Structure-Preserving Guided Image Filtering

Hongyan Wang, Zhixun Su(✉), and Songxin Liang

School of Mathematical Sciences, Dalian University of Technology,
Dalian 116024, China
zxsu@dlut.edu.cn

Abstract. Guided filter behaves as a structure-transferring filter which takes advantage of the guidance image. Nevertheless, it is likely to suffer from structure information loss problem and artifacts would be introduced in practical tasks, e.g., detail enhancement. We in this paper propose to deal with the structure loss problem. We modify the original objective function and develop a re-weighted algorithm to proceed the filtering process iteratively. The proposed filter inherits good properties of guided filter and is more capable in avoiding structure information loss. Many vision tasks can be benefited from the proposed filter. Few applications we outline include flash/no-flash image restoration, image dehazing, detail enhancement, HDR compression, and image matting. Experimental comparisons with relative methods for these tasks demonstrate the effectiveness of the proposed filter.

Keywords: Image filter · Guided filter · Bilateral filter

1 Introduction

Image filtering has attracted many research attentions for years and been witnessed significant advances. The goal is to remove fine-scale details or textures and preserve sharp edges. In computer vision and graphics community, it is a simple and fundamental tool to extract meaningful information for understanding and analyzing images.

Most early proposed image filters are linear translation-invariant filters (e.g., Gaussian and Laplacian filters), which can be explicitly expressed by convolution operator between one input image and a specific filter kernel. They usually achieve poor performance due to their simple forms and lacking of elaborate designing. To better preserve edges, bilateral filter (BF) has been proposed in [1,30], taking both spatial and range information into consideration. By using an additional favourable image instead of the input image, bilateral filter can be extended to joint bilateral filter (JBF) [10,25]. However, one well-known limitation of (joint) bilateral filter is that it may generate gradient reversal artifacts [2,11] in detail enhancement and HDR compression.

Z. Cui et al. (Eds.): IScIDE 2019, LNCS 11935, pp. 114–127, 2019.
https://doi.org/10.1007/978-3-030-36189-1_10

Joint filtering techniques need an input image and a guidance image. Based on a local linear model between the guidance image and the output, the guide filter (GF) [14] is a representative structure-transferring filter and overcomes the gradient reversal limitation. Unluckily, one trouble thing is that the guidance image may be insufficient or unreliable locally, which may lead to unpleasing artifacts. Shen et al. propose the concept of mutual-structure in [28] to address the structure inconsistency problem. Relying on mutual-structures that are contained in both the input and the guidance image to generate the output, MSJF [28] is suitable to specific problems like joint structure extraction and joint segmentation. It does not has structure-transferring property.

Many methods [16,19,20] have been proposed to improve guided filter [14]. However, most of them work on designing various adaptive weights or regularization terms and pay little attention on the structure loss problem. Guided filter [14] applies L_2 norm distance on intensity to formulate fidelity term, leading that some meaningful structures may not be preserved well, particularly near edges. This can be illustrated in Fig. 1. As shown in Fig. 1(b), there is noticeable loss of structural information near the edge. This easily causes errors or artifacts in many applications, e.g., detail enhancement.

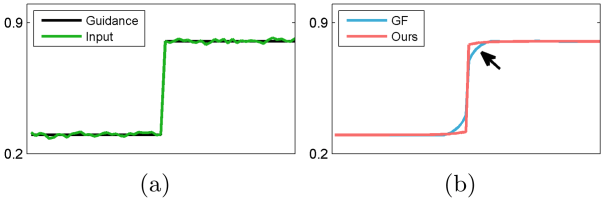


Fig. 1. Illustration of structure loss on 1D signals. The guided filter output losses structural information and blurs the edge. Our method is more capable of preserving the edge and the output edge is sharper.

We in this work propose an algorithm to improve the capability of the guided filter on avoiding structure loss. Our contribution is two-fold. First, we modify the original objective function and develop an efficient algorithm by iterative re-weighting mechanism. Second, we show that the proposed method benefits many vision applications. Experimental results compared with state-of-the-art methods demonstrate the effectiveness of our method.

2 Proposed Model and Optimization

In this section, we propose a new objective function based on the similar assumption as [14]. Then we develop a numerical algorithm and give the iterated solutions of the proposed method.

2.1 Proposed Model

Given an guidance image G, the proposed method is based on the following local model:

$$q_i = a_k G_i + b_k, \forall i \in W_k, \tag{1}$$

where q denote the expected output, and i is the pixel index in the window W_k, which is centered at pixel k. a_k and b_k are two linear transform coefficients. All pixels in W_k are assumed to share the same a_k and b_k. W_k is set to be square with $2r+1$ pixels on the side.

