

# Guided Adaptive Interpolation Filter

This paper is a postprint of a paper submitted to and accepted for publication in IET Image Processing and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library

# Guided Adaptive Interpolation Filter

 ISSN 1751-8644  
 doi: 0000000000  
 www.ietdl.org

 Waseem Waheed<sup>1</sup> Mukhalad Al-nasrawi<sup>2</sup> Guang Deng<sup>1</sup>
<sup>1</sup> Department of Engineering, La Trobe University, Melbourne, Australia

<sup>2</sup> Electrical Power Engineering Technique, Al-Furat Al-Awsat Technical University, Kufa, Iraq

\*, \* E-mail: w.waheed@latrobe.edu.au

**Abstract:** Edge-aware smoothing has proved to be a fundamental technique for various image processing and computer vision tasks. In this paper, we introduce a local, non-iterative and effective edge-preserving filter namely guided adaptive interpolation filter (GAIF). GAIF can be used as a post-processing step after any smoothing filter to improve its edge preservation performance without reformulation. GAIF has an  $O(N)$  computation complexity where  $N$  is the total number of pixels in the image. To further increase the efficiency of GAIF at edge-preservation, two techniques are introduced and demonstrated. GAIF efficiency is demonstrated and compared to state-of-the-art techniques on a number of tasks including image smoothing, flash/no-flash image denoising/fusion, single image dehazing and image details enhancement.

## 1 Introduction

Image smoothing is a fundamental tool for several applications such as edge detection, feature extraction, and image restoration. Conventional linear time-invariant (LTI) filters are utilized to remove noise. Although these filters are computationally efficient, they are oblivious to image content and structures usually resulting in undesirable visual effects. This is due to the use of a spatially-invariant kernels which leads to smoothing or enhancing both; image structure and noise.

To address this problem, researchers have developed and studied numerous non-linear alternatives (spatially varying kernels) called edge-aware filters. The goal of edge-aware filters is to avoid smoothing across significant boundaries while eliminating the unimportant details. There are several edge-aware filters including: bilateral filter [1], weighted least square filter [2], edge-avoiding wavelets [3], guided filter [4],  $L_0$  smoothing [5],  $L_1$  smoothing [6], region covariance [7], domain transform [8], local Laplacian filter [9], weighted median filter [10], fast global smoother [11], fast domain decomposition [12], the bilateral solver [13],  $L_0$  gradient projection [14], side window guided filtering [15] and guided wavelet filter [16].

In addition to edge-aware smoothing, these filters are broadly utilized in numerous applications in image processing and computational photography. Examples include image de-noising [12, 17], detail enhancement [3, 18], image fusion [19, 20], texture smoothing [11, 12, 21–23], single image haze removal [24], tone mapping of high dynamic range (HDR) images [2, 3, 10, 23, 25], anomaly detection in hyper-spectral images [26], object classification accuracy enhancement in hyper-spectral images [27], enhance the output of semantic segmentation algorithms [13], depth super-resolution/up-sampling [11, 13], image colourization [3, 11, 13], image colour quantization [12], scale-space filtering [12, 22], style transfer [10, 12], optical flow estimation [10], compression artifacts removal [14, 22], content-aware resizing and stereo matching [10].

From the earlier review of the current state-of-the-art edge-aware smoothing algorithms, it can be noted that most of them are based on the idea of preserving distinctive structures while smoothing small scale details. Inspired by the success of recently published works on edge-aware filters and their valuable applications, the goal of this work is to propose and investigate a new edge-aware filter called guided interpolation edge-aware filter (GAIF).

This work is motivated by the guided image filter (GIF) [4] and adaptive interpolation filter (AIF) [21]. The key idea of this work is that edge-aware smoothing can be obtained by a local interpolation between the input image and a guidance image which, in the simplest case, could be a linearly smoothed version of the input by using a Gaussian filter. A fundamental difference between this work and

those based on the AIF is that in this work the edges and the flat regions in the resultant image are locally selected from the original image and the guidance image, respectively, through interpolation process. On the other hand, in the AIF, the interpolation process is achieved by an iterative pixel-wise process over the entire image. Although the interpolation process in the AIF is achieved through a linear process, it is an iterative filter. As a result, the proposed filter is computationally more efficient than the AIF.

On the other hand, GIF assumes a patch-level linear model instead of the interpolation in GAIF, in other words, an output pixel is produced as a linear model of the patch centred at the corresponding pixel in a guidance image.

In the following sections, related works are summarized in section 2. The mathematical model of the proposed guided adaptive interpolation filter, an algorithm to solve it, and a way to extend it are presented in section 3. Section 4, is a discussion about the impact of parameter tuning and the smoothing performance. Applications demonstrating the efficiency of the proposed filter are presented in section 5. Finally, a brief discussion and a conclusion about the results are presented in section 6.

## 2 Related Work

Non-linear filters can be divided into two groups based on the locality of the filtering effect; local filters which represent most of the non-linear filters in the literature [28] and global filters which are usually the solutions to optimization problems such as the weighted least-squares based (WLS) filter [2] [29] [30].

### 2.1 Kernel-based filters

Smoothing in kernel-based methods is achieved through a weighted average of the input signal values to yield each element of the output. The kernel is used to measure the similarities between pixels. These similarities are normalized and used as the weights for the averaging. Specifically, the filtered pixel denoted  $y_i$  is computed from the pixels of the input image denoted  $\{x_j\}$  as shown below:

$$y_i = \sum_j W_{ij} x_j \quad (1)$$

where the weight  $W_{ij}$  is a function of the image to be filtered [1] or another image in the case of joint/cross-filtering [31]. Milanfar et al [28] has presented an excellent exposition about this kind of filters. Kernel-based filters are generally considered to be local; because, a

filtered pixel is computed as a weighted average of its surrounding pixels.

## 2.2 Guided Image Filter

The guided image filter (GIF) [4] has received a great deal of attention by the community because it has many desirable properties. For example, the filter formulation, intuitively, makes sense from a statistical regression perspective, and the algorithm is computationally efficient  $O(N)$ . In addition, filter results are very compelling. These qualities contributed to GIF's popularity and motivated other researchers to borrow ideas from it. Ham et al [30], adapted the idea of guidance image to the regularization term at the global level of image rather than the patch level as is done in the GIF. Li et al [32], introduced weight to the regularization parameter to enhance the edge-awareness of the original GIF. Lu et al [33] have proposed another weighting function which is more robust to the regularization parameter.

The original guided filter (GIF) [4] assumes a local linear patch model. A pixel at location  $p$  in the  $k$ th patch  $\Omega_k$  is represented as  $p \in \Omega_k$ . The pixel in output image  $J_{pk}$  (the subscript  $k$  refers to the  $k$ th patch) is related to the corresponding pixel in the guidance image  $G_p$  in the following way:

$$J_{pk} = a_k G_p + b_k \quad \forall p \in \Omega_k \quad (2)$$

where  $a_k$  and  $b_k$  are model parameters for the pixels in  $\Omega_k$ . They are determined by solving the following optimization problem:

$$\arg \min_{a_k, b_k} C(a_k, b_k) = \sum_{p \in \Omega_k} (a_k G_p + b_k - I_p)^2 + \epsilon a_k^2 \quad (3)$$

where  $\epsilon$  is a user specified regularization parameter. Since for a square-shaped patch of  $|\Omega_k|$  pixels, a pixel  $G_p$  belongs to  $|\omega_p|$  overlapping patches. Each resulting in an output patch. GIF takes the average of these outputs as the final filter output  $J_p$ .

$$J_p = \frac{1}{|\omega_p|} \sum_{k \in \omega_p} J_{pk} \quad (4)$$

where  $\omega_p$  is the set of patches to which the pixel  $p$  belongs. This is a simple model averaging process. For completeness here, we mention that other forms of model averaging can be adopted [34].

The idea of using a pair of images to produce the output image was first described in the joint/cross bilateral filter [35] [31] which included the guidance information in an *ad-hoc* fashion. The idea made disciplined in [4] [36] by modelling the image patches as a linear transformation to the corresponding patches in the guidance image. Extensions to this idea include using two guidance images [37], and making the guidance procedure global rather than local [38].

## 2.3 Energy minimization global filters

Most of the optimization-based filters are global filters. In another word, they minimize a cost function calculated over the whole image as opposed to the patch-oriented approach. In [39], Xu et al proposed the relative total variation (RTV) measure to distinguish between structures and texture. Later, RTV is used as a regularizer in a global optimization problem. RTV achieves good texture smoothing. Zhou et al [22] proposed a scale-aware measure and included it in an objective function to achieve scale-aware filtering called Iterative Global Optimization filter (IGO). Liu et al [40] proposed a global optimization model involving truncated Huber function, the resulting model is non-convex and non-smooth, leading to some desirable properties. The authors demonstrated the effectiveness of this model on a number of tasks. A major drawback in these methods is their computational complexity which comes from solving large linear systems [11] [41] [12].

## 2.4 Interpolation based filters

Al-nasrawi et al [21], proposed a pixel level edge-aware smoothing technique that utilises the idea of interpolation between two images, which are the observed/original image and a smooth version of observed/original. The filtering process proceeds in an iterative fashion and the interpolation weights are updated in each iteration based on the residual between the observed/original image and the current estimate image.

Unsharp masking [42] is a classical technique used to improve the sharpness of an image. Two versions of the input image are used to produce the result, a sharp negative version and a smoothed positive. The parameter in unsharp masking is usually fixed throughout the image domain. Main applications of unsharp masking revolve around contrast enhancement.

## 2.5 Edge-preserving filtering

Several edge-preserving smoothing operators have been proposed in the literature. One of the earliest of these operators is the bilateral filter [1], which has been used in numerous applications including HDR tone-mapping [43], [44] and highlight removal [45]. A major drawback in the smoothing performance of the bilateral filter is the gradient-reversal which results in halos when used for image enhancement [2]. Farbman et al [2] tackled the gradient reversal problem by solving a global optimization problem. He et al [4] proposed GIF, a more efficient filter, by solving a local optimization problem but the results still suffer from the halo artefacts [32]. Xu et al proposed an  $L_0$  adaptation of the total variation filter [46], which produces piece-wise constant results. Its performance was demonstrated on a number of applications. Ham et al [47] proposed the static-dynamic image filter which solves a global non-convex optimization problem that involves two guidance images: the current estimate and an external image. The authors have demonstrated its effectiveness at texture removal and depth super-resolution. However, they noted that the filter produces artefacts in flash/no-flash and RGB-NIR denoising tasks.

