



Optimized contrast enhancement for real-time image and video dehazing

Jin-Hwan Kim ^a, Won-Dong Jang ^a, Jae-Young Sim ^b, Chang-Su Kim ^{a,*}

^a School of Electrical Engineering, Korea University, Seoul, Republic of Korea

^b School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea

ARTICLE INFO

Article history:

Received 28 September 2012

Accepted 10 February 2013

Available online 18 February 2013

Keywords:

Image dehazing

Video dehazing

Image restoration

Contrast enhancement

Temporal coherence

Image enhancement

Optimized dehazing

Atmospheric light estimation

ABSTRACT

A fast and optimized dehazing algorithm for hazy images and videos is proposed in this work. Based on the observation that a hazy image exhibits low contrast in general, we restore the hazy image by enhancing its contrast. However, the overcompensation of the degraded contrast may truncate pixel values and cause information loss. Therefore, we formulate a cost function that consists of the contrast term and the information loss term. By minimizing the cost function, the proposed algorithm enhances the contrast and preserves the information optimally. Moreover, we extend the static image dehazing algorithm to real-time video dehazing. We reduce flickering artifacts in a dehazed video sequence by making transmission values temporally coherent. Experimental results show that the proposed algorithm effectively removes haze and is sufficiently fast for real-time dehazing applications.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

An image, captured in bad weather, often yields low contrast due to the presence of haze in the atmosphere, which attenuates scene radiance. Low contrast images degrade the performance of various image processing and computer vision algorithms. Dehazing is the process of removing haze from hazy images and enhancing the image contrast. Histogram equalization or unsharp masking can be employed to enhance the image contrast by stretching the histogram [1]. However, these methods do not consider that the haze thickness is proportional to object depths, which are locally different in an image. Thus, they cannot compensate the contrast degradation in a hazy image adaptively. More sophisticated dehazing algorithms first estimate object depths in a scene. Several dehazing algorithms have been proposed to estimate object depths using multiple images or additional information. For example, object depths are estimated from two images, which are captured in different weather conditions [2,3] or with different degrees of polarization [4,5]. Also, Kopf et al. [6] employed the prior knowledge of the scene geometry for dehazing. These algorithms can estimate scene depths and remove haze effectively, but require multiple images or additional information, which limits their applications.

Recently, single image dehazing algorithms have been developed to overcome the limitation of multiple image dehazing approaches. These algorithms make use of strong assumptions or constraints to remove haze from a single image. Tan [7] maximized the contrast of a hazy image, assuming that a haze-free image has a higher contrast ratio than the hazy image. Tan's algorithm, however, tends to overcompensate for the reduced contrast, yielding halo artifacts. Fattal [8] decomposed the scene radiance of an image into the albedo and the shading, and then estimated the scene radiance based on independent component analysis (ICA), assuming that the shading and the object depth are locally uncorrelated. It can remove haze locally but cannot restore densely hazy images. Kratz and Nishino [9] estimated the albedo and the object depth jointly by modeling a hazy image as a factorial Markov random field (FMRF). Tarel and Hautiere [10] estimated the atmospheric veil, which is the map of blended atmospheric light, and refined the veil using the median filter. He et al. [11] estimated object depths in a hazy image based on the dark channel prior, which assumes that at least one color channel should have a small pixel value in a haze-free image. They also applied an alpha matting scheme to refine the object depths. Ancuti et al. [12] significantly reduced the complexity of He et al.'s algorithm by modifying the block-based approach to a layer-based one. In addition, He et al.'s algorithm has been adopted and improved in many algorithms [13–16].

For video dehazing, Tarel et al. [17] focused on car vision. They partitioned a hazy video sequence into dynamically varying objects and a planar road, and then updated the scene depths only

* Corresponding author. Fax: +82 2 921 0544.

E-mail addresses: arite@korea.ac.kr (J.-H. Kim), wdjang@mcl.korea.ac.kr (W.-D. Jang), jysim@unist.ac.kr (J.-Y. Sim), changsukim@korea.ac.kr (C.-S. Kim).

for the objects using the still image dehazing scheme in [10]. Also, Zhang et al. [18] estimated an initial depth map for each frame of a video sequence, using the algorithm in [11], and then refined the depth map by exploiting spatial and temporal similarities. Oakley and Bu [19] assumed the all pixels in an image had similar depths and subtracted the same offset value from all pixels. Their algorithm is computationally simple, but it cannot adaptively remove haze when a captured image has variable scene depths.

The existing dehazing algorithms often exhibit overstretched contrast [7,9–11] or cannot remove dense haze [8] because of incorrect estimation of scene depths. To overcome these drawbacks, the contrast enhancement should be controlled more adaptively. Furthermore, the conventional video dehazing algorithms suffer from huge computational complexity [18] or low quality restored videos [19]. Therefore, an efficient real-time video dehazing algorithm is required for a wide range of practical applications.

In this work, we propose a fast dehazing algorithm for images and videos based on the optimized contrast enhancement. The proposed algorithm is based on our preliminary work on static image dehazing [20] and video dehazing [21]. We increase the contrast of a restored image to remove haze. However, if the contrast is overstretched, some pixel values are truncated by overflow or underflow. We design a cost function to alleviate this information loss while maximizing the contrast. Then, we find the optimal scene depth for each block by minimizing the cost function. Furthermore, for video dehazing, assuming that the scene radiance of an object point is invariant between adjacent frames, we add a temporal coherence cost to the total cost function. We also implement a parallel computing scheme for fast dehazing. Experimental results demonstrate that the proposed algorithm can estimate object depths in a scene reliably and restore the scene radiance efficiently.

The rest of the paper is organized as follows. Section 2 describes the haze model, which is employed in this work. Section 3 proposes the static image dehazing algorithm, and Section 4 describes the video dehazing algorithm. Section 5 presents experimental results. Finally, Section 6 concludes this work.

2. Haze modeling

The observed color of a captured image in the presence of haze can be modeled, based on the atmospheric optics [2], as

$$\mathbf{I}(p) = t(p)\mathbf{J}(p) + (1 - t(p))\mathbf{A}, \quad (1)$$

where $\mathbf{J}(p) = (J_r(p), J_g(p), J_b(p))^T$ and $\mathbf{I}(p) = (I_r(p), I_g(p), I_b(p))^T$ denote the original and the observed *r*, *g*, *b* colors at pixel position *p*, respectively, and $\mathbf{A} = (A_r, A_g, A_b)^T$ is the global atmospheric light that represents the ambient light in the atmosphere. Also, $t(p) \in [0, 1]$ is the transmission of the reflected light, which is determined by the distance between the scene point and the camera. Since the light traveling a longer distance is more scattered and attenuated, $t(p)$ is inversely proportional to the scene depth, and we have

$$t(p) = e^{-\rho d(p)}, \quad (2)$$

where $d(p)$ is the scene depth from the camera at pixel position *p*, and ρ is the attenuation coefficient determined by weather conditions and commonly assumed to be 1 in typical haze conditions [2]. From (1), note that the scene radiance $\mathbf{J}(p)$ is attenuated with $t(p)$. On the other hand, the atmospheric light \mathbf{A} is weighted by

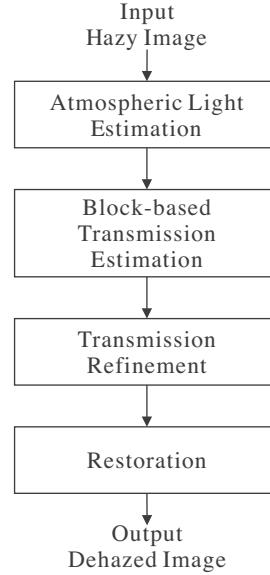


Fig. 1. Block diagram of the proposed static image dehazing algorithm.

$(1 - t(p))$ and plays a more important role, if the scene point is farther from the camera.

3. Static image dehazing

Fig. 1 shows the block diagram of the proposed dehazing algorithm. First, we determine the atmospheric light for an input hazy image. Then, we assume that scene depths are similar within an image block and find the optimal transmission for each block to maximize the contrast of the restored image. Moreover, we also minimize the information loss due to the truncation of pixel values, while enhancing the contrast. Then, we refine the block-based transmission values into the pixel-based ones by employing an edge preserving filter and shiftable windows. Finally, given the transmission map and the atmospheric light, we restore the scene radiance from the input hazy image.

3.1. Atmospheric light estimation

The atmospheric light \mathbf{A} in (1) is often estimated as the brightest color in an image, since a large amount of haze causes a bright color. However, in such a scheme, objects, which are brighter than the atmospheric light, may lead to undesirable selection of the atmospheric light. To estimate the atmospheric light more reliably, we exploit the fact that the variance of pixel values is generally low in hazy regions, e.g., sky. In addition, we propose a hierarchical searching method based on the quad-tree subdivision. More specifically, as illustrated in Fig. 2, we first divide an input image into four rectangular regions. We then define the score of each region as the average pixel value subtracted by the standard deviation of the pixel values within the region. Then, we select the region with the highest score and divide it further into four smaller regions. We repeat this process until the size of the selected region is smaller than a pre-specified threshold. For example, in Fig. 2, the red block is finally selected. Within the selected region, we choose the color vector, which minimizes the distance $\|(I_r(p), I_g(p), I_b(p)) - (255, 255, 255)\|$, as the atmospheric light. By minimizing the distance from the pure white vector $(255, 255, 255)$, we attempt to choose the atmospheric light that is as bright as possible.



Fig. 2. Atmospheric light estimation. By recursively dividing an image into four smaller regions and selecting the region with the highest score, we determine the region that is hazed most densely and then choose the atmospheric light within the region. In this example, the red block is the selected region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Optimal transmission estimation

We assume that scene depths are locally similar, as done in many dehazing algorithms [8,11,17], and find a single transmission value for each block of size 32×32 . Then, for each block with the fixed transmission value t , the haze equation in (1) can be rewritten as

$$\mathbf{J}(p) = \frac{1}{t}(\mathbf{I}(p) - \mathbf{A}) + \mathbf{A}. \quad (3)$$

After estimating the atmospheric light \mathbf{A} , the restored scene radiance $\mathbf{J}(p)$ depends on the selection of the transmission t . In general, a hazy block yields low contrast, and the contrast of a restored block increases as the estimated t gets lower. We attempt to estimate the optimal t so that the dehazed block has the maximum contrast.

