CrossMark

Two-level joint local laplacian texture filtering

Hui Du[1] · Xiaogang Jin[2] · Philip J. Willis[3]

Abstract Extracting the structure component from an image with textures is a challenging problem. This paper presents a novel structure-preserving texture-filtering approach based on the two-level local Laplacian filter. The new texture-filtering method is developed by introducing local Laplacian filters into the joint filtering. Our study shows that local Laplacian filters can also be used for texture smoothing by defining a special remapping function, which is closely related to joint bilateral filtering. This finding leads to a variant of the joint bilateral filter, which produces smooth edges while preserving color variations. Our filter shares similar advantages with the joint bilateral filter, such as being simple to implement and easy to understand. Experiments demonstrate that the new filter can produce satisfactory filtering results with the properties of texture smoothing, smooth edges, and edge shape preserving. We compare our method with the state-of-the-art methods to demonstrate its improve-ments, and apply this filter to a variety of image-editing applications.
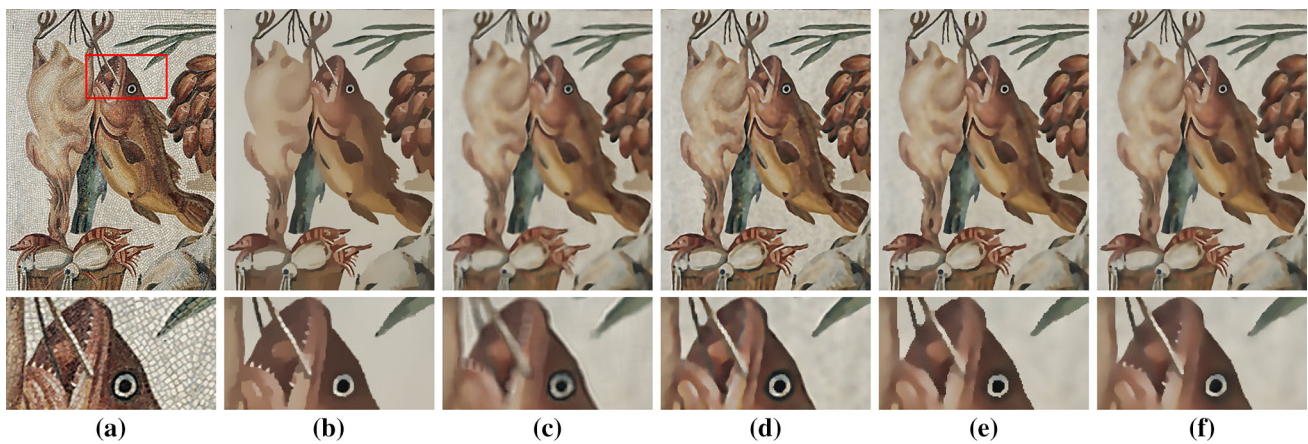
✉ Hui Du
  duhui@zjicm.edu.cn

✉ Xiaogang Jin
  jin@cad.zju.edu.cn

  Philip J. Willis
  P.J.Willis@bath.ac.uk

1 New Media College, Zhejiang University of Media and Communications, Hangzhou 310018, China

2 State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China

3 Department of Computer Science, University of Bath, Bath BA2 7AY, UK

1 Introduction

Edge-preserving image smoothing is very important and useful in many image applications, e.g., detail manipulation, abstraction, tone mapping, and image composition. It seeks to decompose an image into structure and detail components. Most existing work, such as the bilateral filter [25], weighted least squares filtering [8], L_0 smoothing [28] and local Laplacian filters [17], focused on separating structures from details while being edge preserving. These algorithms usually depend on gradient magnitude, pixel intensity and Laplacian pyramid coefficients to obtain satisfactory edge-preserving filtering results. While humans easily recognize the structures of an image which contains rich textures, it is difficult for a computer to automatically extract structures by texture smoothing. Most edge-preserving filtering methods do not explicitly address the structure-preserving texture-smoothing problem. Textures may be considered as strong edges if we apply these approaches straightforwardly to texture smoothing. As a result, textures are preserved instead of being smoothed and unsatisfactory results may arise.

To solve this challenging problem, previous methods [4,6, 12,23,24,29,31] try to suppress textures to obtain structure-preserving texture-smoothing results of images using differ-ent strategies, such as the weighted least squares optimization or the joint bilateral filtering schemes. These solutions can achieve good separation results. However, rich color varia-tions of the input may be flattened somewhat or structure edges of the input may be damaged by these filters. For

Fig. 1 Two-level joint local Laplacian texture filtering. From left to right: **a** input image; **b** Xu et al. [29]; **c** Karacan et al. [12]; **d** Zhang et al. [31]; **e** Cho et al. [6]; **f** our method. In terms of color variations, our method performs better than that of Xu et al. [29]. Our method can preserve partial structures of the fish teeth, which are removed in both results of Zhang et al. [31] and Cho et al. [6]. Zhang et al. [31] produces halo artifacts near strong structure edges which are reduced in our result. Compared with Cho et al. [6], our method produces smoother structure edges

example, after the bilateral texture filtering [6], smooth edges in the original image may show jagged transitions in the resulting structure component, as shown in Fig. 1e. When applying such structure images to image applications (e.g., detail enhancement), undesirable results may arise at structure edges. Thus, to achieve satisfactory results, smooth edges in the extracted structure component should be preserved, and the edge shapes should be as similar to those of the input as possible.

In this paper, we present a new structure-preserving texture-filtering method which can preserve edge shapes and color variations of the input image simultaneously. Our algorithm builds upon local Laplacian filters. While local Laplacian filters can yield high-quality edge-aware filtering results using a Laplacian pyramid, it is still unclear whether they are appropriate for strong texture smoothing. Inspired by Aubry et al. [1], we redefine the remapping function of local Laplacian filters by introducing a guiding image. With this in hand, we find that the two-level joint local Laplacian filter is related to the joint bilateral filter and shares similar advantages with the joint bilateral filter. Moreover, we design a variant of the joint bilateral filter, which essentially is a spatial interpolation-based method between the input image and the result of the joint bilateral filter. The spatial interpolation blending of this filter guarantees that the filtering results own the aforementioned properties, as shown in Fig. 1. Previous methods [6,23,31] have verified that the joint bilateral filtering schemes can effectively suppress textures. Therefore, our approach can also be employed for structure-preserving texture filtering.

The main contributions of our paper include:

– A joint filtering method based on the two-level local Laplacian filter is introduced. We introduce a guidance image into the definition of the remapping function of the two-level local Laplacian filter. This allows us to use this filter to achieve joint filtering.
– A new texture-filtering algorithm using the two-level joint local Laplacian filter is developed, which can produce smooth edges and preserve the original edge shapes as much as possible.