## 2.6 Our contribution

The novel contributions of this work are as follows:

1. General framework for patch-based interpolation is presented, along with algorithms for two special cases.
2. To further enhance the edge-preserving performance of the proposed filter, two weighting functions, that boost or suppress the penalty term based on the image content, are introduced and compared.
3. A relationship between GAIF and GIF is highlighted where a special case of GAIF is also a special case of GIF.

## 3 Guided Adaptive Interpolation Filter

### 3.1 Definition

Smoothing images can readily be achieved using any linear low-pass filter such as the Gaussian filter. However, the resulting filtered image is equally smoothed everywhere regardless of the image contents. To rectify the lack of discriminating power in linear filters, we propose a local, patch-level, interpolation model. This local interpolation is between the corresponding patches of two images. The first image, denoted by  $I$ , is the raw image to be filtered. The second is a smooth image, denoted by  $M$ , produced using any linear or non-linear smoother applied on  $I$  as follows:

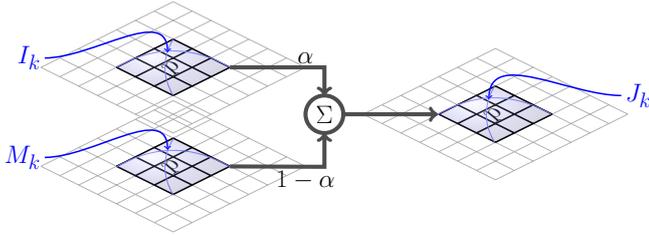
$$M = f(I) \quad (5)$$

where  $f(\cdot)$  is the smoother/filter of choice. More concretely, and adopting the same notation as the GIF,  $J_{pk}$  is the output pixel at location  $p$  due to the local model derived from the  $k$ th patch. We use

$J$  to denote the interpolation between  $I$  and its smooth version  $M$  in the patch  $\Omega_k$ :

$$J_{pk} = \alpha_k I_p + (1 - \alpha_k) M_p \quad \forall p \in \Omega_k \quad (6)$$

where  $\alpha_k$  is an interpolation parameter and is assumed to be constant in the patch  $\Omega_k$ ,  $p$  is the index of pixels within the patch  $\Omega_k$ . Each patch  $\Omega_k$  has  $|\Omega_k|$  pixels. The output patch  $J_{pk}$  in (6) is closer to  $I_p$  if  $\alpha_k$  is higher than 0.5, and this should be the case if the patch  $k$  is part of an edge. Alternatively, the output patch  $J_{pk}$  is closer to  $M_p$  if  $\alpha_k$  is lower than 0.5, this should be the case if the patch  $k$  is not part of an edge, i.e. smooth area. In other words, the images  $I$  and  $M$  are responsible for the edges and the smooth areas in the image  $J$  respectively.



**Fig. 1:** GAIF block-diagram. Patches at index  $k$  in the images  $I$  and  $M$  are interpolated to produce the corresponding patch in image  $J$

To determine the interpolation coefficient  $\alpha_k$ , we seek a solution to (6) by minimising the difference between the approximate image  $J$  and the image to be filtered  $I$ . Concretely, we minimise the following cost function for each patch  $k$ :

$$C(\alpha_k) = \sum_{p \in \Omega_k} |\alpha_k I_p + (1 - \alpha_k) M_p - I_p|^\gamma + \epsilon \alpha_k^2 \quad (7)$$

where  $\epsilon$  is a regularization parameter stopping  $\alpha$  from blowing up and controlling the amount of emphasis placed on  $I$  and  $M$ . The parameter  $\gamma \in \{1, 2\}$  generalises two distinct models which, apparently, have different solutions but are similar in performance. We will show the reason behind that. The solutions of the model (7) are as follows:

- $\gamma = 1$

$$\arg \min_{\alpha_k} C(\alpha_k) = \sum_{p \in \Omega_k} |\alpha_k I_p + (1 - \alpha_k) M_p - I_p| + \epsilon \alpha_k^2 \quad (8)$$

$$\alpha_k = \min \left( 1, \frac{|\Omega_k| \text{MAE}_k}{2\epsilon} \right) \quad (9)$$

where  $|\Omega_k|$  is the number of pixels in the patch and

$$\text{MAE}_k = \sum_{p \in \Omega_k} |I_p - M_p| / |\Omega_k|$$

- $\gamma = 2$

$$\arg \min_{\alpha_k} C(\alpha_k) = \sum_{p \in \Omega_k} (\alpha_k I_p + (1 - \alpha_k) M_p - I_p)^2 + \epsilon \alpha_k^2 \quad (10)$$

$$\alpha_k = \frac{\text{MSE}_k}{\text{MSE}_k + \tilde{\epsilon}_k} \quad (11)$$

where  $\tilde{\epsilon}_k = \epsilon / |\Omega_k|$  and  $\text{MSE}_k = \sum_{p \in \Omega_k} (I_p - M_p)^2 / |\Omega_k|$ .

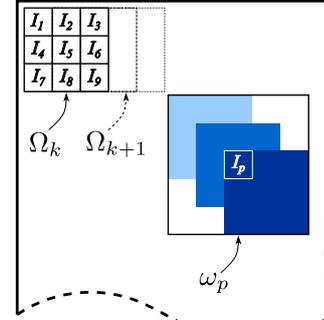
Just as is the case in GIF [4], the pixel at location  $p$  belongs to  $|\omega_p|$  overlapping patches as shown in 2. Which means  $|\omega_p|$  output

patches  $J_{pk}$  are produced per pixel  $p$ . Following the model averaging principle, we take the average of these outputs as the final output as follows:

$$J_p = \frac{1}{|\omega_p|} \sum_{k \in \omega_p} J_{pk} \quad (12)$$

$$= \bar{\alpha}_p I_p + (1 - \bar{\alpha}_p) M_p \quad (13)$$

where  $\bar{\alpha}_p = \frac{1}{|\omega_p|} \sum_{k \in \omega_p} \alpha_k$ .

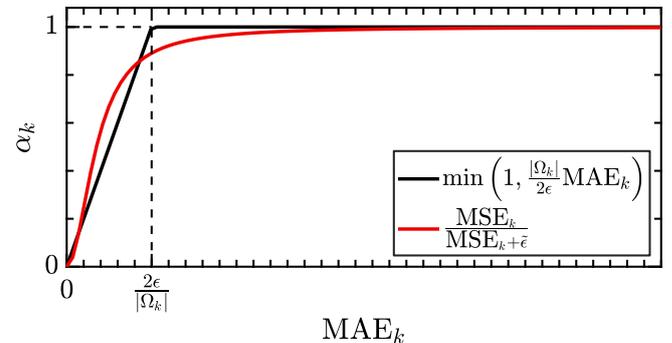


**Fig. 2:** Overlapping patches

**3.1.1 Analysis:** The solutions in (9) and (11) are in terms of the patch MAE and MSE, respectively which are defined earlier. The square of MAE can be considered as an approximation of MSE. As such, to simplify the following analysis we assume  $\text{MSE} \approx \text{MAE}^2$ . With this assumption in mind, a one-to-one comparison is facilitated. In Figure 3, we have started with a pair of  $|\Omega_k|$  and  $\epsilon$  in equation (9) then we found the best corresponding pair of  $|\Omega_k|$  and  $\epsilon$  in equation (11) such that the second function is closest to the first in the  $\ell_2$  sense.

This former parameter tuning allows us to highlight the difference between the two cases when they are the closest to each other. Two key observations here; (1) The two filters treat a patch  $\Omega_k$  of the input image in a generally similar fashion. In other words, at the very low and the very high values of MAE the two functions are equal. (2) The two filters differ around the  $\frac{2\epsilon}{|\Omega_k|}$  point. Before this point, we see higher  $\alpha$  for the  $\gamma = 2$  case signifying that the output patch gets a higher contribution from the input image. After the  $\frac{2\epsilon}{|\Omega_k|}$  point, the opposite occurs, the output patch gets a lower contribution from the input image than in the  $\gamma = 1$  case. Interestingly, the  $\gamma = 1$  filter produces  $\alpha = 1$  for  $\text{MAE} > \frac{2\epsilon}{|\Omega_k|}$ .

However, experimentally we have found it very hard to visually discern the differences between the two images resulting from the two filters thus, from this point onward whenever we mention GAIF, we are referring to the  $\gamma = 2$  case.



**Fig. 3:** Comparison between the solutions in (9) and (11).

### 3.2 Weighted adaptive interpolation filter

From the earlier discussion in 3.1, the proposed scheme produces each output pixel by interpolating the patches centred around the corresponding pixels in the two input images, see Figure 1. The choice of which patch  $I_k$  or  $M_k$  contributes more to the output pixel is encoded as  $\alpha_k$ . In other words, both images  $I$  and  $M$  contribute to the final GAIF filtering result  $J$  as can be seen in (6). More specifically, strong edges are contributed by  $I$  while smooth areas are contributed by  $M$ . This means, it is preferable to have  $\alpha_k \approx 1$  at the locations of edges and  $\alpha_k \approx 0$  elsewhere.

The proposed model in (7), has a regularisation term that puts a cost on choosing  $\alpha = 1$  which is what the solution would be without the extra regularisation term. However, the impact of this regularisation term is controlled by a single tuning parameter  $\epsilon$  which is fixed for the whole image.

