cases [21, 50]. One example of image detail enhancement is illustrated in the second row of Fig. 7(b) and (d). Fig. 8(a) further shows an example of HDR tone mapping.

The reason of the intensity shift in WLS smoothing (or our method of the EP-1 mode) can be obtained through the following analysis. In fact, if we set \( a_d = b_d > I_m, a_s = \epsilon, b_s > I_m, r_d = 0, r_s = 1, \alpha = 0.2, g = f \) in our model, the first iteration of Algorithm 1 \((N = 1)\) can also be considered to be equivalent to WLS smoothing. Our model under this parameter setting can be approximately viewed as a “weighted TV” model. At the same time, it is known that the TV model can lead to intensity shift [28], and this is why the WLS smoothing can result in intensity shift in the results. In contrast, the TV-L1 model does not cause intensity shift [28]. Thus, we can set the parameters of our model as follows: \( a_d = \epsilon, b_d > I_m, a_s = \epsilon, b_s > I_m, r_d = r_s = 1, \alpha = 0.2, g = f \), which can be approximately viewed as a “weighted TV-L1” model. Similarly, the first iteration of Algorithm 1 \((N = 1)\) can be used to smooth the input image. Note that we set \( r_d = 1 \) instead of \( r_d = 0 \) because our experiments show that \( r_d = 1 \) can achieve better performance when we set \( N = 1 \). We denote our model under this parameter setting as the SP-2 mode of our method in Tab. 1. Our experiments show that, with a large value of \( \lambda \), the amplitudes of different structures will decrease at different rates, i.e., the amplitudes of small structures can have a larger decrease than the large ones, as illustrated in Fig. 9(d). At the same time, edges are neither blurred nor sharpened. These properties are desirable for image detail enhancement and HDR tone mapping. Compared with the result of WLS smoothing in Fig. 5(c), edges in the result produced by our method of the SP-2 mode in Fig. 5(d) are better preserved (see the bottom of the 1D signals). Fig. 5(b) and the second row of Fig. 7(c) and (e) further show the results of HDR tone mapping and image detail enhancement obtained with our method of the SP-2 mode. As shown in the figures, the compartmentalization artifacts are properly alleviated. However, we should point out that our method of the SP-2 mode is not prone to enhance image contrast, as shown in the first row of Fig. 7(c) and (e). In this case, it is inferior to the WLS smoothing (our method of the EP-1 mode). In addition, \( \lambda \) in the SP-2 mode of our method is usually much larger than that in the SP-1 mode, for example, the results in Fig. 7(c) is generated with \( \lambda = 20 \).

Image detail enhancement and HDR tone mapping in the first group have long been considered as edge-preserving tasks. In fact, most methods proposed for these tasks in the literature are edge-preserving smoothers [11, 13, 15, 17, 19, 20, 51]. Our method of the EP-1 mode (or WLS smoothing [11]) is also in the spirit of edge-preserving smoothing. In contrast to previous methods, our method of the SP-2 mode provides a promising alternative that handles these tasks in a structure-preserving manner and shows better performance.

3.4.3 Tasks in the Second Group and the Third Group
To sharpen edges that is required by the tasks in the second group and the third group, based on the analysis in Sec. 3.1...
we can set \( a_s = \epsilon, b_s < I_m \) in the smoothness term. For the data term, we can set the parameters as \( a_d = b_d > I_m \), i.e., \( L_2 \) norm penalty in the data term. This results in an edge-preserving smoother, which is denoted as the EP-2 mode of our method in Tab. 1. This kind of parameter setting for sharpening edges is in the same spirit to the SD filter where a Welsch’s penalty function is adopted in the smoothness term. However, as analyzed in Sec. 3.1, our truncated Huber penalty function can better preserve weak edges than the Welsch’s penalty function. This can be validated through our analysis in Sec. 3.1 and the comparison of the 1D smoothing results shown in Fig. 5(e) and (f).

Our method of the EP-2 mode can better preserve edges than the SD filter, however, the weak edges in the large structure are also penalized, as shown in the left part of Fig. 5(j). If we only need to remove the small structure in the middle left of Fig. 5(j) with the rest structures properly preserved, then this will be a quite challenging case, and there are seldom existing smoothers that can properly handle this. The challenge lies on the fact that we need to both preserve large structures with weak edges and small structures with strong edges at the same time. On the one hand, if we perform edge-preserving smoothing, the large structure on the left will be penalized because the corresponding edges are weak, as shown in Fig. 5(e) and (f). On the other hand, if we adopt structure-preserving smoothing, then the small structure in the middle right will be removed due to its small structure size, as shown in Fig. 5(a) and (b).

