Research Article
Fast Video Dehazing Using Per-Pixel Minimum Adjustment

Zhong Luan (ID),[1,2] **Hao Zeng,**[1] **Yuanyuan Shang** (ID),[1,2,3] **Zhuhong Shao,**[2,3] **and Hui Ding**[2,3]

[1]*College of Information Engineering, Capital Normal University, Beijing, China*
[2]*Beijing Advanced Innovation Center for Imaging Technology, Beijing, China*
[3]*Beijing Engineering Research Center of High Reliable Embedded System, Beijing, China*

Correspondence should be addressed to Yuanyuan Shang; syy@bao.ac.cn

To reduce the computational complexity and maintain the effect of video dehazing, a fast and accurate video dehazing method is presented. The preliminary transmission map is estimated by the minimum channel of each pixel. An adjustment parameter is designed to fix the transmission map to reduce color distortion in the sky area. We propose a new quad-tree method to estimate the atmospheric light. In video dehazing stage, we keep the atmospheric light unchanged in the same scene by a simple but efficient parameter, which describes the similarity of the interframe image content. By using this method, unexpected flickers are effectively eliminated. Experiments results show that the proposed algorithm greatly improved the efficiency of video dehazing and avoided halos and block effect.

1. Introduction

In the communities of satellite remote sensing, aviation, shipping, and land transportation, the images and videos obtained by electronic equipment are expected to be clear enough. But, in the real scenes, the existence of haze greatly degrades the quality of the captured images and videos. This not only affects the reliability of the monitoring devices but also may cause potential danger. Therefore, it is imperative to develop a simple and efficient real-time dehazing algorithm.

Despite being an ill-posed problem, single-image dehazing has different types of prior approaches. A priori based methods tend to learn the statistical law of haze-free images. He et al. [1] propose a dark channel prior for haze-free images, and the dehazing effect of this algorithm is impressive. But its performance on the sky region and white objects is unsatisfactory, and the time complexity of soft matting is very high. As the best algorithm at that time, many algorithms improve the DCP method in some particular aspects [2–4]. Chen et al. [5] divide the image into foreground and background based on Fisher's linear discriminant, to process images with a dramatic depth change. Chen et al. [6–9] combine improved DCP method with white balance and local contrast enhances into a unified solution, which achieves good results on

sandstorm weather. Luan et al. [10] maximize image contrast by a cost function that contains the contrast term and the information loss term. This method has obvious advantages in speed; however, some overenhancement appears in their results. Li and Zheng [11] present a simple but effective change of details prior to remove haze from a single image. But this method is mainly stable to local image regions containing objects at different depths. This method works well for most images, but the enhancement for heavy haze regions is not sufficient.

Machine learning has been developing rapidly in recent years. The research focus of dehazing is gradually transformed from image prior to the learning approaches [12–16]. While the effect of dehazing has been substantially improved, undesired results still exist in many challenging scenes, such as street views, thick fog areas, and sky regions. On the other hand, the efficiency of such methods is not dominant.

The main objective of this paper is to develop a fast dehazing algorithm for image and video, where per-pixel method and quad-tree algorithm are utilized. To estimate the transmission map and amend the transmission values that are not accurate enough, we employ the per-pixel method, which can effectively solve the problems of halos and block artifacts near the depth edges and color distortion in the

sky area, also greatly enhancing the speed of estimating the transmission. To further improve the efficiency, the improved quad-tree algorithm is adopted to estimate the atmospheric light. Meanwhile, the proposed algorithm is applied to video dehazing. By keeping the atmospheric light unchanged in the same scene, the proposed algorithm eliminates unexpected flickers in the video dehazing effectively and it achieves fast speed because of using the correlation between two neighbor frames.

2. Single-Image Dehazing

2.1. Haze Modeling. Dehazing is a problem of image restoration; the degradation of a haze image is due to the suspended particles in the turbid air. In this paper, the atmospheric scattering model is used to describe the formation process of haze images. Haze removal adopts the atmospheric scattering model that is used widely in the field of computer vision and computer graphics. Mathematically, the atmospheric scattering model [18] is given as

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

where $I(x)$ is the observed haze image at a pixel position x, J is the original scene radiance, A is the atmospheric light that represents the intensity of the sky light or background light, and t is the medium transmission, determined by the distance from the object to the camera and the turbidity of medium. According to this model, the task of dehazing algorithm is to estimate t and A from the haze image I. The accuracy of estimating these two parameters is the key to improve the dehazing process.

