Semantic Filtering*

Qingxiong Yang
School of Information Science and Technology
University of Science and Technology of China

Abstract

Edge-preserving image operations aim at smoothing an image without blurring the edges. Many excellent edge-preserving filtering techniques have been proposed recently to reduce the computational complexity or/and separate different scale structures. They normally adopt a user-selected scale measurement to control the detail/texture smoothing. However, natural photos contain objects of different sizes which cannot be described by a single scale measurement. On the other hand, edge/contour detection/analysis is closely related to edge-preserving filtering and has achieved significant progress recently. Nevertheless, most of the state-of-the-art filtering techniques ignore the success in this area. Inspired by the fact that learning-based edge detectors/classifiers significantly outperform traditional manually-designed detectors, this paper proposes a learning-based edge-preserving filtering technique. It synergistically combines the efficiency of the recursive filter and the effectiveness of the recent edge detector for scale-aware *edge-preserving filtering. Unlike previous filtering methods, the propose filter can efficiently extract subjectively-meaningful structures from natural scenes containing multiple-scale objects.*

1. Introduction

Edge-preserving filtering has found widespread use in many computer vision and graphics tasks. It is an image smoothing technique that removes low-contrast details/textures while maintaining sharp edges/image structures.

A large category of edge-preserving filters are designed with a specific filter kernel to measure the distance between two pixels in a local region. The distance measurement is then converted to the confidence of an edge between the two pixels for edge-aware filtering. This category is very sensitive to noise/textures. Examples are anisotropic diffusion [34], bilateral filter [42], guided image filter [23] and domain transform filter [20]. The main challenge in this category is accurately including scale measurement for filter design to distinguish textures/noise from image structure. Designing a robust edge-aware filter kernel is surprisingly difficult. There is not an "optimal" solution as the detection of image edges can only be evaluated in a subjective manner. Meanwhile, numerical experiments like the Berkeley Segmentation Dataset and Benchmark (BSDS500) [3] demonstrate that human subjects have various perceptions of edges in a same image.

On the other hand, significant progress has been achieved in machine learning over the past few years. Unlike standard image processing techniques that use strictly static program instructions, it normally builds a model based on example inputs and then uses it to generate predictions or decisions. The performance of a learning-based edge-preserving image filter is thus likely to be closer to human visual system when the example inputs are obtained based on average agreement between sufficient human subjects. The svm-based filter presented in [51] is the first learning-based bilateral filter. As Taylor series expansion of the Gaussian function can be used to approximate the bilateral filters [38], [51] learns a function that maps feature vector comprising of the exponentiation of the pixel intensity, the corresponding Gaussian filtered response, and their products to the corresponding exact bilateral filtered values from the training image.

This paper aims at developing image smoothing techniques that can preserve edges between different-size objects/structures. It is a much more challenging problem. Unlike the bilateral filter, it cannot be approximated using Taylor series expansion and there is not any ground-truth filtered image for training. Nevertheless, there are sufficient hand-labelled segmentation data sets. For instance, BSDS500 [3] contains 12,000 hand-labelled segmentations of 1,000 Corel dataset images from 30 human subjects. Image segmentation and edge detection is closely related to image smoothing technique. They normally pre-smooth the image with a specific low-pass filter for noise reduction.

Because human visual system is capable to understand semantically meaningful structures blended with or formed

*This work was partially supported by the Early Career Scheme through the Research Grants Council of Hong Kong (Project Number: CityU 21201914) and a grant from Qualcomm Technologies, Inc.

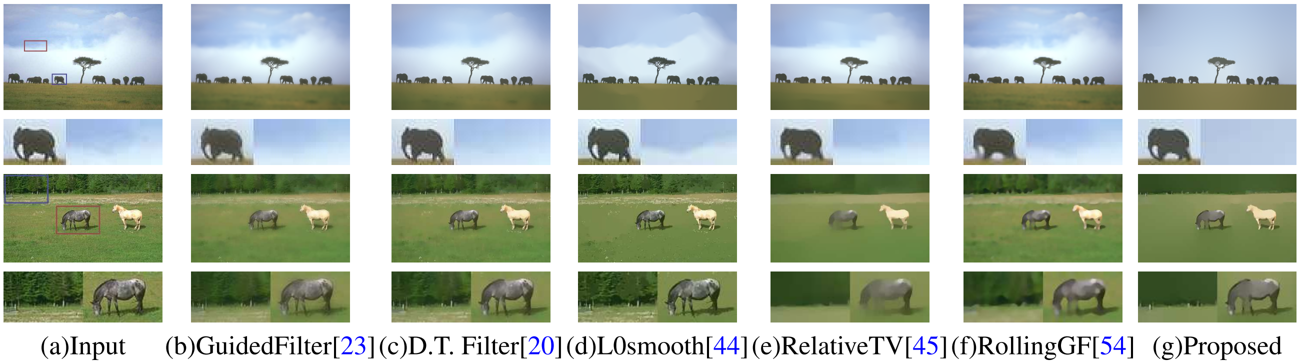

(a)Input (b)GuidedFilter[23] (c)D.T. Filter[20] (d)L0smooth[44] (e)RelativeTV[45] (f)RollingGF[54] (g)Proposed

Figure 1. Natural scenes like (a) contain objects of different sizes and structures of various scales. As a result, the state-of-the-art edge-preserving filters are unlikely to obtain "optimal" smoothing result without parameter adjustment as shown in (b)-(f). *The default parameters included in the implementations published by the authors were used in this experiment*. This paper aims at an efficient scale-aware filtering solution by integrating fast edge-preserving filtering technique with prior knowledge learned from human segmentation results. Unlike previous filters, it can sufficiently suppress image variance/textures inside large objects while maintaining the structures of small objects as shown in (g).

by texture elements [4], a "perfect" segmentation result (which agrees with human subjects) is obviously an excellent guidance for scale-invariant edge-preserving filtering. However, it is very difficult to obtain ideal edges from real life images of moderate complexity. Traditional edge detectors rely on image gradients (followed by non-maximal suppression). Unfortunately, many visually salient edges like texture edges do not correspond to image gradients. As a result, most of the state-of-the-art edge-preserving filters ignore the potential contribution from an edge detector. On the other hand, it has been demonstrated recently that the performance of learning-based edge detectors is approaching human subjects [12, 13].

