The semi-guided bilateral filter

Article *in* IET Image Processing · May 2017

DOI: 10.1049/iet-ipr.2016.0418

CITATIONS
3

READS
170

4 authors:

Ba Son Thai
La Trobe University
7 PUBLICATIONS **8** CITATIONS
SEE PROFILE

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

Guang Deng
La Trobe University
129 PUBLICATIONS **1,147** CITATIONS
SEE PROFILE

Zhuo Su
Sun Yat-Sen University
55 PUBLICATIONS **172** CITATIONS
SEE PROFILE

IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Semi-guided bilateral filter

ISSN 1751-9659
Received on 6th June 2016
Revised 17th February 2017
Accepted on 28th April 2017
doi: 10.1049/iet-ipr.2016.0418
www.ietdl.org

Ba Thai[1] ✉, *Mukhalad Al-nasrawi*[1,2], *Guang Deng*[1], *Zhuo Su*[3]

[1]*Department of Electronic Engineering, La Trobe University, Bundoora, Victoria 3086, Australia*
[2]*Al-Musaib Technical College, Al-Furat Al-Awsat Technical University, 51009 Babylon, Iraq*
[3]*School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, People's Republic of China*
✉ *E-mail: t.thai@latrobe.edu.au*

Abstract: The bilateral filter (BF) is a non-linear filter that spatially smooths images with awareness of large structures such as edges. The level of smoothness applied to a pixel is constrained by a photometric weight, which can be obtained from the same image to be filtered (in case of the original BF) or from a guided image (in case of the joint/cross BF). In this study, the authors propose a new filter called the semi-guided BF which is derived from solving a non-linear constraint least square problem. The proposed filter's photometric weight incorporates information from the image to be filtered and the guided image. They propose a fast implementation of the filter based on layer approximation. They also study the iterative application of the proposed filter and show that the filter can preserve large structures while smoothing out small structures. This makes the proposed filter an efficient and effective tool for structure-aware image smoothing. Experimental results have demonstrated that performance of the proposed filter is comparable to those of the state-of-the-art algorithms.

1 Introduction

The bilateral filter (BF) is an edge-aware image smoothing technique in which it blurs small details such as noise or fragmented contours while preserving large textures such as edges. It was first developed by Aurich and Weule [1] in the form of a non-linear Gaussian filter, and was then reformulated with the name BF by Tomasi and Manduchi [2]. The key idea of the BF is to add to the standard Gaussian low-pass filter an extra photometric weight. This extra weight results in less smoothing in pixels with significant intensity difference. A number of works have been developed to utilise the BF in various image processing and computer vision applications, such as image denoising [2], detail enhancement [3], tone management [4, 5], stylisation [6], haze removal [7] and interpolation [8, 9].

The origin, interpretation, extension and fast implementation of the BF have been intensively studied. As mentioned in [10, 11], the BF can be derived from the result of a robust estimation. It is also related to anisotropic diffusion [12] and weighted least square (WLS) [13]. The connection between the BF and anisotropic diffusion is also studied in [14] in the context of adaptive smoothing. In addition to the original formulation, the BF has been modified in different ways and extended to enhance its performance based on particular applications, such as double BF [15] and recursive bilateral filtering [16]. The computational complexity of the BF has also been improved by its fast implementation methods including polynomial techniques [17, 18], layered approximation [4], bilateral grid [19] and distributive histograms [20].

An important extension of the BF is the joint BF which is applied in the context of digital photography [21]. In this application, the photometric weight for a no-flash image is obtained from a flash image. Using a similar approach, Eisemann and Durand [22] renamed the filter cross BF (CBF). The original BF is a special case of the CBF when the photometric weight is computed from the same image to be filtered. The CBF has been applied in many image processing applications such as dehazing [23], stereo matching [24, 25], image fusion [26] and texture suppression [27].

The CBF has been extended for scale-aware image smoothing by Zhang *et al.* [28] in which it is called the rolling guidance filter (RGF). The key idea of the RGF is to first smooth out the input image by a Gaussian low-pass filter, and then apply an iterative CBF to the same input image in which the photometric weight is derived from the previous filter output. This process will smooth out small structures and recover edges of large structures progressively.

Motivated by the success of the BF and its extensions, in this work, we first formulate the optimisation problem for the image filtering as a non-linear constraint least-square problem and then derive its iterative solution based on the strategy of alternating optimisation [29]. The first iteration is called the semi-guided BF (SBF). The key idea of the SBF is to derive the photometric weight not only from one image but also from different images. This allows the SBF to exploit information from both images. Especially, when applied iteratively the proposed filter has an important property which preserves the shapes of edges while smoothing small structures.

The key contributions and organisation of this work is summarised as follows. In Section 2, we briefly review related work regarding to the BF and its extensions. In Section 3, we formulate the image filtering problem as a non-linear constraint least-square problem and derive an iterative solution. The fast implementation for the first iteration is proposed. We then study the characteristics of the iterative application. In Section 4, we present experimental results of the proposed filter in the context of structure-aware image smoothing and image decomposition based applications. In the final section, we present concluding remarks.

In the remaining of this paper, we use the following notations. Let I, G and Y be the original image, the guided image and the processed image, respectively. We use the subscript to indicate the pixel location, e.g. the pixel of the image I at location p is denoted as I_p. A neighbourhood of a pixel p is denoted as Ω_p. N is the number of pixels in an image. We also define a weight function $w_\sigma(x) = \exp[-x^2/\sigma]$. A normalisation factor C_p for the filtering process of pixel p is defined as $C_p = \sum_{q\in\Omega_p} w_s(||p-q||)w_r(|A_p - B_q|)$, where $||p-q||$ is the Euclidean distance between the two pixel locations p and q, A and B are a pair of images which will be defined accordingly.

2 Related work

2.1 BF and optimisation problem

The BF is defined as

$$Y_p = \frac{1}{C_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|I_p-I_q|)I_q \quad (1)$$

Many studies [10, 11, 30] have shown that the BF can be interpreted from a robust estimation point of view. Paris *et al.* [30] pointed out that the BF is a solution of the following optimisation problem:

$$Y_p = \arg\min_{Y_p}\sum_{p\in N}\left((Y_p-I_p)^2+\sum_{q\in\Omega_p}\Phi(Y_p-Y_q)\right) \quad (2)$$

where Φ is the error norm. The key idea in (2) is to minimise the difference between the input and output while enforcing local smoothness. The BF is the result of this optimisation when $\Phi(x)=w_\sigma(x)$. Different variants of the BF can be obtained with other Φ functions. A study of possible Φ functions can be found in [30].

