ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/330796722

Modified iterative guided texture filtering algorithm

Article *in* Computers & Graphics · January 2019

DOI: 10.1016/j.cag.2018.12.008

CITATIONS
0

READS
21

2 authors:

Mukhalad Al-nasrawi
La Trobe University
5 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Guang Deng
La Trobe University
129 PUBLICATIONS **1,174** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Fingerprint template protection View project

Multi-Focus Image Fusion View project

All content following this page was uploaded by Mukhalad Al-nasrawi on 13 February 2019.

The user has requested enhancement of the downloaded file.

Computers & Graphics 79 (2019) 81–100

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Modified iterative guided texture filtering algorithm ☆

Mukhalad Al-nasrawi [a,b,*], Guang Deng [a]

[a] Department of Engineering, La Trobe University, Bundoora, Victoria 3086, Australia
[b] Al-Musaib Technical College, Al-Furat Al-Awsat Technical University, Babylon 51009, Iraq

ARTICLE INFO

Article history:
Received 8 July 2018
Revised 16 December 2018
Accepted 23 December 2018
Available online 31 January 2019

Keywords:
Structure-texture decomposition
Guided edge-preserving filters
Iterative smoothing
Guidance image
Content-aware image resizing
Image abstraction

ABSTRACT

Structure-texture decomposition smoothing has been extensively studied due to its wide range of applications in computational photography and image processing. In this paper, we propose a new structure-texture decomposition algorithm which is based on two fundamental ideas: (1) guidance image and (2) iterative smoothing. The guidance image is generated by mitigating high-frequency oscillatory components in the original image. The result is then incorporated in a new generic iterative framework which makes use of well-known guided edge-reserving filters such as bilateral filter (BF), guided filter (GF), domain transform filter (DTF), and the extended Bayesian model averaging filter (BMA) called guided Bayesian model averaging filter (GBMA) to achieve texture smoothing. We have presented a detailed study of the proposed algorithm including: guidance image generation, an evaluation of the guided edge-preserving filters which are incorporated in the proposed iterative framework, the number of iterations for the proposed iterative structure, and the selection of guided edge-preserving filter. We demonstrate that the proposed method is a flexible and effective tool for a wide range of image editing applications including: image abstraction, color pencil drawing, content-aware image resizing, and texture editing. In particular, the proposed approach has the best performance in structure-texture decomposition for an image with low-contrast features.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Natural scenes contain objects of various patterns on a broad spatial scales. Small-scale components typically represent texture/noise, while large-scale components generally represent salient objects/distinctive boundaries. All the aforementioned features are very important for human perception. In machine vision, extracting meaningful salient objects from a highly correlated background is an essential task. Using a structure-texture decomposition filter, an image can be separated into two independent components: $f = u + v$, where f, u and v represent the input image, structure components and texture components, respectively.

Similar to structure-texture decomposition filters, edge-preserving filters can also be used as an image decomposition tool. The basic idea of utilizing an edge-preserving filter is to separate an image into two layers, the base layer (prominent objects) and the detail layer (small-scale details). There are many edge-preserving filters including: bilateral filter [1], guided filter [2], L_0-smoothing filter [3], bi-exponential filter [4], BMA filter [5], static-dynamic guidance filter [6], semi-guided bilateral filter [7] and adaptive interpolation filter [8]. These filters tend to smooth out small details while maintaining distinctive structures. However, they are not explicitly designed to address the structure-texture decomposition problem. An edge-preserving filter may regard texture information as strong edges which are preserved. This leads to texture preservation and unsatisfactory results may be obtained.

To solve this problem, many structure-texture decomposition filters have been developed by researchers to address the texture elimination task. These filters include: local extrema filter (LE) [9], relative total variation (RTV) [10], rolling guidance filter (RGF) [11], region covariances filter (RCF) [12], bilateral texture filter (BTF) [13], scale-aware texture smoothing (SATS) [14], Laplacian texture filter (LTF) [15], zero crossing structure decomposition filter (ZCSD) [16], second neighbor anisotropic diffusion filter (SNAD) [17] and relative reductive regression filter (RRRF) [18]. In addition to structure-texture decomposition, these filters are broadly utilized in many applications, such as JPEG artifact removal [15], hatching and image abstraction [19], seamless image cloning and image vectorization [10], seam carving [12], and image inpainting [20].

Inspired by the current development of aforementioned filters and their valuable applications, the aim of this work is to

☆ This article was recommended for publication by M. Kim.

* Corresponding author at: Department of Engineering, La Trobe University, Bundoora, Victoria 3086, Australia.
E-mail address: m.al-nasrawi@latrobe.edu.au (M. Al-nasrawi).

https://doi.org/10.1016/j.cag.2018.12.008
0097-8493/© 2019 Elsevier Ltd. All rights reserved.

investigate a new way for the development of guided edge-preserving filters which will lead to improved performance in terms of handling the presence high-frequency oscillatory components in an image. In the following, we briefly review two fundamental ideas which are related to our work: using guidance information and iterative smoothing.

Using guidance information: Guidance information from the input image or another image can be exploited to guide the filtering process to achieve edge-preserving smoothing. There are two types of guidance: implicit guidance and explicit guidance. Examples of implicit guidance filters are the joint bilateral filter (JBF) [21] and the recursive bilateral filter (RBF) [22]. In the JBF, the range kernel weight is obtained by utilizing the guidance image. In the RBF, the range kernel weight in the current iteration is calculated using the output from the previous iteration. On the other hand, the guided filter (GF) [2] is an example of an explicit guidance filter. In the GF, it is assumed that the image to be filtered is locally related to the guidance image in a linear way. In addition to good edge-preserving properties, details from the guidance image are transferred to the output image. This is a desirable properties for image feathering [2].

Iterative smoothing: In the edge-aware smoothing, the idea is to combine smoothing with edge recovery. It can be implemented iteratively by using an edge-preserving filter and introducing a guidance image [23]. An example of this is the RGF [11] which involves two steps. In the first step, small details are completely eliminated from the original image by a Gaussian filter. In the second step, the result from the first step is used as a guidance image in the JBF [21] to smooth the original image. A recent example is the smooth and iteratively restore filter (SIRF) [24], which follows a similar idea as that of the RGF. The SIRF iteratively restores distinctive edges from the smoothed image. The RGF and the SIRF have been successfully utilized in many applications such as scale-aware smoothing and texture smoothing.

This work is motivated by the fundamental ideas of using the guidance information and iterative smoothing. One key idea of this paper is to extend the Bayesian model averaging filter (BMA) [5] to guided Bayesian model averaging filter by integrating the guidance image into the BMA. Another key idea is based on the well-known guided edge-preserving filters and iterative smoothing. The proposed approach is implemented in two steps. In the first step, pre-smoothing is used to reduce the high-frequency texture information. The result is then processed by one of the commonly used guided edge-preserving filters such as guided Bayesian model averaging filter GBMA, bilateral filter (BF) [1], guided filter (GF) [2], and domain transform filter (DTF) [25]. This work is similar to the RGF [11]. The key difference between this work and those based on the RGF is that in this work, the filtering weight in the range kernel is calculated from the guidance image only once and the resultant image is iteratively filtered from the previous iteration, whereas in the RGF, the filtering weight in the range kernel is iteratively updated to filter the original image. In addition, our approach has the advantage of avoiding rounded edges in large edge structures and blocking effects in small-scale details which appear in the RGF. Such undesirable effects are due to Gaussian smoothing used to generate the guidance image.

The organization of this paper is summarized as follows. In Section 2, we first briefly review edge-preserving filters. In Section 3, we present the proposed structure-texture decomposition method, including the general iterative structure, the guided Bayesian model averaging and other guided edge-preserving filters, guidance image generation, evaluation of guided edge-preserving filters under different parameter settings, the optimal number of iteration, and selection of guided edge-preserving filter. In Section 4, through an analysis of experiment results and a comparison with the state-of-the-art algorithms, we demonstrate that the proposed algorithm is an effective and flexible tool in a wide range of applications including texture smoothing, content-aware image resizing, color pencil sketching, image abstraction, and texture editing. We also demonstrate that the performance of the proposed approach is competitive to the state-of-the-art algorithms. Conclusions are given in Section 5.

2. Related work

Many edge-preserving smoothing techniques have been proposed to separate an image into a structural layer and a textural layer. In the following, we briefly review previous works in the area of structure-texture smoothing relating to the proposed approach.

2.1. Bayesian model averaging

Bayesian model averaging (BMA) is an edge-aware smoothing filter which smooths out fine details in an image while preserving edges. It was firstly developed by Deng [5] as a solution to an optimal estimation problem. The basic idea of the BMA is to estimate a pixel value that belongs to multiple overlapped patches by combining their estimates into a final filtered value.

Notations: To make the notations clear, let I and Y be the input image and the output image. We use the subscript to identify the kth patch of the input image. The patch mean μ_k is calculated from the input image I for the kth patch.

The BMA is defined by the following equations

$$Y = \sum_{k=1}^{M} \alpha_k \beta_k \mu_k + \left(1 - \sum_{k=1}^{M} \alpha_k \beta_k\right) I, \tag{1}$$

$$\beta_k = \frac{N\sigma_o^2}{(N\sigma_o^2 + \sigma_k^2)}, \tag{2}$$

$$\alpha_k = \frac{c}{\sigma_k^2 + \epsilon}, \tag{3}$$

where σ_k is the sample variance for kth patch, σ_o^2 is a user specified parameter, N is the number of pixels in a local patch, c is a normalization factor such that $\sum \alpha_k = 1$, and ϵ is a small number to avoid dividing it by zero. The parameters σ_k and μ_k are obtained from the input image. Therefore, the BMA filter can be called a self-guided BMA.

2.2. Bilateral filter

The BF can be expressed as follows:

$$Y_p = \frac{1}{C_p} \sum_{q \in \Omega_p} \exp\left(-\frac{||p-q||^2}{2\sigma_s^2}\right) \exp\left(-\frac{|A_p - A_q|^2}{2\sigma_r^2}\right) I_q, \tag{4}$$

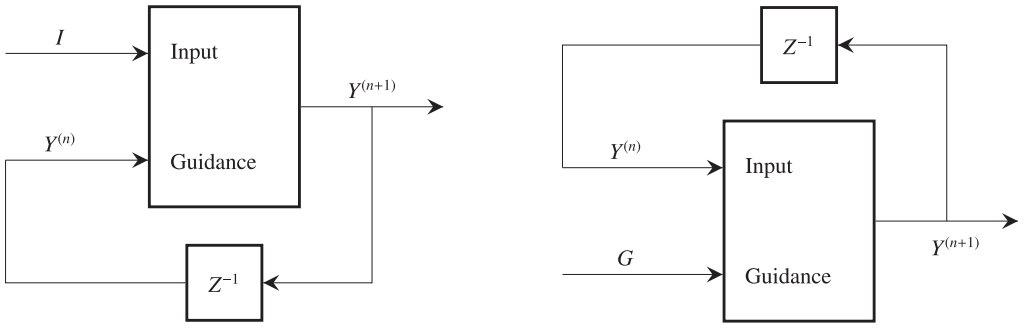
and

$$C_p = \sum_{q \in \Omega_p} \exp\left(-\frac{||p-q||^2}{2\sigma_s^2}\right) \exp\left(-\frac{|A_p - A_q|^2}{2\sigma_r^2}\right), \tag{5}$$

Where Y_p, I_p and A_p are pixels of the output, input images and guidance at location p, respectively. C_p is the normalization factor. Ω_p denotes the neighborhood of pixel p. The two parameters σ_r and σ_s are the scale parameters that control the spatial and range weights, respectively.

An extension of BF is the joint bilateral filter (JBF) which was initially proposed by Petschnigg et al. [21] and later revisited by Eisemann et al. [26]. While in traditional BF the photometric weight is computed from the input image denoted I ($A = I$), in JBF this weight is computed from a guided image denoted G ($A = G$).

Preprint Submitted for review / Computers & Graphics (2018)



(a) RGF ($Y^{(0)} = 0$) (b) EPF ($Y^{(0)} = I$)

Fig. 1. Block diagram of (a) RGF and (b) proposed framework. Z^{-1} denotes a delay function.