2.2. Atmospheric Light Estimation. Although the use of global atmospheric light [10] will have a better effect, its speed cannot meet the real-time requirements. In this article, an improved quad-tree algorithm is proposed to estimate the atmospheric light. It should be mentioned that our quad-tree method and the algorithm in [17] have similar results, but our method has faster speed.

The process of our quad-tree method is summarized as follows. The channel minimum map of image is firstly computed, which aims to avoid mistake estimation of the atmospheric light when there are extreme values in some color channels in a local patch. Secondly, the channel minimum map is equally divided into four blocks and then the mean gray values are calculated. Then divide the block whose mean gray value is maximal into four blocks equally. Repeat the above process until the block's size is less than the given threshold. Choose the maximum value of the last block's pixels in the input image I as the estimation value of atmospheric light. This algorithm converges very fast, and its time complexity is low. The process of the quad-tree algorithm is as shown in Figure 1.

2.3. Transmission Estimation. Based on the statistical characteristics on haze-free images, He et al. [1] proposed an empirical regularity that is called dark channel prior. Then they estimated transmission t as the minimum value in the local area of the minimum channel. However, there will be obvious halos and block artifacts after haze removal by using the above method; using soft matting or guided filter to refine the transmission map can improve this phenomenon but its time complexity is very high. In the algorithm of this paper, we estimate the transmission map by a very simple method and then try to optimize it. Considering that the minimal channel of the haze image part is larger than the clear part, we estimate the preliminary t as follows:

$$t(x) = 1 - \varepsilon \min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c}. \tag{2}$$

Usually, 0.85 is thought to be a proper value for ε. And, in accordance with specific information of the actual haze image, the value of ε could be adjusted suitably. Equation (2) uses per-pixel method to estimate the transmission, instead of dividing the image into blocks, which can preserve full and precise image details. So there is no need to refine the transmission by soft matting [19] or guided filter [20].

Theoretically, the transmission in a local area with the same scene depth should be uniform. But (2) cancels the minimum filter when estimating the transmission, which leads most details of the input image to be kept in the transmission map. To the pixels whose channel minimum values are relatively large, their transmission values calculated by (2) are smaller than the others in a local patch. So that those pixels will seem to be dim after haze removal.

So the transmission values of the pixels whose channel minimum values are relatively large should be increased properly. Here two parameters Δ and K are introduced:

$$\Delta(x) = \left| \min_{c \in \{r,g,b\}} I^c(x) - \min_{c \in \{r,g,b\}} A^c \right|. \tag{3}$$

K is a given threshold value. The pixels satisfying $\Delta(x) < K$ are thought to be the pixels whose channel minimum values are relatively large, which increase their transmission values properly. Otherwise, these pixels are thought to be the pixels whose channel minimum values are relatively small, which decrease their transmission values properly in order to promote the contrast. So here an adjustment parameter α is introduced, and (2) is redefined as

$$t(x) = \alpha(x) \left(1 - \varepsilon \min_c \frac{I^c(y)}{A^c} \right). \tag{4}$$

α is defined by

$$\alpha(x) = \max\left(\sqrt{\frac{K}{\Delta(x)}}, \mu \right), \quad \mu \in (0, 1]. \tag{5}$$

The parameter μ is used to decrease the transmission values properly where channel minimum values are relatively small. The value of μ should be greater than 0.7 in case of some pixels becoming too dark. The significance of the square root in (5) is to weaken excessive enhancement for the pixels whose values are very close to the atmospheric light, as shown in Figure 2. Moreover, after haze removal, the output image may be dark such that a light increment Δl should be added to the output image. Figure 3 shows the results of the period before and after optimizing.

(a)

(b)

(c)

Figure 1: Illustration of quad-tree algorithm. (a) Original image. (b) Channel minimum map. (c) Result of quad-tree division.
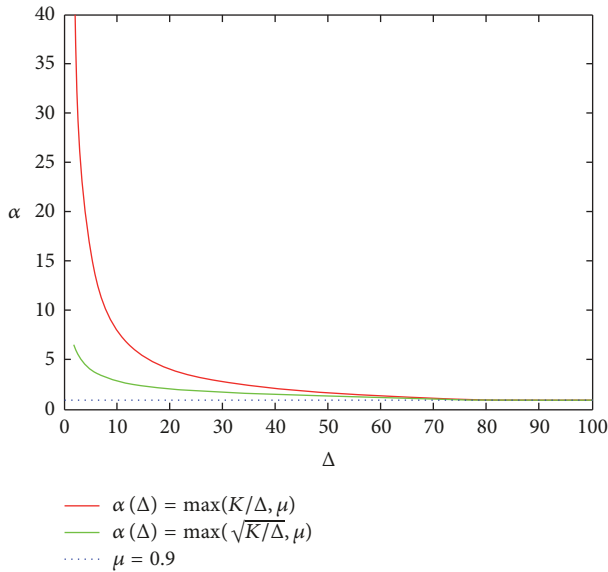

Figure 2: Curve of α relative to Δ ($K = 80$).

