Rolling Guidance Filter

Qi Zhang[†] Xiaoyong Shen[†] Li Xu[‡] Jiaya Jia[†]

[†] The Chinese University of Hong Kong
[‡] Image & Visual Computing Lab, Lenovo R&T

http://www.cse.cuhk.edu.hk/leojia/projects/rollguidance

Abstract. Images contain many levels of important structures and edges. Compared to masses of research to make filters edge preserving, finding scale-aware local operations was seldom addressed in a practical way, albeit similarly vital in image processing and computer vision. We propose a new framework to filter images with the complete control of detail smoothing under a scale measure. It is based on a rolling guidance implemented in an iterative manner that converges quickly. Our method is simple in implementation, easy to understand, fully extensible to accommodate various data operations, and fast to produce results. Our implementation achieves realtime performance and produces artifact-free results in separating different scale structures. This filter also introduces several inspiring properties different from previous edge-preserving ones.

Keywords: image filter, scale-aware processing, edge preserving

1 Introduction

To smooth images while preserving different levels of structures, filter techniques were broadly studied. They are popular in visual processing and are sometimes must-perform operations to remove detrimental or unwanted content. Among all filters, edge-aware ones form a major stream. They include representatives of anisotropic diffusion [20], bilateral filter (BF) [26], guided filter (GF) [13], geodesic filters [7, 11], weighted median filters [18, 34], to name a few. These filters and their variations pursue similar goals to preserve high-contrast edges and remove low-contrast or graduate changes.

Strong edges, measured as large discrepancy between local pixel values, contain important image information. This explains the popularity of these approaches. Meanwhile, we note other than magnitude, another measure that is similarly essential in solving many computer vision problems, including invariant feature construction, data compression, object classification, segmentations, and motion analysis, is about the scale of images/objects/regions.

Natural scenes are composed of objects in different sizes and contain structures of various scales, which deliver diverse information to human. Small structures, usually referred to as details, represent content classified as texture, small objects, and noise, while large scale information generally encodes boundaries,

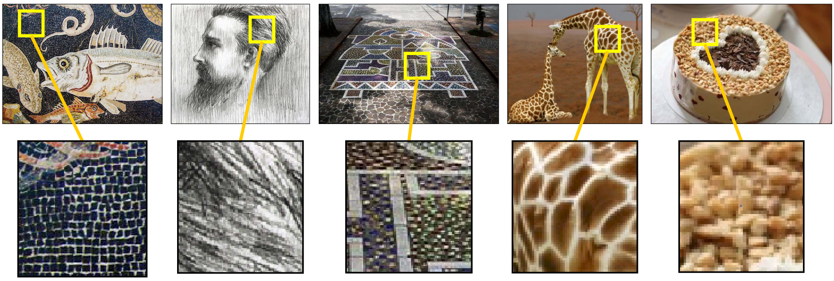


Fig. 1. Examples of high-contrast details in natural images. As explained, edge-aware filters aim to maintain them due to the large magnitude of edges.

slow spatial color transition, and flat regions. The latter tells more about how objects are arranged in the scene. Separation of structures with respect to their scales is an important process that has been discussed in many areas, such as segmentation [3], object detection [10], and saliency detection [31].

To date, incorporating scales in filter design remains a hard problem. Edge-aware methods can hardly separate structure from details because edge strength and object scale are completely different concepts. Several examples are shown in Fig. 1 where magnitudes of gradients in detail/texture regions are high enough to let edge-aware filters preserve them by nature.

Aiming to handle texture, Subr et al. [25] suppressed high-contrast oscillation by averaging local extrema manifolds and Xu et al. [30] used a global optimization method to remove texture. Karacan et al. [14] computed patch-based weighted average using region covariance with the similar objective to remove texture. These methods involve relatively heavy computation due to the need to employ patch-based filtering or solve large linear systems. They also cannot be easily changed to the filter form for realtime and spatially varying operations.

The iterated nonlocal means proposed by Brox and Cremers [5] for noise removal is also related to our approach. But there exist a few essential differences on both objectives and frameworks (detailed later in Section 2). We note the major challenge to propose a scale-aware filter is twofold.

1. it is not clear yet what is the optimal way to define scale in images regarding local pixel information for filter design because structures may not be with obvious boundaries and they are generally irregularly shaped.
2. Spatial color variation in different scales is unavoidably mixed and overlapped, making their separation very difficult.

We address these problems by proposing an effective scale-aware filter that can remove different levels of details in any input natural images. This algorithm is amazingly easy to implement, efficient, and low-cost in computation. Online processing can be achieved even on a single CPU core. Existing techniques for accelerating edge-aware filters can also be adopted to speed up ours. Our major technical contribution is as follows.

1. We introduce a scale measure following scale space theory. This definition brings the breakthrough to control the level of details during filtering.
2. We propose a new rolling guidance method to automatically refine edges that can be preserved in order to preserve large-scale structures optimally.

In addition, our filtering framework is general and can be extended or modified for different special applications. We show our experimental results and applications in Section 5. More are provided in our project website.

2 Related Work

We review popular and related image filters proposed in recent years and discuss the important difference.

Edge-Aware Filter Edge-aware filters are developed in different strategies. But the pursuit is to similarly preserve only high-contrast edges. Technically, average- and optimization-based methods are most widely employed.

