

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/326543184>

# Structure extraction of images using anisotropic diffusion with directional second neighbour derivative operator

Article in *Multimedia Tools and Applications* · July 2018

DOI: 10.1007/s11042-018-6377-7

CITATIONS

0

READS

83

3 authors:



**Mukhalad Al-nasrawi**  
La Trobe University

5 PUBLICATIONS 5 CITATIONS

[SEE PROFILE](#)



**Guang Deng**  
La Trobe University

129 PUBLICATIONS 1,147 CITATIONS

[SEE PROFILE](#)



**Waseem Waheed**  
La Trobe University

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

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



Logarithmic Image Processing [View project](#)



Graph signal processing [View project](#)

# Structure extraction of images using anisotropic diffusion with directional second neighbour derivative operator

Mukhalad Al-nasrawi<sup>1,2</sup> · Guang Deng<sup>1</sup> · Waseem Waheed<sup>1</sup>

Received: 19 December 2017 / Revised: 4 June 2018 / Accepted: 3 July 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** The aim of structure extraction is to decompose an image into prominent structures and textures. In this paper, we present a new structure extraction method which has two main steps. First, high-frequency components due to the texture information in the original image are alleviated by a pre-smoothing filter. The result is then processed by a new anisotropic diffusion algorithm which uses a second neighbour derivative (SND) operator instead of the first neighbour derivative operator. We have demonstrated that the SND operator is better suited for applications such as texture smoothing. We have also presented a detailed study of the proposed method including the selection of the pre-smoothing filter, the number of iterations, and the scale parameter in the anisotropic diffusion algorithm. We have conducted experiments to compare the performance of the proposed method with those state-of-the-art structure extraction algorithms in a wide range of image editing applications such as: superpixel segmentation, texture transfer, contrast enhancement, and pencil drawing. We show that while the running speed of the proposed method is the fastest, its performance is competitive to other methods.

**Keywords** Structure extraction filter · Second neighbour derivative · Superpixel segmentation · Contrast enhancement · Texture transfer

---

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

Guang Deng  
d.deng@latrobe.edu.au

Waseem Waheed  
w.waheed@latrobe.edu.au

<sup>1</sup> Department of Engineering, La Trobe University, Bundoora, Victoria 3086, Australia

<sup>2</sup> Al-Musaib Technical College, Al-Furat Al-Awsat Technical University, Babylon, 51009, Iraq

# 1 Introduction

In computer vision systems, analysing a natural scene with large scale structures and small scale details is a challenging problem. Human visual system can easily handle these problems without image abstraction (small detail elimination). Image salient features are the basic data for human perception, rather than small scale details/textures [3]. In the real world, large structures, important edges, and textures are the principal features that built up the scene. Therefore, segmentation of the overall structure from the highly correlated background is an essential task in computer vision.

Edge-aware filters are essential tools in image decompositions. The key idea of using an edge-aware filter is to decompose an image into two components, large scale structures and small scale details. Existing filters, such as bilateral filter [34], guided filter [13],  $l_0$ -smoothing filter [37], BMA filters [8], domain transform filter [11], and semi-guided bilateral filter [33] tend to remove small details while preserving prominent structures. However, they are not explicitly designed to handle the structure extraction problem. Applying an edge-aware filter on a texture image may lead to texture preservation and undesirable results may arise. This is because some texture components are considered as distinctive edges by these filters.

To address this problem, researchers have studied and developed structure extraction techniques to handle the texture smoothing task. In the following, we only list some well-known algorithms in the structure-preserving and texture smoothing. They include: local extrema filter (LE) [32], rolling guidance filter (RGF) [40], and relative total variation (RTV) [38]. In the past few years, a considerable amount of works have been published in structure extraction filtering which include: region covariances filter (RCF) [16], bilateral texture filter (BTF) [7], Laplacian texture filter (LTF) [9], scale-aware texture smoothing (SATS) [14], zero crossing structure decomposition filter (ZCSD) [15] and reductive regression filter (RRF) [31]. In addition to structure preservation, these filters are widely used in many applications, such as seam carving [16], hatching and image abstraction [41], image seamless cloning and image vectorization [38], image detail enhancement [2], JPEG artifact removal [9], and image inpainting [36]. All aforementioned structure preservation filters can also be used in a very important application which is visual tracking, we refer interested readers to papers [18–22].

Inspired by the recent success of these filters and their valuable applications, the aim of this work is to explore a new structure-texture decomposition filter. The basic idea of this work is based on the classical anisotropic diffusion algorithm [27]. The proposed method is implemented in two main steps. In the first step, a pre-smoothing is utilized to mitigate the high contrast texture information. In the second step, the resultant image from the first step is then processed by the new anisotropic filter. The key difference between this work and the work based on the classical anisotropic diffusion is that in this work a new directional second neighbour derivative (SND) operator is used to obtain the image gradient, while in the classical anisotropic diffusion the first neighbour derivative operator (FND) is used in the gradient calculation. Since the SND operator is less sensitive to the texture information than the FND operator, the proposed filter produces better results in terms of suppressing high-frequency oscillation while preserving distinctive structures.

The organization of this paper is summarized as follows. In Section 2, we present a review of related texture smoothing algorithms. In Section 3, we provide the background information about anisotropic diffusion as well as definitions of the first and second neighbour derivative operators. In Section 4, we present the proposed texture smoothing method, including the selection of pre-smoothing filter, and a discussion of the optimal selection of

parameters from total variation point of view. In Section 5, we present experimental results and comparisons with state-of-the-art approaches in a wide range of applications including: texture smoothing, texture transfer, contrast enhancement, superpixel segmentation and pencil sketch. Conclusions are given in Section 6.

## 2 Related work

Many image decomposition techniques have been proposed to separate an image into a piecewise constant smooth base layer and a detail layer. In the following, we briefly review previous works in the area of structure extraction smoothing relating to the proposed filter.

Edge-preserving smoothing aims at removing fine-scale details and textures in an image without blurring large-scale object boundaries and sharp edges. The bilateral filter [34] is broadly utilized in noise removal and detail smoothing while preserving significant edges. Petschnigg et al. and Kopf et al. proposed an extension of the bilateral filter called joint/cross bilateral filter [17, 28]. In this filter, the range kernel distance is computed based on a guidance image to guide the smoothing process. The guided filter [13] is a local neighbourhood filter which is formulated based on a local linear model. Similar to the bilateral filter, the guided filter smooths an image by considering the content of the guidance image. It has a good edge preserving properties and it is more computationally efficient compared to the bilateral filter. Moreover, the guided filter produces images free of gradient reversal artifacts that the bilateral filter suffers from in detail enhancement applications. Zhang et al. [40] exploited the scale-aware theory to initially produce a smoothed image by eliminating small textures. It can be employed as a guidance image, then applied iteratively using the joint bilateral filter to recover the large edges from the input image. Unfortunately, their results may contain blocking artifacts due to coarse smoothing of the guidance image.

In addition to the local average-based filtering, there are several global optimization-based approaches, such as total variation (TV) [29], weighted least square (WLS) [10], and  $l_0$ -gradient minimization [37]. The total variation (TV) [29] is an edge-preserving regularization framework which has proven to produce good result on smoothing out the texture of irregular shapes by enforcing TV regularization constrains to retain the salient objects edges. The original formulation was extended by some studies reported in [4, 6, 39] which use different norms for both data fidelity and regularization term. They demonstrated that using robust norm could improve image structure-texture separation. Farbman et al. [10] introduced a weighted least square algorithm to overcome problems such as halo artifacts which appears due to coarse smoothing. The proposed global optimization function controls the level of smoothing for the image to be smoothed everywhere except at areas having distinctive gradient values. Xu et al. [37] proposed a robust smoothing filter algorithm which utilizes a sparse gradient measure. The regularization term of  $l_0$ -smoothing is designed to consider the number of image pixels having non-zero gradient as a regularization constrains. Consequently, the proposed approach can remove small gradient magnitudes of image details while retaining and even enhancing the distinctive objects in an image. All the aforementioned edge-aware filters depend on image intensities difference or the gradient magnitudes. Thus, the ongoing smoothing process generally can not recognize texture/small-scale details from the salient image features. As such, unsatisfactory structure-texture separation results might appear. Another limitation of the edge-preserving filter is that they are not explicitly design to handle the existence of high-frequency oscillatory components in an image. Figure 1 shows examples of using edge-preserving filters in texture-structure smoothing.