Let us first review and discuss the three quantitative definitions of the contrast of a restored block. For simplicity, we define the contrast for one color channel.

- **Mean squared error (MSE) contrast:** The MSE contrast, C_{MSE} , represents the variance of pixel values [22], which is given by

$$C_{\text{MSE}} = \sum_{p=1}^N \frac{(J_c(p) - \bar{J}_c)^2}{N}, \quad (4)$$

where $c \in \{r, g, b\}$ is the color channel index, \bar{J}_c is the average of $J_c(p)$, and N is the number of pixels in a block. From (3), C_{MSE} of the restored block can be rewritten as

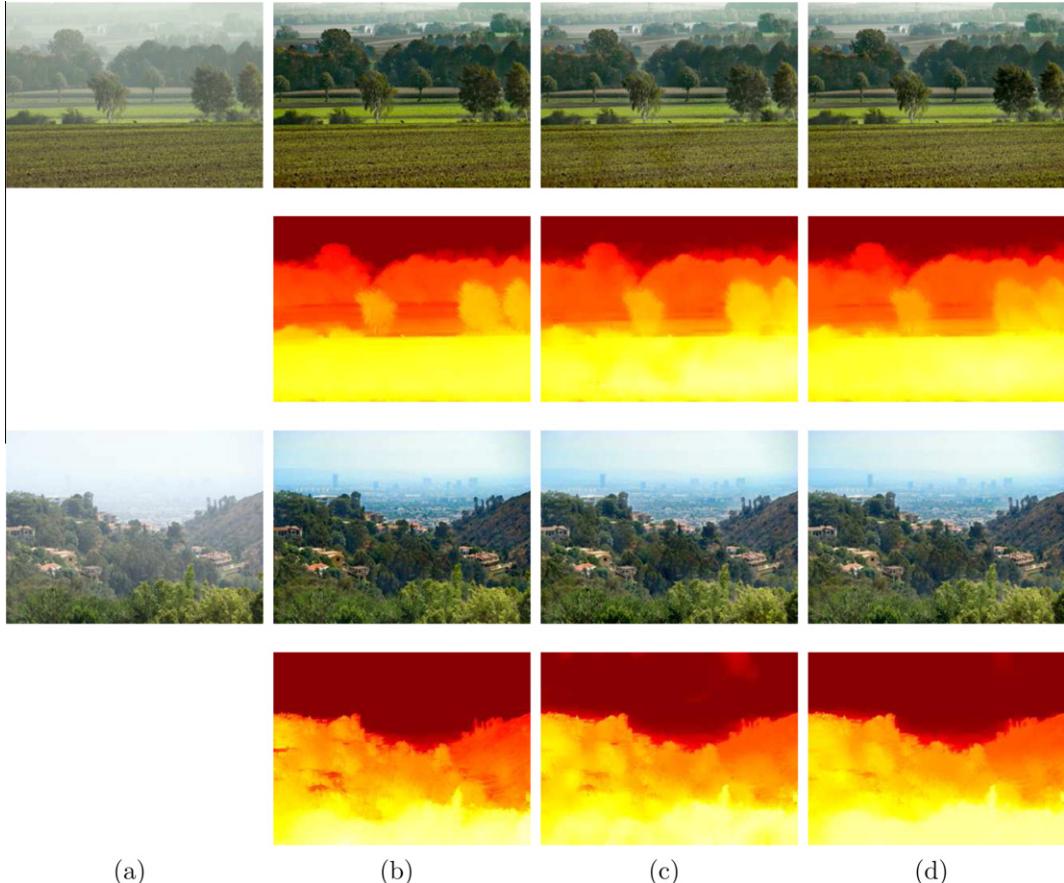


Fig. 3. Comparison of the dehazing results using the three definitions of contrast. (a) Input hazy images. The dehazed images and the corresponding transmission maps obtained by employing (b) the MSE contrast, (c) the Michelson contrast, and (d) the Weber contrast, respectively. In the transmission maps, yellow and red pixels represent near and far scene points, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\mathcal{C}_{\text{MSE}} = \sum_{p=1}^N \frac{(I_c(p) - \bar{I}_c)^2}{t^2 N}, \quad (5)$$

where \bar{I}_c is the average of $I_c(p)$ in the input block. Note from (5) that the MSE contrast is a decreasing function of t .

- **Michelson contrast:** The Michelson contrast, $\mathcal{C}_{\text{Michelson}}$, is typically used for periodic patterns and textures [23]. It is a measure of the difference between the maximum and the minimum values

$$\mathcal{C}_{\text{Michelson}} = \frac{J_{c,\max} - J_{c,\min}}{J_{c,\max} + J_{c,\min}}, \quad (6)$$

where $J_{c,\max}$ and $J_{c,\min}$ denote the maximum and the minimum values of $J_c(p)$. $\mathcal{C}_{\text{Michelson}}$ is also inversely proportional to t , since it can be rewritten as

$$\mathcal{C}_{\text{Michelson}} = \frac{I_{c,\max} - I_{c,\min}}{I_{c,\max} + I_{c,\min} - 2A_c + 2A_c t}, \quad (7)$$

where $I_{c,\max}$ and $I_{c,\min}$ denote the maximum and the minimum values of $I_c(p)$.

- **Weber contrast:** The Weber contrast, $\mathcal{C}_{\text{Weber}}$, is defined as the normalized difference between the background color $J_{c,\text{background}}$ and the object color $J_{c,\text{object}}$ [22], given by

$$\mathcal{C}_{\text{Weber}} = \frac{J_{c,\text{object}} - J_{c,\text{background}}}{J_{c,\text{background}}}. \quad (8)$$

The Weber contrast is widely used to model the human visual system. In practice, we regard each pixel value as an object color and the average pixel value as the background color. Then, we can derive

$$\mathcal{C}_{\text{Weber}} = \sum_{p=1}^N \frac{|J_c(p) - \bar{J}_c|}{N \bar{J}_c}, \quad (9)$$

which has a similar form to the MSE contrast in (4).

Any of these three definitions can be employed to measure the contrasts of restored blocks and dehaze an image. Fig. 3 shows dehazed images and the corresponding transmission maps when we employ the three definitions of contrast, respectively. We see that all three definitions provide similar results. In the remainder of this work, we adopt the MSE contrast to quantitatively measure the contrasts of restored blocks, but it is noted that the other two definitions can be also used as effectively as the MSE contrast for the purpose of dehazing.

Since the contrast measure is inversely proportional to the transmission t , we can select a small value of t to increase the contrast of a restored block. Fig. 4 shows an example of the transformation function in (3), which maps an input pixel value $I_c(p)$ to an output value $J_c(p)$. Note that input values in $[\alpha, \beta]$ are mapped to output values in the full dynamic range $[0, 255]$, where the transmission t determines the valid input range of $[\alpha, \beta]$. When most input values belong to $[\alpha, \beta]$, we can obtain a higher contrast output block. However, when many input values lie outside of $[\alpha, \beta]$, the transformed output values do not belong to the valid output range of $[0, 255]$. In such cases, the underflow or overflow occurs in some pixel values, which are truncated to 0 or 255. It means that the information is lost in the red regions in Fig. 4, degrading the quality of the restored block. In general, the amount of information loss is proportional to the areas of the red regions in Fig. 4, which are in turn proportional to the slope $1/t$ of the transformation function in (3). Therefore, the information loss can be reduced by selecting a large value of t in general. However, when a block contains dense haze, it has a relatively narrow range of input values. Thus, even though it is assigned a small value of t , most of its pixel values be-

long to $[\alpha, \beta]$ and are not truncated. On the contrary, a block with no haze exhibits a broad range of input values and should be assigned a large t to reduce the information loss due to the truncation. Fig. 5 shows examples of restored images according to different values of t . It is observed that, as t gets smaller, more pixel values are truncated in the restored images.

Thus, we should not only enhance the contrast but also reduce the information loss. To this end, we design the contrast cost and the information loss cost and then minimize the two cost functions simultaneously. First, we define the contrast cost, E_{contrast} , by taking the negative sum of the MSE contrasts for three color channels of each block B ,

$$E_{\text{contrast}} = - \sum_{c \in \{r,g,b\}} \sum_{p \in B} \frac{(J_c(p) - \bar{J}_c)^2}{N_B} = - \sum_{c \in \{r,g,b\}} \sum_{p \in B} \frac{(I_c(p) - \bar{I}_c)^2}{t^2 N_B} \quad (10)$$

where \bar{J}_c and \bar{I}_c are the average values of $J_c(p)$ and $I_c(p)$ in B , respectively, and N_B is the number of pixels in B . Note that, by minimizing E_{contrast} , we can maximize the MSE contrasts. Second, we define the information loss cost E_{loss} for block B as the squared sum of truncated values,

$$E_{\text{loss}} = \sum_{c \in \{r,g,b\}} \sum_{p \in B} \left\{ (\min\{0, J_c(p)\})^2 + (\max\{0, J_c(p) - 255\})^2 \right\} \quad (11)$$

$$= \sum_{c \in \{r,g,b\}} \left\{ \sum_{i=0}^{\alpha_c} \left(\frac{i - A_c}{t} + A_c \right)^2 h_c(i) + \sum_{i=\beta_c}^{255} \left(\frac{i - A_c}{t} + A_c - 255 \right)^2 h_c(i) \right\}, \quad (12)$$

where $h_c(i)$ is the histogram of input pixel value i in color channel c , and α_c and β_c denote the intercepts at which the truncation occurs, as illustrated in Fig. 4. In (11), the terms $\min\{0, J_c(p)\}$ and $\max\{0, J_c(p) - 255\}$ denote truncated values due to the underflow and the overflow, respectively. We rewrite the squared sum of the truncated values as (12) using the histogram. It is noted that the histogram uniformness term in our previous work in [20] plays a similar role as the information loss cost. However, the histogram uniformness term may yield incorrect results when a haze-free region has a non-uniform histogram. Last, for block B , we find the optimal transmission t^* by minimizing the overall cost function

$$E = E_{\text{contrast}} + \lambda_L E_{\text{loss}}, \quad (13)$$

where λ_L is a weighting parameter that controls the relative importance of the contrast cost and the information loss cost.

