International Journal of Computer Vision (2018) 126:1245–1265
https://doi.org/10.1007/s11263-018-1091-5

Joint Contour Filtering

Xing Wei[1] · Qingxiong Yang[2] · Yihong Gong[1]

Received: 1 November 2016 / Accepted: 11 April 2018 / Published online: 23 April 2018

Abstract
Edge/structure-preserving operations for images aim to smooth images without blurring the edges/structures. Many exemplary edge-preserving filtering methods have recently been proposed to reduce the computational complexity and/or separate structures of different scales. They normally adopt a user-selected scale measurement to control the detail smoothing. However, natural photos contain objects of different sizes, which cannot be described by a single scale measurement. On the other hand, contour analysis is closely related to edge-preserving filtering, and significant progress has recently been achieved. Nevertheless, the majority of state-of-the-art filtering techniques have ignored the successes in this area. Inspired by the fact that learning-based edge detectors significantly outperform traditional manually-designed detectors, this paper proposes a learning-based edge-preserving filtering technique. It synergistically combines the differential operations in edge-preserving filters with the effectiveness of the recent edge detectors for *scale-aware* filtering. Unlike previous filtering methods, the proposed filters can efficiently extract subjectively meaningful structures from natural scenes containing multiple-scale objects.

Keywords Contour analysis · Edge-preserving filter · Structure-preserving filter · Scale-aware filter

1 Introduction

Edge/structure-preserving filtering[1] has found widespread employment in many computer vision and graphics tasks. It is an image smoothing technique that removes low-contrast details/textures, while maintaining sharp edges/image structures.

A broad category of edge-preserving filters is designed with a specific filter kernel, to measure the distance between two pixels in a local region. The distance measurement is then converted to give the confidence of an edge between the two pixels, for edge-aware filtering. This category is sensitive to noise/textures. Examples are anisotropic diffusion (Perona and Malik 1990; Weickert 1999), bilateral filters (BF) (Tomasi and Manduchi 1998), guided image filters (GF) (He et al. 2013), and domain transform filters (DTF) (Gastal and Oliveira 2011). Another category is proposed to separate meaningful structures from textures by utilizing local statistics, called structure-preserving filtering. Representatives are relative total variation (RTV) (Xu et al. 2012), bilateral texture filter (BTF) (Cho et al. 2014) and rolling guidance filter (RGF) (Zhang et al. 2014). The main challenge in this domain is to accurately include scale measurement and propagation mechanism for filter design, in order to distinguish sharp edges/image structures from details/textures of various sizes. Designing a robust scale-aware filter kernel is surprisingly difficult. There is no "optimal" solution, because the detection of image edges can only be evaluated in a subjective manner. Meanwhile, numerical experiments, such as the Berkeley Segmentation Dataset and Benchmark (BSDS500) (Arbelaez et al. 2011), demonstrate that human subjects have various perceptions of edges in the same image.

[1] The structure-preserving filtering can be considered as a special design of edge-preserving filtering to deal with its limitation in handling textures. In most cases, this paper adopts the phrase "edge-preserving" for a broader concept.

Communicated by S. Soatto.

✉ Qingxiong Yang
liiton.research@gmail.com

Xing Wei
xingxjtu@gmail.com

Yihong Gong
ygong@mail.xjtu.edu.cn

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, Shannxi, China

[2] JingChi Corp., 330 Gibraltar Drive, Sunnyvale, CA 94089, USA

On the other hand, significant progress has been achieved over the past few years in machine learning. Unlike in standard image processing techniques, which use strictly static program instructions, here a model is normally constructed based on example inputs, and then used to generate predictions or decisions. The performance of a learning-based edge-preserving image filter is thus likely to be closer to the human visual system when the example inputs are obtained based on an average agreement between a sufficient number of human subjects. The SVM-based filter presented in Yang et al. (2010) is the first learning-based bilateral filter. As a Taylor series expansion of the Gaussian function can be used to approximate the bilateral filters (Porikli 2008; Yang et al. 2010) learns a function that maps a feature vector consisting of the exponentiation of the pixel intensity, the corresponding Gaussian filtered response, and their products, to the corresponding exact bilateral filtered values from the training image.

This study aims to develop image smoothing methods that can preserve edges between different-sized objects/structures. This presents a considerably more challenging problem. Unlike the bilateral filter, it cannot be approximated using a Taylor series expansion, and there is no ground-truth filtered image available for training. Nevertheless, there are sufficient hand-labeled segmentation datasets. For instance, BSDS500 (Arbelaez et al. 2011) contains 12,000 hand-labeled segmentations of 1000 Corel dataset images from 30 human subjects. Image segmentation and edge detection are closely related to image smoothing techniques. They normally pre-smooth an image using a specific low-pass filter for noise reduction.

Because the human visual system is capable of understanding semantically meaningful structures blended with or formed by texture elements (Arnheim 1956), a "perfect" segmentation result (agreeing with human subjects) obviously provides excellent guidance for scale-aware edge-preserving filtering. However, it is difficult to obtain ideal edges from real-life images that are of moderate complexity. Traditional edge detectors rely on image gradients (followed by non-maximal suppression). Unfortunately, many perceptive inessential textures often have large gradient values. As a result, most state-of-the-art edge-preserving filters ignore potential contributions from edge detectors. On the other hand, it has recently been demonstrated that the performance of learning-based edge detectors is approaching that of human subjects (Dollár and Zitnick 2013, 2015).

This paper proposes a simple seamless combination of the differential operations in filter design and learning-based edge detection, to achieve fast *scale-aware* edge-preserving filtering. We observed that a number of fast edge-preserving filtering methods, including anisotropic diffusion (Perona and Malik 1990), domain transform filters (Gastal and Oliveira 2011), and recursive bilateral filters (RBF) (Yang 2012), operate on the differential structure of the input image. They recursively smooth an image based on the similarity between every pair of neighboring pixels, and are referred to as *anisotropic filters* in this paper. These filters are naturally more sensitive to noise than others like the bilateral filter (Tomasi and Manduchi 1998) or guided image filter (He et al. 2013). They are also unable to separate meaningful structures from textures. However, their computational complexity is comparatively low, as they can be implemented recursively. Another significant advantage is that they can be naturally combined with a state-of-the-art learning-based edge classifier (Dollár and Zitnick 2015), to encourage smoothing within regions until strong edges are reached (Yang 2016). Furthermore, a variety of global image filters, such as weighted least square (WLS) (Farbman et al. 2008) and relative total variation (Xu et al. 2012), also operate on the differential structure of images. These are generally more robust, but slower than anisotropic filters, because they optimize global object functions, and need to solve a large linear system. Fortunately, Min et al. proposed a fast approximation method (FGS) (Min et al. 2014), which works in a similar manner to recursive filters, and has a comparable computational efficiency and memory consumption. This type of global filter also allows the simple incorporation of a learned edge detector. Such an edge classifier is trained using human-labeled contours. According to Arnheim (1956), "the overall structural features are the primary data of human perception, not the individual details." Learning-based edge detectors can thus robustly distinguish the contours of different-sized objects by image noise and textures. When integrated with an anisotropic filter or a global filter, this enables robust structure extraction from natural scenes containing objects of various scales, as demonstrated in Fig. 1. The key reason for this ability is that the recursive and global filter operate on the whole scan line/image and the degree of smoothing is guided by the learned edge map which is trained according to human perception. This make our methods more effective than others which adopt fixed-size kernels and/or low-level features. Figure 1a shows two images containing both large scale (e.g., sky and meadow) and small scale (e.g., animals) objects. Current state-of-the-art edge-preserving filters cannot successfully separate large-scale objects from small ones, as shown in Fig. 1b–f. The proposed filtering technique does not suffer from this limitation, as demonstrated in Fig. 1g. To summarize, we make the following contributions.

– We reform several well-known filters, including the domain transform filter (Gastal and Oliveira 2011), recursive bilateral filter (Yang 2012), the weighted least square filter (Farbman et al. 2008), and the relative total variation filter (Xu et al. 2012). We analyze their inherent limitations, and improve their performance significantly.

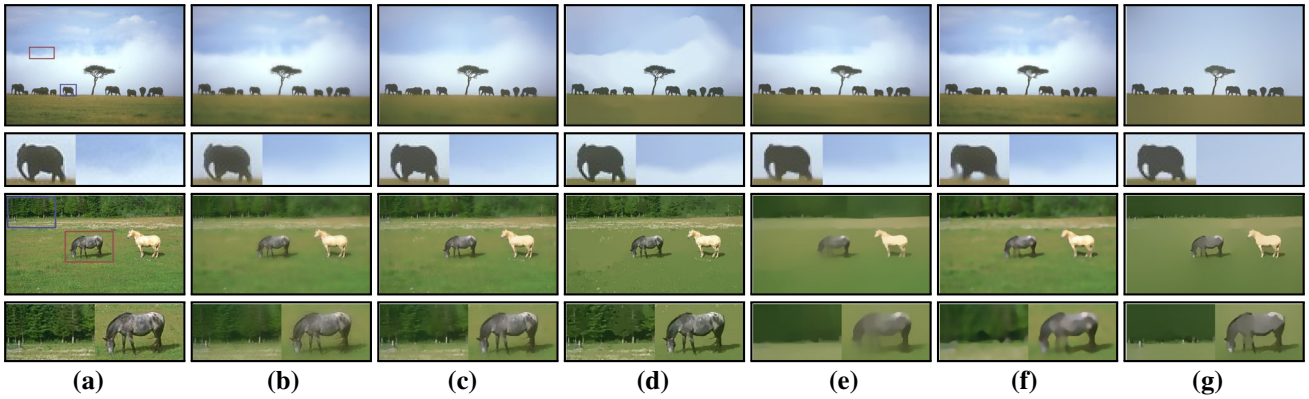

Fig. 1 Natural scenes, such as in **a**, contain objects of different sizes, and structures of various scales. As a result, the state-of-the-art edge-preserving filters are unlikely to obtain "optimal" smoothing results without parameter adjustments, as shown in **b**–**f**. The default parameters included in the implementations published by the authors were employed in this experiment. This study targets an efficient scale-aware filtering solution, by integrating a fast edge-preserving filtering technique with prior knowledge learned from human segmentation results. Unlike previous filters, it can sufficiently suppress image variance/textures inside large objects, while maintaining the structures of small objects, as shown in **g**. **a** Input, **b** GF (He et al. 2013), **c** DTF (Gastal and Oliveira 2011), **d** L_0S (Xu et al. 2011), **e** RTV (Xu et al. 2012), **f** RGF (Zhang et al. 2014), **g** proposed

– The proposed methods are robust to natural scenes containing objects of different sizes and structures of various scales, and thus can successfully extract subjectively meaningful structures from images containing multiple-scale objects.
– Qualitative and quantitative experiments, including large-scale texture removal, local edit propagation, image retargeting, saliency detection, and stereo matching have been conducted, demonstrating that our proposed methods perform favorably against state-of-the-art methods.

2 Related Work

2.1 Edge-Preserving Filtering

The most popular edge-preserving filter is probably the bilateral filter, introduced by Tomasi and Manduchi (1998). It has been applied to many computer vision and computer graphics tasks, and a general overview of these applications can be found in Paris et al. (2009). Let x_i denote the color of an image x at pixel i, and let y_i denote the corresponding filtered value,

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

where j is a pixel in the neighborhood Ω_i of pixel i, and G_{σ_s} and G_{σ_r} are the spatial and range filter kernels measuring the spatial and range/color similarities, respectively. The parameter σ_s defines the size of the spatial neighborhood used to filter a pixel, and σ_r controls how strongly an adjacent pixel is down-weighted because of the color difference. A joint (or cross) bilateral filter (Petschnigg et al. 2004; Eisemann and Durand 2004) is the same as the bilateral filter, except that its range filter kernel G_{σ_r} is computed from another image, named the guidance image.

Brute-force implementations of the bilateral filter are slow when the kernel is large. A number of techniques have been proposed for fast bilateral filtering, based on quantization of the spatial domain and/or range domain (Durand and Dorsey 2002; Pham and van Vliet 2005; Chen et al. 2007; Paris and Durand 2009; Yang et al. 2009; Adams et al. 2009, 2010; Gastal and Oliveira 2012). Other methods reduce the computational complexity using additional constraints on the spatial filter kernel (Weiss 2006; Porikli 2008) or the range filter kernel (Yang 2012).

Besides accelerating the bilateral filter, there also exist efficient bilateral-filter-like techniques derived from anisotropic diffusion (Perona and Malik 1990), weighted least squares (Farbman et al. 2008), wavelets (Fattal 2009), linear regression (He et al. 2013), local Laplacian pyramids (Parisand et al. 2011), domain transforms (Gastal and Oliveira 2011), and L_0 smoothing (L_0S) (Xu et al. 2011).

These edge-preserving filters have been broadly applied in computer vision and graphics. However, they all focus on a relatively small variance suppression, and are vulnerable to textures. The proposed filtering technique differs, in that it can distinguish meaningful structures from textures and image noise.