2 Related work

2.1 Edge-aware filtering

The average-based method is the main kind of edge-aware filter. From a computational point of view, these methods use weighted average schemes to eliminate details, including anisotropic diffusion [18], bilateral filtering [25], joint bilateral filtering [14,19], fast trilateral filtering [21], guided filtering [11], and geodesic filtering [7]. The bilateral filter [25] is one of the most widely employed edge-aware filters, since it preserves important image structures while removing fine-scale details. The output of the bilateral filter is a nonlinear weighted average of the input in the spatial neighborhood defined by the spatial distance and the difference of the range value simultaneously. The joint/cross filter [14,19] smoothes an image by using a guidance image to compute the difference of the range value. The guided image filter [11] is also a local neighborhood filter, like the bilateral filter, but is much faster and avoids gradient reversal that the bilateral filter can suffer from. However, the bilateral filter/the guided filter tends to blur over more edges as the scale of the extracted details increases and also introduces halo artifacts.

To overcome the limitation, Farbman et al. [8] introduced weighted least squares to achieve an edge-preserving multi-scale image decomposition. Furthermore, a few edge-aware filters, including edge-avoiding wavelets [9], local histogram filters [13], local Laplacian filters [17], domain transforms [10], L_0 gradient minimizations [28], and fast local Laplacian filters [1], were proposed to smooth fine-scale details while preserving image structures. Paris et al. [17] proposed a set of image filters called local Laplacian filters to achieve artifact-free edge-preserving results based on the Laplacian pyramid. Aubry et al. [1] further considered a wider class of remapping functions for local Laplacian filters and speeded up local Laplacian filters. Our work is closely related to these two. The main difference between our work and theirs is that we aim at structure-preserving texture smoothing which is not addressed in these two previous works. Recently, Shen et al. [22] used a boosting Laplacian pyramid to produce high-quality exposure fusion results from multiple exposure images.

In addition to the above-mentioned methods, median filters that compute a median [27] or weighted median [16,32] in local patches, and the local mode filter [26], can also remove high-contrast details of an image.

2.2 Structure-preserving texture smoothing

Rudin et al. [20] adopted total variation (TV) successfully to eliminate arbitrary textures of irregular shapes. Furthermore, some improved TV models [2,5,29,30] with different norms were proposed. Buades et al. [5] computed a local total variation and used a simple nonlinear filter pair to decompose an image into structure and texture components. Xu et al. [29] used relative total variation measures to achieve better separations between structure edges and textures. Subr et al. [24] defined details as oscillations between local extrema and smoothed out fine-scale oscillations. Karacan et al. [12] presented a patch-based algorithm that uses region covariances for structure-preserving image smoothing. However, it is time-consuming and may result in overblurred edges. Su et al. [23] applied an iterative asymmetric sampling degenerative scheme to suppress textures, and then used an edge correction operator and a joint bilateral filter to obtain edge-preserving texture suppression results.

Recently, Cho et al. [6] presented a bilateral texture filter based on the idea of patch shift that captures the texture information from the most representative texture patch clear of prominent structure edges. They constructed a guidance image via patch shift on each pixel and employed a joint bilateral filter to obtain the smoothed output. Zhang et al. [31] used a rolling guidance model to smooth the image. Bao et al. [4] proposed a weighted-average filter called the *tree filter* to achieve strong image smoothing with a minimum spanning tree. Since our method is a variant of the joint bilateral filter, our algorithm can also be classified as a joint bilateral filter.

3 Our algorithm

We start from simply summarizing the basic principle of local Laplacian filters and then introduce our algorithm in detail. Given a 2D input image I, local Laplacian filters use a pixel-wise filter with the remapping function $rp(i)$ to generate a transformed image, then compute the Laplacian pyramid L of this image and use the coefficient in the pyramid L as the value of the output pyramid coefficient. Finally, the output pyramid is collapsed to obtain the result.

3.1 Two-level joint local Laplacian filter

Aubry et al. [1] defined the space of remapping functions for local Laplacian filters in the form:

$$r(i) = i - (i - g)f(i - g), \qquad (1)$$

where f is a continuous function, i represents the pixel of the input I and g denotes the coefficient of the Gaussian pyramid. Remapping functions are the same as those of Paris et al. [17] when $f(i - g) = (i - rp(i))/(i - g)$ where rp denotes the remapping functions defined by Paris et al. [17].

We modify the space of remapping functions r by introducing a guidance image M:

$$r(i) = i - (i - g)f(i_m - g_m), \qquad (2)$$

where $g_m = G_l[M](x, y)$ denotes the coefficient of the Gaussian pyramid at level l and position (x, y) of the guidance image M. We will discuss the definition of the guidance image M in Sect. 3.3.

For the two-level filter, we need to compute the two levels $L_0[J]$ and $L_1[J]$ of the Laplacian pyramid of the output image J. We assume the residual $L_1[J]$ of the output remains unprocessed resembling Aubry et al. [1], that is, $L_1[J] = L_1[I]$. The 0th level of the Laplacian pyramid of the output image J is computed as the difference between the transformed image $r(I)$ and the corresponding low pass-filtered image. That is,

$$L_0[J](p) = r(I_p) - [\overline{G}_{\sigma_p} * r(I)](p), \qquad (3)$$

where p denotes a pixel at the position (x, y), $\overline{G}_{\sigma_p}$ is the normalized Gaussian kernel of variance σ_p^2 used to build the pyramid and $*$ indicates the convolution operator. For the finest level of the pyramid, we have $L_0[I] = I - \overline{G}_{\sigma_p} * I$, $g = I_p$ and $g_m = M_p$. Substituting our new definition of the

remapping function into the above expression, we reach the following equation after rearranging:

$$L_0[J](p) = L_0[I](p) + [\overline{G}_{\sigma_p} * (I - I_p) f(M - M_p)](p). \tag{4}$$

We obtain the two-level joint local filter by upsampling the residual, adding it to both sides of the above formula and expanding the convolution:

$$J_p = I_p + \sum_q \overline{G}_{\sigma_p}(q - p) f(M_q - M_p)(I_q - I_p), \tag{5}$$

where $q \in \Omega$ denotes the pixels in the local window Ω centered in the pixel p. The above formula shows that we can achieve joint filtering in a similar spirit to the joint bilateral filter, since the second term of the output J is a weighted average of the pixels in the spatial neighborhood using $\overline{G}_{\sigma_p}$ and the function f of the guidance image M.

3.2 The proposed filter

In this section we first discuss the relationship between the two-level joint local Laplacian filter and the original joint bilateral filter. The definition of the original joint bilateral filter (JBF) is expressed as:

$$\mathrm{JBF}_p = \frac{1}{W_p} \sum_q G_{\sigma_s}(q - p) G_{\sigma_r}(M_q - M_p) I_q,$$
$$W_p = \sum_q G_{\sigma_s}(q - p) G_{\sigma_r}(M_q - M_p), \tag{6}$$

where W_p is the normalization term and M is the guidance image, G_{σ_s} and G_{σ_r} denote the spatial and the range kernels, and they are typically Gaussian functions $G_\sigma(p) = \exp(-p^2/2\sigma^2)$, respectively.

