Vis Comput (2012) 28:713–721
DOI 10.1007/s00371-012-0679-y

ORIGINAL ARTICLE

Fast image dehazing using guided joint bilateral filter

Chunxia Xiao · Jiajia Gan

Published online: 25 April 2012

Abstract In this paper, we propose a new fast dehazing method from single image based on filtering. The basic idea is to compute an accurate atmosphere veil that is not only smoother, but also respect with depth information of the underlying image. We firstly obtain an initial atmosphere scattering light through median filtering, then refine it by guided joint bilateral filtering to generate a new atmosphere veil which removes the abundant texture information and recovers the depth edge information. Finally, we solve the scene radiance using the atmosphere attenuation model. Compared with exiting state of the art dehazing methods, our method could get a better dehazing effect at distant scene and places where depth changes abruptly. Our method is fast with linear complexity in the number of pixels of the input image; furthermore, as our method can be performed in parallel, thus it can be further accelerated using GPU, which makes our method applicable for real-time requirement.

Keywords Image dehazing · Filtering · Image processing · Bilateral filter

1 Introduction

Haze or fog is a common natural phenomenon, which is typically caused by fine suspended particles in the air. In foggy weather, a series of reactions, such as scattering, refraction, and absorption will occur between these particles and light from the atmosphere, which makes the visibility of the scene degraded. The presence of haze will lead to substantially reduce of the visibility of the image scene, which will become a major problem in many computer vision applications, such as video surveillance, remote sensing, navigation, target identification, etc. Thus, haze removal is highly required for receiving high performance of the vision algorithm. Although many image dehazing methods have been proposed, as performing haze removal from a single image is a underconstrained problem, many difficulties are still left to solve.

The scene depth information of the degraded image is an important clue to haze removal. Many methods extract depth information from multiple images and extra information. For instance, binary scattering model is used to extract scene information from color images under different weather conditions [8, 11]. Polarization properties of different scattered light are used to restore the depth information through the polarized light in different directions [14–16]. Narasimhan and Nayar [10] applied the boundary of discontinuous depth to extract scene depth from two grayscale images under different weather conditions. Interactive depth estimation algorithm [9] or known 3D models [6] were also applied to obtain the depth. Although these methods may produce impressive results, they need to use multiple images of the same scene or need user interactions and other extra information, which makes them difficult to meet with real-time requirements of images with changing scenes.

In recent years, many researchers focus on achieving haze removal results from a single degraded image. Through statistics, Tan [17] found that clear images had higher contrast compared with foggy images, thus he maximized the local contrast of the restored image for enhancing image vis-

Electronic supplementary material The online version of this article (doi:10.1007/s00371-012-0679-y) contains supplementary material, which is available to authorized users.

C. Xiao (✉) · J. Gan
School of Computer, Wuhan University, Wuhan, China
e-mail: cxxiao@whu.edu.cn

J. Gan
e-mail: whdxgjj@whu.edu.cn

ibility. The disadvantage was that the color of the restored image was often too saturated. Based on the assumption that the propagation of light and shading parts of the target surface were locally uncorrelated, Fattal [2] first estimated the scene radiance and then derived the transmission image. As this method required sufficient color information, thus it could not process gray level images. He et al. [3] proposed a single image haze removal technology based on Dark Channel Prior (DCP) and produced impressive results, however, this method needed to refine the transmission map through soft matting technology [7], which could be computational intensive.

To improve the efficiency of image dehazing, several methods have been proposed to accelerate the dehazing processing. He et al. [4] proposed a guided filter, and found that the output of a guided filter could be an approximate solution of the Laplacian matting optimization equation [7]. This method greatly reduced the time complexity, however, as the original foggy image was chosen as the reference image, which may lead to incomplete haze removal. Tarel [18] proposed a fast dehazing algorithm using a median filter; this algorithm was very efficient, but as the median filter is not conformal and edge-preserving, although the received atmosphere veil are smooth, it does not respect with the depth information of the scene. In some small edge regions, the desirable dehazing results cannot be achieved.

In this paper, we propose a new single image haze removal algorithm based on reconstructing a more accurate atmosphere veil. Since atmosphere veil mainly depends on scene depth information, which tends to be smooth most of time, thus the desirable atmosphere veil should be smooth most of time except at some places where depth jumps abruptly. Based on this assumption, our basic idea is to apply an image containing the edge information of the input foggy image as the cue of the scene depth information, and use this image as the reference image to filter the initial atmosphere veil [18]. In our method, we compute the min color component map of original foggy image, and filter the map using bilateral filter to receive the reference image. Then we propose a guided joint bilateral filter to filter atmosphere veil [18] to receive a more accurate atmosphere veil. As we remove the undesirable texture details from the atmosphere veil, and add the edge information of the original image into the atmosphere veil, this leads to better dehazing results at places where depth changes abruptly.