For an input image p, the output q is expected to contain major structures associated with p. Details, textures, and noise are expected to be contained in $n = p - q$. Based on these assumptions, we propose the following objective function:

$$E(a_k, b_k) = \sum_{i \in W_k} \left(|a_k G_i + b_k - p_i| + \epsilon a_k^2 \right). \tag{2}$$

To deal with the structure loss, L_1 norm is employed in (2) to formulate the fidelity term.

2.2 Optimization

The data term in (2) is not quadratic, making the optimization problem not a simple linear regression problem. To solve (2), we employ iterative re-weighted least squares (IRLS) algorithm [17] to obtain the iterative solutions of a_k and b_k. IRLS solves a sequence of least square problems within an iterating framework, and every least square problem can be penalized by the reciprocal of absolute error of previous iteration. The cost function at t-th iteration ($t \geq 1, t \in N$) is defined as

$$E(a_k^t, b_k^t) = \sum_{i \in W_k} \omega_i^t (a_k^t G_i + b_k^t - p_i)^2 + \epsilon (a_k^t)^2, \tag{3}$$

where the weight at pixel i is given by $\omega_i^t = 1/\max\left\{|q_i^{t-1} - p_i|,\ \nu\right\}$. ν is a parameter to avoid the zero denominator. We define the q^0 as the output of the guided filter [14].

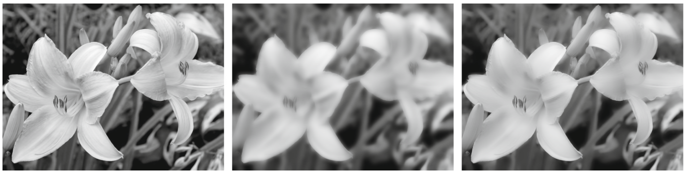
The energy function (3) is a linear regression problem [9]. By setting the derivatives of (3) with respect to a_k^t and b_k^t to zero respectively, we can obtain the iterative solutions of a_k^t and b_k^t:

$$\begin{cases} a_k^t = \dfrac{\frac{1}{|W|}\sum_{i \in W_k} \omega_i^t G_i p_i - \widetilde{G}_k^t \widetilde{p}_k^t \widetilde{\omega}_k^t}{(\widetilde{\sigma}_k^t)^2 + \epsilon}, \\ b_k^t = \widetilde{p}_k^t - a_k^t \widetilde{G}_k^t, \end{cases} \tag{4}$$

where $\widetilde{G}_k^t$ and $(\widetilde{\sigma}_k^t)^2$ denote the weighted mean and weighted variance of G in W_k, given by $\widetilde{G}_k^t = \frac{\sum_{i \in W_k} \omega_i^t G_i}{\sum_{i \in W_k} \omega_i^t}$ and $(\widetilde{\sigma}_k^t)^2 = \frac{1}{|W|}\sum_{i \in W_k} \omega_i^t (G_i - \widetilde{G}_k^t)^2$. $\widetilde{p}_k^t$ denotes the weighted mean of p and $\widetilde{\omega}_k^t$ denotes the mean of all the penalized weights ω_i^t in W_k, given by $\widetilde{\omega}_k^t = \frac{1}{|W|}\sum_{i \in W_k} \omega_i^t$ and $\widetilde{p}_k^t = \frac{\sum_{i \in W_k} \omega_i^t p_i}{\sum_{i \in W_k} \omega_i^t}$.

Similar to [14], overlapping problem appears. In each iteration, we compute $\overline{a}_i^t$ and $\overline{b}_i^t$ by averaging strategy: $\overline{a}_i^t = \frac{1}{|W|}\sum_{k:i \in W_k} a_k^t$ and $\overline{b}_i^t = \frac{1}{|W|}\sum_{k:i \in W_k} b_k^t$. The final output is calculated by $q_i^t = \overline{a}_i^t G_i + \overline{b}_i^t$.

We point out that the calculations of (4) involve $\widetilde{G}_k^t$, $\widetilde{p}_k^t$, and $\widetilde{\sigma}_k^t$, which are associated with the penalized weights $\widetilde{\omega}_i^t$. This is different from the calculations of the solutions (a_k and b_k) of guided filter [14], since the filtering process of [14] is not iterative.



(a) Input/Guidance (b) GF (q^0) (c) Ours

Fig. 2. Comparisons of guided filter and the proposed filter.

3 Discussions

In this section, we discuss some properties of the proposed iterative filter and provide the expressions of our iterative kernel weights. Extension to color images and limitations are also discussed.

3.1 Edge-Preserving Filter

In this section we analyze how does the proposed filter work. When $G = p$, the equations in (4) are simplified to $a_k^t = (\widetilde{\sigma}_k^t)^2 / \left((\widetilde{\sigma}_k^t)^2 + \epsilon\right)$ and $b_k^t = (1 - a_k^t)\,\widetilde{G}_k^t$. As ϵ is positive, for each q_i^t there are two special cases:

- If $(\widetilde{\sigma}_k^t)^2 \ll \epsilon$, then $a_k^t \approx 0$, so $q_i^t \approx b_k^t \approx \widetilde{G}_k^t$.
- If $(\widetilde{\sigma}_k^t)^2 \gg \epsilon$, then $a_k^t \approx 1$ and $b_k^t \approx 0$, so $q_i^t \approx G_i$.

