MAPD: An FPGA-Based Real-Time Video Haze Removal Accelerator Using Mixed Atmosphere Prior

Yanjie Tan, Yifu Zhu, Zhaoyang Huang, Huailiang Tan, and Keqin Li, *Fellow, IEEE*

***Abstract*—Real-time video dehazing plays a key role in helping autonomous driving detect pedestrians or obstacles in severe foggy weather to prevent potential hazards. Existing video dehazing methods achieve good restoration performance but still suffer from oversaturation and low dehazing speed, especially for high-definition (HD, high-resolution) videos. In this article, we propose a mixed atmosphere prior information video dehazing accelerator (MAPD) and implement it on field programmable gate array (FPGA) to achieve real-time haze removal for HD video. MAPD provides a mixed atmospheric light model by applying heterogeneous atmospheric light in the foreground area to balance brightness deviation, and maintaining the global atmospheric light in the background region. Considering the parallel characteristics of FPGA, MAPD leverages the redundant information between adjacent frames to accelerate the dehazing process and designs an indirect transmission estimation to decrease resource consumption. For comparison, we also implement six dehazing solutions (DCP, color ellipsoid prior (CEP), RDCP, FFVD, MHVD, and REFD) on FPGA, and deploy a graphics processing unit (GPU)-based method (D^4) on the platform with Nvidia 3080 GPU. Experiments using two widely used benchmarks show that MAPD increases the performance by up to 36.5%, 53.5%, 36.3%, 33.3%, 11.9%, and 23.3%, decreases resource consumption by up to 79.7%, 75.0%, 74.8%, 25.6%, 22.6%, and 73.9% and enhances FPS for HD videos by up to 241.6%, 145.9%, 151.7%, 68.6%, 50.6%, and 62.4%, compared with DCP, CEP, RDCP, FFVD, MHVD, and REFD. Compared to D^4, MAPD also promotes the dehazing performance by up to 21.8%, and increases FPS by up to 487.0%.**

***Index Terms*—Autonomous driving, field programmable gate array (FPGA), mixed atmosphere prior, real-time, video dehazing.**

I. Introduction

AUTONOMOUS vehicle technology, which plays an important role in helping drivers avoid collision [1] and reduce accidents to save human lives [2], has now become a frontier hot spot because of the advances in automobile intelligence and advanced driver assistant systems. However, unfavorable weather such as fog, which will blur or occlude objects and humans on the road, leads to a grand challenge for the stereo vision system of autonomous vehicles [3]. The blurred video images may mislead stereo vision system modules (such as object detection [4] or object tracking [5]) to make incorrect judgments for potential dangers or sudden situations, and lead to traffic accidents [6]. The figures in Section V-C show the influence of fog weather on object detection. From the first image of Fig. 9(c), it can be observed that two two-wheelers are omitted by the object detection module. This omission perhaps leads the autonomous vehicle system to make a decision to evade to the right and causes an accident at such a close distance. Besides, the loss of contrast and details in the video frame also seriously degrade the performance of other Internet of things (IoT) devices like traffic surveillance cameras [7], smart parking monitors [8], and face recognition terminals [9]. Therefore, a real-time video haze removal method can effectively improve the performance and accuracy of IoT systems (e.g., autonomous vehicle systems) to reduce the number of accidents.

In recent years, there have been several algorithms for single image dehazing proposed to solve the problems caused by haze or fog. They can be categorized into three aspects, i.e., dehazing method based on image enhancement [10], [11], neural network dehazing [12], [13], [14], [15], [16], [17], and prior-based image dehazing algorithms [18], [19], [20], [21]. Wang and Zhou [10] used Fourier and exponential transform to remove image shadow caused by uneven illumination and strengthen image local details. However, these transformation techniques are not suitable for low-luminosity images. Jobson et al. [11] proposed an algorithm called Retinex to enhance fog-degraded images by using color constancy of objects. Unfortunately, Retinex will weaken the original details of the image because of its color range compression through logarithmic transformation. Moreover, these image enhancement dehazing methods ignore the essence of fog image degradation which leads to a lack of dehazing effect and robustness. To get a more accurate clear-day image, researchers begin to use a neural network to process the fog-degraded image. Cai et al. [12] proposed DehazeNet to estimate transmission, and get a remarkable improvement. MSCNN [13] modeled a multiscale deep network structure to estimate the scene transmission and suppress halo artifacts of

Manuscript received 13 November 2022; revised 1 March 2023 and 8 May 2023; accepted 22 June 2023. Date of publication 3 July 2023; date of current version 22 November 2023. This work was supported in part by the Natural Science Foundation of Hunan Province (Image Haze Removal Scheme based on Deep Neural Networks), China, under Grant 2020JJ4216, and in part by the Natural Science Foundation of Changsha, China, under Grant kq2202173. This article was recommended by Associate Editor W. Zhang. *(Corresponding author: Huailiang Tan.)*

Yanjie Tan, Yifu Zhu, Zhaoyang Huang, and Huailiang Tan are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: tanyanjie@hnu.edu.cn; tanhuailiang@hnu.edu.cn).

Keqin Li is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TCAD.2023.3291670

dehazed images. PMS-Net [14] used the convolutional neural network (CNN) to generate the patch map from fog-degraded images and selects the patch size dynamically which can improve the performance in haze removal. However, these neural network dehazing methods have several limitations. First, for a specific scene, the given training set can achieve superior performance, but in other scenarios, the haze removal accuracy drops dramatically. Second, these neural network methods (such as CNN) cannot meet the requirement of real-time processing since the parameters of each layer have to be learned from training [22]. Compared with the two categories of methods mentioned before, prior-based image dehazing algorithms are more widely used because of less computation overhead. Tan [18] used the prior information that the contrast of haze degraded image is higher, and proposed that maximizing the local contrast of haze degraded image can restore haze image. But the restored image tends to be supersaturated and is prone to have halo effects if the depth of field changes suddenly. He et al. [20] proposed a single image haze removal algorithm based on dark channel prior information to recover a clear-day image and obtained significant effects for indoor images. However, the dehazing performance for outdoor images is supersaturated. Bui and Kim [21] used color ellipsoid prior (CEP) information to dehaze a single image by embedding a fuzzy segmentation process to calculate the transmission and perform the refinement process at the same time. But it will lead to visible color artifacts due to overdehazing of the input hazy image. Moreover, all these single image dehazing methods do not fully take into account the feature of abundant correlation information between video frames so that they are hard to satisfy the real-time requirement for video dehazing.

Unlike the single image dehazing approaches, video dehazing is much more complicated [23]. To achieve the goal of real-time video haze removal, we not only need to make full use of the correlation information between video frames to accelerate the process but also cannot ignore the situation of sharp changes in brightness caused by video scene switching. Xu et al. [23] proposed a contrast-limited adaptive histogram equalization-based (CLAHE) approach to clearing the fog from video sequences by fusing the foreground and background images into new frames, while this image enhancement method still neglects the nature of hazy images. Shiau et al. [22] implemented a haze removal algorithm based on the dark channel prior for multiple continuous dynamic image systems. It is also applied to video dehazing by designing a dynamic adjustment strategy for atmospheric light which addresses flicker issues in videos. However, due to the excessive pursuit of brightness, the image pixels are oversaturated. Kumar et al. [24] designed a high-speed very-large-scale integration (VLSI) architecture for video defogging to reduce the power and memory requirements, but their two-stage process for airlight estimation cannot obtain optimal results.

As the preprocessing step of object detection in autonomous driving systems [22], the real-time processing capability of video defogging has become another crucial factor in avoiding potential collisions and accidents. It means more time can be left for other visual perception systems to deal with sudden situations. For example, when a vehicle is traveling at 40 km/h, the total braking distance is 23.1 m if the video dehazing time is 1 s while the distance is decreased to 12.21 m if the video defogging time is lowered to 10 ms. This reduction in distance reduces the chance of collision from nearly 38% to less than 3% based on the computation model of this article [25] (assuming the distance to the obstacle is 50 m). Park and Kim [26] proposed a fast video dehazing system using field programmable gate array (FPGA) that simplifies the atmospheric light solution of the original dark channel algorithm. Majoka and Raja [27] realized a low-delay dehazing system by introducing a two-stage architecture. The work in [22] and [24] leveraged VLSI architecture to achieve real-time video defogging. However, these video haze removal methods are based on the theory of homogeneous atmospheric light prior, which is a simplified model but is not applicable to real-world scenarios, so that the recovery image is inclined to supersaturate or over enhance.

In this article, we propose a new FPGA-based video dehazing accelerator, namely, FPGA-based mixed atmosphere prior information video dehazing accelerator (MAPD), effectively exploiting the hardware parallelization characteristics and the redundant information between adjacent frames for decomposing the video dehazing process, achieving real-time video dehazing and improving dehazing performance. MAPD performs parallel computation on input video frames by dividing independent hardware modules. First, a mixed atmospheric light model is proposed to utilize heterogeneous atmospheric light and global atmospheric light for the foreground and background regions of frames, respectively. Second, unlike the traditional transmission estimation, we combine the hardware characteristics to design an indirect transmission approximation module that can effectively decrease resource consumption. Third, a haze-free video reconstruction module is implemented to gather the mixed atmospheric light and the indirect transmission for the final clear-day output frame. Finally, a program system module is designed to manage the accelerator.

The main contributions of our work are as follows.

1) We propose a new FPGA-based video dehazing accelerator, called MAPD, which takes full advantage of the hardware parallelization characteristics and the redundant information between adjacent frames to efficiently dehazing the foggy videos and decrease dehazing time. It can satisfy the demand for real-time systems and obtain high defogging performance.
2) We design a mixed atmospheric light model to calculate the atmospheric light accurately. We switch the estimation for the foreground region with more complex brightness distributions to the heterogeneous atmospheric light model and keep the global atmospheric light for the background region where brightness varies less. By leveraging the correlation of adjacent frames, we parallelize the estimation on FPGA so that it can finish in one iteration.
3) We explore an indirect transmission approximation module by combining the hardware characteristics to reduce

unnecessary division operations which will lead to extra resource consumption in FPGA.

