

Single image rain and snow removal via guided L0 smoothing filter

Xinghao Ding¹ · Liqin Chen¹ · Xianhui Zheng¹ · Yue Huang¹ · Delu Zeng¹

Received: 14 September 2014 / Revised: 2 April 2015 / Accepted: 23 April 2015 /
Published online: 11 June 2015
© Springer Science+Business Media New York 2015

Abstract Since no temporal information can be exploited, rain and snow removal from single image is a challenging problem. In this paper, an improved rain and snow removal method from single image is proposed by designing a guided L0 smoothing filter. The designed filter is inspired by the previous L0 gradient minimization. Then a coarse rain-free or snow-free image can be obtained with the proposed filter, and the final refined result is recovered by a further minimization operation depending on the observed image. Experimental results show that the proposed algorithm generates better or comparable outputs than the state-of-the-art algorithms in rain and snow removal task for single image.

Keywords Single image rain and snow removal · Guided filter · L0 gradient minimization · Guided L0 smoothing filter

1 Introduction

Outdoor vision system has been applied to a variety of purposes such as navigation, surveillance and so on. However, bad weather conditions usually dissatisfy human seeing, bring difficulties to image processing and decrease the performance of vision algorithms such as feature detection, stereo correspondence, tracking, segmentation, object recognition and so on. Based on the type of visual effects, bad weather conditions can be divided into steady (haze, fog and mist) or dynamic (rain, snow, hail) [8]. Rain or snow is the common bad weather for reducing visibility. An image captured in falling rain or snow is covered with lots of bright streaks or white spots to confuse human vision.

✉ Delu Zeng
dltsang@xmu.edu.cn

¹ Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen, China

The existing methods to remove rain or snow could be classified into two cases: one is for videos, the other is for single images. In the case of videos, Gary and Nayar developed a correlation model capturing the dynamics of rain and a physics-based motion blur model stating the photometry of rain [8, 9]. Zhang et al. presented a detection method combining temporal with chromatic properties of rain [17]. Barnum et al. modeled the rain or snow streaks using a blurred Gaussian model, rain and snow can be detected and removed based on the statistical information in frequency space with different frames [1, 2]. Bossu et al. proposed a rain removal via foreground separation, selection rules and detecting by HOS [3]. These methods well remove rain or snow in videos. However, aforementioned methods are not suitable to the case of single image since there is no reference temporal information. As a result, the tasks become more challenging in the case of single image. For single-image-based approach, Kang et al. proposed a rain removal method by image decomposition, in which the rain components of single image could be removed via performing dictionary learning and sparse coding [12]. However, the method cannot remove the non-orientation snowflakes. Xu et al. proposed a method using guided filter to remove rain or snow [14]. Zheng et al. proposed a new method to remove rain or snow through using guided filter to get the smooth image [16]. Yi-Lei Chen and Chiou-Ting Hsu proposed a novel low-rank appearance model for removing rain streaks [5]. Duan-Yu Chen et al. proposed a visual depth guided color image rain streaks removal method using sparse coding [6]. After removing rain streaks and snowflakes, objects in the image can be seen clearly.

On the other hand, the rain streaks or snowflakes in single image can be considered as the image noise. Some common denoising methods are available for rain or snow removal. For example, the bilateral filter [13] is an edge-preserving smoothing filter which considers the influence of distant pixel and their variance. The non-local means algorithm [4] is a popular image denoising method which is based on a non-local averaging of all pixels in an image. The guided filter, proposed in [10] and [11], transfers the structures of guidance image to the filtering output, and this filter has the edge-preserving smoothing property. Even these denoising methods can directly apply for single image rain or snow removal, the effects are poor and the results are always artificially blurred. Whatsoever, for many single image rain removal methods, these denoising methods can be a good pre-processing, such as [6, 12, 14, 16] and the proposed method.

In this work, a guided L0 smoothing filter is designed based on the L0 gradient minimization [15]. Firstly, a coarse but nearly rain-free or snow-free guidance image is obtained by adopting the traditional guided filter [10, 11]. Then the proposed guided L0 smoothing filter is designed to remove rain or snow. The structure of the paper is given as follows: the background information is introduced in Section 2, followed by the design of the guided L0 smoothing filter in Section 3. In Section 4 the method of removing the rain or snow is detailed. And the experimental results by comparing with other methods and discussions of the limitation are shown in Section 5. The last section is the conclusion.

2 Background

2.1 The physical property of rain or snow

Generally, the size of raindrop is 0.1–3.5mm. Since raindrop usually fall quickly, they are imaged as rain streaks in the image. And only a few pixels are occupied with rain streaks, and snowflakes are similar to rain streaks.

There are three observations on common rain streaks or snowflakes: Firstly, due to the little size and fast speed, raindrop or snowflake is imaged in form of little streaks or little spots with normal exposure of the camera. Secondly, the light of raindrop can be refracted, and the snowflake is white and bright in nature, which means the rain streaks or snowflakes are brighter than the background. Thirdly, the size of rain streaks or snowflakes are small, and they are sparse. The image taken from real-world scenes is usually piecewise smooth. So the degraded background can be restored using the value around the rain streaks or snowflakes.

2.2 The mathematical model of rain or snow image

The following model is commonly used to describe the formation of rain or snow image:

$$I_{in} = (1 - \alpha)I_b + \alpha I_r \quad , \quad (0 \leq \alpha \leq 1), \tag{1}$$

where I_{in} is the input image, I_b is the clean background image, I_r is the rain or snow component and α is the scale parameter.