For pixels located in a flat window, their intensities are approximative and we have $\widetilde{G}_k^t \approx G_i$. Then $(\widetilde{\sigma}_k^t)^2 \to 0$ and $(\widetilde{\sigma}_k^t)^2 \ll \epsilon$, we obtain $a_k^t \approx 0$ and $q_i^t \approx \widetilde{G}_k^t$. In other words, the proposed filter handles pixel in a flat window by weighted averaging to reach the goal of smoothing. On the other hand, only when $(\widetilde{\sigma}_k^t)^2 \gg \epsilon$, pixel centered at this window is preserved. Note that $(\widetilde{\sigma}_k^t)^2$ is influenced by both ω_i^t and structures in the patch W_k. This means that whether the pixel is preserved or not is determined by G, p and q^{t-1}. That is, the criterion "what is an edge" or "structures which are expected to be preserved" is no longer simply measured by the given parameter ϵ like [14]. In the t-th iteration, pixels where p and q^{t-1} are approximate are assigned large weights to reach the similar smoothing effects as guided filter, whereas pixels where p and q^{t-1} are quite different are assigned small weights for proper modifications. This explains why the proposed method is more capable of avoiding structural information loss than guided filter. Visual comparison of guided filter and the proposed filter is shown in Fig. 2.

3.2 Gradient-Preserving Filter

The proposed filter is able to avoid the gradient reversal artifacts. We take detail enhancement for example and follow the algorithm based on base-detail layers decomposition $E = B + \tau D$, where B, D, E denote the base layer, the detail layer and the enhanced image, respectively. τ is a parameter to control the magnification of details.

In practice, the base layer B is generated by filtering on the input image p, and the detail layer D can be viewed as $D = p - B$. This relationship ensures that $\partial D = \partial p - \partial B$. If B can not be consistent with the input signal p and further leads to $\partial D \cdot \partial p < 0$, the gradient reversal artifacts would appear in the enhanced signal after magnifying the detail layer D.

Theoretically, the local linear model (1) indicates that ∂B is a_k^t times of ∂p when $p \equiv G$. For a_k^t, we have $a_k^t = \widetilde{\sigma}_k^2 / (\widetilde{\sigma}_k^2 + \epsilon)$ and it is less than 1. Then we further have $\partial D = \partial p - \partial B = (1 - a_k^t)\partial p$ and $\partial D \cdot \partial p \geq 0$.

An example of 1D signal is shown in Fig. 3. As can be seen in Fig. 3(c), our final enhanced signal avoids gradient reversal artifacts safely and does not introduce over-sharpened artifacts in the enhanced signal.

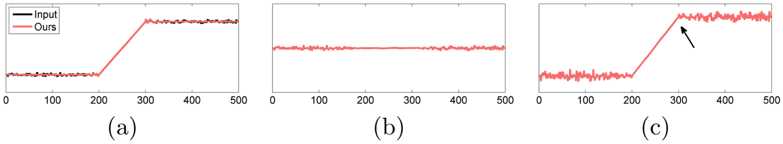


Fig. 3. The proposed filter is gradient-preserving. (a) Input signal and our filtering result. (b) Detail layer. (c) The enhanced signal. The proposed method does not generate unrealistic details near edges and further produces natural enhanced signal without over-sharpened artifacts.

3.3 Iterative Filter Kernel

The filter kernel of the proposed method varies in each iteration. The explicit expressions of kernel weights in t-th iteration can be given by

$$\boldsymbol{W}_{ij}^t = \frac{1}{|W|^2} \sum_{k:(i,j)\in W_k} \boldsymbol{M} \left(\frac{(G_i - \widetilde{G}_k^t)(G_j - \widetilde{G}_k^t)}{(\widetilde{\sigma}_k^t)^2 + \epsilon} + \frac{1}{\widetilde{\omega}_k^t} \right), \quad (5)$$

where $\boldsymbol{M} = \omega_j^t + \boldsymbol{T}_j \boldsymbol{H}\left(|q_j^{t-1} - p_j| - \nu\right)\left(p_j - \widetilde{p}_k^t - a_k^t(G_j - \widetilde{G}_k^t)\right)$, and $\boldsymbol{T}_j = \text{sgn}(q_j^{t-1} - p_j)/(q_j^{t-1} - p_j)^2$. We use $\text{sgn}(\cdot)$ to denote sign function and $\boldsymbol{H}(\cdot)$ to denote heaviside step function (outputting ones for positive values and zeros otherwise).