Elad [10] proposes a local minimisation problem which is reformulated by Caraffa *et al.* [11] as

$$Y_p = \arg\min_{Y_p}\sum_{q\in\Omega_p} w_s(||p-q||)\Phi((Y_p-I_q)^2) \quad (3)$$

The BF is resulted from the first iteration of gradient descent of the above optimisation with $\Phi(x)=w_\sigma(x)$ and $Y^{(0)}=I$. The key idea in (3) is to minimise the residual errors between a pixel in the output and a local neighbourhood in the input.

2.2 CBF and optimisation problem

Initially proposed by Petschnigg *et al.* [21] and later revisited by Eisemann and Durand [22], the CBF is an extension of the BF in which the photometric weight w_r is computed from a guided image. The CBF is defined as

$$Y_p = \frac{1}{C_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|G_p-G_q|)I_q \quad (4)$$

The CBF is the close form solution of the following optimisation problem:

$$Y_p = \operatorname{argmin}_{Y_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|G_p-G_q|)(Y_p-I_q)^2 \quad (5)$$

The key idea in (5) is to minimise sum of weighted square difference between the filter output Y_p and a input pixel I_q within neighbourhood Ω_p. The first weight $w_s(||p-q||)$ is the spatial constraint. The second weight $w_r(||G_p-G_q||)$ is a constraint using the information from the guided image which is physically related to the image to be filtered. The original BF is a special case of the CBF when $G=I$. The CBF was initially used in flash/no-flash applications. It was later extended in other applications such as scale-aware image smoothing [28] and edge-aware texture suppression [27].

2.3 Iterative application of BF and CBF

In this section, we study the formulations and properties of the BF and CBF when they are applied iteratively. Let $Y^{(k)}$ denote the filtering output of an iterative filter at kth iteration ($k \geqslant 0$).

- The iterative BF (IBF) is defined as

$$Y_p^{(k+1)} = \frac{1}{C_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|Y_p^{(k)}-Y_q^{(k)}|)Y_q^{(k)} \quad (6)$$

where $Y^{(0)}=I$.

As proven in [10, 14], the BF can be regarded as a non-iterative form of an anisotropic diffusion. This implies that when the BF is applied iteratively, it tends to diffuse the input image progressively and it eventually leads to a piecewise constant signal. However, due to the constraint of the photometric weight, the diffusion occurs simultaneously with the process of sharpening the edges. This effect is demonstrated in Fig. 1 for a one-dimensional (1D) signal containing a small, a medium and a large structure. The IBF can remove small structures by increasing the number of iterations. However, this also leads to oversharpened edges of medium and large structures. The experiment on an image is shown in Fig. 2 in which the iterative outputs have a cartoon look due to the oversharpening effects. This makes the IBF suitable for segmentation based applications, such as texture separation [30], image and video abstraction [6].

- The iterative CBF is defined as

$$Y_p^{(k+1)} = \frac{1}{C_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|Y_p^{(k)}-Y_q^{(k)}|)I_q \quad (7)$$

where $Y^{(0)}=0$ (a completely black image).

Zhang *et al.* [28] exploited the CBF to set up an iterative filter for scale-aware image smoothing. They named their filter as RGF. The effects of RGF depend on the initialised value of $Y^{(0)}$. Experiments of different settings of $Y^{(0)}$ can be found in [28]. The important property of RGF can be shown in the case $Y^{(0)}=0$ which effectively uses a Gaussian low-pass filter in the first iteration. This process removes small structures and blurs edges of large structures. However, these edges can be recovered when the number of iteration is increased while small structures remain flat as in the first iteration. This behaviour on a 1D signal is demonstrated in Fig. 1 which shows that the RGF has similar oversharpening effect to large edges as that of the IBF. However, unlike the IBF, there is no diffusion occurring in the RGF along with the process of recovering edges. This makes the RGF converge to a signal that contains both medium and large structures having straight edges. The effect of the RGF on a real image is shown in Fig. 2. Due to this important property, the RGF can be utilised in a number of applications, such as scale-aware image smoothing [28], texture separation, detail enhancement and image abstraction (Fig. 3).

3 Semi-guided BF

3.1 Problem formulation and general solution

Inspired by the derivation of BF studied in [10, 11, 30], the proposed filter is formulated as the solution to the following optimisation problem:

$$\{Y_p\}_{p\in N} = \arg\min_{Y_p}\sum_{q\in\Omega_p} w_s(||p-q||)w_r(|I_p-Y_q|)\Phi(Y_p-Y_q) \quad (8)$$

where the term $\Phi(Y_p-Y_q)$ enforces local smoothness in the output, the term $w_r(|I_p-Y_q|)$ is used to constrain the distance between a pixel in the input image and all pixels in a corresponding neighbourhood in the output, and $w_s(||p-q||)$ is a spatial constraint.

We apply the strategy of alternating optimisation [29] to determine the solution for the optimisation. Suppose after kth iteration, we have an image $Y^{(k)}$. At the next iteration, we fix $Y_q^{(k)}$ when solving for Y_p. Equation (8) then becomes

