Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

Efficient Image Haze Removal Approach using Optimized Hardware Architecture Design using Parallel Processing

Harish Babu Gade[1]*, R Anirudh Reddy[2], P Ramakrishna[3], S Pothalaiah[4], Venkata Krishna Odugu[5]

[1]CVR College of Engineering, Senior Assistant Professor, ECE Department, Hyderabad -India
* Corresponding Author Email: harish.sidhu12@gmail.com - ORCID: 0000-0001-7496-5946
[2] B V Raju Institute of Technology, Associate Professor, ECE Department, Hyderabad -India
Email: anirudhreddy.r@bvrit.ac.in - ORCID: 0000-0001-8945-3986
[3]School of Engineering, Anurag University, Associate Professor, ECE Department, Hyderabad -India
Email: ramakrishnaece@anurag.edu.in - ORCID: 0000-0001-7496-5946
[4]Vignana Bharathi Institute of Technology, Professor, ECE Department, Hyderabad -India
Email: pothalaiahs@gmail.com- ORCID: - ORCID: 0000-0002-2545-993X
[5]CVR College of Engineering, Associate Professor, ECE Department, Hyderabad -India
Email: venkatakrishna.odugu@gmail.com - ORCID: 0000-0002-7580-8130

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract: The efficient hardware architecture for the image dehazing approach to remove haze in the photos is needed for many real-time applications that are automatic vehicle systems, CC cameras, multimedia applications. This work suggests the hardware architecture of dehazing which consists of two techniques: a low-complexity Simplified AirLight Estimating (SALE) approach and a Separate Transmission Map Estimation (STME) approach that can work on independently. First, SALE down-samples data, which essentially cuts down on data processing. Then, a Center-Point Compensation (CPC) method is created to make SALE AL predictions better. SALE doesn't use the sorting and filtering that is needed by image dehazing ways. By creating a preluded-AL estimate method, SALE improves the efficiency of hardware architecture data scheduling. STME makes transmission maps quickly and without needing AL data. When SMTE and SALE are processed at the same time, hardware performance goes up. When compared to other advanced studies, the PSNR, and SSIM values good for this work. For the Zynq7000 target FPGA, this suggested hardware design is made real. Based on the results of the experiment, this work uses less space and power.

Keywords: Haze removal, Dehazing, Transmission Map, Airlight Estimation, Zynq FPGA.

1. Introduction

This section provides a detailed explanation of the deteriorated model for image and haze development, the process of the DCP approach, and its separate phases. The DCP approach relies on statistical values of haze-free photographs. In the region-based photos, low-intensity pixels, referred to as black pixels, are present and do not include the sky color. These pixel colors may correspond to any color inside the RGB channel. The intensity of these black pixels is mostly influenced by the AL in the hazy image. Consequently, dark pixels are used to provide an accurate assessment of haze transmission. This haze model may be integrated with soft matting to provide high-quality photographs post-dehazing. This aspect is addressed in this study and is relevant to heavily hazed photos as well.

The creation of haze in an image is represented as a model, shown in Figure 1, mostly used for computer graphics and vision applications, and defined by Eq. (1)

$$\hat{I}(x) = J(x)\bar{t}(x) + \hat{A}\big(1 - \bar{t}(x)\big) \qquad (1)$$

where x shows the coordinates of the image, $\hat{I}$ represents the hazed and J is the haze-free images respectively.

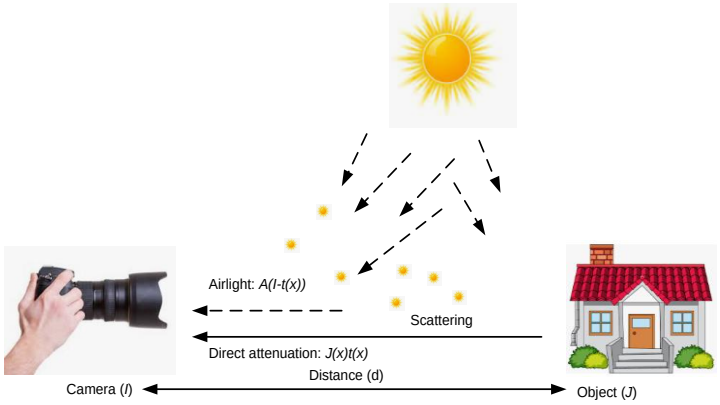


Fig. 1. Image degradation model.