(5) can be proved by the Chain Rule and a series of careful algebraic manipulations. Here we visually show several kernels in Fig. 4.

3.4 Filtering Using Color Guidance Image

Section 2 presents the iterated filtering process for the case of a gray input with a gray guidance image. However, RGB images usually contain more information than gray images. Thus, we develop another proper algorithm for the case of color guidance image. We rewrite model (1) in vector form:

$$q_i = \boldsymbol{a}_k^T \boldsymbol{G}_i + b_k, \forall i \in W_k, \tag{6}$$

where $(\cdot)^T$ denotes matrix transposing operator, $\boldsymbol{G}_i$ denotes RGB intensities at pixel i, and $\boldsymbol{a}_k$ denotes coefficient vector. Note that $\boldsymbol{G}_i$ and $\boldsymbol{a}_k$ are 3×1 vectors, while q_i and b_k are still scalars. Then (2) becomes

$$E(\boldsymbol{a}_k^t, b_k^t) = \sum_{i \in W_k} \omega_i^t(\boldsymbol{a}_k^t \boldsymbol{G}_i + b_k^t - p_i)^2 + \epsilon(\boldsymbol{a}_k^t)^T \boldsymbol{a}_k^t, \tag{7}$$

and the solutions of $\boldsymbol{a}_k^t$ and b_k^t can be obtained by linear regression:

$$\begin{cases} \boldsymbol{a}_k^t = \left(\widetilde{\boldsymbol{\Sigma}}_k^t + \epsilon \boldsymbol{E}\right)^{-1} \left[\frac{1}{|W|} \sum \omega_i^t p_i \boldsymbol{G}_i - \widetilde{\omega}_k^t \widetilde{p}_k^t \widetilde{\boldsymbol{G}}_k^t\right] \\ b_k^t = \widetilde{p}_k^t - (\boldsymbol{a}_k^t)^T \widetilde{\boldsymbol{G}}_k^t \end{cases}, \tag{8}$$

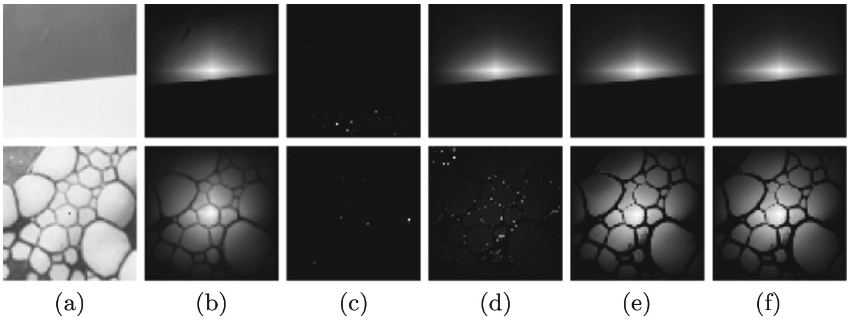
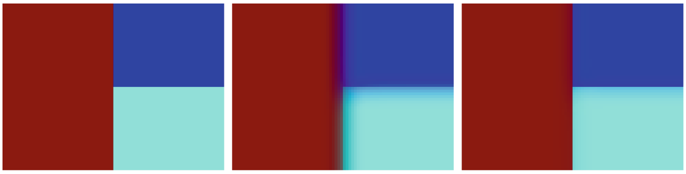


Fig. 4. Several iterative filtering kernels for two special cases. (a) Image patches. (b) Kernels of guided filter ($\boldsymbol{W}^{\boldsymbol{0}}$) at the pixels denoted by the red dots in (a). (c)–(f) Our iterative kernels after 1st, 4th, 7th and 10th iteration. The final kernels are capable in preserving underlying data structure.

where $\boldsymbol{E}$ is a 3×3 matrix with all one elements, and $\widetilde{\boldsymbol{G}}_k^t$ is a 3×1 weighted averaged vector of $\boldsymbol{G}_i$. $\widetilde{\boldsymbol{\Sigma}}_k^t$ is a 3×3 weighted covariance matrix, expressed as $\widetilde{\boldsymbol{\Sigma}}_k^t = \frac{1}{|W|}\sum_{i\in W_k}\omega_i\left(\boldsymbol{G}_i - \widetilde{\boldsymbol{G}}_k^t\right)\left(\boldsymbol{G}_i - \widetilde{\boldsymbol{G}}_k^t\right)^T$.

After dealing with the overlapping problem, the final filter output are given by $q_i^t = \frac{1}{|W|}\sum_{k:i\in W_k}\left(\left(\boldsymbol{a}_k^t\right)^T\boldsymbol{G}_i + b_k^t\right) = \left(\overline{\boldsymbol{a}}_i^t\right)^T\boldsymbol{G}_i + \overline{b}_i^t$.