In fact, we can also set the parameters of the data term as \( a_d = \epsilon, b_d < I_m \). The parameter setting \( a_d = a_s = \epsilon \) makes our model enjoy the structure-preserving property. At the same time, \( b_s < I_m \) enables our model not to penalize large-amplitude edges, which is the edge-preserving property. In this way, our model has the simultaneous edge-preserving and structure-preserving property, which is denoted as the EP&SP mode of our method in Tab. 1. The truncation \( b_d < I_m \) in the data term can help our model to be robust against the outliers in the input image, for example, the noise in the no-flash image and the low-quality depth map. With the simultaneous edge-preserving and structure-preserving smoothing nature, our model is able to handle the challenging case mentioned above. Fig. 5(g) shows the 1D smoothing result of our method of the EP&SP mode. This challenging case is also of practical importance such as the example of clip-art compression artifacts removal shown in Fig. 2. Fig. 9 further shows an example of guided depth map upsampling in the third group. Our method of the EP&SP mode shows better performance than our method of the EP-2 mode in both edge-preserving property and quantitative evaluation, as shown in Fig. 9(c) and (d), respectively. Mean absolute errors (MAE) are adopted as the evaluation metric in Fig. 9(d). Another example can also be found in Fig. 4 where our method of the EP&SP mode can better preserve weak edges than our method of the EP-2 mode. Please refer to the regions labeled with the red arrows in Fig. 4(d) and (e). We adopt the EP&SP mode of our method for all the tasks in the second group and the third group hereafter. We further fix \( \alpha = 0.5, r_d = r_s, N = 10 \). We empirically set \( b_d = b_s = 0.05I_m \sim 0.2I_m \) and \( r_d = r_s \geq 1 \) depending on the applied task and the input noise level.

The structure inconsistency issue in the third group can also be easily handled by our model. Note that \( \mu_{i,j}^s \) in Eq. (13) is computed with the smoothed image in each iteration, as formulated in Eq. (12). It thus can reflect the inherent natures of the smoothed image. The guidance weight \( \omega_{i,j} \) can provide additional structural information.
from the guidance image \( g \). This means that \( \mu^s_{i,j} \) and \( \omega_{i,j} \) can complement each other. In fact, the equivalent guidance weight of Eq. (13) in each iteration is \( \mu^s_{i,j} \omega_{i,j} \), which can reflect the property of both the smoothed image and the guidance image. In this way, it can properly handle the structure inconsistency problem to avoid blurring edges and texture copy artifacts. Similar ideas were also adopted in previous approaches [27, 11].

### 3.5 Dilated Non-local Neighborhood for Computational Cost Reduction

In the EP\&SP mode of our method, we set the neighborhood radius as \( r_d = r_s = 1 \sim 7 \). When the input noise level is high, we usually adopt a non-local neighborhood system where the neighborhood radius is larger than 1. This is different from most previous methods that adopt a 4-connected/8-connected neighborhood system [22] or only \( x \)-axis and \( y \)-axis gradients [3, 5, 8, 50] in the smoothness term. Both the neighborhood radius and \( \lambda \) in our model can be used to control the smoothing strength. We show that a larger neighborhood radius with a smaller \( \lambda \) can yield better performance than a smaller neighborhood with a larger \( \lambda \) especially for heavy input noise. Fig. 10 shows examples of guided depth map upsampling where a larger neighborhood can better preserve edges and smooth noise than a smaller neighborhood. The performance improvement is especially noticeable for large upsampling factors (heavy input noise) as shown in Fig. 10(d).

A larger neighborhood can yield better performance, however, this is also achieved at the expense of higher computational cost, as illustrated in Fig. 11(d). This is due to the reason that with the increasing of the neighborhood radius, more neighbor pixels are involved, which leads to the increase of the computational cost. Inspired by the dilated convolution [52, 53] in the recent deep learning technique, we can adopt a dilated neighborhood to reduce the computational cost. The dilated neighborhood of pixel \( i \) is defined as:

\[
N^D(i) = \{ j \mid j = i - r + t \cdot s; t = 0, 1, 2, \ldots, \lfloor \frac{2 \cdot r + 1}{s} \rfloor \}, \tag{21}
\]

where \( s \) is denoted as the stride of \( N^D(i) \) and \( r \) is the neighborhood radius. When \( s = 1 \), \( N^D(i) \) is the same as the original neighborhood defined in our model. Fig. 11(a) illustrates examples of the dilated neighborhood. As shown in the figure, the parameter setting of \( r = 2, s = 2 \) has the same long range connection between neighbor pixels as \( r = 2, s = 1 \) does, however, the number of involved pixels is greatly reduced, which is the same as that of \( r = 1, s = 1 \). With much fewer neighbor pixels involved, the computational cost is thus reduced. According to our experiments, for a range of neighborhood radiuses, \( s = 2 \) is able to achieve very close performance to \( s = 1 \) with the computational cost greatly reduced, as shown in Fig. 11(b)~(d). Further increasing the value of the stride can lead to blurring edges and noticeable performance drop, as shown in Fig. 11(b) and (c). Note that the measured time is for our method of the EP\&SP mode where the iteration number is \( N = 10 \). For other modes of our method with the iteration number of \( N = 1 \), the required computing time is only around \( \frac{1}{10} \) of that shown in Fig. 11(d). More examples of other tasks are illustrated in Fig. 12.