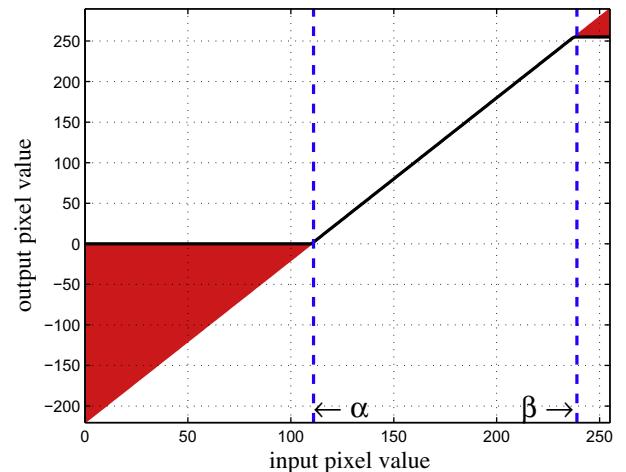


Fig. 4. An example of the transformation function. Input pixel values are mapped to output pixel values according to the transformation function, depicted by the black line. The red regions represent the information loss due to the truncation of output pixel values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

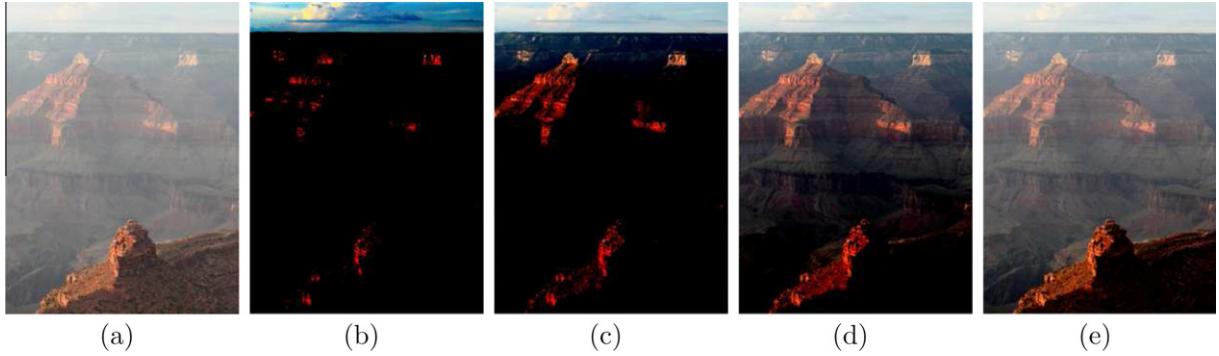


Fig. 5. Relationship between the transmission value and the information loss. A smaller transmission value causes more severe truncation of pixel values and a larger amount of information loss. (a) An input hazy image. The restored dehazed images with transmission values of (b) $t = 0.1$, (c) $t = 0.3$, (d) $t = 0.5$, and (e) $t = 0.7$.

A large value of λ_L in (13) reduces the information loss. In the extreme case of $\lambda_L = \infty$, the optimal transmission value should not yield any information loss, i.e.,

$$\min_{c \in \{r,g,b\}} \min_{p \in B} J_c(p) \geq 0, \quad (14)$$

$$\max_{c \in \{r,g,b\}} \max_{p \in B} J_c(p) \leq 255. \quad (15)$$

These inequalities, together with the relation in (3), impose two constraints for the transmission t , given by

$$t \geq \min_{c \in \{r,g,b\}} \min_{p \in B} \left\{ \frac{I_c(p) - A_c}{-A_c} \right\}, \quad (16)$$

$$t \geq \max_{c \in \{r,g,b\}} \max_{p \in B} \left\{ \frac{I_c(p) - A_c}{255 - A_c} \right\}. \quad (17)$$

The two constraints can be combined into a single constraint

$$t \geq \max \left\{ \min_{c \in \{r,g,b\}} \min_{p \in B} \left\{ \frac{I_c(p) - A_c}{-A_c} \right\}, \max_{c \in \{r,g,b\}} \max_{p \in B} \left\{ \frac{I_c(p) - A_c}{255 - A_c} \right\} \right\}. \quad (18)$$

Notice that E_{contrast} is an increasing function of t . Therefore, the optimal transmission t^* is determined as the smallest value satisfying the constraint in (18). In other words,

$$t^* = \max \left\{ \min_{c \in \{r,g,b\}} \min_{p \in B} \left\{ \frac{I_c(p) - A_c}{-A_c} \right\}, \max_{c \in \{r,g,b\}} \max_{p \in B} \left\{ \frac{I_c(p) - A_c}{255 - A_c} \right\} \right\}. \quad (19)$$

It is worthy to point out that the first constraint in (16) is the same constraint that is employed as the dark channel prior in the He et al.'s algorithm [11]. Using this constraint, [11] provides faithful dehazing results, provided that objects are rarely brighter than the atmospheric light. However, when some objects are brighter than the atmospheric light, [11] fails to estimate the transmission correctly. In contrast, the proposed algorithm employs the additional constraint in (17), which prevents the overflow of restored pixel values. Therefore, the proposed algorithm can estimate the transmission more reliably. Moreover, by controlling λ_L in (13), the proposed algorithm can strike a balance between the contrast enhancement and the information loss.

3.3. Transmission refinement

In Section 3.2, we assumed that all pixels in a block have the same transmission value. However, scene depths may vary spatially within a block, and the block-based transmission map usually yields blocking artifacts. Therefore, by using an edge preserving filter, we refine the block-based transmission map, alleviate the blocking artifacts, and enhance the image details. Edge

preserving filtering attempts to smooth an image, while preserving the edge information [1,24,25]. In this work, we adopt the guided filter [25], which assumes that the filtered transmission $\hat{t}(q)$ is an affine combination of the guidance image $\mathbf{I}(q)$ as follows.

$$\hat{t}(q) = \mathbf{s}^T \mathbf{I}(q) + \psi, \quad (20)$$

where $\mathbf{s} = (s_r, s_g, s_b)^T$ is a scaling vector and ψ is an offset. The scaling vector and the offset are determined for each local window of size 41×41 . For a window W , the optimal parameters, \mathbf{s}^* and ψ^* , are obtained, by minimizing the difference between the initial transmission $t(q)$ found in Section 3.2 and the filtered transmission $\hat{t}(q)$, using the least squares method:

$$(\mathbf{s}^*, \psi^*) = \arg \min_t (\mathbf{s}, \psi) \sum_{q \in W} (t(q) - \hat{t}(q))^2. \quad (21)$$

The window slides pixel by pixel over the entire image, and multiple windows overlap at each pixel position. Therefore, at each pixel position, we can determine the final transmission value as the average of all associated refined transmission values. We call this approach as the centered window scheme.

The centered window scheme reduces blocking artifacts by averaging the refined transmission values of overlapping windows. However, the averaging process may cause blurring in the final transmission map, especially around object boundaries across which depths change abruptly. The blurring in the transmission map, in turn, yields halo artifacts in the dehazed image. To overcome this problem, we employ the shiftable window scheme, instead of the centered window scheme. Note that shiftable windows were used for improving the stereo matching performance [26]. As illustrated in Fig. 6(a), the centered window scheme overlays a window on each pixel so that the window is centered at the pixel. In this example, the window contains multiple objects with different depths, leading to unreliable depth estimation. On the other hand, in the shiftable window scheme in Fig. 6(b), for each pixel, we shift the window within a search range and select the optimal shift position that minimizes the variance of pixel values within the window. Thus, in general, optimal windows are selected at smooth regions and do not contain strong edges. Then, similarly to the centered window scheme, we refine the transmission values of the optimal windows via (20) and determine the final transmission value at each pixel as the average of all associated refined values. Notice that, even though a shiftable window is selected for each pixel, the number of overlapping windows varies according to the pixel position. This is because windows at smooth regions are selected more frequently than those at edge regions. The shiftable window scheme hence can reduce the contributions of unreliable transmission values derived from edge regions, thereby alleviating blurring artifacts.



Fig. 6. Illustration of the shiftable window scheme: (a) centered window and (b) shiftable window.

Fig. 7 shows examples of block-based transmission maps and the corresponding pixel-based refined maps. We see that the pixel-based maps effectively reduce blocking artifacts and preserve image details more accurately. In addition, as compared in **Fig. 7(c)** and **(d)**, the shiftable window scheme alleviates blurring artifacts and provides more faithful transmission values than the centered window scheme, e.g., near the boundary of the bird.

After obtaining the pixel-based transmission map, we dehaze the input image based on **(1)**. However, as suggested in **[11]**, we constrain the minimum transmission value to be greater than 0.1, since a smaller value tends to amplify noise. Furthermore, the restored hazy image often has darker pixel values than the input image. Thus, we apply the gamma correction **[1]** to the restored image with an empirically selected gamma of 0.8.

4. Video dehazing

The dehazing algorithm in Section 3 provides good results on static images. However, when applied to each frame of a hazy video sequence independently, it may break temporal coherence and produce a restored video with severe flickering artifacts. Moreover, its high computational complexity prohibits real-time applications, such as car vision or video surveillance. In this section, we propose a fast and temporally coherent dehazing algorithm for video sequences.

4.1. Temporal coherence

Let us first consider the relationship between the transmission values of consecutive image frames. The transmission values change due to camera and object motions. As an object approaches the camera, the observed radiance gets closer to the original scene radiance. On the contrary, when an object moves away from the camera, the observed radiance becomes more similar to the atmospheric light. Thus, we should modify the transmission value of a scene point adaptively according to its brightness change.

In the proposed video dehazing algorithm, we first convert a video sequence into the YUV color space. We then process only the luminance (*Y*) component, without modifying the chrominance (*U*, *V*) components, to reduce the computational complexity. We empirically observe that dehazing results using the *Y* component only are comparable to those obtained in the RGB color space. Also, notice that the *U* and *V* components are less affected by haze than the *Y* component. Thus, if all *Y*, *U*, *V* components are used for the transmission estimation in the same way as the R, G, B components are used, the estimation becomes unreliable and the dehazing results are degraded severely. Therefore, the *U* and *V* components should not be used in the transmission estimation and the dehazing.

