

Article

VBI-Accelerated FPGA Implementation of Autonomous Image Dehazing: Leveraging the Vertical Blanking Interval for Haze-Aware Local Image Blending

Dat Ngo ^{1,†} , Jeonghyeon Son ^{2,†} and Bongsoon Kang ^{2,*} 

¹ Department of Computer Engineering, Korea National University of Transportation, Chungbuk 27469, Republic of Korea; datngo@ut.ac.kr

² Department of Electronics Engineering, Dong-A University, Busan 49315, Republic of Korea; 1922913@donga.ac.kr

* Correspondence: bongsoon@dau.ac.kr; Tel.: +82-51-200-7703

† These authors contributed equally to this work.

Abstract: Real-time image dehazing is crucial for remote sensing systems, particularly in applications requiring immediate and reliable visual data. By restoring contrast and fidelity as images are captured, real-time dehazing enhances image quality on the fly. Existing dehazing algorithms often prioritize visual quality and color restoration but rely on computationally intensive methods, making them unsuitable for real-time processing. Moreover, these methods typically perform well under moderate to dense haze conditions but lack adaptability to varying haze levels, limiting their general applicability. To address these challenges, this paper presents an autonomous image dehazing method and its corresponding FPGA-based accelerator, which effectively balance image quality and computational efficiency for real-time processing. Autonomous dehazing is achieved by fusing the input image with its dehazed counterpart, where fusion weights are dynamically determined based on the local haziness degree. The FPGA accelerator performs computations with strict timing requirements during the vertical blanking interval, ensuring smooth and flicker-free processing of input data streams. Experimental results validate the effectiveness of the proposed method, and hardware implementation results demonstrate that the FPGA accelerator achieves a processing rate of 45.34 frames per second at DCI 4K resolution while maintaining efficient utilization of hardware resources.

Keywords: autonomous image dehazing; image fusion; real-time processing; vertical blanking interval



Academic Editors: Pierre Rochus, Xing Zhong, Fei Xing and Xiuqing Hu

Received: 30 December 2024

Revised: 23 February 2025

Accepted: 3 March 2025

Published: 5 March 2025

Citation: Ngo, D.; Son, J.; Kang, B. VBI-Accelerated FPGA

Implementation of Autonomous Image Dehazing: Leveraging the Vertical Blanking Interval for Haze-Aware Local Image Blending. *Remote Sens.* **2025**, *17*, 919. <https://doi.org/10.3390/rs17050919>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Remote sensing image systems capture and analyze Earth's surface data using platforms such as satellites, drones, and aircrafts. These systems support applications in climate monitoring, agriculture, urban planning, defense, and disaster management. However, atmospheric conditions such as haze, fog, dust, and pollution degrade image quality, obscuring surface details and complicating analysis. Image dehazing is crucial for mitigating these effects, enhancing visual clarity, contrast, and color fidelity. Over the past two decades, numerous dehazing methods have been developed, many based on the atmospheric scattering model [1]:

$$\mathbf{I}(x, y) = \mathbf{J}(x, y)t(x, y) + \mathbf{A}[1 - t(x, y)], \quad (1)$$

where I , J , t , and A represent the hazy image, the clean image, the transmission map, and the global atmospheric light, respectively. Bold variables denote three-channel values (RGB), while plain text denotes single-channel variables. The indices x and y correspond to pixel coordinates.

Dehazing methods are broadly categorized into non-learning-based and learning-based approaches. While a comprehensive review of these methods is beyond the scope of this work, interested readers are directed to [2–4] for further details. This section focuses on daytime dehazing, highlighting key methods summarized in Table 1.

Table 1. Summary of image dehazing algorithms.

Category	Sub-Category	Year	Paper	Core Technique(s)	Remarks
Non-learning	Prior	2011	[5]	Filtering	- Discovering dark channel prior (DCP) - Exhibiting poor performance in sky and bright regions
	Image enhancement	2012	[6]	Filtering	- White-balancing the input image to facilitate the use of $A = \{1, 1, 1\}$ - Adopting median of medians along lines to estimate the atmospheric veil - Suffering from halo artifacts
	Prior	2015	[7]	Maximum likelihood estimates (MLE)	- Discovering color attenuation prior (CAP) - Adopting linear regression to estimate scene depth from brightness and saturation - Prone to background noise and color distortion
	Image enhancement	2018	[8]	Image fusion	- Dehazing via the fusion of artificially underexposed images - Computationally efficient - Limited in handling diverse haze conditions
	Prior	2018	[9]	Hough transform and weighted least squares (WLS)	- Discovering haze-lines prior (HLP) - Demonstrating improved robustness compared to local priors - Prone to color distortion under non-homogeneous lighting conditions
	Prior	2019	[10]	Filtering and MLE	- Improving CAP to resolve background noise and color distortion - Computationally efficient - Poor performance in dense haze
	Prior	2021	[11]	Filtering and image fusion	- Nighttime dehazing method based on DCP and Retinex theory - Adopting color transfer to handle non-homogeneous lighting - Requiring careful tuning of performance-critical parameters
	Image enhancement	2022	[12]	Image fusion	- Incorporating DCP into the image-fusion-based dehazing scheme - Developing a self-calibrating factor for autonomous dehazing - Supporting real-time dehazing with an FPGA accelerator
	Prior	2022	[13]	Filtering and MLE	- Complete method for preprocessing-dehazing-postprocessing hazy images - Based on improved CAP - Poor performance in dense haze
	Prior	2023	[14]	Filtering	- DCP-based dehazing method - Computing the transmission map as the mean of three maps from each channel - Suffering from halo artifacts
	Prior	2023	[15]	Filtering	- Discovering saturation line prior - Requiring careful tuning of performance-critical parameters
	Prior	2024	[16]	Filtering	- Adopting DCP to estimate transmission map from superpixels - Adopting CAP and BCP (bright channel prior) to estimate atmospheric light - Limited by the adopted priors
	Prior	2024	[17]	Filtering and MLE	- Improving the self-calibrating factor for autonomous dehazing - Supporting real-time dehazing with an FPGA accelerator
Learning	Atmospheric scattering model (ASM)	2016	[18]	Convolutional neural network (CNN)	- Adopting a three-stage CNN for transmission map estimation - Computationally efficient compared to learning-based methods - Suffering from domain-shift issues
	ASM	2020	[19]	CNN	- Estimating the transmission map in a coarse-to-fine manner - Imposing holistic edges for smooth transmission map within the same object - Suffering from domain-shift issues
	End-to-end (E2E)	2020	[20]	Autoencoder (AE)	- Progressive refining dehazing results through boosting and error feedback - Suffering from domain-shift issues
	E2E	2021	[21]	AE	- Exploiting ASM and layer disentanglement to achieve unsupervised dehazing - Limited generalizability
	ASM	2022	[22]	AE	- Exploiting CycleGAN with ASM to achieve unpaired dehazing - Likely to overestimate the transmission map in bright regions
	E2E	2023	[23]	Vision transformer	- Enhancing Swin Transformer’s architecture - Suffering from domain-shift issues - Prone to color distortion in dense haze
	E2E	2023	[24]	CNN	- Adopting curriculum learning to improve dehazing performance - Classifying negative samples into easy, hard, and ultra-hard - Suffering from domain-shift issues

Table 1. Cont.

Category	Sub-Category	Year	Paper	Core Technique(s)	Remarks
Learning	E2E	2023	[25]	AE	- Considering various degradation types to synthesize hazy images - Suffering from domain-shift issues
	E2E	2024	[26]	Diffusion model	- Adopting a region-based diffusion module to handle high-resolution images - Adopting range-null-space decomposition to facilitate reverse diffusion process
	E2E	2024	[27]	Conditional variational AEs	- Adopting conditional variational AEs to generate multiple dehazing results - Fusing the dehazing results to generate a more accurate output

1.1. Non-Learning-Based Dehazing

Non-learning-based methods leverage domain knowledge or image enhancement techniques for dehazing. A widely adopted approach is the dark channel prior (DCP) [5], which exploits the tendency of haze-free images to contain at least one dark channel per local patch, whereas hazy regions exhibit higher brightness. The transmission map is estimated via a channel-wise minimum operation followed by a minimum filter. Global atmospheric light is typically selected from the top 0.15% of pixels with the highest intensities in the dark channel [5,11,14,16].

Despite its effectiveness, DCP struggles with sky regions and bright objects, where its assumptions fail. Furthermore, DCP and other priors, such as the color attenuation prior (CAP) [7] and color ellipsoid prior (CEP) [28], are derived from the analysis of small image patches, limiting their performance on full images. In contrast, the haze-lines prior [9] globally models pixel distributions in RGB space. Clean images form compact clusters, while hazy images extend into haze-lines, whose intersection estimates atmospheric light. The transmission map is derived from the haze-lines' distance to the origin. Although more robust than DCP, the haze-lines prior can introduce color distortions under heterogeneous lighting conditions.

Another category involves image enhancement techniques. For example, ref. [8] applied artificial under-exposure and multiscale image fusion to enhance contrast and suppress haze. Gamma correction with varying gamma values was used to generate under-exposed images, which were fused based on pixel-wise contrast and saturation. A follow-up study [12] improved this approach by integrating the dark channel into the fusion process. Similarly, dark channel information and multiscale image fusion were combined in [29] for both daytime and nighttime image dehazing applications.

While non-learning-based methods are computationally efficient and visually interpretable, they rely on engineered priors, limiting generalizability. For example, DCP fails in bright regions, where its assumptions do not hold.

1.2. Learning-Based Dehazing

To overcome prior-based limitations, deep neural networks (DNNs) have been widely explored. One of the earliest models, DehazeNet [18], used a three-stage convolutional neural network (CNN) to estimate the transmission map, while atmospheric light was derived from DCP. Despite moderate complexity, DehazeNet improved upon prior-based methods but suffered from domain shift due to the lack of real hazy/haze-free image pairs for training.

A subsequent model [19] incorporated multiscale processing, where a CNN branch with large kernels produced a coarse estimate of the transmission map, refined by another branch with smaller kernels to recover fine details. To enforce smoothness within objects, holistic edge priors were employed. While this improved accuracy, domain shift remained an issue.

To address this, unsupervised models, such as YOLY [21] (You Only Look Yourself), adopted a layer disentanglement strategy [30], jointly estimating the transmission map,

atmospheric light, and scene radiance, reconstructing the hazy input using Equation (1). Alternatively, unpaired learning models, such as D^4 [22,31], leveraged arbitrary hazy and haze-free images for training. D^4 estimated the atmospheric scattering coefficient during dehazing and applied it for synthetic rehazing, improving robustness. A later extension [31] incorporated ego-motion synthesis and depth estimation for video dehazing. However, differentiating bright regions from dense haze remained challenging.

Recent advances include conditional variational autoencoders [27], vision transformers [23,32], and diffusion models [26,33], yet domain shift continues to hinder real-world deployment. In summary, while learning-based dehazing improves generalizability over non-learning-based methods, challenges such as high computational cost and domain shift persist, necessitating future research for practical applications.

2. Real-Time Image Dehazing

NTSC (National Television System Committee) and PAL (Phase Alternating Line), the predominant analog television standards, define frame rates of 30 and 25 frames per second (fps), respectively, setting the benchmark for real-time video processing. Despite the transition to digital ATSC (Advanced Television Systems Committee) standards, these frame rate requirements remain unchanged. As real-time processing is essential for practical deployment, achieving these frame rates is a critical criterion for image processing algorithms.

Table 2 presents the processing speeds of various dehazing algorithms at different resolutions, measured in fps. Six benchmark methods with publicly available source codes and parameter configurations were evaluated in MATLAB R2019a and Python 3.9.9 (PyTorch 1.12.0 + cu116). All experiments were conducted on a computer with an Intel Core i9-9900K (3.6 GHz) CPU, 64 GB RAM, and an Nvidia TITAN RTX GPU.

Table 2. Processing speeds (frames per second, fps) of various dehazing methods across different image resolutions. NA indicates that the processing speed could not be measured due to memory exhaustion.

Method	Resolution	VGA	SVGA	HD	FHD	DCI 4K	8K UHD
		640 × 480	800 × 600	1024 × 768	1920 × 1080	4096 × 2160	7680 × 4320
Non-learning	[6]	3.571	1.695	1.316	0.662	0.111	0.027
	[7]	4.545	2.941	1.563	0.662	0.156	0.040
	[13]	10.000	5.882	3.704	1.471	0.339	0.090
Learning	[18]	0.654	0.418	0.258	0.094	0.021	0.006
	[19]	1.852	1.136	0.654	0.292	0.056	0.004
	[21]	0.005	0.003	0.001	0.001	NA	NA

Results indicate that non-learning-based methods are generally faster than learning-based ones, yet even the fastest achieves only 10 fps at VGA resolution, failing to meet real-time requirements. Although lightweight and video-specific DNNs [31,34] exist, they require power-intensive GPUs and remain inadequate for real-time processing.

Another runtime comparison in [28] reports the fastest non-learning-based method, implemented in C/C++, with an execution time of 0.12 s/Mpixels, corresponding to 27.127 fps at VGA resolution, outperforming MATLAB implementations listed in Table 2 and meeting real-time requirements. However, performance drops to 17.361 fps at SVGA resolution, falling short of real-time constraints.

FPGAs offer a promising solution for real-time image dehazing due to their low power consumption and reconfigurability. Table 3 summarizes existing FPGA-based implementations, most of which leverage non-learning methods like DCP for efficient hardware realization. A learning-based FPGA implementation was presented in [35], but

its simplistic CNN architecture, primarily using 1×1 kernels, introduces artifacts such as color distortion, limiting its practicality.

Table 3. Summary of real-time image dehazing implementations.