2.3. Guided filter

The guided filter (GF) [2] is a local neighborhood filter which assumes the smoothed output Y is a linear transformation of the guidance image G in a local patch Ω_p centred at pixel location p:

$$Y_q = a_p G_q + b_p, \qquad \forall q \in \Omega_p \tag{6}$$

where Ω_p is a square-patch of size $(2r+1) \times (2r+1)$, and the linear coefficients a_p and b_p are determined by minimizing the following objective function:.

$$E(a_p, b_p) = \sum_{q \in \Omega_p} \left((a_p G_q + b_q - I_q)^2 + \epsilon a_p^2 \right), \tag{7}$$

where ϵ is a regularization parameter specified by user. The filtered output of the guided filter can be written as:

$$Y_q = \bar{a}_q I_q + \bar{b}_q, \tag{8}$$

where $\bar{a}_q = \frac{1}{|\Omega|} \sum_{p \in \Omega_q} a_p$ and $\bar{b}_q = \frac{1}{|\Omega|} \sum_{p \in \Omega_q} b_p$ are the average coefficients values of all patches which contain pixel I_q.

2.4. Domain transform filter

The domain transform filter replaces the evaluation of computationally expensive edge-preserving kernels in 5-D with a domain transformation t and a lower dimension $\mathcal{H}$. It can be written as

$$Y_p = \int_{\Omega} F(\hat{p}, \hat{q}) \quad I_q \quad dq = \int_{\Omega} \mathcal{H}(t(\hat{p}), t(\hat{q})) \quad I_q \quad dq, \tag{9}$$

where Y_p and I_q are the outputs and the input images, respectively. F and $\mathcal{H}$ are edge-preserving kernels. $\hat{p}$ and $\hat{q}$ are points on a 2-D manifold in $\mathbb{R}^5$. The isometric 1-D domain transform defined as

$$t(u) = \int_0^u 1 + \frac{\sigma_s}{\sigma_r} \sum_{k=1}^{c} |A_k'(x)| \quad dx, \tag{10}$$

where A_k' denotes the derivative of the input image A with respect to x, c is the number of channels in an image. σ_s is the variance of the kernel $\mathcal{H}$ over the signal's spatial domain Ω, and σ_r is the variance of the kernel $\mathcal{H}$ over the signal's range.

To perform the domain transform filter efficiently, the box kernel is defined as

$$\mathcal{H}(t(\hat{p}), t(\hat{q})) = \delta\{|t(\hat{p}) - t(\hat{q})| \leqslant r\}, \tag{11}$$

where the neighborhood size of the filter is defined as r, and and δ is a Boolean function which yields 1 when its argument is true, and 0 otherwise.

3. The proposed algorithm

There are two key steps in the proposed approach, shown in Fig. 1(b). A guidance image is first generated by filtering an image with a simple filter. The result is then incorporated in the proposed iterative structure to guide the smoothing process which is implemented using the well-known guided edge-preserving filters such as BF, GF, DTF, and the proposed GBMA which will be explained in the next subsection. We also provide a detailed discussion on guidance image generation, the stopping criterion for the iteration, selection of guided edge-preserving filters, and the response of the proposed iterative framework under different parameter settings.

3.1. The proposed guided Bayesian model averaging

In this section, we give a brief description of the GBMA which is used in this paper.

Contrary to the self-guided BMA which is presented in Section 2.1, an external-guided BMA can be obtained when the parameters σ_k and μ_k are computed from the guidance image (G). The self-guided BMA can be deemed a special case of the external-guided BMA filter.

The key idea of adopting the guidance image information is to constrain the smoothing process by making use of the information in the guidance image because both the guidance image and the image to be filtered have a physical relation. For instance, in flash/no-flash denoising application using JBF [21], the guidance image G is chosen as the same scene as I, yet with the flash light condition. As such, the guidance image can be considered as a proximate structure of the original image. In BMA, we can exploit this properties and use the guidance image to estimate the sample mean μ_k and variance σ_k. The key idea is that the parameter σ_k estimated from the guidance image would provide a better estimation than that estimated from the original image. This would lead to better filtering results.

3.2. The general iterative structure of the proposed framework

The general structure of the proposed framework follows that of the guided edge-preserving filters (*EPF*) and an iterative method. It can be formulated based on the well-known guided edge-preserving filters such as BF, GF, DTF and the proposed GBMA, which is presented in Section 3.1. In general, we can define the edge-preserving process as a generic smoothing function *EPF*(I, A) that takes two inputs, namely an image to be filtered I and a guided image A. The filtering process can be expressed as

$$Y = EPF(I, A), \tag{12}$$

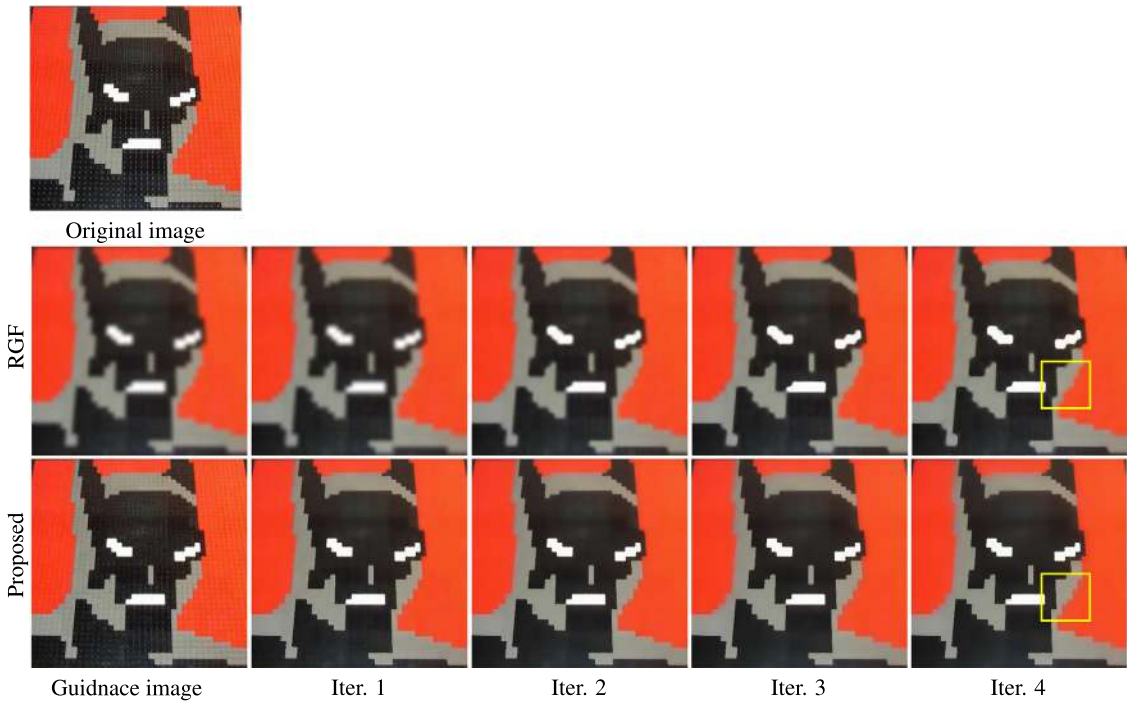


Fig. 2. Structure-texture smoothing results from RGF and our method on "Batman" image. Our approach is able to suppress texture while well preserving the large-scale structures similar to the original edges compared with RGF, as denoted by the yellow square in the last column. The proposed method and RGF have same parameter settings ($\sigma_r = 0.15$, $\sigma_s = 4$, and $N_{iter} = 4$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

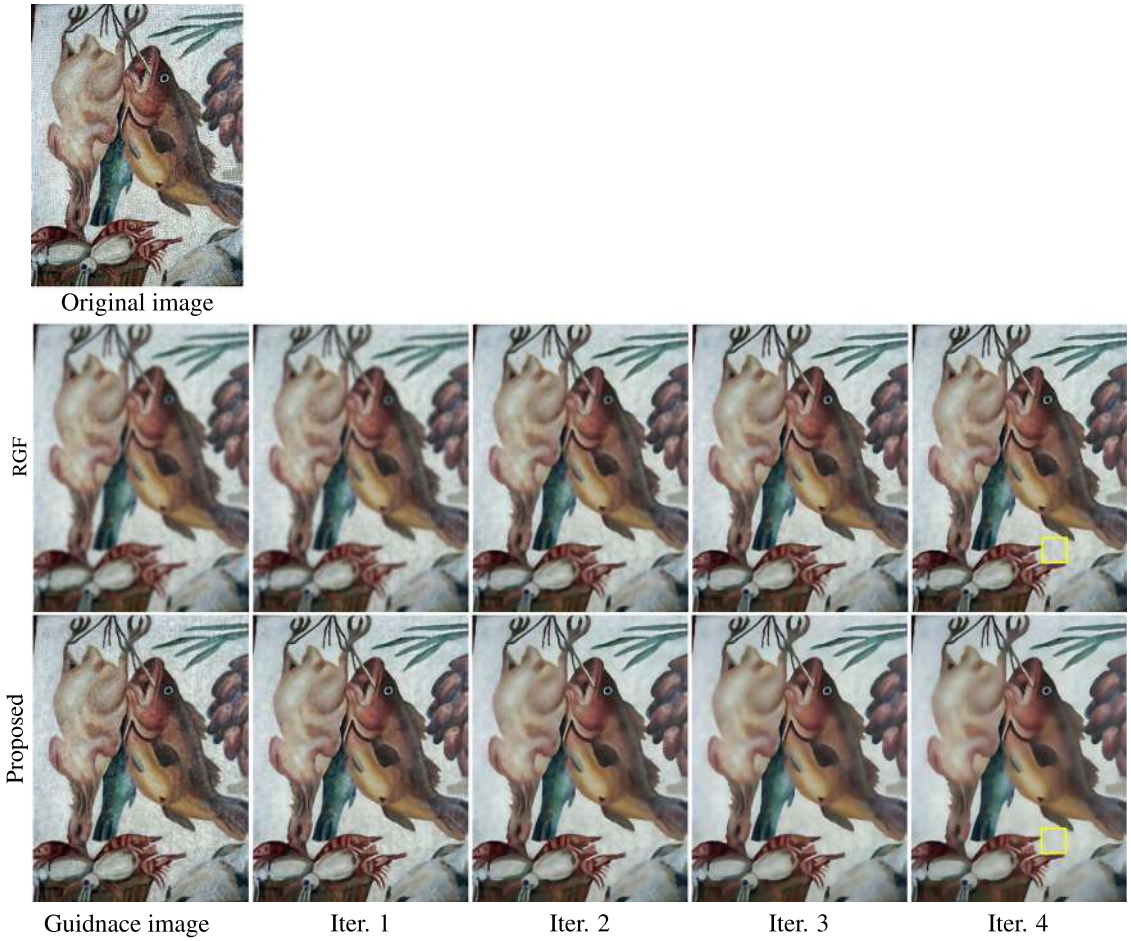


Fig. 3. Structure-texture smoothing results from RGF and our method on "Fish" image. Our approach is able to suppress texture without introducing blocking artifacts compared with RGF, as denoted by the yellow square in the last column. The proposed method and RGF have same parameter settings ($\sigma_r = 0.09$, $\sigma_s = 3$, and $N_{iter} = 4$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

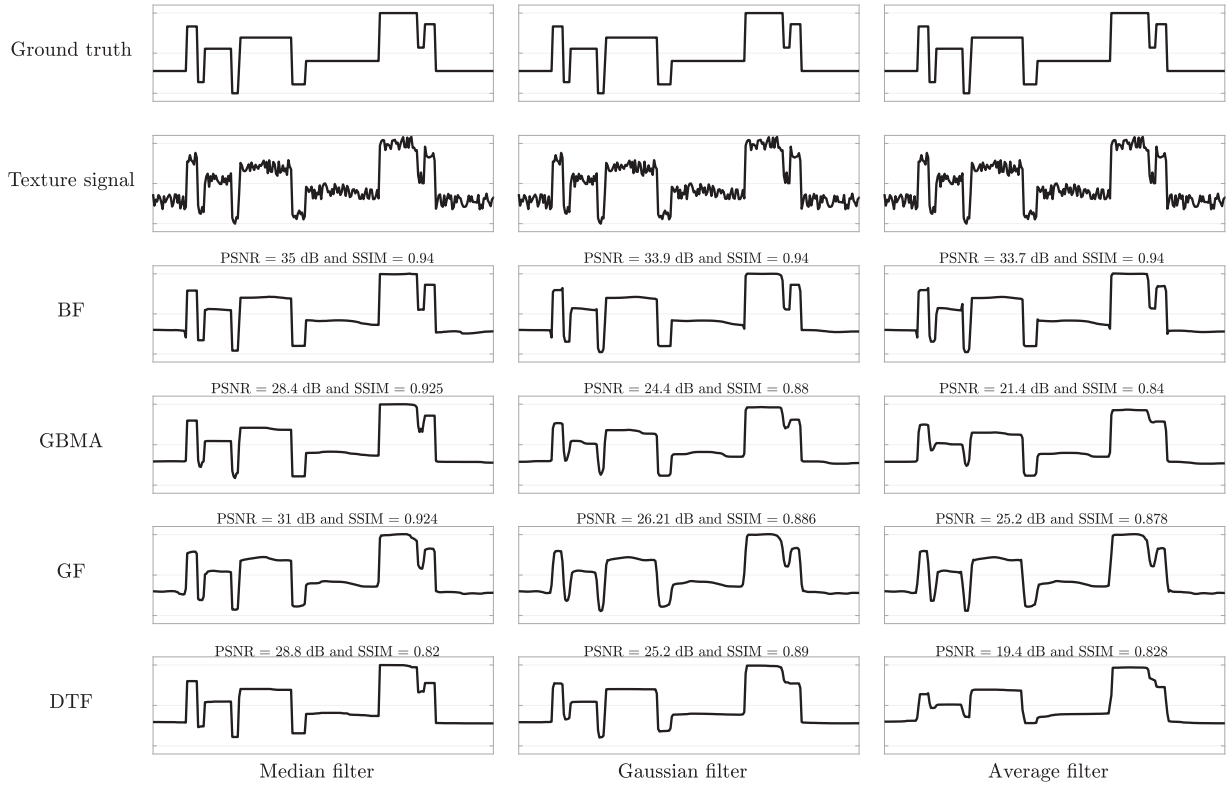


Fig. 4. Results of the proposed approach using different filters as a pre-smoothed stage to obtain the structure-texture decomposition response.