The AirLight (AL) is denoted by $\hat{A}$ and $\bar{t}$ is the amount of light that is not scattered into the atmosphere and reaches the camera. This is called as transmission. The TM $\bar{t}$ can be represented by Eq. (2).

$$\bar{t}(x) = e^{-\beta d(x)} \qquad (2)$$

where the scattering coefficient is denoted by β and d is the depth of the scene. The main objective of the dehazing is the recovery of J, $\widehat{A}$ and $\bar{t}$ from the hazed image $\widehat{I}$. The atmospheric light $\hat{A}$ and TM $\bar{t}$ estimations are not simple because the $\bar{t}$ values are varied with respect to the depth of the scene. Hence, the estimation of the TM $\bar{t}$ is not simple from $\hat{I}$ without any prior assumptions. For clear atmospheric conditions, the β is assumed as zero then $I \approx J$, but, the β is not an ignorable value for hazed images. From Eq. (1), the term $J(x)\bar{t}(x)$ is decreased with the increasing of the scene depth and at the same time, the term $\hat{A}\big(1 - \bar{t}(x)\big)$ is the AL is increased with increasing scene depth. The haze removal method aim is the recovery of J from I after estimating the A and t. This can be expressed in Eq. (3).

$$J(x) = \frac{\hat{I}(x) - \hat{A}}{\bar{t}(x)} + \hat{A} \qquad (3)$$

2. Literature Survey

A unique AL estimated type haze removal process for high-quality remote sensing photos is presented in the study [1]. Here, a differentiable function trains a linear scene depth model's parameters. Second, the scene depth map estimates atmospheric light in hazy remote sensing images. A haze-lines model estimates the TM from the predicted atmospheric light. The estimated AL and TM is used to apply an atmospheric scattering model to remote sensing photographs to remove haze.

The existing hardware dehazing approaches failed assumptions cause artefacts in the recovered photos. To solve this challenge a new dehazing solution using CNN and dark channel prior (DCP) automatically learns crucial features and improves results presented in [2]. The device stores activations in few line buffers and avoids the need for off-chip memory like DRAM. The architecture

is well-suited for applications with limited computing resources, memory, and power budgets, as shown by FPGA and AISC implementations.

The saturation-based hardware method of image dehazing is detailed in [3]. A 15 x 15 window minimum filter accurately estimates AL from blurred down-sampled images. This approach is pixel-based due to saturation-based TME. Unlike patch-based techniques, the offered method suppresses edge halo artefacts without edge identification or image filtering. Seven pipelined stages make up the proposed dehazing system's VLSI architecture. It's implemented on FPGA and 65-nm ASIC platforms. Foggy images can be improved using colour attenuation prior, dark channel prior, or region detection network fog removal [4]. Attenuation and air-light are caused by atmospheric particles like water droplets that absorb and scatter light. This research uses Dark Channel Prior (DCP). DCP must find the TM, which shows image fog intensity. Estimating the dark channel, discovering the TM, improving it, and producing the image without haze are the main steps of this technique. Raspberry pi was used to implement the algorithm. This research employs filters to improve the reconstructed de-hazed image.

A single module design for real-time Retinex-based picture improvement and haze removal is presented in in work [5]. The single-module architecture supporting Retinex-based picture enhancement algorithms by efficiently utilizing their similarities presented. The implementation findings show that the work need only 1% logic circuit overhead. Instead of implementing them individually using different modules, adopting a single module reduces computing complexity and processing especially in consumer electronics. Instead of soft matting for TM, picture enhancement is used, reducing computing complexity that would hinder high-frame-rate real-time processing.

The work [6] provides a hardware-implementable rapid fog removal approach for single images. Here, the estimation of AL using pixel-based median channel of hazy image is presented. High-quality image recovery with lower processing complexity than patch-based dark channel is possible. With median channel prior, FPGA implementation enables real-time dehazing.

A less hardware complexity picture dehazing method is proposed in the work [7]. It uses Depth-Refinement Transmission-Rate Estimation (DTE) and Distribute Airlight Estimation (DAE). DTE generates an accurate depth map to estimate TM by calibrating white-object depth. When estimating AL, DAE accounts for non-uniform haze density by averaging distant illumination sources according to depth. With DTE and DAE, line buffers can be reduced by 33% without sacrificing image dehazing quality.

Image processing is limited by image quality, which hinders data extraction. Traditional image processing software like MATLAB was used. Although the software can extract data from low-quality images, it still has a time-consuming difficulty. Hardware is used in image processing to avoid the issue since efficient image processing produces fast results. Thus, researchers are increasingly interested in hardware-based image-enhancing methods like FPGA. The study [8] analyses 25 research publications from 2016 to 2021 on FPGA hardware implementation in image processing. Many features are analyzed in this work. Linear grey single-channel transmission using a fusional removing fog algorithm streamlines the process [9]. This approach utilizes varying mixes of high-boost filtering and grey single-channel linear transform. A single grayscale channel was created from the RGB channel to enhance defogging visibility without sacrificing results. Fusion should result in linear

transmission of greyscale data. To address the increasing demand for clear images, we also present a reliable real-time FPGA defogging solution.

To reduce time and space, the design optimizes directed filtering data pathways, which speeds up defogging. The shift register in the box filter is divided and placed on the input port for collaborating, conserving storage space, because the mean and square value computations follow the same pixel reading sequence. In addition, employing LUTs instead of the multiplier reduces square value computation module delays and boosts efficiency. A Zynq-based defogging algorithm for low-light photos is proposed in [10]. Many Retinex methods are investigated, analyzed, and compared to find the best one.

