Image Structure Retrieval via L_0 Minimization

Yujing Sun, Scott Schaefer, Wenping Wang

Abstract—Retrieving salient structure from textured images is an important but difficult problem in computer vision because texture, which can be irregular, anisotropic, non-uniform and complex, shares many of the same properties as structure. Observing that salient structure in a textured image should be piece-wise smooth, we present a method to retrieve such structures using an L_0 minimization of a modified form of the relative total variation metric. Thanks to the characteristics shared by texture and small structures, our method is effective at retrieving structure based on scale as well. Our method outperforms state-of-art methods in texture removal as well as scale-space filtering. We also demonstrate our method's ability in other applications such as edge detection, clip art compression artifact removal, and inverse half-toning.

Index Terms—Texture removal, image smoothing, L_0 sparsity

1 Introduction

Painting on textured surfaces has long been an important art form. From oil paintings on canvas to graffiti on brick walls, images on textured surfaces are everywhere. Complicated patterns can also be easily knitted into fine crafts such as a sweater or embroidery using cross-stitch. All these works are similar in that salient structure is blended into texture patterns. The human visual system can easily distinguish the salient image from the underlying texture details. However, the task of retrieving structure from texture patterns for a computer is much more challenging since texture, which can be irregular, anisotropic, non-uniform and complex, shares similar properties with major image structure, namely strong gradients. As a result, it can be difficult for an algorithm to distinguish the main image structure from texture automatically.

Most of the existing detexturing operators [1], [2], [3], [4], [5], [6], [7], if not all, define various metrics to locate texture and then extract structure by blurring an image in textured regions while keeping transitions on structure edges. Recently Xu et al. [7] proposed a method based on a *relative total variation* (RTV) metric, which outperforms most detexturing approaches. Although effective in removing texture, RTV [7] will decrease the transition along major edges, resulting in a blurry effect. While there is a tradeoff between texture removal and edge preservation, by improving the performance of the RTV metric while making use of the sparsity of L_0 norm, our method outperforms RTV [7] and other existing methods in better sharpening structure edges while removing texture patterns.

In addition to texture, small high-contrast features could also cause many image processing techniques, such as content aware editing, edge detection, shape matching etc., to produce unsatisfactory results. Recently, many scale-space operators have been proposed to handle high-contrast details. The Rolling Guided Filter (RGF) [8] is one such method, which removes features based on their scale. However, RGF [8] can shift edges and corners. By combining static and dynamic guidance, SDF [9] fixes the edge shift problem but still has difficulty preserving low-contrast, large structures. Tree Filters [10] and its variants [1], [11] make use of tree-like structure, such as a minimum spanning tree (MST) or segment graph to connect pixels and perform image smoothing globally along the structure. If these connections cross strong edges, these edges will be corrupted [11]. As a result, the tree-like filters can introduce "leaks" on major edges. Thanks to the similar characteristics shared by textures and small structure, our method can work as a scale-space filter and produce better results than state-of-art methods.

In this paper, we show how to combine the detexturing strength of RTV with the edge preserving abilities of L_0 minimization to extract salient structures from images while eliminating texture and/or fine-scale features. Such a combination generates a nonlinear optimization, which can be difficult to solve. Instead we show how to transform the problem into a sequence of simpler optimization steps that converges to the solution to of the overall nonlinear problem. We demonstrate the effectiveness of our method at removing high-contrast details while maintaining structure where the original L_0 gradient minimization [12] often fails and in scale-based filtering where RTV [7] performs poorly.

2 Related Work

Texture Removal In the past few years, texture removal has been well studied. Subr et al. [6] smoothed texture by averaging minimal and maximal envelopes based on the observation that texture patterns oscillate locally between extrema. Buades et al. [2] used a nonlinear low pass-high pass filter pair to compute a local total variation and to perform structure/texture decomposition. In contrast to intensity based methods, Karacan et al. [4] used the covariance of patch features, which leverages the repetitive nature of texture, to remove textures. Cho et al. [3] extended the bilateral filter to perform texture removal (BTF), the core

- *Y. Sun and W. Wang are with the Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: {yjsun, wenping}@cs.hku.hk*
- *S. Schaefer is with the Department of Computer Science, 3112 Texas A& M University, College Station, TX 77843-3112. E-mail: schaefer@cs.tamu.edu*

of which is to capture the texture information by *patch shift* and to ensure proper differentiation of texture and structure. Ono et al. [13] performed cartoon/texture decomposition based on the blockwise low-rank nature of texture but is quite slow. Buades and Lisani [14] presented a directional operator to detexture image by filtering toward largest rate deduction of local total variation. However, all the methods mentioned above over-blurs salient edges to some extent. By constructing tree weights, tree-like filters [1], [10], [11] successfully gets rid of "halo" problems, but one misclassified pixel can corrupt a major edge introducing a "leak" artifact. Su et al. [15] combine multiple filters to remove texture.Total variation (TV) and modifications thereof [5], [16], [17], [18], [19] are known to be one category of classic methods in detexturing by minimizing the L_1 norm of gradients to enforce structural similarity between the input and output. However, the traditional total variation approach blurs major edges when textures have similar or larger gradients than structures. Recently, Xu et al. [7] greatly improved the performance of total variation by defining a relative total variation (RTV) metric.

Structure-preserving filtering Even though not designed for texture/detail removal, edge-preserving image smoothing techniques can achieve a similar goal. Local methods include bilateral filters (BF) [20], [21], [22], [23], [24], [25], guided filters (GF) [8], [9], [26], [27], mode and median filters [28], [29], [30], [31], [32], domain transfer [33], geodesic filters [34], [35], multipoint filters [36], joint filters [37], [38], and scale-space filters [8], [9]. These local filters smooth images by updating each pixel value according to various functions defined upon its spatial and color differences with neighboring pixels. Generally, BF and GF cannot get rid of blurry artifacts [11]. Mode and median filters are effective in removing high contrast noise but tend to produce oscillations in the presence of signals that oscillate quickly. Geodesic filters [34], [35], RGF [8] and SDF [9] can perform scale-space filtering in terms of removing fine scale details without degrading structure edges. SDF [9] sharpens edges well but has difficulty preserving relatively low contrast structures. Global methods are mainly optimization-based including weighted least squares (WLS) [39] and sparsity minimization [5], [12], [40]. For these methods, images are processed by optimizing global functions involving a weighted L_2 norm [39], L_1 norm [5], [40], or L_0 norm [12]. Note that these global methods are all gradient-based operators and thus do not remove high contrast details well. More recently, other image filters [41], [42] to denoise images have been proposed. Although effective, they intend to maintain texture details while reducing noise whereas our goal is to remove texture/detail.

2.1 L_0 Minimization

The L_0 norm directly measures sparsity of a vector, and minimizing the L_0 norm produces sparse solutions. Hence, we take advantage of L_0 minimization to extract salient structure from texture. Besides image smoothing [12], the L_0 norm has been adopted to mesh denoising [43], point set denoising [44], image deblurring [45] and intrinsic image decomposition [46], [47]. However, optimizing the L_0 norm is difficult in practice. Xu et al. [12], He et al. [43], and Sun et al. [44] adopted a splitting scheme to solve an L_0 minimization problem to smooth images, meshes, and point clouds respectively. Since we use a similar splitting scheme in our approach, we will briefly review this technique.