— $\alpha(\Delta) = \max(K/\Delta, \mu)$

— $\alpha(\Delta) = \max(\sqrt{K/\Delta}, \mu)$

⋯ $\mu = 0.9$

3. Video Haze Removal

3.1. Eliminating Unexpected Flickers. In this subsection, the proposed algorithm in Section 2 is applied to video dehazing. When the scene changes in a video, there will be normal flickers. However, in the same scene, smooth transitions should appear between the adjacent frames. But if we use our method to process a video frame by frame directly without other extra operations, there will be some obvious unexpected flickers even in the same scene. Here we call the factors, which cause this problem, instable factors. From (1) we can see that I, t, and A may be the possible factors causing this problem. Generally speaking, the original video usually is smooth and does not have unexpected flickers, so the video itself can not lead to the unexpected flickers' show. On the other hand, every pixel in a frame is processed independently in our method that it will not be affected by its neighbor pixels, which means I is a stable factor and we can ignore the influence of I. From (2) and (4), we can see that $t(x)$ is determined by $I(x)$ and A when ε, K, μ, and Δl are constants. As we know I is a stable factor, the instability of t is caused by A. Based on the above analysis, A is thought to be the only unstable factor that leads to the unexpected flickers. Figure 5 shows the unexpected flickers between serial frames in the same scene. The values of the estimated atmospheric light of the four frames in Figure 4 are (146, 142, and 141); (159, 142, and 142); (159, 142, and 142); and (145, 141, 140). The difference between the values of the estimated atmospheric light will be magnified by dividing by t when using (1) to recover the frames.