Considering the symmetry of the Gaussian kernel and the fact that the weights sum up to 1, the above formula can be rewritten as:

$$\mathrm{JBF}_p = I_p + \frac{1}{W_p} \sum_q G_{\sigma_s}(q - p) G_{\sigma_r}(M_q - M_p)(I_q - I_p). \tag{7}$$

Comparing Eq. 5 with Eq. 7, we see that the two-level joint local Laplacian filter shares similarities with the original JBF. If we define $\overline{G}_{\sigma_p}$ as the spatial weight and f as G_{σ_r}, the only difference between the filters is that the weights are not normalized by $\frac{1}{W_p}$ in the two-level joint local Laplacian filter. This relationship leads to a new joint filter using $\sigma_p = \sigma_s$.

$$J_p = I_p + \sum_q \overline{G}_{\sigma_s}(q - p) G_{\sigma_r}(M_q - M_p)(I_q - I_p). \tag{8}$$
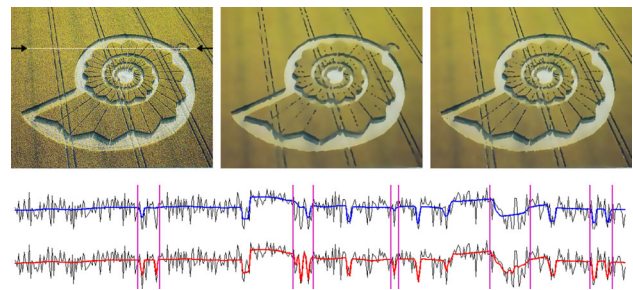
Fig. 2 Comparison of our new filter and the original joint bilateral filter with the same guidance image. From *left* to *right* of the *top row*: input image, the result of the original joint bilateral filter, the result of our filter. The *middle* and the *bottom rows* show the results of the original joint bilateral filter and our new filter of the input 1D signal at the 94th line, respectively. Compared to the original joint bilateral filter, our filter may preserve structure edges better

Figure 2 shows the difference between this new filter and the original joint bilateral filter. The filters have almost the same results within pure texture regions. The main difference between the two filters appears at strong edges. The original joint bilateral filter smooths some strong edges more aggressively than our new filter. When a pixel p is significantly different from its neighbors, i.e., at strong edges, the normalization factor $\sum_q \overline{G}_{\sigma_s}(q - p) G_{\sigma_r}(M_q - M_p)$ is small and the output of this new filter is closer to the original input. That is, this new filter performs better in terms of preserving structure edges of an image.
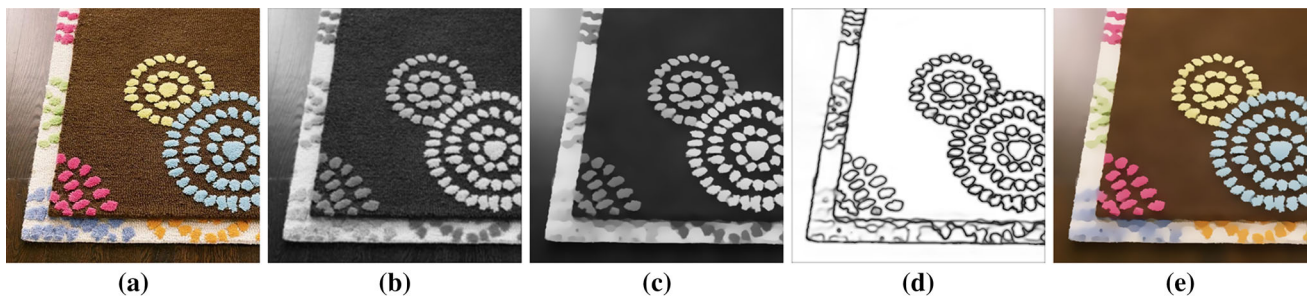
3.3 Guidance image

We now go back to the goal of texture smoothing. Akin to Subr et al. [24], we define textures as fine-scale spatial oscillations of signals. Cho et al. [6] showed that the joint bilateral filter can be effectively applied on structure-preserving texture smoothing by substituting a special texture description image, namely the guidance image, in the range kernel. To guide texture smoothing, the guidance image M should have similar values in a homogeneous texture region and distinguishable values across salient edges. That is, M should represent the approximate structures of the input image.

The computer vision community developed scale space methods to deal with structure identification in images, and the uniqueness of the Gaussian kernel for scale space filtering has been explored [3]. Here, we consider single-scale space and compute the guidance image M by two steps. Specifically, we first apply the Gaussian filter with the standard deviation σ_s to the input image I, i.e.,
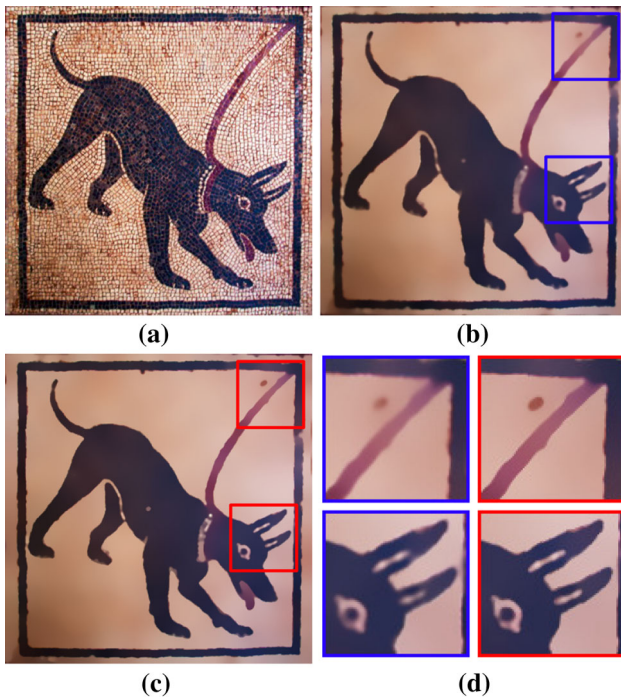
$$D = I * G_{\sigma_s}. \tag{9}$$

After this step, details are eliminated. As a by-product, however, Gaussian filtering often blurs the large-scale struc-

Fig. 3 Overview and intermediate images of our approach. From left to right: **a** input image, **b** the initial blurred image D, **c** the final guidance image M, **d** the interpolation coefficient map μ, and **e** our filtering result J

Fig. 4 The effect of the guidance image. **b** and **c** are the filtering results using guidance images D and M, respectively, **a** input image, **b** Gaussian result, **c** our result, **d** close-ups

tures of the image while eliminating the small-scale structures, and leads to blurring structure edges. Blurred results will be produced if we use D as the guidance image M directly, as shown in Fig. 4. To represent the clear structures of the image, we then employ the original joint bilateral filter with D as the guidance image, to recover edges of the large-scale structures. We finally define our guidance image M as the output of the joint bilateral filter. That is, the guidance image M is defined as follows:

$$M = \text{JBF}(I, D), \tag{10}$$

where D is the result of the Gaussian filtering. Figure 3c shows the guidance image M via the joint bilateral filtering,

which effectively restores structure edges from the initial blurred image D (Fig. 3b).

