Chapter 4

Intensity Based Video Dehazing and its VLSI Architecture

4.1 Introduction

In this Chapter, an intensity-based video dehazing method with low hardware complexity is proposed to effectively restore the visibility of the hazy image frames. In the proposed method, the atmospheric light for each pixel in an image frame is dynamically adjusted for efficient dehazing of hazy scenes. Moreover, a new pixel-to-pixel transmission estimation technique is proposed in this Chapter, which reduces the hardware complexity of the proposed method without compromising its accuracy.

The rest of the Chapter is organized as follows. Background of the proposed method is presented in Section 4.2. The proposed haze removal method and its VLSI architecture are described in Section 4.3. Experimental results and performance analysis are discussed in Section 4.4, and Section 4.5 presents the concluding remarks of the Chapter.

Contributions of the proposed work

Hardware implementation of dehazing algorithms is often governed by the trade-off between hardware logic resource consumption and dehazing performance. Researchers have

tried to minimize hardware resources by simplifying dehazing algorithms as seen in [70]–[74], [76], without much degradation in performance. Still, challenges such as overdehazing and oversaturation of dehazed images exist. Due to this, information in the dehazed images is sometimes lost, which is undesirable. This motivated us to develop a low-complexity video dehazing algorithm and its VLSI architecture, which can be integrated with various real-time image processing systems to increase their robustness in a hazy atmosphere. The key contributions of the proposed work are:

- In this work, a new pixel-based dynamic atmospheric light and transmission estimation technique is proposed using the intensity of individual pixels. This prevents over-dehazing of the hazy images and effectively controls the brightness and saturation of the restored images.

- The proposed method eliminates special filters required to suppress artifacts around edges that occur in patch-based transmission estimation techniques. Thus, the hardware resource requirement is also reduced.

- The proposed dehazing architecture requires 13.8 k gates and operates at 460 MHz when synthesized using 90-nm CMOS technology. Thus, the proposed dehazing architecture can process more than 50 hazy video frames of resolution 3840×2160 in a second.

4.2 Background

Obtaining D using (3.1) requires the knowledge of unknowns A and t but the only known quantity in (3.1) is H. This makes the dehazing process quite challenging. However, using certain characteristics of the hazy image, a fair estimate of A and t can be made. One such characteristic is explored in [37], where the dark channel M^{dark} of an image M is obtained by finding the minimum of all the pixels in a sliding local patch Ω for the three (R, G, B) color channels. Finally, the minimum of the three patch minimums is the dark channel

pixel's value of M^{dark}. So,

$$M^{dark}(x) = \underset{y\in\Omega(x)}{min}\{\underset{c\in(R,G,B)}{min} M^c(y)\}. \tag{4.1}$$

For a hazy image, A can be obtained using its dark channel by the method given in [37]. On rearranging (1.1) and normalizing both sides with A, transmission can be expressed in terms of the dark channel of H and D as

$$t(x) = \frac{1 - \underset{y\in\Omega(x)}{min}\left(\underset{c\in(R,G,B)}{min}\frac{H^c(x)}{A^c}\right)}{1 - \underset{y\in\Omega(x)}{min}\left(\underset{c\in(R,G,B)}{min}\frac{D^c(x)}{A^c}\right)}. \tag{4.2}$$

Since, for a haze-free image, one of its three color channels has very low intensity, its dark channel obtained using (4.1) tends to zero, which leads to an approximation given as

$$D^{dark}(x) = \underset{y\in\Omega(x)}{min}\{\underset{c\in(R,G,B)}{min} D^c(y)\} \to 0. \tag{4.3}$$

This approximation, also known as DCP, is further utilized to simplify the transmission estimation given by (4.2). The simplified transmission equation is represented as

$$t'(x) = 1 - \underset{y\in\Omega(x)}{min}\left\{\underset{c\in(R,G,B)}{min}\frac{H^c(y)}{A^c}\right\}, \tag{4.4}$$

where *t'(x)* is the simplified transmission. Once A and t are estimated, D can be obtained from (3.1) as follows:

$$D(x) = \frac{H(x) - A}{t(x)} + A. \tag{4.5}$$

4.3 The Proposed Method and its VLSI Architecture

4.3.1 The Proposed Method

To simplify the dehazing equation (4.5), knowledge of A and t is essential. A fair estimate of A can be made from the most haze-opaque region, which can be identified using the

