TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: 072/MSI/612

Single Image Haze Removal Using Genetic Algorithm

by

Rupak Pudasaini

A THESIS

SUBMITTED TO DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

NOVEMBER, 2017

SINGLE IMAGE HAZE REMOVAL USING GENETIC ALGORITHM

BY:

RUPAK PUDASAINI

072/MSI/612

SUPERVISED BY:

Dr. BASANTA JOSHI

A THESIS SUBMITTED TO DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PULCHOWK CAMPUS

INSTITUTE OF ENGINEERING

TRIBHUVAN UNIVERSITY

LALITPUR, NEPAL

NOVEMBER, 2017

COPYRIGHT ©

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled "Single Image Haze Removal Using Genetic Algorithm", submitted by Rupak Pudasaini in partial fulfillment of the requirement for the award of the degree of "Master of Science in Information and Communication Engineering".

_____

Supervisor, Dr. Basanta Joshi
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University,
Lalitpur, Nepal.

_____

External Examiner, Mr. Subhash Dhakal
Director, Department of Information Technology
Government of Nepal
Kathmandu, Nepal.

_____

Committee Chairperson, Dr. Dibakar Raj Pant
Head
Department of Electronics and Computer Engineering,
Pulchwok Campus, Institute of Engineering,
Lalitpur, Nepal.

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Single Image Haze Removal Using Genetic Algorithm"**, submitted by Mr. Rupak Pudasaini in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Information and Communication Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

_____

Dr. Dibakar Raj Pant
Head of Department,
Department of Electronics and Computer Engineering,
Pulchowk Campus, Institute of Engineering,
Lalitpur, Nepal.

ABSTRACT

Single image based haze removal is a very challenging field due to the scarcity of ground truth images of the hazy images for the quantitative validation of the work in this field. A haze removal algorithm based on single image combined with the genetic learning model has been addressed in this thesis work. The basics behind this work is simply finding or deriving a depth map model that directly relates to the saturation and brightness features of the haze. The parameters of the model are learned with powerful evolutionary genetic algorithm which is a supervised learning method. Once the depth map is obtained from the given hazy image, the transmission of the scene can be easily restored, and the scene radiation representing the original image can be restored using the atmospheric model. In the atmospheric model, the scattering coefficient, which is the measure of extinction due to scattering of monochromatic radiation as it propagates through a medium that consists of scattering particles, plays a vital role in the dehazing performance. After the completion of the GP process, nearly optimal solution for depth model is obtained. The effect of varying scattering coefficient (β) on the dehazed images was also studied using 10 test images in the research. The results show that the visual quality of the dehazed image n by rate of visible edges, gradient ratio and SSIM is satisfactory in the range β=0.8 to 1.2 in terms of SSIM, β= 1.4 to 1.8 in terms rate of visible edges and β=1.2 to 1.6 in terms of gradient ratio. The SSIM using the GP learning method was obtained 0.74986, whereas the CAP method resulted in SSIM of 0.547907. MSE of the dehazed images was found to be 0.024571 for the GP learning method, and for CAP method, it was found to be 0.050529. The quantitative results prove that the quality of dehazed images that are degraded by haze is improved by the proposed method based on GP learning strategy.

Keywords: Haze, Genetic Programming, Depth Estimation, Scattering Coefficient, Dehazing, Atmospheric Model

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude to my supervisor, **Dr. Basanta Joshi** for his encouragement, suggestion and continuous guidance throughout the course of my thesis work. He has provided me with tremendous insight and guidance throughout my research.

I would also like to thank **Dr. Dibakar Raj Pant**, Head of Department of Electronics and Computer Engineering, Pulchowk Campus for his precious support and guidance.

I would also like to thank **Prof. Dr. Subarna Shakya** for his valuable comments and remarks throughout the course of the thesis work.

I am also very thankful to **Dr. Sanjeeb Prasad Panday** for his suggestions and engagement throughout the course of my research work.

I also am very grateful and pay my sincere gratitude to all the teachers and faculties for their guidance and every bit of help during the period from title search to thesis topic selection stage and thesis performance stage.

I would also like to thank all my friends for their views and ideas regarding the progress of this thesis work.

Lastly, but perhaps most importantly, I would like to thank my family. None of this would have been possible without the love and encouragement of family members. They have always been there to help me navigate through life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CAP	Color Attenuation Prior
CIE	International Commission of Illumination
DCP	Dark Channel Prior
GP	Genetic Program
MRF	Markov Random Field
MSE	Mean Square Error
RMSE	Root Mean Square Error
SSIM	Structural Similarity Index

CHAPTER 1 INTRODUCTION

1.1 Background and motivation

Outdoor images taken in bad weather (e.g., foggy or hazy) usually lose contrast and fidelity, resulting from the fact that light is absorbed and scattered by the turbid medium such as particles and water droplets in the atmosphere during the process of propagation. Moreover, most automatic systems, which strongly depend on the definition of the input images, fail to work normally caused by the degraded images. Therefore, improving the technique of image haze removal will benefit many images understanding and computer vision applications such as aerial imagery, image classification, image/video retrieval, remote sensing and video analysis and recognition.