Let $J_Y^k(p)$ and $I_Y^k(p)$ be the *Y* components of the scene radiance and the observed radiance, respectively, at pixel *p* in the *k*th image frame. We assume that the original radiance of a scene point is the same between two consecutive image frames. Specifically,

$$J_Y^{k-1}(p) = J_Y^k(p). \quad (22)$$

We also assume that the luminance A_Y of the atmospheric light is the same for an entire video sequence. However, when a scene change occurs, A_Y may vary substantially and it should be newly estimated. Thus, in practice, we can employ a scene change detection algorithm, e.g. **[27]**, and estimate the atmospheric light again after each scene change. From **(1)**, we can easily obtain the relationship between the transmission $t_k(p)$ in the current frame and the transmission $t_{k-1}(p)$ in the previous frame,

$$t_k(p) = \tau_k(p)t_{k-1}(p), \quad (23)$$

where $\tau_k(p)$ is the temporal coherence factor, which corrects the transmission according to the change in the observed scene radiances, given by

$$\tau_k(p) = \frac{I_Y^k(p) - A_Y}{I_Y^{k-1}(p) - A_Y}. \quad (24)$$

In **(23)**, we compare two pixels at the same position in the *k*th frame and the $(k-1)$ th frame. However, an object may move and the same scene point may be captured at different pixel positions. To address this issue, the position of a moving object can be tracked, e.g., using the block matching method **[1]** or the optical flow estimation **[28]**. However, conventional motion estimation schemes demand high computational complexity in general, when the size of the searching window increases. Therefore, to achieve fast computation, we do not estimate motion vectors explicitly. Instead, we employ a simple probability model, based on the differential image between the two frames, which is given by

$$w_k(p) = \exp\left(-\frac{(I_Y^k(p) - I_Y^{k-1}(p))^2}{\sigma^2}\right), \quad (25)$$

where σ controls the variance of the probability model. In this work, σ is empirically selected as 10. Note that $w_k(p)$ gets larger as $I_Y^k(p)$ becomes more similar to $I_Y^{k-1}(p)$. Thus, $w_k(p)$ represents the likelihood that the two pixels are the matching ones. Then, we define the temporal coherence factor $\bar{\tau}_k$ for block *B* as

$$\bar{\tau}_k = \frac{\sum_{p \in B} w_k(p)\tau_k(p)}{\sum_{p \in B} w_k(p)}. \quad (26)$$

In other words, the pixel-based factor $\tau_k(p)$ is multiplied by a larger weight $w_k(p)$ in the computation of the block-based factor $\bar{\tau}_k$, when $I_Y^k(p)$ and $I_Y^{k-1}(p)$ are more likely to come from the same scene point.

For each block, we define a temporal coherence cost by taking the squared difference between the transmission t_k in the current



Fig. 7. Transmission map refinement: (a) Input hazy images, (b) the block-based transmission maps, and the pixel-based transmission maps using (c) the centered window scheme and (d) the shiftable window scheme. In the transmission maps, yellow and red colors represent near and far scene points, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

frame and its estimation $\bar{t}_k t_{k-1}$ using the previous frame. However, the squared difference cannot reflect the similarity between the corresponding blocks in the current and previous frames exactly, when a scene change occurs or a new object appears. In such cases, t_k may be far from its estimation $\bar{t}_k t_{k-1}$. Hence, we introduce an additional weight

$$\bar{w}_k = \frac{1}{N_B} \sum_{p \in B} w_k(p), \quad (27)$$

which represents the block similarity between the two frames. Then, we define the temporal coherence cost E_{temporal} as

$$E_{\text{temporal}} = \bar{w}_k(t_k - \bar{\tau}_k t_{k-1})^2. \quad (28)$$

4.2. Cost function optimization

For video dehazing, we reformulate the overall cost in (13) by adding the temporal coherence cost E_{temporal} in (28), i.e.,

$$E = E_{\text{contrast}} + \lambda_L E_{\text{loss}} + \lambda_T E_{\text{temporal}}, \quad (29)$$

where λ_T is a weighting parameter. As λ_T gets larger, we emphasize the temporal coherence more strongly and alleviate flickering artifacts more effectively. However, a large λ_T may fix the optimal transmission value for each block over all frames, causing blurring artifacts and degrading the qualities of restored frames. Therefore, λ_T should be determined by considering the tradeoff between flickering artifacts and the qualities of individual frames.

Note that we first find the optimal transmission t_0^* for each block in the first frame by minimizing the cost function in (13), since there is no previous frame. Then, for subsequent frames, we obtain the optimal transmission t_k^* of each block by minimizing the augmented cost function in (29).

4.3. Fast transmission refinement

For fast dehazing of a video sequence, we reduce the complexity for computing pixel-based transmission values. Only the lumi-



Fig. 8. Dehazing results of the proposed algorithm on the “Cones,” “Forest,” “House,” “Town,” and “Plain” images: (a) the hazy images, (b) the estimated transmission maps, in which yellow and red pixels correspond to near and far scene points, respectively, and (c) the dehazed images. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

nance component is used to compute the pixel-based transmission values via

$$\hat{t}(q) = s_Y I_Y(q) + \psi, \quad (30)$$

where the optimal parameters s_Y^* and ψ^* are obtained by

$$(s_Y^*, \psi^*) = \arg \min_{s_Y, \psi} \sum_{q \in W} (t(q) - \hat{t}(q))^2. \quad (31)$$

This least squares optimization computes only two parameters, as compared with four parameters in (21). It, however, still requires high complexity to compute the least squares for the window around each pixel. Therefore, we sample the evaluation points

using the partially overlapping sub-block scheme in [29] and employ centered windows, instead of shiftable windows, for video dehazing. The partially overlapping scheme may cause blocking artifacts. To alleviate those artifacts, we use a Gaussian window: pixels around the window center have higher weights, whereas pixels farther from the center have lower weights. Then, we obtain the final optimal transmission value for each pixel, by computing the Gaussian weighted sum of the transmission values associated with the overlapping windows. Also, to further reduce the complexity, the proposed algorithm downsamples an input image when computing the transmission.

5. Experimental results

5.1. Static image dehazing

We evaluate the performance of the proposed static image dehazing algorithm on hazy “Cones,” “Forest,” “House,” “Town,” and “Plain” images in Fig. 8(a). “Cones” and “House” were used in [8], and the others were collected from flicker.com. Fig. 8(b) shows the estimated transmission maps, where yellow and red pixels represent near and far scene points, respectively. Fig. 8(c) shows the restored dehazed images when the parameter λ_L in (13) is set to 5. In the “Cones” and “Forest” images, upper areas have denser haze than lower areas, yielding weaker scene radiance and lower contrast. We see in Fig. 8(c) that the lowered contrast in the upper areas is restored in the dehazed images. On the other hand, pixel values in the lower areas are not much influenced by haze. In these lower areas, by employing the information loss cost, the proposed algorithm prohibits selecting too small transmission values and provides high image quality. Moreover, note that the proposed algorithm estimates transmission values reliably even for complex scenes, such as the “House” and “Town” images.

Fig. 9 compares the dehazing results according to the variation of the parameter λ_L in (13). As shown in Fig. 9(a), with a small value of $\lambda_L = 1$, the restored images have significantly increased contrast, but they lose information and contain unnaturally dark pixels due to the truncation of pixel values. On the contrary, with a large value of $\lambda_L = 8$, we can prevent the information loss but cannot remove haze fully. In general, $\lambda_L = 5$ strikes a balance between the information loss prevention and the haze removal effectively. Therefore, we fix λ_L to 5 in all experiments, unless otherwise specified.

We also compare the performance of the proposed algorithm with those of the conventional algorithms on the “Newyork1,” “Newyork2,” and “Mountain” images in Figs. 10–12, respectively. These images were used in [10]. The level control method and the histogram equalization, which are available in the Photoshop [30], stretch the histogram of an input image without considering the local variation of haze thickness. Thus, they do not provide adaptively enhanced results. Tan’s algorithm [7] generates many saturated pixels, since it simply maximizes the contrast of the restored images. Fattal’s algorithm [8] yields more natural results, but it cannot sufficiently remove haze in some regions, for example, the buildings around the horizon in Fig. 12, and the mountains

in Fig. 12(e). Tarel and Hautiere’s algorithm [10] is computationally less complicated, but it changes color tones and exhibits halo artifacts. He et al.’s algorithm [11] only considers the darkest pixel value for dehazing, and it thus removes the shadow of the cloud in Fig. 12(g). On the other hand, the proposed algorithm attempts to prevent the overflow, as well as the underflow, of pixel values during the dehazing procedure, as mentioned in Section 3.2. Therefore, the proposed algorithm can suppress most of the artifacts occurred in the conventional dehazing algorithms.

Next, in Fig. 13, we compare the proposed algorithm with He et al.’s algorithm [11] in more detail. In this test, to assess only the information loss cost without the effects of the contrast cost, we set $\lambda_L = \infty$ in (13) in the proposed algorithm. As shown in Fig. 13(b), He et al.’s algorithm estimates the atmospheric light from the brightest areas in the hazy images, i.e., the white building and the airplane. Thus, He et al.’s algorithm provides incorrect transmission maps in Fig. 13(c). On the other hand, the proposed algorithm estimates the atmospheric light more reliably within the red rectangles in Fig. 13(d), even though the input images include brighter objects than the atmospheric light. The proposed algorithm hence provides higher quality transmission maps in Fig. 13(e). Also, whereas the dark channel prior in He et al.’s algorithm only considers the truncation of dark pixels, the information loss cost in the proposed algorithm additionally prohibits the truncation of bright pixels. Consequently, the proposed algorithm provides more reliable dehazing performance than He et al.’s algorithm. For example, the proposed algorithm alleviates halo artifacts around the tail of the airplane more effectively in Fig. 13(d).

