

# A HW/SW Co-Design of Video Dehazing Accelerator Using Decoupled Local Atmospheric Light Prior

Yanjie Tan, Yifu Zhu, Zhaoyang Huang, Feiteng Nie, Huailiang Tan\*  
The College of Computer Science and Electronic Engineering, Hunan University  
Changsha, Hunan, China

## ABSTRACT

In this paper, we introduce DLAPID, a novel decoupled parallel hardware-software co-design architecture for real-time video dehazing. From a software point of view, DLAPID isolates the atmospheric light operation from the initial transmission estimation to take full advantage of the hardware accelerators' parallelization features. For the hardware implementation, we deploy DLAPID both on FPGA and GPU platforms and validate its effectiveness. Using both real-world driving scenario testing sets and ground-truth datasets, we quantitatively and qualitatively assess the proposed method against several SOTA (state-of-the-art) video dehazing models. The outcomes of our experiments demonstrate that our approach achieves better dehazing performance with lower power consumption and has real-time processing capabilities, thereby preventing potential accidents of autonomous vehicles.

## KEYWORDS

FPGA, decoupled atmospheric light, haze removal, real-time

## 1 INTRODUCTION

With the rapid development of automobile intelligence and advanced driver assistant systems (ADAS), autonomous driving, which is the advanced stage of assisted driving technology, has become a new hot spot. However, there still exist grand challenges in severe weather. For example, smog weather has a tremendous influence on the stereo vision system of autonomous vehicles [15]. It may mislead the stereo vision system to make incorrect judgments for potential dangers or sudden situations. Besides, the other visual perception modules, such as object detection [23], also depend on a clear video of the scene to avoid potential accidents. Therefore, in the case of severe smog weather, fast and effective video dehazing has significant implications for these scenarios that require real-time processing.

In recent years, researchers have proposed numerous video dehazing solutions, which can be divided into two categories, i.e., single-frame-based video dehazing (SFVD) methods [5, 8] and multi-frames-based video haze removal (MFVD) approaches [9, 14, 15]. The former follows the single image haze removal methods to dehaze

\*Corresponding author: Huailiang Tan.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0601-1/24/06

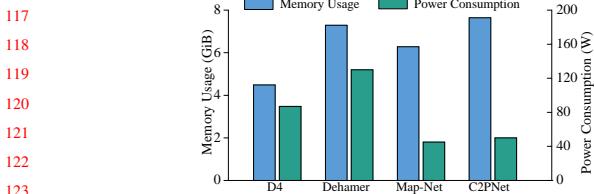
<https://doi.org/10.1145/3649329.3656512>

a foggy video frame-by-frame. However, current SFVD methods primarily focus on exploring the potential of deep learning in enhancing defogging performance [8], and they are hard to deploy in ADAS due to their massive computational resources and power consumption [23]. The latter makes use of the correlation information between video frames to lower computational complexity, but most of them still follow the same simplified atmospheric scattering model, which assumes that the atmospheric light is homogeneous and ignores the variability of the atmosphere in reality. To improve performance, MAPD [15] proposes a non-homogeneous atmosphere dehazing solution, while the coarse-grained foreground and background segmentation makes it difficult to select an accurate atmospheric light value for the background region. Furthermore, the atmospheric light separation in MAPD relies on observation without a derivative basis, weakening the dehazing effect.

Moreover, another critical factor for haze removal methods on autonomous vehicles is the dehazing speed (i.e., dehazing time). If the fog of the videos that are detected by perception sensors on autonomous vehicles cannot be eliminated in real-time, it will affect the subsequent decision-making of other visual perception systems, and even cause serious accidents [22]. To satisfy the real-time requirement, many works have attempted to minimize dehazing time with the help of hardware acceleration techniques, such as GPU or FPGA. However, for the GPU-accelerated methods, the latest neural network (NN) dehazing work, i.e., DW-GAN [4], has compared the processing time for one  $1600 \times 1200$  frame with other SOTA NN-based methods, and its evaluation results present that the running time of dehazing a frame is still up to hundreds of milliseconds or more, which still cannot meet the need for real-time processing. The newest video dehazing work MAP-Net (CVPR 2023 [18]) also admits that although it is able to be faster than many dehazing works, it cannot meet the real-time requirements. In order to accelerate the dehazing process, several FPGA-based solutions are proposed [9, 14, 15]. Although existing FPGA-based methods take full advantage of adjacent frame information to achieve real-time or near-real-time requirements, they still suffer from performance issues, as mentioned in the previous paragraph.