1.1.1 Atmospheric Scattering Model

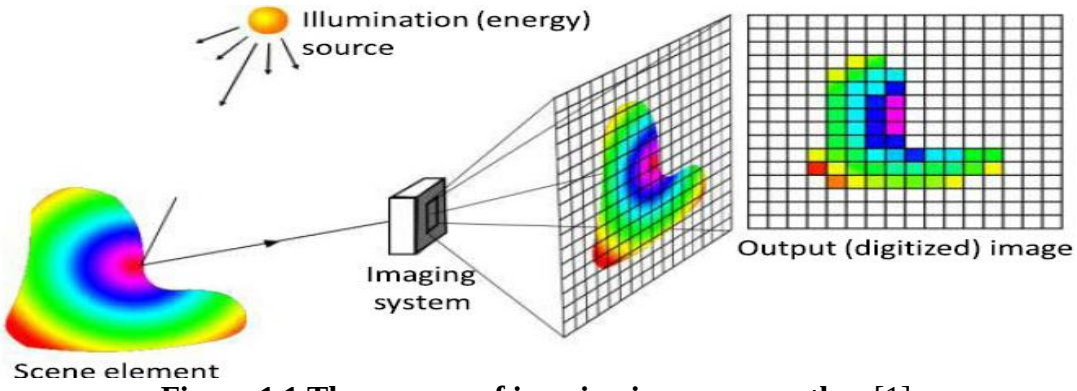


Figure 1.1 The process of imaging in sunny weather.[1]

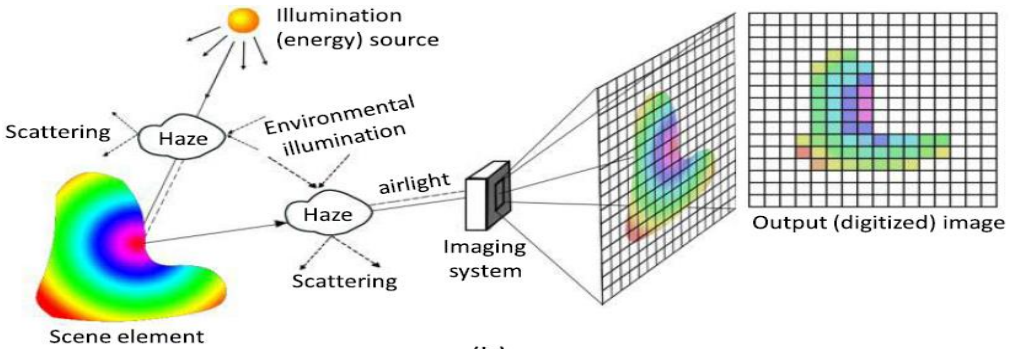


Figure 1.2 The process of image formation in hazy weather[1]

To describe the formation of a hazy image, the atmospheric scattering model, which is proposed by McCartney in 1976, is widely used in computer vision and image processing. Narasimhan and Nayar further derived the model later, and the model can be expressed as follows:

$$I(x) = J(x)t(x) + A\big(1 - t(x)\big) \quad (1)$$

$$t(x) = \exp(-\beta d(x)) \quad (2)$$

where x is the position of the pixel within the image, I is the hazy image, J is the scene radiance representing the haze-free image, A is the atmospheric light, t is the medium transmission, β is the scattering coefficient of the atmosphere and d is the depth of scene. I, J and A are all three-dimensional vectors in RGB space. Since I is known, the goal of dehazing is to estimate A and t, then restore J according to Equation (1). It is worth noting that the depth of the scene d is the most valuable information. Since the scattering coefficient β can be regarded as a constant in homogeneous atmosphere condition, the medium transmission t can be estimated easily according to Equation (2) if the depth of the scene is given. Moreover, in the ideal case, the range of d(x) is [0, +∞) as the scenery objects that appear in the image can be very far from the observer, and we have:

$$I(x) = A, d(x) \to \infty \quad (3)$$

Equation (3) shows that the intensity of the pixel, which makes the depth tend to infinity, can stand for the value of the atmospheric light A. Note that, if d(x) is large enough, t (x) tends to be very small according to Equation (2), and I(x) equals A approximately. Therefore, instead of calculating the atmospheric light A by Equation (3), we can estimate A by the following equation given a threshold $d_{thresold}$:

$$I(x) = A, d(x) \geq dthresold \quad (4)$$

It can also be noticed that it is not hard to satisfy this constraint: $d(x) > d_{thresold}$. In most cases, a hazy image taken outdoor has a distant view that is kilometres away from the observer. In other words, the pixel belonging to the region with a distant view in the image should have a very large depth $d_{threshold}$. Assuming that every hazy image has a distant view, we have:

$$d(x) \geq dthresold, x \in \{x|\forall y : d(y) \leq d(x)\} \quad (5)$$

Based on this assumption, the atmospheric light A is given by:

$$A = I(x), x \in \{x|\forall y : d(y) \leq d(x)\} \quad (6)$$

On this condition, the task of dehazing can be further converted into depth information restoration. However, it is also a challenging task to obtain the depth map from a single hazy image.

1.2 Genetic Programming

Genetic programming (GP) is a collection of evolutionary computation techniques that allow computers to solve problems automatically. Since its inception twenty years ago, GP has been used to solve a wide range of practical problems, producing many human-competitive results and even patentable new inventions. Like many other areas of computer science, GP is evolving rapidly, with innovative ideas, techniques and applications being constantly proposed.

Genetic programming (GP) is an evolutionary computation (EC) technique that automatically solves problems without requiring the user to know or specify the form or structure of the solution in advance. At the most abstract level GP is a systematic, domain-independent method for getting computers to solve problems automatically starting from a high-level statement of what needs to be done.

In genetic programming we evolve a population of computer programs. That is, generation by generation, GP stochastically transforms populations of programs into new, hopefully better, populations of programs, shown in Figure 1.3. GP, like nature, is a random process, and it can never guarantee results. GP's essential randomness, however, can lead it to

escape traps which deterministic methods may be captured by. Like nature, GP has been very successful at evolving novel and unexpected ways of solving problems.

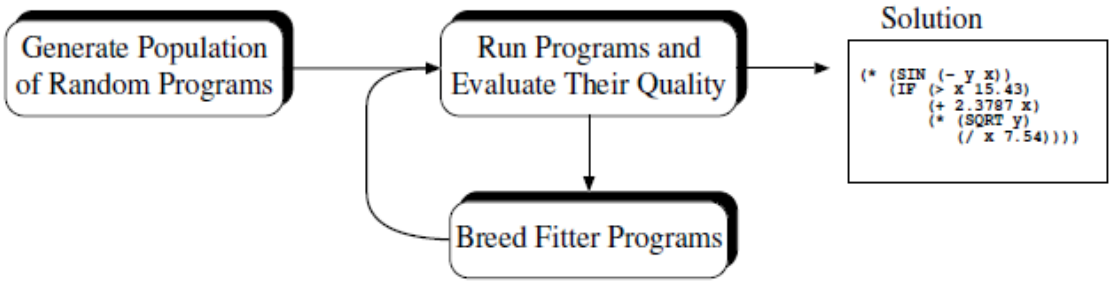


Figure 1.3 The basic control flow for genetic algorithm

Genetic Programming Algorithm

1. Randomly create an initial population of programs from the available primitives
2. **repeat**
    a. Execute each program and ascertain its fitness.
    b. Select one or two program(s) from the population with a probability based on fitness to participate in genetic operations
    c. Create new individual program(s) by applying genetic operations with specified probabilities
3. **until** an acceptable solu
4. tion is found or some other stopping condition is met (e.g., a maximum number of generations is reached).
5. **return** the best-so-far individual.

The basic steps in a GP system are shown in the above algorithm. GP finds out how well a program works by running it, and then comparing its behavior to some ideal. We might be interested, for example, in how well a program predicts a time series or controls an industrial process. This comparison is quantified to give a numeric value called fitness. Those programs that do well are chosen to breed and produce new programs for the next generation. The primary genetic operations that are used to create new programs from existing ones are:

• Crossover: The creation of a child program by combining randomly chosen parts from two selected parent programs.

• Mutation: The creation of a new child program by randomly altering a randomly chosen part of a selected parent program.

1.3 Problem Statement

Single image haze removal has been a challenging problem due to its ill-posed nature. In this research, a powerful method for haze removal from a single input hazy image is developed. By learning the parameters of the model with a supervised learning method, the depth information can be well recovered. With the depth map of the hazy image, it is easy to estimate the transmission and restore the scene radiance via the atmospheric scattering model, and thus effectively remove the haze from a single image. The color attenuation prior method that uses probabilistic learning for estimating depth map model parameters as a linear relationship to saturation and brightness, sometimes gives negative values of depth. Since the negative depth values are impossible to occur, the depth model predicted by this method is not quite accurate. Taking this account, this research aims to establish the true relationship between the saturation and brightness value of a hazy image with the depth map using genetic learning framework.