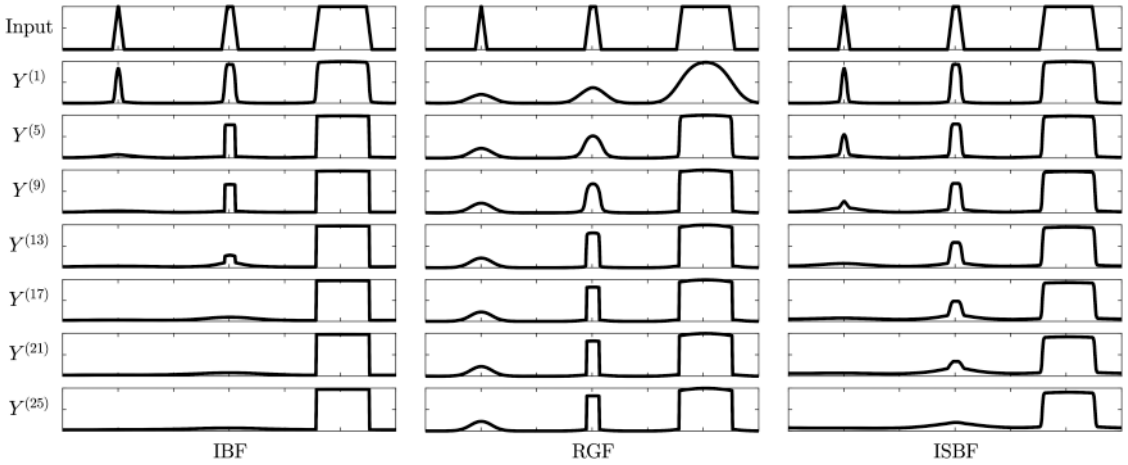

Fig. 1 *Illustration of the effects of IBF (left column), RGF (middle column) and ISBF (right column) on a 1D signal. The first row is the original signal and the following rows are iteratively filtered signals. The two parameters σ_s and σ_r are the same for all filters*

Fig. 2 *Example of IBF (top row), RGF (middle row) and ISBF (bottom row) on a real image. The two parameters are set as $\sigma_s = 3.5$ and $\sigma_r = 0.1$ for all three filters. From top to bottom: IBF, RGF and ISBF. From left to right: 1 iteration, 5 iterations, 9 iterations and 13 iterations*

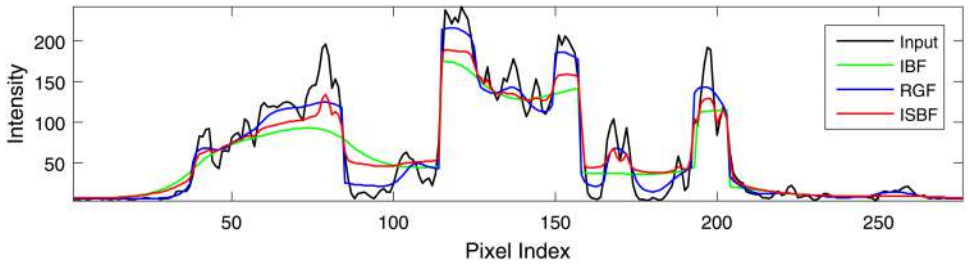

Fig. 3 *One horizontal line extracted from input image and images in last column of Fig. 2*

$$\{Y_p\}_{p \in N} = \arg\min_{Y_p} \sum_{q \in \Omega_p} w_s(||p - q||) w_r(|I_p - Y_q^{(k)}|) \Phi(Y_p - Y_q^{(k)}) \quad (9)$$

The solution satisfies the following equation:

$$\sum_{q \in \Omega_p} w_s(||p - q||) w_r(|I_p - Y_q^{(k)}|) \Phi'(Y_p - Y_q^{(k)}) = 0 \quad (10)$$

where Φ' is the first derivative of Φ with respect to Y_p.

Let $x = Y_p - Y_p$ then $\Phi(x)$ represents the local smoothness cost in the filtered image. This cost is an increasing function of the intensity difference x. Also, in the homogeneous region ($x = 0$), the cost is minimal ($\Phi(x) = 0$). Based on this characteristics of the function $\Phi(x)$, in this work we study a particular case when $\Phi(x) = (1/2)x^2$. Solving (10), we have the solution at $(k+1)$th iteration as

$$Y_p^{(k+1)} = \frac{1}{C_p} \sum_{q \in \Omega_p} w_s(||p - q||) w_r(|I_p - Y_q^{(k)}|) Y_q^{(k)} \tag{11}$$

3.2 One iteration and its fast implementation

When $Y^{(0)} = G$ the first iteration of (11) becomes

$$Y_p = \frac{1}{C_p} \sum_{q \in \Omega_p} w_s(||p - q||) w_r(|I_p - G_q|) G_q \tag{12}$$

where the iteration index is omitted to simplify the notations.

The photometric weight of this filter is obtained from the image to be filtered and a guided image. For these reasons, the proposed filter is called the semi-guided BF (SBF). The BF is a special case of the SBF when $G = I$.

To unify BF related filters, we define a generic filter function $\psi(I, A, B)$ that takes three inputs, including an image to be filtered I and two guided images A and B. The filter output at pixel p is defined as

$$\psi_p(I, A, B) = \frac{1}{C_p} \sum_{q \in \Omega_p} w_s(||p - q||) w_r(|A_p - B_q|) I_q \tag{13}$$

The function ψ can be used to represent the BF, CBF and SBF as shown in Table 1.

A direct implementation for the filter function $\psi(I, A, B)$ consists of two nested loops for p and q. This is the same as the brute-force implementation of the BF. As outlined in [30], the complexity of brute-force BF implementation is $\mathcal{O}(|N|^2)$. Due to the characteristics of the Gaussian filter, the spatial weight is significantly smaller when the spatial distance between two pixels are greater than $2\sigma_s$, where σ_s is the variance of the spatial Gaussian filter. Consequently, the neighbourhood pixel q can be restricted to as $||p - q|| \leq 2\sigma_s$. This reduces the complexity to $\mathcal{O}(|N|\sigma_s^2)$.

We develop a fast algorithm based on the fast implementations of the BF. As summarised in [30], the fast BF includes separable kernel [31], local histograms [20], layered approximation [4] and bilateral grid [19]. Among these techniques, layered approximation and bilateral grid are widely used for their accuracy and fast computational time [19, 30]. The key idea of these two techniques is to subsample the image along the intensity axis and downsample in the spatial domain.

Following the idea of layered approximation outlined in [4, 30], our algorithm consists of three main steps. We first subsample the intensity range of image B into n different values denoted as i_m, where $m = 1, \ldots, n$. For each i_m, we compute a corresponding layer L_m such that its value at pixel p is computed as $L_m(p) = w_r(|I_p - i_m|) I_p$. In the second step, we convolve each layer L_m with the spatial kernel g_{σ_s} and the result is normalised to form a new layer $\bar{L}_m$ that contains exact results of $\psi(I, A, B)$ for pixels with an intensity equal to i_m. In the last step, we perform a linear interpolation between L_{m1} and L_{m2} to obtain results of the filter at location p in which the two closest subsampled values to A_p are i_{m1} and i_{m2}.