Average-based filters include anisotropic diffusion [20], bilateral filter [26, 8, 19, 6, 32, 33], guided filter [13], and geodesic filters [7, 11]. This category basically defines different types of affinity between neighboring pixel pairs by considering intensity/color difference. It smoothes images through weighted average. Large and small affinities are defined for low- and high-contrast regions respectively. A variant is *joint/cross* filtering [21, 16, 13, 11]. They are to smooth an image using guidance when defining the pixel affinity.

Optimization-based methods include total variation (TV) [23], weighted least squares (WLS) [9], and L_0 gradient minimization [29]. These methods restore images by optimizing global functions containing terms defined in L_1 norm, weighted L_2 norm or L_0 norm. All these approaches are not easily accelerated in the form of filter due to the need to solve large linear systems.

Mode/Median Filters Mode and median filters can remove high-contrast details. This class mainly includes mode filter [27, 15], median filter [28, 15], and weighted median filter [18, 34]. These filters compute mode or (weighted) median rather than average in local patches, which inevitably result in higher computational costs. They can perfectly remove salt&pepper noise. But for fast-oscillating signals, local mode and median still produce oscillating results. Thus they cannot optimally remove details in images.

Iterated Nonlocal Means The iterated nonlocal means (INM) proposed by Brox and Cremers [5] is the most related work to ours. There are several major differences nevertheless. First, we aim at scale-aware smoothing while the method of [5] is for texture preserving during noise removal. Second, algorithmically we use iterations to recover blurred edges, while INM adopts fixed point iterations for optimizing a global function. Third, initialization and weight definition are set differently in these two methods.

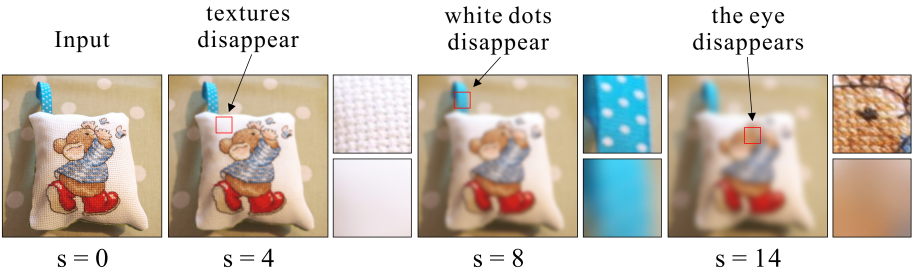


Fig. 2. Illustration of scales. As the Gaussian kernel gets larger, more and more structures disappear.

Texture Smoothing Texture is one type of small-scale oscillations locally. The specialty is on its repeated identical or similar patterns. Subr et al. [25] smoothed texture by averaging two manifolds generated from local minimum and maximum. Xu et al. [30] optimized a global function with relative total variation (RTV) regularization. RTV protects structural edges. This method needs to solve a linear system.

Karacan et al. [14] adopted a weighted-average strategy with the covariance of patch features. It leverages the repetition property of texture and is also time-consuming for pixel affinity computation. Recently, Bao et al. [4] combined bilateral weight with a tree weight defined on a minimum spanning tree. Su et al. [24] combined low-pass filter, edge-preserving filter and L_0 edge correction to achieve the similar goal.

These texture smoothing methods basically make use of the texture repetition property. They are different by nature from the *scale-aware filters* defined in this paper. Our goal is to separate out details, even without repetitive patterns. Only the scale metric is used in our method.

3 Problem Definition and Analysis

We first define the structure scale as *the smallest Gaussian standard deviation* σ_s such that when this σ_s deviation Gaussian is applied to an image, corresponding structure disappears. We denote the convolution process with the input image I and Gaussian $g_v(x, y)$ of variance $v = \sigma_s^2$ as

$$L_v = g_v * I, \tag{1}$$

where $g_v(x, y) = \frac{1}{\sqrt{2\pi v}} \exp(-\frac{x^2+y^2}{2v})$ and $*$ denotes convolution. L_v is the result at scale v. In scale-space theory [17], v is referred to as the *scale parameter*. When the image structure scale is smaller than $\sqrt{v}$ (i.e., σ_s), it will be completely removed in L_v, as claimed in [17]. An illustration is given in Fig. 2. When applying Gaussians with varying σ_s to the image, structures are suppressed differently according to their sizes.

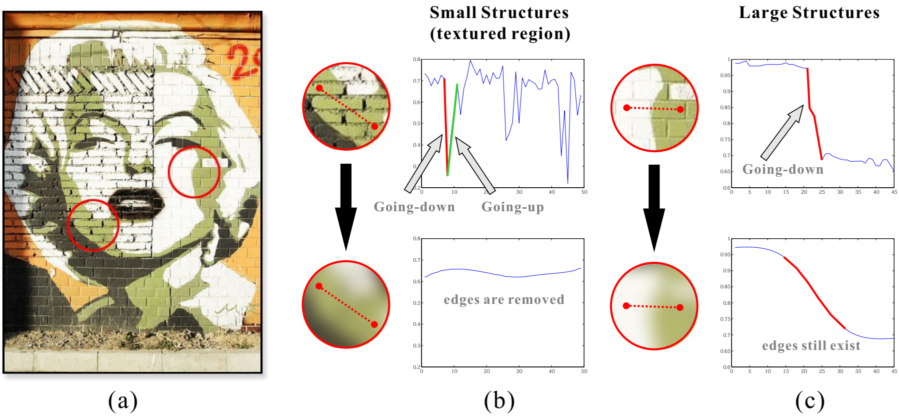


Fig. 3. Comparison of small- and large-structure results after Gaussian filtering. (a) Input image. (b)-(c) 1D signals of pixel values in two lines. The upper signals are the input and the lower ones are results of Gaussian filtering.

