

# A guided edge-aware smoothing-sharpening filter based on patch interpolation model and generalized Gamma distribution

Guang Deng, Fernando Galetto\*, Mukhalad Al-nasrawi, and Waseem Waheed

Smoothing and sharpening are two fundamental image processing operations. The latter is usually related to the former through the unsharp masking algorithm. In this paper, we develop a new type of filter which performs smoothing or sharpening via a tuning parameter. The development of the new filter is based on (1) a new Laplacian-based filter formulation which unifies the smoothing and sharpening operations, (2) a patch interpolation model similar to that used in the guided filter which provides edge-awareness capability, and (3) the generalized Gamma distribution which is used as the prior for parameter estimation. We have conducted detailed studies on the properties of two versions of the proposed filter (self-guidance and external guidance). We have also conducted experiments to demonstrate applications of the proposed filter. In the self-guidance case, we have developed adaptive smoothing and sharpening algorithms based on texture, depth and blurriness information extracted from an image. Applications include enhancing human face images, producing shallow depth of field effects, focus-based image enhancement, and seam carving. In the external guidance case, we have developed new algorithms for combining flash and no-flash images and for enhancing multi-spectral images using a panchromatic image.

**Index Terms**—Edge-aware filter, image smoothing, image sharpening, maximum a posteriori estimate.

## I. INTRODUCTION

**S**MOOTHING and sharpening are two fundamental operations in image processing. Traditionally, smoothing is used to reduce noise, while sharpening is used to enhance details [1]. In recent years, smoothing has found increasingly more applications in graphics, computational photography, and computer vision. Edge-aware smoothing, which preserves sharp edges of objects, has been actively studied. Well known edge-aware filters include [2]–[7]. Among them, the guided filter [3] and its weighted versions [7]–[9] have the advantage of a low computational complexity of  $\mathcal{O}(N)$  in addition to their good performance. Applications of edge-aware filters include detail enhancement, flash no-flash image denoising, upsampling of depth map, image abstraction, image dehazing, tone mapping and contrast enhancement to name a few. These applications have been studied extensively in the literature.

On the other hand, sharpening is usually achieved through the unsharp masking algorithm [1]. Let  $I$  and  $S$  be the observed image to be processed and the sharpened image, respectively. The sharpened image is produced by  $S = I + \gamma Z$  where  $\gamma$  is called the sharpening gain and  $Z$  is the output of a high-pass filter. The use of a linear filter to produce  $Z$  presents two main disadvantages in some applications: high sensitivity to noise and halo artifacts due to overshoot in high contrast regions. Non-linear filters were proposed in [10]–[12] to reduce the effect of noise but not solving the halo artifacts. Edge-aware filters are the main tools to combat the halo effect. In addition, there are many studies on using adaptive gain to perform content adaptive sharpening. For example, in [13], a pixel adaptive gain  $\gamma$  based on the dynamics of the image is proposed to sharpen areas of mid-range contrast, to avoid overshooting in high contrast regions, and to produce minimal sharpening at smooth regions. Attempts have also

been made to formulate the adaptive sharpening problem as an optimization problem [14]. A similar approach was taken by [15] which used the local blurriness to vary the sharpening gain  $\gamma$ . The main goal is to avoid sharpening the very smooth background which is intentionally produced by the photographer to achieve the effect of shallow depth of field.

Although smoothing and sharpening are related through  $Z = I - J$ , where  $J$  is a smoothed version of  $I$ , and edge-aware filters are used to produce  $J$  to minimize the halo effect, these two operations are usually used in different applications. The main motivation of this work is to develop a unified framework such that smoothing and sharpening can be integrated in one filter whose function can be controlled by varying a parameter. Our aim is to develop a filter with a tuning parameter such that when it is set smaller/greater than 1 the filter is smoothing/sharpening. The filter must also have the edge-awareness property such that it does not blur edges when used in smoothing mode and does not create halos when used in sharpening mode. A distinctive advantage of such formulation is that it allows the user to perform selective smoothing and sharpening in different areas of an image to produce results such as smoothing the background while sharpening the main object. In addition, the unification of these two operations in one filter allows the user to have a better control in information fusion applications such as Pan-sharpening [16] and flash-no-flash imaging [3].

The main contributions of this work and organization of this paper are summarized in the following.

- A systematic formulation of a new type of filter (section II-A) based on the Laplacian operator, which unifies smoothing and sharpening operations in one filter. The function and level of smoothing or sharpening are controlled by varying the value of a parameter.
- The development of an edge-aware smoothing-sharpening filter (section III) based on a patch interpolation model similar to that of the guided filter. Parameters of the filter are determined by minimizing the negative posterior probability. The generalized

\* Corresponding author. Deng, Galetto and Waheed are with Department of Engineering, La Trobe University, Bundoora, Victoria 3086, Australia. Phone: +61 3 9479 1410. Fax: +61 3 9471 0524. Email: {d.deng, f.galetto, w.waheed}@latrobe.edu.au. Al-nasrawi is with Al-Furat Al-Awsat Technical University, Technical College of Al-Mussaib, Iraq. Email: com.muk@atu.edu.iq

Gamma distribution is used as the prior. The filter includes the original guided filter as a special case. Self-guidance and external-guidance versions of the filter have been developed and their properties are analyzed. Both versions of the filter are of the same computational complexity as that of the guided filter.

- We have demonstrated the performance of the proposed filter in a number of applications in section IV. Using the self-guided filter, we have developed adaptive smoothing-sharpening algorithms by extracting information of texture, depth, and blurriness to adjust the filter parameter to achieve content-aware processing. Applications include enhancement of images of human face, creating the effect of shallow depth of field, smoothing/sharpening guided by blurriness, and pre-processing an image to achieve better seam carving results. Using the filter in external guidance, we have applied the filter to combine images taken under flash and no-flash conditions, producing much better results than those produced by the guided filter. We have applied the filter to solve the Pan-sharpening [17] problem which combines information from multi-spectral images with a panchromatic image. Both subjective and objective comparison are discussed to validate the applications of the proposed filter.

## II. THE SMOOTHING-SHARPENING FILTER, THE SELF-GUIDED FILTER, AND MAIN IDEAS OF THIS WORK

In this section, we first present a filter called the smoothing-sharpening filter in which smoothing and sharpening is configured through the setting of a parameter. Next, we revisit the basic idea of the guided filter. We then discuss the main idea of the proposed edge-aware smoothing-sharpening filter.

### A. The smoothing-sharpening filter

We develop a unified framework for combining smoothing and sharpening into one filter. We first define the filter as

$$J(n) = I(n) + (1 - \alpha)\Delta I(n) \quad (1)$$

where  $I(n)$  and  $J(n)$  are pixels of the input and output images at location  $n$ ,  $\alpha$  is a parameter, and  $\Delta$  is the discrete Laplace operator (which is referred to as the Laplacian operator in rest of this paper) defined as

$$\Delta I(n) = \mu(n) - I(n) \quad (2)$$

where  $\mu(n) = \frac{1}{N} \sum_{m \in \Omega_n} I(m)$  is the mean of the image calculated over a patch centered at location  $n$ . The symbol  $\Omega_n$  represents the set of pixel indices of the patch and  $N = |\Omega_n|$  is the number of pixels. The particular parameterization of this filter by  $1 - \alpha$  will become clear in the following discussions.

To demonstrate the characteristics of this filter, we consider a simple example of a 1-D filter of which the mean is calculated by an average filter  $\mu(n) = \frac{1}{3} \sum_{m=-1}^1 I(n - m)$ . The impulse response of the filter stated in (1) is then given by  $h(n) = \{(1 - \alpha), (1 + 2\alpha), (1 - \alpha)\}/3$ . The frequency response is calculated as follows

$$H(\omega) = \sum_{n=-1}^1 h(n)e^{-j\omega n} = 1 - \frac{4(1 - \alpha)}{3} \sin^2 \frac{\omega}{2} \quad (3)$$

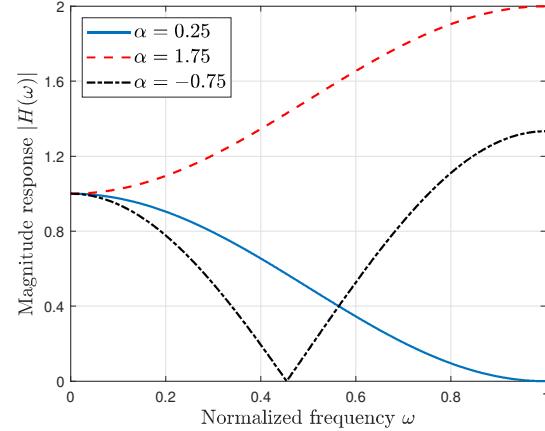


Fig. 1: Magnitude response of the smoothing-sharpening filter. When  $0 < \alpha < 1$ , it is a low-pass filter. When  $\alpha > 1$ , it is a high-frequency emphasis filter. When  $\alpha < 0$ , the filter's function is not well defined.

In Fig. 1, we plot the magnitude response ( $|H(\omega)|$ ) of this filter for three settings of  $\lambda$ . We can see that (a) when  $0 < \alpha < 1$ , it is a low-pass filter, (b) when  $\alpha > 1$ , it is a high-frequency emphasis filter, and (c) when  $\alpha < 0$ , the filter's function is not well defined and is not considered in this paper.

Substitution of (2) into (1), we have

$$J(n) = \alpha I(n) + (1 - \alpha)\mu(n) \quad (4)$$

which is a weighted average between  $I(n)$  and  $\mu(n)$  when  $0 \leq \alpha < 1$ . It is a low-pass filter. On the other hand, when  $\alpha > 1$  we can then re-write (4) as

$$J(n) = \mu(n) + \alpha(I(n) - \mu(n)) \quad (5)$$

Equation (5) is the unsharp masking operation which is a sharpening filter. The equivalence of (5) and (1) shows that the filter stated in (1) can be configured as either a smoothing filter  $0 < \alpha < 1$  or a sharpening filter  $\alpha > 1$ . We can further re-write the filter stated in (4) in the following equivalent form

$$J(n) = \mu(n) - \alpha\Delta I(n) \quad (6)$$

which is the operation of a local mean minus a scaled local Laplacian of the signal.

To use this interpretation in the development of the edge-aware smoothing-sharpening filter, we need to generalize the concept of Laplacian in the following sense. In its original form, the Laplacian at a pixel location  $n$  is defined by (2) where the patch is centered at location  $n$ . The Laplacian can also be written as

$$\Delta I(n) = \frac{1}{N} \sum_{m \in \Omega_n} (I(m) - I(n)) \quad (7)$$

which is the average of the difference between the center pixel  $I(n)$  and each pixel  $I(m)$  in the patch. Using this interpretation, we make the following generalization. For any pixel  $I(q)$  at location  $q \in \Omega_n$ , the Laplacian is defined as

$$\Delta I(q) = \frac{1}{N} \sum_{m \in \Omega_n} (I(m) - I(q)) \quad (8)$$

This definition of the Laplacian thus generalizes the concept from the original one which only applies to the center pixel of the patch to the one which applies to all pixels in the patch. Although the original meaning of the Laplacian is lost in the generalization, we will use the same name in this paper to simplify the terminology. Using this generalization, the filter can be defined for all pixels in the patch as follows

$$J(q) = \mu(n) - \alpha \Delta I(q) \quad (9)$$

$$= \alpha I(q) + (1 - \alpha) \mu(n) \quad (10)$$

The difference between the two filters defined in (6) and (9) is that the former is defined for the center pixel of the patch while the latter is defined for all pixels in the patch. We will call the new filter (stated in (10)) a patch interpolation model, because the output is a weighted average of the input and the patch mean.