When $\alpha = 0$, $I_{in} = I_b$, the observed image is not affected by rain or snow. In rain or snow removal task, we don't need to deal with this situation. When $\alpha \neq 0$, $I_{in} = (1 - \alpha)I_b + \alpha I_r$, we have $I_{in} > I_b$. The main goal of rain or snow removal is to recover the background I_b from the observed image I_{in} . This is an ill-posed problem, and some additional prior information is needed to recover the background.

2.3 The edge characteristic of rain or snow

Based on the relation between the pixels of edges and their surrounding pixels, all edges can be classified into three categories (Fig. 1): step edges, ridge edges and valley edges.

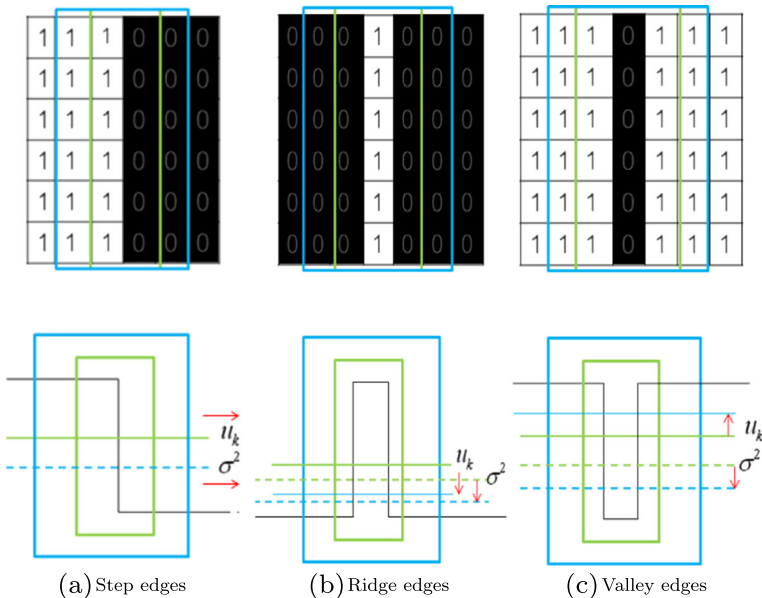


Fig. 1 Intensity profiles of three types of edges along the horizontal

In Fig. 1, let each point of the edge be the center of the corresponding window (the size of window is $k * k$; $k = p, p + 1 \dots$; p is the width of the edge). We will calculate and compare the means and variances in the corresponding windows in order to distinguish three kinds of edges. The mean and variance for step edges almost remains unchanged even if the size of window becomes larger. But for ridge edges and valley edges, their means and variances will change as the window size grows. Concretely, when the size of the window grows for ridge edges, the means become smaller and close to their adjacent pixels, and the variances also become smaller and close to zero; while for valley edges the means become larger and close to the adjacent pixels, and the variances also become smaller and close to zero.

Comparing with the surrounding pixels, the size of rain streaks or snowflakes is smaller and their values are higher than surrounding pixels. So the ridge edges with limited size are considered as rain streaks or snowflakes, and most of the other edges in the background are step edges or valley edges. Therefore, we can distinguish the rain streaks or snowflakes from the background or background defined by texture in this way.

3 The rain-free or snow-free guidance Image

3.1 Guided filter

Guided filter is an edge-preserving smoothing filter, which works well near the edges [10, 11]. Guidance image can be the observed image itself or another reference image. Besides, the guided filter is a fast and nonapproximate linear time algorithm, whose computational complexity is independent of the filtering kernel size. Particularly, guided filter formulates output image I_{out} as follows:

$$I_{out} = \frac{1}{|w|} \sum_{k|i \in w_k} (a_k I_i + b_k), \tag{2}$$

where w_k is a window centered at the pixel k , $|w|$ is the number of pixels in the window, I is guidance image, a_k and b_k are defined as:

$$a_k = \frac{\frac{1}{|w|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \varepsilon},$$

$$b_k = \bar{p}_k - a_k \mu_k \tag{3}$$

where μ_k and σ_k^2 are the mean and variance of I in ω_k , p is the input image, \bar{p}_k is the mean of p in w_k , ε is a regularization parameter to controls the structural similarity. The larger ε is, the smoother the output will be.

3.2 The low frequency part

When the input image is treated as the guidance image ($p = I$), $a_k = \frac{\sigma_k^2}{\sigma_k^2 + \varepsilon}$ and $b_k = (1 - a_k)\mu_k = \frac{\varepsilon}{\sigma_k^2 + \varepsilon} \mu_k$, and the output image using guided filter is formulated by

$$I_{out} = \frac{1}{|w|} \sum_{i \in w_k} \left(\frac{\sigma_k^2}{\sigma_k^2 + \varepsilon} I_i + \frac{\varepsilon}{\sigma_k^2 + \varepsilon} \mu_k \right). \tag{4}$$

The low frequency part of the above mentioned three kinds of edges after using guided filter will change differently. As shown in Fig. 2, we have the following analysis:

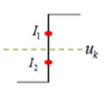
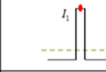
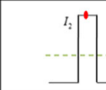
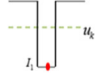
Edges		Pixel	Compare with mean	Output after guided filter
Step edges		I_1	$I_1 > \mu_k$	$\mu_k < I_{out} < I_1$
		I_2	$I_2 < \mu_k$	$I_2 < I_{out} < \mu_k$
Ridge edges	Small size 	I_1	$I_1 > \mu_k$	$I_{out} \rightarrow \mu_k$
	Large size 	I_2	$I_2 > \mu_k$	$\mu_k < I_{out} < I_2$
Valley edges		I_1	$I_1 < \mu_k$	$I_1 < I_{out} < \mu_k$

Fig. 2 Intensity change of three types of edges after using guided filter along the horizontal

1) The low frequency part of step edges after using guided filter is still step edges, but their ranges become smaller, which means that the step edges become smoother after using guided filter (as the step edges in Fig. 2).