In this paper, we propose a new HW/SW co-design real-time dehazing accelerator, namely Decoupled Local Atmosphere Prior Information Dehazing Accelerator (DLAPID), which belongs to a frame-by-frame solution and takes full advantage of the parallelization characteristics by designing several separate modules to efficiently dehaze the fog videos and achieve real-time video haze removal. The main contributions of our work are as follows:

- In the software layer, we explore a novel approximation approach to estimate the airlight instead of the traditional transmission estimation for fully decoupling the dependency of the



**Figure 1: Memory and power consumption comparison on SOTA GPU-based methods for dehazing a 1080p frame.**

atmospheric light in the airlight module and avoiding redundant division operations, which will consume numerous logic resources in some hardware architecture, such as FPGA.

- We estimate the atmospheric light map by a multi-grained local atmospheric light approximation model to eliminate the flicker issues caused by drastic scene changes in the video.
- In the hardware layer, we implement DLAPID both on FPGA and GPU to validate its effectiveness. We conduct a series of experiments with real-world driving scenario datasets to compare our design with other SOTA works. The evaluation results show that our method has a better haze removal effect and achieves real-time processing. The experiments also demonstrate that our solution can avoid potential accidents for autonomous vehicles.

## 2 BACKGROUND AND MOTIVATION

**Atmospheric Scattering Model.** In the field of computer vision and image processing, the atmospheric scattering model (ASM) is widely applied to illustrate the information of a hazy image as follows [7]:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (1)$$

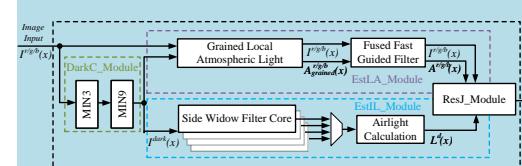
where  $x$  is the spatial coordinate of image pixels,  $I(x)$  and  $J(x)$  are the hazy image and haze-free image respectively.  $A$  is the global atmospheric light.  $t(x)$  represents the medium transmittance, which reflects the hazy concentration of fog-degraded images.  $I(x)$ ,  $J(x)$  and  $A$  are all three-dimensional vectors in RGB space.

In Eq. (1),  $J(x)t(x)$  (i.e., the first polynomial after the equation) demonstrates the direct effect of the scene radiance in the medium, which can be also named direct attenuation. The last polynomial  $A(1 - t(x))$ , namely airlight  $L(x)$ , represents an additive distortion for the scene radiance and results in the change of scene colors. So, if we substitute  $L(x)$  into Eq. (1), we can derive another expression for ASM as follows:

$$I(x) = J(x)\left(1 - \frac{L(x)}{A}\right) + L(x) \quad (2)$$

Obviously, Eq. (1) is an underdetermined equation where the number of known variables (i.e.,  $I(x)$ ) is far less than that of unknowns (i.e.,  $t(x)$  and  $A$ ). For solving this model, some extra constraints or priors have to be introduced, which help calculate the value of  $t(x)$  or  $A$ . With this prior information, the clear-day image/frame  $J(x)$  can be recovered by estimating the global atmospheric light  $A$  and the transmission map  $t(x)$ .

**Motivation.** Although the existing hardware-accelerated video dehazing solutions [5, 8, 9, 14, 15, 18] have achieved a good performance to recover the haze-free video, there still exist several unsolved issues, which will hinder the application of these methods in automated vehicles, as we have demonstrated in Section 1. The key factor is that they lack the ability of a software-hardware (SW/HW) co-design to achieve the perfect combination of defogging performance and real-time performance. For the GPU-based



**Figure 2: High level block diagram of DLAPID.**

approaches, the memory and power consumption pose a critical challenge to satisfy the real-time requirement on automated vehicles, just as shown in Fig. 1 with the latest SOTA models [6, 18, 20, 24].

Compared with the GPU-accelerated solutions, the current FPGA-based video dehazing methods [9, 14] can better achieve the real-time processing due to their full utilization of adjacent frames. However, when the brightness changes dramatically (such as scene change) between two adjacent frames in the video, the flicker problem occurs because the atmospheric light value of the previous frame no longer applies to the current frame, which will lower the quality of the restored video. Although MAPD [15] mitigates the effects of flicker, as shown in Fig. 11 and Fig. 12 of its work, the best way to prevent "flicker" is to avoid using the information from the previous frame.