This paper proposes a simple seamless combination of the recursive filtering technique and the learning-based edge classification technique for fast *scale-aware* edge-preserving filtering. We observed that a number of fast edge-preserving filtering techniques including anisotropic diffusion [34], recursive bilateral filter [47] and domain transform filter [20] operate on the differential structure of the input image. They recursively smoothen an image based on the similarity between *every two neighboring pixels* and are referred to as **Anisotropic Filters** in this paper. These filters are naturally more sensitive to noise than others like the bilateral filter [42] or guided image filter [23]. They cannot separate meaningful structures from textures either. However, their computational complexity is lower as they can be implemented recursively. Another great advantage is that they can be naturally combined with a state-of-the-art learning-based edge classifier [13] to encourage smoothing within regions until reaching strong edges. Such an edge classifier is trained using human-labelled contours. According to [4], "the overall structural features are the primary data of human perception, not the individual details". Learning-based edge detectors can thus robustly distinguish the contours of different-size objects from image noise and textures. When integrating with an anisotropic filter, it enables robust structure extraction from natural scenes containing objects of various scales as demonstrated in Fig. 1. Fig. 1 (a) contains two images with both large scale objects (e.g., sky and meadow) and small scale objects (e.g., animals). Current state-of-the-art edge-preserving filters cannot successfully separate large scale objects from the small ones without parameter tuning as shown in Fig. 1 (b)-(f). The proposed filtering technique does not have this limitation as demonstrated in Fig. 1 (g). Its computational complexity is also low due to the efficiency of the adopted edge classifier and the recursive filters.

Compared with the other edge-preserving filters, the proposed filtering technique has the following advantages:

- It is robust to natural scenes containing objects of different sizes and structures of various scales, and thus can successfully extract subjectively-meaningful structures from images containing multiple-scale objects.

- Its computational complexity is low and real-time performance can be achieved. The major computation cost of the proposed filtering technique resides in the adopted edge classifier [13] and recursive filter, and both can obtain real-time performance.

2. Related Work

2.1. Edge-Preserving Filtering

The most popular edge-preserving filter is likely to be the bilateral filter introduced by Tomasi and Manduchi [42]. It has been used in many computer vision and computer graphics tasks, and a general overview of the applications can be found in [30]. Let x_i denote the color of an image x

at pixel i and y_i denote the corresponding filtered value,

$$y_i = \frac{\sum_{j \in \Omega_i} G_{\sigma_s}(|i-j|) G_{\sigma_r}(|x_i - x_j|) x_j}{\sum_{j \in \Omega_i} G_{\sigma_s}(|i-j|) G_{\sigma_r}(|x_i - x_j|)}, \qquad (1)$$

where j is a pixel in the neighborhood Ω_i of pixel i, and G_{σ_s} and G_{σ_r} are the spatial and range filter kernels measuring the spatial and range/color similarities. The parameter σ_s defines the size of the spatial neighborhood used to filter a pixel, and σ_r controls how much an adjacent pixel is down-weighted because of the color difference. A joint (or cross) bilateral filter [36, 16] is the same as the bilateral filter except that its range filter kernel G_{σ_r} is computed from another image named guidance image.

The brute-force implementation of the bilateral filter is slow when the kernel is large. A number of techniques have been proposed for fast bilateral filtering based on quantization on the spatial domain and/or range domain [15, 37, 9, 29, 50, 49, 2, 1, 21]. Other techniques reduce the computational complexity with additional constraints on the spatial filter kernel [43, 38] or the range filter kernel [47, 48].

Besides accelerating bilateral filter, there are efficient bilateral-filter-like techniques derived from the anisotropic diffusion [34], weighted least square (WLS) measure [18], wavelets [19], linear regression [23], local Laplacian pyramid [31, 5, 32], domain transform [20] and L_0 gradient minimization [44].

These edge-preserving filters have found widespread computer graphics applications. However, they all focus on relatively small variance suppression and vulnerable to textures. The proposed filtering technique is different in that it can distinguish meaningful structures from textures and image noise.

2.2. Structure-Preserving Filtering

Traditional edge-preserving filtering techniques cannot distinguish textured regions/patterns from the major structures of an image. Popular structure-preserving techniques are based on the total variation (TV) model [40]. It uses L1-norm based regularization constraints to enforce large-scale edges, and has demonstrated good separations of structure from texture [28], [52], [6]. Xu et al. [45] proposes relative total variation measures to better capture the difference between texture and structure, and develops an optimization system to extract main structures. Another model was proposed by [41]. It separates oscillations from the structure layer through extrema extraction and extrapolation. Alternatively, better similarity metrics like geodesic [10] or diffusion [17] distances instead of traditional Euclidean distances can enhance the ability of texture-structure separation. Recently, [24] uses second order statistics as a patch descriptor to capture the difference between structure and texture.

Structure-preserving filtering is normally slow before the availability of the rolling guidance filter [54]. Besides the efficiency, [54] also proposes a unique scale measure to control the level of details during filtering. This scale measure is very useful when manual adjustment is required/allowed. Unlike [54] and most of the available structure-preserving filters, this paper aims at extracting meaning structures regardless of the sizes/scales of the objects/structures. This type of technique is desired in computer vision.

