

VLSI Design of Saturation-Based Image Dehazing Algorithm

Bharat Bhushan Upadhyay[✉], *Graduate Student Member, IEEE*, and Kishor Sarawadekar[✉], *Senior Member, IEEE*

Abstract—Hazy weather degrades the vividness of the images captured by real-time systems for applications such as object detection, remote sensing, and surveillance systems. This affects the performance of such real-time systems. The hardware implementation of a real-time haze removal system is imperative to solve these problems. Such a solution is proposed in this article. Here, the saturation-based hardware implementation of an image dehazing system is presented. To estimate atmospheric light more precisely, a 15×15 window minimum filter is implemented, which uses the downsampled hazy image to estimate the atmospheric light. Furthermore, we have employed saturation-based transmission map estimation, which makes our approach pixel-based rather than patch-based. Unlike existing patch-based methods, the proposed method requires neither an edge detection unit nor an image filtering unit to suppress halo artifacts around edges. The VLSI architecture of the proposed dehazing system comprises seven pipelined stages. It is implemented on FPGA as well as ASIC (65-nm technology node) platforms. The ASIC implementation of the proposed dehazing system yielded a maximum throughput of 624 Mpixels/s, which is fast enough to process 3840×2160 resolution at a rate higher than 70 fps with only 13.2k logic gates count.

Index Terms—Image dehazing, real-time application, saturation-based image dehazing, VLSI architecture.

I. INTRODUCTION

A. Necessity of Haze Removal Systems

HAZE adversely affects the performance of real-time computer vision systems by restricting the visibility of the objects in the captured image. This degradation in visibility may lead to the undesirable failure of such systems. A real-time dehazing system can improve a computer vision system's overall performance and make it more resilient even in hazy weather. However, the additional real-time dehazing system should not affect the performance of other subsystems and should be cost-effective. The VLSI implementation of such real-time dehazing systems is extremely challenging as it requires performing sophisticated mathematical operations and numerous iterations, which consumes extensive hardware resources.

B. Literature Review

Various haze removal and vision enhancement algorithms have been proposed earlier to mitigate this situation.

Manuscript received 23 November 2022; revised 11 February 2023 and 28 March 2023; accepted 14 April 2023. Date of publication 17 May 2023; date of current version 28 June 2023. (Corresponding author: Kishor Sarawadekar.)

The authors are with the Department of Electronics Engineering, Indian Institute of Technology (BHU), Varanasi 221005, India (e-mail: skishor.ece@iitbhu.ac.in).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TVLSI.2023.3272018>.

Digital Object Identifier 10.1109/TVLSI.2023.3272018

Recently, prior-based and machine learning-based dehazing algorithms have gained popularity and are mainly reviewed below.

- 1) *Prior-Based Dehazing Methods*: Various prior-based dehazing methods exist in literature. Tan [1] proposed a dehazing method that was independent of image geometry. His method was based on two main observations, viz. contrast of the dehazed image is more than that of the hazy image and variations in atmospheric light are smooth. A cost function based on the above two observations was formulated and optimized. However, this method suffers from halos due to patch-based dehazing. Fattal [2] presented a dehazing method based on the assumption that surface shading of the objects in an image and their transmission are uncorrelated in a local region. Since his method uses the extrapolation technique, it may fail when the input data lacks a sufficient signal-to-noise ratio. A novel, prior-based image dehazing method was proposed in [3], which exploits the statistics of haze-free images. This method is known as dark channel prior (DCP) and is based on an interesting observation that within a local region of natural outdoor images, some of the pixels in the region have very low intensity at least in one of the RGB channels. Although the DCP method is simple and effective, it produces halos around edges and requires some complex post-processing techniques for haze-free image restoration. Another prior-based image dehazing method was introduced in [4], which used a linear model to estimate scene depth based on the pixel's brightness and saturation. However, this method estimates transmission inaccurately wherever atmospheric conditions are nonhomogeneous. Other prior-based dehazing algorithms also exist in [5], [6], [7], and [8]. But DCP being a simple and effective method, many researchers proposed modifications in DCP to enhance its performance [9], [10]. Several filtering techniques have also been proposed to refine the transmission map obtained by DCP. Joint bilateral filtering [11], guided image filtering [12], and median filtering [13] are some of the popular refinement methods proposed by researchers for image dehazing using DCP.

- 2) *Nonprior-Based Dehazing Methods*: An image dehazing method using some linear transformation is proposed in [14]. It is less computationally intensive and recovers the image efficiently, even at sudden edges. A fusion-based technique in which several underexposed hazy images are merged using multiscale Laplacian mixing

to obtain the haze-free image is proposed in [15]. A saturation-based image dehazing method is reported in [16], which derives transmission for individual pixels using saturation of scene radiance. This technique is fast and efficient as it does not require any refinement process. A saturation- and intensity-based image dehazing method is reported in [17], which uses a color line model to maintain smoothness and preserve edges.

- 3) *Machine Learning-Based Methods*: Several machine learning-based image dehazing networks have flourished recently due to the advancement in computing platforms and the availability of a large number of image datasets to train and test the model. DehazeNet [18], an end-to-end convolutional neural network (CNN), is highly efficient in image dehazing as compared with conventional and prior-based methods. However, the results could be better if atmospheric light is dynamically estimated. AOD-Net [19] enhanced its dehazing performance by estimating both the transmission map and atmospheric light in a single unified model rather than separately estimating them. A multiscale CNN is proposed in [20], which uses coarse and fine networks to roughly estimate transmission and then to refine it. However, atmospheric light is also estimated separately in this method, which reduces its performance. PDR-Net [21] uses two separate networks: one for image dehazing and the other for image quality enhancement. This CNN architecture produces high-quality images by optimizing a multiterm loss function. RefineDNet [22] proposes a two-stage weakly supervised framework for image dehazing. In the first stage, the prior-based technique is employed to restore the haze-free image, and in the second stage, adversarial learning is applied to enhance the appearance of the dehazed image. A novel CNN-based image dehazing network (DeHamer) is proposed in [23], which integrates the features of CNN and Transformer to attain the state-of-the-art dehazing performance.

Most of the abovementioned techniques and algorithms require high-performance computing platforms or graphics processing unit (GPU) accelerators for their implementation. Thus, a real-time hardware implementation of aforesaid image dehazing methods is a tedious task.

C. Related Hardware Architectures

A real-time haze removal method based on DCP is proposed in [24], which performs gradient detection to avoid halo artifacts arising due to inaccurate transmission estimation for pixels in a patch lying on the edge. Since this method uses a smaller window size of 3×3 for calculating the dark channel, it may inaccurately estimate the atmospheric light when white objects are present in an image. A faster image dehazing architecture is proposed in [25], but it may also suffer from inaccurate atmospheric light estimation (ALE) due to a smaller window size. A video defogging method is proposed in [26], which also works for single-image dehazing. This method uses complex weights to calculate atmospheric light using the gray levels of the hazy image and suppresses the halo artifacts by employing an edge-preserving filter.