dark channel of the hazy image. However, if the patch size used to obtain the dark channel is small, a white or bright object may misrepresent the most haze-opaque region in the dark channel. On the contrary, a large patch size would lead to high hardware resource consumption. To address these issues, the hazy image was downsampled by a factor of 3, and a patch size of 5×5 was selected to compute the dark channel of the hazy image, which is equivalent to a 13×13 patch in the original hazy image. Down-sampling the hazy image by a small factor doesn't affect the performance because most haze-opaque regions are often wide and continuous. The airlight profile depicted in [39] suggests that the contribution of the airlight is depth-dependent. It means a depth-based adjustment of A would be useful. Further, the impact of haze on the image's pixels presented in [39] indicates that the brightness of pixels in a hazy image increases with depth. So, the pixels in the dark channel of H are further divided into near and far pixels having values less than or greater than a threshold value T_h which is also equal to the mean brightness of the hazy image obtained as

$$T_h = \frac{\sum_{n=1}^{N} H_{gray}}{N}, \tag{4.6}$$

where H_{gray} is the gray value of a pixel and N is the total number of pixels in the hazy image. The brightest pixel of the far group and near group in the dark channel are selected as the representative of the far region and near region atmospheric light A_F and A_N respectively, which can be obtained as follows

$$\begin{aligned} A_F^C &= \{H_{down}(x) | x = max\{H_{down}^{Far\ dark}(x)\}\}, \\ A_N^C &= \{H_{down}(x) | x = max\{H_{down}^{Near dark}(x)\}\}, \\ & \qquad c \in (R, G, B), \end{aligned} \tag{4.7}$$

where H_{down} is the down-sampled hazy image. Finally, A for each pixel is estimated as given below

$$A^C(x) = \begin{cases} A_F^C & \text{If } I_H(x) \geq I_{max} \\ (1 - \eta(x)) * A_F^C + \eta(x) * A_N^C & \text{otherwise,} \end{cases} \tag{4.8}$$

where I_H is the gray value of the pixel in hazy image at x obtained by averaging the pixel values in all the three color channels, I_{max} is the gray value of A_F^C and η is the weighing factor given as

$$\eta(x) = \frac{I_{max} - I_H(x)}{I_{max}}. \tag{4.9}$$

Near pixels with lower intensities have a higher value of η, whereas far pixels with higher intensities have a lower value of η. Hence, η ensures that the contribution of A_F^C for distant pixels is higher, whereas for nearby scenes, the contribution of A_N^C increases. Pixels with intensity greater than I_{max} are brighter than A_F^C and are considered to be very far pixels. Therefore, their estimated A value equals A_F^C only. On the contrary, pixels with intensity less than I_{max} have the weighted contribution of both A_F^C and A_N^C in their A estimation. In [39], it is depicted that the over-estimation of A results in darker dehazed images, whereas the under-estimation of A may lead to the loss of fine details in bright regions. However, these problems are alleviated in the proposed method by the weighted estimation of A using η. Thus, the proposed method efficiently controls the brightness of the dehazed image while preserving fine details. Once A is estimated, t can be obtained using (4.4). However, (4.4) assumes that the transmission in a local patch is constant. This may produce artifacts around the depth discontinuities in the recovered images. A more pragmatic approach was adopted to eliminate this drawback by estimating the transmission for each pixel individually. This is achieved by reducing the patch size in (4.4) to 1×1. Thus, the pixel-wise transmission (t") estimation is mathematically represented as follows

$$t''(x) = 1 - \left\{ \min_{c\in(R,G,B)} \frac{H^c(y)}{A^c} \right\}. \tag{4.10}$$

The transmission obtained using (4.10) varies from pixel to pixel. Moreover, this approach does not produce halo artifacts around the edges. Therefore, the proposed method does not require any special filters or post-processing techniques to suppress artifacts. Consequently, the algorithmic complexity and hardware resource consumption are also reduced. However, the approximation made in (4.3) to simplify (4.2) is not always true. Due to this, the transmission equation obtained in (4.4) and (4.10) under-estimates the value of

transmission ($t' \leq t$) and ($t'' \leq t$), resulting in over-saturation of the dehazed images. To alleviate this problem, we have incorporated an intensity-based correction factor in (4.10) so that the inaccuracy in transmission estimation due to the assumption made in (4.3) can be reduced. The modified transmission estimation equation is as follows

$$t_m(x) = 1 - \omega * \left(\underset{c \in (R,G,B)}{min} \frac{H^c(x)}{A^c} \right) * k(x), \tag{4.11}$$

where t_m is the modified transmission, ω is a factor used to keep a small amount of haze, and *k(x)* is the correction factor which is given as

$$k(x) = 1 - \left(\frac{\underset{c \in (R,G,B)}{min} H^c(x)}{\underset{c \in (R,G,B)}{min} H^c + \underset{c \in (R,G,B)}{max} H^c(x)} \right)^{\gamma}. \tag{4.12}$$