2) If the ridge edges with small size are unaffected by the other edges, their variances are close to 0, then the ridge edges will disappear and tend to the background. While I_i belongs to the ridge edges with large size, it is hard to make I_{out} be equal to background, so the low frequency part of ridge edges with large sizes will remain and become smoothed (as the ridge edges in Fig. 2).

3) the value of low frequency part for valley edges will become larger than the input (as the valley edges in Fig. 2).

So based on the above analysis, we can conclude that the low frequency part using guided filter can be distinguished among the three-types edges, i.e., the step edges will be retained, but become a little smooth (Fig. 3a); the ridge edges with small size will disappear

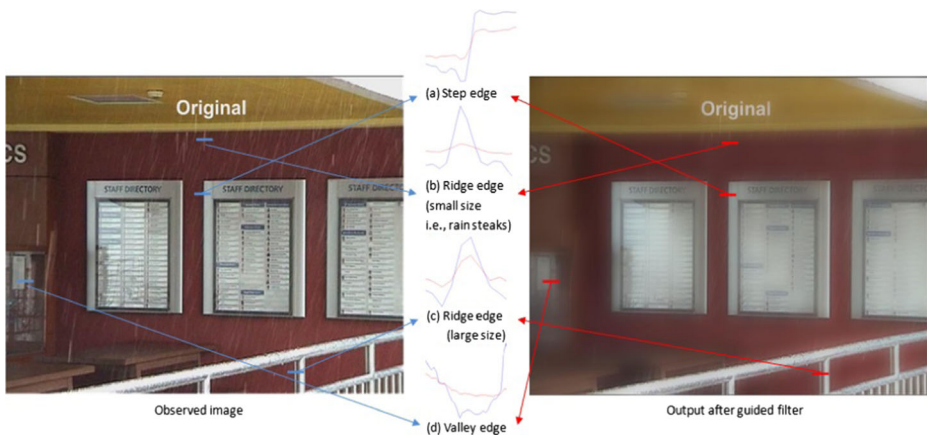


Fig. 3 Intensity change of three types of edges after using guided filter along the horizontal. (Blue line is pixel values of observed image, the red line is the corresponding low frequency part via guided filter)

(Fig. 3b), while the ridge edges with large size will be retained (Fig. 3c); the pixel values of valley edges will become higher in the low frequency part (Fig. 3d).

However, we cannot directly remove rain or snow by this traditional guided filter. In fact, if we regard input image as the guidance image, we can only get a rough result. Because the edges in the output are modified by the input image to approach to the edges in guidance image and it is not the original edges in the input image anymore. The guided filter does not work well in this case.

Nonetheless, guided filter can be a good pre-processing of rain removal method. The low frequency part is an image only keeping the information of contour of background and without remaining the edges caused by rain or snow. It can be regarded as a guidance image in the proposed method. Using both the guidance image and the original observed image, rain or snow can be removed by the proposed guided L0 smoothing filter.

4 Guided L0 smoothing filter

In this section we will firstly briefly review the theory of L0 gradient minimization [15] in brief, and simply explain the corresponding solver. Then based on the model of L0 gradient minimization, the guided L0 smoothing filter is proposed. After that, we present its application to remove rain or snow.

4.1 L0 gradient minimization

By L0 gradient minimization, they can directly optimize the L0 norm to get a piecewise constant output image [15]. It is effective to sharpen prominent edges by increasing the steepness of transition while eliminating low-amplitude structures. In this literature, the following minimization problem (5) is given:

$$\min_f \|f - f^*\|_2^2 + \lambda \|\nabla f\|_0, \quad (5)$$

where f is output, ∇f is the gradients of f , f^* is the observed image, and λ is a weight to control the level of detail. The former term is for fidelity, the later is to constraint the sparsity for the gradient magnitude of the output.

To solve the objective function, auxiliary variable δ is introduced to deal with ∇f , so (5) is written the following minimization:

$$\min_{f, \delta} \|f - f^*\|_2^2 + \beta \|\delta - \nabla f\|_2^2 + \lambda \|\delta\|_0, \quad (6)$$

where β controls the similarity between variables δ and ∇f , and the smoothing level is controlled by λ , δ is a vector with two components: δ_x and δ_y . Then equation (6) is solved through alternatively minimizing [15] δ and f .

4.2 Guided L0 smoothing filter

Recently Xu et al. proposed L0 gradient minimization to sharpen and keep prominent edges, meanwhile smooth low-amplitude structures in line with the observed image [15]. While in the given case (rain or snow removal), we do not intend to keep all the prominent edges

except ones with low gradients, so the pure L0 gradient minimization method will fail in the task of rain or snow removal.