For image haze removal, the time complexity is also a critical problem that needs to be addressed. In some applications, such as remote sensing image processing, video surveillance, and intelligent vehicle, high time complexity of dehazing may make the algorithm impracticable. Both the bilateral filter and the guided joint bilateral filter used in our method can be accelerated in constant time complexity. Furthermore, the proposed filters can be performed in parallel, thus, they can be computed using GPU. Compared with existing methods [2, 3, 17, 18], our method cannot only yield better dehazing quality, but also has low time complexity. For images with moderate size, our method can perform in real time, which makes our method applicable to process video streaming with moderate size.

Fig. 1 *Left*: the input image, *middle*: the dehazing result of [18], *right*: our result

2 Background

In computer vision, the formation of haze images is usually described by the atmosphere attenuation process:

$$I(x) = J(x)t(x) + A\big(1 - t(x)\big) \tag{1}$$

where $I(x)$ is the observed haze image, $J(x)$ is scene irradiance (the clear haze-free image), A is the overall atmosphere light, and $t(x)$ is the medium transmission, when the atmosphere is homogenous, $t(x) = e^{-\beta d(x)}$. Here, β is the scattering coefficient of the atmosphere, and d is the scene depth. The goal of haze removal is to recover $J(x)$, A and $t(x)$ from $I(x)$. The term $J(x)t(x)$ is the direct attenuation, which indicates that haze will induce the scene radiance to attenuate exponentially with the scene depth d in the medium. The term $A(1 - t(x))$ is the atmospheric veil (atmospheric scattering light), which causes fuzzy, color shift, and distortion in the scene.

Tan [17] enhanced image visibility by maximizing the contrast of the resulting image, and reformulated the problem as maximizing atmosphere veil $V(x) = A(1 - t(x))$ assuming that $V(x)$ is smooth most of the time, except along edges with large depth jumps. As the optimization function was computationally intensive, Tarel et al. [18] proposed a fast visibility restoration algorithm by using a filtering approach to compute atmosphere veil $V(x)$. They supposed that a desired $V(x)$ should meet the following two constraints: (1) the value $V(x)$ is positive ($V(x) \geq 0$) at each pixel; (2) the value of $V(x)$ is not higher than the min components of $I(x)$, $V(x) \leq W(x)$, where the notation $W(x) = min_{c\in\{r,g,b\}}(I^c(x))$ is the min color components of $I(x)$.

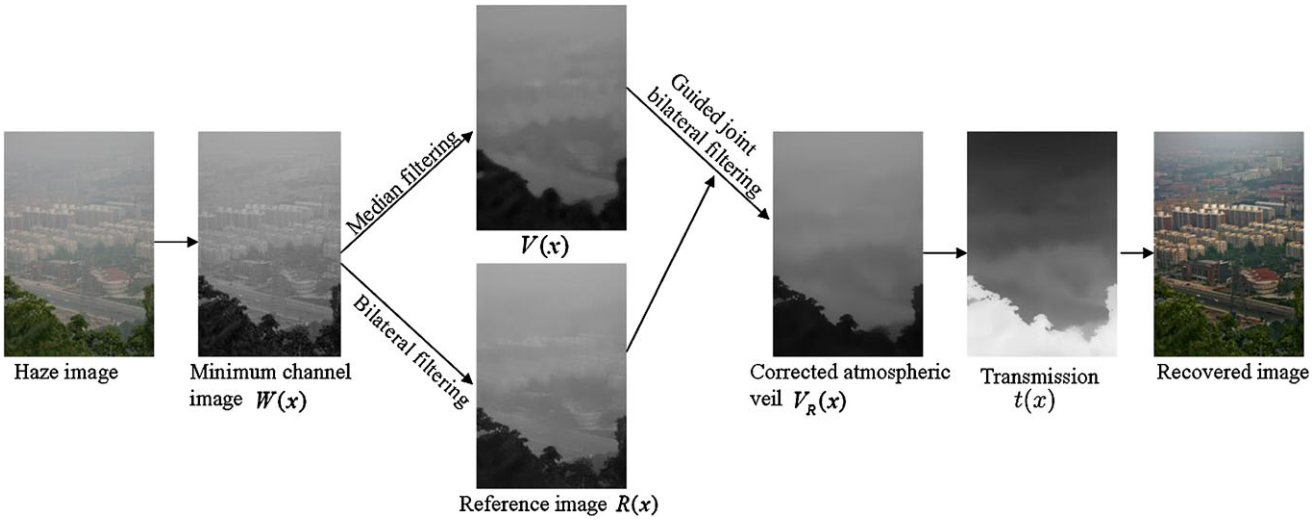

Fig. 2 The flow chart of the proposed method

With these two constraints and observations, Tarel et al. [18] used median filtering to yield desirable function $V(x)$. They first filtered the $W(x)$ using a median filter to receive $B(x)$ and to alleviate the affect of contrasted texture for the haze removal, they also applied the difference of the local mean $B(x)$ and local standard deviation of $W(x)$. Finally, they multiplied $C(x)$ by a scale factor $p \in [0, 1]$ to control the strength of visibility restoration. The atmospheric scattering light $V(x)$ was calculated as the following steps:

$$\begin{aligned} &(1)\ W(x) = min_{c\in\{r,g,b\}}(I^c(x)) \\ &(2)\ B(x) = median_{\Omega}(W(x)) \\ &(3)\ C(x) = B(x) - median_{\Omega}(|W - B|)(x) \\ &(4)\ V(x) = max(min(pC(x), W(x)), 0) \end{aligned} \tag{2}$$

where Ω is a square window of median filter.