A similar dehazing architecture is proposed in [27] with a modified edge-preserving filter to eliminate halo artifacts in the recovered images. The atmospheric light is also dynamically adjusted to enhance the visual quality of the dehazed image. However, this technique also uses a smaller window size to calculate the dark channel, which makes it prone to inappropriate global atmospheric light calculation, which is further used to calculate local atmospheric light. An image dehazing engine with parallel processing cores for calculating airlight and transmission rate is presented in [28], which achieves better performance at the cost of extra hardware. Depth-based transmission and ALE are proposed in [29], where the depth of the scene is calculated and carefully calibrated to differentiate distant objects from white objects. A saturation-based image dehazing architecture is presented in [30], which computes transmission on a pixel-by-pixel basis. Although it consumes fewer hardware resources, still its performance is limited by the smaller window size used for the ALE.

D. Contributions of the Proposed Work

The hardware implementation of an image dehazing algorithm is mainly constrained by on-chip memory, which is required to store image frames and hardware resources required to implement dehazing logic. Performing complex mathematical operations with the least logic resources without affecting the output image quality is a challenging task while implementing any image dehazing algorithm on hardware. In this article, we propose an image dehazing architecture and its hardware implementation based on DCP and saturation of hazy and haze-free images. The proposed method estimates the atmospheric light using the concept of dark channel and transmission map using saturation of hazy and haze-free images. Several aforesaid methods have also used DCP as the basis to calculate atmospheric light or airlight. However, due to the small patch size, minimum filtering is likely to select the wrong candidate for atmospheric light when large white objects are present in the hazy image. A large window of size 15×15 is suitable for minimum filtering to obtain atmospheric light as shown in [3]. The implementation of such a large-size minimum filter requires not only a large number of hardware resources but also a large number of line buffers (LBs) to store partial image frames required for minimum filtering. The key contributions of this article are the following.

- 1) The existing designs use 3×3 size minimum filter architectures. However, we designed a 15×15 size minimum filter architecture with comparable hardware resources to improve the quality of the dehazed images in real time.
- 2) The saturation-based transmission estimation method employed in this work operates on a pixel-to-pixel basis, and it does not introduce artifacts around depth discontinuities. This eliminates the necessity of an edge-preserving filter required to suppress halo artifacts.

The qualitative and quantitative results show that the proposed method provides better results than the similar methods in existence and that it consumes moderate hardware resources. Thus, it is a suitable candidate for real-time applications.

The remaining article consists of four sections. Section II discusses the image formation model and the DCP method. The proposed minimum filter and the dehazing architecture are elaborated in Section III. Section IV presents the experimental results and their analysis. Finally, the proposed work is summarized and concluded in Section V.

II. BACKGROUND

The most popular hazy image formation model widely used in machine vision is given below

$$H(x) = D(x)t(x) + A(1 - t(x)) \quad (1)$$

where H is the hazy image, D is the image without haze or dehazed image, t is the transmission of the medium, A is the atmospheric light, and x is the coordinate of a particular pixel located in the image. On rearranging (1), we get

$$D(x) = \frac{H(x) - A}{t(x)} + A. \quad (2)$$

It would be an easy task to obtain dehazed image D using (2) if t and A were known. Here, the only known quantity is H , which makes the dehazing problem ill-posed. However, if some prior knowledge is applied to estimate unknown quantities in (2), the dehazing process will be much simplified. Using the DCP technique, the dark channel of any image I is given as

$$I^{\text{dark}}(x) = \min_{y \in \Omega(x)} \left\{ \min_{c \in (R, G, B)} I^c(y) \right\} \quad (3)$$

where I^c is the R, G, B color channel of I and $\Omega(x)$ is a local region or patch with a pixel having coordinate x located at the center of the patch. If I is a natural haze-free image, the dark channel of I resulting out of two minimum operators has very low-intensity pixels, which can be represented as

$$I^{\text{dark}}(x) = \min_{y \in \Omega(x)} \left\{ \min_{c \in (R, G, B)} I^c(y) \right\} \rightarrow 0. \quad (4)$$

This is because for outdoor natural images, in a local region, some pixels have very low intensity in one of the three color channels. Assuming A is known, on normalizing (1) with A and obtaining dark channel of both sides of the normalized equation, we get

$$\min_{y \in \Omega(x)} \left\{ \min_c \frac{H^c(y)}{A^c} \right\} = t(x) \min_{y \in \Omega(x)} \left\{ \min_c \frac{D^c(y)}{A^c} \right\} + 1 - t(x) \\ c \in (R, G, B). \quad (5)$$

Since D is a dehazed image and A is always positive, using the result of (4), we get

$$\min_{y \in \Omega(x)} \left\{ \min_{c \in (R, G, B)} \frac{D^c(y)}{A^c} \right\} = 0. \quad (6)$$

Therefore, using (5) and (6), transmission can be obtained as

$$t(x) = 1 - \min_{y \in \Omega(x)} \left\{ \min_{c \in (R, G, B)} \frac{H^c(y)}{A^c} \right\}. \quad (7)$$

With the knowledge of A , we can easily obtain t using DCP. However, (6) is not always true. Moreover, transmission is not always constant within a local region or a patch, which results

in inaccurate transmission estimation using (7) especially when a patch comprises objects with different depths. This leads to halo artifacts, and some postprocessing techniques are required to mitigate those artifacts. Hence, scene restoration becomes a complex task.

In [16], patch size is reduced to 1×1 , which causes the transmission to vary from pixel to pixel, making transmission estimation more realistic. Moreover, the assumption of DCP presented in (4) and (6) is also discarded in [16]. Thus, with the patch size of 1×1 and using (5), transmission can be obtained as given below

$$t(x) = \frac{1 - \left(\min_{c \in (R, G, B)} \frac{H^c(x)}{A^c} \right)}{1 - \left(\min_{c \in (R, G, B)} \frac{D^c(x)}{A^c} \right)}. \quad (8)$$

Since, the actual dehazed image is unknown and we simply estimate the parameters of the dehazed image, we have replaced the dehazed image D with an estimated dehazed image D' . For any image M , if $S_M(x)$ denotes the saturation value of a pixel in M at some location x , then $S_M(x)$ is defined as

$$S_M(x) = 1 - \frac{\min_{c \in R, G, B} M^c(x)}{K_M(x)}. \quad (9)$$

Here, $K_M(x)$, which denotes the intensity of a pixel in M located at x , is defined as

$$K_M(x) = \frac{M^R(x) + M^G(x) + M^B(x)}{3}. \quad (10)$$

Using the result of (9) in (8), we get

$$t(x) = \frac{1 - K_{Hn}(x)(1 - S_{Hn}(x))}{1 - K_{D'n}(x)(1 - S_{D'n}(x))} \quad (11)$$

where $K_{Hn}(x)$, $K_{D'n}(x)$, $S_{Hn}(x)$, and $S_{D'n}(x)$ are obtained by normalizing H and D' with respect to A . $K_{D'n}(x)$ can be obtained in terms of $K_{Hn}(x)$ by normalizing (2) with respect to A as follows:

$$K_{D'n}(x) = \frac{K_{Hn}(x) - 1}{t(x)} + 1. \quad (12)$$

Finally, when $K_{D'n}(x)$ is substituted from (12) in (11), a simple linear equation is obtained, which on simplification and rearrangement yields t as

$$t(x) = 1 - K_{Hn}(x) \left(1 - \frac{S_{Hn}(x)}{S_{D'n}(x)} \right). \quad (13)$$