Year	Paper	Platform	Core Technique(s)	Adopted Tool(s)	Remarks
2013	[36]	FPGA (Altera Stratix EP1S10F780C6)	Filtering	Verilog HDL	- Replacing soft matting in DCP with an edge-preserving filter - Designing an 11-stage pipelined FPGA accelerator - Achieving a throughput of 58.43 Mpixels/s
2015	[37]	DSP (TMS320C6678)	Filtering	C/C++ (OpenMP)	- Direct implementation of DCP - Achieving 21.277 fps at 600×600 resolution
2016	[38]	FPGA (Altera Stratix EP1S10F780C6)	Filtering	Verilog HDL	- Direct implementation of DCP - Designing a 15-stage pipelined FPGA accelerator - Achieving a throughput of 116 Mpixels/s
2017	[39]	GPU (NVIDIA Tegra K1)	Total variation regularization	C/C++ (OpenCV and CUDA)	- Estimating the transmission map using locally adaptive neighborhoods and order statistics - Achieving 32.4 fps at 800×600 resolution
2018	[40]	FPGA (Altera Cyclone II)	Filtering	NA	- Simplifying DCP to improve processing speed - Achieving 4.386 fps at 320×240 resolution
2019	[41]	ASIC (TSMC's 0.13 μm)	Filtering	Verilog HDL	- Replacing soft matting in DCP with an edge-preserving filter - Adjusting atmospheric light between successive frames to avoid flickering - Designing a 7-stage pipelined ASIC accelerator - Achieving a throughput of 200 Mpixels/s
2021	[35]	FPGA (Xilinx XC7Z020-3CLG484)	CNN	Verilog HDL	- Adopting a lightweight CNN to estimate the transmission map from the input image and its dark channel - Achieving a throughput of 200 Mpixels/s
2022	[12]	FPGA (Xilinx XC7Z045-2FFG900)	Image fusion	Verilog HDL	- Fusing the input image and its dehazed result to achieve autonomous dehazing - Developing a self-calibrating factor to guide the fusion process - Achieving a throughput of 271.37 Mpixels/s
2023	[42]	FPGA (Xilinx XC7Z020-CLG484-1)	Filtering	Verilog HDL	- Estimating the pixel-wise transmission map from saturation - Downsampling the input image before estimating atmospheric light - Designing a 7-stage pipelined FPGA accelerator - Achieving a throughput of 85.2 Mpixels/s
2024	[43]	FPGA (Xilinx XC7K325T-2FFG900C)	Image fusion	NA	- Fusing high-boost filtering and linear stretching results of the input image to achieve dehazing - Achieving a throughput of 72.299 Mpixels/s
2024	[44]	FPGA (Xilinx XC7Z020-CLG484-1)	Filtering	Simulink HLS	- Simplifying DCP to improve processing speed - Incorporating color cast correction into the dehazing process - Achieving a throughput of 50 Mpixels/s

This paper presents an improved real-time FPGA-based dehazing implementation, addressing limitations in our prior work [12]. The previous approach neglected spatial haze variations during fusion due to the inability to compute fusion weights within the active frame interval. The proposed method leverages the vertical blanking interval to precompute fusion weights that account for local haze distribution. This enhancement improves dehazing quality while maintaining real-time performance, achieving 45.34 fps at DCI 4K resolution.

The main contributions of this paper are as follows:

- A novel hardware architecture exploiting the vertical blanking interval for real-time video processing, mitigating flickering and enabling high-quality autonomous image dehazing;
- A fast and compact FPGA implementation capable of processing DCI 4K video at 45.34 fps.

3. Proposed FPGA Implementation

Figure 1 presents the block diagram of the proposed autonomous dehazing system. The input image undergoes preprocessing via unsharp masking, followed by dehazing using an improved color attenuation prior (ICAP). The dehazing process is controlled by a self-calibrating weight to ensure the dehazing strength adapts to the input haze

conditions. A haziness degree evaluator [45] calculates the average haze density and a patch-based haze density map, which are used to compute the self-calibrating weight and local blending weights, respectively. The input image is then fused with its dehazed counterpart using the blending weights. The fusion result undergoes postprocessing through adaptive tone remapping, also controlled by the self-calibrating weight, to maintain enhancement consistency with the input haze conditions.

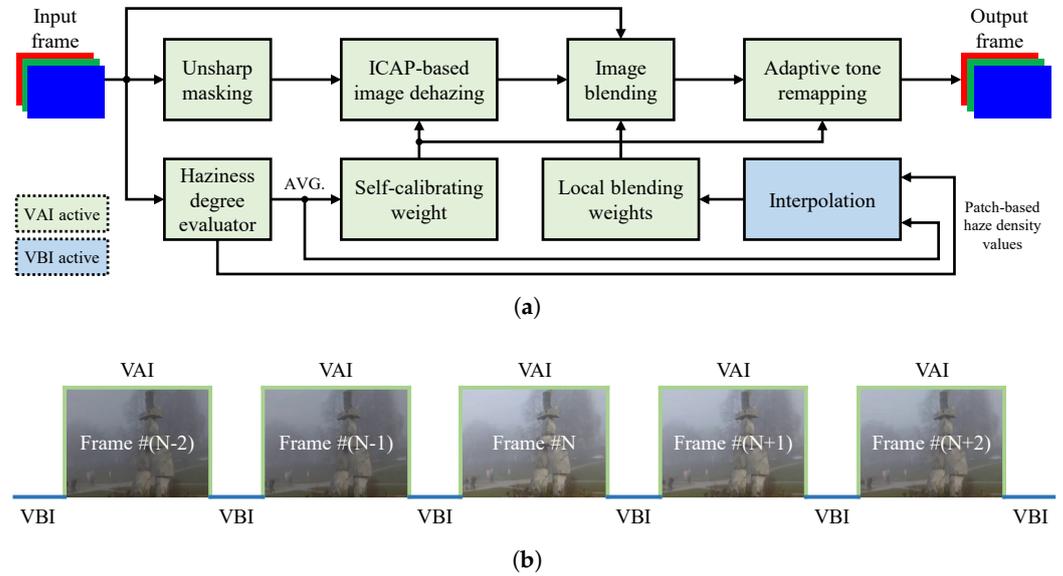


Figure 1. Overview of the proposed FPGA implementation. (a) Simplified block diagram. (b) Input frame data with video active signal. ICAP, VAI, and VBI stand for improved color attenuation prior, video active interval, and video blanking interval, respectively.

Figure 1a illustrates the operational timeline of the system, where green blocks correspond to processes executed during the video active interval (VAI), and the blue block represents computations performed during the video blanking interval (VBI). The VAI and VBI are depicted in Figure 1b. The blue block, a key contribution of this work, differentiates the proposed system from our previous design [12]. It computes local blending weights based on patch-based haze density values, enabling haze-aware fusion and significantly improving autonomous dehazing performance. Consequently, this component is the primary focus of this section.

3.1. Unsharp Masking

Haze is generally a smooth, low-frequency component in an image, except at depth discontinuities, which reduces the clarity of image details. To counteract this effect, unsharp masking (UM) is applied. As the luminance channel captures shapes, patterns, and fine details of objects in the scene, the UM process is performed on the luminance channel, which is obtained through the following color space conversion:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.183 & 0.614 & 0.062 \\ -0.101 & -0.338 & 0.439 \\ 0.439 & -0.399 & -0.040 \end{bmatrix} \begin{bmatrix} I^R \\ I^G \\ I^B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}, \quad (2)$$

where $\{I^R, I^G, I^B\}$ are the red, green, and blue color channels of the input image \mathbf{I} , respectively, Y is the luminance, and $\{Cb, Cr\}$ are the chrominance components.

The enhancement formula for image detail is given by:

$$Y_e = Y + \omega \cdot e, \tag{3}$$

where Y_e represents the luminance of the enhanced image, ω is the scaling weight, and e denotes the image details. The scaling weight ensures proportional enhancement, assigning higher weights to less detailed regions and lower weights to regions with richer details. This weight is computed as a piecewise linear function of the local variance v , bounded by two predefined thresholds, i.e., ω_1 and ω_2 , as shown in Equation (4). The local variance v is computed using Equation (5), where \otimes represents the convolution operation and U is the averaging kernel defined in Equation (6). Image details e are extracted by convolving the luminance channel with a Laplacian operator, as defined in Equation (8).

$$\omega = \begin{cases} \omega_1 & v < v_1 \\ \left(\frac{\omega_2 - \omega_1}{v_2 - v_1}\right)v + \frac{\omega_1 v_2 - \omega_2 v_1}{v_2 - v_1} & v_1 \leq v \leq v_2 \\ \omega_2 & v > v_2 \end{cases} \tag{4}$$

$$v_s = Y^2 \otimes U - (Y \otimes U)^2, \tag{5}$$

$$U \triangleq \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9, \tag{6}$$

$$e = Y \otimes \nabla^2, \tag{7}$$

$$\nabla^2 \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{8}$$

After enhancement, the image is converted back to the RGB color space for the subsequent dehazing step. The conversion is defined as:

$$\begin{bmatrix} I_e^R \\ I_e^G \\ I_e^B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.793 \\ 1.164 & -0.213 & -0.534 \\ 1.164 & 2.115 & 0 \end{bmatrix} \begin{bmatrix} Y_e - 16 \\ Cb - 128 \\ Cr - 128 \end{bmatrix}. \tag{9}$$

Figure 2 illustrates the simplified data path of the unsharp masking process. Modules represented by plain blocks perform simple arithmetic operations, which are easily implemented using building blocks such as adders, multipliers, and multiplexers. Shaded blocks, on the other hand, involve two-dimensional filtering and require line memories to access pixel neighborhoods within the kernel. For clarity and to maintain the readability of the main text, the detailed implementation of image filters is provided in Appendix A.

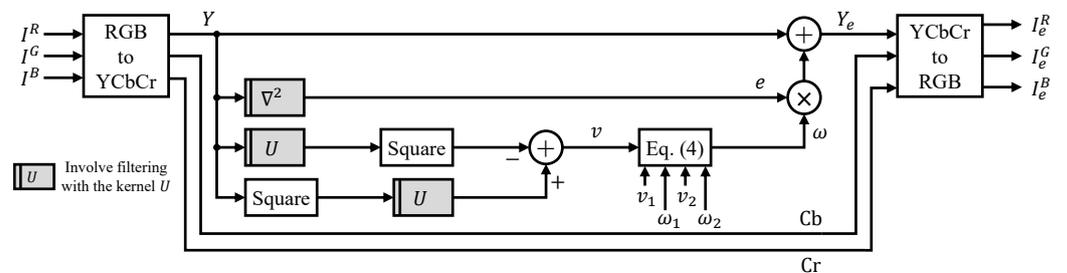


Figure 2. Simplified data path of the unsharp masking process. Plain modules perform pixel-wise operations, while shaded modules involve two-dimensional filtering (refer to Appendix A).

3.2. Dehazing via Improved Color Attenuation Prior

Given the preprocessed image $\mathbf{I}_e = \{I_e^R, I_e^G, I_e^B\}$, the dehazing process can be performed by isolating \mathbf{J} from the Koschmieder model, as follows:

$$\mathbf{J} = \frac{\mathbf{I}_e - \mathbf{A}}{t} + \mathbf{A}. \tag{10}$$

Two variables, \mathbf{A} and t , need to be determined for the dehazing process. Regarding \mathbf{A} , it is typically approximated as the brightest pixel in the image, that is, $\mathbf{A} = \{255, 255, 255\}$. However, in cases when the input image contains shiny objects or artificial light sources, setting $\mathbf{A} = \{255, 255, 255\}$ may lead to over-dehazing, which darkens the entire image and causes the loss of shadow details. In the proposed system, this limitation of using $\mathbf{A} = \{255, 255, 255\}$ is well-addressed by incorporating adaptive tone remapping (described in Section 3.4), providing an effective solution to mitigate such undesirable effects.

The transmission map t is estimated using the improved color attenuation prior (ICAP) [10], chosen for its simplicity and effectiveness. As noted earlier, haze is depth-dependent and can be expressed as $t = \exp(-\beta_c d)$, where d is the scene depth and β_c is the atmospheric scattering coefficient. Accurately estimating β_c is highly complex, as it depends on the physical and chemical properties of suspended particles in the atmosphere [46]. Consequently, a common practice in image dehazing is to set $\beta_c = 1$. In ICAP, a linear model is employed to estimate d from saturation S and brightness V , as follows:

$$d = \theta_0 + \theta_1 S + \theta_2 V, \tag{11}$$

where $\{\theta_0, \theta_1, \theta_2\}$ are model parameters estimated through maximum likelihood estimation. The estimated scene depth is then refined using a modified hybrid median filter (mHMF), an edge-preserving smoothing filter, to ensure that the scene depth remains smooth except at depth discontinuities. As detailed in Section 3.5, the scene depth is further multiplied by a self-calibrating weight to adjust the dehazing strength according to the haze condition of the input image. Finally, before substituting the transmission map into Equation (10), a no-black-pixel (NBP) constraint [6] is applied to prevent undershoots that could occur during the dehazing process.