The factor γ controls the rate of decrease of *k* with the increase in the intensity of the pixels in H. The correction factor in (4.11) helps in a fair estimation of transmission, as it would using the exact transmission estimation equation given by (4.2) on reducing the patch size to 1×1. Since D in (4.2) is unknown, it cannot be directly used to estimate *t*. However, using some prior information, the behavior of (4.2) can be mimicked to estimate the transmission precisely. Since haze increases the intensity of the pixels in an image, it is obvious that

$$\underset{c \in (R,G,B)}{D^c}(x) \leq \underset{c \in (R,G,B)}{H^c}(x). \tag{4.13}$$

For patch size 1×1, the denominator of (4.2) satisfies the condition $0 \leq 1 - \left(\underset{c \in (R,G,B)}{min} \frac{D^c(x)}{A^c} \right) \leq 1$. For low-intensity pixels in H, the intensity of the corresponding pixels in D will also be low. Hence, $D << A$ and $1 - \left(\underset{c \in (R,G,B)}{min} \frac{D^c(x)}{A^c} \right) \approx 1$. However, with increase in the intensity of the pixels in H, the intensity of the corresponding pixels in D will also increase, leading to a decrease in the value of the denominator of (4.2). This will result in the non-linear boosting of *t* with the increase in the intensity of the pixels in H. A similar non-linear boost in t_m with the increase in the intensity of the pixels in H is also obtained from (4.11). This is because the correction factor *k* in (4.13) decreases nonlinearly with

(a) (b) (c) (d) (e)

Figure 4.1: Dehazing results for different values of γ. (a) Hazy image. (b) Result with $\gamma = 1.0$. (c) Result with $\gamma = 1.5$. (d) Result with $\gamma = 2.0$. (e) Result with $\gamma = 2.5$.

the increase in the intensity of the pixels in M_H. Since k always reduces the second term in (4.11), t_m also experiences a nonlinear boost with the increase in the intensity of pixels in M_H. The increase in the intensity of pixels due to haze is depth-dependent [104]. Near pixels see lower intensity boosting due to haze compared to farther pixels. This leads to higher values of k for near pixels in M_H and vice versa. The nonlinear variation of k with the intensity of the pixels in M_H causes the distant pixels to experience a higher boost in t_m compared to the near ones. This boosting pattern is desirable as haze reduces the transmission of the distant pixels much more than the closer ones. Finally, (4.5) is utilized to produce the dehazed image. The dehazing results obtained with different values of γ are depicted in Fig. 4.1, which shows that the dehazing performance increases with increase in the γ values. However, for higher values of γ, dehazed images may be oversaturated. The appropriate value of γ is selected based on an ablation study presented in Section 4.4.

4.3.2 The Proposed VLSI Architecture

The block-level architecture of the proposed design is presented in Fig. 4.2. It consists of six blocks named a pixel buffer (PB), an atmospheric light estimation (ALE), an average gray value calculation (AGVC), a pixel-wise atmospheric light adjustment (PALA), a pixel-wise transmission estimation (PTE), and a scene dehazing (SD) block. There are ten pipeline stages in this design. The PB block supplies the pixel values to the remaining blocks for dark channel calculation, T_h calculation, t estimation, and scene dehazing.

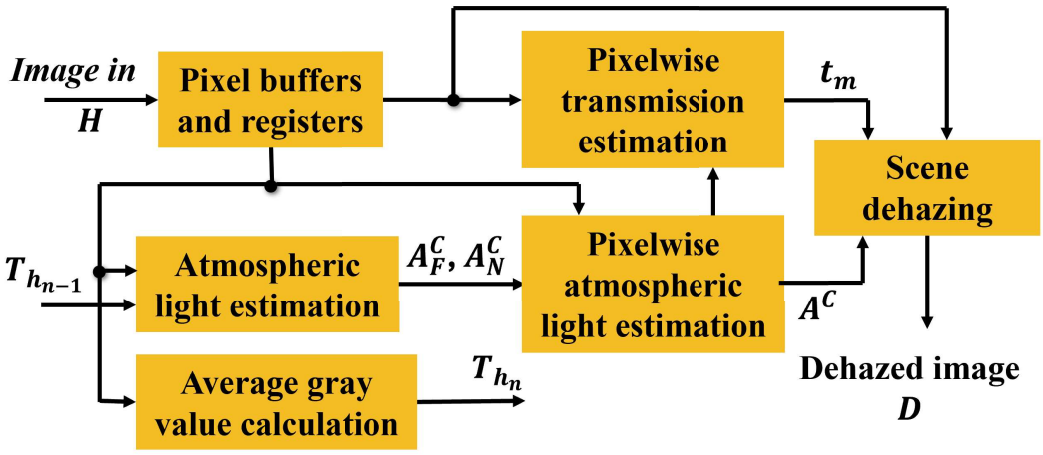


Figure 4.2: Block-level representation of the proposed method.

First, the dark channel of the downsampled hazy image is obtained using (4.1).