Consequently, in this paper a guided L0 smoothing filter is proposed. It takes advantage of the property of guided filter with L0 gradient minimization. Unlike the original L0 gradient minimization, the edges of enhanced result can be preserved or smoothed according to the rain-free/snow-free guidance image. Specifically, the observed image's edges can be reserved if the corresponding locations of the guidance image is of large gradient magnitudes, and they will be smoothed if the corresponding locations are of low gradient magnitudes.

Figure 4 shows we can only obtain the sparsity via the final iteration of L0 gradient minimization (i.e., H_{last}), but can not obtain the gradient magnitude. β can solve this problem in the iteration process of L0 gradient minimization. However, the gradient magnitude of guidance image cannot be obtained. In this paper, the level of guidance image's gradient can be perceived in the output image by solving the following iteration

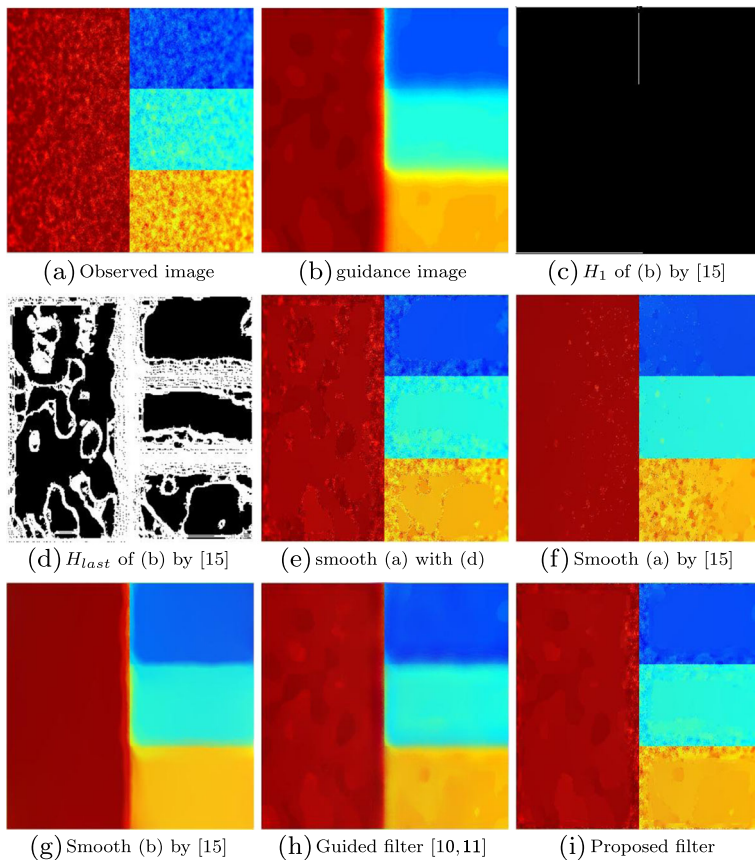


Fig. 4 Comparisons of L0 gradient minimization [15], guided filter [10, 11] and proposed guided L0 smoothing filter. **a** observed image; **b** guidance image; **c** the first iteration corresponding to **(b)** by [15] ($\lambda = 0.0005$); **d** the final iteration corresponding to **(b)** by [15] ($\lambda = 0.0005$); **e** smooth **(a)** with **(d)**; **f** smooth **(a)** by [15]; **g** smooth **(b)** by [15]; **h** smooth **(a)** with guided filter [10, 11]; **i** smooth **(a)** with proposed filter

process. In this way, the prominent edges of the observed image can be protected. Therefore, we call the proposed algorithm to be guided L0 smoothing filter.

Firstly, when f^k is known, we optimize the δ^k as (7), where λ must be small to keep all gradients information

$$\min_{\delta^k} \beta^k \|\nabla f^k - \delta^k\|_2^2 + \lambda \|\delta^k\|_0 \quad (k = 1, 2, 3 \dots). \tag{7}$$

We can solve (7) as follows [15]:

$$\delta^k = \begin{cases} 0 & \nabla f^k \leq \frac{\lambda}{\beta} \\ \nabla f^k & \text{others} \end{cases} \tag{8}$$

Secondly, when f^k and δ^k are known, we solve f^{k+1} by:

$$\min_{f^{k+1}} \|f^{k+1} - f^*\|_2^2 + \beta^k \|\delta^k - \nabla f^{k+1}\|_2^2. \tag{9}$$

Then the expression (9) is equivalent to (10):

$$\begin{cases} \min_{f^{k+1}} \|f^{k+1} - f^*\|_2^2 + \beta^k \|\nabla f^{k+1} - H \cdot \nabla f^k\|_2^2 \\ H = \begin{cases} 0 & \delta^k = 0 \\ 1 & \delta^k \neq 0 \end{cases} \quad (k = 1, 2, 3 \dots) \end{cases} . \tag{10}$$

The objective function is quadratic and thus is a convex optimization problem. So we can use the least square method and Fourier transformation to solve it. Then the solution of (9) or (10) is:

$$f^{k+1} = \text{ifft} \left(\frac{\text{fft}(f^*) + \beta(\text{fft}(\partial_x^T) \text{fft}(\delta_x^k) + \text{fft}(\partial_y^T) \text{fft}(\delta_y^k))}{\text{fft}(1) + \beta(\text{fft}(\partial_x^T) \text{fft}(\partial_x) + \text{fft}(\partial_y^T) \text{fft}(\partial_y))} \right), \tag{11}$$

where fft is the fast Fourier transform operator and ifft is its inverse. ∂_x and ∂_y denote difference operators in the horizontal and vertical directions, respectively.