When the parameter  $\alpha$  is fixed, the filter does not have an edge-awareness capability. This is a major problem of the filter. We will show in the next section that the idea of the guided filter provides a solution to this problem by adaptively setting the parameter  $\alpha$ .

### B. The self-guided filter

We revisit the basic idea of the guided filter [3]. A square patch of radius  $r$  has  $N = (2r + 1)^2$  pixels. Let  $\Omega_k$  represent the  $k$ th patch in which a pixel at location  $q$  is denoted  $I_k(q)$  where  $q \in \Omega_k$  and the subscript  $k$  indicates the patch. A linear model is imposed on each pixel of the patch such that

$$J_k(q) = a_k I_k(q) + b_k, \quad (11)$$

where  $J_k(q)$  is the desired output. The two patch dependent parameters  $a_k$  and  $b_k$  are determined by solving a regularized least squares problem with the following cost function

$$c = \frac{1}{2N} \sum_{q \in \Omega_k} [J_k(q) - I_k(q)]^2 + \frac{\epsilon}{2} a_k^2, \quad (12)$$

where the second term is the regularization and  $\epsilon$  is a user defined parameter. Solving  $\partial c / \partial b_k = 0$ , we have  $b_k = (1 - a_k) \mu_k$  where  $\mu_k = \frac{1}{N} \sum_{q \in \Omega_k} I_k(q)$  is the patch mean. By Substitution of this result into (11), we have another patch model

$$J_k(q) = a_k I_k(q) + (1 - a_k) \mu_k \quad (13)$$

which is an interpolation between the pixels in the patch and the patch mean.

A new regularized least squares problem is obtained by substitution of (13) into (12) which results in the following cost function

$$d = \frac{\sigma_k^2}{2} (a_k - 1)^2 + \frac{\epsilon}{2} a_k^2 \quad (14)$$

where  $\sigma_k^2 = \frac{1}{N} \sum_{q \in \Omega_k} [I_k(q) - \mu_k]^2$  is the patch variance. Solving  $\partial d / \partial a_k = 0$ , we have

$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \epsilon} \quad (15)$$

Let  $I(p)$  represent the pixel to be processed at location  $p \in \Omega_k$ . It can be shown that the pixel  $I(p)$  belongs to  $N$

overlapping patches [3]. Since each patch model produces one output

$$J_k(p) = a_k I(p) + (1 - a_k) \mu_k, \quad (16)$$

there are  $N$  modelling results  $\{J_k(p)\}_{k=1:N}$ . To aggregate these results, a weighted average [7], [8] is performed:

$$J(p) = \sum_{k=1}^N w_k J_k(p) = I(p) \sum_{k=1}^N w_k a_k + \sum_{k=1}^N w_k (1 - a_k) \mu_k \quad (17)$$

where  $\sum_{k=1}^N w_k = 1$ . The original guided filter [3] uses the fixed weight  $w_k = 1/N$ .

### C. Main idea of the proposed filter

The key to our development is the patch interpolation model which is stated in (13) and the smooth-sharpening filter stated in (10). Comparing the two, we can see that they are in a similar form. There are two key differences.

- In its general form (1), the smoothing-sharpening filter can be configured as either smoothing or sharpening. It is not obvious how to set the parameter  $\alpha$  such that the filter has the edge-aware capability.
- On the other hand, in the guided filter case, the filter parameter  $a_k$  is specifically determined for edge-awareness. Referring to (15), for the case  $\epsilon \ll \sigma_k^2$  which indicates strong texture/edge inside the patch, the algorithm assigns  $a_k \rightarrow 1$  such that  $J_k(q) \rightarrow I(q)$ . For the case  $\epsilon > \sigma_k^2$  which indicates a smooth patch, the algorithm assigns  $a_k \rightarrow 0$  such that  $J_k(q) \rightarrow \mu_k$ . As a result, edge-aware smoothing is achieved. However, it is always a smoothing filter because  $a_k = \frac{\sigma_k^2}{\sigma_k^2 + \epsilon} < 1$ .

Based on the above observations, to develop an edge-aware smoothing-sharpening filter we should find a generalization of the guided filter such that it is possible to set  $a_k \geq 1$ . We show in the next section that the development is based on two key ideas:

- 1) using the patch interpolation model stated in equation (13), and
- 2) using the principle of maximum a posteriori to determine the parameter  $a_k$  to overcome the limitation of  $a_k < 1$  in the original guided filter.

## III. THE EDGE-AWARE SMOOTHING-SHARPENING FILTER

We first develop the self-guided edge-aware smoothing-sharpening filter in section III-A. We then develop the external-guided version of the filter in section III-B. Implementation and computational complexity of the proposed filter are discussed in section III-C.

### A. The self-guided form and its properties

#### 1) The self-guided smoothing-sharpening filter

We define the linear Gaussian observation model for the patch data as follows

$$I_k(q) = J_k(q) + r(q) \quad (18)$$

where  $r(q)$  is a realization of an i.i.d. zero mean Gaussian random variable with variance  $\tau^2 N$  ( $\tau > 0$ ). Here we use the

patch size  $N = |\Omega_k|$  to parameterize the noise variance such that the result does not depend on  $N$ .

To determine  $\alpha_k$ , we treat it as a random variable and use the principle of maximum a posteriori. More specifically, using Bayes rule we can write the negative log-posterior as a cost function  $D(\alpha_k)$  by ignoring constants as follows

$$\begin{aligned} D(\alpha_k) &= -\log p(\alpha_k | \{I_k(q)\}) \\ &= -\log p(\{I_k(q)\} | \alpha_k) - \log p(\alpha_k) \end{aligned} \quad (19)$$

where based on the observation model stated in (18), the negative log-likelihood for the patch is given by

$$\begin{aligned} -\log p(\{I_k(q)\} | \alpha_k) &= \frac{1}{2\tau^2 N} \sum_{q \in \Omega_k} (I_k(q) - J_k(q))^2 \\ &= \frac{\sigma_k^2}{2\tau^2} (\alpha_k - 1)^2 \end{aligned} \quad (20)$$

Compared with the first term of the cost function of the guided filter stated in (12), the above negative log-likelihood has an extra parameter  $\tau$ . We set  $\tau = 1$  in this work which allows us to include the guided filter as a special case in the proposed filter.

Substitution of (20) into (19) the cost function can be written as

$$D(\alpha_k) = \frac{\sigma_k^2}{2} (\alpha_k - 1)^2 - \log p(\alpha_k). \quad (21)$$

Compared with the cost function of the guided filter, the above cost function is different in two aspects: (a) the parameter  $b_k$  is implicitly defined in the patch interpolation model, and (b) the regularization term  $\frac{\epsilon}{2}\alpha_k^2$  (which is the negative of logarithm of zero mean Gaussian) is replaced by the negative log-prior which permits us to develop different filters.

In this work, we consider the generalized Gamma distribution as the prior:

$$p(\alpha_k) \propto \alpha_k^\eta e^{-\frac{1}{2}(\alpha_k/\theta)^h} \quad (22)$$

where  $\theta > 0$  is a scale parameter. We set  $h = 2$  and  $\eta \geq 0$  to control the shape of the distribution. Substituting (22) into (21) we have the cost function in which constant terms are omitted

$$D(\alpha_k) = \frac{\sigma_k^2}{2} (\alpha_k - 1)^2 + \frac{1}{2\theta^2} \alpha_k^2 - \eta \log \alpha_k \quad (23)$$

We can easily see the motivation and justification of such settings. When  $\eta = 0$ , the cost function is the same as that of the guided filter with the setting  $\epsilon = 1/\theta^2$ . When  $\eta > 0$ , the cost function has an extra term  $-\eta \log \alpha_k$  compared with the cost function of the guided filter. We will show that this extra term permits the filter to be configured as either a smoothing filter ( $0 < \alpha_k \leq 1$ ) or a sharpening filter ( $\alpha_k > 1$ ).

Another justification is mathematical simplicity. The cost function under this parameter setting is convex leading to a unique minimum. Indeed, the generalized Gamma distribution allows us to explore other settings of parameters such as  $h \neq 2$ . However, for such a setting, the cost function may not be convex and is difficult to optimize. Therefore, we do not pursue study in this direction.

We now determine the filter parameter  $\alpha_k$  by minimizing the cost function  $D(\alpha_k)$  which is equivalent to maximising

the posterior. Solving  $\partial D / \partial \alpha_k = 0$ , we obtain the optimal value

$$\alpha_k = \frac{1}{2} \left\{ \frac{\sigma_k^2}{\sigma_k^2 + \frac{1}{\theta^2}} + \sqrt{\left( \frac{\sigma_k^2}{\sigma_k^2 + \frac{1}{\theta^2}} \right)^2 + \frac{4\eta}{\sigma_k^2 + \frac{1}{\theta^2}}} \right\} \quad (24)$$

## 2) Properties

To reveal how the proposed filter generalizes the original guided filter, we set  $\epsilon = 1/\theta^2$  such that the guided filter's parameter stated in (15) is given by

$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \epsilon} = \frac{\sigma_k^2}{\sigma_k^2 + 1/\theta^2} \quad (25)$$

Substitution (25) into (24), we have

$$\alpha_k = \frac{1}{2} \left\{ a_k + \sqrt{a_k^2 + \frac{4\eta}{\sigma_k^2 + \epsilon}} \right\} \quad (26)$$

We can clearly see that the parameter of the proposed filter can be expressed as a function of the parameter of the original guided filter. Their relationship is discussed in the following. Based on the interpolation model stated in equation (13), we can prove that the proposed filter can be configured by setting  $\eta$  relative to  $\epsilon$  as follows.

- When  $\eta < \epsilon$ , we can show that  $\alpha_k < 1$  which leads to a smoothing filter. In an extreme case when  $\eta = 0$ , the proposed filter is reduced to the original guided filter. It is also interesting to note that for this setting  $\alpha_k > a_k$  which means the proposed filter always performs a lower degree of smoothing than the original guided filter.
- When  $\eta = \epsilon$ , we can show that  $\alpha_k = 1$  which leads to no filtering.
- When  $\eta > \epsilon$ , we can show that  $\alpha_k > 1$  which leads to a sharpening filtering. The sharpening gain is defined as  $\gamma_k = \alpha_k - 1$  which is an increasing function of  $\eta$ .

In light of the above discussion and to simplify parameter settings, we introduce another parameter  $\kappa$  to replace  $\eta$  by defining  $\eta = \kappa/\theta^2 = \kappa\epsilon$ . The parameter of the proposed filter can be re-written as

$$\alpha_k = \frac{1}{2} \left\{ a_k + \sqrt{a_k^2 + 4\kappa(1 - a_k)} \right\} \quad (27)$$

The advantage of this new parameterization is that the filter can be configured as smoothing by setting  $0 \leq \kappa < 1$  and as sharpening by setting  $\kappa > 1$ .

For the smoothing case, we plot  $\alpha_k$  for different settings of  $\kappa$  including the case with  $\kappa = 0$  which is the guided filter. Results are shown in Fig. 2. We can see that as  $\kappa$  is increased,  $\alpha_k$  is less adapted to  $\sigma_k^2$  and is closer to the constant 1 (no filtering) for  $\sigma_k^2 > T$  where  $T$  is a signal dependent threshold.

