CrossMark

Edge-aware smoothing through adaptive interpolation

Mukhalad Al-nasrawi[1,2] · Guang Deng[1] · Ba Thai[1]

Abstract The goal of edge-aware filtering is to smooth out small-scale structures while preserving large object boundaries. A fundamental idea to design such filters is to avoid smoothing across strong edges. In this paper, we explore a new approach which iteratively adds the edge information back to a smoothed image. We study the smoothed image as the starting point of the iteration and the optimal stopping criterion. We demonstrate that in a wide range of applications the proposed technique can produce competitive results as those of state-of-the-art edge-aware filters. In particular, the proposed algorithm has the best performance in texture smoothing.

Keywords Edge-aware smoothing · Adaptive interpolation · Texture smoothing · Detail enhancement · Content-aware resizing

1 Introduction

Edge-preserving filtering has been extensively studied in computer graphics and image processing to suppress noise from images and/or extract important image contents [17,

22]. Edge-preserving filters are widely deployed in many applications, such as image matting and haze removal [9], image colorization [10], HDR tone mapping [7], details manipulation [6], and image super resolution [27–29]. Some well-known algorithms include: adaptive smoothing [20], anisotropic diffusion [18], total variation [21], robust statistics [2], bilateral filter [25], nonlocal mean filter [3], Ł$_0$-minimization-based filter [30], BMA filters [5], weighted least-square-based filter [6], SVM-based filter [31], geodesic filter [7], bi-exponential filter [24], and guided filter [9]. The relationship between recent image smoothing algorithms is studied in [14].

From the above brief review of current state-of-the-art edge-aware filtering algorithms, it can bee seen that most of them are based on the idea of smoothing out unwanted detail information by avoiding smoothing across significant edges. Motivated by the current development and the successful applications of edge-aware filters, the aim of this work is to explore new ways for the development of filters which will lead to improved performance in terms of smoothing texture and preserving edges. In the following, we briefly review the basic ideas of two algorithms which are directly related to our work: the rolling guidance filter and unsharp masking.

The rolling guidance filter (RGF) [32] is a scale-aware filter [13] which involves two steps. In the first step, small structures are completely removed by a Gaussian low-pass filter. In the second step, a joint bilateral filter is adopted to restore the large image structures iteratively. Recently, the smooth and iterative restore (SIR) filter [12], which follows a similar idea of the RGF, is used to iteratively restore strong edges from the smoothed image.

Unsharp masking [8] is a classical technique which is used to improve the sharpness of an image. The basic idea is to smooth the image and extract the edge information. The enhanced image is produced by adding the edge informa-

✉ Mukhalad Al-nasrawi
  m.mukhalad@latrobe.edu.au

  Guang Deng
  d.deng@latrobe.edu.au

  Ba Thai
  t.thai@latrobe.edu.au

1 Department of Electronic Engineering, La Trobe University, Bundoora, VIC 3086, Australia

2 Al-Musaib Technical College, Al-Furat Al-Awsat Technical University, Babylon 51009, Iraq

tion to the smoothed image. Edge-aware smoothing filters [4,5,9,19] have been used to produce the smoothed image.

This work is inspired by the RGF and unsharp masking. The basic idea of this work is that the edge-aware smoothing can be achieved by iteratively adding significant edges to a smoothed image. A fundamental difference between this work and those based on the RGF is that in this work the edges are added back through a linear process similar to that of unsharp masking, while in the RGF the edges are restored through the guided edge-aware filters such as the bilateral filter. As such, the proposed algorithm is computationally more efficient than the RGF. In addition, by iteratively adding the edges back, the resulting image can be regarded as an adaptive interpolation between the smoothed image and the original image.

The rest of the paper is organized as follows. In Sect. 2, we describe the proposed edge-preserving filter, including the basic iterative algorithm and its convergence, the interpretations of the filter from geometric and nonlinear interpolation points of view, and a discussion of the stopping criterion in terms of a convex optimization problem. In Sect. 3, through experimental results and comparison with other algorithms, we demonstrate that the proposed algorithm is an efficient and effective tool for many applications including detail enhancement, texture smoothing, and seam carving based resizing. Conclusions are presented in Sect. 4.

2 The proposed algorithm

2.1 The basic algorithm

At pixel location n, let $I(n)$ and $Y_k(n)$ be the original image and the processed image at the kth iteration. The proposed algorithm can be written as

$$Y_{k+1}(n) = Y_k(n) + \phi(D_k(n))D_k(n) \tag{1}$$

where $D_k(n) = I(n) - Y_k(n)$ and $\phi(x)$ is a nonlinear function. Since the proposed algorithm uses pixel-wise operations, to simplify notation, the dependence on the pixel location n is not explicitly expressed in the following discussion. For example, we will use Y_k, I and D_k to replace $Y_k(n)$, $I(n)$ and $D_k(n)$.