where the above function can be used to represent the filter output of self-guidance as $Y = EPF(I, I)$ when $A = I$ and external-guidance as $Y = EPF(I, G)$, when $A = G$ (G is the guidance image). We can iteratively implement the proposed framework by following similar idea as the rolling guidance filter (RGF) [11]. Let $Y^{(n)}$ indicate the filter output at nth iteration ($n \geq 1$) which can be written as the following

$$Y^{(n+1)} = EPF(Y^{(n)}, G), \tag{13}$$

where $Y^{(0)} = I$, and G is the guidance which can be obtained from a simple smoothing filter such as Gaussian low-pass filter, average filter, and median filters. To highlight the difference between the proposed framework and the rolling guidance filter (RGF), we can describe the RGF as follows

$$Y^{(n+1)} = RGF(I, Y^{(n)}), \tag{14}$$

where $Y^{(0)} = 0$. The key difference between the proposed framework and the RGF is that in the RGF, the guidance image is iteratively updated in each iteration, while the image to be filtered remains the same. In the proposed framework, the guidance image remains the same, while the input image is iteratively updated. The block diagrams of the proposed framework and the RGF are shown in Fig. 1.

3.3. Comparison to RGF

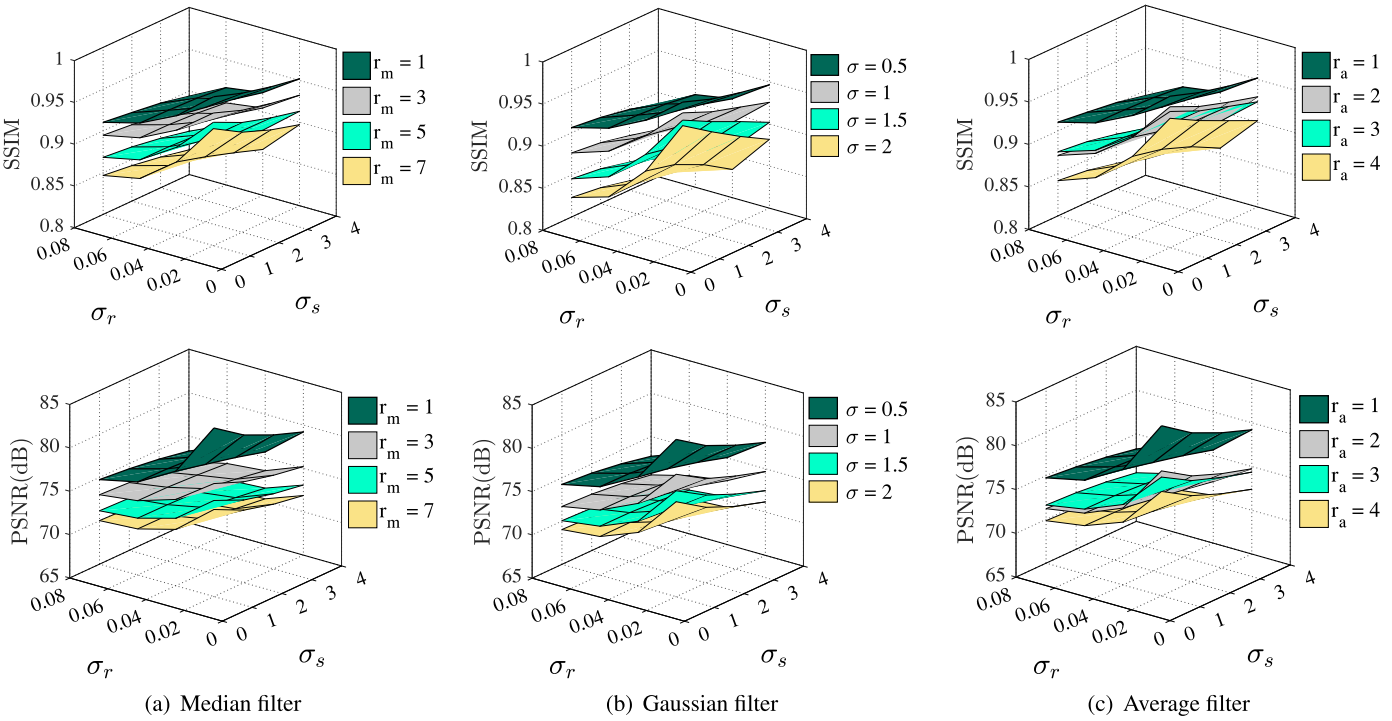
Generally, the proposed filter and RGF share similar idea which comprises two steps. The first step is to eliminate small-scale structures (textures). In the second step, a guided edge-preserving filter is used to handle the large-scale structures (distinctive objects). However, they are different in terms of implementing this idea. In the proposed approach, the smoothing weight in the range kernel is determined from the guidance image only once and iteratively filtering the resultant image from the previous iteration. On the other hand, RGF does the opposite. In the RGF, the smoothing weight in the range kernel is iteratively updated to filter the original image.

Another key difference is in guidance image properties. In RGF, the guidance image is generated by using a Gaussian filter to remove the texture components. However, this process not only blurs the textures but also severely degrades the image's structure as well. This leads to two defects, producing rounded edges (non-articulated restoration of large-scale edges), and blocking artifacts in small-scale details (blocked artifacts the textures areas). Such effects are illustrated in *the second rows* in Figs. 2 and 3, respectively. On the other hand, in the proposed method, the guidance image should be reasonably generated (not necessary removing all texture components) while preserving the distinctive objects similar to the original image. This results in producing better edges without introducing blocking artifacts. This results in *the third rows* in Figs. 2 and 3, respectively.

3.4. Guidance image generation

In the proposed algorithm, a pre-smoothing step is required to generate a guidance image which should have a relatively flat area in homogeneous texture regions and preserve prominent structures. More specifically, the guidance image can be deemed as a proximate structure of the original image. The Gaussian filter targets image structures at a certain scale. Structures with a scale smaller than the scaling parameter of the Gaussian filter will be smoothed out. Inspired by this property, we follow a similar idea to mitigate the effect of texture components.

To obtain the guidance image, an interesting question is: what would be the best filter to generate the guidance image? We perform experiments using three simple filters: Gaussian low-pass filter, average filter, and median filter.



(a) Median filter (b) Gaussian filter (c) Average filter

Fig. 5. Different parameter settings of the pre-smoothed image utilizing median filter, Gaussian filter, and average filter in the proposed iterative framework which adopts BF as a guided edge-preserving filter. The SSIM and PSNR were employed to evaluate the image quality in Fig. 20. The first row and the second row show the SSIM and the PSNR of the image resulted from the proposed algorithm, respectively.

We create a 1-D signal which has variable structures. This signal is used as the ground truth. Then, we add an artificial texture to the signal. The resulting signal is processed by one of the three aforementioned filters. The radius of the neighborhood for the median filter and the average filter are $r_m = 3$ and $r_a = 3$, respectively. The scale parameter of the Gaussian kernel is $\sigma_g = 0.8$. Results from the pre-smoothed filters are used as the guidance image. We tune the parameters of these filters in such a manner that they produce the highest structure similarity index (SSIM) [27] and peak-signal-to-noise-ratio (PSNR). From Fig. 4, we make the following observations. The results of the proposed method that are associated with the median filter have the highest SSIM and PSNR. Thus, the median filter is adopted to produce the guidance image in the proposed approach.

We also study the performance of the proposed approach under different parameter settings of the pre-smoothed filters and the guided edge-preserving filters on image. Similar to the 1-D signal generation, we add texture information to a texture-free image (ground truth image) to simulate the texture components, as shown in Fig. 20(b). The resulting image will be smoothed by pre-smoothed filters. The parameters of the aforementioned pre-smoothed filters as well as the guided edge-preserving filters are tuned to yield different smoothing results. The corresponding PSNR and SSIM of these results are shown in Figs. 5–8. We can make the following observations. The PSNR and SSIM of the proposed framework which utilizes the median filter and the average filter are close to each other. These results are summarized in Table 1. Since the median filter has a good edge-preserving ability. Thus, the median filter will be adopted as the pre-smoothing filter to generate the guidance image.

3.5. Evaluation of filters

In this section, we present the results of the proposed framework under different parameter settings of the four different guided edge-preserving filters. Since the proposed framework is iterative, we conducted all experiments on each filter for a fixed number of iterations.

3.5.1. GBMA

The GBMA has two parameters, σ_0^2 (the variance of the prior distribution) and r (the patch radius). It is clearly seen from Fig. 9 that for small values of σ_0^2, the small-scale detail/texture and prominent structures are preserved, while for larger values of r, distinctive objects as well as small-scale structures are smoothed out.

3.5.2. BF

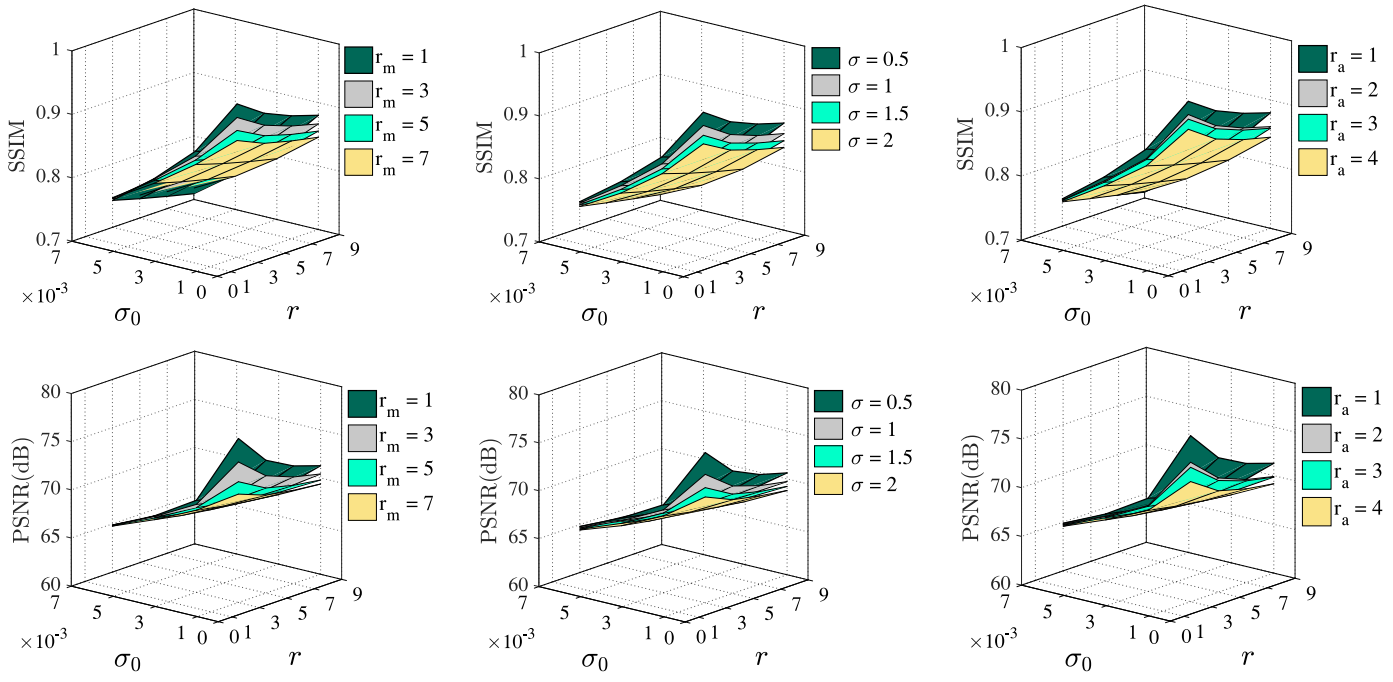
In Fig. 10, we illustrate the results of using BF under different settings. There are two parameters in BF. They are the spatial scale parameter σ_s and the range scale parameter σ_r. We can clearly observe from Fig. 10 that when σ_r is relatively small, small details such as tiles and flowers are preserved. A larger value of σ_s will result in smoothing out distinctive objects such as the angel's face as well as the detail on the wings.