Saturation and intensity values are used in [17] to estimate transmission where a fitting coefficient ψ is introduced to control the degree of refinement of the initial transmission map. This gives a modified transmission equation as

$$t(x) = 1 - \psi \frac{K_H}{A} \left(1 - \frac{S_H(x)}{S_D(x)} \right). \quad (14)$$

Once the transmission is estimated, scene recovery can be achieved using (2). However, $S_{D'n}(x)$ is still an unknown quantity, and it is required to estimate transmission. In [16], an approximate condition showing $S_{D'n}(x) \geq S_{Hn}(x)$ was obtained, which is sufficient for haze removal. This approximation can be utilized to calculate $S_{D'n}(x)$ using the concept of stretch functions. Thus, with the knowledge of A ,

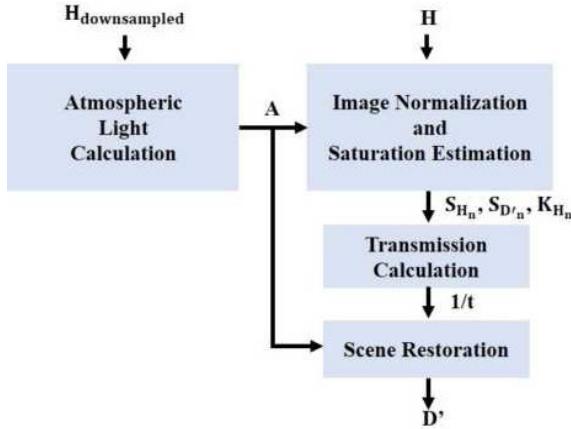


Fig. 1. Flow diagram of the implemented dehazing algorithm.

dehazing can be achieved easily using the saturation of the input hazy image. While DCP serves as a simple yet efficient method to estimate atmospheric light, its hardware implementation is quite complex due to the large size of the minimum filter (optimally 15×15) and sorting of the top 0.1% brightest pixels in the dark channel. In [24], [25], and [27], A is calculated using a 3×3 size minimum filter, which makes their method susceptible to inaccurate estimation of A , which sometimes over saturates the recovered images.

The entire process of image dehazing using the above mathematical equations can be represented in the form of a flow diagram, as shown in Fig. 1. It is also clear from Fig. 1 that atmospheric light A should be known prior to the start of the dehazing process. Saturation-based transmission map estimation is efficient in representing fine textures and edges in the recovered haze-free image, and hence it is employed in the proposed method. The proposed dehazing architecture is further elaborated in Section III.

III. PROPOSED HARDWARE ARCHITECTURE

A complete block diagram of the proposed dehazing architecture is depicted in Fig. 2. There are seven pipeline stages in this architecture. First, the downsampled hazy image is fed to the ALE unit through LBs and an RB. The ALE module performs 15×15 minimum filter operation to estimate atmospheric light A using the concept of DCP. This requires scanning of the downsampled image. Once the value of A is determined, the image normalization and saturation estimation module calculates the saturation of hazy S_{H_n} and haze-free image $S_{D'_n}$ using the normalized image. Saturation and normalized image pixel values are further sent to the transmission estimation module, which calculates the reciprocal of the transmission t . Finally, the scene restoration unit utilizes A , H , and t values to generate the haze-free image D' . In this design, complex dividers are replaced with lookup tables to reduce hardware cost. Pipeline registers (PRs) are used to curtail the critical path length as well as to transfer the intermediate results from one stage to the next.

A. ALE Module

Single image dehazing is much simplified if A would have been known beforehand. A simple yet compute-intensive

technique to estimate A is proposed in [3], which requires a minimum filter with a large window size. In addition, the sorting process is employed to estimate A , which makes it unfit for hardware implementation. While estimating A , the technique of [3] was adopted in [24] and [27] to obtain the dark channel. However, these hardware architectures were presented with the following two changes.

- 1) The minimum filter size was reduced to 3×3 .
- 2) Sorting process was eliminated.

Consequently, the requirements for the number of LBs and other hardware resources have been reduced. It has also reduced the latency. However, with the smaller size of the minimum filter, the accuracy of (4) reduces, the dark channel becomes brighter, and the recovered image gets oversaturated. A complex method is adopted in [26] to estimate A , where a threshold is computed using the gray level of the hazy image. Based on this threshold value, the entire image is divided into bright and dark parts, and the average weight of each part is further calculated to estimate A , thereby limiting the overall speed of the design. In the proposed method, we use the concept of DCP [3] on the downsampled hazy image H to compute A . It provides a fair estimate of A and does not affect the speed of the overall architecture. Furthermore, we propose to avoid sorting the top 0.1% of the brightest pixels in the dark channel to reduce the execution time.

In the proposed architecture, the downsampled hazy image H_{DS} is first fed to the ALE module through LBs and an RB, as shown in Fig. 2. Since the most haze-opaque region in the hazy image gives a better estimate of A , downsampling the hazy image by a small factor would still retain a sufficient haze-opaque region. We have used a downsampling factor of 2 in this work, which reduces the size of the hazy image to half. Consequently, the size of the LBs required to store the image pixels for minimum filtering has also been reduced to half.

A 15×15 window can be divided into 25 nonoverlapping 3×3 windows holding pixel values of consecutive 15 rows and 15 columns. However, if an image is downsampled by a factor of 2, each 3×3 window will hold pixel information of alternate rows and columns. Thus, only nine 3×3 windows with alternate rows and columns are sufficient to implement a 15×15 window. Initially, in Stages 2 and 3, 3×3 minimum filtering is performed on the downsampled hazy image in raster scan format. This is implemented using two LBs and a set of registers for each R, G, and B channel. The results of 3×3 minimum filtering are further stored in LBs in Stage 3, as shown in Fig. 3(a). Once 3×3 minimum filtering results up to M15 and N15 are available in the LBs, 3×3 minimum filtering is performed again in Stages 4 and 5, as shown in Fig. 3(b). This requires five LBs and a set of registers for each R, G, and B channel. Furthermore, a min3 operation is performed in Stage 6 to determine the minimum of all three color channels to obtain the dark channel of the hazy image. Finally, in the run time, the pixel in the dark channel that appears as the brightest pixel in the dark channel is chosen as the atmospheric light, as shown in Fig. 4. In this work, it is assumed that the minimum value of A cannot be less than 100, which is quite reasonable for hazy images.