Intuitively, it is expected that in the smooth areas of the image, the value of D_k is small. There is little edge information to add to the smoothed image. As such, a natural requirement is that $Y_{k+1} \approx Y_k$, which requires the function $\phi(x)$ to have the property $\lim_{|x| \to 0} \phi(x) = 0$. On the other hand, around the edge, the value of D_k is expected to be large due to the smoothing effect. Edge information must be added to the smoothed image. In an extreme case, it is desirable to have $Y_k = I$ such that the edge information is fully

restored. As such, the function $\phi(x)$ should have the property $\phi(x) = 1$ when $|x|$ is sufficiently large.

With these considerations, we can define $\phi(x)$ as

$$\phi(x) = 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{2}$$

where σ is a scale parameter which controls how fast the function is increased to 1 or decreased to 0. For example, the value of $|D_k|$ is said to be sufficiently large when $|D_k| \geq 3\sigma$. In this case, we have $\phi(D_k) = 0.99$ which strongly restores the edge information.

2.2 Adaptive interpolation

2.2.1 Interpolation

The proposed algorithm can be interpreted as an adaptive interpolation between the initial smoothed image Y_0 and the original image. Indeed, we can rewrite Eq. (1) as a weighted average of Y_k and I

$$Y_{k+1} = (1 - \phi(D_k))Y_k + \phi(D_k)I \tag{3}$$

To examine the interpolation effects, let us denote the smoothed image $Y_0 = S$ which is produced by an image smoothing algorithm. We can then determine the output of the proposed algorithm in terms of S and I as follows

$$Y_{k+1} = \Phi_k S + (1 - \Phi_k)I \tag{4}$$

where

$$\Phi_k = \prod_{i=1}^{k}(1 - \phi(I - Y_i)) = \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{k} D_i^2\right) \tag{5}$$

We now prove that the above iterative algorithm converges to the original image I. Since $\sum_{i=1}^{k} D_i^2 \geq \sum_{i=1}^{k-1} D_i^2$, we have

$$\begin{cases} \Phi_k < \Phi_{k-1}, & Y_k \neq I \tag{6} \\ \Phi_k = \Phi_{k-1}, & Y_k = I \tag{7} \end{cases}$$

We define the absolute difference

$$\epsilon_k = |Y_{k+1} - I| = \Phi_k|S - I| \tag{8}$$

Using (6), we have $0 \leq \epsilon_k < \epsilon_{k-1}$ which means ϵ_k is a decreasing function of k. As such, the output is getting closer to the original image as the number of iteration increases. On the other hand, using (7), we have $\epsilon_k = \epsilon_{k-1}$ which implies $Y_{k+1} = Y_k = I$. There is no further change in the output in successive iterations. In fact, using (4), we can clearly

Fig. 1 The convergence of the proposed algorithm as iteration increases from the initial smooth image (S) (k = 0) to the image at iteration (k = 4). The mean squared difference $\delta(k)$ is also shown

see that in this case $\epsilon_k = 0$ which means $\Phi_k = 0$. Therefore, the output of the iterative algorithm will converge to the original image I. An example is shown in Fig. 1 which demonstrates the convergence of the algorithm measured by the mean squared difference defined as:

$$\delta(k) = \frac{1}{N} \sum_{n=1}^{n=N} |Y_k(n) - I(n)|^2 \tag{9}$$

2.2.2 A geometrical interpretation

If we regard I and S as two points in the N-dimensional space, where N is the number of pixels, then the interpolation results are a sequence of points $\{Y_k\}$. Initially Y_k (k is small) is close to S. As the iteration progresses, Y_k is getting close to I. Let us imagine that there is a curve starting at point S and ending at point I. The curve links all points $\{Y_k\}$. The desirable edge-aware filtering result is then a point in this curve. Therefore, there are two factors that make critical influence to the filtering result. One is the starting point S, and the other is the stopping criterion.