3.5.3. GF

We evaluate the effect of different parameter settings of GF, as shown in Fig. 11. The GF has two parameters ϵ (regulation parameter) and r (patch radius). Setting a large value of r leads to fine detail as well as significant edges being blurred. On the other hand, setting a relatively small value of ϵ leads to the smoothing out of small-scale/texture while preserving the image's distinctive objects.

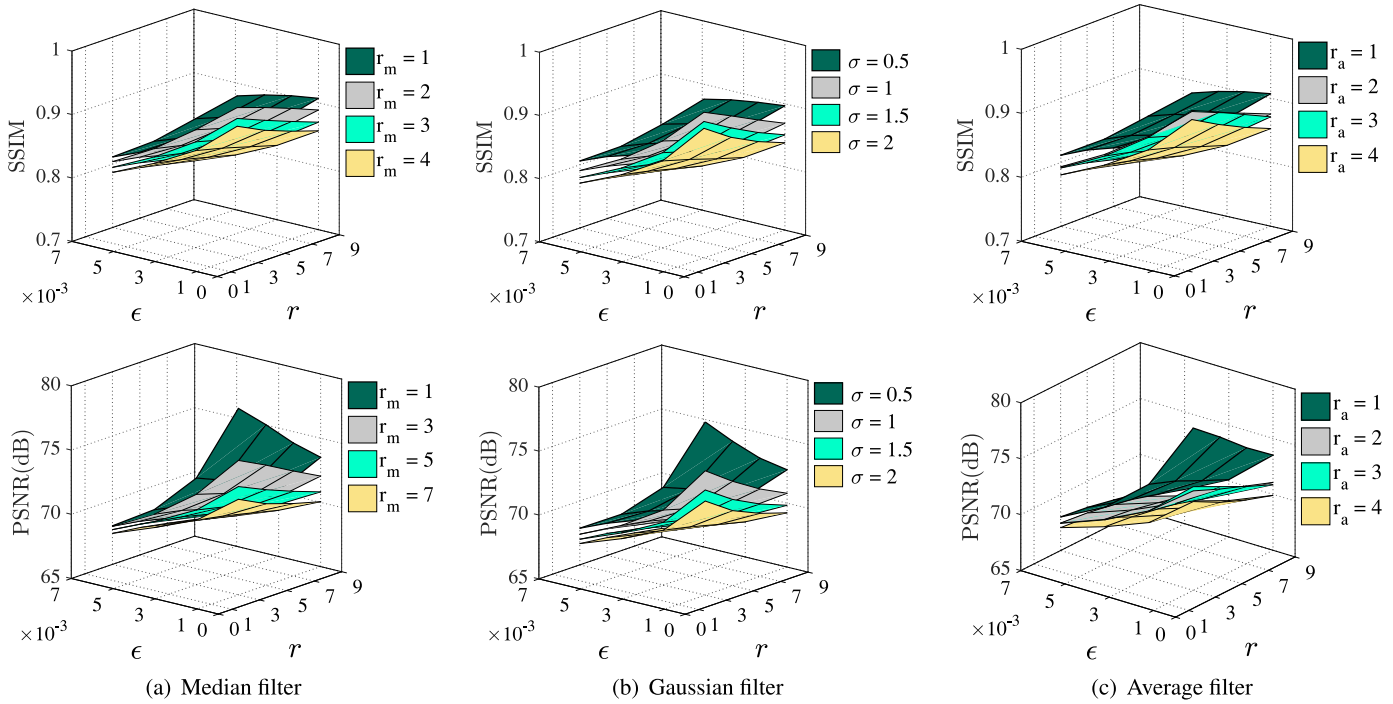
3.5.4. DTF

In Fig. 12, we present results of using DTF for different combinations of the spatial parameter σ_s and the range parameter σ_r. We can observe from this figure that when σ_s is relatively large, sharp edges and small-scale details are blurred. On the other hand, setting a smaller value of σ_r leads to texture preservation.



(a) Median filter (b) Gaussian filter (c) Average filter

Fig. 6. Different parameter settings of the pre-smoothed image utilizing median filter, Gaussian filter, and average filter in the proposed iterative framework which adopts GBMA as a guided edge-preserving filter. The SSIM and PSNR were employed to evaluate the image quality in Fig. 20. The first row and the second row show the SSIM and the PSNR of the image resulted from the proposed algorithm, respectively.



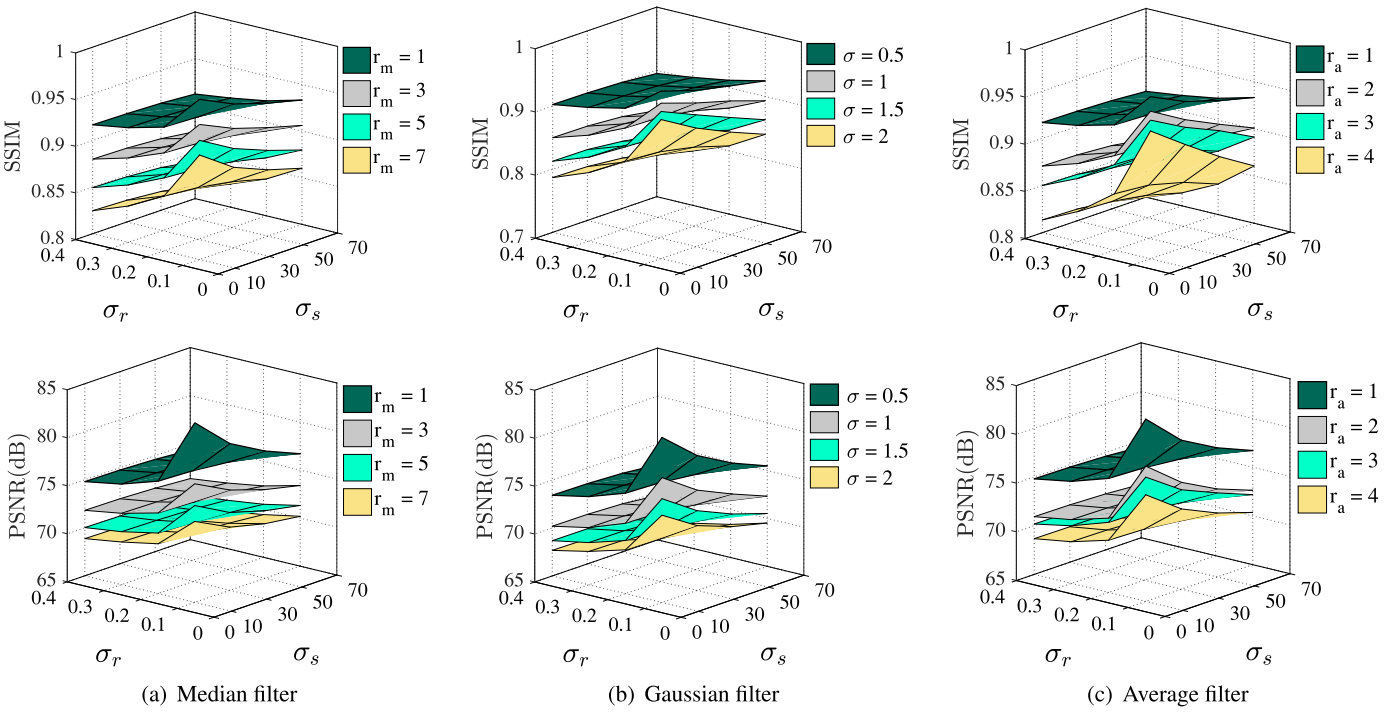
(a) Median filter (b) Gaussian filter (c) Average filter

Fig. 7. Different parameter settings of the pre-smoothed image utilizing median filter, Gaussian filter, and average filter in the proposed iterative framework which adopts GF as a guided edge-preserving filter. The SSIM and PSNR were employed to evaluate the image quality in Fig. 20. The first row and the second row show the SSIM and the PSNR of the image resulted from the proposed algorithm, respectively.

3.6. Stopping criterion and guided edge-preserving filters selection

3.6.1. Stopping criterion

As the number of iterations n increases, the results of the proposed framework eventually converges into a constant image. The output image might reach the unsatisfactory results. Therefore, knowing the number of iterations that yields an optimal trade-off between texture removal and structure preservation is important. We develop a stopping criterion leading to a smoothed image which has salient object features similar to the original image. To study the stopping criterion, we recall the well-known total variation (TV) algorithm [28] which has the cost function defined as:



(a) Median filter (b) Gaussian filter (c) Average filter

Fig. 8. Different parameter settings of the pre-smoothed image utilizing median filter, Gaussian filter, and average filter in the proposed iterative framework which adopts DTF as a guided edge-preserving filter. The SSIM and PSNR were employed to evaluate the image quality in Fig. 20. The first row and the second row show the SSIM and the PSNR of the image resulted from the proposed algorithm, respectively.

Fig. 9. Effect of setting different values of σ_o and r on the proposed method which adopts GBMA as a guided edge-preserving filter in the proposed framework. The number of iterations ($n = 10$).

Fig. 10. Effect of setting different values of σ_r and σ_s on the proposed method which adopts BF as a guided edge-preserving filter in the proposed framework. The number of iterations ($n = 10$).

Table 1
The best PSNR and SSIM of the image in Fig. 20.

Edge-preserving filters	Measure	Median ($r_m = 1$)	Gaussian ($\sigma = 0.5$)	Average ($r_a = 1$)
BF ($\sigma_r = 0.02, \sigma_s = 1$)	PSNR	***84.435***	83.167	84.430
	SSIM	***0.9782***	0.9738	0.9778
GBMA ($\sigma_o = 3, r = 0.001$)	PSNR	77.027	75.679	77.012
	SSIM	0.9418	0.9312	0.9411
GF ($\epsilon = 0.001, r = 3$)	PSNR	79.577	78.552	79.517
	SSIM	0.9548	0.9512	0.9508
DTF ($\sigma_r = 0.1, \sigma_s = 10$)	PSNR	83.700	82.218	83.700
	SSIM	0.9718	0.9649	0.9711

$$J(n) = ||I^{(0)} - I^{(n)}||_2^2 + \lambda||\nabla I^{(n)}||_1, \quad (15)$$

where $I^{(n)}$ is the smoothed image from our framework, $I^{(0)}$ is the original image, n is the number of iterations, λ is the regularization parameter, and $||\nabla I^{(n)}||_1$ is the magnitude of the gradient. To obtain desirable structure-texture decomposition results, Eq. (15) forces the output image to meet the following requirements: (1) it must be close to the original image, and (2) it must be flat in the textured area and have distinctive edges at the image object boundaries. Using this cost function, we can develop a stopping criterion for the proposed approach. We incorporate the four different guided edge-preserving filters which are BF, GF, DTF, and GBMA in our iterative framework. We then conduct experiments using the proposed method and compare the results with the RGF. For each guided edge-preserving filter including the RGF, the cost function (15) is calculated for the output in each iteration. Using the "Fish" image shown in Fig. 16(a), Fig. 13 shows the cost as a function of the number of iterations obtained under different parameter settings (parameters of the guided edge-preserving filters, the RGF, and the regularization parameter of the cost function λ). We can clearly see that every filter has a different minimum cost which represents the optimal structure-texture decomposition. The minimum cost of BF, GF, DTF, and GBMA are (0.5335, 0.6342, 0.6772, and 0.6916) and their corresponding iterations are (10, 19, 5, and 16), respectively. On the other hand, the minimum cost of the RGF is 0.8520 which occurs at iteration 4. The key difference between the RGF and the proposed method is that although the RGF needs less iterations to converge, the cost of the RGF is higher than the proposed algorithm. Since the BF has the smallest cost among all filters tested, it is adopted to produce the structure-texture decomposition smoothing result in the proposed method.