Note this definition of scales may not correspond to the actual size or radius of a pattern because the latter is hard to measure given the complexity of image structures. But it tells the relative information. If a structure gets larger, its scale according to our definition must increase, and vice versa.

Now that Gaussian filter sizes determine structure scales, is it possible to directly use Gaussian for scale-aware filtering? The answer is obviously negative because Gaussian filter blurs all edges no matter they are strong or weak. In what follows, we propose a new framework based on this scale definition to retain a level of structures according to their original appearance.

3.1 Our Observation

Gaussian filtering produces blurred images. We analyze result difference on small and large structures. On the one hand, edges of structures with scales below the smoothing scale are completely removed according to the Gaussian average mechanism. On the other hand, large-scale structures are blurred instead of eliminated. To understand the effects, we use the example in Fig. 3. The close-ups in (b) are the 1D signals in a textured region, which originally contain many fast changes with edges going upwards and downwards. When applying Gaussian filter, average of the edges removes nearly all texture patterns. For the large-scale intensity variation shown in Fig. 3(c), Gaussian filter only blurs it and the edge can still be found.

This observation corresponds to an intriguing fact – that is, when edges of small structures are removed by Gaussian filtering, visually there is no clue to manifest they ever exist in the input by only inspecting the result. Contrarily, by watching blurred large-scale structures that are filtered in the same way, one can still be aware of their existence without referring to the input image.

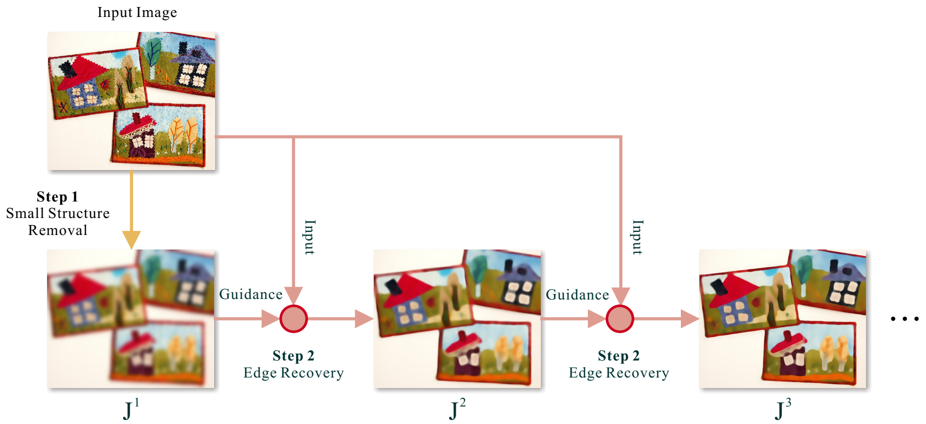


Fig. 4. Flow chart of our method. It contains two steps respectively for small structure removal (Section 4.1) and edge recovery (Section 4.2). Step 2 is an iterative process. The final result is obtained in 3–5 iterations.

We note this finding is the key to developing the rolling-guidance filter scheme. It enlists the power of distinguishing between structures in terms of scales without needing to know the exact form (or model) of texture, details, or noise.

4 Our Method

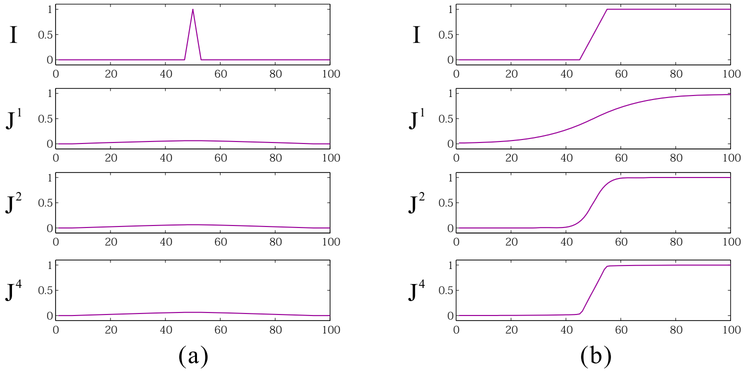
Our method is composed of two main steps, i.e., small structure removal and edge recovery. We explain them in what follows along with a general and simple modification of our rolling guidance framework in Section 4.3. We also discuss the difference between our work and similar techniques in Section 4.4 and show its variants in Section 4.5. Fig. 4 illustrates the work flow.