2.3 Stopping criterion and starting point selection

2.3.1 Stopping criterion

To study the stopping criterion, we recall an early work [15] on edge-aware filtering which is based on the following objective function

$$J(k) = \sum_{n=1}^{N} D_k(n) + \lambda \sum_{n=1}^{N} \varphi\left(h_k(n) + v_k(n)\right) \tag{10}$$

where $h(n)$ and $v(n)$ are the absolute value of the horizontal and vertical gradient of the image at pixel $Y_k(n)$, and $\varphi(x)$ is defined as $\varphi(x) = |x|^\alpha$, $0 < \alpha < 1$. The objective function imposes two requirements for the desirable edge-aware fil-
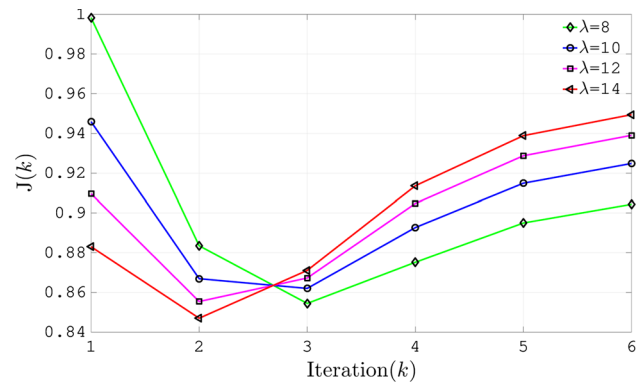
Fig. 2 Cost function $J(k)$ averaged over the 4 test images under the test condition $\alpha = 0.5$ and different settings of λ

tering result: (1) It must be close to the original image, and (2) it must be smooth except at locations of sharp edges. We can numerically determine k such that this objective function is minimized for a fixed regularization parameter λ.

Since the above objective function has led to good edge-aware filtering results, a natural way to develop a stopping criterion for the proposed algorithm is to evaluate the objective function at each iteration Y_k. The output image at the kth iteration which results in the smallest cost should be used as the stopping criterion. We have conducted experiments by applying the proposed algorithm to 4 test images. The cost function Eq. (10) is calculated for the output of each iteration. Figure 2 shows the average costs $J(k)$ over 4 test images which are "Flower," "Flautist," "House corner," and "Tulips." To simplify our experiment, we fix $\alpha = 0.5$ and calculate the cost function for a range of values of k (the number of iterations) and λ. Results are shown in Fig. 2 which are obtained by using the four tested images. From these results, we can clearly see that the minimum cost is reached when $k = 2$ or 3. Therefore, we can choose $k = 2$ or 3 as the stopping criterion. We also remark that both parameters α and λ are not required in the proposed algorithm. They are used in the cost function which is used to experimentally determine the stopping criterion.

Fig. 3 *First row* the initial smoothed images using Gaussian blur ($\sigma_o = 4$), (9×9) mean filter and (9×9) median filter, respectively. *Second row* the proposed filtering results from the respective three initial smoothed images. The filter parameters are: $\sigma = 0.4$ and $k = 2$. The artifacts due to the Gaussian and mean filters are pointed out by *red arrows* (color figure online)

2.3.2 Starting point selection

The initial smoothed image (starting point S) can be selected based on the following experiments. We have used simple smoothing filters: Gaussian low-pass filter, the mean filter, and the median filter to remove small-scale structure. We tune the parameters of these three filters to produce similar smoothing effects. Results are shown in Fig. 3. We can make the following observations. In the smooth areas of the image, results of the proposed filter that used these three filters as the starting point are almost the same. However, in areas with sharp edges, the results of the proposed filter associated with the mean filter and Gaussian filter have ringing phenomena around the edges, while the result associated with the median filter is almost free of such artifacts. Therefore, the median filter is adopted to produce the initial smoothed image in the proposed filter. The neighborhood size of median filter is defined as (r).

3 Results

3.1 Experimental study of the proposed algorithm

3.1.1 Parameters settings and smoothing results

There are two parameters in the proposed filter. They are the scale parameter (σ) for the nonlinear function $\phi(x)$ and

the size (r) of the neighborhood for the median filter. The smoothing performance of the proposed filter has been tested using the "Flower" image shown in Fig. 4 by setting different values of these two parameters. We can see that setting a large value of r will smooth out more details, while setting a smaller value of σ will lead to preserving textures such as those on the flower leaves.

3.1.2 Running time

Because the proposed filter only requires pixel-wise computation, it has competitively low running time without software and hardware optimization. The complexity of the proposed filter is $\mathcal{O}(r \log(r) N + kN)$.[1] We compare the running time of the proposed filter with some of the most computationally efficient edge-aware filters. Two images (Flautist and Flower) are used in the experiment which is aimed at smoothing the texture. All results are produced by averaging the running time of 10-run of each filter. All filters are implemented in MATLAB which is run in a PC with 16 GB memory and an Intel-i7 processor running at 3.4 GHz. Results are presented in Table 1 which shows that the running time of the proposed algorithm is faster than the domain transform filter (DTF) [7], the rolling guidance filter (RGF) [32] and the bilateral grid (BG) implementation of the bilateral filter [16], but is slower than the guided filter (GF) [9].