Next, we study the sharpening gain  $\gamma_k$  as a function of  $\kappa$  ( $\kappa > 1$ ) and the patch variance  $\sigma_k^2$ . The interpolation model can be re-written in a sharpening filter form

$$J_k(q) = I_k(q) + \gamma_k(I_k(q) - \mu_k) \quad (28)$$

Fig. 3 shows the sharpening gain as a function of the patch variance  $\sigma_k^2$  for various settings of  $\kappa$ . We can make the following observations. (a) The sharpening gain is a decreasing function of the patch variance. This is a desirable

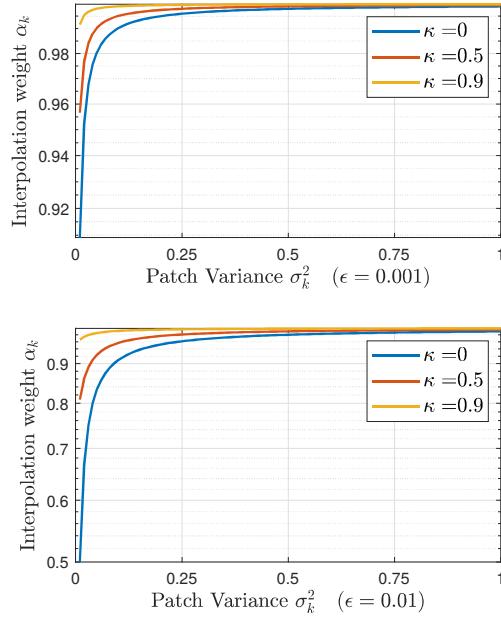


Fig. 2: Interpolation weight  $\alpha_k$  as a function of patch variance  $\sigma_k^2$  and filter parameter  $\kappa$  under two fixed settings of  $\epsilon$  (top  $\epsilon = 0.001$ , bottom  $\epsilon = 0.01$ ). As  $\kappa$  is increased,  $\alpha_k$  is less adapted to  $\sigma_k^2$  and is close to the constant 1 for  $\sigma_k^2 > T$  where  $T$  is a signal dependent threshold.

property of the sharpening filter. It performs a higher degree of enhancement on a patch of smaller variance which is usually due to low contrast. (b) The level of enhancement can be controlled by setting the parameter  $\kappa$ . A bigger value will lead to a bigger sharpening gain for the patch of fixed variance. (c) The parameter  $\epsilon$  also controls the sharpening gain. For the same setting of  $\kappa$ , a bigger value of  $\epsilon$  will lead to a bigger value of the sharpening gain.

### B. The guided form and its properties

In this section, we develop a guided version of the proposed filter by a further generalization of smoothing-sharpening formulation stated in section (II-A) and the guided filter.

#### 1) The guided smoothing-sharpening formulation and the guided filter

Referring to the filter formulation stated in (9), we can see that one of the key components is the Laplacian which is a second derivative operation and is thus sensitive to noise in the image  $I$ . When a guidance image  $G$ , which is assumed to have a higher signal-to-noise ratio, is available, a generalization is to replace the Laplacian calculated on  $I$  by the Laplacian calculated on  $G$  such that the filter can be written as:

$$J(q) = \mu(n) - \alpha \Delta G(q) \quad (29)$$

where  $\mu(n)$  is the mean of the patch centered at  $I(n)$  and

$$\Delta G(q) = \frac{1}{N} \sum_{m \in \Omega_n} (G(m) - G(q)) \quad (30)$$

Although this idea is technically sound, how to determine  $\alpha$  remains a problem. We will revisit the basic idea of the original guided filter and show how this problem can be solved

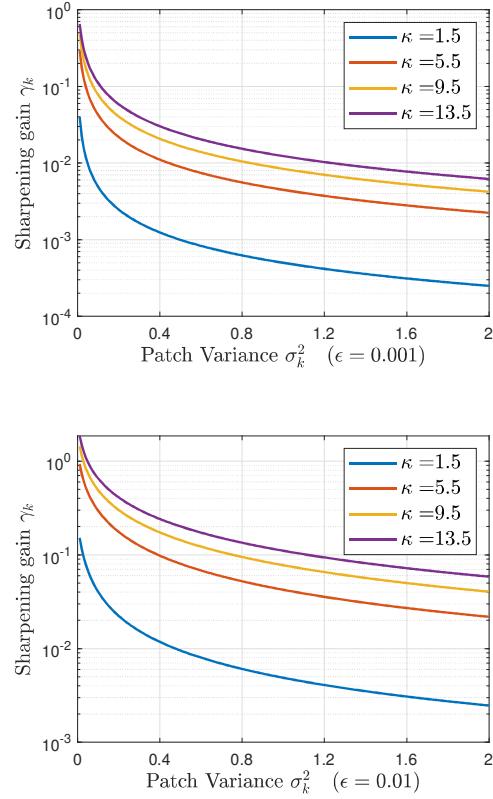


Fig. 3: The sharpening gain  $\gamma_k$  is a decreasing function of the patch variance  $\sigma_k^2$  for various settings of  $\kappa$  under two fixed settings of  $\epsilon$  (top  $\epsilon = 0.001$ , bottom  $\epsilon = 0.01$ ). The sharpening gain  $\gamma_k$  is bigger for a bigger value of  $\kappa$  and is a decreasing function of the patch variance  $\sigma_k^2$ .

by using a similar approach as the development of the self-guided version of the proposed filter described in the previous section.

In the original guided filter, the patch model is given by

$$J_k(q) = a_k G_k(q) + b_k \quad (31)$$

The two parameters  $a_k$  and  $b_k$  are determined by minimization of the cost function over the patch data

$$\{a_k, b_k\} = \min_{a_k, b_k} \sum_{q \in \Omega_k} [J_k(q) - I_k(q)]^2 + \frac{\epsilon}{2} a_k^2 \quad (32)$$

It can be shown that

$$b_k = \mu_k - a_k \nu_k \quad (33)$$

where  $\mu_k$  and  $\nu_k$  are the mean for the  $k$ th patch of image  $I$  and  $G$ , respectively. The parameter  $a_k$  is given by

$$a_k = \frac{\phi_k}{\varsigma_k^2 + \epsilon} \quad (34)$$

where  $\varsigma_k^2$  is the patch variance of  $G$ , and  $\phi_k$  is the sample covariance

$$\phi_k = \frac{1}{N} \sum_{q \in \Omega_k} (G_k(q) - \nu_k)(I_k(q) - \mu_k) \quad (35)$$

Substituting (33) into (31), we have a new patch model

$$J_k(q) = \mu_k + a_k(G_k(q) - \nu_k) = \mu_k - a_k \Delta G_k(q) \quad (36)$$

where  $\Delta G_k(q) = \nu_k - G_k(q)$  is the generalization of the Laplacian defined in (8).

Comparing the guided filter model stated in (36) with the proposed smoothing-sharpening model stated in (29), we can see that they are similar. Simply put, if we start with the filter model stated in (29), we will determine the same filter as the guided filter which is a smoothing filter. In the next section, we address the problem of how to develop a smoothing-sharpening guided filter.

### 2) The proposed guided smoothing-sharpening filter

To use (36) to develop the smoothing and sharpening filter, we face a new difficulty that  $a_k$  can be positive or negative. This is unlike the case for the self-guidance version developed in section III-A1 where it is always the case  $a_k \geq 0$ . Referring to (34), the sign of  $a_k$  is defined by the sign of the covariance  $\phi_k$  such that

$$\text{sign}(a_k) = \text{sign}(\phi_k) \quad (37)$$

The role of the sign of the covariance  $\phi_k$  can be explained as follows. Because  $\Delta I_k(q)$  is replaced by  $\Delta G_k(q)$ , it requires that the two patches  $I_k$  and  $G_k$  must be correlated. If they are positively correlated, then it is expected the two Laplacians are of the same sign. If they are negatively correlated, then it is expected the two Laplacians are of opposite sign. Thus a correction of the sign of the Laplacian calculated on the guidance image is required. The definition of  $a_k$  stated in (34) automatically satisfies this requirement.

In light of the above discussion, we can develop the guided version of the smoothing-sharpening filter by defining the guided patch interpolation model as the following

$$J_k(q) = \mu_k + \text{sign}(\phi_k)\alpha_k(G_k(q) - \nu_k) \quad (38)$$

such that  $\alpha_k$  is a positive parameter. We can then follow the same procedure as that presented in section III-A1 for the development. More specifically, the negative log-likelihood is

$$\begin{aligned} -\log p(\{J_k(q)\}|\alpha_k) &= \frac{1}{2\tau^2 N} \sum_{q \in \Omega_k} (I_k(q) - J_k(q))^2 \\ &= \frac{\varsigma_k^2}{2} \alpha_k^2 - |\phi_k| \alpha_k \end{aligned} \quad (39)$$

where we have set  $\tau = 1$  as before. Using the same generalized Gamma distribution as the prior, we obtain the negative log-posterior as the cost function  $D(\alpha_k)$

$$\begin{aligned} D(\alpha_k) &= -\log p(\{J_k(q)\}|\alpha_k) - \log p(\alpha_k) \\ &= \frac{\varsigma_k^2}{2} \alpha_k^2 - |\phi_k| \alpha_k + \frac{1}{2\theta^2} \alpha_k^2 - \eta \log \alpha_k \end{aligned} \quad (40)$$

We also follow the same parameter settings of the self-guided form by letting  $\epsilon = 1/\theta^2$  and  $\eta = \kappa\epsilon$ . Solving  $\partial D/\partial \alpha_k = 0$  and re-arranging the results, we have

$$\alpha_k = \frac{1}{2} \left\{ \frac{|\phi_k|}{\varsigma_k^2 + \epsilon} + \sqrt{\left( \frac{|\phi_k|}{\varsigma_k^2 + \epsilon} \right)^2 + \frac{4\kappa\epsilon}{\varsigma_k^2 + \epsilon}} \right\} \quad (41)$$

### 3) Properties

We can clearly see that when  $G = I$ , we have  $\varsigma_k^2 = \phi_k = \sigma_k^2$ . The guided version of the proposed filter stated by (42) reduces to its self-guided version stated by (26). The relationship between the proposed filter and original guided

filter can be revealed by substitution of (34) into (41), which results in the following

$$\alpha_k = \frac{1}{2} \left\{ |a_k| + \sqrt{a_k^2 + \frac{4\kappa\epsilon}{\varsigma_k^2 + \epsilon}} \right\} \quad (42)$$

We can also see that  $\alpha_k \geq |a_k|$  from (42). The original guided filter is a special case of the proposed filter when  $\kappa = 0$  leading to  $\alpha_k = |a_k|$ .

Next we discuss under what parameter setting the filter is smoothing or sharpening. The analysis presented in section II-A can not be used, because the Laplacian is calculated on the guidance image rather than on the image to be processed. The following analysis is based on an observation that a smoothing filter will reduce the patch variance while a sharpening filter will increase the patch variance. It is also assumed that the patch mean is not changed by the filter. This is a reasonable assumption because smoothing and sharpening usually do not change the average brightness of the image.