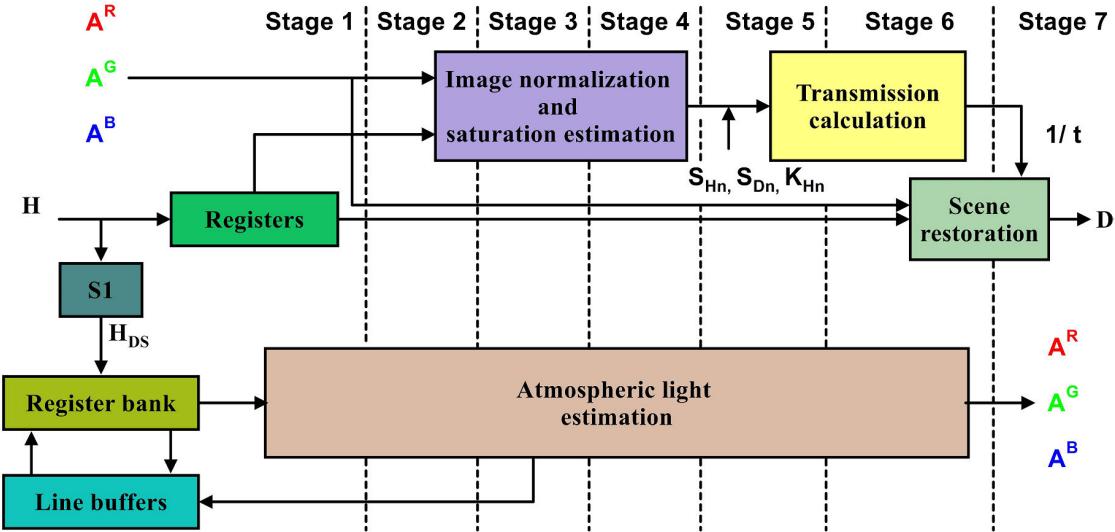
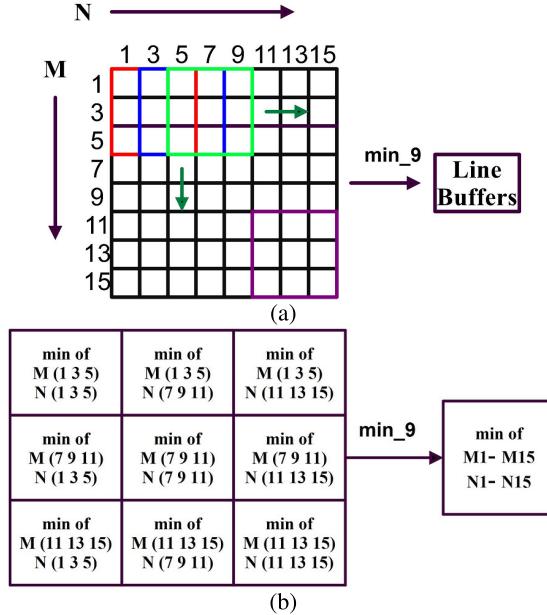


Fig. 2. Block diagram of the proposed VLSI architecture for dehazing.

Fig. 3. (a) 3×3 minimum filter operation is performed on the downsampled hazy image, and the result is stored in LBs and (b) a 15×15 minimum filter for the downsampled hazy image using a 3×3 minimum filter.

The size of the LUTs in Stage 2 of the image normalization and saturation estimation module gets reduced with this assumption, without any compromise in the performance of the proposed method. This fact is verified experimentally on image datasets used for performance evaluation, as discussed in Section IV. For an image of width w , a total of 21 LBs of size $w/2$ are required, which is equivalent to 10.5 LBs of width w . The architecture for the estimation of A is shown in Fig. 4.

B. Image Normalization and Saturation Estimation Module

The overall architecture of this module is shown in Fig. 5. Once A is available from the ALE module in Stage 7,

H is normalized in all three color channels with its respective A in Stage 2. Normalized hazy image is used to calculate the saturation of input hazy image in Stage 3 using (9). Furthermore, the contrast stretch function [16] is employed in Stage 4 to estimate the saturation of the haze-free image D . From (12) and (13), it can be easily derived that $S_{D'}(x) \geq S_H(x)$. A simple and hardware-friendly contrast stretch function satisfying this condition is given below

$$S_{D'}(x) = S_H(x)(2.0 - S_H(x)). \quad (15)$$

Thus, the saturation information of a hazy image can be utilized to estimate the saturation information of a haze-free image. In the process of estimating the saturation values of the hazy and haze-free images, mathematical and logical operations are performed such that the saturation values of the hazy and haze-free images are scaled to 12 bits to maintain accuracy and prevent loss of data for smaller values. Apart from (15), several other stretch functions are also available. However, they require exponential functions to be implemented in hardware, which is resource-consuming. Moreover, it is shown in [16] that the dehazing outcome does not depend extensively on the choice of stretch function, which was also verified experimentally by us on various image datasets discussed in Section IV before choosing the contrast stretch function given by (15).

C. Transmission Estimation Module

Fig. 6 depicts the hardware architecture of this module. This module receives the normalized pixel value and saturation value from the image normalization and saturation estimation module in Stage 5. This module calculates the reciprocal of transmission by reordering (13) in Stage 6. The calculated value of transmission is further passed to the scene restoration module in Stage 7 to obtain the dehazed image. In this work, we have modified (13) by incorporating fitment factor ψ used

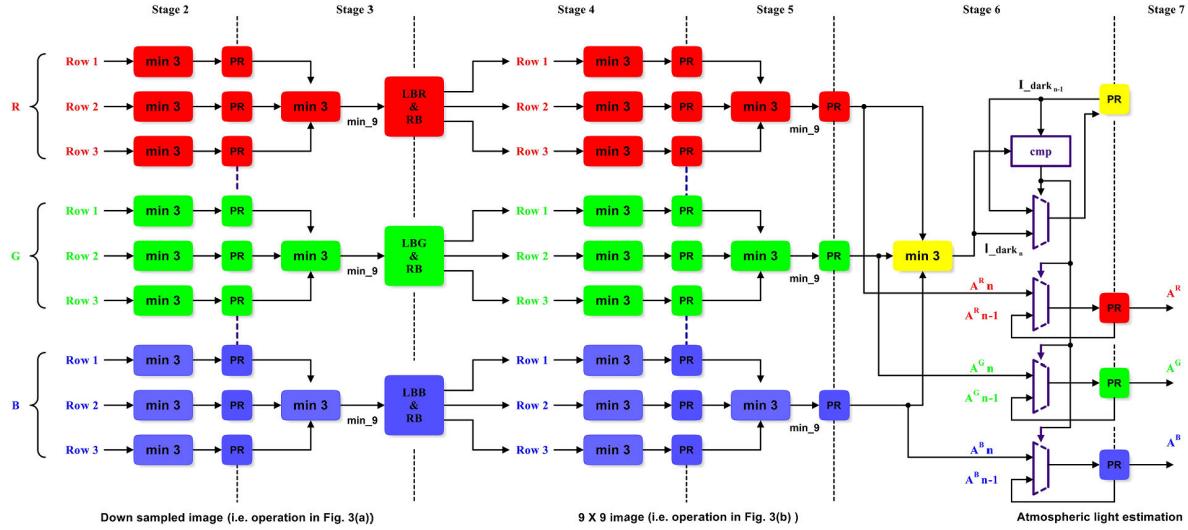


Fig. 4. Architecture of ALE.