### 4 Applications and Experimental Results

Our method is applied to various tasks in the first to the fourth groups to validate the effectiveness. Comparisons with the state-of-the-art approaches in each application are also presented. When adopting our method for image smoothing, the intensity values of the input image are first normalized into range \([0, 1]\) before the smoothing, they are then normalized back to their original range for quantitative
input images, the values of the other parameters are fixed as those in Tab. 1. Fig. 13 and Fig. 14 show visual comparison of image detail enhancement and tone mapping results produced by different smoothers, respectively. As shown in the figures, there are either gradient reversals or halos in the results of the compared methods. Some results even contain both gradient reversals and halos, as shown in Fig. 13(b) and Fig. 14(a) and (b). In contrast, no gradient reversals and halos exist in the results of WLS smoothing and our method of the SP-2 mode. However, the tone mapping result of WLS smoothing contains slight compartmentalization artifacts caused by intensity shift, as shown in the region labeled with the yellow box in Fig. 14(g). This is properly eliminated in the result of our method of the SP-2 mode, shown in Fig. 14(h).

We collect 40 HDR images to further quantitatively evaluate the performance of all the compared approaches. All the tone mapping results are evaluated in terms of tone mapping quality index (TMQI) proposed by Yeganeh et al. [64]. TMQI first evaluates the structural fidelity and the naturalness of the tone mapped images. The two measurements are then adjusted by a power function and averaged to give a final score ranging from 0 to 1. Larger values of TMQI indicate better quality of the tone mapped images, and vice versa. Tab. 2 shows the mean TMQI, structural fidelity and naturalness of the results produced by each of the compared methods. As shown in the table, our method is able to achieve the best performance in most cases.

4.2 Tasks in the Second Group

We also apply our method to clip-art compression artifacts removal, and the EP&SP mode of our method is adopted. The input image is used as the guidance image in our method, i.e., \( g = f \). Clip-art images are piecewise constant with sharp edges. When they are compressed in JPEG format with a low quality factor, there will be edge-related artifacts, and the edges are usually blurred as shown in Fig. 15(a). Therefore, when removing the compression artifacts, the edges should also be sharpened in the restored image. We evaluate all the compared methods with 30 different collected images. These images are first compressed in JPEG format with the quality factor ranging from 10 to 90. The parameters of our method are set as follows: \( r_d = r_s = 2, \lambda = 0.4, b_d = b_s = 0.15 \) for the compression quality factor of 10, \( \lambda \) is divided by 2 and \( b_d = b_s \) is decreased by 0.01.
TABLE 2
Quantitative evaluation of HDR tone mapping results. The best results are in bold. The second best results are underlined.

| Method            | AMF | BLF | DTF-NC | DTF-IC | DTF-RF | EAW | FG | L₀ norm | SG-WLS | SWF | Ours (EP-1) | Ours (SP-2) |
|-------------------|-----|-----|--------|--------|--------|-----|----|---------|--------|-----|-------------|-------------|
| Naturalness       | 0.8531 | 0.8491 | 0.8491 | 0.8491 | 0.8598 |     |    |         |        |     | 0.8408 | 0.8407 |
| Fidelity          | 0.8257 | 0.8227 | 0.8263 | 0.8155 | 0.8338 | 0.8398 | 0.8175 | 0.8265 | 0.8425 | 0.8407 | 0.8461 |
| TMQI              | 0.8614 | 0.8586 | 0.8570 | 0.8488 | 0.8525 | 0.8541 | 0.8536 | 0.8534 | 0.8603 | 0.8496 | 0.8666 | 0.8712 |

TABLE 3
Quantitative comparison of clip-art compression artifacts removal results. The best results are in bold. The second best results are underlined.