4) We conduct a series of evaluations on real-world hazy video datasets to compare MAPD with six FPGA-based solutions (DCP, CEP, RDCP, FFVD, MHVD, and REFD) and a graphics processing unit (GPU)-based method (D^4) to demonstrate the effectiveness of our work. For comparison, the first six methods are also implemented in the same FPGA platform and D^4 is evaluated with the Nvidia 3080 GPU. The experimental results show that MAPD has superior dehazing performance and achieves real-time, 244 FPS for a 480p video and 39 FPS for a 1080p video.

The remainder of this article is organized as follows. Section II introduces background and motivation. Section III illustrates a detailed design of MAPD and the relevant algorithms. Section IV describes the hardware implementation of MAPD. Section V presents the evaluation results. Section VI introduces related work. In Section VII, we summarize our results and provide a number of conclusions. Finally, Section VII provides the acknowledgments.

II. Background and Motivation

A. Atmospheric Scattering Model

To describe the formation of fog in a blurred fog image, Koschmieder [28] developed a simplified atmospheric scattering model. Most haze removal algorithms are derived from this model, and it can be expressed as follows [29]:

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

where x is the coordinate of the pixel. $J(x)$ represents the clear-day image while $I(x)$ means the fog-degraded image. A is the global atmospheric light. These three variables are all 3-D vectors in RGB (RED, GREEN, BLUE) space. $t(x)$ means the transmission coefficient, which indicates the fog density of a hazy image. However, in this equation, $I(x)$ is the only known variable whose value is much less than that of unknowns, i.e., this is an underdetermined equation. In order to solve this equation, some additional constraints or priors have to be obtained. With this priori information, the haze-free image $J(x)$ could be restored by estimating the global atmospheric light A and the transmission $t(x)$.

B. Fast Guided Filtering and Side Window Filtering

Fast guided filtering (FGF) is an accelerated version of guided filtering (GF) which is used to smooth the input image while keeping the edges at the same position as in the guidance image. According to the GF theory, the filtering output image J can be expressed as follows:

$$J(x) = \overline{a}_x G(x) + \overline{b}_x \tag{2}$$

where $G(x)$ is the guidance image, and $\overline{a}_x$ and $\overline{b}_x$ are the average linear coefficients of all local patches overlapping x which can be computed by a mean filter, respectively. Since the main computation is a series of box filters, the time complexity of GF is $O(N)$. Compared with the original GF algorithm, FGF speeds up the time from $O(N)$ to $O(N/s^2)$ by subsampling the input image and the guidance image with a subsampling ratio s and then upsampling to the original size after the computation of box filters finish.

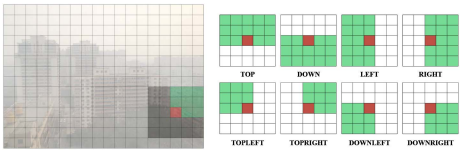

Fig. 1. Side window filter. The green rectangles present the candidate templates and the red ones mean the target pixel.

Unlike most traditional filters that align the center of the window with the pixels being processed, the side window filtering (SWF) uses its corner or side instead to improve the edge preserving capability of the image significantly. As shown in Fig. 1, SWF preserves the edge information by computing eight candidate templates of the target pixel and then picking the one with the minimal distance from the pixel to be processed. More detail information about FGF and SWF can be found in [30] and [31], respectively.

C. Motivation

Although the current video dehazing methods have gained decent performance, there still exist some issues that need to be addressed.

First, most of them are based on the assumption of the homogeneous atmospheric scattering model, while this simplified model is not suitable for the atmospheric light distribution in reality because of the diverse composition of the atmosphere [12], [32]. Due to the uneven distribution of the atmospheric particle, estimating a global atmospheric light A for the whole image will lead to color distortion. Fig. 2(a)–(d) (i.e., DCP [20], CEP [21], RDCP [33], and MHVD [24]) demonstrates this phenomenon, especially for the sky regions in these figures (more details will be discussed in Section V). Shiau et al. [22] proposed an optimized global atmospheric light estimation by calculating an atmospheric light value for the foreground and background areas, respectively, and assigning different weights for them. However, its final atmospheric light is still limited to the global value for the entire image so that the restored image tends to be supersaturated, as shown in Fig. 2(e) (i.e., FFVD). Du et al. [34] proposed a fusion defogging method based on gray-scale and linear transmission to produce a more brightly restored image, but it also had the side effect of quality noise that severely blurs the visual effect of images, as shown in Fig. 2(f) (i.e., REFD). In the area of GPU-based dehazing solutions, Yang et al. [15] designed a self-augmented image dehazing framework to improve the dehazing performance by decomposing the estimation of transmission into the prediction for density and depth map, while its estimation for the atmospheric light still followed the homogeneous atmospheric scattering model. Besides, it also suffered from the overestimation of transmission for extremely bright areas, leading to a supersaturated recovered result, as shown in Fig. 2(g) (i.e., D^4). Based on the above experimental results and analysis, we propose a new mixed atmosphere scattering prior algorithm that accurately calculates the atmospheric light,

Fig. 2. Dehazing performance on reality images. (a) DCP. (b) CEP. (c) RDCP. (d) MHVD. (e) FFVD. (f) REFD. (g) D^4. (h) Our MAPD.

TABLE I
DEHAZING TIME OF A 1080P VIDEO FRAME FOR DIFFERENT METHODS

Method	DCP	CEP	RDCP	FFVD	MHVD	REFD
Time on FPGA (ms)	87.1	62.7	64.9	42.99	38.3	41.4

which presents a superior dehazing performance, as shown in Fig. 2(h). More evaluation details are illustrated in Section V.

Second, with the improved quality of video captured by cameras in autonomous driving systems, it will take a longer time to dehaze the frames in the video, which makes it harder to meet real-time requirements. Table I represents the dehazing time of a frame in a 1080p (1920×1080) video for six FPGA-based solutions (DCP, CEP, RDCP, FFVD, MHVD, and REFD). It can be seen that even with FPGA acceleration, it takes at least 38 ms (26 FPS) to dehaze a frame of video. If the video resolution is increased to 2K (2560×1440), then the frame rate will be less than 14 fps, which obviously cannot satisfy the demand for real-time dehazing systems. Therefore, we propose a new FPGA-based video haze removal accelerator by implementing our mixed atmosphere scattering prior algorithm on FPGA to achieve real-time video dehazing.

III. DESIGN

In this section, we describe the MAPD design, which leverages the parallelization characteristics of FPGA by designing several separate modules to efficiently dehaze the fog-degraded video with lower time consumption.

A. MAPD Overview

As shown in Fig. 3, the processing system (PS) module is responsible for video frame buffer control and dehaze parameter initialization. The programmable logic (PL) module is the main component of MAPD, which consists of three submodules, namely, mixed atmospheric light module (CalM_Module), transmission calculation module (CalT_Module), and image reconstruction module (RecJ_Module), respectively. The hazy video frame $I_n(x)$ is the input and the hazy-free video frame is obtained as the output. In CalM_Module, the heterogeneous atmospheric light $A_n(x)$ and the global atmospheric light A_n^{back} are calculated first, then they are combined based on the mixed atmosphere scattering prior information to generate the final coarse-grained atmosphere map $M_n(x)$. In order to eliminate the block effect,

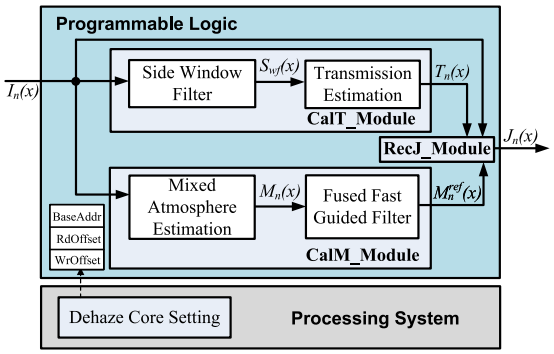

Fig. 3. Accelerator hardware circuit. The accelerator can be divided into two main parts, which consist of the processing system module and the PL module.

the final atmospheric light map $M_n^{\text{ref}}(x)$ will be refined from $M_n(x)$ by performing a fused fast guided filter operation that is slightly modified from the original filter algorithm to reduce time consumption. In the CalT_Module, considering that hardware implementation needs to make full use of the parallel characteristics of FPGA, we first approximate the transmission by the side window filter and then estimate an indirect transmission map $T_n(x)$ with the fixed transformation parameter. More details will be introduced in the following sections.