5.2. Video dehazing

We evaluate the performance of the proposed video dehazing algorithm on the “Riverside,” “Intersection,” and “Road View” sequences in Figs. 14–16. We set the parameter λ_T in (29) to 1 to strike a balance between flickering and blurring. We implement the proposed video dehazing algorithm in two different versions. First, it is implemented without the fast transmission refinement techniques, i.e. the partially overlapping sub-block scheme with the centered Gaussian windows and the downsampled computation of the transmission, in Section 4.3. Second, it is implemented with those techniques. Also, for comparison, we provide the results of the proposed static image dehazing algorithm and Zhang et al.’s

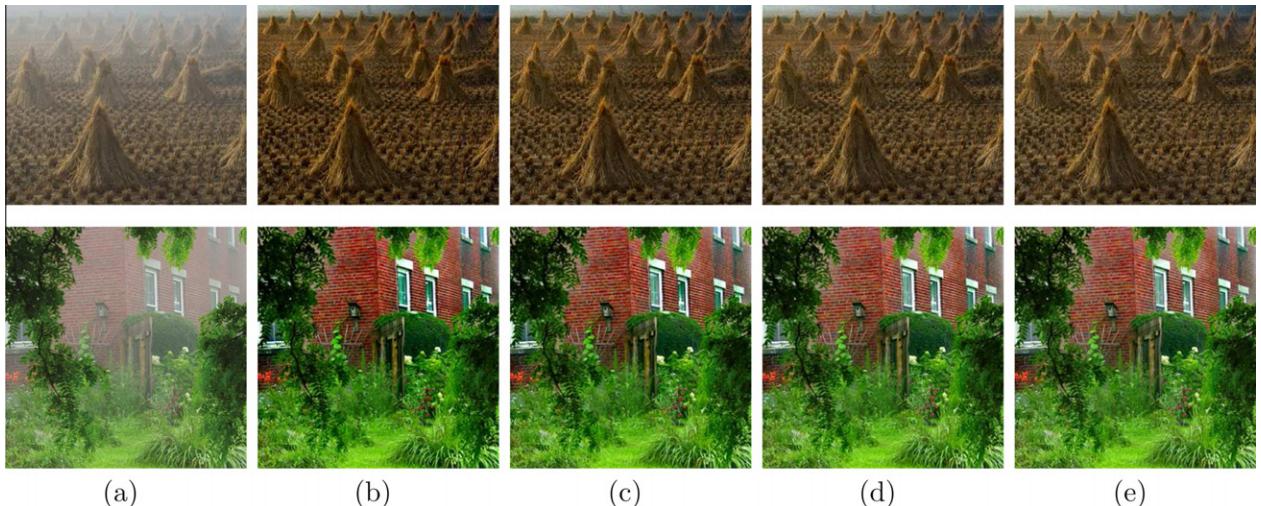


Fig. 9. Dehazing results on the “Cones” and “House” images, according to the parameter λ_L in the cost function in (13). (a) The input hazy images. The dehazed images at (b) $\lambda_L = 1$, (c) $\lambda_L = 2$, (d) $\lambda_L = 5$, and (e) $\lambda_L = 8$, respectively.

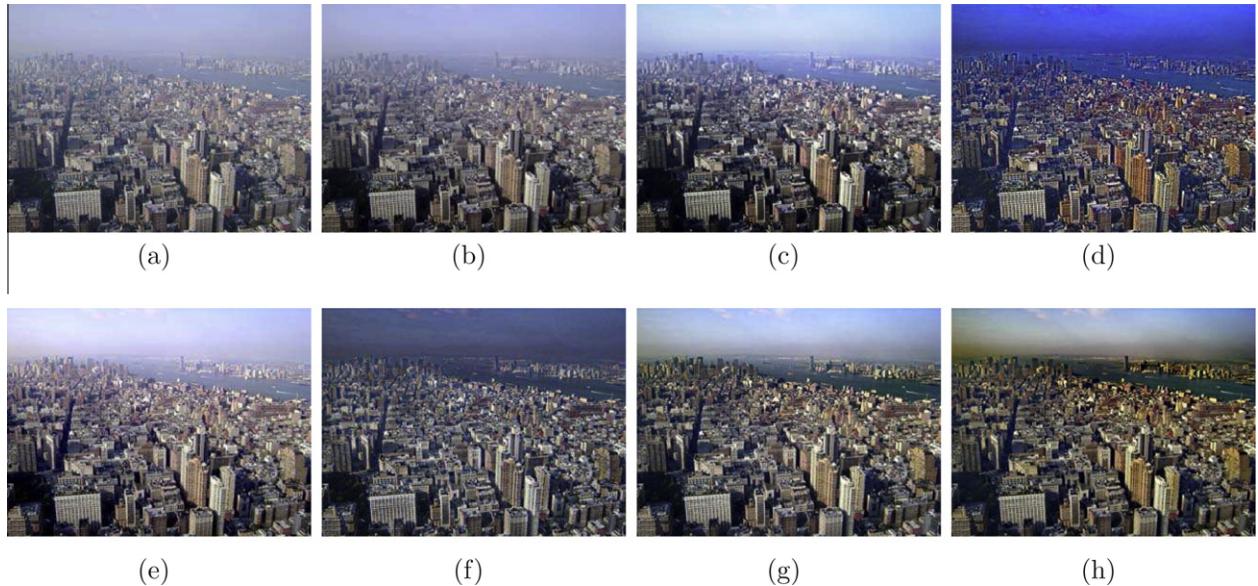


Fig. 10. Comparative results of the proposed algorithm and the conventional algorithms on the “Newyork1” image. (a) The input hazy image. The dehazed images obtained by (b) the level control method, (c) the histogram equalization method, (d) Tan’s algorithm [7], (e) Fattal’s algorithm [8], (f) Tarel et al.’s algorithm [10], (g) He et al.’s algorithm [11], and (h) the proposed algorithm.

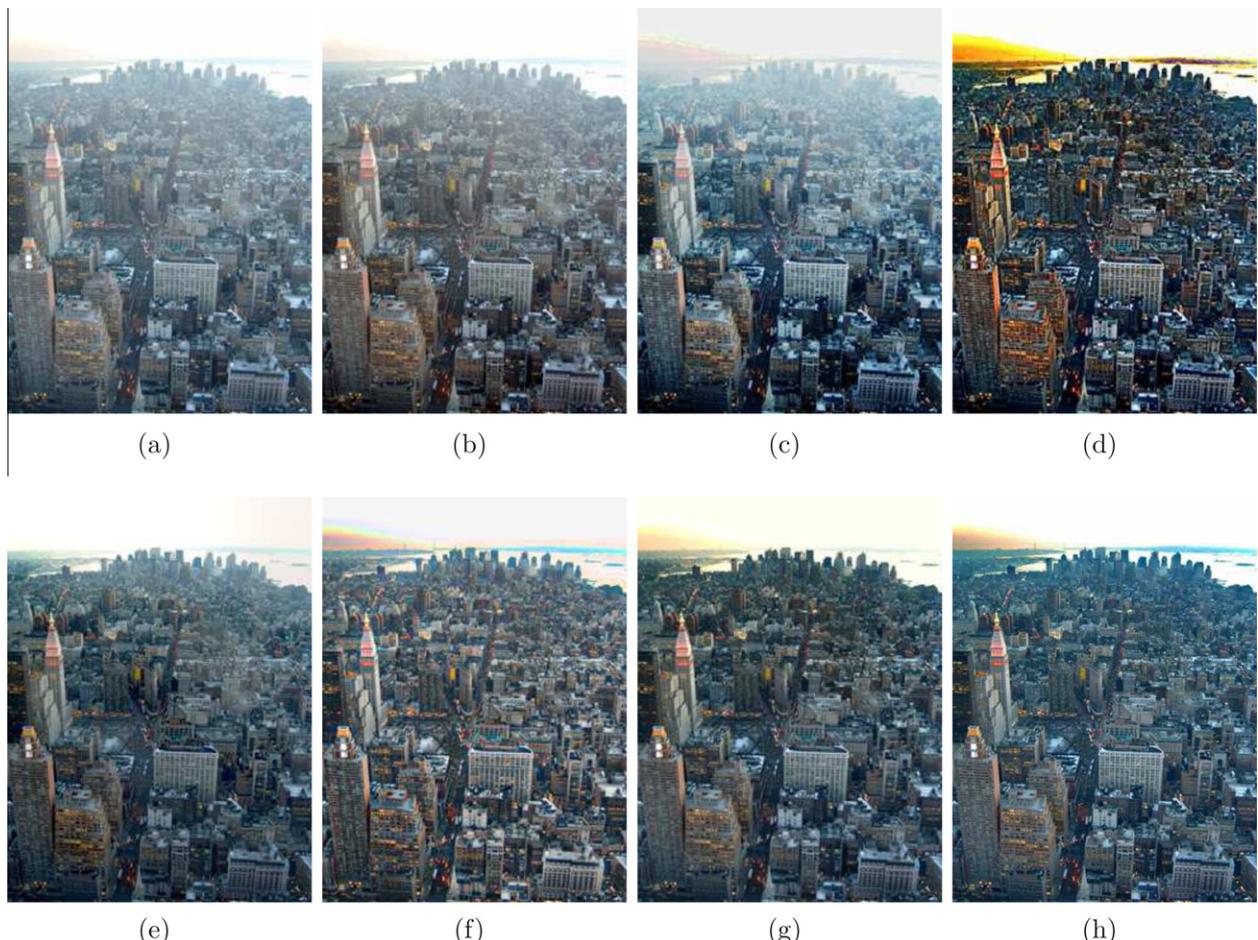


Fig. 11. Comparative results of the proposed algorithm and the conventional algorithms on the “Newyork2” image. (a) The input hazy image. The dehazed images obtained by (b) the level control method, (c) the histogram equalization method, (d) Tan’s algorithm [7], (e) Fattal’s algorithm [8], (f) Tarel et al.’s algorithm [10], (g) He et al.’s algorithm [11], and (h) the proposed algorithm.