Finally, when s^k and δ^k are known, we solve s^{k+1} from (12)

$$\begin{cases} \min_{s^{k+1}} \|s^{k+1} - s^*\|_2^2 + \beta^k \|\nabla s^{k+1} - H \cdot \nabla s^k\|_2^2 \\ H = \begin{cases} 0 & \delta^k = 0 \\ 1 & \delta^k \neq 0 \end{cases} \quad (k = 1, 2, 3 \dots) \end{cases} . \tag{12}$$

The solution of (12) is:

$$s^{k+1} = \text{ifft} \left(\frac{\text{fft}(s^*) + \beta(\text{fft}(\partial_x^T) \text{fft}(H \cdot \nabla s_x^k) + \text{fft}(\partial_y^T) \text{fft}(H \cdot \nabla s_y^k))}{\text{fft}(1) + \beta(\text{fft}(\partial_x^T) \text{fft}(\partial_x) + \text{fft}(\partial_y^T) \text{fft}(\partial_y))} \right). \tag{13}$$

The above entire procedure is summarized in Algorithm 1. In the proposed method, most of the parameter settings are same to L0 gradient minimization's, i.e., $\beta_0, \beta_{\max}, \kappa$ are fixed as $2\lambda, 10^5, 2$, respectively. λ is set to 0.0005. Figure 4 shows the comparison among L0 gradient minimization, guided filter and the proposed guided L0 smoothing filter. Figure 4i

is the result of proposed filter, its prominent edges are closer to the observed image than the guided filter’s result [10, 11], and it is as smoothed as the result of guided filter.

Algorithm 1 :

Input: observed image s^* , guidance image f^* , parameters $\lambda, \beta_0, \beta_{\max}$, rate κ

Initialization: $f^1 \leftarrow f^*, s^1 \leftarrow s^*, \beta^1 \leftarrow \beta_0, k \leftarrow 1$

repeat:

with f^k , solve for δ^k in (8);

with f^k and δ^k , solve for f^{k+1} in (11);

with s^k and δ^k , solve for s^{k+1} in (13);

$\beta \leftarrow \kappa\beta, k++$;

Until $\beta \geq \beta_{\max}$

Output: The result of guided L0 smoothing filter s .

5 The frame of the proposed method

Figure 5 explicitly shows the framework of the proposed method. Firstly, low frequency part of the observed image will be obtained by the traditional guided filter, i.e., we obtain a blurred guidance image without rain or snow. Because the rain streaks are small and sparse to hardly cover the background, and observed image has real background information, the proposed guided L0 smoothing filter is used to restore the rain-free/snow-free result. However, the valley edges do not need to be restored, because they don’t belong to rain streaks or snowflakes. So at last, the minimization operation was taken between the observed image and the rain-free or snow-free result to get the final refined result.

6 Experimental results

Since we know that the rain streaks almost don’t appear in the horizontal direction and their intensities are higher than their adjacent pixels in horizontal direction. Hence, all