2.2 Structure-Preserving Filtering

Traditional edge-preserving filtering techniques cannot distinguish textured regions from the major structures in an

image. Popular structure-preserving techniques are based on the total variation (TV) model (Rudin et al. 1992; Chambolle and Darbon 2009). This uses L_1 norm-based regularization constraints to enforce large-scale edges, and has demonstrated the effective separation of structures from textures (Meyer 2001; Yin et al. 2005; Aujol et al. 2006). Xu et al. (2012) proposed relative total variation measures for better capturing the differences between textures and structures, and developed an optimization system to extract main structures. Another model based on local extrema (LEX) was proposed by Subr et al. (2009). This separates oscillations from the structure layer by extrema extraction and extrapolation. Alternatively, the use of superior similarity metrics instead of traditional Euclidean distances, such as geodesics (Criminisi et al. 2010) or diffusion (Farbman et al. 2010) distances, can enhance the performance of texture-structure separation. Karacan et al. (2013) employed region covariances (RCV) as patch descriptors, and leveraged the repetition property of textures to capture the differences between structures and textures. Cho et al. (2014) proposed a bilateral texture filter, which also relies on a patch-based analysis of texture features, and integrates its results into the range filter kernel of the conventional bilateral filter. Recently, some new total variation frameworks and nonlinear filters (Gilboa 2014; Buades and Lisani 2016; Zeune et al. 2016) have been proposed for multi-scale texture analysis. Structure-preserving filtering was typically slow before the availability of the rolling guidance filter (Zhang et al. 2014). Besides its efficiency, the work of Zhang et al. (2014) also proposes a unique scale measure to control the level of details during filtering. This scale measure is considerably useful when manual adjustment is required.

2.3 Contour Detection

Edge/contour detection is a fundamental task in computer vision and image processing. Traditional approaches, such as the Sobel operator (Duda and Hart 1973), detect edges by convolving the input image with local derivative filters. The most popular edge detector, the Canny detector (Canny 1986), makes extensions by adding non-maximum suppression and hysteresis thresholding steps. These approaches apply low-level interpolation of the image structures, and an overview can be found in Ziou and Tabbone (1998). Recent studies have focusing on utilizing machine learning techniques. These either train an edge classifier based on local image patches (Dollár et al. 2006; Lim et al. 2013; Ren and Liefeng 2012; Gupta et al. 2015; Dollár and Zitnick 2013; Zitnick and Dollár 2014; Dollár and Zitnick 2015), or make use of learning techniques for cue combination (Arbelaez et al. 2011; Zheng et al. 2007; Catanzaro et al. 2009; Zitnick and Parikh 2012). Deep neural networks have also recently been applied to edge detection (Kivinen et al. 2014), and dominate the current leading methods (Bertasius et al. 2015a, b; Xie and Tu 2015; Shen et al. 2015; Yang et al. 2016).

Traditional edge detectors rely on image gradients, while many visually salient edges, such as texture edges, do not correspond to image gradients. As a result, they are not suitable for structure-preserving filtering. However, state-of-the-art detectors (Dollár and Zitnick 2013, 2015) are learned using human labeled segmentation results, including sufficient texture edges. As a result, they contain useful and accurate structural information, which can be adopted for robust structure-preserving filtering.

3 Joint Contour Filtering

3.1 Anisotropic Filtering

Anisotropic diffusion is a traditional edge-aware filtering technique (Perona and Malik 1990). It is modeled using partial differential equations, and implemented as an iterative process. The recently proposed domain transform filter (Gastal and Oliveira 2011) and recursive bilateral filter (Yang 2012) are closely related to anisotropic diffusion, and can achieve a real-time performance.

3.1.1 Domain Transform Filter

Given a one-dimensional (1D) signal, the DTF (Gastal and Oliveira 2011) applies a distance-preserving transformation to the signal. A perfect distance-preserving transformation does not exist, but a simple approximation is given simply by the sum of the spatial distances (e.g., one-pixel distances) and color/intensity differences between every pair of pixels. Let x denote the 1D input signal, and let t denote the transformed signal,

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

In practice, two additional parameters, σ_s and σ_r, are included in Eq. (2) to adjust the amount of smoothness:

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

As can be seen from Eq. (3), this transform operates on the differential structure of the input signal, which is the same as in anisotropic diffusion, but with much faster results. A standard low-pass filter (e.g., Gaussian filter) with a kernel defined by σ_s will be used to smooth the transformed signal [Eq. (3)] without blurring the edges, and the final result is

obtained by transforming the smoothed signal back to the original domain.

3.1.2 Recursive Bilateral Filter

The recursive bilateral filter, presented in Yang (2012), is also closely related to anisotropic diffusion. A traditional bilateral filter has two Gaussian filter kernels, a spatial filter kernel and a range filter kernel, as shown in Eq. (1). The spatial filter kernel was reduced to a box filter in Weiss (2006) and Porikli (2008), and the range filter kernel was reduced to a polynomial range filter in Porikli (2008), in order to reduce the computational complexity. The recursive bilateral filter, presented in Yang (2012), employs a similar constraint. It assumes that the range filter can be decomposed into a recursive product (and that the spatial filter can be implemented recursively, which is standard for most of the fast bilateral filtering approaches).

Again, let x denote the 1D input signal of a causal recursive system of order n, and let y denote the output. Then,

$$y_i = \sum_{l=0}^{n-1}(a_l \cdot x_{i-l}) - \sum_{k=1}^{n}(b_k \cdot y_{i-k}), \tag{4}$$

where a_l and b_k are coefficients designed for a specific recursive filter. The RBF (Yang 2012) extends the above recursive system for bilateral filtering by modifying the coefficients at each pixel location:

$$a_l^{new} = R_{i,i-l} \cdot a_l, \tag{5}$$
$$b_k^{new} = R_{i,i-k} \cdot b_k, \tag{6}$$

where

$$R_{k,i} = \prod_{j=k}^{i-1} R_{j,j+1} = exp\left(-\frac{\sum_{j=k}^{i-1}(x_j - x_{j+1})^2}{2\sigma_r^2}\right) \tag{7}$$

is the range filter kernel, and σ_r is a constant used to control the intensity/color similarity measurement between pairs of pixels. An anti-causal recursive filter of the same order is required to compute responses from right to left. Note that the normalization factor is omitted in Eqs. (5) and (6), as this can be directly computed from the above equations by setting each x_i equal to one. Similarly to anisotropic diffusion and the DTF, the RBF also operates on the differential structure of the input signal, as can be seen from Eq. (7). These filters are referred to as *anisotropic filters* in this paper. As with the DTF (Gastal and Oliveira 2011) and RBF (Yang 2012), 2D signals will be filtered using the 1D operations by performing separate passes along each dimension of the signal. It is also demonstrated in Gastal and Oliveira (2011) that artifact-free filtered images can be obtained by performing filtering iteratively, and the filter kernel size (defined by σ_s and σ_r) should be reduced after every iteration to converge.

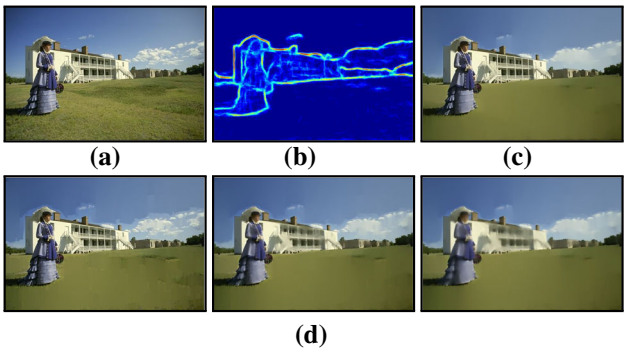

Fig. 2 Direct use of the edge confidence as guidance may introduce visible artifacts or over-smooth the image, as can be seen in **d**. **c** shows that the combination of the edge confidence in **b** and the image gradients can effectively suppress the potential artifacts resulting from incorrect edge detection. **a** Input, **b** edge confidence, **c** proposed, **d** joint filtering directly using **b** (w.r.t. different parameters)

3.2 Scale-Aware Anisotropic Filtering

This anisotropic filters presented in Sect. 3.1 can be implemented recursively, and thus the computational complexity is relatively low. However, they are sensitive to image noise, and cannot distinguish textures from structures. Available solutions either manually design low-level vision models and descriptors (Rudin et al. 1992; Xu et al. 2012; Karacan et al. 2013) to capture the differences between structures and textures, or simply adopt a texture scale parameter (Zhang et al. 2014). The performance of these filters is excellent when perfect parameters are employed. Nevertheless, natural photos contain objects of different sizes and structures of various scales, which are difficult to describe in terms of a unified low-level feature. In contrast, it has already been demonstrated in closely related research (such as on edge detection) that high-level features learned from human-labeled data can significantly outperform manually-designed features. This section makes use of the state-of-the-art structured learning based edge classifier (Dollár and Zitnick 2013, 2015), to achieve structure-preserving filtering while maintaining its efficiency. The sufficient human-labeled training examples from the BSDS500 benchmark (Arbelaez et al. 2011) enable the proposed filtering technique to be robust for various texture scales.

The anisotropic filters presented in Sect. 3.1 accumulate the image gradients in order to measure the distance between two pixels, as can be seen from Eqs. (3) and (7). However, it is clear that texture edges do not correspond to image gradients. A straightforward learning-based solution is to train a deep learning architecture to map a local patch to a "per-

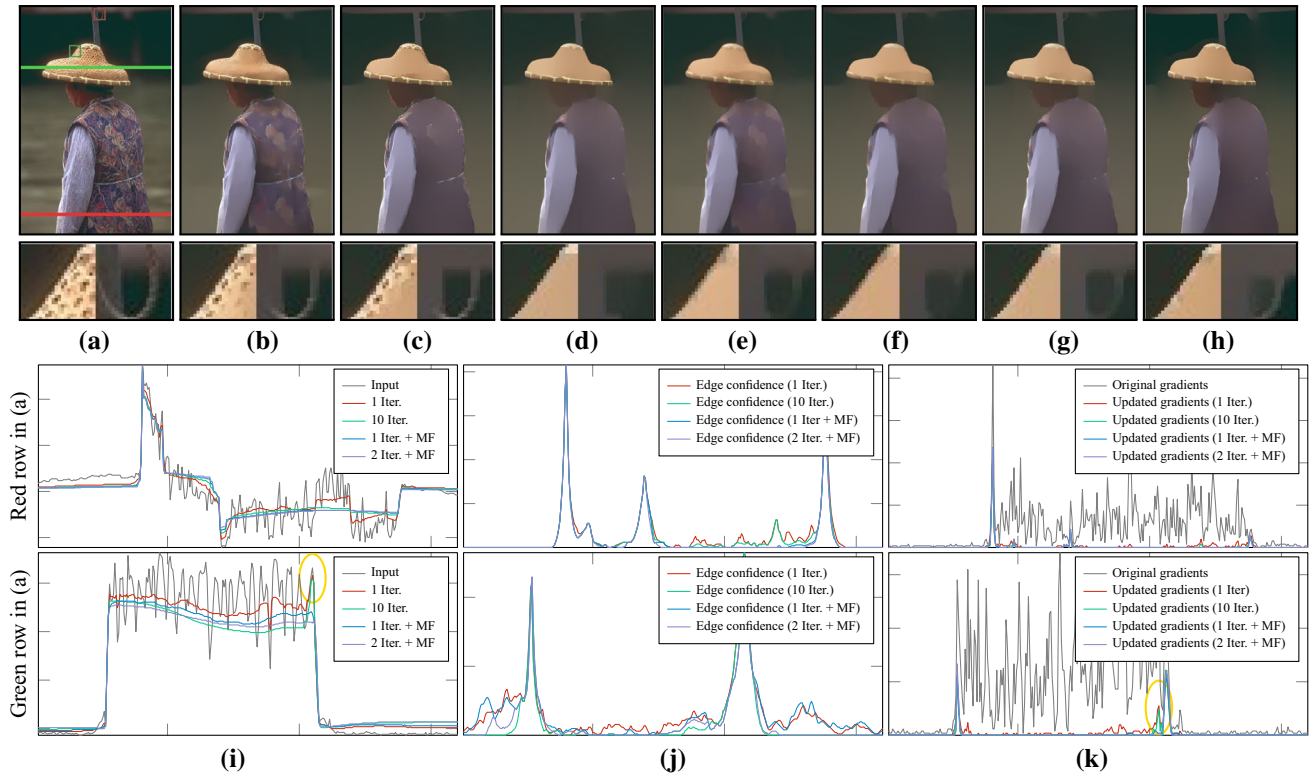

Fig. 3 The integration of anisotropic filters with an edge detector can successfully remove textures, except for small-scale textures around the edges, as can be seen in **b**–**c**. A simple and robust solution is to remove these textures by using a rolling guidance filter (Zhang et al. 2014) in advance, as demonstrated in **d**. However, this will be relatively slow. To improve the efficiency, this study instead uses the differential structure of the median filtered image to smooth the original input image, as shown in **g**–**h**. The yellow ellipses will be discussed in the text. **a** Input, **b** 1 Iter., **c** 10 Iter., **d** Zhang et al. (2014) + 2 Iter. **e** 1 Iter. (MF), **f** 2 Iter. (MF), **g** our cDTF, **h** our cRBF, **i** pixel values (on the red channel), **j** edge confidence, **k** image gradients (Color figure online)

fect" image gradient value, so that it is low inside of a region and high around texture edges. However, this solution will be slow. A simpler but much faster solution is thus adopted in this work. By taking advantage of the inherent structure in edges in a local patch, Dollár and Zitnick (2015) proposes a generalized structured learning approach for edge classification. This has been demonstrated to be highly robust to textures, as well as very efficient. A direct solution to structure smoothing is to employ the edge confidence computed from Dollár and Zitnick (2015) as the guidance image in the DTF and the range filter in RBF, to smooth the input image. This type of filter is called a *joint/cross* (bilateral) filter in the literature (Petschnigg et al. 2004; Eisemann and Durand 2004). Specifically, let f_j denote the confidence of an edge at pixel j. Then, Eqs. (3) and (7) are modified as follows to suppress gradients (resulting from textures) inside a region:

$$t_i = x_0 + \sum_{j=1}^{i} 1 + \frac{\sigma_s}{\sigma_r} \cdot f_j, \tag{8}$$

$$R_{k,i} = exp\left(-\frac{\sum_{j=k}^{i-1} f_j^2}{2\sigma_r^2}\right). \tag{9}$$

This is an effective solution given a perfect edge classifier, which does not exist in practice. This may introduce visible artifacts or over-smooth the image, as shown in Fig. 2d.

