SOME INVESTIGATIONS ON DEVELOPMENT OF AN EMBEDDED FRAMEWORK FOR REAL-TIME IMAGE/VIDEO DE-HAZE APPLICATIONS

Submitted in partial fulfillment of the requirements
for the award of the degree of
DOCTOR OF PHILOSOPHY

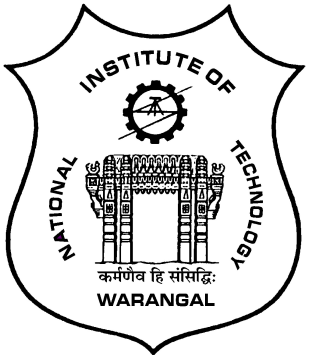
by

PRATHAP SOMA

(Roll No: 717133)

Supervisor:

Dr. RAVI KUMAR JATOTH
Associate Professor



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA STATE-506004, INDIA

2021

Dedicated to

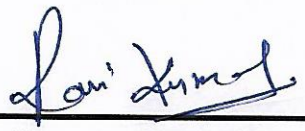
My family & Teachers

Approval Sheet

This thesis entitled **"Some Investigations on development of an embedded framework for real-time image/video de-haze applications"** by **Prathap Soma** is approved for the degree of Doctor of Philosophy.

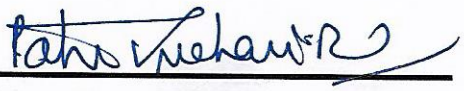
Examiners

Supervisor(s)



Dr. J. Ravi Kumar
Associate Professor

Chairman



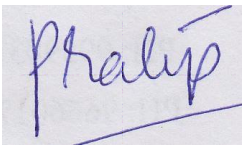
Dr. Patri Sreehari Rao
Associate Professor and Head

Date: 13th Jan 2022

DECLARATION

This is to certify that the work presented in the thesis entitled "**Some Investigations on development of an embedded framework for real-time image/video de-haze applications**" is a bonafide work done by me under the supervision of **Dr. J. Ravi Kumar**, Associate Professor, Department of Electronics and Communication Engineering, National Institute of Technology Warangal, India and was not submitted elsewhere for the award of any degree.

I declare that in my own words, this written submission reflects my ideas and where the ideas or words of others have been used, I have cited and referenced the original sources sufficiently. I also declare that in my submission, I have adhered to all academic honesty and integrity standards and have not misrepresented or fabricated or falsified any idea/data/truth/source. I recognize that any infringement of the above would be a cause for the Institute's disciplinary action and may also invoke punitive action from sources that have not been properly cited or from which if necessary, proper permission has not been obtained.



Prathap Soma

(Roll No: 717133)

Date:01-07-2021

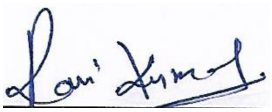
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

TELANGANA STATE-506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled "**Some Investigations on development of an embedded framework for real-time image/video de-haze applications**", which is being submitted by **Mr. Prathap Soma (Roll No: 717133)**, in partial fulfillment for the award of the degree of Doctor of Philosophy to the Department of Electronics and Communication Engineering of National Institute of Technology Warangal, is a record of bonafide research work carried out by him under my supervision and has not been submitted elsewhere for any degree.



Dr. J. Ravi Kumar
(Research Supervisor)
Associate Professor
Department of E.C.E.
N.I.T. Warangal
Warangal - 506004, India

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor **Dr. J. Ravi Kumar** for the continuous support of my Ph.D. study and related research, for his patience, motivation, and guidance with his moral values. My sincere thanks to him for providing me an opportunity to join the institute as a Ph.D. research scholar and giving me access to the research facilities. I find words inadequate to thank him for enabling me to complete this work in spite of all obstacles. The thesis would not have seen the light of the day without his insistent support and cooperation. Also, I think that you deserve an extra note of thanks for all of your ongoing leadership and support. Finally, I just wanted to thank you for being such a great advisor in my personal life too. I have enjoyed working with you so much over the past few years.

My special words of thanks should also go to **Prof. L. Anjaneyelu**, Head of the department, Electronics and Communication Engineering Department, NIT Warangal for his valuable suggestions and support that he shared during my research tenure.

Here, I also would like to take this privilege to thankful my DSC members, Dr. Patri Sreehari Rao, Department of Electronics and Communication Engineering, NIT Warangal, Dr. S. Anuradha, Department of Electronics and Communication Engineering, NIT Warangal and Dr. U.S.N. Raju, Assistant Professor, Department of Computer Science Engineering, NIT Warangal for their continuous support, suggestions and advice during my research period whenever required.

I also appreciate the encouragement from teaching staff, non-teaching members and fraternity of Dept. of E.C.E. in N.I.T. Warangal. They have always been encouraging and supportive towards my research.

I thank my fellow labmates (Dr. Hathiram Nenavath, Dr. Jailsingh Bhookya, Mr. Kishor Ingle, Mr. A. B. Ahadit, Mr. M. Vijaya Kumar) for the stimulating discussions, we were working together before deadlines, and for all the fun we have had in the last few years.

It is my pleasure to show my indebtedness to my co-scholars at NITW like Mr. B. Roshan, Dr. R. Shashank, Dr. V. Santhosh Kumar, Dr. M. A. Mushahhid Majeed, Mr. T. Sunil Kumar and friends for their help during the course of this work. Their timely help and friendship shall always be remembered during my stay.

I owe my deepest gratitude towards my better half (**Sukanya Soma**) for her continuous support, love and understanding of my goals and aspirations. Her patience and sacrifice will remain my inspiration throughout my life. Without her help, I would not have been able to complete much of what I have done and become who I am. Also, I love my daughter (**Maansa Samyuktha**) for giving me happiness and support during the complete research. I appreciate my daughter for abiding my ignorance and the patience she showed during my research. I consider myself the luckiest in the world to have such a lovely and caring family, standing beside me with their love and unconditional support.

My heartfelt special thanks goes to my family members (parents **Padma-Yakaiah(Late)**, brothers family (Narender-Sinduja) for supporting me spiritually throughout my life in general. Their unconditional love, support, self-sacrifices and patience has always been the source of my strength. I shall be grateful forever because of their kindness and love.

Moreover, I find no words inadequate to express any form of acknowledgment to my sister's family **Mamatha-Venugopal & Rajeshwari-Premchand** for their love, constant inspiration, encouragement, financial help and moral support in any situation, I shall be grateful forever to them.

Last but not least, I want to mention my sincere thanks to **SERB, New Delhi, India** for their financial support, which has supported me so much during this research.

Finally, I thank God, for filling me every day with new hopes, strength, purpose and faith.

ABSTRACT

Haze is a phenomenon of absorption or scattering of particles in the atmosphere. It influences the attenuation of the radiance along the path towards the camera. Subsequently, captured images suffer from less contrast and quality limiting the visibility of the distant scenes. As it changes the colors of the original scene and reduces the contrast becomes an annoying problem to photographers, it declines the visibility of the pictures and acts as a threat to several applications like object detection, outdoor surveillance; it also declines the clarity of the underwater images and satellite images. The removal of haze from images is vital and widely required in the computer vision field. However, haze is dependent on an unknown depth; hence haze removal is still a challenging problem. If multiple images are used for de-hazing, different types of image capturing devices are needed to shoot multiple images in the same location with different weather conditions, which not only has a high economic cost but also difficult in the application for a dynamic scene. The capturing of multiple images of the same scene in different weather conditions is infeasible for outdoor vision systems with high real-time demand; whereas, the problem is ill-posed if the input is only a single hazy image. Due to recent developments in computer vision, it is now possible to de-haze the image and increase its contrast. With a single image as input, many techniques have been proposed. These techniques are physically sound based on theories from the methodology. Unfortunately, so costly in terms of computational complexity hence they are not suitable for real-time applications.

This thesis focuses on developing single image dehazing algorithms with low computational complexity and allows to eliminate the artefacts & halos in the final dehazed image. The developed algorithms are based on a strong and statistically based approach called dark channel prior. Experimental simulations are performed with a variety of standard image databases. The validity of the algorithms is analyzed quantitatively with five metrics namely Average Contrast of the Output Image, Mean Square Error, Peak Signal to Noise Ratio, Percentage of Haze Improvement and Structural Similarity Index (SSIM) are compared to the available state-of-the-art methods. Computation complexity is one of the important metrics and is taken into consideration in the evaluation of the algorithm performances. Based on the comparative analysis, the developed algorithms have been mapped on different hardware platforms such as DSP TMS320C6748, Zynq 702, Zynq 706, Raspberry pi3 and Jetsom Nano for real-time image dehazing applications used in computer vision. Its validity and effectiveness are evaluated with image/video materials. These results have proven that real-time image dehazing is possible for new computer vision applications like high resolution, outdoor surveillance, onboard vehicle camera applications and many more.

Contents

List of Figures

List of Tables

List of Abbreviations

1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional
ACOI	Average Contrast of Output Image
AI	Artificial Intelligence
ATC	Average Time Cost
ASIC	Application Specific Integrated Circuit
BRAM	Blocked Random Access Memory
CCD	Charge Coupled Device
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DCP	Dark Channel Prior
DOP	Direction of Polarization
DSP	Digital Signal Processing
EOF	End of Frame
ET	Execution Time
FF	Flip-Flop
FMRF	Factorial Markov Random Field
FPGA	Field Programmable Gate Array
GANs	Generative Adversarial Networks
GPIO	General Purpose Input/Output
HDL	Hardware Descriptive Language
HLS	High Level Synthesis
HSL	Hue, Saturation, and Light
ICA	Independent Component Analysis
ILA	Integrated Logic Analyzer
IP	Intellectual Property

LSB	Least Significant Bit
LUT	Look Up Table
MATLAB	Matrix Laboratory
MSE	Mean Square Error
PHI	Percentage of Haze Improvement
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
RTL	Register Transfer Level
SSIM	Structural Similarity Index
SoC	System on Chip
VIO	Virtual Input/ Output

Chapter 1

Introduction

Digital image processing [1] is a subcategory of Digital Signal Processing that deals with digital images. An image is a two-dimensional array represented as I(x, y) where x, y represent the position of the element. The value at coordinate (x, y) is called intensity, sometimes referred to as a pixel, picture element, or image element. Image processing techniques have been widely used in many applications wherever the subjective quality of images is essential for human perception.

During the process of capturing the real-world scene as a digital image through an image sensor, different types of noises are superimposed on the images. The most commonly superimposed noise models are additive [2] and multiplicative [3].

Additive noise model: An image sensor consists of many detectors, the most obvious example being a CCD array which is a two-dimensional array of detectors, one per pixel of the detected image. If individual detectors do not have an identical response, then the fixed pattern detector response will be combined with the detected image. If this fixed pattern is purely additive, then the detected image is just:

$$f(i, j) = s(i, j) + b(i, j) \tag{1.1}$$

where s(i, j) is the true image and b(i, j) the fixed pattern noise. This noise model follows the probability distribution functions such as Gaussian, Laplacian, Poisson, and uniform distribution functions, etc. The removal of this type of noise can be done using linear filtering techniques [4] . But, de-noising via linear filters normally does not give satisfactory results

since both noise and edges contain high-frequency components. Therefore, some nonlinear filtering techniques have been proposed. They are weighted Median [5], Wavelet-based image filters [6], Laplace [7], Sobel [8] , and so on.

Multiplicative noise model: It is commonly found in many real-world signal processing applications. Unlike additive noise, this kind of noise is much more difficult to remove from the corrupted signal, mainly because of its multiplicative nature. Let us consider an image as an example, when noise corrupts the bright area of the image, the corresponding pixels are multiplied with high-intensity values, thus its random variation will increase, or be "magnified". On the other hand, if the noise is multiplied to the darker area, the change of random variation can be less significant. Therefore, the noise variation greatly depends on the intensity levels of the image pixels being corrupted. If the noise is multiplicative, the detected image can be represented as

$$f(i, j) = s(i, j) * b(i, j) \tag{1.2}$$

where s(i, j) is the true image and b(i, j) the fixed pattern noise. To develop effective approaches for removing multiplicative noise, various filtering techniques [9] have been proposed by assuming statistical characteristics of the noise are available. Many researchers have used wavelet-based [10, 11, 12] approaches for de-noising problem. To effectively remove such type of noise, modeling of noise formation is needed. But, it is not easy to establish an appropriate statistical model for noise by simply examining the corrupted image. Hence, the computational complexity of removing multiplicative noise is more.

Image de-hazing aims to remove haze from a hazy image, so as to restore the original image. However, noise, edge, and texture are high frequency components; it is difficult to distinguish them in the process of de-hazing and the de-hazed images could inevitably lose some details. Overall, recovering meaningful information from hazy images in the process of haze removal to obtain high quality images is an important problem nowadays.

Haze is also a type of noise that consists of both additive and multiplicative models. When the images captured in hazy weather conditions are seriously degraded by scattering of atmospheric particles, contrast, color, and object features are difficult to identify by human vision. To remove the haze from images, modeling of haze formation of the image is needed. It is explained in section 1.1 briefly.

Image De-hazing is one of the applications in the field of image/video processing. The light reflected from an outdoor object gets scattered even before reaching the camera in almost

all the outdoor scenes. The reason for this effect is particles that are present in the atmosphere. They include mist, dust, and fumes and the particles are present in the way of propagation and deflect the light. In scenes where the haze is present for long distances, the atmospheric particles affect adversely due to which the contrasts get reduced and the colors present on the surface become faint. Hence, the photographs are degraded and they lose visual appeal. They provide bad visibility of the contents present in the scene and also decrease the quality of photography.

The same effect can also be seen in satellite imaging [13] and it is important to know as it has many applications including land-use planning [14], archeology, web mapping, and environmental studies. In a hazy environment, the light should have traveled in straight paths but the light that is scattered previously replaces this light. That scattered light is known as air-light. The contrast of the resultant image undergoes multiplicative and additive losses because of the presence of air-light. The resultant image looks noisy and called is hazy image.

The development of image de-hazing methods has been beneficial to many real-world applications, including driver assistance systems [15, 16] outdoor video surveillance [17, 18, 19], analysis of remote sensing imagery [20, 21, 22, 23], and video-assisted transportation [24, 25]. These techniques can also be transferred to underwater image enhancement [26, 27, 28] and images acquired in rain or snow [29, 30]. The broad prospects for applicability have attracted much attention from researchers in the field of computer vision and image processing in recent years.

There are three major methods available to remove haze from images. One is image enhancement method, the second is image fusion-based method and the last one is image restoration method based on the physical model.

The image enhancement method is one which improves the visual quality of the hazy image by enhancing the contrast and highlights the image details. The important thing about this method is that it doesn't care about the quality of the final de-hazed image and may cause certain information losses. Examples of image enhancement methods include the Retinex algorithm[31, 32, 33, 34], wavelet techniques [35, 36, 37, 38, 39], and histogram equalization[40, 41, 42, 43, 44, 45]. Nevertheless, the techniques are widely employed because of their elegance.

The image fusion-based method [46, 47, 48, 49, 50] enhances the useful and required information from multiple source channels to eventually produce a high-quality restored image. This method doesn't require any physical model. It means the method doesn't estimate the atmospheric light and transmission map or depth map. Sometimes post-processing techniques are also not required. Since there is no depth map estimation it is difficult to obtain good quality de-hazed image and several halos and artifacts may be present in the final recovered image.

The analysis of image de-hazing based on image restoration includes studying the process of degradation of haze images using a physical model. Once the physical model is established, the unknown parameters need to be estimated accurately by taking some assumptions so a de-hazed image can be achieved. The image restoration methods are further divided into two; One is multiple image or polarization restoration method and another is single image restoration method. In the polarization method, the haze can be removed by considering many hazy images taken from dissimilar degrees of polarization and at different weather conditions. In this case, more than one image is needed to recover the de-hazed image. Narasimhan and Nayar [51] proposed a theory on multiple-image restoration method. The calculations required in this method are more and, the quality of the de-hazed image depends on multiple input images. Hence, many researches in haze removal are based on single-image restoration.

In Section 1.1 the model for image restoration is described which can be normally used in determining the formation of the image in a hazy environment.

1.1 Haze image formation model

Fig. 1.1 represents the haze formation model of an image. When light passes through some scattering medium, it gets attenuated along the original course and hence gets distributed to all directions. The process can be commonly modeled mathematically by assuming that for short distances, linear relation exists between distance which is traveled and a fraction of the light deflected. More formally, along the infinitesimally shorter distances dr, light fraction absorbed can be taken by βdr where β is the coefficient of medium extinction caused by light scattering. Integrating the above process along a line ray which is emerging from the viewer, in the case of β which is varying especially, gives:

$$T(x) = e^{\left(-\int \beta(r(s))ds\right)} \tag{1.3}$$

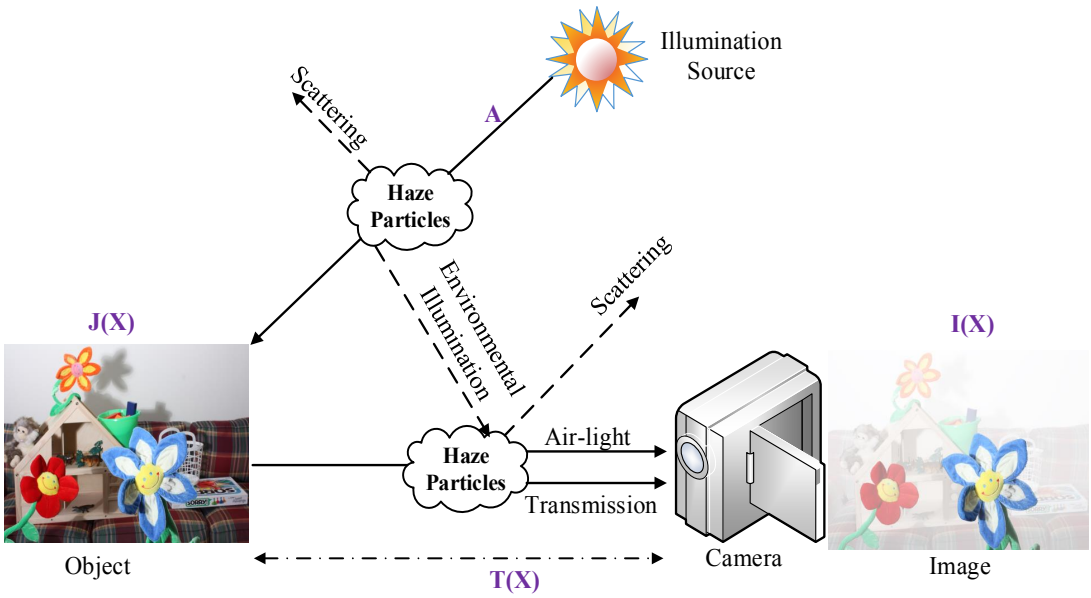


Figure 1.1: Haze image formation model.

where r stands for arc-length parameterization of the ray. $T(x)$ which is a fraction called transmission and it denotes the relative amount of light that survived the entire length of the path between the observer and the surface point present in the scene, at $r(d)$, without getting scattered. If there is no blackbody radiation is present, the light scattering process conserves energy, which means that the fraction of light that is scattered from a particular direction gets replaced by the same amount of light fraction which is scattered from other directions. This equation expresses this conservation law called Radiative Transport Equation. Assuming that the above-added light gets dominated by the light which underwent many scattering events, it allows us for approximating it as both uniform and isotropic in space. The constant light presented above-called air light or veiling light is used for approximating the actual in-scattering term in the full equation for radiative transport to obtain the following image formation model [52]:

$$I(x) = J(x)\,T(x) + A\,(1 - T(x)) \tag{1.4}$$

Equation 1.4 describes three color channels of an image, where $I(x)$ is observed image, A is the air-light, $J(x)$ is the surface or scene radiance at a point of intersection of the scene, and real-world line ray corresponds to pixel and lastly $T(x)$ is transmission media along the path. In 1.4, the product term $J(x)T(x)$ is known as direct attenuation, the second term $A[1 - T(x)]$ is an additive component and termed air-light. Three unknown components are present in the above equation. The goal of this work is to recover an image showing the scene $J(x)$ through a clear haze-free medium. Other parameters affecting emission of radiation are not eliminated.

The recovery of the radiance map is achieved by extraction of raw data $I(x)$ from the camera as

$$J(x) = A + \left(\frac{I(x) - A}{max\left(T(x), t_0\right)} \right) \tag{1.5}$$

The above equation 1.5 represents the recovery of the de-haze image from the hazy image, where t_0 is the control parameter. The scene radiance is usually not as bright as an atmospheric light, and the image after haze removal looks dim. So, we increase the exposure of $J(x)$ for display by taking t_0. The typical value of t_0 is considered 0.1.

1.2 History

In 1998, based on Mie scattering law [53], Oakley et al.[54] began research on images taken in bad weather conditions. He assumed that the depth information of the haze image was known and achieved scattering attenuation compensation by estimating the pixel's weights of the scattered and reflected image. It is a model-based method and obtained satisfactory de-hazing results. In those days his approach became a research hotspot in the field of image processing. But, the limitation of this method was that it was suitable only for gray images. Later, Tan et al. [55, 56] improved this method by studying the relationship between image contrast quality and wavelength and obtained the results of the degraded image restoration method to color images. Thereafter, Robinson et al. [57] constructed a dynamic and real-time weather system, which was based on the atmospheric scattering model. According to atmospheric scattering theory, the scattering of air particles is divided into two parts: first is the attenuation of reflected light from the object surface to the camera, and the second is the scattering of air-light reaching the camera. Based on this model, Hautière et al. [58] estimated the visibility distance using side geographical information. They computed the scene depth by modeling the depth value of each point as a Euclidean distance function and used the 3D geographical model to remove the haze. Kopf et al. [59] introduced a deep photosystem that uses the existing digital terrain to provide basic information. A three-dimensional model of the scene was built, and then the depth information values, as well as the structure of the color image and texture, were obtained.

In the last decade, some researchers have made significant progress in understanding the degradation mechanism and foggy image modeling using atmospheric scattering theory, and have suggested image processing methods to improve image clarity. At last, we can say that the physical model based methods can be used for de-hazing effectively.

This method is based on the assumption that the depth of the scene is known and the quality of the recovered image is good. However, the real-time applicability of this method is seriously limited due to the hardware requirements and need for accurate scene depth information.

1.3 Literature Survey

This chapter is devoted to the review of research work reported in literature in the area of image dehazing and its embedded based design. The various methods adopted for enhancing the quality of the haze images and its hardware implementation issues are reviewed. The essential necessity of embedded system design in the field of image processing and its advantages over general purpose processors are also reviewed. The consolidated findings and results of various researchers for different methods and techniques are discussed.

The literature review was carried out in three parts. In the first part, review of single image restoration based de-hazing techniques used for various applications was done. The importance of a physical model for enhancing the quality of the haze image was also reviewed. In the second part, traditional/existing algorithms used for image de-hazing were reviewed. The advantages and disadvantages of this algorithm were also noted. The literature related to the implementation of image de-hazing algorithms on various hardware platforms such as FPGA, Digital Signal Processors, ASIC, were included in the third part.

1.4 Single image restoration method

Based on the studies, single image restoration based de-hazing [60] works well in areas where the quality of acquired images is decreased by the influence of the surrounding medium. While this medium is usually the atmosphere and the factors affecting images are caused by weather, digital images are captured in other environments as well especially in water. The aim of image restoration-based de-hazing method is to investigate the causes of image degradation and analyze the imaging process, after which the scene is recovered using inverse transformation [61]. There are two generalized models available to analyze haze degradation, namely, degradation model and physical model based on atmospheric scattering.

1.4.1 Degradation model

According to the degradation model [62] the haze formation model can be expressed as

$$f'(x) = h'(g(x)) \tag{1.6}$$

where h' is the restored function, f' is the restored image and g(x) is degraded image by haze particles in the atmosphere it can be expressed as

$$g(x) = f(x) * h(x) + n(x) \tag{1.7}$$

where f(x) is the original scene, h(x) is the degraded function and n(x) is the noise function. Based on the process of restored and degraded function the de-hazed image can be recovered easily. This model has some drawbacks since it ignores how the hazy image's contrast is degraded through noise function and degraded function. It is not able to express the complex factors of weather conditions. As a result, the above model fails to produce satisfactory results for hazy images.

1.4.2 Physical model

In 1998, based on Mie atmospheric scattering law[53], Oakley et al. [54] started research work on images taken in bad weather conditions.

According to Mie atmospheric scattering law, the scattering of atmospheric particles is classified into two parts: one is due to the attenuation of reflected light from the object surface to the camera, and the other is the scattering of air-light reaching to the camera. Therefore, McCartney [63] proposed that the imaging mechanism in bad weather should be described by a light attenuation model and an air-light imaging model, which forms a theoretical basis of a hazy image with characteristics of blur and low contrast that can be used to understand the degradation mechanism of foggy images, thus enabling degraded images to be restored. This model is explained briefly in Section 1.1.

Based on this physical model, plenty number of de-hazing methods have been proposed, many of them have achieved satisfactory results. Several representative methods based on a physical model will now be introduced in the following section.

1.5 Single Image De-hazing With knowing Additional (Depth) Information

Tan et al.[55, 56] improved Oakley et al.'s [54] method by studying the relationship between image contrast quality and wavelength, and extending the degraded image restoration method to color images. Thereafter, Robinson et al. [57] constructed a dynamic and real-time weather system, which was based on the atmospheric scattering model. According to atmospheric scattering theory, the scattering of air particles is divided into two parts: first is due to the attenuation of reflected light from the object surface to the camera, and the second is due to the scattering of air-light reaching the camera. Based on this model, Hautière et al. [58] estimated the visibility distance using side geographical information they computed of the scene depth by modeling the depth value of each point as a Euclidean distance function and used 3D geographical model to remove the haze. Kopf et al. [59] introduced a deep photosystem that uses existing digital terrain to provide basic information. Firstly, a three-dimensional model of the scene was built, and then the depth information values, as well as the structure of the color image and texture, were obtained.

This method is based on the assumption that the depth of the scene is known and that the quality of the recovered image is good. However, the real-time applicability of this method is seriously limited due to hardware requirements and the need for accurate scene depth information.

1.6 Multi-image Dehazing Methods

The estimation of transmission map can also be done by using two or more images of the same scene. The consideration of multiple images can be divided into two: Same scene under different degree of polarization (Polarization Technique) and another is same scene under different weather conditions.

1.6.1 Polarization based technique

Schechner et al. [52, 64] investigated the formation of foggy image mechanism based on atmospheric scattering to demonstrate the physical concept of the polarization effect. Initially, by changing the direction of the polarizer, more images were collected; then, the atmospheric

polarization coefficient, the contrast and accurate colors of the image were estimated. After that, the depth of the scene was obtained and then the de-hazed image was recovered. In this process, the depth information and atmospheric light value can also be estimated. But, this method needs some primary information about the sky and so has some drawbacks in terms of implementation. Later, Shwartz et al. [65] overcame this limitation by proposing a blind classification method. Based on his studies, the sky information can be ignored by assuming that if there is no correlation between the airlight component and the transmission component in some parts of the scene, an independent component analysis (ICA) approach can be used to restore the airlight component and other relevant information in order to improve the image's visibility and colour and achieve the goal of de-hazing. In addition, the authors processed the noise that was introduced during the de-hazing process. They successfully used some regularization method and adaptive weights related to the scene depth to remove noise. Based on angle-of-polarization (AOP) distribution analysis, the authors proposed a polarimetric de-hazing [66] approach to improve the contrast and range of visibility of images. [67] provided a new polarization hazy imaging model that considers the joint polarization effects of air-light and object radiance. It is an effective method for synthesizing the optimal polarized-difference (PD) of the image. Treibitz and Schechner [68] experimented with different angles of polarized filter. These filters were quantitatively evaluated based on their signal-to-noise ratio (SNR) to estimate the dehazing results. A quality assessment method suitable for polarization analysis images in hazy conditions is proposed in [69] . Reference [70] proposed a method to estimate the haze parameters from the polarization information of two known objects at different distances, and the estimated parameters were used to de-haze image.

These methods were greatly dependent on the direction of polarization (DOP) of sky light. They can enhance contrast of the image under thin fog and dense fog. The de-hazing effect may be greatly reduced because of inaccuracies in information.Furthermore, finding the maximum and minimum degrees of polarization under the same scene during rapid scene changes is difficult, and the process is complicated, so this type of image restoration method is not possible in real-time applications.

1.6.2 Different weather conditions

Narasimhan and Nayar [71, 72] extensively researched the estimation of depth information of a scene from different viewpoints and different weather conditions. They observed that the intensity and contrast of the image depend primarily on two parameters. They are atmospheric light and the scattering of atmospheric particles. According to this method, initially the color

changes of the degraded images are calculated, and following this the atmospheric scattering models of the different degraded images are combined to estimate the depth information. Finally, a 3-D image structure is obtained called de-hazed image. Later, Sun et al. [73] enhanced the above method using gradient field. In his method, the depth information of the scene is estimated using the partial derivative of the atmospheric light and Poisson equation for realizing the de-hazed image. Chen et al. [74] took a sample of a hazy image and a corresponding clear image of the same scene to forming the optical model. The depth ratio with corresponding points was estimated and then the restored image was obtained using optical model.

In addition to the work of Nayar and Narasimhan [71, 72], Wu and Dai [75] used segmentation step to verify the changes in the scene, such as planes moving across the field of view. As a result, this method could account for ambiguous areas.

These de-hazing techniques are simple and can accomplish good results. But, two or more different images of the same scene in different weather conditions are required and difficult to obtain. Hence, the use of these methods for de-hazing for real-time monitoring situations is not suitable.

1.6.3 Single Image Dehazing Method With Prior Knowledge

Single image de-hazing is still an under-researched problem. In recent years, based on prior knowledge of the haze image formation, some de-hazing algorithms have been proposed. Using two prior conditions, Tan et al. proposed effective de-hazing method. One is that the contrast of the image is higher in clear image than hazy image. Second is that the attenuation of the hazy image is a continuous function of distance and is smooth. In this method, firstly, the color of light is determined, and then each color channel's brightness is separated. At last, the air-light of the image is converted to white. This method can obtain de-hazing with only one image. But, serious "halo" effects may easily appear due to sudden changes of depth, leading to over saturation of the color when the haze is more. Based on the same assumptions, Ancuti et al. [76] suggested another de-hazing method which is optimized to preserve the color distribution and local contrast. It is suitable for tackling image de-hazing based on local feature points. However, over-enhancement effect still appeared in the final de-hazed image. Fattal [77] assumed that the surface shading and the scene transmission are locally uncorrelated to estimate the thickness of the haze. They used independent component analysis (ICA) and Markov random field to estimate the scene. This method is physically sound and produces good results when there is sufficient prior color information of the image. However, it produces inaccurate outcome when

there is heavy haze and may fail in cases where the original assumptions are invalid. Kratz and Nishino [78] assumed that a hazy image was composed of two layers: scene albedo and scene depth. They achieved relative depth information using Factorial Markov Random Field (FMRF). Even so, the color in the obtained image using this method was over-saturated.

He et al. [79] proposed an impressive solution based on the dark channel prior (DCP) of hazy images that can effectively overcome the drawbacks of the above algorithms to some extent. The statics of the outdoor haze-free images are the basis for this algorithm. The statics shows that in an image, most of the local regions include dark pixels in at-least one of the RGB color channels. The dark pixels are defined as low-intensity pixels i.e., its intensity is close to zero. The DCP algorithm is an important breakthrough in the field of single image de-hazing. DCP provides a new concept for researchers, but refinement of transmission map requires high computations. However, the performance of this method fails in the sky region. In Addition to that, when the image contains large bright areas such as sky, water or white objects, the dark channel prior assumptions become invalid. As it is the best method, most de-hazing algorithms are developed in some particular aspect (especially at sky regions) by improving DCP (Dark Channel Prior) method. Many improvements were later done to refine the coarse transmission map based on DCP, such as WLS edge-preserving smoothing [80] , bilateral filtering [81], joint bilateral filtering [82], a joint trilateral filter [83], guided image filtering [84], weighted guided image filtering [85], content adaptive guided image filtering [86], smooth filtering [87], anisotropic diffusion [88], window adaptive method [89], edge-preserving and mean filters [90], adaptively subdivided quadtree [91], edge-guided interpolated filter [92], an adaptive Wiener filter [93], guided trigonometric bilateral filters [94], median filter and gamma correction [95], Laplacian-based gamma correction [96], fuzzy theory and weighted estimation [97], and a fusion strategy [98], to refine the transmission image. Moreover, the computational complexity of the methods is more. It is difficult to implement them on hardware. Hence, for real-time applications, de-hazing is still a challenging task.