Figure 3 illustrates the simplified data path of the ICAP-based image dehazing process. Saturation and brightness values are computed using the formulas in Equations (12) and (13). The mHMF implementation is based on the optimized merging-sorting network [47]. The NBP constraint is applied using a filter based on the kernel U , defined in Equation (6), with further details provided in Appendix A. The computation and implementation of the self-calibrating weight are discussed in Section 3.5.

$$S = \frac{\max_{c \in \{R,G,B\}} I_e^c - \min_{c \in \{R,G,B\}} I_e^c}{\max_{c \in \{R,G,B\}} I_e^c}, \tag{12}$$

$$V = \max_{c \in \{R,G,B\}} I_e^c. \tag{13}$$

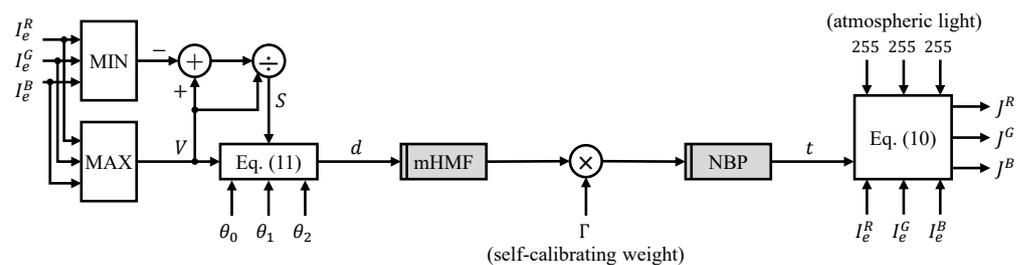


Figure 3. Simplified data path of the dehazing process. mHMF and NBP are abbreviations for the modified hybrid median filter and no-black-pixel, respectively.

ICAP is selected for its computational efficiency rather than dehazing performance, making it well-suited for real-time processing. However, its primary limitation stems from its simplicity. The linear equation for scene depth estimation assumes a typical scenario where haze density increases with distance from the camera. This assumption breaks down in dense or heterogeneous haze conditions, where haze distribution is not strictly correlated with scene depth. Consequently, Equation (11) fails to provide accurate estimates, leading to reduced performance.

3.3. Image Blending

The input image I and its dehazed counterpart J are fused using local blending weights α , as expressed in the following equation:

$$\mathbf{B} = \alpha \mathbf{J} + (1 - \alpha) \mathbf{I}, \quad (14)$$

where \mathbf{B} represents the blended result. The derivation of α is deferred to Section 3.5, as it constitutes the main contribution of this paper. The rationale for employing image blending can be summarized as follows:

- If the input image is haze-free, applying a dehazing operation introduces visual artifacts. In this case, $\alpha = 0$ ensures the input image remains unchanged.
- For images affected by mild or moderate haze, the dehazing strength must be controlled to prevent over-dehazing. Here, α is set to vary linearly between zero and unity, that is, $0 < \alpha < 1$.
- For densely hazed images, full-scale dehazing is required. Thus, $\alpha = 1$ is used to completely suppress the contribution of the input image in the blending process.

The implementation of the image blending process is straightforward, requiring only multipliers and adders, as illustrated in Figure 4.

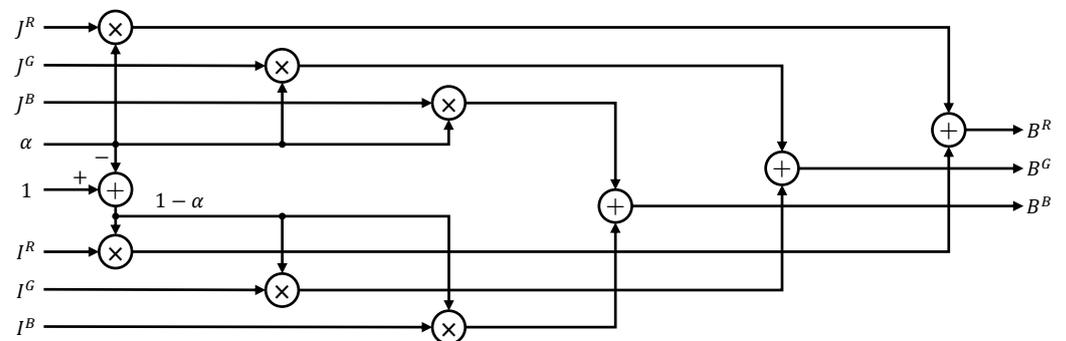


Figure 4. Simplified data path of the blending process.

3.4. Adaptive Tone Remapping

In layperson's terms, image dehazing involves removing haze from an image, which can make the image appear dimmer than before. As a result, a postprocessing step is necessary to enhance image quality. Specifically, as the luminance channel directly influences image brightness, luminance enhancement is applied to address this issue. However, luminance enhancement alone can lead to unnatural appearance; thus, chrominance expansion is also necessary. To achieve this, the proposed system employs adaptive tone remapping (ATR), which consists of two stages: luminance enhancement, followed by chrominance expansion to maintain a natural appearance. These steps are expressed as below:

$$Y_{\mathbf{B}_f} = Y_{\mathbf{B}} + \Gamma \cdot g_1(Y_{\mathbf{B}}) \cdot g_2(Y_{\mathbf{B}}), \quad (15)$$

$$\text{Ch}_{\mathbf{B}_f} = \text{Ch}_{\mathbf{B}} \left[1 + \frac{Y_{\mathbf{B}_f}}{Y_{\mathbf{B}}} \cdot g_3(Y_{\mathbf{B}}) \right], \quad (16)$$

where $\{Y_{\mathbf{B}}, Ch_{\mathbf{B}}\}$ and $\{Y_{\mathbf{B}_f}, Ch_{\mathbf{B}_f}\}$ denote the luminance and chrominance components of the image before (\mathbf{B}) and after (\mathbf{B}_f) ATR. The self-calibrating weight Γ , introduced in Section 3.5, Equation (24), ensures that the enhancement is proportional to the amount of haze removed. If the input image is haze-free, no dehazing is performed, leading to $\Gamma = 0$ and no enhancement. Conversely, for hazy images, $0 < \Gamma \leq 1$ is enforced to proportionally scale the enhancement according to the dehazing strength. Chrominance (Ch) is derived by subsampling the blue-difference (Cb) and red-difference (Cr) chroma channels according to the 4:2:2 ratio.

The luminance enhancement formula involves two components: a non-linear gain $g_1(\cdot)$ and a linear weight $g_2(\cdot)$, as defined in Equations (17) and (18). The non-linear gain is derived from the adaptive luminance point (ALP), which constrains the luminance range to prevent over-enhancement. The user-defined exponent θ is used for performance tuning. The linear weight is a linear function of the luminance $Y_{\mathbf{B}}$, with its slope and intercept determined by two parameters, m and b .

$$g_1(Y_{\mathbf{B}}) = \frac{Y_{\mathbf{B}}}{221} \left[255 \left(1 - \frac{Y_{\mathbf{B}} - \text{ALP}}{255} \right)^{\theta} \left(\frac{255 - Y_{\mathbf{B}}}{255} \right) \right]^2, \quad (17)$$

$$g_2(Y_{\mathbf{B}}) = \frac{m}{255} Y_{\mathbf{B}} + b. \quad (18)$$

The adaptive luminance point ALP is computed as:

$$\text{ALP} = \begin{cases} 0.04 + \frac{0.02}{255}(L_{0.9} - L_{0.1}) & \bar{Y}_{\mathbf{B}} > 128 \\ 0.04 - \frac{0.02}{255}(L_{0.9} - L_{0.1}) & \bar{Y}_{\mathbf{B}} \leq 128 \end{cases}, \quad (19)$$

where $\bar{Y}_{\mathbf{B}}$ is the average luminance and L_k is the luminance value corresponding to a cumulative distribution function (CDF) such that $\text{CDF}(L_k) = k$, with $0 \leq k \in \mathbb{R} \leq 1$.

The chrominance expansion formula involves a piecewise linear function $g_3(\cdot)$ of the luminance $Y_{\mathbf{B}}$, as defined in Equation (20). The thresholds L_{high} and L_{low} are predefined luminance bounds. According to the Helmholtz–Kohlrausch effect [48], luminance enhancement narrows the color gamut in chromaticity coordinates. To mitigate this effect, the ratio $Y_{\mathbf{B}_f} / Y_{\mathbf{B}}$ is multiplied by $g_3(\cdot)$ to counterbalance the narrowing of the color gamut. The appearance of $Y_{\mathbf{B}_f} / Y_{\mathbf{B}}$ in Equation (16) enables the self-calibrating weight Γ to implicitly affect chrominance expansion as well.

$$g_3(Y_{\mathbf{B}}) = \begin{cases} 0.7 & Y_{\mathbf{B}} < L_{\text{low}} \\ 0.7 - 0.26 \frac{Y_{\mathbf{B}} - L_{\text{low}}}{L_{\text{high}} - L_{\text{low}}} & L_{\text{low}} \leq Y_{\mathbf{B}} \leq L_{\text{high}} \\ 0.44 & Y_{\mathbf{B}} > L_{\text{high}} \end{cases}. \quad (20)$$

Figure 5 illustrates the simplified data path of adaptive tone remapping. The color space conversions are the same as described in Section 3.1, except that the blue-difference and red-difference chroma components are combined into a single chroma channel using the 4:2:2 subsampling scheme. The subsampling process is efficiently implemented using multiplexers. The average luminance $\bar{Y}_{\mathbf{B}}$ and the CDF are computed on a frame-wise basis, leveraging the high temporal similarity between consecutive video frames. This eliminates the need for frame buffers, thereby optimizing resource usage.

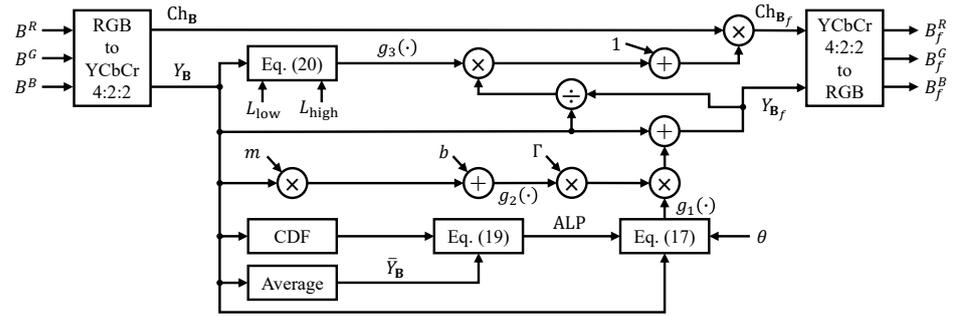


Figure 5. Simplified data path of the adaptive tone remapping process. CDF is the abbreviation for cumulative distribution function.

3.5. Haze Density Estimation and Haze-Aware Weighting

For achieving autonomous dehazing, estimating a haze density map from an arbitrary input image is a critical prerequisite. In the proposed system, a haziness degree evaluator [45] is employed to compute the haze density map as follows:

$$\rho_I = 1 - \hat{t}, \tag{21}$$

where \hat{t} is obtained by optimizing the cost function $O(t)$ in Equation (22). This cost function is formulated using image features such as saturation S , brightness V , sharpness σ , and the dark channel D , all of which are expressed as functions of the transmission map t . The regularization parameter is denoted as λ . Optimization of $O(t)$ aims to determine \hat{t} such that saturation, brightness, and sharpness are maximized, while the dark channel is minimized.

$$O(t) = \frac{S(t)V(t)\sigma(t)}{D(t)} + \frac{\lambda}{t}. \tag{22}$$

To maintain conciseness, interested readers are referred to [45] (Section 3.4 and Appendix A) for a detailed explanation of the derivation of \hat{t} . The formula for computing the haze density map is given as follows:

$$\rho_I = I_{m\Psi} + \frac{I_{mc}v}{\lambda} - \sqrt{\frac{I_{mc}v}{\lambda} \left(\frac{I_{mc}v}{\lambda} - 255 + I_{m\Psi} \right)}, \tag{23}$$

where $I_{m\Psi} = \min_{(x,y) \in \Psi} \left[\min_{c \in \{R,G,B\}} I^c(x,y) \right]$ is the result of applying a minimum filter to the minimum channel, $I_{mc} = \max_{c \in \{R,G,B\}} I^c - \min_{c \in \{R,G,B\}} I^c$ is the difference between the maximum and minimum color channels, and v is the local variance as defined in Equation (5).

The self-calibrating weight Γ is defined as a piecewise function of the average haze density $\bar{\rho}_I$, expressed as:

$$\Gamma = \begin{cases} 0 & \bar{\rho}_I \leq \rho_1 \\ \left(\frac{\bar{\rho}_I - \rho_1}{\rho_2 - \rho_1} \right)^n & \rho_1 < \bar{\rho}_I \leq \rho_2, \\ \left(\frac{\Gamma_u - 1}{1 - \rho_2} \right) (\bar{\rho}_I - \rho_2) + 1 & \bar{\rho}_I > \rho_2 \end{cases}, \tag{24}$$

where ρ_1 and ρ_2 are two user-defined thresholds. For $\rho_I \leq \rho_1$, $\Gamma = 0$ is used to zero the dehazing strength, allowing the input haze-free image to remain unchanged. For $\rho_1 < \bar{\rho}_I \leq \rho_2$, Γ increases exponentially, controlled by the exponent n (set to 0.1), allowing for a gradual increase in dehazing strength from mild to moderate haze conditions. For $\bar{\rho}_I > \rho_2$, Γ varies linearly, capped by an empirically determined upper bound $\Gamma_u = 1.2$. The

final transmission map is modified as $t = \exp(-\beta_c \Gamma d)$, enabling dynamic control of the dehazing process.