As median filter itself is not edge-preserving and not conformal, many edge information in the resulting image $V(x)$ are lost after twice median filtering, which include the edges where the depth value changes abruptly, while large jumping in the atmosphere veil is very important for image recovery. If the edge information is missing, the algorithm cannot detect the haze at these locations, which will result in incomplete haze removal. As Fig. 1 shows, after applying the method [18] based on median filtering, there are still some fogs in the gap between the leaves.

3 Fog removal based on filtering

We observe that the variation of the atmosphere veil $V(x) = A(1-t(x))$ mainly depends on the depth d of the scene, that is, the distance of the objects to the viewer. Thus, the desired atmosphere veil should be smooth and its intensity should gradually vary according to the scene depth, and at the regions with same depth that intensity of $V(x)$ should have similar value. When we infer the $V(x)$ from the $W(x)$, there is much texture information as well as the edge information remaining in the $W(x)$. The edge information is important as they suppose possible large depth jumps among the objects, such as the edges between the leaves and the wall in Fig. 7. Thus, it is necessary to restore edge information.

We think one appropriate filtering method should keep the atmosphere veil smooth, preserve the depth jumps, and remove the redundant texture information. To reach the above goal, we try to correct the atmosphere veil $V(x)$. The purpose of this correction operator is to restore the edge information in $V(x)$, and reduce texture details in $V(x)$, as texture information does not suggest depth variation, for example, the brick texture information on the wall in the Fig. 7. We take an image containing the edge information of the source image as the reference image, and propose a guided joint bilateral filter to enhance the atmosphere veil $V(x)$ achieved using median filtering, as shown in Fig. 2.

3.1 Guided joint bilateral filter

The bilateral filter, originally introduced by [19], has been widely used in computer graphics and computer vision community [12]. The bilateral filter computes the filter output at a pixel as a weighted average of neighboring pixels and is able to preserve edges of processed images. For each pixel in the image I, let $\Omega(x)$ be the local patch centered at x, $I(x)$ and $I(y)$ be the corresponding intensity value of pixel x and y, then the filtered intensity value of x is

$$I^B(x) = \frac{\sum_{y\in\Omega(x)} f(x-y)\cdot g(I(x)-I(y))\cdot I(y)}{\sum_{y\in\Omega(x)} f(x-y)\cdot g(I(x)-I(y))} \tag{3}$$

where f and g are the spatial and range filter kernels, respectively.

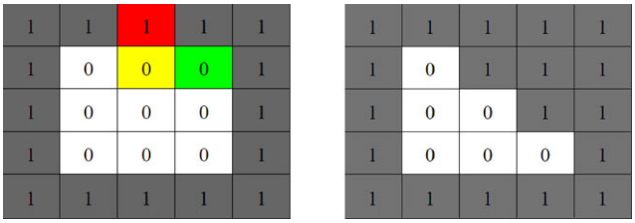

Fig. 3 Guided Joint Bilateral Filter. *Left*: image to be filtered, *right*: reference image. "0" and "1" represent the intensity value of the underlying pixel

The bilateral filter is generalized to the joint bilateral filter in [5, 13], in which the range kernel is computed based on another guidance image D, then the filtered intensity value of x is

$$I_D^B(x) = \frac{\sum_{y\in\Omega(x)} f(x-y)\cdot g(D(x)-D(y))\cdot I(y)}{\sum_{y\in\Omega(x)} f(x-y)\cdot g(D(x)-D(y))} \tag{4}$$

Thus, the filtered image I_D^B will gain the edge information of reference image D. The joint bilateral filter is particular favored when the input image is not reliable to provide edge information, e.g., when it is very noisy or is an intermediate result in image processing.

Joint bilateral filter can enforce the edge information of the filtered image to be similar to the reference image, however, since it only considers the difference between pixel x and neighboring y in reference image D, its correction capacity is still limited. As shown in Fig. 3, the left is the image to be filtered; the edge information of which is not complete and accurate. The right image contains the cue of edge details. To refine the edge feature of the left image, we take the right image as the reference image and filter the left one using joint bilateral filter. Suppose the window size is 3 pixels, considering the edge pixel marked by yellow color, whose correct value should be 1, in order to make the filter value close to 1, we need to give large weight to pixel with value 1 and give small weight to pixel with value 0 in its 3×3 local patch. For the pixel marked by red color with value 1, the responding range weight $g(D(x)-D(y)) = g(1-1) = g(0)$, which satisfies our needs, but for the pixel marked by green color with value 0, the responding range weight $g(D(x)-D(y)) = g(1-1) = g(0)$ is still very large.