The local patch of size 5×5 slides in raster scan format in all three color channels. To find the minimum value of the patch, the minimum value of each column of the patch is stored in a column minimum register set (CMRS). The minimum of all CMRS values is the current patch's minimum value. The oldest column is replaced with a new column as the patch slides, while the other four remain the same. The minimum value of the newly added column is updated in CMRS in place of the minimum value of the deleted column. This operation and the method to determine a minimum of five values are depicted in Fig. 4.3. For an image of size w×h, where *w* represents the width of the image, and *h* represents the height of the image, four line buffers of size *w* are required to store the pixels in each R, G, and B channels, as shown in Fig. 4.3, if the patch size is 5×5. However, an image with a downsampling factor *r* requires line buffers of size *w/r* to store a complete row. Thus, in our design, twelve line buffers of size *w/3* are required, which is equivalent to four line buffers of size *w*.

The ALE unit estimates the dark channel pixel value for the current patch by finding the minimum among the R, G, and B patches. Since continuous frames in a video have a high correlation, the atmospheric light estimation unit utilizes the T_h value of the previous frame to divide the dark channel pixels into near and far groups. Further, the pixels with the highest intensity in dark channels of far and near groups are continuously updated in

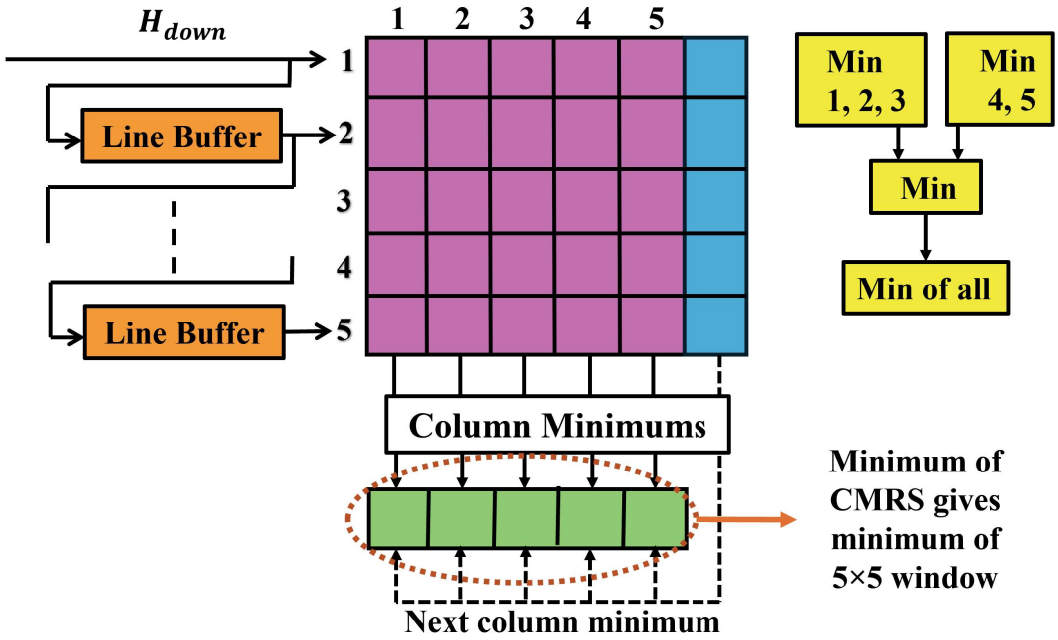


Figure 4.3: Method for calculating the minimum value of the sliding patch.