Based on these observations and analyses, we propose a decoupled frame-by-frame video dehazing design to completely strip the calculations of the atmospheric light  $A$  from transmission map  $t(x)$  by re-deriving the ASM equation and transforming the approximation of  $t(x)$  to the airlight  $L(x)$ . Therefore, the parallel potential of hardware can be fully exploited to accelerate the dehazing processing. Furthermore, we introduce a local atmospheric light approximation model to address the resource problems caused by the calculations of global variables (such as  $A$ , the details are described in Section 3.2) in our frame-by-frame decoupled architecture and further enhance the dehazing performance.

## 3 DESIGN

**DLAPID Overview.** In this section, we'll introduce the complete dehazing algorithm process of DLAPID, which can help take full advantage of the parallelization capabilities of hardware. Fig. 2 shows a high-level block diagram of the accelerator that comprises the dark channel calculation module (DarkC\_Module), the local atmospheric light estimation module (EstLA\_Module), the indirect airlight calculation module (EstIL\_Module), and the haze-free frame restoration module (ResJ\_Module) and their interconnections. The hazy frame  $I(x)$  is the input and the haze-free frame  $J(x)$  is obtained as the output. In DarkC\_Module, the dark channel map of  $I(x)$  is computed and sent to the next two modules. In EstIL\_Module and EstLA\_Module, the decoupled airlight map  $L^d(x)$  and the local atmospheric light map  $A(x)$  are initially estimated in parallel respectively, and then the clear-day frame  $J(x)$  is finally calculated by combining  $A(x)$ ,  $L^d(x)$ , and the hazy frame  $I(x)$ . The details are introduced in the following subsections.

**Decoupled Airlight Module.** According to the famous dark channel prior dehazing algorithm [7], the transmission  $t(x)$  could be estimated simply by Eq. (3):

$$t(x) = 1 - \theta \min_{y \in \Omega(x)} \left( \min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c} \right) \quad (3)$$

where  $\theta$  is a constant parameter to maintain very little haze for the distant objects [7]. However, we can observe that the estimation

of  $t(x)$  relies on the result of  $A$ , which ought to be computed first. This coupling is not conducive to the parallelization of modules and requires extra resources to cache frame data [15]. The local atmospheric light module, which will be introduced later, can alleviate this situation, while decoupling the calculations may be the best choice. Therefore, we re-derive Eq. (1) based on the dark channel prior theory to obtain a decoupled transmission approximation that can separate the atmospheric light  $A$ .

Just as the derivation in [7], we also assume that the atmospheric light  $A$  is given (it will be calculated in Section 3.3) and the transmission in a local patch  $\Omega(x)$  is constant, which is denoted as  $\tilde{t}(x)$ . Then we put the minimum operation on both sides as follows:

$$\min_{y \in \Omega(x)} \left( \min_c I^c(y) \right) = \tilde{t}(x) \times \min_{y \in \Omega(x)} \left( \min_c J^c(y) \right) + (1 - \tilde{t}(x)) \times \min_c A^c \quad (4)$$

where  $I^c$  and  $J^c$  are a color channel of  $I$  and  $J$  respectively and the minimum operator  $\min_c$  is performed on each pixel to select the minimum value of three channels. According to the definition of dark channel [7], Eq. (4) can be expressed as follows:

$$I^{dark}(x) = \tilde{t}(x) \times J^{dark}(x) + (1 - \tilde{t}(x)) \times \min_c A^c \quad (5)$$

where  $I^{dark}(x)$  and  $J^{dark}(x)$  are the dark channel of  $I(x)$  and  $J(x)$  respectively. Since  $J(x)$  is the haze-free frame,  $J^{dark}(x)$  tends to be zero. Therefore, we can estimate the transmission  $t(x)$  as follows:

$$\tilde{t}(x) = 1 - I^{dark}(x) / \min_c A^c \quad (6)$$

From Eq. (6), it can be seen that the atmospheric light  $A$  has been stripped from the filtering operation. In order to further decouple the  $A$ , we estimate the airlight  $\tilde{L}(x)$ , which is described in Section 2, instead of the direct estimation for transmission  $t(x)$  as follows:

$$\tilde{L}(x) = A \times (1 - \tilde{t}(x)) = \frac{A}{\min_c A^c} \times I^{dark}(x) \quad (7)$$