Table 1 Relation of ψ to the BF, CBF and SBF

BF (1)	CBF (4)	SBF (12)
$Y_p = \psi_p(I, I, I)$	$Y_p = \psi_p(I, G, G)$	$Y_p = \psi_p(G, I, G)$

Algorithm: Fast implementation of the generic filter function $\psi(I, A, B)$

Inputs:

- Image I, guided images A and B.
- Gaussian parameters σ_s and σ_r. Number of layers n.

Output: Y

1. Subsample n values (i_m where $m = 1, \ldots, n$) from min(B) to max(B) and build n layers as
$$L_m(p) = w_r(|I_p - i_m|) I_p$$
2. Let g_{σ_s} and g_{σ_r}, respectively, be the spatial and intensity kernel of a Gaussian filter with variances σ_s and σ_r. Let $\div$ denote a per-pixel division. Further denote the sum of the weights at each pixel as $g_{\sigma_s} \oplus g_{\sigma_r}$. Convolve each layer L_m with the spatial kernel g_{σ_s} and normalise the result as
$$\bar{L}_m = g_{\sigma_s} \otimes L_m \div g_{\sigma_s} \oplus g_{\sigma_r}$$
3. For each pixel p in image A, find the two closest values i_{m1} and i_{m2} and perform a linear interpolation to obtain the result as
$$Y_p = \frac{A_p - i_{m1}}{i_{m2} - i_{m1}} \times \bar{L}_{m2} + \frac{i_{m2} - A_p}{i_{m2} - i_{m1}} \times \bar{L}_{m1}$$

Since we follow the same structure of the layered approximation to implement our filter, our implementation has the similar complexity as outlined in [30] which is $\mathcal{O}\big(|N| + (|N|/\sigma_s^2)(|R|/\sigma_r)\big)$ where N and R are, respectively, the number of pixels and intensity range of image I. However, in our proposed generic filter, we subsample the guided image B instead of image I, such that R is the intensity range of image B.

3.3 Iterative algorithm and its property

Equation (11) represents the formulation of our iterative SBF (ISBF) where $Y^{(0)} = I$. In order to exploit the fast implementation of $\psi(I, A, B)$ in iterative ways, we set up the IBF, RGF and ISBF as following:

- IBF
$$Y^{(k+1)} = \psi(Y^{(k)}, Y^{(k)}, Y^{(k)}) \tag{14}$$
where $Y^{(0)} = I$.
- RGF
$$Y^{(k+1)} = \psi(I, Y^{(k)}, Y^{(k)}) \tag{15}$$
where $Y^{(0)} = 0$.
- ISBF
$$Y^{(k+1)} = \psi(Y^{(k)}, I, Y^{(k)}) \tag{16}$$
where $Y^{(0)} = I$.

The block diagram of these algorithms is shown in Fig. 4.

In order to study the effects of the proposed ISBF, we experiment on the same 1D signal used for the IBF and the RGF as shown in Fig. 1. For a subjective comparison, the parameters σ_s and σ_r are kept the same for all three filters.

Suppose that p is the pixel to be filtered and q is its neighbourhood. The photometric weight w_r of the pixel q at kth

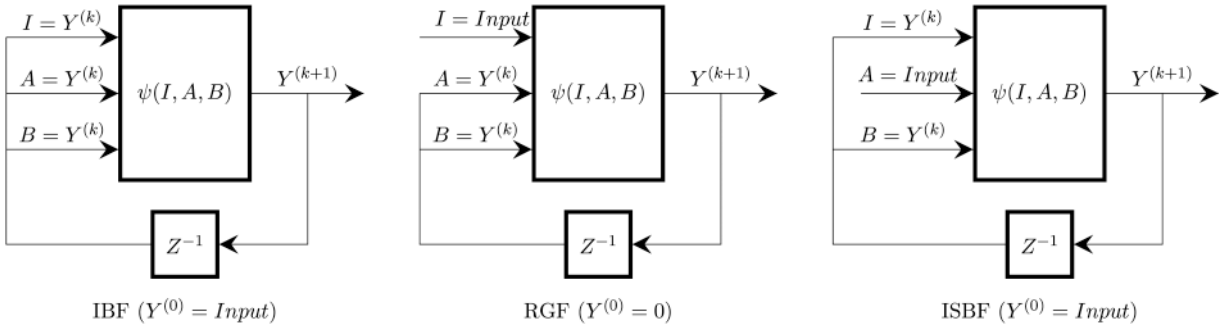

Fig. 4 *Block diagram of the IBF, the RGF and the proposed filter ISBF. Z^{-1} denote a delay function. The filter function $\psi(I, A, B)$ is defined in (13)*

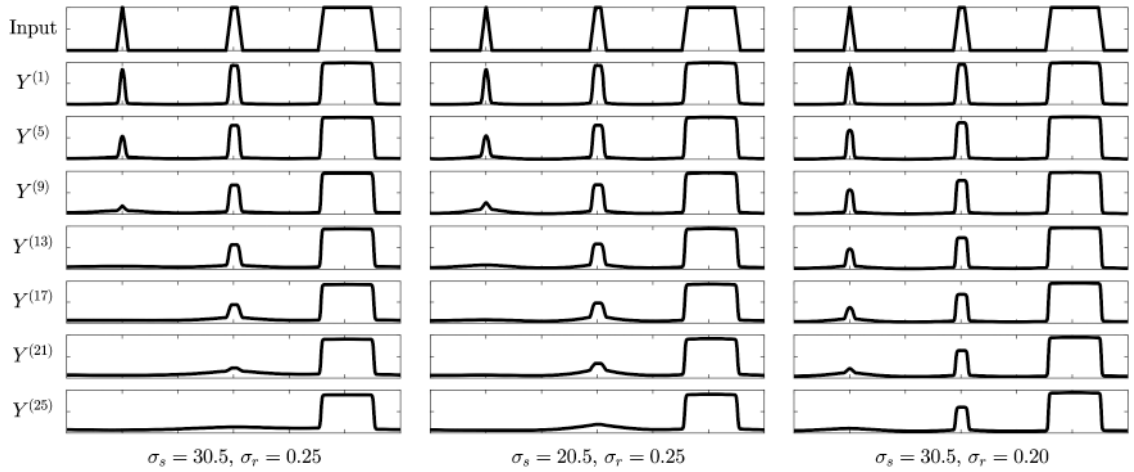

Fig. 5 *Effect of setting different values of σ_s and σ_r on ISBF filtering results for a 1D signal. The first row is the input signal, the following rows are filtering outputs*