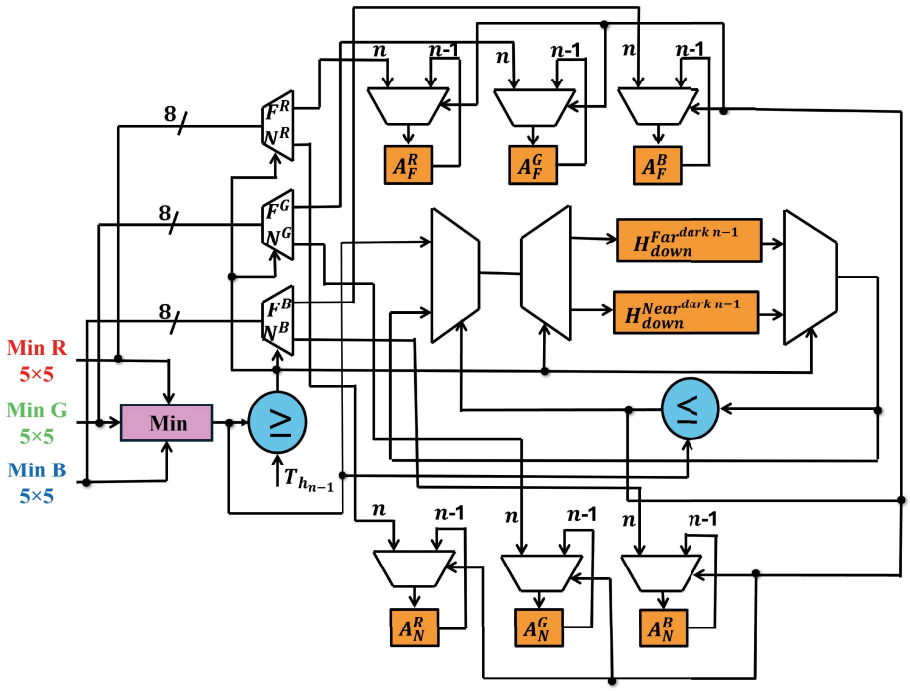


Figure 4.4: Architecture of atmospheric light estimation (ALE).

registers along with their corresponding R, G, and B values in H_{down} as shown in Fig. 4.4. Finally, after a complete scan, value of A_F and A_N are obtained in registers. Simultaneously, the center pixel value of each patch is sent to the AGVC unit consisting of an

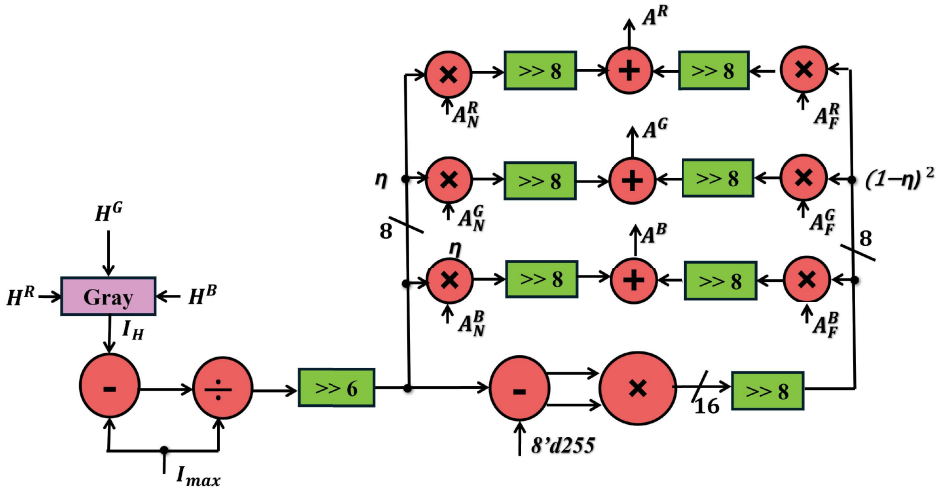


Figure 4.5: Architecture for pixel-wise atmospheric light adjustment (PALA) unit.

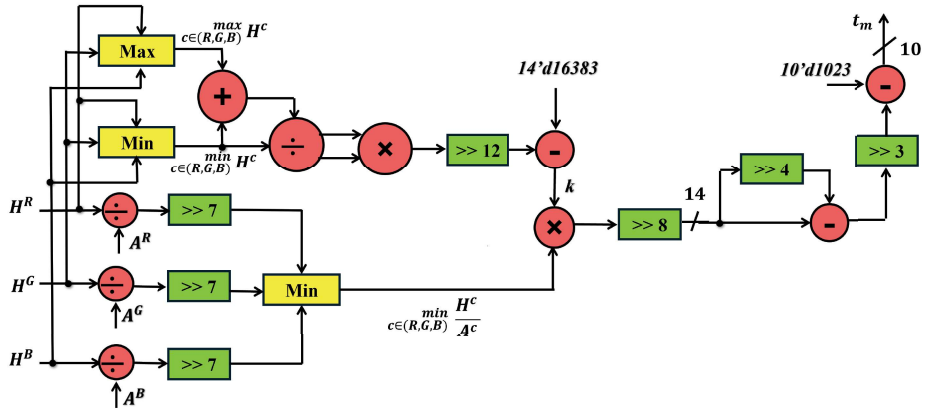


Figure 4.6: Architecture for pixel-wise transmission estimation (PTE) unit.

adder to find the sum of all the pixels in H_{down} which is divided by the number of pixels in H_{down} to obtain T_h of the current frame. Once A_F and A_N are estimated in the ALE unit, they are sent to the PALA unit for dynamic adjustment. The PALA unit, presented in Fig. 4.5, estimates the weighing factor η according to (4.8). The division operation is attained using a lookup table (LUT) based multiplier that holds the reciprocal values of the denominator scaled up by 2^{14}. The shifting operation in this unit is performed to scale down the values to match the bit widths. Depending on the gray value of the current pixel, an appropriate value of atmospheric light is passed to the PTE unit shown in Fig. 4.6. It utilizes the atmospheric light values obtained from the PALA unit and the current pixel value from the PB unit to estimate pixel transmission using (4.11) and (4.12). Complex dividers are replaced with LUT-based multipliers to reduce hardware complexity. The