Actually, the atmospheric light usually keeps unchanged in the same scene in a short period of time. When different atmospheric light values are used to process two neighboring frames in the same scene, it may bring an unexpected flicker. A better strategy is to use the last frame's atmospheric light
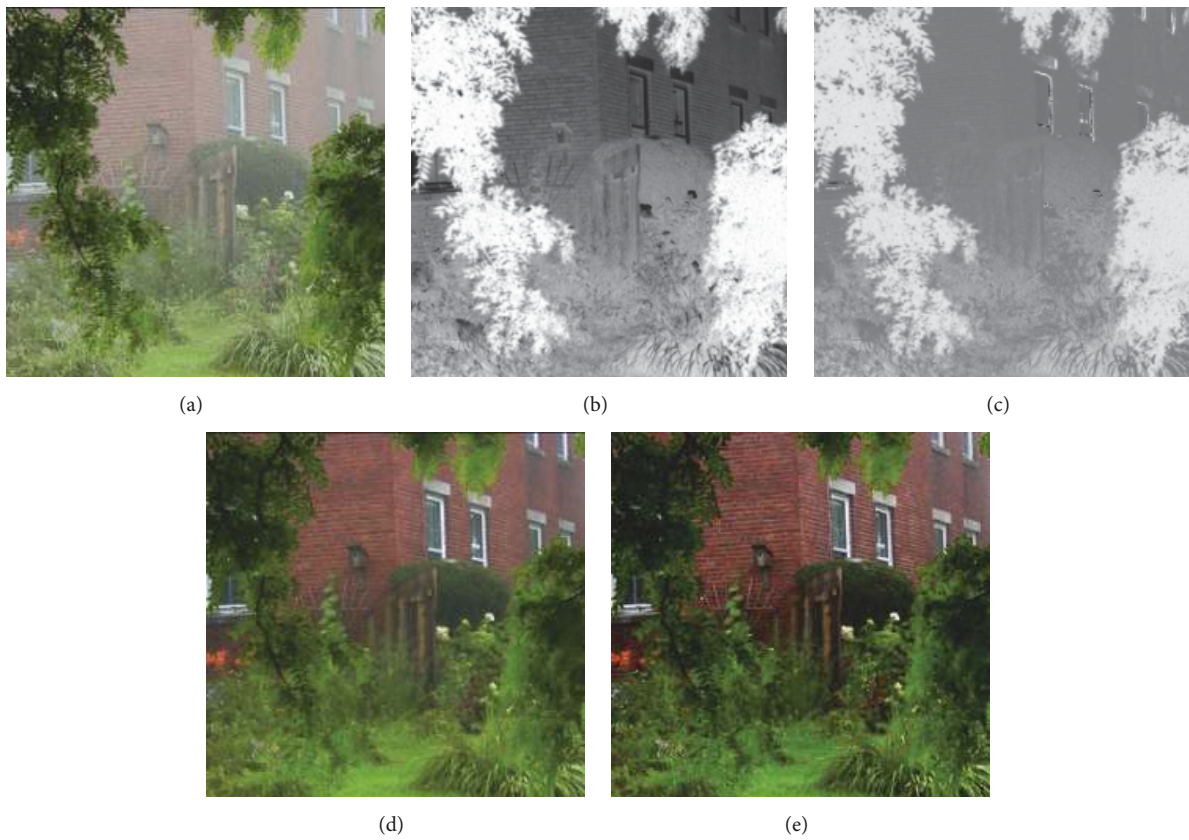
Figure 3: Results of before and after optimizing. (a) Input haze image. (b) Estimated transmission map before optimizing. (c) Estimated transmission map after optimizing. (d) Recovered image by (b). (e) Recovered image by (c). ($\varepsilon = 0.85, K = 80, \mu = 0.9,$ and $\Delta l = 0$).
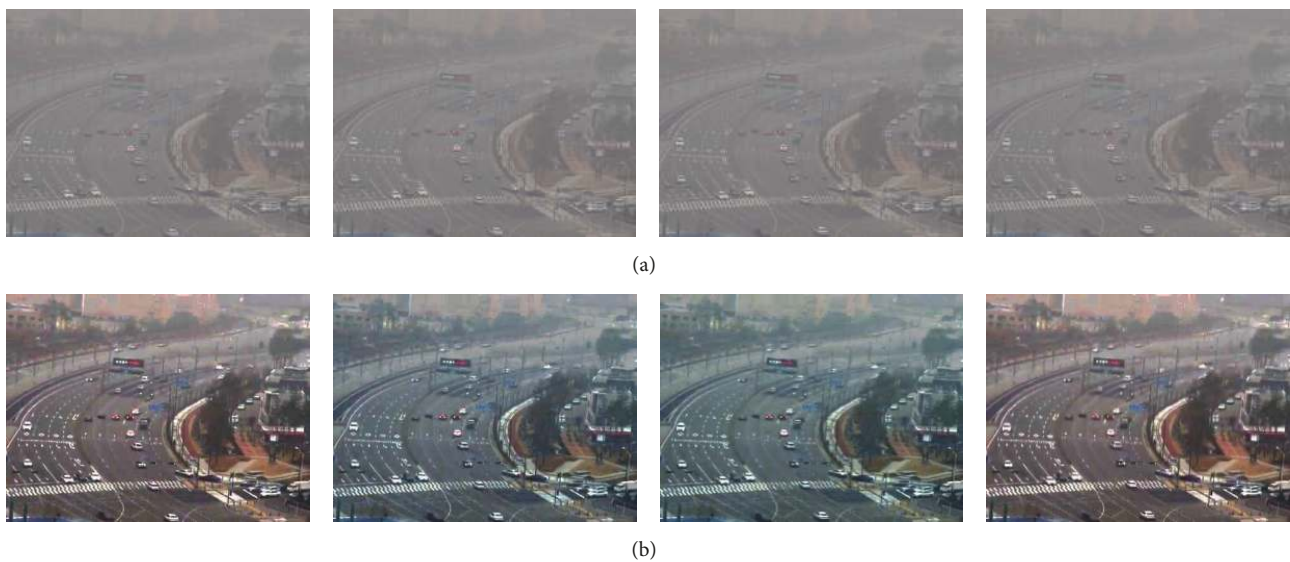

Figure 4: Unexpected twinkle in the same scene. (a) Serial input frames. (b) Output frames. ($\varepsilon = 0.95, K = 35, \mu = 0.9,$ and $\Delta l = 15$).

(a)

(b)

(c)

FIGURE 5: Three short fragments of a video. (a), (b), and (c) are three fragments of continuous frames. The ΔI_{ave} values of these frames are shown in Table 1.

value to recover the current frame instead of recalculating. This operation not only can avoid the unexpected flickers but also saves time.

To distinguish whether two neighboring frames are in the same scene, we put forward a simple and efficient method as follows. Firstly, two parameters ΔI and ΔI_{ave} are, respectively, defined by

$$\Delta I (x) = \mathrm{sum} \left| I_{\mathrm{current}} (x) - I_{\mathrm{last}} (x) \right|,$$
$$\Delta I_{\mathrm{ave}} = \frac{\mathrm{sum} (\Delta I)}{\mathrm{size} (\Delta I)}, \tag{6}$$

where I_{current} and I_{last} denote the current input frame and the last input frame. Then, if ΔI_{ave} is small enough, the current frame and the last frame are considered in the same scene and the atmospheric light A is unchanged. Otherwise, we recalculate the atmospheric light A.