The L_0 norm of a vector ϑ is defined as the number of non-zero entries. Given a signal $\hat{X}$ and a differential operator f, X can be minimized by

$$\min_X |X - \hat{X}|^2 + \lambda |f(X)|_0, \tag{1}$$

where the first term is a data fidelity term, λ is the smoothing parameter and $|f(X)|_0$ denotes the L_0 norm of $f(X)$. To optimize Equation 1, an auxiliary variable μ can be introduced to form

$$\min_{X,\mu} |X - \hat{X}|^2 + \beta |f(X) - \mu|^2 + \lambda |\mu|_0, \tag{2}$$

in which β is set to a small value initially and controls how quickly this problem approaches Equation 1.

Given an initial guess for X, we minimize Equation 2 by first holding X constant and solving for μ,

$$\min_\mu \beta |f(X) - \mu|^2 + \lambda |\mu|_0.$$

The solution of this minimization is given by $\mu_i = 0$ if $|f(X)_i|^2 < \frac{\lambda}{\beta}$ or $\mu_i = f(X)_i$ otherwise. Next, we hold μ constant and solve for X,

$$\min_X |X - \hat{X}|^2 + \beta |f(X) - \mu|^2.$$

This equation is quadratic in X and has a trivial global minimum. By repeating solving the two minimization problems with $\beta = 2\beta$, $f(X)$ is forced to approach μ when β approaches infinity.

2.2 Relative Total Variation

As stated above, RTV [7] is effective at distinguishing texture and structure. We review the computation of the metric here.

The RTV Υ of an image I is defined pixel-wise and contains two factors: the windowed total variation Γ, and the windowed inherent variation Φ. The windowed total variation counts the absolute spatial difference within a patch centered at pixel p in the d direction, which is written as

$$\Gamma_d(p) = \sum_{q \in \Omega(p)} w_{p,q} \cdot |\partial_d I_q|,$$

where $\Omega(p)$ is the rectangular region centered at pixel p and d is the direction. The windowed inherent variation Φ is defined as

$$\Phi_d(p) = |\sum_{q \in \Omega(p)} w_{p,q} \cdot \partial_d I_q|,$$

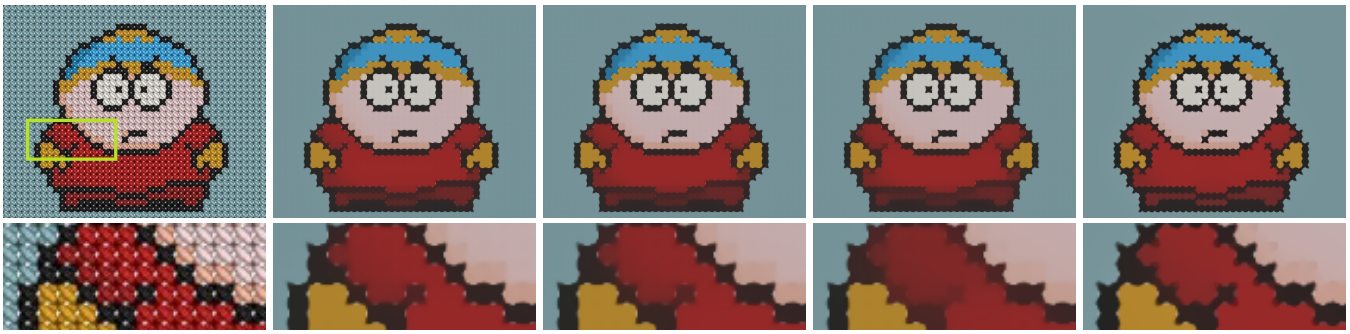
where $w_{p,q}$ is a weighting function

$$w_{p,q} = \exp(-\frac{|p-q|^2}{2\sigma^2}). \tag{3}$$

Then, relative total variation Υ is formed by combining Γ and Φ,

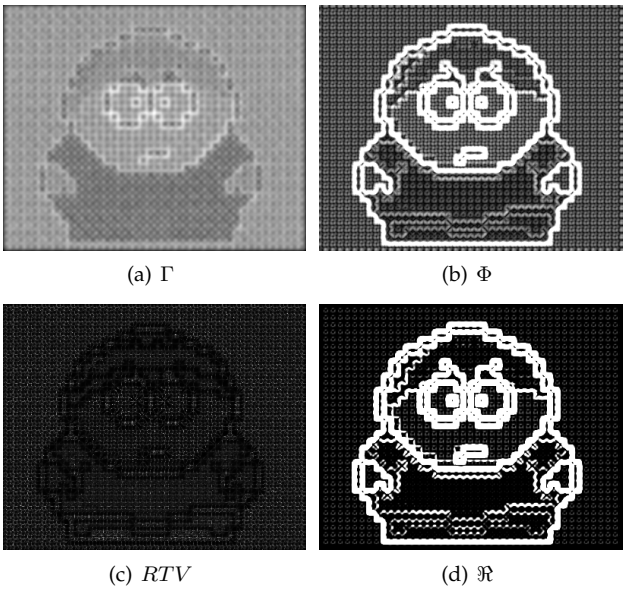
$$\Upsilon(p) = \frac{\Gamma_x(p)}{\Phi_x(p) + \epsilon} + \frac{\Gamma_y(p)}{\Phi_y(p) + \epsilon}$$

in which ϵ is a small positive number to avoid division by zero.



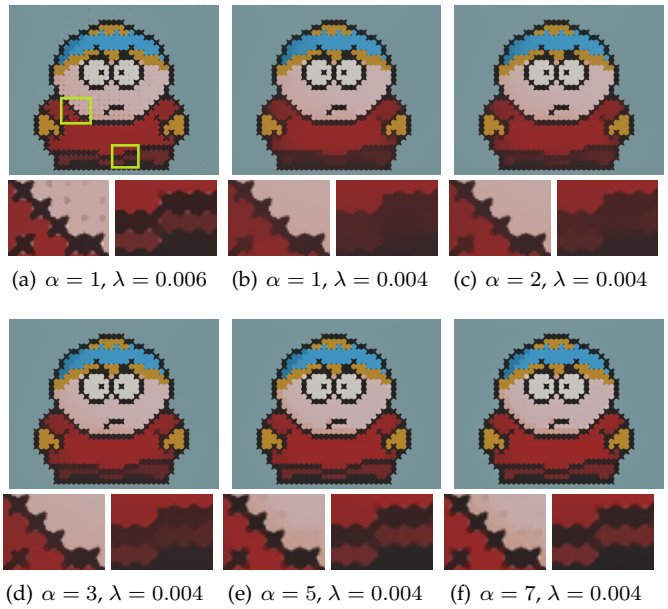
(a) Input (b) RTV, $\lambda = 0.014$ (c) RTV, $\lambda = 0.016$ (d) RTV, $\lambda = 0.018$ (e) our method, $\lambda = 0.004$

Fig. 1. Comparison between RTV [7] and our method on the *boy* image, $\sigma = 2.5$. *Input image courtesy Mark Delaney.*



(a) Γ (b) Φ (c) RTV (d) $\Re$

Fig. 2. Comparison between Γ, Φ, RTV and our modified RTV metric ($\Re$) on the boy image.