Through this transformation, we decompose the estimation of airlight  $L(x)$  into the multiplication of two independent terms, where the left term is a polynomial of  $A$  and the right term represents the dark channel of  $I(x)$ . Therefore, in the hardware implementation, we can move the left term to the frame restoration module that completely decouple the atmospheric light module and the airlight module and eliminate additional cache consumption due to waiting for atmospheric light calculations. Since the assumption that  $\tilde{t}(x)$  in a local patch is a constant will lead to depth discontinuities sometimes [7], we employ a side window filter (SWF) [21] to refine the airlight map and preserve the edge information:

$$\tilde{L}(x) = \frac{A}{\min_c A^c} \times SWF(I^{dark}(x)) \quad (8)$$

where  $SWF(I^{dark}(x))$  is the result of side window filter for dark channel  $I^{dark}(x)$ . When we substitute Eq. (8) into Eq. (2), the clear-day frame can be recovered as follows:

$$J(x) = \frac{A[I(x) - L(x)]}{A - L(x)} = \frac{\min_c A^c \times I(x) - A \times \tilde{L}^d(x)}{\min_c A^c - \tilde{L}^d(x)} \quad (9)$$

where  $\tilde{L}^d(x) = \theta \times SWF(I^{dark}(x))$  is denoted as the decoupled airlight. Following the principle in work [7], we also keep the constant parameter  $\theta$  to make the recovered frame seem natural. Moreover, another benefit of utilizing Eq. (9) to recover the haze-free frame is the minimization of the division operation which will consume a lot of logic resources in FPGA. Obviously, there exists an extra division operation in Eq. (8), while it is eliminated by transforming  $t(x)$  to  $L(x)$ .

**Local Atmospheric Light Module.** As we illustrated in Section 2, atmospheric light plays a significant role in dehazing processing.

For some complex real-world non-homogeneous scenarios, where the haze density could vary from one area to another in frames with heterogeneous haze, the global atmospheric light is no longer applied [3]. Inspired by the derivation in the work [13] that the darkest pixels in a transmission map can represent the atmospheric light, we further propose a novel multi-grained local atmospheric light algorithm to obtain a more accurate map for the atmospheric light according to the heterogeneous atmospheric scattering model [3]:

$$I(x) = J(x) \left( 1 - \frac{L(x)}{A(x)} \right) + L(x) \quad (10)$$

where  $A(x)$  indicates the intensity of atmospheric light for each pixel. The procedure of estimating local atmosphere map  $A(x)$  are as follows. Firstly, based on the assumption that the haze level will keep unchanged in a local region [11], the input frame can be divided into chunks. Secondly, since the darkest pixel, which means the minimum value, of the transmission map in each chunk can represent the local atmospheric light of the chunk, we pick the pixel with the maximum dark channel value in each chunk as the coarse-grained atmospheric light according to the derivation of Eq. (6). Finally, a fast guided filter operation is performed to smooth the block effect and generate the final fine-grained atmospheric light map  $A(x)$ . Note that we also assume that the local atmospheric light in a local patch is constant, so the derivation for transmission or airlight in Seciton 3.2 can still hold.

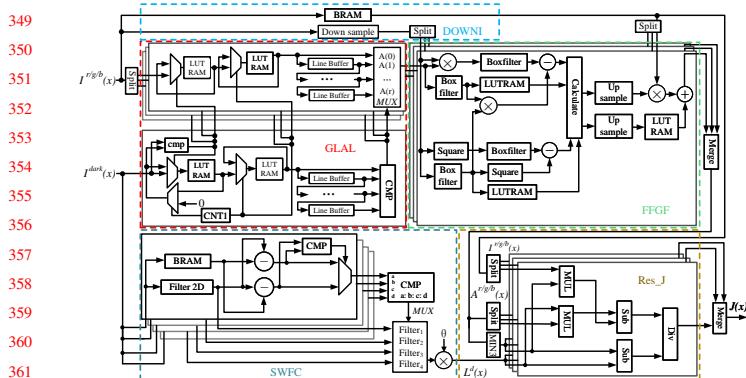
## 4 IMPLEMENTATION

Benefiting from the software design that the atmospheric light variable is stripped from the airlight formula successfully, the dehazing process for estimating the local atmospheric light and the decoupled airlight can be implemented in parallel. In this section, we implement our dehazing solution on the FPGA and GPU hardware platforms respectively according to the software design principle mentioned in Section 3. More hardware implementation details are introduced in the following subsections.

### 4.1 FPGA-based Implementation

We first present the hardware implementation details of our DLAPID architecture on FPGA using HLS (High-Level Synthesis). We take full advantage of the parallel characteristics of FPGA to accelerate our proposed dehazing algorithm and achieve real-time. Fig. 3 shows the entire hardware architecture diagram that can be divided into three main parts, i.e., the local atmospheric light module (EstLA\_Module, at the top half of Fig. 3), the decoupled airlight module (EstIL\_Module, in the bottom left half of Fig. 3), and the haze-free image restoration module (ResJ\_Module, in the bottom right half of Fig. 3). Note that both EstLA\_Module and EstIL\_Module need the calculation results of the dark channel map for  $I(x)$ , we abstract it as a pre-submodule (DarkC\_Module) in our hardware design. For the circuit of DarkC\_Module, we follow the design of research [9], which consists of a MIN3 filter (three input minimum filter) and a MIN9 filter (nine input minimum filter) to produce a dark channel value corresponding to a  $3 \times 3$  local patch of the pixel  $x$ . Therefore, one of the inputs in Fig. 3 is  $I^{dark}(x)$ .