2.3. Edge Detection

Edge detection is a fundamental task in computer vision and image processing. Traditional approaches like the Sobel operator [14] detects edges by convolving the input image with local derivative filters. The most popular edge detector - Canny detector [7] makes extensions by adding non-maximum suppression and hysteresis thresholding steps. These approaches use low-level interpolation of the image structures, and an overview can be found in [57]. Recent works focusing on utilizing machine learning techniques. They either train an edge classifier based on local image patches [11, 26, 39, 22, 12, 58, 13] or make use of learning techniques for cue combination [3, 56, 8, 59]. Deep neural networks were also applied to edge detection recently [25].

Traditional edge detectors rely on image gradients while many visually salient edges like texture edges do not correspond to image gradients. As a result, they are not suitable for structure-preserving filtering. However, the state-of-the-art detectors [12, 13] are learned from human-labelled segmentation results including sufficient texture edges. As a result, they contain useful and accurate structural information that can be adopted for robust structure-preserving filtering.

3. Scale-Aware Structure-Preserving Filtering

3.1. Anisotropic Filtering

Anisotropic diffusion is a popular edge-aware filtering technique [34]. It is modeled using partial differential equations and implemented as an iterative process. The recent domain transform filter (**DTF**) [20] and the recursive bilateral filter [47] are closely related to anisotropic diffusion and can achieve real-time performance. Due to page limit, only a brief introduction of DTF is presented in this section. The proposed filtering technique is identical to DTF except that the distance measurement in DTF (Eq.3) will be adjusted using edge confidence computed from an edge classifier and the proposed filter need to repeat iteratively until converge. As a result, a fully understand DTF is indeed not required.

Given a one-dimensional (1D) signal, DTF applies a distance-preserving transformation to the signal. A perfect distance-preserving transformation does not exist, but a simple approximation is the sum of the spatial distance (e.g., one pixel distance) and color/intensity difference between every two pixels. Let x denote the 1D input signal, and t denote the transformed signal,

$$t_i = x_0 + \sum_{j=1}^{i}(1 + |x_j - x_{j-1}|). \tag{2}$$

Similar to the bilateral filter, two additional parameters σ_s and σ_r will be included in Eq. (2) to adjust the amount of smoothness in practice:

$$t_i = x_0 + \sum_{j=1}^{i}(1 + \frac{\sigma_s}{\sigma_r}|x_j - x_{j-1}|). \tag{3}$$

As can be seen from Eq. (3), this transform operates on the differential structure of the input signal, which is the same as the anisotropic diffusion. The recursive implementation of a standard low-pass filter like the Gaussian filter with kernel defined by σ_s will be used to smooth the transformed signal (Eq.3) without blurring the edges, and the smoothed image is the output of DTF.

DTF filters 2D signals using the 1D operations by performing separate passes along each dimension of the signal. [20] demonstrates that artifact-free filtered images can be obtained by performing filtering iteratively and the filter kernel size (defined by σ_s and σ_r) should be reduced iteratively to guarantee convergence.

3.2. Structure-Preserving Anisotropic Filtering

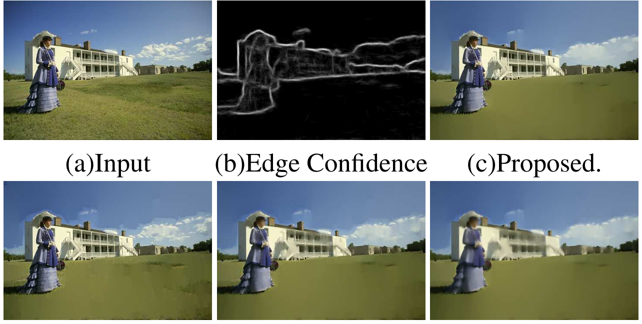
The anisotropic filters can be implemented recursively and thus the computational complexity is relatively low. However, they are sensitive to image noise and cannot distinguish textures from structures. Available solutions either manually design low-level vision model(s) and descriptor(s) [40, 45, 24] to capture the difference between structure and texture or simply adopt a texture scale parameter [54]. The performance of these filters is excellent when perfect parameters are used. Nevertheless, natural photos contain objects of different sizes and structures of various scales which are hard to describe using a unified low-level feature. In contrast, it has already been demonstrated on closely related research (like edge detection) that high-level features learned from human-labelled data can significantly outperform manually-designed features. This section makes use of the state-of-the-art structured learning based edge classifier [12, 13] for structure-preserving filtering while maintain its efficiency. The sufficient human-labelled training examples from BSDS500 benchmark [3] enable the proposed filtering technique to be robust to various texture scales.

DTF accumulates the image gradients to measure the distance between two pixels as can be seen from Eq. (3). However, it is clear that texture edges do not correspond to image gradients. A straightforward learning-based solution is training a deep learning architecture to map a local patch to a "perfect" image gradient value so that it is low inside a region and high around texture edges. However, it will be slow. A simpler but much faster solution is thus adopted in this paper. By taking the advantage of the inherent structure in edges in a local patch, [13] proposes a generalized structured learning approach for edge classification. It has been demonstrated to be very robust to textures and very efficient.

A straight-forward solution to structure-preserving smoothing is using the edge confidence computed from [13] as the guidance in DTF to smooth the input image. This type of filter is named *cross/joint* (bilateral) filter in the literature [36, 16]. Let f_j[1] denote the confidence of an edge at pixel j. f_j is the output of [13] at pixel j and represents the probability of an edge at j. Eq. (3) can be modified as follows to suppress gradients (due to textures) inside a region:

$$t_i = x_0 + \sum_{j=1}^{i}(1 + \frac{\sigma_s}{\sigma_r} \cdot f_j), \tag{4}$$

The corresponding filter is referred to as *cross DTF*. This is a good solution given a perfect edge classifier which however, does not exist in practice. It may introduce visible artifacts or blur the image as shown in Fig.3(d).