We show an example in Fig. 5. By comparing the results of using gray guidance and RGB guidance visually, we can see that edges in Fig. 5(c) are preserved better than that in Fig. 5(b).



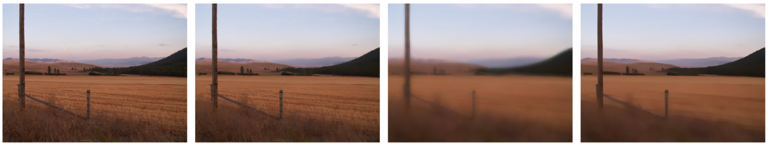
(a) Input image (b) Gray guidance (c) RGB guidance

Fig. 5. Comparisons on color guidance image and gray guidance image.

In addition, filtering a gray input with a color guidance image is also very useful for some vision tasks, such as dehazing and image matting. These applications can be found in Sect. 4.

3.5 Limitations

The proposed method would not work well if there is complex texture patterns contained in the image, or it is incapable of removing dense textures. We show an example in Fig. 6. As can be seen, gradient-based RTV method [33] performs better than the proposed method.



Input $r = 8, \epsilon = 0.2^2$ $r = 16, \epsilon = 0.6^2$ RTV [33]

Fig. 6. One failure example.

4 Applications and Experimental Results

The proposed method can be applied to a variety of computer vision tasks. Several tasks we outline in this section are flash/no-flash image restoration, image dehazing, detail enhancement, HDR compression, and image matting.

Parameter Settings. We state parameter settings first. Empirically, we set window radius $r < 100$ and the regularization parameter $\epsilon < 1$. ν is set to be 0.0001 in all experiments. The number of iterations is set to be 5.

Running Time. The proposed algorithm has been implemented in MATLAB on a PC with Intel Xeon E5630 CPU and 12 GB RAM. It takes about 0.4 s to process a 321 × 481 gray image without code optimization. For the color case, processing an image with the same size takes about 23.8 s.

4.1 Flash/No-Flash Image Restoration

Flash/No-Flash Denoising. The observed flash image can be viewed as a guidance image to facilitate denoising a noisy no-flash input. Figure 7 shows an example. The compared methods include joint BF, GF [14], and WLS [11]. As can be seen, the results shown in Fig. 7(c) (joint BF) contain noticeable gradient reversal artifacts. GF is incapable of preserving edges and can not produce sharp edges in some regions (Fig. 7(d)). The results of WLS (Fig. 7(e)) contain artifacts generated by the intensive noises. Our result is visually better than others.

Flash/No-Flash Deblurring. One common way for flash/no-flash deblurring is to generate an image which both preserves the ambiance lighting and contains clear edges and details. This process can be simply finished by the base-detail layers decomposition model mentioned in Sect. 3.2, which may save much computation compared with existing deblurring methods. The base layer B is generated by filtering on the no-flash image p guided by the flash image G in order to maintain the ambient lighting. The detail layer D is produced by $D = G - \widehat{G}$, where $\widehat{G}$ denotes a self-guidance filtered output of G. Then we can combine the base layer B and the detail layer D to generate a blur-free image.

A challenging case is that some saturated regions may appear in the blurry no-flash image, as shown in Fig. 8(a). Two representative blind image deblurring methods [22,24], fail to produce clear structure around the saturated region, as shown in Fig. 8(b)–(c). Severe artifacts can be found in these results. Even for methods [15,23], which are specifically proposed to deal with outliers, their deblurred results shown in Fig. 8(d)–(e) are still unpleasing. We also provide results of several filtering methods, including joint BF, GF [14], WLS [11], L_0 gradient minimization [32], domain transform filter (DTF) [12], RTV [33], RGF [34], and MSJF [28]. As shown in Fig. 8(g)–(o), ours is visually the best.

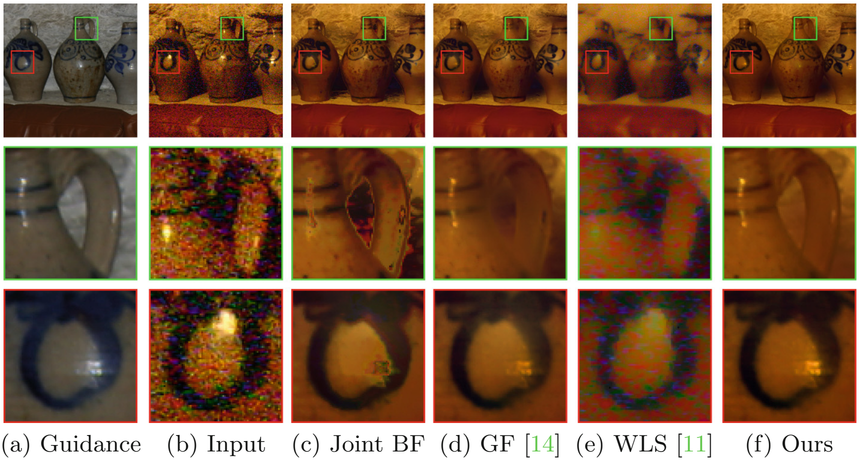


Fig. 7. Denoising with flash/no-flash image pair. The no-flash image (input) suffers from severe noises while the structures and edges in the flash image (guidance) are quite clear. Compared with the results shown in (c)–(e), the proposed filtering method produce a noise-free result with clear edges.