This paper proposes the use of edge confidence to adjust the original distance measure in DTF and the range filter in RBF:

$$t_i = x_0 + \sum_{j=1}^{i} 1 + \frac{\sigma_s}{\sigma_r} \cdot f_j |x_j - x_{j-1}|, \tag{10}$$

$$R_{k,i} = exp\left(-\frac{\sum_{j=k}^{i-1} f_j (x_j - x_{j+1})^2}{2\sigma_r^2}\right). \tag{11}$$

The combination of the edge confidence and the image gradient can effectively suppress the potential artifacts resulting from incorrect edge detection. Because that our methods integrate the edge/contour map for joint filtering, we refer to the proposed filters as joint contour DTF and RBF (cDTF and cRBF).

The gray curves in Fig. 3k represent the original image gradients of the red and green rows in Fig. 3a, respectively. Traditional anisotropic filters are vulnerable to textures in

these two rows. The red curves in Fig. 3j represent the edge confidence detected from these two rows, respectively. The peaks in the two red curves correspond to the edges of the red and green rows in Fig. 3a. The edge confidence is used to suppress the image gradients inside the textured regions, and enable texture removal according to Eqs. (10) and (11). The red curves in Fig. 3k represent the modified image gradients, which correspond to the $f_j|x_j - x_{j-1}|$ values in Eq. (10). Note that the variance of the image gradients inside the textured regions has been significantly suppressed, and thus the resulting anisotropic filter can successfully remove most of the textures (e.g., the hat), as can be seen in Fig. 3b and the red curves in (i). A new edge confidence can be obtained from the filtered image, and used to further suppress the textures.

The proposed edge-preserving filtering technique iteratively computes edge confidence using a learning-based edge classifier, and applies this to suppress the textures until convergence is achieved. As in Gastal and Oliveira (2011), the filter kernels (determined by σ_s and σ_r in Eq. 10) are iteratively reduced (by half), in order to guarantee convergence. Figure 3c presents the filtered image after 10 iterations. This shows that most of the visible textures are removed. The green, blue, and purple lines in the first row of Fig. 3i–k correspond to the pixel intensities, edge confidence measurements, and updated image gradients of the red row in (a) after 2, 3, and 10 iterations, respectively. This shows that the proposed filter converges rapidly (after only around three iterations).

3.2.1 Suppressing Small-Scale Textures and Structures

The filters presented in Eqs. (10) and (11) cannot sufficiently remove small-scale textures around highly-confident edges, as shown in the close-ups below Fig. 3b, c. This is because of the imperfect confidence measurements around a textured edge. As shown in the yellow ellipse in Fig. 3k, large image gradients around texture edges cannot be effectively suppressed, even after a large number of iterations. As a result, the original pixel values will be preserved, as can be seen from the yellow ellipse in Fig. 3i. Applying a small median filter to the input image cannot significantly affect the edge confidence around highly-confident edges, as can be seen in the blue and purple lines in Fig. 3j. However, this is very effective for removing textures around edges, as demonstrated in Fig. 3e–f and the blue and purple lines in the second row of Fig. 3i, k (see the values around the two yellow ellipses). Nevertheless, a median filter will, of course, remove thin-structured objects, as shown in the close-ups under Fig. 3e, f. Let x^{MF} denote the median filter result for the input signal x. In this study, *only* x^{MF} is used to compute the image gradients in Eqs. (10) and (11). Let

$$R_{k,i}^{MF} = exp\left(-\frac{\sum_{j=k}^{i-1} f_j\left(x_j^{MF} - x_{j+1}^{MF}\right)^2}{2\sigma_r^2}\right). \tag{12}$$

Then, the proposed cDTF and cRBF are computed as follows:

$$t_i = x_0 + \sum_{j=1}^{i} 1 + \frac{\sigma_s}{\sigma_r} f_j |x_j^{MF} - x_{j-1}^{MF}|, \tag{13}$$

$$y_i = \sum_{l=0}^{n-1} (R_{i,i-l}^{MF} \cdot a_l \cdot x_{i-l}) - \sum_{k=1}^{n} (R_{i,i-k}^{MF} \cdot b_k \cdot y_{i-k}). \tag{14}$$

Figure 3g, h present the images filtered using the proposed cDTF [Eq. (13)] and cRBF [Eq. (14)]. They both successfully remove the textures in Fig. 3a–c, while better preserving the details around thin-structured objects. An alternative solution is to directly apply the rolling guidance filter (Zhang et al. 2014) to the input image to remove the small-scale textures, and the result is presented in Fig. 3d. However, this method will be relatively slow.

Meanwhile, some applications are desired to generate a multi-scale representation. In this case, the median filter can serve as a scale selector, similarly to the other initial blurring operators employed in the previous structure-preserving filters (Zhang et al. 2014; Cho et al. 2014), and its size σ_m should be adjusted according to the scale of structures to be removed (see Figs. 4 and 18). For the other experiments, we set $\sigma_m = 2$ as a constant. The values of σ_m, σ_s, and σ_r will be reduced by half after every iteration, in order to guarantee convergence, and convergence will typically be achieved after as few as two iterations, as described in Sect. 5.1.

3.3 Scale-Aware Global Image Smoothing

Low-level image processing operations are often formulated as the minimization of an energy function comprising the data and prior terms,

$$E = E_{data} + \lambda \cdot E_{prior}, \tag{15}$$

where λ controls the relative importance of the two parts. A general data fidelity term for local appearance adjustment (Dani et al. 2004; Lischinski et al. 2006) can be expressed as

$$E_{data} = \sum_{p \in \Omega} (s_p - u_p)^2, \tag{16}$$

where s_p is the output editing label, and u_p denotes an exemplary input label. Ω represents the entire image in image filtering, while for interactive editings such as colorization, image matting, and tone adjustment, Ω represents

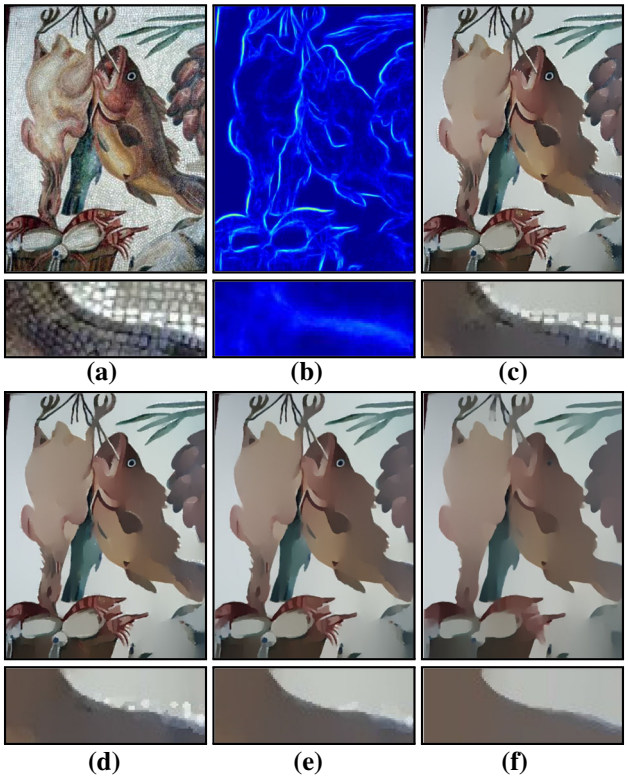

Fig. 4 Different median filter (MF) sizes σ_m will affect the texture removal result around edges. **a** Input, **b** edge map, **c** without MF, **d** MF ($\sigma_m = 1$), **e** MF ($\sigma_m = 2$), **f** MF ($\sigma_m = 4$)

the user-drawn stroke set. Sophisticated data terms have also been designed to better deal with complex spatially-varying images or sparse user inputs, such as the all-pairs constraint (An and Pellacini 2008) and the Gaussian Mixture Model (GMM) (Xu et al. 2013). This work focuses on the simplest L_2 norm data term which is defined only on individual pixels. As a result, there is a trivial solution in which the output label is strictly equal to the input. A simple prior is to enforce the smoothness of the output image, and some representative methods are the weighted least square (Farbman et al. 2008) and relative total variation (Xu et al. 2012).

3.3.1 Weighted Least Square

The smoothness constraint of the WLS filter is adaptively enforced using a spatially varying weighting function $w_{p,q}(g)$, defined on a guidance image g:

$$E_{prior}^{WLS} = \sum_{q \in \mathrm{N}(p)} w_{p,q}(g)(s_p - s_q)^2, \tag{17}$$

where N is the four- or eight-neighbor system, and $w_{p,q}$ represents the affinity between two pixels p and q, and is typically defined as a Gaussian with standard deviation σ_r:

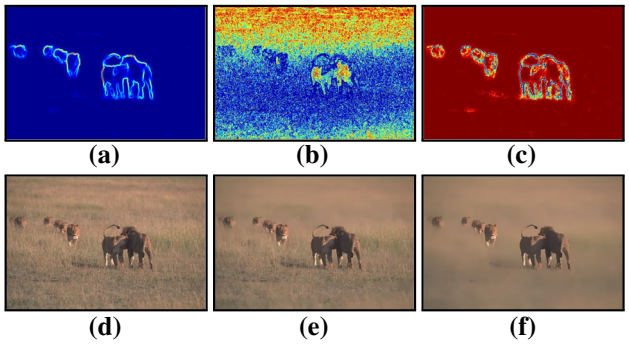

Fig. 5 The affinity map of the WLS filter computed by the Euclidean distance of pixel values is vulnerable to textures, as shown in **b**. A cleaner affinity map is obtained in **c** by combining learned edges, and this can effectively remove textures. **a** Input edge, **b** Euclidean distance, **c** edge-adaptive distance, **d** input image, **e** result using (**b**), **f** result using (**c**)

$$w_{p,q}(g) = exp\left(-(g_p - g_q)^2/2\sigma_r^2\right). \tag{18}$$

The similarity measurement has an important effect on the performance of the WLS filter. It is originally designed on the Euclidean distance (Farbman et al. 2008) between pixel values, and is further improved by using the diffusion distance (Donoho et al. 2006; Farbman et al. 2010), which better accounts for the global distribution of pixels in their feature space. As was pointed out in the previous sections, learning-based methods have demonstrated significant advantages over manually designed low-level vision models and descriptors. Thus, the integration of a higher level of image understanding into low-level image operators is promising. Based on this observation, we introduce the idea of using the learned edge confidence to define the similarity measurement, and we refer to this as the *edge-adaptive distance*:

$$w_{p,q}(g) = exp\left(-f_p(g_p - g_q)^2/2\sigma_r^2\right). \tag{19}$$

Figure 5 illustrates the affinity map calculated by $\sum_{q \in \mathrm{N}(p)} w_{p,q}(g)$ at each pixel location p. As can be seen from Fig. 5b, The affinity computed using the Euclidean distance of pixel values suffers when it comes to textures. Figure 5c shows the refined affinity, using (a). Note that the refined affinity is often high everywhere except for at salient edges, which not only allows the removal of perceptive inessential textures, but also facilitate the propagation of sparse user inputs for interactive image editing, such as colorization (see Fig. 10 for an example).

3.3.2 Relative Total Variation

The RTV measurement is proposed in order to better capture the differences between textures and structures. It is based on

the observation that a major structural edge in a local window contributes more gradients in similar directions than textures with complex patterns do. The definition of RTV is

$$E^{RTV}_{prior} = \frac{G_{\sigma_s} * |\partial_x S|}{|G_{\sigma_s} * \partial_x S| + \varepsilon} + \frac{G_{\sigma_s} * |\partial_y S|}{|G_{\sigma_s} * \partial_y S| + \varepsilon}, \tag{20}$$

where $\partial(\cdot)$ is the discrete gradient operator, $*$ is the convolution operator, and G_{σ_s} is a Gaussian filter with standard deviation σ_s. The division in Eq. (20) is to be understood element-wise, and ε is a small positive number inserted to avoid division by zero. The RTV measure does not require prior texture information, and can also remove non-uniform and multiple-scale textures. Essentially, the RTV measure penalizes *all* textures with scales smaller than that corresponding to σ_s. Thus, the inherent limitation of RTV is that *it cannot remove large-scale textures without blurring small-scale structures*. Based on the previous success of integrating learned edges into edge-preserving filters, we propose joint contour RTV (cRTV), which allows the efficient extraction of subjectively meaningful structures from natural scenes containing multiple-scale textures. The formula for cRTV is as follows:

$$E^{cRTV}_{prior} = \frac{G_{\sigma_s} * |\partial_x S|}{f \cdot |G_{\sigma_s} * \partial_x S| + \varepsilon} + \frac{G_{\sigma_s} * |\partial_y S|}{f \cdot |G_{\sigma_s} * \partial_y S| + \varepsilon}. \tag{21}$$

An example is presented in Fig. 6, to demonstrate the limitations of the original RTV. Figure 6a shows a road surface covered with textures of various scales. Variation maps with different scale parameters are shown in the first row. A small ($\sigma_s = 3$) windowed variation measure is not sufficient to remove large-scale textures on the ground, as shown in (b). This phenomenon is alleviated when the window size is increased ($\sigma_s = 8$ and $\sigma_s = 15$). However, small-scale objects, such as pedestrians, become blurred, as shown in (c) and (d). On the contrary, the proposed cRTV can effectively remove multi-scale textures using a small σ_s, as shown in (e).