The efficient hardware architectures and image dehazing techniques for efficient haze removal of single images are described in works [11-17] using different simplified techniques.

In this work, the key contributions are given as:

- The image haze removal approach uses a SALE Approach and an independent STME.
- SALE avoids photo dehazing's sorting and computational filters. The preluded-AL estimate approach developed by SALE improves hardware architecture data scheduling.
- STME generates transmission maps effectively without AL data. Hardware throughput increases when SMTE and SALE are processed together.
- Compared to advanced studies, this study performs well in PSNR, and SSIM,
- The Zynq7000 target FPGA implements this hardware design. This study has higher area and energy efficiency, according to experiments.

3. Proposed Hardware Architecture

The main steps in the image dehazing process are: AL Estimation (ALE), TM Estimation (TME), and scene recovery. If an input hazy image is processed by these three steps in sequentially as shown in Figure 2, then the time delay is very high, and the throughput of the image process is less to process all image pixels. In the process of image dehazing, many arithmetic operations, shifting and sorting operations are needed. Hence, the delay is more, and throughput is less. To overcome this problem, in this work the parallel processing of ALE and TM Estimation (TME) are carried out as shown in Figure 3. An imbalance in computation between them is another and calculation gap is mainly caused by the extensive sorting operations in AL estimate. The TME is halted for few cycles while ALE is completed, resulting in reduced hardware consumption.

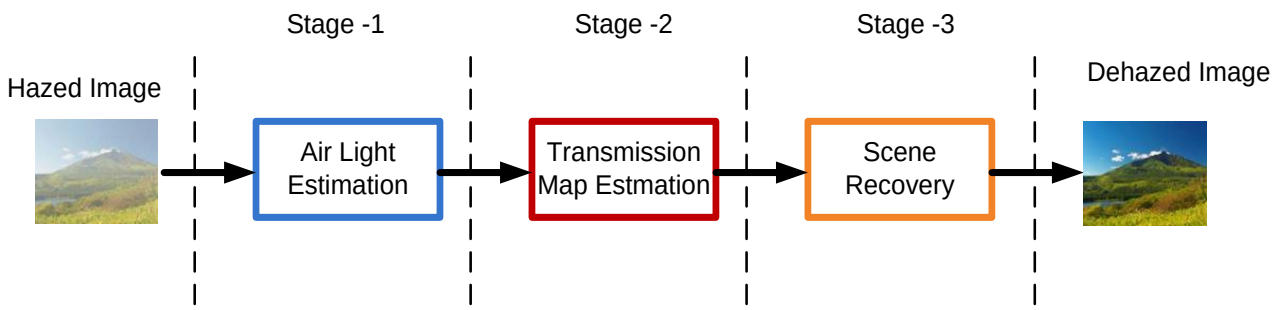


Fig. 2. General dehazing process

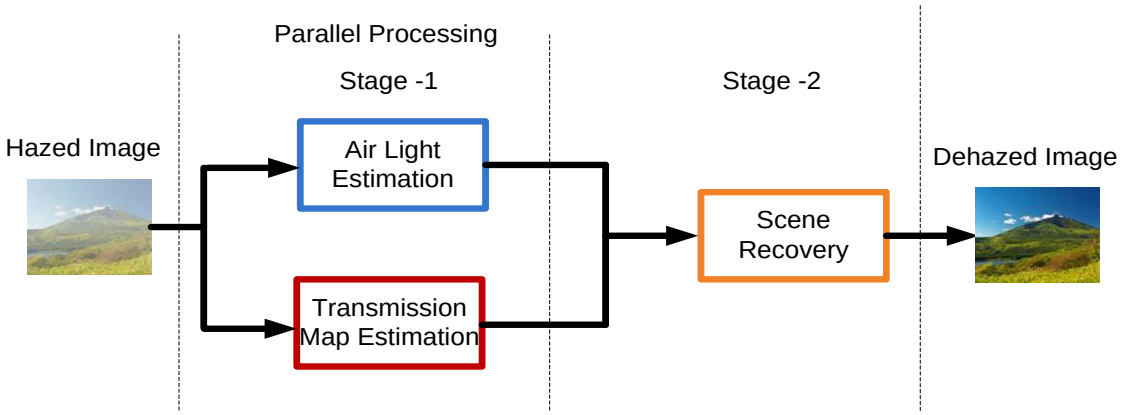


Fig. 3. Parallel processing in dehazing process

Their hardware architectures face a barrier in calculating the mismatch between *t* AL and TM estimations. To overcome this problem, in the hardware architecture, a simplified AirLight Estimation (SALE) and Separate Transmission Map Estimation (STME) which is interdependent type along with Scene Recovery Module (SRM) considered as shown in Figure 4.