Based on the above observation, in order to preserve edge and correct the imperfect edge details, we not only need to consider the difference in the reference image, but also require to further consider the difference between the image to be filtered and the reference image, give larger weights to pixels with smaller deviation and smaller weights to pixels with larger deviation. Using this filtering operation, we could correct the image toward the reference image. This filter is called guided joint bilateral filter, and the filtered intensity value of x is

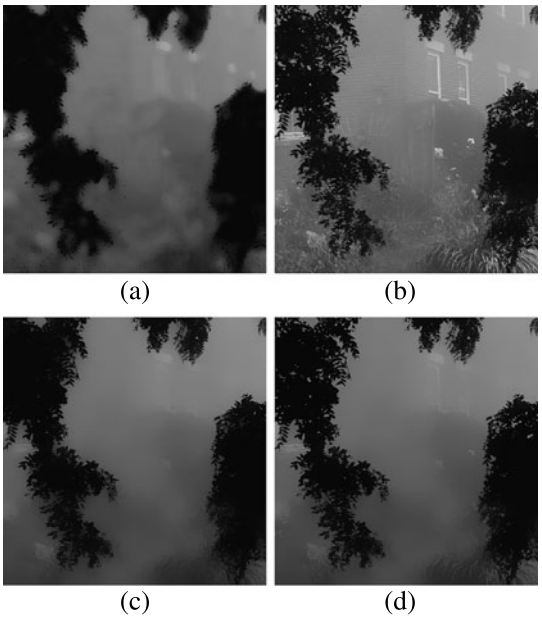

Fig. 4 Guided Joint Bilateral Filter. (**a**) Image to be filtered, (**b**) reference image, (**c**) and (**d**) joint bilateral filter results without using and using term $h(I(y)-D(y))$

$$I_D^G(x) = \frac{1}{k}\sum_{y\in\Omega(x)} f(\|y\|)\cdot g\big(\|D(y)\|\big) \cdot h\big(I(y)-D(y)\big)\cdot I(y) \tag{5}$$

where $k = \sum_{y\in\Omega(x)} f(\|y\|)\cdot g(\|D(y)\|)\cdot h(I(y)-D(y))$ is the normalizing factor, the functions $f(\|y\|) = e^{-(x-y)^2/2\sigma_s^2}$, $g(\|D(y)\|) = e^{-(D(x)-D(y))^2/2\sigma_r^2}$, and $h = e^{-x^2/2\sigma_t^2}$ is also a Gaussian kernel function. With this improvement, as shown in Fig. 3, the range weight given to pixel 0 marked by green color is $g(D(x)-D(y))\cdot h(I(y)-D(y)) = g(1-1)\cdot h(1-0) = g(0)\cdot h(1)$, which is obviously reduced. Figure 4 gives the joint bilateral filter results using and without using term $h(I(y)-D(y))$. Obviously, our method produces better results to recovery edge information.

As the atmosphere veil $V(x)$ is computed by [18], this method loses a lot of edge information. We try to add edge information to the original image of $I(x)$ to $V(x)$, meanwhile reducing texture details of $V(x)$. As the minimum luminance channel map $W(x)$ contains the edge features and texture details of the input image, we propose to use $W(x)$ as the cue to enhance the edge information of $V(x)$ around the regions with the abrupt depth jumps. We first use a bilateral filter on $W(x)$ to filter out some texture details, while the edge features can be well preserved:

$$R(x) = \frac{\sum_{y\in\Omega(x)} f(x-y)\cdot g(W(x)-W(y))\cdot W(y)}{\sum_{y\in\Omega(x)} f(x-y)\cdot g(W(x)-W(y))} \tag{6}$$

Then we take the filtered image $R(x)$ as a reference image to filter the atmosphere veil $V(x)$ by using the guided joint filter:

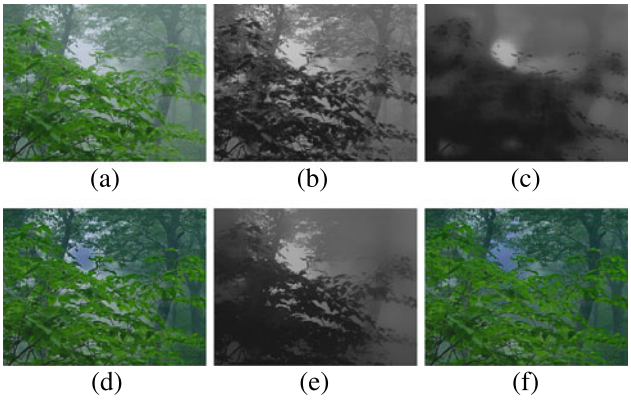

Fig. 5 (**a**) Input image I, (**b**) minimal component W of I, (**c**) atmospheric veil V computed using [18], (**d**) restored result using the V computed in (**c**), (**e**) the atmospheric veil V_R of our method, (**f**) our final dehazing result

$$V_R(x) = \frac{1}{k} \sum_{y \in \Omega(x)} f(\|y\|) \cdot g\big(\|R(y)\|\big) \cdot h\big(V(y) - R(y)\big) \cdot V(y) \tag{7}$$

where k is the normalizing factor, the range filter kernel $g(\|R(y)\|) = e^{-(R(x)-R(y))^2/2\sigma_r^2}$. Here, the term $g(R(x) - R(y))$ and term $h(V(y) - R(y))$ work together to preserve the edge information and remove the useless texture information of the image $R(x)$, which produces a more accurate atmospheric veil $V_R(x)$.

Figure 5 shows the atmospheric veil before and after guided joint bilateral filtering. It is easy to find that the filtered atmospheric veil is completely smooth as a whole. We can see the intensity of the atmospheric scattering map is gradually increasing, that is, the intensity of the close scene is lower than those of the distant scene, which respects with the depth information of the scene, and also conforms with the observation of Tan [17] that the variation of atmospheric veil mainly depends on the distance of objects to the viewer. In addition, our results also keep the necessary large jumps to restore images with jumpy depth. As shown in Fig. 5, depth of the leaves and the neighboring space is different, however, using the filtering of Tarel et al. [18], the regions containing the leaves are smooth, while our method keeps the existing jumps well, and the outline of the tiny leaves is also effectively restored. With the received smooth atmospheric veil as well as recovered depth jumps, the haze is removed completely.

3.2 Atmosphere light estimation

In many single image dehazing methods, the overall atmosphere light A is usually estimated from pixels with most dense haze, for example, taking the highest luminance value of the image as the ambient light. But the brightest pixel may be the white objects. He et al. [3] proposed that the intensity of the dark channel is a rough approximation of the thickness of the haze. Similar to [3], we use the dark channel to improve the accuracy of the atmosphere light: firstly, choosing the brightest pixels (0.2 %) in the dark channel, which correspond to the foggiest regions; then, choosing the brightest pixel among the pixels at the same position in the input foggy image as the overall atmosphere light A.

3.3 Scene radiance recovering

Given the atmosphere veil $V_R(x)$ and the overall atmosphere light A, the transmission $t(x)$ can be obtained by: $t(x) = 1 - \omega V_R(x)/A$, where the parameter $\omega \in (0, 1]$ is used to preserve little haze in the distant scene and makes the recovered image more natural. In most experiments in this paper, the value of ω is 0.95. Similar to [3], the final scene irradiance (the clear haze-free image) can be computed as $J(x) = \frac{I(x)-A}{max(t(x),t_0)} + A$, where t_0 is the lower bound of transmission, which is used to avoid noises in very dense haze regions.

3.4 Algorithm process flow

The main process of this algorithm is as follows:

(1) Calculate the min components $W(x)$ of $I(x)$ and estimate the atmosphere light A;
(2) Yielding initial atmospheric veil $V(x)$ using median filtering (Eq. (2));
(3) Filtering $W(x)$ using bilateral filter to get the reference image $R(x)$ (Eq. (6));
(4) Filtering $V(x)$ taking $R(x)$ as the reference image to receive corrected atmospheric scattering light $V_R(X)$ (Eq. (7));
(5) Calculate the transmission $t(x)$;
(6) Get the recovered image $J(x)$.

Figure 2 gives the flow chart of this method. It can be seen that the time-consuming parts of the whole algorithm are the median filter in step 2, bilateral filtering in step 3, and the guided joint bilateral filtering in step 4; other steps can be completed in $O(1)$. In the next section, we will present methods to accelerate the steps 2, 3, and 4.

4 Dehazing acceleration

To receive real time image dehazing results, we can accelerate the three filters used in this paper, the bilateral filter, the guided joint bilateral filter, and the median filter. Yang et al. [24] proposed a fast bilateral filter method with $O(1)$ time complexity for arbitrary spatial and range domain. The key of this acceleration method is to decompose the bilateral

filter into a set of spatial domain filters. Inspirited by [24], we first accelerate the guided joint bilateral filtering.