### 3.4 Texture smoothing via interpolation

By now, we have obtained our texture-smoothing filter via the two-level joint local Laplacian filter. We will further prove that this new filter is equivalent to a spatially varying interpolation between the input image and the result of the joint bilateral filter. We find that this interpolation-based texture-smoothing scheme is simple and intuitive.

We first define μ_p as $\sum_q \overline{G}_{\sigma_s}(q - p)G_{\sigma_r}(M_q - M_p)$. Since $\overline{G}_{\sigma_s} = \frac{1}{\sqrt{2\pi\sigma_s^2}}G_{\sigma_s}$ is a normalized Gaussian kernel and $\sum_q G_{\sigma_s}(q - p)G_{\sigma_r}(M_q - M_p)(I_q - I_p) = W_p(\text{JBF}_p - I_p)$ according to the expression of the original JBF, we can rewrite the formula of our new joint filter and substitute μ_p into the expression. Finally, we obtain the following formula:

$$
\begin{aligned}
J_p &= I_p + \sum_q \overline{G}_{\sigma_s}(q - p)G_{\sigma_r}(M_q - M_p)(I_q - I_p) \\
&= I_p + \frac{1}{\sqrt{2\pi\sigma_s^2}}\sum_q G_{\sigma_s}(q-p)G_{\sigma_r}(M_q-M_p)(I_q - I_p) \\
&= I_p + \frac{1}{\sqrt{2\pi\sigma_s^2}}W_p(\text{JBF}_p - I_p) \\
&= I_p + \mu_p(\text{JBF}_p - I_p) \\
&= (1 - \mu_p)I_p + \mu_p\text{JBF}_p, \tag{11}
\end{aligned}
$$

where μ_p is the interpolation coefficient. In uniform regions of the guidance image M, the coefficient μ_p is large because $M_q - M_p$ is close to zero. On the contrary, in discontinuity regions of the guidance image, the coefficient μ_p is small around edges and this leads our new joint filter having a weaker filtering effect. This also explains why our new joint filter can effectively preserve the original edge shapes of the input.

In experiments, we found that a single iteration of the proposed joint filter may not be sufficient for obtaining a
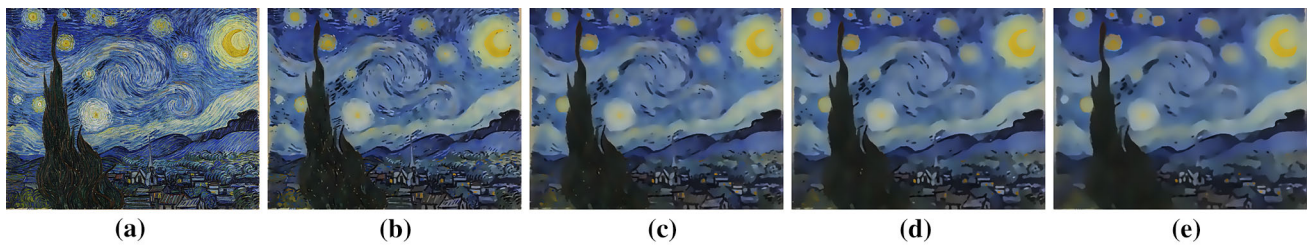
Fig. 5 Different filtering results by progressively increasing the scale σ_s. In this example, we set $\sigma_r = 0.04$ and $N_{\text{iter}} = 5$. **a** Input image, **b** $\sigma_s = 3$, **c** $\sigma_s = 4$, **d** $\sigma_s = 5$, **e** $\sigma_s = 6$

Algorithm 1 Joint Local Laplacian Texture Filtering

Require: Image I, N_{iter}, σ_s and σ_r
Ensure: Filtered image J
1: **for** $k = 1 \rightarrow N_{iter}$ **do**
2:      $D \leftarrow I * G_{\sigma_s}$ ▷ Gaussian blurring of I
3:      $M \leftarrow JointBilateralFiltering(I, D)$
4:      $\mu \leftarrow \sum_q \overline{G}_{\sigma_s}(q - p)G_{\sigma_r}(M_q - M_p)$
5:      $JBF \leftarrow JointBilateralFiltering(I, M, \sigma_s, \sigma_r)$
6:      $J \leftarrow (1 - \mu)I + \mu JBF$
7:      $I \leftarrow J$
8: **end for**

---

desired texture-smoothing result. Therefore, we apply our filter in an iterative manner resembling the method of Cho et al. [6]. Algorithm 1 depicts our final algorithm and Fig. 3 visualizes the flow of our approach with intermediate images.

4 Analysis

4.1 Parameters

In our implementation, all pixel values are firstly normalized to the range [0, 1] and the spatial kernel half-size of the joint bilateral filter is set to σ_s. Our algorithm can remove textures of different scales by varying the value of σ_s, since the spatial standard deviation σ_s indicates the scale of textures to be suppressed (Fig. 5). Because there are no explicit measures to clearly distinguish scales of the main structures and textures of an image, the user specifies the value of σ_s directly to generate the final result. We find that satisfactory results can be achieved when $\sigma_s \in \{3, 4, 5, 6, 7\}$, $\sigma_r \in \{0.04, 0.05, 0.055\}$ and $N_{\text{iter}} \in \{5, 6, 7, 8, 9, 10\}$. Alternatively, we may also tweak the range standard deviation σ_r^k for the kth iteration by setting $\sigma_r^k = \sigma_r/\min(k, \lambda)$, where σ_r denotes the initial range standard deviation and λ is set to 3 or 4 in our experiments. This enables our algorithm to restore the original color variations of the object surface, as shown in Fig. 1.

4.2 Color image

For a color image, we first convert it to grayscale and compute the guidance image M from this grayscale image. Then,
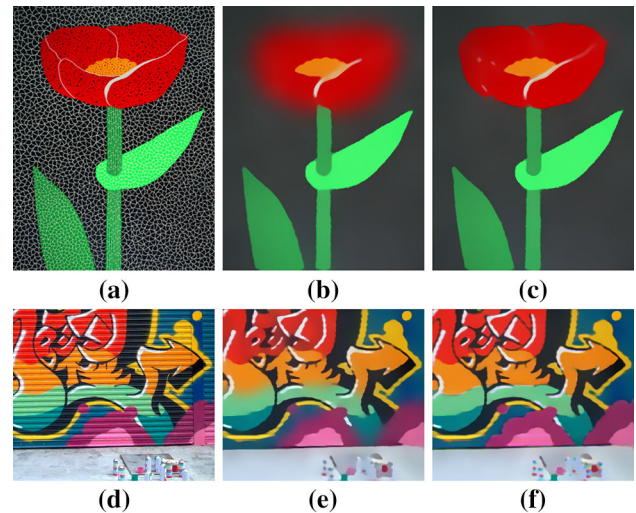
Fig. 6 Color image filtering. **a** Input image. **b** The result using the $rgb2gray()$ function of MATLAB. **c** The result using contrast-preserving decolorization [15]. **d** Input image. **e** The result using grayscale guidance image. **f** The result using color guidance image. *Color guidance image* preserves structures of an image better

our proposed filter is applied on each color channel using M to obtain the filtering result. We employ the contrast-preserving decolorization algorithm [15], which aims to preserve the original color contrast as much as possible, to obtain grayscale images for most examples in this paper. Compared to simply computing a grayscale one via a linear combination of R, G, and B channels with fixed weight [e.g., the $rgb2gray()$ function in Matlab], using contrast-preserving decolorization usually produces better filtering results, although it adds some computation cost, as shown in the first row of Fig. 6.