1.7 Review of Hardware implementation

Digital image/video processing algorithms are difficult to implement on a processor due to factors like huge amount of data present in an image, and the complex operations that need to be performed on the image. A single image of size 256×256 can be considered to perform a single operation (3x3 convolution/masking) and requires about 0.2 million computations (ad-

dition/subtraction, multiplication, padding, and shifting) without considering the overhead of loading and recovering pixel values.

In recent years, a new class of computing hardware has been gaining research interest. Configurable computing hardware has some of the advantages of both general purpose computing processor and application specific hardware. Reconfigurable hardware devices in the form of Field Programmable Gate Arrays (FPGAs) [99], DSP Processors [100], Raspberry Pi [101], Jetson Nano [102] and so on [103] have been proposed as viable system building blocks in the construction of high performance systems at an economical price. This type of hardware consists of a large number of functional units. The functionality of the hardware is determined by how interconnections between functional units are configured and in some cases how the functional units themselves are configured. By changing the configuration the hardware can be made to perform a completely different function. Since the structure of the hardware has effectively been changed for the specific function to be implemented, many types of computations can be performed at speeds close to those obtained using application specific hardware. In addition, the configuration can be changed relatively quickly from one function to another.

Computational efficiency requires improvement. Implementing a haze removal algorithm is a challenge in embedded systems, which require real-time performance. Several image haze removal methods are implemented on DSP, GPU, and ASIC. However, the DSP system is far from satisfying the real-time requirements [104]. The GPU cannot be used in embedded systems because of its high cost and power demand [105]. In [106], a 11-stage pipelined hardware architecture was implemented to satisfy real-time haze removal. An ASIC for guided image filtering [84] was designed in [107]. Both the systems in [106] and [107] require a consistent amount of memory and have low performance in processing large images. A real-time hardware accelerator for single-image haze removal was designed in [108]. However, haze removal results have not been satisfactory. [105] suggested a single image de-hazing on Raspberry Pi; it requires 5 seconds to process a single image of dimensions 320x240 and 9 Seconds for an image of 640x480. In [106], histogram & dark channel based real-time image de-hazing was proposed. But this model fails at hardware implementation. Rahul [109] proposed hardware architecture for a single image dehazing system on ZC702 but this required extensive hardware resources. Similarly, [110] proposed HDL based single image dehazing on ZC706 to produce satisfactory results but the system fails at reproducing the de-hazed image. Therefore, the effective utilization of hardware resources of an image processing algorithm is nontrivial.

1.8 Findings in literature

Fig 1.2 represents the basic building blocks involved in existing de-hazing algorithm based on restoration method. The hazy image undergoes the following blocks and produces dehazed image as output. The first most block is dark channel (the darkest pixel among RGB channels),

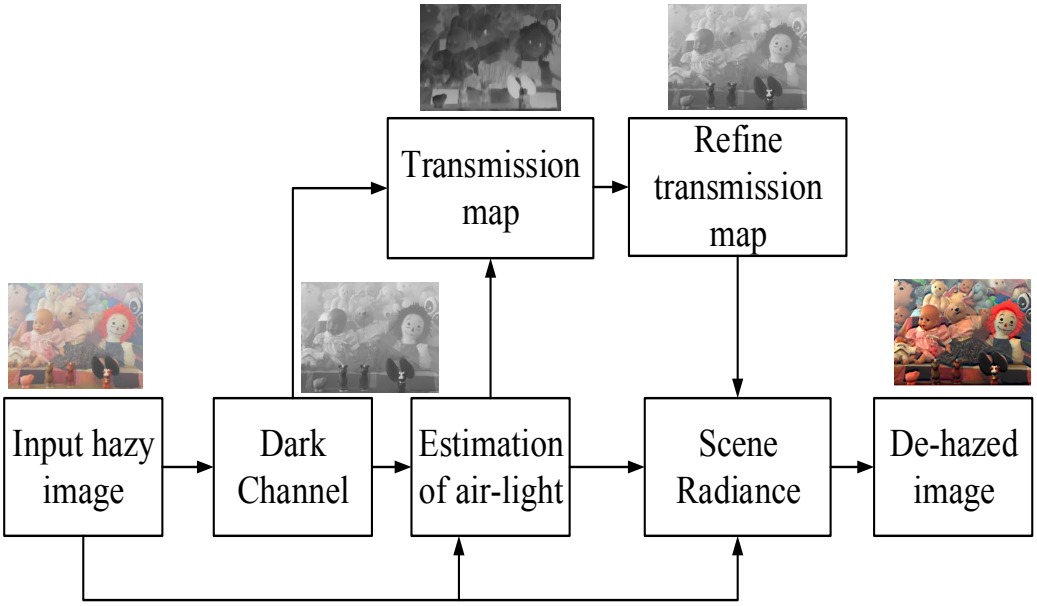


Figure 1.2: Basic building blocks in restiration based dehazing algorithm (DCP)

followed by atmospheric light (the top 0.1% of brightest intensity pixels in the dark channel, corresponding to pixels in the hazy image), transmission map (Normalization of dark channel with respect to atmospheric light). The transmission map produced by this method is inaccurate. Hence, a refined transmission map is needed. The computational complexity for an image of dimensions 25×25×3 with a patch size of 5×5 is described in Table 1.1. It can be observed that the number of computations are more, especially at atmospheric light and refine transmission map step. These computations need to be reduced.

Besides, high complexity operations place a significant role in efficient implementation, such as exponential function, sorting operation, floating-point multiplication, full image buffer, and data transfer between processor & DRAM. These operations decrease the performance of hardware architecture.

Table 1.1: Computational complexity of existing method

Image size	Patch size	Parameter	Computations
25×25×3	5×5	Dark channel	27×27×3×25
		Atmospheric light	2×25×25×100
		Transmission estimation	25×25×3×25×2
		Refined Transmission	25×25×2×2×2
		Final De-haze image	3×25×25×3

1.9 Motivation and Problem statement

1.9.1 Motivation

Based on the discussion, the following problems can be formulated and constitute the motivation for the work presented in this thesis.

The quality of the haze-degraded image is a challenging task, especially rapid de-hazing on a single image for its application in moving systems. As depth information is dependent on the quantum of haze in the atmosphere is unknown and hence acquisition for the depth of single two dimensions image is an unconstrained problem. If multiple images are used to de-haze, different types of imaging devices are needed to shoot multiple images of the same location, which not only has a high cost but also is difficult to apply for a dynamic scene. The collection of multiple images of the same scene in different weather conditions is infeasible for outdoor vision systems with high real-time demand. Therefore, it is necessary to design and implement a rapid de-hazing system on a single image, which can improve the reliability of all kinds of outdoor vision systems in foggy weather.

1.9.2 Problem statement

The goal of this work is to design an algorithm with low computational complexity. Implement an embedded platform for real-time image/video de-hazing systems with a focus on achieving good quality de-hazed images. This has to be achieved in short time, using low memory, and resource. So the problem can be stated as: “Design and development of an embedded framework for real-time image/video de-haze applications”. To achieve this, various aspects need to be taken into account to map a de-hazing application efficiently on different hardware platforms. These aspects include computational time and memory hot-spots, extraction of parallelism and pipeline, partitioning application, data transfer among IP cores, etc.

Thesis Title: **" Some investigations on development of an embedded framework for real-time image/video de-haze applications"**

1.10 Objectives

The existing haze removal methods suffer from high computation complexity owing to, wide-ranging matrix multiplication/division, sorting, and floating-point operations. In low-speed processors, these methods cannot meet the user timing requirements for real-time image processing applications. Therefore, an efficient and low-complexity haze removal methods needs to be proposed for real-time applications. The objectives of the thesis include seeking a good solution for the problem of embedded system design of de-hazing and reduce number of computations, estimating accurate transmission map, preserving the edges and avoiding halos & artifacts in the final de-hazed image. Most computationally complex step of refine transmission map need to be eliminated. The algorithms designed should work in all type of environmental conditions such as sky & non-sky regions, in the presence of one or more atmospheric light and utilize fewer computations. This thesis mainly focuses on the following objectives:

- Estimate an accurate transmission map with fewer computations and remove halos & artifacts in the final de-hazed image.
- Develop fast and efficient de-haze algorithms for computer vision applications.
- Implement various hardware architectures and compare them for image/video de-hazing systems.
- Develop an embedded solution for image de-hazing using IP cores.

1.11 Thesis organization

This thesis consists of six chapters and brief outline of the idea is summarized as follows:

Chapter 1 presents the introduction. It includes a background of the study, history of the research area, motivation towards the problem statement and scope of the research.
It also discusses the latest literature review related to research into de-hazing system design and its implementation. The literature review focuses on depth-based de-hazing methods from previous works for hardware implementation for real-time computer vision applications. A brief review of the progress of embedded platform implementation for real-time image/video

de-hazing systems is also highlighted. Then, the gaps of this study are also identified to develop novel techniques that fulfill objectives of the study.

Chapter 2 is devoted to developing a pixel-by-pixel approach for a single image de-hazing system and its effectiveness is verified with simulation results. Later, the implementation of pixel-by-pixel single image de-hazing technique on TMS 320C6748 DSP Processor is illustrated.

Chapter 3 validates the development of contrast-enhanced image/video de-hazing techniques for real-time applications. Prototype on Jetson Nano and Raspberry Pi is demonstrated. The algorithm performance is observed and evaluated with real-time videos with a frame rate of 24 FPS (Frames per Second) without any noticeable delay from input to output.

Chapter 4 focuses on developing a novel pixel, gray image-based de-hazing algorithm. Its advantages are verified with the help of quantitative and qualitative analysis. Based on comparative analysis, two different hardware architectures for the proposed algorithm are implemented. Out of them one is on DSP Processor (TMS320C6748) with floating pointing operations and another on Zynq-706 with fixed-point operations. The comparisons are made and illustrated.

Chapter 5 discusses the advantages of vector-based patch moment over the square and the development of a memory-efficient vector-based de-hazing algorithm using a median filter. The proposed method is compared with existing state-of-the methods. At last, prototyping of a vector-based technique together with YCbCr color space de-hazing algorithm on Zynq 702 FPGA using hardware-software co-design approach is presented.

Chapter 6 This chapter summarizes the thesis conclusions from the contributions and provides a brief discussion of work that maybe done in future.

References are given at the end of the chapters.

Chapter 2

Implementation of image de-hazing algorithm on DSP TMS320C6748 Processor

This Chapter discusses an attempt to design and implement a pixel-by-pixel approach for single image dehazing systems. First, the pixel-by-pixel based single image dehazing algorithm is designed. Secondly, the same algorithm is implemented on hardware platform of DSP TMS320C6748 processor .

2.1 Introduction

Figure 2.1 shows haze formation model of an image, when the camera trying to capture the image because of water droplets, mist and dirt particle's scattering, the captured image looks like noisy. This noise can be called as haze or fog. Haze is a phenomenon of absorption or scattering of particles in the atmosphere. It influences the attenuation of the radiance along the path towards the camera. Subsequently, captured images suffer from less contrast and quality limiting the visibility of the distant scenes. As it changes the colors of the original scene and reduces the contrast becomes an annoying problem to photographers, it declines the visibility of the pictures and acts as a threat to the several applications like object detection, outdoor surveillance; it also declines the clarity of the underwater images and satellite images. The air light or also called veiling light, is used for approximating the actual in-scattering term in full

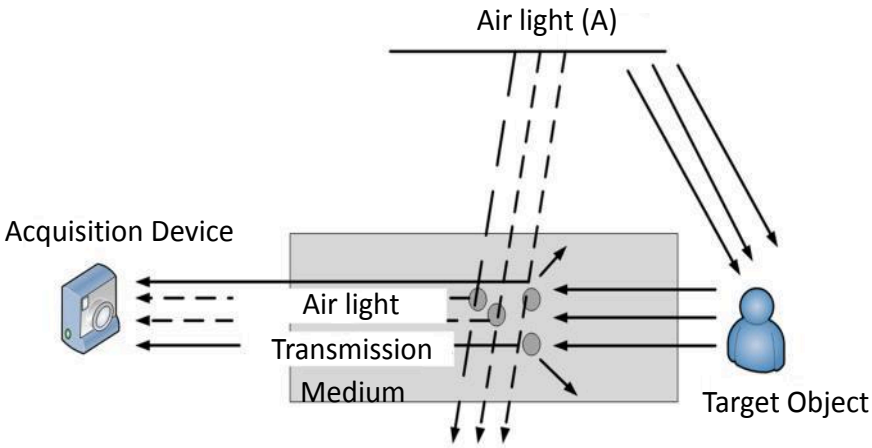


Figure 2.1: Haze formation model.

equation for radiative transport to obtain the following image formation model:

$$I(x) = J(x)\,T(x) + A\,(1 - T(x)) \tag{2.1}$$

Equation 2.1 describes three colour channels of an image where $I(x)$ is observed image, A is air-light colour vector, $J(x)$ is the vector of surface radiance at a point of intersection of the scene, and real-world line ray corresponds to pixel and lastly $T(x)$ is transmitted along the path. The degradation model is generally used in the description of the image formation when haze is present. In the described model, an assumption is made that two components gets summed up to give the final degraded image. Those components are surface radiance which is unknown and contribution of air-light and these two components present in the three-channel colour vectors, are multiplied by a transmission coefficient. This is a scalar coefficient which gives out the each pixel visibility. The colour values present in three surfaces and also transmission values present at every pixel need to be determined to recover an image which is haze-free. The goal of this work is recovering, an image showing the scene through a clear haze-free medium. Also, it is assumed that input image is given as in true scene radiance values. The recovery of radiance map is achieved by extraction of raw data from the camera or by inversion of the curve of total acquisition response. Equation 2.2 represents recovery of de-haze image from hazy image:

$$J(x) = A + \left(\frac{I(x) - A}{max\,(T(x), t_0)}\right) \tag{2.2}$$

This can be achieved by DCP [79] and is based on following observations on outdoor images which are free from haze: In many of patches in non-sky regions, at least any one color channel has few pixels which are low intensity and it is almost zero. Equivalently, the minimum intensity in such patch is close to zero. For describing the above observation formally, the dark channel concept is described. For an input image , its $J^{dark}(x)$ given by

$$J^{dark}(x) = \overset{min}{x\epsilon\Omega}(x)\left(c\epsilon \overset{min}{\{r,g,b\}}\left(J^{c}(x)\right)\right) \tag{2.3}$$

Where J^c color channel of an image J. $\Omega(x)$ local patch which is centered at x. Dark channel can be defined as result of following minimum operators: which is performed for every pixel, and is a minimum filter. After this, top 0.1 percentage of brightest pixel are selected from the dark channel of the hazy image. This pixel corresponds to the objects which covered most by the haze. Of these pixels, highest intensity pixels in an image $I(x)$ is considered for an atmospheric light.

This chapter is structured as follows. Section 2.2 Implementation issues are discussed. In Section 2.3, Methodology used for implementation, Section 2.4 provides an experimental results and discussions. Finally, we draw a conclusion in Section 2.5.

2.2 Implementation issues

The implementation of image de-hazing algorithm on hardware platform is major concern. The issues which are raised during the implementation are discussed in detail.

2.2.1 Computational Complexity

From the literature, it can be observed that the number of computations are more at atmospheric light and refine transmission map step. By employing pixel based approach these computations can be reduced 25 times. In addition, this technique allows to eliminate the halos and artifacts produced in the transmission map. Hence the refined transmission step is not required. This is the way the number of computations are reduced and resultant de-hazed image doesn't contain any halos and artifacts.

2.2.2 The Selection of Hardware Platform

The implementation of digital image technology involves three major elements; they are processor (hardware), software tools, and technical assistance from the third-party systems. The DaVinci technology lays great ground for integrating the three major elements mentioned above. Based on DaVinci technology an ideal core power for signal processing as well as image processing applications a lowest cost DSP device has to choose. For which variety kinds of DSP processors have been compared. Finally, TMS320c6748 [111] is chosen to be the core device of the system. TMS320c6748 (Texas Instruments) evaluation module is chosen as the hardware platform of data processing and Code Composer [112] V6.0 simulator is chosen for implementing simulations.

TMS320c6748 evaluation module is small in size but has plenty of functions, and compact in structure. The hardware feature diagram of TMS320c6748 is shown in Fig. 2.1. The key feature of this board is operating speed; is up to 375 MHz for fixed point and 456MHz for floating point operations. The PCI Host Driver extends configuration options for developers by authorizing them to plug the TMS320c6748 EVM directly into the PC. The EVM can also be used in a standalone fashion, connected to a PC using on-board emulation (XDS 100v3 USB Emulator) through a standard JTAG connector. This processor includes many peripherals, which makes it very suitable for DSP embedded system applications. Its main applications include: signal processing, IP camera, vision system, video surveillance system, portable media and video telephone.

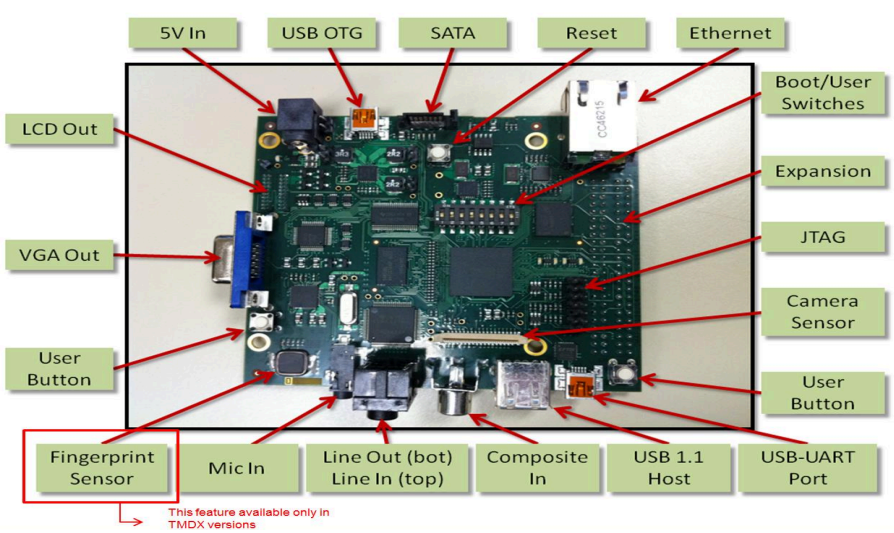


Figure 2.2: Hardware features of TMS320C6748 DSP Processor.

2.2.3 Speed-up Techniques

- To speed-up the implementation process, all floating-point calculations should be replaced with a fixed-point arithmetic. The TMS320c6748 DSP processor has both fixed and floating central processor unit.
- Perform patch moment in pixel wise order, i.e. scan each position of the image pixels rather than matrix form. This technique can reduce the amount of computations in about two times.
- Utilization of processor cache or internal DRAM can also significantly improve performance of operation. It is worth to cache only the data, which is the most frequently used in the system. The image frequently used for processing is stored in DRAM for further operations. Using the linker command, DRAM or processor cache can be selected.
- Use YUV color format rather than RGB. Generally, YUV [113] is a color space used as part of a color image for fast processing of data. It encodes color image pixels by considering human perception into account, allowing to reduce the bandwidth (BW) of chrominance components, thereby enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a "direct" RGB-representation.

The YUV color space usually involves loss of information. Therefore, in this implementation, we used RGB color space with fixed point operations using DRAM.

2.3 Methodology

The flow of hardware implementation is presented in figure 2.2. At the first step, an RGB input image is taken from the well-known standard database of Middlebury [114] and also from flickr website. The hazy images of 640*480 resolution are considered for verification (Suitable for DRAM of processor).

At pre-processing stage the image is cast into a standard data type of "unsigned integer 8-bit"(uint8) format, to use fixed point arithmetic operations. Next, each channel then converted to header form (Raw form) suitable for reading the image in Code Composer. Followed by implementing a de-haze algorithm in ‘C’ code. If it is free of errors then, the project is successfully built by the assembler/compiler/linker and, then debugging process invoked to program

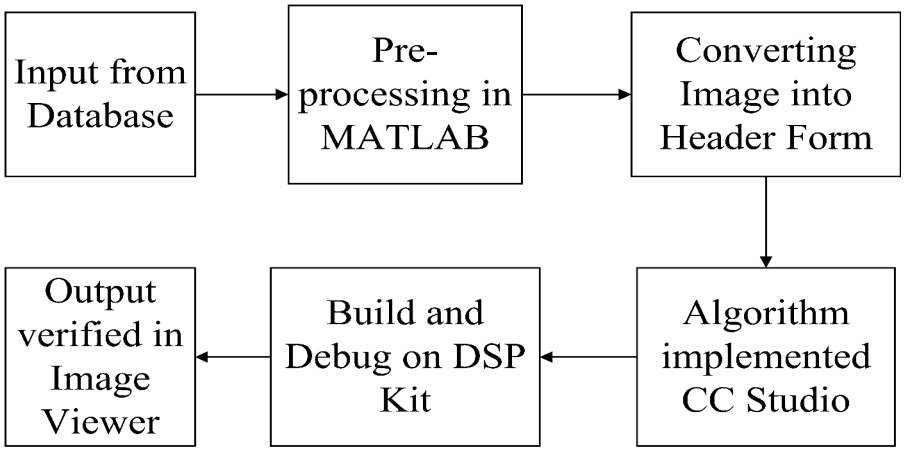


Figure 2.3: Flow of hardware implementation

into the device. The standalone executable file (.out, .hex, .axf, etc.) was generated during the build process, used for programming the device. The implementing de-haze algorithm in code composer as follows.

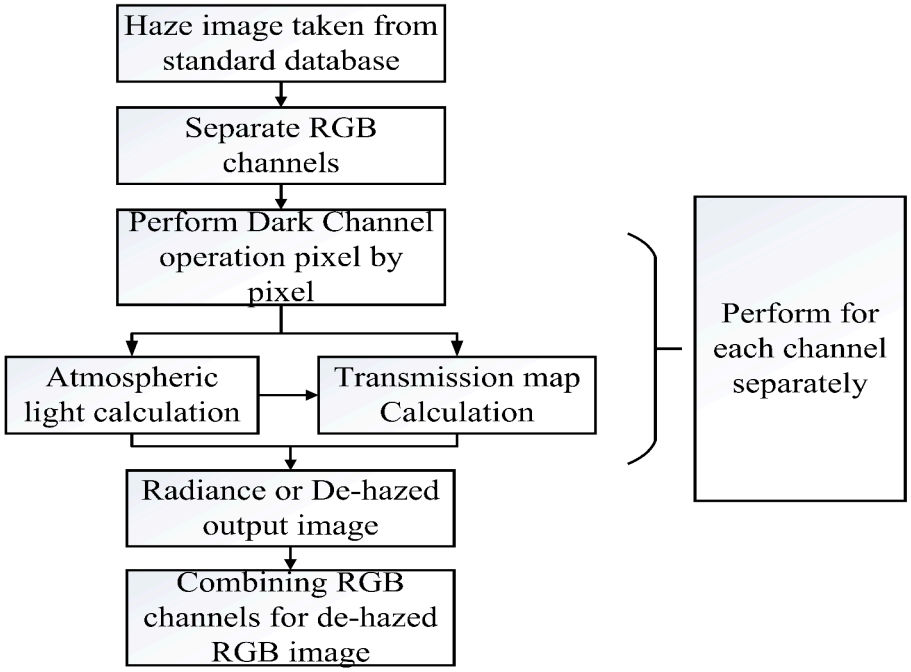


Figure 2.4: Basic Steps involved in hardware implementation of image De-hazing using TMS320C6748

The corresponding pseudo-code of the proposed framework is illustrated in algorithm **??** Figure 2.4 represents basic steps involved for implementing image de-hazing on DSP TMS320C6748 processor. The process is applied on each color channel individually and combined at the final step. After separating the RGB channels minimum among the RGB pixels are calculated and considered it as dark channel. Later, to estimate the transmission, we assumed an atmospheric light (A) as constant and is known prior. Then the haze model equation

Algorithm 1 Proposed de-haze algorithm

Result: Produces de-hazed image

initialization dark_img Input_frame, RGB_img, Transmission, air_light, dehaze_RGB, scene_radiance

Convert RGB to CSV form using MATLAB

while *each channel of RGB_img* **do**

| find dark_img=from RGB_img

end

while *each channel of RGB_img* **do**

| **while** *dark_img* **do**

| | find air_light

| **end**

end

while *elements in RGB_img* **do**

| Transmission=dark_imge/air_light

end

while *elements in RGB_img* **do**

| dehaze_RGB=air_light+(RGB_img-air_light)/Transmission

end

while *each channel of RGB_img* **do**

| combine dehaze_RGB

end

scene_radiance=dehaze_RGB

is normalized with respect to atmospheric light resulting is transmission map. Algorithm used to calculate this light constant is based on the key observations of hazy images, i.e., top 0.1 percentage of the brightest pixel are selected from the dark channel of the hazy image. These pixels correspond to the objects which covered most by the haze. Of these pixels, highest intensity pixels in an image (I) is considered for an atmospheric light.

Moreover it is also assumed that, the value of transmission over the local patch re-mains fairly constant copmpared with existing method that is $\Omega(x) = 1$. The radiance of the actual scene can be recovered according to eq. 2.2 when the air-light component and transmission map is known.

2.4 Results and Discussion

To verify the validity of implemented design, we choose a well-known standard data-base Middlebury with 640*480 resolution of image. This database covers all the challenging factors.

Figure 2.5: Results of de-haze system on hardware platform

The fig. 2.5 and **??** demonstrates results of de-haze algorithm on hardware platform. In addition to visual observation, the quantitative examination has been done. The comparison of de-hazing technique which was implemented on hardware has made with MATLAB 8.1 program on i5 CPU 3 GHz and 8GB memory with OS window 8.1 using different quality metrics of Percentage of perceptual haze density (Visibility Analysis), and execution time (ET). It is observed that perceptual haze is almost similar in both cases. The outcome of each step of proposed algorithm in each channel is represented in fig. 2.5. It can be seen that final dehazed image doesn't allows halos and artifacts. The further refinement of transmission map is not required. The qualitative comparison of algorithm is represented in fig. **??**. The table 2.1 represents the execution time and haze improvement of haze algorithm in two different aspects such as in MATLAB environment and on hardware environment. The process time

Table 2.1: Performance analysis of proposed algorithm on hardware platform

Database	Execution Time	Percentage of Haze improvement	
	Hardware platform (m Sec)	MATLAB	Hardware platform
Shopvac	13.86	20.3642	20.1972
Toys	16.59	54.8686	53.7846
Pumpkins	24.92	45.3664	43.9418
Backpack	15.43	26.5988	26.5432
Recycle	43.94	67.3824	67.2681
Shelves	54.35	22.3979	22.3872
Sword1	24.58	42.9285	42.8924
Bicycle	23.18	42.1782	42.0356

needed for executing in a hardware platform is very less and is in terms of nanoseconds. It is observed that proposed algorithm produces de-hazed image on TMS320C6748 is less than 50 milliseconds. While implementing the algorithm on hardware to speed it up, we employed "uint8" format. Consequently, it resulted in loss of some amount of information and percentage of haze improvement is varied which is negligibly small.

Figure 2.7: Experimental setup of image de-hazing system

Fig 2.7 indicates experimental test setup of proposed single image de-hazing algorithm on TMS320C6748 DSP processor.

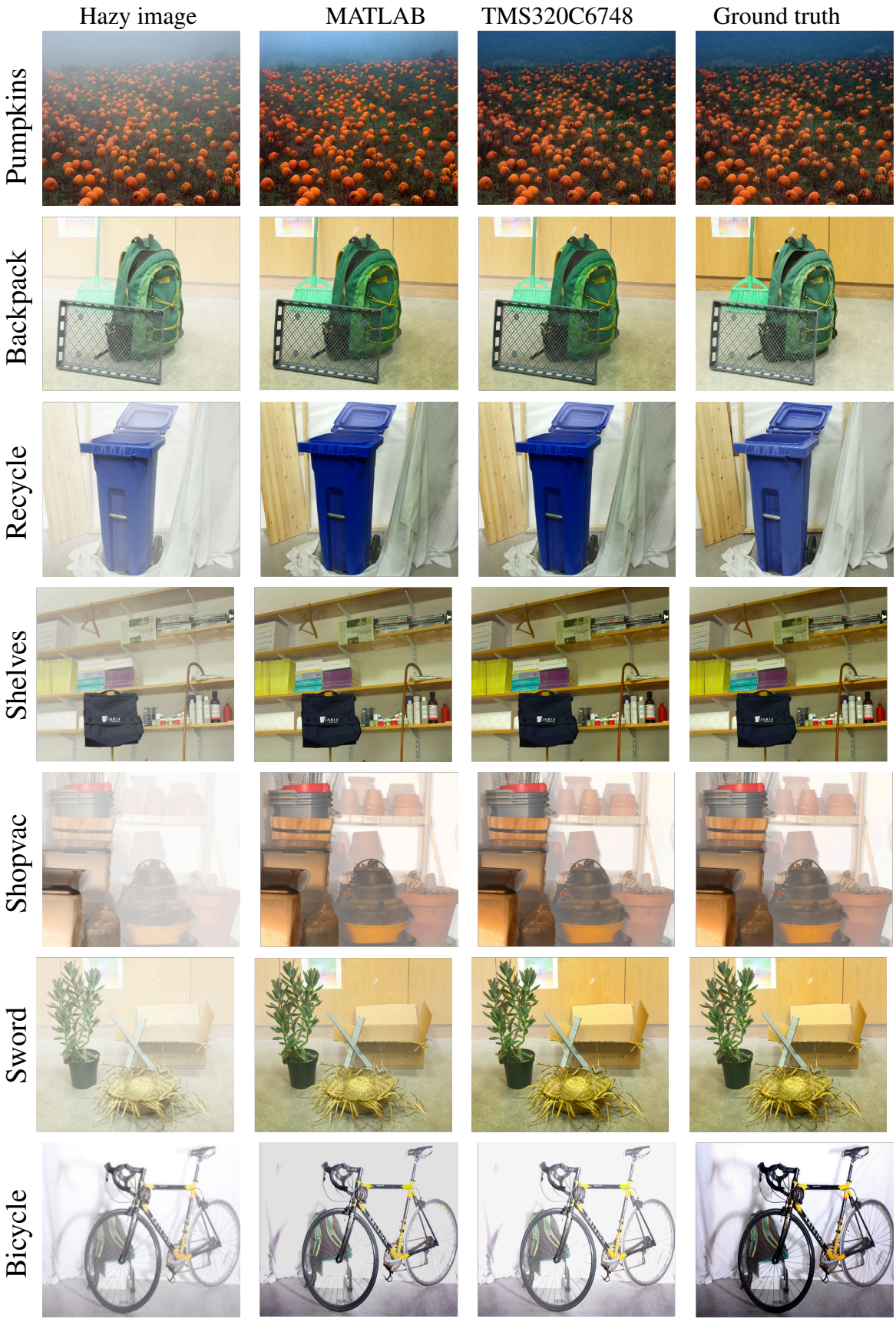


Figure 2.6: Qualitative comparison between simulation and hardware implementation

2.5 Summary

In this chapter, we have implemented de-haze algorithm on TMS320C6748 based on DaVinci technology and is achieved in the offline image platform. The use of fixed point technique and pixel based operation mostly diminish the development cycles and its complexity. The correctness and efficiency is justified using quantitative and qualitative evaluation parameters. This demonstration shows a feasible framework for the hardware acceleration of computer vision applications. In future, by interfacing camera module to the DSP processor same algorithm can be implemented for real-time applications.