| Method            | PSNR | SSIM |
|-------------------|------|------|
| Compression Quality | 10   | 20   | 30   | 40   | 50   | 60   | 70   | 80   | 90   |
| JPEG              | 31.15 | 0.9482 | 0.9622 | 0.9685 | 0.9692 | 0.9757 | 0.9794 | 0.9874 | 0.9930 |
| Wang [6]          | 31.27 | 0.9518 | 0.9664 | 0.9725 | 0.9729 | 0.9792 | 0.9811 | 0.9883 | 0.9923 |
| BTF [15]          | 32.12 | 0.9657 | 0.9759 | 0.9789 | 0.9814 | 0.9843 | 0.9866 | 0.9879 | 0.9923 |
| L₀ norm            | 31.16 | 0.9522 | 0.9639 | 0.9743 | 0.9796 | 0.9811 | 0.9837 | 0.9851 | 0.9890 |
| Region Fusion [4] | 31.91 | 0.9626 | 0.9761 | 0.9793 | 0.9815 | 0.9833 | 0.9852 | 0.9862 | 0.9905 |
| deep prior [57]   | 28.52 | 0.9401 | 0.9437 | 0.9459 | 0.9468 | 0.9497 | 0.9516 | 0.9529 | 0.9544 |
| Ours              | 33.56 | 0.9823 | 0.9878 | 0.9983 | 0.9910 | 0.9928 | 0.9933 | 0.9947 | 0.9962 |

(a) input (b) AMF (c) GF (d) SWF
(e) SG-WLS (f) L₀ norm (g) WLS/ours (EP-1) (h) ours (SP-2)

Fig. 13. Image detail enhancement results of different approaches. (a) Input image. Smoothed image and 3× detail enhanced image of (b) AMF [19] (σₛ = 20, σᵣ = 0.25), (c) GF [3] (r = 20, ϵ = 0.15²), (d) SWF [56] (r = 7, iteration = 1), (e) SG-WLS [55] (r = 2, τ = 1, λ = 50), (f) L₀ norm smoothing [5] (λ = 0.02), (g) our method of the EP-1 mode/WLS [1] (λ = 1, α = 1.2) and (h) our method of the SP-2 mode (λ = 20).

(a) BF (b) AMF (c) GF (d) SWF
(e) SG-WLS (f) L₀ norm (g) WLS/ours (EP-1) (h) ours (SP-2)

Fig. 14. HDR tone mapping results of different approaches. Result of (a) BLF [17] (σₛ = 20, σᵣ = 0.25), (b) AMF [19] (σₛ = 16, σᵣ = 0.12), (c) GF [3] (r = 20, ϵ = 0.1²), (d) SWF [56] (r = 10, iteration = 2), (e) SG-WLS [55] (r = 2, τ = 1, λ = 75), (f) L₀ norm smoothing [5] (λ = 0.01), (g) our method of the EP-1 mode/WLS [1] (λ = 10, α = 1.2) and (h) our method of the SP-2 mode (λ = 20).

when the quality factor is increased by 10. The values of the other parameters are fixed as those in Tab. 1.

Fig. 15 shows the visual comparison of the results produced by different methods. The approach proposed by Wang et al. [6] can seldom handle heavy compression artifacts. Their result is shown in Fig. 15(c). Cho et al. [15] also applied their bilateral texture filter (BTF) to compression artifacts removal, however, their method can blur edges...
TABLE 4
Quantitative comparison on the noisy simulated ToF data. Results are evaluated with the MAE between the upsampled depth map and the ground-truth depth map. The best results are in **bold**. The second best results are underlined.

| Method  | Real | JPEG | SD Filter | DKN | FBS | FG1 | SGF | DJF | FGI | GF | Image 60x64 to 183x183 |
|---------|------|------|-----------|-----|-----|-----|-----|-----|-----|----|------------------------|
| Mean    | 3.13 | 5.17 | 3.09      | 3.04| 3.03| 3.02| 3.01| 3.01| 3.00| 2.99| 3.05                   |
| Median  | 2.97 | 4.98 | 2.96      | 2.93| 2.92| 2.91| 2.90| 2.90| 2.89| 2.88| 2.94                   |
| RMSE    | 2.31 | 3.86 | 2.26      | 2.18| 2.17| 2.16| 2.15| 2.15| 2.14| 2.13| 2.19                   |
| MAE     | 1.87 | 2.94 | 1.83      | 1.80| 1.79| 1.78| 1.77| 1.77| 1.76| 1.75| 1.81                   |
| PSNR    | 15.04| 14.48| 14.71     | 14.70| 14.68| 14.67| 14.66| 14.66| 14.65| 14.64| 14.62                 |

TABLE 5
Quantitative comparison on real ToF data. The errors are calculated as the MAE to the measured ground-truth depth maps. The best results are in **bold**. The second best results are underlined.