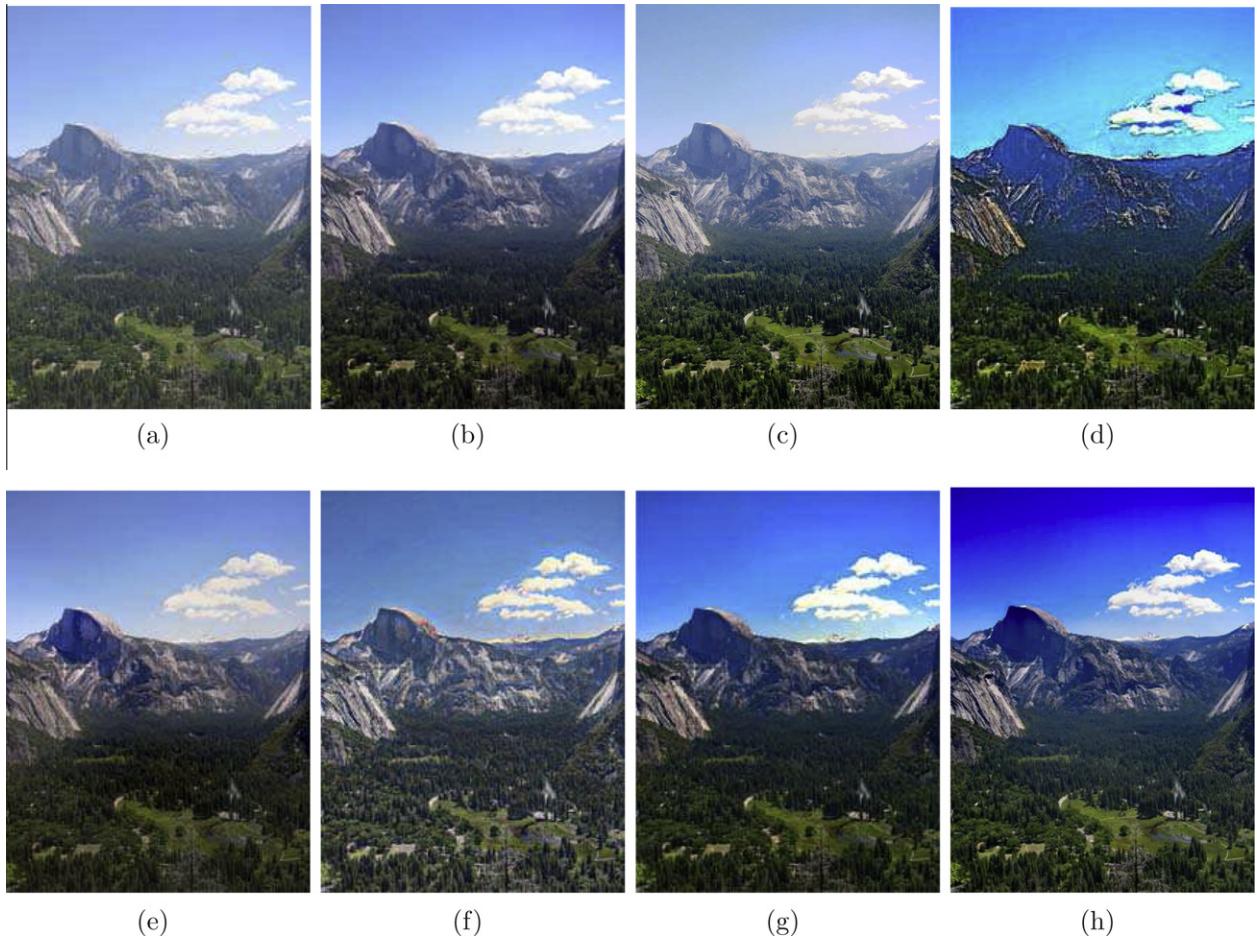


Fig. 12. Comparative results of the proposed algorithm and the conventional algorithms on the “Mountain” image. (a) The input hazy image. The dehazed images obtained by (b) the level control method, (c) the histogram equalization method, (d) Tan’s algorithm [7], (e) Fattal’s algorithm [8], (f) Tarel et al.’s algorithm [10], (g) He et al.’s algorithm [11], and (h) the proposed algorithm.

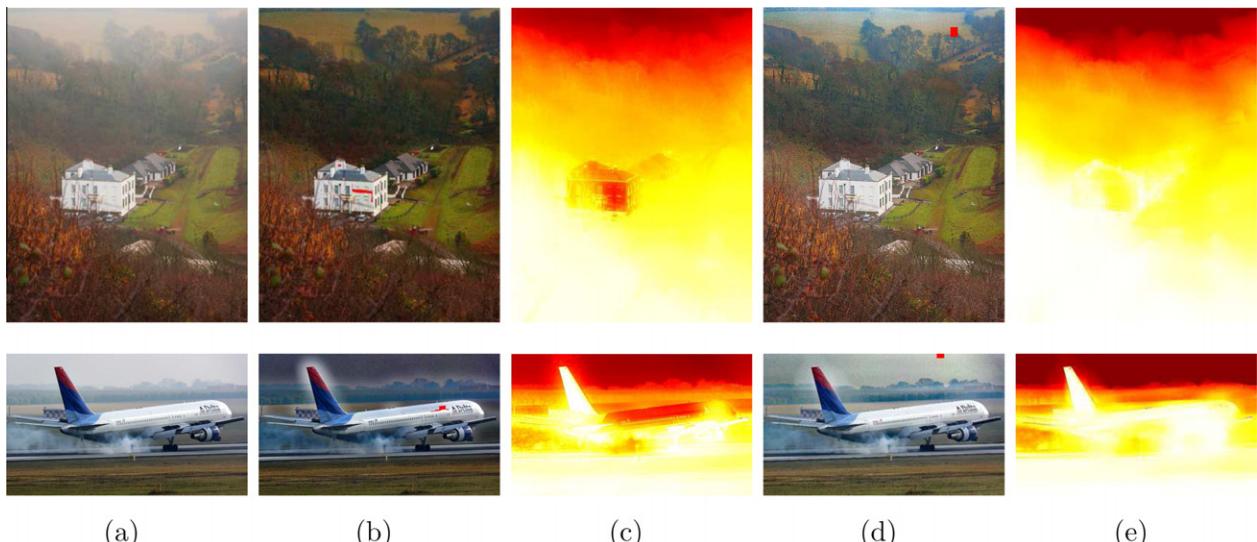


Fig. 13. The comparison of the proposed algorithm with He et al.’s algorithm [11]. (a) The input hazy images. (b) The dehazed images and (c) the transmission maps obtained by He et al.’s algorithm. The red areas in (b) represent the top 0.1% of the brightest pixels in the dark channels, in which the biggest pixel values are selected as the atmospheric light. (d) The dehazed images and (e) the transmission maps obtained by the proposed algorithm. The red rectangular areas in (d) are selected to determine the atmospheric light. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

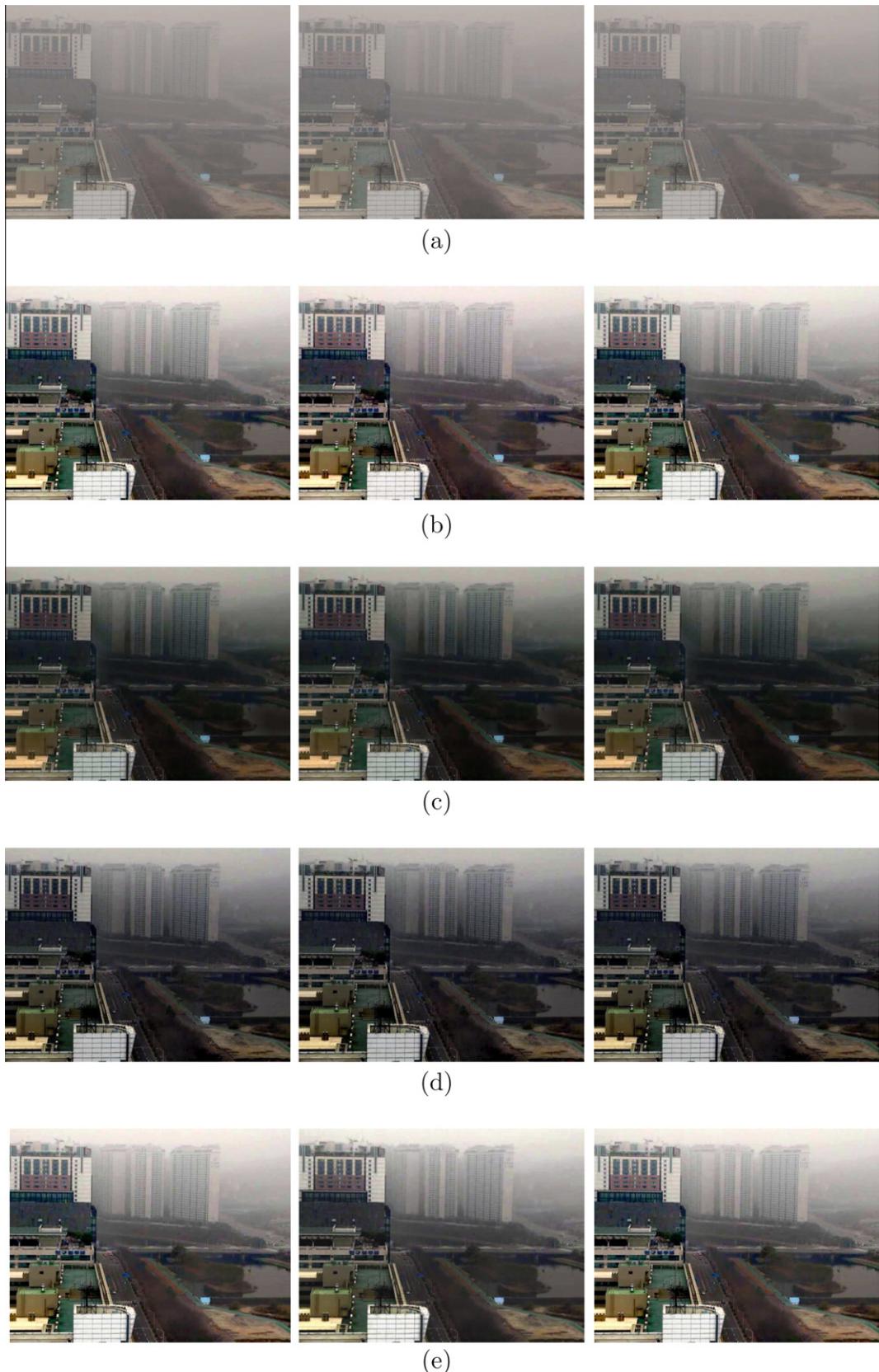


Fig. 14. Video dehazing on the “Riverside” sequence. (a) The input hazy sequence. The dehazed sequences by (b) the static image dehazing algorithm, (c) Zhang et al.’s algorithm [18], (d) the proposed algorithm without the fast transmission refinement, and (e) the proposed algorithm with the fast transmission refinement. The frame numbers of the left, middle, and right columns are 10, 12, and 14, respectively.