Chapter 3

Contrast enhanced image/video de-hazing based on transmission estimation using HSL color model

This chapter focuses on the development of an efficient and simple real-time video dehazing algorithm using the HLS color model on embedded hardware platforms; Raspberry Pi and Jetson Nano. The performance evaluation metrics used for effective verification of proposed de-hazing algorithm with existing state of the art methods are described.

3.1 Introduction

Improving the visibility of hazy images and videos' is desirable in robot navigation, security surveillance, and other computer vision applications. The presence of fog significantly damages the quality of captured images and videos; this affects the surveillance system's reliability and poses a potential danger. Therefore, developing a simple and efficient real-time video de-hazing algorithm is essential. Removing haze is a challenging problem since transmission depends on the unknown depth and varies with light. Therefore, atmospheric light and transmission map estimation is the core step involved in haze removal algorithms. Various types of prior methods are available for single image dehazing. These methods are used to understand the haze-free images statistically. The contributions of this work are summarized as follows:

- We estimated an accurate transmission map using the HSL (Hue, Saturation, and Light) color model together with RGB (Red, Green, and Blue) color space.
- In addition to reducing the number of computations, this method preserved the edges and avoided the halos and artifacts. This can be accomplished by employing the median of the pixels. Another advantage of this method, eliminate the most computationally complex step of refine transmission map.
- Initially, this method was analyzed for a single image and verified its advantages with five existing classical methods, later extended it for real-time video.
- At last, we mapped this method on Raspberry Pi3 and Jetson Nano with 24fps (frames per second) without noticeable delay from input to output.

3.2 Proposed method

We tend to find a more compact imaging model that rewrites the original image formation model (eq. 2.1) as follows.

$$\bar{I}(x) = \bar{J}(x)T(x) \tag{3.1}$$

where $\bar{I}(x) = A - I(x)$ and $\bar{J}(x) = A - J(x)$. From the equation (3.1)

$$\bar{J}(x) = \frac{\bar{I}(x)}{T(x)} \tag{3.2}$$

In RGB image, the intensity of the pixels range from 0 to 255. To simplify the calculation, the intensities are normalized in the range [0, 1]. i.e. $0 \leqslant \bar{J}(x) \leqslant 1$. Then the following equation (3.2) can be written as:

$$0 \leqslant \frac{\bar{I}(x)}{T(x)} \leqslant 1 \qquad i.e., \quad 0 \leqslant \frac{A - I(x)}{T(x)} \leqslant 1 \tag{3.3}$$

In equation (3.3), a small change in $T(x)$ leads to larger change in the entire term. Hence based on the equation 3.3, it is necessary to estimate an accurate transmission map. As the existing algorithm uses RGB color space (3 channels) to estimate the transmission map it requires more computations. Whereas the proposed method works based on the light component (single channel) of the image. Therefore it can reduce the number of computations. Figure 3.1 depicts the proposed framework for a single image de-hazing system; later, it can be extended to a real-time video de-hazing without modification in the work-flow.

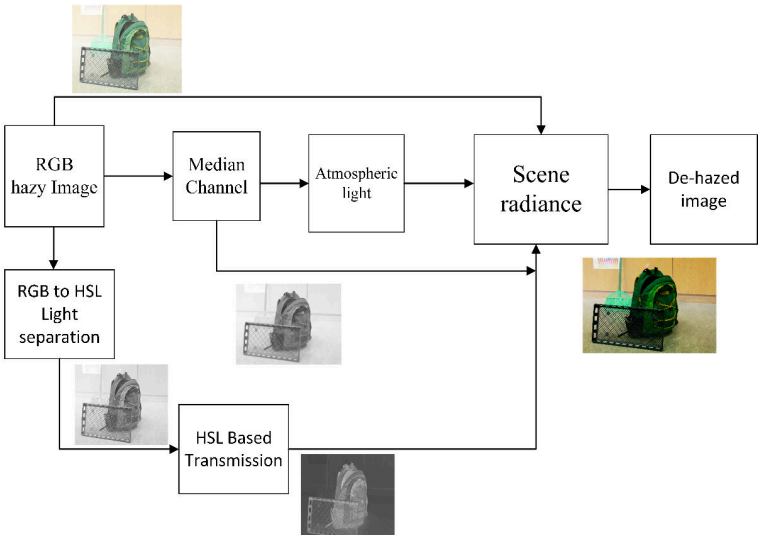


Figure 3.1: Block diagram of proposed frame work

3.2.1 Median channel estimation:

A median is nothing but the middle value or average of the middle values within the list. It is a non-linear signal processing technique based on statistics. Identifying the median is that pixels within the mask are arranged based on their gray levels. Later, the middle value of this group is stored to replace the original value. It can be represented as

$$M(x, y) = med(N(x - i, y)), i \epsilon W \tag{3.4}$$

Where $M(x, y)$ & $N(x, y)$ are de-noise and noisy image respectively, W is a mask. As any image's information will hide in the middle of the image, instead of finding the minimum of pixels in , we employed median value.

$$N^{median}(x) = \underset{x \epsilon w(x)}{median} \left(\underset{c \epsilon \{r,g,b\}}{median} \left(N^c(x) \right) \right) \tag{3.5}$$

3.2.2 Atmospheric Light Estimation (A):

The determination of the unknown parameter 'A' is a challenging task. In most of the existing de-haze methods, atmospheric light is estimated from the most haze-opaque pixel. For example, the pixels with the highest intensity are considered as atmospheric light in [79] and

further refined in [84]. In our method, the top 0.1% of brightest intensity pixels in the median channel (3.5), corresponding pixels in the hazy image, are used as atmospheric light.

3.2.3 Transmission map estimation (T(x)):

When the dark channel failed, it produces an inaccurate transmission map. Hence, we proposed the transmission map such that it should be independent of the dark channel. It can be derived as follows.

Generally, the transmission map $T(x)$ is a function of two parameters and can be expressed as:

$$T(x) = exp^{-\beta d(x)} \tag{3.6}$$

where β scattering coefficient and $d(x)$ is depth of the scene ranges from $[0, +\infty)$. From equation (3.6), when d tends to infinity, the captured image is very far from the observer, and at the same time $T(x)$ tends to zero, which means that there is no transmission map. This case is ambiguous; another observation from equation (3.6) is that if T tends to infinity, no image exists, i.e., image de-hazing depends on transmission map $T(x)$. Therefore, from eq. (3.3) and (3.6) the estimation of an accurate transmission map is needed.

In the proposed method, HSL based transmission map is given by:

$$T(x) = exp^{-\beta L^*(x)} \tag{3.7}$$

Where $L^*(x)$ denotes modified light intensity value; based on studies on hazy image formation and Mie scattering model in [51] , the value of β is strongly correlated with the wavelength (λ) of every channel (RGB). It is determined by the amount of haze, object distance, and camera angle. Usually, a camera angle of 60° is sufficient to capture images with sky regions. Therefore, the scattering coefficients (β) for red, green, and blue channels corresponding to a camera angle of 60° are selected as 0.3324 (λ=700μm), 0.3433 (λ=520μm) and 0.3502 (λ=440μm) respectively. The original light intensity value is derived from the HSL color map and can be stretched to real depth as follows.

$$L^*(x) = \frac{\tau}{L^a} L(x) \tag{3.8}$$

τ describes the depth map range, L^a indicates the top 80 percentile value of lightness L. The modified transmission map utilizes relative wavelengths. Hence it provided better accuracy and worked well in real-time sky and non-sky regions. Figure 3.2 shows the visual comparison

of different algorithm's transmission maps for bicycle database. It is observed that the proposed method can estimate an accurate transmission map when compared with other existing algorithms. These observations are discussed in Sec 3.3.

3.2.4 Scene radiance J(x):

Once, the transmission ($T(x)$) and air-light (A) is known, the scene radiance ($J(x)$) can be determined as

$$J(x) = \left(\frac{I(x) - A}{max(T(x), t_o)} \right) \tag{3.9}$$

where t_0 is the control parameter. The scene radiance is usually not as bright as an atmospheric light, and the image after haze removal looks dim. So, we increase the exposure of $J(x)$ for display by taking t_0. The typical value of t_0 is considered 0.1.

3.3 Performance analysis of proposed method:

To verify all algorithms' efficiency, we developed an experimental setup with the hardware Lenovo computer consisting of an Intel(R) i5-4690S 3.2 GHz CPU and 8GB RAM. The test images are taken from a well-known standard de-haze database called Middlebury, which contains 1400 test images related to natural landscapes, indoor scenes, and aerial images. The outcome assessment includes subjective (visual), objective (performance metric) evaluation, and computational complexity. They are performed qualitatively and quantitatively.

3.3.1 Qualitative or Visual analysis:

Figure 3.2 represents the comparison of the transmission-map. It is observed that the background of the image disappeared in Zhu [115], Tarel [116] and D. Berman [117] methods, whereas halos are seen in He's method [79] and transmission becomes darker in Cai's method [118]. As a result, the output de-hazed image produced by these methods is associated with the problem of over enhancement and background color shift (it can be seen in Figure 3.3). Therefore, the transmission map estimated by other methods is not clear, and information loss exists. In contrast, the transmission-map estimated by the proposed method is accurate & clear.

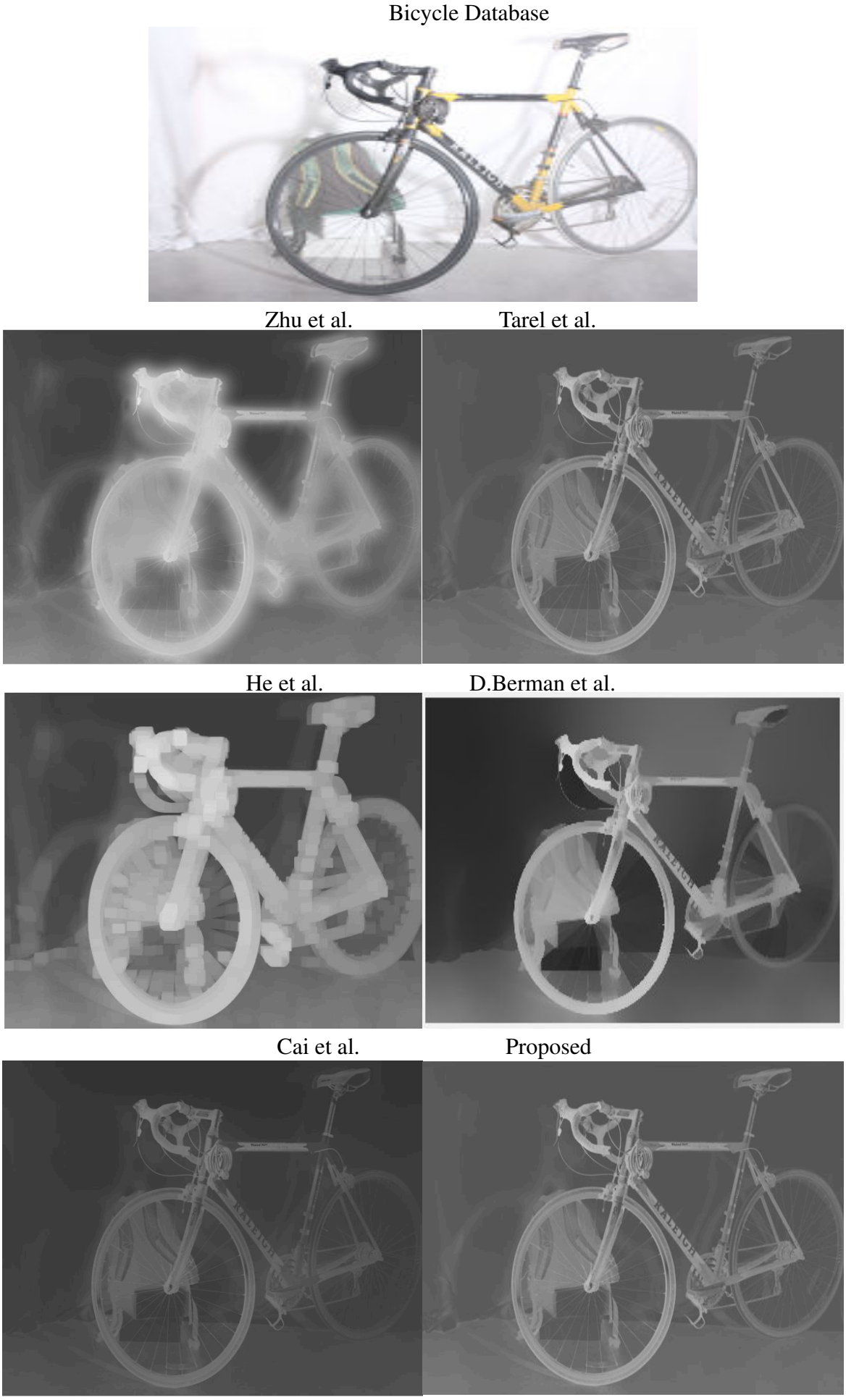


Figure 3.2: Transmission depth map of different algorithms.

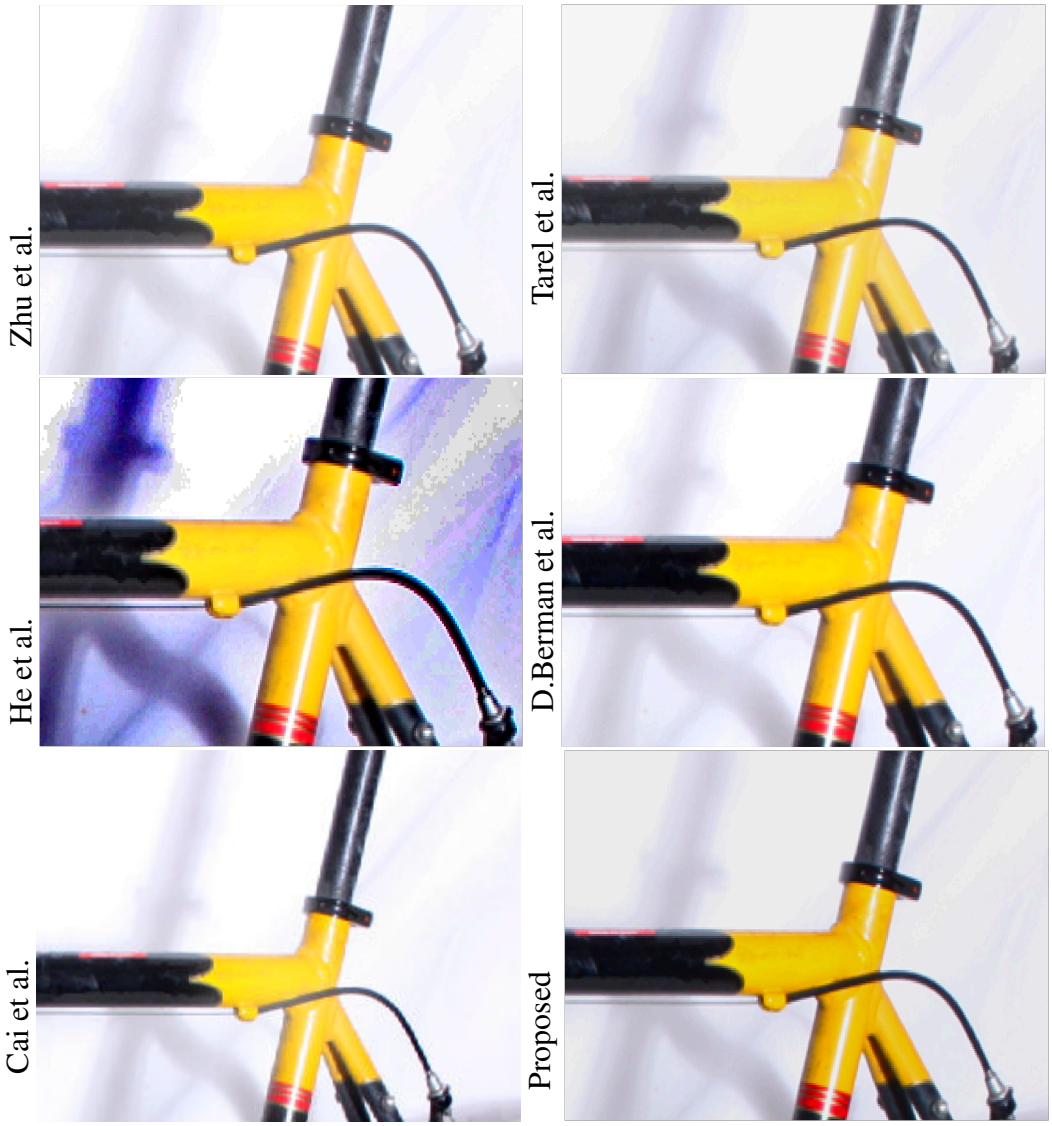


Figure 3.3: Closer view of restored images of different algorithms.

Figure 3.3 depicts the closer view of restored images of different algorithms. It is observed that the methods, Zhu, D. Berman, and He are suffered from background color-shifting and the over enhancement is associated with Tarel's method. Because of the under extraction of some hazy input images' features, Cai's method produces undersaturated images. Whereas, the proposed framework is disallowing halos & artifacts and background color shifting.

To further demonstrate the performance of the proposed method, we employed some standard test images. Figure 4.11 illustrates the few randomly selected experimental results. The motorcycle, vintage, adirondack, office and chairs images are examples of thick haze. Whereas, bicycle, backpack are examples of thin haze. The quality of these images in the proposed method is closer to the ground truth images, with no halo effect and over-saturation in color. It suggests the advantage of the proposed method.

3.3.2 Quantitative analysis:

To further validate the proposed method's effectiveness, we examine the experimental results with six quantitative metrics. They are PSNR [119], PHI, SSIM [120], MSE [119], ACOI, ATC.

3.3.3 PERFORMANCE EVALUATION METRICS

Evaluation metrics gives information about which scheme of the method can be preferable. In this paper, five performance metrics are used for evaluating the efficiency of the proposed algorithm.

PSNR and MSE:

PSNR and MSE are well-identified performance metrics for measuring the degree of error because these represent overall error content in the entire output image.

PSNR is defined as the "logarithmic ratio of peak signal energy (P) to the mean squared error (MSE) between output N_i and input M_j images". It can be expressed as

$$PSNR = 20 \times log\left[\frac{max(P))}{\sqrt{MSE}}\right] \tag{3.10}$$

$$MSE = \frac{1}{kn}\sum_{i=0}^{k-1}\sum_{j=0}^{n-1}\left\|M_i - N_j\right\|^2 \tag{3.11}$$

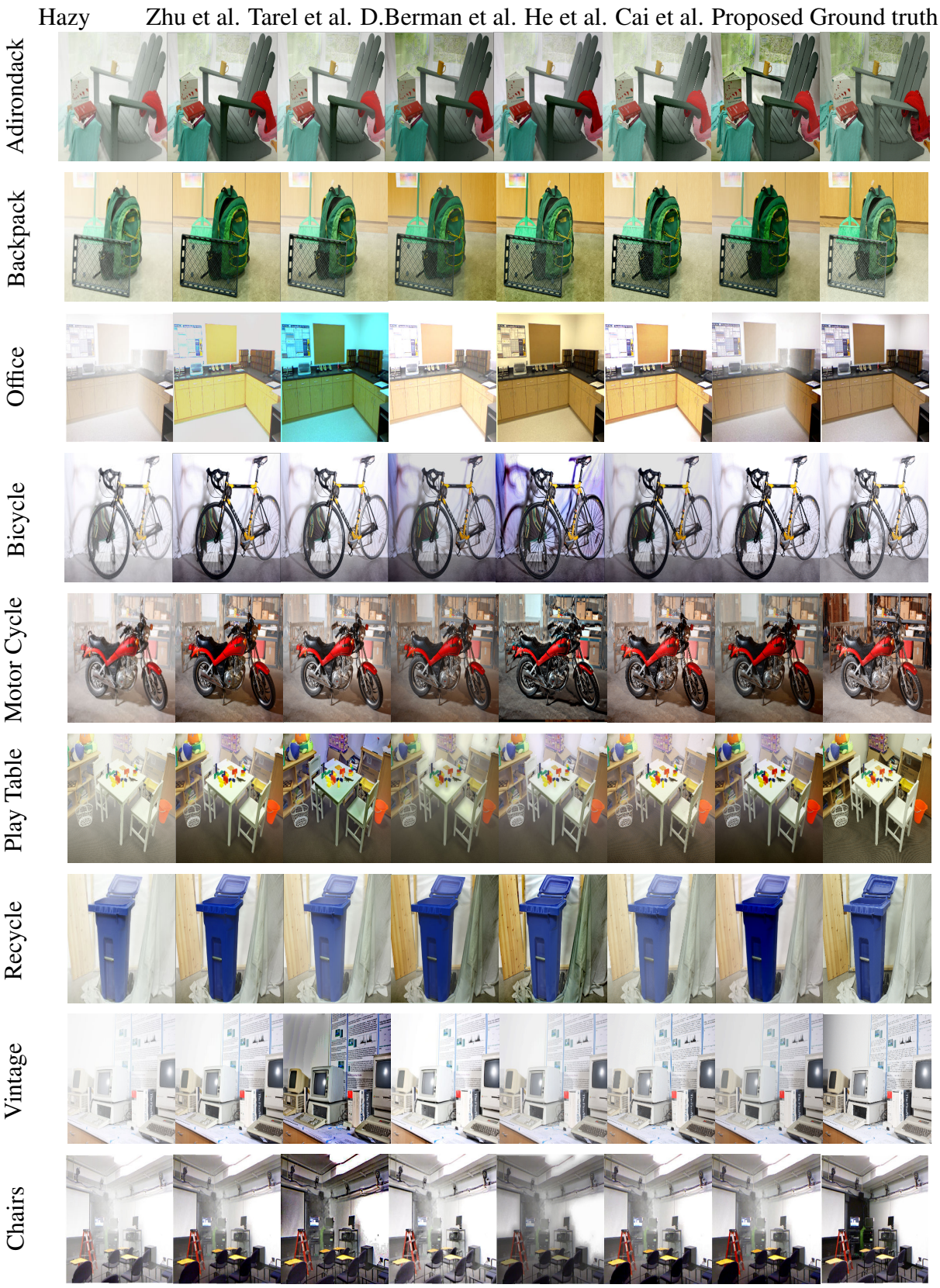


Figure 3.4: Visualization results of different de-hazing systems.

where P is the maximum value of the pixel in an image (typical value of P=255), MSE is Mean Squared Error, k, n are the no. of rows, no. of columns of the image respectively. Generally, the value of PSNR would be desirably high.

Computation time or Average time cost (ATC):

It represents the amount of time needed to complete an algorithm. The unit is in seconds. The average time cost of our proposed algorithm is estimated with the other four existing algorithms in 21 number of iteration.

Average contrast of output image (ACOI):

$$C = \frac{1}{MN}\left(max\left(\frac{L_{max} - L_{min}}{L_{max} + L_{min}}\right)\right) \tag{3.12}$$

Here, L_{min} L_{max} is the minimum, maximum luminance of an output image correspondingly. M, N row, and column of an output image respectively. Generally, the maximum value of it directs that an image is more quality.

Structural similarity index (SSIM):

It is a method for quantifying the similarity between two images. The SSIM can be observed as a quality measure of one of the images being compared with the other image is regarded as of perfect quality. The value of SSIM is ranging from 0 to 1. The maximum value of 1 indicates that the two images are structurally similar while a value of 0 indicates no structural similarity.

PSNR gives the image structure's integrity. From Table 3.1, the value of PSNR is high in the Cai's and proposed method. Hence, the de-hazed images are clear and natural over other methods.

PHI, SSIM is designed to reflect the traditional method like PSNR, which has proven inconsistent with human eye perception. If the PHI, SSIM values are high, then the target image is more similar to the reference image. These metrics help us to compare the difference among the de-hazed images. From table 3.1, our proposed method has higher PHI, SSIM, and ACOI. Therefore, the reliability of the proposed method is validated thoroughly.

Another quantitative metric used for validation is MSE. It is observed that the value of MSE is low for almost all image databases. Since the median of pixels is employed, our method's mean square error is a little higher and negligibly small. In comparison, our approach is balanced at MSE and achieves effective de-hazing without sacrificing MSE.

To verify the efficiency of all methods, we have selected the average computational time (ATC). The computational cost is estimated for image dimensions of 800×600 and tabulated in table 3.1. It is observed that He et al.'s method is based on soft matting, Cai et al. & Zhu et al.'s methods are based on feature extraction. In contrast, D. Berman's work is based on color clus-

Table 3.1: Comparison of quantitative metrics of different de-hazing algorithms

Database	Method	PSNR (dB)	ATC (Sec)	ACOI	MSE	PHI	SSIM
Adirondack	Zhu et. al.	25.83	1.441288	0.0123	1.63	12.1976	0.7612
	Tarel et al.	26.14	2.885601	0.0123	1.52	10.3668	0.0482
	He et al.	45.14	2.038838	0.0268	1.87	07.4004	0.6813
	D. Berman et al.	45.77	2.163758	0.9657	1.72	26.1645	0.7765
	Cai et al.	46.51	1.877650	0.0486	1.45	20.7800	0.4912
	Proposed	**47.83**	**1.155503**	**1.4617**	**1.07**	**27.3232**	**0.7871**
Backpack	Zhu et. al.	22.48	1.489615	0.0537	2.58	15.9559	0.5077
	Tarel et al.	43.15	1.865297	0.0313	2.75	15.6305	0.0217
	He et al.	43.76	2.035956	0.0103	2.92	18.4239	0.0888
	D. Berman et al.	**44.78**	2.875439	1.6754	2.77	18.6745	0.5908
	Cai et al.	44.73	1.857791	0.1084	2.54	05.6306	0.4454
	Proposed	43.47	**1.212019**	**1.7119**	**2.07**	**21.5011**	**0.6652**
Bicycle	Zhu et. al.	23.98	1.472224	0.0382	2.49	16.3358	0.7719
	Tarel et al.	23.98	1.898116	0.1195	2.49	15.7125	0.2270
	He et al.	23.98	1.987793	0.0059	2.49	23.2702	0.7494
	D. Berman et al.	22.07	3.675125	1.8302	3.91	17.8601	0.7617
	Cai et al.	**44.72**	1.815449	0.0914	**2.19**	13.6584	0.4446
	Proposed	44.43	**1.127629**	**2.7532**	2.35	**29.2658**	**0.8822**
Office	Zhu et. al.	**26.11**	1.393388	0.0448	1.55	16.6764	0.782
	Tarel et al.	45.14	1.845369	0.0964	1.73	07.6522	0.0425
	He et al.	46.75	2.029721	0.0193	1.38	13.1743	0.8354
	D. Berman et al.	45.18	3.790123	1.2765	2.12	16.6510	0.7219
	Cai et al.	46.61	1.823412	0.0973	1.35	08.4635	0.3781
	Proposed	47.10	**1.146288**	**1.8996**	**1.25**	**23.6419**	**0.9465**
Umbrella	Zhu et. al.	26.85	1.525979	0.0170	1.29	05.2107	0.7973
	Tarel et al.	43.47	1.773976	0.0257	1.71	01.2454	0.0222
	He et al.	43.35	1.964879	-0.0265	1.95	06.6677	0.6674
	D. Berman et al.	42.17	2.451839	1.2794	2.12	14.1248	0.6814
	Cai et al.	42.45	1.841045	0.0957	2.07	**14.8767**	0.4367
	Proposed	**47.58**	**1.126600**	**1.4858**	**1.08**	12.1262	**0.7991**
Motorcycle	Zhu et. al.	43.16	1.418704	0.0329	1.51	03.2373	0.6854
	Tarel et al.	43.16	1.906796	0.0341	1.51	06.6550	0.2098
	He et al.	43.16	2.029526	-0.0195	1.51	04.1131	0.4782
	D. Berman et al.	42.59	2.756102	**2.1738**	2.31	**27.6529**	0.6571
	Cai et al.	43.16	1.809674	0.0587	1.06	06.1840	0.2284
	Proposed	**43.72**	**1.151892**	1.7430	**0.98**	20.3847	**0.8591**
Couch	Zhu et. al.	42.35	1.32560	1.4556	2.19	16.1343	0.7492
	Tarel et al.	42.72	1.824160	1.5482	2.72	16.5521	0.1996
	He et al.	42.18	1.985484	**1.9527**	2.05	15.5750	0.8486
	D. Berman et al.	42.97	1.581258	1.3218	2.88	18.8065	0.7816
	Cai et al.	**43.04**	1.828032	0.1192	**1.59**	02.4292	0.5238
	Proposed	42.56	**1.115626**	1.4576	2.19	**32.1741**	**0.9397**
Chairs	Zhu et. al.	42.08	4.295092	0.0495	2.49	16.0677	0.7898
	Tarel et. al.	42.75	1.750527	**0.6080**	2.49	48.5453	0.1320
	He et al.	42.05	1.987452	0.1857	2.54	16.5410	0.2365
	D. Berman et al.	41.96	2.187521	0.6099	1.32	33.7685	0.7851
	Cai et al.	42.08	1.818684	0.1352	2.15	17.3826	0.5012
	Proposed	**45.71**	**1.715283**	0.4681	**0.91**	**48.7666**	**0.8528**

Table 3.2: Comparison of average time cost (in sec) of the dehazing methods in MATLAB.

S.No.	Image Size	He et al.	Tarel et al.	D. Berman et al.	Zhu et al.	Cai et al.	Proposed
1	600x400	2.168750	1.790873	2.361475	1.056444	1.879861	1.076313
2	800x600	3.494644	3.850005	3.851637	1.986920	3.158457	1.689875
3	1024x768	5.523964	6.504838	5.421678	2.987708	5.785461	2.683828
4	1366x1080	10.415705	12.146995	9.534128	5.895126	11.587427	4.532522
5	1080x1920	15.008219	16.902982	15.812619	8.331968	15.236518	6.664180

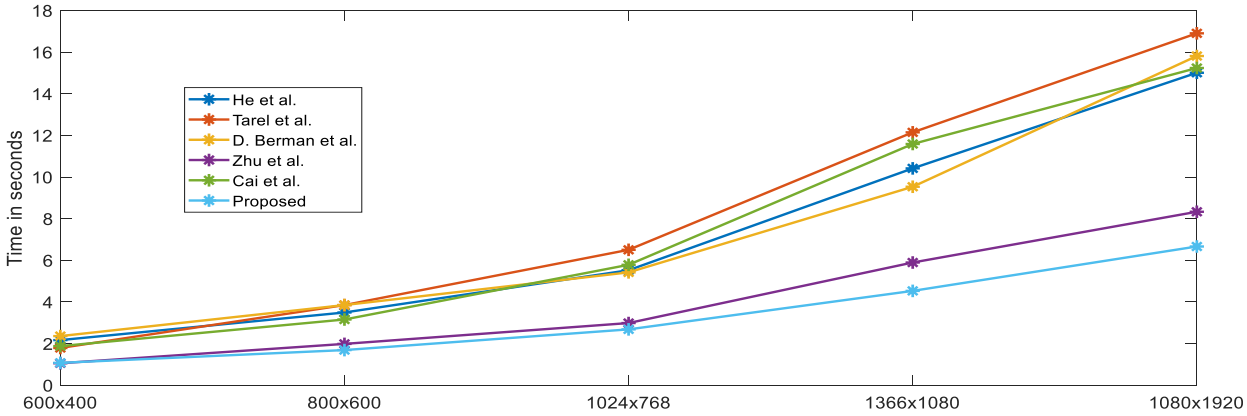


Figure 3.5: Average time cost of the dehazing algorithms

tering, and Tarel et al.'s is based on atmospheric veil inference; hence, these methods have high computational complexity. Whereas our method is based on the HSL model and median filter, thereby it took minimum computational time over other methods, approximately 1.715283 Seconds. From Table 3.1, out of six metrics, our algorithm competitive in at least four metrics.