The proposed framework is summarized in Algorithm 1.

Algorithm 1: Joint Contour Filtering

Input: image I, iterations n_{iter}, parameters $\phi = \{\sigma_{\mathbf{r}}, \sigma_{\mathbf{s}}, \lambda\}$
Output: texture filtered image J

$J^0 \leftarrow I$
for $t = 1$ ***to*** n_{iter} **do**
  $M^t \leftarrow$ median blurring of J^{t-1}
  $E^t \leftarrow$ edge detecting of J^{t-1}
  $J^t \leftarrow$ joint filtering of J^{t-1} using M^t and E^t as guidance
  $\phi \leftarrow 0.5 \times \phi$
end

4 Applications

4.1 Large-Scale Texture Removal

Several methods (Xu et al. 2012; Karacan et al. 2013; Zhang et al. 2014; Cho et al. 2014) have recently been proposed for extracting meaningful structures from highly-textured images. This facilitates subsequent image manipulation operations, including visual abstraction, detail enhancement, and scene understanding. These methods typically focus on removing small-scale textures, and achieve an excellent performance in this scenario. Nevertheless, few methods deal effectively with large-scale textures. Large-scale texture removal is challenging, especially when there are severe variations inside the textured regions, as shown in Fig. 7a. Figure 7b–d present the intermediate results of the proposed cRTV filter. Our cRTV filter iteratively updates the structure image and edge map. Empirically, between three and five iterations are sufficient to suppress textures. We employ four iterations in our experiments, which is the same as in the original RTV (Xu et al. 2012). Figure 7e–f show that our method is helpful for the image segmentation task when the state-of-the-art method, MCG (Arbeláez et al. 2014), fails. Figure 8 visually compares cRTV with state-of-the-art structure-preserving filters. The proposed method performs favorably against the others, which either cannot remove large-scale textures sufficiently or blur salient structures.

4.2 Local Edit Propagation

Many image and video processing methods are performed with the assistance of user strokes, such as colorization (Dani et al. 2004), tone adjustment (Lischinski et al. 2006), matting (Levin et al. 2006), intrinsic decomposition (Bousseau et al. 2009), and white balance correction (Boyadzhiev et al. 2012). Such methods are intended to perform spatially-variant editing by propagating user-guided information without crossing prominent edges. This section validates the benefits of employing an edge-adaptive distance in the place of the classical Euclidean distance (Farbman et al. 2008) or diffusion distance (Farbman et al. 2010) in the framework of the WLS filter. We show that there exist practical scenarios in which an edge-adaptive distance can significantly improve the editing results compared with traditional distance measures. Figure 9 demonstrates the performance of these three distance measures. The goal is to select different regions, and apply editing effects individually. These two images are challenging, because they either have significant variations inside a region, or contain different objects sharing similar appearances. The performance for the Euclidean and diffusion distance is unsatisfactory in these two cases, while our method generates visually pleasing results. The editing

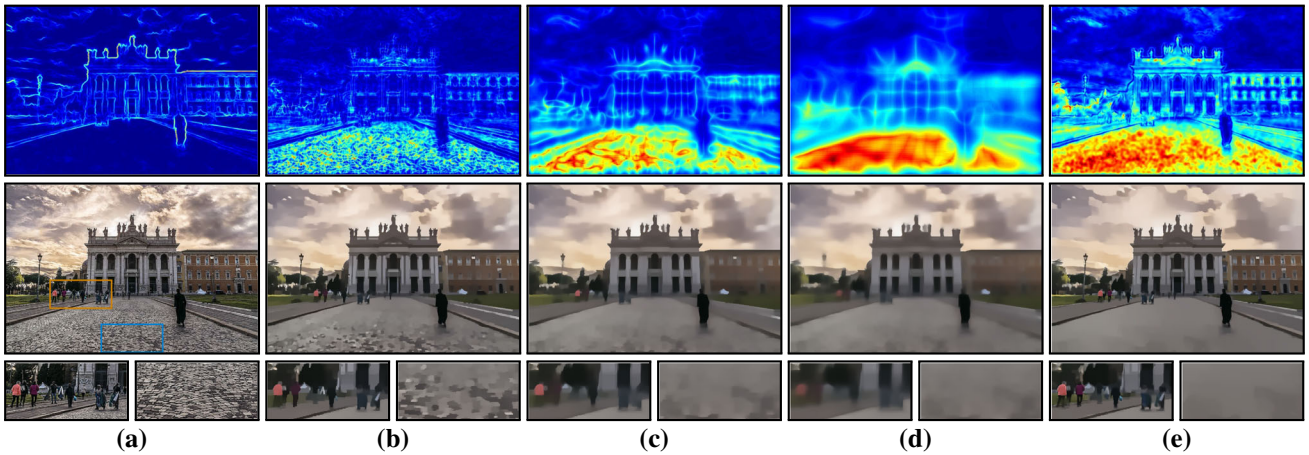

Fig. 6 Comparison of RTV and the proposed cRTV. The spatial parameter σ_s in Eqs. (20) and (21) controls the window size for computing the windowed variations, as shown in the first row. As shown in **b**–**d**, a small value of σ_s is not sufficient to remove large-scale textures, while a large value blurs salient structures. In contrast, cRTV enables the extraction of subjectively meaningful structures using a small σ_s. **a** Image and edge map, **b** RTV, $\sigma_s = 3$, **c** RTV, $\sigma_s = 8$, **d** RTV, $\sigma_s = 15$, **e** our cRTV, $\sigma_s = 3$

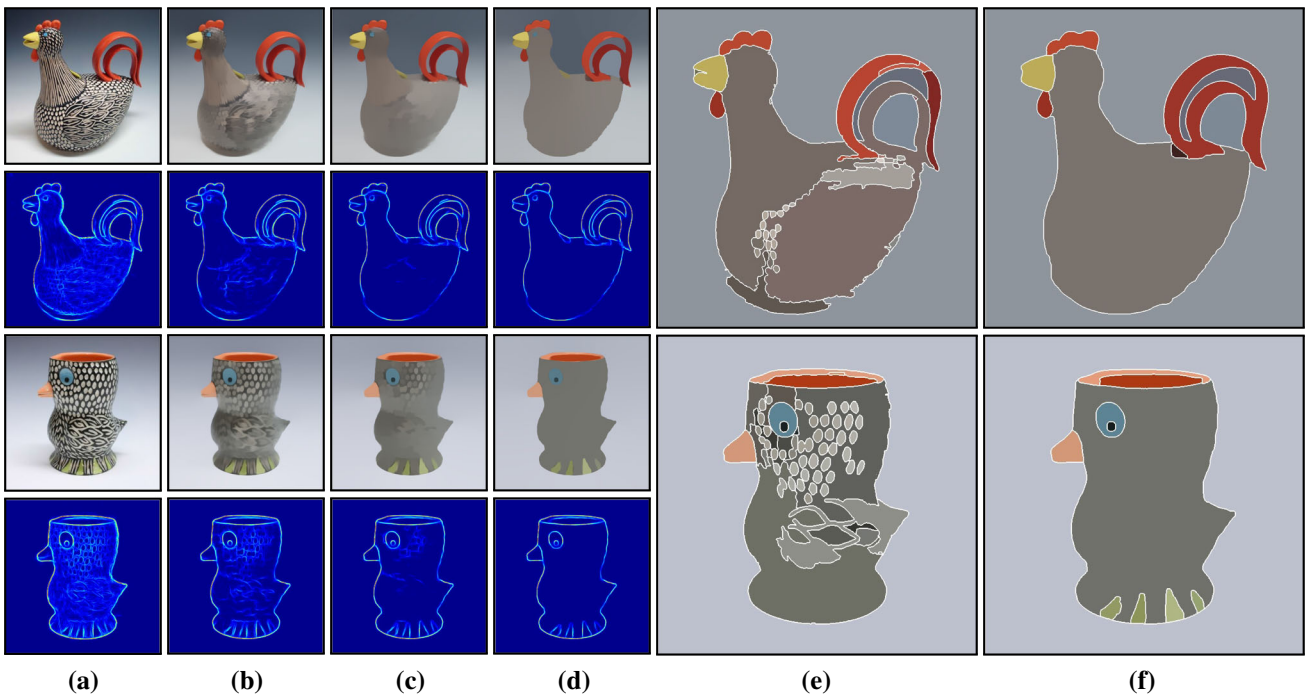

Fig. 7 Large-scale structure–texture separation is challenging, especially when there are severe variations inside the textured region such as in **a**. The proposed cRTV filter iteratively updates the structure image and edge map. Empirically, between three and five iterations are sufficient to suppress textures, as shown in **b**–**d**. **e**, **f** show the MCG (Arbeláez et al. 2014) segmentation results on the original image and our filtered image, respectively. Our method removes textures effectively, and is helpful for extracting more accurate segments. **a** Input, **b** 1st Iter., **c** 2nd Iter. **d** Final output, **e** MCG (Arbeláez et al. 2014) on the original image, **f** MCG (Arbeláez et al. 2014) on our filtered image

results for the first image were obtained in Kyprianidis and Döllner (2008) and Kyprianidis and Kang (2011).

Figure 10 presents the colorization results for two challenging images, where the classical methods (Dani et al. 2004; Fattal 2009)[2] and the WLS filter with the Euclidean or diffusion distance fail. The first image contains very sparse color strokes, and the second has a complicated fore-

[2] The implementations with the default parameters published by the authors were employed.

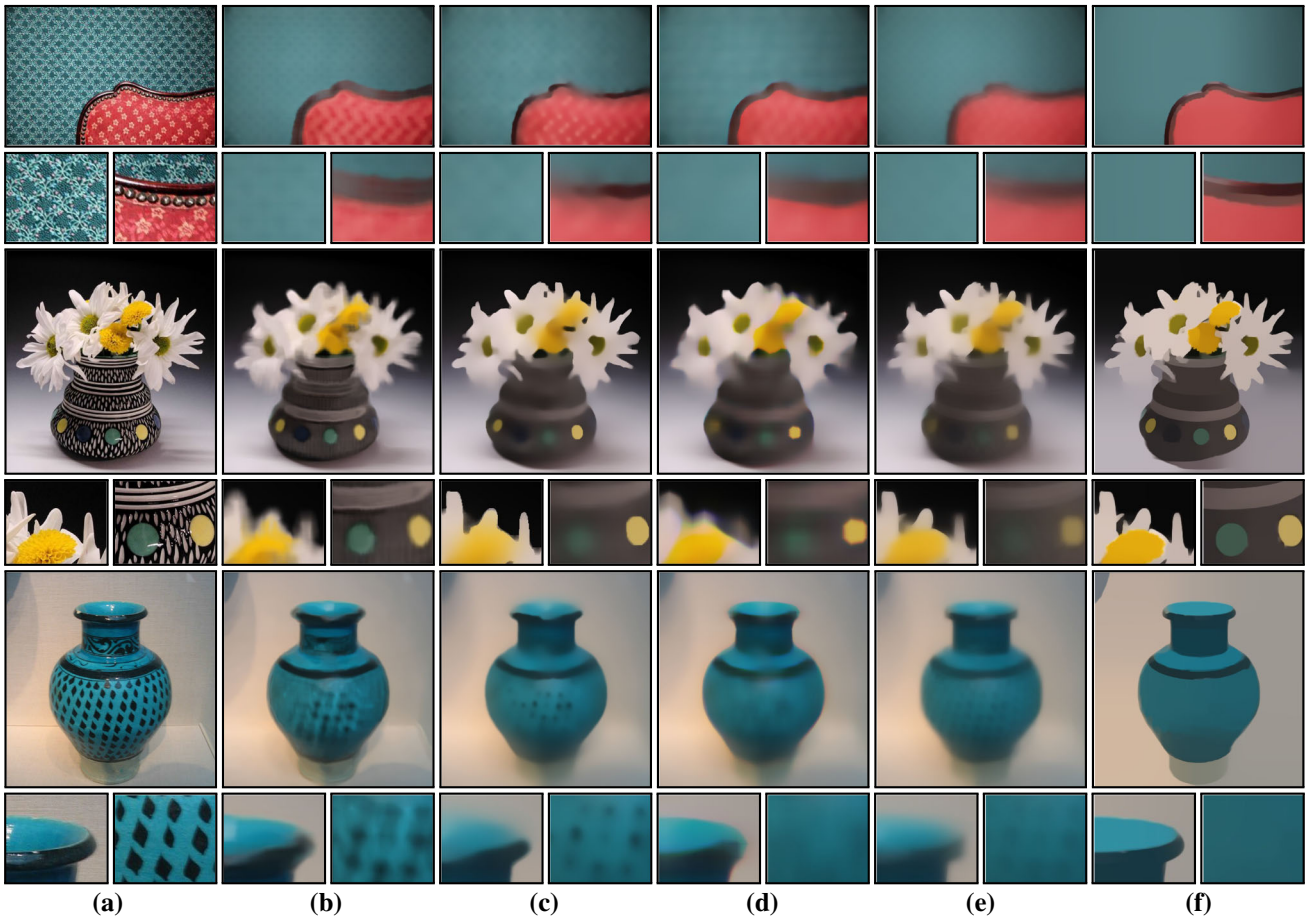

Fig. 8 Visual comparison with state-of-the-art structure-preserving filters for large-scale texture removal. Parameters: RCV (Karacan et al. 2013)($\sigma_s = 0.5, k = 31$, M1), BTF (Cho et al. 2014)($k = 15, n_{iter} = 3$), RGF (Zhang et al. 2014)($\sigma_s = 15, \sigma_r = 0.1$), RTV (Xu et al. 2012)($\sigma_s = 25, \lambda = 0.04$), and cRTV($\sigma_s = 25, \lambda = 0.04$). **a** Input, **b** RCV (Karacan et al. 2013), **b** BTF (Cho et al. 2014), **c** RGF (Zhang et al. 2014), **d** RTV (Xu et al. 2012), **e** our cRTV

ground/background. As discussed in Sect. 3.3.1, a clean affinity map can be obtained when integrating the learned edges into the affinity measurement. This affinity map is often high everywhere apart from salient edges, and thus our method facilitates the propagation of sparse inputs on complex images.