4.1 Small Structure Removal

The first step is to remove small structures. As aforementioned, Gaussian filter is related to our structure scale determination. We express this operator in a weighted average form, which takes the input image I and outputs image G. Letting p and q index pixel coordinates in the image and σ_s denote the standard deviation, we write the filter as

$$G(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\left(- \frac{||p-q||^2}{2\sigma_s^2} \right) I(q), \tag{2}$$

where $K_p = \sum_{q \in N(p)} \exp(-\frac{||p-q||^2}{2\sigma_s^2})$ is for normalization and $N(p)$ is the set of neighboring pixels of p. This filter completely removes structures whose scale is



(a) (b)

Fig. 5. 1D signal examples and their results in rolling guidance. (a) One small structure. (b) One edge of a large structure.

smaller than σ_s as claimed in the scale space theory. It is implemented efficiently by separating kernels in perpendicular directions. Approximation by box filter is also feasible.

4.2 Edge Recovery

The iterative edge recovery step forms the major contribution in our method. In this process, an image J is iteratively updated. We denote J^{t+1} as the result in the t-th iteration. Initially, J^1 is set as G in Eq. (2), which is the output of Gaussian filtering. The value of J^{t+1} in the t-th iteration is obtained in a joint bilateral filtering form given the input I and the value in previous iteration J^t:

$$J^{t+1}(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\Big(- \frac{\|p-q\|^2}{2\sigma_s^2} - \frac{\|J^t(p) - J^t(q)\|^2}{2\sigma_r^2} \Big) I(q), \qquad (3)$$

where

$$K_p = \sum_{q \in N(p)} \exp\Big(- \frac{\|p-q\|^2}{2\sigma_s^2} - \frac{\|J^t(p) - J^t(q)\|^2}{2\sigma_r^2} \Big)$$

for normalization. I is the same input image used in Eq. (2). σ_s and σ_r control the spatial and range weights respectively.

This expression can be understood as a filter that smoothes the input I guided by the structure of J^t. This process is different by nature from how previous methods employ joint bilateral filter – we iteratively change the guidance image in passes. It yields illuminating effects, explained below. We name this iterative operation *rolling guidance*.

To demonstrate how it works, we show simple 1D examples in Fig. 5 where one small structure and one edge of a large structure are presented. The four rows show inputs and J^t obtained by rolling guidance respectively. Since this process uses J^t to compute the affinity between pixels, it makes resulting structures similar to J^t. Put differently, it yields structure transform from J to I.

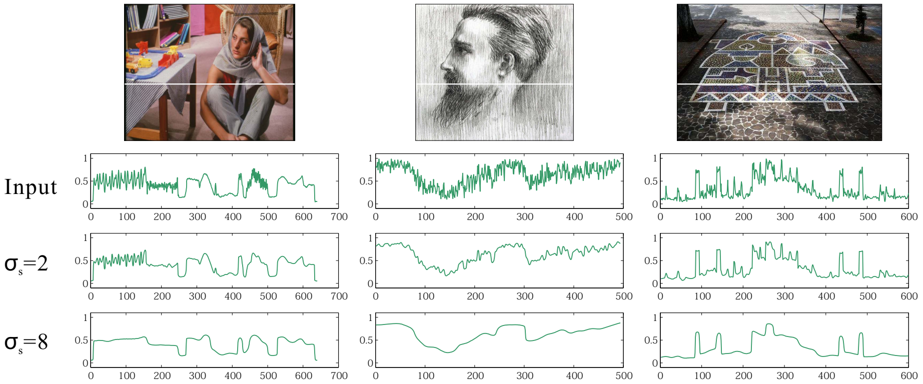


Fig. 6. 1D examples from real images. The curves are inputs and filtering results under different σ_s. Large-scale edges and changes are preserved well.

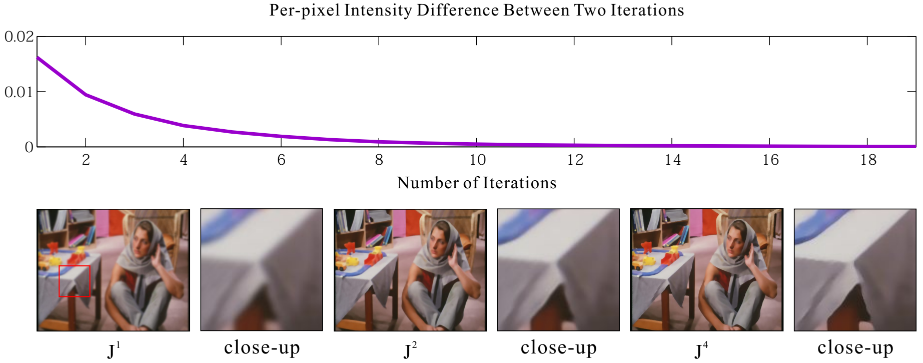
Small Structure In the first example (Fig. 5(a)), since the edges of the small structure are completely removed in J^1 by Gaussian filter, J^1 is mostly flat. In Eq. (3), the term $\|J^t(p) - J^t(q)\|$ is almost zero for any (p, q) pairs, which makes the joint bilateral filter behave like a Gaussian filter due to the inoperative range weight. Therefore, the output J^2 remains flat. All following iterations cannot add the detail back.