(a)Input (b)Edge Confidence (c)Proposed.

(d) Cross DTF (w.r.t. different parameters).

Figure 3. Direct use of the edge confidence as the guidance may introduce visible artifacts or blur the image as shown in (d) (zoom in for details). (c) shows that the combination of the image gradients from (a) and the edge confidence in (b) can effectively suppress the potential artifacts due to imperfect edge detection.

This paper proposes to use the edge confidence to adjust the original distance measurement in DTF:

$$t_i = x_0 + \sum_{j=1}^{i}(1 + \frac{\sigma_s}{\sigma_r} \cdot f_j|x_j - x_{j-1}|), \tag{5}$$

The combination of the edge confidence and the image gradients can effectively suppress the potential artifacts cause by incorrect edge detection.

[1] f_j will be re-computed after every iteration.

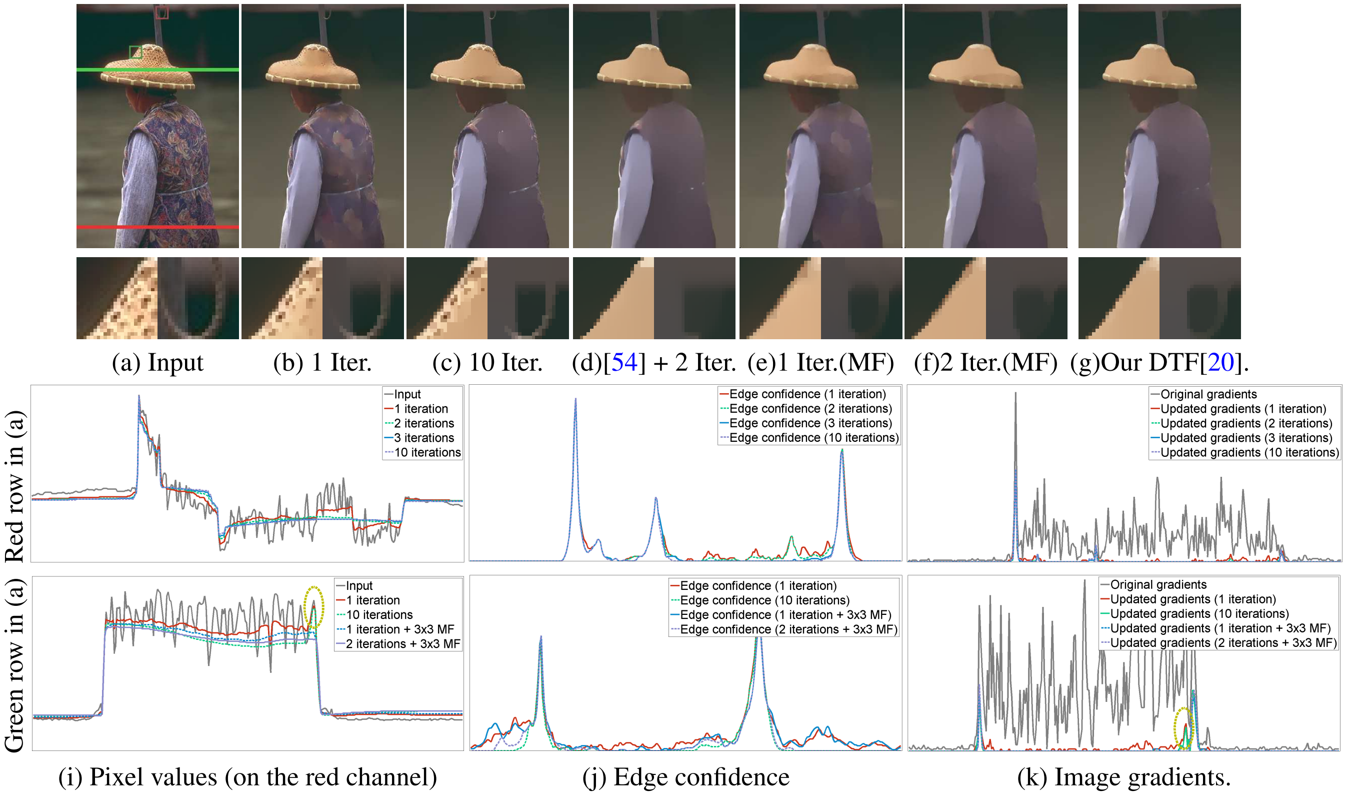


Figure 2. Integration of anisotropic filters with edge detector can successfully remove textures except for small-scale textures around the edges as can be seen in (b)-(c). A simple and robust solution is removing these textures using rolling guidance filter [54] in advance as demonstrated in (d). However, it will be relatively slow. To improve the efficiency, this paper alternatively uses the differential structure of median filtered image to smooth the original input image as shown in (g). Dashed ellipses will be discussed in text.

The gray curves in Fig. 2 (k) represent the original image gradients of the red and green rows of Fig. 2 (a), respectively. Traditional anisotropic filters are vulnerable to textures in these two rows. The red curves in Fig. 2 (j) represent the edge confidence detected from these two rows, respectively. The peaks in the two red curves correspond to the edges in the red and green rows in Fig. 2 (a). The edge confidence is used to suppress the image gradients inside the textured regions and enable texture removal according to Eq. (5). The red curves in Fig. 2 (k) represent the modified image gradients (which correspond to the $f_j|x_j - x_{j-1}|$ values in Eq. 5). Note that the variance of the image gradients inside the textured regions were significantly suppressed, and thus the resulted anisotropic filter can successfully remove most of the textures (e.g., the hat) as can be seen in Fig. 2 (b) and the red curves in (i). New edge confidence can be obtained from the filtered image and used to further suppress the textures/noise.