**For EstLA\_Module,** it can be divided into two stages (i.e., GLAL and FFGF). Firstly, for estimating the coarse-grained atmosphere map, we divide the frame into chunks that are  $r \times r$  in size and pick the pixel with the maximum dark channel value as the local atmospheric light value of the chunk in stage GLAL. Based on the ablation study



**Figure 3: Hardware Implementation on FPGA.**

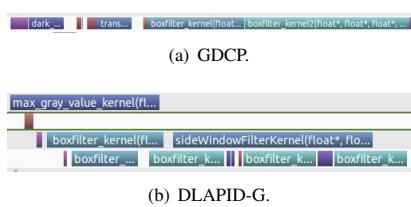
result (due to page limitation, we don't present it in the paper), we set the value of  $r$  to 16. Secondly, a fast guided filter operation is needed to eliminate the noises and remove the block effect, as shown in stage FFGF. From Section 3.3, we can see that multiple stages in EstLA\_Module need to use the input frame  $I(x)$ . Since the frame data on FPGA is loaded as a data stream, we parallelize the downsampling operation for  $I(x)$ , which should be performed in fast guided filtering, with the coarse-grained atmospheric light estimation to reduce cache resources consumption and speed up the calculation of fine-grained atmospheric light map, as shown at the top of Fig. 3 (i.e., the DOWNI circuit module). Furthermore, considering  $A_c(x, y)$  has been split into chunks earlier, the downsampling for the local atmospheric light is unnecessary and will consume additional hardware resources. So, we fuse these steps to design the circuit of our part-fused fast guide filter, as shown in stage FFGF. By removing the redundant subsample operation, the sampling operations are reduced by 25%. Finally, the final atmosphere map  $A(x)$  is output to ResJ Module.

**For EstIL\_Module**, as we have discussed in Section 3.2, the operation of the atmospheric light is separated during the calculation of the airlight. Therefore, we can move the hardware implementation for the minimum operator:  $\min_c A^c$  from EstIL\_Module to the final ResJ\_Module so that the data dependency between EstIL\_Module and EstLA\_Module can be eliminated to achieve parallelism. Firstly, we take  $I^{dark}(x)$  as the input and send it to the side window filter cores, which are the circuits of the side window filter [21]. Considering the balance between performance and overhead, we reduce the number of convolution kernels from eight to four (i.e., UP, DOWN, LEFT, and RIGHT). Secondly, the decoupled airlight map  $L^d(x)$  is calculated by a multiplier according to the definition in Section 3.2.

**For ResJ\_Module**, it has three signal inputs, i.e.,  $I(x)$ ,  $L^d(x)$ , and  $A(x)$ . Since  $L^d(x)$  is the single-channel map, we first expand it to a three-channel signal by a duplication function. Meanwhile, we use a MIN3 filter to calculate the minimum channel values of  $A(x)$ . Then, the subtractor, multiplier, and divider are called in turn to calculate the final result and generate the haze-free frame.

## 4.2 GPU-based Implementation

Our proposed DLAPID is also capable of implementing on the GPU-based platform to further accelerate the dehazing task. We use the NVIDIA Visual Profiler (NVP) tool to demonstrate the parallelism comparison between our GPU version with CUDA (DLAPID-G) and another non-NN based study [16], namely GDCP, which doesn't



**Figure 4: The visual profile result of DCP and DLAPID-G.** decouple the atmospheric light and the transmission. As shown in Fig. 4, it is clear that our decoupled architecture is more suitable for utilizing the parallel characteristics of GPU. The estimation for transmission in GDCP still relies on the value of atmospheric light so they are still series in the hardware implementation. Compared to GDCP, we further decouple the calculations of atmospheric light and transmittance to parallelize them, just as Fig. 4(b) presented. The evaluation results in Section 5 demonstrate that DLAPID-G can also achieve real-time processing.