Let N represents the total number of grayscale values, then the intensity value of pixel x in reference image $D(x) \in \{0, \ldots, N-1\}$. Suppose $D(x) = k$, for each pixel y and each intensity value $k \in \{0, \ldots, N-1\}$, we define

$$W_k(y) = g\big(k - D(y)\big) \cdot h\big(I(y) - D(y)\big) \tag{8}$$

$$J_k(x) = W_k(y) \cdot I(y) \tag{9}$$

Then, the guided joint filter (Eq. (5)) can be decomposed into N linear filters:

$$J_k^G(x) = \frac{\sum_{y \in \Omega(x)} (f(x-y) \cdot J_k(y))}{\sum_{y \in \Omega(x)} (f(x-y) \cdot W_k(y))} \tag{10}$$

Then the filtered image can be expressed as

$$I_D^G(x) = J_{D(x)}^G(x) \tag{11}$$

We sample N' gray level $\{L_0, \ldots, L_{N'-1}\}$ from N, assuming $D(x) \in [L_k, L_{k+1}]$, then $I_D^G(x)$ can be obtained by linear difference between $J_k^G(x)$ and $J_{k+1}^G(x)$ [24].

As both the $W_k(y)$ and $J_k(x)$ can be precomputed and stored in two-dimensional array, which eliminates the redundancy calculation. Thus, from Eq. (10), we can find that the main cost is separately spatial filtering on $J_k(x)$ and $W_k(y)$. We apply Deriche's recursive method [1] to approximate Gaussian filter, which can get $O(1)$ time complexity, and the filtered result is very close to the result using exact filtering method. The total time complexity is $O(1)$ for the guided joint bilateral filtering invariant to filter kernel size.

The bilateral filter (Eq. (6)) can be accelerated in the same way as described in [24] with $O(1)$ time complexity invariant to filter kernel size. Median filter in step 2 also can be accelerated using the method in [24] in an approximate way with $O(1)$ time complexity. Weiss [20] proposed a fast and exact median filter with $O(\log r)$, where r is the size of the window radius. Though [20] can yield the exact median result, we find the median filter acceleration [24] can receive desirable results in our experiments. The whole time complexity of our dehazing algorithm is linear in the number of pixels of the image. As the bilateral filter, the guided joint bilateral filter and the median filter can be implemented parallel, thus, our method can be further accelerated using GPU (Graphics Processing Unit).

5 Fast video dehazing

Compared with single haze removing, video haze removal is more challenging for the following reasons. First, the video data is usually much larger than a single image, thus efficiency is critical for video dehazing. Secondly, with the moving camera as well as the moving object in the video, the atmosphere veil is also varying according to a changing scene. Finally, it is important to maintain the temporal coherence of the dehazing results. As our dehazing method can perform single image dehazing in real time, thus it is possible to employ the proposed method for fast atmosphere veil computing to accelerate video dehazing.

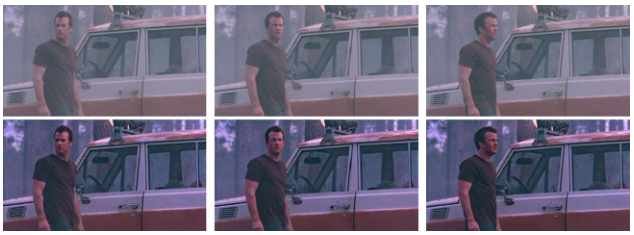

Fig. 6 Video dehazing. *Top row*, the input video, *bottom row*, our dehazed results. From *left* to *right*, the 48th, 62th, and 81th frame

To generate temporal-coherent dehazing results for live video streaming, we first compute the atmosphere veil $V_R(x)$ for each frame as it arrives in the input video streaming, using the same atmosphere veil recovering method as the single image. Then, instead of computing atmosphere light A in a current single frame, we estimate the A using several neighboring frames information around current frame (usually 5 frames), which makes more robust atmosphere light in the varying scene. Then based on $V_R(x)$ and A, we compute the transmission $t(x)$, and finally recover the scene irradiance $J(x)$. Figure 6 shows the video dehazing results using our method. For input video with size 1024×560, we efficiently remove the haze in the video streaming with an average 2.687 seconds to dehaze one frame. The experiments show that the proposed dehazing acceleration techniques make our method applicable for efficient video dehazing.

6 Experimental results

In this section, we show and discuss the results of the proposed method, and compare with several state of the art image dehazing methods in both image restoration quality and the time complexity. We also give several video streaming dehazing results to illustrate the affectivity and efficiency of the proposed method. Our approach is implemented in C++ on a Pentium Dual-Core CPU E5200@2.50 GHz with 2 GB RAM.