**Fig. 1** A comparison on using different global optimization-based edge-preserving filters in texture smoothing. **a** Input image. **b** TV [29] ( $\lambda = 0.4$ ) **c** WLS [10] ( $\lambda = 2, \alpha = 3$ ) **d**  $L_0$ -smoothing [37] ( $\lambda = 0.1, \kappa = 2$ )

Due to limitations of edge-aware smoothing algorithms, many researchers have studied and developed algorithms which have good ability to identify meaningful structures from textures in an image. Xu et al. [38] employed a non-linear relative total variation (RTV) metric which allows accurate identification of textures and smooths them out. They demonstrated that adopting such spatially varying metric can yield good structure-texture separation results. Karacan et al. [16] introduced a patch-based approach to decompose an image into large structural and small textural components. Region covariance descriptors which encode intensity and orientation are used as the base to compare the similarity between two patches on a Riemannian manifold. However, it may produce over smoothed results due to the overlapping patches. It is also computationally expensive. Cho et al. [7] introduced a patch shift mechanism into the joint bilateral filter to achieve structure-texture decomposition. They also proposed an improved measure of the relative total variation called the modified relative total variation to enhance the filtering performance. The joint bilateral filter is employed by incorporating results from patch shift in the range kernel to obtain the smoothed output.

### 3 Anisotropic diffusion and the SND operator

In this section, we first give a brief description of the anisotropic diffusion filter. Definitions of the first neighbour derivative (FND) and the second neighbour derivative (SND) are also provided. Then, a justification for using the SND operator in our structure extraction filter is presented. Finally, we generalize the 1-D SND operator to the 2-D SND operator.

#### 3.1 Anisotropic diffusion

Anisotropic diffusion is a non-linear diffusion technique used in image denoising via partial differential equation. Perona and Malik modified the classical diffusion filter to avoid the edge blurring problem in conventional diffusion filter by introducing stopping functions to stop the diffusion across significant boundaries [27]. The discrete version of the anisotropic diffusion equation is given by

$$I^{k+1}(x, y) = I^k(x, y) + \lambda \sum_{d=1}^8 S(|\nabla_d I^k(x, y)|) \nabla_d I^k(x, y) \quad (1)$$

where  $I^k(x, y)$  represents an image,  $k$  denotes iterations (discrete time steps), and  $x, y$  denote the coordinates of the pixel to be processed. The constant  $\lambda \in (0, 1/4]$  is a value that determines the diffusion rate.  $\nabla_d I^k(x, y)$  is the derivative of an image  $I$  in  $d^{\text{th}}$  direction,

$d = 1 \dots 8$  corresponds to an angle  $(d - 1)\pi/4$ . For example, we present the derivative operators in (2) and (3) for  $d = 1$  and  $d = 2$  which correspond to two directions  $0^\circ, 45^\circ$ , respectively.

$$\nabla_1 I(x, y) = I(x, y + 1) - I(x, y) \tag{2}$$

$$\nabla_2 I(x, y) = I(x - 1, y + 1) - I(x, y) \tag{3}$$

Their corresponding filter masks are shown in Fig. 2a and b.

The stopping function reduces diffusivity at distinctive boundaries. Two widely used stopping functions are:

$$S(|\nabla_d I(x, y)|) = \exp \left[ - \left( \frac{|\nabla_d I(x, y)|}{\sigma} \right)^2 \right] \tag{4}$$

and

$$S(|\nabla_d I(x, y)|) = \frac{1}{1 + \left( \frac{|\nabla_d I(x, y)|}{\sigma} \right)^2} \tag{5}$$

where  $\sigma$  is a scale parameter which has an important role in smoothing texture in an image. These functions satisfy the following requirement;  $S(|X|) \rightarrow 0$  when  $|X| \rightarrow \infty$  such that the diffusion stopped at edges.

### 3.2 The second neighbour derivative operator

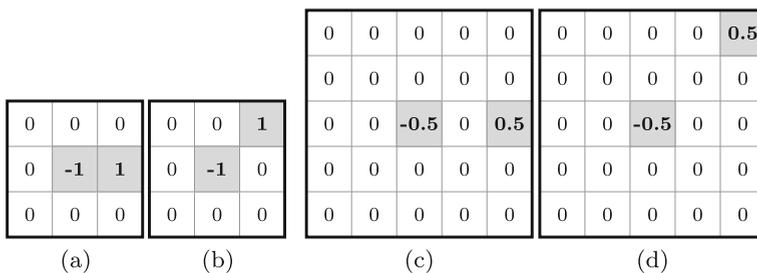
#### 3.2.1 Definition

In the discrete domain, a common way to define a derivative operator is the numerical differentiation. For simplicity, we only consider the 1-D signal. The discrete derivative at location  $n$  denoted by  $\nabla I(n)$  is defined as

$$\nabla I(n) = \frac{I(n + \tau) - I(n)}{\tau} \tag{6}$$

where  $\tau$  is a positive constant and  $\nabla I(n)$  measures the difference between two pixels. Mathematically, we can use  $\tau = 1$  which leads to the first neighbour difference (FND) operator

$$\nabla^{(1)} I(n) = I(n + 1) - I(n) = M * I \tag{7}$$



**Fig. 2** Filter masks corresponding to the FND and SND operators. **a** and **b** the FND masks. **c** and **d** the proposed SND masks

where the convolution mask is  $M = [-1 \ 1]$ . The FND operator is indeed used in anisotropic diffusion filter to determine the image gradient.

On the other hand, setting  $\tau = 2$ , we obtain the second neighbour difference (SND) operator

$$\nabla^{(2)}I(n) = \frac{I(n+2) - I(n)}{2} \quad (8)$$

The SND operator  $\nabla^{(2)}I(n)$  can be regarded as the smoothed version of  $\nabla^{(1)}I(n)$  and is given by:

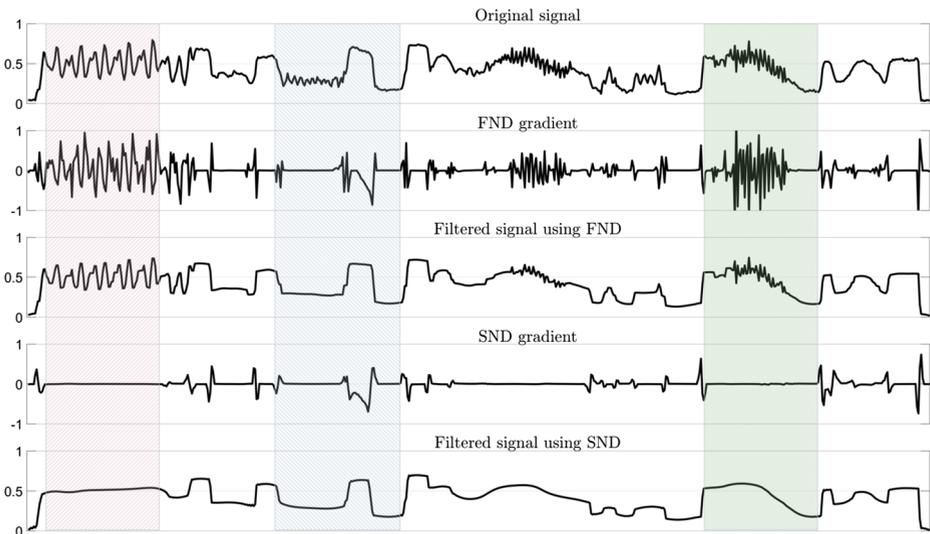
$$\nabla^{(2)}I(n) = h * \nabla^{(1)}I(n) \quad (9)$$

where  $h = [1 \ 1]/2$  which is the simplest low-pass filter. Intuitively, the derivative calculated using the SND can be regarded as the result of low-pass filtering the derivative calculated using the FND. The low-pass filter suppresses high-frequency components in  $\nabla^{(1)}I(n)$ . As such, anisotropic diffusion using the SND will be less sensitive to texture, leading to better smoothing of the texture component in an image.

### 3.2.2 Characteristics of the SND operator

In this section, we provide a justification why the SND operator is better than the FND in terms of texture removal. As mentioned in the previous section, this is because the SND operator can be considered as a low-pass filtered version of the FND operator. Therefore, the SND operator produces an output with less high-frequency oscillatory components which are due to the texture. By replacing the FND with the SND in the anisotropic diffusion algorithm, as we will show later, we observe an improvement in texture smoothing performance.