3.2 Applications and comparisons

In this section, we present applications of the proposed algorithm and compare the results with some state-of-the-art filters including guided image filter (GF) [9], rolling guidance filter (RGF) [32], region covariance filter (RCF) [11], Bayesian model averaging (BMA) [5] and semi-guided bilateral filter (SGBF) [23].

3.2.1 Image noise removal

To study the performance of different edge-preserving filters in the presence of noise, we artificially create an image which has structures of different sizes (i.e., small, medium, and large). This image is used as a reference image. We then add a uniform noise between -10 and 10 to the reference image to generate a noisy image. One row of the reference image and the noisy image as well as the filtering results are shown in Fig. 5. We adjust parameters of each filter such that the largest value of PSNR is achieved. The quantitative performance of noise removal is measured by the peak-signal-to-noise-ratio (PSNR) which is shown in Table 2. We can clearly see

---

[1] One of the reviewers of this paper pointed out this result.
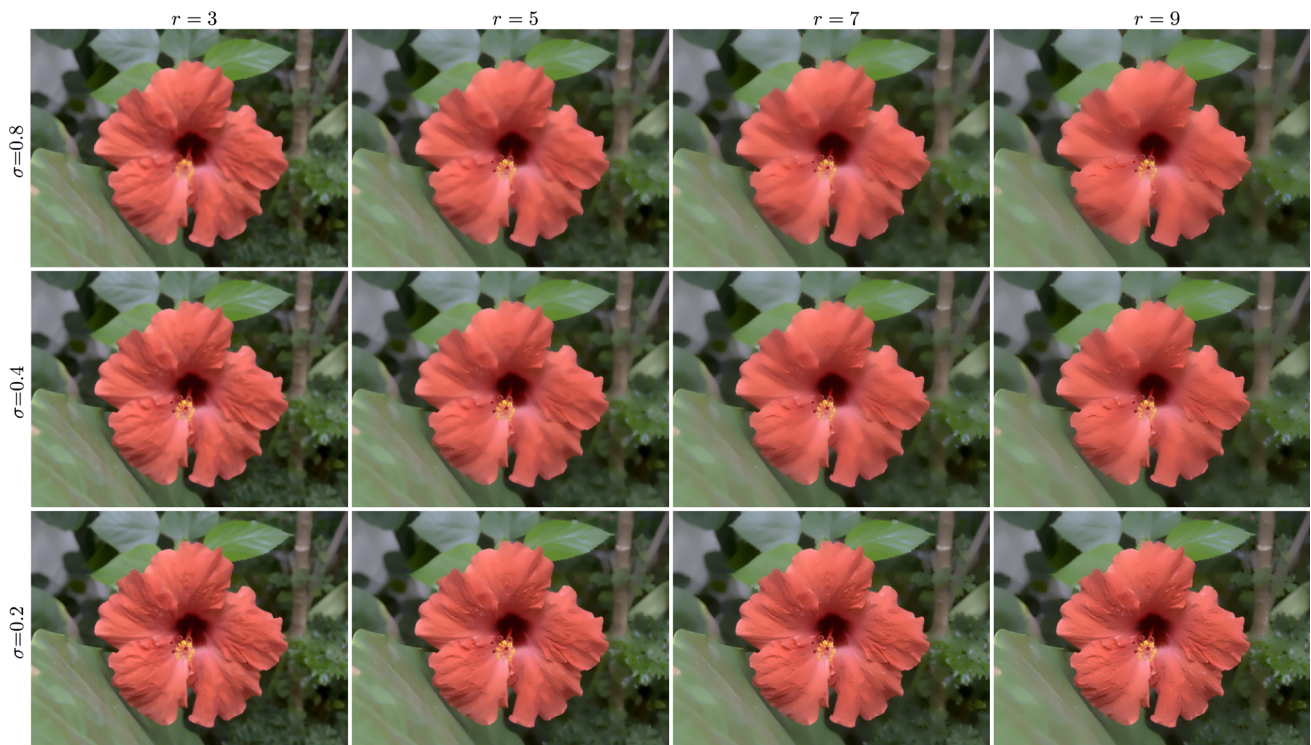
Fig. 4 Effects of different parameters settings on smoothing results of the proposed filter

Table 1 Running time (s) of experiments on "Flower" shown in Fig. 1 and "Flautist" shown in Fig. 7

	GF	RGF	DTF	BG	Proposed
Flautist	0.014	1.175	0.072	0.049	0.040
Flower	0.060	1.567	0.181	0.174	0.078