iteration is a decreasing function of $|I_p - Y_q^{(k)}|$. The role of the photometric weight w_r is to control the smoothing level caused by a Gaussian filter characterised by a spatial weight w_s. Since $Y^{(0)} = I$, the ISBF begins by filtering the image with a BF that produces a smoother version of the input I. This implies $|I_p - Y_q^{(1)}| < |I_p - Y_q^{(0)}|$ which leads to $|Y_p^{(2)}| < |Y_p^{(1)}|$. The same effect can be observed when the number of iterations is increased. Consequently the level of sharpness of iterative outputs is decreased, which can be represented as $Y^{(k)} < Y^{(k-1)} < \cdots < Y^{(1)} < Y^{(0)} = I$. In addition, the level of smoothness depends on the size of structures. For example, small structures tend to be smoothed out faster than large structures. Hence, large structures can be preserved if the filter is stopped at an appropriate iteration. This effect is shown in Fig. 1 in which the proposed filter can progressively remove small structures without sharpening large edges. The filtering result of the proposed filter on a real image is shown in Fig. 2 in which small furs on the cat face are smoothed out progressively while the resultant images look naturally. A plot of one horizontal line extracted from the results of this experiment is presented in Fig. 3 which shows the difference in the filtering results by using the IBF, RGF and the proposed ISBF.

3.4 Parameters settings and convergence

3.4.1 Parameters settings and filtering results: An important property of the proposed filter is to smooth an image at different speed depending on the size of structures. This property is characterised by two parameters σ_s and σ_r. Similar to the BF, the role of σ_s in the proposed filter is to determine the spatially smoothing level. A neighbourhood pixel has more contribution on filtering the central pixel at larger value of σ_s. In addition, the role of w_r (a decreasing function of σ_r) is to reduce the smoothing effects applied to pixels with large intensity variation usually locating at edges. By selecting a smaller value of σ_r, the large edges can be retained in the first iteration. However, this requires more iterations to smooth out small structures. These effects are demonstrated in Figs. 5 and 6 for a 1D signal and a real image, respectively. In order to preserve edges, the two parameters need to be set to values such that the desirable structures are retained in the first iteration because the ISBF does not recover edges. It then depends on a particular application to adjust σ_r to an appropriate value. For example, σ_r should be increased if less iterations are required in small structure removal applications.

3.4.2 Convergence: Due to the smoothing process, the proposed filter eventually converges into a piecewise constant signal (an image having same intensity value for all pixels). Since the convergent result of the ISBF does not contain valuable information, we only study its convergent or smoothing speed. Knowing the convergent speed of the filter can be useful when determining the number of iterations for a particular application.

The ISBF produces progressively smoother versions of the original input image. This implies that the amount of small textures contained in the output at an iteration is less than that of in the output at the previous iteration. In addition, the convergent speed is mainly determined by small textures as they are smoothed out faster than large structures. As a result, when the number of iterations is increased, the variation between outputs of two successive iterations decreases. The sum of square difference (SSD) between two adjacent iterations is shown in Fig. 7. The figure also demonstrates that the SSD of the first iteration depends on the σ_s and σ_r for the first bilateral filtering process. Our empirical experiments have shown that after about 15 iterations, the variation of outputs between two successive iterations is visually unrecognisable.

4 Applications and results

In this section, we first highlight the important property of our ISBF filter in the context of structure-aware image smoothing. We

Fig. 6 *Effect of setting different values of σ_s and σ_r on ISBF filtering results for a real image. Top row: $\sigma_s = 8.5$, $\sigma_r = 0.05$. Middle row: $\sigma_s = 3.5$, $\sigma_r = 0.05$. Bottom row: $\sigma_s = 3.5$, $\sigma_r = 0.15$. From left to right: 1 iteration, 5 iterations, 9 iterations and 13 iterations*

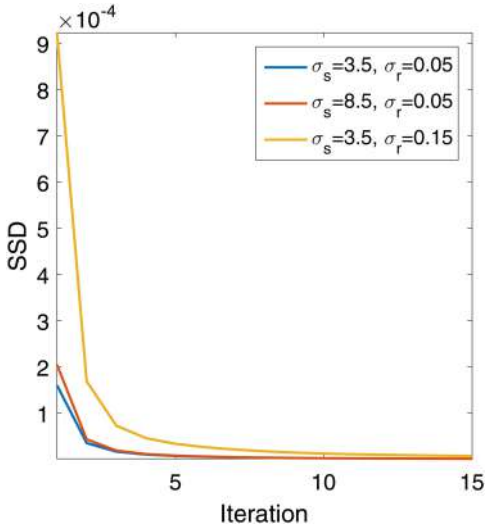

Fig. 7 *SSD between outputs of two successive iterations for the image in Fig. 6*

then demonstrate its effective applications based on image decomposition.

4.1 Structure-aware image smoothing

As discussed above, the proposed ISBF can remove small structures iteratively while maintaining natural shapes of edges of the remaining structures. Also, by adjusting its parameters, the proposed ISBF can eliminate structures at different scales. These properties make the proposed filter an effective tool for structure-aware image smoothing in which both sharp edges and large structures are preserved. These properties are demonstrated in the following three experiments in which the proposed filter is compared with the state-of-the-art filters, including guided image filtering (GIF) [32], domain transform (DMT) [33], gradient minimisation (L0) [34], WLS [35], Bayesian model averaging (BMA) filter [36] (GIF is proven to be a special case of BMA filter), BM3D [37] and relative total variation (RTV) [38].

In the first experiment which is shown in Fig. 8, the input image contains a number of stars at different scales. Its background also has different colours blended together naturally. This experiment aims to remove tiny stars while keeping the stars with significant sizes. Small stars have similar brightness as the big stars. Due to the effects of small structures removal, large stars are blurred in the edge-aware filters, such as BMA, BM3D, DMT and WLS. Conversely, scale-aware filters, such as L0, RGF and RTV can preserve sharpness of large stars. However, the background is unexpectedly segmented by these filters. On the contrary, due to the property of removing small structures without sharpening edges, our filter produces the desirable result such that small stars are removed, big stars maintain their sharpness and background has pleasant colour transition. As discussed in Section 3.4, since the purpose of this experiment is to remove small stars the σ_s would be set to a big value; however, in order to minimise the blur effects on big stars the value of σ_s cannot be too big. Also, the convergent iteration is empirically experimented to be about 15 iterations for most images. Consequently, the σ_s, σ_r and number of iterations for this experiment are set to 3.5, 0.1 and 10 iteratively, respectively.