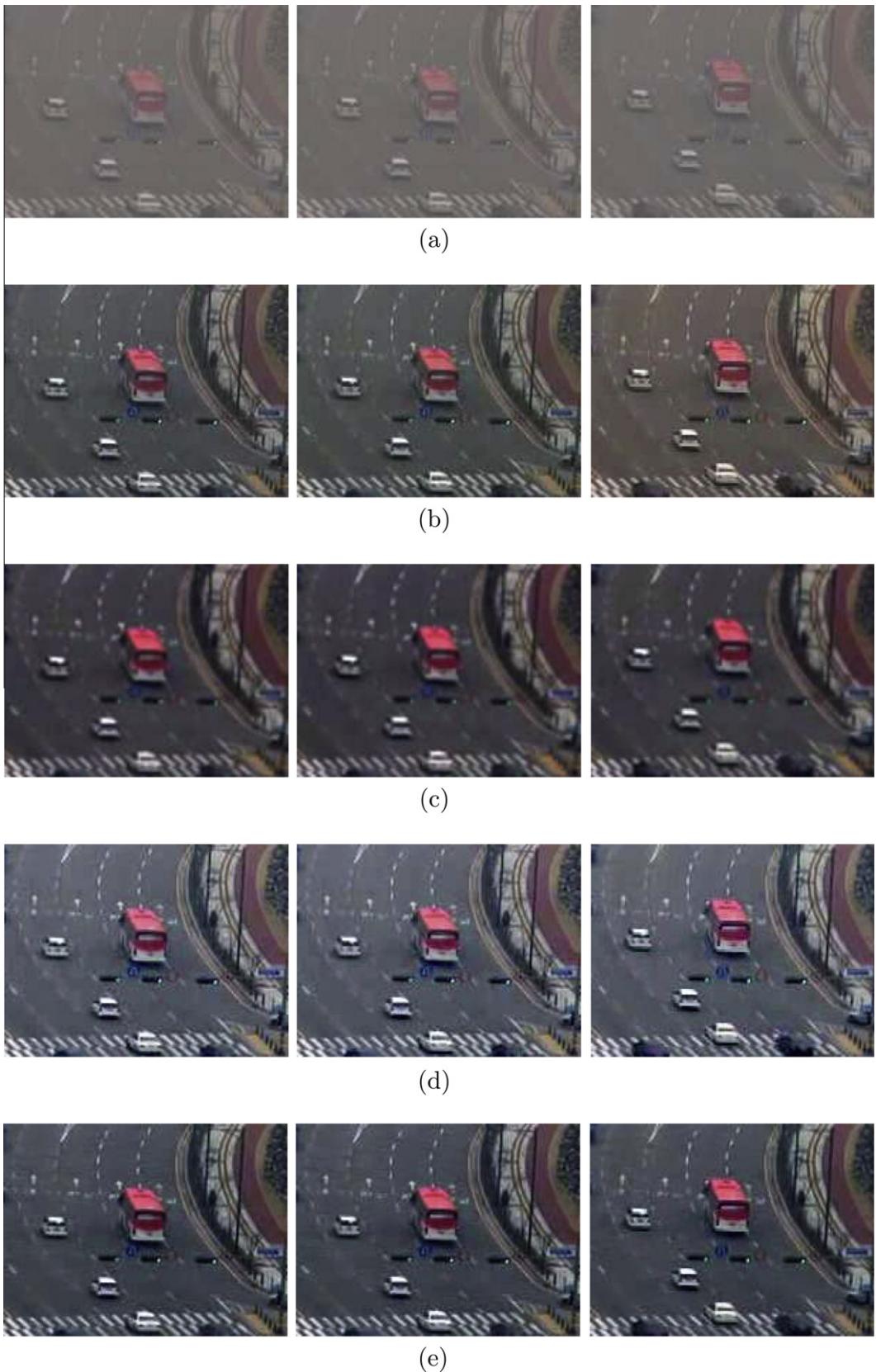


Fig. 15. Video dehazing on the “Intersection” sequence. (a) The input hazy sequence. The dehazed sequences by (b) the static image dehazing algorithm, (c) Zhang et al.’s algorithm [18], (d) the proposed algorithm without the fast transmission refinement, and (e) the proposed algorithm with the fast transmission refinement. The frame numbers of the left, middle, and right columns are 7, 8, and 14, respectively.

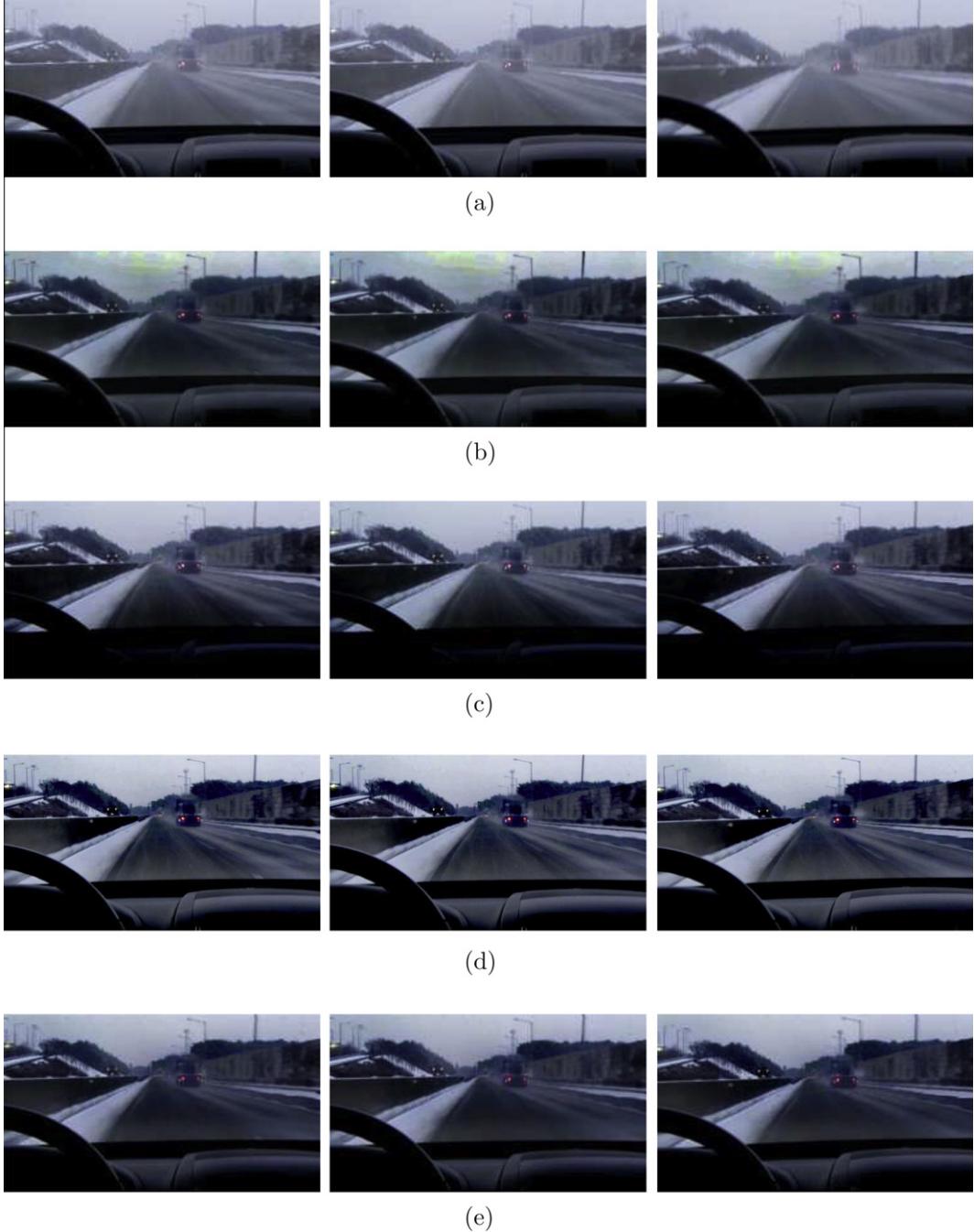


Fig. 16. Video dehazing on the “Road View” sequence. (a) The input hazy sequence. The dehazed sequences by (b) the static image dehazing algorithm, (c) Zhang et al.’s algorithm [18], (d) the proposed algorithm without the fast transmission refinement, and (e) the proposed algorithm with the fast transmission refinement. The frame numbers of the left, middle, and right columns are 7, 8, and 12, respectively.

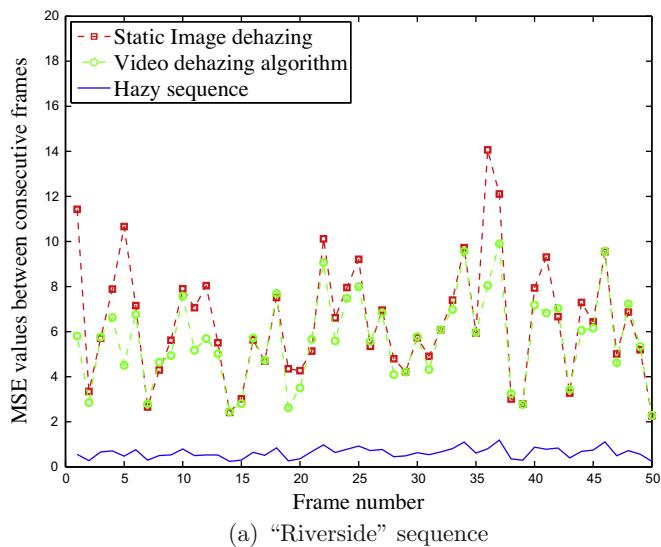
algorithm [18]. The static image dehazing algorithm is applied to each frame of the hazy sequences independently. It causes flickering artifacts in Figs. 14(b), 15(b) and 16(b), due to the variations of the estimated atmospheric light among frames. The small differences in the atmospheric light are amplified with low transmission values, which severely change the color tones of restored frames. On the contrary, Zhang et al.’s algorithm and the proposed video dehazing algorithm yield temporally coherent dehazing results by suppressing flickering artifacts. However, Zhang et al.’s algorithm is essentially based on He et al.’s static image dehazing algorithm [11]. Therefore, it may cause the overflow of pixel values. We see that the proposed algorithm removes haze more effectively and

naturally than Zhang et al.’s algorithm, especially in Figs. 14 and 15. Also, note that the proposed algorithm with the fast refinement techniques provides faithful output images in Figs. 14(e), 15(e) and 16(e), whose qualities are comparable to those of the images in Figs. 14(d), 15(d) and 16(d) without the fast techniques.