Color-to-gray conversion inevitably results in information loss. Alternatively, we may use full color information to compute the guidance image. We first apply the Gaussian filter and the joint bilateral filter in each channel of the color image to construct the color guidance image M. The interpolation coefficient of each channel is then computed via the corresponding channel of the guidance image M. Finally, we apply the joint bilateral filter by using the color M, and blend the input with the result of the joint bilateral filter using the inter-

polation coefficient to obtain the final result. The second row of Fig. 6 shows an example.

5 Results and applications

In this section, we show the results produced by our algorithm and compare our approach with the state-of-the-art methods [4,6,12,29,31]. We carefully tuned the parameters to generate results from these methods. The experiments show that our method produces satisfactory filtering results and is comparable to previous methods on structure-preserving texture smoothing.

Figures 1, 7 and 8 show comparisons between our method and previous methods [4,6,12,29,31]. In Fig. 1, our method can effectively remove textures in both small and large regions. By gradually decreasing the value of σ_r, our method performs better in terms of color variations than the method of Xu et al. [29]. Compared with the method of Karacan et al. [12], our method effectively preserves structure edges, which are overblurred in their result. Our method can even

preserve partial structures such as the fish teeth, which are removed by Zhang et al. [31] and Cho et al. [6]. The method of Cho et al. [6] damages the original smoothness of structure edges and leads to jagged transitions. In contrast, our method can yield smooth edges and preserve the original edge shapes well.

Figure 7 shows that both our method and Xu et al.'s method [29] can preserve the original structure edge shapes, which are altered in the results of Zhang et al. [31] and Cho et al. [6]. But with the method of Xu et al. [29], it is difficult to eliminate textures located near a structure edge, as pointed out in Cho et al. [6] and it may mistake parts of structures for textures (see Fig. 9). As shown in Fig. 8, Bao et al.'s approach [4] effectively removes textures in pure texture regions, but may retain textures located near prominent structure edges. The approach by Karacan et al. [12] may oversmooth structures while removing texture effectively. Our method can separate image structures and textures more cleanly. More results are supplied in the supplementary material.
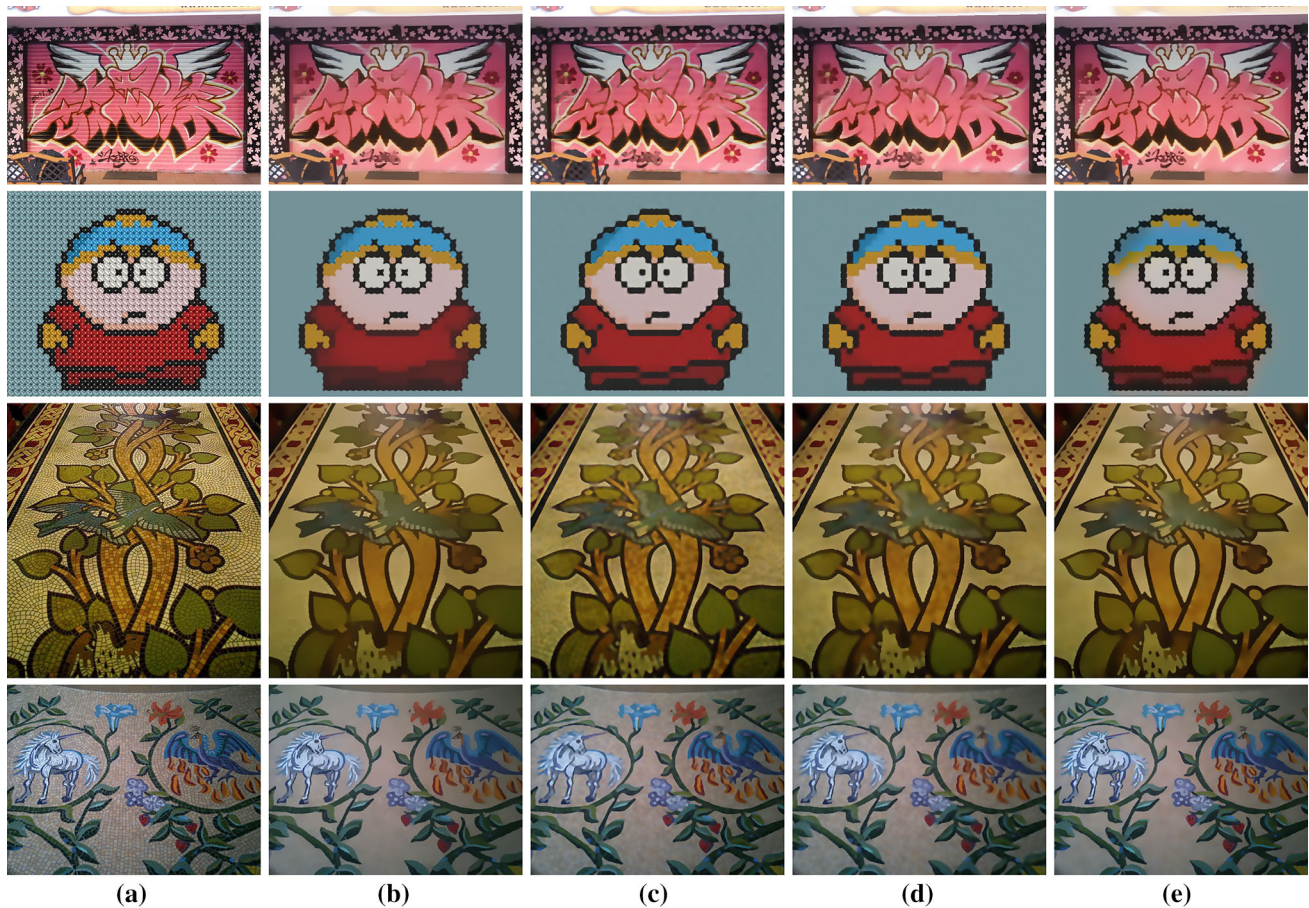