| Method  | Real | JPEG | SD Filter | DKN | FBS | FG1 | SGF | DJF | FGI | GF | Image 60x64 to 183x183 |
|---------|------|------|-----------|-----|-----|-----|-----|-----|-----|----|------------------------|
| Mean    | 3.13 | 5.17 | 3.09      | 3.04| 3.03| 3.02| 3.01| 3.01| 3.00| 2.99| 3.05                   |
| Median  | 2.97 | 4.98 | 2.96      | 2.93| 2.92| 2.91| 2.90| 2.90| 2.89| 2.88| 2.94                   |
| RMSE    | 2.31 | 3.86 | 2.26      | 2.18| 2.17| 2.16| 2.15| 2.15| 2.14| 2.13| 2.19                   |
| MAE     | 1.87 | 2.94 | 1.83      | 1.80| 1.79| 1.78| 1.77| 1.77| 1.76| 1.75| 1.81                   |
| PSNR    | 15.04| 14.48| 14.71     | 14.70| 14.68| 14.67| 14.66| 14.66| 14.65| 14.64| 14.62                 |

Fig. 15. Clip-art compression artifacts removal results of different methods. (a) Input compressed image and (b) the corresponding ground-truth image. Result of (c) the approach proposed by Wang et al. [6], (d) BTF [15] (k = 3, n_T = 2), (e) L_0 norm smoothing [5] (A = 0.01), (f) region fusion approach [4] (λ = 0.05), (g) deep image prior [57] and (h) our method of the EP×SP mode (r_L = r_S = 2, λ = 0.4, b_d = b_s = 0.15).
Our method of the EP&SP mode is applied to the guided depth map restoration. We first test our method on the simulated dataset constructed from the Middlebury dataset [66]. The simulated dataset contains six depth maps and four upsampling factors for each depth map, as listed in Tab. 4. We fix $r_d = r_s = 5$ for all the upsampling factors, and $\lambda = 0.1/0.25/0.5/0.95, b_d = b_s = 0.1/0.1/0.08/0.07$ for $2 \times /4 \times /8 \times /16 \times$ upsampling. The other parameters are fixed as those in Tab. 1. We compare our method against the state-of-the-art approaches including the very recently proposed deep learning based methods such as deep joint filtering (DJF) [63] and deformable kernel network (DKN) [63]. Note that these two approaches do not perform $2 \times$ upsampling, the corresponding results are thus omitted in Tab. 4. Fig. 16 shows the visual comparison between our result and the results of the compared methods. Our method shows better performance in preserving sharp depth edges and avoiding texture copy artifacts, as illustrated in the highlighted regions of Fig. 16. Tab. 4 also shows the quantitative evaluation of the results produced by different methods. Mean absolute error (MAE) between the upsampled depth map and the ground-truth one is adopted as the evaluation metric, which is also widely used in previous work [65, 59, 55, 58]. As Tab. 4 shows, our method can achieve the best or the second best performance among all the compared approaches.

We further validate our method on the real ToF data introduced by Ferstl et al. [10]. The real dataset contains three low-resolution depth maps captured by a ToF depth camera and the corresponding highly accurate ground-truth depth maps captured with structured light. The upsampling factor for the real dataset is $\sim 6.25 \times$. The parameters our method for this dataset are as follows: $r_d = r_s = 3, b_d = b_s = 0.08, \lambda = 0.5$. The visual comparison in Fig. 17 and the quantitative comparison in Tab. 5 show that our method can outperform the compared methods and achieve state-of-the-art performance.

We also apply our method of the EP&SP mode to flash/no flash image filtering. Images captured in low-light condition usually contain heavy noise. A high-quality image can be obtained with the help of flash. This flash image can be used as the guidance to smooth out the noise in the no-flash image. However, the flash can cause shadows in the image that do not exist in the no-flash image, as...
shown in the highlighted regions in Fig. 18(a). This structure inconsistency can lead to blurring edges/texture copy artifacts in the smoothed image if the smoothing procedure is not robust against the structure inconsistency. As the test images are collected from different datasets, the noise level and light condition are quite different. We thus only fix $r_d = r_s = 1$ for different test images. The other parameters which are not fixed in Tab. 1 differs for different test images, and they are detailed in the figure caption. Fig. 18 shows the visual comparison of the results produced by different approaches. The results of GF [3], deep image prior [57] and the optimal scale map filter [23] suffer from blurring edges, as highlighted in Fig. 18(b), (c) and (d), respectively. The DJF [62] can avoid these artifacts, however, the noise is less smoothed in its result, as shown in Fig. 18(e). The results of SD filter [7] in Fig. 18(f), mutually guided image filter (muGIF) [65] in Fig. 18(g) and our result in Fig. 18(h) can properly preserve sharp edges and smooth the noise.