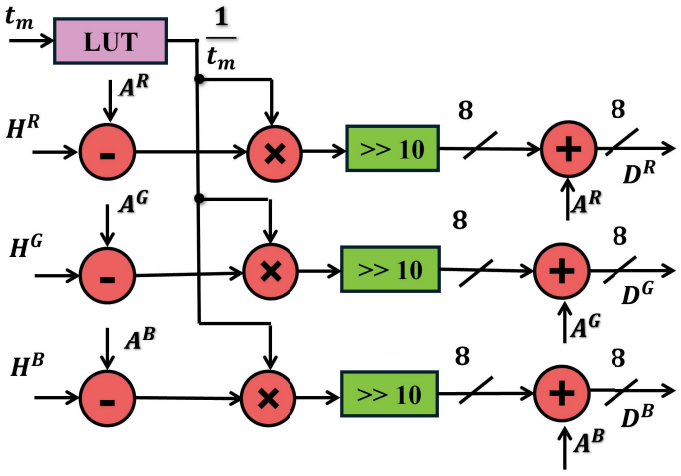
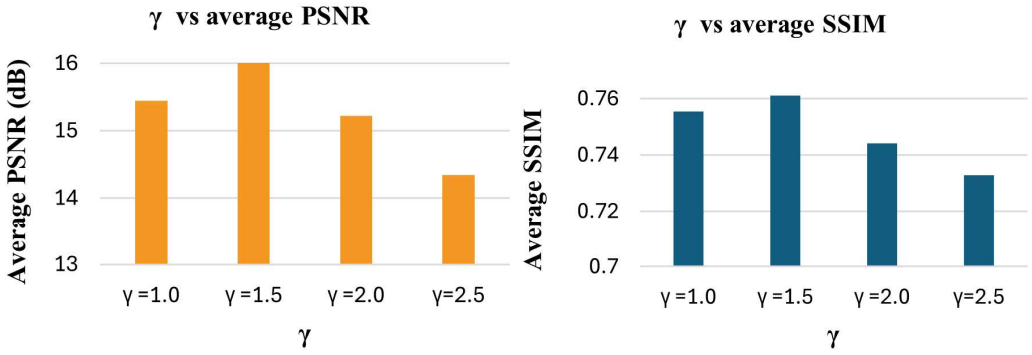


Figure 4.7: Architecture for scene dehazing (SD) unit.



(a) (b)

Figure 4.8: (a) Plot of γ vs average PSNR. (b) Plot of γ vs average SSIM.

values of γ and ω are chosen as 2 and 0.9375, respectively. Thus, the multiplication of ω is achieved using an adder and a shifter rather than a multiplier to reduce hardware costs.

Once t_m is obtained, it is sent to the SD unit, which performs dehazing based on (4.5). The PB unit, as depicted in Fig.4.2, supplies the pixel values of the hazy image to the SD unit. Simultaneously, the SD unit receives A from the PALA unit and reciprocal of t_m from LUT, generating the final dehazed picture as shown in Fig.4.7. Although this method works well for video with a high correlation between successive frames, flickering occurs if A of successive frames differ by three, as shown in [71]. To reduce flicker in such cases, A_F and A_N of the current frame is obtained by averaging A_F and A_N of current and previous frames. The algorithmic flow of the proposed method is given in Algorithm 1.

Algorithm 1 The proposed method

1: **Input:**Hazy image H
Output: Dehazed image D
2: Compute the dark channel of downsampled input hazy image H using (4.1).
3: From the obtained dark channel of H, compute $A_F^C = max\{H_{down}^{Far\ dark}(x)\}$ and $A_N^C = max\{H_{down}^{Near\ dark}(x)\}$ using T_H of the previous frame.
4: Find the difference between A_F^C and A_N^C of the previous and the current frame. If the difference is greater than three, use the corresponding average value of A_F^C and A_N^C of the previous and current frame; otherwise, proceed with the A_F^C and A_N^C value of the current frame.
5: Obtain the weighing factor η of the current pixel using (4.9).
6: Use η to find A^C of the current pixel using (4.8).
7: Obtain transmission of the current pixel using (4.11) and (4.12).
8: Obtain dehazed pixel value using (4.5).