To make the discussion easy to follow, we first define the patch variance for the three images as

$$\text{Original image : } \sigma_k^2 = \frac{1}{N} \sum_{q \in \Omega_k} (I_k(q) - \mu_k)^2 \quad (43)$$

$$\text{Guided image : } \varsigma_k^2 = \frac{1}{N} \sum_{q \in \Omega_k} (G_k(q) - \nu_k)^2 \quad (44)$$

$$\text{Filtered image : } \tau_k^2 = \frac{1}{N} \sum_{q \in \Omega_k} (J_k(q) - \mu_k)^2 \quad (45)$$

In equation (45), we substitute  $J_k(q)$  by the patch interpolation model stated in (38) and use the definition stated in (44), we obtain

$$\tau_k^2 = \alpha_k^2 \varsigma_k^2 \quad (46)$$

Next we identify parameter settings that lead to reduced variance, i.e.,  $\tau_k^2/\sigma_k^2 \leq 1$  for smoothing or increased variance, i.e.,  $\tau_k^2/\sigma_k^2 > 1$  for sharpening. It can be shown that

$$\frac{\tau_k^2}{\sigma_k^2} = \frac{1}{2} \rho_k^2 \hat{\varsigma}_k^2 \left\{ 1 + \sqrt{1 + \frac{4\kappa\epsilon}{\sigma_k^2 \rho_k^2 \hat{\varsigma}_k^2}} \right\} + \frac{\kappa\epsilon \hat{\varsigma}_k^2}{\sigma_k^2} \quad (47)$$

where  $\hat{\varsigma}_k^2 = \frac{\varsigma_k^2}{\varsigma_k^2 + \epsilon}$  and  $\rho_k$  ( $|\rho_k| \leq 1$ ) is the cross correlation coefficient of the two patches defined as

$$\rho_k = \frac{\phi_k}{\sigma_k \varsigma_k} \quad (48)$$

where  $\phi_k$  is the sample covariance between the corresponding patches of the original and guided images. It is defined in equation (35).

In a special case in which  $\kappa = 0$  leading to the original guided filter, we have the following results

$$\frac{\tau_k^2}{\sigma_k^2} = \rho_k^2 \eta_k^2 = \rho_k^2 \left[ \frac{\varsigma_k^2}{\varsigma_k^2 + \epsilon} \right]^2 < 1 \quad (49)$$

This means that the original guided filter is always a smoothing filter.

However, there is a highly non-linear relationship between the ratio  $\tau_k^2/\sigma_k^2$  and the two filter parameters  $\kappa$  and  $\epsilon$  for the two patches in  $I$  and  $G$  where the variances ( $\sigma_k^2$  and  $\varsigma_k^2$ ) and

correlation coefficients ( $\rho_k$ ) can be calculated. In theory, we can calculate the required parameter settings such that for two given patches the ratio is greater than one or less than one. But doing so adds considerable computation burden in practice. A practical approach is to let the user set the two parameters such that the desirable result is produced. In this regard, we can see that the ratio  $\tau_k^2/\sigma_k^2$  is an increasing function of  $\frac{\kappa\epsilon}{\sigma_k^2}$ . For the case where  $\epsilon$  and  $\sigma_k^2$  are fixed, increasing  $\kappa$  will increase the ratio and the patch is more likely to be sharpened (or less smoothed). For the case where  $\kappa$  and  $\epsilon$  are fixed, a patch with a larger/smaller value of variance is less/more likely to be sharpened. This is a desirable feature because more sharpening should be applied to areas of less variance.

### C. Implementation and computational complexity

The implementation is similar to that of the original guided filter. For each patch, we calculate the parameter  $\alpha_k$  using either (27) for the self-guided version or (41) for the guided version. We then perform the weighted average operation. More specifically, for the self-guided version, the filter output is given by

$$J(p) = I(p) \sum_{k=1}^N w_k \alpha_k + \sum_{k=1}^N w_k (1 - \alpha_k) \mu_k \quad (50)$$

For the guided version, the filter output is given by

$$J(p) = G(p) \sum_{k=1}^N w_k \beta_k + \sum_{k=1}^N w_k (\mu_k - \beta_k \nu_k) \quad (51)$$

where  $\beta_k = \text{sign}(\phi_k)\alpha_k$  and  $a_k$  is given by (34).

We calculate the weight for the self-guided and the guided case as follows

$$w_k = \frac{c_k}{1 + (\sigma_k^2/(s\bar{\sigma}^2))^2} \quad (52)$$

and

$$w_k = \frac{c_k}{1 + (\varsigma_k^2/(s\bar{\varsigma}^2))^2} \quad (53)$$

where  $c_k$  is a normalization factor to ensure  $\sum_{k=1}^N w_k = 1$ ,  $s$  is a user defined scale parameter,  $\bar{\sigma}^2$  and  $\bar{\varsigma}^2$  are the average of  $\sigma_k^2$  and  $\varsigma_k^2$  over the whole image.

The proposed guided smoothing-sharpening filter can be implemented in MATLAB of which the code is shown in Appendix. The self-guided version has a similar implementation. We assume the four parameters are given: patch radius ( $r$ ), Kappa ( $\kappa$ ), Epsilon ( $\epsilon$ ), and Scale ( $s$ ). We can see from the brute-force implementation that the proposed filter has an  $\mathcal{O}(N)$  complexity which is the same as that of the original guided filter. It can be implemented by using 7 linear filters.

For color images, we can process each color component individually. Alternatively, we can convert the image from RGB to HSV color space. Filtering is performed on the value channel. The processing result is then combined with the hue and saturation channels and is converted back to RGB.

## IV. APPLICATIONS EXAMPLES

There are two purposes of this section: validation (section IV-A) of the theoretical analysis of the proposed filter, and demonstration of successful applications in (a) adaptive smoothing and sharpening (section IV-B) based on extracted information of texture, depth and blurriness, and (b) information fusion (section IV-C) for denoising and creating high resolution multi-spectral images.

### A. Effects of parameter settings for the self-guidance case

We demonstrate the properties of the filter and confirm the theoretical analysis presented in section III-A2. The proposed filter has 3 user defined parameters: (a) The patch radius  $r$ , (b) the sharpening/smoothing gain  $\kappa$ , and (c) the regularization parameter  $\epsilon$ . In this section we study the effect of these parameters on the processed image.

In Fig. 4 we demonstrate the effect of varying  $\kappa$  by keeping the rest of the parameters fixed. Fig. 4a and 4b show the effect of smoothing when  $0 < \kappa < 1$ . Smaller  $\kappa$  values increase the smoothing level on the result image. Fig. 4d and 4e show the effect of sharpening when  $\kappa > 1$ . Larger  $\kappa$  values produce a sharper result. The image shown in Fig. 4c is produced by setting  $\kappa = 1$ . We can verify that it is exactly as the original image. Thus when  $\kappa = 1$  the filter produces no smoothing or sharpening effect.

Next, we study the effect of patch size and  $\epsilon$  for sharpening and smoothing separately. To set the filter in smoothing mode we set a fixed  $\kappa = 10^{-2}$ . In Fig. 5 the results are organized in such a way that the radius varies from 5 to 10 column-wise while  $\epsilon$  varies from  $10^{-2}$  to 1 row-wise. Results shown in this figure clearly show the edge preserving capabilities of the filter in smooth mode. Setting a larger  $\epsilon$  value produces a more washed out result, while increasing the radius of the filter also produces a stronger smoothing result.

The results shown in Fig. 6 are obtained by setting  $\kappa = 20$  to demonstrate the sharpening effects as a function of different  $\epsilon$  and  $r$  values. These results confirm that the sharpening gain increases with the increase in  $\epsilon$  as previously stated in Fig. 3. Increasing the radius impacts the variance for each pixel position. As a result, it produces a change in the sharpening gain.

### B. Application in adaptive smoothing-sharpening

As mentioned in the previous section, the value of  $\kappa$  controls the sharpening/smoothing gain of the proposed filter, in this section we propose three pixel-adaptive smoothing and sharpening algorithms by defining  $\kappa$  as a non-linear transformation of a feature map such as depth, blurriness or texture.

The non-linear transformation used in this paper is simply a variation of the Gompertz<sup>1</sup> function which is a sigmoid function. It is defined as:

$$g(t) = ae^{-be^{-ct}} \quad (54)$$

<sup>1</sup>[https://en.wikipedia.org/wiki/Gompertz\\_function](https://en.wikipedia.org/wiki/Gompertz_function)



Fig. 4: Effect of varying  $\kappa$  for settings  $r = 11, \epsilon = 0.01, scale = 1$ . (a) and (b) Smoothing ( $0 < \kappa < 1$ ). (c) Original image ( $\kappa = 1$ , no filtering). (d) and (e) Sharpening ( $\kappa > 1$ ).

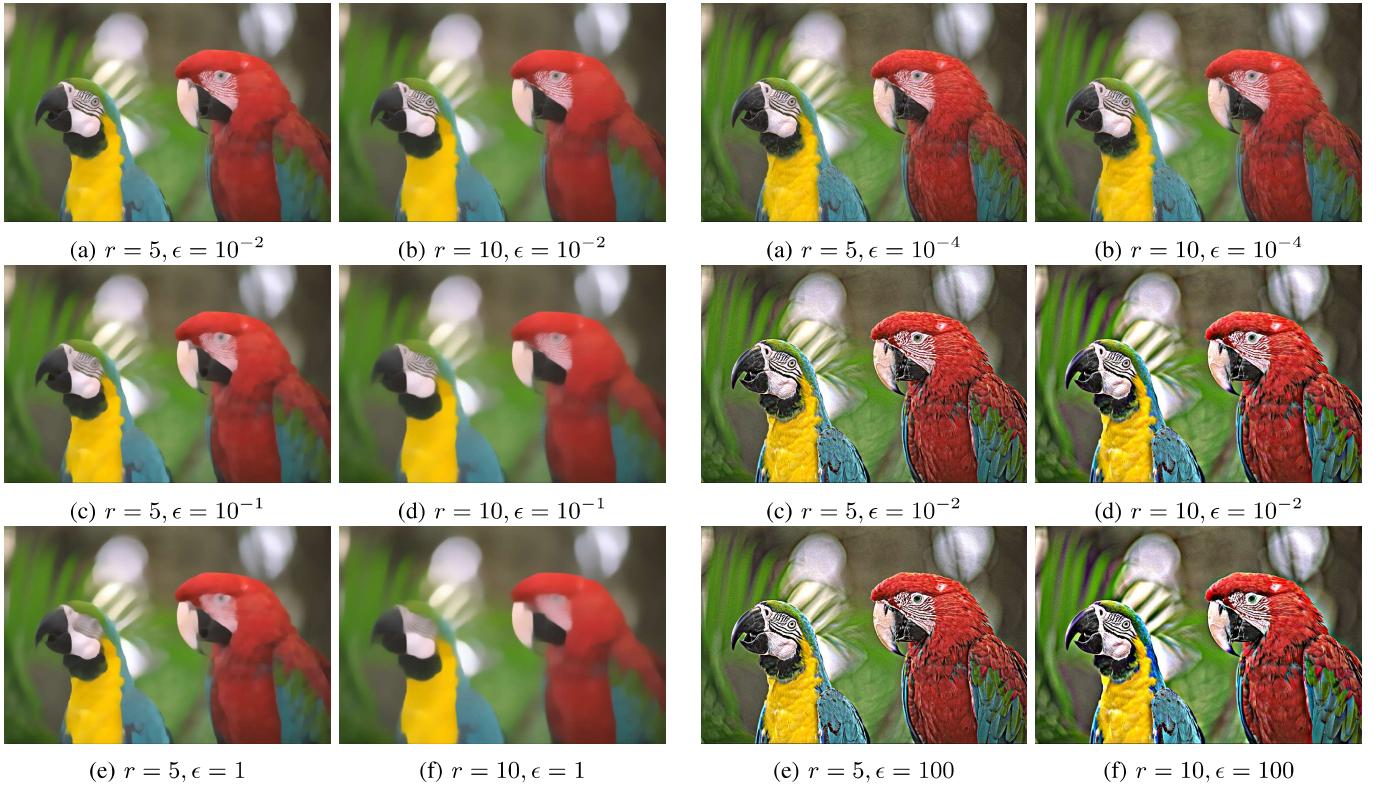


Fig. 5: Effect of varying  $\epsilon$  and  $r$  in smoothing mode for fixed settings of  $\kappa = 10^{-2}$  and  $scale = 0.25$ .

where  $a$ ,  $b$  and  $c$  are three parameters. Let us consider  $t$  ( $t \in [0, 1]$ ) as a feature map extracted from the input image, the parameter of the proposed filter  $\kappa$  is defined as:

$$\kappa = (\kappa_{max} - \kappa_{min})e^{-0.69 \times e^{-c(t-t_0)}} + \kappa_{min} \quad (55)$$

where  $\kappa_{min}$  and  $\kappa_{max}$  are the minimum and the maximum values that the gain  $\kappa$  will take,  $c$  is the growth rate of the transformation and  $t_0$  is the value of  $t$  that produces  $\kappa_i = (\kappa_{max} - \kappa_{min})/2$ .