Large Structure In the second example (Fig. 5(b)), we show how rolling guidance processes a large scale edge. In the first iteration, J^1 is blurred and I is sharp. Since each-iteration process takes weighted average on I, result J^2 is smoother than, or at most with similar sharpness as I. It is also guaranteed to be sharper than J^1 because J^2 is smoothed weaker than J^1 due to the involvement of range weights. Notice that the range weight is no longer inoperative as in the small structure case. Now the order of smoothness can be expressed as

$$I \leq J^2 < J^1.$$

In following iterations, following the same analysis, sharpness of J^{t+1} is always between I and J^t since more range difference is involved in computing the weight and less averaging is preformed around the edge. It is conclusive that the sharpness is restored gradually and eventually goes back to original degree in the input, as shown in the bottom of Fig. 5(b).

Fig. 5 sheds light on understanding the new property of our scale-aware filter. It not only optimally preserves large scale edges, but also smoothes texture and other details. There is no need in prior to know how the structures are formed. As long as they are kept, even partially, after Gaussian smoothing, our method can recover the large-scale shape with nice edge preserving. Whether an edge is preserved or not is not dependent of its magnitude, marking the inherent difference to other edge-preserving filters.



J^1 close-up J^2 close-up J^4 close-up

Fig. 7. Plot of difference between input and output images in iterations. The difference of two successive iterations reduces monotonically and the result is guaranteed not an all-constant image. We use $\sigma_s = 4$ and $\sigma_r = 0.1$ for this example. Please view them in the original resolutions to compare all details.

By changing the value of σ_s, this framework can be used to remove structures in different scales. Fig. 6 shows the 1D examples from real images. Structures of different scales are removed gradually while faithfully preserving necessary edges when σ_s grows.

Convergence Another fascinating property is that the rolling guidance iterations converge rapidly. We plot difference between J^t and J^{t+1} from all iterations in Fig. 7. For most average-based filters, such as Gaussian filter, bilateral filter [26], domain-transform filter [11], and guided filter [13], if they are performed repeatedly, the result of any image is eventually a constant-value one as all pixels are averaged to the same intensity. Unlike these filters, the rolling guidance procedure converges to a meaningful image faithful to the input no matter how many iterations are performed, which is worth further theoretical study. Empirically, We have tested this filter on thousands of images and failed it on *none* of them. Therefore, it is a filter with very attractive and unique features.

4.3 Combination of the Two Steps

In our framework, steps described in Sections 4.1 and 4.2 can be combined into one, by starting rolling guidance simply from a constant-value image. In Eq. (3), if we set all values in J^t to a constant C, i.e., $\forall p, J^t(p) = C$, it updates to

$$J^{t+1}(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\Big(- \frac{||p-q||^2}{2\sigma_s^2}\Big) I(q). \tag{4}$$

The form in Eq. (4) is exactly the same as Eq. (2). Therefore, we save one step by starting rolling guidance from J^0, where $\forall p, J^0(p) = C$. Correspondingly, the step to evolve from J^0 to J^1 is exactly the one explained in Section 4.1; later

Algorithm 1 Rolling Guidance Using Bilateral Filter

```
Input: $I$, $\sigma_s$, $\sigma_r$, $N^{\text{iter}}$
Output: $I^{\text{new}}$
Initialize $J^0$ as a constant image
for t:= 1 to $N^{\text{iter}}$ do
    $J^t \leftarrow JointBilateral(I, J^{t-1}, \sigma_s, \sigma_r)$ {Input: $I$; Guidance: $J^{t-1}$ }
end for
$I^{\text{new}} \leftarrow J^{N^{\text{iter}}}$
```

iterations correspond to the step in Section 4.2. This modification makes our algorithm even more general and easier to understand. Algorithm 1 depicts our final scale-aware filter construction.

4.4 Comparison with Other Techniques

In this section, we compare our method with other related ones and clarify their inherent difference.

Successively Performed Bilateral Filtering When successively applying bilateral filtering or other average-based edge-aware filters, a general expression becomes

$$J^0 = I, \quad J^{t+1} = f(J^t), \tag{5}$$

where J^0 is the initial image, J^t is the result in the t-th iteration, and $f(\cdot)$ is the filter. This process has two fundamental differences to ours.

First, successively performing edge-aware filters does not remove small-scale structures. Instead, they could over-smooth large-scale ones. One example is shown in Fig. 8(b)-(c). Second, average-based filters do not converge as ours as discussed above. A final blurred image is obviously not what we want in scale-aware filtering.

Joint Bilateral Upsampling Similarly denoting input image as I and output as J, this technique can be expressed as

$$J(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\Big(-\frac{||p-q||^2}{2\sigma_s^2} - \frac{||I(p)-I(q)||^2}{2\sigma_r^2}\Big) M(q_{\downarrow}), \tag{6}$$

where M is a downsampled image and $p_{\downarrow}$ is the corresponding coordinate of p in M. If we set M as a downsampled input image $I_{\downarrow}$, the formulation changes to

$$J(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\Big(-\frac{||p-q||^2}{2\sigma_s^2} - \frac{||I(p)-I(q)||^2}{2\sigma_r^2}\Big) I_{\downarrow}(q_{\downarrow}), \tag{7}$$

which seems similar to our algorithm.