To account for the spatial heterogeneity of haze, the input image is divided into 8×8 patches. The local haze density ρ_{Ω_i} for the i -th patch Ω_i is computed as:

$$\rho_{\Omega_i} = \max \left(\bar{\rho}_I, \frac{1}{|\Omega_i|} \sum_{\forall (x,y) \in \Omega_i} \rho_I(x,y) \right), \quad (25)$$

where $|\Omega_i|$ represents the number of pixels in Ω_i . Based on the local haze density, the local blending weight α_i is defined as a piecewise linear function, with thresholds ρ_1 and ρ_2 consistent with the definitions in Equation (24).

$$\alpha_i = \begin{cases} 0 & \rho_{\Omega_i} < \rho_1 \\ \frac{\bar{\rho}_I - \rho_1}{\rho_2 - \rho_1} & \rho_1 \leq \rho_{\Omega_i} \leq \rho_2 \\ 1 & \rho_{\Omega_i} > \rho_2 \end{cases}. \quad (26)$$

Figure 6 demonstrates the impact of abrupt transitions in local haze densities on blended results, causing blocky artifacts. On the left of Figure 6 is a hazy image superimposed by local haze densities. The pink rectangle masks a region with abrupt transitions in haze densities: $0.1531 \rightarrow 0$ and $0.1122 \rightarrow 0$. These values were min–max normalized for ease of interpretation. The image on the right is the blended result obtained using 8×8 local weights. It is observed that abrupt transitions cause blocky artifacts, as shown by the pink rectangle in the blended result.

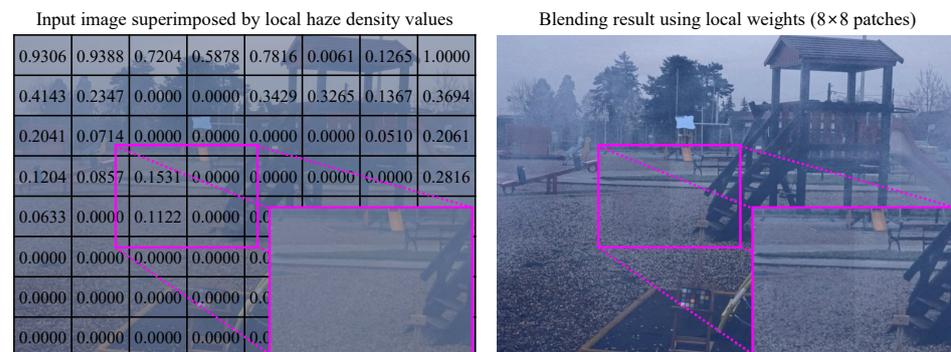


Figure 6. Input image and the corresponding blended result using 8×8 weights. Local haze densities are min–max normalized for ease of interpretation.

To mitigate this problem, a 2×2 low-pass filter is applied to smooth transitions, followed by $4 \times$ bilinear interpolation to upscale local haze densities from 8×8 to 29×29 . Figure 7 illustrates how this process significantly reduces blocky artifacts by ensuring more gradual transitions between patches. Considering the same region as in Figure 6, the 2×2 local haze densities turn into 5×5 , demonstrating more gradual transitions both horizontally and vertically. Using the interpolated results, blocky artifacts are now effectively eliminated, allowing the proposed system to produce high-quality images.

Figure 8 illustrates the simplified data path for computing the self-calibrating weight and local blending weights. To optimize performance, the 2×2 low-pass filter and $4 \times$ interpolation modules are implemented during the video blanking interval (VBI) using a dual-port (DP) RAM for 8×8 local haze density values and a single-port (SP) RAM for 29×29 interpolated results. During the video active interval (VAI), the system computes 8×8 local haze densities, stores them in DP RAM, and retrieves interpolated results from SP RAM, as illustrated in Figure 9. Executing the low-pass filter and $4 \times$ bilinear interpolation during VBI ensures that the interpolated haze densities are available for the next frame;

hence, the name VBI-accelerated of the proposed system. The detailed implementation of the 2×2 low-pass filter is provided in Appendix B.

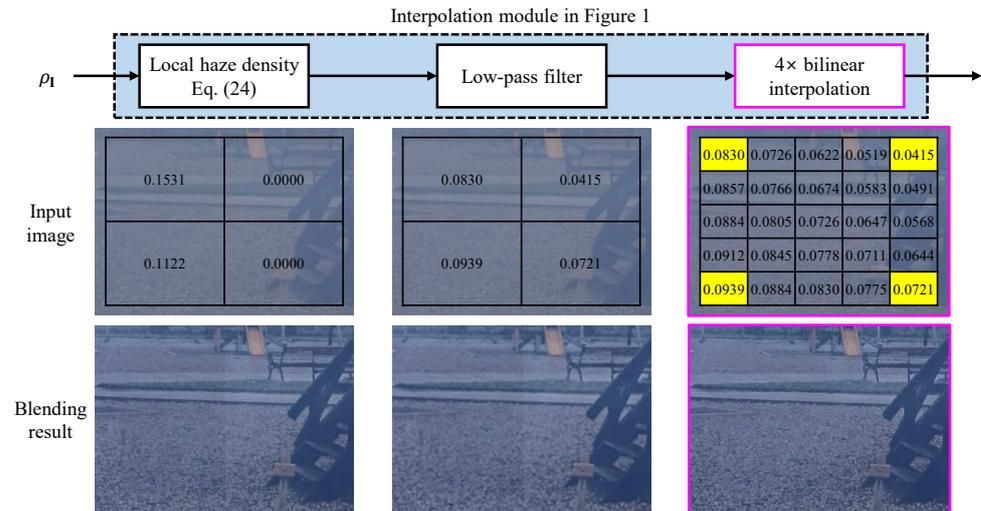


Figure 7. Effect of interpolation on the blended result. Local haze densities are min–max normalized for ease of interpretation.

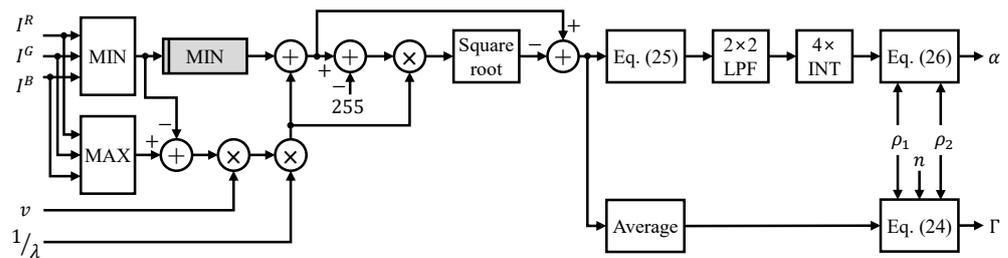


Figure 8. Simplified data path for computing the self-calibrating weight and local blending weights. LPF and INT are abbreviations for low-pass filter and interpolation, respectively.

The $4 \times$ bilinear interpolation is implemented using two architectures: INTERP_1 , a naive implementation of bilinear interpolation, and INTERP_2 , an optimized design leveraging overlapping sliding windows. Details about these two architectures are provided in Appendix C. Implementation on an XCZU7EV-2FFVC1156 MPSoC device [49], obtained using Vivado 2023.1 [50], is summarized in Table 4. It demonstrates that INTERP_2 achieves higher efficiency, occupying fewer hardware resources and operating at higher frequencies. Consequently, INTERP_2 is adopted in the proposed system.

Table 4. Implementation results of the $4 \times$ interpolation module. INTERP_1 refers to the formula-based architecture, while INTERP_2 denotes the proposed architecture that leverages overlapping sliding windows. LUT stands for look-up table and the symbol # represents quantities.

Xilinx Vivado v2023.1					
Device		XCZU7EV-2FFVC1156-2-E			
Design		INTERP_1		INTERP_2	
Slice Logic Utilization	Available	Used	Utilization	Used	Utilization
Slice registers (#)	460,800	1944	0.42%	1506	0.33%
Slice LUTs (#)	230,400	1663	0.72%	1350	0.59%
Block RAMs (#)	312	0.5	0.16%	0.5	0.16%
Minimum period (ns)	-	1.414		1.397	
Maximum frequency (MHz)	-	707.214		715.820	

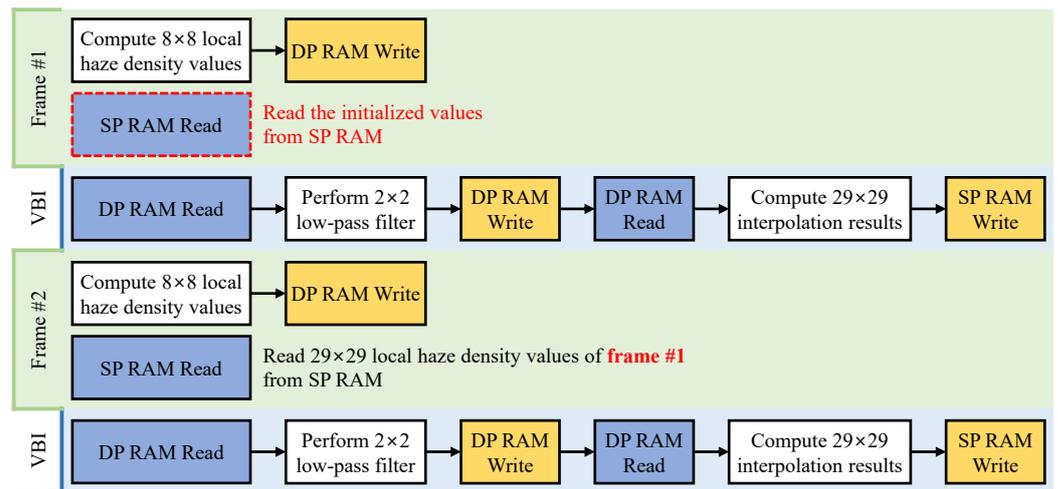


Figure 9. Proposed implementation scheme of the 2×2 low-pass filter and $4 \times$ interpolation. DP RAM and SP RAM refer to dual-port RAM and single-port RAM, respectively.

4. Evaluation

This section presents a comparative evaluation of the proposed autonomous dehazing system against six benchmark methods, abbreviated as DCP [5], CAP [7], DehazeNet [18], YOLY [21], MB-TaylorFormer [32], and FCDM (Frequency Compensated Diffusion Model) [33]. The evaluation encompasses both qualitative and quantitative analyses to assess dehazing effectiveness. Additionally, FPGA implementation results are examined and compared with existing designs (Section 2) to demonstrate real-time processing capabilities. Finally, the practical utility of the proposed system is validated through its application in aerial object detection, highlighting its potential for real-world deployment.

Table 5 summarizes the parameters of the proposed system and their settings, which are used in all subsequent experiments.

Table 5. Summary of the parameters of the proposed autonomous dehazing system. The self-calibrating weight and local blending weight processes share the same values for ρ_1 and ρ_2 .

Process	Parameter	Value	Remark	
Unsharp masking	v_1	0.001	Scaling weight calculation in Equation (4)	
	v_2	0.010		
	ω_1	2.5		
	ω_2	1.0		
ICAP-based dehazing	θ_0	0.180069	Scene depth estimation in Equation (11)	
	θ_1	1.014740		
	θ_2	-0.734965		
Haziness degree evaluator	λ	-1	Regularization parameter in Equation (23)	
Self-calibrating weight and Local blending weights	n	0.1	Self-calibrating calculation in Equation (24) and local blending weight calculation in Equation (26)	
	ρ_1	0.8811		
Adaptive tone remapping	ρ_2	0.9344	$g_1(\cdot)$ in Equation (17), θ depends on the input image	
	θ	$1.5 / (\bar{Y}_{\mathbf{B}} - L_{0.1})$		
	m	5		$g_2(\cdot)$ in Equation (18)
	b	6		
	L_{low}	128		$g_3(\cdot)$ in Equation (20)
L_{high}	255			

4.1. Qualitative Evaluation

Figure 10 illustrates the results of the proposed autonomous dehazing system and six benchmark methods applied to five natural images. Two of these images are haze-free,

while the remaining three are affected by mild, moderate, and dense haze, respectively. The average haze density $\bar{\rho}_1$ is compared against two thresholds, $\rho_1 = 0.8811$ and $\rho_2 = 0.9344$, to determine the haze condition.

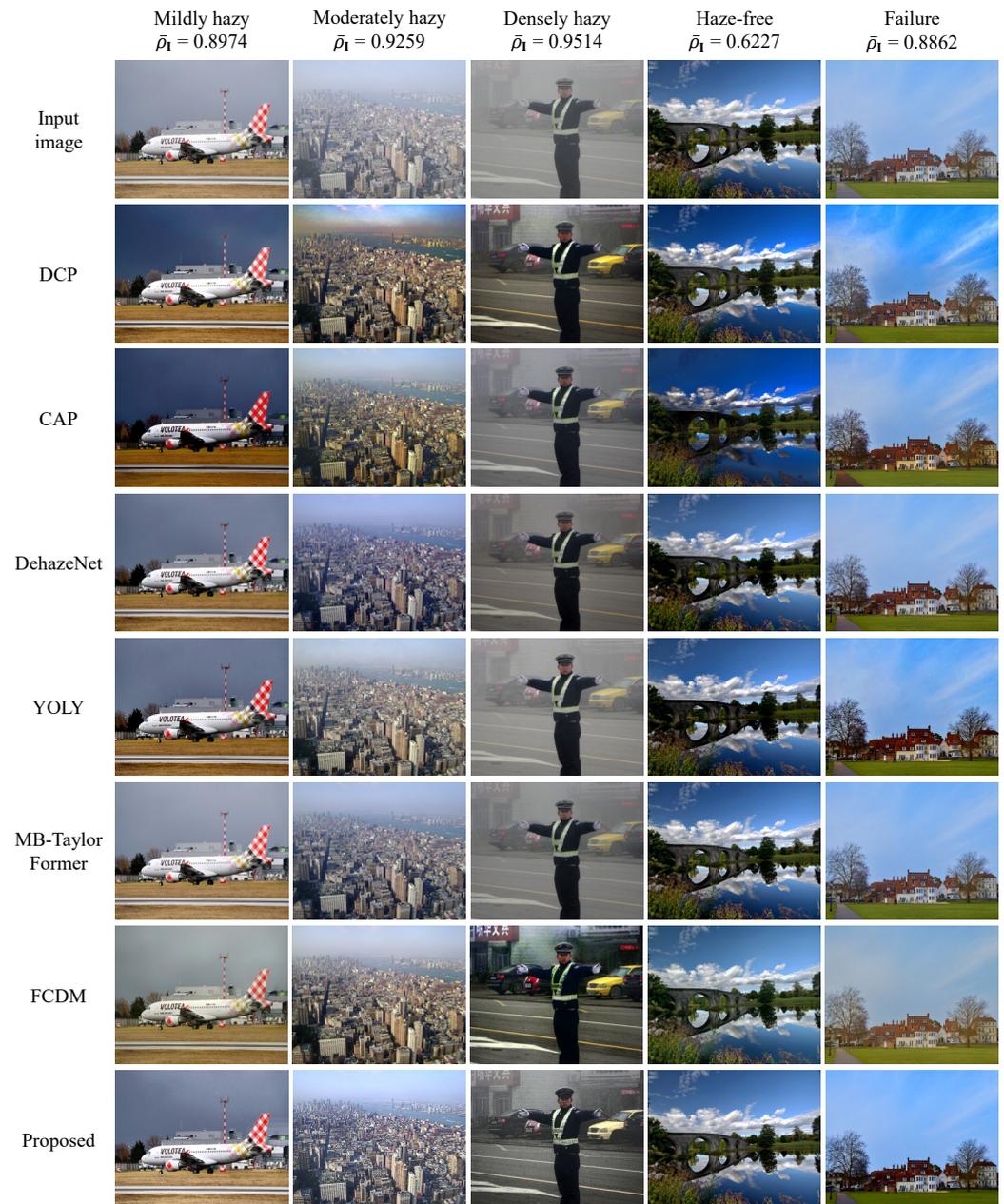


Figure 10. Qualitative evaluation results of the proposed system and six benchmark methods on haze-free, mildly hazy, moderately hazy, and densely hazy images. The average haze density $\bar{\rho}_1$ was compared against two thresholds, $\rho_1 = 0.8811$ and $\rho_2 = 0.9344$, to determine the haze condition.