4.2 Image Dehazing

We follow the widely used the hazy image formation model $\boldsymbol{I}(x) = \boldsymbol{J}(x)T(x) + \boldsymbol{A}(x)(1 - T(x))$, where $\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{A}, T$ denote the observed hazy image, the scene radiance, the global atmospheric light and the medium transmission, respectively. x

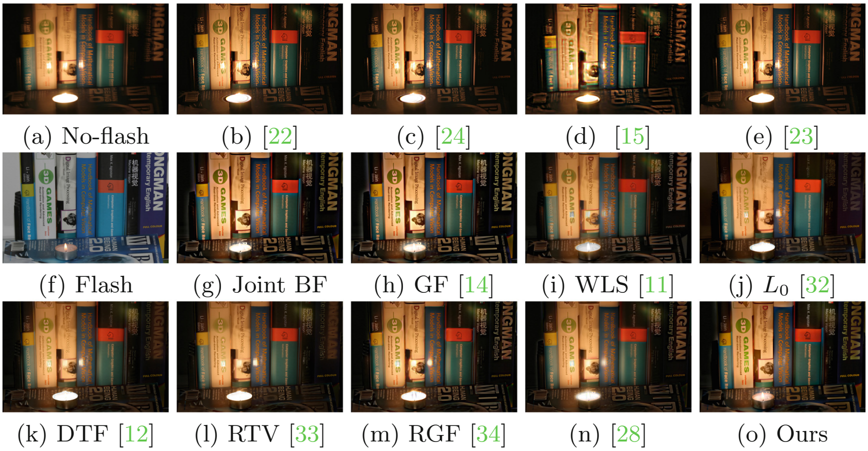


Fig. 8. Blur removal example with flash/no-flash image pair. Results shown in (b)–(e) are restored by deblurring methods [15,22–24], respectively (blur kernel size: 33 × 33; running time of kernel estimation process: 465.7 s, 2020.6 s, 306.8 s, and 1933.4 s). Results shown in (g)–(o) require no kernel estimation process (running time: 1.6 s, 1.1 s, 7.4 s, 10.2 s, 78.4 s, 17.5 s, 6.4 s, 23.8 s and 7.8 s).

is the pixel index. We estimate the atmospheric light $\boldsymbol{A}$ and the raw transmission map $T^0(x)$ within the framework [13] and refine $T^0(x)$ by filtering $T^0(x)$ instead of solving the matting Laplacian matrix like [13], which is very slow. As can be seen in Fig. 9, our refined transmission map T (Fig. 9(d)) contains more meaningful structures than T^0 (Fig. 9(b)). The final dehazed result is shown in Fig. 9(e). The competitive dehazing methods include DCP [13], BCCR [21], NLD [3], DehazeNet [5], and MSCNN [26]. Our result is visually comparable to the results of conventional methods DCP, BCCR and NLD, and is a little better than that of learning-based methods DehazeNet and MSCNN.

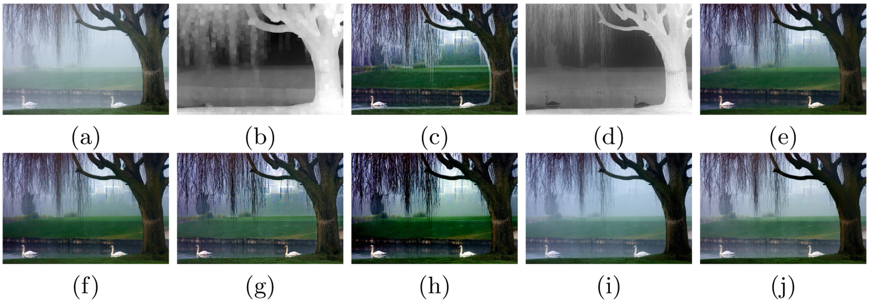
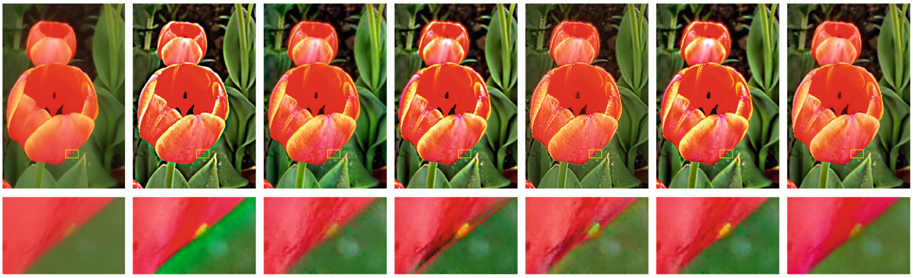