The proposed structure-preserving filtering technique iteratively compute edge confidence using structured-learning based edge classifier and use it to suppress the textures until convergence. Same as [20], the filter kernels, that is σ_s and σ_r in Eq.(5) are iteratively reduced by half to guarantee convergence[2]. Although it will not affect the transform in Eq.(5), reducing σ_s will decrease the filter kernel of the low-pass filter used to smooth the transformed signal (see the text below Eq.3 for details), and thus convergence can be guaranteed. Fig. 2 (c) presents the filtered image after 10 iterations. It shows that most of the visible textures are removed. The green, blue and purple lines in the first row of Fig. 2 (i)-(k) correspond to the pixel intensities, edge confidence measurements and updated image gradients of the red row in (a) after 2, 3 and 10 iterations, respectively. It shows that the proposed filter converges quickly (after only about 3 iterations).

3.2.1 Suppress small-scale textures around edges

The filter presented in Eq. (5) cannot sufficiently remove small-scale textures around highly-confident edges as shown in the close-ups below Fig. 2 (b)-(c). This is due to the imperfect confidence measurements around a texture edge. As shown in the yellow ellipse in Fig. 2 (k), large image gradients around texture edges cannot be effectively suppressed even after a large number of iterations. As a

[2] σ_s = 0.4 and σ_r =0.04 in all the conducted experiments.

result, the original pixel values will be preserved as can be seen from the yellow ellipse in Fig. 2 (i). Applying a small median filter (**MF**) to the input image cannot significantly affect the edge confidence around highly-confident edges as can be seen in the blue and purple lines in Fig. 2 (j). However, it is very effective for removing textures around edges as demonstrated in Fig. 2 (e)-(f) and the blue and purple lines in the second row of Fig. 2 (i) and (k) (see the values around the two yellow ellipses). Nevertheless, median filter will of course remove thin-structured objects as shown in the close-ups under Fig. 2 (e)-(f). Let $x^{\mathbf{MF}}$ denote the median filter result of the input signal x, this paper ONLY uses $x^{\mathbf{MF}}$ to compute the image gradients in Eq. (5). The proposed structure-preserving DTF are computed as follows:

$$t_i = x_0 + \sum_{j=1}^{i}(1 + \frac{\sigma_s}{\sigma_r} f_j |x_j^{\mathbf{MF}} - x_{j-1}^{\mathbf{MF}}|), \quad (6)$$

Fig. 2 (g) presents the filtered images with the proposed DTF (Eq. (6)). They both successfully remove the textures in Fig. 2 (a)-(c) while better preserving the details around thin-structured objects. An alternative solution is directly applying rolling guidance filter [54] to the input image to remove the small-scale textures and the result is presented in Fig. 2 (d). However, it will be relatively slow.

The median filter size, σ_s and σ_r will be reduced by half after every iteration to guarantee convergence, and it normally converges after as few as two iterations. As a natural scene contains textures of different scales, the size of the employed median filter should be adjusted accordingly. Otherwise it may either blur structures or cannot completely remove all textures as shown in Fig. 4. This paper directly relies on the level of textureness which is represented by the percentage of low confident edges detected by [13] to decide the median filter size.

3.2.2 Computational complexity

The computation cost of the proposed filtering technique resides in the adopted state-of-the-art edge detector [13], DTF and the median filter [43, 35]. They can all run in real-time and thus the whole pipeline will be very fast if the number of iterations is low. This paper uses the BSDS500 benchmark [3] to analyze the convergence problem. The average PSNR values from the filtered images between every two iterations are presented in Fig. 5. As can be seen, it is safe to stop after only **2 iterations** as the PSNR value computed from the filtered images after 2 and 3 iterations is higher than 40 dB (which corresponds to almost invisible difference). In practice, a downsample-version is sufficient for both the edge detector [13] and the median filter when the image resolution is relatively large (e.g., one megapixel). As a result, the computation cost mainly resides in the adopted anisotropic filter which operates on the full-resolution input image.

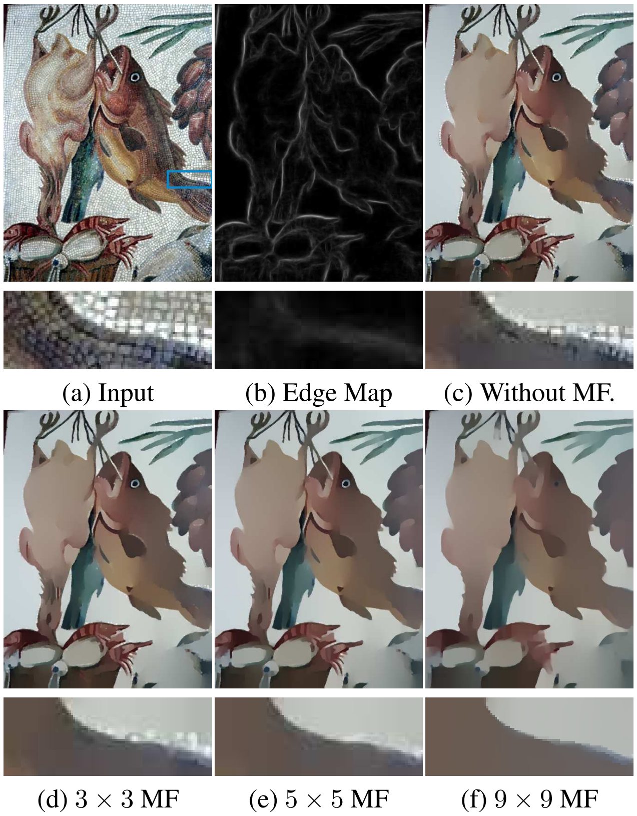

(a) Input (b) Edge Map (c) Without MF.
(d) 3×3 MF (e) 5×5 MF (f) 9×9 MF

Figure 4. Different median filter size will affect the texture removal result around edges and thus the filter size should be adjusted according to the scene content.