B. Mixed Atmospheric Light Model

As we described in Section II, atmospheric light plays a significant role in haze removal to recover a clear-day image. FFVD, which is another FPGA-based video defogging method, optimizes the estimation of global atmospheric light by using a weighting function. It first picks a subglobal atmospheric light value for the foreground portion and the background portion, respectively, and then weights these two values into a new global atmospheric light. Although it makes the haze-free images look brighter, there exists oversaturation in some image pixels because of the unnecessary weighting for these areas. The essential reason is that this article [22] still employs the global value for the entire image, while the atmospheric particle distribution is uneven in reality. We further observe that the grayscale distribution of the foreground area is in a wide range while the background region usually concentrates in a small interval, which means the atmospheric light varies less in the background region. Based on this observation and analysis, we establish the mixed atmospheric light model by applying heterogeneous atmospheric light in the foreground area to balance brightness deviation and maintain the global atmospheric light in the background region. We follow the design of Shiau et al. [22] to distinguish the foreground and background regions. The segmentation threshold can be calculated as follows:

$$\text{threshold} = \frac{\underset{x \in I}{\text{sum}}\left(I_{\text{gray}}(x)\right)}{w \times h} \tag{3}$$

where w and h are the width and height of the input frame $I(x)$, respectively, and $I_{\text{gray}}(x)$ means the grayscale image of $I(x)$. Through this segmentation, we can divide the pixels into

Algorithm 1 Heterogeneous Atmospheric Light Algorithm

Input:
$I_n(x, y)$: Input hazy frame
r: Length of the heterogeneous atmospheric light chunk
Output:
$A_n(x, y)$: Heterogeneous atmospheric light map
1) /* Iterate the atmospheric light chunks */
2) *for* $(x, y) \leftarrow (1, 1)$ to $(h/r, w/r)$ **do**
3) /* Iterate the chunk and find the max value */
4) $I_{maxg} \leftarrow -1$
5) **for** $(x_c, y_c) \leftarrow (1, 1)$ to (r, r) **do**
6) **if** $(I_{\text{gray}}(x_c, y_c) < threshold)$
7) $I_{maxg} \leftarrow \text{MAX}(I_{\text{gray}}(x_c, y_c), I_{maxg})$
8) **end if**
9) **end for**
10) **if** $(I_{maxg} == -1)$
11) /* Belong to background regions, ignore this chunk */
12) $A_n(x, y) \leftarrow -1$
13) **else**
14) $A_n(x, y) \leftarrow \text{I}(\arg(I_{maxg}))$
15) **end if**
16) *end for*
17) *return* $A_n(x, y)$
18) *End*

two categories. Those whose grayscale value is less than the threshold belong to the foreground region. Otherwise, they are part of the background region. The visual result is illustrated in Fig. 9(c) of Section V. So, mixed atmospheric light map $M_n(x)$ is expressed in the following:

$$M_n(x) = \begin{cases} A_n(x), & x \in I_{\text{fore}} \\ A_n^{\text{back}}, & x \in I_{\text{back}} \end{cases} \tag{4}$$

where I_{fore} and I_{back} are the foreground region and background region, respectively, n is the current input frame number. A_n^{back} represents the global atmospheric light for the background region while $A_n(x)$ means the heterogeneous atmospheric light for the foreground region. In the background region, we follow the traditional global atmospheric light solutions [18], i.e., the pixel with the maximum grayscale value will be selected and assigned to A_n^{back}. The expression is shown in the following:

$$A_n^{\text{back}} = \left\{ I_n(y) \mid y = \text{Coordinate}\left(\max_{x \in I_{\text{back}}} I_{\text{gray}}(x) \right) \right\} \tag{5}$$

where Coordinate indicates the coordinate of the pixel.

Algorithm 1 shows the procedure of estimating $A_n(x)$, which is the heterogeneous atmospheric light map for foreground regions. First, we separate the region into chunks (steps 1–4). Second, the maximum grayscale value $I_{\max}$ in each chunk is calculated (steps 5–9). If the chunk contains a mix of foreground and background pixels, we only consider the foreground pixels (steps 6–8). Third, if $I_{\max}$ is equal to -1, it means that this chunk belongs to the background area and will be ignored, otherwise the related pixel is taken as the atmospheric light of the chunk (steps 10–16). Finally, the heterogeneous atmospheric light map $A_n(x)$ is generated after the iteration finishes (step 17).

After we finish the estimation for A_n^{back} and $A_n(x)$, a coarse-grained atmospheric light map $M_n(x)$ can be generated by mixing them. Because the chunk separation operations during the estimation result in the block effect which can be seen in the third column of Fig. 9(b), we follow up with a FGF operation [30] to smooth the output and obtain the final mixed atmosphere map $M_n^{\text{ref}}(x)$. So, the (1) can be updated as follows:

$$I_n(x) = J_n(x)t(x) + M_n^{\text{ref}}(x)[1 - t(x)]. \tag{6}$$

C. Transmission Estimation Module

For dehazing a fog-degraded image according to (1), there are two critical values (i.e., the atmospheric light and the transmission) needed to be estimated. In Section III-B, we have calculated the atmospheric light with the mixed atmosphere scattering model. In this section, we will combine the hardware characteristics of FPGA to accurately estimate the value of the transmission.

In the widely used dark channel model [20], the transmission $t(x)$ is estimated as follows:

$$t(x) = 1 - \omega \min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right) \tag{7}$$

where $I^c(y)$ means a color channel of $I(y)$, and $\Omega(x)$ is a local patch centered at where the transmission is presumed to be a constant. A^c represents the given global atmospheric light, and ω is a constant parameter range from zero (exclusive) to one (inclusive) to hold a little haze for distant objects. $\min_{y \in \Omega(x)}(\min_c(I^c(y)))$ is the dark channel of the image $(I^c(y))$. Based on the dark channel prior theory, we observe that the global atmospheric light values for each color channel are very close. Therefore, the transmission map can be approximated from the following:

$$t(x) = 1 - \omega \frac{\min_{y \in \Omega(x)} \left(\min_c I^c(y) \right)}{A^c} \tag{8}$$

but the transmission tends to be discontinuous in local areas if the value of image pixels changes suddenly, especially for the outlines of objects, which leads to phenomena like strong halos. Although a smaller patch can address this problem, it will decrease the accuracy of the transmission map. Guided filter [30], [35] is another solution that can alleviate the phenomena, but the irreversible loss of edge information reserve is still unsolved. To address this issue, we use a side window filter [31] instead of the minimal filtering dark channel to preserve the edge information which will spread along the normal direction in the dark channel prior dehazing algorithm. By moving the filter anchor from the center to the edge of the windows, we suppress the reverse diffusion of the edge during minimum filtering. Combined with the mixed atmospheric light, the transmission $t(x)$ can be approximated using

$$t(x) = 1 - \omega \times \frac{swf\left(\min_c \left(I^c(x) \right) \right)}{M(x)} \tag{9}$$

however, this approximation may result in some performance penalties because the observation may not be valid in the foreground area. Fortunately, the performance loss is within acceptable limits, especially when compared to the reduction in resource consumption. We evaluated the influence of this

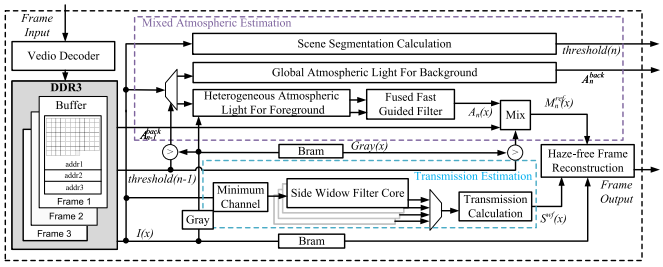

Fig. 4. High-level block diagram of the dehazing accelerator implementation, which comprises mixed atmospheric estimation, transmission calculation, and haze-free frame reconstruction modules and their interconnections.

approximation through an ablation experiment, and further details are presented in Section V.

After the mixed atmospheric light $M_n^{\text{ref}}(x)$ and the transmission $T_n(x)$ are estimated, the haze-free output $J_n(x)$ can be obtained by transforming (6) as follows:

$$J_n(x) = \frac{I_n(x) - M_n^{\text{ref}}(x)}{T_n(x)} + M_n^{\text{ref}}(x). \tag{10}$$

IV. Implementation

In this section, we present the implementation details of the MAPD architecture on the FPGA platform. We take full advantage of the parallel characteristics of FPGA to accelerate our proposed video dehazing algorithm, as shown in Fig. 4. Besides, for the experimental comparisons in Section V, DCP, CEP, RDCP, FFVD, MHVD, and REFD algorithms are also implemented on the same FPGA platform. More details are introduced in the following sections.

A. Implementation of Mixed Atmospheric Light Module

From Section III-B, we can see that multiple iterations for the whole frame are needed to generate the final mixed atmospheric light map (such as the threshold calculation and the atmospheric light estimation), which will lower the dehazing speed for the video. Considering the parallel characteristics of FPGA, we should parallelize them so that the estimation can finish in one iteration. However, the data dependency in the process is still an issue. For example, we have to calculate the segmentation threshold before the estimation for the atmospheric light. Fortunately, unlike the single image, there exists abundant correlated redundant information between video frames that we can leverage. Based on this observation, we adopt a parallel implementation flow by using the information in adjacent frames. As shown in Fig. 4, we reuse the values of threshold and global atmospheric light from the previous frame for the current frame, which can enable parallelization of these processes to reduce the dehazing time.