We also conduct further experiments using our iterative framework on ten different texture images. Fig. 14 shows the average costs $J(n)$ over ten images. To simplify our experiment, we fixed $\lambda = 0.4$ and determine the cost as a function of iterations (n). We can clearly see in Fig. 14 that for BF, GF, DTF and GBMA, the minimum cost occurs at four different iterations ($n = 15$, $n = 17$, $n = 6$,

Fig. 11. Effect of setting different values of ϵ and r on the proposed method which adopts GF as a guided edge-preserving filter in the proposed framework. The number of iterations ($n = 10$).

and $n = 7$, respectively). At these iterations, the resultant images will achieve the best results in terms of texture removal while preserving significant edge information similar to the original image. Therefore, we can choose the iterations (n) which correspond to the minimum costs as a stopping criterion. We also note that parameter λ is not required in the proposed approach, but is utilized to experimentally calculate the stopping criterion.

3.6.2. Guided edge-preserving filters selection

In this section, we experimentally demonstrate that the proposed algorithm can be used as a flexible tool in structure-texture decomposition by adopting different types of guided edge-preserving filters. We compare results of the proposed framework using four different guided edge-preserving filters. By carefully tuning their parameters, these filters produce similar structure-texture decomposition effects. To highlight the difference between the four guided edge-preserving filters, we conduct an experiment on a 1-D signal which has variable structures. We divide the 1-D signal into three regions. Each region in the resultant signals are compared with their corresponding regions in other signals. The three regions are illustrated in Fig. 15: a texture+shading (right bar), a distinctive structure edge (middle bar), and a texture region (left bar). In Fig. 15 (right bar), we can clearly observe that while the results for BF, GF, and GBMA are similar, the results for DTF have a small jaggedness. On the other hand, for the region of the signal with the distinctive edge (middle bar), DTF and GBMA produce similar outputs in which the distinctive edges are preserved. It is also noted that BF produces sharp edges, while GF produces blurry edges. In the texture region (left bar), it is clearly seen that the results of all the filters are similar in terms of texture removal. In summary, of the four filters, the BF produces sharper and jagged-free edges.

To further highlight the difference visually, we conduct an experiment on images to extract structures from a highly correlated background. The results are shown in Fig. 16. It can be clearly seen that all filters successfully eliminate the repetitive textures while preserving the prominent structures. However, the GF blurred the significant edges.

3.7. Summary

Guided edge-preserving filters can be implemented in the same structure shown in Fig. 1 and produce different structure-texture smoothing results. The BF produces the best results in terms of texture smoothing. This is confirmed by the largest SSIM and PSNR which are presented in Figs. 4, 5, and Table 1. It also achieves the minimum cost measured by the total variation, as shown in Figs. 13(a) and 14. Fig. 15 shows that the BF provides a good trade-off between texture smoothing and edges preservation without introducing jagged and blurred edges. In addition, the overall structure of the resultant signal is similar to the original signal. Therefore, the BF is chosen to demonstrate the applications of the proposed method in the next section.

4. Results and comparisons

All experiments were conducted using a PC with an Intel-i7 processor running at 3.40 GHZ with 32 GB RAM. MATLAB is utilized as the programming language.

Fig. 12. Effect of setting different values of σ_s and σ_r on the proposed method which adopts DTF as an edge-preserving filter in the proposed framework. The number of iterations ($n = 10$).

Table 2
A comparison of the running time (seconds) results of the proposed method using BF, DTF, GF and GBMA which produce texture smoothing of images in Figs. 16(a) ("Fish" image), 18, and 19, respectively.

Example	Img. size	BF	DTF	GF	GBMA
Fig. 16	600×450	0.62	16.00	0.92	***0.31***
Fig. 18	640×454	0.66	18.42	1.00	***0.33***
Fig. 19	639×640	1.05	26.02	1.34	***0.54***

4.1. Convergence analysis

In this study, we test the convergence of the proposed algorithm utilizing BF, GF, DTF, and GBMA. Results for the "Fish" image are shown in Fig. 17. The convergence is measured by the cost function (Eq. (15)). As the number of iterations increase, the resultant image converges to a flat image. More intuitively, referring to Eq. (15), the smoothed image $I^{(n)}$ reaches the mean value. On the other hand, the second term in Eq. (15) approaches zero. This leads to a constant cost $J(n)$ when $n \rightarrow \infty$.

4.2. Running time

Table 2 shows a comparison of the running time of the proposed approach which uses BF, DTF, GF, and GBMA. The running time of the proposed filter is calculated by adding the running time of the pre-smoothed filter to the running time of the guided edge-preserving filter. From Table 2, we can clearly see that GBMA is faster than DTF, GF, and BF. It is also worth mentioning that in our implementation we employ the bilateral grid [29] which is among the fastest implementation of bilateral filter.

In Table 3, we also compare the running time for the various algorithms to process the images in Figs. 16(a) ("Fish" image), 18, and 19, respectively. The experiments show that the running time of the proposed filter is faster than the relative total variation filter (RTV) [10], rolling guidance filter (RGF) [11], bilateral texture filter (BTF) [13], spanning tree filter (STF), static and dynamic guidance filter (SDF) [6], interval gradient filter (IGF) [30], and scale-aware filter (SAF) [14].

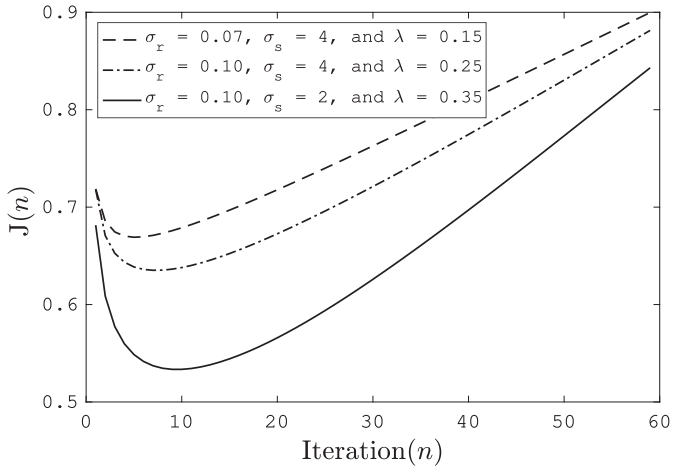
4.3. Applications

Extracting a meaningful structure from a highly-textured scene is an important step in many applications. In this section, we present applications in image structure-texture decomposition, image abstraction, pencil drawing, content-aware image resizing, and editing.

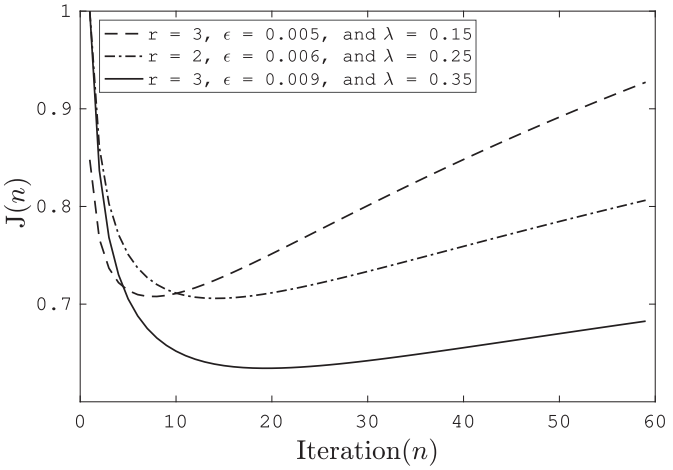
4.3.1. Structure-texture decomposition

We experimentally demonstrate that the proposed approach can be used as an effective tool in structure-texture decomposition smoothing. In Fig. 18, we compare our approach with a number of state-of-the-art structure-texture decomposition algorithms, including relative total filter (RTV) [10], rolling guidance filter (RGF) [11], spanning tree filter (STF) [31], bilateral texture filter (BTF) [13], static and dynamic guidance filter (SDF) [6], scale-aware filter (SAF) [14], and interval gradient filter (IGF) [30]. To produce results for these algorithms, we manually tune parameters such that the prominent structures in different scales are preserved and textures are eliminated.

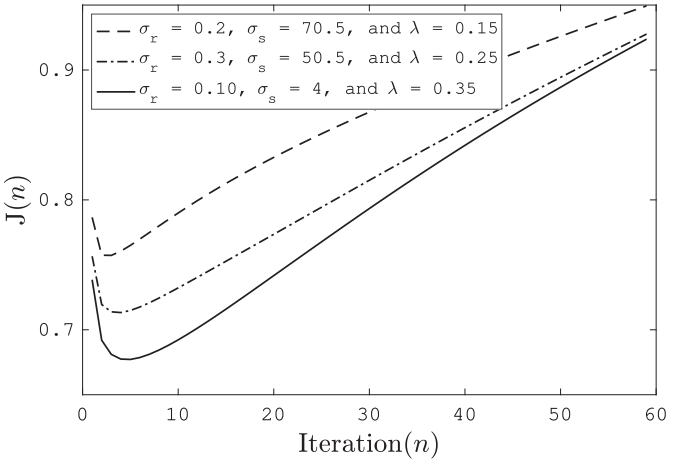
The results are shown in Fig. 18. In the texture regions denoted by the red rectangles. It can be clearly observed that the results



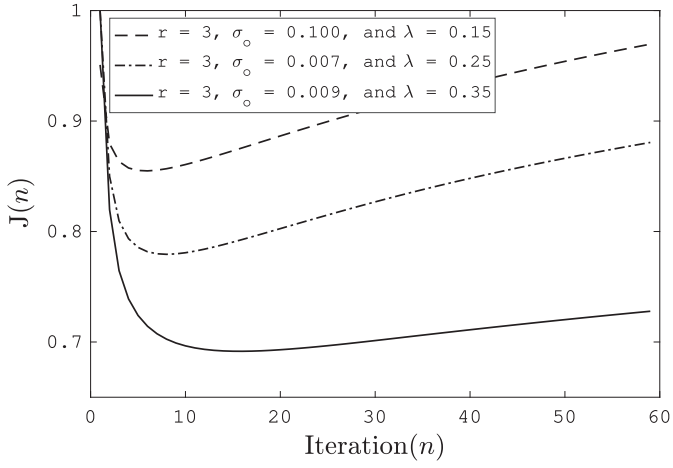
(a) The minimum cost of BF is $J(n) = 0.5335$. It occurs at iteration $n = 10$.



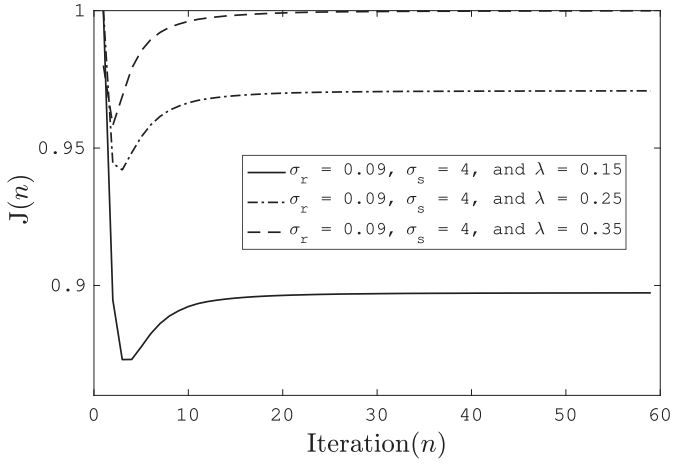
(b) The minimum cost of GF is $J(n) = 0.6343$. It occurs at iteration $n = 19$.



(c) The minimum cost of DTF is $J(n) = 0.6772$. It occurs at iteration $n = 5$.



(d) The minimum cost of GBMA is $J(n) = 0.6916$. It occurs at iteration $n = 16$.



(e) The minimum cost of RGF is $J(n) = 0.8520$. It occurs at iteration $n = 4$.

Fig. 13. Cost functions $J(n)$ for different guided edge-preserving filters (BF, GF, DTF, GBMA, and RGF) under different parameter settings using "Fish" image which is shown in Fig. 16(a).