Fig. 7 Comparison with the state-of-the-art methods. The interpolation scheme ensures that our filter can effectively preserve the original edge shapes. Our method performs better in terms of preserving small objects while eliminating textures (*the bottom row*). The parameter values of the state-of-the-art methods are listed in the supplementary material. **a** Input images, **b** Xu et al. [29], **c** Zhang et al. [31], **d** Cho et al. [6], **e** Our method
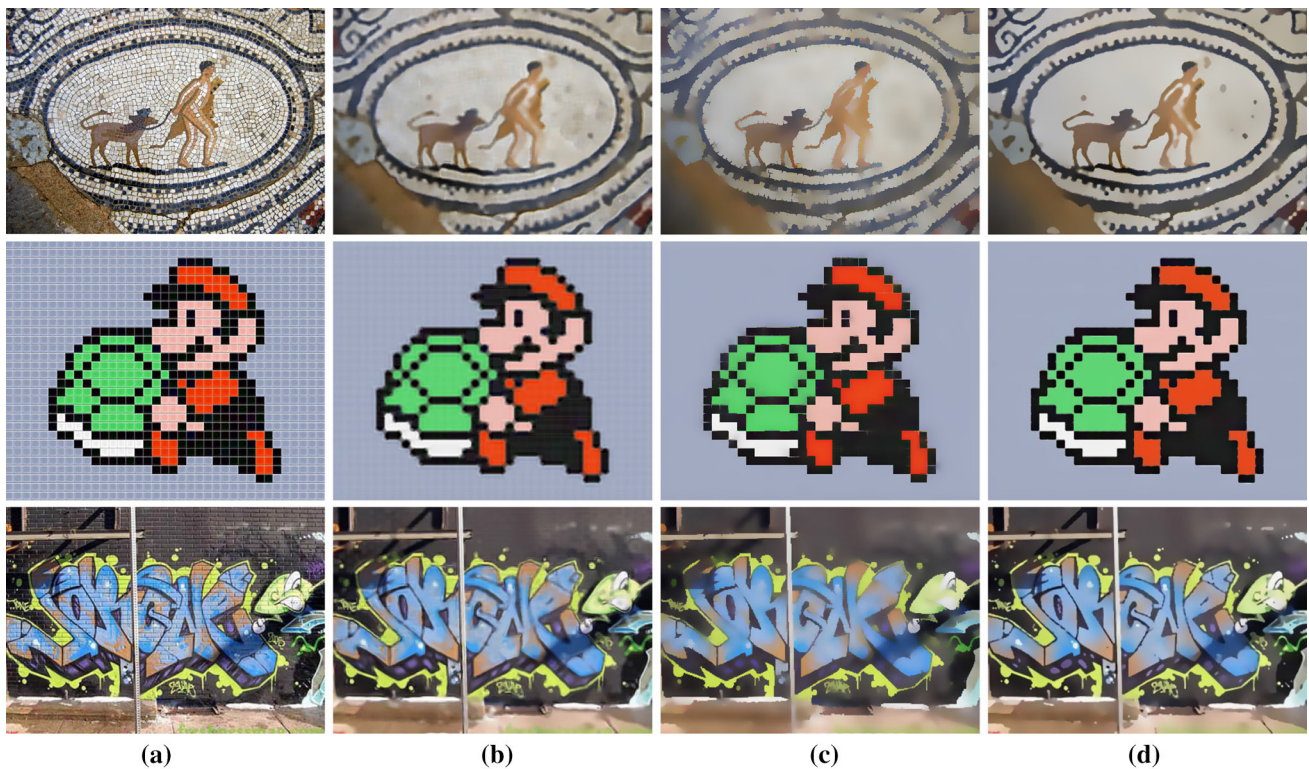
Fig. 8 Texture smoothed by Karacan et al. [12], Bao et al. [4] and our method. The parameter values of the state-of-the-art methods are listed in the supplementary material. **a** Input images, **b** Karacan et al. [12], **c** Bao et al. [4], **d** our method
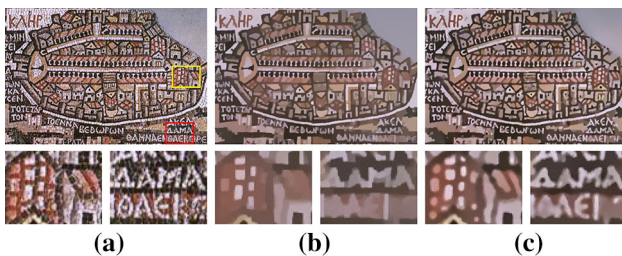
Fig. 9 Smoothing textures whose scales and appearance are similar to those of structures. Some structures are removed by Xu et al. [29]. **a** Input image, **b** Xu et al. [29], **c** our method

Table 1 Performance of our method (in seconds) and parameter values

Example	Size of input	σ_s	σ_r	N_{iter}	Times
Figure 7 (row 1)	627×441	5	0.05	6	2.864
Figure 7 (row 2)	400×324	4	0.04	11	5.565
Figure 7 (row 3)	495×536	4	0.05	7	3.509
Figure 7 (row 4)	900×675	5	0.05	5	8.903
Figure 8 (row 1)	500×356	7	0.04	8	2.427
Figure 8 (row 2)	500×406	5	0.055	8	2.798
Figure 8 (row 3)	400×284	3	0.055	6	1.673

Our experimental environment involves a computer with a CPU of Intel Core I5-4590, 4GB memory, and Matlab version 2013b. For a 400×300 grayscale image, it takes about 0.23 s for $\sigma_s = 5$ and $\sigma_r = 0.05$ to accomplish one single iteration process. The computation time of our unoptimized implementation and experimental parameter values are shown in Table 1. Note that we construct the color guidance images to yield corresponding filtering results for the second and fourth rows of Fig. 7, which inevitably adds to the computation time. We believe that the performance of our method can be further improved using GPU acceleration or available acceleration methods for the bilateral filter.
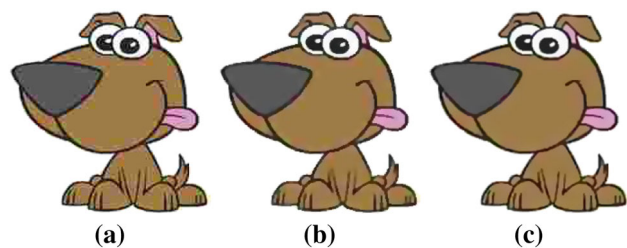
5.1 Applications

While our focus is on structure-preserving texture filtering in this work, we can apply our filter to a variety of image applications, including detail enhancement, edge detection, inverse halftoning, JPEG artifact removal, image abstraction and image composition. We show some of them in this section. More application results are contained in our supplementary material.