Fig. 5 shows the hardware architecture of the mixed atmospheric light module which is divided into five stages. First, in stage 1, we turn the input frame into a grayscale image by calculating the grayscale value of each pixel and then classify the pixels into background and foreground portions by the comparator, which will be handled by the global atmospheric light for background (GALB) and heterogeneous atmospheric light for foreground (HALF) circuit modules, respectively. Second, the GALB and HALF circuits execute in parallel to calculate the relevant atmospheric light values in stage 2 and stage 3. For the background regions, the GALB circuit will latch the pixel with the maximum grayscale value until the frame finish signal (EOF) comes and outputs the global atmospheric light. For the foreground portions, since a local patch of the pixel will be selected to calculate its heterogeneous atmospheric light, the pixels of all rows are needed to be cached resulting in the high overhead of BRAM resources. To further reduce the resource consumption of FPGA, we design a preprocessing unit called the row–column circuit in the HALF module, which contains a row subsampling subcircuit (Row_Sample) and a column buffer subcircuit (Col_Buffer) to reduce the consumption of BRAM. the Row_Sample consists of several triggers and muxes to select the maximum grayscale value from every r incoming consecutive data which will be sent to the Col_Buffer. Since the image data is loaded as a data stream, there is a necessary to design a column buffer (i.e., Col_Buffer) to cache the subsampled row pixels. The Col_Buffer is made up of r Linebuffers and every Linebuffer is composed of w/r trigger. With the data flowing in, the Linebuffer move their cached data to the next until none of them are empty. Then, the data from the same column will be compared by the comparator and output the pixel with maximum grayscale value as the heterogeneous atmospheric light value of the $r \times r$ chunk. Based on the ablation study result in Section V-H, the value of r is set to eight. After the heterogeneous atmospheric light map $A_n(x)$ is obtained, we need to perform a fast guided filter operation to eliminate the block effect caused by the local chunk in the heterogeneous atmosphere estimation. Since we have divided the grayscale image into chunks earlier, there is no need to subsample the $A_n(x)$ again, which will consume extra hardware resources. So, we combine these two steps to design the hardware architecture of our fused fast guide filter. As shown in stage 4, we first split the $A_n(x)$ into three color channels (red, blue, green) and compute the local linear coefficients. Second, the upsampling circuits are used to bilinearly upsample the two coefficient maps to the original size. Finally, The refined results are obtained by the multiplier and adder circuits. With our fused fast guide filter, the sampling operations and mean-filter operations are reduced by 50% and 22%, respectively. In stage 5, the final mixed atmospheric light map $M_n^{\text{ref}}(x)$ will be generated by merging the global atmospheric light and the heterogeneous atmospheric light map.

In most cases, our solution works well because there is a temporal relation between the adjacent frames of the video. However, when drastic changes occur in adjacent frames, the flicker problem is caused [36], which means that the global atmospheric light and threshold of the previous frame are no longer valid for the current frame. Reusing the threshold segmentation will result in bias in the global atmospheric light value of the background area and mislead the next frame. To address these issues, we propose a two-step prediction strategy, as shown in the left and top right of Fig. 5. First, we compute a global atmospheric light G_n by selecting a small area of the image data stream and then compare it with G_{n-1} to predict the flicker scene. If the difference is greater than a

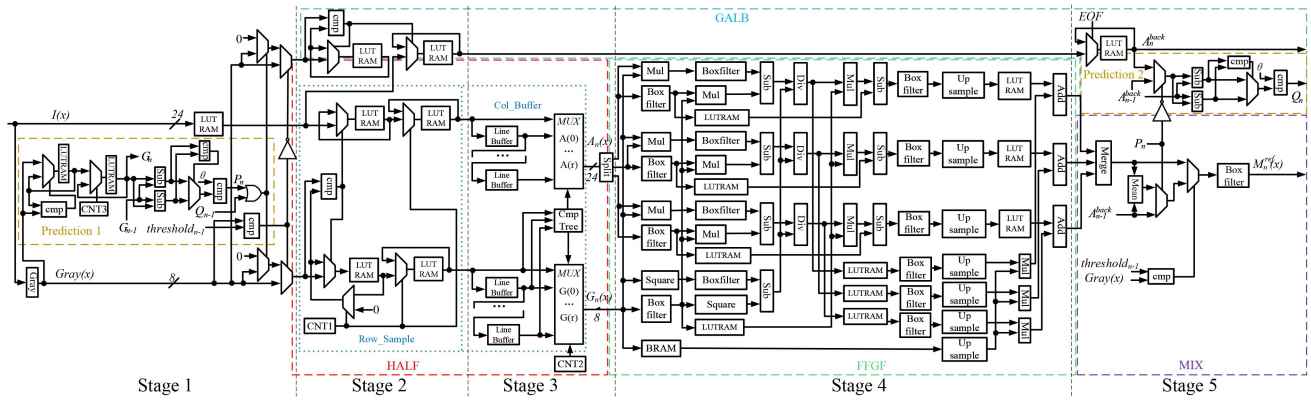

Fig. 5. Hardware architecture of mixed atmospheric light module, which consists of five stages. Stage 1: Grayscale conversion circuit module. Stage 2 and Stage 3: HALF and GALB circuit modules. Stage 4: FFGF circuit module. Stage 5: MIX circuit module.

specific threshold θ, we switch back to heterogeneous atmospheric light estimation for the current frame. Meanwhile, since the threshold segmentation is invalid, a global atmospheric light for the entire image is also computed for the next frame. To prevent the failure of the first step prediction, which would affect the next frame, we calculate the difference between the global atmospheric light of the current frame (i.e., A_n^{back}) and the previous frame (i.e., A_{n-1}^{back}), and compare it with θ to predict the next frame. Since the work [22] has observed hundreds of frames after video defogging and summarized the optimal value of θ, we also set it to three by following their setting. Additionally, when the flicker scene is predicted, we average the global atmospheric light of the previous frame and the atmospheric light value of the current frame to yield a smooth result. With the prediction strategy, MAPD can also achieve good effects in drastically changing scenes. More visual results are shown in Section V-D.

B. Indirect Transmission Estimation Module

In Section III-C, we get the transmission map $T_n(x)$ by (9). However, there exist several issues in the hardware implementation. First, in most traditional dehazing methods, the mixed atmospheric light map $M_n^{\text{ref}}(x)$ is assumed to be known, when the transmission is calculated. It means that we have to wait until the estimation of $M_n^{\text{ref}}(x)$ is finished, i.e., the calculations of $M_n^{\text{ref}}(x)$ and $T_n(x)$ cannot be processed in parallel, resulting in increased dehazing time. Shiau et al. [22] used the atmospheric light of the previous frame to calculate the transmission, while it is not suitable for scenes with drastic changes in foreground and background regions. Second, during the process of (9), there is a division operation on $M_n^{\text{ref}}(x)$. However, when we leverage (10) to recover the haze-free frame, another division operation is performed on $T_n(x)$. Since there will be a lot of hardware resource consumption for division operation in FPGA, it is necessary to simplify the calculation before designing the circuit. Combining (9) and (10), we can transform the expression of the haze-free output $J_n(x)$ as follows:

$$J_n(x) = \left(\frac{I_n(x) - S_n^{wf}(x) \times \omega}{M_n^{\text{ref}}(x) - S_n^{wf}(x) \times \omega} \right) \times M_n^{\text{ref}}(x) \qquad (11)$$

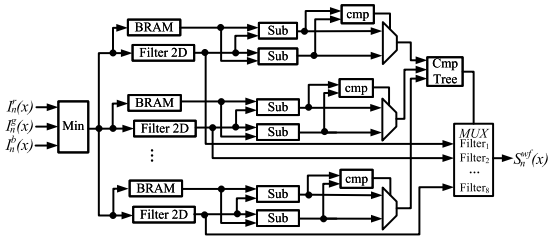

Fig. 6. Hardware architecture of indirect transmission estimation module.

where $S_n^{wf}(x)$ is the side window filter result for the input frame $I_n(x)$. From (11), it can be seen that the extra division operation is eliminated successfully. Moreover, the calculation of $S_n^{wf}(x)$ no longer depends on $M_n^{\text{ref}}(x)$, which means that the indirect transmission estimation module can be parallelized and has no negative effect on scenes with drastic changes. Based on this transformation, we design the indirect transmission estimation hardware module as shown in Fig. 6, which depicts a hardware implementation of side-window filtering for the minimal single-channel of the input frame $I_n(x)$.

C. Haze-Free Frame Reconstruction Module

After we obtain the value of $M_n^{\text{ref}}(x)$ and $S_n^{wf}(x)$, the haze-free frame $J_n(x)$ can be calculated by (11). Fig. 7 shows the hardware implementation of the haze-free frame reconstruction module. First, we divide the input frame $I_n(x)$ and the mixed atmospheric light map $M_n^{\text{ref}}(x)$ into three single-channel signals, and leverage a multiplier to get the value of $S_n^{wf}(x) \times \omega$. Second, for each channel, two subtractors are used to calculate the numerator and denominator of (11), respectively. Finally, the haze-free frame is reconstructed through the multiplier and divider.

V. Evaluation

In this section, we run a set of experiments to evaluate the dehazing performance of MAPD on an FPGA platform, namely, MZ7100FA, by using multiple types of video and image datasets, which demonstrates that MAPD has several advantages in haze removal performance and dehazing speed compared with other methods. We also implement DCP, CEP,

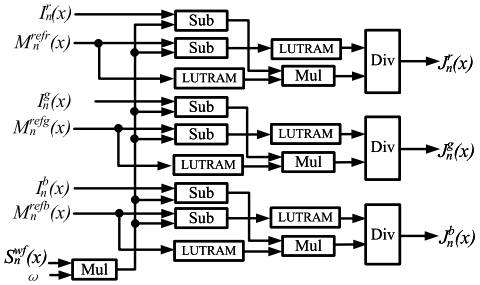

Fig. 7. Hardware implementation of haze-free frame reconstruction module.

RDCP, FFVD, MHVD, and REFD on the FPGA platform and deploy D^4 on the Nvidia GPU platform for comparison.

A. Experimental Setup

The experimental platform is a Xilinx XC7Z100-2FFG900I FPGA SoC (System on Chip) with an ARM Cortex-A9 dual-core processor. As we have depicted in Section III-A, the only MAPD components implemented on the PS side are the video frame buffer controller and dehaze parameter initialization. All other components are deployed on the PL side. This also applies to the implementation of the comparison algorithms which we will mention later. For comparison, we select two classical haze removal algorithms, i.e., DCP and CEP, and implement them on our FPGA platform. Since these two algorithms are both derived from the simplified atmospheric scattering model, we implement the hardware modules of their estimation for the global atmospheric light A and the transmission $t(x)$, and then restore the haze-free frame by a reconstruction hardware module, whose hardware implementation flow is similar to MAPD. In addition, we also compare our MAPD to other FPGA-based solutions, such as RDCP, FFVD, MHVD, and REFD, which are all ported to our FPGA hardware platform for a fair comparison. All experiments are performed on our FPGA hardware platform, where the clock frequency is set to 100 MHz and the bit-width is set to eight. To measure the runtime power consumption of the whole SoC, we plug a power meter in the FPGA platform. Moreover, since GPUs are commonly used in autonomous driving, we select the latest GPU-based dehazing solution, namely, D^4 [15], for comparison. Following the guidelines in this article [15], we deployed D4 on our platform equipped with the Nvidia 3080 GPU. Considering the differences in platforms, we only compare MAPD with D^4 in terms of dehazing performance and dehazing time, while also presenting the resource utilization and power consumption results of D^4 in Section V-F.