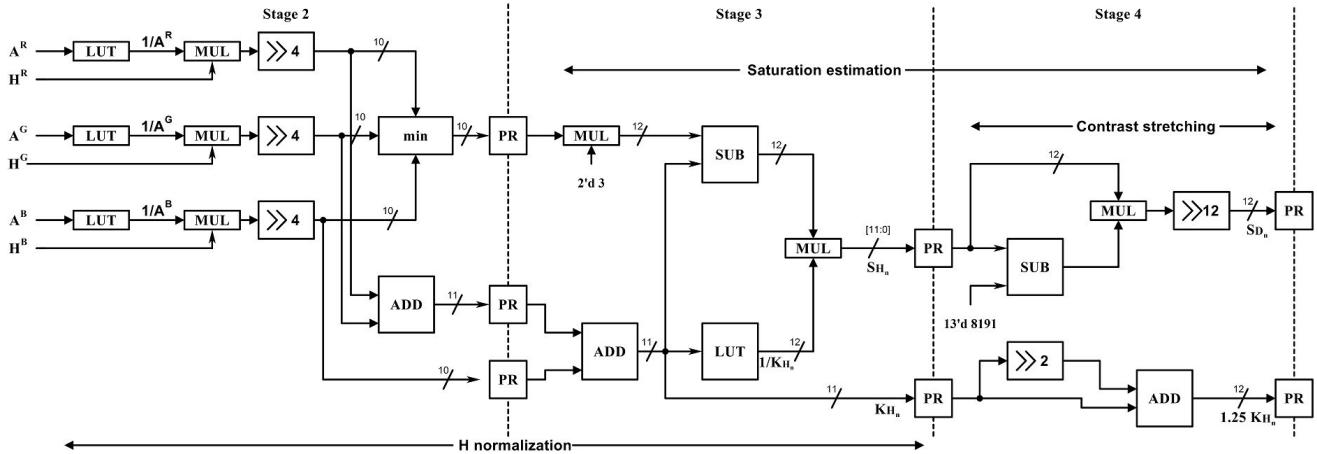


Fig. 5. Architecture of the image normalization and saturation estimation module.

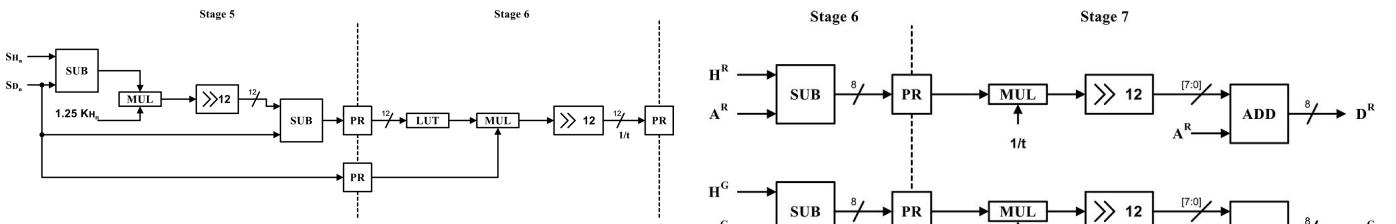


Fig. 6. Architecture of the transmission estimation module.

in (14). However, instead of using a complex iterative procedure to find the optimum value of ψ , we fixed it at 1.25 so that hardware implementation becomes easy and recovered images are neither over- nor underdehazed.

D. Scene Restoration Module

This is the final module of the proposed dehazing method, and its architecture is shown in Fig. 7. This module utilizes A and $1/t$ values along with H to restore the dehazed image D' using (2). In the last stage, i.e., Stage 7, the output of the multiplier, which represents the first term in (2), is scaled down to 12 bits and added to A to produce 8-bit output.

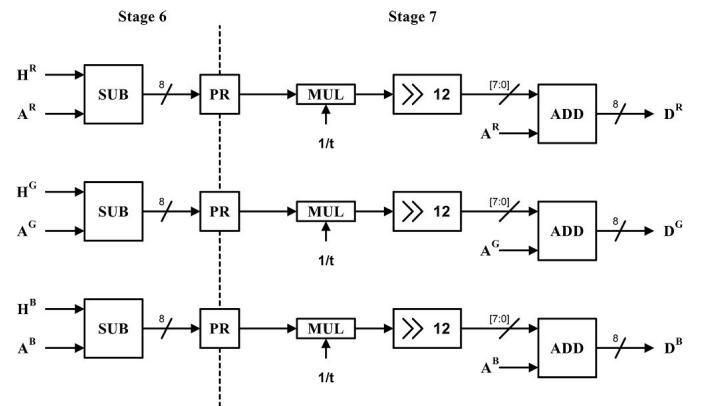


Fig. 7. Architecture of the scene restoration module.

IV. RESULTS AND DISCUSSION

This section consists of the qualitative and quantitative performance evaluation of the proposed dehazing architecture and its comparison with the existing hardware architectures. Although several image dehazing methods have been published, only a few of them have been implemented on

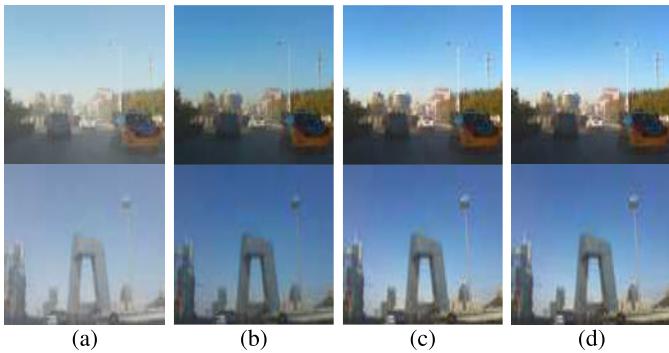


Fig. 8. Restored images using different A . (a) Hazy images. (b) and (c) Restored images with DCP using the 3×3 and 15×15 size minimum filters, respectively. (d) Restored images with the proposed method for the 15×15 size minimum filter.

hardware. The proposed work is aimed at developing an effective method to dehaze images in real time. Therefore, we selected those methods that were implemented in hardware and compared their performance with the proposed method.

First, we performed MATLAB simulations to determine an appropriate size of the minimum filter window. We experimented with different window sizes starting from 3×3 to 15×15 . However, the results for the 3×3 and 15×15 size minimum filters only are reported here. Initially, we estimated atmospheric light A using the DCP [3] algorithm to restore images, and results with 3×3 and 15×15 minimum filters are depicted in Fig. 8(b) and (c), respectively. Next, we used the proposed method (detailed in Section III) to estimate atmospheric light A and repeated the experiments. The results with the proposed method for the 15×15 size minimum filter are depicted in Fig. 8(d). It is clear from Fig. 8 that when a 3×3 size minimum filtering window is used, the estimation of A is inaccurate. Hence, restored images are oversaturated.

However, when images are restored using A with a 15×15 size minimum filter, the output of the proposed method looks more natural and its visual quality is similar to that of the DCP [3] method. We used ten hazy images (whose ground truth is also available) from the publicly available and widely used synthetic objective testing set (SOTS) dataset, which comprises indoor and outdoor synthetic hazy images, and the NYU dataset, which comprises indoor images of varying depths to perform quantitative analysis of the proposed design. These test images are shown in Fig. 9. CIEDE2000 [31] metric represents color fidelity in the recovered image, and a lower value of CIEDE2000 implies a less color difference between the recovered image and the ground truth. We applied the proposed method and computed CIEDE2000 for these test images with window sizes starting from 3×3 to 15×15 . However, the bar chart plots for the 3×3 and 15×15 size minimum filters only are presented in Fig. 10. It can be observed from Fig. 10 that a smaller window size (i.e., 3×3) has resulted in a higher value of CIEDE2000 index for most of the test images because the 3×3 size minimum filter overestimates the value of A . On the contrary, color fidelity in the restored images is better for most of the test images with the minimum filter of size

15×15 . Furthermore, with the 15×15 size minimum filter, the CIEDE2000 value does not differ much for the original and downsampled hazy image, as shown in Fig. 10. This shows that the estimation of A does not get affected even if the image is downsampled and better estimation is possible with a larger window size, as mentioned in [3]. Moreover, downsampling the hazy image reduces scan time to estimate A , and the proposed method requires only $w/2 \times h/2$ clock cycles to estimate A , whereas the methods of [24], [26], [27] require almost $w \times h$ clock cycles to estimate A . Therefore, we used the downsampled version of the hazy image and 15×15 size minimum filters in our hardware implementation.