The results of all algorithms are tuned such that the results are of about the same PSNR (27.8 dB for the "Flautist" image and 37.6 dB for the "Flower" image)
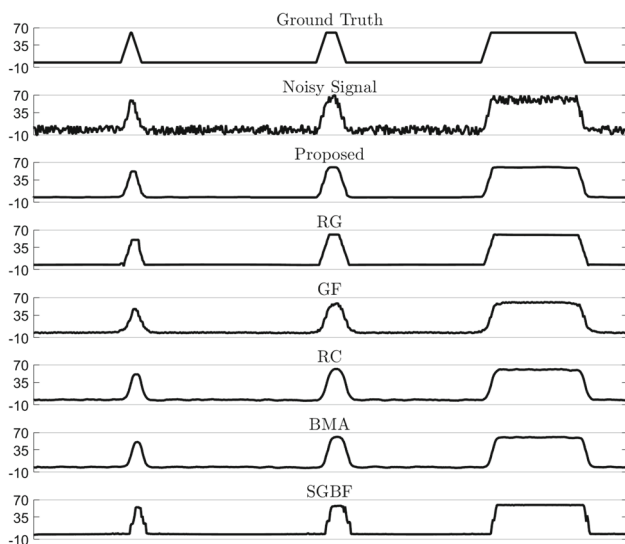
Fig. 5 A comparison of results in noise removal

Table 2 The PSNR (dB) for results in Fig. 5

GF	RGF	RCF	BMA	SGBF	Proposed
28.79	35.41	34.03	34.90	28.54	35.92

that while RCF, GF, BMA, and SGBF achieve PSNR below 35 dB, the proposed filter and RGF achieve PSNR greater than 35 dB. Visually, the filters (RCF, GF, and BMA) do not only smooth out the noise but also blur large edges. The SGBF produces results with jagged edges. Both the RGF and the proposed filter produce similar results which are relatively close to the ground truth signal.

An interesting question is: what would be the performance of the proposed filter if the nonlinear function $\phi(x)$ defined in (2) is replaced by other functions? To study this problem, we recognize that Eq. (2) is indeed the so-called Welsch function which is a cost function typically used in robust estimation [33]. Therefore, we investigate the performance of the proposed filter by using other functions used in robust estimation. These functions are listed in Table 3. We apply the proposed algorithm with different choices of $\phi(x)$ to process the noisy image used in the above experiment. The results are measured by PSNR which are also listed in Table 3. We can clearly see that the proposed function (the Welsch function) leads to the highest PSNR.

**Table 3** Robust cost functions and the achieved peak-signal-to-noise-ratio (PSNR) in the proposed filter

Type	$\phi(x)$	PSNR (dB)
Proposed	$1 - \exp(-(x^2/2c^2))$	35.92
L_1	$|x|$	29.82
$L_1 - L2$	$2(\sqrt{1 + x^2/2} - 1)$	30.29
L_p	$|x|^v/v$	35.24
Fair	$c^2[|x|/c - \log(1 + |x|/c)]$	32.83
Cauchy	$\frac{c^2}{2}\log(1 + (x/c)^2$	27.13
German-McClure	$\frac{x^2/2}{1+x^2}$	23.76
Tukey	$\begin{cases} \frac{c^2}{6}(1 - [1 - (x/c)^2]^3) \\ (c^2/6) \end{cases}$	35.00