We use four datasets in this article—1) RESIDE [37]; 2) I-HAZE (35 indoor image pairs) [38]; 3) O-HAZE (45 outdoor image pairs) [39]; and 4) REVIDE (2031 frame pairs) [40]. RESIDE is a realistic single-image dehazing dataset, while I-HAZE and O-HAZE are real-world scene datasets. REVIDE is the real-world high-definition video haze removal dataset that contains four kinds of indoor fog-degraded videos (i.e., the Laboratory style, the Corridor style, the Western style, and the Eastern style). We run these datasets to evaluate our proposed method and the other seven dehazing algorithms. For REVIDE, the video stored in the SD card is first decoded into video frames on the PS side and then sent to the dehazing core on the PL side for haze removal processing. Finally, the haze-free frames are output to the screen or encoded and saved back to the SD card. For other image datasets, the process is similar to REVIDE except that there is no need to encode and decode images.

B. Qualitative Comparison on Real-World Images

To fully assess the proposed MAPD, we evaluate the dehazing performance in the case of different hazy images on a real-world dataset of over 40 images collected by ColorLines [41]. As shown in Fig. 8(b), DCP helps reveal some details masked by fog, but darkens some regions of the recovered image, resulting in color variations within dense fog (such as the city skyline in the third image). CEP in Fig. 8(c) enhances the scene to some extent but leaves a lot of white artifacts around object boundaries (like tree leaves). For most examples, RDCP in Fig. 8(d) and MHVD in Fig. 8(f) provide more satisfactory results. However, color distortion still occurs in some areas, especially in regions of the sky and darker backgrounds. As shown in Fig. 8(e), FFVD achieves a brighter visual experience in most scenes while exacerbating the color distortion phenomenon. Fig. 8(g) shows that REFD does not perform well on real-world images because the fog cannot completely be removed and even some noise is generated. The recovery of object details in D^4, as shown in Fig. 8(h), is impressive, but some areas, such as the top left region in the first and third images and the trees in the second picture, appear oversaturated. In comparison, MAPD preserves most of the details and compensates for the brightness of local areas by using the heterogeneous atmospheric light in the foreground area. Additionally, MAPD addresses the common phenomena of illumination deviation and contour fuzziness present in real outdoor dehazing images, resulting in a visually appealing output.

C. Qualitative Performance for Autonomous Vehicles

To better demonstrate the benefit of our proposed method in real-time systems, we discuss several use-cases of MAPD utility in this section. We select four typical types of scenarios in the autonomous vehicle for evaluation. Fig. 9 shows the six fog-degraded images, the haze-free results dehazed by MAPD, and some visualizations of intermediate results.

MAPD is an efficient pre-PS for a variety of autonomous vehicle tasks (e.g., object detection and target tracking). Fig. 9(a)–(c) demonstrates the effectiveness of defogging in lanes, which will be beneficial to enhance the safety of autonomous driving and advanced driver assistant systems. MAPD can help the object detection module distinguish the blurred pedestrians or objects in fog in advance, which can reserve more time for processing to avoid potential accidents in sudden situations. For example, in Fig. 9(a), the in-vehicle cameras show clearer and more natural images after dehazing with MAPD, which makes the cars in the fog more prominent and easy to recognize. Fig. 9(b) and (c) exhibits similar results. Fig. 9(d) extends MAPD to application scenarios beyond road

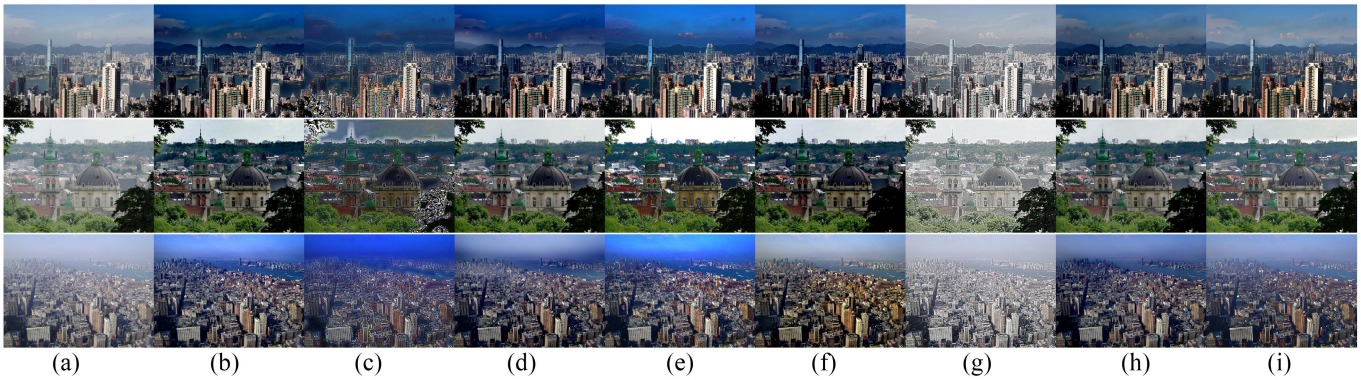

Fig. 8. Qualitative comparison of different haze removal methods on real-world images collected by Color-Lines [41]. (a) Hazy image. (b) DCP. (c) CEP. (d) RDCP. (e) FFVD. (f) MHVD. (g) REFD. (h) D4. (i) MAPD.

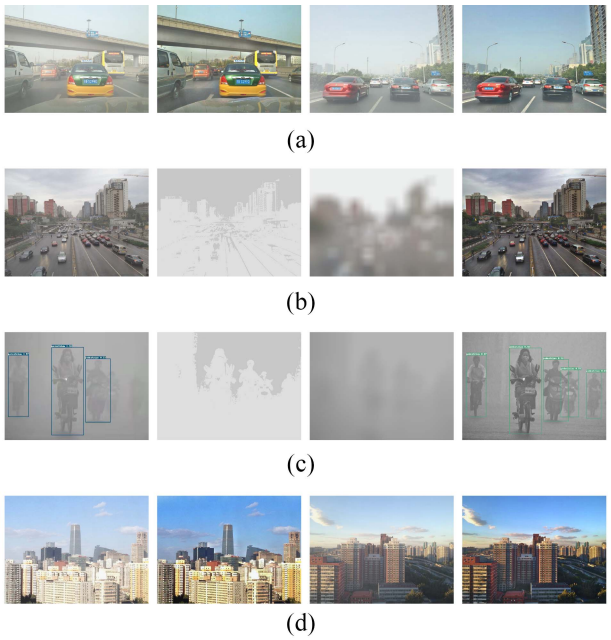

Fig. 9. Dehazing performance on various scenarios. The first and the last row consists of two image pairs, respectively, which include a fog-degraded image and the haze-free result dehazed by MAPD. The image pairs in the second and the third row are listed from left to right as the hazy image, the visual image for segmentation, the visual image for the mixed atmospheric light map, and the haze-free result dehazed by MAPD. (a) Automobile driver assistant systems. (b) Roadway traffic camera. (c) Pedestrain prone zebra crossing and intersection. (d) Aerial(drone-based) building.

traffic, such as aerial imagery from drones. In severe fog weather, drones run the risk of crashing into buildings, so the real-time defogging system has become a major advantage of air vehicle technology. In our experimental results, the building surfaces and roofs in the distance are restored to their original colors and distinguished from the sky, which is conducive for drones to plan a route to avoid a collision. In addition, we present the visualization images for segmentation and mixed atmospheric light map, as shown in the second and third columns of Fig. 9(b) and (c), respectively. From these intermediate figures, we demonstrate that the segmentation for the foreground area and background region achieves well effects, even in severe fog weather.

To further demonstrate the effectiveness of MAPD, we use the recently popular YOLOv4 object detection algorithm [42] to detect objects on haze-free images dehazed by DCP, CEP, RDCP, FFVD, MHVD, REFD, D^4, and MAPD, respectively. The experimental results are shown in Fig. 10(j). In the first row, only MHVD and MAPD recognize the distant car, while other methods, such as DCP, CEP, RDCP, FFVD, REFD, and D^4, omit that car. However, MHVD does not detect the foremost car in the second row. Moreover, in the last row, all methods except our proposed MAPD failed to detect the truck in the front right, which means that the stereo vision system modules could be misled to make incorrect judgments and lead to a traffic accident. The evaluations show that MAPD recovery better in detail and brightness in the foreground regions that can help autonomous driving cars detect more objects.

D. Qualitative Performance on Hazy Video

For evaluating real video applications, the case we consider is a high-definition (HD, the resolution is 1920×1080) video dehazing from the REVIDE dataset by splicing the video clips into a whole video. We compare our proposed method with MHVD and FFVD, which are also both the video defogging approaches. Fig. 11 shows the qualitative performance comparison for adjacent frames. It can be seen that MAPD has superior performance than MHVD and FFVD in color details and brightness recovery. Especially when drastic changes occur in adjacent frames, MAPD shows better flicker reduction performance than MHVD and FFVD.