This leads us to question the possibility of automatically adjusting the tuning parameter  $\epsilon$  such that the impact of the regularisation parameter becomes negligible, allowing the model to pick  $\alpha_k \approx 1$  at the edges, and amplify the impact of the regularisation parameter in smooth areas to allow the solution to be  $\alpha_k \approx 0$

To this end, we propose the following model:

$$\arg \min_{\alpha_k} C(\alpha_k) = \sum_{p \in \Omega_k} (\alpha_k I_p + (1 - \alpha_k) M_p - I_p)^2 + \tilde{\epsilon}_k \alpha_k^2 \quad (14)$$

where  $\tilde{\epsilon}_k = \epsilon \theta_k$  which means that  $\epsilon$  is tuned at the patch level rather than globally as is the case in the initial model (7). Li et al [32] proposed such formulation for the GIF. In this work, two variants of  $\theta_k$  are proposed as follows:

Define a re-scaling function

$$\phi(x) = A - \frac{Ax}{\beta + |x|} \quad (15)$$

$$\textbf{Variant 1:} \quad \theta_k = \phi(\eta_1 * \eta_2) \quad (16)$$

$$\text{where } \eta_1 = \frac{1}{N} \sum_{p \in \Omega_k} |I_p - \mu_p|$$

$$\text{and } \eta_2 = \frac{1}{N} \sum_{k \in \omega_p} \frac{1}{\eta_1 + c}$$

$$\textbf{Variant 2:} \quad \theta_k = \phi(\sigma_p(\eta_{1:5}(I))) \quad (17)$$

$$\text{where } \eta_i(I) = \text{MEDFILT}(I, 3 + 2(i - 1))$$

where MEDFILT represents a 2D median filter operation.

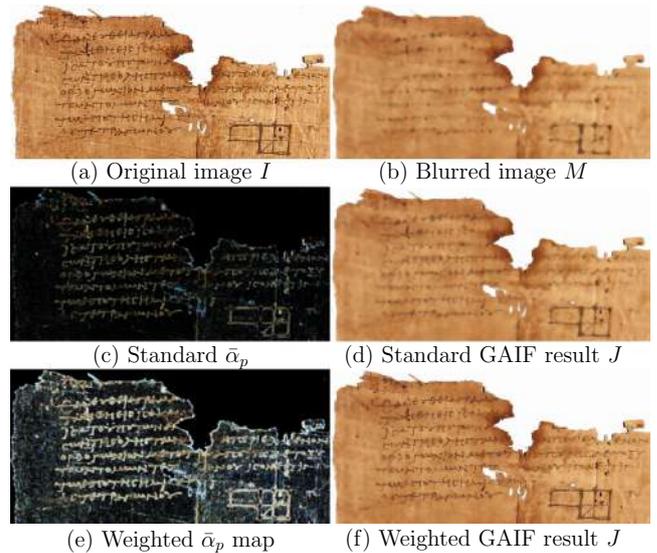
The parameters in equation (15) were found and set empirically to  $A = 5$  and  $\beta = 0.025$  throughout this paper. The role of the rescaling function in (15) is twofold, firstly; it flips the sign of its argument, secondly; it makes the output saturate at  $2A$  for negative inputs and saturate at 0 for large positive inputs. In other words, the rescaling function  $\phi(\cdot)$  makes sure that the value of  $\theta_k$  is always positive and bounded thus avoiding potential numerical issues.  $c$  in (16) is a small constant\*.  $\eta_{1:5}$  in (17) means a stack of  $\eta_i : i \in \{1..5\}$ .

It is important here to note that these variants result in slightly different smoothing effects and this can be clearly seen in Figure 4.

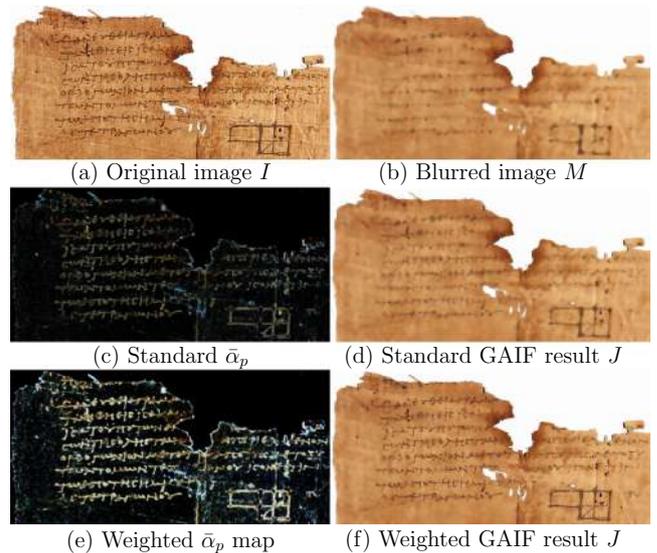
In variant 1, the pixels in a window are used in the computation of  $\theta_k$  which measures the relative mean absolute deviation of the central pixel in a window to the mean absolute deviations of the surrounding pixels followed by the re-scaling function  $\phi(x)$  to bound the scaling of  $\alpha_k$  within the range  $\{0,5\}$ .

In variant 2, the input image  $I$  is filtered with five median filters of increasing windows sizes producing five different values for each pixel, the standard deviation of the five samples is re-scaled with  $\phi(x)$  to produce the final scaling parameter  $\theta_k$ .

Figures 4 5 illustrate the way  $\bar{\alpha}_p$  changes after adding the weighted regularization across the image domain. Darker regions represent areas where the filter leans towards the smooth version i.e more emphasis on  $M$ , while light regions capture important edges where the filter leans towards the original version i.e more emphasis on  $I$ .



**Fig. 4:** Variant 1 of weighted GAIF demonstration. (a) is input image  $I$ . (b) is median filtered version of  $I$  with window size = 11. (c) and (e) are  $\bar{\alpha}_p$  image in the case of standard GAIF (11) and variant 1 of the weighted GAIF (14) respectively. Light areas represent more contribution from  $I$  than  $M$  and darker regions represent more contribution from  $M$  than  $I$ . (d) and (f) are the results of GAIF and weighted GAIF respectively with  $\epsilon = 1$ .



**Fig. 5:** Variant 2 of weighted GAIF demonstration. (a) is input image  $I$ . (b) is median filtered version of  $I$  with window size = 11. (c) and (e) are  $\bar{\alpha}_p$  image in the case of standard GAIF (11) and variant 2 of the weighted GAIF (14) respectively. Light areas represent more contribution from  $I$  than  $M$  and darker regions represent more contribution from  $M$  than  $I$ . (d) and (f) are the results of GAIF and weighted GAIF respectively with  $\epsilon = 1$ .

\* $c$  was fixed throughout this paper as  $c = 1 \times 10^{-6}$

### 3.3 Filter kernel

GAIF is a local filter, in particular, the resulting image  $J$  is a linear local combination of the input image  $I$  and a smoothed version of  $I$  namely  $M$  as follows:

$$J_p = \alpha_k I_p + (1 - \alpha_k) M_p, \quad \forall p \in \Omega_k \quad (18)$$

which can be written as follows:

$$J_p = \sum_{j \in \Omega_k} W_{pj}(I, M) I_j \quad (19)$$

where the  $W_{pj}(I, M)$  is the filter kernel. The kernel depends on both  $I$  and  $M$ . Generally, we filter  $I$  to produce  $M$ . One such case is to filter  $I$  with a box filter to produce  $M$ , the GAIF filter kernel is similar to the self-guided case of the GIF [4] which has following explicit formula:

$$W_{pj}(I) = \frac{1}{|\Omega_k|^2} \sum_{p \in \Omega_k} \sum_{j \in \Omega_k} \left( 1 + \frac{(I_p - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon} \right) \quad (20)$$

Details can be found in Appendix 7

### 3.4 $O(N)$ Time exact algorithm

A major advantage of the proposed filter over the global energy minimization schemes is that it is an  $O(N)$  complexity exact algorithm.  $O(N)$  means that the filter computational complexity depends linearly only on the number of pixels. In other words, it is independent of the window size, which allows the user to choose any windows size without additional computational cost as is the case with the bilateral filter [1], non-local means [48], SD [47], RTV [39], and the more recent filter by Wang et al [49] just to name few. This property is shared with the guided image filter [4]. Table 1 provides a summary of the computational complexities of some of the well-known techniques in the literature for comparison.

**Table 1** Computational complexities of some of the well-known filters in the literature.  $N$  is the total number of pixels in an image,  $|\Omega_k|$  is number of pixels in the patch,  $w$  is the number of pixels in a window, and  $n$  is the number of iterations.

Filter	Complexity
Bilateral	$O( \Omega_k .N)$
NLM	$O( \Omega_k .w.N)$
SD	$O(n. \Omega_k .N)$ Assuming that the linear system can be solved in $O(N)$
RTV	$O(n. \Omega_k .N)$
GF	$O(N)$

## 4 Parameters Setting and Details Smoothing

### 4.1 Parameter setting

GAIF has two parameters to tune; the radius  $r$  of a patch  $\Omega_k$ , which is an odd integer and can take the values  $\{3, 5, 7, \text{etc}\}$ , the second parameter being the regularization parameter  $\epsilon$ . We have empirically observed that increasing  $r$  results in better edge preservation. On the other hand, increasing  $\epsilon$  was found to result in increasingly smoother images. Those observations can be verified in Figure 6.

The behaviour of GAIF at various values of  $\epsilon$  and  $r$  can be explained by referring to 3.3. For the special case where  $M$  is an average filter, the relationship between the output pixels and the input pixels is encoded in the explicit formula in (20). From (20), we notice that; increasing  $\epsilon$  results in an averaging effect while increasing  $r$ , which increases the number of pixels in  $\Omega_k$ , increases the reliance on the image  $I$ , hence more edges.

### 4.2 The role of $M$

The image  $M$  in (6) is responsible for the smoothing effect hence, it is important to demonstrate the role it plays in the GAIF filtering. The choice of  $f(\cdot)$  in (5) can be considered as a tuning parameter with the highest gains in smoothing performance achieved for non edge-aware filters ( $f(\cdot)$ ). To this end, we compare the effect of using Gaussian and Median filters to construct the image  $M$  on various images. Figures 8,9,10,11,12,13,14,15 demonstrate the role of different  $M$  images on the filtering outcome of the four images in 7.