4.4 Tasks in the Fourth Group

We finally apply our method to image texture removal which belongs to the tasks in the fourth group. It aims at extracting salient meaningful structures while removing small complex texture patterns. Many meaningful structures can be formed by or appear over textured surfaces in natural images. Extracting these structures is challenging but is of great practical importance, which can benefit a number of applications, such as image vectorization, edge simplification and detection, content-aware image resizing [68], etc. The challenge of this task is that it requires structure-preserving smoothing rather than edge-preserving smoothing. Fig. 19(a) shows a classical example of image texture removal: the small textures with strong edges should be smoothed out while the salient structures with weak edges should be preserved. The SP-1 mode is adopted in our method for this task. The value of $\lambda$ varies for different input images as the size of the structures to be removed can be quite different among different images. The other parameters are fixed as those in Tab. 1. Fig. 19 shows the results of the recent state-of-the-art approaches and ours. The joint convolutional analysis and synthesis sparse (JCAS) model [26] and rolling guidance filter (RGF) [14] can well remove the textures, but the resulting edges are also blurred. The relative total variation (RTV) method [8], mutually guided...
image filtering (muGIF) [65], the deep learning approach based on fully-convolutional networks (FCN) [32] and decoupled learning approach [38] cannot completely remove the textures, in addition, the weak edges of the salient structures have also been smoothed out in their results. Our method can both preserve the weak edges of the salient structures and remove the small textures.

4.5 Computation Efficiency Analysis

We also analyze the computation efficiency of our method. The test is performed on an i5 CPU with 32GB memory and a NVIDIA Titan RTX GPU. Both color images and gray images of 5 classical image resolutions are used for evaluation, which is detailed in Tab. 6. We compare our method of different modes against some baselines including deep learning based methods [32, 57, 38, 62] in the first group, local average based approaches [19, 67, 8] in the second group and global optimization based ones [5, 7, 8, 65] in the third group, as shown in Tab. 5. All the deep learning based approaches are tested on the GPU while the rest compared methods are evaluated on the CPU.

Generally, most of the deep learning approaches and the local methods are much faster than ours and the other global methods. Our method of the EP-1 mode and the SP-2 mode also runs quite fast, which is the fastest among the compared global methods. When \( r_d = 1, r_s = 1 \) ( \( r = 1 \) in Tab. 5), the computational cost of our method of the other modes (SP-1, EP-2 and EP&SP) is also not quite high, and our method is slightly faster than muGIF [65]. However, as the value of \( r_d \) and \( r_s \) increases, the computational cost of our method increases greatly as shown in Tab. 6. It drops when we adopt the dilated neighborhood with \( s = 2 \). This can lead to a speedup of \( 2.5 \times \) to our method, which is now slightly faster than the SD filter [7].

5 Conclusion and Limitations

We propose a generalized framework for edge-preserving and structure-preserving image smoothing. We first introduce the truncated Huber penalty function which shows strong flexibility. Then a robust framework is presented. When combined with the flexibility of the truncated Huber penalty function, our framework is able to achieve different or even contradictory smoothing behaviors under different parameter settings. This is different from most previous approaches of which the inherent smoothing natures are usually fixed. Our method is also able to yield the smoothing behavior that is seldom achieved by previous approaches. It thus enables our method capable of more challenging cases which are not well handled by previous approaches. An efficient numerical solution to our model is proposed with the convergence theoretically guaranteed. We further provide a simple yet effective solution to reduce the computational cost of our method with the performance still maintained. The effectiveness of our method is demonstrated through comprehensive experimental results in a number of applications.

The limitations of our method are twofold. The first one is that our method has more parameters, which makes our method more complex than most existing approaches. This should be treated as the tradeoff between the complexity and the flexibility: it is these parameters that enable our model to enjoy the strong flexibility. In addition, we have fixed most of the parameters for different tasks as illustrated in Tab. 4. The property analysis of our model in Sec. 3.4 can also work as the guideline for the choice of the rest unfixed parameters. The second one is that our method is not time efficient as shown in Tab. 6. Our method is thus not applicable to real-time image processing tasks.

APPENDIX A

This appendix presents the proof of Eq. 7 and Eq. 8 in Sec. 3.3. Given the Huber penalty function \( h(x) \) and truncated Huber penalty function \( h_T(x) \) defined as Eq. 2 and Eq. 1, then \( h(x) \) and \( h_T(x) \) are related through the following equation:

\[
h_T(x) = \min_y g(x, y)
\]

where \( g(x, y) = h(x - y) + (b - \frac{1}{2}) |y|_0 \), s.t. \( b \geq a \),

here \( |y|_0 \) is the \( L_0 \) norm of \( y \):

\[
|y|_0 = 0 \quad \text{if} \quad y = 0 \quad \text{and} \quad |y|_0 = 1 \quad \text{if} \quad y \neq 0.
\]