Furthermore, to demonstrate the effectiveness of the prediction strategy, we conduct an ablation evaluation by removing the prediction hardware designs (denoted as MAPD_NP). The experimental results are presented in the fourth row of Fig. 11. Compared to MAPD_NP, MAPD achieves superior dehazing performance in drastically changing scenes (Frame 692). Since the prediction strategy successfully detects the drastic changes that occur in adjacent frames and switches back to the heterogeneous atmospheric light estimation for Frame 692, it avoids image color distortion caused by reusing the segmentation threshold and global atmospheric light of the previous frame (i.e., Frame 691).

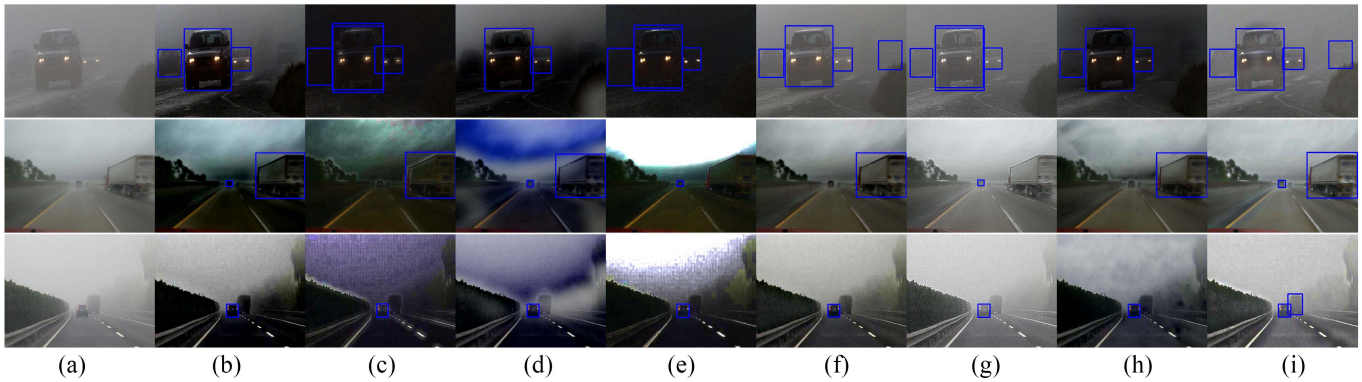

Fig. 10. Object detection comparison using YOLOv4 [42]. Only MHVD and MAPD recognize the distant car in the first case, and all results of other seven methods fail to detect the truck in the third case. Our method works well on all cases. (a) Hazy image. (b) DCP. (c) CEP. (d) RDCP. (e) FFVD. (f) MHVD. (g) REFD. (h) D4. (i) MAPD.

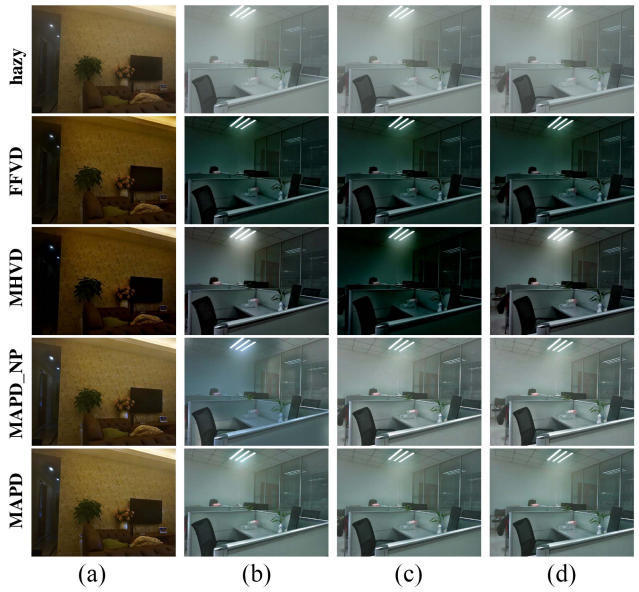

Fig. 11. Dehazing performance on adjacent frames. Frame 691 is the previous scene and Frame 692 to 694 are the current scenes. (a) Frame 691. (b) Frame 692. (c) Frame 693. (d) Frame 694.

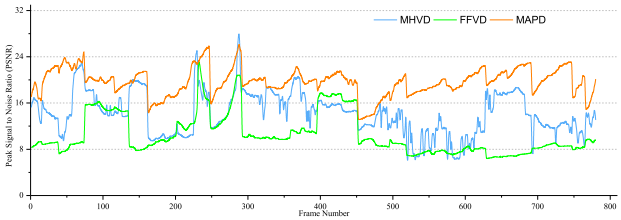

Fig. 12. Quantitative dehaze performance (PSNR) of MHVD, FFVD, and MAPD for video dehazing.

E. Quantitative Dehaze Performance Comparison

In order to further evaluate the effectiveness of MAPD for video defogging, we use the well-known PSNR (signal-to-noise ratio) indicator to perform a quantitative comparison. The quantitative comparison is performed on the computer with a python script by exporting the dehazing images from the FPGA platform. Fig. 12 shows the experimental results for MAPD and FFVD. In Fig. 12, the abscissa is the frame number and the ordinate means the PSNR which indicates the stochastic error between the haze-free image and the clear-day image. It demonstrates that MAPD outperforms MHVD and FFVD in PSNR and is more pronounced when the scene changes drastically. Especially from frames 480 to 650, MHVD and FFVD have severe video flicker phenomena, while MAPD almost eliminates these flicker issues.

Moreover, we also compare MAPD with DCP, CEP, RDCP, FFVD, MHVD, REFD, and D^4 for a quantitative evaluation by adding another two objective indicators, namely, structural similarity (SSIM) and color difference full reference model (CIEDE2000), in the case of two different workloads (I-HAZE and O-HAZE) collected from the real world to make the comparison more convincing.

Table II shows the quantitative performance results for seven haze removal methods with PSNR, SSIM, and CIEDE2000. For PSNR and SSIM, the high value indicates well dehazing performance, while it is the opposite for CIEDE2000. On the PSNR indicator, MAPD achieves the best dehazing performance for the I-HAZE workload while D^4 performs better in the O-HAZE workload. For the I-HAZE workload, MAPD promotes the PNSR by 33.3%, 53.5%, 22.0%, 29.1%, 5.3%, 8.2%, and 14.5%, compared to DCP, CEP, RDCP, FFVD, MHVD, REFD, and D^4. For the O-HAZE benchmark, the PNSR of MAPD also outperforms that of DCP, CEP, RDCP, FFVD, and REFD by 10.6%, 19.9%, 10.4%, 18.8%, and 23.3%, and is very close to MHVD. Although MAPD has a 6.3% lower PSNR than D^4, it exhibits a more stable performance because MAPD does not rely on the trained dataset. The quantitative results of CIEDE2000 show a similar trend.

In the middle row of Table II, we present the experimental results of SSIM, which is an image quality evaluation index to measure the structural information maintenance ability of an algorithm. It can be seen that MAPD performs the best SSIM on the I-HAZE dataset. Although the SSIM indicator for the O-HAZE workload in MAPD is not the best among all methods, it is very close to the highest value. In addition, MAPD promotes the SSIM by 23.0% and 7.3% compared with CEP, 9.4%, 5.4% compared with RDCP, 15.7%, 10.0% compared

TABLE II
QUANTITATIVE DEHAZING PERFORMANCE

	Benchmark	Method							
		DCP	CEP	RDCP	FFVD	MHVD	REFD	D^4	MAPD
PSNR	I-HAZE	12.40	10.77	13.55	12.81	15.70	15.28	14.44	**16.53**
	O-HAZE	14.74	13.59	14.76	13.72	16.29	13.22	**17.41**	16.30
SSIM	I-HAZE	0.69	0.66	0.74	0.70	0.81	0.77	0.77	**0.81**
	O-HAZE	**0.77**	0.70	0.72	0.69	0.75	0.69	0.76	0.76
CIEDE	I-HAZE	20.01	23.86	19.92	19.05	14.42	14.18	16.25	**12.70**
	O-HAZE	21.23	21.83	22.01	21.92	15.56	19.06	**15.10**	15.83
ACC	RESIDE	69.4%	57.3%	69.5%	55.3%	72.7%	58.1%	68.7%	**86.9%**

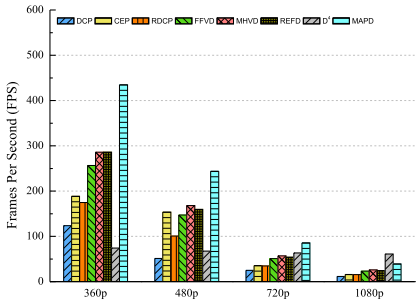

Fig. 13. Comparison of FPS for different resolution. The x-axis represents the resolutions, and the y-axis shows the value of FPS.

with FFVD and also exceeds REFD by 5.6%, and 9.2% in both scenes, respectively. The evaluation results demonstrate that MAPD is superior in restoring image structural information to other dehazing methods because of the mixed atmospheric light module.

The last row of Table II displays the quantitative results for object detection. We select hundreds of images related to autonomous driving scenarios from the RESIDE dataset to compute the object detection accuracy indicator (denoted as ACC). Compared with the other seven methods, MAPD promotes the ACC by 25.2%, 51.6%, 25.0%, 57.1%, 19.5%, 49.6%, and 26.5%, respectively. The more accurate map of atmospheric light and transmission in MAPD enables recovered images to preserve more details, which can help autonomous driving cars detect more objects.

F. Video Dehazing Time

We also measure the video dehazing time to evaluate seven haze removal methods under different video resolutions. For comparison, we transcode the video generated in Section V-D to different resolutions (360p, 480p, 720p, and 1080p). Since DCP, CEP, and RDCP need to cache the data of the current frame to wait for the result of the global atmospheric light in transmission calculation, their BRAM resources exceed the limit of our platform in the 1080p experiment. REFD also has to traverse the whole image to obtain the maximum and minimum values in their gray-scale comparison module, it has the same issue in the 1080p experiment. In addition, we observe that the synthesis results are consistent with the actual results in the other three resolutions. Based on these reasons, we use the synthesis results for DCP, CEP, RDCP, and REFD in the 1080p experiment, while FFVD, MHVD, and our MAPD still employ the actual results because of the application of adjacent frame information. Besides, for D^4, we measure the dehazing time by recording the time of its dehazing core algorithm.