Further, the proposed algorithm's average computational time is compared in 21 iterations with different dimensions of images. Table 3.2 indicates the average computational time of different de-hazing algorithms. The proposed algorithm takes lower execution time, among other algorithms. Figure 3.5 shows the comparison of average time cost of de-hazing algorithms. The proposed algorithm's computational time for lower dimensional images (600×400, 800×600, and 1024×768) is balanced with Zhu et al.'s method. When the image dimensions are increased to 1366×1080 and 1080×1920), our method is dominating. Table 3.3 represents a comparison of the proposed algorithm with five state-of-the-art algorithms in aggregated fashion for a set of 100 test images. From the mean, min and max value of each metric, the proposed algorithm gives better results than other methods.

Besides, the values of the scattering co-efficient (β) and relative wavelength (λ) are chosen as constant in the proposed method. Hence, the proposed algorithm can only remove the haze within a distance of 5 Kilometers range.

3.4 Real-time Video acceleration:

3.4.1 On Raspberry Pi 3:

We employed Raspberry Pi 3 [121] for accelerating the algorithm in real-time applications. Raspberry Pi 3 is a powerful mid-range computer. It has built-in software, which enables programmers to program and design animation, game, or video. Besides, they can develop programming using Python language [122]; Python is the core language in the Raspbian operating system. Figure 3.6 represents hardware test setup of our proposed de-hazing algorithm for real-time video captured on Raspberry Pi 3. The de-hazed output is displayed on the monitor through HDMI. Figure 3.7 represents process of the real-time video acceleration. At initial, the duration of the video is calculated and is stored in EOF (end of frames). Later, the algorithm is applied to each frame for every incremented count value until the condition 'Count < EOF' satisfied and the successive frames are sent to the display unit.

Figure 3.6: Implementation of real-time video dehazing on Raspberry Pi3

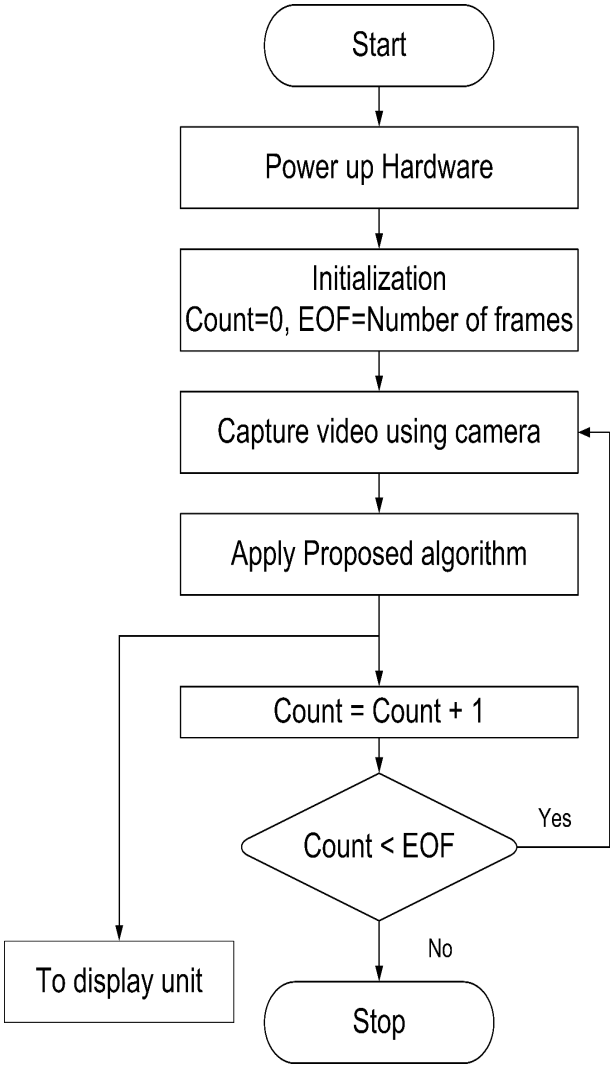


Figure 3.7: Flowchart of video acceleration on Raspberry Pi & Jetson Nano

Figure 3.8 describes the architecture level implementation. It can be observed that the proposed implementation requires a 3-stage pipeline structure.

Stage 1: The camera captures the input frames is called reading phase.

Stage 2: Executing phase: A parallel processing technique is employed for estimating the median channel, atmospheric light, and transmission map.

Stage 3: Displaying Phase: Displays successively processed frames.

The processing time required to execute the proposed algorithm on Raspberry Pi3 is slightly higher. It can be further reduced by accessing the GPU. Hence, we have chosen another hardware platform with GPU.

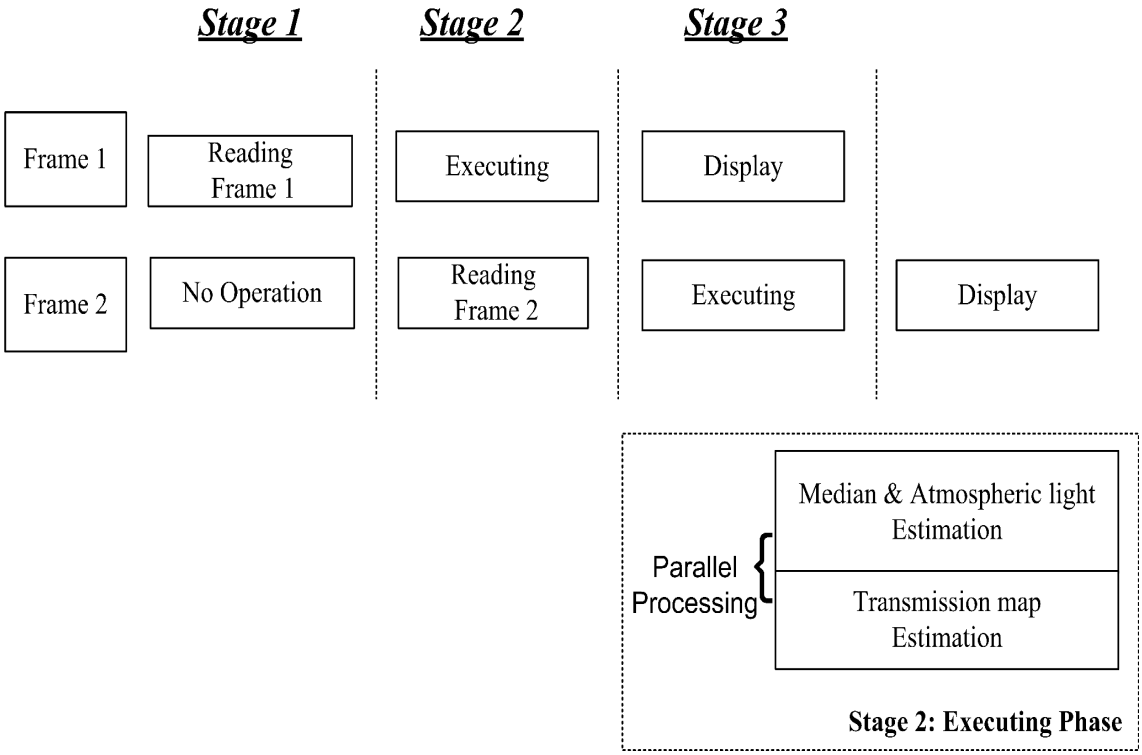


Figure 3.8: Architecture of proposed method on Raspberry Pi & Jetson Nano

3.4.2 On Jetson Nano:

NVIDIA Jetson Nano [102] is an AI (Artificial Intelligence) computer. It has 128-core Maxwell GPU and a quad-core arm A57 processor. It supports 4×1080 @ 30 FPS. Hence, we have chosen Jetson Nano to enhance the fastness of our algorithm for real-time applications. Figure 3.9 represents the hardware test setup of the proposed algorithm on Jetson Nano. The implementation process is almost similar to Raspberry Pi except at GPU accessing using Nvidia CUDA [123] Compiler (nvcc). Ideally, the execution time on GPU is at least 100 times faster than the same on a CPU.

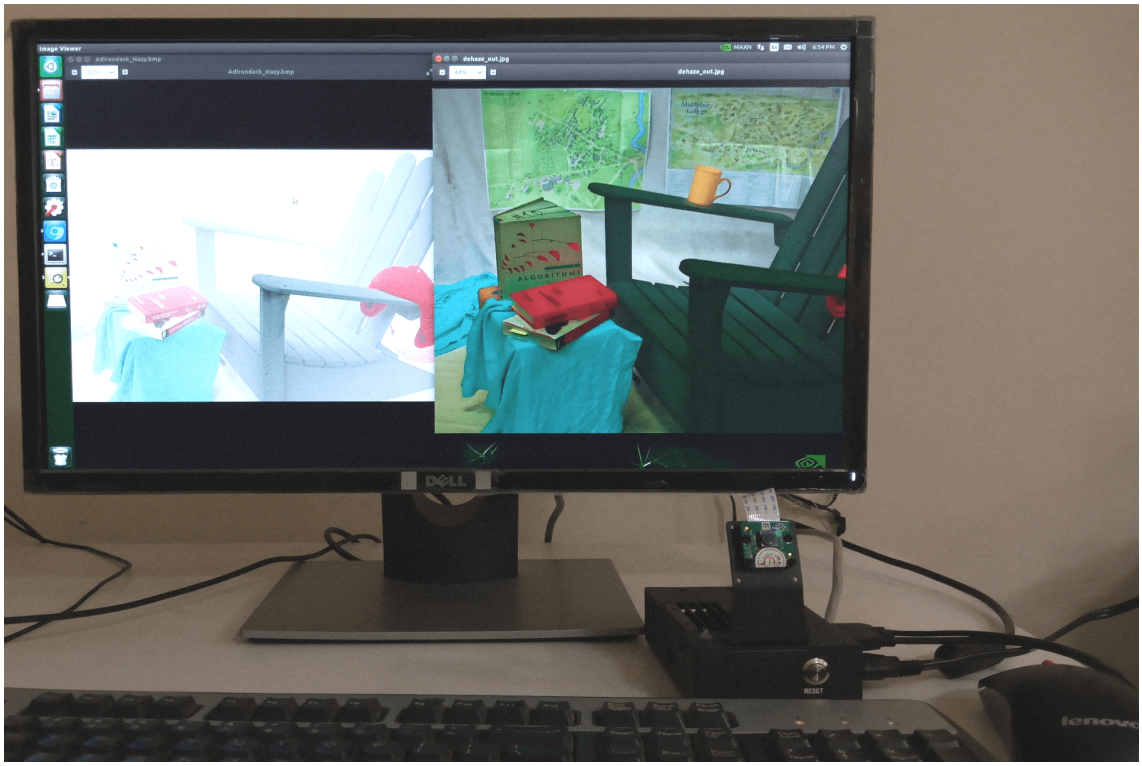


Figure 3.9: Implementation of real-time video dehazing on Jetson Nano

3.4.3 Observations:

When applied to each frame of a hazy video, it exhibits good results because we employed HSL based transmission and processed the only light component from the HSL color map without changing the hue and saturation components. The atmospheric light is calculated using RGB components from the median of the pixels. It is estimated for every scene changes; this is the only step in the proposed algorithm that takes significant processing time. Its computational complexity is $o(n^2)$, where n is the number of pixels in a frame.

Eq. 3.13 represents the condition for real-time image processing. If any image processing algorithm gets satisfied with this eq 3.13, it can be called real-time image processing.

$$Processing\ Time/Frame < Capturing\ Time/Frame \tag{3.13}$$

The proposed algorithm's running time in Raspberry Pi3 and Jetson Nano @ GPU for a frame size of 1080×1920 @ 24fps is approximately 39 milliseconds 17 milliseconds, respectively (1000 msec/24 frames= 41.66 msec/frame). Therefore, this method has low computational complexity and satisfies the condition. Hence, it is confirmed that proposed method is suitable

for real-time applications such as transport, car vision, or video surveillance. The proposed method potentially solves the major problem faced by Railways-ensuring timely and accident-free trains during haze time.

Figure 3.10 depicts off-line video frames, proving that our implemented design can work for all environmental conditions, i.e., sky and non-sky regions.

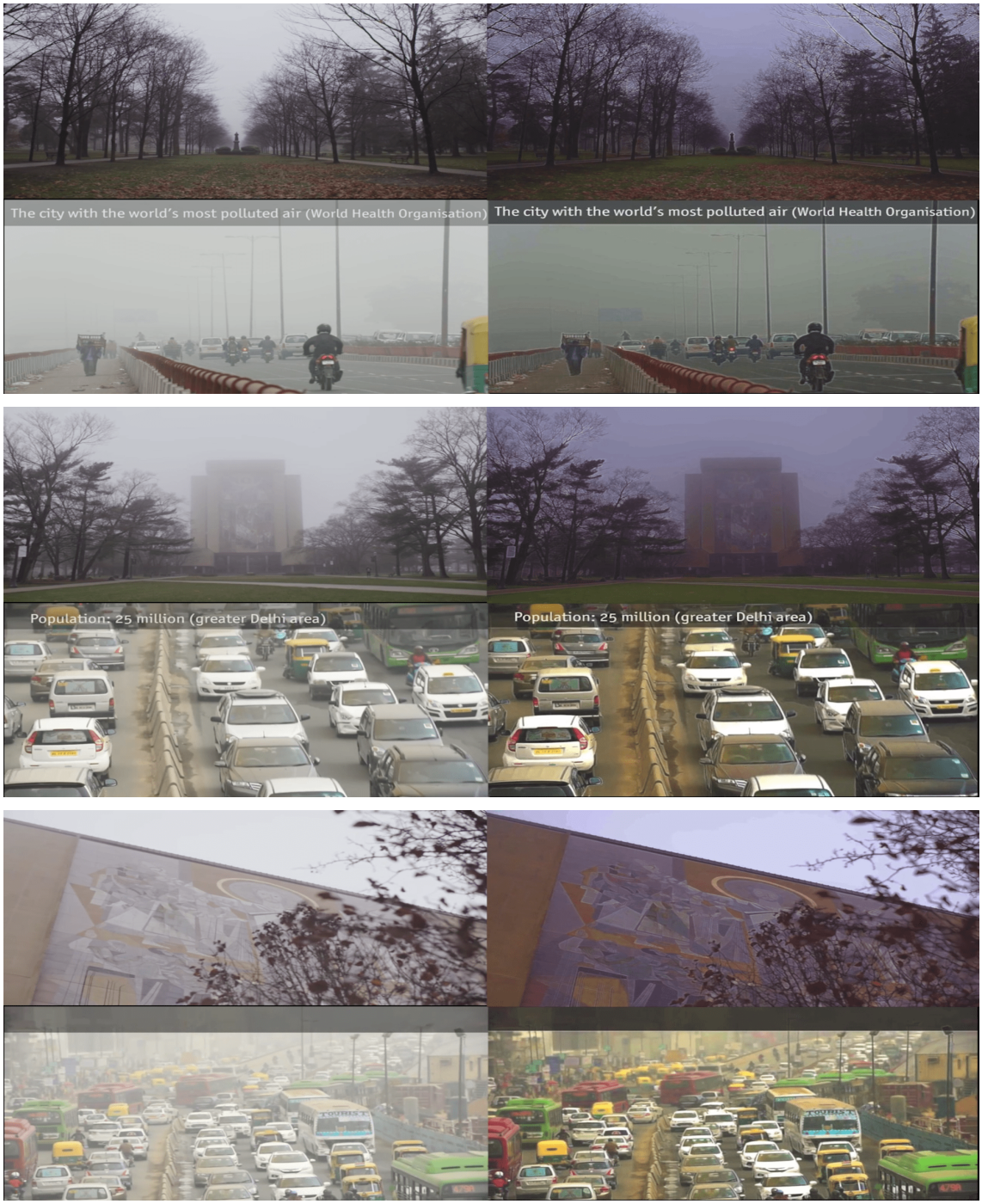


Figure 3.10: 1st, 115 and 120th Frames of the off line video (left side hazy frame & right side dehazed frame)

3.5 Summary

This chapter has proposed a new framework for single image de-hazing later extended to real-time video processing and mapped on Raspberry Pi 3 and Jetson Nano. We have presented three new contributions. Our first contribution refers to 'estimating the accurate transmission map using the HSL color model together with RGB color space. Second contribution is 'preserving the edges and avoiding halos & artifacts by employing the median of the pixels. Our last contribution 'real-time acceleration', is presented in Section 3.4 and confirmed by results shown in Table 3.2.

Our proposed algorithm may lead to a faithful interpretation of objects surrounded by a dense haze when dealing with images plagued by heavy, thick haze. The comparison experiments show that the proposed method can improve haze removal results by providing more accurate air lights.

Table 3.3: Comparison of proposed method in aggregated fashion.

Method	PSNR (dB)			ATC (Sec)			ACOI			MSE			PHI			SSIM		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Zhu et.al.	22.08	22.75	23.52	1.165025	1.725212	4.295092	0.0107	0.1905	1.4556	1.29	1.92	2.58	03.2373	12.6681	16.6764	0.5077	0.7339	0.7973
Tarel et. al.	22.08	22.73	23.47	1.653807	1.933738	2.885601	0.0123	0.3350	1.5482	1.51	2.06	2.75	01.2454	14.7474	48.5453	0.0217	0.1056	0.2270
He et. al.	22.05	22.95	23.35	1.964879	2.079973	2.660146	-0.0195	0.2937	1.9527	1.38	2.03	2.49	04.1131	12.5073	23.2702	0.0888	0.5852	0.8486
D. Berman et.al.	20.97	22.51	26.61	1.523481	2.555671	3.790123	0.6099	1.3461	2.1738	1.32	2.54	3.91	14.1248	21.8703	33.7685	0.5819	0.7042	0.7851
Cai et. al.	22.08	22.69	23.16	1.809674	1.834878	1.877650	0.0090	0.0848	0.1352	0.98	1.70	2.54	02.4292	10.7635	20.7800	0.2284	0.41215	0.5238
Proposed	22.10	23.91	28.97	1.052370	1.200356	1.715283	1.4617	1.6078	2.7532	0.87	1.41	2.35	12.1262	27.1289	48.7666	0.6312	0.8181	0.9465

Chapter 4

Implementation of a Novel, Fast and Efficient Image De-Hazing Algorithm on Zynq 706 and DSP TMS320C6748 Processor

This chapter focuses on two different implementation strategies to carry out the goal of hardware implementation. One method is on Zynq-706 FPGA and another on DSP processor (TMS320C6748).

4.1 Introduction

Improving the visibility of hazy images is desirable for robot navigation, security surveillance, and other computer vision applications. The presence of fog significantly damages the quality of the captured image, which does not only affect the reliability of the surveillance system but also produce potential danger. Therefore, developing as well as implementing a simple and efficient image de-hazing algorithm is essential. The reconfigurable computing devices like Field Programmable Gate Array and Digital Signal Processing (DSP) processors are used to implement these image processing applications.

4.2 Proposed Method

Based on atmospheric scattering theory, McCartney [63] has described a haze image model in terms of light attenuation and air-light. This model is commonly used to define the haze image formation and is given as

$$I(x) = J(x)t(x) + A[1 - T(x)] \tag{4.1}$$

where x is the image pixel coordinate, $I(x)$ is input hazy image, $J(x)$ is output de-haze image, A is air light, and $T(x)$ is transmission media. In (4.1) the product term $J(x)T(x)$ is known as direct attenuation, the second term $A[1-T(x)]$ is an additive component and termed as air-light. Three unknown components are present in the above equation. The basic idea of de-hazing is restoring $J(x)$ from $I(x)$ by estimating the parameters A and $T(x)$. Then the final dehazed image can be represented in (4.2)

$$J(x) = A + \left(\frac{I(x) - A}{max(T(x), t_0))}\right) \tag{4.2}$$

Using method [79] de-hazing can be done. But, this method suffers from high computation complexity owing to, wide-ranging matrix multiplication/division, sorting, and floating-point operations. In low-speed processors, this method cannot meet the user timing requirements for the real-time image processing applications. Therefore, an efficient and low-complexity haze removal method using pixel-based and gray image-based is proposed for real-time applications. Figure 4.1 represents the block diagram of the proposed algorithm based on the minimum number of computations.

4.2.1 Estimation of average channel

The DCP [79] method employed a minimum channel of the pixels as a dark channel of the image. As a result, it produces halos and artifacts in the de-hazed image. The effect of employing a minimum of pixels with a patch-based moment is represented in Figure 4.2. The white region becomes shaded, resulting in will lead to producing undesired pixel intensity which causes a halo effect.

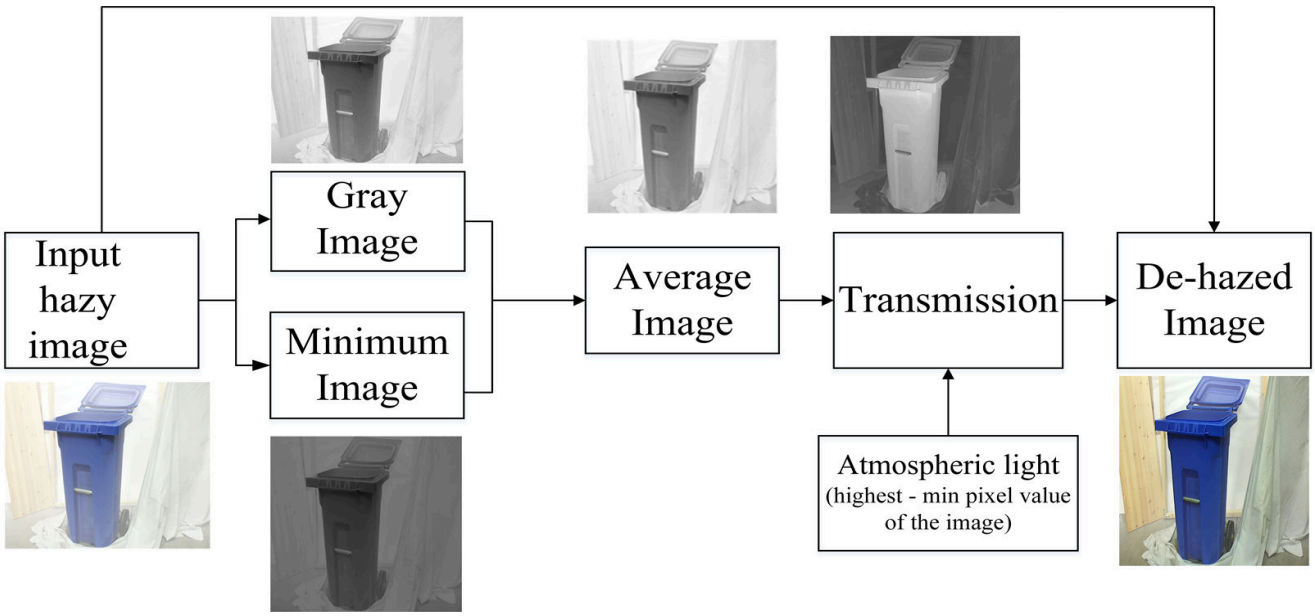


Figure 4.1: Block diagram of proposed algorithm.

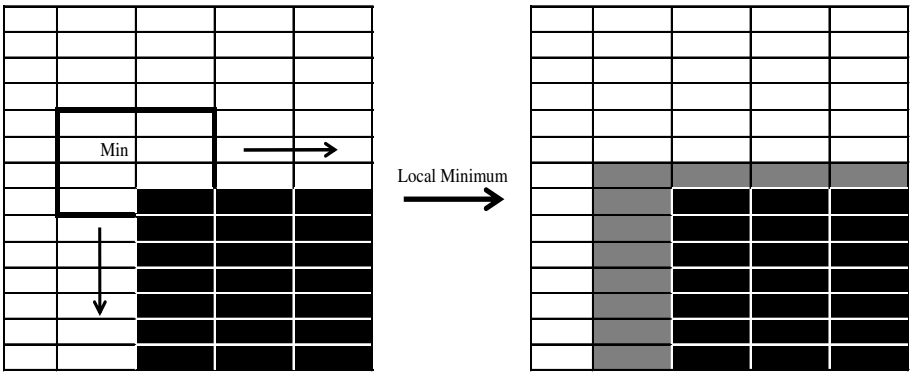


Figure 4.2: Halo effect.

$$I_{average} = \frac{I_{dark} + I_{gray}}{2} \tag{4.3}$$

The I_{dark} can be estimated using [79]. The I_{gray} can be estimated using (4.4). The mean of these can be termed as $I_{average}$ and it significantly reduce the halos and artifacts present in the final dehazed image.

$$I_{gray} = 0.3 * R + 0.59 * G + 0.11 * B \tag{4.4}$$

4.2.2 Estimation of Atmospheric light (A)

It is noticed that the value of the air light is always closed to higher pixel value within the image. By conducting several simulation experiments in MATLAB with different values of atmospheric light ranging from 0 to 1 for floating-point datatype and 0 to 255 for unsigned integer format (uint8), the atmospheric light value is assumed as maximum pixel value within the dark image. It can be calculated as "255 – *min*" for uint8 format and "1 – *min*" for floating-point format. Figure 3.2 represents the effect of variation of atmospheric-light on de-hazed images, it is observed that the flower database produces accurate de-hazing output at A=0.7, whereas table database at A=0.9.

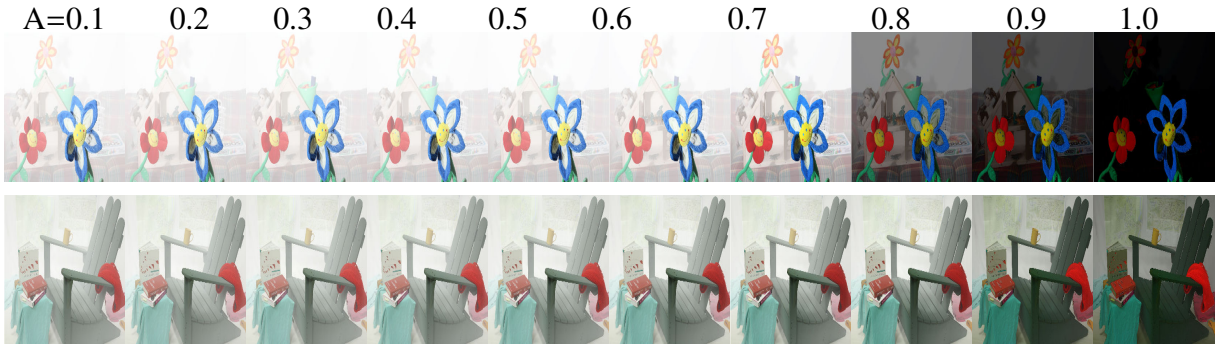


Figure 4.3: Variation of atmospheric light on hazy images (Flower, Table database).

4.2.3 Estimation of Transmission map (T(x))

The transmission can be estimated by normalizing the average image with atmospheric light. Later, recovered scene radiance can be obtained as a de-hazed image. The transmission map of various algorithms are visually compared and is shown in Figure 4.4. It is observed that our proposed method is estimated an accurate transmission map when compared with other existing algorithms.

4.3 Performance estimation:

The proposed algorithm is qualitatively and quantitatively compared in MATLAB 2017a with four existing state-of-art algorithms namely, Zhu et al. [115], Tarel et al. [116], He et al. [79] and Tripathi et al. [124] in terms of PSNR, ATC, ACOI, MSE, PHI, and SSIM for

Figure 4.4: Transmission depth map of different algorithms a.Zhu et al. b. Tarel et al., c.Our Proposed d. He et al., e. Tripathi et al..

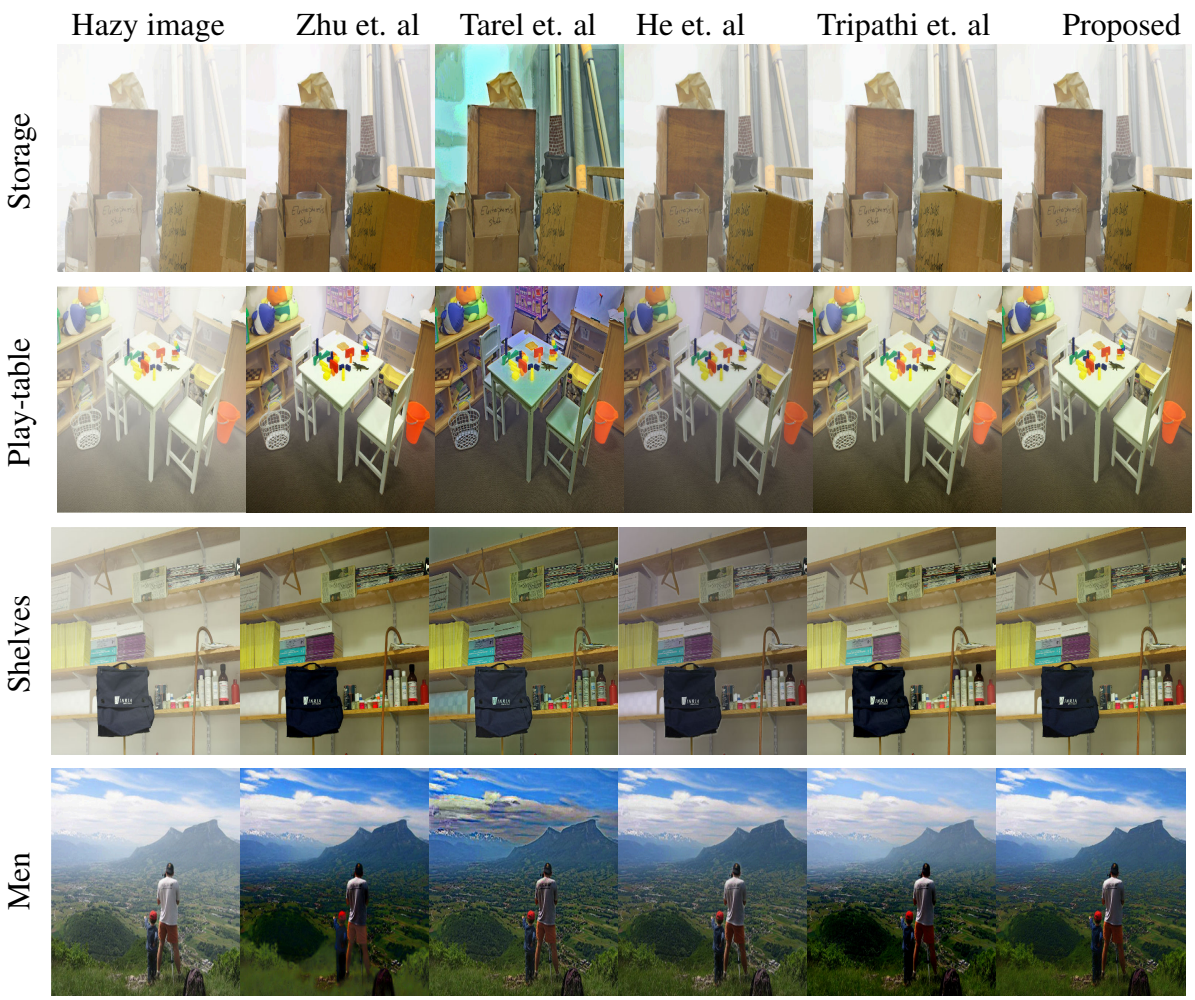


Figure 4.5: Simulation results of de-hazing techniques

an input image dimensions of 640×480. The comparison of quantitative metrics of the proposed algorithm is represented in Table 4.1 and corresponding visual comparison using qualitative metrics are represented in Figure 4.5. It is observed that the proposed algorithm takes an optimum amount of execution time. The SSIM value indicates that the proposed method produces good quality of the de-hazed images. The PSNR and PHI values of our method are high which indicating that haze particle elimination, as well as visualization of a de-hazed image is good. Thereby, it eliminates hallos & artifacts in the recovered output image, reduces the time-consuming computational step of refining transmission in DCP, and produces accurate transmission map.

Table 4.1: Comparison of quantitative metrics.