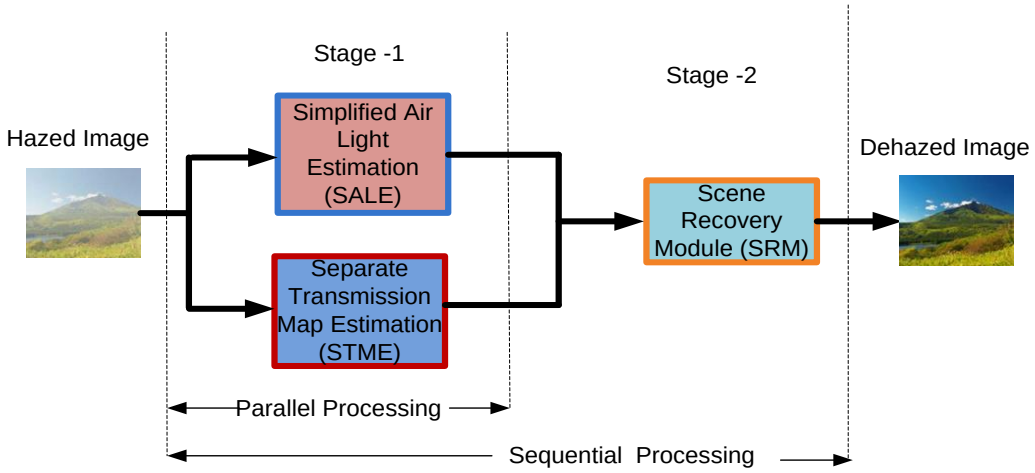


Fig. 4. Modified basic processes of image dehazing

This section describes the modules of the hardware-structure of efficient image dehazing method. The architecture is divided into four components: SALE, STME, SRM, and the Controller module. Because of the Down Sampling Ratio (DSR), SALE processes faster than STME and SRM. Thus, STME and SRM have three levels of parallelism, while SALE has only one. The controller is a Finite State Machine (FSM)-based block serves two basic functions, as described below. First, it organizes the signals required to swap data paths between SALE, STME, and SRM. Second, it ensures that SALE, STME, and SRM can effectively communicate with one another.

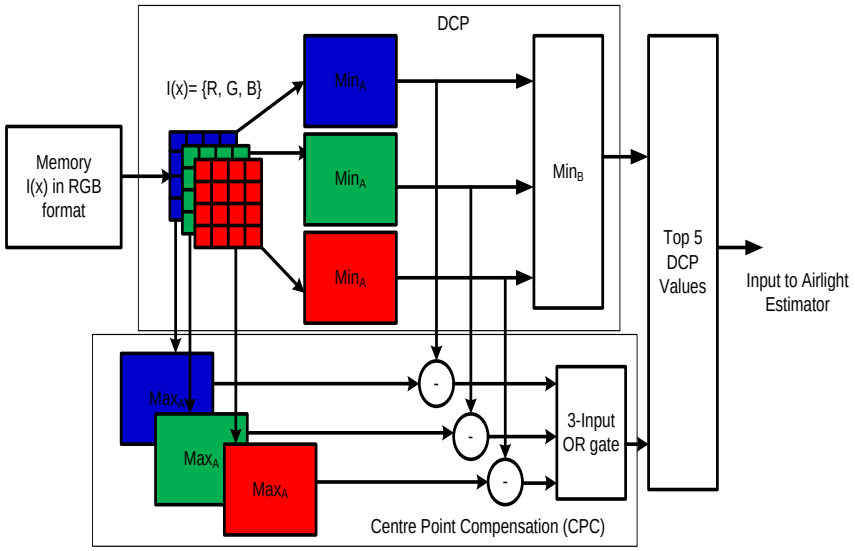


Fig. 5. Generation of top 5 DCP under SALE along with DCP and CPC

DCP, CPC, and the AL estimator are all part of the SALE module as shown in Figure 5. DCP retrieves the down-sampled $I_s(x)$ from external memory. Then it retrieves $I^{DS}(x)$ from $I_s(x)$. Min_A chooses three distinct minimums for every RGB channel. The minimum of them is then picked up by Min_B as $I^{DS}(x)$, according to (4).

$$I^{DS}(x) = \min_{y \in \Omega(x)} \left\{ \min_{c \in \{R,G,B\}} I_s^c(y) \right\} \qquad (4)$$

CPC (5) determines the coordinates of the four adjacent pixels that should be added to E.

$$\left\{ \left[\min_{y1 \in \Omega`(x)} I^c(y1) - \min_{y2 \in \Omega`(x)} I^c(y2) \right] > JND^c \right\} |_{c \in \{R,G,B\}} \quad (5)$$

The maximum operator in Equation (5) is realized by Max_A. Just like Min_A, the lowest operator in Eq. (5) finds the specific minimal on each RGB channel. Therefore, in Eq. (5) for CPC, Min_A is shared alongside the minimal operator. Then, using three subtractors, we can get the RGB channel's minimum and maximum differences. Equation (5) is met if one of the three sign bits is logic-1.