(a) $\alpha = 1, \lambda = 0.006$ (b) $\alpha = 1, \lambda = 0.004$ (c) $\alpha = 2, \lambda = 0.004$ (d) $\alpha = 3, \lambda = 0.004$ (e) $\alpha = 5, \lambda = 0.004$ (f) $\alpha = 7, \lambda = 0.004$

Fig. 3. Demonstration of our method using different α.

3 Approach

The ability of L_0 minimization has been well demonstrated in image smoothing [12], however, the same approach does not work well with texture patterns when the image gradient in textured regions is more prominent than that of salient edges, as shown in Figure 6, 7 and 9. The problem with minimizing the L_0 norm of the image gradients is that image gradients are a poor measure of whether or not texture exists within the image. In contrast, RTV distinguishes between structure and texture well. Hence, we would like to find an image close to the input whose measure of texture is zero. Moreover, the structures are typically sparse compared to the entire image, which makes the L_0 norm an ideal candidate for this problem.

However, we must make a few modifications to RTV to use this metric with an L_0 optimization. First, RTV is high in textured regions and low at structural edges in the image. While such a metric makes sense for an L_2 minimization, it is not compatible with the sparsity measured by the L_0 norm. Instead we modify the RTV metric slightly to make the metric small in textured regions and large along structure edges in the image. Then optimizing the L_0 norm of this modified metric will find an image whose measure of texture is zero except along structure edges that are sparse within the image.

Hence, our modified RTV metric is given as

$$\Re(p) = \Re_x(p) + \Re_y(p)$$

where

$$\Re_d(p) = s\Phi_d(p)^\alpha \Gamma_d(p).$$

Compared with the original RTV metric, we multiply by Φ rather than divide by Φ to achieve the desired behavior on different regions of the image. We include α as an exponent to accenuate the structure from the texture (we choose $\alpha = 7$ in our experiments), and s is simply a normalization factor. Figure 3 shows the results of using different values of α and figure 2 shows the difference between the original RTV metric on an image and our modified form $\Re$. Note that since RTV metric divides Γ by $(\Phi + \epsilon)$, the range of values is quite large if ϵ is close to zero since Φ may be zero in

some regions. To visualize the original RTV values in an image, we use a large $\epsilon = 0.1$ in figure 2.

3.1 L_0 Relative Total Variation Minimization

Let I be the input image, S the computed result, and $\Re$ the modified RTV metric distinguishing structure from texture, we solve the structure retrieval problem by minimizing

$$\min_S \sum_p |S_p - I_p|^2 + \lambda |\Re|_0,$$

where λ is the parameter controlling the smoothness of the image. The first term makes sure that the structure image does not stray too far away from the input and second term is the L_0 norm of the modified RTV metric, which attempts to remove texture while preserving structure.

Similar to [12], we adopt a splitting scheme to minimize this nonlinear objective function. Note that due to the nonlinear terms in $\Re$, the solution to this energy function is not trivial as was the case in the original L_0 smoothing work [12]. Hence, we will show how we transform the nonlinear portion of $\Re$ so that the optimization can be solved in an iterative manner.

We introduce auxiliary vectors ϕ and ρ corresponding to $\Re_x$ and $\Re_y$ and minimize the following energy function

$$\begin{aligned}\min_{S,\phi,\rho} \sum_p (S_p - I_p)^2 + \beta((\Re_x(p) - \phi_p)^2 + (\Re_y(p) - \rho_p)^2) \\ + \lambda|\phi + \rho|_0.\end{aligned} \tag{4}$$

Subproblem 1: Solve for ϕ and ρ.

To minimize this energy, we perform an alternating optimization where we first hold S constant and solve for ϕ and ρ,

$$\begin{aligned}\min_{\phi,\rho} \quad & \beta \sum_p ((\Re_x(p) - \phi_p)^2 + (\Re_y(p) - \rho_p)^2) \\ & + \lambda|\phi + \rho|_0,\end{aligned} \tag{5}$$

which reaches its minimum when

$$(\phi_p, \rho_p) = \begin{cases} (0,0) & \Re_x(p)^2 + \Re_y(p)^2 \le \frac{\lambda}{\beta}, \\ (\Re_x(p), \Re_y(p)) & otherwise. \end{cases} \tag{6}$$

Subproblem 2: Solve for S.

Fixing ϕ and ρ, we minimize equation 4 over S

$$\min_S \sum_p (S_p - I_p)^2 + \beta((\Re_x(p) - \phi_p)^2 + (\Re_y(p) - \rho_p)^2). \tag{7}$$

Since $\Re$ is nonlinear, this energy function is difficult to solve. Now we show how Eq. 7 can be decomposed into a simpler optimization. To simplify, let $\Psi_d(p) = s\Phi_d(p)^\alpha$, and $\Re_d(p)$ can be rewritten as

$$\Re_d(p) = \Psi_d(p) \cdot \Gamma_d(p).$$

Without loss of generality, we first consider the term in the x direction in Equation 7, $(\Re_x(p) - \phi_p)^2$. Expanding $\Gamma_x(p)$ in $\Re_x(p)$ yields

$$\begin{aligned}&(\Re_x(p) - \phi_p)^2 = (\Psi_x(p) \cdot \sum_{q \in \Omega(p)} w_{p,q} |\partial_x S_q| - \phi_p)^2 \\ &= (\Psi_x(p) \cdot w_{p,p} |\partial_x S_p| + \Psi_x(p) \cdot \sum_{q \in \Omega(p)}^{q \neq p} w_{p,q} |\partial_x S_q| - \phi_p)^2 \\ &\approx (\hat{\Psi}_x(p) \cdot w_{p,p} |\partial_x S_p| + \hat{\Psi}_x(p) \cdot \sum_{q \in \Omega(p)}^{q \neq p} w_{p,q} |\partial_x \hat{S}_q| - \phi_p)^2 \\ &= (k_{xp} |\partial_x S_p| + b_{xp})^2.\end{aligned} \tag{8}$$

where

$$k_{xp} = \hat{\Psi}_x(p) \cdot w_{p,p},$$

and

$$b_{xp} = \hat{\Psi}_x(p) \cdot \sum_{q \in \Omega(p)}^{q \neq p} w_{p,q} |\partial_x \hat{S}_q| - \phi_p.$$

$\hat{\Psi}$ and $\hat{S}$ are results from the previous iteration, and therefore k_{xp} and b_{xp} can be treated as constants. Thus, Equation 8 reaches minimum when $|\partial_x S_p| = -\frac{b_{xp}}{k_{xp}}$.Expanding $-\frac{b_{xp}}{k_{xp}}$, we obtain

$$\begin{aligned}|\partial_x S_p| = -\frac{b_{xp}}{k_{xp}} &= -\frac{\hat{\Psi}_x(p) \cdot \sum_{q \in \Omega(p)}^{q \neq p} w_{p,q} |\partial_x \hat{S}_q| - \phi_p}{\hat{\Psi}_x(p) \cdot w_{p,p}} \\ &= \begin{cases} -\sum_{q \in \Omega(p)}^{q \neq p} \frac{w_{p,q} |\partial_x \hat{S}_q|}{w_{p,p}} & \phi_p = 0, \\ |\partial_x \hat{S}_p| & \phi_p = \hat{\Re}_x(p). \end{cases}\end{aligned}$$

Note that $-\sum_{q \in \Omega(p)}^{q \neq p} \frac{w_{p,q} |\partial_x \hat{S}_q|}{w_{p,p}}$ is always negative, which is not possible unless $|\partial_x \hat{S}_p| = 0$. Thus, Equation 8 is minimized under the condition

$$\partial_x S_p = \begin{cases} 0 & \phi_p = 0, \\ \partial_x \hat{S}_p & \phi_p = \hat{\Re}_x(p). \end{cases}$$

A similar argument holds for $\Re_y(p)$.