Database	Method	PSNR(dB)	ATC (Sec)	ACOI	MSE	PHI	SSIM
	Zhu et. al.	26.19	1.478520	0.0107	1.52	12.1976	0.6342
	Tarel et. al.	26.54	2.885601	0.6080	1.53	14.2584	0.0236
Men	He et. al.	45.87	2.038838	0.1258	1.78	10.4658	0.7676
	Tripathi et. al.	42.16	1.628883	0.5399	1.69	27.0457	0.6749
	Proposed	**46.54**	**1.155503**	**1.4617**	**1.34**	**27.3232**	**0.7736**
	Zhu et. al.	26.99	1.44128	0.0123	1.63	12.976	0.7612
	Tarel et. al.	26.99	2.885601	0.0123	1.63	10.3668	0.0482
Play Table	He et. al.	24.99	2.038838	0.0123	1.63	7.4004	0.6813
	Tripathi et. al.	26.76	1.628883	0.4578	2.35	25.5258	0.7423
	Proposed	**43.78**	**1.052765**	**1.2154**	**1.58**	**26.7324**	**0.8395**
	Zhu et. al.	23.14	1.455901	0.0488	2.83	1.3772	0.7642
	Tarel et. al.	24.19	1.813378	0.0487	2.43	0.4644	0.0125
Shelves	He et. al.	25.62	2.024570	0.0046	1.93	1.5700	0.8431
	Tripathi et. al.	43.27	1.631953	0.1752	1.44	4.5346	0.2355
	Proposed	**44.59**	**0.894809**	**1.4300**	**0.54**	**9.0500**	0.8337
	Zhu et. al.	25.76	1.418704	0.0329	1.61	3.2373	0.8166
	Tarel et. al.	25.57	1.906796	0.0341	1.76	6.6550	0.0917
Storage	He et. al.	42.61	2.029526	-0.0195	1.87	4.1131	0.8169
	Tripathi et. al.	45.82	1.619069	0.1238	1.53	5.9353	0.6730
	Proposed	**46.16**	**1.148978**	**0.6168**	**0.31**	**10.6060**	0.9139
	Zhu et. al.	24.73	1.428657	-0.0359	2.25	13.7933	0.8133
	Tarel et. al.	22.96	1.831700	-0.0343	2.79	13.1588	0.5378
Shopvac	He et. al.	28.58	2.020214	-0.0686	2.53	18.9791	0.8152
	Tripathi et. al.	24.47	1.588451	0.1011	2.05	**34.3493**	0.53757
	Proposed	24.95	**0.796819**	**2.0290**	**2.02**	3.6349	**0.9382**
	Zhu et. al.	25.11	1.393388	0.0448	1.75	16.6764	0.7069
	Tarel et. al.	45.72	1.845369	0.0964	1.82	7.6522	0.2151
Classroom	He et. al.	45.58	2.029721	0.0193	1.61	13.1743	0.8049
	Tripathi et. al.	45.09	1.602608	0.3582	1.54	18.9678	0.2151
	Proposed	**45.86**	**1.146288**	**1.8996**	**1.13**	**23.6419**	**0.9089**
	Zhu et. al.	43.50	1.525979	0.017	1.29	5.2107	0.7973
	Tarel et. al.	44.65	1.773976	0.0257	1.71	1.2454	0.0222
Umbrella	He et. al.	38.78	1.964879	-0.0265	1.95	6.6677	0.6674
	Tripathi et. al.	45.59	1.603365	0.2060	1.29	4.9916	0.1274
	Proposed	23.58	**1.295478**	**2.5145**	2.58	**9.7845**	**0.8121**
	Zhu et. al.	23.35	0.999786	3.8515	2.61	10.7208	0.6732
	Tarel et. al.	21.15	1.845405	3.0461	4.83	34.6196	0.1156
Library	He et. al.	44.74	2.388612	4.9670	2.11	20.1557	0.7420
	Tripathi et. al.	**44.83**	1.635244	2.4354	2.19	27.7894	0.1156
	Proposed	24.35	**0.853564**	1.4576	2.12	**15.7623**	**0.7702**

4.4 Hardware Architecture

4.4.1 Implementation on Zynq-706 FPGA

Since the existing state-of-art algorithms are difficult to implement on the hardware platform, the 14-stage pipeline structure based on Zynq-706 FPGA [125] for single image haze removal is designed. The implemented algorithm takes full advantage of powerful parallel processing and the ability to perform the same on the hardware platform. Figure 4.6 represents the hardware architecture of the proposed de-hazing algorithm on Zynq FPGA. Initially, the design includes RAM inference using MIF (Memory Initialization File), which specifies the initial content of a memory block. Three .mif files are sufficient for storing the input image data. With the help of a counter, the data which is stored in the .mif file can be invoked. In the proposed design, Each .mif stores 256×256=65536 pixels data. Therefore, A 16-bit counter is used for extracting the address of 64Kb .mif data. Then three 8-bit comparators are employed for finding minimum pixel among R, G and B, followed by a gray image of the input image using shift operations can be obtained using the following equation.

$$I_{gray\ modified} = 0.25 * R + 0.5 * G + 0.25 * B \tag{4.5}$$

The (4.5) is not the same as the traditional definition (4.4). Still, it makes almost no difference. It is much easier to be implemented in hardware design, to reduce the resource utilization, we employed a cut-off operation (shifting) instead of multiplication, thereby achieving a gray image of an input.

Later, a 10-bit adder is used and then truncating the LSB of the sum and carry bit so that an 8-bit data is given as input to further stages. Once the gray and dark channel is achieved, the mean of them using left shift operation is considered as the average channel of a hazy input image. Since the uint8 format of the image is employed, the value of atmospheric light calculated as the "255-minimum" pixel value of the input image. Moreover, the visibility content of the output is not much effected compared with the software platform. The normalization of the average channel to the atmospheric light results in a transmission map. An 8-bit divider circuit is employed for this operation. The above process is run in parallel for R, G, and B channels of the input image. Finally, using (4.2) modified pixel values are obtained as a de-haze image. The detailed 14-stage pipelining structure in the form of RTL of the proposed algorithm on the Zynq-706 platform using HDL is represented in Figure 4.7. In order to obtain the output in a

systematic manner, we introduced two new Intellectual Property (IP) cores, i.e., Virtual Input/ Output (VIO) and Integrated Logic Analyzer (ILA) these cores are responsible for verifying the de-haze algorithm. Figure 4.8 represents the top-level arrangement of IP cores with the de-haze algorithm. Once the algorithm is implemented on hardware, verification of the algorithm can

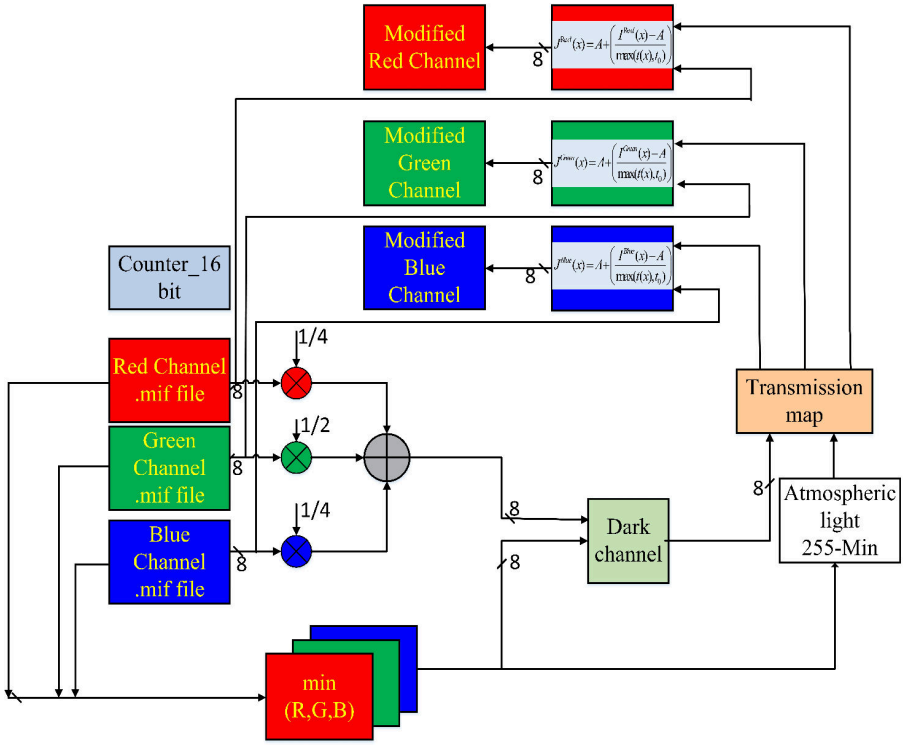


Figure 4.6: Proposed Hardware architecture of image de-hazing.

be done using VIO console. Figure 4.9 represents simulation results of the handwritten HDL code of image de-hazing. It can be observed that the first three signals within the waveform are input red, green and blue pixels respectively, the corresponding gray, dark channel and average channel of these pixels are indicated in the bottom of these waveforms. It is very hard to analyze for 256×256=65536 pixels. But, using the Tcl command the corresponding pixel values which are currently being executed on the FPGA can be exported to the spreadsheet, and qualitative analysis is verified in MATLAB.

4.4.2 Implementation on DSP Processor (TMS320C6748)

Based on DaVinci technology, an ideal core power for signal processing as well as image processing applications, the lowest cost DSP device has to choose. For which various kinds of DSP processors have been compared. Finally, TMS320C6748 [111] is chosen to be the core device of the system. The TMS320C6748 evaluation module is chosen as the hardware platform

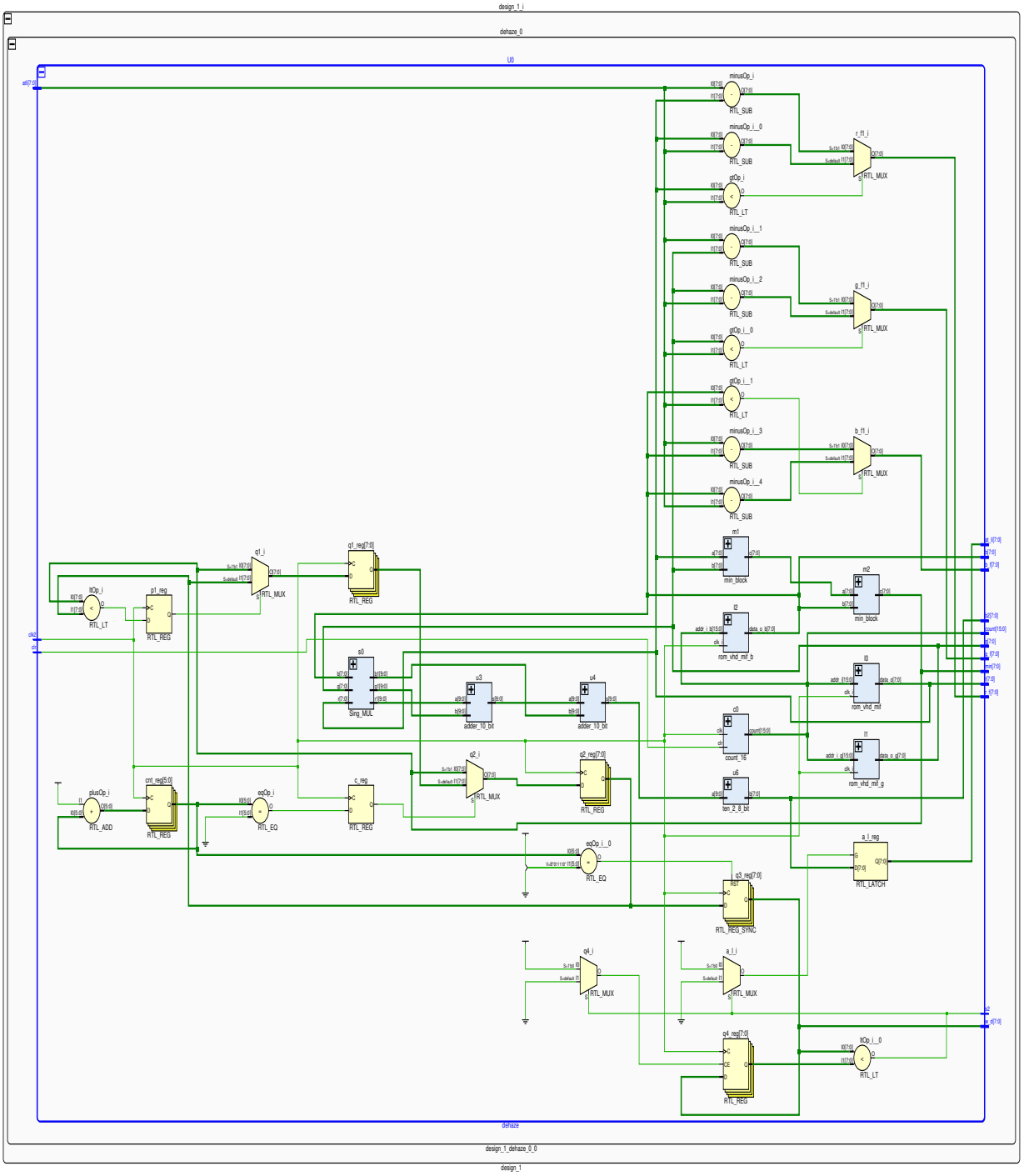


Figure 4.7: Schematic and elaborated design structure.

of data processing, and Code Composer V6.0 simulator [100] is chosen to implement simulation.

The hardware implementation flow of the proposed algorithm on TMS320C6748 is presented in Figure 4.10. In initial step, an input image is taken from the standard database of size

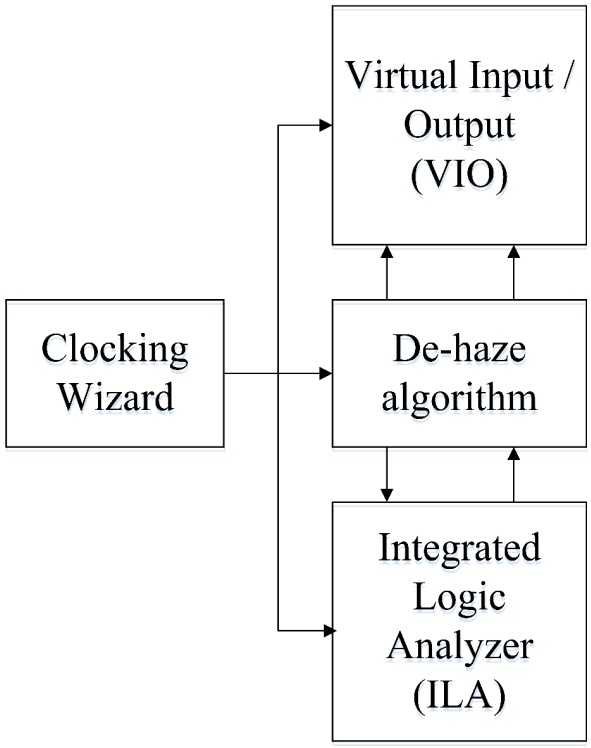


Figure 4.8: Test setup of de-hazing algorithm in Vivado.

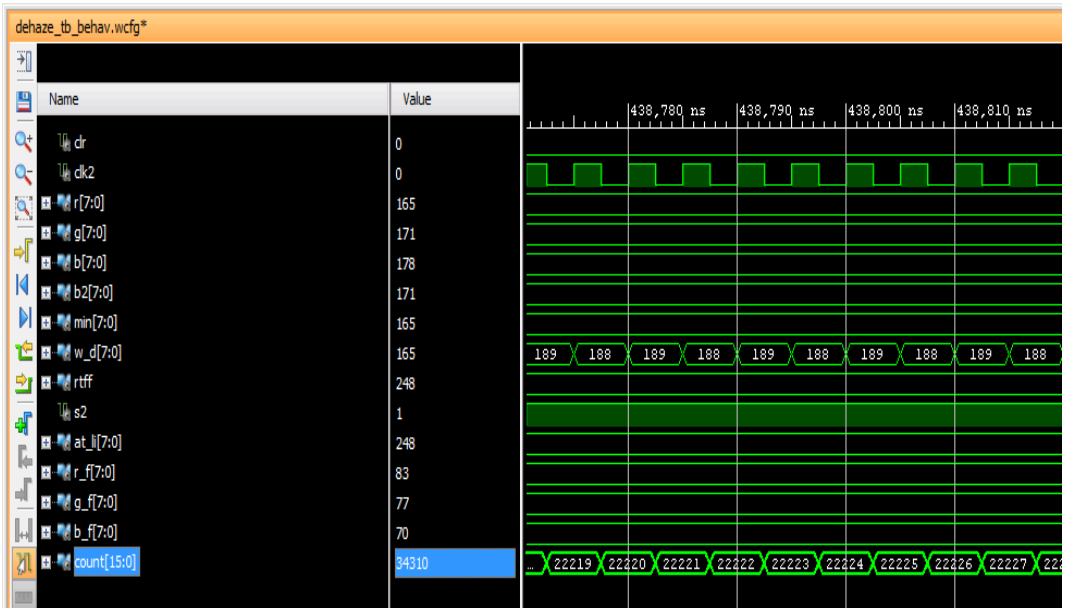


Figure 4.9: Simulation results of image de-hazing system.

256 by 256 for experimental verification. At the pre-processing stage, an image is cast into a standard datatype 'floating point (double precision)' format. The implementation of the de-haze algorithm in the code composer as follows. In the first step, we implemented a minimum filter for an image of M×N×3 pixels. Using (4.4) gray image of the input image is calculated. The overall cost O(2×M×N) time. Since the floating format of the image is employed, therefore, in transmission estimation each pixel value is escalating to 1. The value of atmospheric light

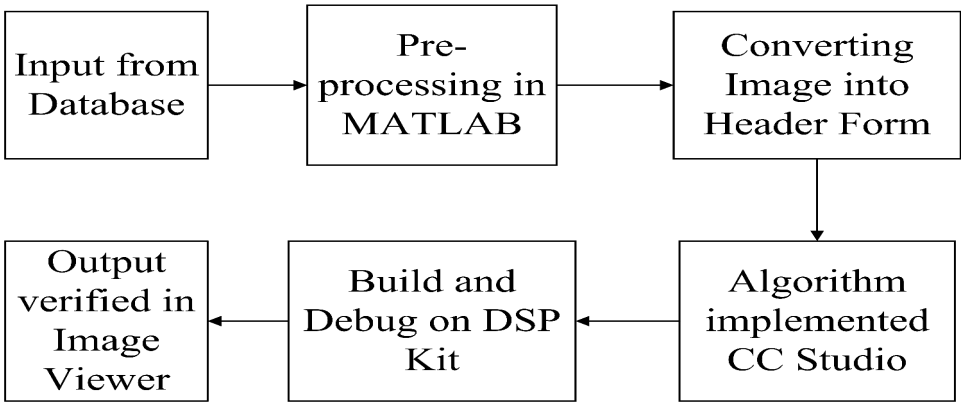


Figure 4.10: Flow of hardware implementation on TMS320C6748.

assumed as the "1-minimum" pixel value of the image. Figure 4.11 represents basic steps involved for implementing image de-hazing on DSP TMS320C6748 processor. The process is applied on each color channel individually and combined at the final step.

4.5 Results and Discussion

The validity of the proposed algorithm, which is implemented on two hardware platforms is verified using a well-known standard database of Middlebury [114] and from the Flickr website. To speed up the algorithm while implementing it on Zynq FPGA, the “uint8” format for representing the pixels is employed. Consequently, there is a negligible amount of loss in information resulting in the variation of the percentage of haze improvement. Whereas the floating-point operations are added as an advantage for obtaining better results in DSP processor-based implementation. In these two implementations, a refined transmission map (Guided filter in case of DCP) was eliminated, and this technique is helpful for disallowing the artifacts and hallows. The intensity variation between the haze and de-hazed image can be observed in Figure 4.12. Here red, green and blue lines represent the variation of RGB pixels of the image. It is observed that the brightness or intensity value of the de-hazed image is less compared with haze image pixels.

Table 4.2 indicates the speed of operation of the above-mentioned techniques. It is observed that the processing time required for executing the haze removal algorithm is optimal in two hardware platforms. The Zynq-706 FPGA with HDL based implementation took less amount of time when compared with DSP processor-based implementation. Since a 14-stage

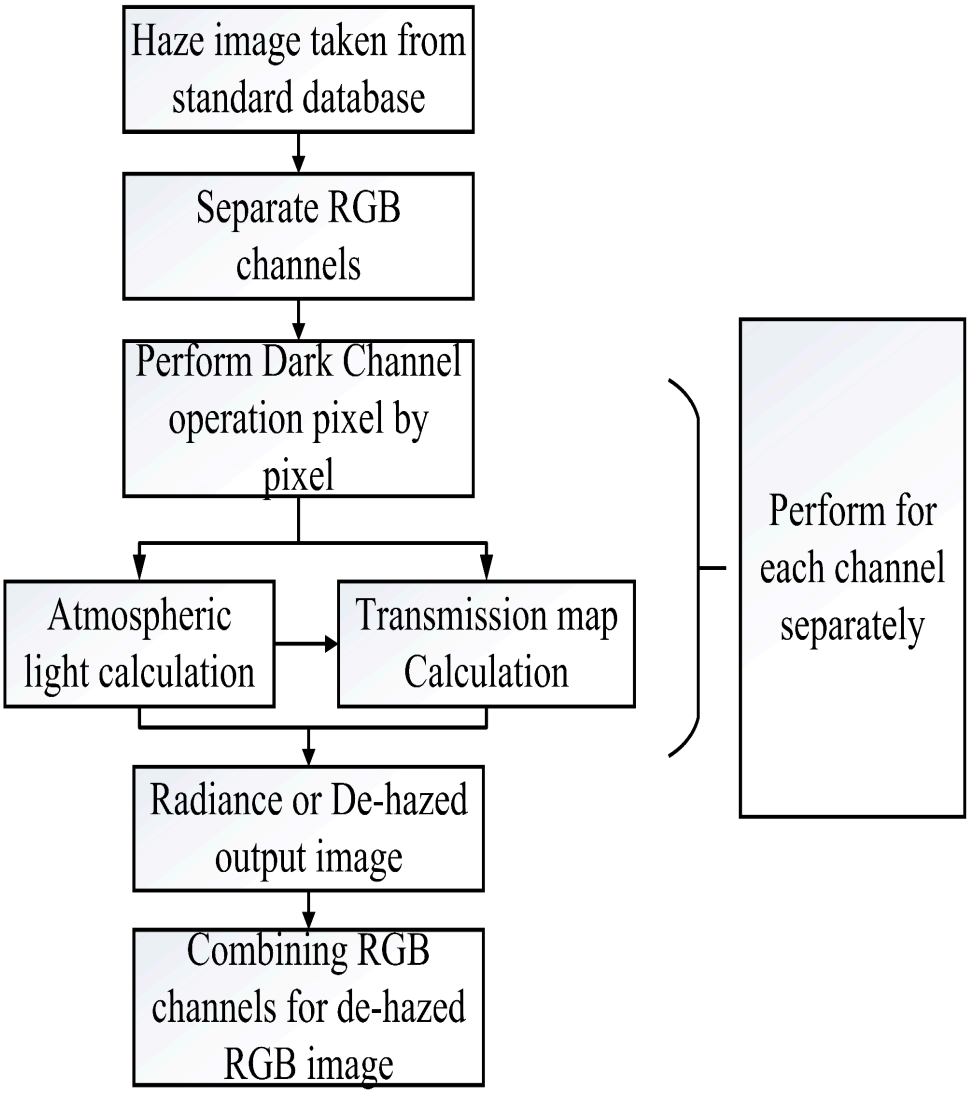


Figure 4.11: Basic Steps involved in hardware implementation of image De-hazing using TMS320C6748.

Table 4.2: Execution time on hardware platforms.

Parameter	On TMS320C6748	On Zynq-706 Using HDL Code
Max. operating clock frequency (MHz)	456	933
Throughput	333Mpixels/sec	385Mpixels/sec
Execution time	4.8 ns	3.59 ns

pipeline line structure is used, resulting in a number of clock pulses ($256 \times 256 \times 3 \times 14 = 2752512$) are required for de-haze operation theoretically.

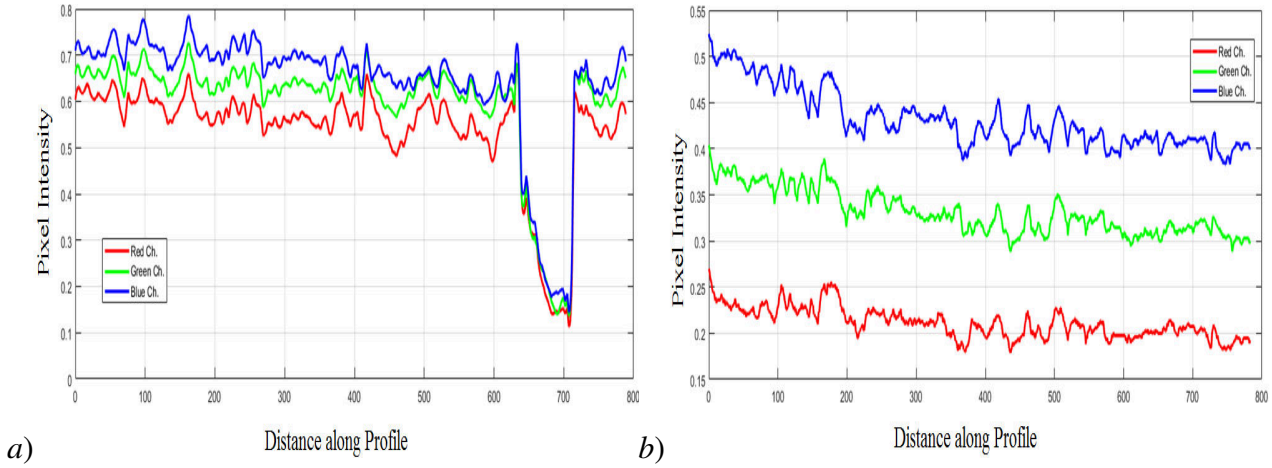


Figure 4.12: Sharp variation of brightness of the image pixels a. Haze image b.Dehazed image.

Table 4.3: Utilization summary of hardware resources.

Resources	Utilization % (HDL)	Available Resources	Resource Utilized
LUT	1.12	218600	2664
LUTRAM	0.3	70400	211
FF	1.63	218600	3562
BRAM	44.04	545	240
IO	0.55	362	2
BUFG	12.5	32	4

Hardware resource utilization summary of the proposed algorithm on the Zynq platform is presented in Table 4.3, It is noticed that the algorithm took less number of hardware resources. A very less number of other resources (LUT (Look Up Table), LUTRAM, FF (Flip-Flops), IO (Input Output), and MMCM) are utilized when compared with BRAM. Since storing the entire image or frame is not an efficient way of using on-chip memory (BRAM) resources on FPGA. Such a storage mechanism (.mif based) generally affects BRAM memory resources. This algorithm requires approximately 44 percentage of BRAM resources, which can be considered as a major challenging issue in hardware implementation. It can be overcome by storing a small portion of the image data inside the FPGA using line buffer instead of BRAMs, i.e., a portion of the image data or few selected lines of image data can be stored in a line buffer for manipulation and then, the remaining part of the image will be stored.

Figure 4.13 & Table 4.4 represents qualitative and quantitative comparison of de-haze algorithm respectively. Since the cutoff operation has employed for finding the gray image in HDL based implementation resulting in high MSE compared with processor-based implementation. PSNR and ACOI values are suggested that both hardware-based de-hazing algorithms produce a better result. When throughput is the user constraint, Zynq based hardware platform

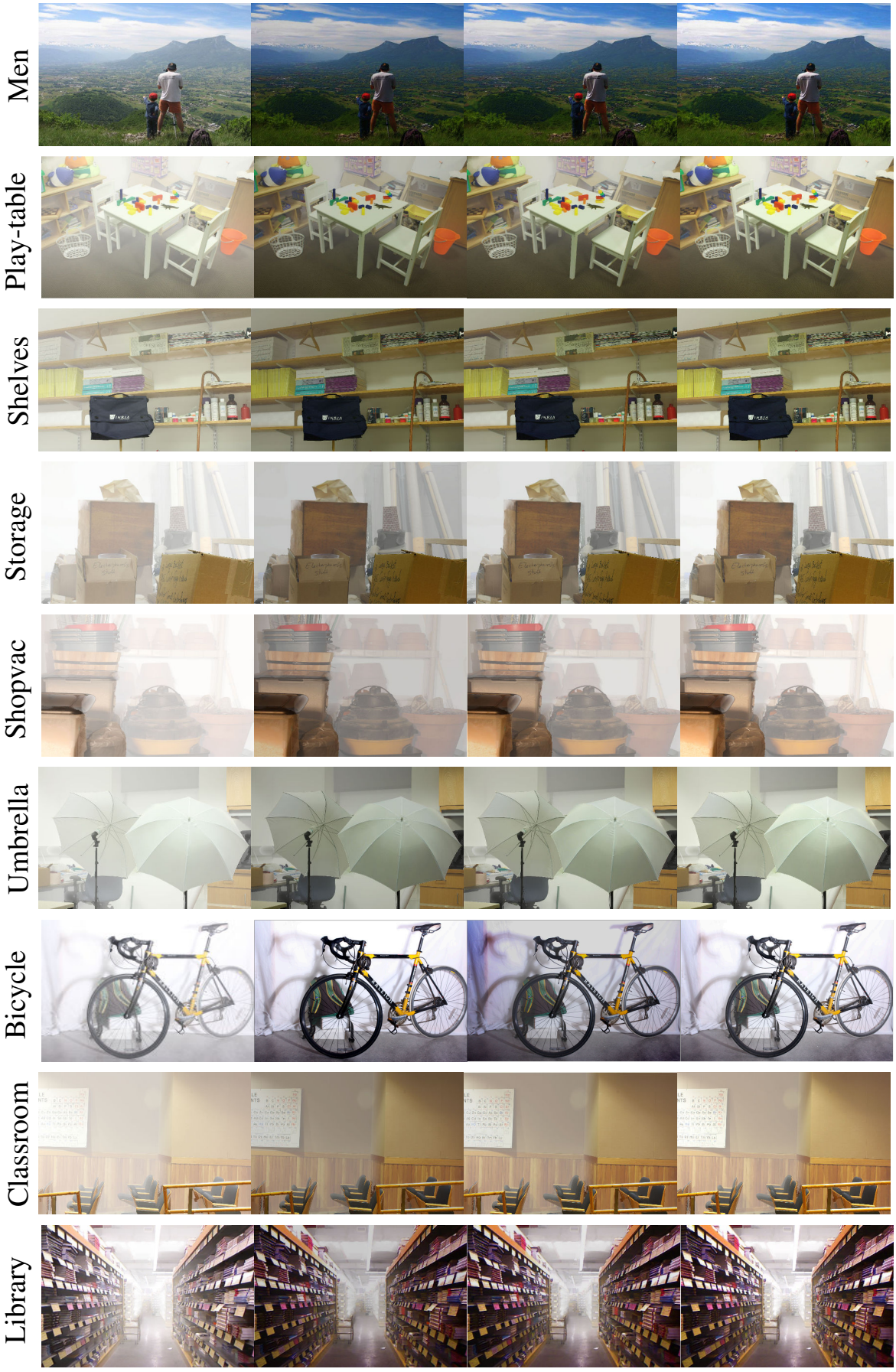


Figure 4.13: Implementation results of de-hazing system (Input hazy image, output on DSP processor, output on hand-written HDL based design, output of MATLAB from left to right).

Table 4.4: Quantitative performance of de-haze algorithm on different hardware platforms.

Database	Method	ACOI	MSE	PSNR	PHI
Men	DSP	0.4972	1.3467	46.8381	16.5585
	HDL	0.5000	1.5812	46.1409	15.4700
Play Table	DSP	0.4918	1.7845	45.6156	19.3389
	HDL	0.4887	1.9741	45.1771	14.3087
Shelves	DSP	0.4800	1.3590	46.7986	24.5587
	HDL	0.4984	1.5782	46.1491	23.4660
Storage	DSP	0.4725	1.6739	45.8935	37.8124
	HDL	0.4995	1.4528	46.5087	38.2875
shopvac	DSP	0.4952	1.5731	46.1632	38.7377
	HDL	0.2510	1.6328	46.0014	32.5731
umbrella	DSP	0.2036	1.3298	46.8929	16.8774
	HDL	0.2037	1.5439	46.2446	15.0265
Bicycle	DSP	0.4512	1.5328	46.2759	29.7640
	HDL	0.3191	1.6529	26.1903	21.2604
classroom	DSP	0.5000	1.5543	46.2154	16.0200
	HDL	0.4514	1.8910	45.3638	08.3962
Library	DSP	0.4000	1.5342	46.2719	27.4581
	HDL	0.2608	1.6124	46.0560	11.7153

Figure 4.14: Hardware Platform Test Setup; Left: on DSP Processor TMS320C6748, Right: on Zynq 706 All programmable SoC.

is well suited and if the quality of the output image requirement is the user constraint then DSP processor-based hardware implementation is well suited for de-haze applications. The hardware test setup of the proposed algorithm is represented in Figure 4.14.