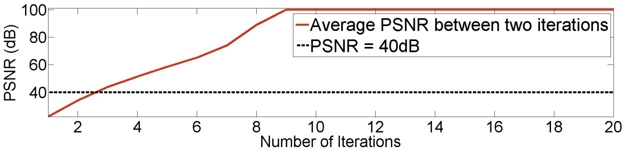
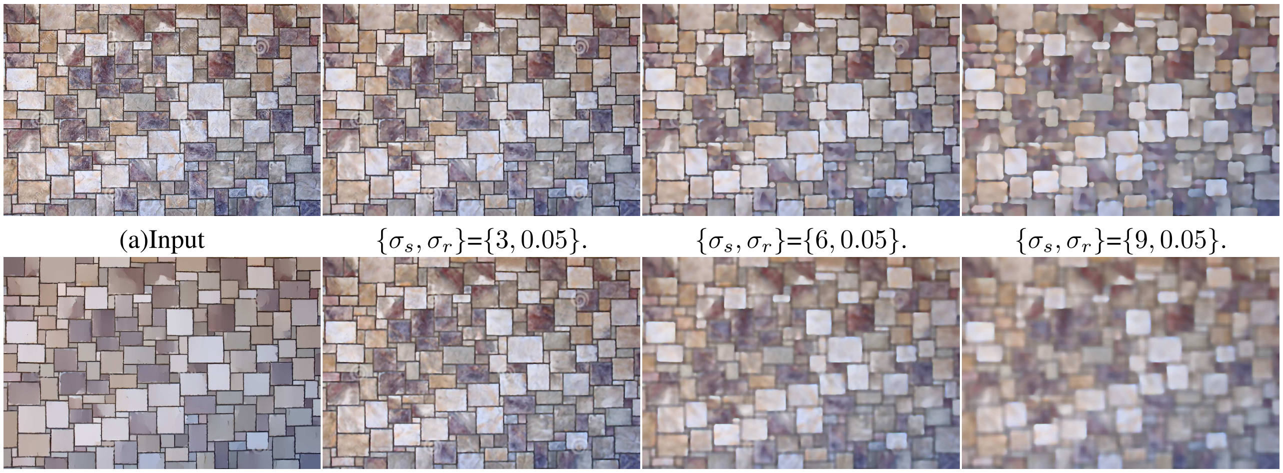


Figure 5. Average PSNR values computed from filtered images between every two iterations. PSNR values larger than 100 was shown as 100 for better visualization (as it can be infinity).

The complete comparison with the other state-of-the-art filters is presented in Table 1. The last row presents the runtime when the rolling guidance filter [54] is used as a pre-smoothing step (to remove small-scale textures around edges). Note that both the proposed filter and [54] are significantly faster than the others. However, [54] is not suitable for images containing various structure scales as shown in Fig. 6.

3.3. Comparison with the state-of-the-art

Fig. 7 visually compare the proposed filtering technique with the state-of-the-art filters under constant param-



(a)Input

$\{\sigma_s, \sigma_r\}$=$\{3, 0.05\}$.

$\{\sigma_s, \sigma_r\}$=$\{6, 0.05\}$.

$\{\sigma_s, \sigma_r\}$=$\{9, 0.05\}$.

(b)Proposed

(c)$\{\sigma_s, \sigma_r\}$=$\{3, 0.15\}$.

(d)$\{\sigma_s, \sigma_r\}$=$\{6, 0.15\}$.

(e)$\{\sigma_s, \sigma_r\}$=$\{9, 0.15\}$.

Figure 6. Proposed filter and the rolling guidance filter [54] are significantly faster than the others. Rolling guidance filter requires an estimate of the structure scale and can effectively smooth out the corresponding textures. However, it is not suitable for scenes containing objects/structures of multiple scales as can be seen from (c)-(e). It either cannot sufficiently remove textures in a large-scale object or blur small-scale objects. The computational complexity of proposed filter is close to [54] but is more robust to this limitation as demonstrated in (b).

Method	Runtime(Sec./Mp)
Relative TV[45]	35
L0 Smoothing[44]	18
WMF[55]	18
Covariance M2[24]	614
Rolling GF[54](with D.T. Filter[20])	0.15
Proposed (with D.T. Filter[20])	0.14
Proposed (with [54]+D.T. Filter[20])	0.27

Table 1. Exact runtime for processing a one megapixel color image on an Intel i7 3.4GHz CPU.

eter setting[3]. The adoption of learning-based edge detection technique[13] enable the proposed filter to be robust to natural scenes containing objects of different sizes and structures of various scales.

Quantitative evaluation of structure-preserving filtering is difficult, and thus the state-of-the-art methods [44, 54, 45] present only visual evaluation. In this section, we propose to numerically evaluate the improvement over the state-of-the-art saliency detection algorithm when the original image is processed by the state-of-the-art filters. We choose the ECSSD dataset [46] which is known to be difficult and Minimum Barrier Saliency (MBS) detection algorithm [53][4] which is the latest algorithm that outperforms all the others on this dataset. The precision-recall curves which evaluate the overall performance of a saliency detection method are presented in Fig. 8. Note that the proposed filter consistently outperforms the state-of-the-art filters [44, 54, 45]. The corresponding mean absolute errors (MAE) [33] and weighted-F-measure scores [27] are presented in Table 2 which show that the proposed filter has the lowest error rate and the highest weighted-F-measure score.