4.3 Image Retargeting

The seam carving method proposed in Avidan and Shamir (2007) resizes an image by taking its content into account. It uses a gradient-based energy function to measure the relative importance of each pixel. As described throughout this paper, natural scenes often have high gradient values on inessential textures, so that the seam carving method may generate unsatisfactory results. Content-aware image editing is also related to saliency detection, which will be analyzed in detail in Sect. 4.4. This section provides visual results for the application of image retargeting. Figure 11 shows a beach photograph that has high gradient values on the sand, and thus the entire sand region in the resized image is preserved using the original seam carving method. In contrast, our cRBF and cWLS remove the textured area effectively, and deliver superior resized images.

4.4 Saliency Detection

The quantitative evaluation of image filtering is difficult, and thus the state-of-the-art methods (Xu et al. 2011, 2012; Zhang et al. 2014) provide only a visual evaluation. In this section, we propose evaluating the improvement over the state-of-the-art saliency detection algorithm numerically, where the original image is preprocessed by state-of-the-art filters. We use the ECSSD (Yan et al. 2013) dataset, which is challenging, and the minimum barrier saliency (MBS) detection algorithm (Zhang et al. 2015), which is both fast and accurate. Saliency detection aims to locate outstanding objects/regions in images, which is closely related

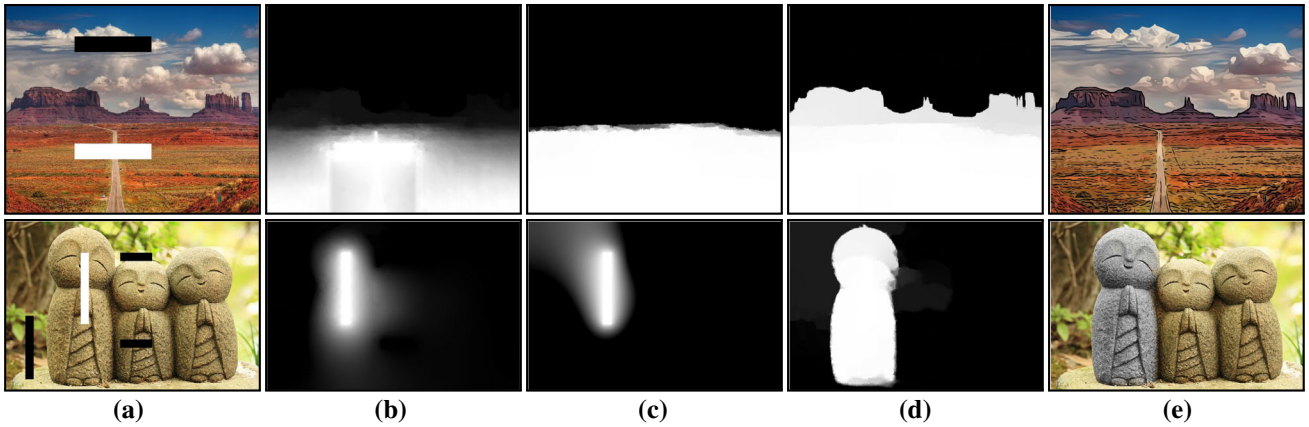

Fig. 9 Comparison of three distance measures in the framework of the WLS filter for local edit propagation. These two images either have large variations in the same region or contain different objects sharing similar appearances, which are difficult to process with conventional distance measures. **a** Input, **b** Euclidean distance (Lischinski et al. 2006), **c** diffusion distance (Farbman et al. 2010), **d** edge-adaptive distance, **e** editing results

Fig. 10 Two difficult colorization cases. The first image shows an example with sparse color strokes, and the second contains a complex foreground/background. We compare two classical methods (Dani et al. 2004; Fattal 2009) with the WLS filter with three different distance measures. The methods of Dani et al. (2004), Fattal (2009) and the two conventional distance measures for the WLS filter produce results with artifacts, while our method enables the propagation of sparse inputs on complex images. The result obtained by cRBF is also displayed. **a** Input, **b** Dani et al. (2004), **c** Fattal (2009), **d** Euclidean (Lischinski et al. 2006), **e** Diffusion (Farbman et al. 2010), **f** cWLS, **f** cRBF (Color figure online)

to selective perception in the human vision system. One common challenge in saliency detection is when the foreground/background contains salient/complex patterns, as shown in Fig. 12. Structure-preserving filters effectively abstract undesirable details while maintaining relevant structures, which is beneficial for this task. The precision-recall curves that evaluate the overall performance of a saliency detection method are illustrated in Fig. 13. Note that the proposed filter consistently outperforms the state-of-the-art filters (Xu et al. 2011, 2012; Zhang et al. 2014). The corresponding mean absolute errors (MAE) (Perazzi et al. 2012) and weighted-F-measure scores (WFM) (Margolin et al. 2014) are presented in Table 1, which shows that the proposed filter has the lowest error rate and the highest weighted-F-measure score. We have also combined the proposed method with a more recent edge detector (Yang et al. 2016), and

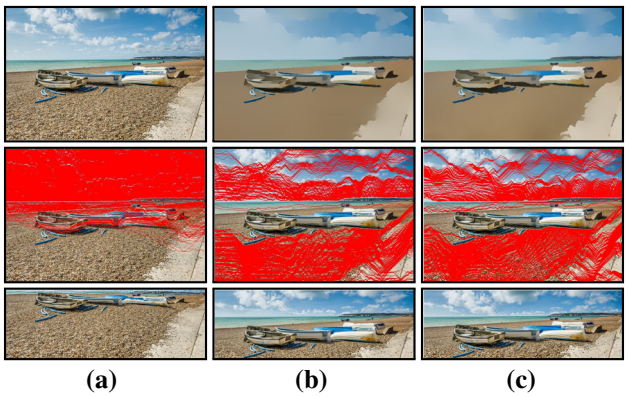

Fig. 11 Seam carving results. Top to bottom: input image, filtered image using cRBF and cWLS, eliminated seams, and the resized output images. **a** Original input, **b** our cRBF, **c** our cWLS

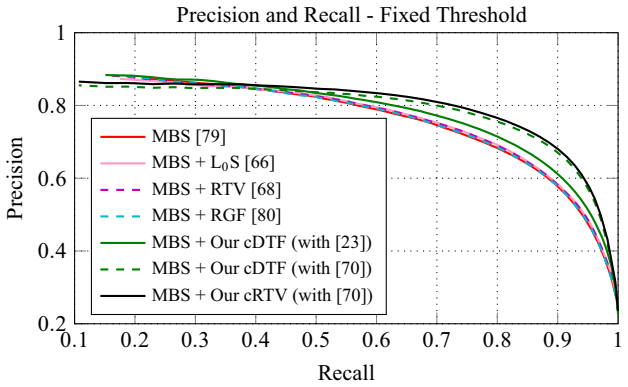

Fig. 13 Precision-recall curves for saliency detection. Note that preprocessing using structure-preserving filters can outperform the original MBS method on average, and the proposed filter significantly outperforms the others

the performance is illustrated in the last row of Table 1. The saliency detection accuracy is significantly further improved, which verifies that our method can directly integrate progress in edge detection, and could have broader benefits in other applications.

4.5 Stereo Matching

A local stereo matching algorithm generally performs (subsets of) the following four steps: cost volume computation, cost aggregation, disparity computation (winner-takes-all), and disparity refinement. The work of Yoon and Kweon (2006) is the first to employ an edge-preserving filter (i.e., BF) to perform cost aggregation. The basic idea of this technique is to transfer the structural information in the guidance image to the cost volume, in order to aggregate the cost without crossing edges. However, the guidance image may contain textures, which are not desired to be transferred. Thus, an edge-preserving filter that is robust to textures can be helpful.

Table 2 presents the results of several edge-preserving filters on the Middlebury benchmark. It can be seen that the proposed method outperforms the traditional edge-preserving filters.

5 Analysis

We have presented techniques to address two important problems in image filtering. Namely, that edge-preserving filtering (DTF, RBF, and WLS) is vulnerable to textures,

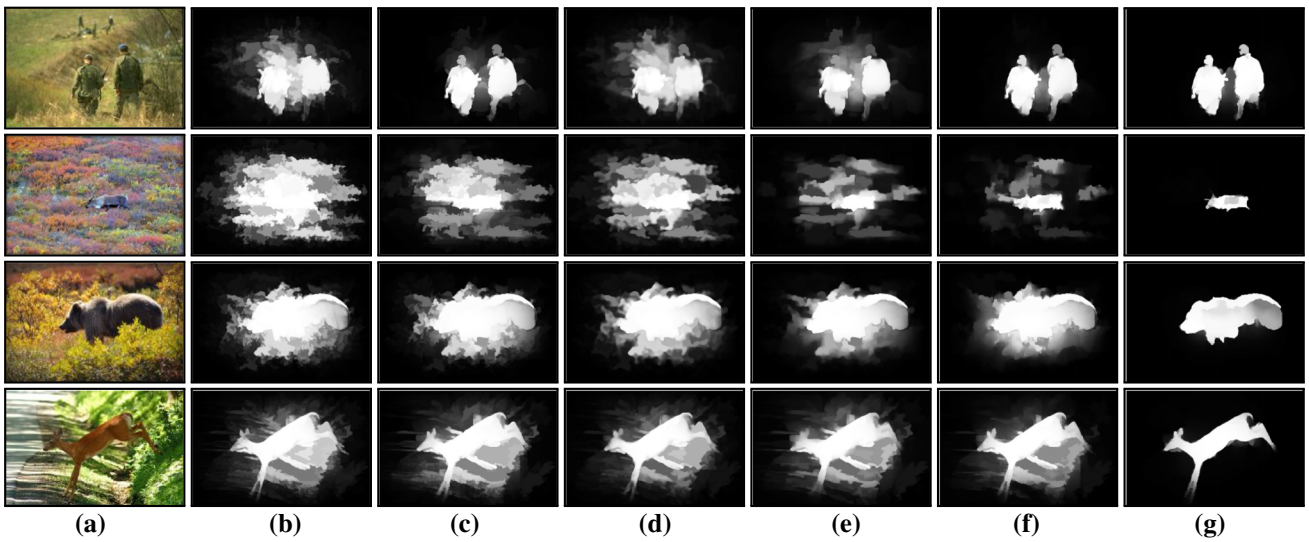

Fig. 12 Structure-preserving filtering for saliency detection. Complex background patterns hamper the state-of-the-art method (Zhang et al. 2015) in locating salient objects. Structure-preserving filters effectively abstract unnecessary details, which is beneficial for this task. **a** Input, **b** MBS, **c** MBS + L_0S, **d** MBS + RGF, **e** MBS + RTV, **f** MBS + cDTF (Dollár and Zitnick 2013), **g** MBS + cDTF (Yang et al. 2016)

Table 1 Quantitative evaluation of the state-of-the-art filters using the minimum barrier saliency (MBS) detection method (Zhang et al. 2015) on the ECSSD (Yan et al. 2013) dataset. The proposed filter has the lowest error and the highest weighted-F-measure score accuracy

Method	MAE ↓	WFM ↑
MBS (Zhang et al. 2015)	0.1707	0.5612
MBS + L_0S (Xu et al. 2011)	0.1674	0.5668
MBS + RTV (Xu et al. 2012)	0.1660	0.5630
MBS + RGF (Zhang et al. 2014)	0.1698	0.5606
MBS + Our cDTF (with Dollár and Zitnick 2013)	0.1578	0.5846
MBS + Our cDTF (with Yang et al. 2016)	0.1402	0.6278
MBS + Our cRTV (with Yang et al. 2016)	0.1339	0.6339

Our method can directly integrate recent progress in edge detection, such as Yang et al. (2016), and significantly further improve saliency detection

Table 2 Stereo matching

Method	Tsukuba			Venus			Teddy			Cones			Avg.Error
	Non	All	Disc	Non	All	Disc	Non	All	Disc	Non	All	Disc	
BF (Yoon and Kweon 2006)	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67
GF (Rhemann et al. 2011)	1.51	1.85	7.61	0.20	0.39	2.42	6.16	11.8	16.0	2.71	8.24	7.66	5.55
RBF (Yang 2012)	1.85	2.51	7.45	0.35	0.88	3.01	6.28	12.1	14.3	2.80	8.91	7.79	5.68
Our cRBF	1.80	2.14	6.86	0.29	0.50	2.42	5.90	11.3	14.3	2.48	7.82	7.10	5.25

Quantitative evaluation of the performance of several edge-preserving filters on the Middlebury benchmark. Parameters for cRBF: $\sigma_s = 0.02$, $\sigma_r = 0.03$

and structure-preserving filtering (RTV) is hampered by multiple-scale objects. Figure 14 visually compares the proposed cRBF method with the state-of-the-art filters under default/constant parameter settings. The adoption of a learning-based edge detection technique (Dollár and Zitnick 2015) enables the proposed filters to be robust to natural scenes containing objects of different sizes and structures of various scales.