## 5 EVALUATION

## 5.1 Experimental Setup

The FPGA platform is equipped with an XC7Z100-2FFG900I FPGA chip, which includes an ARM Cortex-A9 dual-core processor. All components mentioned in Section 4 are deployed on the PL (Programmable Logic) side, while only the dehazing parameter initialization is implemented on the PS (Processing System) side. The clock frequency of our FPGA platform is set to 100 MHZ, and the bit-width is set to eight. This also applies to the implementation of the comparison FPGA-based methods. For comparison, we select three FPGA-based video dehazing accelerators (FFVD [14], MHVD [9], and MAPD [15]) and all port to the same FPGA platform. Considering GPUs are widely used in autonomous driving, we further compare our method with the latest SOTA GPU-based deep learning (NN) works (MAP-Net [18] and C2PNET [24], which are both deployed on the platform with the Nividia 3070Ti GPU by using their released code. Moreover, to demonstrate our GPU-based version (DLAPID-G), we finally implement a traditional GPU-based algorithm (GDCP [16]) on the GPU platform. We use the real-world task-driven testing set (RTTS) [10], which contains over 4000 real-world hazy images covering mostly traffic and driving scenarios, to demonstrate the effectiveness of our haze removal method in autonomous driving.

## 5.2 Evaluation Results

**Visual Comparison on Real-world Hazy Driving Scenes.** We first compare DLAPID with the other six dehazing methods of GDCP, FFVD, MHVD, MAPD, MAP-Net, and C2PNET. Since DLAPID-G exhibits a very similar visual effect to DLAPID because of the same dehazing algorithm, we only present the dehazing effect of DLAPID in this section to save page space. Fig. 5 shows the visual results including one light-haze scene, one medium-haze scene, and one dense-haze scene. It can be seen that DLAPID clears the majority of the haze and recovers the most image details than the compared results. It also prevents the supersaturation problem and maintains the color details of images, even for dense-haze level traffic scenarios.

To further illustrate the effectiveness of DLAPID in autonomous driving, we use the popular object detection algorithms (YOLOv4

**Light-haze****Medium-haze****Dense-haze**

(a) The hazy images (b) GDCP (c) FFVD (d) MHVD (e) MAPD (f) MAP-Net (g) C2PNET (h) DLAPID

**Figure 5: Qualitative comparison of different haze removal methods.**

[1] and PP-YOLOE [19]) to detect objects on dehazing images of our proposed methods and the comparison solutions. Fig. 6 presents the object detection results of the haze images. It can be seen that only the haze-free images of DLAPID successfully help the object detection system recognize all vehicles or two-wheelers in severe fog weather, which means that the stereo vision system modules can make the correct judgments to avoid the potential accident with the help of DLAPID.

We further quantitatively calculate the object detection accuracy (abbreviated as ACC), which is defined as the quotient of the number of objects identified and the total count of objects in the frames, by selecting hundreds of hazy images of traffic scenes related to autonomous vehicles. The quantitative results in Table 1 also demonstrate that DLAPID is an efficient pre-processing system for a variety of autonomous vehicle tasks.

**Table 1: The Object Detection Accuracy Comparison**

Method	GDCP	FFVD	MHVD	MAPD	MAP-Net	C2PNET	DLAPID
Algorithm	62.2%	55.3%	68.7%	86.9%	64.9%	60.1%	<b>88.6%</b>
YOLOV4	66.1%	62.5%	69.8%	78.7%	68.3%	68.4%	<b>81.2%</b>

**Quantitative Dehaze Performance and Time.** To compare the haze removal effect more objectively, we select two widely-used no-reference IQA metrics (BRISQUE [12] and FADE [2]) to compare DLAPID with existing methods. As shown in Table 2, the BRISQUE of DLAPID outperforms by 15.6%, 13.1%, 23.3%, 26.3%, 27.4%, and 24.4% respectively. The quantitative evaluations for FADE also present similar results, which indicate that our DLAPID achieves the best performance for real-world scenes.

**Table 2: Quantitative Dehazing Performance (Small Value Means Better Performance)**

Metrics \ Method	GDCP	FFVD	MHVD	MAPD	MAP-Net	C2PNET	DLAPID
BRISQUE	32.23	31.29	35.46	36.87	37.48	35.97	<b>27.19</b>
FADE	1.1	1.7	1.78	1.33	1.89	2.89	<b>0.89</b>

We also measure the dehazing speed of eight methods under different video resolutions. As shown in Table 3, for the FPGA-based methods (FFVD, MHVD, MAPD, and DLAPID), the dehazing time of DLAPID is very close to MAPD and lower than FFVD and MHVD in all image resolutions which decreases the time by up to 40.5% and 55.8% respectively. This is because DLAPID and MAPD both decouple the atmospheric light from the transmittance