We demonstrate this property of the SND operator by conducting an experiment on a 1-D signal which has different structures to highlight the difference between the FND and the SND. Three regions are presented in Fig. 3: a texture region (red band), a strong structure edge (grey band), and a texture+shading (green band). For the signal with only texture



**Fig. 3** A comparison on the FND and the SND for filtering a 1-D signal with variable structures (highlighted in red, grey and green) and their corresponding smoothing results

information, as shown in Fig. 3 (red band), we can make the following observations. The FND operator leads to an output with high degree of fluctuations, while the SND operator is much less sensitive to the texture. On the other hand, for the part of the signal with prominent structure (grey band), the FND operator produces an output with oscillations near the edge, while the SND operator produces less oscillations. Such oscillations lead to jagged edges in texture smoothing. As shown in Fig. 3 (green band) which represents the signal with texture+shading, it is clearly seen that the response of the FND operator fails to suppress oscillations in the signal, whereas the SND operator smooth them out better. In summary, we can draw the following conclusion. The SND operator is more robust than the FND operator in the smoothing out of textures.

To further highlight the difference between using the two operators in texture smoothing, we conducted an experiment using the proposed algorithm (which is detailed in Section 4) to remove the texture in an image. Results are shown in Fig. 4. It can be clearly observed that the performance of the SND operator is better than the FND operator in terms of texture removal while maintaining significant objects' boundaries.

### 3.2.3 The 2-D SND operator

To extend 1-D SND into 2-D, we use the 2-D coordinates of the pixel. The 2-D SND operator is defined as

$$\nabla_d^{(2)} I(x, y) = \frac{I(x+i, y+j) - I(x, y)}{2} \quad (10)$$

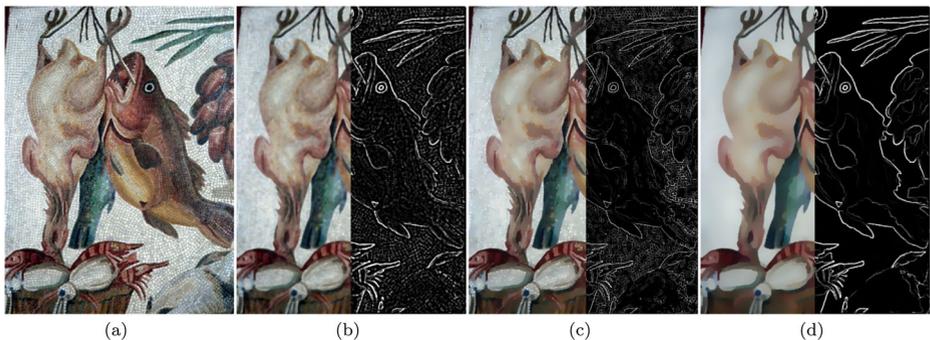
where  $i \in [-2, 0, 2]$ , and  $j \in [-2, 0, 2]$ . For simplicity, we only present two examples for the 2-D SND operators in two directions  $0^\circ$  and  $45^\circ$  which are given as

$$\nabla_1^{(2)} I(x, y) = \frac{I(x, y+2) - I(x, y)}{2} \quad (11)$$

and

$$\nabla_2^{(2)} I(x, y) = \frac{I(x-2, y+2) - I(x, y)}{2} \quad (12)$$

Their corresponding filter masks are shown in Fig. 2c and d.



**Fig. 4** A comparison on the effect of the FND and the SND map. **a** Input image. **b** Pre-smoothed image response and its gradient map. **c** Extracted structure and its gradient map using anisotropic diffusion with the FND operator. **d** Extracted structure and its gradient map using the proposed method with the SND operator

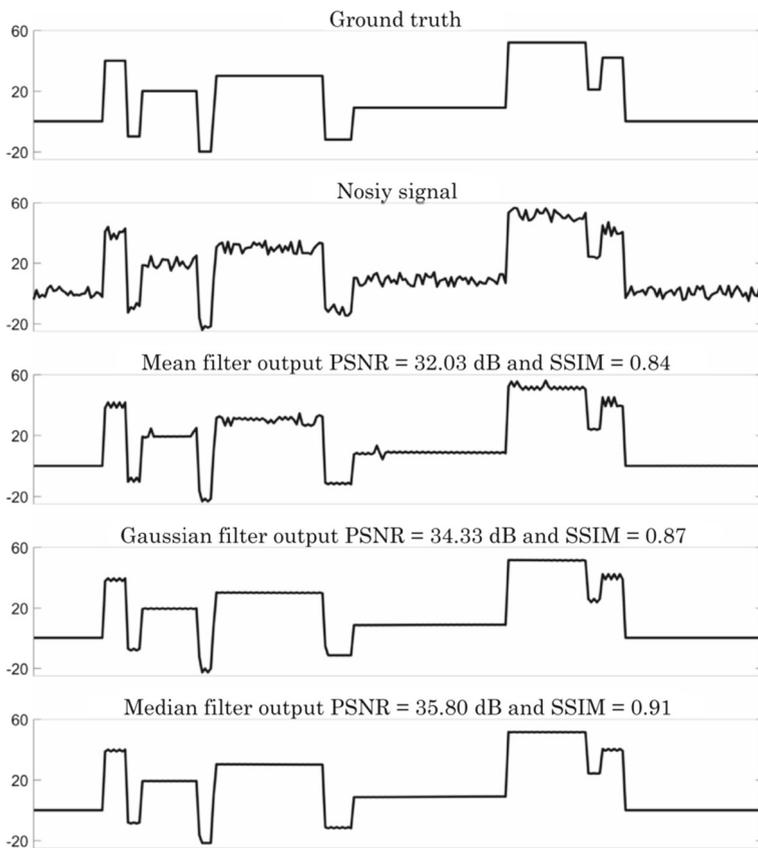
## 4 The proposed algorithm

### 4.1 Two key steps of the algorithm

There are two key steps in the proposed algorithm. An image is first pre-smoothed by a simple filter. The result is further smoothed by the anisotropic diffusion algorithm in which the derivative operations are implemented by using the SND operator. More specifically, referring to (1), the directional derivative operator  $\nabla_d$  is replaced by the SND operator  $\nabla_d^{(2)}$ . In the following, we provide a detailed discussion on the pre-smoothing filter, the stopping criterion for the anisotropic diffusion, and the parameter selection for the stopping function.

### 4.2 Pre-smoothing filter selection

In the proposed approach, a pre-processing step is used to help texture elimination in the second step. After the smoothing, homogeneous texture regions should become a relatively flat area (similar values) while salient edges should be preserved. It is clearly seen in Fig. 4b that applying a pre-smoothing filter is not sufficient to get rid of all texture information. A



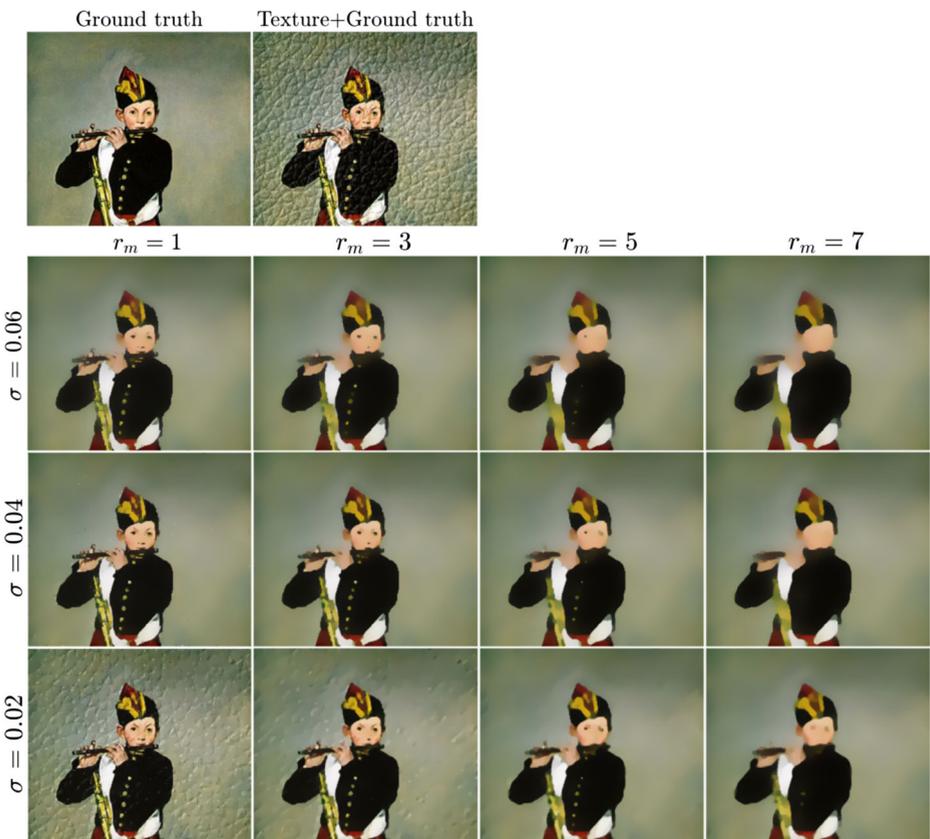
**Fig. 5** Results of the proposed method by using different filters as a pre-smoothed stage to obtain the texture smoothing response