We can use our filter in layer decomposition and achieve detail enhancement results by enhancing the detail layer. A single iteration is sufficient for detail enhancement using our method. The interpolation scheme of our method yields bet-
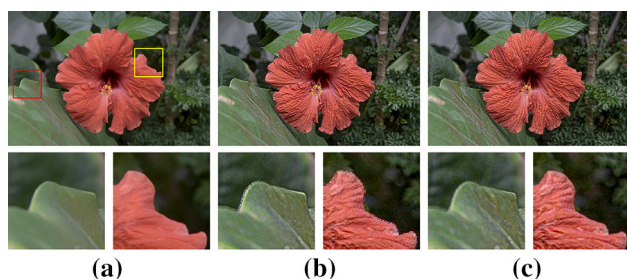
Fig. 10 Detail enhancement by Cho et al. [6] and our method. Parameters: Cho et al. [6] ($k = 3$, $N_{\text{iter}} = 5$), our method ($\sigma_s = 4$, $\sigma_r = 0.04$). **a** Input image, **b** Cho et al. [6], **c** our method

Fig. 13 Using patch shift [6] as the guidance image. **a** Input image. **b** The result of Cho et al. [6]. **c** The result of our method using the guidance image of Cho et al. [6]

Fig. 11 Comparison of cartoon JPEG artifact removal. **a** Input image. **b** The result of Cho et al. [6]. **c** The result of our method
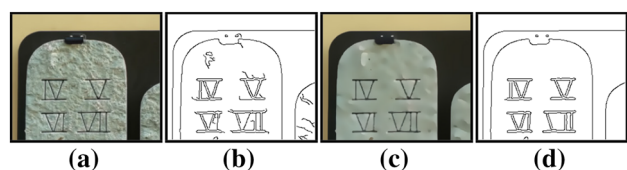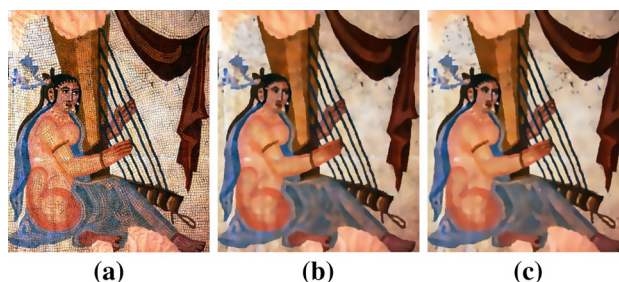

Fig. 12 Edge detection by our method. **a** Input image. **b** Canny edge detection result of **a**. **c** Our extracted structure component. **d** Our corresponding edge map. The threshold value for the Canny method in **b** and **d** is 0.38

ter results with smoother edges than those of Cho et al. [6]. Fig. 10 shows an example. Our method can also be used to remove severe compression artifacts of cartoon JPEG images. Our restoration results are comparable to those of previous methods, as shown in Fig. 11. Due to the ability to remove details and textures, our method is helpful to find the main edges of an image, as shown in Fig. 12. Directly using Canny edge detector on the input image cannot produce a cleaner map of structure edges. In contrast, our method produces a cleaner structure result, which makes detection of structure edges easy.

6 Discussion

The proposed method can be regarded as complementary to previous texture-filtering approaches. From a formal standpoint, filtering results of our method are affected by the interpolation coefficient and the guidance image. Based on

the contents of images, different filtering results can be achieved by setting different values of the interpolation coefficient and using different guidance images. If we set the interpolation coefficient μ_p to 1 for all image pixels, our method is the same as the original joint bilateral filter. Our method will accept various guidance images. For instance, other guidance image construction methods, such as patch shift [6], can also be employed here to better distinguish textures from structures. In this case, our method that produces smooth edges is a variant of Cho et al. [6]. Figure 13 shows such an example. The interpolation scheme plays a key role in distinguishing our approach from other methods. We believe that the applicability of the proposed new filter could be further explored in other extensive research on visual media processing.

The notion of using the content of a guidance image to influence the filtering process is related to the guided image filter [11]. Technically speaking, the guided image filter is also a neighborhood operation that takes into account the statistics of a region in the corresponding spatial neighborhood in the guidance image when calculating the value of the output pixel. Compared with the joint bilateral filter, the guided image filter is much faster and can avoid gradient reversal artifacts near edges. However, one possible problem of the guided filter is that halos may appear near some edges, as indicated in [11]. Therefore, the results of detail enhancement using the guided filter may exhibit undesired halos around edges. Figure 14 shows an example for detail enhancement. The joint bilateral filter result has noticeable gradient reversal artifacts near some edges. In comparison, our approach produces cleaner edges with no halos or far fewer halos, and without gradient reversal artifacts, due to the interpolation property of our method.

We have demonstrated the robustness of our method using a variety of input images in the paper and our supplementary material. To further evaluate the performance of our approach, a ground truth comparison was conducted. We compare images that have been distorted and smoothed with ground truth data which has no compression artifacts. Due
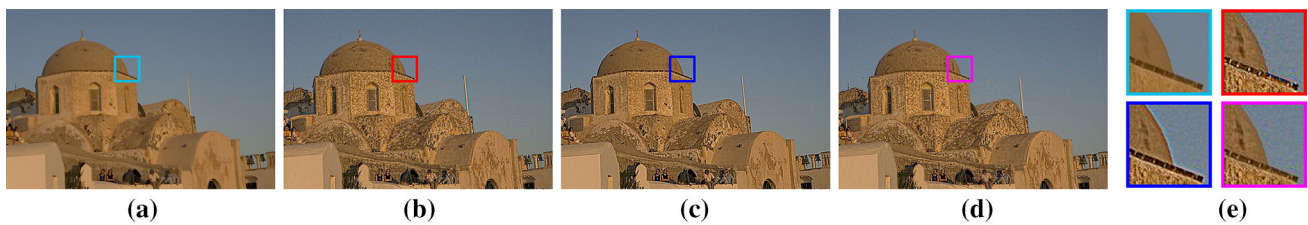
Fig. 14 The joint bilateral filter suffers from gradient reversal artifacts, which can be avoided in the guided filter. However, the guided filter produces halos near edges. In comparison, our approach produces the filtering result with much fewer halos and fewer gradient reversal artifacts. For the guided filter, we set r to 2 and ϵ to 0.01. In this example, we use a 5× detail enhancement. **a** Input image, **b** joint bilateral filter, **c** guided filter, **d** our filter, **e** close-ups
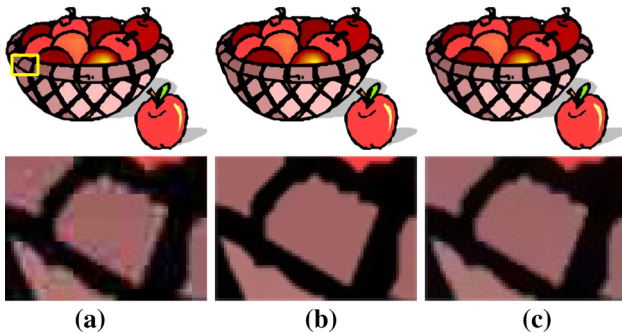
Fig. 15 Comparison with ground truth. The second row shows close-ups of the marked yellow region. Our filtering result closely matches the ground truth. **a** Distorted image, **b** ground truth, **c** our filtering

Fig. 16 Failure example. Using a single scale leads to removing small-scale structures. **a** Input image, **b** our result
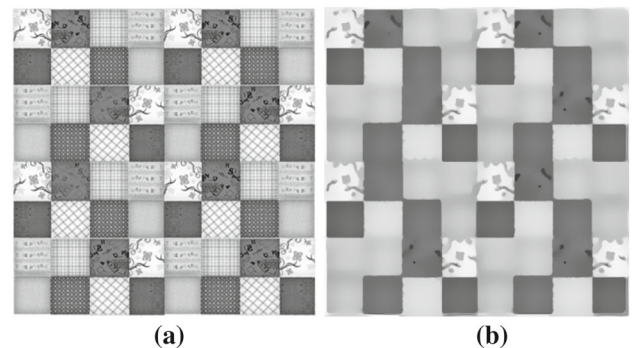
to the interpolation step, our approach helps filtering results to recover the original shapes of edges as much as possible while removing JPEG artifacts. As Figure 15 shows, filtering results yielded using our method are visually close to ground truth.