The computational cost for the proposed filtering technique resides in the adopted edge detector (Dollár and Zitnick 2015), the anisotropic filter [DTF (Gastal and Oliveira 2011), RBF (Yang 2012), FGS (Min et al. 2014)], and the median filter (Weiss 2006; Perreault and Hbert 2007; Yang et al. 2015). These can all run in real-time, and thus the whole pipeline will be fast if the number of iterations is low. In practice, a downsample version is sufficient for both the edge detector (Dollár and Zitnick 2015) and the median filter when the image resolution is relatively large (e.g., 1 megapixel). As a result, the computational cost mainly resides in the adopted anisotropic filter, which operates on the full-resolution input images. The computational complexity of the adopted anisotropic filters (DTF, RBF, and FGS) is independent of the filter kernel size. These filter a 2D image by performing alternative horizontal and vertical 1D filtering, and the number of arithmetic operations required for each pixel is also considerably low. For example, the DTF and RBF use two multiplication operations at every pixel location to filter a 1D signal, and the FGS uses six. Note that the computational complexity of the proposed filters and the RGF (Zhang et al. 2014) is much lower than for the other structure-preserving filters. However, RGF bears the same limitation as RTV, in that it is not suitable for images that contain various structure scales, as shown in Fig. 15. On the other hand, the RGF can be more efficient than ours on the GPU since the recursive computation is not fully parallelized compared to the pixel-wise operations.

5.1 Convergence Analysis

In this study, we used the BSDS500 benchmark (Arbelaez et al. 2011) to analyze the convergence problem. Similarly to Gastal and Oliveira (2011), the filtered result obtained after n iterations is evaluated by comparing it with the result obtained for the same image after 15 iterations, which can be considered as artifact-free practically. We use the structural similarity (SSIM) index (Wang et al. 2004), as recommended by Gastal and Oliveira (2011), to perform a numerical comparison, because SSIM provides an image-quality metric that is consistent with human perception.

Figure 16 illustrates the similarity measured for various numbers of filtering iterations. The curves represent the maximum errors obtained on the BSDS500 dataset for the parameters $\sigma_s \in \{0.01, 0.05, 0.1, 0.2, 0.4, 0.6\}$, $\sigma_r \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, $\lambda \in \{10, 100, 500, 1000, 5000, 10000\}$ (for cWLS), and $\lambda \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$ (for

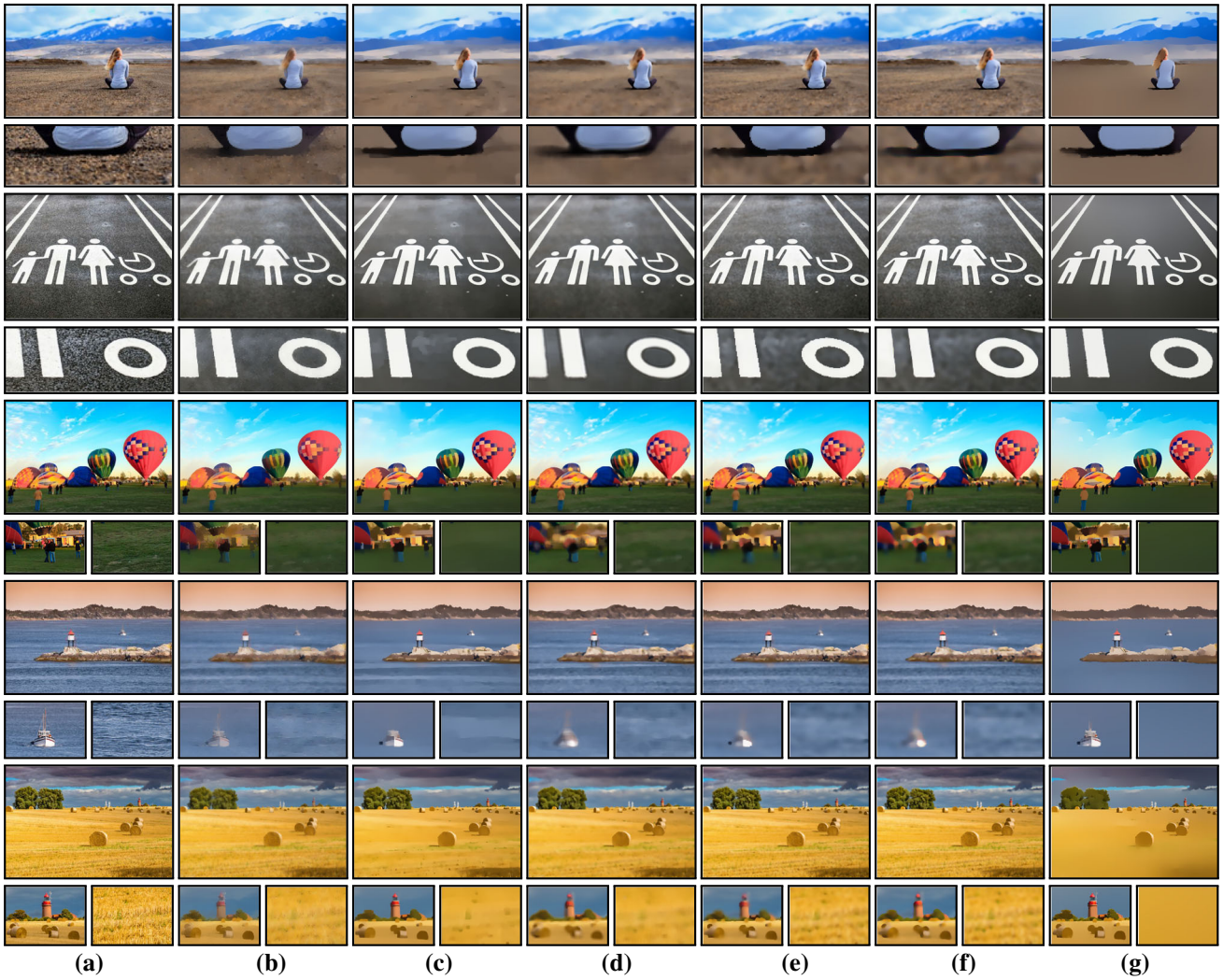

Fig. 14 Visual comparison with the state-of-the-art structure-preserving filters under constant/default parameter settings. Unlike the state-of-the-art filters, the proposed filter is more robust to various object scales, owing to the adoption of a learning-based edge detection technique (Dollár and Zitnick 2015). Parameters: LEX (Subr et al. 2009) (kernel size: 3×3), RTV (Xu et al. 2012) ($\sigma_s = 3, \lambda = 0.01$), RCV (Karacan et al. 2013) ($\sigma = 0.2, k = 9$, M1), BTF (Cho et al. 2014) ($k = 3, n_{iter} = 3$), RGF (Zhang et al. 2014) ($\sigma_s = 3, \sigma_r = 0.1$), and cRBF ($\sigma_s = 0.15, \sigma_r = 0.015$). **a** Input, **b** LEX (Subr et al. 2009), **c** RTV (Xu et al. 2012), **d** RCV (Karacan et al. 2013), **e** BTF (Cho et al. 2014), **f** RGF (Zhang et al. 2014), **g** our cRBF

cRTV), which is sufficient to cover all practical cases. As can be seen, it is safe to stop after only two iterations,[3] as the SSIM values computed from the filtered images after two iterations are close to or higher than 0.98.

5.2 Analysis of Parameters

Most edge-preserving filters have two important free parameters: the range parameter and the spatial parameter.[4] For example, σ_r is the range parameter of the DTF, RBF, and WLS, and σ_s is the spatial parameter of the DTF, RBF, and RTV. In general, the range parameter controls the simplification level based on color differences/edge confidence, and the spatial parameter determines the scale of texture to be removed. Although the proposed methods are mainly designed to be robust to various object scales, as shown in Fig. 14, they can be easily extended to generate a multi-scale representation via different spatial parameters. This subsection presents a parameter study in the two respects described above.

Figure 17 illustrates the effects of employing our cDTF, cRBF, and cWLS methods with different range parameter values. Note that the increasing the range parameter results

[3] Experiments conducted in this study use two iterations, except for the large-scale texture removal task in Sect. 4.1, which requires more iterations to smooth large-scale highly-textured images.

[4] Please note that when a spatial parameter is a fractional number, it represents the percentage of width/height of the image.

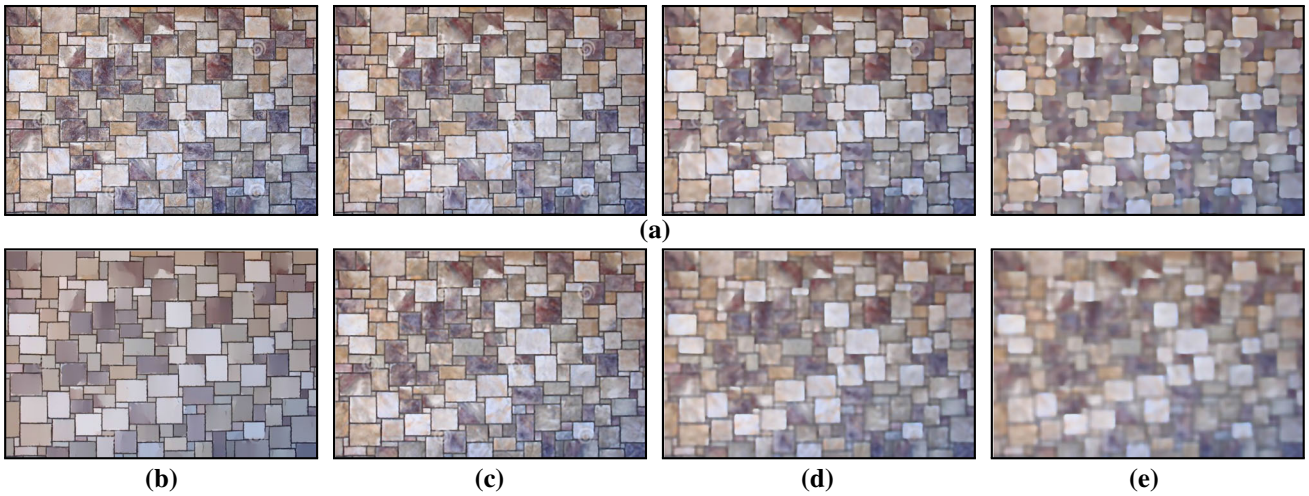

Fig. 15 The RGF (Zhang et al. 2014) and proposed cDTF and cRBF methods are significantly faster than the others. RGF requires an estimate of the structure scale, and can effectively smooth out the corresponding textures. However, it is not suitable for scenes containing objects/structures of multiple scales, as can be seen from **c** to **e**. It either fails to sufficiently remove textures in a large-scale object, or blurs small-scale objects, whereas our proposed filter does not have this limitation, as demonstrated in **b**. **a** Input $\{\sigma_s, \sigma_r\} = \{3, 0.05\}$. $\{\sigma_s, \sigma_r\} = \{6, 0.05\}$. $\{\sigma_s, \sigma_r\} = \{9, 0.05\}$, **b** our cDTF, **c** $\{\sigma_s, \sigma_r\} = \{3, 0.15\}$, **d** $\{\sigma_s, \sigma_r\} = \{6, 0.15\}$, **e** $\{\sigma_s, \sigma_r\} = \{9, 0.15\}$

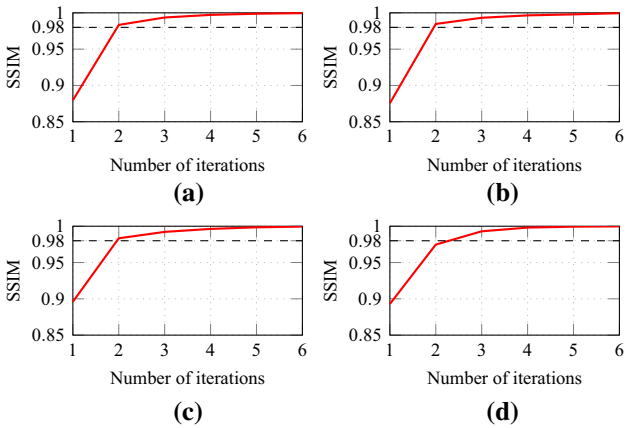

Fig. 16 Convergence analysis. Similarity measured using SSIM between the filtered images and their "ideal" results as a function of numbers of iterations. **a** cDTF, **b** cRBF, **c** cWLS, **d** cRTV

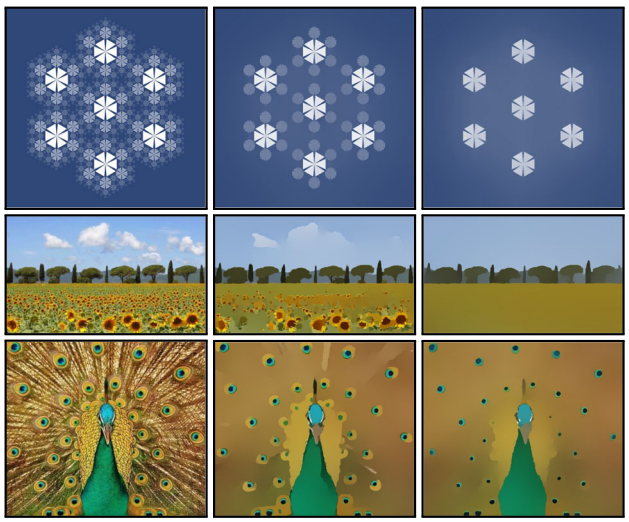

Fig. 17 The effects of the range parameters. From left to right: input, results using small range parameters, results using large range parameters. From top to bottom: cDTF ($\sigma_s = 0.3, \sigma_r = 0.1, 0.2$), cRBF ($\sigma_s = 0.5, \sigma_r = 0.1, 0.4$), cWLS ($\lambda = 1000, \sigma_r = 0.02, 0.04$)

in a hierarchy of image simplifications, and at each smoothing level the structures that contain outstanding colors are well preserved.