4.6 Summary

This chapter presented a fast and efficient de-hazing algorithm well suited for implementing on any hardware platform. The entire work is carried out on i5 CPU 3 GHz and 8GB memory with OS window 8.1 with MATLAB 2017a. Vivado version of 2015.4 and Code composer studio v6.0 are used for implementation. A Zynq (xc7z045ffg900-2) based hardware architecture, and DSP processor-based image de-hazing algorithms are implemented successfully.

Chapter 5

Design of an Embedded Single image de-hazing system using YCbCr color format on Zynq 702

In this chapter of the thesis, a new approach for efficient development of image de-hazing system is explained. Initially a memory efficient algorithm is developed in RGB color format. Later, it is further optimized with respect to the memory using YCbCr color format and implemented on ZYNQ 702 all programmable SoC using Hardware/Software co-design approach.

5.1 Introduction

Memory is the major limiting factor in the field of image processing when wish to implement on FPGAs. Let us consider a simple image processing operation used in most of the algorithms called convolution operation [126].

The convolution is an important operation on an image and is performed in the spatial domain. Where each pixel and its neighbors are multiplied by kernel or patch, the results are summed up and the result is used to measure the new pixel value. The process is represented in Fig. 5.1. This exhibits a pattern of local computations due to the dependence of a restored pixel value on its 1-neighbors. The same process is repeated on a significant amount of data for a number of iterations. There have also been some difficulties related to this algorithm when we wish to implement it on hardware. The process is also in sequential manner, i.e. until the next

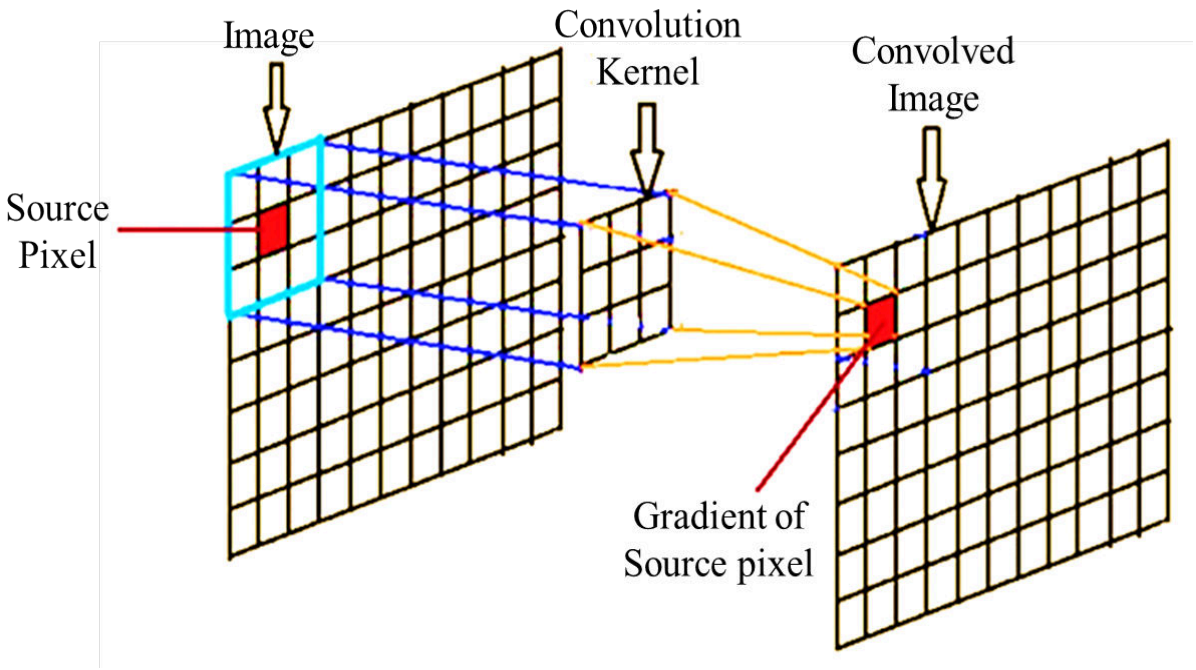


Figure 5.1: Convolution operation on Sub-window of the frame (X)

iteration step can begin, all pixels of the image will move through the current restoration stage. This requires a fast and wider bandwidth.

Hence, in this chapter we focused on eliminating the refine transmission step (for reducing the computational complexity) and optimized memory management of the pixels for efficient implementation of image processing algorithms on hardware platforms.

5.2 Methodology

The DCP [79] employed the minimum channel of the pixels as a dark channel of the image. As a result, it produces halos and artifacts in the de-hazed image. To remove this effect the estimated transmission map need to be further refined, thereby total computations are increased.

This problem can be resolved in our proposed method as, in Figure. 5.2, the points on diagonal line in the RGB cube ranges from (0, 0, 0) to (255, 255, 255). Foggy areas (Brighten pixels) in an image are mostly scattered nearby this diagonal line, indicated with blue points in Figure.4. Whereas object related pixels are surrounded nearer to the origin which are indicated with pink points. There is less special continuity. By determining variogram (V) [127] for each pixel of the hazy image, we can get the spatial continuity of an original image. Generally, in

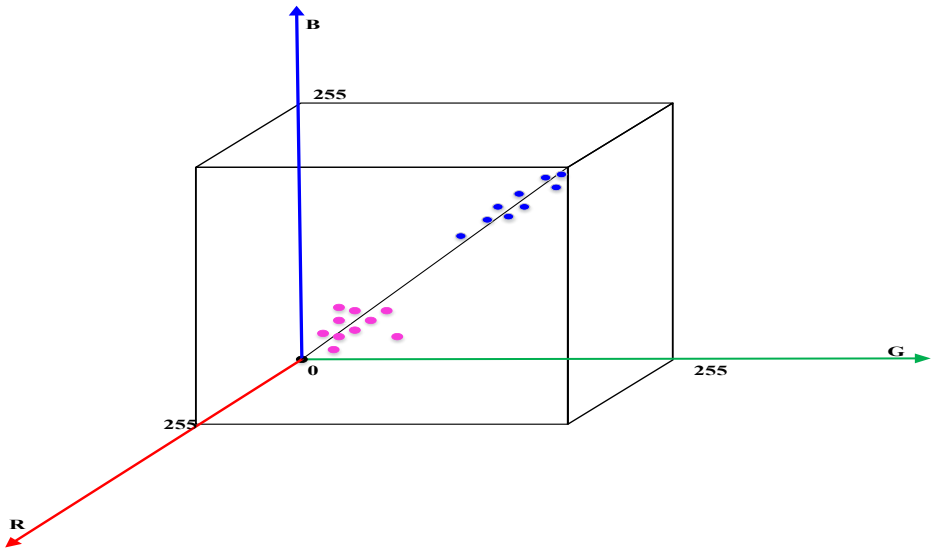


Figure 5.2: RGB color space of hazy image model

sky regions, the value of 'V' is very small. However, in non-sky areas, 'V' is larger. To measure 'V', we used the concept of color fluctuations. According to the depth of field, bright objects will become dimmer as the distance increases. This phenomenon is more observable under the impact of thick haze. The pink points in Figure. 5.2 shows, an object's color is unsteady at a distance. To stretch the value of "V" and make it more effective we employed the median of the pixels.

Another key aspect proposed in this chapter is about memory. To address this issue, assume that the ' X ' image or frame will be stored as an array (A) of 8-bit elements. The 'A' array has columns and rows and is usually stored in BRAM [128] like a row by row or a column by column manner (ixj). Generally, the computer or processor does not keep track of each element of an array's address. Relatively, it keeps track of address of the first pixel and is called the base address. It can use measure the specific pixel's address .

$$Loc\left[A\left(i,j\right)\right] = Base\_adrs + \left(no\_of\_Columns(or)rows \times i\right) + j \tag{5.1}$$

The moment of mask or patch in the image is varying from top to bottom as well as left to right end. In a particular row, when elements are finished, the mask moves one position right and down. Therefore, nine-pixel numbers (if patch size-3x3) are involved during the convolution process at each step. Sequentially, it extracts nine pixels in each step. Because a single

pixel's data is replicated, therefore this strategy is costly in terms of memory storage. Hence, a sequential pattern of pixels employed like a row vector manner. Most exceptional de-hazed output can be obtained by combing the above light of ideas. The process of proposed method as: Initially, an image is converted to a row vector model. As a result, the square patch will be changed as rectangular. Figure 5.3 indicates the patch moment in existing and proposed algorithm respectively.

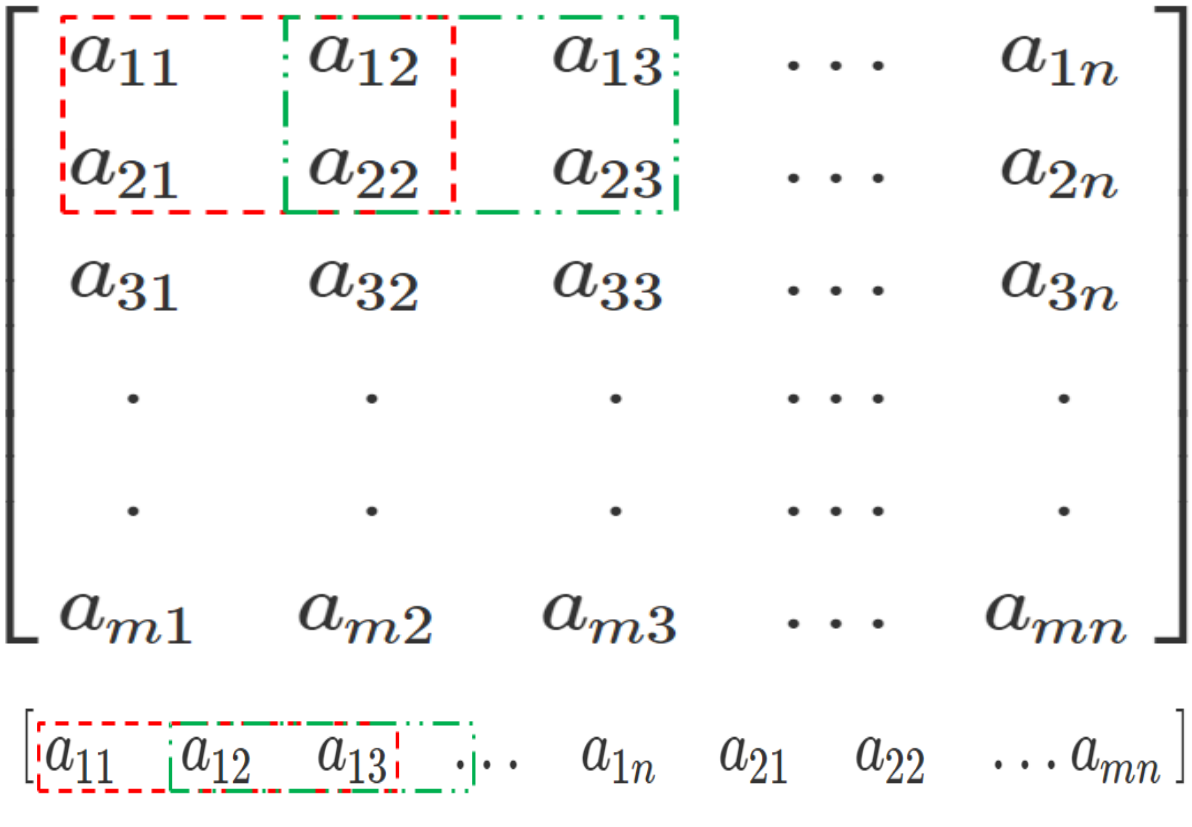


Figure 5.3: Patch Moment. a. Square b. Rectangular

When observing transmission maps with different window sizes, we find a tip to eliminate the refine transmission step as well as halos and artifacts in the recovered image. It can be done by estimating the median channel of the pixels instead of a dark channel. Furthermore, note that in DCP produces a good de-hazed image with a patch size of 16x16. Therefore, in the proposed method, the median channel is determined with a 16x1 patch size.

In the next step, the median [129] of the pixels is employed as a dark channel. Since the information of any image will hide in the middle of the image so, instead of finding the minimum of pixels in [79] we proposed the median. Median is nothing but middle value or average of the central values within the list. It is a non-linear signal processing technique based on statistics. The noisy value of any digital image or the sequence is replaced with its median

value of the neighborhood (mask). Generally, noise elimination is employed at a pre-processing step to improve the outcomes in post-processing (ex: Edge detection). The pixels within the mask are ranked in the order of their gray levels, and the median value of the group is stored to replace the noisy value. The median of the pixels can be obtained as

$$M(x,y) = med\{I(x-i,y), i\epsilon W\} \tag{5.2}$$

Where $M(x,y)$, $I(x,y)$ are output de-noise image and the original noisy image respectively, W is a single-dimensional mask. Sometimes the mask shape may be linear, circular, cross and square. In our method, we considered linear mask with size W=16×1. Then, equation (1) can be modified as

Step 1: Taking the median in the local patch in the hazy input image individually on each R, G&B channels from equation (1)

$$\underset{x\epsilon\Omega(p)}{median}\left(I^c(p)\right) = \widetilde{t(p)} \ \underset{x\epsilon\Omega(p)}{median}\left[J^c(p)\right] + A^c\left(1-\widetilde{t(p)}\right) \tag{5.3}$$

Step 2: Applying normalization with air-light then equation (8) can be re-written as

$$\underset{p\epsilon\Omega(p)}{median}\left(\frac{I^c(p)}{A^c}\right) = \widetilde{t(p)} \ \underset{p\epsilon\Omega(p)}{median}\left(\frac{J^c(p)}{A^c}\right) + \left(1-\widetilde{t(p)}\right) \tag{5.4}$$

Step 3: Obtaining the minimum value of pixel from three RGB color channels as

$$\overset{min}{c}\left(\underset{p\epsilon\Omega(p)}{median}\left(\frac{I^c(p)}{A^c}\right)\right) = \widetilde{t(p)} \ \underset{c}{min}\left(\underset{p\epsilon\Omega(p)}{median}\left(\frac{J^c(p)}{A^c}\right) + \left(1-\widetilde{t(p)}\right)\right) \tag{5.5}$$

We considered an assumption according to the DCP, minimum value of median of three color channels approximated to zero. This can be represented as

$$J^{dark}(p) = \overset{min}{p\epsilon\Omega(p)}\left(\underset{c\epsilon\{r,g,b\}}{median}\left(J^c\right)\right) \cong 0 \tag{5.6}$$

Step 4: The patch transmission obtained as

$$\widetilde{t(p)} = 1 - \min_{c}\left(\underset{p\epsilon\Omega(p)}{median}\left(\frac{I^c}{A^c}\right)\right) \tag{5.7}$$

The existence of haze is a basic indication for a human to identify depth. Even, if the pictures are captured in good weather conditions, the image will appear to be abnormal if the entire haze is removed and it seems to depth may be lost.

It can be defeated by adding a small quantity of fog (ω =0.25), with the constant parameter ($0< \omega \leq 1$) in the above equation.

$$\widetilde{t(p)} = 1 - \omega \times \min_{c}\left(\underset{p\epsilon\Omega(p)}{median}\left(\frac{I^c}{A^c}\right)\right) \tag{5.8}$$

Here the combining patch's transmission is equal to transmission depth of the entire image. Once, the transmission and air-light is known the scene radiance J(p) can be determined as

$$J(p) = \left(\frac{I(p) - A}{max\left(t(p), t_0\right)}\right) + A \tag{5.9}$$

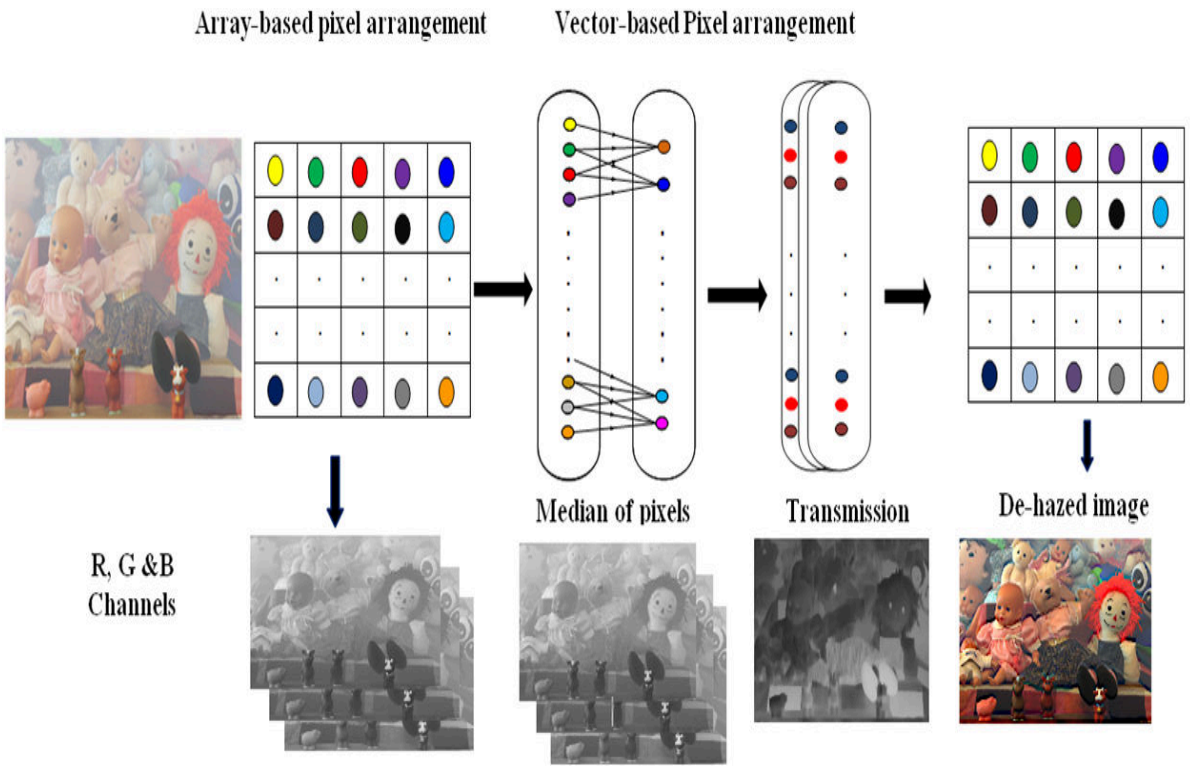


Figure 5.4: Architecture of proposed algorithm

The above whole process is done in the row vector model. At last, after finding the scene radiance for a row vector, it is reshaped into the matrix (image) form. The architecture of the

proposed algorithm with array-based pixel arrangement and corresponding rectangular patch moment is represented in Figure 5.4. To understand the variation of the image at each succeeding step, we demonstrated the architecture with images. Figure 5.5, represents the block diagram. Initially, matrix-based pixel arrangement converted to the vector-based manner and then with a patch size of 16x1 the median of pixels is obtained. Later, atmospheric light, transmission map, and scene radiance are predicted. Finally, vector-based arrangement converted to a matrix-based manner for visualization purposes.

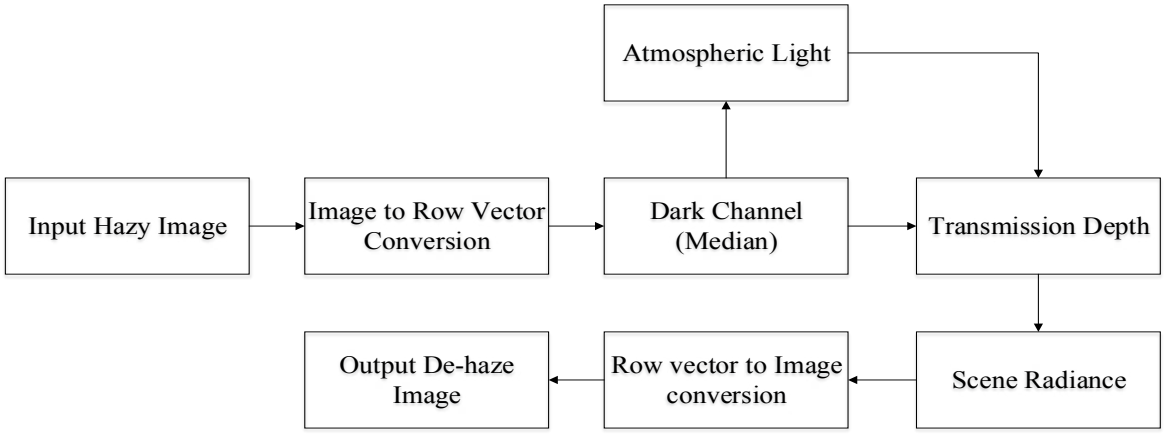


Figure 5.5: Block diagram of proposed algorithm

5.3 Results and discussion

The simulations are performed in MATLAB 17a running on Intel i5 CPU 3 GHz and 8GB memory with OS window 8.1. To verify the validity of our proposed algorithm, we choose a well-known standard database Middlebury [114]. This database covers all the challenging factors. The proposed algorithm compared for an image dimensions of 640x480 with a patch size 16x16. They are Zhu et. al [115], Tarel et. al [116], He et. al [79] and Tripathi et. al [124] in terms of MSE [119], PSNR [119], average time cost, average contrast of the output image and visibility analysis (percentage of haze improvement, Structural Similarity Index [120]).

From the comparison table 5.1, the proposed approach taking minimum average time cost; produces output with high contrast and equal mean square error (MSE) i.e., for canon image the MSE is 2.07 similar for other methods and haze improvement of the proposed algorithm gives a significant result. In the motorcycle database, He et. al and Tripathi et. al produces halos

Table 5.1: Comparison of proposed algorithm with existing methods

Database	Method	PSNR in dB	ATC in Sec	ACOI	MSE	PHI	SSIM
	Zhu et. al.	22.99	1.44128	0.0123	1.63	12.1976	0.7612
	Tarel et. al.	22.99	2.885601	0.0123	1.63	10.3668	0.0482
Play table	He et. al.	22.99	2.038838	0.0123	1.63	7.4004	0.6813
	Tripathi et. al.	21.76	1.628883	0.4578	2.35	25.5258	0.7423
	Proposed	23.95	1.597643	1.4617	1.98	29.5147	0.9882
	Zhu et. al.	23.42	1.857462	0.1238	2.07	15.2154	0.6270
	Tarel et. al.	23.42	1.258642	0.0341	2.07	14.3258	0.1042
Canon	He et. al.	23.42	1.647523	-0.0195	2.07	19.2158	0.6994
	Tripathi et. al.	23.44	1.525874	0.6560	2.07	22.2174	0.8910
	Proposed	23.52	1.104532	1.7119	2.07	23.2587	0.9881
	Zhu et. al.	22.08	4.295092	0.0495	2.49	16.0677	0.7898
	Tarel et. al.	0.765	1.750527	0.6080	2.49	43.5453	0.1320
Vintage	He et. al.	22.05	1.987452	0.1857	2.54	16.5410	0.2365
	Tripathi et. al.	25.18	1.841230	1.8547	2.49	11.2654	0.4163
	Proposed	25.92	1.325487	2.7458	2.49	18.9784	0.9955
	Zhu et. al.	23.16	1.418704	0.0329	1.51	3.2373	0.6854
	Tarel et. al.	23.16	1.906796	0.0341	1.51	6.6550	0.2098
Motor Cycle	He et. al.	23.16	2.029526	-0.0195	1.51	4.1131	0.4782
	Tripathi et. al.	23.16	1.619069	0.1238	1.51	25.9353	0.5843
	Proposed	24.58	1.597488	2.1254	1.03	20.3847	0.9889
	Zhu et. al.	25.18	1.425781	0.0478	2.42	18.1245	0.8408
	Tarel et. al.	25.18	1.754215	0.7584	1.59	18.4758	0.2579
Flowers	He et. al.	25.18	1.914254	-0.2548	2.42	17.4789	0.8638
	Tripathi et. al.	25.18	1.574123	0.9954	2.42	16.2587	0.6587
	Proposed	25.72	1.072587	1.8222	2.42	19.0012	0.9953
	Zhu et. al.	28.64	1.547842	0.1258	1.55	16.6764	0.7226
	Tarel et. al.	28.67	2.754218	0.0258	1.55	7.6522	0.2093
Play Room	He et. al.	28.67	2.147896	0.1458	1.55	13.1743	0.7910
	Tripathi et. al.	28.67	1.785421	0.9985	1.55	18.9678	0.8425
	Proposed	28.52	1.058414	1.5285	1.55	23.6419	0.9893
	Zhu et. al.	25.45	1.574562	0.017	1.29	11.1458	0.7239
	Tarel et. al.	25.45	1.975125	0.0257	1.29	16.1249	0.1575
Toys	He et. al.	25.45	1.587452	-0.0265	1.29	18.1246	0.8480
	Tripathi et. al.	25.45	1.254789	0.206	1.29	20.0128	0.5483
	Proposed	25.45	1.160618	1.4858	1.29	22.0196	0.9870
	Zhu et. al.	22.08	1.472224	0.0382	2.49	16.3358	0.7719
	Tarel et. al.	22.08	1.898116	0.1195	2.49	15.7125	0.2270
Bicycle	He et. al.	22.08	1.987793	0.0059	2.49	23.2702	0.7494
	Tripathi et. al.	22.07	1.542246	0.608	2.49	28.5453	0.7675
	Proposed	22.58	1.452662	2.7532	2.19	29.2658	0.9928

and artifacts. Whereas, in the proposed method we employed the median of the pixels along with row-based patch moment, which performs well on removing artifacts and halos around the depth discontinuities. Zhu et.al produces contrast-enhanced output and they look unnatural.

Tarel et. al produces natural de-hazed as an output but recovered image has noticeable haze. Proposed method has highest SSIM value among other state-of-the-art methods. These evaluations parameters are more trustworthy for analysis. and the bold value in the Table 5.1 indicates the better evaluation metric value produced by the corresponding method.

An example of a bi-cycle image from a standard database chosen for illustration, Figure **??**

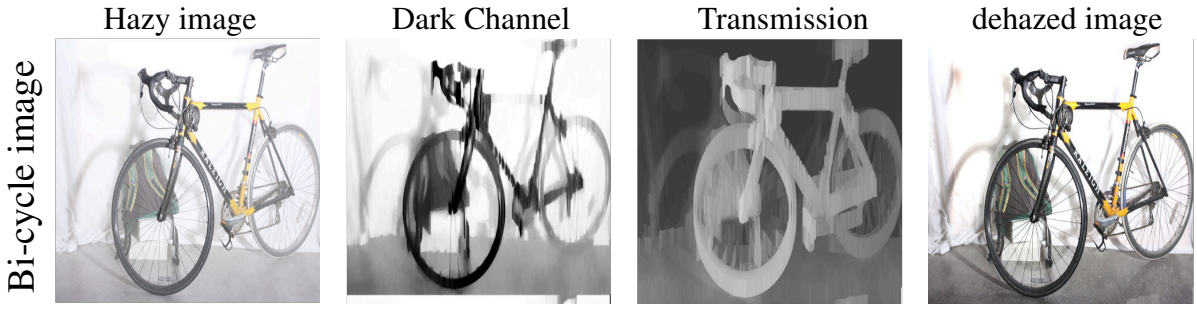


Figure 5.6: Result of proposed method.

denotes a bi-cycle hazy image with a resolution of 800x600.

Figure 5.7 shows the qualitative comparison of five state-of-the-art algorithms on different real-world images.

This idea significantly reduces the computational complexity and halos & artifacts produced in the dehazed image compared with existing algorithms. Since this method is based on RGB color format, the computational time is still high, and it can be further reduced by changing the color format to (Luminance, blue-difference Chroma and red-difference Chroma) YCbCr [130].
As the RGB color space always requires considerable amount of memory to store. Whereas, in the case of YCbCr, the chroma components are horizontally sub-sampled, and their resolution is halved compared to the luminance counterpart. Therefore, this YCbCr color space is mostly employed in high-end digital video formats and still images.

The key contributions of hardware implementation of proposed algorithm are listed below:

- We employed YCbCr (4:2:2) color model to reduce the memory requirements.
- In order to reduce the number of operations, the luminance component of the haze image is considered as a transmission map.
- The PL part was implemented using IP cores, and its application code is written in Xilinx SDK. Finally, validated on Zynq 702 FPGA.