An appropriate threshold is needed to be chosen for ΔI_{ave} to distinguish different scene correctly. Figure 5 shows three short fragments of a video. We can see each fragment changes scene at the third frame. Table 1 shows the ΔI_{ave} values of frames in Figure 5. It can be observed that, between two neighboring frames, the value of ΔI_{ave} is either very small or very large. In this paper, based on a large number of experiments, the threshold of ΔI_{ave} is set to 30.

3.2. Improving Efficiency. To further enhance the efficiency of video processing, we use the relativity between two neighboring frames. As mentioned above, our method adopts the approach of per-pixel processing to evaluate the transmission, and $t(x)$ is determined by $I(x)$ and A when ε, K, μ, and

TABLE 1: The ΔI_{ave} values of frames in Figure 5.

	Frame 1	Frame 2	Frame 3	Frame 4
Fragment 1	6.24	13.14	120.22	12.10
Fragment 2	5.06	6.00	114.81	7.67
Fragment 3	21.74	13.10	209.23	3.87

Δl are constants. However, from (1) we can see that $J(x)$ is determined by $I(x)$, A, and $t(x)$. So, in our method, $J(x)$ is only determined by $I(x)$ and A.

In Section 3.1, we keep the atmospheric light A of the current frame the same with the last frame if they are in the same scene. When the current frame and the last frame are in the same scene, and $I_{\mathrm{current}}(x)$ is equal to $I_{\mathrm{last}}(x)$, $J_{\mathrm{current}}(x)$ must be equal to $J_{\mathrm{last}}(x)$ where J_{current} is the current output frame and J_{last} is the last output frame. In this case, we do not need to recalculate $J_{\mathrm{current}}(x)$. For the neighboring frames in the same static scene, there are only a few pixels needed to be recalculated for haze removal. This will save a lot of time for us. But when the frames are in a dynamic scene, almost all the pixel values will change, such that almost all these pixel values of the current frame need to be recalculated. We can judge whether $I_{\mathrm{current}}(x)$ is equal to $I_{\mathrm{last}}(x)$ or not by judging whether $\Delta I(x)$ is equal to zero or not.

4. Experimental Results

To evaluate the performance of the proposed algorithm, experiments are performed on a computer with Intel(R) Core(TM) i7-6800K CPU @3.40 GHz and 32.00 GB RAM.

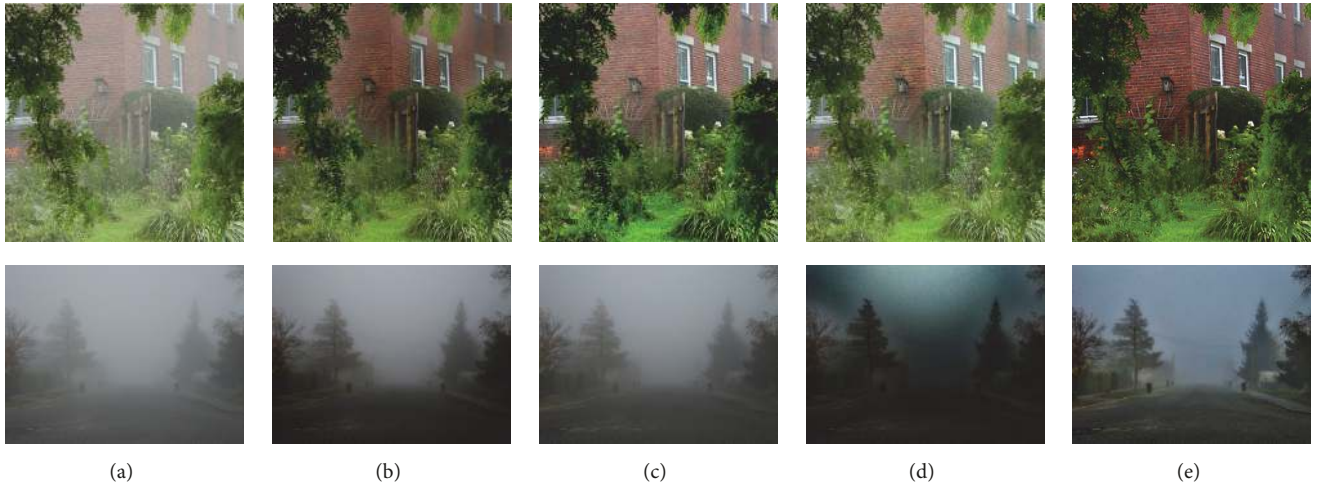| (a) | (b) | (c) | (d) | (e) |

FIGURE 6: Results of image dehazing by different algorithms: (a) original image. (b) Cai et al.'s result [13]. (c) Li and Zheng's result [11]. (d) Kim et al.'s result [17]. (e) Our result ($\varepsilon = 0.85, K = 80, \mu = 0.9, \Delta l = 0$ and $\varepsilon = 0.85, K = 60, \mu = 0.8,$ and$\Delta l = 0$).