In our method, the window size Ω of median filter is very important to the recovery results. If Ω is too small, it will cause white object larger than Ω removed as fog in the original image; if Ω is too large, it will cause over blurring in the filtered image, and lots of edge detail lost in the generated atmospheric scattering light. In this paper, the window size is adaptively chosen as: $size(\Omega) = floor[2 \times (max(h, w)/50)] + 1$, where $floor[\cdot]$ represents the rounding operation, and w and h are the width and height

Table 1 Time consumption comparison with He et al. [3], He et al. [4], Tarel [18], and Fattal [2]

Image size	He et al. [3]	He et al. [4]	Tarel [18]	Fattal [2]	the proposed algorithm
441×450	18.67 minutes	16.579 seconds	0.521 seconds	26.496 seconds	1.016 seconds
384×399	13.11 minutes	12.766 seconds	0.362 seconds	20.457 seconds	0.719 seconds
651×509	39.87 minutes	27.516 seconds	1.076 seconds	44.241 seconds	1.325 seconds

Fig. 7 From *top* to *down*: the original images; atmosphere veil obtained through median filtering; atmosphere veil obtained through guided joint bilateral filtering; the transmission map; the dehazed images

of the input image, respectively. In the bilateral filtering and guided joint bilateral filtering steps, we can get desirable filtered results using both small and large range kernel variances σ_r and σ_t. In this paper, $\sigma_s = 0.03 \times min(h, w)$, $\sigma_r = 20$, $\sigma_t = 20$.

Figure 7 shows atmosphere veil compared with Tarel et al. [18]. From the second row, we can find that the atmosphere veil obtained only through the median filter lose much tiny edge information, such as the edges of the leaves. However, at these locations, the depth value changes abruptly, which is very important for the image restoration. By filtering the initial atmosphere veil using guided joint bilateral filter, as the third row shows, the generated atmosphere veil is smoother, the useless texture details are further removed, and the edge information is recovered effectively. We also present the computed transmission map $t(x)$ in the fourth row, and the results of $t(x)$ conform to actual depth of scenes. The final dehazed images are shown in the fifth row.

In Fig. 8, we compare with the state of the art dehazing methods based on single image, such as He et al. [3], He et al. [4], Tarel [18], and Fattal [2]. We can find that, for most images, He et al. [3] can produce impressive recovery results. In [4], He et al. further proposed a new guided filter, and took the source foggy image as a reference image to refine transmission map. Compared with [3], the running speed of this method is significantly improved, while this method may sacrifice image dehazing quality. Method [18] is also very fast, however, it sometimes may fail in processing tiny boundaries with abrupt depth jumps, and note that the haze between the small leaves cannot be removed. Fattal's method [2] is physically based and can produce satisfactory dehazing results in most cases, however, this approach cannot handle heavy haze images well. Furthermore, this approach is based on the assumption that the transmission and surface shading are locally uncorrelated, when this assumption is broken, this approach may be ineffective. Our method could get satisfactory dehazing results in distant scenes and places where the depth changes abruptly, as shown in the last column of Fig. 8, and the haze between the leaves are removed completely.

In Table 1, we give the time consumption comparison with He et al. [3], He et al. [4], Tarel [18], and Fattal [2]. We can find that our method is much faster than [3] and [2], also is faster than [4]. The performance of our method is comparable with Tarel [18], although we use two more filtering operations, as its time-complexity is $O(1)$, thus our method is still very fast.

Figure 9 shows the restored results of Dunhuang frescos. The frescos of Dunhuang are valuable materials for researching on the Chinese art history. However, under the influence of time and natural environment, some local regions of frescoes become fuzzy and many details are missing. Enhanced by our dehazing algorithm, the result images

Fig. 8 From *left* to *right*: the original foggy images, recovered images using He et al. [3], He et al. [4], Tarel [18], Fattal [2] and the proposed algorithm, respectively

are well restored. As shown in Fig. 9, the overall contrast of the restored images has been significantly increased and many hidden details are visible.

We also present two video dehazing results (see Electronic supplementary material) one input video is high resolution, and the other one is low resolution video; both are with heavy haze. Figure 6 shows the video dehazing results using our method. Using our method on CPU, for processing high resolution video with size of $1024 \times 560 \times 125$, it takes average 2.687 seconds to dehaze one frame. As our method can be performed in parallel, we run our algorithm on GPU, and the GPU acceleration is based on CUDA [32] and run on a NVIDIA GeForce GTX 285 (1 GB) graphics card. It takes average 0.231 seconds to process one frame. The experiments show that our method is an efficient tool for video dehazing.

7 Conclusion

In this paper, we propose a new fast dehazing method from single image based on filtering. We firstly obtain an initial atmosphere scattering light through median filtering, then refine it by using a guided joint bilateral filtering to generate a new atmosphere veil which recover the depth edge information. Finally, the scene radiance can be restored using the atmosphere attenuation model. Our method is fast with linear complexity in the number of pixels of the input image, which makes our method applicable for a real-time requirement. The experimental results demonstrate that the proposed method cannot only achieve better haze removal

Fig. 9 Restored results of Dunhuang frescos. *Left*: the input images, *right*: the recovered images using the proposed algorithm

results, but also have capability to process video streaming dehazing in real-time.