In the first column, corresponding to a mildly hazy scenario, the over-dehazing limitations of benchmark methods are evident, particularly for non-learning-based approaches such as DCP and CAP. Significant color distortions are observed in the cloudy sky, aircraft, and grass. Although learning-based methods—DehazeNet, YOLY, MB-TaylorFormer, and FCDM—mitigate this issue to some extent, color changes in the grass and sky remain noticeable. In contrast, the proposed system accurately categorizes the image as mildly hazy and applies reduced dehazing strength, effectively preventing over-dehazing artifacts.

For the moderately hazy image in the second column, DCP, CAP, and DehazeNet continue to exhibit color distortion due to over-dehazing. Meanwhile, YOLY, MB-TaylorFormer,

FCDM, and the proposed system achieve more favorable results, effectively removing haze without introducing significant artifacts. In the dense haze scenario (third column), DCP, CAP, DehazeNet, and FCDM demonstrate strong dehazing performance, whereas YOLY and MB-TaylorFormer leave visible residual haze. The reduced performance of YOLY and MB-TaylorFormer can be attributed to domain shift, where the input data differ from the training set, impairing their effectiveness. FCDM addresses this issue through a sophisticated data augmentation strategy, generating various haze levels and swapping haze statistics in the low-frequency component between images. This approach enhances generalizability and mitigates domain shift. In contrast, the proposed system employs a simpler yet effective strategy, leveraging the self-calibrating weight for adaptive dehazing strength and local blending weights to enhance textual details, yielding robust performance across both moderate and dense haze conditions.

The fourth column displays a haze-free image where no dehazing should be applied. However, benchmark methods, unable to discern the haze-free condition, perform unnecessary dehazing, resulting in color distortion in the sky. This issue is more pronounced in non-learning-based methods. Notably, FCDM, the most recent learning-based method, introduces minimal color changes and produces an output nearly identical to the input. The proposed system correctly identifies the image as haze-free, setting the dehazing strength to zero and preserving the original image without introducing artifacts.

In the fifth column, another haze-free image is presented. This image contains a large sky region with a haze-like cloud veil, leading to a false positive in the haziness degree evaluation, as discussed in [45], where the haze-free image is misclassified as hazy. Consequently, the proposed system miscategorizes the image as mildly hazy, applying dehazing and causing a slight loss of fine details in the tree twigs. This issue affects most methods, except for MB-TaylorFormer, which produces the most visually satisfactory result in this case.

Figure 11 demonstrates the performance of the proposed system on an aerial image and three synthetic hazy variants generated using the process described in [12]. While differences among the seven methods are visually discernible, determining the best approach remains challenging, as none introduce significant artifacts. However, the proposed system effectively removes haze while preserving color fidelity, whereas benchmark methods introduce varying degrees of color alteration. Section 4.4 further evaluates the impact of dehazing on YOLOv9's performance in aerial object detection.

4.2. Quantitative Evaluation

To conduct a quantitative assessment, the tone-mapped image quality index (TMQI) [51] and the feature similarity extended to color images (FSIMc) [52] are employed. TMQI and FSIMc are full-reference image quality assessment metrics, ranging from zero to unity, where higher values indicate better performance. Five public datasets—FRIDA2 [6], D-HAZY [53], O-HAZE [54], I-HAZE [55], and Dense Haze [56]—are used to compute the average TMQI and FSIMc values. A summary of these datasets is provided in Table 6, while the corresponding average TMQI and FSIMc values are presented in Table 7.

Table 6. Summary of the five public datasets used in the quantitative evaluation.

Dataset	Haze-Free (#)	Hazy (#)	Remark
FRIDA2	66	264	Road scene images generated by computer graphics
D-HAZY	1472	1472	Synthetic indoor images
O-HAZE	45	45	Real outdoor images (haze generator)
I-HAZE	30	30	Real indoor images (haze generator)
Dense Haze	50	50	Real indoor and outdoor images (haze generator)

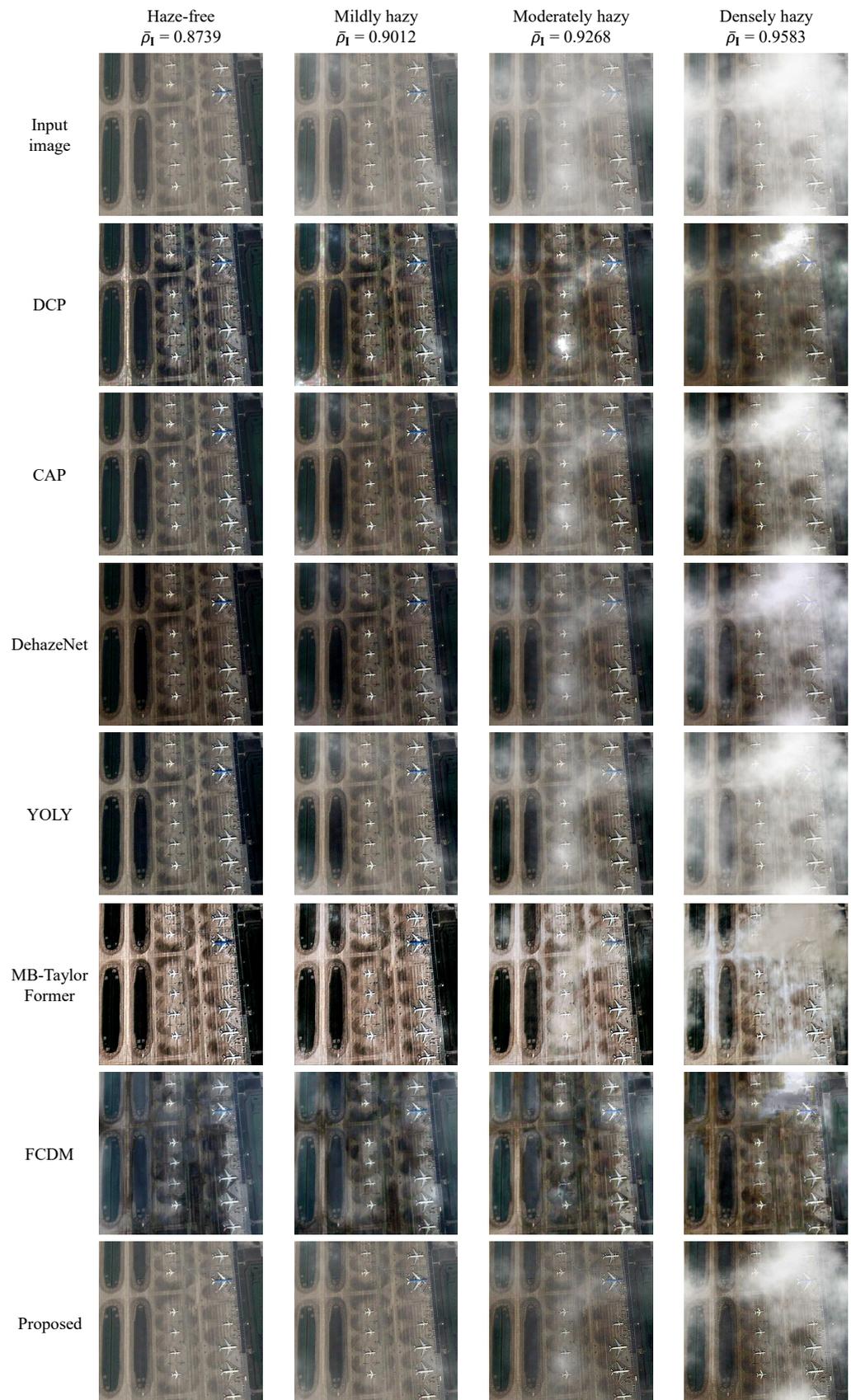


Figure 11. Qualitative evaluation results of the proposed system and six benchmark methods on an aerial image under haze-free, mildly hazy, moderately hazy, and densely hazy conditions. The average haze density $\bar{\rho}_1$ was compared against two thresholds, $\rho_1 = 0.8811$ and $\rho_2 = 0.9344$, to determine the haze condition.

Table 7. Average TMQI and FSIMc values computed on five public datasets. The best and second-best results are boldfaced and italicized, respectively. MB-TF is the abbreviation for MB-TaylorFormer.

Dataset	Method	DCP	CAP	DehazeNet	YOLY	MB-TF	Proposed
TMQI ↑	FRIDA2	0.7291	0.7385	0.7366	0.7176	0.7631	0.7345
	D-HAZY	0.8631	0.8206	0.7966	0.6817	0.7428	0.7861
	O-HAZE	0.8403	0.8118	0.8413	0.6566	0.8732	0.9048
	I-HAZE	0.7319	0.7512	0.7598	0.6936	0.8655	0.8319
	Dense-Haze	0.6383	0.5955	0.5723	0.5107	0.7237	0.6120
	Total	0.7357	0.7336	0.7312	0.6520	0.7761	0.7466
FSIMc ↑	FRIDA2	0.7746	0.7918	0.7963	0.7849	0.7158	0.8027
	D-HAZY	0.9002	0.8880	0.8874	0.7383	0.7727	0.8772
	O-HAZE	0.8423	0.7738	0.7865	0.6997	0.8420	0.8319
	I-HAZE	0.8208	0.8252	0.8482	0.7564	0.8692	0.8726
	Dense-Haze	0.6419	0.5773	0.5573	0.5763	0.7976	0.5869
	Total	0.7746	0.7693	0.7725	0.7111	0.7544	0.7865

FCDM is excluded from the quantitative evaluation due to its implementation constraints. Unlike the proposed system and other benchmark methods, which process variable-sized images, FCDN resizes all inputs to square dimensions, producing correspondingly shaped outputs. This discrepancy impacts TMQI and FSIMc scores, so quantitative results including FCDM are provided in Appendix D.

TMQI combines multiscale structural similarity with a naturalness measure, where the latter is derived from the intensity statistics of 3000 natural scene images. As shown in Table 7, the proposed system demonstrates strong performance on real-world outdoor and indoor images, ranking first and second on the O-HAZE and I-HAZE datasets, respectively. On the Dense-Haze dataset, the system ranks third, following MB-TaylorFormer and DCP. MB-TaylorFormer, rather than removing haze directly, reconstructs a haze-free image using features extracted from the input hazy image. This approach is particularly effective for dense haze scenarios, although its high computational cost is a notable drawback. DCP, well-known for its over-dehazing tendencies, performs favorably in dense haze conditions but is less suitable in general contexts due to its tendency to produce unnatural results. In contrast, the proposed system, with its self-calibrating capability, achieves robust performance across varying haze conditions. However, its lower performance on Dense-Haze stems from the inherent limitations of its dehazing process, which is based on the ICAP method. Overall, the proposed system is ranked second in terms of TMQI.

FSIMc, the second metric, extends structural similarity to assess color images. Table 7 reveals an interesting trend: non-learning-based methods exhibit higher FSIMc values compared to learning-based methods. This observation can be attributed to the following: learning-based methods such as YOLY and MB-TaylorFormer are trained to generate haze-free images. Due to the high computational and memory demands during training, these methods are typically trained on resized images rather than the original full-resolution images, which likely affects their performance under FSIMc evaluation. DehazeNet, another learning-based method, performs better than YOLY and MB-TaylorFormer because it estimates the transmission map rather than generating a complete haze-free image. The proposed system performs well across most datasets, with the exception of Dense-Haze, for reasons consistent with those identified in the TMQI evaluation. Nevertheless, it achieves the highest overall ranking in terms of FSIMc.

4.3. FPGA Implementation Result

The proposed autonomous dehazing system was implemented using Verilog HDL (IEEE Standard 1364-2005) [57] at the register transfer level (RTL), leveraging the portability and reusability offered by this design methodology. For example, the modified hybrid median filter was directly reused from our previous work in [47]. Moreover, as the RTL design primarily models signal flow, it is particularly well-suited for FPGA implementations, enabling efficient deployment using the simplified data paths outlined in Section 3.