As a result, objective function in Equation 7 can be transformed to the following quadratic optimization problem

$$\min_S \sum_p (S_p - I_p)^2 + \beta((\partial_x S_p - \eta_p)^2 + (\partial_y S_y - \gamma_p)^2), \tag{9}$$

where η, γ are auxiliary variables that depend solely on ϕ, ρ

$$(\eta_p, \gamma_p) = \begin{cases} (0,0) & (\phi_p, \rho_p) = (0,0), \\ (\partial_x S_p, \partial_y S_p) & (\phi_p, \rho_p) = (\Re_x(p), \Re_y(p)). \end{cases} \tag{10}$$

Algorithm 1 depicts pseudocode for this minimization process. Since Equation 9 converges to the solution to Equation 7 and is a quadratic function, our optimization has a closed-form solution. Results from different iterations are shown in Figure 4.

Algorithm 1 Structure Retrieval via L_0 Minimization

Input: image I, smoothing weight λ, detail scale σ, parameters β_0, β_{max}.
Initialization: $S^{(0)} \leftarrow I$, $\beta \leftarrow \beta_0$, $i \leftarrow 0$
repeat
  Compute $\Re$ from $S^{(i)}$ at each pixel p.
  Find ϕ_p and ρ_p in Eq. 6.
  Using ϕ_p and ρ_p, compute η_p and γ_p in Eq. 10.
  Using η_p and γ_p, minimize Eq. 9 for $S^{(i+1)}$.
  $\beta \leftarrow 2\beta$, $\sigma \leftarrow \sigma/2$, i++.
until $\beta > \beta_{max}$
Output: resulting image S

(a) Input (b) Iter 5 (c) Iter 15

Fig. 4. Results at different iterations. $\sigma = 3, \lambda = 0.005$. *Input image from Pinterest user Community of Jesus.*

3.2 Analysis

Parameters. Our method has two main parameters: σ (defined in Equation 3), which determines the spatial scale of texture/feature, and λ, which controls the amount of smoothing. The influence of increasing σ and λ is demonstrated in Figure 8. As shown in the figure, σ is the key parameter to distinguish salient structures from texture/details. We use tunable ranges of parameters for $\sigma \in [1, 5]$ and $\lambda \in [0.001, 0.005]$ in all the experiments. In our method, we decrease σ at each iteration by setting $\sigma = \sigma/2$ since we found that gradually decreasing σ helps to produce sharper edges.

Convergence. Our L_0 optimization has a closed-form solution at each step and reduces the error since both of the subproblems to solve for ϕ and ρ in Equation 5 and the sub-problem to solve for S in equation 9 have closed-form solutions. In practice, we only perform a finite number of iterations.

Implementation. In terms of timings, our algorithm took 10 to 30 iterations to produce the results in the paper. In each iteration, most of the computation time is spent calculating $\Re$ and on solving for S in Equation 9, which we accelerate using the Fourier Transform. When processing color images, for each channel, we compute $\Re$ and perform the L_0 minimization process separately. The proposed method takes about 5 seconds to process a 600×400 image with a single channel on an Intel Xeon X5460 3.16GHz with 8GB memory.

4 Experiments and Discussion

In all the experiments, we fine-tuned parameters for every method. Unless stated otherwise, we use $\mu = 50, v = 400$ for SDF [9] and $k = 5$ for RGF [8]. For BNN [13], we process each image channel separately.

4.1 Texture Removal

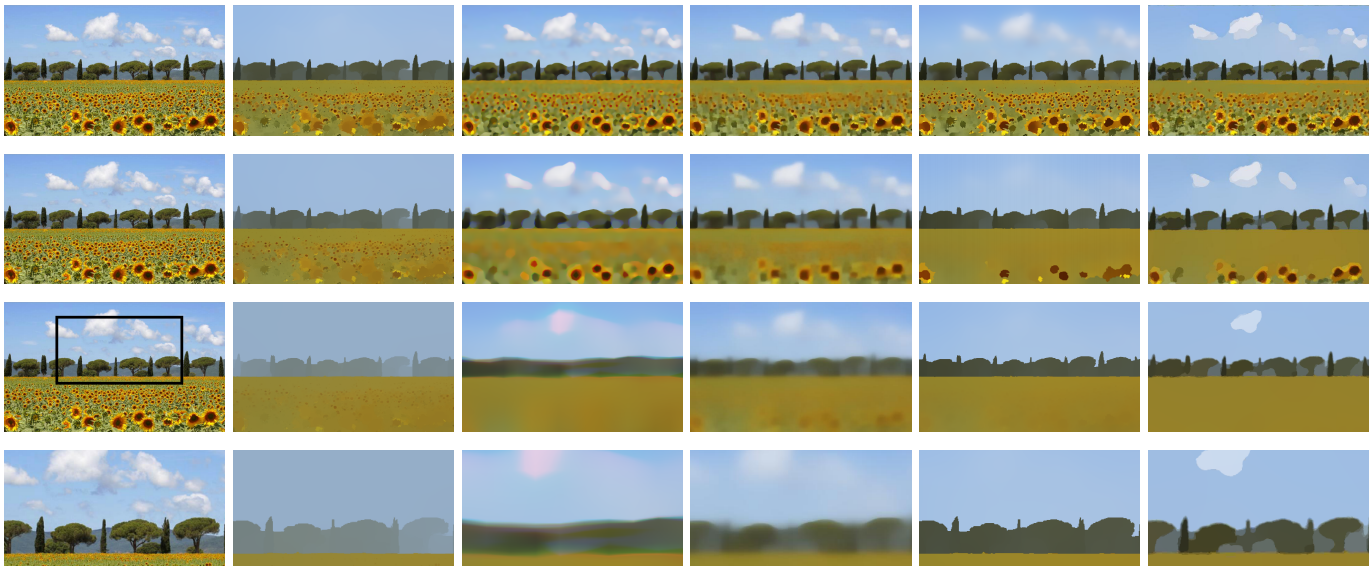
Texture Type. We consider texture to be surface patterns that are similar in appearance and local statistics [49], which can be isotropic or anisotropic. Our method makes no assumptions about the regularity, isotropic shape, or uniformity of the textures. Figure 6 shows a comparison of different methods that target texture removal on an image containing multiple types of texture, both isotropic and anisotropic. Scale-space filtering methods can be used for texture removal as well, and the comparison in Figure 7 compares our method with texture removal methods as well as scale-space filters on an image with irregular and nonuniform textures.

L_0 gradient minimization [12] is effective at sharpening and preserving strong edges but the gradient-based formulation prevents this method from removing high-contrast texture patterns. RGF [8] can remove texture details, but structures are over-smoothed as well. RGF [8] also results in a bumpy output due to the locality of its computations. SDF [9] has a better performance but still blurs certain features.