TABLE 2: Time comparison of different algorithms.

Image resolution	Kim et al. [17]	Cai et al. [13]	Li and Zheng [11]	Ours
510×510	0.28	2.08	0.82	0.19
576×768	0.48	3.43	1.49	0.23
$1,024 \times 768$	0.94	5.91	2.58	0.65
$2,048 \times 1,280$	2.91	19.14	7.69	1.96

TABLE 3: Speed comparison between Kim et al.'s algorithm and our algorithm.

	Video 1	Video 2	Video 3	Video 4
Frame size	640×480	480×360	640×356	640×480
Kim et al.'s results	56.1 fps	86.0 fps	80.6 fps	56.8 fps
Our results	98.5 fps	175.1 fps	113.3 fps	95.0 fps

For the image haze removal, the experiments are implemented by using MATLAB R2017a and, for the video haze removal, the OpenCV is used.

4.1. Single-Image Haze Removal. Experiments have been done firstly to evaluate the efficiency of the proposed scheme for single image, and its performance is compared with those of the work reported in [11, 13, 17].

The dehazing results by using different algorithms are presented in Figure 6. In the "brick wall" photo, the plant at the foreground undergoes oversaturation in (b) and (c). At the same time, information loss occurred in (b) due to the overdarkness of some areas. There are obviously halo and block artifacts near the depth edges in (e). Our method generates better result (e) than other approaches; (e) not only can reserve almost all the details information of depth edge but also avoids generating halo and block artifacts. In the "road" photo, haze is not removed well in (b) and (c). Noise is amplified in (e) which was overenhanced. The contrast of (e) is significantly better than the other results.

Table 2 shows the time comparison of different algorithms performed on varying image size. It can be observed that our proposed algorithm cost the least time whatever the images size is. The result demonstrates that the proposed algorithm is practical to image dehazing.

4.2. Video Haze Removal. Next, experiments are carried out to see the validity of the proposed algorithm when it is applied to video dehazing. It should be noted that, in the video part, we only compare our algorithm to [17] for three reasons: (1) the efficiency of algorithm [11, 13] is not enough for real-time video dehazing, which is the main purpose of this article. (2) In the parameter optimization step, both [11] and [13] use the guided filter [20] to smooth their parameter maps, which would cause some halo artifacts as in [17]. (3) Without the atmospheric light strategy we proposed in the video dehazing part, none of [11, 13, 17] can eliminate unexpected flickers, which leads to approximate experimental results.

The results are shown in Figures 7 and 8. It can be seen that Kim's algorithm not only generated obvious halos and block artifacts near the depth edges but also brought distinct local flickers to the video. The problem of color distortion still exists. However, our algorithm can overcome the above-mentioned shortcomings.

Table 3 lists the cost time of Kim's algorithm and ours tested on different sizes. Obviously, our proposed algorithm is more efficient. It attributes to the fast image haze removal and the timesaving operation.