The evaluation results are shown in Fig. 13. It can be seen that MAPD obtains the highest FPS (i.e., the minimum dehazing time) in almost all video resolutions except 1080p, compared to the other seven methods. Compared with DCP, CEP, RDCP, FFVD, MHVD, REFD, and D^4, MAPD promotes the FPS by 252.2%, 130.4%, 148.7%, 69.6%, 47.8%, 52.2%, and 487.0% for the 360p video, and also achieves 377.1%, 59.0%, 142.2%, 65.6%, 45.4%, 52.9%, and 262.5% improvement for the 480p video. When the video resolution is increased to 720p, MAPD outperforms by 242.7%, 141.9%, 147.6%, 68.4%, 48.7%, 59%, and 34.8%, respectively. Even for a high-resolution video dehazing (1080p), MAPD improves the FPS by 241.6%, 145.9%, 151.7%, 68.6%, 50.6%, and 62.4%, respectively, compared with the other six FPGA-based methods, and is still up to 39 FPS. D^4 achieves a relatively stable dehazing time at all resolutions and is even ahead in 1080p resolutions. However, the consumption of GPU memory and runtime power is up to 5728 MB and 135 W, respectively, which makes it a challenge to achieve real-time on autonomous vehicles [4]. The evaluation results demonstrate that our proposed methods can satisfy the demand for real-time systems such as autonomous driving.

G. Resource Utilization and Power Consumption

In this section, we compare the overhead of the REVIDE workload in DCP, CEP, RDCP, FFVD, MHVD, REFD, and MAPD. Fig. 14 shows the utilization of hardware resources at 720p resolution. As shown in Fig. 14, MAPD achieves significant gains in BRAM utilization. Compared with DCP, CEP, RDCP, FFVD, MHVD, and REFD, MAPD reduces BRAM utilization by 74.9%, 75.0%, 74.8%, 25.6%, 22.6%, and 73.9%. The reason is that we entirely decouple the atmospheric light calculations in our indirect transmission estimation module, which eliminates the requirement for additional memory to cache the current frame while waiting for the atmospheric light results. To address this issue, MHVD and FFVD also use the information from the previous frame, but they do not perform well in the scenes with drastic changes in foreground and background regions, as shown in Fig. 11. The experimental results also demonstrate that the other four methods (i.e., DCP, CEP, RDCP, and REFD) are unable to meet the requirement of high-definition video (1080p or even 2K resolution) haze removal because their BRAM utilization is nearly 100% at 720p resolution. DCP even has to convert some BRAM resources into LUT resources to run the dehazing program on board.

Table III shows the runtime power consumption in 720p and 1080p. The first row depicts that at 720p resolution MAPD lowers the power consumption by 12.5%, 11.1%, 6.7%, 5.1%, and 9.7%, respectively, compared to DCP, CEP, FFVD, and REFD, and is similar to MHVD. For the 1080p resolution evaluation, as we have discussed in Section V-F, DCP, CEP,

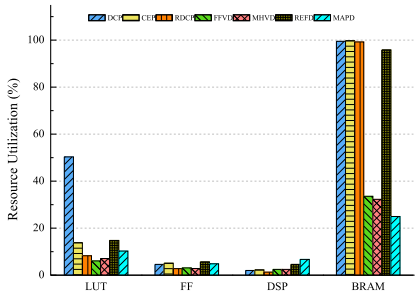

Fig. 14. Resource utilization for running video dehazing application. The *x*-axis expresses the type of hardware resources (e.g., LUT, FF, DSP, an BRAM), and the *y*-axis represents the resource utilization rate.

TABLE III
POWER CONSUMPTION (W) AT DIFFERENT RESOLUTION

Resolution \ Method	DCP	CEP	RDCP	FFVD	MHVD	REFD	MAPD
720p	6.4	6.3	6.0	5.9	5.5	6.2	5.6
1080p	\	\	\	6.6	6.0	\	5.9

TABLE IV
ABLATION STUDY ON O-HAZE

Metrics	Performance			Overhead			
	PSNR	SSIM	CIEDE	LUT	FF	DSP	BRAM
MAPD_NT	16.79	0.78	15.59	32517	30480	136	332
MAPD	16.30	0.76	15.83	28378	26760	136	188

RDCP, and REFD are unable to measure the runtime power consumption because of the exceeded resources. Therefore, we use backslashes instead, as shown in the second row. Compared to the other two FPGA-based solutions (i.e., FFVD and MHVD) which can be performed on our FPGA platform, MAPD reduces the power consumption by 10.6% and 1.7%, respectively. The experimental results demonstrate that MAPD has a smaller proportion of whole resources and power consumption and reserves excellent defogging performance.

H. Ablation Study

In this section, we first aim to validate the effectiveness of our proposed transmission approximation. We replace (9) with (7), where A^c is also changed to $M(x)$, and use (10) to recover the haze-free image while keeping all other settings unchanged (denoted as MAPD_NT). The ablation results are presented in Table IV. Compared to MAPD_NT, MAPD incurs only a slight reduction in dehazing performance of up to 2.92% (PSNR), while simultaneously reducing resource consumption by up to 43.37% (BRAM). This indicates that decoupling atmospheric light from transmission significantly reduces overhead with a relatively small performance penalty.

Second, for illustrating the advantage of our proposed FPGA accelerator, we extend our algorithm to the GPU platform, denoted as MAPD_G, and compare it with our FPGA accelerator, as shown in Table V. Observing Table V, we can see that the dehazing performance between MAPD and MADP_G is very close because they follow the same haze removal procedure. Although the dehazing time of MAPD_G is a little superior to MAPD, its power consumption is as high as 89W, which is hard to be applied for autonomous vehicles [4]. The evaluation results demonstrate that our FPGA accelerator is more suitable to meet the limited resource requirement of autonomous vehicles.

TABLE V
EVALUATIONS FOR MAPD AND MAPD_G ON O-HAZE AT 720P RESOLUTION

Metrics	PSNR	SSIM	CIEDE	Time (ms)	Power (W)
MAPD	16.30	0.76	15.83	11.7	5.9
MAPD_G	16.36	0.75	15.81	9.2	89

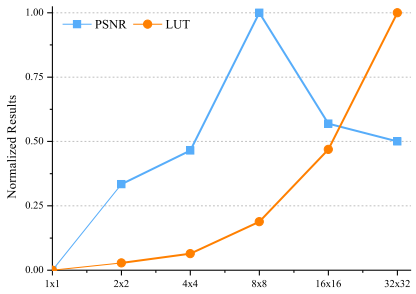

Fig. 15. Normalized PSNR and LUT for MAPD. The *x*-axis expresses the different size of the local patch and the *y*-axis represents the normalized result of dehazing performance (PSNR) and resource consumption (LUT).

Finally, to evaluate the impact of chunk size r on performance and resources, we increase r from 1 to 32 and measure the PSNR and LUT, respectively. Fig. 15 presents the normalized experimental results. It can be seen that the dehazing performance shows a trend of first increasing and then decreasing. This is because the confidence level of locating the most haze-opaque pixel for the local patch increases initially with the chunk size. However, for images with heterogeneous haze, the haze level may vary sharply from one region to another [43]. Thus, excessive chunk size may cause the estimated value of atmospheric light in one area to be taken in another area, which leads to a decrease in performance. Besides, a larger chunk size will also bring more resource consumption. Based on the ablation experiment and analysis, we set the value of r to eight.

VI. RELATED WORK

Video Haze Removal Methods: Most video dehazing methods try to recover the haze-free frames of a hazy video by leveraging the inverse function of (1) with estimated atmospheric particles and transmission through the fog-degraded frames, which is similar to the single image haze removal algorithms. Guo et al. [44] proposed a Gaussian-based dark channel algorithm for the atmospheric light calculation and a fusion-based transmission method with two different transmission models combined to establish an optical scattering model. They also combined an improved CLAHE to improve the fine-detail performance and maintain the color fidelity. RDCP [33] simplified the atmospheric light calculation of DCP to speed up the dehazing process and estimated an optimized medium transmission to handle the sky regions that are flawed in

DCP. Kumar et al. [24] employed a smaller tile-based method for the atmospheric light estimation to reduce the memory overhead. For the transmission calculation, an improved dark channel prior approach with Gaussian filtering was proposed to overcome halo artifacts in DCP, which are caused by abrupt changes in the frame pixel values. However, these video dehazing algorithms are all improvements to the classic DCP algorithm, which assumes that the global atmospheric light is homogeneous. In contrast, the atmospheric particle in reality is general an uneven distribution. Shiau et al. [22] realized the problem and extended the dark channel prior model to generate brighter and more colorful frames by calculating an appropriate atmospheric light according to the threshold value with a weighting technique. But the atmospheric light values of the bright portion and the dark portion are still global constants for their respective regions. It results in that the true atmospheric light of the entire image, which is the weighted sum of atmospheric light values in bright and dark regions, is still homogeneous. AtJwD [43] aimed to handle nonhomogeneous haze images where the haze distribution is uneven, i.e., some regions in an image are covered with denser haze while other areas are covered with shallower haze. It constructed a neural network architecture to combine the direct estimation approach for the haze-free image and the indirect physical parameters calculation method by estimating a spatially varying weight map. Although AtJwD obtained improved image quality, the computational resources and power overhead cannot meet the requirement of real-time processing. Besides, it does not consider the video dehazing which is much more complicated than the single image haze removal. Our MAPD designs a mixed atmospheric light model with an adaptive estimating scheme for the foreground region and the background area and makes full use of adjacent frames to support nonhomogeneous video dehazing.