The locality of Region Covariance [4] creates similar results to the other local methods, causing a "halo" artifact. BTF [3] can better preserve structure due to the *patch shift* algorithm, but the method also fails to smooth the image sufficiently in a global manner and blurs edges to some extent. The classic total variation method [16] overblurs the structure too much. Even though the Directional filter [14] improves the edges preservation by filtering along the direction of largest local total variation rate deduction, blurry artifacts are still quite noticeable. BNN [13] cannot well maintain edges as well. Tree Filtering [1] is a weighted-average filter combining bilateral weights with a tree weight and produces smoother result. However, it suffers from the "leak" problem: certain major edges are corrupted. RTV [7] performs detexturing as a global optimization. But our method produces sharper edges while removing textures more effectively than RTV [7]. Even though $\Re$ is computed locally, our smoothing process is a global L_0 minimization, which produces higher quality results. Additional results produced by our method are included in the supplemental materials.

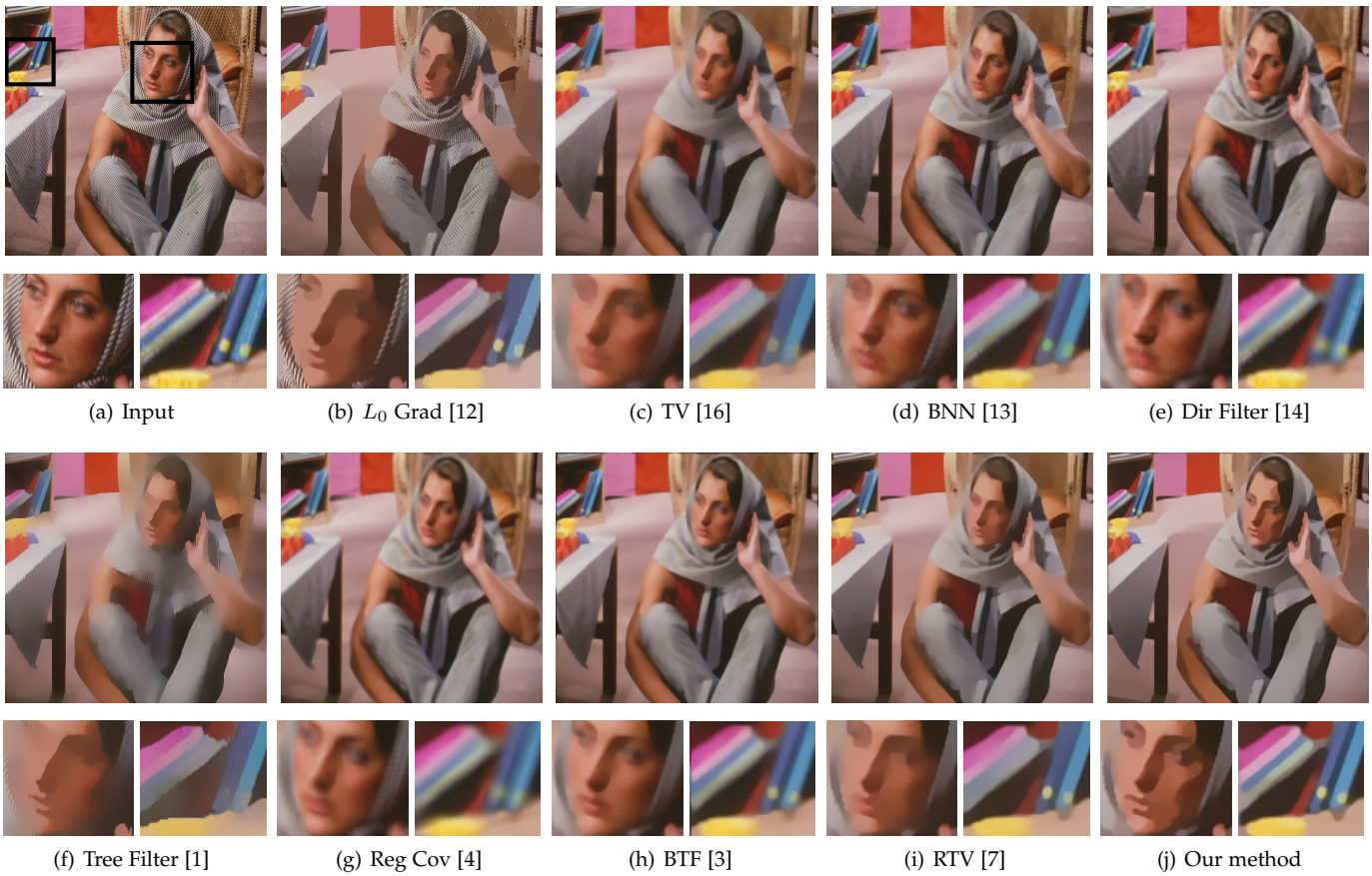
4.2 Scale-space Filtering

By varying σ, which controls the size of feature, our method can perform scale-based filtering as well. In figure 5, we show comparisons of our method with the state of the art on different smoothing scales. L_0 gradient minimization [12] and WLS [39] perform image smoothing by solving a global minimization on gradients. Since these methods are gradient-based operators, higher contrast features are



(a) Input (b) WLS [39] (c) RGF [8] (d) RTV [7] (e) SDF [9] (f) Our

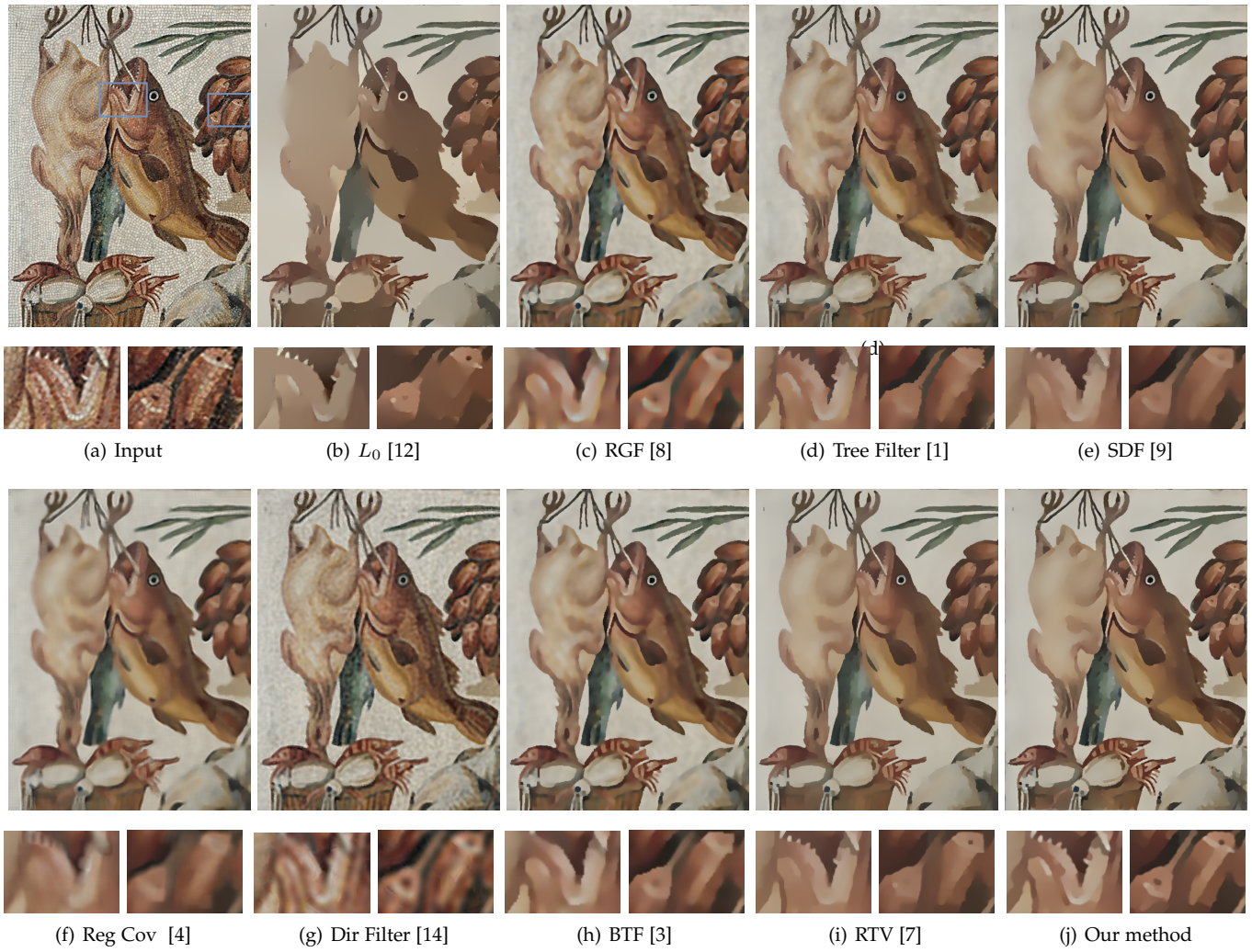
Fig. 5. Comparison of filtering based on scale. For all methods, parameters gradually increase from top to bottom. The last row shows the close-ups of the specified regions in row 3, respectively. (b) WLS [39] $\lambda = 5 \times 10^3, 3 \times 10^4, 2 \times 10^5, \mu = 40$. (c) RGF [8] $\sigma_s = 4, 10, 40, \sigma_r = 0.05$. (d) RTV [7] $\sigma = 4, 12, 20, \lambda = 0.02$. (e) SDF [9] $\lambda = 200, 1200, 3000$.(f) Ours $\sigma = 4, 12, 20, \lambda = 0.005$. *Input image from [9].*