Table 8 presents the resource utilization of the proposed system on the XCZU7EV-2FFVC1156 MPSoC device, showing an occupation of 10.80%, 23.21%, and 18.11% for slice registers, look-up tables (LUTs), and block RAMs, respectively. This level of utilization is reasonable, given that dehazing algorithms typically serve as preprocessing steps in high-level computer vision systems. The proposed system achieves a minimum clock period of 2.49 ns, corresponding to a maximum frequency of 401.45 MHz, which translates to a throughput of 401.45 Mpixels/s.

Given the maximum frequency $f_{\max} = 401.45$ MHz, the processing speed in terms of frames per second (fps) can be computed as:

$$\text{FPS} = \frac{f_{\max}}{(H + B_V)(W + B_H)}, \quad (27)$$

where $\{H, W\}$ represent the image height and width and $\{B_V, B_H\}$ denote the vertical and horizontal blanking intervals, measured in pixels. For modern digital camera systems, $B_V = 1$ and $B_H = 1$ can be assumed. Considering a DCI 4K resolution ($H = 2160$ and $W = 4096$), it requires $(2160 + 1)(4096 + 1) = 8,853,617$ clock cycles to process a single frame. Thus, the system achieves a processing speed of approximately $401.45 \times 10^6 / 8,853,617 \approx 45.34$ fps, rendering it highly suitable for real-time computer vision applications. Table 9 summarizes the processing speed of the system across various video standards, ranging from Full HD to DCI 4K.

Table 8. Hardware implementation result of the proposed autonomous dehazing system.

Xilinx Vivado v2023.1			
Device	XCZU7EV-2FFVC1156-2-E		
Slice Logic Utilization	Available	Used	Utilization
Slice registers (#)	460,800	49,757	10.80%
Slice LUTs (#)	230,400	53,484	23.21%
Block RAMs (#)	312	56.5	18.11%
Minimum period (ns)		2.49	
Maximum frequency (MHz)		401.45	

Table 9. Maximum processing speeds in frames per second for different video standards.

Standard	Resolution	Required Clock Cycles (#)	Processing Speed (fps)	
Full HD	1920 × 1080	2,076,601	193.32	
Quad HD	2560 × 1440	3,690,401	108.78	
4K	UW4K	3840 × 1600	6,149,441	65.28
	UHD TV	3840 × 2160	8,300,401	48.36
	DCI 4K	4096 × 2160	8,853,617	45.34

For 8K UHD resolution (7680×4320), the proposed system achieves only 12.10 fps, falling short of real-time processing requirements. However, as shown in Table 8, the target FPGA can accommodate four instances of the proposed system. Figure 12 illustrates two strategies for processing 8K UHD video. On the left, two instances operate on 7680×2160 frame segments, achieving a maximum processing speed of

24.19 fps, approaching real-time performance. On the right, four instances process 3840×2160 segments, increasing the speed to 48.37 fps, thereby meeting both NTSC and PAL real-time requirements.

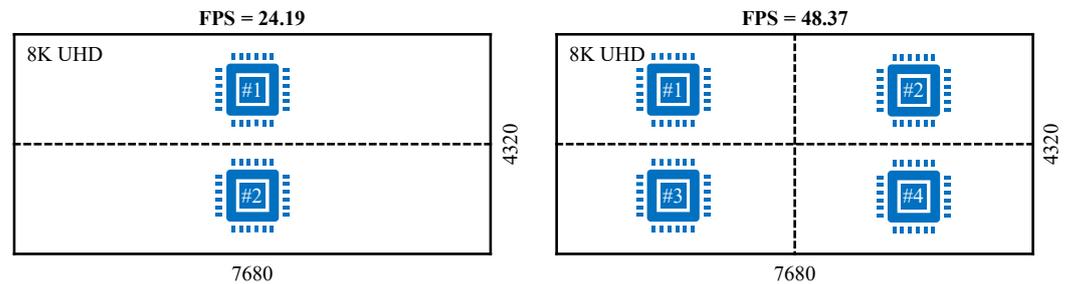


Figure 12. Strategies for processing 8K UHD video.

The proposed system demonstrates a significant advantage in processing speed compared to all real-time dehazing implementations reported in Table 3. In Table 10, it is compared against three specific implementations: the DCP with fast airlight estimation (DCP-FAE) [40], the direct implementation of DCP (DCP-DI) [38], and an earlier autonomous dehazing system from our prior work [12]. The proposed system outperforms DCP-FAE across all evaluation metrics, requiring fewer hardware resources while delivering significantly faster processing speeds. A similar trend is observed when compared to DCP-DI, despite the unavailability of complete implementation details for the latter. Furthermore, as detailed in Section 4.1, DCP-based methods lack the autonomous dehazing capability, a key feature of the proposed system.

Table 10. Comparison with other real-time dehazing implementations. NA stands for not available.

Hardware Utilization	DCP-FAE [40]	DCP-DI [38]	Previous Work [12]	Proposed System
Slice registers (#)	53,400	NA	53,216	49,757
Slice LUTs (#)	64,000	NA	49,799	53,484
DSPs (#)	42	NA	0	0
Memory (Mbits)	3.2	NA	1.4	2.1
Maximum frequency (MHz)	88.70	116.00	271.37	401.45
Maximum video resolution	SVGA	Quad HD	DCI 4K	DCI 4K
Autonomous dehazing	Unequipped	Unequipped	Equipped	Equipped

Finally, compared to our previous work, the proposed system exhibits minimal differences in logic utilization but requires additional memory resources due to the implementation of local image blending. However, as this feature operates during the vertical blanking interval, it does not affect the overall processing speed. Notably, the proposed system achieves a $1.48\times$ improvement in processing speed compared to our prior implementation.

4.4. Aerial Object Detection Result

Aerial images in Figure 11 are provided to YOLOv9 [58] for object detection, with results shown in Figure 13 and summarized in Table 11. Under haze-free conditions, all six benchmark methods apply dehazing indiscriminately, unintentionally degrading image quality and detection performance. As discussed in Section 4.2, YOLY and MB-TaylorFormer generate haze-free images rather than performing explicit dehazing. While effective for moderate and dense haze, this approach leads to performance degradation in haze-free and mild haze scenarios, as evidenced by a sharp decrease in the number of detected aircraft. FCDM, a diffusion-based dehazing model, performs slightly better than MB-TaylorFormer and YOLY, as it avoids false detections. Meanwhile, DCP, CAP, and

DehazeNet cause some reduction in detection performance, though less severe than that of YOLY and MB-TaylorFormer. In contrast, the proposed system correctly identifies haze-free inputs and bypasses dehazing, preserving detection performance.

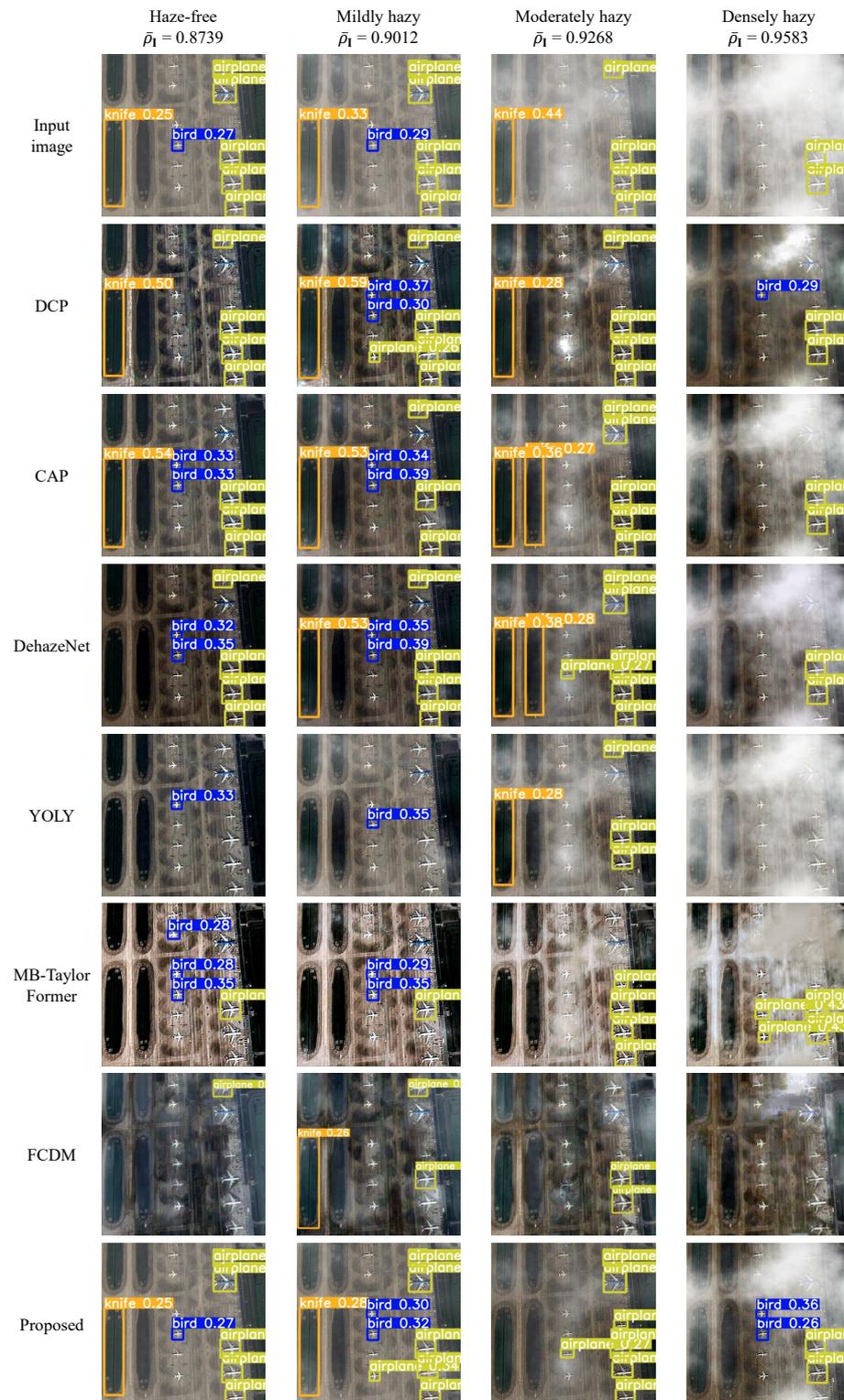


Figure 13. Aerial object detection results of YOLOv9 on an aerial image under haze-free, mildly hazy, moderately hazy, and densely hazy conditions. The average haze density $\bar{\rho}_1$ was compared against two thresholds, $\rho_1 = 0.8811$ and $\rho_2 = 0.9344$, to determine the haze condition. Yellow, blue, and orange labels represent aircrafts, birds, and knives, respectively.

Table 11. Summary of YOLOv9 detection results on an aerial image under haze-free, mildly hazy, moderately hazy, and densely hazy conditions.

Method	Case	Haze-Free		Thin		Moderate		Dense	
		Aircraft (#)	Failure (#)						
Input		5	2	5	2	4	1	2	0
DCP		4	1	5	3	4	1	2	1
CAP		3	3	3	3	5	2	2	0
DehazeNet		4	2	4	3	6	2	2	0
YOLY		0	1	0	1	3	1	0	0
MB-TaylorFormer		1	3	1	2	4	0	4	0
FCDM		1	0	2	1	2	0	0	0
Proposed		5	2	6	3	7	0	2	2

Under mild haze, a similar trend is observed. Most methods reduce detection performance, whereas DCP maintains the number of detected aircraft, and the proposed system results in one additional aircraft detected. In moderate haze, YOLY and MB-TaylorFormer improve performance as expected. While YOLY reduces the number of detected aircraft by one, MB-TaylorFormer preserves YOLOv9's original detection performance. FCDM does not benefit YOLOv9 and leads to the disappearance of two detected aircraft. In contrast, CAP and DehazeNet enhance detection rates by 25% and 50%, respectively. The proposed system achieves a 75% improvement, while also eliminating false detections.

Under dense haze, the proposed system, though not the best-performing method, still preserves detection performance by maintaining two detected aircraft. In this scenario, MB-TaylorFormer achieves the highest improvement at 100%. Interestingly, despite its strong qualitative and quantitative performance, FCDM results in zero detections. Across all haze conditions, YOLOv9 does not benefit from images preprocessed by this method. Overall, the proposed system outperforms other approaches in preprocessing input images for YOLOv9-based object detection.

5. Conclusions

This paper presents a real-time FPGA implementation of an autonomous dehazing method. A haziness degree evaluator quantifies the haze density of the input image, classifying it into four categories: haze-free, mildly hazy, moderately hazy, or densely hazy. Based on this classification, a self-calibrating weight is derived, enabling the proposed system to adaptively adjust its dehazing strength for different haze conditions. Additionally, local haze density values are used to compute local blending weights, enhancing the local textures of the final result. Unlike previous approaches, which often omit this feature due to implementation challenges during the video active interval, this work introduces a method to perform local blending during the video blanking interval. This ensures high processing speed without impacting system throughput.

Qualitative and quantitative evaluations demonstrate the effectiveness of the proposed system in dehazing, while FPGA implementation results validate its efficient hardware utilization and high throughput. The system achieves a processing rate of 45.34 frames per second for DCI 4K video. Furthermore, experiments on aerial images under various haze conditions highlight its suitability for integration into high-level computer vision applications, such as YOLOv9 for object detection.