Fig. 9. An image dehazing example. (a) Input hazy image. (b) Raw transmission map T^0. (c) Dehazed by (b). (d) Our refined T. (e) Dehazed by (d). (f) DCP [13]. (g) BCCR [21]. (h) NLD [3]. (i) DehazeNet [5]. (j) MSCNN [26]. It cost about 57 s and 40 s for (f) (using matting Laplacian) and (e) (ours) to refine T^0.

4.3 Detail Enhancement and HDR Compression

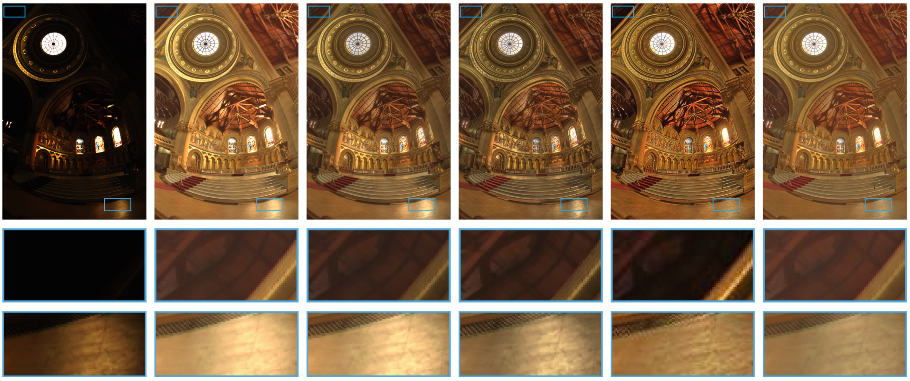
Detail Enhancement. The detail enhancement algorithm has been described in Sect. 3.2. Figure 10 shows comparisons of using GF [14], LLF [29], RTV [33], mRTV [8], RoG [4] and the proposed filter on an example. The results shown in Fig. 10(b)–(f) suffer from unrealistic artifacts (See close-ups). In comparison, our method avoids to generate unrealistic artificial details (Fig. 10(g)).

HDR Compression. Different from detail enhancement, HDR compression aims to generate low dynamic range image by compressing the base layer at some rate while preserving the details. We show an example in Fig. 11 compared with some filter methods [4,11,29,32]. To display we convert the input HDR radiance to a logarithmic scale and then map the result to $[0, 1]$ (See Fig. 11(a)). For each result, we show two close-ups of a highlighted area and a dark area. The proposed method produces clean result with natural details, whereas the other methods either suffer from aliasing or fail to compress the range properly.



(a) Input (b) GF (c) LLF (d) RTV (e) mRTV (f) RoG (g) Ours

Fig. 10. A detail enhancement example compared with [4,8,14,29,33]. The guided filter generates an over-enhanced results with unrealistic artifacts due to structural information loss, as shown in (b). Our result is visually more natural with little unrealistic artifacts than (b)–(f).

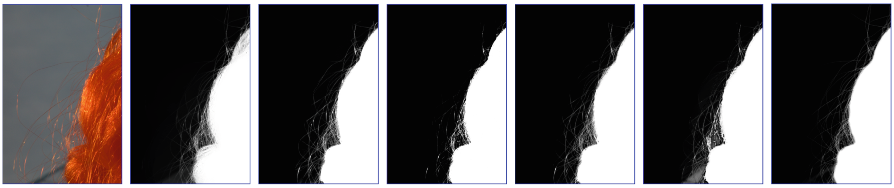


Input WLS [11] LLF [29] L_0 [32] RoG [4] Ours

Fig. 11. HDR compression. Close-ups show that the proposed method compresses the highlighted area and the dark area effectively.

4.4 Image Matting

An accurate matte can be generated from filtering a coarse binary mask with the guidance of corresponding clear image. We compare our method with image matting methods [6,7,18,27,31]. Our result shown in Fig. 12(g) is visually comparable with the results shown in Fig. 12(b)–(f). Nevertheless, the proposed method does not require another user-assisted input but a coarse binary mask. In comparison, all the competitive methods require user-assisted input (either scribbles or trimap image) for labeling.



(a) Input (b) [18] (c) [31] (d) [6] (e) [27] (f) [7] (g) Ours

Fig. 12. An image matting example.