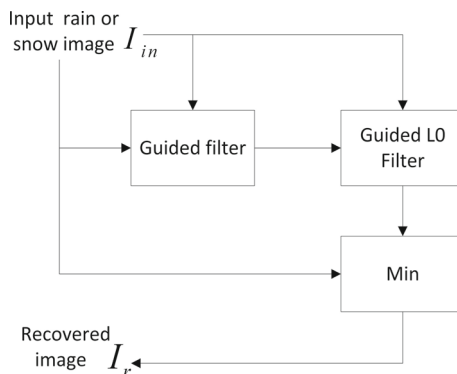


Fig. 5 Block diagram of the proposed removal method

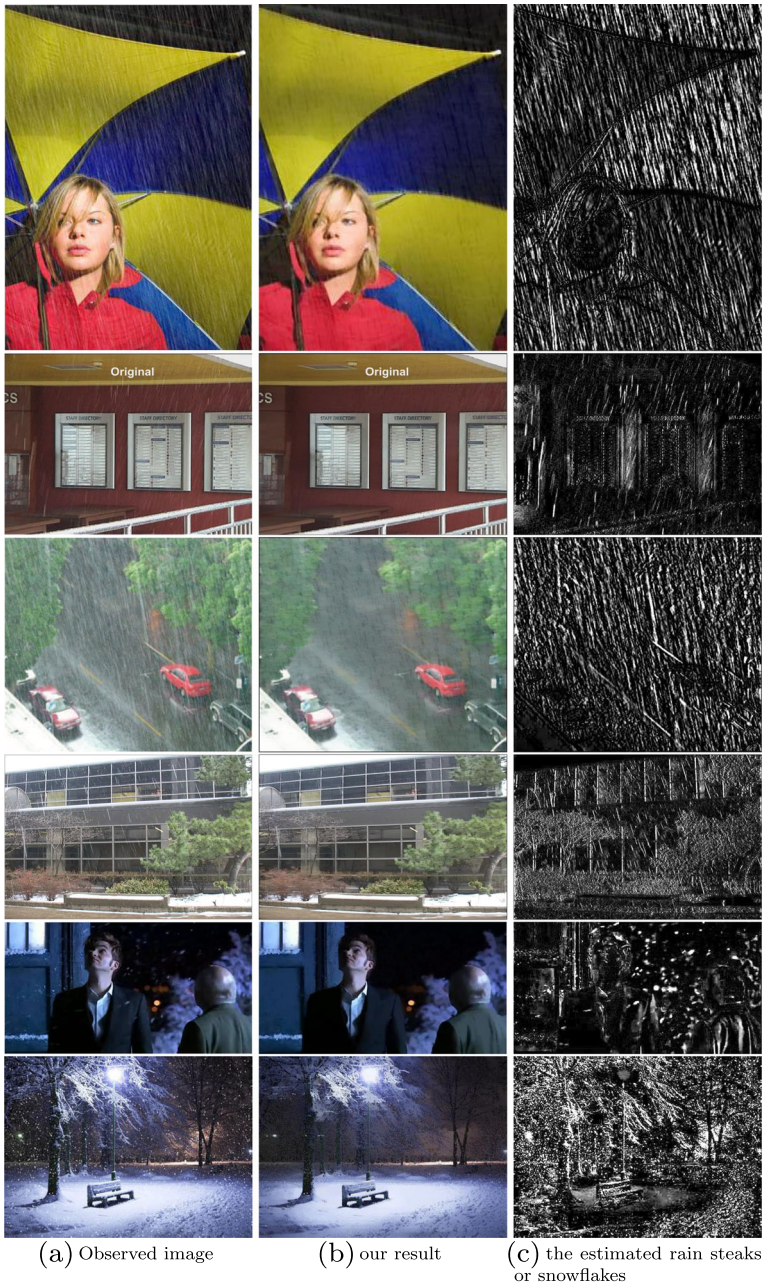


Fig. 6 The results of proposed method and the estimated rain streaks or snowflakes

of the rain streaks removal experiments should be carried out in the horizontal direction. But for snowflakes removal experiments, they are performed both in horizontal and vertical directions.

6.1 The results of proposed rain and snow removal methods

The most commonly used data is applied to test our proposed method. In the Fig. 6, the left column images are the observed image, the middle images are the restored background by the proposed method, and the right column images are the estimated rain streaks or snowflakes. the estimated rain streaks or snowflakes image can be obtained by subtracting our result with the observed image. Even some details will lose such as little ridge edges or some pixels of edges, rain or snow can be estimated with low error rate through the proposed method. So our method can improve the visual quality of rain image or snow image to get an almost clear and accurate background.

6.2 Comparisons

In Figure 7 we compare our rain removal result with the methods of Xu et al. [14] and the coauthors's previous work in [16]. It shows that the proposed method yields better visual quality against other methods for comparisons. Particularly, the proposed method does not create blurred artifacts, and the restored image is closer to the background in the observed image. For example, the result of Fig. 7b and c are not clear, but the proposed result is clearer.

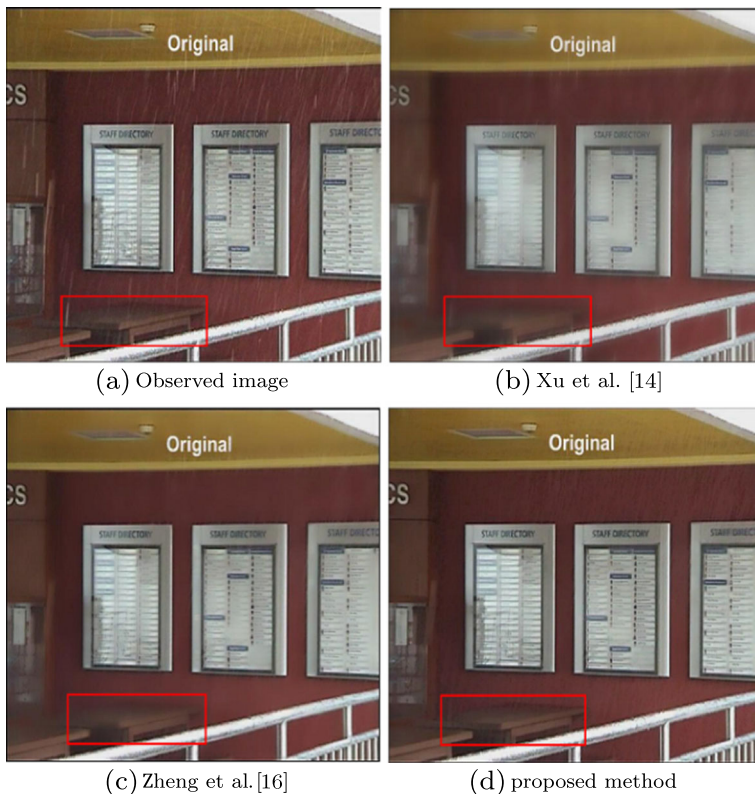


Fig. 7 Comparison of rain removal results with other methods

Figure 8 shows the comparisons of snow removal results obtained by previous methods [14, 16] with the proposed method. Clearly, the proposed method achieves much more accurate result of snowflake removal than the previous methods.