We show an example of the non-linear transformation in Fig. 7, where there are two well defined areas in the

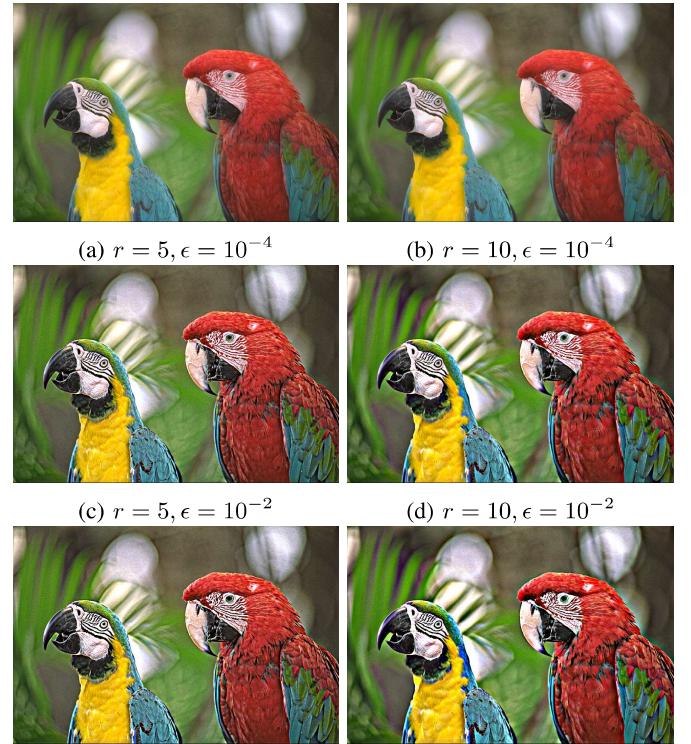


Fig. 6: Effect of varying  $\epsilon$  and  $r$  in sharpening mode for fixed settings of  $\kappa = 20$  and  $scale = 1$ .

function: the smoothing region where the  $\kappa_{min} \leq \kappa < 1$  and the sharpening region where  $1 < \kappa \leq \kappa_{max}$ . We can control the level of smoothing or sharpening by changing the values of  $\kappa_{min}$  and  $\kappa_{max}$ . Setting  $\kappa_{min} = 1$  will cancel the smoothing effect also setting  $\kappa_{max} = 1$  will cancel or not produce any sharpening on the image. In the following subsections we produce content adaptive  $\kappa$  by using the non-linear transformation on feature maps.

### 1) Texture guided smoothing and sharpening of face images

A challenge in sharpening portraits by a non-adaptive unsharp masking algorithm is that undesirable effect on

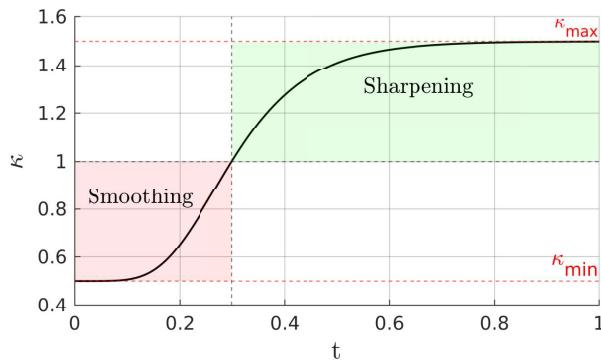


Fig. 7: Non-linear transformation, using  $\kappa_{min} = 0.5$ ,  $\kappa_{max} = 1.5$ ,  $c = 10$ ,  $t_0 = 0.3$

skin regions is usually produced. Examples are shown in Figures 8(b) and 9(g,h) in which the skin part of the image is sharpened. To solve this problem, we first estimate a binary skin map using one of the many algorithms for skin segmentation, e.g., [18]–[20]. We then transform the binary skin map by using the non-linear transformation in (55) to obtain pixel-adaptive  $\kappa$ , which is used in the proposed filter to sharpen non-skin regions only while gently smoothing the skin region to produce a notable face enhancement.

In Fig. 8 we compare the result of the proposed method against the popular unsharp masking (UM). UM algorithm has a fixed gain ( $\gamma$ ) to amplify the high frequency components of the image. It can be seen that our method produces a more aesthetically pleasing result in skin region while still sharpening the non-skin region.

In Fig. 9 we compare our method with other state-of-the-art sharpening methods such as generalized unsharp masking (GUM) [21], unsharp masking [1] and contrast adaptive sharpening (CAS)<sup>2</sup>. Our filter produces a more natural and aesthetically appealing effect on the image than UM and its performance is similar to GUM and CAS.

#### 2) Depth guided smoothing for shallow depth of field

Modern mobile phones can have multiple high resolution cameras to capture high quality images which can be used to estimate the depth map of a scene. We combine this capability and the proposed algorithm to produce a shallow depth of field (SDoF) effect which is frequently used to emphasize the main object. Traditionally, SDoF effect is achieved by using a lens with a large aperture in a SLR camera. The proposed algorithm of pixel-adaptive smoothing and sharpening can be used as post-processing tool to create the SDoF effect. The idea is to obtain the depth map from the phone and perform the non-linear transformation to determine  $\kappa$  which is used in the filtering process. In this experiment, we used a Samsung Galaxy Note 9 phone in live focus mode to capture the image and depth map information. We note that methods such as [22], [23] can be used to estimate the depth map from a single image if only one camera is available.

The depth map  $D$  is a gray-scale image with values in the range  $[0,1]$ . A closer object in the scene has a smaller  $D$  value. So the feature map is defined as  $t = 1 - D$ . Applying the non-

linear transformation, the depth information is mapped to  $\kappa$  which is a decreasing function of  $D$  leading to progressively smoothing effect as the distance increases. At the same time the closer objects are sharpened to correct slight out of focus or blur.

In Fig. 10 and Fig. 11 we show two different results where the foreground is sharpened and the background is smoothed to produce the SDoF effect. We can see that the background is smoothed with a natural appearance to simulate the defocus blur, also the foreground is sharpened reducing the blur due to movement in Fig. 10 and out of focus in Fig. 11.

#### 3) Content-aware seam carving

We present an application of the background smoothing algorithm detailed in previous section as a pre-processing step for seam carving. Seam carving was introduced in [24] as an effective tool for resizing an image without significant change to main objects. The idea is to delete pixels of unimportant details in an image. The importance of a pixel is measured by a function of gradient. A natural image often contains details such as trees, sand, grass which are usually less important compared to the object of interest such as human. However, a direct application of gradient-based seam-carving may lead to unsatisfactory results. An example is shown in Fig. 12b. A solution to this problem is to use content-aware image resizing. To avoid elimination of information in foreground, we pre-process the input image by using the proposed SDoF algorithm which not only smooths out details in the background but also emphasizes the object of interest by sharpening it. Results are shown in Fig. 12d which shows that after SDoF filtering the seams are not running over the boy's face. As a result, the seam-carving algorithm produces a better output image.

#### 4) Blurriness guided sharpening and smoothing

When an object in a scene is outside the focal plane it is defocused with a blur level directly proportional to the distance from the focal plane [25]. Due to the limited depth of field intrinsic in most optical systems, the defocus blur is present in most of images. When the depth information is not available, the defocus blur becomes the simplest depth cue in an image [26] and it is widely used by photographers to make the main object of the scene to stand out from the background. Sometimes, due to wrong focal settings, images need to be sharpened or deblurred to achieve a more pleasant result.

When an image has defocus blur present, the sharpening process becomes a challenge. Sharpening highly defocused regions produces artifacts and sharpening in focus regions can lead to over-sharpening. To tackle these problems we propose a method by sharpening and smoothing an image adaptively using the defocus-blur map to compute a spatially varying  $\kappa$  map. The key behind the success of our method is the estimation of the defocus-blur map.

Methods have been developed to estimate the defocus-blur map. Some of the methods use multiple images or special hardware [27]–[29] while others use a single image. These methods fall into two main categories. The first one is the traditional image processing methods such as [30], [31] which perform a frequency domain analysis to estimate the defocus-blur map or [32], [33] which use changes in gradient to estimate the blur level at edges and then interpolate those level

<sup>2</sup><https://www.amd.com/en/technologies/radeon-software-fidelityfx>

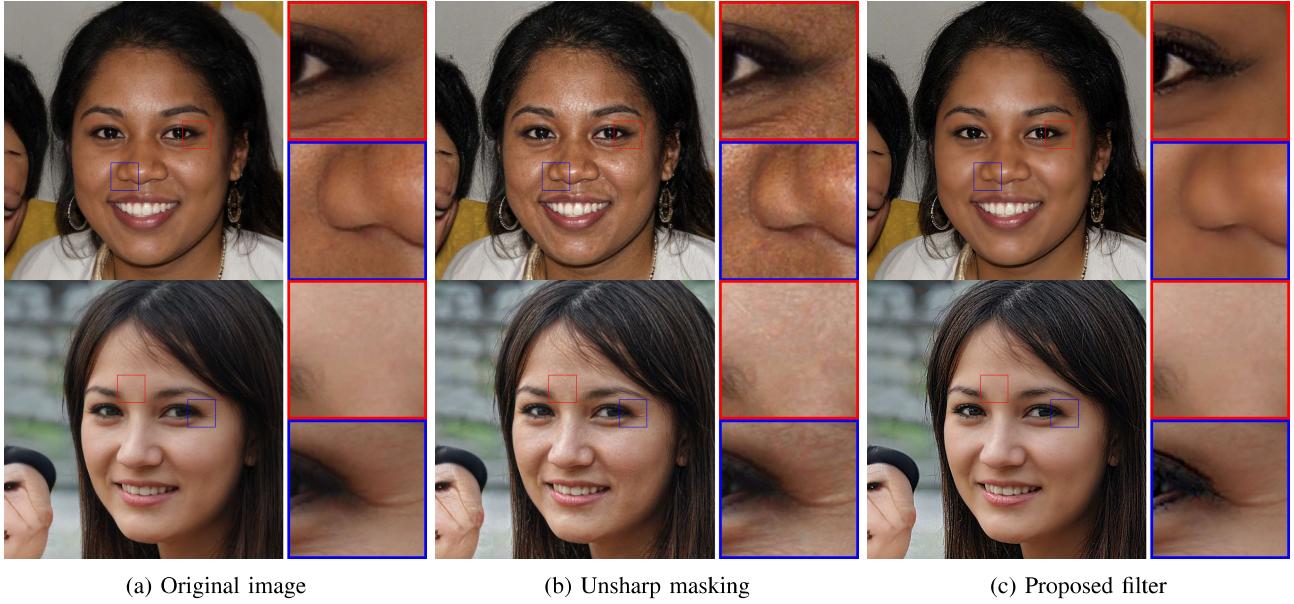


Fig. 8: Face enhancement results. (a) Original image. (b) UM ( $\gamma = 1.1$ ). (c) The proposed filter ( $r = 3, \epsilon = 0.01, N_{iter} = 1, \kappa_{max} = 5, \kappa_{min} = 0.1$ ). The proposed method does not sharpen skin regions producing more aesthetically pleasing result.