**Table 3: Dehazing Time (ms) at Different Resolution**

Resolution	FPGA-based methods				GPU-based methods			
	FFVD	MHVD	MAPD	<b>DLAPID</b>	NNs		non-NNs	
					MAP-Net	C2PNET	GDCP	<b>DLAPID-G</b>
360p	3.9	4.92	2.3	2.35	115	222	1.39	1.15
480p	6.79	8.67	4.11	4.12	176.4	389	1.53	1.19
720p	19.7	25.8	11.7	11.73	531.7	1160	2.35	1.33
1080p	42.99	57.9	25.49	25.58	1128.4	2600	3.66	2.3

equation so that the latency caused by the data dependency is reduced. However, the separation strategy in MAPD relies on observation without a derivative basis, its dehazing performance is lower than DLAPID, which has been demonstrated in Table 2. This also applies to the comparison of GDCP and DLAPID-G. For the NN-based work, MAP-Net and C2PNET show a relatively poor dehazing inference time on our 3070Ti GPU platform and as we have illustrated in Fig. 1, the tremendous memory and power-hungry requirements make these powerful GPUs usually too costly to be deployed on automated vehicles with limited memory and computation constraints. The evaluations of dehazing time demonstrate that DLAPID can achieve real-time (or near-real-time), which means that more time is left for other visual perception systems to deal with sudden situations.

Although both DLAPID and DLAPID-G can satisfy the real-time requirement in automated vehicles and the latter that belongs to a GPU dehazing design is even faster. However, the execution time GPU hardware platform may fluctuate due to the uncontrollable OS scheduling [17]. As shown in Fig. 7, which presents the distribution of the time cost to dehaze a video, the dehazing time of DLAPID-G fluctuates in the range from -3.9% to 23.9%. In contrast, the execution time on DLAPID is more stable because we can control the clock cycles and runtime pipeline of the FPGA accelerator programs, which demonstrates that DLAPID has critical-level timing determinism at nanosecond and makes it more suitable for hard real-time systems, such as automated vehicles.

**Table 4: Runtime power consumption**

Method	FPGA-based methods				GPU-based methods			
	FFVD	MHVD	MAPD	<b>DLAPID</b>	NNs		non-NNs	
					MAP-Net	C2PNET	GDCP	<b>DLAPID-G</b>
Power (W)	6.6	6.0	5.9	5.8	45	50	46	45

**Power Consumption.** We also measure the runtime power consumption of the whole SoC by plugging a power meter in the FPGA platform. Table 4 shows experimental results for the 1080p video. It can be seen that the power consumption of DLAPID is only 5.8W

## YOLOV4



## PP-YOLOE



Figure 6: Object detection comparison.

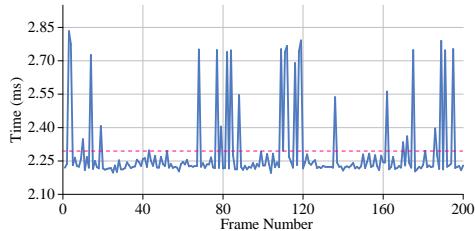


Figure 7: The time cost of DLAPIID-G to dehaze a video.

while the GPU-based methods consume much more power, which indicates that DLAPIID is suitable for being deployed on autonomous vehicles with limited resources and low power requirements.

## 6 CONCLUSION

In this paper, we have presented an SW/HW co-design dehazing architecture to enhance performance and satisfy real-time requirement. Our experimental results demonstrate that our method has a better dehazing effect and achieves real-time, 39 FPS for dehazing a  $1920 \times 1080$  video. The method can also promote the accuracy of object detection in autonomous vehicles to reduce potential dangers.

## ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under grants 62302158 and the Hunan Provincial Natural Science Foundation of China (2023JJ40175).