This is easily accomplished with a three-input OR gate. ALE employs Top-5 DCP collector to estimate the top five brightest DCPs from E, as illustrated in Figure 5.

The condition checker of Figure 6 of the SALE module uses external memory to retrieve the $I(x)$ within five 11 × 11 windows, with centers at C1-C5. The $I(x)$ samples with RGB ≥200 are collected into $\Omega_p(x)$. As a result, SALE is produced by Eq. (6), where it is usual to use a divider since the denominator L is not constant. Having said that, there are more gates since the divider logic is more complicated.

$$A_{LAE}^{c} = \frac{\sum_{y \in \Omega_p(x)} I^c(y)}{L} \qquad (6)$$
$$c \in \{R,G,B\}$$

To address this issue, progressive-accumulating is recommended for A_{PRE} and A_{LAE} in Figure 6. This technique is divided into two components: ACC_A/AVG_A for A_{PRE} and ACC_B/AVG_B for SALE. A_{PRE} is made up of an accumulator (ACC_A) and a 1-bit logic-right shifter (AVG_A). Data scheduling of ACC_A/AVG_A for A_{PRE}.

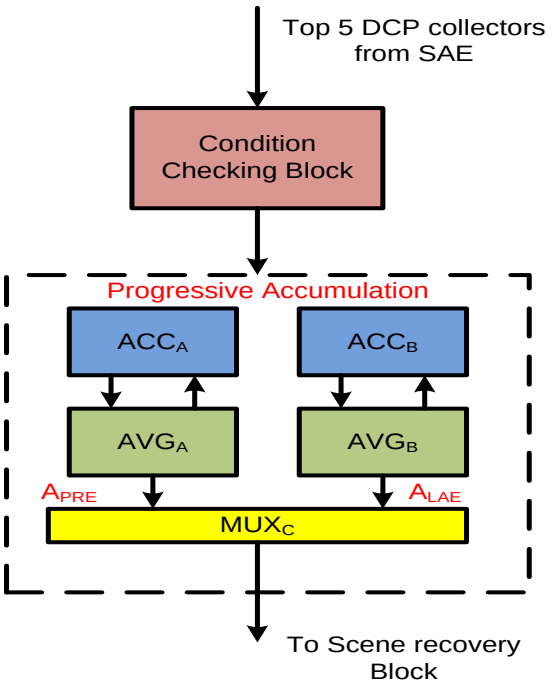


Fig. 6. Simplified Airlight Estimation (SALE) module

STME consists of three components: RGB2SV, Depth refinement, and TME. RGB2SV as shown in Figure 7 converts RGB to Saturation *S(x)* and Value *V(x)* using Eqs. (7) and (8), respectively. Min_C and Max_B implement the min. and max. operators in Eqs. (7) and (8), respectively.

$$S(x) = \begin{cases} 0; max_{c\in\{R,G,B\}} I^c(x) = 0 \\ 1 - \frac{min_{c\in\{R,G,B\}} I^c(x)}{max_{c\in\{R,G,B\}} I^c(x)}; \ otherwise \end{cases} \quad (7)$$

$$V(x) = \max_{c\in\{R,G,B\}} I^c(x) \quad (8)$$

V(x) is obtained immediately from Max_B, which is likewise the result of the maximal operator in Eq. (7). As a result, Max_B is shared with *S(x)*. The reciprocal of the maximal operator in Eq. (7) is converted to a look-up table, LUT_A (256 × 9). The results of LUT_A and Min_C are sent to the multiplier. Its product is subtracted from one, and the subtraction result is sent to MUX_A. MUX_A assigns the subtraction result for Max_B = 0 or 0 for Max_B = 0 to *S(x)*. *S(x)* and *V(x)* are provided to the distance estimator, which generates *d(x)*, using Eq. (9). After that, it goes to Depth Refinement (DR), where the minimum filter reduces it to d(x), as shown in Eq. (10).

$$d(x) = \theta_0 + \theta_1 V(x) + \theta_2 S(x) \quad (9)$$

$$d`(x) = \min_{y\in\Omega_r(x)} d(y) \quad (10)$$

With the help of a buffer, Min_D is able to implement this minimum operator. Then, the Line Buffer (LB) stores the information of the filter masks, and d(x) is sent on to the sharp LPF and Sharp Filter (SF). Figure 8 shows how the DR module returns d(x), which is progressively loaded into the LB. All of the window data for the LPF and Sf filters is stored in the LB. It can share the data and window

sizes of both filters. Following this method, SALE and STME data streaming will go smoothly, allowing for their concurrent processing. There are a total of nine FFs in the filter masks; five of these are shared by the SF. The SF and LPF share these nine flip-flops, which are called the Data-reused buffer. The output of the SF is subsequently subtracted from JND^{Gray}'s. Whether *d(x)* or the LPF output is sent to the TME via MUX_B depends on the sign-bit of the subtraction result.