Next, we selected the existing methods [24], [25], [26], [27] as they were implemented in hardware. We implemented their algorithm in MATLAB and compared their performance with that of the proposed dehazing algorithm. Finally, we implemented the proposed algorithm on FPGA as well as ASIC platforms. Fig. 11 shows the visual results obtained on various hazy images from the standard datasets. It is clear from Fig. 11 that the images recovered using the methods presented in [24], [26], and [27] suffer from oversaturation, especially in the sky region. This is due to the overestimation of atmospheric light resulting from a small size minimum filter which is used to obtain the dark channel. Moreover, the method of [26] performs overdehazing in the sky region, as can be seen in the result of Image 2 depicted in Fig. 11(d). Indoor images recovered using the proposed method are also smoother than the existing methods.

Quantitative performance evaluation was carried out by computing peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [32], and CIEDE2000 metrics of the dehazed image with respect to the ground truth in MATLAB on all the images in SOTS, NYU, and O-HAZE datasets, and the results are presented in Tables I–III. A higher value of SSIM implies that the structural information of the recovered image is well preserved by the dehazing technique. From the results obtained from Tables I–III, it can be observed that the proposed dehazing architecture produces superior results than the existing hardware architectures. Moreover, the result of the proposed method is comparable to the state-of-the-art deep learning methods except for the PSNR and SSIM results of [22], which performs the best on the SOTS dataset. However, their results on NYU datasets are unavailable. Furthermore, deep learning-based methods require high-performance GPUs and processors for their implementation, which is quite impractical for real-time applications. For real-time analysis, we tested the proposed as well as the existing methods using MATLAB on Intel's i7-9700 @ 3-GHz processor with 8-GB RAM, and the execution time for different image sizes is presented in Table IV. It is clear from these results that the proposed method performs much better than the methods of [24], [26], [27]. However, the method of [25] is the fastest because it does not use DCP to estimate atmospheric light.

The proposed dehazing architecture is designed using Verilog HDL and implemented on ZynQ7 XC7Z020CLG484-1 FPGA using Xilinx Vivado Design Suite. The FPGA implementation results of the proposed design are presented in

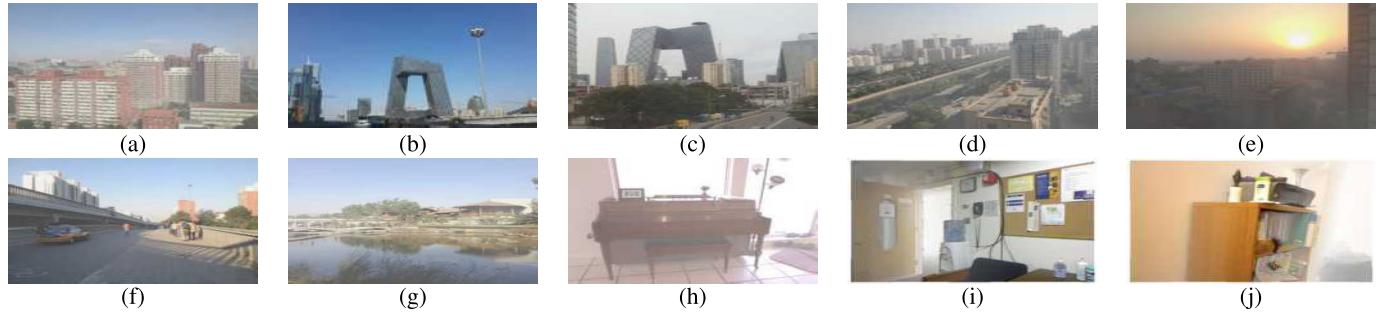


Fig. 9. Test images used for quantitative performance evaluation. (a) Image 1. (b) Image 2. (c) Image 3. (d) Image 4. (e) Image 5. (f) Image 6. (g) Image 7. (h) Image 8. (i) Image 9. (j) Image 10.

TABLE I
PERFORMANCE COMPARISON OF VARIOUS DEHAZING METHODS ON SOTS DATASET USING PSNR AND SSIM METRICS

	Deep learning methods						VLSI architectures											
	[18]		[19]		[22]		[24]		[25]		[26]		[27]		The proposed			
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SOTS (Outdoor)	22.46	0.8514	20.29	0.8765	-	-	18.83	0.8120	18.29	0.8241	15.23	0.6562	15.53	0.6459	21.42	0.8791		
SOTS(indoor)	21.14	0.8472	19.06	0.8504	-	-	18.42	0.7933	16.33	0.7697	16.30	0.7496	17.09	0.7704	18.68	0.8310		
Average	21.80	0.8493	19.68	0.8634	24.23	0.9431	18.63	0.8026	17.31	0.7969	15.76	0.7029	16.31	0.7082	20.05	0.8551		

TABLE II

PERFORMANCE COMPARISON OF VARIOUS DEHAZING METHODS ON NYU AND O-HAZE DATASETS USING PSNR AND SSIM METRICS

	Deep learning methods				VLSI architectures											
	[18]		[20]		[24]		[25]		[26]		[27]		The proposed			
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NYU	12.84	0.7175	12.26	0.7000	11.41	0.6723	11.10	0.6512	10.86	0.6422	11.85	0.6889	12.71	0.7356		
O-HAZE	15.30	0.4110	17.14	0.4370	14.37	0.3502	15.60	0.4233	13.85	0.4565	13.73	0.3292	16.45	0.4411		

TABLE III
PERFORMANCE COMPARISON ON SOTS, NYU, AND O-HAZE DATASETS USING CIEDE2000 METRIC

Dataset	Deep learning methods			VLSI architectures					
	[18]	[19]	[20]	[24]	[25]	[26]	[27]	The proposed	
PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SOTS	8.8481	7.6742	10.7991	9.8256	10.3588	12.1012	11.0216	7.5471	
NYU	15.8782	16.6028	17.4497	18.9124	18.0135	17.2362	16.5489	15.6824	
O-HAZE	16.8700	15.0800	19.7600	20.2100	17.2561	19.2855	21.6108	15.4280	

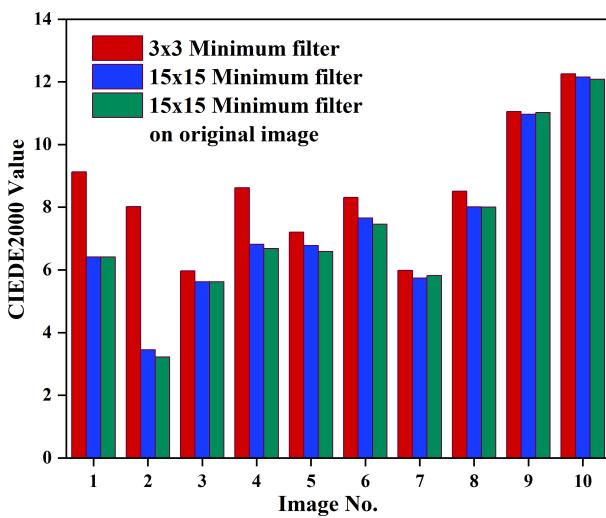


Fig. 10. Bar plot of CIEDE2000 metric with the proposed method for 3 × 3 and 15 × 15 window minimum filters.