In the second experiment which is shown in Fig. 9, we demonstrate the effectiveness of the ISBF in background smoothing applications. It is sometimes required to smooth the background of images to highlight the important information conveyed in the foreground. However, smoothing the background is often at the cost of losing details or sharpness in the foreground. In this experiment, we purposely adjust parameters of all filters to produce outputs such that the background is smoothed while keeping important information of the foreground. As can be seen in Fig. 9, our filter effectively smooths out the background while retaining details and sharpness in the foreground. Other methods tend to flatten details in the foreground or reduce the overall sharpness. In order to increase the smoothness level of the

Fig. 8 *Scale-aware smoothing results. Parameters are adjusted such that tiny stars are removed while big stars are retained. From left to right and top to bottom: input, BMA (r = 9, $\sigma_s = 0.0001$), BM3D ($\sigma_s = 30$), DMT ($\sigma_s = 5.5$, $\sigma_r = 1.0$, 5 iterations), WLS ($\lambda = 2.5$, $\alpha = 1.0$), L0 ($\lambda = 0.1$), RGF ($\sigma_s = 3.5$, $\sigma_r = 0.05$, 5 iterations), RTV ($\lambda = 0.002$, $\sigma = 3$, 3 iterations), our method ($\sigma_s = 3.5$, $\sigma_r = 0.1$, 10 iterations)*

Fig. 9 *Background smoothing results. All filters are tuned for their best performance in terms of background smoothing. From left to right and top to bottom: input, BMA (r = 25, $\sigma_s = 0.00005$), BM3D ($\sigma_s = 250$), DMT ($\sigma_s = 20.5$, $\sigma_r = 1.5$, 5 iterations), WLS ($\lambda = 2.5$, $\alpha = 1.5$), L0 ($\sigma_s = 0.075$), RGF ($\sigma_s = 9.5$, $\sigma_r = 0.05$, 5 iterations), RTV ($\lambda = 0.05$, $\sigma = 3$, 3 iterations), our method ($\sigma_s = 6.5$, $\sigma_r = 0.1$, 7 iterations)*

background, in this experiment the σ_s is increased to 6.5 compared to that of 3.5 in the first experiment.

In the third experiment, we aim to demonstrate the performance of the proposed filter in terms of smoothing noise while preserving important edges. For a subjective assessment, parameters of all filters are tuned such that they produce similar peak signal-to-noise ratio (PSNR) values. The experimental results are shown in Fig. 10 in which the proposed filter produces competitive results in comparison with other state-of-the-art edge-aware image filters.

The computational time of the all filters in these two experiments is presented in Table 2. By exploiting the fast implementation of a single iteration iteratively, the ISBF has competitive running time in comparison with other state-of-the-art filters.

4.2 Image decomposition based applications

Image decomposition is a process of separating an image into a piecewise smooth base layer and one or more detail layers. The base layer contains large structures with significant variations in intensity. Edge-aware smoothing operators are often used to extract the image base layer. The difference between the original image and the base layer defines the detail layer. These layers are used in various photographic applications including detail manipulation, pencil sketching and texture and details transfer. The following experiments demonstrate effectiveness of the proposed filter in image decomposition based applications.

4.2.1 Detail manipulation: Fattal *et al.* [39] proposed a method to employ BF-based image decomposition for shape and detail enhancement. Their method produces a sequence of outputs which