The minimum of Eq. (22) is achieved on the condition:

\[
y = \begin{cases} 
0, & |x| \leq b \\
0, & |x| > b 
\end{cases}
\]
Proof:

Case 1: when $|x| < a$:

- when $y = 0$, since $|x - 0| < a$, we have:
  $$g(x, 0) \triangleq g_1(x, y) = \frac{1}{2a}x^2,$$
  (24)
  - when $y \neq 0$, we have:
  $$g(x, y) \triangleq \left\{ \begin{array}{ll}
g_{2}(x, y) = \frac{1}{2a}(|x - 0|^2 + (b - \frac{a}{2})), & |x - y| < a \\
g_{3}(x, y) = |x - y| - \frac{a}{2} + (b - \frac{a}{2}), & |x - y| \geq a
\end{array} \right..$$
  (25)

We have the following inequalities:

$$\left\{ \begin{array}{ll}
b - \frac{a}{2} \leq g_1(x, y) \leq \frac{a}{2} \\
g_2(x, y) \geq b \\
g_3(x, y) \geq b
\end{array} \right.,$$
(26)

since $b \geq a$, then $b - \frac{a}{2} \geq \frac{a}{2}$, and thus $g_2(x, y) \geq b - \frac{a}{2} \geq \frac{a}{2}$.

Finally, we have:

$$g_1(x, y) < g_2(x, y) < g_3(x, y).$$
(27)

Accordingly, the minimum of Eq. (22) is $g_1(x, y)$ which is achieved when $y = 0$:

$$\min_y g(x, y) = g_1(x, y) = g(x, 0) = \frac{1}{2a}x^2.$$  
(28)

Case 2: when $a \leq |x| \leq b$:

- when $y = 0$, since $|x - 0| \geq a$, we have:
  $$g(x, 0) \triangleq g_1(x, y) = |x| - \frac{a}{2},$$
  (29)
  - when $y \neq 0$, we have:
  $$g(x, y) \triangleq \left\{ \begin{array}{ll}
g_{2}(x, y) = \frac{1}{2a}(|x - 0|^2 + (b - \frac{a}{2})), & |x - y| < a \\
g_{3}(x, y) = |x - y| - \frac{a}{2} + (b - \frac{a}{2}), & |x - y| \geq a
\end{array} \right..$$
  (30)

We have the following inequalities:

$$\left\{ \begin{array}{ll}
b \leq g_1(x, y) \leq b - \frac{a}{2} \\
b - \frac{a}{2} \leq g_2(x, y) < b \\
g_3(x, y) \geq b
\end{array} \right..$$
(31)

we then have:

$$g_1(x, y) \leq g_2(x, y) < g_3(x, y).$$
(32)

Accordingly, the minimum of Eq. (22) is $g_1(x, y)$ which is achieved when $y = 0$:

$$\min_y g(x, y) = g_1(x, y) = g(x, 0) = |x| - \frac{a}{2}.$$  
(33)

Case 3: when $|x| > b$:

- when $y = 0$, since $|x - 0| > b$, we have:
  $$g(x, 0) \triangleq g_1(x, y) = |x| - \frac{a}{2},$$
  (34)
  - when $y \neq 0$, we have:
  $$g(x, y) \triangleq \left\{ \begin{array}{ll}
g_{2}(x, y) = \frac{1}{2a}(|x - 0|^2 + (b - \frac{a}{2})), & |x - y| < a \\
g_{3}(x, y) = |x - y| - \frac{a}{2} + (b - \frac{a}{2}), & |x - y| \geq a
\end{array} \right..$$
  (35)

We have the following inequalities:

$$\left\{ \begin{array}{ll}
\frac{a}{2} \leq g_1(x, y) < b - \frac{a}{2} \\
g_2(x, y) < b \\
g_3(x, y) \geq b
\end{array} \right.,$$
(36)

from Eq. (36) we can observe that the minimum value among $g_1(x, y), g_2(x, y)$ and $g_3(x, y)$ is $b - \frac{a}{2}$ which is achieved when $y = x$ in $g_2(x, y)$, i.e.:

$$\min_y g(x, y) = g_2(x, y) = x = b - \frac{a}{2}.$$  
(37)

Finally, Eq. (28), Eq. (33) and Eq. (37) consist with Eq. (22) and the corresponding optimum conditions are equal to Eq. (23). By simply replacing $x$ with $\nabla^*_i, j$ and $y$ with $l^*_i, j$, we get Eq. (7) and Eq. (8) in Sec. 3.3.