Despite its advantages, the proposed system inherits a limitation from the use of ICAP for dehazing—reduced performance in dense or heterogeneous haze conditions. This issue stems not only from the simplicity of the linear equation used for scene depth estimation, but also from the constraints of the atmospheric scattering model. Future

research may explore more flexible or advanced physical models to further enhancing dehazing performance.

Author Contributions: Conceptualization, B.K.; methodology, D.N. and B.K.; software, D.N. and J.S.; data curation, D.N. and J.S.; writing—original draft preparation, D.N.; writing—review and editing, D.N., J.S., and B.K.; visualization, D.N. and J.S.; supervision, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by research funds from Dong-A University, Busan, Korea.

Data Availability Statement: Data are available in a publicly accessible repository. The data presented in this study are openly available in [6,53–56].

Acknowledgments: The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Two-dimensional filters are implemented using line memories, registers, and a dedicated logic circuit, customized to the specific filter type, as illustrated in Figure A1. Line memories are mapped to block RAMs on the FPGA, with each memory delaying the input stream by one line, where the line length corresponds to the width of the image. In other words, line memories provide vertical delay to the input stream. Conversely, registers are used to delay the input stream horizontally, with each register introducing a one-pixel delay. For example, the 3×3 filter shown in Figure A1 comprises two line memories and six registers, enabling access to nine pixels within the filter kernel: z_1, z_2, \dots, z_9 .

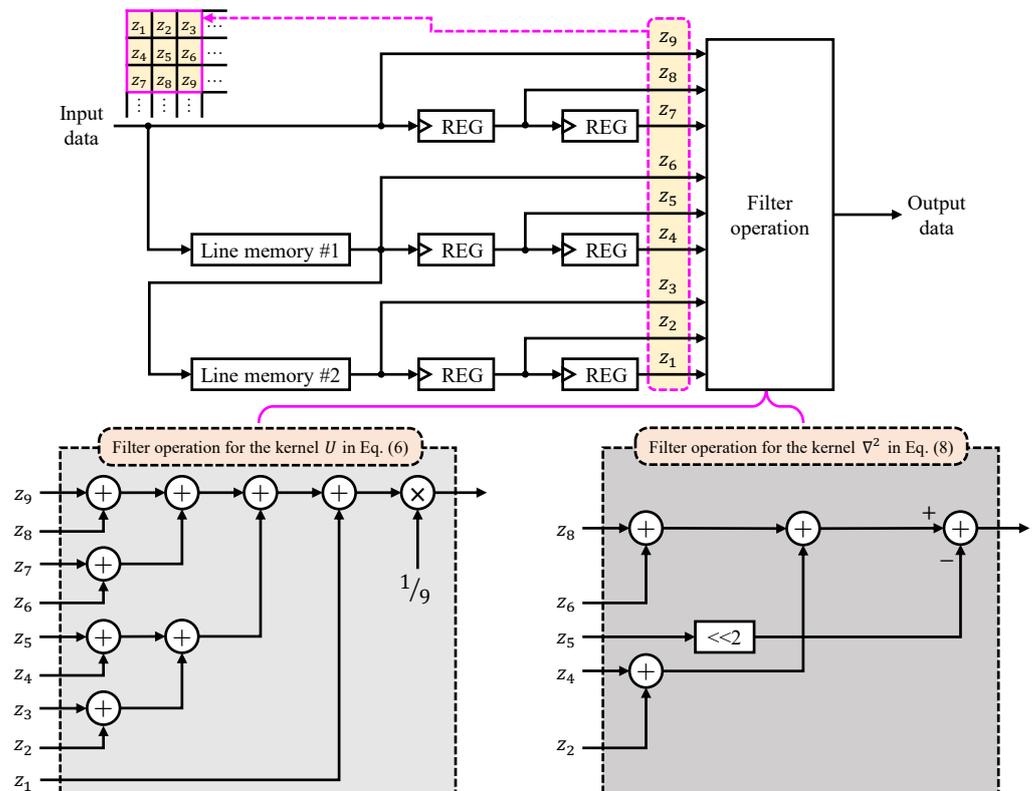


Figure A1. Simplified data path of 3×3 two-dimensional filters. REG is an abbreviation for register.

These pixels are subsequently processed by a dedicated logic circuit, designed according to the filter type. For example, Figure A1 illustrates two specific implementations: the moving average filter (kernel U in Equation (6)) and the Laplacian filter (kernel ∇^2 in Equation (8)). For the modified hybrid median filter described in Section 3.2, the filtering

operation module is replaced with an optimized merging sorting network, as detailed in [47].

Appendix B

The filter architecture presented in Appendix A operates during the video active interval, whereas the 2×2 low-pass filter for smoothing local haze density values functions during the video blanking interval. This requires a separate architecture, depicted in Figure A2. The 8×8 local haze density map ρ_Ω , computed using Equation (25), is stored in a dual-port RAM. This RAM features two address ports and two data ports, enabling efficient implementation of the 2×2 sliding window.

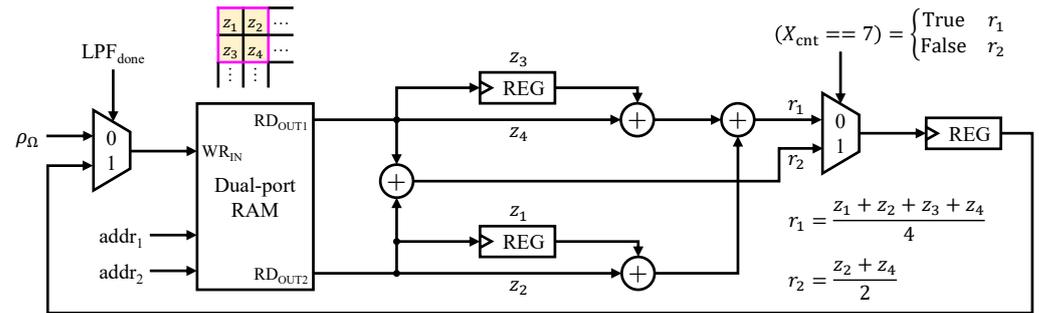


Figure A2. Simplified data path of the 2×2 low-pass filter.

The first address port, $addr_1$, is driven by a counter incrementing from 0 to 63, facilitating the retrieval of z_1 and z_3 within the sliding window. The second address port, $addr_2$, is controlled by a counter that increases from 8 to 63, followed by wrapping back to 56 to 63, ensuring the retrieval of z_2 and z_4 . Boundary padding is not applied in this implementation. Instead, boundary values are averaged directly to generate the filtering results, as illustrated by a multiplexer controlled by the comparison $X_{cnt} == 7$, where X_{cnt} represents a horizontal counter.

The data written to the dual-port RAM are selected through a multiplexer controlled by the signal LPF_{done} . When the computation of the filtering result is complete ($LPF_{done} = 1$), the multiplier routes the filtering result to the write port of the dual-port RAM. Conversely, when $LPF_{done} = 0$, the multiplier routes the local haze density values to the write port, allowing the dual-port RAM to store new 8×8 local haze density values.

Appendix C

Figure A3 illustrates two simplified data paths for implementing $4 \times$ bilinear interpolation. The first data path, shown in Figure A3a and denoted as $INTERP_1$, is derived directly from the interpolation formulas. Local haze density values processed by the low-pass filter are stored in a dual-port RAM, from which they are read to perform the interpolation. The interpolated results are subsequently stored in a single-port RAM, to be read later for computing the self-calibrating weight and local blending weights. However, this data path requires a substantial number of adders and multiplexers, increasing implementation complexity.

The second data path, depicted in Figure A3b and referred to as $INTERP_2$, is designed to exploit the overlap when sliding the window. Let X_{cnt} and Y_{cnt} represent the horizontal and vertical counters, respectively, both ranging from 0 to 7. Three cases are considered when interpolating the 8×8 local haze density map:

- Case 1: $(X_{cnt} < 7)$ AND $(Y_{cnt} < 7)$
For this case, instead of computing all values within the 5×5 interpolated patch, it suffices to compute only the upper-left 4×4 patch. The last column of the 5×5 patch

is computed when the 2×2 window slides one position to the right. Similarly, the last row is computed when the 2×2 window slides one position downward.

- Case 2: $(X_{cnt} == 7)$ AND $(X_{cnt} == 7)$
When the window is at the last position, that is, the 63rd position in the 8×8 local haze density map, no computation is required as the interpolated result is identical to the input value.
- Case 3: otherwise
For this case, when the window reaches the rightmost position, it is sufficient to compute the first four values of the last column of the 5×5 interpolated patch. Similarly, when the window is at the bottommost position, only the first four values of the last row of the 5×5 interpolated patch need to be computed.

In the proposed $INTERP_2$ data path, four multiplexers are employed to select the appropriate weight (1, 2, 3, 4, 6, 8, 9, 12, or 16) for implementing the interpolation formulas. This design significantly reduces the number of required adders compared to the $INTERP_1$ data path shown in Figure A3a, leading to a more efficient hardware implementation.

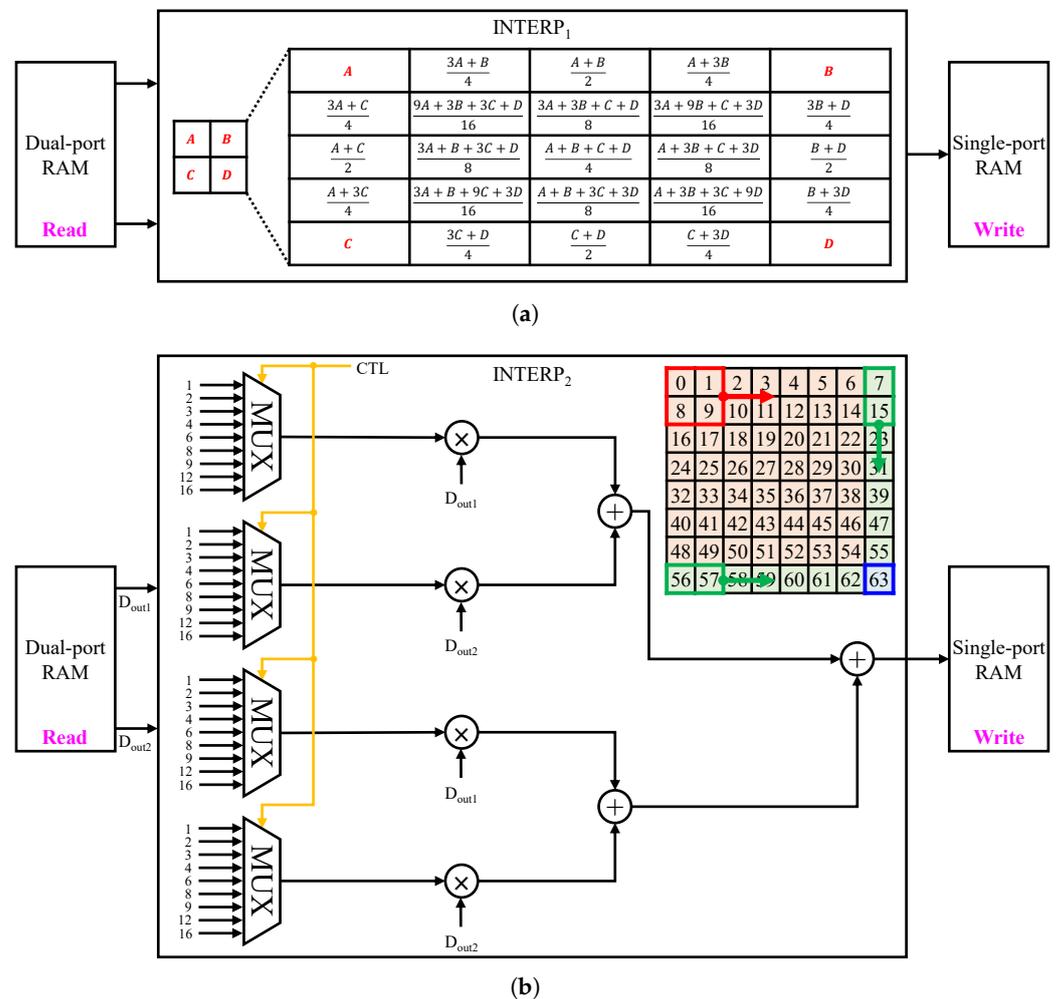


Figure A3. Simplified data paths of $4 \times$ bilinear interpolation. (a) Formula-based data path. (b) Proposed data path leveraging the overlap when sliding the window.

Appendix D

Table A1 presents the average TMQI and FSIMc scores, including FCDM [33]. It should be noted that FCDM resizes all input images to square dimensions and produces correspondingly shaped outputs, unlike the proposed system and other benchmark methods, which can process variable-sized images. To compute TMQI and FSIMc scores for FCDM,

ground-truth images are resized to match its output dimensions. This resizing affects the final scores, making direct comparisons with other methods challenging. Therefore, these results are provided for reference only.

Table A1. Average TMQI and FSIMc values computed on five public datasets. The best and second-best results are boldfaced and italicized, respectively. MB-TF is the abbreviation for MB-TaylorFormer.