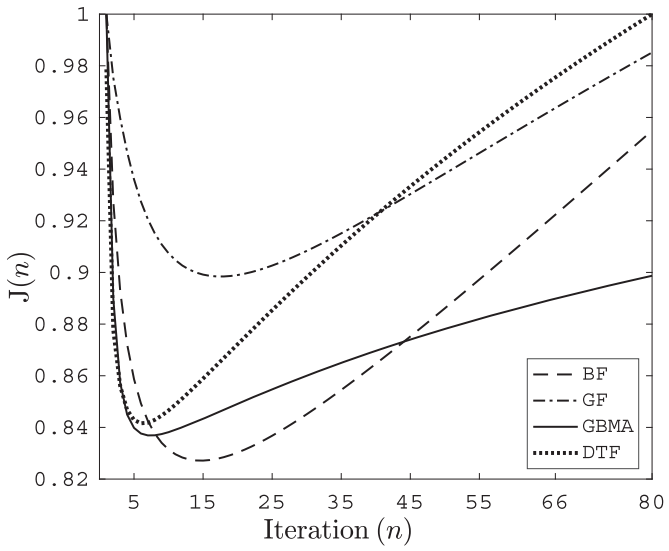


Fig. 14. Cost function $J(n)$ averaged over the 10 test texture images for four different guided edge-preserving filters (BF, GF, DTF, and GBMA) and the regularization parameter $\lambda = 0.4$. For each edge-preserving filter, the minimum cost $J(n)$ which occurs at iterations 15, 17, 6, and 7, respectively.

of RTV, RGF, BTF, STF, IGF, and SAF fail to smooth out the textures. They also blur the small-scale structures, which are denoted by the blue rectangles in Fig. 18. We can also see that our result is competitive with SDF in terms of texture elimination and outperforms it in terms of preserving the small-scale structures, as shown in Fig. 18(f) and (i). On the other hand, the proposed algorithm runs 11-times faster than SDF. It is also worth mentioning that both STF and the proposed filter produce results which are free of the staircase artifacts. Such artifacts appear in results of other filters (see Mona Lisa's cheek). In summary, the proposed approach achieves better trade-off between texture smoothing and structure preservation without introducing staircase effects.

We also demonstrate that the proposed algorithm can be used in structure-texture decomposition for an image with low-contrast details. In Fig. 19, it can be clearly seen that our filter preserves the low-contrast features better than RTV, RGF, BTF, SDF, IGF, and SAF, especially details such as eyelashes and hair, as indicated by the red and blue squares in Fig. 19(i). On the other hand, STF achieves comparable results to our result except that our result has better contrast than STF in terms of overall appearance.

We also provide an objective evaluation for the proposed approach. The quantitative evaluation has been made by adding texture components to a texture-free image which is employed as the ground truth. An example is illustrated in Fig. 20. In this experiment, parameters of each filter are purposely tuned to produce an output such that it preserves the global structures and eliminates the texture components as much as possible. The peak-signal-to-noise-ratio (PSNR) and the structure similarity index (SSIM) [27] are used to measure the quality of the structure-texture smoothing performance. The qualitative evaluation of the proposed method is confirmed by the largest PSNR and SSIM associated with our results. The PSNR and SSIM values shown in Table 4 show that our algorithm outperforms RTV, RGF, BTF, STF, SDF, IGF, and SAF.

4.3.2. Image abstraction

The non-photorealistic rendering algorithm [32,33] obtains the stylization effect by removing the low-contrast features of an image while maintaining the high-contrast content by utilizing an edge-aware filter. The extracted prominent edges are overlaid on the smoothed image to generate a cartoon-like of the image. We find that abstracting an image by smoothing out the small-scale elements rather than the low-contrast features while maintaining the large-scale structure objects can achieve better

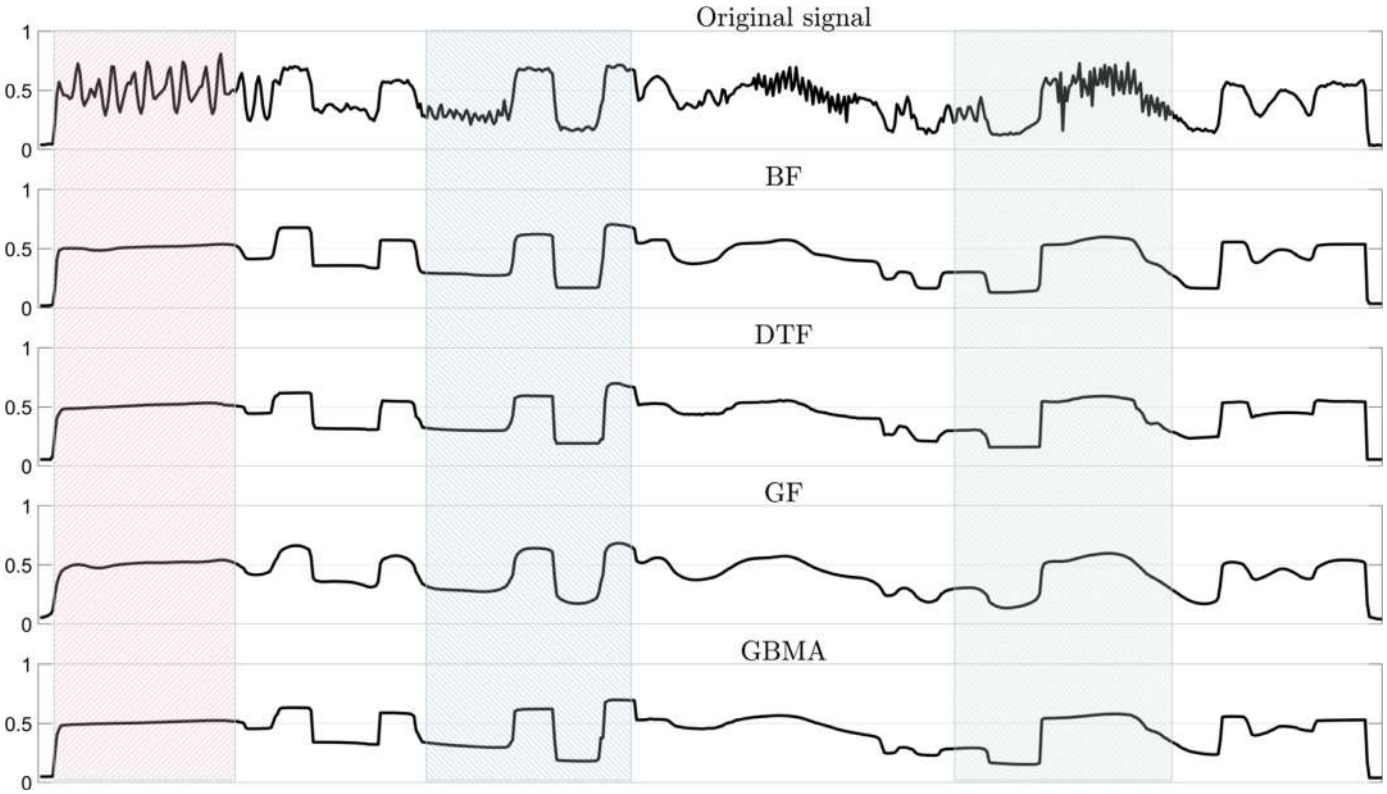


Fig. 15. A comparison of the proposed algorithm for filtering a 1-D signal using different edge-preserving filters.

Fig. 16. A comparison of the proposed algorithm's results using different guided edge-preserving filters.

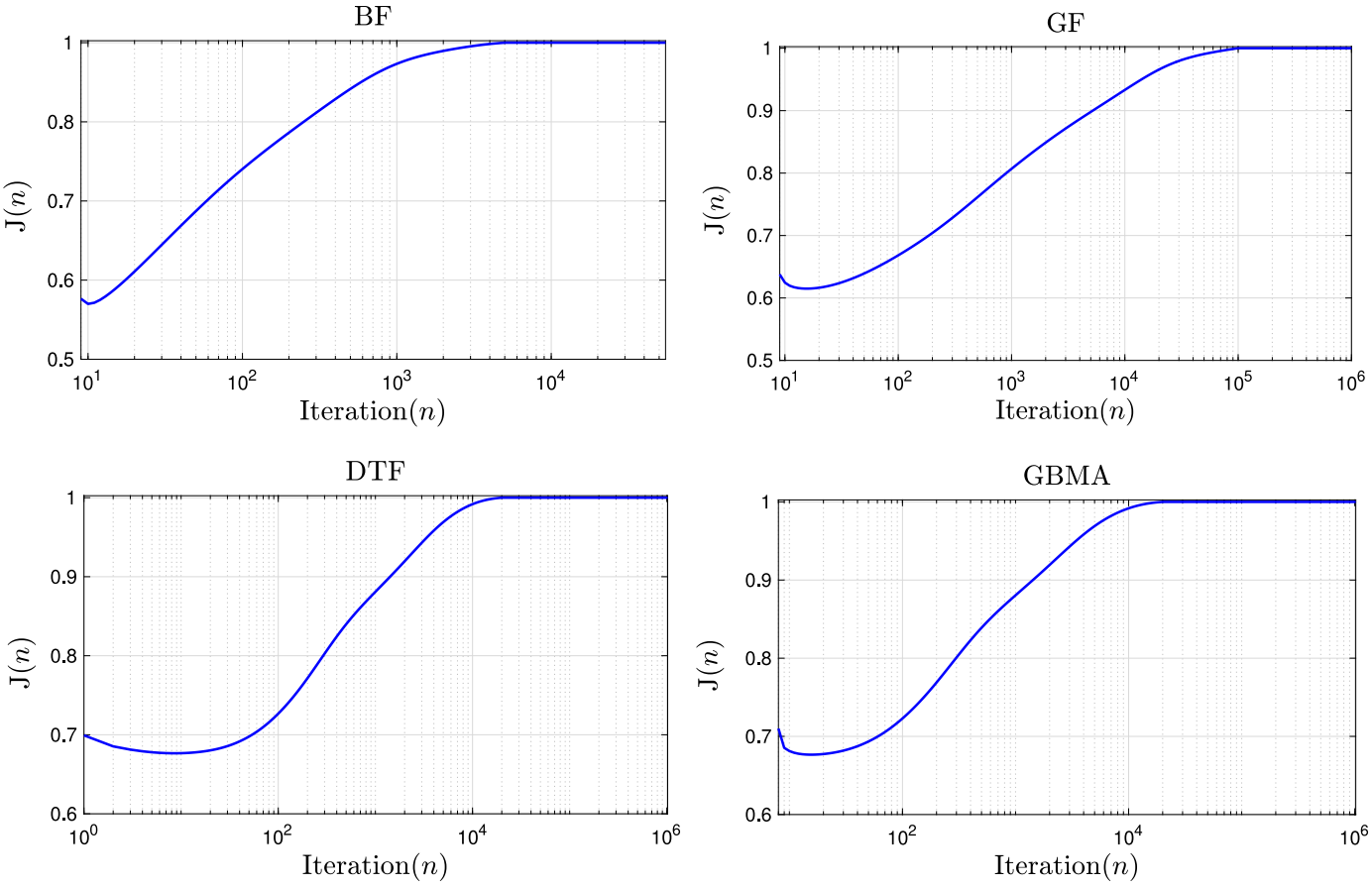


Fig. 17. The convergence of the proposed algorithm results using different guided edge-preserving filters.

Table 3
Running time (seconds) of texture smoothing experiments of images in Figs. 18, 19, and 16(a) ("Fish" image), respectively. Results of the proposed filter using BF are listed in the last column.