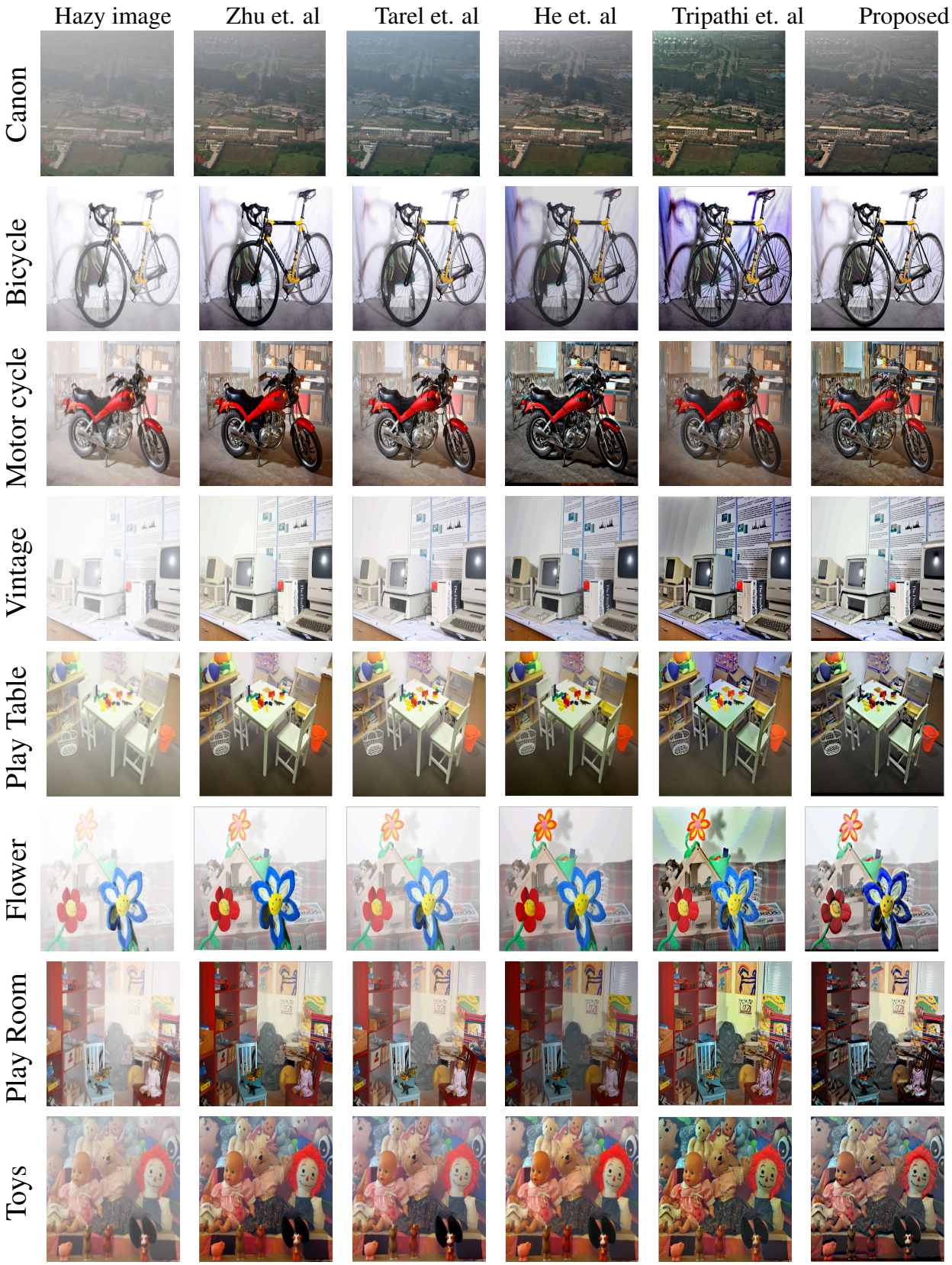


Figure 5.7: Qualitative comparison of different Dehazing algorithms on Middlebury standard challenging database

The Fig.4 represents the block diagram of proposed method The following steps can be

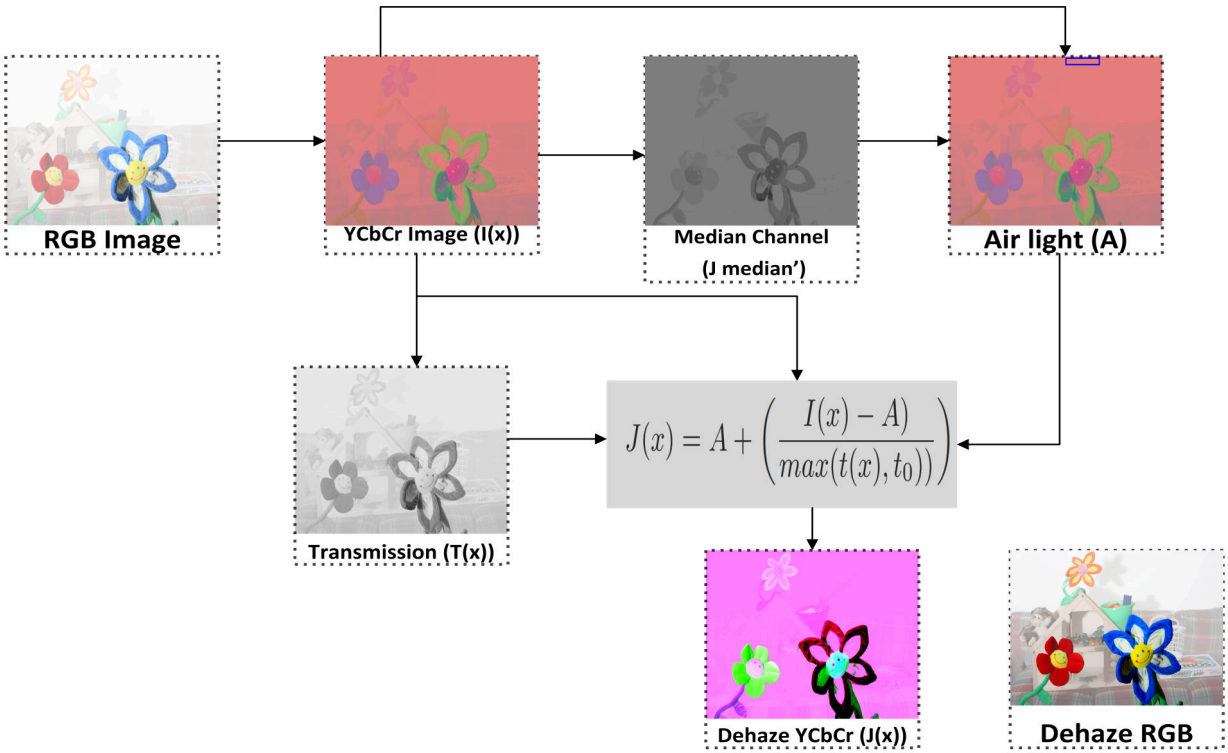


Figure 5.8: Block diagram of proposed method

used to estimate the de-hazed image in the proposed method.

Step 1. Conversion: Converting RGB color space into YCbCr color space using equations 5.10, 5.11 & 5.12

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \tag{5.10}$$

$$Cb = 0.564(B - Y) \tag{5.11}$$

$$Cr = 0.713(R - Y) \tag{5.12}$$

Step 2. Median Channel: We have made an extensive and comparative work and finally got a conclusion that most of the haze-free image has pixels, each pixel is further represented by three color channels namely Y, Cb & Cr out of which one of the channels should have minimum intensity it can be obtained by taking the median of the pixels. The Eq. 5.13 represents the median channel

$$J^{median'} = \underset{x \epsilon \Omega}{median} \left(\underset{c \epsilon \{Y,Cb,Cr\}}{median} (YCbCr\ image) \right) \tag{5.13}$$

where, Ω represents a vector patch of size 16×1 and c represents Y, Cb and Cr channels. Based on this way, haze-free pixels are extracted from the YCbCr color space.

Step 3. Air light: The $J^{median'}$ is used to estimate air light value. It can be calculated by

considering the average of YCbCr pixels to the highest intensity values in the $J^{med'}$. These pixels are represented in Fig.5.8 at air light (A) with a blue color rectangular box. Eq. 5.14 represents the air light value

$$Air\ light\ (A) = \frac{1}{2M}\left(\sum_{i=1}^{M} YCbCr\ image + \sum_{i=1}^{M} J^{med'}\right) \tag{5.14}$$

where M is the total number of pixels in the image.

Step 4. Transmission map: The transmission map can be estimated directly from the luminance component of the haze image since structural information of any image is mostly associated with its luminance component.

$$Transmission\ T(x) = Luminance\ (Y) \tag{5.15}$$

Step 5. Scene radiance: Once the unknown parameters (air light (A), transmission map (T(x))) are estimated the scene radiance can be obtained as

$$J(x) = \left(\frac{I(x) - A}{max(T(x), t_o)}\right) \tag{5.16}$$

In order to deal with noise amplification a constant parameter t_o=0.1 is used.

Step 6: Finally, Convert the scene radiance from YCbCr color space into RGB color space.

The corresponding pseudo-code of the proposed framework is illustrated in algorithm 2.

5.4 Hardware-Software Co-design approach:

The design strategy is divided into two parts: Programmable Logic (PL) part, Processing system (PS) part. PL part is related to the Hardware (HW) and PS part is related to the Software (SW). The PL part is designed in the Xilinx Vivado using Intellectual Property (IP) cores & integrated with the PS part in Xilinx SDK and validated on Zynq 702.

5.4.1 Programmable Logic (PL) part:

Fig. 5.9 represents the IP core arrangement of real-time single image de-hazing system in accordance to their working principle. In the initial stage, the image can be read from the real world using a High-Density Multilayer Interconnect (HDMI) or Camera module.

Once the image has been read, it can be passed through the Color Filter Array (CFA) [131].

Algorithm 2 Proposed de-haze algorithm

Result: Produces de-hazed image

initialization YCbCr_img, Input_frame, RGB_img, Transmission, air_light, dehaze_YCbCr, scene_radiance

Convert RGB to YCbCr image using eq 2, 3 & 4 Separate Y(Luminance) component from the Input frame Transmission=Y(Luminanace)

while *each channel of YCbCr_img* **do**
  find median_img=from YCbCr_img
end
while *each channel of YCbCr_img* **do**
  while *median_img* **do**
    find air_light
  end
end
while *elements in YCbCr_img* **do**
  dehaze_YCbCr=air_light+(YCbCr_img-air_light)/Transmission
end
while *each channel of YCbCr_img* **do**
  Convert dehaze_YCbCr to RGB
end

scene_radiance=dehaze_RGB

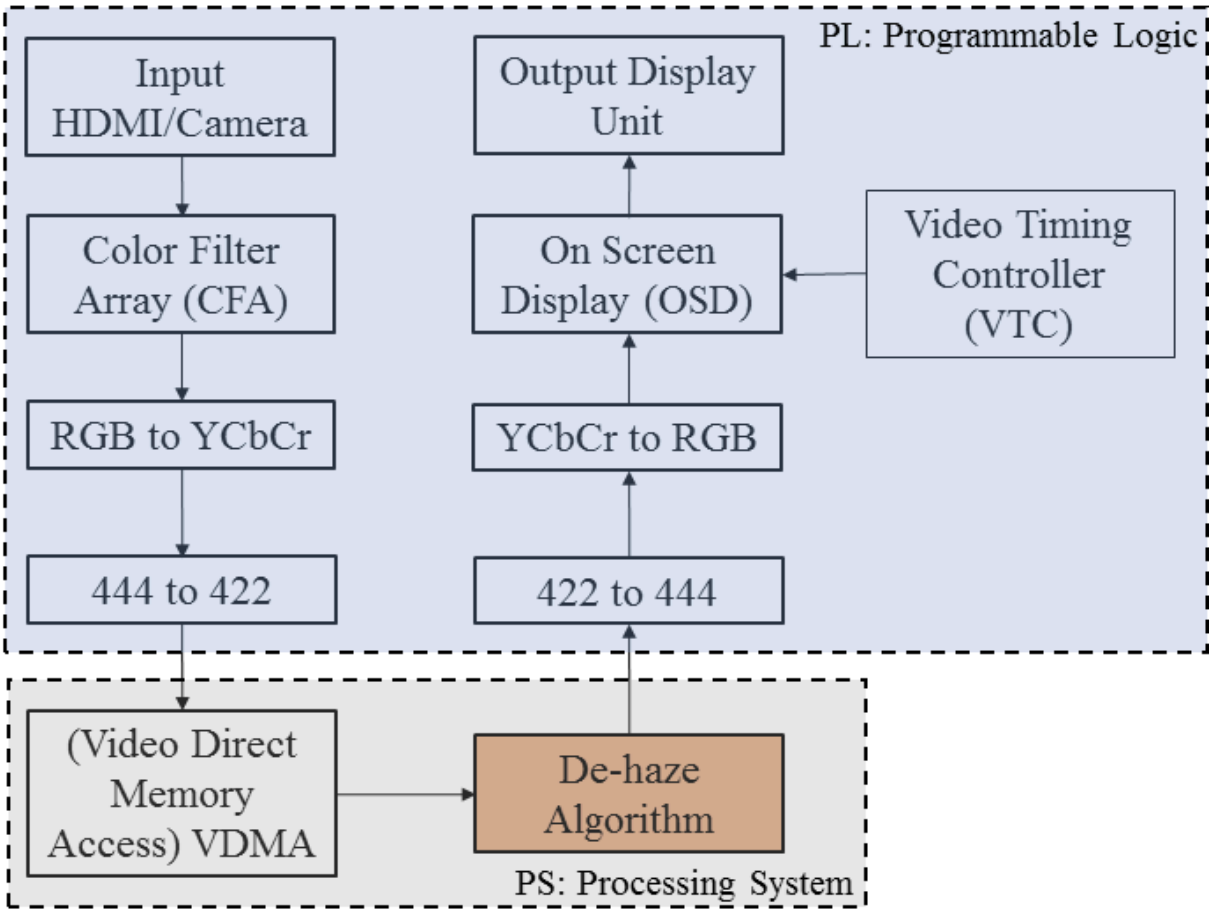


Figure 5.9: IP core arrangement of the de-hazing system

Since the images captured by a CMOS/CCD image sensor are monochrome in nature, they contains only one color information. However, all three primary colors (Red, Green, and Blue) information is needed at each pixel to reconstruct the color image. Based on the information available at neighboring pixels, using the padding and interpolation technique, the missing information of other pixels were recreated [132]. CFA can do the process of recreating the missing color information. The "s_axis_video_tdata" is used as an in_stream of an 8-bit pixel (monochrome) which is converted into "m_axis_video_tdata" as an out_stream of 24-bit pixels (RGB). The image data from CFA is fed to the RGB to YCbCr converter [133].

The YCbCr color format requires less memory and bandwidth to store the image information than the RGB. It means that the RGB pixel requires 24-bits of data to store its information, whereas 12-bits are required for YCbCr (444). Hence, a converter from RGB to YCbCr is needed. The throughput of this converter is one pixel per cycle that is it outputs one pixel per clock. The YCbCr (444) can be further decimated to YCbCr (422) using the 444 to 422 converter through axi_streams. Hence, the memory required to store the number of pixels in this format is reduced by 75 times compared with the RGB. The streaming data is further fed to the Video Direct Memory Access (VDMA) IP.

The purpose of VDMA [134] is to directly access the Advanced eXtensible Interface (AXI4-stream) data and feed it to the peripherals. It has an input data width support of multiples of 8 up to 1024 bits and the address width of size 32 and 64. The address width, input data width of proposed method is 32 bits and 8 bits respectively. The throughput is 96 frames per second. The advantages of this IP in the proposed method is discussed in section 5.4.2. The another most important IPs used in the PL part are On-Screen Display (OSD) [135] and Video Timing Controller VTC [136].

The OSD is an on-screen video processing block used to display the frames in a sequential manner. It supports up to eight layers. The proposed design utilizes only two layers. One layer is used to display the output on monitor, the other layer simultaneously performs the de-haze operation. As these layers belong to RGB color channels, each layer produces 24 -bits of information per pixel at a time. The layer size of the proposed method is 1080×1920×3 at a frame rate of 60 frames per second.

The video timing controller VTC is considered the heart of this video processing system as it synchronizes all the IPs in the PL part. It generates vertical and horizontal timing signals. They are used to find the start and end of the frames. It synchronizes 16 frames at a time. The

maximum clocks per line and maximum frames per line supported by this IP are 8192. The proposed method utilizes 4096 clocks per line for each frame.

The additional IPs used in the PL are 422 to 444 converter and YCbCr to RGB converter. These IPs performs an inverse operation of 444 to 422 and RGB to YCbCr respectively.
The arrangement of IP cores in accordance to their working principle for dehazing is called hardware design of the proposed method. The corresponding software can be programmed as follows.

5.4.2 Processing System (PS) part:

The PS part contains Dual-core ARM Cortex A9 Processor and VDMA. It works in both PS and PL part. It accesses the data through the AXI stream bus. The VDMA is acting as image memory unit. The captured frame data is usually stored in this memory. When the user-defined interrupt gets generated from the PL part, the VDMA accepts that interrupt and stops capturing the frame data. Now, the image memory contains current frame data. We performed the de-hazing operation on those pixels according to **algorithm 2**. This way the hardware (PL) is programmed with software (PS). The modified pixels are fed to the PL part through the AXI stream and then displayed.

5.4.3 Architecture of the image dehazing system:

The architecture of the proposed de-hazing are represented in Fig. 5.10. There are two main parts in architecture; PL and PS part. PL part: The video input sends the sequence of frames to the resizer. According to the user requirement, image dimensions can be selected. 1080×1920 image dimensions are employed in the proposed method. With the help of frame buffers, each frame can be processed through AXI stream bus to the PS part.

The content in the image memory of VDMA is YCbCr color format, where the component Y is a luminance component of the image. The estimation of transmission-map is always complex, and more number of computations are required. The luminance component is directly used as a transmission map in the proposed framework to reduce the number of calculations. The corresponding median image is calculated by employing the median value among the YCbCr pixels. The average of these values and YCbCr is used as air light. The scene radiance is ob-

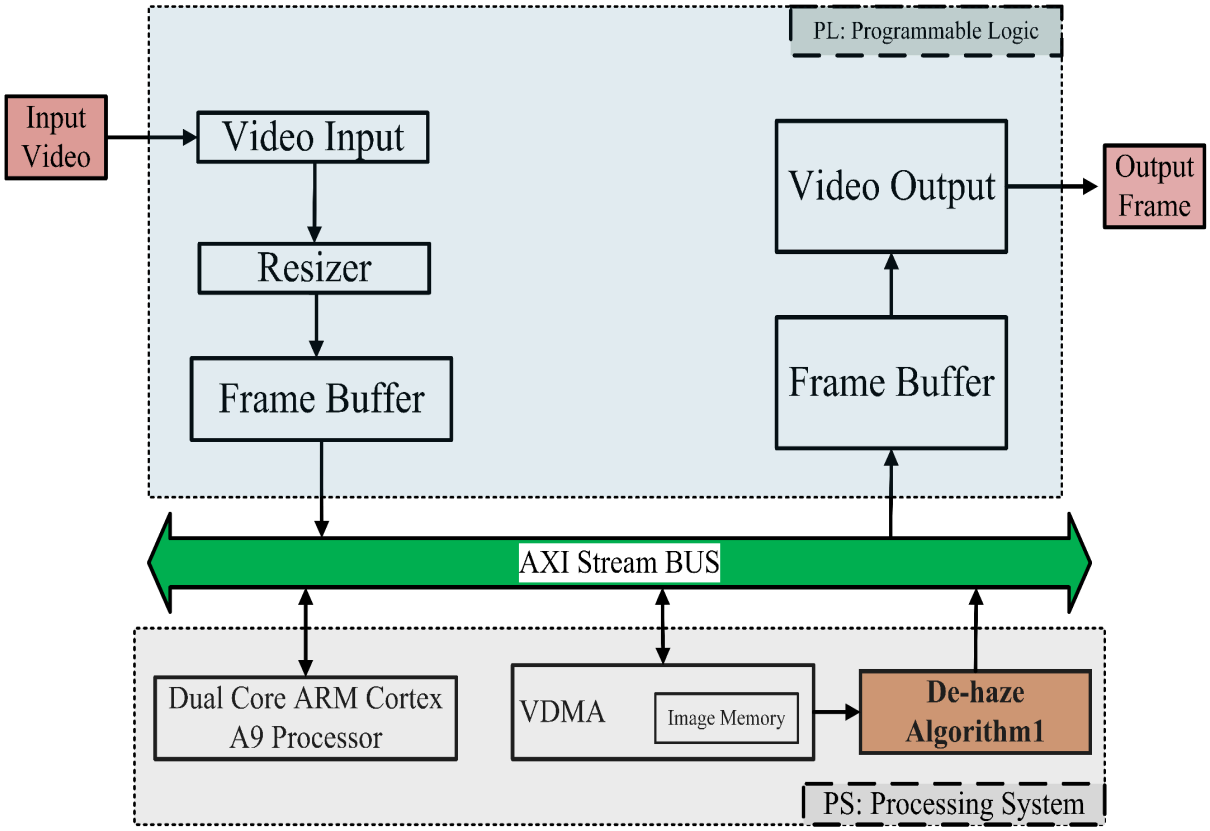


Figure 5.10: The architecture of the proposed dehazing system

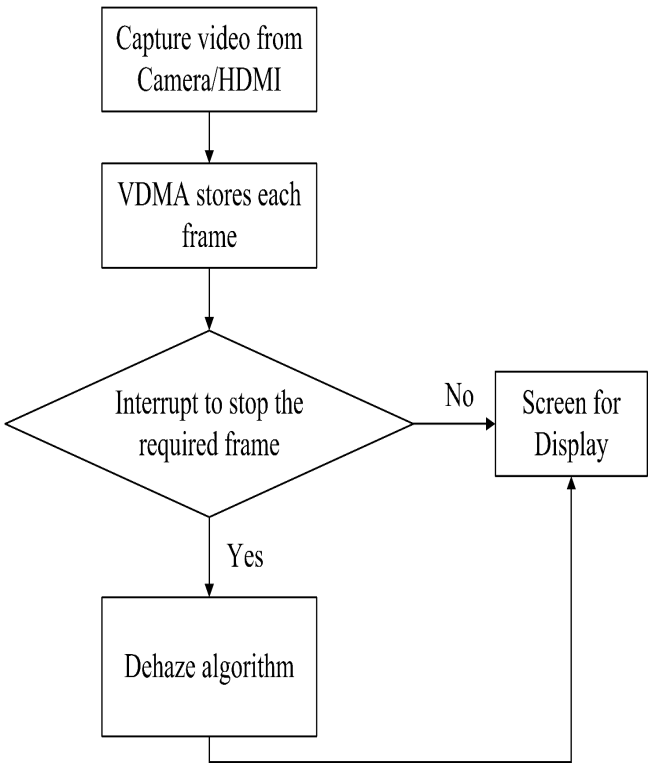


Figure 5.11: Flow chart of the proposed dehazing system

tained by converting the dehazed YCbCr format into an RGB format. At last, scene radiance will be obtained using equation 5.16. Once the algorithm gets executed through AXI stream bus, the frame data is fed to the PL part.
The process of proposed dehazing system is represented in Fig. 5.11

5.5 Results and Discussion

The proposed algorithm is simulated in MATLAB and compared with the DCP and Rahul's method [109] in terms of Time Complexity (TC) in Seconds, Mean Square Error (MSE), and Structural Similarity Index (SSIM). It is observed from Table 5.2 that the values

Table 5.2: Quantitative Comparison of the proposed algorithm

Database	**Method**	**TC**	**SSIM**	**MSE**
City	He et. al	2.422470	0.8781	0.0185
	Rahul's	NA	0.8972	0.0146
	Proposed	1.640968	0.9431	0.0127
Cones	He et. al	2.436265	0.8207	0.0207
	Rahul's	NA	0.8874	0.0178
	Proposed	1.843669	0.8925	0.0120
Canon	He et. al	2.413335	0.8973	0.0133
	Rahul's	NA	0.8978	0.0173
	Proposed	1.589763	0.9018	0.0018
Men	He et. al	2.556302	0.8699	0.0278
	Rahul's	NA	0.8915	0.0172
	Proposed	1.600895	0.8946	0.0128

of TC and MSE are low and SSIM is high in the proposed algorithm compared with the other two algorithms. It is proven that proposed algorithm produces better results. Fig. 5.12 depicts the visual quality comparison of various algorithms. It can be observed that our method can effectively remove the haze particles as compared with He et al. [79] and Rahul's [109] method. At the edges, the details of the building images are clear and sharper in the proposed method. The sky regions are becoming dimmer in He et al. whereas the shaded outcome is appearing in Rahul's method. Based on these results, we map the proposed method on the Zynq 702 platform. Table 5.3 represents the percentage of hardware resources utilized by the algorithms. It is noticed that the proposed method uses less number hardware resources compared with Rahul's method. The proposed method decreased the number of Flip-Flops by 2.5 times, BRAMs by two times, and BRAMs by six times compared with Rahul's approach. The additional hardware

resources such as IO, BUFG, LUTRAMs utilization report is not available for Rahul's method. Since the hardware resources, number of computations (no refine transmission map) and memory requirements (YCbCr used) are less we can say that proposed algorithm performs satisfactorily compared with existing methods.

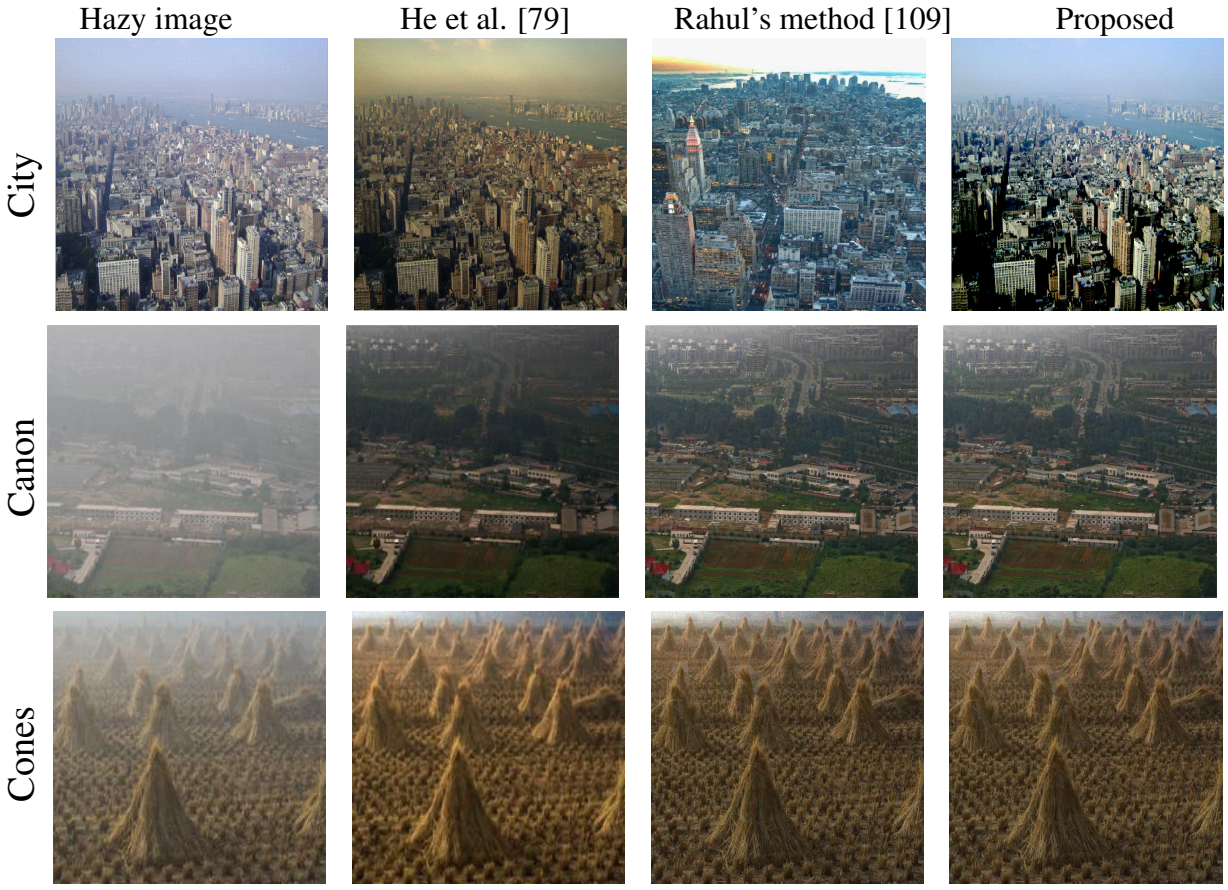


Figure 5.12: Visualization results of different de-hazing systems.

Table 5.3: Hardware utilization summary

Resources	Available	Rahul's	Proposed
BRAM	140	9	5
DSP Slices	220	97	16
FF	106400	6508	2538
LUT	53200	8519	1878
IO	200	–	65
BUFG	32	–	7
LUTRAM	17400	–	1297

Finally, the hardware test setup of the proposed method is depicted in Fig. 5.13. Here, the input is taken from the laptop using an FMC card through an HDMI cable. The algorithm

ran on the Zynq board and displayed the corresponding resultant de-hazed image on another monitor.

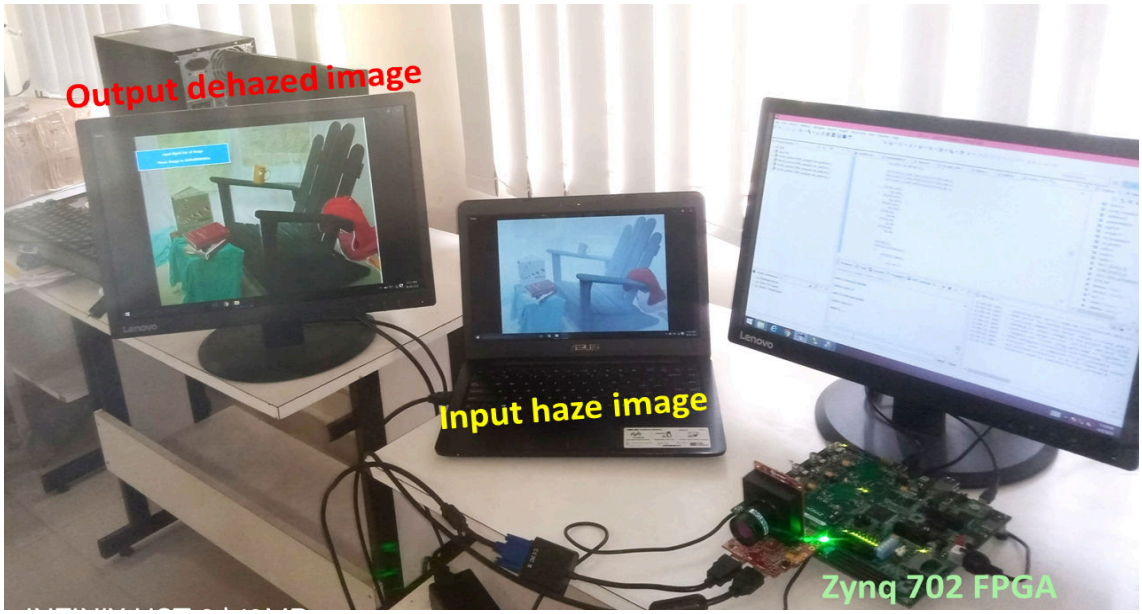


Figure 5.13: Hardware Test Setup of the proposed method

5.5.1 Summary:

Based on qualitative and quantitative metrics we concluded that the proposed approach took minimum computation time; produces output with high contrast and has almost equal mean squared error (MSE). The haze improvement of the proposed algorithm gives a better result than state-of-the-art methods. Our method has a common limitation like all de-hazed algorithms: the haze imaging model is valid only for non-sky regions and it has another drawback of dealing with hazy images which are obtained at long distance, e.g. aerial images. In the future, we will focus on the above aspects.

This chapter implemented a simple and low complex de-hazing algorithm based on the YCbCr color space. The hardware-software co-design approach using IP's was proposed. The comparison results demonstrated that the proposed algorithm effectively removes the haze on the Zynq hardware platform. The YCbCr color format added an advantage to the proposed algorithms in terms of bandwidth, processing speed, hardware resource utilization, and restored image quality.

Chapter 6

Conclusions and Future Scope

In this thesis, we have designed and implemented an efficient de-hazing algorithm on different embedded hardware platforms like, DSP TMS320C6748, Raspberry Pi, in-built GPU device Jatson Nano and high computational complexity devices like Zynq FPGA's. The conclusions are drawn as follows:

6.1 Conclusions

In this thesis, chapters 1 presented the problem of de-hazing in digital images and videos by explaining the physical conditions affecting the light propagation in transparent medium. Haze removal algorithm based on dark channel prior method is analyzed on depth in different aspects. The phenomena of haze removal based on dark channel prior is wrong in estimating the accurate transmission map, and produces some blurs at the edges. To solve this various image filters are used. These filters requires more number of computations. Consequently, effects the efficient hardware implementation.

In chapter 2, pixel based image de-hazing on DSP TMS320C6748 Processor was implemented. We observed that pixel-by-pixel approach effectively eliminates the total number of computations by reducing the step of refine transmission map. It works only for off-line images. Therefore, the implementation of real-time image/video de-hazing is still a challenging task. This challenging task of implementation of de-hazing algorithm for real-time applications is described in chapter 3. In this chapter, a fast and efficient video Dehazing system with reduced computational complexity for real-time computer vision applications was introduced.

It estimated the accurate transmission map using the HSL (Hue, Saturation, and Light) color model together with RGB (Red, Green, and Blue) color space with less number of computations. Later, this algorithm was successfully implemented on Raspberry Pi3 and Jetson Nano. The qualitative and quantitative analysis demonstrated that the implemented method attained better de-hazing results and can be efficiently used for real-time video de-hazing applications.

Accurate transmission map estimation and efficient hardware implementation is further experimented on high computational complexity devices like Zynq FPGA's. The pixel wise and gray image-based technique was implemented on Zynq 706 FPGA and is represented in chapter 4. The key advantages were, estimated the accurate transmission map and achieved faster execution without noticeable degradation of the quality of the de-hazed image. It was implemented on Zynq-706 and DSP Processor. The results obtained show that Zynq-706 based hardware implementation processing speed was 1.33 times faster when compared to DSP processor-based implementation for an image dimensions of 256×256.

The chapter 5 presented an implementation of memory efficient de-hazing algorithm on Zynq 702 FPGA. The rectangular patch moment and median filter added an advantage to this method that processor's speed gets increased, halos and artifacts are reduced in final de-hazed image. This algorithm was initially applied on RGB pixels further optimized using YCbCr color space and implemented on Zynq 702 all programmable SoC based on hardware/software co-design approach. As, YCbCr color space reduces the storage allocation of image pixels the processing speed is high.