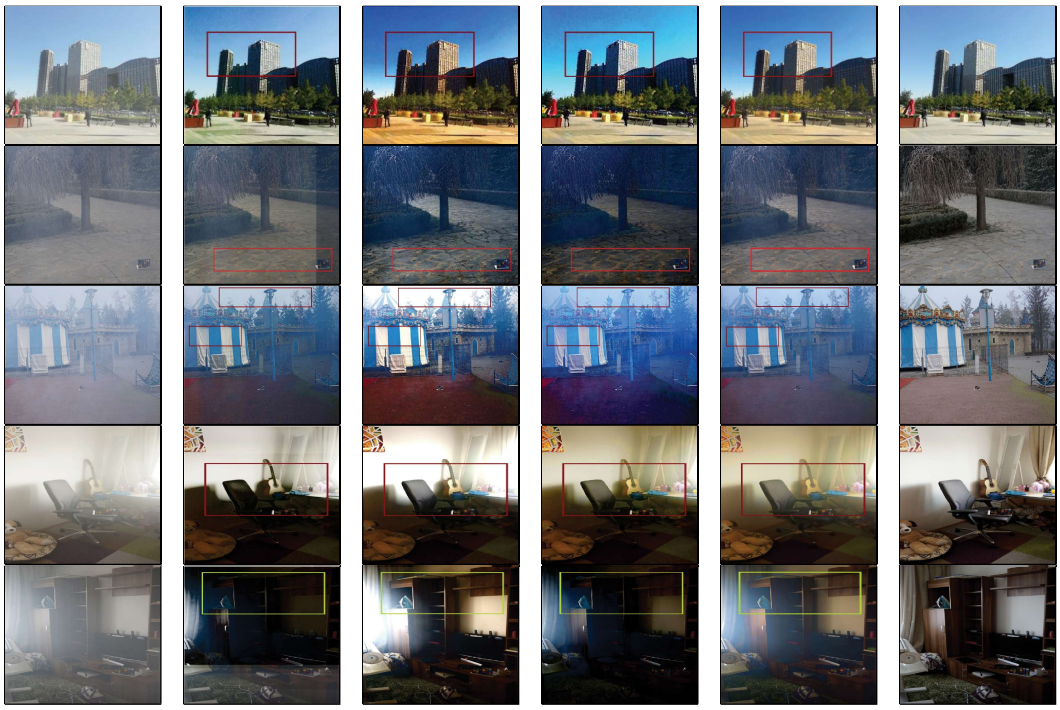
4.4 Performance Evaluation and Results Analysis

4.4.1 Ablation Study

The proposed algorithm was implemented in MATLAB, and an ablation study was performed to acquire an optimum value of γ used in (4.12). For this purpose, a subset of 200 images was created from the RESIDE [102], I-Haze [105], and O-HAZE [103] datasets. Then the average peak signal-to-noise ratio (PSNR) and the average structural similarity (SSIM) were obtained for various values of γ. The result of this ablation study is presented in Fig. 4.8. It depicts that the highest average PSNR and SSIM values are obtained for γ values around 1.5, but as seen in Fig. 4.1, the dehazing performance is better for higher values of γ. Thus,$\gamma = 2$ was chosen for which the quantitative performance is still high, and the dehazed images are not oversaturated. Moreover, it eases the hardware implementation of the PTE module. Henceforth, all the results are obtained using $\gamma = 2$.

4.4.2 Qualitative Analysis

The visual results exhibited by the proposed method are shown in Fig. 4.9 along with the results of [70], [71], and [72]. The dehazed images obtained using method [70] usually suffer from color distortions, as seen in the second row of Fig. 4.9. This occurs due to a discontinuity in the atmospheric light map. The images recovered using methods [71] and



(a) (b) (c) (d) (e) (f)

Figure 4.9: Dehazed results (a) Hazy image. (b) Results with [70]. (c) Results with [71]. (d) Results with [72]. (e) The proposed method. (f) Ground truth.

[72] are oversaturated compared to the ground-truth images because a smaller patch size is used in transmission estimation. Moreover, method [71] leads to overdehazing as seen in rows three, four, and five of Fig. 4.9. The results obtained using the proposed method are neither oversaturated nor overdehazed compared to their ground truth versions. This is achieved by proper adjustment of A and appropriate correction in the estimated t. Further, the proposed method was tested on real-world images from the RTTS dataset [102], and Fig. 4.10 shows the visual results obtained using the proposed method. From Fig. 4.10, it can be seen that the proposed method restores the visibility of the real-world hazy images significantly. To evaluate the video dehazing capability of the proposed method, video frames from the REVIDE dataset [106] were used, and the results are presented in Fig. 4.11. The proposed approach worked well for the test video's adjacent frames without significant flicker.

(a) (b) (c) (d)

Figure 4.10: Image dehazing results for real-world images (Top row - hazy images and bottom row - dehazed images).

(a) Frame 26 (b) Frame 27 (c) Frame 28 (d) Frame 29

Figure 4.11: Video dehazing results for different video frames (Top row - hazy image frames and bottom row - dehazed image frames).