(a) Input (b) L_0 Grad [12] (c) TV [16] (d) BNN [13] (e) Dir Filter [14]

(f) Tree Filter [1] (g) Reg Cov [4] (h) BTF [3] (i) RTV [7] (j) Our method

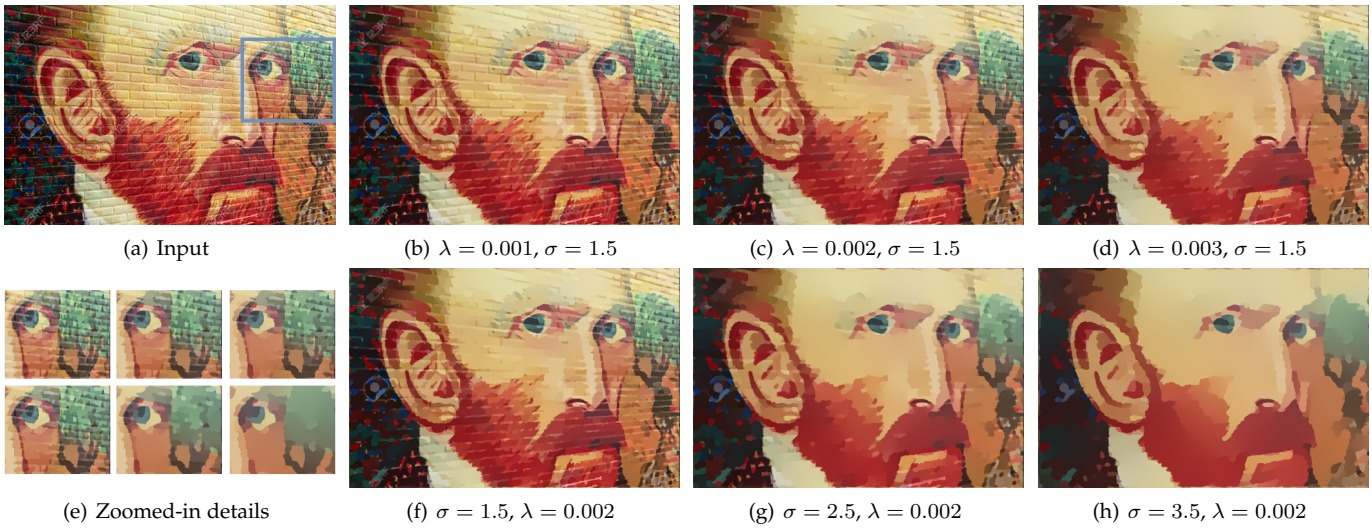
Fig. 6. Comparison with state-of-art detexturing methods on the *Barbara* image. Parameters: (b) $\lambda = 0.03$ (c) $\sigma = 3$ (d) $\gamma = 0.1, \lambda = 0.5$ (e) $\sigma = 4$ (f) $\sigma = 0.05, \sigma_s = 4, \sigma_r = 0.03$ (g) $k = 9, \sigma = 0.3, Model2$ (h) $k = 5, n_{itr} = 2$ (i) $\sigma = 2.5, \lambda = 0.01$ (j) $\sigma = 2.5, \lambda = 0.001$.



(a) Input (b) L_0 [12] (c) RGF [8] (d) Tree Filter [1] (e) SDF [9]

(f) Reg Cov [4] (g) Dir Filter [14] (h) BTF [3] (i) RTV [7] (j) Our method

Fig. 7. Another comparison of our method with existing image smoothing and detexturing methods. (b) $\lambda = 0.08$ (c) $\sigma_s = 5, \sigma_r = 0.05$ (d) $\sigma = 0.02, \sigma_s = 2, \sigma_r = 0.05$ (e) $\lambda = 50$, Gaussian kernel $([10, 10], 4)$ (f) $k = 15, \sigma = 0.2$, Model 1 (g) $\sigma = 4$ (h) $k = 5, n_{iter} = 3$ (i) $\sigma = 2, \lambda = 0.015$ (j) $\sigma = 2, \lambda = 0.002$.



(a) Input (b) $\lambda = 0.001, \sigma = 1.5$ (c) $\lambda = 0.002, \sigma = 1.5$ (d) $\lambda = 0.003, \sigma = 1.5$

(e) Zoomed-in details (f) $\sigma = 1.5, \lambda = 0.002$ (g) $\sigma = 2.5, \lambda = 0.002$ (h) $\sigma = 3.5, \lambda = 0.002$

Fig. 8. The effect of parameters on our optimization. λ is the overall smoothing parameter and does not significantly affect high-contrast texture removal (top row). σ distinguishes salient structures from textures (bottom row). *Input image from Tidy.ro label Discover.*

(a) Input (b) L_0 [12] (c) RGF [8] (d) RTV [7] (e) SDF [9] (f) Our method

Fig. 9. Another comparison on scale-space filtering at the resolution when the fence is removed. (b) $\lambda = 0.08$ (c) $\sigma_s = 5, \sigma_r = 0.07$ (d) $\lambda = 0.01, \sigma = 4$ (e) $\lambda = 60$, Gaussian kernal $([15, 15], 5)$ (f) $\lambda = 0.001, \sigma = 4$. *Input image from [48].*

always removed after lower contrast features without regards to feature size. RGF [8] and SDF [9] are scale-based separators. When increasing the smoothing parameters, larger structures are gradually removed. RGF [8] tends to shift/round off the structure edges while SDF [9] fixes the problem by using joint static and dynamic guidance. Even though SDF [9] is effective at localizing edges, it cannot preserve low-contrast, large structure edges well. As shown in figure 5(e), while the sunflowers in the foreground are maintained, the clouds, which are apparently larger than the flowers, are completely smoothed out. In addition, our method better preserves the shape of the trees as shown in figure 5 (row 4). Even though RTV [7] is not design for scale-based filtering, we add it in the comparison since our method is related to RTV. As shown, RTV [7] could smooth images based on scale but over-blurs certain edges to a great extent.