Example	Img. size	RTV	RGF	BTF	STF	SDF	IGF	SAF	Proposed
Fig. 16	600 × 450	2.03	1.10	10.75	0.62	6.83	5.11	1.64	***0.62***
Fig. 18	640 × 454	2.09	1.53	35.28	0.89	6.87	7.13	1.37	***0.66***
Fig. 19	639 × 640	2.49	1.50	15.83	1.34	6.93	9.44	2.44	***1.05***

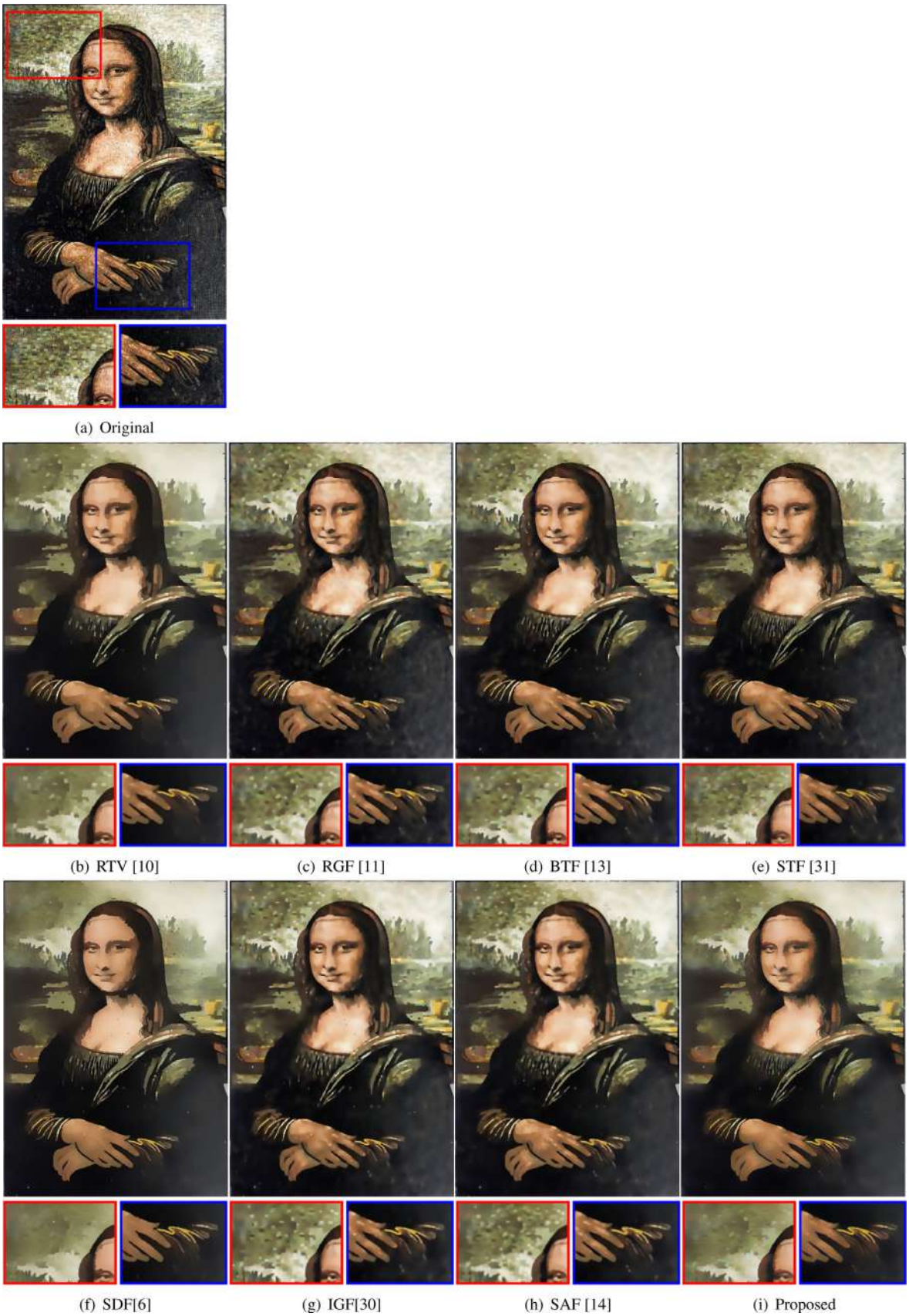


Fig. 18. A comparison of the structure-texture decomposition results for the "Mona Lisa" image with a number of state-of-the-art algorithms. (a) Original image, (b) RTV ($\sigma = 3, \lambda = 0.01, \epsilon = 0.02, N_{iter} = 4$), (c) RGF ($\sigma_r = 0.07, \sigma_s = 3, N_{iter} = 10$), (d) BTF ($k = 3, N_{iter} = 10$), (e) STF ($\sigma_s = 3, \sigma_r = 0.01, \sigma = 0.06, N_{iter} = 4$), (f) SDF ($\sigma = 0.5, \mu = 70, \nu = 300, \lambda = 50, steps = 5$), (g) IGF ($\sigma = 1.5, \epsilon = 0.03^2$), (h) SAF ($\sigma = 3.5, \sigma_r = 0.05, N_{iter} = 3$), (h) Our filter using BF ($\sigma_s = 4, \sigma_r = 0.1, N_{iter} = 5$).

Fig. 19. A comparison of the structure-texture decomposition results of the proposed method with a number of state-of-the-art algorithms. (a) Original image, (b) RTV ($\sigma = 2, \lambda = 0.005, \epsilon = 0.02, N_{iter} = 4$), (c) RGF ($\sigma_r = 0.035, \sigma_s = 4, N_{iter} = 8$), (d) BTF ($k = 3, N_{iter} = 3$), (e) STF ($\sigma_s = 3.5, \sigma_r = 0.03, \sigma = 0.004, N_{iter} = 4$), (f) SDF ($\sigma = 0.5, \mu = 100, \nu = 600, \lambda = 50, steps = 3$), (g) IGF ($\sigma = 1.1, \epsilon = 0.03^2$), (h) SAF ($\sigma = 2, \sigma_r = 0.03, N_{iter} = 6$), (h) Our filter using BF ($\sigma_s = 4, \sigma_r = 0.04, N_{iter} = 5$).

non-photorealistic rendering results. This is because smoothing small size features from an image and emphasising the overall image structures is helpful for human perception. Since the proposed approach preserves the low-contrast details, our abstraction results lead to a better overall result.

In this experiment, we purposely tune the parameters of all filters to produce outputs such that they remove small-scale details while retaining large-scale objects. Fig. 21 shows the non-photorealistic results using our filter, the region covariance filter (RCF) and the relative reductive regression filter (RRRF) [18]. The

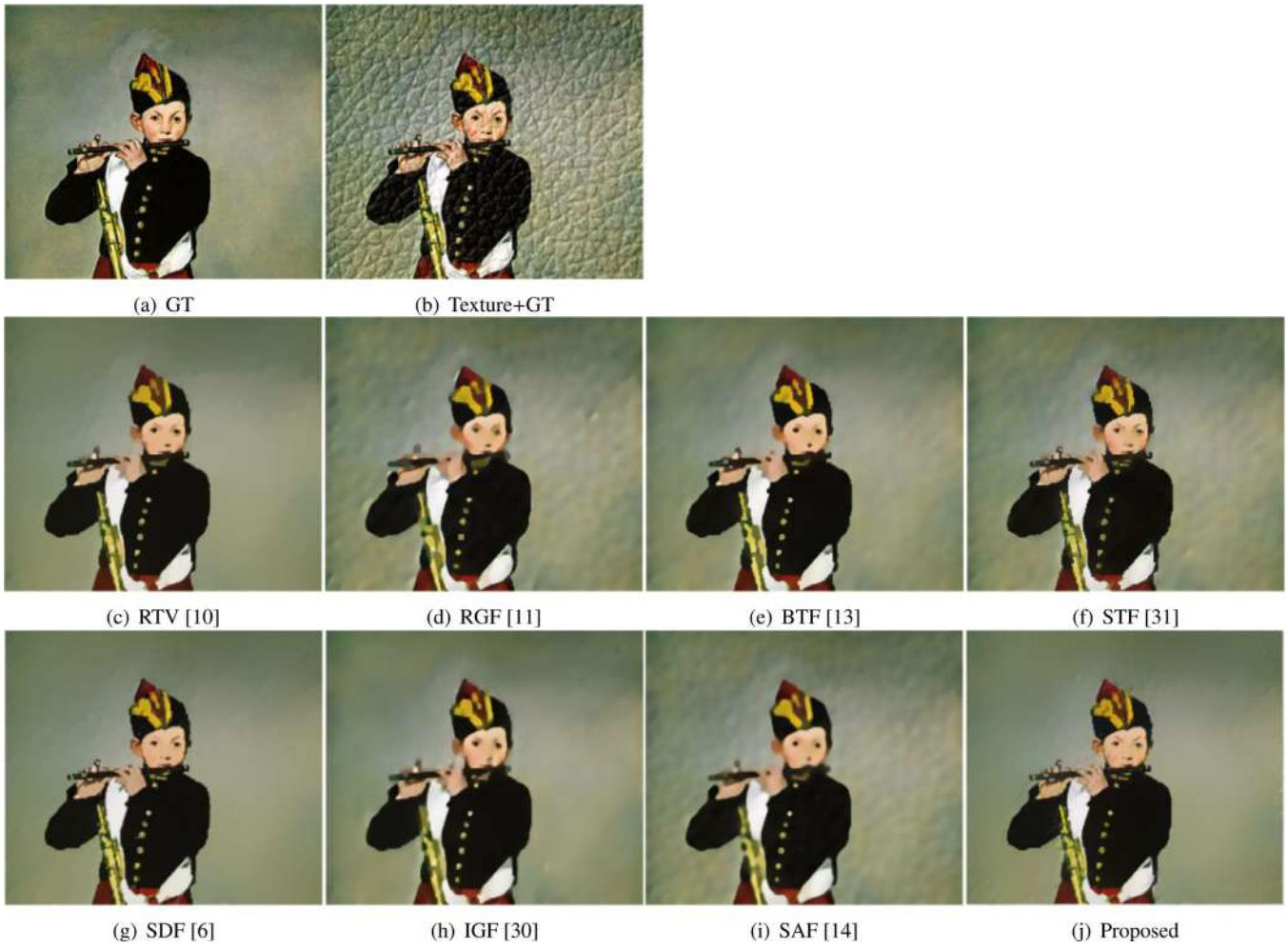


Fig. 20. A comparison of the structure-texture decomposition smoothing results for the "Flautist" image with a number of state-of-the-art algorithms. (a) Ground truth (GT), (b) Texture+ground truth, (c) RTV ($\sigma = 2.5, \lambda = 0.015, \epsilon = 0.02, N_{iter} = 4$), (d) RGF ($\sigma_r = 0.05, \sigma_s = 3.9, N_{iter} = 5$), (e) BTF ($k = 3, N_{iter} = 15$), (f) STF ($\sigma_s = 1, \sigma_r = 0.05, \sigma = 0.002, N_{iter} = 9$), (g) SDF ($\sigma = 2, \mu = 200, \nu = 500, \lambda = 200, steps = 5$), (h) IGF ($\sigma = 2, \epsilon = 0.03^2$), (i) SAF ($\sigma = 2.4, \sigma_r = 0.07, N_{iter} = 5$), (j) Our filter using BF ($\sigma_s = 4.1, \sigma_r = 0.06, N_{iter} = 15$).

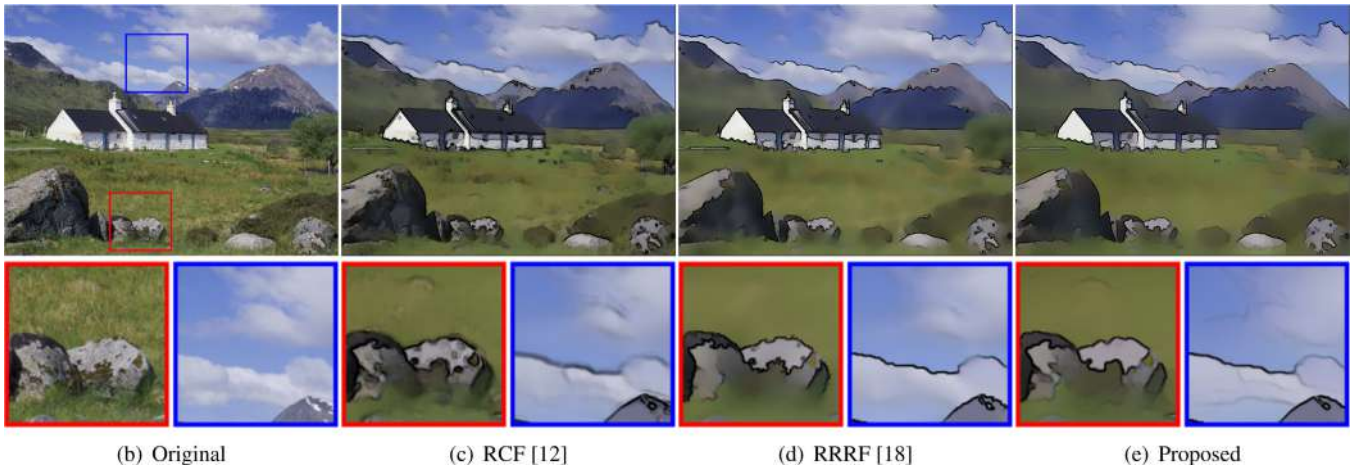


Fig. 21. A comparison of our image's non-photorealistic results with different filtering methods. (a) Original image, (b) RCF ($k = 15,\ r = 3,\ \sigma = 0.09,\ Model\ 1$), (c) RRRF ($\sigma_s = 5,\ \sigma_r = 0.06,\ N_{iter} = 5$), (e) Proposed ($\sigma_s = 4,\ \sigma_r = 0.04,\ N_{iter} = 15$).

figure demonstrates that the image resulting from our filter preserved the low-contrast features, such as the details in the clouds and grass, better than RCF and RRRF, as shown in Fig. 21(e). It is also clearly seen that RCF and RRRF fail to smooth out the small-scale details, such as the details on the rock, as shown in Fig. 21(c) and (d) (the red squares). In addition, the result obtained from RCF has a ghosting effect around the significant boundaries. On the other hand, although RRRF produces sharp edges, it also has irregular edge boundaries.