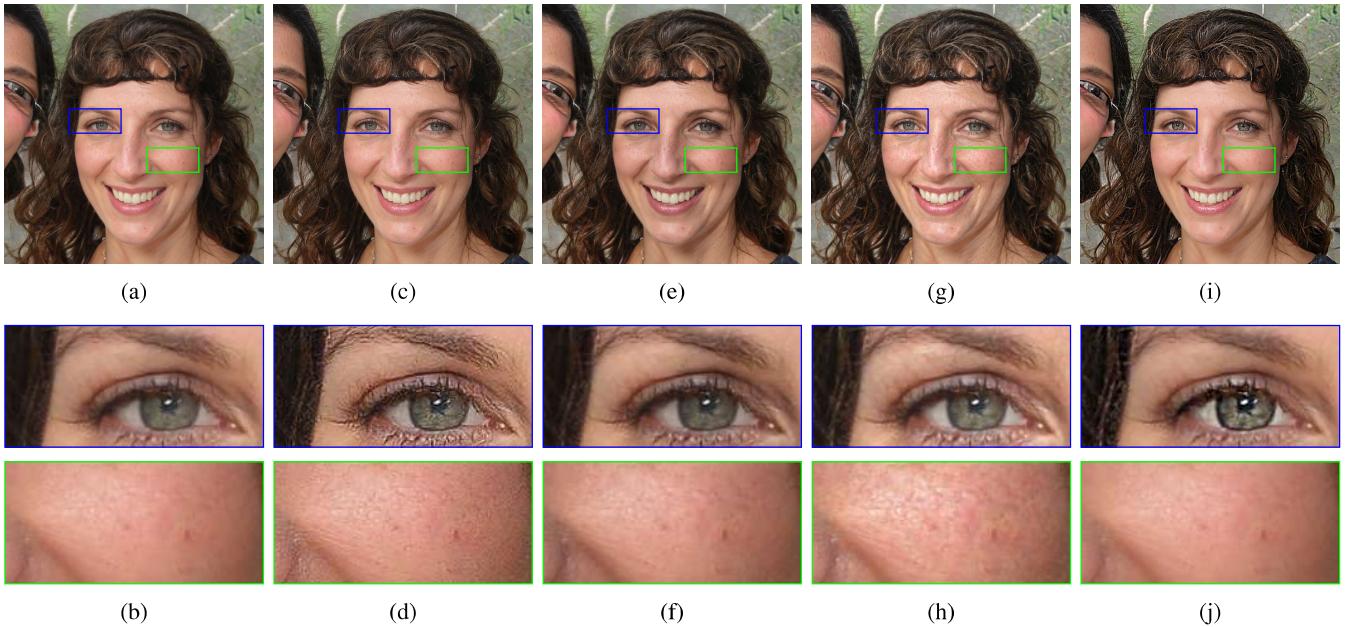


Fig. 9: Face enhancement. (a, b) Original images. (c, d) CAS results. (e,f) GUM results. (g,h) UM results. (i,j) Proposed filter results.

to the rest of the image using a matting algorithm. The second one is the machine learning based methods such as [34] which uses an end-to-end CNN to estimate the defocus map and [35] which estimates the blur map by using a regression tree field (RTF) model.

In this work we use entropy, which is a measure of pixel variation in a local area, as an indicator for the defocus. The defocus-blur map  $D$  at a pixel location is defined as the entropy of a patch centered at that particular pixel. The map is then refined by using the guided filter [3] which uses the original image as the guidance. The refined defocus-blur map is non-linearly transformed by equation (54) to

obtain the desired  $\kappa$  map. The proposed filter with the pixel-adaptive  $\kappa$  is applied to the image. The result is adaptive smoothing-sharpening based on the local entropy information as a measure of defocus-blur.

In Fig. 13 we first show an image which contains different levels of focal blur. The boy's face (image (a)) is in focus while the background is slightly out of focus. The refined entropy map (image (b)) is produced by the MATLAB function `entropyfilt` with a window size of 33x33 pixels and the result is refined by using guided filter [3] ( $r = 32, \epsilon = 0.01$ ). The  $\kappa$  map is shown in image (c).

In the second row of Fig. 13 we show 3 different results.



(a)



(b)



(c)



(d)

Fig. 10: Shallow depth of field (SDoF) effect. (a,b) Original image. (c,d) SDoF using the proposed filter ( $r = 3, \epsilon = 10, N_{iter} = 1, \kappa_{max} = 2, \kappa_{min} = 0$ ). Our filter not only smooths the background, it also sharpens the object to correct small movements and out of focus.

Image (d) is the result of smoothing areas with some degrees of focal blur. The background is successfully blurred while the boy's face remains sharp leading to a pleasing SDoF effect. Image (e) is the result of sharpening the out-of-focus regions to increase the depth of field. The background appears sharper producing the sensation that the scene's depth is slightly larger than the original. In both cases, the boy's face which is in focus, is not changed (achieved by setting  $\kappa = 1$ ). However, we should point out that the proposed filter can be easily configured to sharpen the in-focus objects by adjusting the parameters of the non-linear transformation. As a comparison, image (f) shows the effect of sharpening with a fixed  $\kappa$  value for the whole image, leading to an image of non natural appearance.

To further demonstrate the advantage of the proposed smoothing-sharpening filter, we compare results from the gradient domain guided filter (GDGF) [9], the weighted guided filter (WGF) [8], and the side window guided filter (SWGF<sup>3</sup>) [36]. We tune parameters of these 3 filters such that the background is maximally smoothed while preserving information of the face shown in Fig. 13(a) as much as



(a)



(b)



(c)



(d)

Fig. 11: Shallow depth of field (SDoF). (a,b) Original image. (c, d) SDoF using the proposed filter ( $r = 1, \epsilon = 100, N_{iter} = 10, \kappa_{max} = 2, \kappa_{min} = 0$ ). Our filter not only smooths the background, it also sharpens the object.

possible. Results are shown in Fig. 14 which clearly shows that the proposed filter has the best capability for blurring the background while preserving the information of the face which is almost unchanged. On the other hand, both GDGF and WGF can blur the background to some extend at the cost of smoothing the face. For the SWGF, we have to choose a relatively small patch size to prevent the face being overly smoothed. As a result, there is little effect on smoothing the background.

### C. External guided smoothing and sharpening

#### 1) Flash/no-flash sharpening

In this section we demonstrate another application of the proposed filter where guided smoothing and sharpening is required. For example, a picture taken under low light condition contains a high level of noise due to the use of high ISO setting. One of the successful applications of the original guided filter [3] is in processing images captured under low light condition without using the flash. One such image is shown in Fig. 15a. The idea is to use another image captured with flash-on as a guidance to enhance the one without flash. In the original implementation, the guided filter with parameters  $r = 8$  and  $\epsilon = 0.004$  are used. Result is

<sup>3</sup><https://github.com/YuanhaoGong/SideWindowFilter>

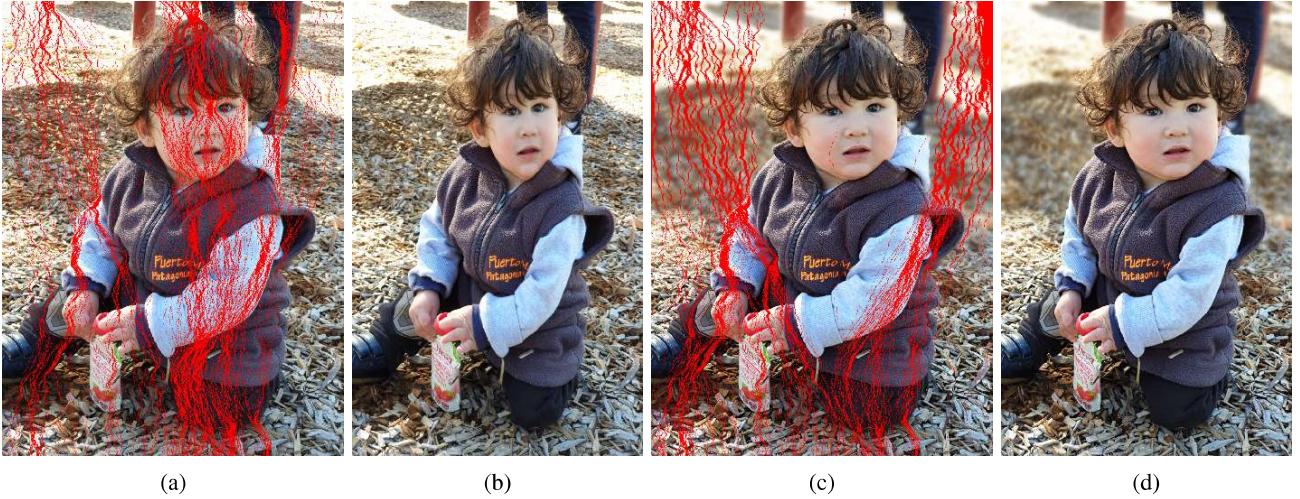


Fig. 12: Seam carving results. (a) Original image and seams to be removed. (b) Removed seams in (a). (c) Image after background smoothing and seam to be removed. (d) Removed seams in (c). The proposed background filter leads to a better result.

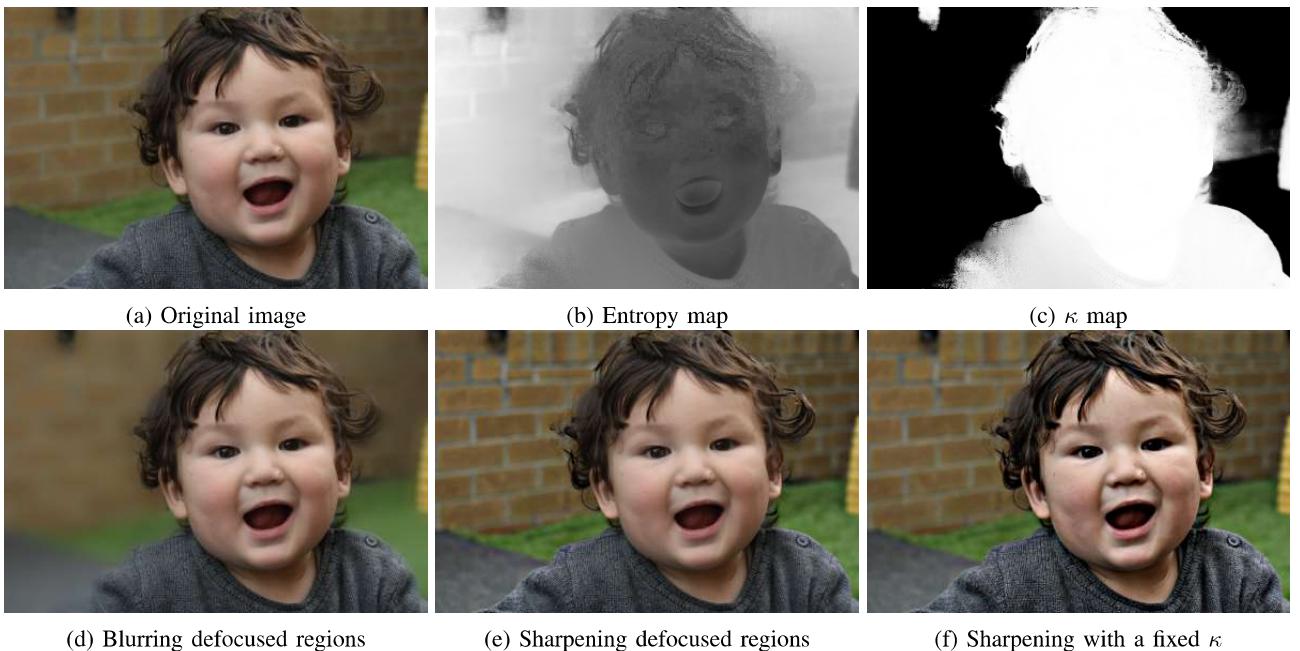


Fig. 13: Sharpening/smoothing guided by defocus. (a) Original image. (b) Refined entropy map. (c)  $\kappa$  map. (d) Result of blurring defocused regions to produce a shallow depth of field. (e) Sharpening defocused regions to increase the depth of field. (f) Sharpening the whole image with a fixed  $\kappa$ .

shown in Fig. 15(c), where we can see that noise has been greatly reduced while the color information is preserved. This is however at the cost of loss of details, e.g., details on the wall and on the vases. This is evident when we compare the result with the guidance image (with-flash).