In Fig. 9, the comparison of snow removal results is shown among the common denoising methods and the state-of-the-art rain removal methods. Seen from the results, the proposed method works better than the common denoising methods (Bilateral filter [13], K-SVD [7], Nonlocal filter [4], guided filter [10, 11]), especially achieving a greater improvement for the effect of rain removal or the restored background (for example in the zoom of red box). Compared with the state-of-the-art image rain removal methods, although the proposed method cannot remove all the rain streaks like MCA method [12] and method [10, 11], the corresponding result is clearer than the comparison methods [10–12]. Moreover, although the methods [12, 14] enhance the details of background, the naturalness is preserved better and the value is closer to the observed image in proposed method (for example in the zoom in of red box).

6.3 Limitations

The proposed method is based on the fact that the region value for rain or snow is smaller and brighter than background. As a result, there are some limitations. Figure 6 (third row) shows the case that rain streaks are too dense and cover most parts of the background, so we can not find the background in any patches, and the rain streaks can not be completely removed. In Fig. 6 (fourth row), the rain streaks are too bright, the window frames and the branches are smaller and brighter than the adjacent pixels as well, the proposed method

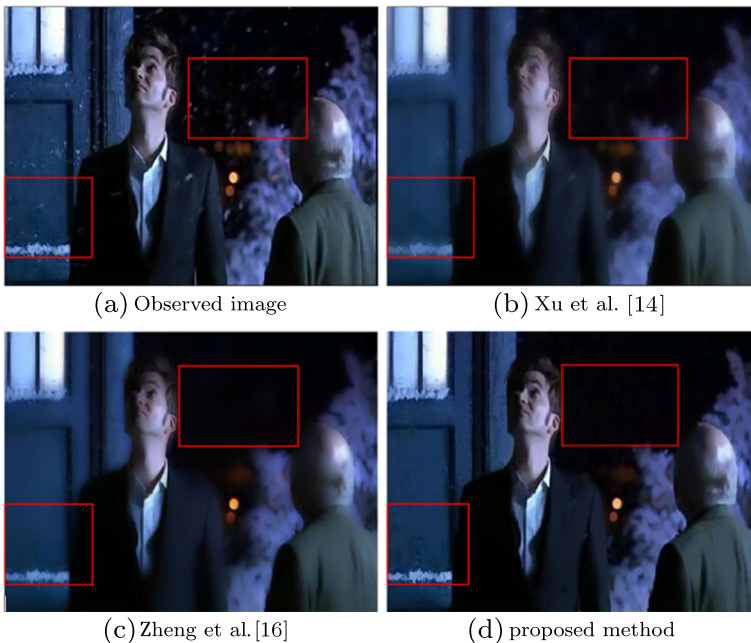


Fig. 8 Comparisons of rain removal results with other methods

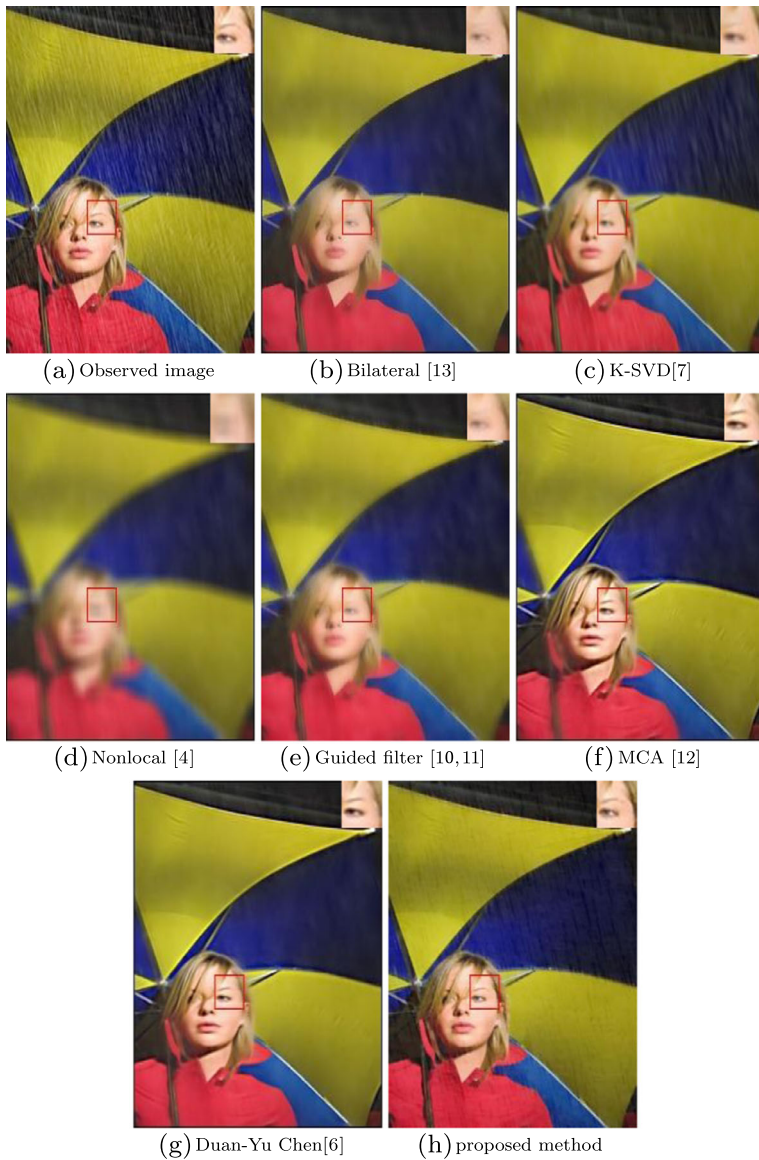


Fig. 9 Comparison of rain removal results with other methods

can not remove all rain streaks without losing background information. However, these problems are still challenging to be worthy of further researching for this filed.