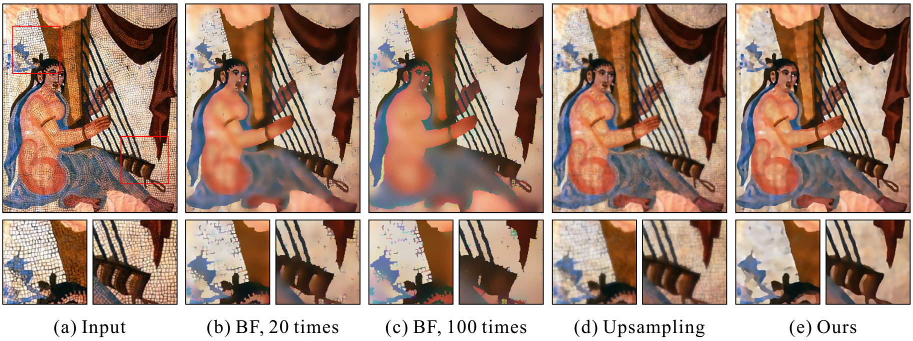


Fig. 8. Comparison with other techniques. (a) Input image. (b) Successively performing bilateral filter for 20 times with $\sigma_s = 3$ and $\sigma_r = 0.1$. (c) Successively performing bilateral filter for 100 times with the same parameters. (d) Joint bilateral upsampling. Downsampling is with Gaussian filter with $\sigma_s = 3$ and upsampling is with $\sigma_s = 3$ and $\sigma_r = 0.1$. (e) Our result with the same parameters in 5 iterations.

In fact, the difference between these processes can be understood in three major aspects. (1) Joint bilateral upsampling uses the clear image as guidance. Contrarily, our method uses a blurry image to guide filtering. (2) Joint bilateral upsampling does not remove small-scale structures because the guidance image already contains many of them, as illustrated in Fig. 8(d). (3) It is not an iterative process. If conducted successively, joint bilateral upsampling results in a very blurry image like the case of bilateral filter.

4.5 Extensible Rolling Guidance Design

Our rolling guidance is a general framework with the freedom to use almost all types of joint filters. Our method allows for iteratively replacing the guidance image with the filtering result in previous pass. It is thus not restricted to joint bilateral filter. Average-based joint filters, such as domain-transform [11], guided filter [13], and recursive bilateral filter [32], all can be employed in our framework.

In Fig. 9, we compare results generated using bilateral filter, domain-transform filter, and guided filter as the rolling guidance. They generate similar scale-aware results although, when applying alone, they are edge-preserving. Difference can be observed as well among the results. In particular, applying bilateral filter guidance creates smoothly curved edges, as shown in Fig. 9(b). Domain-transform filter guidance, on the contrary, generates many horizontal and vertical edges as shown in (c), since it performs filtering by scanning the image in these directions. Finally, guided filter as guidance produces smoother results as shown in (d). It is in accordance with the explanation provided in the original paper [13].

The generality of this rolling guidance framework makes it extensible in many ways to suit various applications and scenarios. It also provides loads of possibilities for further approximation and acceleration when replacing the bilateral filter by others.

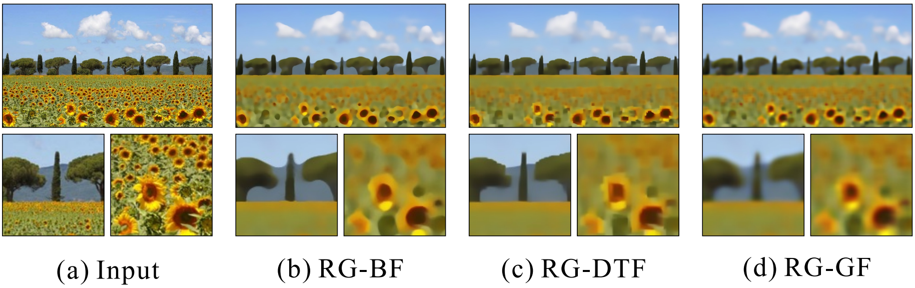


Fig. 9. Results when setting the rolling guidance as bilateral filter (b), domain-transform filter (c), and guided filter (d). The parameters are as follows. (b) ($\sigma_s = 5, \sigma_r = 0.1$); ($\sigma_s = 2.8, \sigma_r = 0.1$); ($\sigma_s = 4, \sigma_r = 0.1$). (c) ($\sigma_s = 10, \sigma_r = 0.1$); ($\sigma_s = 6, \sigma_r = 0.1$); ($\sigma_s = 6, \sigma_r = 0.1$). (d) ($r = 6, \epsilon = 0.003$); ($r = 3, \epsilon = 0.0012$); ($r = 3, \epsilon = 0.0036$).

Table 1. Running time comparison. "s/Mp" stands for "seconds per mega-pixel". Our method uses 4 iterations. The first iteration uses fast Gaussian filter.