**APPENDIX B**

This appendix provides the proof of Eq. (11) and Eq. (12) in Sec. 3.3. The proof is based on the following theory in convex optimization [69]: for a function $f(x)$, its conjugated function $g(y)$ is defined as $g(y) = \max_x \{yx - f(x)\}$ which is convex. If $f(x)$ is continuous and convex, then we further have $f(x) = \max_y \{yx - g(y)\}$.

Given $h(x)$ defined as Eq. (2), it is clear that $h(x)$ is symmetric, i.e., $h(-x) = h(x)$. Thus, we only need to focus on its property for $x \geq 0$. A new function $\theta(x)$ is first defined as:

$$\theta(x) = h(\sqrt{x}), \ x \geq 0.$$  
(38)

The function $-\theta(x)$ is convex. Defining its conjugated function as:

$$\psi(y) = \max_{x \geq 0} \phi(x, y), \ \text{where} \ \phi(x, y) = (-y)x - (-\theta(x)),$$
(39)

Then by using the property of the convexity of $-\theta(x)$, we have:

$$\theta(x) = \max_y \{(y) - \psi(y)\} = \min_y \{(yx + \psi(y)\}$$
(40)

Since we have $x \geq 0$ in Eq. (39) and Eq. (40), then $x$ can be replaced with $x^2$. By using $\theta(x') = h(x)$, we have:

$$h(x) = \min_y \{yx^2 + \psi(y)\}$$
(41)

By setting $x = \nabla^*_i, j - l^*_i, j$ and $y = \mu^*_i, j$, we get Eq. (11) in Sec. 3.3.

The optimum condition of $\psi(y, x)$ in Eq. (39) can be obtained by setting $\frac{\partial \psi(x, y)}{\partial x} = -\dot{y} + \theta'(x) = 0$, i.e.:

$$y = \theta'(x) = \frac{h'(\sqrt{x})}{2\sqrt{x}} = \frac{h(x)}{2x}.$$  
(42)

By setting $x = \nabla^*_i, j - l^*_i, j$, $y = \mu^*_i, j$, and substituting the expression of $h'(x)$ into Eq. (42), we get Eq. (12) in Sec. 3.3.

**REFERENCES**

[1] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, “Edge-preserving decompositions for multi-scale tone and detail manipulation,” in ACM Transactions on Graphics (TOG), vol. 27, no. 3. ACM, 2008, p. 67.

[2] R. Fattal, M. Agrawala, and S. Rusinkiewicz, “Multiscale shape and detail enhancement from multi-light image collections,” in ACM Transactions on Graphics (TOG), vol. 26, no. 3. ACM, 2007, p. 51.

[3] K. He, J. Sun, and X. Tang, “Guided image filtering,” IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 35, no. 6, pp. 1397–1409, 2013.
Pingping Zhang received the B.E. degree in mathematics and applied mathematics from Henan Normal University (HNU), Xinxiang, China, in 2012, and the Ph.D. degree in signal and information processing from the Dalain University of Technology (DUT), Dalain, China, in 2020. He is currently an Associate Professor with the School of Artificial Intelligence, DUT. His research interests include deep learning, saliency detection, object tracking, and semantic segmentation.

Yinjie Lei (Member, IEEE) received the M.S. degree in image processing from Sicuan University (SCU), China, in 2009, and the Ph.D. degree in computer vision from The University of Western Australia (UWA), Australia, in 2013. Since 2017, he has been serving as the Vice Dean of the College of Electronics and Information Engineering, SCU, where he is currently an Associate Professor. He has authored over 60 journals/conference papers. His main research interests include deep learning, 3D vision, and semantic segmentation.

Xiaolin Huang received the B.S. degree in control science and engineering, and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. From 2012 to 2015, he worked as a postdoctoral researcher in ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that he was selected as an Alexander von Humboldt Fellow and working in Pattern Recognition Lab, the Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. From 2016, he has been an Associate Professor at Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. In 2017, he was awarded by "1000-Talent Plan" (Young Program). His current research areas include machine learning and optimization.

Jie Yang received his Ph.D. from the Department of Computer Science, Hamburg University, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China. He has led many research projects (e.g., National Science Foundation, 863 National High Tech. Plan). His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.

Michael Ng received the B.S. and M.Phil. degrees from the University of Hong Kong in 1990 and 1992, respectively, and the Ph.D. degree from the Chinese University of Hong Kong in 1995. He was a Chair Professor in Department of Mathematics at Hong Kong Baptist University from 2006 to 2019. He is currently a Chair Professor in Research Division of Mathematical and Statistical Science at The University of Hong Kong. His research interests include bioinformatics, image processing, scientific computing, and data mining. He was selected for the 2017 Class of Fellows of the Society for Industrial and Applied Mathematics. He obtained the Feng Kang Prize for his significant contributions in scientific computing. He serves on the Editorial Board members of several international journals.