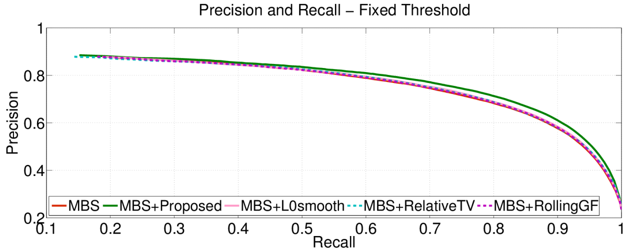


Figure 8. Precision-recall curves for saliency detection. Note that the combination of the structure-preserving filters can outperform the original MBS method on average and the proposed filter consistently outperforms the others.

Method	MBS	MBS+ L0S[44]	MBS+ RTV[45]	MBS+ RGF[54]	MBS+ **Proposed**
MAE	0.1707	0.1674	0.1660	0.1698	**0.1578**
WFM	0.5612	0.5668	0.5630	0.5606	**0.5846**

Table 2. Mean absolute errors (MAE) and weighted-F-measure scores (WFM). The proposed filter has the lowest error and the highest weighted-F-measure score.

[3] The default parameters included in the implementations published by the authors of [44, 54, 45] were used.

[4] The implementation published by the authors was used.

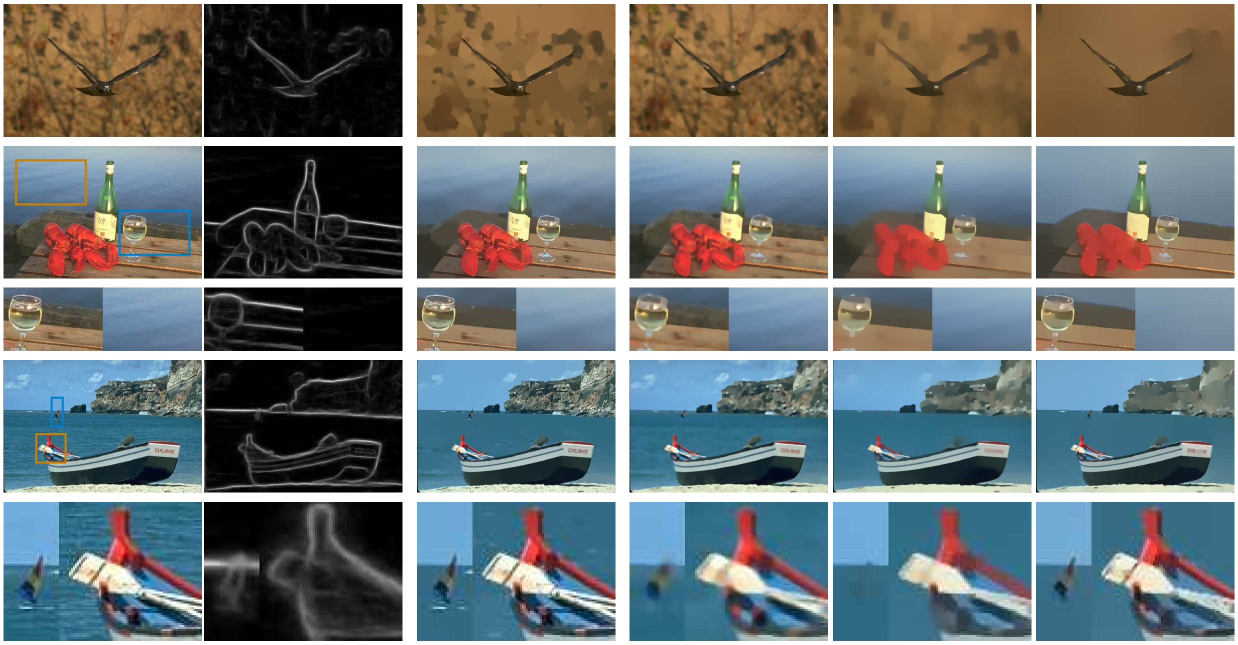


Figure 7. Visual comparison with the state-of-the-art structure-preserving filters under constant/default parameter setting. Unlike the state-of-the-art filters, the proposed filter is more robust to various object scales due to adoption of learning-based edge detection technique[13].

4. Conclusion

An efficient structure-preserving filtering technique is proposed in the paper. Unlike the current state-of-the-art structure-preserving filters that use low-level vision features to design the filter kernel, the proposed technique is developed based on high-level understanding of the image structures. It is a seamless combination of the fast anisotropic filtering technique(s) with the structure learning based edge detector. The use of edge models trained from human-labelled data sets enables the proposed filter to better preserve the structure edges that can be detected by the human visual system. As a result, it is more robust to objects/structures with different sizes/scales.

The proposed technique cannot be directly applied to other edge-preserving filters like the guided filter [23] and most of the quantization-based fast bilateral filters [15, 37, 9, 29, 50, 2, 1, 21]. Fig.9(a)-(b) show that the guided filter is vulnerable to textures when a constant filter kernel is used. A simple extension is adjusting the edge confidence to adaptively control the guided filter kernel so that small kernel will be used around edges. Fig.9(c)-(d) present the edge confidence before and after minimum filtering, and Fig.9(e) presents the guided filtered image using an adaptive kernel based on the edge confidence in (d). It outperforms the original guided filter for suppressing textures while capable of maintaining the most salient structure edges. However, the quality is obviously lower than the integration of an anisotropic filter as shown in Fig.9(f). A better extension for other edge-preserving filters will be investigated in the near future.