Table V. These results show that the proposed design consumes only 1537 logic elements (LEs) and operates at 85.2-MHz frequency. Although the method of [25] consumes the least LEs and can operate at 116 MHz, due to dynamic

TABLE IV
EXECUTION TIME REQUIREMENTS (UNIT: SECONDS)

Architecture	Image size			
	550×413	832×776	1165×709	1800×1574
[24]	0.062	0.183	0.242	0.967
[25]	0.002	0.005	0.008	0.036
[26]	0.078	0.238	0.316	1.102
[27]	0.086	0.262	0.344	1.292
The proposed	0.028	0.091	0.128	0.545

ALE, it may result in discontinuous layers in the recovered images, which degrades the quality of the recovered image as can be seen in the result of Image 4 shown in Fig. 11(c). The methods of [24], [25], [27] have used Intel (Altera) FPGA manufactured at a higher technology node with LEs as basic building blocks, whereas the proposed architecture is implemented using configurable logic blocks (CLBs) present in the AMD-Xilinx FPGA. The size and complexity of these building blocks (LEs and CLBs) differ from each other. Furthermore, the existing architectures were implemented using a different (Quartus) design suite.

The ASIC implementation of the proposed architecture is also carried out using the Synopsis Design Vision tool at

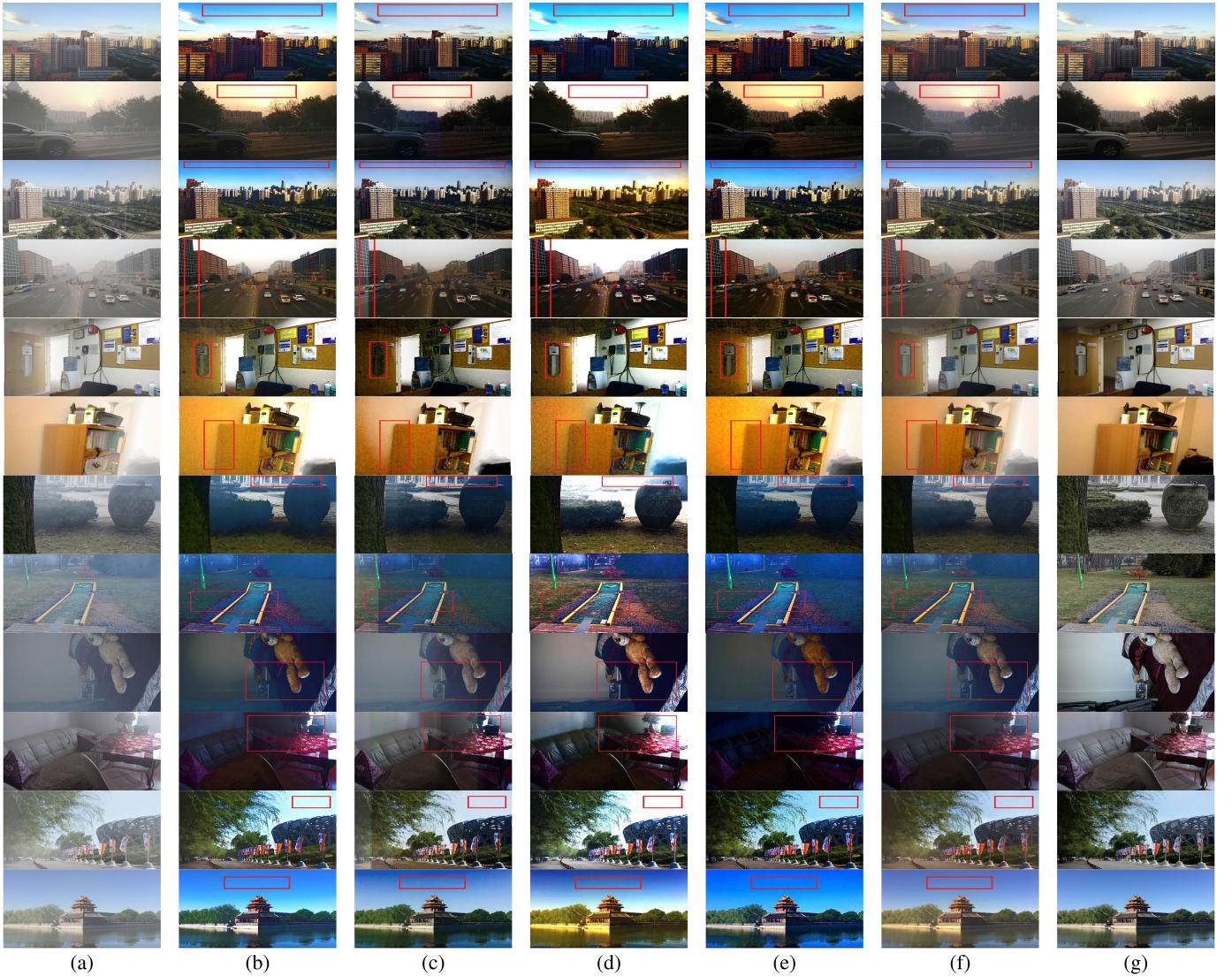


Fig. 11. Simulation results of recovered images obtained with different haze removal methods. (a) Hazy image, (b) results with [24], (c) results with [25], (d) results with [26], (e) results with [27], (f) proposed method, and (g) ground truth.

TABLE V
FPGA IMPLEMENTATION RESULTS

Architecture	[24]	[25]	[27]	The proposed
Family	Stratix	Stratix	Stratix	Zynq7000
Device	EP1S10F780C6	EP1S10F780C6	EP1S10F780C6	XC7Z020CLG484-1
No. of LEs [†]	1607	1094	3169	1537*
Registers	454	539	651	547
Frequency (MHz)	58.43	116	58.82	85.2
Throughput (Mpixels/s)	58.43	116	58.82	85.2
Line buffers	6	6	6	10.5

[†]The logic elements (LEs) of Xilinx and Intel FPGAs are different from each other.

* Total LUT count

a 65-nm CMOS technology node, and the results are presented in Table VI. Since the designs of [24], [26], [27] are synthesized at 130 nm, we have also normalized our ASIC implementation results to 130 nm [28] for a fair comparison as follows:

$$\text{Scaling factor } x = \frac{65 \text{ nm}}{130 \text{ nm}} \quad (16)$$

$$\text{frequency}_{130 \text{ nm}} = \text{frequency}_{65 \text{ nm}} \times x \quad (17)$$

$$\text{Power}_{130 \text{ nm}} = \text{Power}_{65 \text{ nm}} \times \frac{1}{x^2} \quad (18)$$

TABLE VI
ASIC IMPLEMENTATION RESULTS

Architecture	[24]	[26]	[27]	[29]	The proposed
CMOS Technology	130nm	130nm	130nm	180nm	65nm
Gate count (K)	12.8	23.7	18.6	14.4	13.2
Frequency (MHz)	200	200	200	250[346]*	624[312]*
Throughput (Mpixels/s)	200	200	200	250[346]*	624[312]*
Power @ 200MHz (mW)	11.9	13.4	NA	15.2[7.88]*	2.62[10.48]*
Power delay product (PJ)	59.5	67	NA	22.77*	33.59*