Algorithms	Grayscale Image (s/Mp)	Color Image (s/Mp)
Local Extrema [25]	95	–
RTV [30]	14	35
Zhang et al. [34]	5	16
Covariance M2 [14]	198	614
Ours with BF [33]	0.2	–
Ours with BF [1]	–	2
Ours with GF [13]	0.12	0.8
Ours with DTF [11]	**0.05**	0.15

5 Experiments

We conduct experiments to more extensively evaluate this filter. Besides Fig. 6, we show another interesting example in Fig. 10 to exhibit the unique property of rolling guidance filter. The input image contains a few square patterns in different scales. Results of edge-aware filters are shown in (a). They do not remove even the smallest scale square because all edges are with high contrast. Our results are shown in (b) using different σ_s. Squares are progressively removed according to their scales in an ascending order. More results of our scale-aware filter on natural images are shown in Fig. 11.

Our experiments are conducted on a PC with an Intel i7 3.4GHz CPU and 8GB memory. Only a single thread is used without involving any SIMD instructions. Since acceleration of edge-aware filtering was well studied, many tools are available [8, 19, 28, 22, 33, 32, 2, 1, 11–13]. They can be used directly in our framework to accelerate rolling guidance. Our implementation is easy with only several lines of code based on Algorithm 1. In Table 1, we tabulate running time of implementing our method using different rolling guidance options (Section 4.5). It also

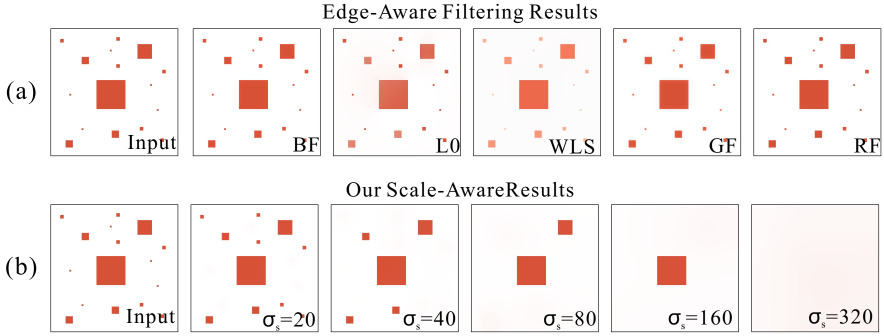


Fig. 10. (a) Edge-aware filtering results on the image composed of squares in different sizes. We apply bilateral filter [26] ($\sigma_s = 40$, $\sigma_r = 0.1$), L_0 filter [29] ($\lambda = 1000$), WLS filter [9] ($\lambda = 5$, $\alpha = 1.2$), guided filter [13] ($\epsilon = 0.01$, $r = 20$), and recursive filter [11] ($\sigma_s = 10$, $\sigma_r = 0.1$). (b) Our results under different σ_s. σ_r is set to 0.1. 4 iterations are used. The recursive filter [11] is used in rolling guidance.

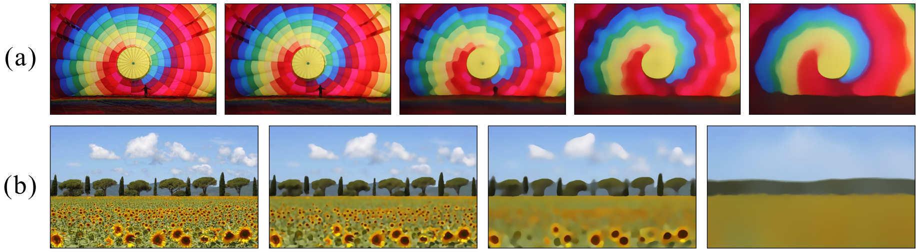


Fig. 11. Scale-aware filtering results on natural images.

lists time of running related texture smoothing methods. Our implementation is 10-100 times faster than the previous fastest method [34]. The implementation with DTF [11] achieves realtime performance even for one-megapixel images.

Applications Due to the useful scale-aware property and fast speed, our filter profits a variety of applications, including detail enhancement, tone-mapping, denoising, edge extraction, de-halftoning, JPEG artifact removal, multi-scale structure decomposition, segmentation, saliency detection, optical flow, and stereo matching.

We show results of texture smoothing and virtual contour restoration in this section. More applications can be found on our project website. Texture on the object surface is generally of small scale structures. Our algorithm can painlessly remove them in different levels. Meanwhile, global color variation and main edges can be preserved. Our implementation is much faster than all other texture smoothing tools. Results and comparisons are provided in Fig. 12.

In natural images, not all contrast that human perceive can be expressed in form of large gradients. Virtual edges are thus common and correspond to

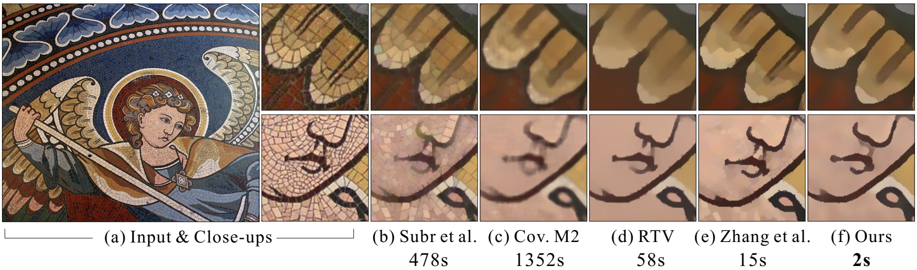


Fig. 12. Texture smoothing results and close-ups. (b)-(e) are results of [25], [14], [30], and [34] respectively. Parameters are (b) $k = 5$, (c) $\sigma = 0.3$, $k = 9$, (d) $\lambda = 0.015$, $\sigma = 5$, (e) $r = 3$, $\sigma_r = 0.3$, 10 iterations, (f) $\sigma_s = 5$, $\sigma_r = 0.1$.