To tackle this problem, we first apply the guided filter in an iterative manner. We represent the filter operation as  $J = GF(I, G)$  where  $GF$  denotes the guided filter. The iteration is performed as:  $J^{(0)} = I$  and  $J^{(n)} = GF(J^{(n-1)}, G)$ . Using the parameter settings  $r = 25$  and  $\epsilon = 10^{-6}$ , we perform 10 iterations and show the result in Fig. 15(d). Comparing the original GF result with the iterative GF result, we can see that the latter has retained more details of the original scene than

the former.

Next, we test the proposed guided smoothing-sharpening filter in the same iterative way, i.e., same parameter settings with 10 iterations. The proposed filter has two extra parameters:  $\kappa$  and the scale  $s$ . For simplicity, we set  $s = 1$  and  $\kappa = 10$  and 100 to study the sharpening effect. Results in Fig. 15(e) and (f) which show that the proposed filter does indeed produce sharper results than the iterative GF. To make a quantitative comparison, we calculate the total variation of the image. The total variation for image  $I$  is defined as

$$TV(I) = \sum_{c \in \{R, G, B\}} \sum_n |I_h^{(c)}(n)| + |I_v^{(c)}(n)| \quad (56)$$

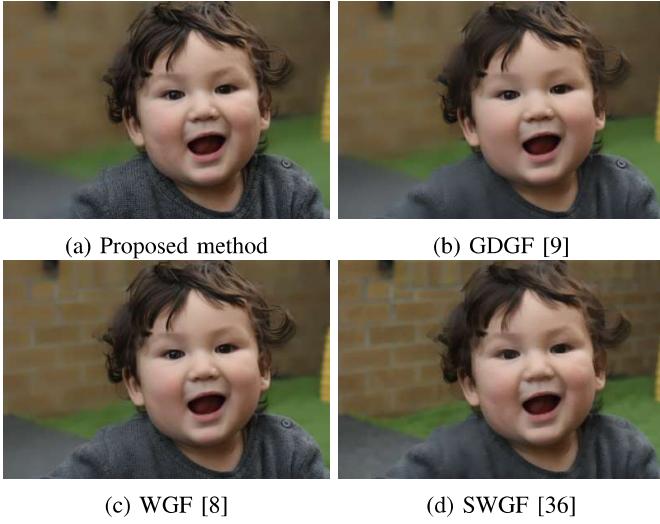


Fig. 14: Background smoothing. Comparison with other guided filters (a) Proposed method (same as Fig.13(d)). (b) GDGF [9] ( $r = 7, \epsilon = 0.01$ ). (c) WGF [8] ( $r = 7, \epsilon = 0.001$ ). (d) SWGF [36] ( $r = 3, N_{iter} = 3$ ).

TABLE I: The total variation of the image produced by the proposed iterative smoothing-sharpening filter. It is an increasing function of  $\kappa$  indicating the image appear to be sharper by using a larger setting of  $\kappa$ .

$\kappa$	0	10	50	100	200
$TV(\times 10^4)$	1.13	1.33	1.56	1.77	2.09

where  $I_h^{(c)}(n) / I_v^{(c)}(n)$  is the first derivative of the image along the horizontal/vertical direction at location  $n$  and the superscript  $c$  is used to indicate the color channel. Since the total variation is the sum of absolute values of the first derivative, we can use it as an indication of the sharpness of the image in this application. The total variations for different settings of  $\kappa$  are shown in Table I, where  $\kappa = 0$  corresponds to the iterative guided filter. We can see that the sharpness of the image is indeed an increasing function of  $\kappa$ .

Another issue is related to the number of iterations. In general, for  $\kappa > 1$ , more iterations tend to produce a higher degree of sharpening effect. How to set the number of iterations is application dependent and it can be a user specified parameter. For the image shown in Fig. 15, we empirically found that between 5 to 10 iterations and a setting of  $10 < \kappa < 100$  result in visually pleasing images.

## 2) Pan-sharpening

Multi spectral (MS) images usually have low spatial resolution but are rich in spectral information. On the other hand panchromatic images (called PAN images) have low spectral resolution but have high spatial resolution. For example, the IKONOS and QuickBird imaging sensors capture a PAN image with a spatial resolution of 1 and 0.6 m respectively and a MS image with a spatial resolution of 4 and 2.6 m respectively [37]. Pan-sharpening is a technique that combines information of MS images with PAN images to produce a high spatial resolution image with large spectral information. Pan-sharpening is a useful tool in many remote

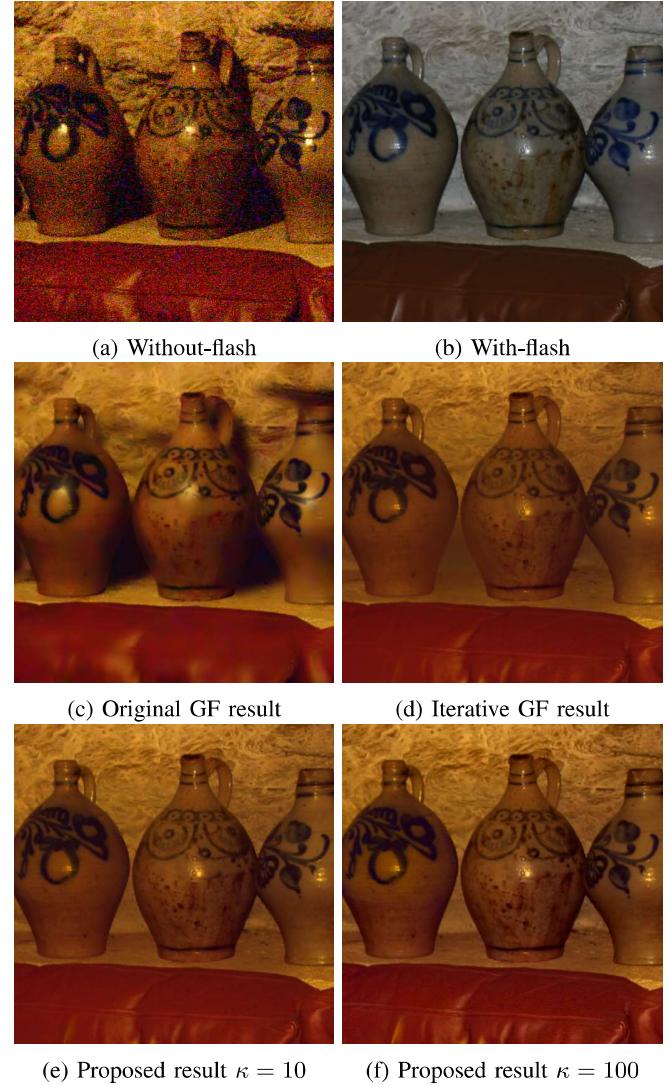


Fig. 15: Flash/no flash sharpening. (a) Image without flash. (b) Image with flash. (c) GF ( $r = 8, \epsilon = 0.004$ ). (d) Iterative GF ( $r = 25, \epsilon = 10^{-6}, N_{iter} = 10$ ). (e) The proposed filter ( $r = 25, \epsilon = 10^{-6}, N_{iter} = 10, s = 1, \kappa = 10$ ). (f) The proposed filter ( $r = 25, \epsilon = 10^{-6}, N_{iter} = 10, s = 1, \kappa = 100$ ).

sensing applications.

Pan-sharpening techniques have been continuously developed over the years. Some frequently used techniques include: Brovety transform (BT) [37], intensity-hue-saturation (IHS) [38], principal component analysis (PCA) [16], wavelet transform (WT) [39], [41], guided filter based methods such as [17], [42], and P+XS [40]. We use some of these methods to compare with the result of the proposed algorithm.

The proposed method consists of 3 steps similar to the GF-based approach [17]. The MS image is first up-sampled using nearest neighbor interpolation. The result is then processed by the proposed filter which uses the PAN image as a guidance to transfer the high spatial resolution information from the guidance image to the low resolution MS image. The last step is a histogram matching on the result image using the original MS image as a reference.

Fig. 16a and Fig. 16b show the up-sampled MS image and

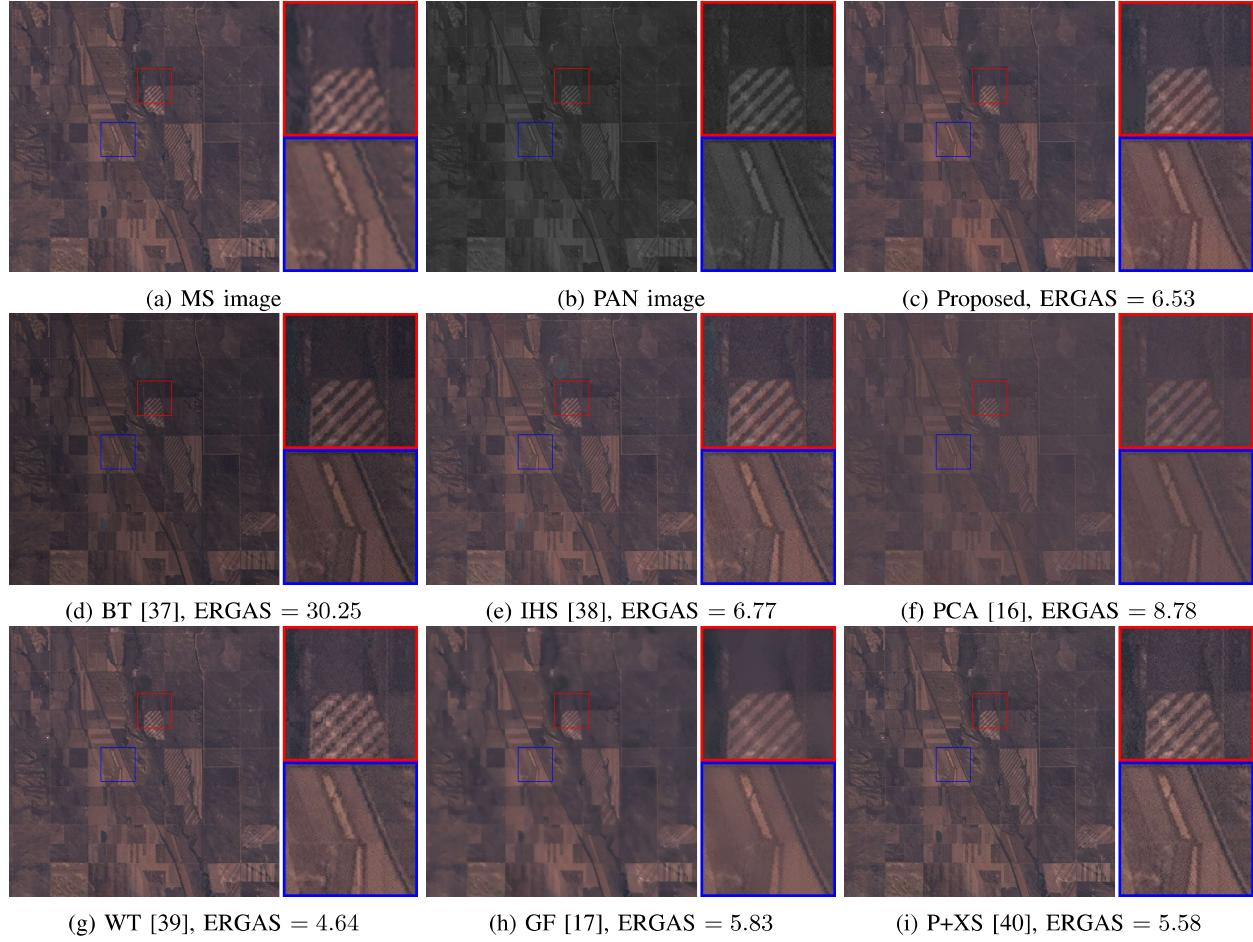


Fig. 16: Pan-sharpening results and comparison with other methods. (a) Original low resolution multi spectral image. (b) High resolution panchromatic image. (c) Proposed filter. (d) to (i) Other state of the art methods. The proposed filter successfully transfers the details from the high spatial resolution panchromatic image to the up-sampled multi spectral image while preserving the spectral resolution.