diffusion process is needed to further suppressing the texture. The property of the Gaussian kernel can be used to deal with structure identification in images [25]. After Gaussian filtering, any structure with a scale smaller than the scaling parameter of the Gaussian kernel will disappear. Motivated by this property, we adopted similar idea to alleviate the effect of high-frequency textures.

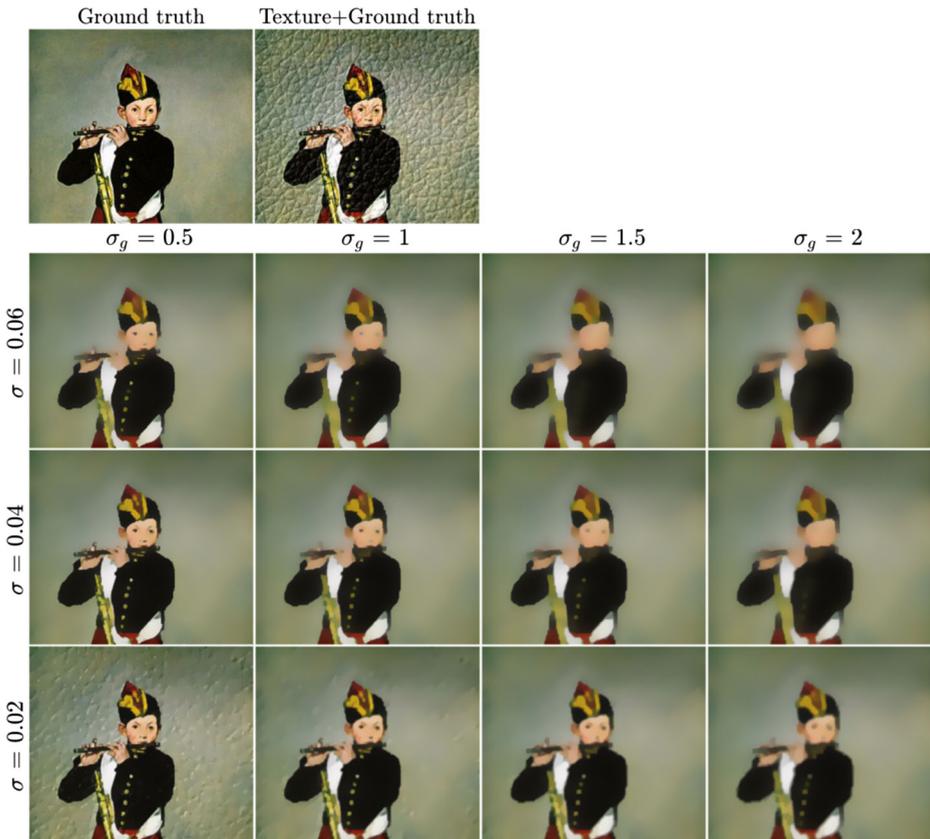
An interesting question is: what would be the best filter for obtaining the pre-smoothed image? To answer this question, we conducted experiments by employing various simple smoothing filters: median filter, Gaussian low-pass filter, and mean filter.

We artificially create an image which has different structures. This image is employed as a reference image. We then add noise of uniform distribution to simulate the high-frequency oscillatory components. The resulting noisy image  $I$  will be processed by one of the three mentioned filters. We tuned parameters of these three filters in such a way that they can achieve highest structure similarity index (SSIM) [35] and peak-signal-to-noise-ratio (PSNR). One row of the reference image, the noisy image, and the filtering results as well as their corresponding SSIM and PSNR, are shown in Fig. 5. We can clearly observed that the median filter can achieve the best SSIM and PSNR.

To study the performance of the proposed filter over images, we add texture components to a texture-free image which is employed as the ground truth, as illustrated in Fig. 6b.



**Fig. 6** Results of the proposed filter using different settings of parameters for the median filter as a pre-smoothing step. The neighbourhood size of the median filter is defined as ( $r_m$ )

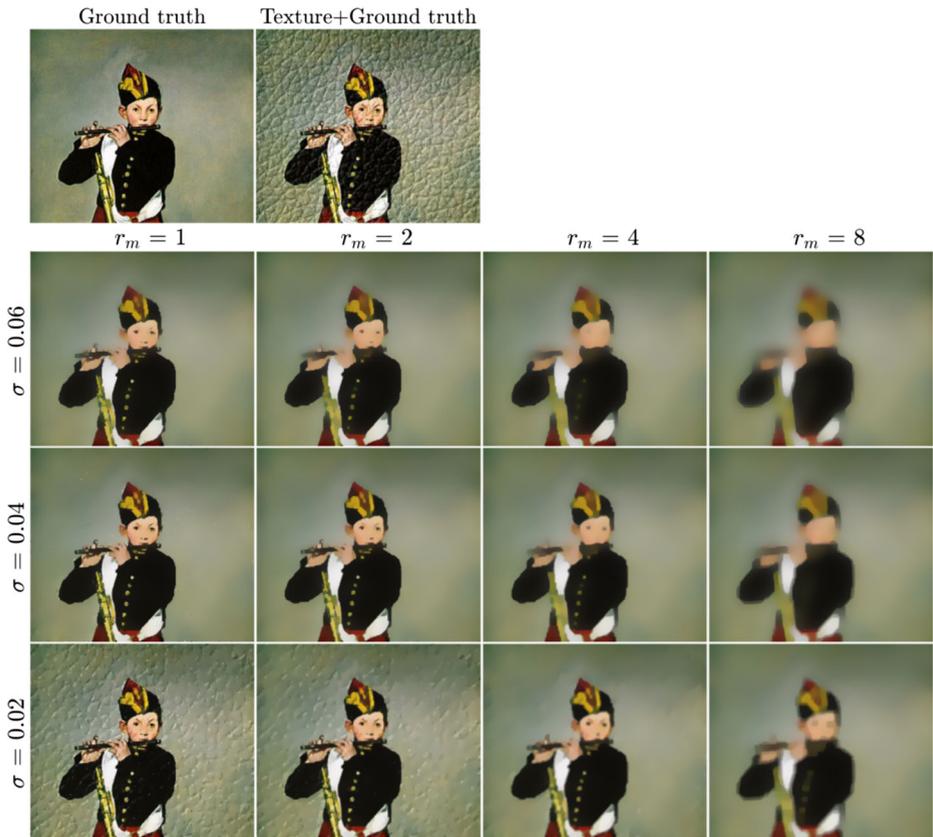


**Fig. 7** Results of the proposed filter using different settings of parameters for the Gaussian filter as a pre-smoothing step. The scale parameter of Gaussian kernel is defined as ( $\sigma_g$ )

The resulting textured image will be processed by pre-smoothing filters. The radius of the neighbourhood for the median filter and the mean filter is  $r_m$ . The Gaussian has a scale parameter  $\sigma_g$ . Each aforementioned parameters of the pre-smoothed filters as well as the diffusion coefficient ( $\sigma$ ) are adjusted to produce various smoothing effect, as shown in Figs. 6, 7, and 8 and their corresponding SSIM and PSNR are presented in Fig. 9. We can make the following observations. From Figs. 6, 7, and 8, we can see that the results of the proposed method exhibit robust response to perturbations in parameters settings of pre-smoothing filters and the diffusion coefficient. It is also observed that the PSNR and SSIM results of the proposed filter which uses the median filter and the mean filter are close to each other. These results are summarized in Table 1. Since the median filter has a proven ability in preserving the edges. Therefore, the median filter is adopted as the pre-smoothing filter.