Figure 18 illustrates the effects of employing our cDTF method with different spatial parameter values. A simplification hierarchy of discs of various sizes is produced by employing different spatial parameters. The size of median filter σ_m is determined by σ_s. Similarly to other initial blurring operators (e.g., Gaussian filter and box filter) in previous work (Zhang et al. 2014; Cho et al. 2014), the median filter can be served as a scale selector that determines the scale of texture to be removed. Specifically, in the RGF (Zhang et al. 2014), the first iteration of joint bilateral filtering uses a constant guidance image which is identical to Gaussian smoothing. In each iteration of the BTF (Cho et al. 2014), a texture-free guidance image is first obtained using the box filter. As discussed in Cho et al. (2014), the basic requirement of this procedure is to have the image texture properly smoothed out so that even a structure-preserving smoothing technique may be employed here (see Fig. 3d for an example). However, we use a simple median filter to achieve this goal in the light of simplicity and efficiency.

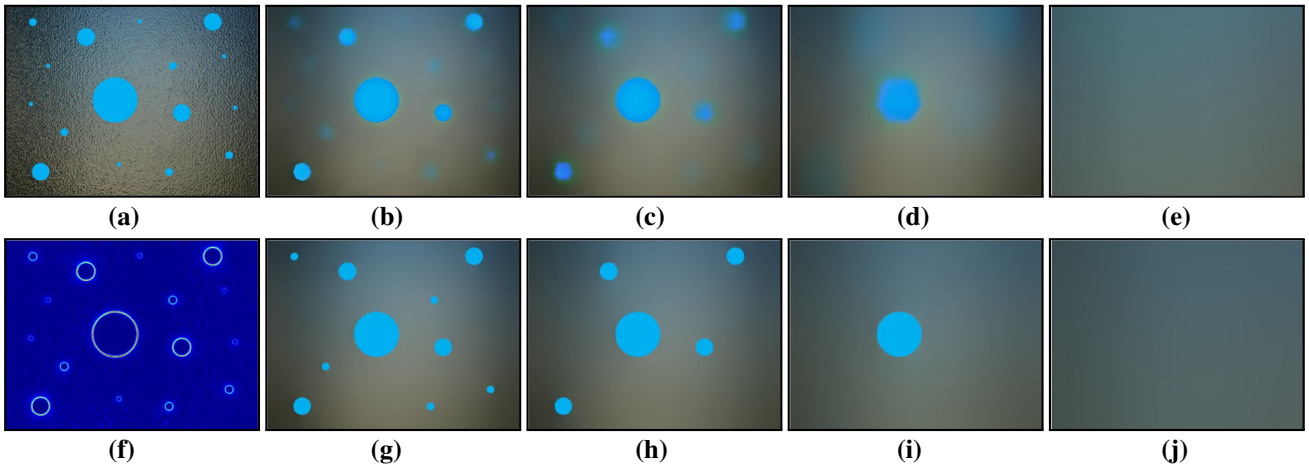

Fig. 18 The effects of spatial parameters. A simplification hierarchy of discs of various sizes is generated via different spatial parameters. From top to bottom and from left to right: input image, RGF ($\sigma_r = 0.1, \sigma_s = 15, 25, 50, 200$), input edge map, cDTF ($\sigma_r = 0.04, \sigma_s = 0.08, 0.15, 0.3, 0.6, \sigma_m = 8, 15, 30, 60$). **a** Input, **b** RGF ($\sigma_s = 15$), **c** RGF ($\sigma_s = 25$), **d** RGF ($\sigma_s = 50$), **e** RGF ($\sigma_s = 200$), **f** edge map, **g** our cDTF ($\sigma_s = 0.08$), **h** our cDTF ($\sigma_s = 0.15$), **i** our cDTF ($\sigma_s = 0.30$), **j** our cDTF ($\sigma_s = 0.60$)

5.3 Analysis of Different Edge Detectors

The edge detector (Dollár and Zitnick 2013) takes an image patch as input, and predicts the local edge confidence via random forest classifiers. Although the learned classifiers are better able to distinguish "true" edges from textures, they may fail in regions of low contrast, owing to a lack of global reasoning. On the contrary, state-of-the-art edge detectors (Kivinen et al. 2014; Bertasius et al. 2015a, b; Shen et al. 2015; Xie and Tu 2015; Yang et al. 2016) rely on deep learning techniques, which can learn hierarchical features ranging from local to global.

Figure 19 presents two examples that contain low-contrast regions. Note that the faint edges are difficult to recognize through a local patch, but are easier to infer from the whole image (Fig. 19a). Thus, the estimated edge map of the global method (Yang et al. 2016) (Fig. 19c) is more accurate than that of the local one (Dollár and Zitnick 2013) (Fig. 19b). Figure 19d, e show the results of the proposed cRTV filter for the input edge maps in (b) and (c), respectively. Of course, an accurate edge map is helpful for producing a sharp filtered image (Fig. 19e). However, when the edge detector fails in low-contrast regions, the filtered image (Fig. 19d) will be blurred in those regions, in a similar manner as for other methods (Xu et al. 2012; Zhang et al. 2014; Cho et al. 2014) (Fig. 19f–h) that rely on color differences.

5.4 Comparison of Proposed Filters

The smoothing framework of WLS optimizes a global objective that provides more advantages than the recursive filters DTF and RBF. And it is also possible to incorporate a recursive filter into such a framework to deal with complex cases. In this section, we conduct an experiment on image colorization, using imprecise user scribbles to demonstrate the differences between the proposed filters. Recall that the original data term of WLS defined in Eq. (16) enforces a hard constraint with respect to the given observation, through a per-pixel cost function. However, this assumption may be violated in applications where the input data is inaccurate. An and Pellacini (2008) proposed using the aggregated data term to handle erroneous input data in several image editing applications. However, the all-pairs constraint in An and Pellacini (2008) requires solving a dense linear system, which is computationally expensive. A similar but more efficient data aggregation mechanism is discussed in Xu et al. (2009), Min et al. (2014), and is formulated as follows:

$$E_{data} = \sum_{p \in \Omega} \left(\sum_{q \in N_D(p)} c_{p,q}(g)(s_p - u_q)^2 \right), \tag{22}$$

where N_D represents a set of neighbors used to aggregate the input data. In contrast to the smooth prior defined in the N_4/N_8 neighbors method, in the case of $N_D(p)$ it is recommended to use more neighbors to integrate large supports. Here, $c_{p,q}$ is defined as an edge-preserving filter kernel, e.g., the bilateral filter kernel $\exp(-(p-q)^2/2\delta_s^2 - (g_p - g_q)^2/2\delta_r^2)$, and it determines the contribution of each neighbor. Combining the smooth term with Eq. (17), the final image is obtained by solving the following linear system (Farbman et al. 2008; Min et al. 2014):

$$(\mathbf{D} + \lambda \mathbf{L})\mathbf{s} = \mathbf{C}\mathbf{u}, \tag{23}$$

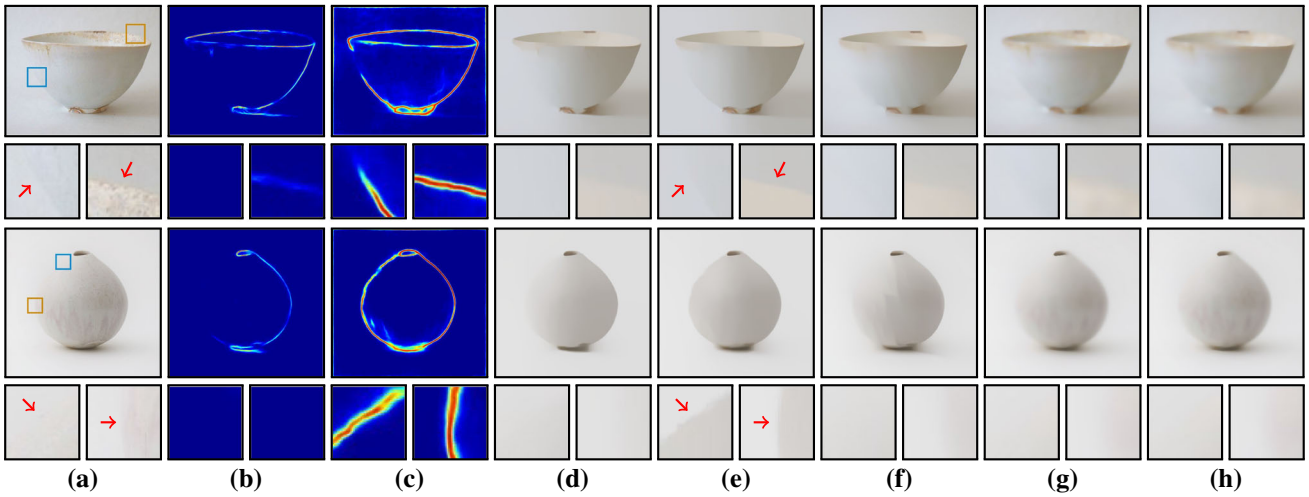

Fig. 19 Filtering results on low-contrast images. **a** Note that the faint edges are difficult to recognize via a local patch, but are easier to infer through the whole image. **b**, **c** Thus, the edge detector (Yang et al. 2016) that takes advantage of global reasoning is more accurate than the local one (Dollár and Zitnick 2013). **d**, **h** Of course, an accurate edge map is helpful for producing a sharp filtered image. However, when the edge detector fails in low-contrast regions, the filtered image will be blurred in those areas, similarly to other methods that rely on color differences. **a** Input, **b** edge map (Dollár and Zitnick 2013), **c** edge map (Yang et al. 2016), **d** ours with (**b**), **e** ours with (**c**), **f** RTV (Xu et al. 2012), **g** BTF (Cho et al. 2014), **h** RGF (Zhang et al. 2014) (Color figure online)

where $\mathbf{L}$ is a five-point/nine-point spatially inhomogeneous Laplacian matrix, $\mathbf{C}$ is a kernel matrix whose nonzero elements are given by the weights $c_{p,q}$, and $\mathbf{D}$ is a diagonal matrix whose diagonal values are the sum of the weights $\sum_{q\in N_D(p)} c_{p,q}$. The un-normalized bilateral filtering $\mathbf{Cu}$ and the sum of bilateral weights $\mathbf{D}$ can be efficiently computed using an $O(N)$ bilateral filter, such as the RBF. Specifically, the two terms are computed by solving the recursive system defined in Eq. (4) twice, with the input image $\mathbf{u}$ and a constant all-ones image, respectively. If $c_{p,q}$ is a normalized bilateral kernel (e.g., DTF), then $\sum_{q\in N_D(p)} c_{p,q} = 1$ (i.e., $\mathbf{D}$ is an identity matrix), and $\mathbf{Cu}$ is the exact smoothed input data. In this case, Eq. (23) becomes

$$(\mathbf{I} + \lambda\mathbf{L})\mathbf{s} = \mathbf{Cu}. \tag{24}$$

Compared with the original WLS (Farbman et al. 2008) system $(\mathbf{I} + \lambda\mathbf{L})\mathbf{s} = \mathbf{u}$, we can easily see that Eq. (24) just represents two successive smoothing operations on the input signal $\mathbf{u}$,

$$\mathbf{s} = (\mathbf{I} + \lambda\mathbf{L})^{-1}(\mathbf{C})\,\mathbf{u}. \tag{25}$$

In comparison with the hard data constraint, the pre-smoothed soft constraint is more robust against errors that may exist in the given observation. In fact, any edge-preserving filter or hybrid of such filters can be iteratively applied to Eq. 25. Figure 20 presents a colorization example where the colors of the input scribbles are imprecise. We apply the proposed cDTF, cRBF, and cWLS methods iteratively to handle this case. As can be seen from Fig. 20, the recursive approaches (d, g) do not deal with imprecise inputs effectively compared with the global one (b, c), and require more iterations (e, h) to obtain competitive results (An and Pellacini 2008; Min et al. 2014). In contrast, the recursive filters can be directly applied to the global framework (Eq. 25) to achieve a better performance (f, i).

6 Conclusion

In this paper, an efficient scale-aware edge-preserving filtering framework has been proposed. Unlike the current state-of-the-art filters, which use low-level vision features to design the filter kernel, the proposed technique is developed based on a higher-level understanding of the image structures. The integration of edge models trained from human-labeled datasets enables the proposed filters to better preserve structured edges that can be detected by the human visual system. As a result, it is more robust to objects/structures of different sizes/scales.