4.4.3 Quantitative Analysis

The quantitative performance of the proposed technique is measured in terms of PSNR and SSIM metrics measured over the O-haze and the I-haze datasets. Table 4.1 presents the quantitative performance results of methods [70], [71], [72], [76] and the proposed method. A higher PSNR and SSIM value signifies better dehazing efficiency. For $\gamma = 1.5$, the proposed method scores the highest PSNR for the O-Haze dataset and the second highest for the I-Haze dataset. However, for $\gamma = 2$, the best results are obtained by [76] as it utilizes special filters for atmospheric and transmission map refinement. Yet, for $\gamma = 2$, the proposed design yields results comparable to [76] for PSNR values, and better results

Table 4.1: Comparison of PSNR (in dB) and SSIM metrics for different methods over the O-Haze and the I-Haze datasets.

	Metrics	Designs					
		[70]	[71]	[72]	[76]	The proposed $\gamma = 1.5$	The proposed $\gamma = 2.0$
O-HAZE	**PSNR** ↑	15.60	13.85	13.73	16.30	16.52	15.92
	SSIM ↑	0.42	0.45	0.33	0.76	0.46	0.45
I-HAZE	**PSNR** ↑	13.67	13.43	12.12	16.53	16.41	16.02
	SSIM ↑	0.52	0.55	0.38	0.81	0.62	0.60

Table 4.2: FPGA resource utilization.

Methods	Resources			
	LUT	FF	DSP	BRAM
[76]	32517	30480	136	332
The Proposed	1575	634	20	18

Table 4.3: ASIC implementation results.

Methods	CMOS Technology (nm)	Gate Count (k)	Frequency (MHz)	Throughput (Mpixels/s)	Power (mw)
[71]	130	23.7	200	200	13.7
[72]	130	18.6	200	200	NA
[73]	180	36.2	200[278]*	600[833]*	16.5[8.42]*
[74]	180	14.4	250[347]*	250[347]*	15.2[7.75]*
The proposed	90	13.8	470[325]*	460[325]*	4.15[8.66]*

* Normalized to 130 nm [73]

than the designs of [70], [71], and [72] for the I-Haze dataset and SSIM value over the O-Haze dataset.

The proposed VLSI architecture was designed using Verilog, and the AMD-Xilinx Vivado design suite was used to synthesize and implement it on ZynQ7 XC7Z020CLG484-1 FPGA. The RTL design of the proposed architecture was synthesized for ASIC implementation using 90-nm technology node in the Cadence Genus tool. A summary of the RTL synthesis results for FPGA and ASIC platforms are presented in Table 4.2 and Table 4.3,

Figure 4.12: Object detection results on hazy images and their corresponding dehazed images (Top row - hazy images and bottom row - dehazed images).

respectively. The results in Table 4.2 are computed for the O-Haze dataset like [76] for a fair comparison. Being computationally light, the proposed design utilizes only 4.84% of LUTs compared to [76] as the proposed method doesn't require special filters for atmospheric light and transmission refinement. ASIC implementation results of the proposed architecture given in Table 4.3 are also scaled to 130-nm for a fair comparison with [71], [72], [73], and [74]. From the data presented in Table 4.3, it can be observed that the proposed design requires only 58.2%, 74.2%, 38.1%, and 95.8% of gate count compared to [71], [72], [73], and [74], respectively. Edge detection and patch-based transmission estimation in [71] and [72] mainly increase their gate count. Further, Table 4.3 shows that the power dissipation of the proposed design is comparable to other methods at 130-nm technology. Moreover, the proposed design operates at 460 MHz. So, it is possible to process more than 50 frames of 3840×2160 resolution in a second, which is sufficient to dehaze images in real-time.

4.4.4 Suitability for Autonomous Vehicles

To assess the suitability of the proposed algorithm for implementation in autonomous vehicles, we performed object detection in the hazy image and the corresponding dehazed image using YOLOv4 [107]. The result of object detection is shown in Fig. 4.12. Compared to hazy images, more objects are detected in the corresponding dehazed images using the proposed method, even in dense haze conditions. This proves that the proposed dehazing method can be employed in autonomous vehicles for collision avoidance under significantly low visibility conditions.

4.5 Concluding Remarks

In this Chapter, a real-time video dehazing system and its hardware architecture are presented. The proposed video dehazing system uses an intensity-based dynamic approach to estimate atmospheric light and transmission for each pixel in the hazy image. The visual results show that the dehazed images obtained using the proposed method are neither overdehazed nor oversaturated. No post-processing techniques are required in the proposed method because pixel-to-pixel transmission estimation employed in the proposed method efficiently suppresses the halo artifacts around depth discontinuities. The proposed hardware architecture's optimal resource consumption makes it an appropriate choice for cost-effective real-time dehazing applications such as autonomous vehicles.