3.2.2 Detail enhancement

An unsharp masking algorithm is used in the detail enhancement. Figure 6 shows the results for the test image "Tulips." We can see that the proposed filter produces a similar result as those of the GF. It can also be observed that results of GF and the proposed filter are free of gradient reversal and ringing artifacts. However, RGF produces reversal artifacts.
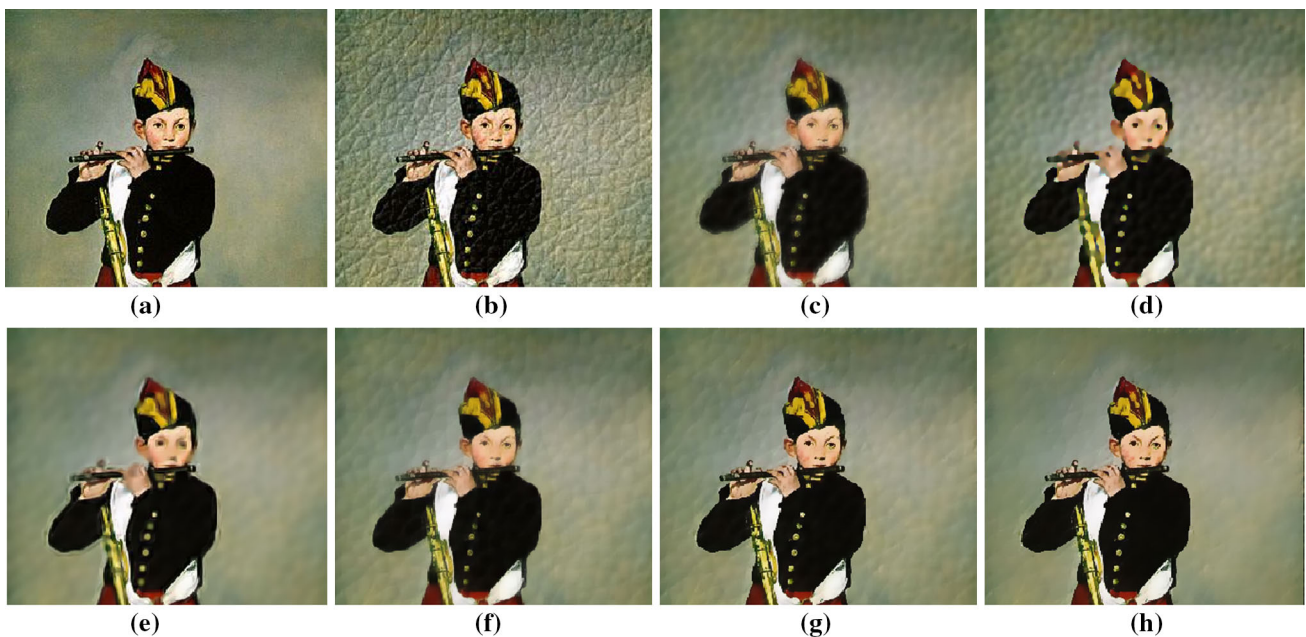
3.2.3 Texture smoothing

In this section, we demonstrate that the proposed filter is an effective tool for smoothing out texture information while preserving large objects and sharp edges.
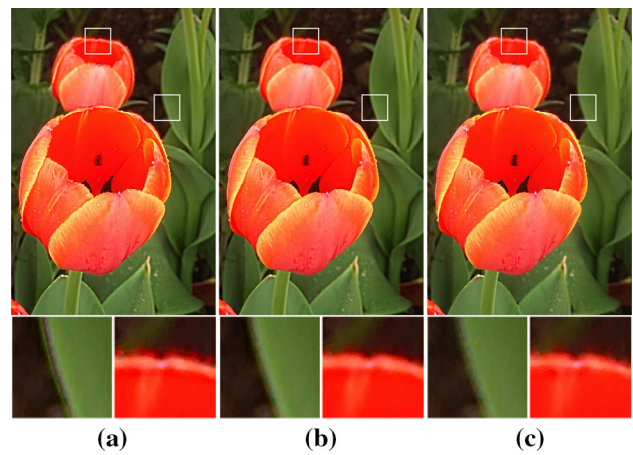
Fig. 6 Detail enhancement results. The parameters are $\sigma_s = 7$, $\sigma_r = 0.1$ for RGF, $r = 7$, $\epsilon = 0.1^2$ for GF, and $r = 7$, $\sigma = 0.1$ for the proposed filter. The detail is boosted by 4 times. **a** RGF. **b** GF. **c** Our results

To perform a quantitative evaluation of the smoothing performance, we add texture to a texture-free image which is used as the ground truth. We adjust parameters of each filter such that it preserves the overall image structure and eliminates the texture component as much as possible. The quality of the smoothing result is measured by the structure similarity index (SSIM) [26].

Results shown in Fig. 7 clearly indicate that the proposed filter effectively smooths out the texture while preserving most of the important information of the original image. In comparison, other filters (GF, RGF, BMA, and SGBF) still


Fig. 7 Texture smoothing results. All filters are adjusted for their best performance in terms of achieving the best SSIM. **a** Ground truth image. **b** Ground truth + texture. **c** GF ($r = 5$, $\epsilon = 0.05$). **d** RGF ($\sigma_s = 2.1$, $\sigma_r = 0.1$). **e** RCF ($\sigma = 0.05$, $k = 3$). **f** BMA ($r = 4$, $\epsilon = 0.002$). **g** SGBF ($\sigma_s = 2.1$, $\sigma_r = 0.1$). **h** ($r = 4$, $\sigma = 0.4$)

Table 4 The objective evaluation of the results using SSIM and PSNR for images in Fig. 7