Another scale-space filtering comparison is shown in Figure 9. L_0 minimization fails to remove the fine-scale details of the fence at this resolution while losing some of the coarser details in the image. Both RGB and RTV remove fine-scale features but blur edges significantly. SDF performs well but still blurs some of the edges of the church. In contrast, our approach removes fine-scale details without blurring sharp edges.

4.3 Comparision with L_0 Gradient [12] and RTV [7]

As shown in Section 3.1, we solve the strucure retrieval problem in an l_0 minimization manner whereas the large image gradient in these textured regions effectively prevents the original L_0 gradient minimization [12] from removing these textures/features as demonstrated in Figure 6, 7 and 9.

Compared to RTV, our method better preserves sharp edges when eliminating texture patterns/details due to the used of the L_0 norm. In Figure 1, we show RTV results with varying λ. Similar to our method, RTV produces smoother result with larger λ. However, certain edges are smoothed out gradually as well. For fairness, we use the same σ for RTV and our method to process each test image and turning the smoothing parameter λ to produce the best results for both methods. λ is chosen based on the minimum principle, which is to take the minimal value that can completely removal texture patterns. Note that scale of λ in RTV and our method are different. For all the examples, we choose [0.01, 0.03] for RTV as instructed by the original paper [7] and [0.001, 0.005] for our method. As shown in all the examples, our method can better maintain salient edges than RTV [7].

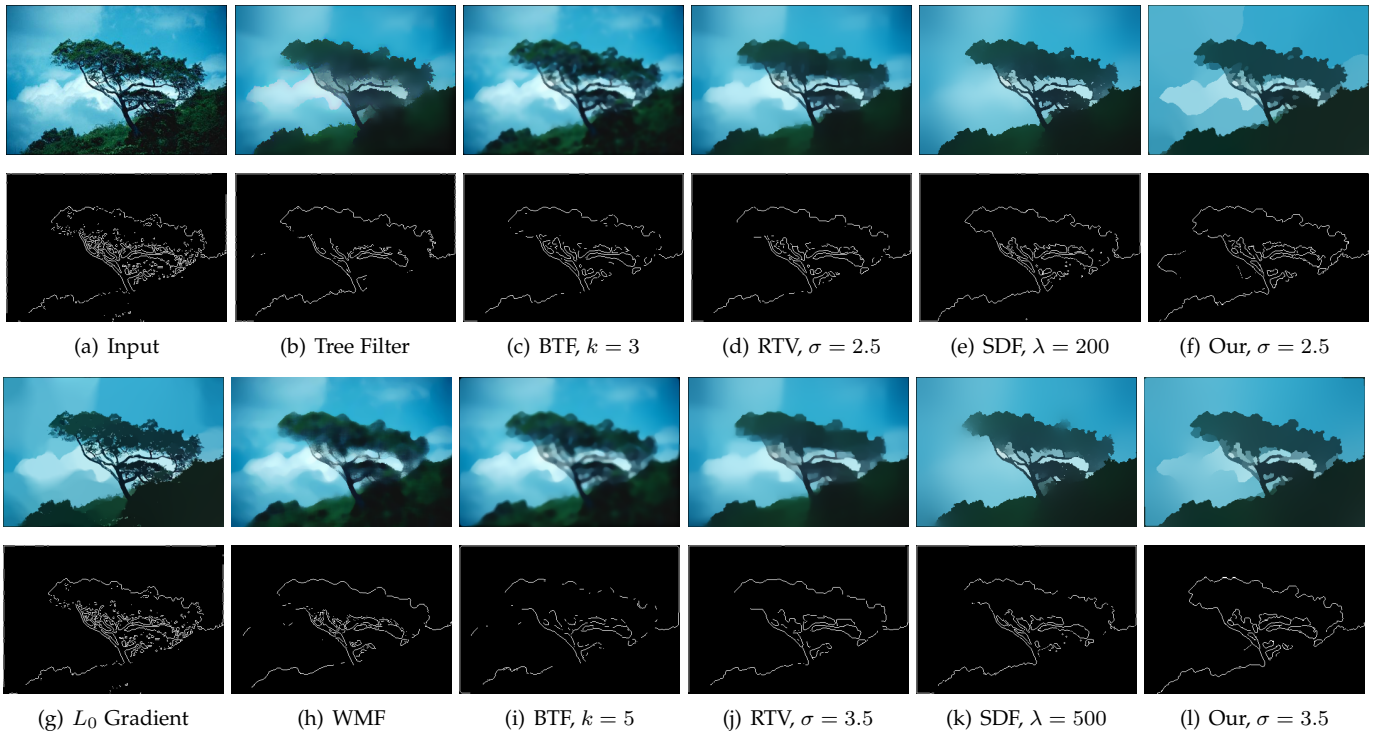
5 Applications

Texture smoothing is a useful tool of image editing and can have various applications. In this section, we show various applications including edge enhancement/simplification, clip-art artifact removal, and inverse halftoning.

5.1 Edge Simplification and Enhancement

Images, especially natural images, may contain trivial high-contrast texture/details/noises, which are not useful for humans to perceive the image content and potentially hinder edge detection operators from extracting meaningful edges. Edge simplification is an important pre-processing tool whose applications include image abstraction, segmentation, scene understanding, and shape matching to name a few.

Our method is capable of enhancing and simplifying edges. By reducing these trivial details, our method can help to produce clean edge maps. In figure 10, we apply the sobel edge detection operator to the result of various image smoothing methods. All of the filters improve the performance of the sobel edge detector in some ways. Among the methods, SDF [9] and our method have the best performance. Note that SDF completely removed the low-contrast large structure (figure 10(e)) and blurs some of the structure edges when details are fully smoothed (figure 10(k)).