5 Conclusion

In this paper, we propose to modify the measurement function of fidelity term in the objective function of guided filter to address its structural information loss limitation and improve its capability on preserving structures. We then develop an efficient iterative re-weighting algorithm to solve the proposed model. We analyze the attractive properties of our method. The extension to color guidance image (with gray input) leads the proposed filter to benefit some specific tasks, e.g., image matting. We also outline other applications which can be benefited from the proposed method. We expect to apply it to more practical applications.

Acknowledgements. This work has been partially supported by National Natural Science Foundation of China (No. 61572099).

References

1. Aurich, V., Weule, J.: Non-linear Gaussian filters performing edge preserving diffusion. In: Sagerer, G., Posch, S., Kummert, F. (eds.) Mustererkennung, 17. DAGM-Symposium, pp. 538–545. Springer, Heidelberg (1995). https://doi.org/10.1007/978-3-642-79980-8_63
2. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. ACM ToG **25**(3), 637–645 (2006)
3. Berman, D., Treibitz, T., Avidan, S.: Non-local image dehazing. In: CVPR, pp. 1674–1682 (2016)
4. Cai, B., Xing, X., Xu, X.: Edge/structure preserving smoothing via relativity-of-Gaussian. In: ICIP, pp. 250–254 (2017)
5. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: an end-to-end system for single image haze removal. IEEE TIP **25**(11), 5187–5198 (2016)
6. Chen, Q., Li, D., Tang, C.K.: KNN matting. IEEE TPAMI **35**(9), 2175–2188 (2013)
7. Cho, D., Tai, Y.-W., Kweon, I.: Natural image matting using deep convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 626–643. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_39
8. Cho, H., Lee, H., Kang, H., Lee, S.: Bilateral texture filtering. ACM ToG **33**(4), 128:1–128:8 (2014)
9. Draper, N.R., Smith, H.: Applied Regression Analysis. Wiley series in Probability and Mathematical Statistics, 2nd edn. Wiley, New York (1981)

10. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. ACM ToG **23**(3), 673–678 (2004)
11. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM ToG **27**(3), 67:1–67:10 (2008)
12. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM ToG **30**(4), 1–12 (2011)
13. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE TPAMI **33**(12), 2341–2353 (2011)
14. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE TPAMI **35**(6), 1397–1409 (2013)
15. Hu, Z., Cho, S., Wang, J., Yang, M.: Deblurring low-light images with light streaks. In: CVPR, pp. 3382–3389 (2014)
16. Kou, F., Chen, W., Wen, C., Li, Z.: Gradient domain guided image filtering. IEEE TIP **24**(11), 4528–4539 (2015)
17. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM ToG **26**(3), 70 (2007)
18. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE TPAMI **30**(2), 228–242 (2008)
19. Li, Z., Zheng, J., Zhu, Z., Yao, W., Wu, S.: Weighted guided image filtering. IEEE TIP **24**(1), 120–129 (2015)
20. Liu, W., Chen, X., Shen, C., Yu, J., Wu, Q., Yang, J.: Robust guided image filtering. Computing Research Repository abs/1703.09379 (2017). http://arxiv.org/abs/1703.09379
21. Meng, G., Wang, Y., Duan, J., Xiang, S., Pan, C.: Efficient image dehazing with boundary constraint and contextual regularization. In: ICCV, pp. 617–624 (2013)
22. Pan, J., Hu, Z., Su, Z., Yang, M.: Deblurring text images via L0-regularized intensity and gradient prior. In: CVPR, pp. 2901–2908 (2014)
23. Pan, J., Lin, Z., Su, Z., Yang, M.: Robust kernel estimation with outliers handling for image deblurring. In: CVPR, pp. 2800–2808 (2016)
24. Pan, J., Sun, D., Pfister, H.: Blind image deblurring using dark channel prior. In: CVPR, pp. 1628–1636 (2016)
25. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M.F., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM ToG **23**(3), 664–672 (2004)
26. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.-H.: Single image dehazing via multi-scale convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 154–169. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_10
27. Shahrian, E., Rajan, D., Price, B.L., Cohen, S.: Improving image matting using comprehensive sampling sets. In: CVPR, pp. 636–643 (2013)
28. Shen, X., Zhou, C., Xu, L., Jia, J.: Mutual-structure for joint filtering. IJCV **125**(1–3), 19–33 (2017)
29. Sylvain, P., Samuel, W.H., Jan, K.: Local laplacian filters: edge-aware image processing with a laplacian pyramid. Commun. ACM **58**(3), 81–91 (2015)
30. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, pp. 839–846 (1998)
31. Varnousfaderani, E.S., Rajan, D.: Weighted color and texture sample selection for image matting. IEEE TIP **22**(11), 4260–4270 (2013)
32. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L_0 gradient minimization. ACM ToG **30**(6), 174:1–174:12 (2011)

33. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM ToG **31**(6), 139:1–139:10 (2012)
34. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 815–830. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_53