	GF	RGF	RCF	BMA	SGBF	Proposed
SSIM	0.922	0.917	0.934	0.934	0.933	0.936
PSNR (dB)	26.83	27	27.74	27.65	26.79	27.18
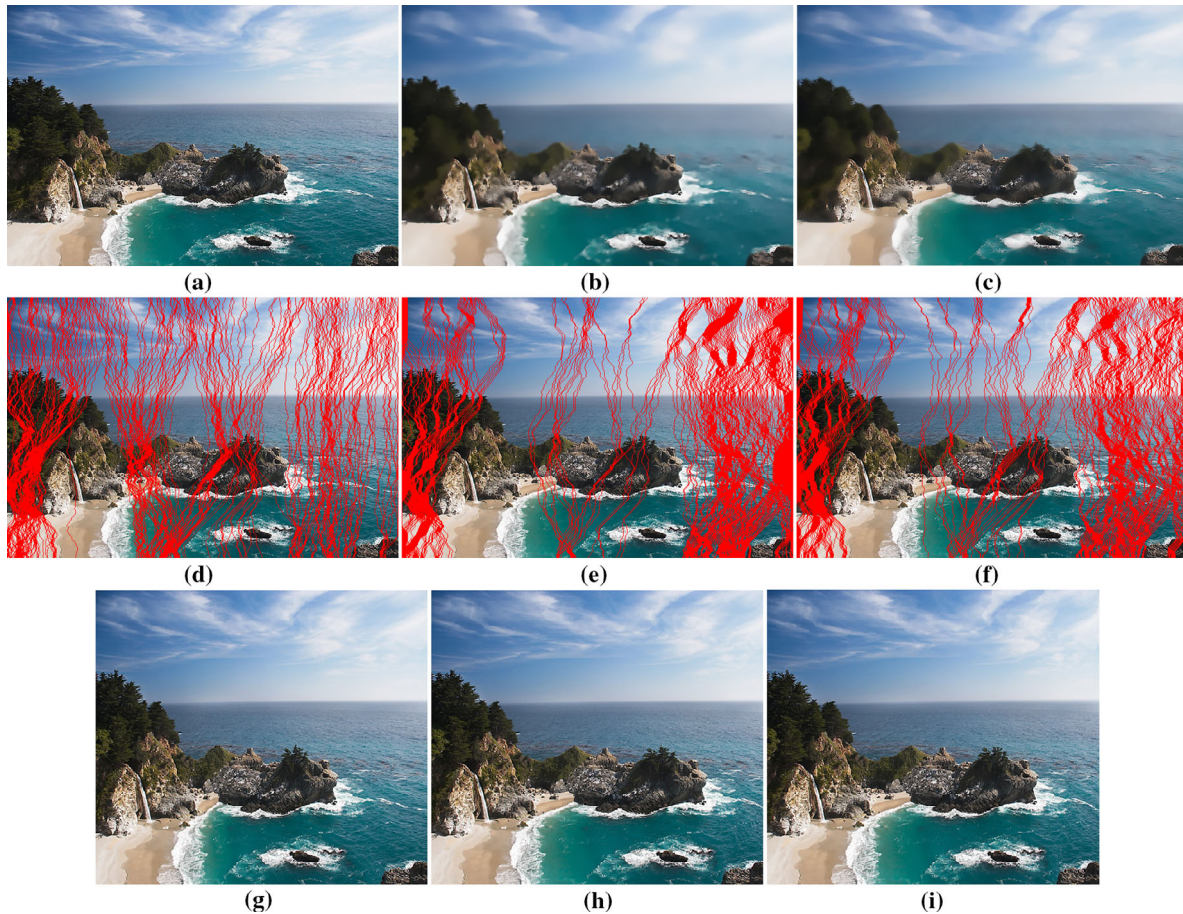
Fig. 8 Seam carving in a natural science. *Left column* original method. *Middle column* results of [11]. *Right column* results of the proposed filter. *Rows* from *top* to *bottom*: original image, smoothed images, seams to be eliminated and the seam carving results

preserve texture to certain degree because these filters treat texture as edges. On the other hand, the RCF is successful in texture removal at the cost of blurring object boundaries. The subjective evaluation of the proposed filter is confirmed by the largest SSIM associated with result produced by the proposed method. However, the PSNR results are mixed. The PSNR value of the image produced by our filter is among the best as shown in Table 4.

3.2.4 Content-aware image resizing

Content-aware image resizing was introduced in [1]. The importance of the pixels has been calculated based on gradient energy function. In a natural scene, details such as grass, waves, sand, tree are less important compared to objects of interest. It can be observed in Fig. 8d that the vertical seams pass through the tree on the left and middle, whereas it crosses waves on the right, making the results in Fig. 8g unacceptable. To address this problem, an edge-aware filter is used to smooth the image and the result is used to calculate the seam. We compare the performance of the proposed filter with the structure–texture decomposition approach in [11]. Results shown in Fig. 8 show that the performance of the two approaches is indeed similar. The advantage of the proposed filter is that it runs 25-times faster than the region covariances filter [11].