the PAN images respectively. The images are from United States Geological Survey database<sup>4</sup>. Fig. 16c shows the output of the proposed algorithm using  $r = 11, \epsilon = 0.1, \kappa = 1.2, scale = 0.5$ , the details of the PAN image are successfully transferred to the up-sampled MS image and the spectral resolution is preserved. In the second and third rows of Fig. 16 we present the results of the Pan-sharpening process using BT, IHS, PCA, WT, GF and P+XS algorithms. We can observe that our method produces sharp and high contrast results with large spatial and spectral information.

We use ERGAS [43] to quantify the quality of a Pan-sharpened image. ERGAS is a metric that calculates the spectral distortion. Ideally, its value should be zero. The ERGAS values for results produced by different methods are shown in the caption of Fig. 16. We can see that the result of the proposed filter is of about the same quality as those produced by state-of-the-art methods. Thus, the proposed filter is a new tool for Pan-sharpening with the ability to perform pixel-wise sharpening or smoothing.

## V. CONCLUSIONS

Smoothing and sharpening are two fundamental operations in image processing. They are usually related through the unsharp masking algorithm. In this paper, we have developed a new filter which can perform smoothing and sharpening depending on the setting of a parameter  $\kappa$ . The filter is a smoothing filter or a sharpening filter when  $0 < \kappa < 1$  or  $\kappa > 1$ . The systematic unification of these two operations in one filter is based on (a) a new Laplacian based filter formulation which unifies the smoothing and sharpening operations, (b) a patch interpolation model similar to the guided filter which provides the edge-awareness capability, and (c) the generalized Gamma distribution as the prior for parameter estimation. As a result the filter allows pixel-adaptive image smoothing/sharpening by adapting  $\kappa$  to local characteristics such as texture, depth, and blurriness. Based on the patch interpolation model, the proposed filter uses the guidance information in two ways. In self-guidance the proposed filter uses information of the image to be processed and has the ability to use other information to adapt  $\kappa$ . In external-guidance, the filter is similar to the guided filter, but has an extra ability of adaptive smoothing-sharpening. In addition, the proposed filter has the desirable edge-awareness

<sup>4</sup><https://earthexplorer.usgs.gov/>

property which retains sharp edges in smoothing and does not suffer from the halo effect in sharpening.

Using the filter in self-guidance we have developed adaptive smoothing-sharpening algorithms based on information of texture, depth and blurriness to enhance human face images, to create the effect of shallow depth of field, to perform adaptive processing based on local blurriness, and to pre-process an image to achieve better seam carving results. Using the filter in external guidance, we have combined images of under flash and no-flash conditions, producing much better results than those produced by using the guided filter. We have also demonstrated the successful application of the filter to solve the Pan-sharpening problem which combines information from multi-spectral images with a panchromatic image.

## APPENDIX

A brute-force implementation of the proposed filter in MATLAB is presented as the following code. We assume parameters such as: patch radius ( $r$ ), Kappa ( $\kappa$ ), Epsilon ( $\epsilon$ ), and Scale ( $s$ ), are provided by the user.

```

pad='symmetric';
N=(2*r+1)^2;
h=ones(2*r+1)/N;
%patch mean of I
mu=imfilter(I,h,pad);
%patch mean of G
nu=imfilter(G,h,pad);
%patch cov
phi=imfilter(I.*G,h,pad)-mu.*nu;
%patch var of G
vS=imfilter(G.*G,h,pad)-nu.*nu;
a=phi./(vS+Epsilon);
Beta=(a+sign(phi)).*sqrt(a.^2+4*kappa...
    *Epsilon./(vS+Epsilon))/2;
%weight calculation
w=vS./(s*mean(vS(:)));
w=1./((1+w.^2));
nor=imfilter(w,h,pad);
%final output
A=imfilter(Beta.*w,h,pad);
B=imfilter((mu-Beta.*nu).*w,h,pad);
J=(G.*A+B)./nor;

```

## REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, *Digital image processing*. New York, NY: Pearson, 2018.
- [2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, pp. 839–846, 1998.
- [3] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2012.
- [4] G. Deng, "Guided wavelet shrinkage for edge-aware smoothing," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 900–914, 2016.
- [5] X. Wei, Q. Yang, and Y. Gong, "Joint contour filtering," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1245–1265, 2018.
- [6] H. Yin, Y. Gong, and G. Qiu, "Side window guided filtering," *Signal Processing*, vol. 165, pp. 315–330, 2019.
- [7] Z. Sun, B. Han, J. Li, J. Zhang, and X. Gao, "Weighted guided image filtering with steering kernel," *IEEE Trans. Image Process.*, vol. 29, pp. 500–508, 2020.
- [8] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 120–129, 2014.
- [9] F. Kou, W. Chen, C. Wen, and Z. Li, "Gradient domain guided image filtering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4528–4539, 2015.
- [10] S. K. Mitra, H. Li, I.-S. Lin, and T.-H. Yu, "A new class of nonlinear filters for image enhancement," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 2525–2526, 1991.
- [11] G. Ramponi, "A cubic unsharp masking technique for contrast enhancement," *Signal Processing*, vol. 67, no. 2, pp. 211–222, 1998.
- [12] G. Ramponi, N. K. Strobel, S. K. Mitra, and T.-H. Yu, "Nonlinear unsharp masking methods for image contrast enhancement," *Journal of Electronic Imaging*, vol. 5, no. 3, pp. 353–367, 1996.
- [13] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 505–510, 2000.
- [14] F. Kou, W. Chen, Z. Li, and C. Wen, "Content adaptive image detail enhancement," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 211–215, 2015.
- [15] W. Ye and K.-K. Ma, "Blurriness-guided unsharp masking," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4465–4477, 2018.
- [16] V. P. Shah, N. H. Younan, and R. L. King, "An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1323–1335, 2008.
- [17] J. Qu, Y. Li, and W. Dong, "Hyperspectral pansharpening with guided filter," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2152–2156, 2017.
- [18] D. Dahmani, M. Cheref, and S. Larabi, "Zero-sum game theory model for segmenting skin regions," *Image Vis. Comput.*, vol. 99, p. 103925, 2020.
- [19] E. Buza, A. Akagic, and S. Omanovic, "Skin detection based on image color segmentation with histogram and k-means clustering," in *Proc. 10th Int. Conf. on Electrical and Electronics Engineering*, pp. 1181–1186, IEEE, 2017.
- [20] S. L. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color and edge information," in *Proc. Seventh Int. Symposium on Signal Processing and Its Applications*, vol. 1, pp. 525–528, 2003.
- [21] G. Deng, "A generalized unsharp masking algorithm," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1249–1261, 2010.
- [22] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. CVPR*, pp. 5162–5170, 2015.
- [23] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. CVPR*, pp. 270–279, 2017.
- [24] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, p. 10, 2007.
- [25] A. P. Pentland, "A new sense for depth of field," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 4, pp. 523–531, 1987.
- [26] G. Mather, "Image blur as a pictorial depth cue," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 263, no. 1367, pp. 169–172, 1996.
- [27] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, 2007.
- [28] D. T. Vu, B. Chidester, H. Yang, M. N. Do, and J. Lu, "Efficient hybrid tree-based stereo matching with applications to postcapture image refocusing," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3428–3442, 2014.
- [29] C. Zhou, S. Lin, and S. Nayar, "Coded aperture pairs for depth from defocus," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 325–332, IEEE, 2009.
- [30] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," in *Proc. IEEE CVPR*, pp. 2512–2519, IEEE, 2010.
- [31] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar, "Estimating spatially varying defocus blur from a single image," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4879–4891, 2013.
- [32] X. Zhang, R. Wang, X. Jiang, W. Wang, and W. Gao, "Spatially variant defocus blur map estimation and deblurring from a single image," *J. Vis. Commun. Image Represent.*, vol. 35, pp. 257–264, 2016.
- [33] H. Hu and G. De Haan, "Low cost robust blur estimator," in *Proc. Int. Conf. Image Process., ICIP*, pp. 617–620, IEEE, 2006.
- [34] J. Lee, S. Lee, S. Cho, and S. Lee, "Deep defocus map estimation using domain adaptation," in *Proc. IEEE CVPR*, pp. 12222–12230, 2019.
- [35] L. D'Andres, J. Salvador, A. Kochale, and S. Süstrunk, "Non-parametric blur map regression for depth of field extension," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1660–1673, 2016.

- [36] H. Yin, Y. Gong, and G. Qiu, "Side window filtering," in *Proc. IEEE Conf. CVPR*, pp. 8750–8758, 2019.
- [37] T. Tu, Y.-C. Lee, C.-P. Chang, and P. S. Huang, "Adjustable intensity-hue-saturation and Brovey transform fusion technique for IKONOS/QuickBird imagery," *Optical Engineering*, vol. 44, no. 11, pp. 1 – 10, 2005.
- [38] T.-M. Tu, S.-C. Su, H.-C. Shyu, and P. S. Huang, "A new look at IHS-like image fusion methods," *Information Fusion*, vol. 2, no. 3, pp. 177–186, 2001.
- [39] J. Nunez, X. Otazu, O. Fors, A. Prades, V. Pala, and R. Arbiol, "Multiresolution-based image fusion with additive wavelet decomposition," *IEEE Trans Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1204–1211, 1999.
- [40] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé, "A variational model for P+XS image fusion," *Int. J. Comput. Vis.*, vol. 69, no. 1, pp. 43–58, 2006.
- [41] X. Otazu, M. González-Audicana, O. Fors, and J. Núñez, "Introduction of sensor spectral response into image fusion methods. application to wavelet-based methods," *IEEE Trans Geosci. Remote Sens.*, vol. 43, no. 10, pp. 2376–2385, 2005.
- [42] J. Liu and S. Liang, "Pan-sharpening using a guided filter," *International Journal of Remote Sensing*, vol. 37, no. 8, pp. 1777–1800, 2016.
- [43] L. Wald, "Quality of high resolution synthesised images: Is there a simple criterion?," in *Proc. Int. Conf. Fusion Earth Data*, pp. 99–103, Jan 2000.