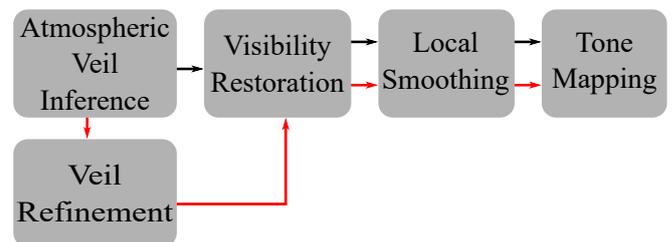
### 4.3 Details smoothing

Figures 16 17 18 are comparisons between the smoothing results. The top part merges two filtered version of the eye image, upper-left is the eye smoothed using the named filter, lower-right is the smoothed with GAIF. Here the GAIF smoothed image is a result of filtering the original image, guided by the smoothed image using the named filter. Filters used in the comparison include the classical Gaussian and median filters, guided filter (GIF) [4], sub-window box filter (SWF) [50], weighted least-squares (WLS) [2], static-dynamic filter (SD) [47], rolling guidance filter (RGF) [51] and relative total variation filter (RTV) [39]. In the close-up views, it is clear that GAIF is better preserving the eyelashes in comparison with other filters, meanwhile smoothing the other parts of the face. This result demonstrates the efficiency of GAIF at improving on the smoothing results of linear and non-linear filters. This improvement comes at a minimal computational cost and no reformulation of these filters is required.

## 5 Applications and Experimental Results

### 5.1 Single image haze removal

Tarel et al [52, 53] proposed a method for single image dehazing which works in the four steps summarised in Figure 19 (black arrows path). To improve on the results of this technique, a veil refinement step is introduced which involves filtering the inferred veil image using an edge-aware filter. We use GAIF to perform the veil refinement step, more specifically, the image  $I$  is the inferred veil image while the image  $M$  is a mean or median filtered version of  $I$ . The authors in [52, 53] have considered both median and bilateral filters for veil refinement. Figures 20 and 21 are comparisons between three state-of-the-art techniques, including the techniques proposed in [52, 53], dark channel prior [24] and our technique. The top parts of the figures are the results of various techniques and the bottom parts are close-up views of two regions of the images. GAIF is resulting in the clearest and sharpest result among the four techniques.



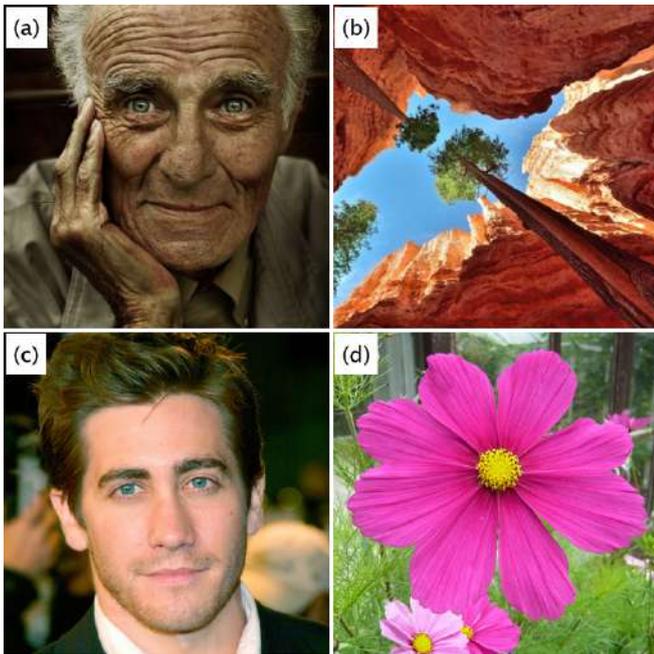
**Fig. 19:** Single image de-hazing steps proposed in [52]. The black arrows refer to the original model and the red path includes the GAIF as the veil refinement step.

### 5.2 Flash/No flash fusion denoising

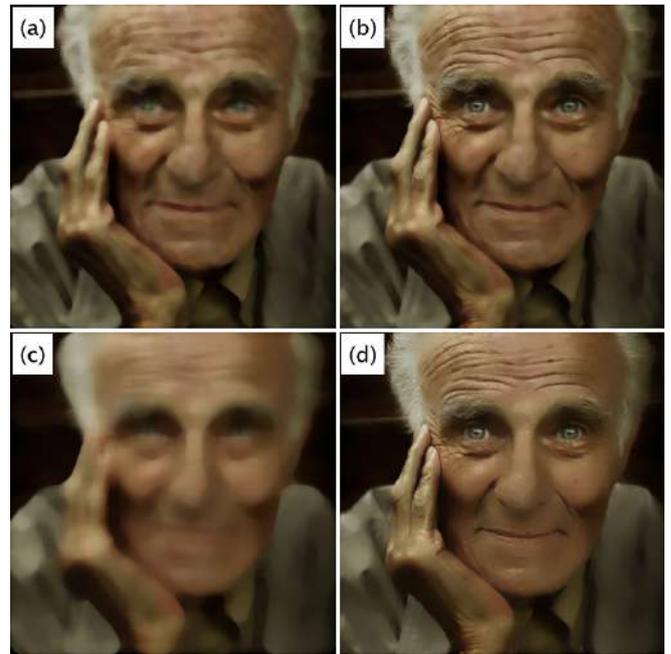
Denoising an image taken without flash by utilizing another version of the same image with flash is a common digital photography problem [31]. In Figure 23, we compare three representative filters namely; guided filter (GIF) [4], joint bilateral filter (JBF) [31] and semi-guided bilateral filter (SGBF) [54]. GAIF is used in this application as a



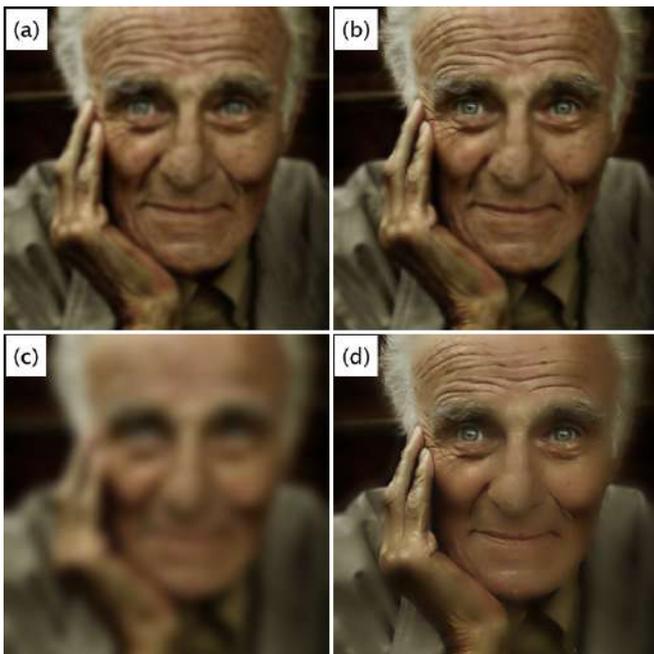
**Fig. 6:** Edge-preserving smoothing using GAIF filter with different kernel sizes  $r$  and values of the regularization parameter  $\epsilon$ . Lower  $\epsilon$  values correspond to sharper outputs. Larger patch radius  $r$  results in better edge preservation.



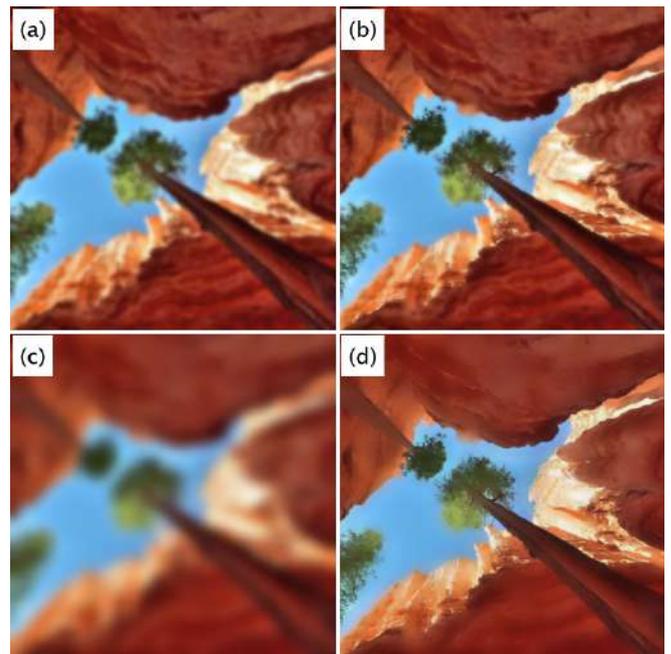
**Fig. 7:** Input images used to demonstrate the role of the smooth image  $M$ .



**Fig. 9:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Median filter with window size 9 and 21 respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (a) while  $M$  being the image (a) and (c) respectively.



**Fig. 8:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Gaussian filter with  $\sigma = 2$  and  $\sigma = 7$  respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (a) while  $M$  being the image (a) and (c) respectively.



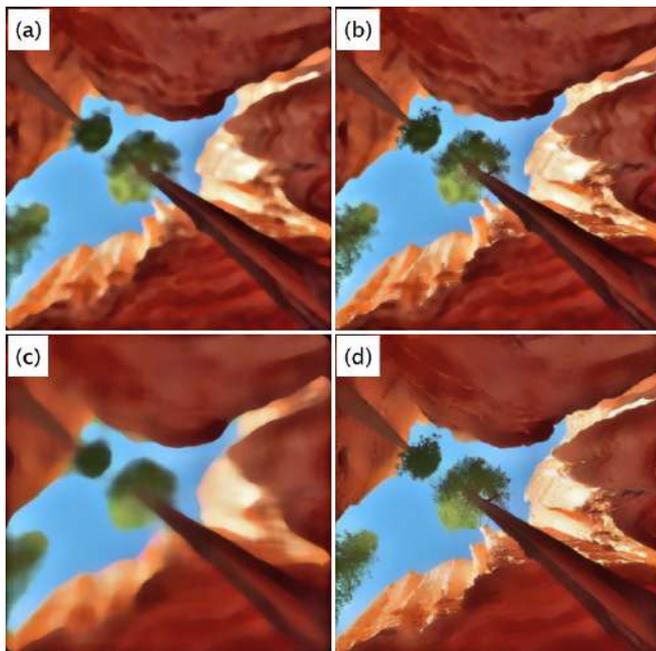
**Fig. 10:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Gaussian filter with  $\sigma = 2$  and  $\sigma = 7$  respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (b) while  $M$  being the image (a) and (c) respectively.

replacement to the bilateral filter in the scheme proposed in [31]. More specifically, GAIF is used to filter both images the flash and no-flash where  $I$  is the image to be filtered and  $M = H * I$  where  $H$  is a linear average filter as shown in Figure 22. In Figure 23, not only did GAIF denoise the no-flash image, but it also filled the dark regions between the jugs with details from the flash image producing a significantly better result than JBF and GIF. However, comparable results to SGBF, but slightly sharper, especially the drawings on the jugs.