### 4.3 Stopping criterion and optimal smoothing parameter selection

In the second step, anisotropic diffusion using the directional SND operator is applied to the pre-smoothed image. There are two parameters in this algorithm: the number of iterations



**Fig. 8** Results of the proposed filter using different settings of parameters for the mean filter as a pre-smoothing step. The neighbourhood size of the mean filter is defined as ( $r_m$ )

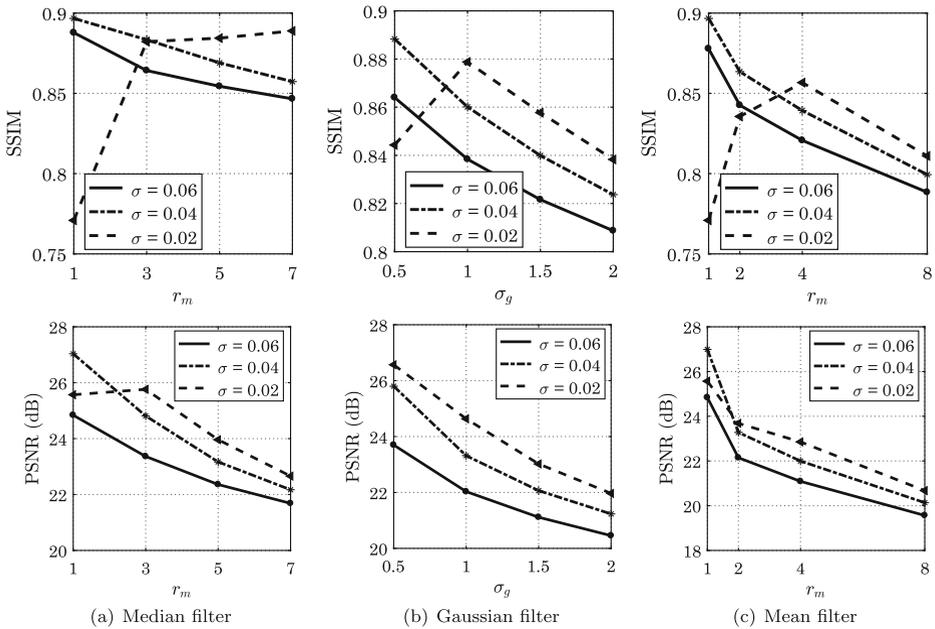
and the scale parameter in the stopping function. In this subsection, we study how to set these two parameters.

### 4.3.1 The number of iterations

It is well known that as the number of iterations  $k$  increases, the result of anisotropic diffusion eventually converges into a segmented piecewise constant image. Therefore, the resultant image might not achieve the desirable structure extraction goal. Hence, it is important to identify the number of iterations that provides an optimal trade-off between structure extraction and texture elimination.

In this subsection, we study the stopping criterion leading to a smoothed image which has salient object features similar to the original image. To develop the criterion, we use the cost function of total variation [29] which is defined as:

$$J(k) = \sum_{n=1}^N |I^0(n) - I^k(n)|^2 + \gamma \sum_{n=1}^N |\nabla I^k(n)| \tag{13}$$



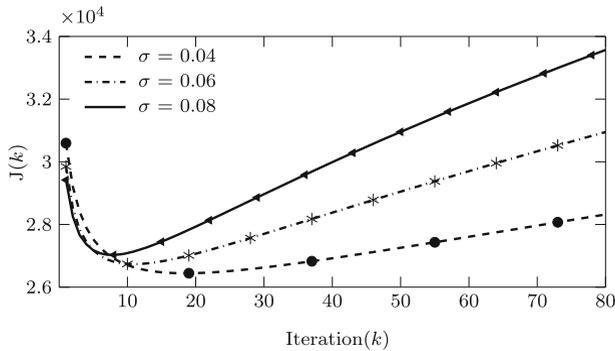
**Fig. 9** The effect of different parameters settings of the pre-smoothed image using median filter, Gaussian filter, and mean filter on the proposed method. The PSNR and SSIM measures were employed to evaluate the image quality in Figs. 6, 7, and 8. The first row and the second row show the SSIM and the PSNR of the image resulted from the proposed algorithm, respectively

where  $I^0(n)$  is the original image,  $I^k(n)$  is the smoothed image from the proposed filter at pixel  $n$ ,  $k$  is the number of iterations, and  $|\nabla I^k(n)|$  is the magnitude of the gradient. The cost function forces the output image to satisfy the following requirements to achieve structure preservation results: (1) it must be smoothed in the textured area except at the location of prominent boundaries, and (2) it must be close to the original image.

To develop a stopping criterion for the proposed anisotropic diffusion step, we can choose the desired output image at the  $k^{\text{th}}$  iteration that corresponding to the smallest cost. We have conducted experiments by applying the proposed approach on 10 different texture images. The cost function (13) is determined for the output of each iteration. The average costs  $J(k)$  over 10 images are shown in Fig. 10. To simplify our experiment, we fix  $\gamma = 0.45$  and determined the cost function for a range of values of the diffusion parameters  $\sigma$  (0.04, 0.06, and 0.08) and  $k$  (the number of iterations). We can clearly observe that the minimum cost occurs at three different iterations ( $k = 19$ ,  $k = 10$ , and  $k = 8$ ) corresponding to different settings of  $\sigma$ . An interesting question is: what would be the best number of iterations for texture smoothing? To answer this question, we can choose the output image produced in iteration  $k$  which is associated with the global minimal cost. This image will achieve the best

**Table 1** The best SSIM and PSNR of the images in Figs. 6, 7, and 8

Measure	Median ( $r_m = 1, \sigma = 0.04$ )	Gaussian ( $\sigma_g = 0.5, \sigma = 0.04$ )	Mean ( $r_m = 1, \sigma = 0.04$ )
SSIM	0.8969	0.8883	0.8964
PSNR (dB)	27.033	25.801	26.995



**Fig. 10** Cost function  $J(k)$  averaged over the 10 test texture images under different diffusion parameter settings of  $\sigma$  (0.04, 0.06, and 0.08, respectively) and the regularization parameter  $\gamma = 0.45$ . For each  $\sigma$ , minimum cost  $J(k)$  occurs at iterations (19, 10, and 8, respectively)

trade-off between texture smoothing and preserving significant information of the original image. We can observe that for the three settings of  $\sigma$ , the cost function is relatively small for  $k$  in the range from 10 to 20. Therefore, we can choose  $k$  in this range. It is also worth mentioning that the parameter  $\gamma$  is not required in the proposed filter, but is used in the calculation of the cost function to study the stopping criterion.

#### 4.3.2 The scale parameter of the stopping function

The scale parameter of the stopping function  $\sigma$  controls the amount of smoothing. Setting a large value of  $\sigma$ , both texture and important structural information will be filtered out. On the other hand, by setting a smaller value of  $\sigma$ , the texture components will be preserved. Therefore, the scale parameter  $\sigma$  controls a trade-off between texture smoothing and distinctive boundaries preservation. To determine the optimal value, as we discussed in the previous section, we set different values of  $\sigma$  to determine the cost. The particular value of  $\sigma$  which achieves the minimum cost is the optimal. From Fig. 10, we can see that minimum cost occurs in three different  $k$ 's values at (19, 10, 8), respectively. Therefore, we can choose  $\sigma$  at  $k = 19$  as the optimal value which means the optimal value is  $\sigma = 0.04$ .