Our method also has a common limitation with most dehazing methods. Similar to [3], when there are some objects whose color is similar to the atmospheric light in the scene, the color will become very dark after dehazing. As the applied atmosphere attenuation model is relatively simple. for more complex situations, such as strong sunlight on

Fig. 10 Failed example. *Left*: the input image, *right*: the recovered image using the proposed algorithm

the sky region, the dehazing results may not be satisfactory, as Fig. 10 shows. Therefore, our future work is to develop more applicable model to address this problem. Finally, it is important to make some tone adjustments to receive satisfactory dehazing results, thus we would like to use the efficient edit propagation method [23] incorporating fast image clustering [21], and similar regions matching [22] to work on this direction.

Acknowledgements This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), NSFC (No. 61070081), Open Project Program of the State Key Laboratory for Novel Software Technology (kfkt2010B05), the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1208), and Luojia Outstanding Young Scholar Program of Wuhan University. Thanks to Peng Yin for the thoughtful discussions on the guided joint filter, and thanks to Xiangyun Hu for proofreading the manuscript.

References

1. Deriche, R.: (1993) Recursively implementing the Gaussian and its derivatives. Research Report 1893, INRIA
2. Fattal, R.: Single image dehazing. In: ACM Transactions on Graphics (TOG), vol. 27, p. 72. ACM, New York (2008)
3. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: CVPR 2009, pp. 1956–1963. IEEE Press, New York (2009)
4. He, K., Sun, J., Tang, X.: Guided image filtering. In: ECCV 2010, pp. 1–14 (2010)
5. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Trans. Graph. **26**(3), 96 (2007)
6. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: Model-based photograph enhancement and viewing. In: ACM Transactions on Graphics (TOG), vol. 27, p. 116. ACM, New York (2008)
7. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)
8. Narasimhan, S., Nayar, S.: Chromatic framework for vision in bad weather. In: CVPR 2000, vol. 1, pp. 598–605. IEEE Press, New York (2000)
9. Narasimhan, S., Nayar, S.: Interactive (de) weathering of an image using physical models. In: IEEE Workshop on Color and Photometric Methods in Computer Vision (2003)
10. Narasimhan, S., Nayar, S.: Contrast restoration of weather degraded images. In: ACM SIGGRAPH ASIA 2008 Courses. ACM, New York (2008)
11. Narasimhan, S., Nayar, S.: Vision and the atmosphere. In: ACM Siggraph Asia 2008 Courses, p. 69. ACM, New York (2008)
12. Paris, S., Kornprobst, P., Tumblin, J.: Bilateral Filtering: Theory and Applications. Now, Boston (2009)
13. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. In: ACM Transactions on Graphics (TOG), vol. 23, pp. 664–672. ACM, New York (2004)
14. Schechner, Y., Narasimhan, S., Nayar, S.: Instant dehazing of images using polarization. In: CVPR 2001, vol. 1, pp. 1–325. IEEE Press, New York (2001),
15. Schechner, Y., Narasimhan, S., Nayar, S.: Polarization-based vision through haze. In: ACM SIGGRAPH ASIA 2008 Courses. ACM, New York (2008)
16. Shwartz, S., Namer, E., Schechner, Y.: Blind haze separation. In: CVPR 2006, vol. 2, pp. 1984–1991 (2006)
17. Tan, R.: Visibility in bad weather from a single image. In: CVPR 2008 (2008)
18. Tarel, J., Hautiere, N.: Fast visibility restoration from a single color or gray level image. In: ICCV 2009, pp. 2201–2208. IEEE Press, New York (2009)
19. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV 1998, pp. 839–846. IEEE Press, New York (1998)
20. Weiss, B.: Fast median and bilateral filtering. In: ACM Transactions on Graphics (TOG), vol. 25, pp. 519–526. ACM, New York (2006)
21. Xiao, C., Liu, M.: Efficient mean-shift clustering using Gaussian kd-tree. In: Computer Graphics Forum, vol. 29, pp. 2065–2073. Wiley, New York (2010)
22. Xiao, C., Liu, M., Yongwei, N., Dong, Z.: Fast exact nearest patch matching for patch-based image editing and processing. IEEE Trans. Vis. Comput. Graph. **17**(8), 1122–1134 (2011)
23. Xiao, C., Nie, Y., Tang, F.: Efficient edit propagation using hierarchical data structure. IEEE Trans. Vis. Comput. Graph. **17**(8), 1135–1147 (2011)
24. Yang, Q., Tan, K., Ahuja, N.: Real-time o (1) bilateral filtering. In: CVPR 2009, pp. 557–564. IEEE Press, New York (2009)

Chunxia Xiao received the Ph.D. from the State Key Lab. of CAD& CG, Zhejiang University, in 2006, and is currently an associate professor at the School of Computer, Wuhan University, China. During October 2006−April 2007, he worked as a post-doc at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. His research interests include digital geometry processing, image and video processing, and computational photography.

Jiajia Gan is a master student at the School of Computer Science, Wuhan University, China. Her research interests include image and video editing, computational photography.