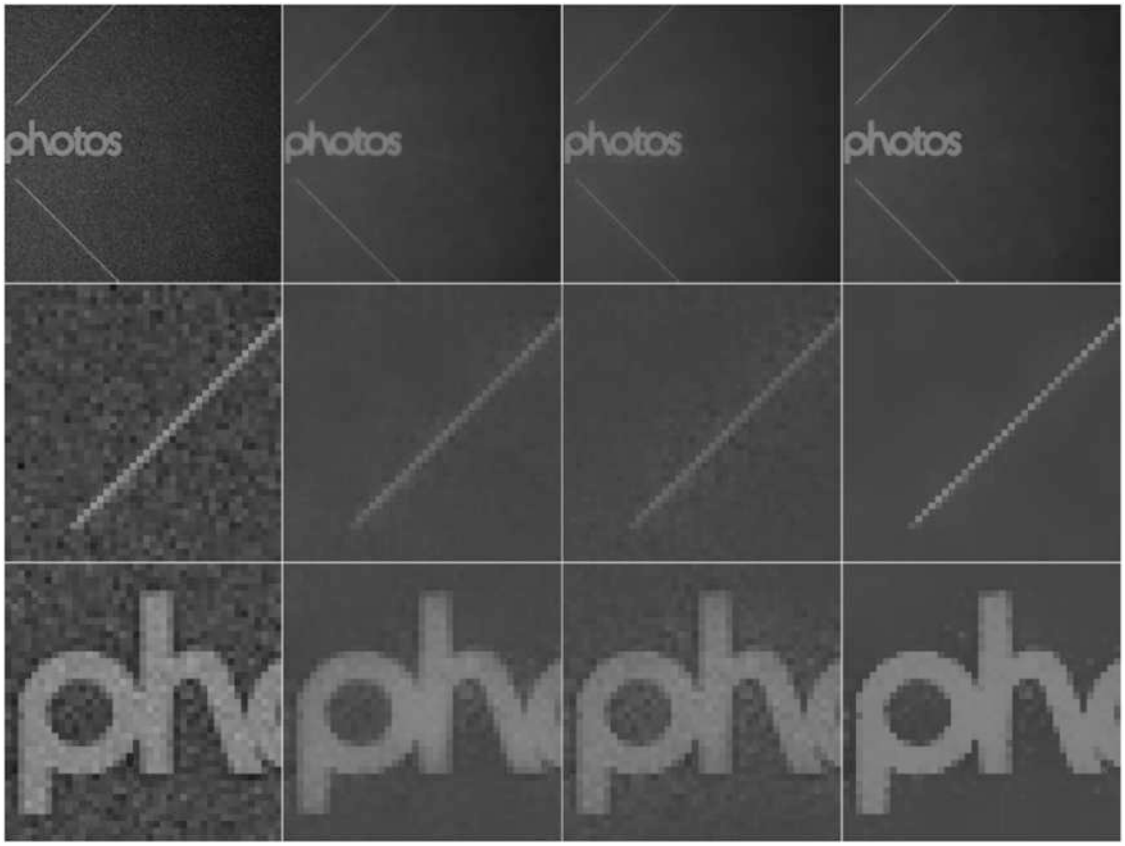

Fig. 10 *Edge-aware image filtering results. All filters are tuned to produce similar PSNR. Top row, from left to right: input, BMA mode 3 ($r = 3$, $\sigma_s = 255 \times 10^{-6}$, PSNR = 30.41), GIF ($\epsilon = 0.0105$, $\omega = 11$, PSNR = 30.43), our method ($\sigma_s = 4.5$, $\sigma_r = 0.1$, 5 iterations, PSNR = 30.40). Middle and bottom row are close-ups of corresponding images in the top row*

Table 2 Running time (seconds) of experiments on images in Figs. 8 and 9. All algorithms are implemented using MATLAB and run in a 3.6 GHZ Intel processor

Image	Method							
	BMA	BM3D	DMT	WLS	L0	RGF	RTV	ISBF
stars (0.48 MB)	0.20	2.13	13.05	2.53	1.40	1.35	1.72	2.35
leopard-lion (0.38 MB)	0.14	1.83	8.61	1.70	1.00	0.60	2.56	0.78

Fig. 11 *Detail enhancement based on multi-scale image decomposition. The running time (seconds) of Farbman et al. [35] and our method are 2.48 and 0.36 s, respectively*
(a) Input, ***(b)*** Farbman *et al.* [35], ***(c)*** Our result

contain details in different scales. Farbman *et al.* [35] improved this method with the use of their WLS edge-aware filtering technique. The decomposition results in different layers at progressively finer scales are obtained by taking difference between the two outputs of the WLS filter at successive iterations. Using a similar approach, we apply our filter to manipulate image details at different scales. Let $Y^{(0)}, Y^{(1)}, \ldots, Y^{(k)}$ denote progressively smoother versions of an image I and $Y^{(0)} = I$. The detail layers are obtained as $d^{(i)} = Y^{(i)} - Y^{(i+1)}$ which can be manipulated individually to produce a sequence of detail layers. An example of fine-scale detail enhancement is shown in Fig. 11 which demonstrates that the proposed filter produces similar result as that of Farbman *et al.* [35] at faster computational time.

4.2.2 Image stylisation: Image stylisation transforms a real image into a non-photorealistic image. The base layer is utilised together with other image information. For example, edges or gradients can be added to the base layer to produce cartoon-like or pencil-like images, respectively. Lu *et al.* [40] exploited image gradients to generate pencil drawing effects. The resultant of pencil drawing can be visually unpleasant due to the appearances of fragmented contours. In order to eliminate the undesirable gradients, image smoothing is applied to the input image prior to generating its pencil sketch. Fig. 12 shows pencil drawing effects using our filter, the BMA and L0 filters. The figure demonstrates that the image resulted from our filter has more details than that of BMA and L0 filters and has higher contrast than that of BMA and L0 filters.

Fig. 12 *Colour pencil sketching results*
(a) BMA, ***(b)*** L0, ***(c)*** Our method

Fig. 13 *Flash/no-flash detail transfer with denoising. In our implementation, the base layer and detail layer were extracted with ISBF ($\sigma_s = 8.5$, $\sigma_r = 0.2$, 5 iterations)*
(a) Image with flash, ***(b)*** Image without flash, ***(c)*** Petschnigg *et al.* [21], ***(d)*** Our result

4.2.3 Flash/no-flash detail transfer with denoising: Petschnigg *et al.* [21] proposed a technique to use a pair of images for applications in detail transfer with denoising. They observed that the no-flash image of a low-light environment contains important ambient illumination information, whereas the flash image of the same scene contains valuable amounts of details. Their method was to compose base layer of the no-flash image with detail layer of the flash image. Consequently, the resultant image has less noise than the no-flash image. It also contains more natural scene illumination than the flash image. They exploited the BF and CBF to extract base layer and detail layer, respectively. In this experiment, we utilise our ISBF filter for both base layer and detail layer extractions. As shown in Fig. 13, our result has better visibility in shadow areas and sharper edges than that of the Petschnigg *et al.*'s result.

5 Conclusion

In this paper, we have formulated the image filtering problem as a non-linear constraint least square problem and used the strategy of alternating optimisation to derive a new filter called the ISBF. One iteration of the proposed filter is called the SBF for which we have developed a fast implementation based on the idea of layer approximation. We have presented a detailed study of the important property of our filter in image smoothing with respect to shapes of structures. We have tested the proposed filter and compare its performance with those state-of-the-art filters in a number of applications. Experimental results show that the proposed filter is an efficient and effective alternative tool for a wide range of applications.