6.1 Limitation

The limitation of our method is that it uses a uniform scale to measure textures of an image. As a result, our method may mistake some small-scale structures for textures if the scale of the image textures is larger than that of the structures, and yield unsatisfactory texture-filtering results. Figure 16 shows an example where small-scale structures are smoothed out, because the scale of image textures is visually larger than that of structures. A multiscale version of our approach might help to better measure texture features of different scales and improve filtering results.

7 Conclusion

We have presented a new filter for texture smoothing based on the two-level local Laplacian filter. By introducing a guidance image in the remapping function, the two-level joint local Laplacian filter has the capability of structure-preserving texture smoothing while preserving color variations. We further introduce a variant of the joint bilateral texture filter. Our method better preserves structure edges than the joint bilateral filter, while it retains the simplicity of the joint bilateral filter. Extensive experiments have been conducted to demonstrate the effectiveness of our method.

In the future, we plan to investigate a scale-adaptive texture-smoothing method. This could improve the quality of structure–texture separation. Although using our current guidance image produces satisfactory results in most cases, we also try to develop more sophisticated texture measures to create the guidance image.

References

1. Aubry, M., Paris, S., Hasinoff, S.W., Kautz, J., Durand, F.: Fast local Laplacian filters: theory and applications. ACM Trans. Gr. **33**(5), 167:1–167:14 (2014)
2. Aujol, J.-F., Gilboa, G., Chan, T., Osher, S.: Structure-texture image decomposition-modeling, algorithms, and parameter selection. Int. J. Comput. Vis. **67**(1), 111–136 (2006)

3. Babaud, J., Witkin, A.P., Baudin, M., Duda, R.O.: Uniqueness of the gaussian kernel for scale-space filtering. IEEE Trans. Pattern Anal. Mach. Intell. **8**(1), 26–33 (1986)

4. 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. **23**(2), 555–569 (2014)

5. Buades, A., Le, T.M., Morel, J.-M., Vese, L.A.: Fast cartoon + texture image filters. IEEE Trans. Image Process. **19**(8), 1978–1986 (2010)

6. Cho, H., Lee, H., Kang, H., Lee, S.: Bilateral texture filtering. ACM Trans. Gr. **33**(4), 128:1–128:8 (2014)

7. Criminisi, A., Sharp, T., Rother, C., P'erez, P.: Geodesic image and video editing. ACM Trans. Gr. **29**(5), 134:1–134:15 (2010)

8. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Trans. Gr. **27**(3), 67:1–67:10 (2008)

9. Fattal, R.: Edge-avoiding wavelets and their applications. ACM Trans. Gr. **28**(3), 1–10 (2009)

10. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Gr. **30**(4), 69:1–69:12 (2011)

11. He, K., Sun, J., Tang, X.: Guided image filtering. In: Proceedings of the 11th European Conference on Computer Vision, pp. 1–14 (2010)

12. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Trans. Gr. **32**(6), 176:1–176:11 (2013)

13. Kass, M., Solomon, J.: Smoothed local histogram filters. ACM Trans. Gr. **29**(4), 100:1–100:10 (2010)

14. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Trans. Gr. **26**(3), 96:1–96:5 (2007)

15. Lu, C., Xu, L., Jia, J.: Contrast preserving decolorization. In: IEEE International Conference on Computational Photography, pp. 1–7 (2012)

16. Ma, Z., He, K., Wei, Y., Sun, J., Wu, E.: Constant time weighted median filtering for stereo matching and beyond. In: The IEEE International Conference on Computer Vision (ICCV), pp. 49–56 (2013)

17. Paris, S., Hasinoff, S.W., Kautz, J.: Local Laplacian filters: edge-aware image processing with a Laplacian pyramid. ACM Trans. Gr. **30**(4), 68:1–68:12 (2011)

18. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)

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

20. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)

21. Shen, J., Jin, X., Sun, H.: High dynamic range image tone mapping and retexturing using fast trilateral filtering. Vis. Comput. **23**(9–11), 641–650 (2007)

22. Shen, J., Zhao, Y., Yan, S., Li, X.: Exposure fusion using boosting Laplacian pyramid. IEEE Trans. Cybern. **44**(9), 1579–1590 (2014)

23. Zhuo, S., Luo, X., Deng, Z., Liang, Y., Ji, Z.: Edge-preserving texture suppression filter based on joint filtering schemes. IEEE Trans. Multimed. **15**(3), 535–548 (2013)

24. Subr, K., Soler, C., Durand, F.: Edge-preserving multiscale image decomposition based on local extrema. ACM Trans. Gr. **28**(5), 147:1–147:9 (2009)

25. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision, p. 839C846 (1998)

26. van de Weijer, J., van den Boomgaard, R.: Local mode filtering. In: Computer Vision and Pattern Recognition (CVPR), pp. 428–433 (2001)

27. Weiss, B.: Fast median and bilateral filtering. ACM Trans. Gr. **25**(3), 519–526 (2006)

28. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l0 gradient minimization. ACM Trans. Gr. **30**(6), 174:1–174:12 (2011)

29. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Trans. Gr. **31**(6), 139:1–139:10 (2012)

30. Yin, W., Goldfarb, D., Osher, S.: Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In: Proceedings of the Third International Conference on Variational, Geometric, and Level Set Methods in Computer Vision, pp. 73–84 (2005)

31. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Computer Vision—ECCV 2014, pp. 815–830 (2014)

32. Zhang, Q., Xu, L., Jia, J.: 100+ times faster weighted median filter. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2830–2837 (2014)

Hui Du is currently a faculty member of the Zhejiang University of Media and Communications. He received his MSc degree from the Zhejiang University of Technology in 2004 and his PhD degree from the State Key Lab of CAD & CG, Zhejiang University in 2013. His research interests include image processing and computer vision.

Xiaogang Jin is a professor of the State Key Lab of CAD & CG, Zhejiang University. He received his BSc degree in computer science in 1989, and MSc and PhD degrees in applied mathematics in 1992 and 1995, all from Zhejiang University. His current research interests include virtual try-on, insect swarm simulation, traffic simulation, implicit surface modeling and applications, creative modeling, sketch-based modeling and image processing. He received an ACM Recognition of Service Award in 2015. He is a member of the IEEE and ACM.

Philip J. Willis is Professor of Computing, Director of the Centre for Digital Entertainment and founded the Media Technology Research Centre at the University of Bath. Externally, he is a Fellow and past Chair of the Eurographics Association and became the first Eurographics-ACM SIGGRAPH joint member. He was a founding member of UKCRC, the UK Computing Research Committee. His research interests are in color raster graphics, computer games, virtual reality and animation, and film technologies. He has an underlying interest in picture and object representations, especially in the balance between discrete and continuous representations.