Fig. 22. Color pencil sketching results.

Fig. 23. Content-aware image resizing in texture images. (a) Original image. (b) The image smoothed by the proposed approach. (c) and (d) show the seams to be eliminated by the approach in [35] and the modified [35] by our algorithm, respectively. Our result in (f) is more visually appealing than in (e).

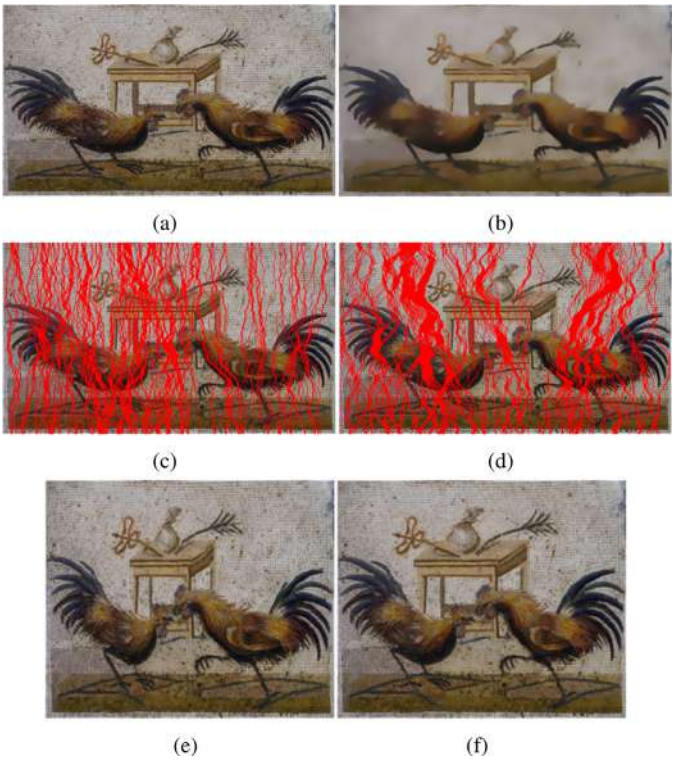


Fig. 24. More content-aware image resizing results in texture images. (a) Original image. (b) The image smoothed by the proposed approach. (c) and (d) show the seams to be eliminated by the approach in [35] and the modified [35] by our algorithm, respectively. Our result in (f) is more visually appealing than in (e).

Table 4
The quantitative evaluation of the structure-texture smoothing results using PSNR and SSIM for images in Fig. 20. The PSNR and SSIM of the synthesized image (shown in Fig. 20(b)) are 0.621 and 22.859 (dB), respectively. Results of the proposed algorithm are listed in the last column.

Measure	RTV	RGF	BTF	STF	SDF	IGF	SAF	Proposed
SSIM	0.891	0.880	0.894	0.881	0.872	0.896	0.885	**0.911**
PSNR (dB)	26.322	24.788	25.463	24.730	23.640	25.771	24.925	**27.774**

Fig. 25. Texture replacement results. (a) Original image. (b) The structure layer extracted by the proposed approach. (c) The resulting image after adding (b) to another texture image.

4.3.3. Color pencil sketching

Pencil sketching is a non-photorealistic transformation of a real image. The structure-texture decomposition result is used with other image information to achieve this effect. For example, gradients or edges can be added to the smoothed version of the original image to produce pencil-like or cartoon-like images. Lu et al. [34] manipulated image gradients to generate pencil sketching effects. In the texture image, the pencil drawing can yield visually unappealing results due to the high-frequency oscillatory components. Therefore, we can pre-process the original image using the proposed algorithm to obtain a better result. Fig. 22 presents the pencil sketching effects using our filter and the approach in [34]. The figure demonstrates that the results from our filter are more visually pleasing and more abstract than the results obtained from the algorithm in [34].

4.3.4. Content-aware image resizing

Seam carving is used in content-aware image resizing [35]. Every seam is generated by a connected path of pixels which have a low energy cost. The energy function is defined in terms of the gradient magnitude. Important objects in an image can be protected by these seams after image resizing. In a texture image, applying content-aware image resizing directly might not produce visually appealing results because of the correlation between textures and objects in the scene. More specifically, the existence of textures might lead to misleading seams carving. Therefore, some important content could be eliminated. To solve this problem, we apply the proposed structure-texture decomposition approach on the original image to alleviate the effect of textures prior to seam carving. The filtered version of the image is used as a guidance map for the original seam carving algorithm to produce the final results.

For example, Fig. 23 shows three faces: left, middle and right. It can be clearly observed that most of the vertical seams pass through the left-hand side of the left face and the middle face in the scene which leads to unpredicted results, as shown in Fig. 23(e). By using the proposed filter, many of the vertical seams no longer cross through the important details, rather they now target the less important details such as the tiles in the background, as shown in Fig. 23(d). This is due to using the smoothed image produced by our algorithm which leads to more satisfactory results, as shown in Fig. 23(f).

Another example is shown in Fig. 24. It can be seen in Fig. 24(c) that the vertical seams are distributed randomly over the whole image without considering image objects. This leads to a distorted result after applying content-aware resizing, such as the noticeable distortion in the sticks as well as the rooster on the left-hand side of the image, as shown in Fig. 24(e). On the other hand, after processing by our filter, although the vertical seams still pass through the roosters as in Fig. 24(d), most of the vertical seams are in areas of unimportant repetitive details such as tiles in the background and some of the roosters' feathers. This leads to a less distorted resized image, as shown in Fig. 24(f).

4.3.5. Texture editing

Using the proposed method, we are able to decompose the highly-textured image into a structure layer and texture layer. We can simply replace the texture layer with another texture image. Thus, the texture components in the original image are replace by the new texture components. Fig. 25 illustrates an example.

5. Conclusion

In this paper, a new technique for structure-texture decomposition smoothing is introduced. The proposed method comprises two steps. A guidance image is first generated and the result is then processed by one of the guided edge-preserving filters such as the extended Bayesian model averaging filter (guided Bayesian model averaging (GBMA)), bilateral filter (BF), guided filter (GF), and domain transform filter (DTF). We compare the responses of GBMA, BF, GF, and DTF in areas of an image with texture and significant edges. We observe that the response of the proposed filter using the BF produces the best results among all filters tested. We also study an important issue such as guidance image generation, the stopping criterion for the iteration, and guided edge-preserving selection. The experiment results and comparisons show that in a wide range of applications, the proposed algorithm achieves competitive structure-texture decomposition smoothing results. In particular, the proposed filter has the best performance in the structure-texture decomposition for a low-contrast texture image.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: Proceedings of the sixth international conference on computer vision. Washington, DC: IEEE Computer Society; 1998. p. 839–46.
[2] He K, Sun J, Tang X. Guided image filtering. IEEE Trans Pattern Anal Mach Intell 2013;35(6):1397–409.
[3] Xu L, Lu C, Xu Y, Jia J. Image smoothing via L_0 gradient minimization. ACM Trans Graph 2011;30(6) 174:1–174:12.
[4] Thévenaz P, Sage D, Unser M. Bi-exponential edge-preserving smoother. IEEE Trans Image Process 2012;21(9):3924–36.
[5] Deng G. Edge-aware BMA filters. IEEE Trans Image Process 2016;25(1):439–54.
[6] Ham B, Cho M, Ponce J. Robust image filtering using joint static and dynamic guidance. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 4823–31.
[7] Thai B, Al-nasrawi M, Deng G, Su Z. Semi-guided bilateral filter. IET Image Process 2017;11(7):512–21.
[8] Al-nasrawi M, Deng G, Thai B. Edge-aware smoothing through adaptive interpolation. Signal Image Video Process 2018;12(2):347–54.
[9] Subr K, Soler C, Durand F. Edge-preserving multiscale image decomposition based on local extrema. ACM Trans Graph 2009;28(5) 147:1–147:9.
[10] Xu L, Yan Q, Xia Y, Jia J. Structure extraction from texture via relative total variation. ACM Trans Graph 2012;31(6):139.
[11] Zhang Q, Shen X, Xu L, Jia J. Rolling guidance filter. In: Proceedings of the European conference on computer vision. Springer; 2014. p. 815–30.
[12] Karacan L, Erdem E, Erdem A. Structure-preserving image smoothing via region covariances. ACM Trans Graph 2013;32(6) 176:1–176:11.
[13] Cho H, Lee H, Kang H, Lee S. Bilateral texture filtering. ACM Trans Graph 2014;33(4) 128:1–128:8.
[14] Jeon J, Lee H, Kang H, Lee S. Scale-aware structure-preserving texture filtering. In: Proceeidings of the Computer Graphics Forum, vol. 35; 2016. p. 77–86.
[15] Du H, Jin X, Willis PJ. Two-level joint local Laplacian texture filtering. Vis Comput 2016;32(12):1537–48.
[16] Jiang X, Yao H, Liu S. How many zero crossings? A method for structure-texture image decomposition. Comput Graph 2017;68:129–41.
[17] Al-nasrawi M, Deng G, Waheed W. Structure extraction of images using anisotropic diffusion with directional second neighbour derivative operator. Multimedia Tools Appl 2018b. doi:10.1007/s11042-018-6377-7.
[18] Su Z, Zeng B, Miao J, Luo X, Yin B, Chen Q. Relative reductive structure-aware regression filter. J Comput Appl Math 2018;329:244–255.
[19] Zhou Z, Wang B, Ma J. Scale-aware edge-preserving image filtering via iterative global optimization. IEEE Trans Multimedia 2018;20(6).
[20] Wu H, Xu D, Yuan G. Region covariance based total variation optimization for structure-texture decomposition. Multimedia Tools Appl 2017:1–21.
[21] Petschnigg G, Szeliski R, Agrawala M, Cohen M, Hoppe H, Toyama K. Digital photography with flash and no-flash image pairs. ACM Trans Graph 2004;23(3):664–72.
[22] Yang Q. Recursive bilateral filtering. In: Proceedings of the European conference on computer vision. Springer; 2012. p. 399–413.
[23] Deng G. Guided wavelet shrinkage for edge-aware smoothing. IEEE Trans Image Process 2017;26(2):900–14.
[24] Kniefacz P, Kropatsch WG. Smooth and iteratively restore: a simple and fast edge-preserving smoothing model. In: Proceedings of the thirty-ninth annual workshop of the austrian association for pattern recognition (OAGM); 2015. p. 1–9. arXiv: 1505.06702.
[25] ESL G, Oliveira MM. Domain transform for edge-aware image and video processing. ACM Trans Graph 2011;30(4) 69:1–69:12.
[26] Eisemann E, Durand F. Flash photography enhancement via intrinsic relighting, 23; 2004. p. 673–8.
[27] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.
[28] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Phys D Nonlinear Phenom 1992;60(1–4):259–68.
[29] Chen J, Paris S, Durand F. Real-time edge-aware image processing with the bilateral grid, 26; 2007. p. 103.
[30] Lee H, Jeon J, Kim J, Lee S. Structure-texture decomposition of images with interval gradient, 36; 2017. p. 262–74.
[31] Bao L, Song Y, Yang Q, Yuan H, Wang G. Tree filtering: efficient structure-preserving smoothing with a minimum spanning tree. IEEE Trans Image Process 2014;23(2):555–69.
[32] DeCarlo D, Santella A. Stylization and abstraction of photographs. ACM Trans Graph 2002;21(3):769–76.
[33] Winnemöller H, Olsen SC, Gooch B. Real-time video abstraction. ACM Trans Graph 2006;25(3):1221–6.
[34] Lu C, Xu L, Jia J. Combining sketch and tone for pencil drawing production. In: Proceedings of the symposium on non-photorealistic animation and rendering; 2012. p. 65–73.
[35] Avidan S, Shamir A. Seam carving for content-aware image resizing. ACM Trans Graph 2007;26(3) 10:1–10:9.

View publication stats