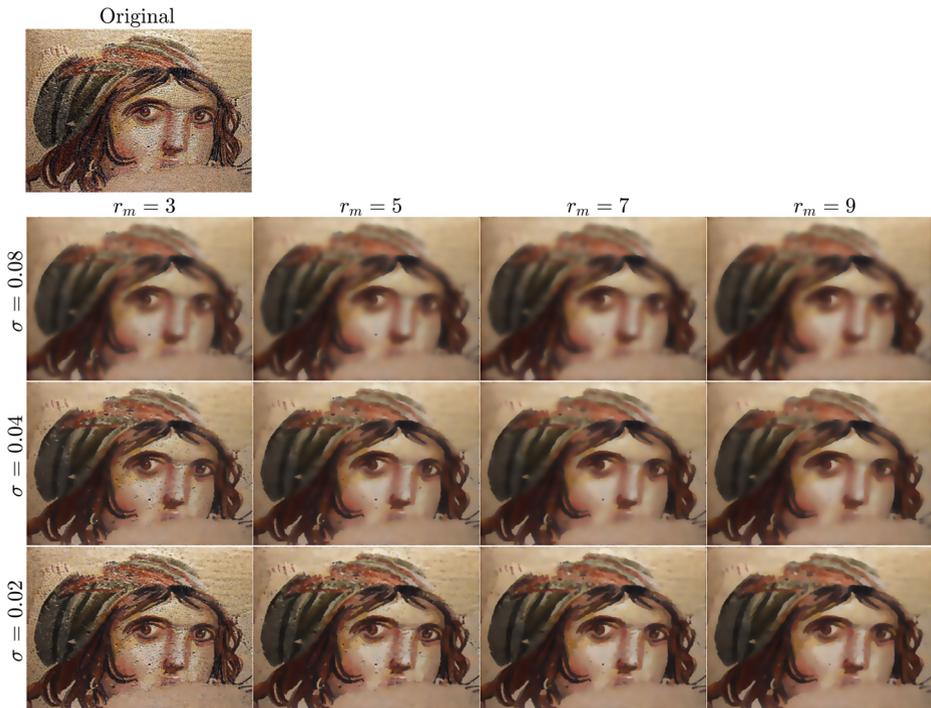
## 5 Applications and comparisons

All experiments were conducted using MATLAB on a PC with an Intel-i7 processor running at 3.2 GHZ and 32 GB RAM.

### 5.1 Experimental study of the proposed approach

#### 5.1.1 Effects of parameters settings

In Fig. 11, we present results of proposed filter under different settings of the two parameters. They are the scale parameter ( $\sigma$ ) of the non-linear stopping function that adjust the amount of diffusion and the size ( $r_m$ ) of neighbourhood for the median filter which yields



**Fig. 11** Effect of setting different values of  $\sigma$  and  $r_m$  on the proposed filter results for a texture image

the pre-smoothed image. We can see from Fig. 11 that when ( $\sigma$ ) is relatively large (corresponding to higher diffusion strength), major edges such as the face as well as small texture (sand in background) are blurred, while setting a smaller value of ( $r_m$ ) preserves small details such as tiles and dots in the face.

## 5.2 Applications

In this Section, we present smoothing results and various applications of the proposed filter and make a comparison with state-of-the-art algorithms.

### 5.2.1 Structure extraction

We demonstrate that the proposed filter is an effective tool for smoothing out texture information while preserving large objects and sharp edges. We compare the performance of our filter with a group of state-of-the-art texture smoothing algorithms such as relative total

**Table 2** Running time (seconds) of texture smoothing experiments in Figs. 4 and 12. Results of the proposed filter are listed in the last column

Example	Image size	RTV	RGF	BTF	STF	SDF	IGF	SAF	Proposed
Fig. 4	600×450	1.89	0.76	12.14	0.64	6.34	0.63	0.90	0.20
Fig. 12	675×900	4.04	1.02	43.82	2.24	25.0	1.71	3.50	0.63



**Fig. 12** A comparison of texture smoothing results for the “Fish” image with state-of-the-art algorithms. **a** Original image, **b** RTV ( $\sigma = 6, \lambda = 0.02, \epsilon = 0.02, N_{iter} = 4$ ), **c** RGF ( $\sigma_r = 0.08, \sigma_s = 5, N_{iter} = 5$ ), **d** BTF ( $k = 5, N_{iter} = 5$ ), **e** STF ( $\sigma_s = 5, \sigma_r = 0.05, \sigma = 0.02, N_{iter} = 5$ ), **f** SDF ( $\sigma = 2, \mu = 50, \nu = 400, \lambda = 200, steps = 1$ ), **g** IGF ( $\sigma = 4.3, \epsilon = 0.03^2$ ), **h** SAF ( $\sigma = 4, \sigma_r = 0.1, N_{iter} = 5$ ), **i** Our filter ( $\sigma = 0.04, \lambda = 0.25, N_{iter} = 19$ )

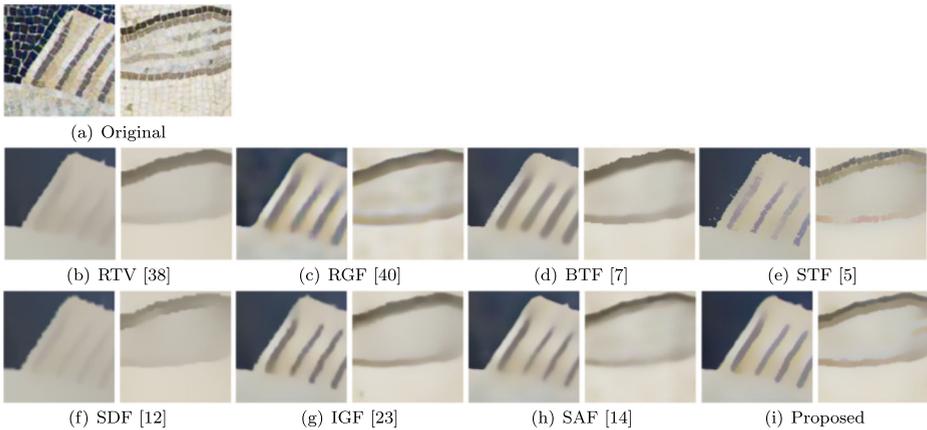
variation (RTV) [38], rolling guidance (RGF) [40], bilateral texture filter (BTF) [7], spanning tree filter (STF) [5], static and dynamic filter (SDF) [12], interval gradient filter (IGF) [23], and scale-aware texture smoothing (SAT) [14].

### 5.2.2 Running speed

Table 2 shows a comparison of the average running time of 10-run for our method versus a group of state-of-the-art algorithms in texture smoothing. Results show that the running time of the proposed filter is much faster than spanning tree filter (STF) [5], rolling guidance filter (RGF) [40], relative total variation (RTV) [38], bilateral texture filter (BTF) [7], static and dynamic filtering (SDF) [12], scale-aware filter (SAF) [14] and interval gradient filter (IGF) [23].

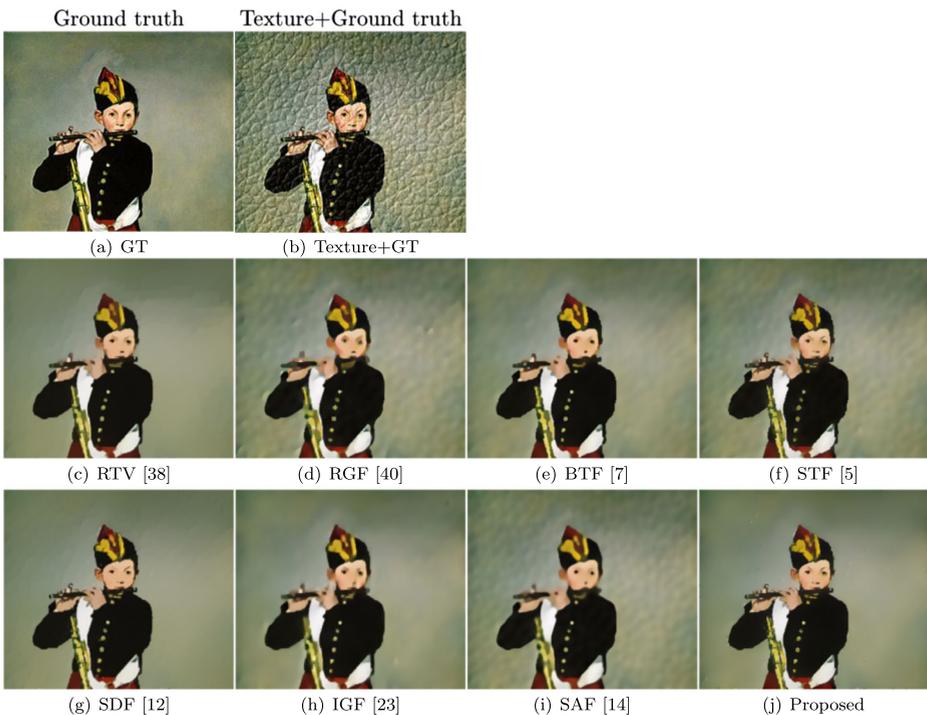
Results are shown in Figs. 12 and 13. It can be clearly seen that from the close up view in Fig. 13e the result of STF fails to smooth out texture around distinctive boundaries. On the other hand, while SDF and RTV tend to eliminate oscillatory textures, they also over smooth small details such as details in dorsal and pectoral fins of the fish, which are shown in Fig. 13b and f. We can also observe that the structure extraction results from RGF, BTF, IGF, SAF and our results are similar to each other except that our results have well preserved small strips in dorsal and pectoral fins. In terms of overall visual appearance, the contrast and colour of the image produced by the proposed filter appears to be more natural and similar to the original image.