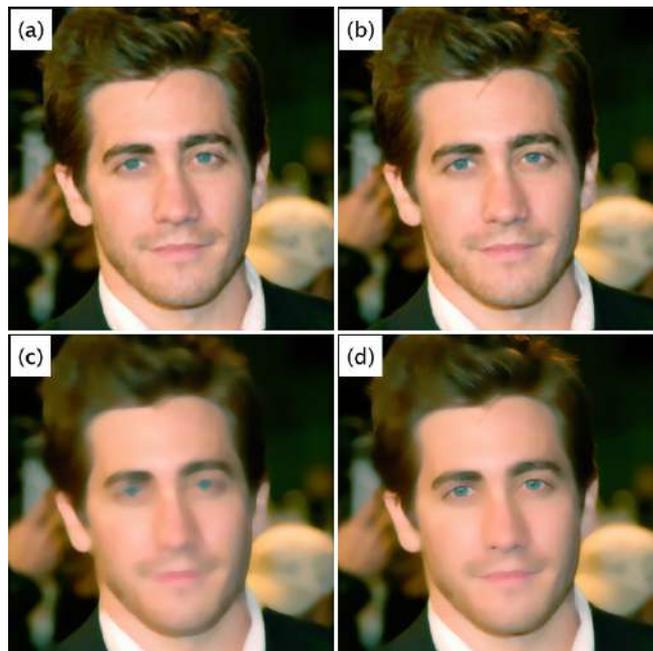
### 5.3 Image detail enhancement

Enhancing the details of an image starts with decomposing the image into base and details layers followed by details amplification. To avoid halo artifacts, edge-aware filters are used to produce the base layer as follows:

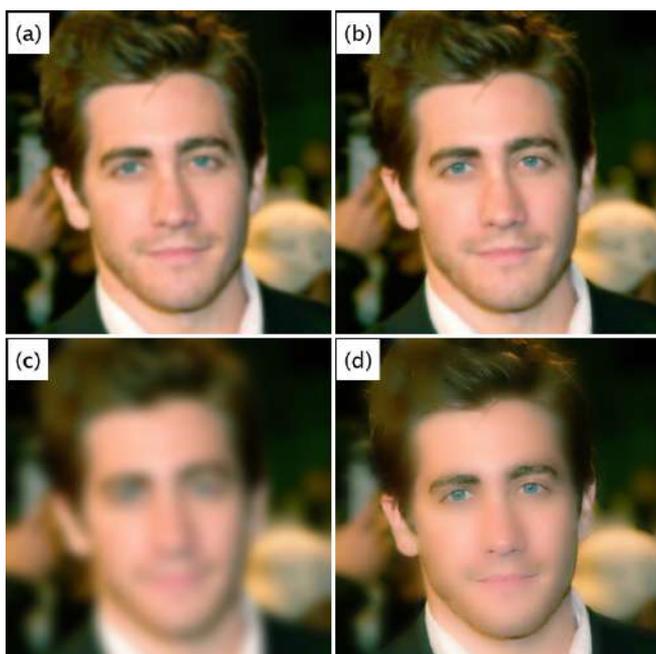
$$J = \text{GAIF}(I, M) + \gamma(I - \text{GAIF}(I, M)) \quad (21)$$



**Fig. 11:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Median filter with window size 9 and 21 respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (b) while  $M$  being the image (a) and (c) respectively.



**Fig. 13:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Median filter with window size 5 and 13 respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (c) while  $M$  being the image (a) and (c) respectively.



**Fig. 12:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Gaussian filter with  $\sigma = 2$  and  $\sigma = 7$  respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (c) while  $M$  being the image (a) and (c) respectively.



**Fig. 14:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Gaussian filter with  $\sigma = 2$  and  $\sigma = 7$  respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (d) while  $M$  being the image (a) and (c) respectively.

where  $\gamma$  is a magnification factor used to amplify the details, and  $M$  is median filtered version of  $I$ . Figures 25 and 26 compare the performance of GAIF to two representative techniques; weighted least-squares (WLS) [2] and semi-guided filter [54]. We observe that the proposed filter preserves the original edges intact, in other words, the filtered image has the edges of the input image as can be seen in Figure 25.

#### 5.4 Edge detection

Another application of edge-aware filters is edge-detection. Images are preprocessed using edge-aware filtering followed by an edge-detection algorithm. To demonstrate the potential of GAIF at improving edge-detection, in Figure 27, we conducted an experiment on a synthetically generated image with known edges then we added noise to it. To detect the edges of the noisy image, we preprocessed it with GAIF followed by edge-detection. We used the "edge" command



**Fig. 15:** The role of the smooth image  $M$ . (a) and (c) are  $M$  images produced using a Median filter with window size 9 and 21 respectively. (b) and (d) are GAIF filtered images with  $I$  being the original image in 7 (d) while  $M$  being the image (a) and (c) respectively.

in MATLAB for edge detection which is an implementation of the Canny edge-detection algorithm.

## 6 Discussion and Conclusion

In this paper, a novel filtering technique is presented with a number of applications in image processing and computer vision. GAIF achieves edge-preservation by interpolating between two patches. As a result, the filter can improve the results of linear and non-linear filters.

GAIF is a computationally efficient edge-preserving filter with a computational complexity of  $O(N)$  where  $N$  is the number of pixels in the image. We have demonstrated the efficiency of GAIF on a number of problems including single image haze-removal, flash/no-flash image fusion, image detail enhancement and edge detection.

Finally, we note that GAIF models the image as an interpolation at the patch level between the input image and a smoothed version of which. This means that; the filter is constructed with two images,  $I$  and  $M$ , produced using the same type of sensor in mind. Hence, it's not straight forward to utilize the filter for tasks that require cross-domain fusion such as matting/feathering [55] and super-resolution. It is our future plan to adapt GAIF to more applications including multi-sensor fusion, image vectorisation, colourisation, non photo-realistic rendering and low-light image enhancement.

## Acknowledgment

We would like to acknowledge the anonymous reviewers' constructive feedback that have improved the overall quality of this work.

## 7 Explicit kernel proof

In the special case of  $M$  is a box filter (linear), GAIF is equivalent to the self-guided GIF. Specifically, the model of the self-guided GIF is:

$$\arg \min_{a_k, b_k} C(a_k, b_k) = \sum_{p \in \Omega_k} (a_k I_p + b_k - I_p)^2 + \epsilon a_k^2 \quad (22)$$

optimizing for  $b_k$  results in the following:

$$b_k = \mu_k - a_k \mu_k = (1 - \alpha_k) \mu_k \quad (23)$$

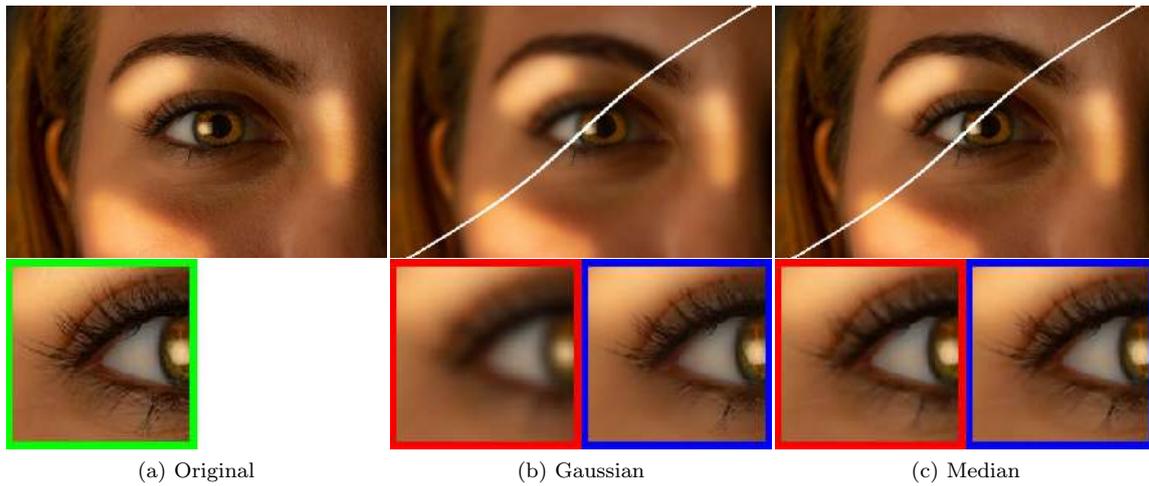
where

$$\mu_k = \frac{1}{|\Omega_k|} \sum_{p \in \Omega_k} I_p$$

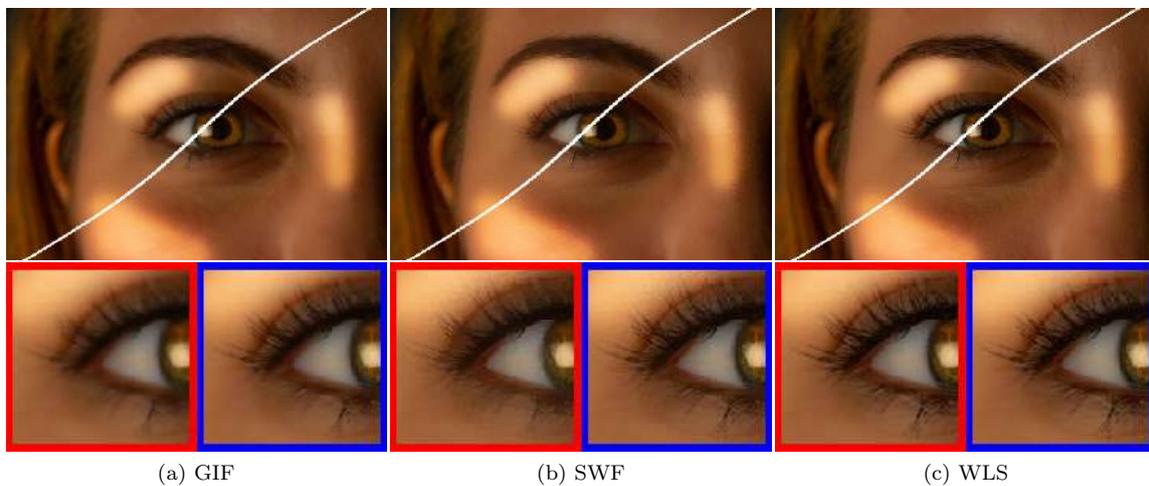
substituting  $b_k$  in the GIF model yields:

$$\arg \min_{a_k, b_k} C(a_k) = \sum_{p \in \Omega_k} (a_k I_p + (1 - \alpha_k) \mu_k - I_p)^2 + \epsilon a_k^2 \quad (24)$$