The proposed technique cannot be directly applied to other edge-preserving filters, such as the guided filter (He et al. 2013), and most of the quantization-based fast bilateral filters (Durand and Dorsey 2002; Pham and van Vliet 2005; Chen et al. 2007; Paris and Durand 2009; Yang et al. 2009; Adams et al. 2009, 2010; Gastal and Oliveira 2012). Such filters do not rely on recursive operations thus may be more suitable for GPU parallelization. Figure 21a–d shows that the guided filter is vulnerable to textures when a constant filter

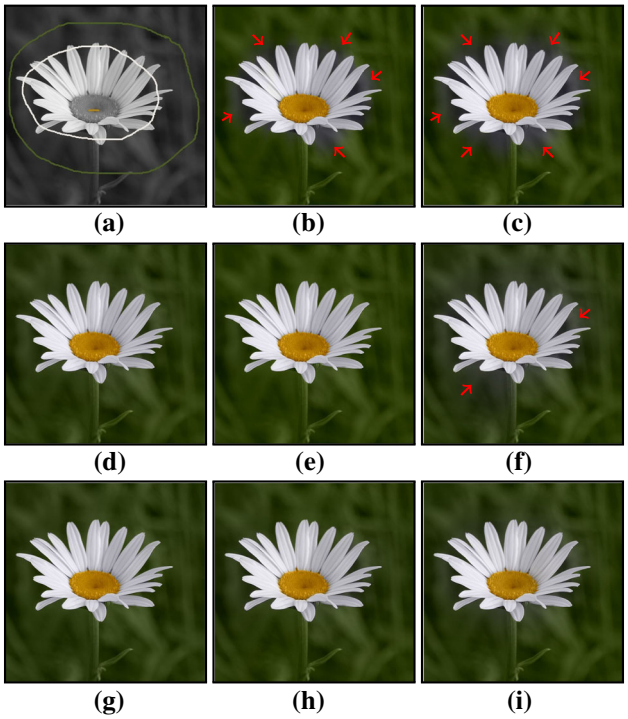

Fig. 20 Colorization with imprecise color scribbles. Iteratively applying an edge-preserving filter is helpful for handling imprecise inputs. However, the recursive filters cRBF and cDTF are not as effective as the cWLS, and require more iterations to obtain competitive results. In contrast, the recursive filters can be directly applied in Eq. (25) to obtain a better performance. **a** Input, **b** cWLS after 2 iter., **c** cWLS after 10 iter., **d** cRBF after 2 iter., **e** cRBF after 10 iter., **f** cRBF + cWLS., **g** cDTF after 2 iter., **h** cDTF after 10 iter., **i** cDTF + cWLS (Color figure online)

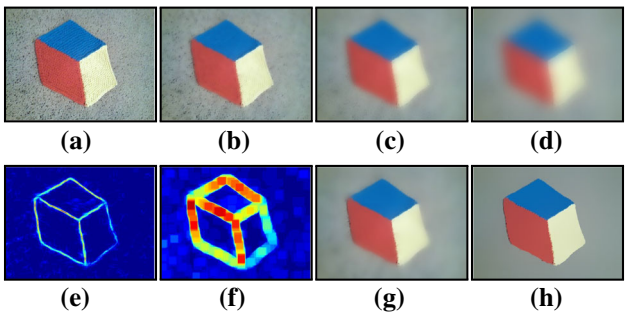

Fig. 21 Limitations. The guided filter is not effective for removing textures, as can be seen in **b**–**d**. The proposed technique can be adjusted for integration with a guided filter, but the quality will be lower than for anisotropic filters (e.g., the recursive bilateral filter in **h**). **a** Input, **b** $r = 4$, $\varepsilon = 0.2^2$, **c** $r = 10$, $\varepsilon = 0.5^2$ guided filter (He et al. 2013), **d** $r = 20$, $\varepsilon = 0.9^2$, **e** Conf., **f** Min Conf., **g** Conf. + GF, **h** our cRBF

kernel is employed. A simple extension is to adjust the edge confidence to adaptively control the guided filter kernel, so that a small kernel will be used around edges. Figure 21e–f present the edge confidence before and after minimum filtering, and Fig. 21g shows the guided filtered image obtained using an adaptive kernel based on the edge confidence in (f). This outperforms the original guided filter in terms of suppressing textures, while remaining capable of maintaining the most salient structure edges. However, the quality is obviously lower than that achieved with the integration of an anisotropic filter, as shown in Fig. 21h. A generalized extension for other edge-preserving filters will be investigated in the future.

Acknowledgements We thank all the reviewers for valuable comments. This work was supported by the National Basic Research Program of China (Grant No. 2015CB351705), the State Key Program of National Natural Science Foundation of China (Grant No. 61332018).

References

Adams, A., Baek, J., & Davis, A. (2010). Fast high-dimensional filtering using the permutohedral lattice. *CGF*, *29*(2), 753–762.

Adams, A., Gelfand, N., Dolson, J., & Levoy, M. (2009). Gaussian kd-trees for fast high-dimensional filtering. *ACM TOG (SIGGRAPH)*, *28*, 21:1–21:12.

An, X., & Pellacini, F. (2008). Appprop: All-pairs appearance-space edit propagation. *ACM TOG (SIGGRAPH Asia)*, *27*(3), 40:1–40:9.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detectionand hierarchical image segmentation. *IEEE TPAMI*, *33*, 898–916.

Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., & Malik, J. (2014). Multiscale combinatorial grouping. In *CVPR*.

Arnheim, R. (1956). *Art and visual perception: A psychology of the creative eye*. Berkeley: University of California Press.

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

Avidan, S., & Shamir, A. (2007). Seam carving for content-aware image resizing. *ACM TOG (SIGGRAPH)*, *26*(3), 10.

Bertasius, G., Shi, J., & Torresani, L. (2015a). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*.

Bertasius, G., Shi, J., & Torresani, L. (2015b). High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In *ICCV*.

Bousseau, A., Paris, S., & Durand, F. (2009). *User-assisted intrinsic images*. *ACM TOG (SIGGRAPH Asia)*, *28*, 130:1–130:10.

Boyadzhiev, I., Bala, K., Paris, S., & Durand, F. (2012). User-guided white balance for mixed lighting conditions. *ACM TOG (SIGGRAPH Asia)*, *31*(6), 200:1–200:10.

Buades, A., & Lisani, J. L. (2016). Directional filters for color cartoon+texture image and video decomposition. *Journal of Mathematical Imaging and Vision*, *55*(1), 125–135.

Canny, J. (1986). A computational approach to edge detection. In *IEEE TPAMI*.

Catanzaro, B., Su, B. Y., Sundaram, N., Lee, Y., Murphy, M., & Keutzer, K. (2009). Efficient, high-quality image contour detection. In *ICCV*.

Chambolle, A., & Darbon, J. (2009). On total variation minimization and surface evolution using parametric maximum flows. *IJCV*, *84*(3), 288–307.

Chen, J., Paris, S., & Durand, F. (2007). Real-time edge-aware image processing with the bilateral grid. *ACM TOG (SIGGRAPH)*, *26*(3), 103.

Cho, H., Lee, H., Kang, H., & Lee, S. (2014). *Bilateral texture filtering*. *ACM TOG (SIGGRAPH)*, *33*(4), 128:1–128:8.

Criminisi, A., Sharp, T., Rother, C., & Perez, P. (2010). Geodesic image and video editing. *ACM TOG*, *29*(5), 134.

Dani, A. L., Lischinski, D., & Weiss, Y. (2004). Colorization using optimization. *ACM TOG (SIGGRAPH)*, *23*, 689–694.

Dollár, P., Tu, Z., & Belongie, S. (2006). Supervised learning of edges and object boundaries. In *CVPR*.

Dollár, P., & Zitnick, C. L. (2013). Structured forests for fast edge detection. In *ICCV*.

Dollár, P., & Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE TPAMI*.

Donoho, D., Chui, C., Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, *21*(1), 5–30.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Durand, F., & Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. *ACM TOG (SIGGRAPH)*, *21*(3), 257–266.

Eisemann, E., & Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *ACM TOG (SIGGRAPH)*, *23*(3), 673–678.

Farbman, Z., Fattal, R., & Lischinski, D. (2010). Diffusion maps for edge-aware image editing. *ACM TOG (SIGGRAPH Asia)*, *29*(6), 145:1–145:10.

Farbman, Z., Fattal, R., Lischinski, D., & Szeliski, R. (2008). Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM TOG (SIGGRAPH)*, *27*(3), 67.

Fattal, R. (2009). Edge-avoiding wavelets and their applications. *ACM TOG (SIGGRAPH)*, *28*(3), 1–10.

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

Gastal, E., & Oliveira, M. (2012). Adaptive manifolds for real-time high-dimensional filtering. *ACM TOG (SIGGRAPH)*, *31*(4), 33:1–33:13.

Gilboa, G. (2014). A total variation spectral framework for scale and texture analysis. *SIAM Journal of Imaging Sciences*, *7*(4), 1937–1961.

Gupta, S., Arbeláez, P. A., Girshick, R. B., & Malik, J. (2015). Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation. *IJCV*, *112*(2), 133–149.

He, K., Sun, J., & Tang, X. (2013). Guided image filtering. *IEEE TPAMI*, *35*, 1397–1409.

Karacan, L., Erdem, E., & Erdem, A. (2013). Structure-preserving image smoothing via region covariances. *ACM TOG (SIGGRAPH Asia)*, *32*(6), 176:1–176:11.

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

Kyprianidis, J. E., & Döllner, J. (2008). Image abstraction by structure adaptive filtering. In *Proceedings of EG UK theory and practice of computer graphics, Manchester, United Kingdom, 2008* (pp. 51–58).

Kyprianidis, J. E., & Kang, H. (2011). Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum*, *30*(2), 593–602.

Levin, A., Lischinski, D., & Weiss, Y. (2006). A closed form solution to natural image matting. In *CVPR*.

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

Lischinski, D., Farbman, Z., Uyttendaele, M., & Szeliski, R. (2006). Interactive local adjustment of tonal values. *ACM TOG (SIGGRAPH)*, *25*(3), 646–653.

Margolin, R., Zelnik-Manor, L., & Tal, A. (2014). How to evaluate foreground maps. In *CVPR*.

Meyer, Y. (2001). *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. Providence: American Mathematical Society.

Min, D., Choi, S., Lu, J., Ham, B., Sohn, K., & Do, M. N. (2014). Fast global image smoothing based on weighted least squares. *IEEE TIP*, *23*(12), 5638–5653.

Paris, S., & Durand, F. (2009). A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, *81*, 24–52.

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

Parisand, S., Hasinoff, S. W., & Kautz, J. (2011). Local laplacian filters: Edge-aware image processing with a Laplacian pyramid. *ACM TOG (SIGGRAPH)*, *30*(4), 68:1–68:12.

Perazzi, F., Krahenbuhl, P., Pritch, Y., & Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *CVPR* (pp. 733–740).

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEE TPAMI*, *12*, 629–639.

Perreault, S., & Hbert, P. (2007). Median filtering in constant time. *IEEE TIP*, *16*(9), 2389–2394.

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

Pham, T. Q., & van Vliet, L. J. (2005). Separable bilateral filtering for fast video preprocessing. In *ICME*.

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

Rhemann, C., Hosni, A., Bleyer, M., Rother, C., & Gelautz, M. (2011). Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*.

Ren, X., & Liefeng, B. (2012). Discriminatively trained sparse code gradients for contour detection. In *NIPS*.

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

Shen, W., Wang, X., Wang, Y., Bai, X., & Zhang, Z. (2015). Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR* (pp. 3982–3991).

Subr, K., Soler, C., & Durand, F. (2009). Edge-preserving multiscale image decomposition based on local extrema. *ACM ToG (SIGGRAPH Asia)*, *28*(5), 147.

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *ICCV* (pp. 839–846).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, *13*(4), 600–612.

Weickert, J. (1999). Coherence-enhancing diffusion filtering. *IJCV*, *31*(2–3), 111–127.

Weiss, B. (2006). Fast median and bilateral filtering. *ACM TOG (SIGGRAPH)*, *25*(3), 519–526.

Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of IEEE international conference on computer vision*.

Xu, K., Li, Y., Ju, T., Hu, S. M., & Liu, T. Q. (2009). Efficient affinity-based edit propagation using k-d tree. *ACM ToG (SIGGRAPH Asia)*, *28*(5), 118:1–118:6.

Xu, L., Lu, C., Xu, Y., & Jia, J. (2011). Image smoothing via l0 gradient minimization. *ACM TOG (SIGGRAPH Asia)*, *36*(6), 174.

Xu, L., Yan, Q., & Jia, J. (2013). A sparse control model for image and video editing. *ACM TOG (SIGGRAPH Asia)*, *32*(6), 197.

Xu, L., Yan, Q., Xia, Y., & Jia, J. (2012). Structure extraction from texture via relative total variation. *ACM TOG (SIGGRAPH Asia)*, *31*(6), 139.

Yan, Q., Xu, L., Shi, J., & Jia, J. (2013). Hierarchical saliency detection. In *CVPR*.

Yang, J., Price, B., Cohen, S., Lee, H., & Yang, M. H. (2016). Object contour detection with a fully convolutional encoder-decoder network. In *CVPR*.

Yang, Q. (2012). Recursive bilateral filtering. In *ECCV* (pp. 399–413).

Yang, Q. (2016). Semantic filtering. In *CVPR*.

Yang, Q., Ahuja, N., & Tan, K. (2015). Constant time median and bilateral filtering. *IJCV*, *112*(3), 307–318.
Yang, Q., Tan, K. H., & Ahuja, N. (2009). Real-time o(1) bilateral filtering. In CVPR (pp. 557–564).
Yang, Q., Wang, S., & Ahuja, N. (2010). Svm for edge-preserving filtering. In *CVPR* (pp. 1775–1782).
Yin, W., Goldfarb, D., & Osher, S. (2005). Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In *VLSM* (pp. 73–84).
Yoon, K. J., & Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE TPAMI*, *28*(4), 650–656.
Zeune, L., van Dalum, G., Terstappen, L. W. M. M., van Gils, S. A., & Brune, C. (2016). Multiscale segmentation via Bregman distances and nonlinear spectral analysis.CoRR arXiv:1604.06665.
Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., & Mech, R. (2015). Minimum barrier salient object detection at 80 fps. In *ICCV*.
Zhang, Q., Shen, X., Xu, L., & Jia, J. (2014). Rolling guidance filter. In *ECCV*.
Zheng, S., Tu, Z., & Yuille, A. (2007). Detecting object boundaries using low-,mid-, and high-level information. In *CVPR*.
Ziou, D., & Tabbone, S. (1998). Edge detection techniques: An overview. *IEEE TPAMI*, *8*, 537–559.
Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV*.
Zitnick, C. L., & Parikh, D. (2012). The role of image understanding in contour detection. In *CVPR*.