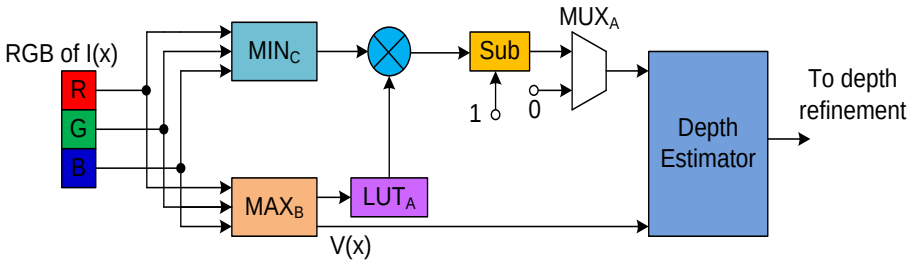


Fig. 7. RGB to SV converter

The TME converts t(x) to SRM in accordance with Eq. (2). In Eq. (2), the exponential term is converted to a LUT, i.e LUT_B (256 x 9). The exponential term in t(x) represents a complicated mathematical calculation, as mentioned before in Eq. (2), which is an aspect of *d(x)*. The most effective tool for this task would be a look-up table. You can use either floating-point or fixed-point notation to express *t(x)* and *d(x)*. A lot more computing power is needed to process data in floating-point format compared to fixed-point format because of all the extra hardware resources needed. The study does not take the floating-format into account, and this design approach significantly reduces hardware throughput.

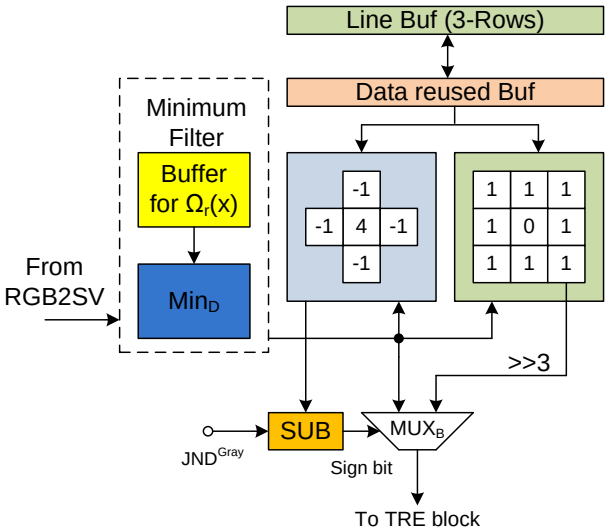


Fig. 8. The Depth Refinement Module

SRM as shown in Figure 9 is responsible for restoring $J(x) = \{J^R(x), J^G(x), J^B(x)\}$. It obtains the $I(x) = \{I^R(x), I^G(x), I^B(x)\}$, ALE from SLAE, and t(x) from STME. Eq. (22) consists of an adder, subtractor, and multiplier to *1/max{t(x), t0}*. LUT_C (256×9) transforms *t(x)* into *1/max{t(x), t0}*. If LB_2 data scheduling is used, the AL buffer is active to store AL estimation results from SALE. If LB_1 is used, the AL buffer can be omitted.