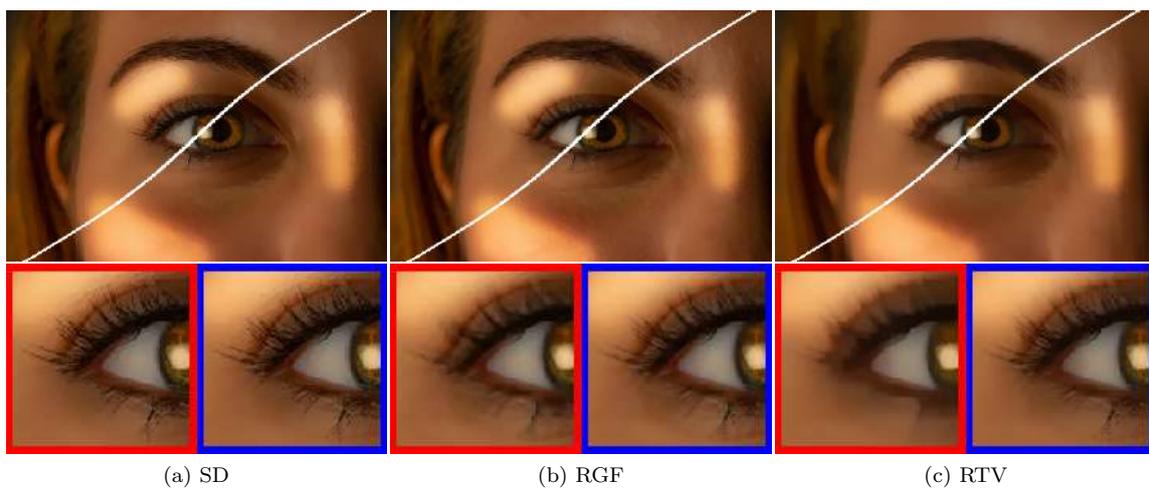
Equation (24) is equivalent to GAIF with  $M = \mu_k$ .



**Fig. 16:** Image smoothing (b) Gaussian  $\sigma = 6$ , (c) median  $5 \times 5$  and GAIF  $\epsilon = 0.04$  for all cases. The top part compares two results, to the left is a result of the indicated filter, to the right is a result of filtering with GAIF. The bottom part is zoomed-in versions of the top part, the red box is for the result of the indicated filter and the blue box is for the GAIF result.



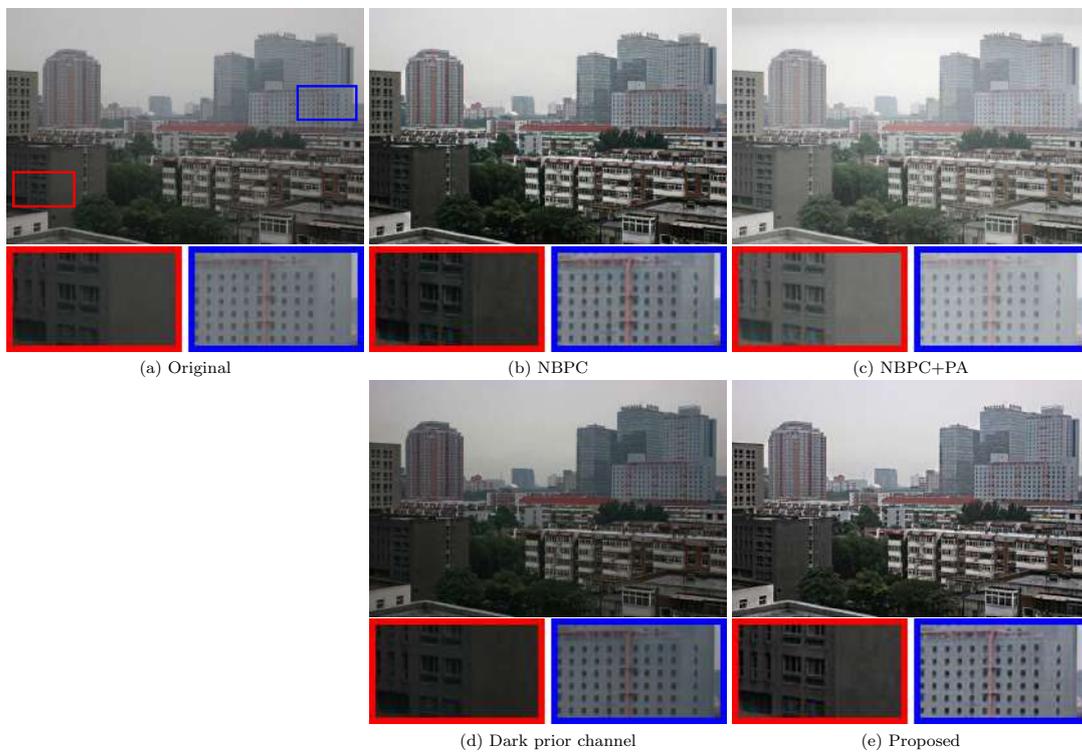
**Fig. 17:** Image smoothing (a) GIF  $r = 5$ ,  $\epsilon = 0.01$ , (b) SWF  $r = 5$ , (c) WLS  $\lambda = 0.05$ ,  $\alpha = 1$  and GAIF  $\epsilon = 0.04$  for all cases. The top part compares two results, to the left is a result of the indicated filter, to the right is a result of filtering with GAIF. The bottom part is zoomed-in versions of the top part, the red box is for the result of the indicated filter and the blue box is for the GAIF result.



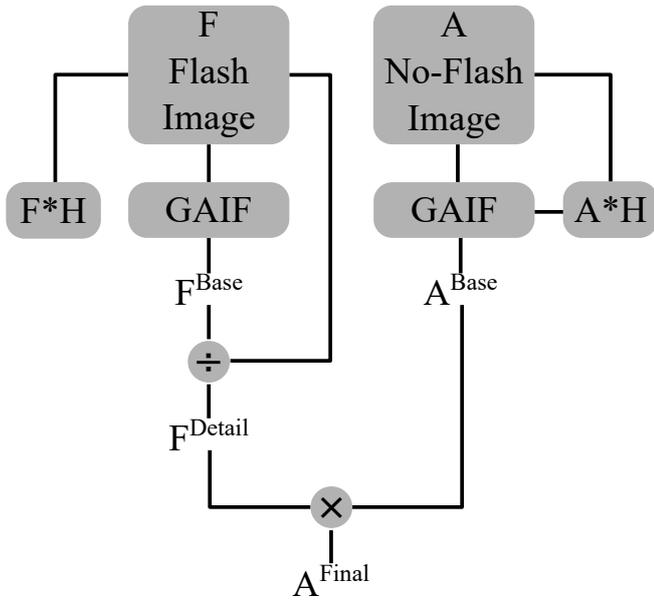
**Fig. 18:** Image smoothing (a) SD  $\lambda = 5$ ,  $\mu = 50$ ,  $\nu = 400$ , iter = 10, (b) RGF  $\sigma_s = 3$ ,  $\sigma_r = 0.01$ , iter = 4, (c) RTV  $\lambda = 0.005$ ,  $\sigma = 3$  and GAIF  $\epsilon = 0.04$  for all cases. The top part compares two results, to the left is a result of the indicated filter, to the right is a result of filtering with GAIF. The bottom part is zoomed-in versions of the top part, the red box is for the result of the indicated filter and the blue box is for the GAIF result.



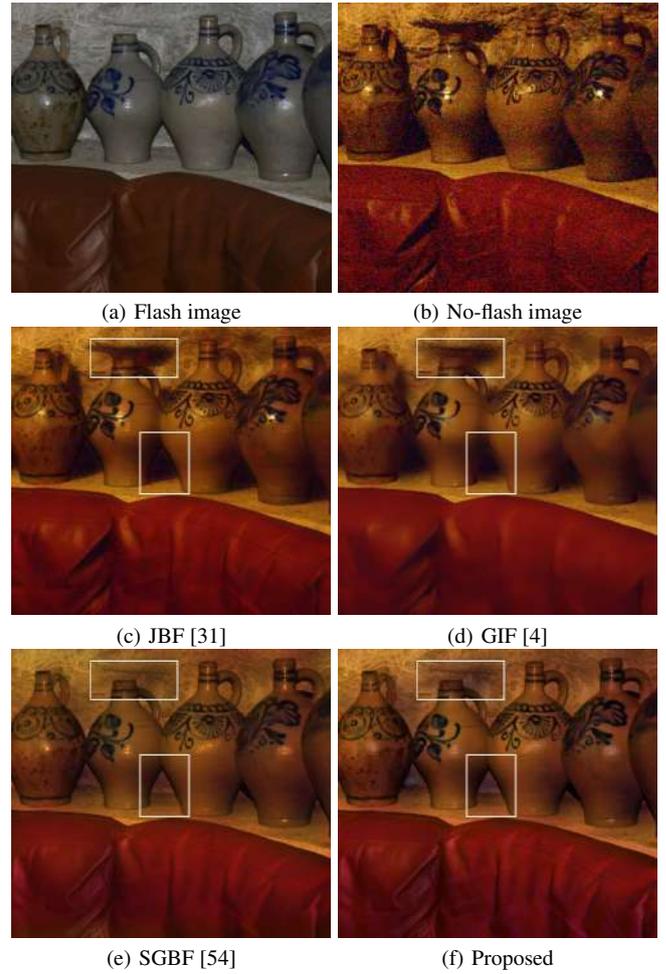
**Fig. 20:** A comparison of image haze removal on road image. GAIF is used to filter the atmospheric veil estimate in the no-black pixel constraint (NBPC) technique [52]. In comparison, dehazing using the original NBPC[52], NBPC+PA [53] and dark prior channel [24] techniques are evaluated.