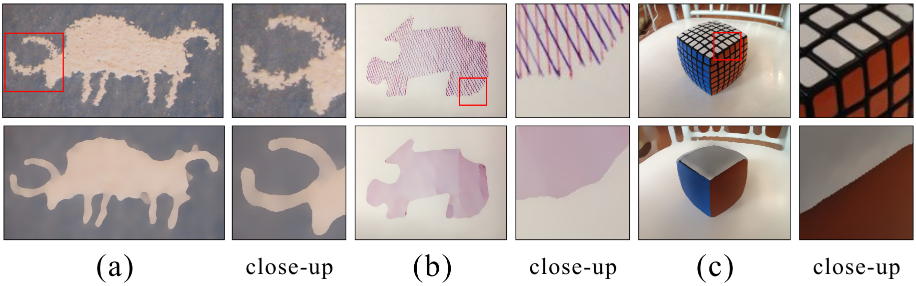


Fig. 13. Virtual contour restoration. Large contrast naturally forms region boundaries in human perception. Our filter can simulate this process.

large-scale image information possibly useful for segmentation. Our algorithm can restore these originally nonexistent edges reflecting large scale contrast, as shown in Fig. 13.

6 Concluding Remarks

We have presented a new type of filter for scale-aware operations. It is the first in its kind to remove small-scale structures while preserving other content, parallel in terms of importance to previous edge-preserving filters. This framework is simple to implement, greatly extensible to accommodate various tools for rolling guidance, and yields decent performance. Our future work will be to apply it to more applications.

Acknowledgements

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 412911).

References

1. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. Comput. Graph. Forum 29(2), 753–762 (2010)
2. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian kd-trees for fast high-dimensional filtering. ACM Transactions on Graphics (TOG) 28(3), 21 (2009)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 898–916 (2011)
4. Bao, L., Song, Y., Yang, Q., Yuan, H., Wang, G.: Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE Transactions on Image Processing 23(2), 555–569 (2014)
5. Brox, T., Cremers, D.: Iterated nonlocal means for texture restoration. In: Scale Space and Variational Methods in Computer Vision, pp. 13–24. Springer (2007)
6. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Trans. Graph. 26(3), 103 (2007)
7. Criminisi, A., Sharp, T., Rother, C., Perez, P.: Geodesic image and video editing. ACM Trans. Graph. 29(5), 134 (2010)
8. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Transactions on Graphics (TOG) 21(3), 257–266 (2002)
9. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Trans. Graph. 27(3) (2008)
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010)
11. Gastal, E.S., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Transactions on Graphics (TOG) 30(4), 69 (2011)
12. Gastal, E.S., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. ACM Transactions on Graphics (TOG) 31(4), 33 (2012)
13. He, K., Sun, J., Tang, X.: Guided image filtering. In: ECCV, pp. 1–14. Springer (2010)
14. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Transactions on Graphics (TOG) 32(6), 176 (2013)
15. Kass, M., Solomon, J.: Smoothed local histogram filters. ACM Transactions on Graphics (TOG) 29(4), 100 (2010)
16. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Trans. Graph. 26(3), 96 (2007)
17. Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. Journal of applied statistics 21(1-2), 225–270 (1994)
18. Ma, Z., He, K., Wei, Y., Sun, J., Wu, E.: Constant time weighted median filtering for stereo matching and beyond. In: ICCV. IEEE (2013)
19. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: ECCV, pp. 568–580. Springer (2006)
20. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
21. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM transactions on graphics (TOG) 23(3), 664–672 (2004)

22. Porikli, F.: Constant time o (1) bilateral filtering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)
23. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1), 259–268 (1992)
24. Su, Z., Luo, X., Deng, Z., Liang, Y., Ji, Z.: Edge-preserving texture suppression filter based on joint filtering schemes. IEEE Transactions on Multimedia 15(3), 535–548 (2013)
25. Subr, K., Soler, C., Durand, F.: Edge-preserving multiscale image decomposition based on local extrema. ACM Transactions on Graphics (TOG) 28(5), 147 (2009)
26. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: International Conference on Computer Vision (ICCV). pp. 839–846. IEEE (1998)
27. van de Weijer, J., Van den Boomgaard, R.: Local mode filtering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. II–428 (2001)
28. Weiss, B.: Fast median and bilateral filtering. ACM Trans. Graph. 25(3), 519–526 (2006)
29. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l 0 gradient minimization. ACM Transactions on Graphics (TOG) 30(6), 174 (2011)
30. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Transactions on Graphics (TOG) 31(6), 139 (2012)
31. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1155–1162 (2013)
32. Yang, Q.: Recursive bilateral filtering. In: ECCV, pp. 399–413. Springer (2012)
33. Yang, Q., Tan, K.H., Ahuja, N.: Real-time o(1) bilateral filtering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 557–564 (2009)
34. Zhang, Q., Xu, L., Jia, J.: 100+ times faster weighted median filter. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)