To investigate the efficiency of all implemented hardware platforms we have considered a common test image of Bicycle. The thickness of the haze is moderate in this test image. In case of DSP processor based implementation, the haze was removed throughly but at the edges some small amount of halos were found. The processing speed was approximately 50 milliseconds for an image size of 640×480. In the case of Raspberry Pi3 and Jetson Nano implementation 39 milliseconds and 17 milliseconds respectively, for an image dimensions of 1080 × 1920. This approach satisfies the implementation of real-time image/video de-hazing algorithm on embedded platform. The final dehazed images has no halos & artifacts, furthermore the quality is good. It can be further investigated on high end computational devices, such as Zynq 702 and Zynq 706 FPGA's. The approaches for implementation of de-hazing algorithm on these hardware platforms gives satisfactory results for single image de-hazing. These approaches utilizes less number of hardware resources, bandwidth and processing speed.

Thus, we concluded that the development of image/video de-hazing on different hardware platforms such as DSP Processor (TMS320C6748), Raspberry Pi3, Jetson Nano, ZYNQ 702, and ZYNQ 706 FPGA, was investigated.

6.2 Future Scope

During the thesis work, some issues were identified. These issues are the basis for future scope. A more specific summary is provided next.

The single image de-hazing algorithms implemented on Zynq FPGAs can be extended for real-time video de-hazing in the future. Would like to implement more faithful and robust algorithms for image de-hazing and later extend it to the problem of video de-hazing. Still, implementing of real-time video de-hazing algorithm for heavy fog applications on complex devices like FPGAs is a challenging task. An IP has to be implemented for the entire de-hazing system and escalate it to video de-hazing.

In future, Convolutional Neural Network (CNN) based low complexity architecture can be designed for real-time video processing, computer vision applications.

Appendix A

Zynq FPGA

For image processing and computer vision applications Xilinx company introduced Zynq-7000 family. The Xilinx Zynq-7000 All Programmable SoC integrates processing system (PS) and Xilinx programmable logic (PL) in a single chip, and is fabricated on 28 nm technology.

Figure A.1 represents the structural design of Zynq FPGA family. The yellow region belonging to programmable logic (PL), blue color region indicates the processing system (PS) part. These two regions are interconnected through AXI bus interface. The Processing System contains

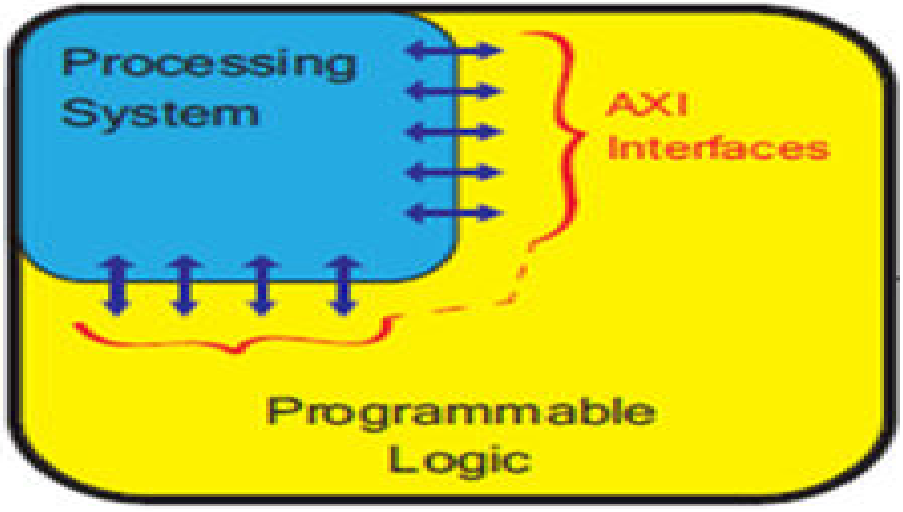


Figure A.1: ZYNQ structural design

ARM Cortex- A9 processor, 252 Kilobyte on-chip memory, 512 Kilobytes of L2 Cache, 32 KB L1 D-Cache and ICache, Direct Memory Access (DMA) controller, Timers, watchdog timers, and other peripheral interfaces. Similarly, The Programmable Logic contains configuration

Table A.1: Resource comparison between Zynq 702 and Zynq 706

Resources	Zynq 702	Zynq 706
LUT	53200	218600
LUTRAM	17400	70400
FF	106400	218600
BRAM	140	540
IO	200	362
BUFG	32	32

storage based on SRAM type; programmable logic blocks (CLBs), Blocked RAM (BRAM), DSP blocks, programmable I/O Blocks, and additional resources. The PS is interconnected to the PL through multiple ARM AMBA AXI ports. During testing process the cores (PS and PL) operate with an internal supply voltage of 1.0 V and an auxiliary supply voltage of 1.8 V. This evaluation board is provided with features common with many systems which perform embedded programs, which includes component memory and DDR3 SODIMM, Ethernet PHY, PCI interface, I/O for general purpose, and interfaces for UART. Some more features are supported by usage of cards of FPGA mezzanine which are attached to higher pin count and lower pin count connectors.

Table A.1 represents the number of hardware resources available in Zynq702 and Zynq 706 FPGAs. It can be observed that the number of hardware resources are more in Zynq 706 compared with Zynq 702 FPGA. Hence, the computational efficiency is more in Zynq 706 all programmable SoC.

Appendix B

DSP TMS320C6748 Processor

TMS320c6748 evaluation module is small in size and has plenty of functions, and compact in structure. This board has more peripherals, which makes it very suitable for DSP embedded system applications. Its main applications include: signal processing. IP camera, vision system, video surveillance system, portable media and video telephone. The key feature of this board is operating speed of the C674x processor is up to 375 MHz for fixed point and 456 MHz for floating-point operations. Before hardware prototype is available and it is designed to use the PC as a host during inceptive development. The PCI host driver extends configuration options for developers by authorizing them to plug the TMS320c6748 EVM directly into the PC. The EVM can also be used in a standalone fashion, connected to a PC using on-board emulation (XDS 1003 USB Emulator), or through a standard JTAG connector. The functional diagram of DSP TMS320C6748 processor is depicted in Fig. B.1.

Additional features:

- Real-Time Clock (RTC) With 32-kHz Oscillator and Separate Power Rail
- Supports All SATA Power-Management Features
- Three 64-Bit General-Purpose Timers (Each Configurable as Two 32-Bit Timers)
- 32-Bit Load-Store RISC Architecture
- 4KB of Instruction RAM Per Core
- 512 Bytes of Data RAM Per Core

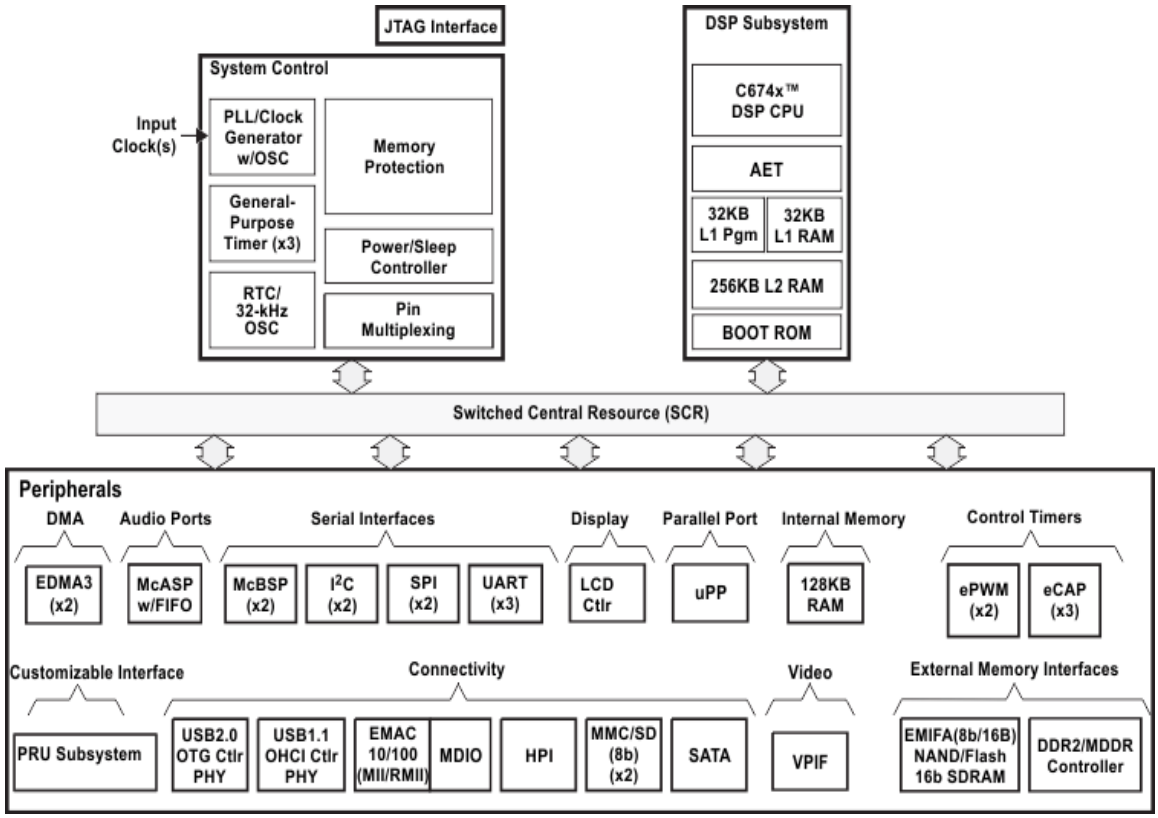


Figure B.1: Functional diagram of DSP Processor

References

[1] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital image processing," 2009.

[2] S. I. Olsen, "Noise variance estimation in images," in *Proc. 8th SCIA*, 1993, pp. 25–28.

[3] Z. Long and N. H. Younan, "Denoising of images with multiplicative noise corruption," in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–4.

[4] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, pp. 1–12, 2019.

[5] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filtering under structural constraints," *IEEE transactions on signal processing*, vol. 43, no. 3, pp. 591–604, 1995.

[6] S. D. Ruikar and D. D. Doye, "Wavelet based image denoising technique," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 2, no. 3, 2011.

[7] L. J. Van Vliet, I. T. Young, and G. L. Beckers, "A nonlinear laplace operator as edge detector in noisy images," *Computer vision, graphics, and image processing*, vol. 45, no. 2, pp. 167–195, 1989.

[8] O. R. Vincent, O. Folorunso *et al.*, "A descriptive algorithm for sobel image edge detection," in *Proceedings of informing science & IT education conference (InSITE)*, vol. 40. Informing Science Institute California, 2009, pp. 97–107.

[9] B. Chen, J.-L. Cai, W.-S. Chen, and Y. Li, "A multiplicative noise removal approach based on partial differential equation model," *Mathematical Problems in Engineering*, vol. 2012, 2012.

[10] A. Kaur and K. Singh, "Speckle noise reduction by using wavelets," in *NCCI 2010-National Conference on Computational Instrumentation CSIO Chandigarh*, 2010.

[11] Y.-M. Huang, M. K. Ng, and Y.-W. Wen, "A new total variation method for multiplicative noise removal," *SIAM Journal on imaging sciences*, vol. 2, no. 1, pp. 20–40, 2009.

[12] J. M. Bioucas-Dias and M. A. Figueiredo, "Multiplicative noise removal using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1720–1730, 2010.

[13] R. Richter, "Atmospheric correction of satellite data with haze removal including a haze/clear transition region," *Computers & Geosciences*, vol. 22, no. 6, pp. 675–681, 1996.

[14] D. Glover and T. Jessup, *Indonesia's fires and haze: the cost of catastrophe*. ISEAS–Yusof Ishak Institute Singapore, 1999.

[15] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 20–37, 2006.

[16] R. Sato, K. Domany, D. Deguchi, Y. Mekada, I. Ide, H. Murase, and Y. Tamatsu, "Visibility estimation of traffic signals under rainy weather conditions for smart driving support," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 1321–1326.

[17] S.-C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE transactions on circuits and systems for video technology*, vol. 21, no. 1, pp. 1–14, 2010.

[18] M. I. Chacon-Murguia and S. Gonzalez-Duarte, "An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3286–3298, 2011.

[19] I. Yoon, S. Kim, D. Kim, M. H. Hayes, and J. Paik, "Adaptive defogging with color correction in the hsv color space for consumer surveillance system," *IEEE transactions on consumer electronics*, vol. 58, no. 1, pp. 111–116, 2012.

[20] S. Tao, H. Feng, Z. Xu, and Q. Li, "Image degradation and recovery based on multiple scattering in remote sensing and bad weather condition," *Optics Express*, vol. 20, no. 15, pp. 16 584–16 595, 2012.

[21] A. Makarau, R. Richter, R. Müller, and P. Reinartz, "Haze detection and removal in remotely sensed multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5895–5905, 2014.

[22] J. Liu, X. Wang, M. Chen, S. Liu, X. Zhou, Z. Shao, and P. Liu, "Thin cloud removal from single satellite images," *Optics express*, vol. 22, no. 1, pp. 618–632, 2014.

[23] X. Pan, F. Xie, Z. Jiang, and J. Yin, "Haze removal for a single remote sensing image based on deformed haze imaging model," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1806–1810, 2015.

[24] S.-C. Huang, B.-H. Chen, and Y.-J. Cheng, "An efficient visibility enhancement algorithm for road scenes captured by intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2321–2332, 2014.

[25] M. S. Shehata, J. Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and A. Radmanesh, "Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 349–360, 2008.

[26] N. Carlevaris-Bianco, A. Mohan, and R. M. Eustice, "Initial results in underwater single image dehazing," in *Oceans 2010 Mts/IEEE Seattle*. IEEE, 2010, pp. 1–8.

[27] P. Drews, E. Nascimento, F. Moraes, S. Botelho, and M. Campos, "Transmission estimation in underwater single images," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 825–830.

[28] J. Y. Chiang and Y.-C. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE transactions on image processing*, vol. 21, no. 4, pp. 1756–1769, 2011.

[29] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi, "Adherent raindrop detection and removal in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1035–1042.

[30] Z. Jia, H. Wang, R. Caballero, Z. Xiong, J. Zhao, and A. Finn, "Real-time content adaptive contrast enhancement for see-through fog and rain," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 1378–1381.

[31] M. L EH, "Lightness and retinex theory," *J. Opt. Soc. Am.*, vol. 61, no. 1, pp. 1–11, 1971.

[32] T. J. Cooper and F. A. Baqai, "Analysis and extensions of the frankle-mccann retinex algorithm," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 85–92, 2004.

[33] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE transactions on image processing*, vol. 6, no. 3, pp. 451–462, 1997.

[34] X. Xu, Q. Chen, P.-a. Wang *et al.*, "A fast halo-free image enhancement method based on retinex," *Journal of Computer-Aided Design & Computer Graphics*, vol. 20, no. 10, pp. 1325–1331, 2008.

[35] Y. Du, B. Guindon, and J. Cihlar, "Haze detection and removal in high resolution satellite image with wavelet analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 1, pp. 210–217, 2002.

[36] S.-d. Zhou, M. Wang, F. Huang, Z.-h. LIU, and S. YE, "Color image defogging based on intensity wavelet transform and color improvement [j]," *Journal of Harbin University of Science and Technology*, vol. 16, no. 4, pp. 59–62, 2011.

[37] J. John and M. Wilscy, "Enhancement of weather degraded color images and video sequences using wavelet fusion," in *Advances in Electrical Engineering and Computational Science*. Springer, 2009, pp. 99–109.

[38] Z. Rong and W. L. Jun, "Improved wavelet transform algorithm for single image dehazing," *Optik*, vol. 125, no. 13, pp. 3064–3066, 2014.

[39] J.-L. Starck, F. Murtagh, E. J. Candès, and D. L. Donoho, "Gray and color image contrast enhancement by the curvelet transform," *IEEE Transactions on image processing*, vol. 12, no. 6, pp. 706–717, 2003.

[40] R. Dale-Jones and T. Tjahjadi, "A study and modification of the local histogram equalization algorithm," *Pattern Recognition*, vol. 26, no. 9, pp. 1373–1381, 1993.

[41] T. K. Kim, J. K. Paik, and B. S. Kang, "Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 82–87, 1998.

[42] H. Xu, G. Zhai, X. Wu, and X. Yang, "Generalized equalization model for image enhancement," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 68–82, 2013.

[43] W. L. Jun and Z. Rong, "Image defogging algorithm of single color image based on wavelet transform and histogram equalization," *Applied Mathematical Sciences*, vol. 7, no. 79, pp. 3913–3921, 2013.

[44] G. Yadav, S. Maheshwari, and A. Agarwal, "Foggy image enhancement using contrast limited adaptive histogram equalization of digitally filtered image: Performance improvement," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2014, pp. 2225–2231.

[45] M. F. Al-Sammaraie, "Contrast enhancement of roads images with foggy scenes based on histogram equalization," in *2015 10th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2015, pp. 95–101.

[46] C. O. Ancuti, C. Ancuti, and P. Bekaert, "Effective single image dehazing by fusion," in *2010 IEEE international conference on image processing*. IEEE, 2010, pp. 3541–3544.

[47] C. O. Ancuti, C. Ancuti, C. Hermans, and P. Bekaert, "Image and video decolorization by fusion," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 79–92.

[48] W. Wang and X. Yuan, "Recent advances in image dehazing," 2017.

[49] Z. Wang and Y. Feng, "Fast single haze image enhancement," *Computers & electrical engineering*, vol. 40, no. 3, pp. 785–795, 2014.

[50] Z. Ma, J. Wen, C. Zhang, Q. Liu, and D. Yan, "An effective fusion defogging approach for single sea fog image," *Neurocomputing*, vol. 173, pp. 1257–1267, 2016.

[51] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 820–827.

[52] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[53] A. J. Preetham, P. Shirley, and B. Smits, "A practical analytic model for daylight," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 91–100.

[54] J. P. Oakley and B. L. Satherley, "Improving image quality in poor visibility conditions using a physical model for contrast degradation," *IEEE transactions on image processing*, vol. 7, no. 2, pp. 167–179, 1998.

[55] O. P. J. TAN K, "Physics-based approach to color image enhancement in poor visibility conditions [j]," *Optical Society of America*, vol. 18, no. 10, pp. 2460–2467, 2001.

[56] K. Tan and J. P. Oakley, "Physics-based approach to color image enhancement in poor visibility conditions," *JOSA A*, vol. 18, no. 10, pp. 2460–2467, 2001.

[57] M. J. Robinson, D. W. Armitage, and J. P. Oakley, "Seeing in the mist: real time video enhancement," *Sensor Review*, 2002.

[58] N. Hautiere and D. Aubert, "Contrast restoration of foggy images through use of an onboard camera," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* IEEE, 2005, pp. 601–606.

[59] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski, "Deep photo: Model-based photograph enhancement and viewing," *ACM transactions on graphics (TOG)*, vol. 27, no. 5, pp. 1–10, 2008.

[60] Y.-T. Peng, K. Cao, and P. C. Cosman, "Generalization of the dark channel prior for single image restoration," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2856–2868, 2018.

[61] S. M. Mirhosseini, "Haze reduction from image by adaptive inverse filter and haar wavelet transform."

[62] H. S. Baird, "The state of the art of document image degradation modelling," in *Digital Document Processing*. Springer, 2007, pp. 261–279.

[63] E. J. McCartney, "Optics of the atmosphere: scattering by molecules and particles," *New York*, 1976.

[64] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Polarization-based vision through haze," *Applied optics*, vol. 42, no. 3, pp. 511–525, 2003.

[65] S. Shwartz, E. Namer, and Y. Y. Schechner, "Blind haze separation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1984–1991.

[66] F. Liu, L. Cao, X. Shao, P. Han, and X. Bin, "Polarimetric dehazing utilizing spatial frequency segregation of images," *Applied optics*, vol. 54, no. 27, pp. 8116–8122, 2015.

[67] S. Fang, X. Xia, X. Huo, and C. Chen, "Image dehazing using polarization effects of objects and airlight," *Optics express*, vol. 22, no. 16, pp. 19 523–19 537, 2014.

[68] T. Treibitz and Y. Y. Schechner, "Polarization: Beneficial for visibility enhancement?" in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 525–532.

[69] C. Li, W. Lu, S. Xue, Y. Shi, and X. Sun, "Quality assessment of polarization analysis images in foggy conditions," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 551–555.

[70] D. Miyazaki, D. Akiyama, M. Baba, R. Furukawa, S. Hiura, and N. Asada, "Polarization-based dehazing using two reference objects," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 852–859.

[71] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *International journal of computer vision*, vol. 48, no. 3, pp. 233–254, 2002.

[72] S. G and S. K. Nayar, Narasimhan, "Removing weather effects from monochrome images," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2. IEEE, 2001, pp. II–II.

[73] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 315–321.

[74] C. Gong, W. Tang, and Z. He-qin, "A novel physics-based method for restoration of foggy day images," *Journal of Image and Graphics*, vol. 5, no. 13, pp. 887–893, 2008.

[75] D. Wu and Q. Dai, "Data-driven visibility enhancement using multi-camera system," in *Enhanced and Synthetic Vision 2010*, vol. 7689. International Society for Optics and Photonics, 2010, p. 76890H.

[76] C. Ancuti and C. O. Ancuti, "Effective contrast-based dehazing for robust image matching," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1871–1875, 2014.

[77] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, pp. 1–9, 2008.

[78] F. Raanan, "Dehazing using color-lines," *ACM transactions on graphics (TOG)*, vol. 34, no. 1, pp. 1–14, 2014.

[79] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[80] D. Park, D. K. Han, and H. Ko, "Single image haze removal with wls-based edge-preserving smoothing filter," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 2469–2473.

[81] J. Yu, C. Xiao, and D. Li, "Physics-based fast single image fog removal," in *IEEE 10th International Conference on Signal Processing Proceedings*. IEEE, 2010, pp. 1048–1052.

[82] C. Long, G. Bao-long, B. Juan, and Z. Juan-juan, "Algorithm of single image fog removal based on joint bilateral filter," *Journal of Beijing University of Posts and Telecommunications*, vol. 35, no. 4, p. 19, 2012.

[83] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 41–50, 2014.

[84] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European conference on computer vision*. Springer, 2010, pp. 1–14.

[85] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Transactions on Image processing*, vol. 24, no. 1, pp. 120–129, 2014.

[86] B. Li, S. Wang, J. Zheng, and L. Zheng, "Single image haze removal using content-adaptive dark channel and post enhancement," *IET Computer Vision*, vol. 8, no. 2, pp. 131–140, 2014.

[87] J.-B. Wang, N. He, L.-L. Zhang, and K. Lu, "Single image dehazing with a physical model and dark channel prior," *Neurocomputing*, vol. 149, pp. 718–728, 2015.

[88] A. K. Tripathi and S. Mukhopadhyay, "Single image fog removal using anisotropic diffusion," *IET Image processing*, vol. 6, no. 7, pp. 966–975, 2012.

[89] F. Fang, F. Li, and T. Zeng, "Single image dehazing and denoising: a fast variational approach," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 969–996, 2014.

[90] Z. Li and J. Zheng, "Edge-preserving decomposition-based single image haze removal," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5432–5441, 2015.

[91] M. Ding and R. Tong, "Efficient dark channel based image dehazing using quadtrees," *Science China Information Sciences*, vol. 56, no. 9, pp. 1–9, 2013.

[92] X. Zhu, Y. Li, and Y. Qiao, "Fast single image dehazing through edge-guided interpolated filter," in *2015 14th IAPR international conference on machine vision applications (MVA)*. IEEE, 2015, pp. 443–446.

[93] K. B. Gibson and T. Q. Nguyen, "Fast single image fog removal using the adaptive wiener filter," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 714–718.

[94] H. Lu, Y. Li, and S. Serikawa, "Underwater image enhancement using guided trigonometric bilateral filter and fast automatic color correction," in *2013 IEEE international conference on image processing*. IEEE, 2013, pp. 3412–3416.

[95] M. Ju, C. Ding, D. Zhang, and Y. J. Guo, "Gamma-correction-based visibility restoration for single hazy images," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 1084–1088, 2018.

[96] L. Zhuang, Y. Ma, Y. Zou, and G. Wang, "A novel image dehazing algorithm via adaptive gamma-correction and modified amef," *IEEE Access*, vol. 8, pp. 207 275–207 286, 2020.

[97] J.-G. Wang, S.-C. Tai, and C.-J. Lin, "Image haze removal using a hybrid of fuzzy inference system and weighted estimation," *Journal of Electronic Imaging*, vol. 24, no. 3, p. 033027, 2015.

[98] Y. Li, Q. Miao, R. Liu, J. Song, Y. Quan, and Y. Huang, "A multi-scale fusion scheme based on haze-relevant features for single image dehazing," *Neurocomputing*, vol. 283, pp. 73–86, 2018.

[99] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-programmable gate arrays*. Springer Science & Business Media, 1992, vol. 180.

[100] S. Qureshi, *Embedded image processing on the TMS320C6000TM DSP: examples in code composer studioTM and MATLAB*. Springer Science & Business Media, 2005.

[101] A. Pajankar, *Raspberry Pi Image Processing Programming*. Springer, 2017.

[102] A. Kurniawan, "Introduction to nvidia jetson nano," in *IoT Projects with NVIDIA Jetson Nano*. Springer, 2021, pp. 1–6.

[103] M. Wnuk, "Remarks on hardware implementation of image processing algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 1, p. 105, 2008.

[104] A. G. Khodary and H. A. Aly, "A new image-sequence haze removal system based on dm6446 davinci processor," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 703–706.

[105] J. Zhang and S. Hu, "A gpu-accelerated real-time single image de-hazing method using pixel-level optimal de-hazing criterion," *Journal of real-time image processing*, vol. 9, no. 4, pp. 661–672, 2014.

[106] Y.-H. Shiau, H.-Y. Yang, P.-Y. Chen, and Y.-Z. Chuang, "Hardware implementation of a fast and efficient haze removal method," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 8, pp. 1369–1374, 2013.

[107] C.-C. Kao, J.-H. Lai, and S.-Y. Chien, "Vlsi architecture design of guided filter for 30 frames/s full-hd video," *IEEE Transactions on circuits and systems for video technology*, vol. 24, no. 3, pp. 513–524, 2013.

[108] Z. Liang, H. Liu, B. Zhang, and B. Wang, "Real-time hardware accelerator for single image haze removal using dark channel prior and guided filter," *IEICE Electronics Express*, vol. 11, no. 24, pp. 20 141 002–20 141 002, 2014.

[109] R. Kumar, B. K. Kaushik, and R. Balasubramanian, "Fpga implementation of image dehazing algorithm for real time applications," in *Applications of Digital Image Processing XL*, vol. 10396. International Society for Optics and Photonics, 2017, p. 1039633.

[110] P. Soma and R. K. Jatoth, "Implementation of a novel, fast and efficient image de-hazing algorithm on embedded hardware platforms," *Circuits, Systems, and Signal Processing*, vol. 40, no. 3, pp. 1278–1294, 2021.

[111] T. Instruments, "Tms320c6748 dsp technical reference manual," 2016.

[112] C. C. Studio, "Getting started guide," *Texas Instruments.–2001*, 2006.

[113] M. Podpora, G. P. Korbas, and A. Kawala-Janik, "Yuv vs rgb-choosing a color space for human-machine interaction." in *FedCSIS (Position Papers)*, 2014, pp. 29–34.

[114] C. Ancuti, C. O. Ancuti, and C. De Vleeschouwer, "D-hazy: A dataset to evaluate quantitatively dehazing algorithms," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2226–2230.

[115] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE transactions on image processing*, vol. 24, no. 11, pp. 3522–3533, 2015.

[116] J.-P. Tarel and N. Hautiere, "Fast visibility restoration from a single color or gray level image," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2201–2208.

[117] D. Berman, S. Avidan *et al.*, "Non-local image dehazing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1674–1682.

[118] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[119] K. Zhang, S. Wang, and X. Zhang, "New metric for quality assessment of digital images based on weighted mean square error," in *Second International Conference on Image and Graphics*, vol. 4875. International Society for Optics and Photonics, 2002, pp. 491–497.

[120] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[121] M. Richardson and S. Wallace, *Getting started with raspberry PI*. " O'Reilly Media, Inc.", 2012.

[122] P. Umesh, "Image processing in python," *CSI Communications*, vol. 23, p. 2, 2012.

[123] S. Cass, "Nvidia makes it easy to embed ai: The jetson nano packs a lot of machine-learning power into diy projects-[hands on]," *IEEE Spectrum*, vol. 57, no. 7, pp. 14–16, 2020.

[124] A. K. Tripathi and S. Mukhopadhyay, "Efficient fog removal from video," *Signal, Image and Video Processing*, vol. 8, no. 8, pp. 1431–1439, 2014.

[125] L. H. Crockett, R. Elliot, M. Enderwitz, and R. Stewart, *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Strathclyde Academic Media, 2014.

[126] J. Ludwig, "Image convolution," *Portland State University*, 2013.

[127] F. Gui and L. Q. Wei, "Application of variogram function in image analysis," in *Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP'04. 2004.*, vol. 2. IEEE, 2004, pp. 1099–1102.

[128] P. Garcia, D. Bhowmik, R. Stewart, G. Michaelson, and A. Wallace, "Optimized memory allocation and power minimization for fpga-based image processing," *Journal of Imaging*, vol. 5, no. 1, p. 7, 2019.

[129] G. Devarajan, V. Aatre, and C. Sridhar, "Analysis of median filter," in *ACE'90. Proceedings of [XVI Annual Convention and Exhibition of the IEEE In India]*. IEEE, 1991, pp. 274–276.

[130] T.-L. Lin, P.-S. Liaw, S.-L. Chen, C.-H. Hsia, and S.-Y. Lin, "Chroma 422 subsampling for bayer pattern in h. 264 video coding," *Electronics Letters*, vol. 52, no. 8, pp. 608–609, 2016.

[131] P. Guide, "Logicore ip color filter array interpolation v4. 0," 2016.

[132] T. Acharya and P.-S. Tsai, "Computational foundations of image interpolation algorithms." *Ubiquity*, vol. 2007, no. October, pp. 4–1, 2007.

[133] P. Guide, "Logicore ip ycrcb to rgb color-space converter v6. 00."

[134] P. Kumbhare and V. Krishna, "Axi vdma reference design," 2012.

[135] Pankaj and V. Krishna, Kumbhare, "Designing high-performance video systems in 7 series fpgas with the axi interconnect," *Xilinx, Inc., San Jose, CA, USA, Appl. Note*, vol. 7, pp. 1–24, 2012.

[136] I. Xilinx LogiCORE, "Video timing controller, product guide, 2012."

List of Publications

1. P. Soma, R. K. Jatoth, and H. Nenavath, "Fast and memory efficient de-hazing technique for real-time computer vision applications," SN Applied Sciences, vol. 2, no. 3, pp. 1–10, 2020. (**ESCI/SCOPUS** - Springer).

2. P. Soma, R. K. Jatoth, "Implementation of a novel, fast and efficient image de-hazing algorithm on embedded hardware platforms," Circuits, Systems, and Signal Processing, vol. 40, no. 3, pp. 1278–1294, 2021. (**SCIE** - Springer).

3. P. Soma, R. K. Jatoth, "An efficient and contrast-enhanced video de-hazing based on transmission estimation using hsl color model," The Visual Computer, pp. 1–12, 2021.. (**SCIE** – Springer)

4. P. Soma, R. K. Jatoth "An Embedded Solution for Image De-hazing based on YCbCr Color Space". *Microprocessors and Microsystems*. (**SCIE**– Elsevier- Under Review)

List of conferences

1. P. Soma and R. K. Jatoth, "Hardware implementation issues on image processing algorithms," in 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018, pp. 1–6.

2. Prathap Soma, Ravi Kumar Jatoth, and Hathiram Nenavath. "Implementation of Single Image De-hazing System on DSP TMS320C6748 Processor". in Soft Computing: Theories and Applications, pp. 405-415. Springer, Singapore, 2020.

Patent Publication

1. Prathap Soma, Ravi Kumar Jatoth "AN IP CORE FOR REAL-TIME IMAGE/VIDEO DEHAZING APPLICATIONS" applied for an INDIAN patent with *Application no: 202141005281*. (published online) (**This work not included in Thesis**)