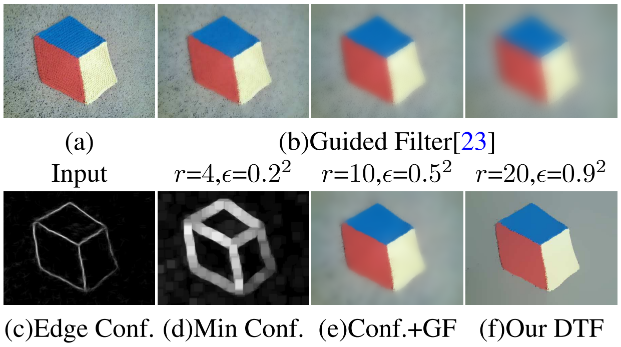


Figure 9. Limitation. The guided filter is not effective for removing textures as shown in (b). The proposed technique can be adjusted for the integration with guided filter but the quality will be lower than anisotropic filters.

References

[1] A. Adams, J. Baek, and A. Davis. Fast high-dimensional filtering using the permutohedral lattice. *Comput. Graph. Forum*, 29(2):753–762, 2010.

[2] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph.*, 28:21:1–21:12, July 2009.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33:898–916, 2011.

[4] R. ARNHEIM. *Art and Visual Perception: A Psychology of the Creative Eye*. University of California Press, 1956.

[5] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand. Fast local laplacian filters: Theory and applications. *ToG*, 33(5), 2014.

[6] J. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *IJCV*, 67(1):111–136, 2006.

[7] J. Canny. A computational approach to edge detection. *PAMI*, 1986.

[8] B. Catanzaro, B.-Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer. Efficient, high-quality image contour detection. In *ICCV*, 2009.

[9] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. In *Siggraph*, volume 26, 2007.

[10] A. Criminisi, T. Sharp, C. Rother, and P. Perez. Geodesic image and video editing. *ToG*, 29(5), 2010.

[11] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006.

[12] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[13] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[14] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[15] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *Siggraph*, volume 21, 2002.

[16] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *Siggraph*, 23(3):673–678, 2004.

[17] Z. Farbman, R. Fattal, and D. Lischinski. Diffusion maps for edge-aware image editing. *ToF*, 29(6):145:1–145:10, 2010.

[18] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ToG*, 27(3), 2008.

[19] R. Fattal. Edge-avoiding wavelets and their applications. *ToG*, 28(3):1–10, 2009.

[20] E. Gastal and M. Oliveira. Domain transform for edge-aware image and video processing. *TOG*, 30(4):69:1–69:12, 2011.

[21] E. Gastal and M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *TOG*, 31(4):33:1–33:13, 2012.

[22] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013.

[23] K. He, J. Sun, and X. Tang. Guided image filtering. *PAMI*, 35:1397–1409, 2013.

[24] L. Karacan, E. Erdem, and A. Erdem. Structure-preserving image smoothing via region covariances. *ToG*, 32(6):176:1–176:11, 2013.

[25] J. J. Kivinen, C. K. Williams, and N. Heess. Visual boundary prediction: A deep neural prediction network and quality dissection. In *AISTATS*, 2014.

[26] J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013.

[27] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps. In *CVPR*, 2014.

[28] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, Boston, MA, USA, 2001.

[29] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, 81:24–52, January 2009.

[30] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–73, 2009.

[31] S. Parisand, S. W. Hasinoff, and J. Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ToG*, 30(4):68:1–68:12, 2011.

[32] S. Parisand, S. W. Hasinoff, and J. Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *Communications of the ACM*, 58(3), 2015.

[33] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[34] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12:629–639, 1990.

[35] S. Perreault and P. Hbert. Median filtering in constant time. *TIP*, 16(9):2389–2394, 2007.

[36] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *Siggraph*, 23(3):664–672, 2004.

[37] T. Q. Pham and L. J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *International Conference on Multimedia and Expo*, 2005.

[38] F. Porikli. Constant time o(1) bilateral filtering. In *CVPR*, 2008.

[39] X. Ren and B. Liefeng. Discriminatively trained sparse code gradients for contour detection. In *NIPS*, 2012.

[40] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[41] K. Subr, C. Soler, and F. Durand. Edge-preserving multiscale image decomposition based on local extrema. *ToG*, 28(5), 2009.

[42] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[43] B. Weiss. Fast median and bilateral filtering. In *Siggraph*, volume 25, pages 519–526, 2006.

[44] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2011.

[45] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via natural variation measure. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2012.

[46] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013.

[47] Q. Yang. Recursive bilateral filtering. In *ECCV*, pages 399–413, 2012.

[48] Q. Yang. Recursive approximation of the bilateral filter. *IEEE Transactions on Image Processing*, 24(6):1919–1927, 2015.

[49] Q. Yang, N. Ahuja, and K.-H. Tan. Constant time median and bilateral filtering. *International Journal of Computer Vision*, 112(3):307–318, 2015.

[50] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time o(1) bilateral filtering. In *CVPR*, pages 557–564, 2009.

[51] Q. Yang, S. Wang, and N. Ahuja. Svm for edge-preserving filtering. In *CVPR*, pages 1775–1782, 2010.

[52] W. Yin, D. Goldfarb, and S. Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In *VLSM*, pages 73–84, 2005.

[53] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, 2015.

[54] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *ECCV*, 2014.

[55] Q. Zhang, L. Xu, and J. Jia. 100+ times fasterweighted median filter (wmf). In *CVPR*, 2014.

[56] S. Zheng, Z. Tu, and A. Yuille. Detecting object boundaries using low-,mid-, and high-level information. In *CVPR*, 2007.

[57] D. Ziou and S. Tabbone. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis*, 8:537C559, 1998.

[58] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

[59] C. L. Zitnick and D. Parikh. The role of image understanding in contour detection. In *CVPR*, 2012.