We also provide a quantitative evaluation for the proposed algorithm. The quantitative evaluation has been conducted by adding a texture to a texture-free image which is utilized



**Fig. 13** Texture smoothing results of dorsal and pectoral fins parts of the fish

as the ground truth, as shown in Fig. 14a and b. We purposely tune parameters of each filter to yield an output such that it retains the meaningful structure and removes the texture element as much as possible. The structure similarity index (SSIM) [35] as well as the



**Fig. 14** A comparison of texture smoothing results for the “Flautist” image with a group 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 ( $\sigma = 0.04, \lambda = 0.25, N_{iter} = 20$ )

**Table 3** The objective evaluation of the texture smoothing results using SSIM and PSNR for images in Fig. 14. The SSIM and PSNR of the synthesized image (shown in Fig. 14b) are 0.5330 and 21.244 (dB), respectively. Results of the proposed approach are listed in the last column

Measure	RTV	RGF	BTF	STF	SDF	IGF	SAF	Proposed
SSIM	0.8914	0.8809	0.8941	0.8810	0.8720	0.8961	0.8851	0.8967
PSNR (dB)	26.322	24.788	25.463	24.730	23.640	25.771	24.925	27.033

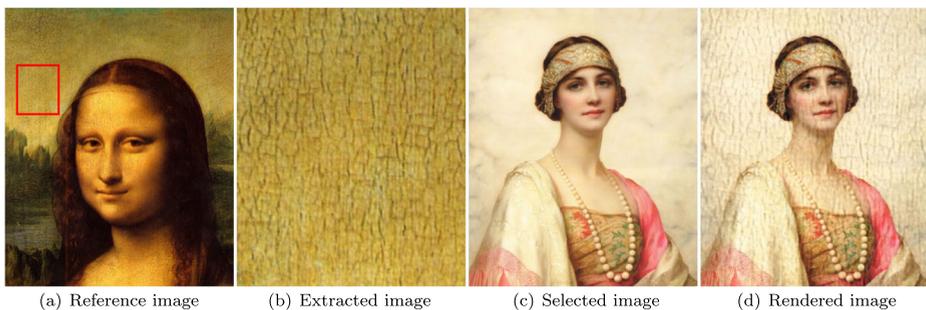
peak-signal-to-noise-ratio are adopted to measure the quantity of the texture smoothing performance. The subjective evaluation of the proposed algorithm is confirmed by the largest SSIM and PSNR associated with result produced by the proposed method. Table 3 shows that the SSIM and PSNR values of the image produced by our algorithm outperforms RTV, RGF, BTF, STF, SDF, IGF, and SAF.

### 5.2.3 Texture transfer

In this application, we utilize the proposed filter in texture transfer which is discussed in [30]. The key idea is to extract desirable textures from a reference image and to embed them into a target image. An example of this application is shown in Fig. 15 in which we transfer cracks from the Mona Lisa painting (the red rectangular in Fig. 15a) into the “An Elegant Beauty” painting Fig. 15c. We first select a region in the Mona Lisa image that contains suitable textures. We then apply the proposed filter to extract the texture layer from this selected region. Finally, the extracted textures are added to the target image. In order to preserve the colour of the target image, the texture transfer is implemented in the luminance component of the YUV colour space. As a result, the rendered image has a natural look with a historical sense conveyed by cracks.

### 5.2.4 Contrast enhancement

Contrast enhancement is a commonly used tool in image processing. In most contrast enhancement algorithms, the main assumption is that the input image is free of noise and compression artifacts. However, in practical situations, compression occurs due to the data streaming for example via the internet or/and captured by cameras. An example is presented in Fig. 16a. A direct application of a contrast enhancement algorithm on the image will



**Fig. 15** Texture in image **a** is extracted and presented in **b**. The texture is blended into the target image **c**. The result is shown in **d**



**Fig. 16** Image contrast enhancement using tone curve adjustment and texture smoothing. Structure extracted images in (b) and (f) of the original image in (a) are obtained by the method in [24] and the proposed filter, respectively. A direct boosted structure+texture on images (b) and (f) are presented in images (c) and (g). The only boosted structure on images (b) and (f) are shown in images (d) and (h). Enhanced results which are used the boosted structure and the processed texture by the method in [24] and the proposed approach are shown in images (e) and (i), respectively

not only improve its appearance but also unintentionally amplify noise/artifacts, especially JPEG compression artifacts, as shown in Fig. 16c and g. To overcome this problem, Yu et al. [24] proposed an approach to avoid boosting JPEG artifacts while improving image appearance. They adopted the texture-structure decomposition method to separate the original image into two layers: structure and texture layers which are processed separately. The structure layer is enhanced based on tone curve adjustment. Then, the texture layer is processed carefully to remove the compression artifacts. The final output is obtained by recombining the two layers. Since the proposed filter can extract image structure while smooth out texture/noise, it can also be used for image contrast enhancement application such as in [24]. We compare the performance of the proposed filter with the contrast enhancement approach in [24]. Results shown in Fig. 16e and i demonstrate that the performance of the two approaches are indeed similar except that our result is sharper, especially around the clouds in the sky.

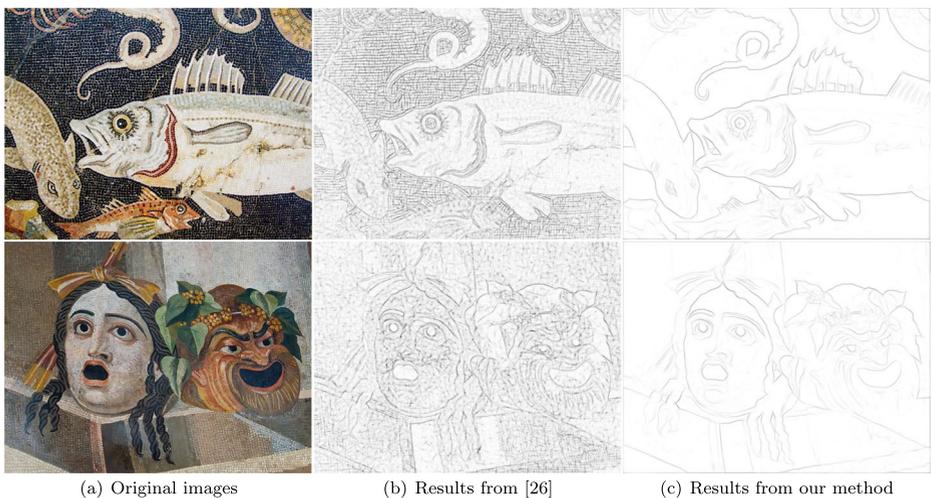
### 5.2.5 Superpixel segmentation

In recent years, superpixel segmentation has become an increasingly active research area in image processing and computer vision. In [1], Achanta et al. introduced a new approach which adopts the k-means clustering to yield efficient superpixel segmentation results. This approach is called simple linear iterative clustering (SLIC). Although the SLIC yields a uniform and compact superpixel segmentation, it does not handle a texture image well due to the existence of high-frequency oscillatory components. A direct application of the SLIC on the texture image leads to segments with irregular boundaries, as shown in Fig. 17b. Since the proposed filter has texture-structure decomposition capability, we can use it to



**Fig. 17** Results of superpixel segmentation. First column: original images. Second column: results of superpixel linear iterative clustering (SLIC). Third column: results of our method

improve superpixel segmentation results by mitigating the effect of high oscillatory textures. This will help produce better segmented regions. In Fig. 17, we illustrate a comparison of our superpixel segmentation with the state-of-the-art algorithm [1]. We can see that our results can produce a uniform segmented area. Due to the structure extraction properties of our filter, the boundaries of each segment has more regular patterns compared to the SLIC results.