## REFERENCES

- [1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934* (2020).
- [2] Lark Kwon Choi, Jaehye You, and Alan Conrad Bovik. 2015. Referenceless prediction of perceptual fog density and perceptual image defogging. *IEEE Transactions on Image Processing* 24, 11 (2015), 3888–3901.
- [3] Yuxin Feng, Xiaozhe Meng, Fan Zhou, Weisi Lin, and Zhuo Su. 2023. Real-world Non-homogeneous Haze Removal by Sliding Self-attention Wavelet Network. *IEEE Transactions on Circuits and Systems for Video Technology* (2023).
- [4] Minghan Fu, Huan Liu, Yankun Yu, Jun Chen, and Keyan Wang. 2021. Dw-gan: A discrete wavelet transform gan for nonhomogeneous dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 203–212.
- [5] Vijay M Galshetwar, Prashant W Patil, and Sachin Chaudhary. 2021. Single frame-based video dehazing with adversarial learning. In *International Conference on Computer Vision and Image Processing*. Springer, 36–47.
- [6] Chun-Le Guo, Qixin Yan, Saeed Anwar, Runmin Cong, Wenqi Ren, and Chongyi Li. 2022. Image dehazing transformer with transmission-aware 3D position embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5812–5820.
- [7] Kaiming He, Jian Sun, and Xiaou Tang. 2010. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence* 33, 12 (2010), 2341–2353.
- [8] Akshay Juneja, Vijay Kumar, and Sunil Kumar Singla. 2023. Aethra-net: Single image and video dehazing using autoencoder. *Journal of Visual Communication and Image Representation* 94 (2023), 103855.
- [9] Rahul Kumar, Raman Balasubramanian, and Brajesh Kumar Kaushik. 2021. Efficient Method and Architecture for Real-Time Video Defogging. *IEEE Transactions on Intelligent Transportation Systems* 22, 10 (2021), 6536–6546. <https://doi.org/10.1109/TITS.2020.2993906>
- [10] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. 2019. Benchmarking Single-Image Dehazing and Beyond. *IEEE Transactions on Image Processing* 28, 1 (2019), 492–505. <https://doi.org/10.1109/TIP.2018.2867951>
- [11] Kareem Metwaly, Xueli Li, Tiantong Guo, and Vishal Monga. 2020. Nonlocal channel attention for nonhomogeneous image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 452–453.
- [12] Anish Mittal, Anush K Moorthy, and Alan C Bovik. 2011. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*. IEEE, 723–727.
- [13] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. 2016. Single image dehazing via multi-scale convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II* 14. Springer, 154–169.
- [14] Yeu-Horng Shiau, Yao-Tsung Kuo, Pei-Yin Chen, and Feng-Yuan Hsu. 2017. VLSI design of an efficient flicker-free video defogging method for real-time applications. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1 (2017), 238–251.
- [15] Yanjie Tan, Yifu Zhu, Zhaoyang Huang, Huailiang Tan, and Keqin Li. 2023. MAPD: An FPGA-Based Real-Time Video Haze Removal Accelerator Using Mixed Atmosphere Prior. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2023).
- [16] Chia-Chi Tsai, Cheng-Yen Lin, and Jiun-In Guo. 2019. Dark channel prior based video dehazing algorithm with sky preservation and its embedded system realization for ADAS applications. *Optics express* 27, 9 (2019), 11877–11901.
- [17] Chenren Xu, Shuang Jiang, Guojie Luo, Guangyu Sun, Ning An, Gang Huang, and Xuanze Liu. 2020. The case for fpga-based edge computing. *IEEE Transactions on Mobile Computing* 21, 7 (2020), 2610–2619.
- [18] Jiaqi Xu, Xiaowei Hu, Lei Zhu, Qi Dou, Jifeng Dai, Yu Qiao, and Pheng-Ann Heng. 2023. Video Dehazing via a Multi-Range Temporal Alignment Network with Physical Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18053–18062.
- [19] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yunling Du, et al. 2022. PP-YOLOE: An evolved version of YOLO. *arXiv preprint arXiv:2203.16250* (2022).
- [20] Yang Yang, Chaoyue Wang, Risheng Liu, Lin Zhang, Xiaojie Guo, and Dacheng Tao. 2022. Self-augmented unpaired image dehazing via density and depth decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2037–2046.
- [21] Hui Yin, Yuanhao Gong, and Guoping Qiu. 2019. Side window filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8758–8766.
- [22] Shu Dong Zhang and Zhi Bo Zhang. 2015. Contributing Factors and the Control of Haze. In *Advanced Materials Research*, Vol. 1073. Trans Tech Publ, 2751–2754.
- [23] Pu Zhao, Geng Yuan, Yuxuan Cai, Wei Niu, Qi Liu, Wujie Wen, Bin Ren, Yanzhi Wang, and Xue Lin. 2021. Neural Pruning Search for Real-Time Object Detection of Autonomous Vehicles. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 835–840. <https://doi.org/10.1109/DAC18074.2021.9586163>
- [24] Yu Zheng, Jiahui Zhan, Shengfeng He, Junyu Dong, and Yong Du. 2023. Curricular contrastive regularization for physics-aware single image dehazing. *arXiv preprint arXiv:2303.14218* (2023).