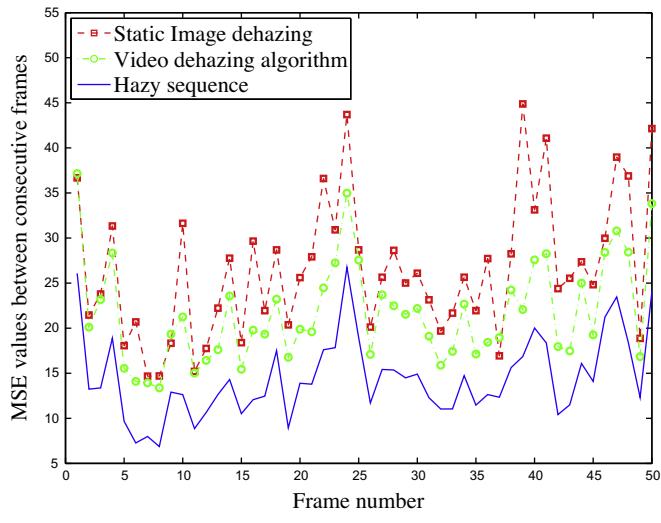
Zhang et al.’s algorithm demands high memory and computational complexities, since it uses the information in at least three frames to estimate the transmission map of a frame. On the contrary, the proposed algorithm requires only the information in the previous frame. We test the complexity of the proposed algorithm using a personal computer with an Intel Core i5-2500K processor and 4 GB memory. When the proposed algorithm is

implemented without the fast refinement techniques, it provides the processing speeds of 7.5, 7.6, and 8.1 frames per second (fps) on “Riverside,” “Intersection,” and “Road View,” respectively. We improve the speeds by employing the fast refinement techniques. Moreover, we use parallel programming tools, SIMD [31] and OpenMP [32], for faster computation. We use the SIMD in the transmission refinement step, by performing the computation for four pixels in parallel, and apply the OpenMP to restore pixel values using four processor cores in parallel. Consequently, the proposed algorithm with the fast transmission refinement achieves real-time video dehazing and performs at 31.8, 36.8, and 46.1 fps on “Riverside,” “Intersection,” and “Road View,” respectively. However, this complexity is still too high to be employed in applications with limited computing resources, such as car vision. Further complexity reduction is one of the future research issues.

Next, we quantitatively show how the proposed video dehazing algorithm suppresses flickering artifacts by employing the temporal coherence cost in the optimization. Fig. 17 plots the MSE between two consecutive frames in the “Riverside” and “Road View” sequences. When the static image dehazing algorithm is independently applied to each frame, the MSE curves experience



(a) “Riverside” sequence



(b) “Road View” sequence

Fig. 17. Comparison of the MSE's between consecutive frames. The proposed video dehazing algorithm causes less fluctuations than the static image dehazing algorithm, which is independently applied to each frame.

Table 1

Comparison of the average temporal deviation measurements. A smaller temporal deviation indicates less flickering artifacts.

	Input sequence	Static image dehazing	Proposed video dehazing
“Riverside”	0.3078	2.4386	2.2079
“Road View”	6.5611	7.0752	5.1355

relatively large fluctuations as compared with the input hazy sequences, especially between 35–45 frames in Fig. 17(a) and 5–10 frames and 35–45 frames in Fig. 17(b). These fluctuations are caused by abrupt changes in color tones between consecutive frames, which result in flickering artifacts. On the other hand, the video dehazing algorithm alleviates the fluctuations and reduces the flickering artifacts efficiently.

We also quantify the flickering artifacts based on the sensitivity model of flickering perception [33]. The original flicker sensitivity function in [33] measures the flickering of a temporal sinusoidal signal, and it cannot be directly applied to assess the flickering in a dehazed video. Thus, similarly to [34], we modify their sensitivity function as follows. At each pixel position, we extract a temporal sequence of pixel values through a video clip, filter the sequence based on the human perception model as in [33], and then compute the temporal standard deviation of the filtered sequence. A smaller temporal deviation indicates less flickering artifacts in general. Hence, we employ the average temporal deviation over all pixel positions as a measure of flickering artifacts. Table 1 compares the average temporal deviation measurements. Note that the proposed video dehazing algorithm exhibits smaller average temporal deviations than the static image dehazing algorithm.

We make the dehazing results available as video clips at our project website,¹ so that the reduction of flickering artifacts can be assessed subjectively. Moreover, we provide more dehazing results on other images and videos on the website. These experimental results also confirm that the proposed algorithm is a promising technique for dehazing.

6. Conclusions

In this work, we proposed a dehazing algorithm based on the optimized contrast enhancement. The proposed algorithm first selects the atmospheric light in a hazy image using the quadtree-based subdivision. Then, since a hazy image has low contrast, the proposed algorithm determines transmission values, which are adaptive to scene depths, to increase the contrast of the restored image. However, some pixels in the restored image can be saturated, resulting in information loss. To overcome this issue, we incorporated the information loss cost into the optimized transmission computation. We also extended the static image dehazing algorithm to the real-time video dehazing algorithm, by employing the temporal coherence cost. Experimental results demonstrated that the proposed algorithm is capable of removing haze effectively and restoring images faithfully, as well as achieving real-time processing.

Acknowledgments

The work of J.-H. Kim, W.-D. Jang, and C.-S. Kim was supported partly by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-011031), and partly by Basic Science Research Program through the NRF of Korea funded by the MEST (No. 2012-0000916). The work of J.-Y. Sim was

¹ <http://mcl.korea.ac.kr/projects/dehazing/>.

supported by Basic Science Research Program through the NRF of Korea funded by the MEST (2010-0006595).

References

- [1] R. Gonzalez, R. Woods, *Digital Image Processing*, third ed., Prentice-Hall, 2007.
- [2] S. Narasimhan, S. Nayar, Vision and the atmosphere, *Int. J. Comput. Vis.* 48 (3) (2002) 233–254.
- [3] S. Narasimhan, S. Nayar, Contrast restoration of weather degraded images, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (6) (2003) 713–724.
- [4] S. Shwartz, E. Namer, Y. Schechner, Blind haze separation, in: Proc. IEEE CVPR, 2006, pp. 1984–1991.
- [5] Y. Schechner, S. Narasimhan, S. Nayar, Instant dehazing of images using polarization, in: Proc. IEEE CVPR, 2001, pp. 325–332.
- [6] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, D. Lischinski, Deep photo: model-based photograph enhancement and viewing, *ACM Trans. Graph.* 27 (5) (2008) 1–10.
- [7] R. Tan, Visibility in bad weather from a single image, in: Proc. IEEE CVPR, 2008, pp. 1–8.
- [8] R. Fattal, Single image dehazing, *ACM Trans. Graph.* 27 (3) (2008) 1–9.
- [9] L. Kratz, K. Nishino, Factorizing scene albedo and depth from a single foggy image, in: Proc. IEEE ICCV, 2009, pp. 1701–1708.
- [10] J. Tarel, N. Hautière, Fast visibility restoration from a single color or gray level image, in: Proc. IEEE ICCV, 2009, pp. 2201–2208.
- [11] K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (12) (2011) 1956–1963.
- [12] C. Ancuti, C. Hermans, P. Bekaert, A fast semi-inverse approach to detect and remove the haze from a single image, in: Proc. ACCV, 2011, pp. 501–514.
- [13] J. Yu, Q. Liao, Fast single image fog removal using edge-preserving smoothing, in: Proc. IEEE ICASSP, 2011, pp. 1245–1248.
- [14] P. Carr, R. Hartley, Improved single image dehazing using geometry, in: Proc. DICTA, 2009, pp. 103–110.
- [15] L. Schaul, C. Fredembach, S. Süstrunk, Color image dehazing using the near-infrared, in: Proc. IEEE ICIP, 2009, pp. 1629–1632.
- [16] X. Dong, X. Hu, S. Peng, D. Wang, Single color image dehazing using sparse priors, in: Proc. IEEE ICIP, 2010, pp. 3593–3596.
- [17] J. Tarel, A. Hautière, A. Cord, D. Gruyer, H. Halmaoui, Improved visibility of road scene images under heterogeneous fog, in: Proc. IEEE Intelligent Vehicles Symposium (IV), 2010, pp. 478–485.
- [18] J. Zhang, L. Li, Y. Zhang, G. Yang, X. Cao, J. Sun, Video dehazing with spatial and temporal coherence, *Vis. Comput.* 27 (6) (2011) 749–757.
- [19] J. Oakley, H. Bu, Correction of simple contrast loss in color images, *IEEE Trans. Image Process.* 16 (2) (2007) 511–522.
- [20] J.-H. Kim, J.-Y. Sim, C.-S. Kim, Single image dehazing based on contrast enhancement, in: Proc. IEEE ICASSP, 2011, pp. 1273–1276.
- [21] J.-H. Kim, W.-D. Jang, Y. Park, D.-H. Lee, J.-Y. Sim, C.-S. Kim, Temporally coherent real-time video dehazing, in: Proc. IEEE ICIP, 2012, pp. 969–972.
- [22] E. Peli, Contrast in complex images, *J. Opt. Soc. Am. A* 7 (10) (1990) 2032–2040.
- [23] A. Michelson, *Studies in Optics*, Dover Publications, 1995.
- [24] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proc. IEEE ICCV, 1998, pp. 839–846.
- [25] K. He, J. Sun, X. Tang, Guided image filtering, in: Proc. ECCV, 2010, pp. 1–14.
- [26] R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science, Springer, 2010.
- [27] B.-L. Yeo, B. Liu, Rapid scene analysis on compressed video, *IEEE Trans. Circ. Syst. Video Technol.* 5 (6) (1995) 533–544.
- [28] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. International Joint Conference on Artificial Intelligence (IJCAI), 1981, pp. 674–679.
- [29] J.-Y. Kim, L.-S. Kim, S.-H. Hwang, An advanced contrast enhancement using partially overlapped sub-block histogram equalization, *IEEE Trans. Circ. Syst. Video Technol.* 11 (4) (2001) 475–484.
- [30] Photoshop, CS5, Adobe systems, San Jose, CA, 2010.
- [31] D. Patterson, J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, fourth ed., The Morgan Kaufmann Series in Computer Architecture and Design, Elsevier Morgan Kaufmann, 2009.
- [32] B. Chapman, G. Jost, R. Pas, *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*, The MIT Press, 2007.
- [33] J. Rovamo, K. Donner, R. Näsänen, A. Raninen, Flicker sensitivity as a function of target area with and without temporal noise, *Vis. Res.* 40 (28) (2000) 3841–3851.
- [34] C.-Y. Hsu, C.-S. Lu, S.-C. Pei, Temporal frequency of flickering-distortion optimized video halftoning for electronic paper, *IEEE Trans. Image Process.* 20 (9) (2011) 2502–2514.