**Fig. 18** Pencil sketching results

### 5.2.6 Image pencil sketch

Pencil sketch is a non-photorealistic rendering of a real image. This effect can be generated by adding gradients or edges to the smoothed version of the input image to produce a different visual impression, such as pencil-like or cartoon-like images. Lu et al. [26] generate pencil drawing effects by manipulating image gradients. In the texture image, applying the pencil sketch directly can not produce visually pleasing results because of the high correlation of textures with objects in the scene. To produce a better result, texture filtering can be applied to the original image as a pre-processing step and the result is used to obtain the pencil drawing. This leads to a more abstract image and shows the object of interest better. Figure 18 presents pencil sketching results using our filter and the algorithm in [26]. It can be clearly observed that the result from our filter has a clear and more abstract pencil drawing than method in [26].

## 6 Conclusion

In this paper, a new technique for structure extraction smoothing is introduced. The proposed filter has two main steps. A pre-smoothing filter is first applied to the input image, the result is then processed by a new anisotropic diffusion algorithm which uses the second neighbour derivative (SND) operator. A key difference between the classical anisotropic diffusion and the proposed filter is that the first neighbour derivative (FND) operator is replaced by the SND operator. We have compared the responses of the FND and SND to areas of an image with the texture and edges. We observed that while the SND is less sensitive to texture compared to the FND, the SND performs equally well in preserving edge information. These observations provide strong evidence for the justification of the use of the SND. We have also studied three key issues related to the proposed algorithm: the selection of the pre-smoothing filter, the number of iterations, and the scale parameter which controls the smoothing effect. While we have demonstrated that the performance of the proposed technique in a wide range of applications and conducted comparison with results of state-of-the-art methods, the proposed technique has more application in other computer vision tasks such as visual tracking. We will consider this in our future work<sup>1</sup>. We also show that the running speed of proposed technique is the fastest and its performance is among the best.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slc superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282
2. Al-nasrawi M, Deng G, Thai B (2018) Edge-aware smoothing through adaptive interpolation. *SIViP* 12(2):347–354
3. Arnheim R (1956) *Art and visual perception: a psychology of the creative eye*. University of California Press, Berkeley & Los Angeles

---

<sup>1</sup>One of the reviewers of this paper suggested this application.

4. Aujol JF, Gilboa G, Chan T, Osher S (2006) Structure-texture image decomposition—modeling, algorithms, and parameter selection. *Int J Comput Vis* 67(1):111–136
5. Bao L, Song Y, Yang Q, Yuan H, Wang G (2014) Tree filtering: efficient structure-preserving smoothing with a minimum spanning tree. *IEEE Trans Image Process* 23(2):555–569
6. Buades A, Le TM, Morel JM, Vese LA (2010) Fast cartoon + texture image filters. *IEEE Trans Image Process* 19(8):1978–1986
7. Cho H, Lee H, Kang H, Lee S (2014) Bilateral texture filtering. *ACM Trans Graph* 33(4):128:1–128:8
8. Deng G (2016) Edge-aware bma filters. *IEEE Trans Image Process* 25(1):439–454
9. Du H, Jin X, Willis PJ (2016) Two-level joint local Laplacian texture filtering. *Vis Comput* 32(12):1537–1548
10. Farbman Z, Fattal R, Lischinski D, Szeliski R (2008) Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans Graph* 27(3):67:1–10
11. Gastal ESL, Oliveira MM (2011) Domain transform for edge-aware image and video processing. *ACM Trans Graph* 30(4):69:1–69:12
12. Ham B, Cho M, Ponce J (2015) Robust image filtering using joint static and dynamic guidance. In: *Proc IEEE conference on computer vision and pattern recognition*, pp 4823–4831
13. He K, Sun J, Tang X (2013) Guided image filtering. *IEEE Trans Pattern Anal Mach Intell* 35(6):1397–1409
14. Jeon J, Lee H, Kang H, Lee S (2016) Scale-aware structure-preserving texture filtering. In: *Computer graphics forum*, vol 35, pp 77–86
15. Jiang X, Yao H, Liu S (2017) How many zero crossings? a method for structure-texture image decomposition. *Comput Graph* 68:129–141
16. Karacan L, Erdem E, Erdem A (2013) Structure-preserving image smoothing via region covariances. *ACM Trans Graph* 32(6):176:1–176:11
17. Kopf J, Cohen MF, Lischinski D, Uyttendaele M (2007) Joint bilateral upsampling. *ACM Trans Graph* 26(3):96–99
18. Lan X, Ma AJ, Yuen PC (2014) Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In: *Proc IEEE conference on computer vision and pattern recognition*, pp 1194–1201
19. Lan X, Ma AJ, Yuen PC, Chellappa R (2015) Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. *IEEE Trans Image Process* 24(12):5826–5841
20. Lan X, Yuen PC, Chellappa R (2017) Robust mil-based feature template learning for object tracking. In: *Proc conference on artificial intelligence (AAAI)*, pp 4118–4125
21. Lan X, Zhang S, Yuen PC (2016) Robust joint discriminative feature learning for visual tracking. In: *Proc international joint conference on artificial intelligence*, pp 3403–3410
22. Lan X, Zhang S, Yuen PC, Chellappa R (2018) Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker. *IEEE Trans Image Process* 27(4):2022–2037
23. Lee H, Jeon J, Kim J, Lee S (2017) Structure-texture decomposition of images with interval gradient. In: *Computer graphics forum*, vol 36, pp 262–274
24. Li Y, Guo F, Tan RT, Brown MS (2014) A contrast enhancement framework with jpeg artifacts suppression. In: *European conference on computer vision*, Springer, pp 174–188
25. Lindeberg T (1994) Scale-space theory: a basic tool for analyzing structures at different scales. *J Appl Stat* 21(1-2):225–270
26. Lu C, Xu L, Jia J (2012) Combining sketch and tone for pencil drawing production. In: *Proc symposium on non-photorealistic animation and rendering*, pp 65–73
27. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
28. Petschnigg G, Szeliski R, Agrawala M, Cohen M, Hoppe H, Toyama K (2004) Digital photography with flash and no-flash image pairs. *ACM Trans Graph* 23(3):664–672
29. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Phys D, Nonlinear Phenomena* 60(1-4):259–268
30. Su Z, Luo X, Deng Z, Liang Y, Ji Z (2013) Edge-preserving texture suppression filter based on joint filtering schemes. *IEEE Trans Multimedia* 15(3):535–548
31. Su Z, Zeng B, Miao J, Luo X, Yin B, Chen Q (2017) Relative reductive structure-aware regression filter. *J Comput Appl Math* 329:244–255
32. Subr K, Soler C, Durand F (2009) Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans Graph* 28(5):147:1–147:9
33. Thai B, Al-nasrawi M, Deng G, Su Z (2017) Semi-guided bilateral filter. *IET Image Process* 11(7):512–521
34. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: *6Th international conference on computer vision*, pp 839–846

35. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
36. Wu H, Xu D, Yuan G (2017) Region covariance based total variation optimization for structure-texture decomposition. *Multimedia Tools and Applications*
37. Xu L, Lu C, Xu Y, Jia J (2011) Image smoothing via  $L_0$ , gradient minimization. *ACM Trans Graph* 30(6):174:1–174:12
38. Xu L, Yan Q, Xia Y, Jia J (2012) Structure extraction from texture via relative total variation. *ACM Trans Graph* 31(6):139
39. Yin W, Goldfarb D, Osher S (2005) Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In: *Variational, geometric, and level set methods in computer vision*, Springer, pp 73–84
40. Zhang Q, Shen X, Xu L, Jia J (2014) Rolling guidance filter. In: *European conference on computer vision*, Springer, pp 815–830
41. Zhou Z, Wang B, Ma J (2018) Scale-aware edge-preserving image filtering via iterative global optimization. *IEEE Trans Multimedia* 20(6):1392–1405



**Mukhalad Al-nasrawi** received his B.S. and M.E. degrees in University of Baghdad, Iraq, in 2002 and 2008 respectively. And now he is pursuing his PhD degree in department of electronic engineering, La Trobe University, Australia. His current research interests include edge-aware smoothing, image processing, and computer vision.



**Guang Deng** is Reader and Associate Professor in the Department of Engineering, La Trobe University. His current research interests include Bayesian signal processing, lossless image compression and generalized linear image processing.



**Waseem Waheed** received his Bsc degree in Electrical engineering in 2009 from the Salahaddin University and the Masters degree in Telecommunications in 2014 from La Trobe University. He is currently a Ph.D. student in Electronic Engineering at La Trobe University. His research interests include signal processing, image processing, computer vision and Machine learning.