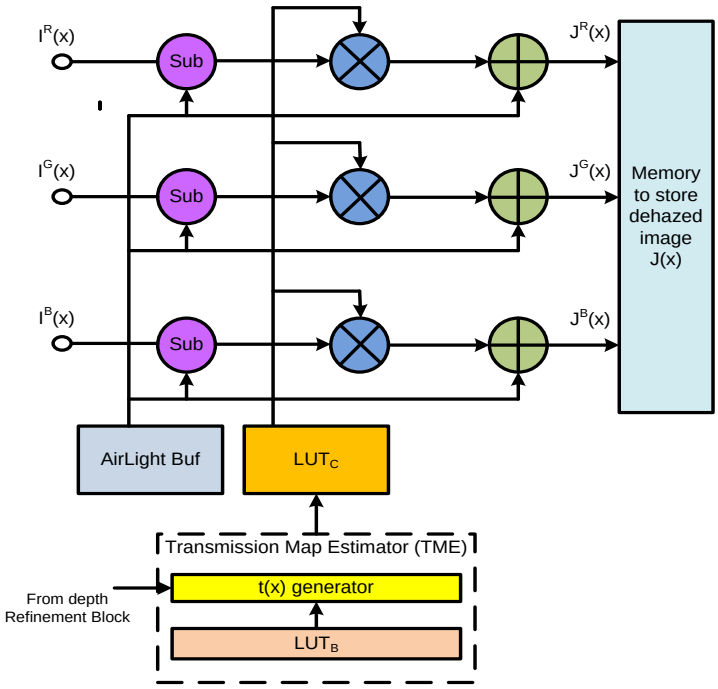


Fig. 9. Scene Recovery block with 3- level parallelism

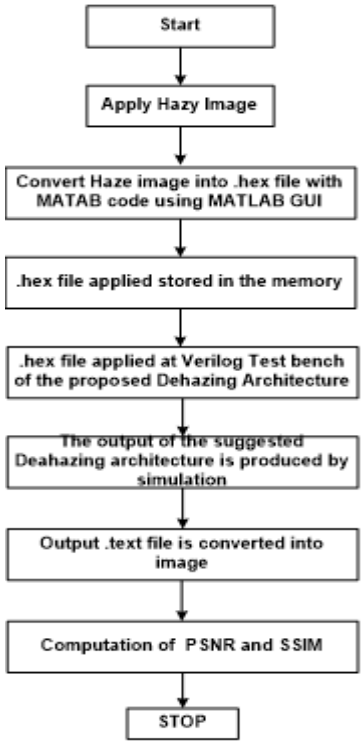
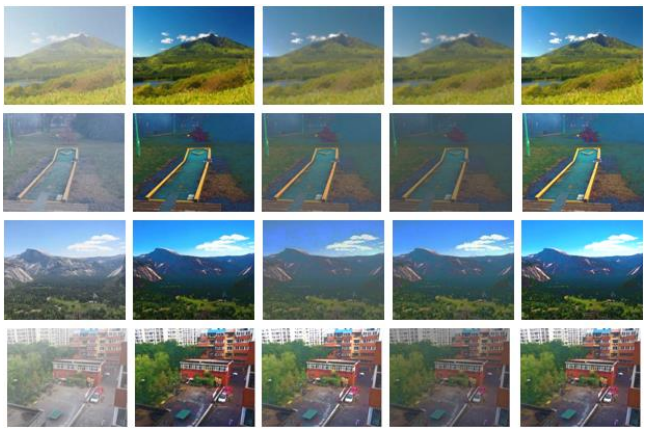


Fig. 10. Flow diagram of the image dehazing setup work

4. Results

This hardware architecture is realized using Verilog HDL and simulated by the Xilinx Vivado tool for the Zynq7 series FPGA family with the device number XC7Z020CLG484-1. The sample dataset

images are considered and applied to the proposed hardware architecture in the simulation environment. The flow diagram of the entire image-defogging process is shown in Figure 10. In Figure 10, the image conversion, input feeding, and processing by the suggested model steps are described.



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

Fig. 11. (a) Hazed (b) Ground Truth (c) Zhang et al. [11] (d) Lee et al. [12] (e) proposed work images

The performance metrics related to the output image that is dehazed image are analyzed by PSNR and SSIM values. The PSNR and SSIM values are compared to the existing works in Table 1. Next, the hardware-wise parameters are compared in Table 2.

Table 1. Image quality metrics comparison

	PSNR			SSIM		
	Zhang et al. [11]	**Lee et al. [12]**	**Prop. Work**	**Zhang et al. [11]**	**Lee et al. [12]**	**Prop. work**
Im-1	**20.5**	**19.89**	**21.9**	**0.84**	**0.92**	**0.97**
Im-2	**19.6**	**20.2**	**22.0**	**0.91**	**0.89**	**0.92**
Im-3	**20.8**	**20.9**	**21.8**	**0.90**	**0.89**	**0.91**
Im-4	**21.3**	**21.1**	**21.5**	**0.91**	**0.88**	**0.94**
Avg.	**20.55**	**20.52**	**21.8**	**0.89**	**0.895**	**0.942**

The avg. of the PSNR values is calculated for the four images and the avg. PSNR of the current method is improved by 6.08%, and 6.23% than Zhang et al. [11] and Lee et al. [12], respectively. The average SSIM also improved by 5.84% and 5.25% than the structures of Zhang et al. [11] and Lee et al. [12], respectively.

Table 2. Comparison of hardware architecture parameters

	Zhang et al. [11]	**Lee et al. [12]**	**Proposed work**
FPGA LEs	**1.63k**	**1.56k**	**1.32k**
Frequency	**116**	**99**	**100**
Throughput	**116**	**99**	**100**
Power (mW)	**22.8**	**16.5**	**10.59**

It can be observed that from Table 2, the number of logic elements decreased by 19% and 17.6% for the current architecture of the image dehazing than the studies of Zhang et al. [11] and Lee et al. [12],

respectively. The power of the suggested work is decreased by 53.55% and 35.8% than the structures of Zhang et al. [11] and Lee et al. [12].