**Fig. 21:** A comparison of image haze removal on city image. GAIF is used to filter the atmospheric veil estimate in the no-black pixel constraint (NBPC) technique [52]. In comparison, dehazing using the original NBPC[52], NBPC+PA [53] and dark prior channel [24] techniques are evaluated.



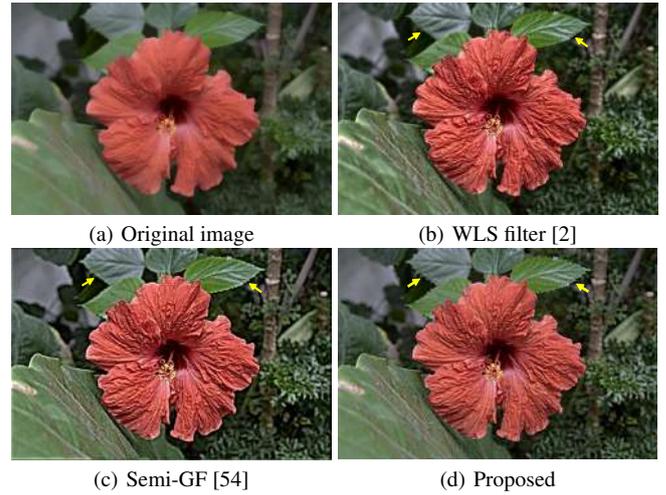
**Fig. 22:** Flash/No flash denoising algorithm.



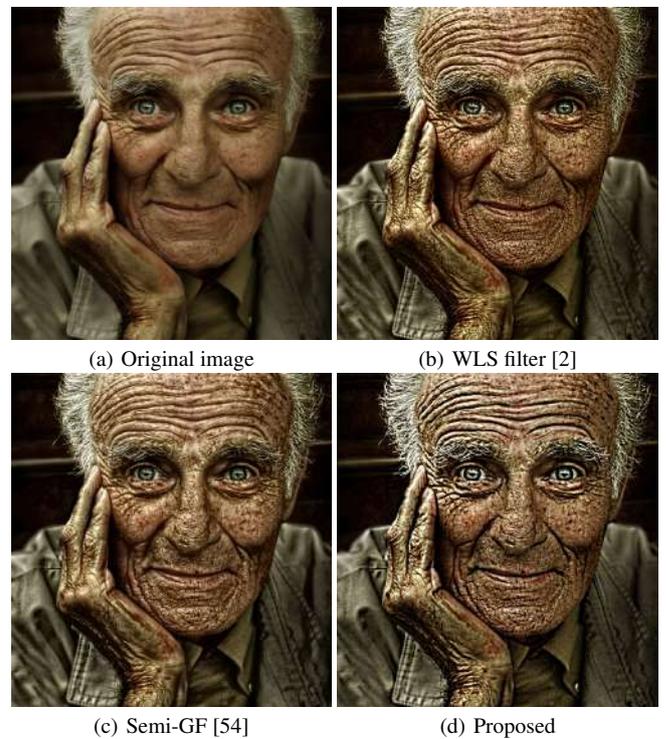
**Fig. 23:** A comparison of image flash/no-flash denoising between GAIF with  $(F_{base} : M = \text{boxfilter}(I, w = 50), \Omega_k = 3, \epsilon = 1, A_{NR} : M = \text{boxfilter}(I, w = 50), \Omega_k = 5, \epsilon = 20)$ , the joint bilateral filter [31], guided image filter [4] with  $(r = 9, \epsilon = 0.0004)$  and Semi-guided filter [54] with  $(F_{base} : \sigma_s = 8.5, \sigma_r = 0.5, N = 5, A_{NR} : \sigma_s = 8.5, \sigma_r = 0.35, N = 5)$ . The proposed filter is superior to the first two and is on par with (e) but at less computational complexity. In (f) GAIF manages to capture all the important details as annotated.



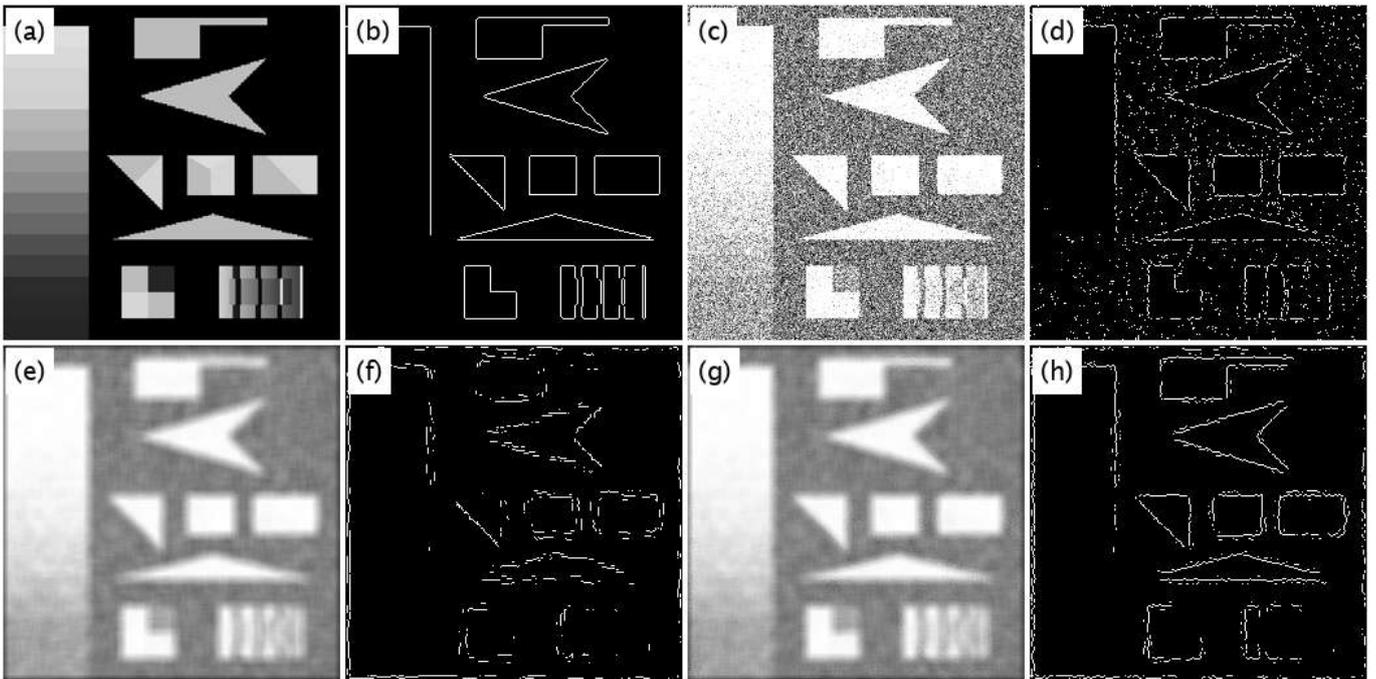
**Fig. 24:** A comparison of image flash/no-flash denoising between GAIF with ( $F_{base} : M = \text{boxfilter}(I, w = 50), \Omega_k = 3, \epsilon = 1, A_{NR} : M = \text{boxfilter}(I, w = 50), \Omega_k = 5, \epsilon = 20$ ), the joint bilateral filter [31], guided image filter [4] with ( $r = 9, \epsilon = 0.0004$ ) and Semi-guided filter [54] with ( $F_{base} : \sigma_s = 8.5, \sigma_r = 0.5, N = 5, A_{NR} : \sigma_s = 8.5, \sigma_r = 0.35, N = 5$ ). The proposed filter is superior to the first two and is on par with (e) but at less computational complexity. In (f) GAIF manages to capture all the important details as annotated.



**Fig. 25:** A comparison of Image details enhancement performance between GAIF ( $M = \text{MEDFILT}(I, w = 13), \Omega_k = 3, \epsilon = 0.1$ ), weighted least-squares filter [2] ( $\lambda = 0.125, \alpha = 1.2$ ) and the semi-guided filter [54] ( $\sigma_s = 3.5, \sigma_r = 0.05, N = 5$ ). As annotated, the proposed filter excels at preserving the true edges of the input image while achieving comparable smoothing performance in other regions.



**Fig. 26:** A comparison of Image details enhancement performance between GAIF ( $M = \text{MEDFILT}(I, w = 13), \Omega_k = 3, \epsilon = 0.00001$ ), weighted least-squares filter [2] ( $\lambda = 0.125, \alpha = 1.12$ ) and the semi-guided filter [54] ( $\sigma_s = 3.5, \sigma_r = 0.05, N = 5$ ).



**Fig. 27:** Edge detection by preprocessing image using GAIF. (a) is input image, (b) is the edge map of the input image, (c) is the input image plus a gaussian noise with  $\mu = 0.5$  and  $\sigma^2 = 0.05$ , (d) is the edge map of the image in (c), (e) is average filtered version of the image in (c) with window size 9, (f) edge map of the image in (e), (g) is GAIF filtered image with  $I$  is the image in (c) and  $M$  is the image in (e), finally (h) is the edge map of the image in (g).