4 Conclusion

In this paper, a new technique for edge-aware image smoothing is introduced. The proposed technique is based on the idea

of iteratively adding the edge information to the smoothed image. This is different from most existing edge-aware techniques in which the aim is to perform the smoothing operation in such a way to avoid smoothing across edges. A detailed study of the proposed technique is presented, which includes the choice of the smoothed image as the starting point of the iteration and the optimal stopping criterion. Experimental results and comparisons have demonstrated that in a wide range of applications the proposed technique can produce competitive smoothing results as those of state-of-the-art edge-aware filters. In particular, the proposed filter has the best performance in texture smoothing.

References

1. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Trans. Graph. **26**(3), 10:1–10:9 (2007)
2. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. Int. J. Comput. Vis. **19**(1), 57–91 (1996)
3. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. SIAM Multiscale Model. Simul. **4**(2), 490–530 (2005)
4. Deng, G.: A generalized unsharp masking algorithm. IEEE Trans. Image Process. **20**(5), 1249–1261 (2011)
5. Deng, G.: Edge-aware BMA filters. IEEE Trans. Image Process. **25**(1), 439–454 (2016)
6. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Trans. Graph. **27**(3), 67 (2008)
7. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Graph. **30**(4), 69:1–69:12 (2011)
8. Gonzalez, R., Woods, R.: Digital Image Processing. Pearson/Prentice Hall, London/Englewood Cliffs (2008)
9. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1397–1409 (2013)
10. Hussein, A.A., Yang, X.: Colorization using edge-preserving smoothing filter. Signal Image Video Process. **8**(8), 1681–1689 (2014)
11. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Trans. Graph. **32**(6), 176:1–176:11 (2013)
12. Kniefacz, P., Kropatsch, W.G.: Smooth and iteratively restore: a simple and fast edge-preserving smoothing model. In: The 39th Annual Workshop of the Austrian Association for Pattern Recognition(OAGM), pp. 1–9. eprint arXiv:1505.06702 (2015)
13. Lindeberg, T.: Scale-space theory: a basic tool for analyzing structures at different scales. J. Appl. Stat. **21**(1–2), 225–270 (1994)
14. Milanfar, P.: Symmetrizing smoothing filters. SIAM J. Imaging Sci. **6**(1), 263–284 (2013)
15. Nikolova, M., Ng, M.K., Zhang, S., Ching, W.K.: Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization. SIAM J. Imaging Sci. **1**(1), 2–25 (2008)
16. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. Int. J. Comput. Vis. **81**(1), 24–52 (2009)
17. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: theory and applications. Found. Trends Comput. Graph. Vis. **4**(1), 1–73 (2009)
18. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
19. Polesel, A., Ramponi, G., Mathews, V.J.: Image enhancement via adaptive unsharp masking. IEEE Trans. Image Process. **9**(3), 505–510 (2000)
20. Saint-Marc, P., Chen, J.S., Medioni, G.: Adaptive smoothing: a general tool for early vision. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 514–529 (1991)
21. Strong, D., Chan, T.: Edge-preserving and scale-dependent properties of total variation regularization. Inverse Probl. **19**(6), S165 (2003)
22. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer, New York (2010)
23. Thai, B., Al-nasrawi, M., Deng, G., Su, Z.: The semi-guided bilateral filter. IET Image Process. **11**(7), 512–521 (2017)
24. Thevenaz, P., Sage, D., Unser, M.: Bi-exponential edge-preserving smoother. IEEE Trans. Image Process. **21**(9), 3924–3936 (2012)
25. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: 6th International Conference on Computer Vision, pp. 839–846. IEEE Computer Society, Washington, DC (1998)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
27. Wei, D.: Image super-resolution reconstruction using the high-order derivative interpolation associated with fractional filter functions. IET Signal Process. **10**(9), 1052–1061 (2016)
28. Wei, D., Li, Y.: Reconstruction of multidimensional bandlimited signals from multichannel samples in linear canonical transform domain. IET Signal Process. **8**(6), 647–657 (2014)
29. Wei, D., Li, Y.M.: Generalized sampling expansions with multiple sampling rates for lowpass and bandpass signals in the fractional Fourier transform domain. IEEE Trans. Signal Process. **64**(18), 4861–4874 (2016)
30. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L_0 gradient minimization. ACM Trans. Graph. **30**(6), 174:1–174:12 (2011)
31. Yang, Q., Wang, S., Ahuja, N.: SVM for edge-preserving filtering. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1775–1782 (2010)
32. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: European Conference on Computer Vision, pp. 815–830. Springer (2014)
33. Zhang, Z.: Parameter estimation techniques: a tutorial with application to conic fitting. Image Vis. Comput. **15**(1), 59–76 (1997)