5. Conclusions

The architecture for a hardware-efficient image dehazing technique is proposed in this work. The image haze removal method uses two techniques: a simplified and low-complexity AirLight Estimating (SALE) Approach and an independently working Separate Transmission Map Estimation (STME). First, SALE effectively reduces data processing by down-sampling. A Center-Point Compensation approach (CPC) is then devised to improve SALE AL estimates. SALE avoids picture dehazing methods' sorting and computationally demanding filters. SALE enhances hardware architecture data scheduling efficiency by developing a preluded-airlight estimate technique. STME efficiently creates transmission maps without AL data dependency. The simultaneous processing of SMTE and SALE increases hardware throughput. This work shows competitive performance in PSNR, and SSIM, compared to other advanced investigations. This suggested hardware architecture is realized for the target FPGA of Zynq7000 The implementation results show that this study has lower area and power.

References

[1] Zhu, Z., Luo, Y., Wei, H., Li, Y., Qi, G., Mazur, N., ... & Li, P. (2021). Atmospheric light estimation based remote sensing image dehazing. *Remote Sensing*, *13*(13), 2432. https://doi.org/10.3390/rs13132432.

[2] Kumar, R., Kaushik, B. K., Raman, B., & Sharma, G. (2021). A hybrid dehazing method and its hardware implementation for image sensors. *IEEE Sensors Journal*, *21*(22), 25931-25940. DOI: 10.1109/JSEN.2021.3118376

[3] Upadhyay, B. B., & Sarawadekar, K. (2023). VLSI design of saturation-based image dehazing algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *31*(7), 959-968. DOI: 10.1109/TVLSI.2023.3272018

[4] More, V. N., & Vyas, V. (2022). Removal of fog from hazy images and their restoration. *Journal of King Saud University-Engineering Sciences*. https://doi.org/10.1016/j.jksues.2022.01.002

[5] Kasauka, D., Sugiyama, K., Tsutsui, H., Okuhata, H., & Miyanaga, Y. (2019). An architecture for real-time retinex-based image enhancement and haze removal and its FPGA implementation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *102*(6), 775-782. https://doi.org/10.1587/transfun.E102.A.775

[6] Kang, H. J., Kim, Y. H., & Lee, Y. H. (2015). FPGA implementation for enhancing image using pixel-based median channel prior. *International Journal of Multimedia and Ubiquitous Engineering*, *10*(9), 147-154. DOI:10.14257/ijmue.2015.10.9.16

[7] Lee, Y. H., & Tang, S. J. (2020). A design of image dehazing engine using DTE and DAE techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(7), 2880-2895. DOI:10.1109/TCSVT.2020.3034250

[8] Azhari, Z. I., Setumin, S., Rosli, A. D., & Bakar, S. J. A. (2023). A systematic literature review on hardware implementation of image processing. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, *12*(1),19-28. DOI: http://doi.org/10.11591/ijres.v12.i1.pp19-28

[9] Du, G., Wu, J., Cao, H., Xing, K., Li, Z., Zhang, D., & Wang, X. (2021). A real-time effective fusion-based image defogging architecture on FPGA. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *17*(3), 1-21. https://doi.org/10.1145/3446241

[10] Zhang, C., Bi, S., Jiang, T., Wang, J., & Mao, W. (2020, December). Implementation of ZYNQ for image defogging. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (Vol. 9, pp. 1971-1977). IEEE. DOI:10.32620/reks.2023.2.08

[11] Zhang, B., & Zhao, J. (2016). Hardware implementation for real-time haze removal. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *25*(3),1188-1192. https://doi.org/10.1109/TVLSI.2016.2622404

[12] Lee, Y. H., & Wu, B. H. (2018). Algorithm and architecture design of a hardware-efficient image dehazing engine. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(7), 2146-2161. https://doi.org/10.1007/s11554-023-01356-x

[13] Babu, G. H., & Venkatram, N. (2020). A survey on analysis and implementation of state-of-the-art haze removal techniques. *Journal of Visual Communication and Image Representation*, *72*, 102912. https://doi.org/10.1016/j.jvcir.2020.102912

[14] Babu, H., & Venkatram, N. (2022). An efficient image dahazing using Googlenet based convolution neural networks. *Multimedia Tools and Applications*, *81*(30), 43897-43917. https://doi.org/10.1007/s11042-022-13222-2

[15] Babu, G. H., & Venkatram, N. (2023). ABF de-hazing algorithm based on deep learning CNN for single I-Haze detection. *Advances in engineering software*, *175*, 103341. DOI:10.1016/j.advengsoft.2022.103341

[16] Babu, G. H., Odugu, V. K., Venkatram, N., Satish, B., Revathi, K., & Rao, B. J. (2023). Development and performance evaluation of enhanced image dehazing method using deep learning networks. *Journal of Visual Communication and Image Representation*, *97*, 103976. DOI:10.1016/j.jvcir.2023.103976

[17] Gade, H. B., Odugu, V. K., N, V., & K, R. (2024). Efficient single image-based dehazing technique using convolutional neural networks. *Multimedia Tools and Applications*, 1-23. DOI:10.1007/s11042-024-18784-x