* Normalized to 130nm

$$\text{Scaling factor } y = \frac{130 \text{ nm}}{180 \text{ nm}} \quad (19)$$

$$\text{frequency}_{180 \text{ nm}} = \text{frequency}_{130 \text{ nm}} \times y \quad (20)$$

$$\text{Power}_{180 \text{ nm}} = \text{Power}_{130 \text{ nm}} \times \frac{1}{y^2}. \quad (21)$$

ASIC implementation results show that the proposed design comprises only 13.2k gates and can operate at 624 MHz, consuming only 2.62-mW power at 200 MHz. Although the gate count of the proposed design is higher than that of [24], it can operate at a higher frequency and consumes less power than the rest of the designs except [29], which is more power

efficient and can operate at a higher frequency. However, the power and speed of the proposed architecture are comparable to those of [29] with a lower gate count. Despite using a 15×15 size minimum filter, the hardware cost and the computation time of the proposed design have not increased significantly. This is the major advantage of the proposed design.

V. CONCLUSION

In this article, we present a seven-stage pipelined image dehazing architecture based on the DCP and saturation of the input hazy image. The proposed dehazing architecture estimates atmospheric light based on the concept of DCP using an optimum-sized minimum filter. Furthermore, the proposed dehazing architecture utilizes saturation-based transmission map estimation. Therefore, it works on a pixel-to-pixel basis, thereby eliminating the requirement of an edge detection and image filtering unit, which further reduces the hardware cost as well as suppresses halo artifacts around edges. Despite consuming some extra LBs as compared with other existing methods, the qualitative and quantitative results obtained with the proposed method are superior to the existing methods implemented on hardware platforms except [29]. Although the qualitative results of [29] are unavailable, the hardware implementation results of the proposed architecture are comparable to it. ASIC implementation results show that this method can easily process 4k (3840×2160) resolution frames at a rate higher than 70 fps, making it a preferable candidate for real-time image dehazing applications such as remote sensing and advanced driver assistance system (ADAS). However, the performance of this design is inefficient under a dense hazy condition like other existing image dehazing hardware architectures. There is scope to mitigate this problem.

REFERENCES

- [1] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [2] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, Aug. 2008, doi: [10.1145/1360612.1360671](https://doi.org/10.1145/1360612.1360671).
- [3] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.
- [4] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3522–3533, Nov. 2015.
- [5] Y.-H. Lai, Y.-L. Chen, C.-J. Chiou, and C.-T. Hsu, "Single-image dehazing via optimal transmission map under scene priors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 1–14, Jan. 2015.
- [6] L. He, J. Zhao, N. Zheng, and D. Bi, "Haze removal using the difference-structure-preservation prior," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1063–1075, Mar. 2017.
- [7] F. Yuan and H. Huang, "Image haze removal via reference retrieval and scene prior," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4395–4409, Sep. 2018.
- [8] T. M. Bui and W. Kim, "Single image dehazing using color ellipsoid prior," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 999–1009, Feb. 2018.
- [9] H. Xu, J. Guo, Q. Liu, and L. Ye, "Fast image dehazing using improved dark channel prior," in *Proc. IEEE Int. Conf. Inf. Sci. Technol.*, Mar. 2012, pp. 663–667.
- [10] W. Jin, Z. Mi, X. Wu, Y. Huang, and X. Ding, "Single image dehaze based on a new dark channel estimation method," in *Proc. IEEE Int. Conf. Comput. Sci. Automat. Eng. (CSAE)*, vol. 2, May 2012, pp. 791–795.
- [11] C. Xiao and J. Gan, "Fast image dehazing using guided joint bilateral filter," *Vis. Comput.*, vol. 28, nos. 6–8, pp. 713–721, Jun. 2012, doi: [10.1007/s00371-012-0679-y](https://doi.org/10.1007/s00371-012-0679-y).
- [12] Z. Lin and X. Wang, "Dehazing for image and video using guided filter," *Open J. Appl. Sci.*, vol. 2, no. 4B, pp. 123–127, 2012.
- [13] S.-C. Huang, B.-H. Chen, and W.-J. Wang, "Visibility restoration of single hazy images captured in real-world weather conditions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1814–1824, Oct. 2014.
- [14] W. Wang, X. Yuan, X. Wu, and Y. Liu, "Fast image dehazing method based on linear transformation," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1142–1155, Jan. 2017.
- [15] A. Galdran, "Image dehazing by artificial multiple-exposure image fusion," *Signal Process.*, vol. 149, pp. 135–147, Aug. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168418301063>
- [16] S. E. Kim, T. H. Park, and I. K. Eom, "Fast single image dehazing using saturation based transmission map estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 1985–1998, 2020.
- [17] L. Y. He, J.-Z. Zhao, and D.-Y. Bi, "Effective haze removal under mixed domain and retract neighborhood," *Neurocomputing*, vol. 293, pp. 29–40, Jun. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231218302662>
- [18] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: An end-to-end system for single image haze removal," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5187–5198, Nov. 2016.
- [19] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "AOD-Net: All-in-one dehazing network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4780–4788.
- [20] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 154–169.
- [21] C. Li, C. Guo, J. Guo, P. Han, H. Fu, and R. Cong, "PDR-Net: Perception-inspired single image dehazing network with refinement," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 704–716, Mar. 2020.
- [22] S. Zhao, L. Zhang, Y. Shen, and Y. Zhou, "RefineDNet: A weakly supervised refinement framework for single image dehazing," *IEEE Trans. Image Process.*, vol. 30, pp. 3391–3404, 2021.
- [23] C. Guo, Q. Yan, S. Anwar, R. Cong, W. Ren, and C. Li, "Image dehazing transformer with transmission-aware 3D position embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022.
- [24] Y. H. Shiau, H. Y. Yang, P. Y. Chen, and Y. Z. Chuang, "Hardware implementation of a fast and efficient haze removal method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1369–1374, Aug. 2013.
- [25] B. Zhang and J. Zhao, "Hardware implementation for real-time haze removal," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1188–1192, Mar. 2017.
- [26] Y.-H. Shiau, Y.-T. Kuo, P.-Y. Chen, and F.-Y. Hsu, "VLSI design of an efficient flicker-free video defogging method for real-time applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 238–251, Jan. 2019.
- [27] Y.-T. Kuo, W.-T. Chen, P.-Y. Chen, and C.-H. Li, "VLSI implementation for an adaptive haze removal method," *IEEE Access*, vol. 7, pp. 173977–173988, 2019.
- [28] Y.-H. Lee and B.-H. Wu, "Algorithm and architecture design of a hardware-efficient image dehazing engine," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2146–2161, Jul. 2019.
- [29] Y.-H. Lee and S.-J. Tang, "A design of image dehazing engine using DTE and DAE techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2880–2895, Jul. 2021.
- [30] B. B. Upadhyay, S. K. Yadav, and K. P. Sarawadekar, "VLSI architecture of saturation based image dehazing algorithm and its FPGA implementation," in *Proc. IEEE 65th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2022, pp. 1–4.
- [31] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Res. Appl.*, vol. 30, no. 1, pp. 21–30, Feb. 2005. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/col.20070>
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.