5. Conclusion

This paper introduces a real-time dehazing algorithm for single image and video, the transmission parameter t is
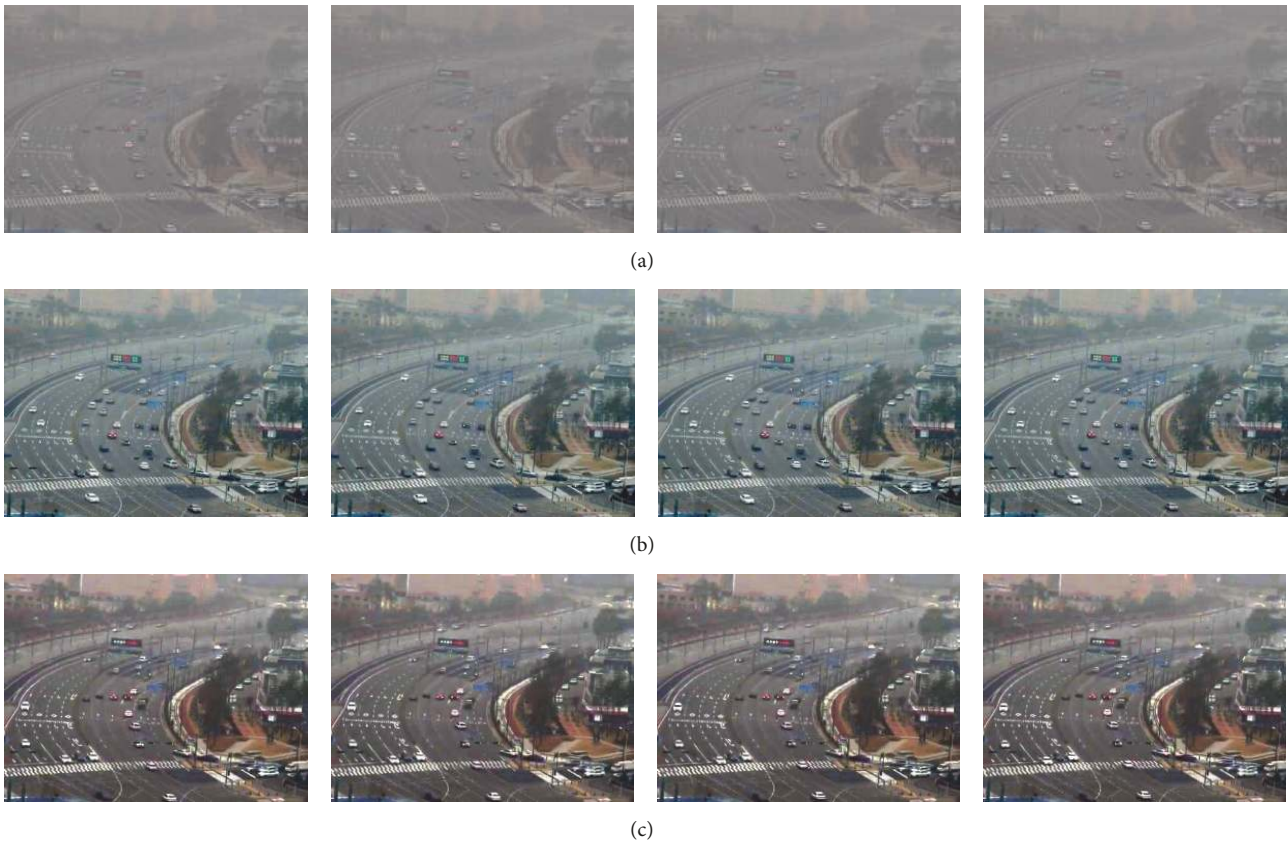
(a)

(b)

(c)

Figure 7: Results of video dehazing by different algorithms: (a) serial input frames. (b) Kim et al.'s result. (c) Our result ($\varepsilon = 0.95, K = 35, \mu = 0.9,$ and $\Delta l = 15$).
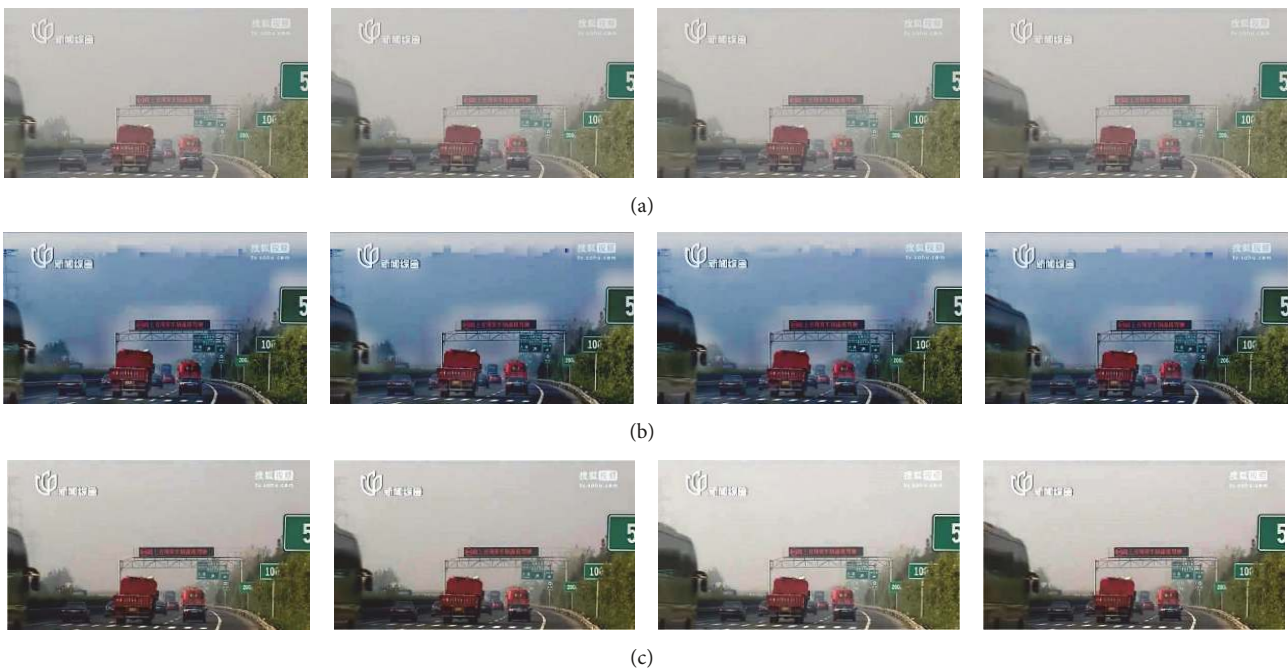