6 References

[1] Aurich, V., Weule, J.: 'Non-linear Gaussian filters performing edge preserving diffusion'. Proc. of the DAGM Symp., 1995, pp. 538–545
[2] Tomasi, C., Manduchi, R.: 'Bilateral filtering for gray and color images'. Proc. Int. Conf. Computer Vision (ICCV 1998), Bombay, India, 1998, pp. 839–846
[3] Zhang, B., Allebach, J.P.: 'Adaptive bilateral filter for sharpness enhancement and noise removal', *IEEE Trans. Image Process.*, 2008, **17**, (5), pp. 664–678
[4] Durand, F., Dorsey, J.: 'Fast bilateral filtering for the display of high-dynamic-range images', *ACM Trans. Graph.*, 2002, **21**, (3), pp. 257–266
[5] Paris, S., Durand, F.: 'Two-scale tone management for photographic look', *ACM Trans. Graph.*, 2006, **25**, (3), pp. 637–645
[6] Winnemöller, H., Olsen, S.C., Gooch, B.: 'Real-time video abstraction', *ACM Trans. Graph.*, 2006, **25**, (3), pp. 1221–1226
[7] Tripathi, A.K., Mukhopadhyay, S.: 'Single image fog removal using bilateral filter'. Proc. Int. Conf. Signal Processing, Computing and Control (ISPCC 2012), Hong Kong, 2012, pp. 1–6
[8] Han, J.-W., Kim, J.-H., Cheon, S.-H., *et al.*: 'A novel image interpolation method using the bilateral filter', *IEEE Trans. Consum. Electron.*, 2010, **56**, (1), pp. 175–181
[9] Hung, K.-W., Siu, W.-C.: 'Fast image interpolation using the bilateral filter', *IET Image Process.*, 2012, **6**, (7), pp. 877–890
[10] Elad, M.: 'On the origin of the bilateral filter and ways to improve it', *IEEE Trans. Image Process.*, 2002, **11**, (10), pp. 1141–1151
[11] Caraffa, L., Tarel, J.-P., Charbonnier, P.: 'The guided bilateral filter: When the joint/cross bilateral filter becomes robust', *IEEE Trans. Image Process.*, 2015, **24**, (4), pp. 1199–1208
[12] Perona, P., Malik, J.: 'Scale-space and edge detection using anisotropic diffusion', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990, **12**, (7), pp. 629–639
[13] Lagendijk, R.L., Biemond, J., Boekee, D.E.: 'Regularized iterative image restoration with ringing reduction', *IEEE Trans. Acoust. Speech Signal Process.*, 1988, **36**, (12), pp. 1874–1888
[14] Barash, D.: 'Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (6), pp. 844–847
[15] Jondhale, K., Sable, A.: 'Modified double bilateral filter for sharpness enhancement and noise removal'. Proc. Int. Conf. Advances in Computer Engineering (ACE), 2010, pp. 295–297
[16] Yang, Q.: 'Recursive bilateral filtering'. European Conf. on Computer Vision (ECCV 2012), 2012, pp. 399–413
[17] Dai, L., Yuan, M., Zhang, X.: 'Accelerate bilateral filter using Hermite polynomials', *Electron. Lett*, 2014, **50**, (20), pp. 1432–1434
[18] Dai, L., Yuan, M.: 'Optimal fitting polynomial for linear time bilateral filters', *Electron. Lett*, 2015, **51**, (16), pp. 1249–1251
[19] Paris, S., Durand, F.: 'A fast approximation of the bilateral filter using a signal processing approach', *Int. J. Comput. Vis.*, 2009, **81**, (1), pp. 24–52
[20] Weiss, B.: 'Fast median and bilateral filtering', *ACM Trans. Graph.*, 2006, **25**, (3), pp. 519–526
[21] Petschnigg, G., Szeliski, R., Agrawala, M., *et al.*: 'Digital photography with ash and no-ash image pairs', *ACM Trans. Graph.*, 2004, **23**, (3), pp. 664–672
[22] Eisemann, E., Durand, F.: 'Flash photography enhancement via intrinsic relighting', *ACM Trans. Graph.*, 2004, **23**, (3), pp. 673–678
[23] Xiao, C., Gan, J.: 'Fast image dehazing using guided joint bilateral filter', *Visual Comput.*, 2012, **28**, (6–8), pp. 713–721
[24] Mattoccia, S., Giardino, S., Gambini, A.: 'Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering'. Computer Vision – ACCV 2009, 2010, pp. 371–380
[25] Fu, L., Peng, G., Song, W.: 'Histogram-based cost aggregation strategy with joint bilateral filtering for stereo matching', *IET Comput. Vis.*, 2016, **10**, pp. 173–181
[26] Kumar, B.S.: 'Image fusion based on pixel significance using cross bilateral filter', *Signal Image Video Process.*, 2015, **9**, (5), pp. 1193–1204
[27] Su, Z., Luo, X., Deng, Z., *et al.*: 'Edge-preserving texture suppression filter based on joint filtering schemes', *IEEE Trans. Multimedia*, 2013, **15**, (3), pp. 535–548
[28] Zhang, Q., Shen, X., Xu, L., *et al.*: 'Rolling guidance filter'. European Conf. on Computer Vision (ECCV 2014), 2014, pp. 815–830
[29] Bezdek, J.C., Hathaway, R.J.: 'Some notes on alternating optimization'. Advances in Soft Computing – AFSS 2002, 2002, pp. 288–300
[30] Paris, S., Kornprobst, P., Tumblin, J., *et al.*: 'Bilateral filtering: theory and applications'. Foundations and Trends in Computer Graphics and Vision, 2009
[31] Pham, T.Q., Van Vliet, L.J.: 'Separable bilateral filtering for fast video preprocessing'. IEEE Int. Conf. on Multimedia and Expo, 2005, pp. 454–457

[32] He, K., Sun, J., Tang, X.: 'Guided image filtering', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (6), pp. 1397–1409
[33] Gastal, E.S., Oliveira, M.M.: 'Domain transform for edge-aware image and video processing', *ACM Trans. Graph.*, 2011, **30**, (4), Article 69, doi: 10.1145/2010324.1964964
[34] Xu, L., Lu, C., Xu, Y., *et al.*: 'Image smoothing via l0 gradient minimization', *ACM Trans. Graph.*, 2011, **30**, (6), Article 174, doi: 10.1145/2024156.2024208
[35] Farbman, Z., Fattal, R., Lischinski, D., *et al.*: 'Edge-preserving decompositions for multiscale tone and detail manipulation', *ACM Trans. Graph.*, 2008, **27**, (3), article 67, doi:10.1145/1399504.1360666
[36] Deng, G.: 'Edge-aware bma filters', *IEEE Trans. Image Process.*, 2016, **25**, (1), pp. 439–454
[37] Dabov, K., Foi, A., Katkovnik, V., *et al.*: 'Image denoising by sparse 3-d transform domain collaborative filtering', *IEEE Trans. Image Process.*, 2007, **16**, (8), pp. 2080–2095
[38] Xu, L., Yan, Q., Xia, Y., *et al.*: 'Structure extraction from texture via relative total variation', *ACM Trans. Graph.*, 2012, **31**, (6), article 139, doi: 10.1145/2366145.2366158
[39] Fattal, R., Agrawala, M., Rusinkiewicz, S.: 'Multiscale shape and detail enhancement from multi-light image collections', *ACM Trans. Graph.*, 2007, **26**, (3), article 51, doi:10.1145/1275808.1276441
[40] Lu, C., Xu, L., Jia, J.: 'Combining sketch and tone for pencil drawing production'. Proc. Int. Symp. Non-Photorealistic Animation and Rendering (NPAR 2012), 2012, pp. 65–73