7 Conclusions

In this work, an improved rain and snow removal method is proposed. A designed guided L0 smoothing filter is designed in the method. It is based on the fact that small rain streaks

or snowflakes are brighter than the adjacent pixels. After using the guided filter, the low frequency parts of three kinds of edges can be distinguished by the means and variances in the window [16]. So we take the observed image and the low frequency part as the inputs of the designed filter to acquire a rough result. Then further by the minimization operation between the rough result and the observed image to achieve the final refined result. The results show that the proposed approach is more effective in rain and snow removal task than the recent methods.

Acknowledgments The project is supported by the National Natural Science Foundation of China (No. 30900328, 61172179, 61103121, 81301278), the Natural Science Foundation of Fujian Province of China (No. 2012J05160), The National Key Technology R&D Program (2012BAI07B06), the Fundamental Research Funds for the Central Universities (No. 2011121051, 2013121023), the Research Fund for the Doctoral Program of Higher Education under Grant 20120121120043.

References

1. Barnum P, Kanade T, Narasimhan SG (2007) Spatio-temporal frequency analysis for removing rain and snow from videos. In: Proceedings of the 1st international workshop on photometric analysis for computer vision-PACV, p 8 p
2. Barnum PC, Narasimhan S, Kanade T (2010) Analysis of rain and snow in frequency space. *Int J Comput Vis* 86(2–3):256–274
3. Bossu J, Hautire N, Tarel JP (2011) Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *Int J Comput Vis* 93(3):348–367
4. Buades A, Coll B, Morel JM (2008) Nonlocal image and movie denoising. *Int J Comput Vis* 76(2):123–139
5. Chen YL, Hsu CT (2013) A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: Proceedings of 2013 IEEE international conference on computer vision, pp 1968–1975
6. Chen DY, Chen CC, Kang LW (2014) Visual depth guided color image rain streaks removal using sparse coding. *IEEE Trans Circ Syst Video Technol* 24(8):1430–1455
7. Elad M, Aharon M (2006) Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans Image Process* 15(12):3736–3745
8. Garg K, Nayar SK (2007) Vision and rain. *Int J Comput Vis* 75(1):3–27
9. Garg K, Nayar SK (2004) Detection and removal of rain from videos. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, vol 1, pp 1-528–I-535
10. He K, Sun J, Tang X (2010) Guided image filtering. In: Proceedings of European Conf Comput Vis, pp 1–14
11. He K, Sun J, Tang X (2013) Guided image filtering. *IEEE Trans Pattern Anal Mach Intell* 35(6):1397–1409
12. Kang LW, Lin CW, Fu YH (2012) Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans Image Process* 21(4):1742–1755
13. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Proceedings of 6th international conference on computer vision, pp 839–846
14. Xu J, Zhao W, Liu P, Tang X (2012) An improved guidance image based method to remove rain and snow in a single image. *Comput Inf Sci* 5(3):49
15. Xu L, Lu C, Xu Y, Jia J (2011) Image smoothing via L 0 gradient minimization. *Proc ACM Trans Graph* 30(6):174
16. Zheng X, Liao Y, Guo W, Fu X, Ding X (2013) Single-image-based rain and snow removal using multi-guided filter. In: Proceedings of neural information processing, pp 258–265
17. Zhang X, Li H, Qi Y, Leow WK, Ng TK (2006) Rain removal in video by combining temporal and chromatic properties. In: Proceedings of 2006 IEEE international conference on multimedia and expo, pp 461–464



Xinghao Ding received the B.S. and Ph.D. degrees from Department of Precision Instruments, Hefei University of Technology, Hefei, China, in 1998 and 2003, respectively. Since 2006, he has been a Professor with the Xiamen University, Xiamen, China. His main research interests include image processing, sparse signal representation, and machine learning.



Liqin Chen received the B.S. degree from Department of Electrical Engineering, Ocean University of China, Qingdao, China. Currently she is an undergraduate student in the Department of Communication Engineering, Xiamen University, Xiamen, China. Her research interests include sparse signal representation, image processing.



Xianhui Zheng received the B.S. degree from electronic and information engineering, Huaqiao University, Xiamen, China, and M.S. degree from Department of Electrical Engineering, Xiamen University, Xiamen, China. His main research interests include sparse signal representation, image processing.



Yue Huang received the B.S. from Department of Electrical Engineering, Xiamen University, and Ph.D. degrees from Department of Biomedical Engineering, Tsinghua University, Beijing, China, in 2005 and 2010, respectively. Since 2010, she has been an assistant professor with the Xiamen U University, Xiamen, China. Her main research interests include sparse signal representation, and machine learning.



Delu Zeng received the B.S. and M.S. degrees in applied mathematics and the Ph.D. degree in electronic and information engineering from South China University of Technology, Guangzhou, China, in 2003, 2005, and 2009, respectively. He is now an associate professor in the school of Information Science and Technology in Xiamen University, China. His research interests include partial differential equations, machine learning, and their applications in image/video processing, i.e., image segmentation and object tracking.