(a) Input (b) Tree Filter (c) BTF, $k = 3$ (d) RTV, $\sigma = 2.5$ (e) SDF, $\lambda = 200$ (f) Our, $\sigma = 2.5$

(g) L_0 Gradient (h) WMF (i) BTF, $k = 5$ (j) RTV, $\sigma = 3.5$ (k) SDF, $\lambda = 500$ (l) Our, $\sigma = 3.5$

Fig. 10. Comparison on Edge detection and Simplification. Other parameter settings are: $\sigma_s = 3, \sigma_r = 0.03, \sigma = 0.1$ for tree filter, $n_{itr} = 3$ for BTF, $\lambda = 0.01$ for RTV, $\lambda = 0.03$ for L_0 gradient minimization, $r = 5, \sigma = 300$ for WMF and $\lambda = 0.003$(f), 0.005(l) for our method. *Input image from [9].*

(a) Input (b) RGF [8] (c) SDF [9] (d) Reg Cov [4] (e) BTF [3] (f) Our

Fig. 11. Application of inverse halftoning. *Input image from Digital Anarchy.*

5.2 Clip-Art Compression Artifact Removal

Lossy compressed cartoon images may contain severe artifacts. However, these cartoon images are often piecewise constant color before compression, and compression artifacts correlate with edges in the image. General image smoothing and denoising methods are not suitable to remove these artifacts [12] and will blur otherwise sharp boundaries. To remove these compression artifacts, [50] used a prior training procedure. In contrast, our method can faithfully reduce these artifacts without a prior learning process. Thanks to the L_0 minimization framework, both the traditional L_0 gradient minimization method [12] and our method demonstrate impressive results in this application. But when the artifacts have higher contrast, edges could by smoothed by the traditional L_0 gradient method. However, small high-contrast details have little influence on our method, and thus our method can produce sharper edges than the L_0 gradient method. A comparison of different methods on this application is shown in Figure 12.

5.3 Inverse Halftone

Our method can also be applied to perform inverse halftoning, which aims at removing stipple dots from the halftone image. Although our method is not design to solve this problem, it demonstrates good performance in terms of keeping structures while reducing the dots. We show an inverse halftoning application in Figure 11. RGF suffers from "bumpy" artifacts and rounded features while SDF removes the shading on the face. Both reg cov [4] and BTF [3] applied their approaches to solve this problem, but our methods better preserves transitions along edges and maintains more details and shading.

(a) Input (b) SDF (c) L_0 Grad

(d) BTF (e) RTV (f) Our

Fig. 12. Comparison on removal compression artifacts from cartoon images. Parameters: (b) $\lambda = 30$ (c) $\lambda = 0.03$, (d) $k = 3, n_{itr} = 5$ (e) $\sigma = 2, \lambda = 0.015$,(f) $\sigma = 2, \lambda = 0.0025$. *Input image from [7].*

6 Conclusion and Limitations

To conclude, we have presented a method to retrieve salient structure. By incorporating an L_0 minimization with a modified form of the RTV metric, we demonstrate our method's ability in detexture and scale-space filtering.

Our method does have some limitations, which cannot handle images with small-scale features. When an image contains fine details comparable to textures in size, our method could fail because $\Re$ may fail to distinguish such details from textures. However, this failure is also the reason our method performs well when used for scale-based filtering.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments and Dr. Hao Xiong for his great help and discussion on the early stage. This work was supported by NSF Career award IIS 1148976.

References

[1] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang, "Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree," *Image Processing, IEEE Transactions on*, vol. 23, no. 2, pp. 555–569, 2014.

[2] A. Buades, T. M. Le, J.-M. Morel, L. Vese *et al.*, "Fast cartoon+ texture image filters," *Image Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 1978–1986, 2010.

[3] H. Cho, H. Lee, H. Kang, and S. Lee, "Bilateral texture filtering," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 128, 2014.

[4] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 176, 2013.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[6] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 147, 2009.

[7] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 139, 2012.

[8] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 815–830.

[9] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[10] Q. Yang, "Stereo matching using tree filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 4, pp. 834–846, 2015.

[11] F. Zhang, L. Dai, S. Xiang, and X. Zhang, "Segment graph based image filtering: Fast structure-preserving smoothing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 361–369.

[12] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, Dec. 2011.

[13] S. Ono, T. Miyata, and I. Yamada, "Cartoon-texture image decomposition using blockwise low-rank texture characterization," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1128–1142, 2014.

[14] A. Buades and J. Lisani, "Directional filters for color cartoon+ texture image and video decomposition," *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 125–135, 2016.

[15] Z. Su, X. Luo, Z. Deng, Y. Liang, and Z. Ji, "Edge-preserving texture suppression filter based on joint filtering schemes," *Multimedia, IEEE Transactions on*, vol. 15, no. 3, pp. 535–548, 2013.

[16] V. Duval, J.-F. Aujol, and Y. Gousseau, "The tv-l1 model: a geometric point of view," *Multiscale Modeling & Simulation*, vol. 8, no. 1, pp. 154–189, 2009.

[17] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition-modeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.

[18] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*. American Mathematical Soc., 2001, vol. 22.

[19] W. Yin, D. Goldfarb, and S. Osher, "Image cartoon-texture decomposition and feature selection using the total variation regularized l^ 1 functional," in *VLSM*. Springer, 2005, pp. 73–84.

[20] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 103, 2007.

[21] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 257–266, 2002.

[22] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *ECCV 2006*. Springer, 2006, pp. 568–580.

[23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[24] Q. Yang, "Recursive bilateral filtering," in *ECCV 2012*. Springer, 2012, pp. 399–413.

[25] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time o(1) bilateral filtering," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, 2009, pp. 557–564.

[26] K. He, J. Sun, and X. Tang, "Guided image filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, 2013.

[27] F. Kou, W. Chen, C. Wen, and Z. Li, "Gradient domain guided image filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4528 – 4539, 2015.

[28] M. Kass and J. Solomon, "Smoothed local histogram filters," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 100, 2010.
[29] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2013, pp. 49–56.
[30] J. van de Weijer and R. Van den Boomgaard, "Local mode filtering," in *Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Conference on*, 2001, pp. II–428.
[31] B. Weiss, "Fast median and bilateral filtering," *Acm Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 519–526, 2006.
[32] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (wmf)," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 2830–2837.
[33] E. S. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 69, 2011.
[34] A. Criminisi, T. Sharp, C. Rother, and P. Pérez, "Geodesic image and video editing." *ACM Trans. Graph.*, vol. 29, no. 5, p. 134, 2010.
[35] X. Chen, S. Kang, J. Yang, and J. Yu, "Fast patch-based denoising using approximated patch geodesic paths," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1211–1218.
[36] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do, "Cross-based local multipoint filtering," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 430–437.
[37] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 96, 2007.
[38] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 664–672, 2004.
[39] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 67, 2008.
[40] S. Bi, X. Han, and Y. Yu, "An l-1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 78, 2015.
[41] H. Liu, R. Xiong, J. Zhang, and W. Gao, "Image denoising via adaptive soft-thresholding based on non-local samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 484–492.
[42] J.-H. R. Chang and Y.-C. F. Wang, "Propagated image filtering," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 10–18.
[43] L. He and S. Schaefer, "Mesh denoising via l0 minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, Jul. 2013.
[44] Y. Sun, S. Schaefer, and W. Wang, "Denoising point sets via l0 minimization," *Computer Aided Geometric Design*, vol. 35, pp. 2–15, 2015.
[45] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1107–1114.
[46] X. Nie, W. Feng, L. Wan, H. Dai, and C.-M. Pun, "Intrinsic image decomposition by hierarchical l 0 sparsity," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2014, pp. 1–6.
[47] S. Ding, B. Sheng, Z. Xie, and L. Ma, "Intrinsic image estimation using near-l_0 sparse optimization," *The Visual Computer*, vol. 33, no. 3, pp. 355–369, 2017.
[48] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.
[49] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009, pp. 93–117.
[50] G. Wang, T.-T. Wong, and P.-A. Heng, "Deringing cartoons by image analogies," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 4, pp. 1360–1379, 2006.