(a)

(b)

(c)

Figure 8: Results of video dehazing by different algorithms: (a) serial input frames. (b) Kim et al.'s result. (c) Our result ($\varepsilon = 0.85, K = 90, \mu = 0.80,$ and $\Delta l = 20$).

estimated by a per-pixel method instead of a block method, and the airlight A is estimated by our improved quad-tree method. Experiment results demonstrated the superior efficiency of the proposed algorithm. Future work will concentrate on applying the proposed algorithm to the images with deep scene depth and configure out the regulation of how to adjust the parameters in our algorithm when the scene changes in a video.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] K. M. He, J. Sun, and X. O. Tang, "Single image haze removal using dark channel prior," in *Proceeding of Conference on Computer Vision and Pattern Recognition*, vol. 33, pp. 1956–1963, 2011.

[2] L. Zeng and Y. Dai, "Single image dehazing based on combining dark channel prior and scene radiance constraint," *Journal of Electronics*, vol. 25, no. 6, pp. 1114–1120, 2016.

[3] B. Li, S. Wang, J. Zheng, and L. Zheng, "Single image haze removal using content-adaptive dark channel and post enhancement," *IET Computer Vision*, vol. 8, no. 2, pp. 131–140, 2013.

[4] X. Liu, H. Zhang, Y. Y. Tang, and J.-X. Du, "Scene-adaptive single image dehazing via opening dark channel model," *IET Image Processing*, vol. 10, no. 11, pp. 877–884, 2016.

[5] B.-H. Chen, S.-C. Huang, and J. H. Ye, "Hazy image restoration by bi-histogram modification," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, article 50, pp. 1–17, 2015.

[6] B.-H. Chen, S.-C. Huang, and F.-C. Cheng, "A high-efficiency and high-speed gain intervention refinement filter for haze removal," *Journal of Display Technology*, vol. 12, no. 7, Article ID 7384437, pp. 753–759, 2016.

[7] S.-C. Huang, B.-H. Chen, and Y.-J. Cheng, "An efficient visibility enhancement algorithm for road scenes captured by intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2321–2332, 2014.

[8] B.-H. Chen and S.-C. Huang, "Edge Collapse-Based Dehazing Algorithm for Visibility Restoration in Real Scenes," *Journal of Display Technology*, vol. 12, no. 9, Article ID 7450147, pp. 964–970, 2016.

[9] S.-C. Huang, B.-H. Chen, and W.-J. Wang, "Visibility restoration of single hazy images captured in real-world weather conditions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1814–1824, 2014.

[10] Z. Luan, Y. Shang, X. Zhou, Z. Shao, G. Guo, and X. Liu, "Fast single image dehazing based on a regression model," *Neurocomputing*, vol. 245, pp. 10–22, 2017.

[11] Z. Li and J. Zheng, "Edge-preserving decomposition-based single image haze removal," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5432–5441, 2015.

[12] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2995–3002, IEEE, Columbus, Ohio, USA, June 2014.

[13] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: an end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[14] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9906, pp. 154–169, 2016.

[15] M. Wang, J. Mai, Y. Liang, R. Cai, T. Zhengjia, and Z. Zhang, "Component-Based Distributed Framework for Coherent and Real-Time Video Dehazing," in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 321–324, Guangzhou, China, July 2017.

[16] B. Li, X. Peng, and Z. Wang, End-to-End United Video Dehazing and Detection, 2017.

[17] J.-H. Kim, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, "Optimized contrast enhancement for real-time image and video dehazing," *Journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 410–425, 2013.

[18] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233–254, 2002.

[19] L. Anat, L. Dani, and W. Yair, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2015.

[20] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.