Hardware-Accelerated Defogging Methods: In order to apply the dehazing methods to realistic applications like automatic driving, the real-time requirement is another key factor that should be taken into consideration. Researchers have implemented their haze removal methods on various hardware platforms, such as application specific integrated circuit (ASIC), GPU, and FPGA, to achieve the demand of real-time applications. Kumar et al. [24] designed an ASIC implementation for their video defogging method and achieved a high throughput. But the ASIC implementation lacks in terms of flexibility. Most neural network-based methods [12], [15], [16], [17] deploy their models on a single GPU to achieve real-time requirements, while the computational resources and power consumption of these methods are high, as well as the memory usage. Compared with GPU, FPGA can achieve the goal of low-cost and real-time processing, which makes it suitable for hardware implementation of video dehazing methods [22], [24], [33]. However, existing FPGA-based video dehazing methods neglect to deal with nonhomogeneous haze videos which are more general in real-world scenarios. Our MAPD proposes an FPGA-based video dehazing accelerator by implementing the mixed atmospheric light model on FPGA to adapt to the heterogeneous situation and meet the real-time requirement.

VII. Conclusion

In this article, we have proposed a new FPGA-based accelerator, namely, MAPD, for real-time video dehazing in autonomous vehicles. By taking full advantage of the hardware parallelization characteristics, MAPD decomposes the entire video dehazing process into different separate modules to reduce the data independence and achieve real-time video dehazing. For calculating the atmospheric light, MAPD establishes a mixed atmospheric light model, which uses heterogeneous atmospheric light in the foreground area to balance brightness deviation and keeps the global atmospheric light in the background region. For the transmission estimation, MAPD designs an indirect transmission estimation transformation to lower the logic resource consumption in FPGA. Evaluations using I-HAZE, O-HAZE, and REVIDE workloads show that MAPD promotes the dehazing performance and can be extended to more IoT applications including traffic surveillance systems and road navigation markings.

Acknowledgment

The authors are grateful to the anonymous reviewers for their helpful feedback.

References

[1] I. Yaqoob, L. U. Khan, S. A. Kazmi, M. Imran, N. Guizani, and C. S. Hong, "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges," *IEEE Netw.*, vol. 34, no. 1, pp. 174–181, Jan./Feb. 2020.

[2] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4316–4336, Jul. 2021.

[3] P. Kohli and A. Chadha, "Enabling pedestrian safety using computer vision techniques: A case study of the 2018 Uber Inc. Self-driving car crash," in *Proc. Future Inf. Commun. Conf.*, 2019, pp. 261–279.

[4] P. Zhao et al., "Neural pruning search for real-time object detection of autonomous vehicles," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, 2021, pp. 835–840.

[5] H.-K. Chiu, J. Li, R. Ambruş, and J. Bohg, "Probabilistic 3D multi-modal, multi-object tracking for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* 2021, pp. 14227–14233.

[6] I. Ogunrinde and S. Bernadin, "A review of the impacts of defogging on deep learning-based object detectors in self-driving cars," in *Proc. SoutheastCon*, 2021, pp. 1–8.

[7] D. Ngo, S. Lee, Q.-H. Nguyen, T. M. Ngo, G.-D. Lee, and B. Kang, "Single image haze removal from image enhancement perspective for real-time vision-based systems," *Sensors*, vol. 20, no. 18, p. 5170, 2020.

[8] G. Satyanath, J. K. Sahoo, and R. K. Roul, "Smart parking space detection under hazy conditions using convolutional neural networks: A novel approach," 2022, *arXiv:2201.05858*.

[9] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, p. 1188, 2020.

[10] M. Wang and S.-D. Zhou, "The study of color image defogging based on wavelet transform and single scale retinex," in *Proc. Int. Symp. Photoelectron. Detection Imag. Adv. Imag. Detectors Appl.*, 2011, Art. no. 81940F.

[11] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE Trans. Image Process.*, vol. 6, pp. 451–462, 1997.

[12] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: An end-to-end system for single image haze removal," *IEEE Trans. Image Process.*, vol. 25, pp. 5187–5198, 2016.

[13] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 154–169.

[14] W.-T. Chen, J.-J. Ding, and S.-Y. Kuo, "PMS-Net: Robust haze removal based on patch map for single images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11681–11689.
[15] Y. Yang, C. Wang, R. Liu, L. Zhang, X. Guo, and D. Tao, "Self-augmented unpaired image dehazing via density and depth decomposition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2037–2046.
[16] A. Mehra, M. Mandal, P. Narang, and V. Chamola, "ReviewNet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4256–4266, Jul. 2021.
[17] Z. Zheng et al., "Ultra-high-definition image dehazing via multi-guided bilateral learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 16180–16189.
[18] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[19] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, 2008.
[20] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.
[21] T. M. Bui and W. Kim, "Single image dehazing using color ellipsoid prior," *IEEE Trans. Image Process.*, vol. 27, pp. 999–1009, 2017.
[22] Y.-H. Shiau, Y.-T. Kuo, P.-Y. Chen, and F.-Y. Hsu, "VLSI design of an efficient flicker-free video defogging method for real-time applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 238–251, Jan. 2019.
[23] Z. Xu, X. Liu, and X. Chen, "Fog removal from video sequences using contrast limited adaptive histogram equalization," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, 2009, pp. 1–4.
[24] R. Kumar, R. Balasubramanian, and B. K. Kaushik, "Efficient method and architecture for real-time video defogging," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6536–6546, Oct. 2021.
[25] M. Guzek, "Driver's reaction time in the context of an accident in road traffic," in *Proc. Int. Sci. Conf. Transp. 21st Century*, 2019, pp. 184–193.
[26] Y. Park and T.-H. Kim, "A video dehazing system based on fast airlight estimation," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2017, pp. 779–783.
[27] M. N. Majoka and G. Raja, "FPGA based image de-hazing architecture for real time applications," in *Proc. Intell. Syst. Conf. (IntelliSys)*, 2017, pp. 609–613.
[28] H. Koschmieder, "Theorie der horizontalen sichtweite," *Beitrage Physik Freien Atmosphare*, vol. 33, no. 12, pp. 171–181, 1924.
[29] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *Int. J. Comput. Vis.*, vol. 48, no. 3, pp. 233–254, 2002.
[30] K. He and J. Sun, "Fast guided filter," 2015, *arXiv:1505.00996*.
[31] H. Yin, Y. Gong, and G. Qiu, "Side window filtering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8758–8766.
[32] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE Trans. Image Process.*, vol. 24, pp. 3522–3533, 2015.
[33] B. Zhang and J. Zhao, "Hardware implementation for real-time haze removal," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1188–1192, Mar. 2017.
[34] G. Du et al., "A real-time effective fusion-based image defogging architecture on FPGA," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 3, pp. 1–21, 2021.
[35] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[36] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Flicker effects in adaptive video streaming to handheld devices," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 463–472.
[37] B. Li et al., "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, pp. 492–505, 2018.
[38] C. Ancuti, C. O. Ancuti, R. Timofte, and C. De Vleeschouwer, "I-HAZE: A dehazing benchmark with real hazy and haze-free indoor images," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2018, pp. 620–631.
[39] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer, "O-Haze: A dehazing benchmark with real hazy and haze-free outdoor images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 754–762.
[40] X. Zhang et al., "Learning to restore hazy video: A new real-world dataset and a new method," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9239–9248.
[41] R. Fattal, "Dehazing using color-lines," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 1–14, 2014.
[42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YoloV4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[43] K. Metwaly, X. Li, T. Guo, and V. Monga, "Nonlocal channel attention for nonhomogeneous image dehazing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 452–453.
[44] J.-M. Guo, J.-Y. Syue, V. R. Radzicki, and H. Lee, "An efficient fusion-based defogging," *IEEE Trans. Image Process.*, vol. 26, pp. 4217–4228, 2017.

Yanjie Tan received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2011 and 2015, respectively, and the Ph.D. degree from Hunan University, Changsha, China, in 2021.

He is a Postdoctoral Fellow with the College of Computer Science and Electronic Engineering, Hunan University. His current research interests include real-time system and image and video processing.

Yifu Zhu received the B.S. degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2019. He is currently pursuing the M.S. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

His current research interests include FPGA and real-time system.

Zhaoyang Huang received the B.S. degree in computer science and technology from the China West Normal University of Computer Science, Nanchong, China, in 2020. She is currently pursuing the M.S. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

Her current research interests include embedded systems and storage systems.

Huailiang Tan received the B.S. degree from Central South University, Changsha, China, in 1992, and the M.S. degree from Hunan University, Changsha, in 1995, and the Ph.D. degree from Central South University in 2001.

He has more than eight years of industrial research and development experience in the field of information technology. He was a Visiting Scholar with Virginia Commonwealth University, Richmond, VA, USA, from 2010 to 2011. He is currently a Full Professor of Computer Science and Technology with Hunan University. His research interests include high-performance I/O, image and video processing, and embedded systems.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a SUNY Distinguished Professor of Computer Science with the State University of New York, New Paltz, NY, USA. He is also a National Distinguished Professor with Hunan University, Changsha, China. He has authored or coauthored over 840 journal articles, book chapters, and refereed conference papers. He holds over 70 patents announced or authorized by the Chinese National Intellectual Property Administration.

Dr. Li has received several best paper awards. He is among the World'S Top Five Most Influential Scientists in parallel and distributed computing based on a composite indicator of Scopus citation database. He is currently an Associate Editor of the *ACM Computing Surveys* and *CCF Transactions on High Performance Computing*. He has served for the editorial boards of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, and IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.