Dataset	Method	DCP	CAP	DehazeNet	YOLY	MB-TF	FCDM	Proposed
TMQI↑	FRIDA2	0.7291	0.7385	0.7366	0.7176	0.7631	0.7907	0.7345
	D-HAZY	0.8631	0.8206	0.7966	0.6817	0.7428	<i>0.8447</i>	0.7861
	O-HAZE	0.8403	0.8118	0.8413	0.6566	0.8732	0.9154	<i>0.9048</i>
	I-HAZE	0.7319	0.7512	0.7598	0.6936	0.8655	0.8191	<i>0.8319</i>
	Dense-Haze	0.6383	0.5955	0.5723	0.5107	0.7237	<i>0.6589</i>	0.6120
	Total	0.7357	0.7336	0.7312	0.6520	<i>0.7761</i>	0.7918	0.7466
FSIMc↑	FRIDA2	0.7746	0.7918	0.7963	0.7849	0.7158	0.8230	<i>0.8027</i>
	D-HAZY	0.9002	0.8880	0.8874	0.7383	0.7727	<i>0.8883</i>	0.8772
	O-HAZE	<i>0.8423</i>	0.7738	0.7865	0.6997	0.8420	0.8865	0.8319
	I-HAZE	0.8208	0.8252	0.8482	0.7564	0.8692	0.8974	<i>0.8726</i>
	Dense-Haze	0.6419	0.5773	0.5573	0.5763	0.7976	<i>0.7056</i>	0.5869
	Total	0.7746	0.7693	0.7725	0.7111	0.7544	0.8233	<i>0.7865</i>

References

- Lee, Z.; Shang, S. Visibility: How Applicable is the Century-Old Koschmieder Model? *J. Atmos. Sci.* **2016**, *73*, 4573–4581. [\[CrossRef\]](#)
- Ngo, D.; Lee, S.; Ngo, M.T.; Lee, G.D.; Kang, B. Visibility Restoration: A Systematic Review and Meta-Analysis. *Sensors* **2021**, *21*, 2625. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sahu, G.; Seal, A.; Bhattacharjee, D.; Nasipuri, M.; Brida, P.; Krejcar, O. Trends and Prospects of Techniques for Haze Removal From Degraded Images: A Survey. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *6*, 762–782. [\[CrossRef\]](#)
- Saad Saoud, L.; Elmezain, M.; Sultan, A.; Heshmat, M.; Seneviratne, L.; Hussain, I. Seeing Through the Haze: A Comprehensive Review of Underwater Image Enhancement Techniques. *IEEE Access* **2024**, *12*, 145206–145233. [\[CrossRef\]](#)
- He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353. [\[CrossRef\]](#)
- Tarel, J.P.; Hautiere, N.; Caraffa, L.; Cord, A.; Halmaoui, H.; Gruyer, D. Vision Enhancement in Homogeneous and Heterogeneous Fog. *IEEE Intell. Transp. Syst. Mag.* **2012**, *4*, 6–20. [\[CrossRef\]](#)
- Zhu, Q.; Mai, J.; Shao, L. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Trans. Image Process.* **2015**, *24*, 3522–3533. [\[CrossRef\]](#)
- Galdran, A. Image dehazing by artificial multiple-exposure image fusion. *Signal Process.* **2018**, *149*, 135–147. [\[CrossRef\]](#)
- Berman, D.; Treibitz, T.; Avidan, S. Single Image Dehazing Using Haze-Lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 720–734. [\[CrossRef\]](#)
- Ngo, D.; Lee, G.D.; Kang, B. Improved Color Attenuation Prior for Single-Image Haze Removal. *Appl. Sci.* **2019**, *9*, 4011. [\[CrossRef\]](#)
- Tang, Q.; Yang, J.; He, X.; Jia, W.; Zhang, Q.; Liu, H. Nighttime image dehazing based on Retinex and dark channel prior using Taylor series expansion. *Comput. Vis. Image Underst.* **2021**, *202*, 103086. [\[CrossRef\]](#)
- Lee, S.; Ngo, D.; Kang, B. Design of an FPGA-Based High-Quality Real-Time Autonomous Dehazing System. *Remote Sens.* **2022**, *14*, 1852. [\[CrossRef\]](#)
- Ngo, D.; Lee, G.D.; Kang, B. Single Image Dehazing With Unsharp Masking and Color Gamut Expansion. *IEEE Access* **2022**, *10*, 102462–102474. [\[CrossRef\]](#)
- Dwivedi, P.; Chakraborty, S. Single image dehazing using extended local dark channel prior. *Image Vis. Comput.* **2023**, *136*, 104747. [\[CrossRef\]](#)
- Ling, P.; Chen, H.; Tan, X.; Jin, Y.; Chen, E. Atmospheric Light Estimation Based Remote Sensing Image Dehazing. *IEEE Trans. Image Process.* **2023**, *32*, 3238–3253. [\[CrossRef\]](#)
- Liang, S.; Gao, T.; Chen, T.; Cheng, P. A Remote Sensing Image Dehazing Method Based on Heterogeneous Priors. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5619513.

17. Ngo, D.; Kang, B. A Symmetric Multiprocessor System-on-a-Chip-Based Solution for Real-Time Image Dehazing. *Symmetry* **2024**, *16*, 653. [[CrossRef](#)]
18. Cai, B.; Xu, X.; Jia, K.; Qing, C.; Tao, D. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Trans. Image Process.* **2016**, *25*, 5187–5198. [[CrossRef](#)]
19. Ren, W.; Pan, J.; Zhang, H.; Cao, X.; Yang, M.H. Single Image Dehazing via Multi-scale Convolutional Neural Networks with Holistic Edges. *Int. J. Comput. Vis.* **2020**, *128*, 240–259. [[CrossRef](#)]
20. Dong, H.; Pan, J.; Xiang, L.; Hu, Z.; Zhang, X.; Wang, F.; Yang, M.H. Multi-Scale Boosted Dehazing Network with Dense Feature Fusion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2154–2164. [[CrossRef](#)]
21. Li, B.; Gou, Y.; Gu, S.; Liu, J.Z.; Zhou, J.T.; Peng, X. You Only Look Yourself: Unsupervised and Untrained Single Image Dehazing Neural Network. *Int. J. Comput. Vis.* **2021**, *129*, 1754–1767. [[CrossRef](#)]
22. Yang, Y.; Wang, C.; Liu, R.; Zhang, L.; Guo, X.; Tao, D. Self-augmented Unpaired Image Dehazing via Density and Depth Decomposition. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 2027–2036. [[CrossRef](#)]
23. Song, Y.; He, Z.; Qian, H.; Du, X. Vision Transformers for Single Image Dehazing. *IEEE Trans. Image Process.* **2023**, *32*, 1927–1941. [[CrossRef](#)] [[PubMed](#)]
24. Zheng, Y.; Zhan, J.; He, S.; Dong, J.; Du, Y. Curricular Contrastive Regularization for Physics-Aware Single Image Dehazing. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 5785–5794. [[CrossRef](#)]
25. Wu, R.Q.; Duan, Z.P.; Guo, C.L.; Chai, Z.; Li, C. RIDCP: Revitalizing Real Image Dehazing via High-Quality Codebook Priors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 22282–22291. [[CrossRef](#)]
26. Huang, Y.; Lin, Z.; Xiong, S.; Sun, T. Diffusion Models Based Null-Space Learning for Remote Sensing Image Dehazing. *IEEE Geosci. Remote. Sens. Lett.* **2024**, *21*, 8001305. [[CrossRef](#)]
27. Ding, H.; Xie, F.; Qiu, L.; Zhang, X.; Shi, Z. Robust Haze and Thin Cloud Removal via Conditional Variational Autoencoders. *IEEE Trans. Geosci. Remote. Sens.* **2024**, *62*, 5604616. [[CrossRef](#)]
28. Bui, T.M.; Kim, W. Single Image Dehazing Using Color Ellipsoid Prior. *IEEE Trans. Image Process.* **2018**, *27*, 999–1009. [[CrossRef](#)]
29. Ancuti, C.; Ancuti, C.O.; De Vleeschouwer, C.; Bovik, A.C. Day and Night-Time Dehazing by Local Airlight Estimation. *IEEE Trans. Image Process.* **2020**, *29*, 6264–6275. [[CrossRef](#)]
30. Li, B.; Gou, Y.; Liu, J.Z.; Zhu, H.; Zhou, J.T.; Peng, X. Zero-Shot Image Dehazing. *IEEE Trans. Image Process.* **2020**, *29*, 8457–8466. [[CrossRef](#)]
31. Yang, Y.; Guo, C.L.; Guo, X. Depth-Aware Unpaired Video Dehazing. *IEEE Trans. Image Process.* **2024**, *33*, 2388–2403. [[CrossRef](#)]
32. Qiu, Y.; Zhang, K.; Wang, C.; Luo, W.; Li, H.; Jin, Z. MB-TaylorFormer: Multi-branch Efficient Transformer Expanded by Taylor Formula for Image Dehazing. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 12756–12767. [[CrossRef](#)]
33. Wang, J.; Wu, S.; Yuan, Z.; Tong, Q.; Xu, K. Frequency compensated diffusion model for real-scene dehazing. *Neural Netw.* **2024**, *175*, 106281. [[CrossRef](#)]
34. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015. [[CrossRef](#)]
35. Kumar, R.; Kaushik, B.K.; Raman, B.; Sharma, G. A Hybrid Dehazing Method and its Hardware Implementation for Image Sensors. *IEEE Sens. J.* **2021**, *21*, 25931–25940. [[CrossRef](#)]
36. Shiau, Y.H.; Yang, H.Y.; Chen, P.Y.; Chuang, Y.Z. Hardware Implementation of a Fast and Efficient Haze Removal Method. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1369–1374. [[CrossRef](#)]
37. Bai, L.; Wu, Y.; Xie, J.; Wen, P. Real Time Image Haze Removal on Multi-core DSP. *Procedia Eng.* **2015**, *99*, 244–252. [[CrossRef](#)]
38. Zhang, B.; Zhao, J. Hardware Implementation for Real-Time Haze Removal. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 1188–1192. [[CrossRef](#)]
39. Diaz-Ramirez, V.H.; Hernández-Beltrán, J.E.; Juárez-Salazar, R. Real-time haze removal in monocular images using locally adaptive processing. *J. Real-Time Image Process.* **2019**, *16*, 1959–1973. [[CrossRef](#)]
40. Park, Y.; Kim, T.H. Fast Execution Schemes for Dark-Channel-Prior-Based Outdoor Video Dehazing. *IEEE Access* **2018**, *6*, 10003–10014. [[CrossRef](#)]
41. Shiau, Y.H.; Kuo, Y.T.; Chen, P.Y.; Hsu, F.Y. VLSI Design of an Efficient Flicker-Free Video Defogging Method for Real-Time Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 238–251. [[CrossRef](#)]
42. Upadhyay, B.B.; Sarawadekar, K. VLSI Design of Saturation-Based Image Dehazing Algorithm. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2023**, *31*, 959–968. [[CrossRef](#)]

43. Lv, T.; Du, G.; Li, Z.; Wang, X.; Teng, P.; Ni, W.; Ouyang, Y. A fast hardware accelerator for nighttime fog removal based on image fusion. *Integration* **2024**, *99*, 102256. [[CrossRef](#)]
44. Bilal, M.; Masud, S.; Hanif, M.S. Efficient Framework for Real-Time Color Cast Correction and Dehazing Using Online Algorithms to Approximate Image Statistics. *IEEE Access* **2024**, *12*, 72813–72827. [[CrossRef](#)]
45. Ngo, D.; Lee, G.D.; Kang, B. Haziness Degree Evaluator: A Knowledge-Driven Approach for Haze Density Estimation. *Sensors* **2021**, *21*, 3896. [[CrossRef](#)]
46. Arun, R.; Anoop, P.; John, S.; Chen, J.; Deepa, V.; Varma, R. Atmospheric Aerosol Scattering Coefficients in a Broad Visible Spectral Region. *Aerosol Sci. Eng.* **2021**, *5*, 373–382. [[CrossRef](#)]
47. Ngo, D.; Lee, S.; Lee, G.D.; Kang, B. Single-Image Visibility Restoration: A Machine Learning Approach and Its 4K-Capable Hardware Accelerator. *Sensors* **2020**, *20*, 5795. [[CrossRef](#)] [[PubMed](#)]
48. Fairchild, M. *Color Appearance Models*, 3rd ed.; Wiley: West Sussex, UK, 2013.
49. Xilinx. ZCU104 Evaluation Board: User Guide (UG1267). Available online: <https://docs.amd.com/v/u/en-US/ug1267-zcu104-eval-bd> (accessed on 12 July 2023).
50. Xilinx. Vivado Design Suite User Guide: Release Notes, Installation, and Licensing (UG973). Available online: <https://docs.amd.com/r/en-US/ug973-vivado-release-notes-install-license/Release-Notes> (accessed on 15 April 2023).
51. Yeganeh, H.; Wang, Z. Objective Quality Assessment of Tone-Mapped Images. *IEEE Trans. Image Process.* **2013**, *22*, 657–667. [[CrossRef](#)]
52. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [[CrossRef](#)] [[PubMed](#)]
53. Ancuti, C.; Ancuti, C.O.; De Vleeschouwer, C. D-HAZY: A dataset to evaluate quantitatively dehazing algorithms. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2226–2230. [[CrossRef](#)]
54. Ancuti, C.O.; Ancuti, C.; Timofte, R.; De Vleeschouwer, C. O-HAZE: A Dehazing Benchmark with Real Hazy and Haze-Free Outdoor Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 867–8678. [[CrossRef](#)]
55. Ancuti, C.; Ancuti, C.O.; Timofte, R.; Van Gool, L.; Zhang, L.; Yang, M.H.; Patel, V.M.; Zhang, H.; Sindagi, V.A.; Zhao, R.; et al. NTIRE 2018 Challenge on Image Dehazing: Methods and Results. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018.
56. Ancuti, C.O.; Ancuti, C.; Timofte, R.; Van Gool, L.; Zhang, L.; Yang, M.H.; Guo, T.; Li, X.; Cherukuri, V.; Monga, V.; et al. NTIRE 2019 Image Dehazing Challenge Report. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 2241–2253. [[CrossRef](#)]
57. *IEEE Std 1364-2005*; IEEE Standard for Verilog Hardware Description Language; (Revision of IEEE Std 1374-2001); IEEE: Piscataway, NJ, USA, 2006; pp. 1–590. [[CrossRef](#)]
58. Wang, C.Y.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.