## 8 References

- 1 Tomasi, C., Manduchi, R. 'Bilateral filtering for gray and color images'. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Bombay, India, Jan 1998. pp. 839–846
- 2 Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: 'Edge-preserving decompositions for multi-scale tone and detail manipulation', *ACM Trans Graph*, 2008, **27**, (3), pp. 67:1–67:10.
- 3 Fattal, R.: 'Edge-avoiding wavelets and their applications', *ACM Trans Graph*, 2009, **28**, (3), pp. 22:1–22:10.
- 4 He, K., Sun, J., Tang, X.: 'Guided image filtering', *IEEE Trans Pattern Anal Mach Intell*, 2012, **35**, (6), pp. 1397–1409
- 5 Xu, L., Lu, C., Xu, Y., Jia, J.: 'Image smoothing via  $L_0$  gradient minimization', *ACM Trans Graph*, 2011, **30**, (6), pp. 174:1–174:12.
- 6 Bi, S., Han, X., Yu, Y.: 'An  $L_1$  image transform for edge-preserving smoothing and scene-level intrinsic decomposition', *ACM Trans Graph*, 2015, **34**, (4), pp. 78:1–78:12.
- 7 Karacan, L., Erdem, E., Erdem, A.: 'Structure-preserving image smoothing via region covariances', *ACM Trans Graph*, 2013, **32**, (6), pp. 176:1–176:11.
- 8 Gastal, E.S.L., Oliveira, M.M.: 'Domain transform for edge-aware image and video processing', *ACM Trans Graph*, 2011, **30**, (4), pp. 69:1–69:12.
- 9 Paris, S., Hasinoff, S.W., Kautz, J.: 'Local laplacian filters', *Communications of the Association for Computing Machinery*, 2015, **58**, (3), pp. 81–91
- 10 Zhang, Q., Xu, L., Jia, J. '100+ times faster weighted median filter (wmf)'. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, Jun 2014. pp. 2830–2837
- 11 Min, D., Choi, S., Lu, J., Ham, B., Sohn, K., Do, M.N.: 'Fast global image smoothing based on weighted least squares', *IEEE Trans Image Process*, 2014, **23**, (12), pp. 5638–5653
- 12 Kim, Y., Min, D., Ham, B., Sohn, K.: 'Fast domain decomposition for global image smoothing', *IEEE Trans Image Process*, 2017, **26**, (8), pp. 4079–4091
- 13 Barron, J.T., Poole, B. 'The fast bilateral solver'. In: Computer Vision – ECCV 2016. (Cham: Springer International Publishing, 2016. pp. 617–632
- 14 Ono, S.: ' $l_0$  gradient projection', *IEEE Trans Image Process*, 2017, **26**, (4), pp. 1554–1564
- 15 Yin, H., Gong, Y., Qiu, G.: 'Side window guided filtering', *Signal Process*, 2019, **165**, pp. 315–330.
- 16 Deng, G.: 'Guided wavelet shrinkage for edge-aware smoothing', *IEEE Trans Image Process*, 2017, **26**, (2), pp. 900–914
- 17 Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: 'Deterministic edge-preserving regularization in computed imaging', *IEEE Trans Image Process*, 1997, **6**, (2), pp. 298–311
- 18 Fattal, R., Agrawala, M., Rusinkiewicz, S.: 'Multiscale shape and detail enhancement from multi-light image collections', *ACM Trans Graph*, 2007, **26**, (3).
- 19 Li, Z.G., Zheng, J.H., Rahardja, S.: 'Detail-enhanced exposure fusion', *IEEE Trans Image Process*, 2012, **21**, (11), pp. 4672–4676
- 20 Guan, J., Cham, W. 'Quality estimation based multi-focus image fusion'. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). New Orleans, LA, Mar 2017. pp. 1987–1991
- 21 Al-nasrawi, M., Deng, G., Thai, B.: 'Edge-aware smoothing through adaptive interpolation', *Signal, Image and Video Processing*, 2018, **12**, (2), pp. 347–354.
- 22 Zhou, Z., Wang, B., Ma, J.: 'Scale-aware edge-preserving image filtering via iterative global optimization', *IEEE Trans Multimedia*, 2018, **20**, (6), pp. 1392–1405
- 23 Cai, B., Xing, X., Xu, X. 'Edge/structure preserving smoothing via relativity-of-gaussian'. In: 2017 IEEE International Conference on Image Processing (ICIP). Beijing, Sep 2017. pp. 250–254
- 24 He, K., Sun, J., Tang, X.: 'Single image haze removal using dark channel prior', *IEEE Trans Pattern Anal Mach Intell*, 2011, **33**, (12), pp. 2341–2353
- 25 Badri, H., Yahia, H., Aboutajdine, D.: 'Fast edge-aware processing via first order proximal approximation', *IEEE Trans Vis Comput Graphics*, 2015, **21**, (6), pp. 743–755
- 26 Li, S., Zhang, K., Hao, Q., Duan, P., Kang, X.: 'Hyperspectral anomaly detection with multiscale attribute and edge-preserving filters', *IEEE Geosci Remote Sens Lett*, 2018, **15**, (10), pp. 1605–1609
- 27 Xia, J., Bombrun, L., Adali, T., Berthoumieu, Y., Germain, C.: 'Spectral-spatial classification of hyperspectral images using ica and edge-preserving filter via an ensemble strategy', *IEEE Trans Geosci Remote Sens*, 2016, **54**, (8), pp. 4971–4982
- 28 Milanfar, P.: 'A tour of modern image filtering: New insights and methods, both practical and theoretical', *IEEE Signal Process Mag*, 2012, **30**, (1), pp. 106–128
- 29 Talebi, H., Milanfar, P.: 'Global image denoising', *IEEE Trans Image Process*, 2013, **23**, (2), pp. 755–768
- 30 Ham, B., Cho, M., Ponce, J.: 'Robust guided image filtering using nonconvex potentials', *IEEE Trans Pattern Anal Mach Intell*, 2018, **40**, (1), pp. 192–207
- 31 Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: 'Digital photography with flash and no-flash image pairs', *ACM Trans Graph*, 2004, **23**, (3), pp. 664–672.
- 32 Li, Z., Zheng, J., Zhu, Z., Yao, W., Wu, S.: 'Weighted guided image filtering', *IEEE Trans Image Process*, 2015, **24**, (1), pp. 120–129
- 33 Lu, Z., Long, B., Li, K., Lu, F.: 'Effective guided image filtering for contrast enhancement', *IEEE Signal Process Lett*, 2018, **25**, (10), pp. 1585–1589
- 34 Deng, G.: 'Edge-aware BMA filters', *IEEE Trans Image Process*, 2016, **25**, (1), pp. 439–454
- 35 Eisemann, E., Durand, F.: 'Flash photography enhancement via intrinsic relighting', *ACM Trans Graph*, 2004, **23**, (3), pp. 673–678.
- 36 Qiu, T., Wang, A., Yu, N., Song, A.: 'Llsure: Local linear sure-based edge-preserving image filtering', *IEEE Trans Image Process*, 2013, **22**, (1), pp. 80–90
- 37 Lu, K., You, S., Barnes, N.: 'Double-guided filtering: Image smoothing with structure and texture guidance'. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA). Sydney, NSW, Nov 2017. pp. 1–8
- 38 Li, Z., Zheng, J.: 'Single image de-hazing using globally guided image filtering', *IEEE Trans Image Process*, 2018, **27**, (1), pp. 442–450
- 39 Xu, L., Yan, Q., Xia, Y., Jia, J.: 'Structure extraction from texture via relative total variation', *ACM Trans Graph*, 2012, **31**, (6), pp. 139:1–139:10.
- 40 Liu, W., Zhang, P., Huang, X., Yang, J., Shen, C., Reid, L.: 'A generalized framework for edge-preserving and structure-preserving image smoothing'. (arXiv, 2019)
- 41 Krishnan, D., Fattal, R., Szeliski, R.: 'Efficient preconditioning of laplacian matrices for computer graphics', *ACM Trans Graph*, 2013, **32**, (4), pp. 142:1–142:15.
- 42 Gonzalez, R.C.: 'Digital image processing'. 4th ed. (Pearson, 2018)
- 43 Durand, F., Dorsey, J.: 'Fast bilateral filtering for the display of high-dynamic-range images', *ACM Trans Graph*, 2002, **21**, (3), pp. 257–266.
- 44 Han, J., Kim, J., Cheon, S., Kim, J., Ko, S.: 'A novel image interpolation method using the bilateral filter', *IEEE Trans Consum Electron*, 2010, **56**, (1), pp. 175–181
- 45 Yang, Q., Wang, S., Ahuja, N. 'Real-time specular highlight removal using bilateral filtering'. In: Daniilidis, K., Maragos, P., Paragios, N., editors. Computer Vision – ECCV 2010. (Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. pp. 87–100
- 46 Rudin, L.I., Osher, S., Fatemi, E.: 'Nonlinear total variation based noise removal algorithms', *Physica D*, 1992, **60**, (1), pp. 259–268.
- 47 Ham, B., Cho, M., Ponce, J. 'Robust image filtering using joint static and dynamic guidance'. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, Jun 2015. pp. 4823–4831
- 48 Buades, A., Coll, B., Morel, J.: 'A non-local algorithm for image denoising'. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2. San Diego, CA, USA, Jun 2005. pp. 60–65 vol. 2
- 49 Wang, W., Li, F., Ng, M.K.: 'Structural similarity-based nonlocal variational models for image restoration', *IEEE Trans Image Process*, 2019, **28**, (9), pp. 4260–4272
- 50 Gong, Y., Liu, B., Hou, X., Qiu, G. 'Sub-window box filter'. In: 2018 IEEE Visual Communications and Image Processing (VCIP). Taichung, Taiwan, Dec 2018. pp. 1–4
- 51 Zhang, Q., Shen, X., Xu, L., Jia, J. 'Rolling guidance filter'. In: Computer Vision – ECCV 2014. (Cham: Springer International Publishing, 2014), pp. 815–830
- 52 Tarel, J.P., Hautiere, N. 'Fast visibility restoration from a single colour or gray level image'. In: 2009 IEEE 12th International Conference on Computer Vision. Kyoto, Sep 2009. pp. 2201–2208
- 53 Tarel, J.P., Hautiere, N., Cord, A., Gruyer, D., Halmaoui, H. 'Improved visibility of road scene images under heterogeneous fog'. In: 2010 IEEE Intelligent Vehicles Symposium. San Diego, CA, Jun 2010. pp. 478–485
- 54 Thai, B., Al-nasrawi, M., Deng, G., Su, Z.: 'Semi-guided bilateral filter', *IET Image Process*, 2017, **11**, (7), pp. 512–521
- 55 He, K., Sun, J., Tang, X. 'Fast matting using large kernel matting laplacian matrices'. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, CA, Jun 2010. pp. 2165–2172