1.4 Objectives

- To remove the haze from the single image using Genetic learning framework
- To study the effect of varying scattering coefficient on the dehazed images
- To compare the dehazing capabilities of the proposed method based on genetic learning strategy and Color Attenuation Prior method

1.5 Scope of work

Haze comes as a result of some atmospheric phenomena, and images captured by a digital camera can be uniformly corrupted by a large percentage of haze particles. These images usually look hazier and more faded as compared with haze-free versions. Image corruption due to haze has significantly harmed many outdoor computer vision techniques, hindering their ability to achieve their image analytic tasks, such as tracking people counting and crowds' analysis, action, recognition, human-machine interaction, license plate

recognition, and so on. By using proposed haze removal algorithm, the stability and robustness of the visual system can be improved.

CHAPTER 2 LITERATURE REVIEW

The traditional technique of image processing like histogram-based dehazing methods [2] were used early by the researchers to remove haze from a single image. Since the single image cannot provide much information, the dehazing effect was limited. In this paper, block-overlapped histogram equalization system for enhancing contrast of image sequences was proposed.

Later, researchers tried to improve the dehazing performance with multiple images. Schechner et al. [3] proposed polarization based methods that are used for dehazing with multiple images which are taken with different degrees of polarization. It is based on the fact that usually airlight scattered by atmospheric particles is partially polarized. Polarization filtering alone cannot remove the haze effects, except in restricted situations. This method, however, worked under a wide range of atmospheric and viewing conditions. The image formation process is analyzed, taking into account polarization effects of atmospheric scattering. Then the process was inverted to enable the removal of haze from images. The method can be used with as few as two images taken through a polarizer at different orientations.

Narasimhan et al. [4] proposed haze removal approaches with multiple images of the same scene under different weather conditions. In this paper, a geometric framework was developed for analyzing the chromatic effects of atmospheric scattering. First, a simple color model was studied for atmospheric scattering and verified for fog and haze. Then, based on the physics of scattering, several geometric constraints were derived on scene color changes, caused by varying atmospheric conditions. Finally, using these constraints we algorithms were derived for computing fog or haze color, depth segmentation, extracting three-dimensional structure, and recovering true scene colors, from two or more images taken under different but unknown weather conditions.

Narasimhan et al. [5] conducted dehazing based on the given depth information. In this paper, the question of de-weathering a single image using simple additional information provided interactively by the user was addressed. Effective color and contrast restoration using several images taken under poor weather conditions were demonstrated.

Recently, significant progress has been made in single image dehazing based on the physical model. Under the assumption that the local contrast of the haze-free image is much higher than that in the hazy image, Tan [15] proposes a novel haze removal method by maximizing the local contrast of the image based on Markov Random Field (MRF). Although Tan's approach is able to achieve impressive results, it tends to produce over-saturated images.

He et al. [6] discover the dark channel prior (DCP) that, in most of the non-sky patches, at least one-color channel has some pixels whose intensities are very low and close to zero. With this prior, they estimate the thickness of haze, and restore the haze-free image by the atmospheric scattering model. The DCP approach is simple and effective in most cases. However, it cannot well handle the sky images and is computationally intensive.

Zhu et al. [1] proposed a simple but powerful color attenuation prior for haze removal from a single input hazy image. By creating a linear model for modeling the scene depth of the hazy image under this novel prior and learning the parameters of the model with a supervised learning method, the depth information can be well recovered. With the depth map of the hazy image, the transmission can be easily restored and restore the scene radiance via the atmospheric scattering model, and thus effectively remove the haze from a single image.

Recently, machine learning techniques allow more accurate and faster implementation of image processing and computer vision tasks. By taking advantage of machine learning and principle of the atmospheric scattering, several approaches have been developed for dehazing. Gibson et al. [9] have tried to estimate the medium transmission by applying an example-based learning framework [10] originally proposed for super-resolution.

Wang et al. [11] systematically investigated different haze-relevant features in a learning framework to identify the best feature combination for image dehazing. It was shown that the dark-channel feature is the most informative one for this task, while other haze-relevant features also contribute significantly in a complementary way.

As an alternative machine learning framework, genetic programming (GP), a powerful evolutionary process inspired by natural phenomenon, can be employed to discover haze

in the scene. In image dehazing, an effective haze estimator is attempted to be found by combining various features. GP has demonstrated its ability to extract meaningful information from input data. For example, Davis et al. [7] have employed GP for discriminative feature selection in multivariate data analysis.

More recently, Liu et al. [8] exploited GP to learn the spatiotemporal representation of visual features for human action recognition. Extracting discriminative and robust features from video sequences is the first and most critical step in human action recognition.

CHAPTER 3 METHODOLOGY

3.1 Block Diagram

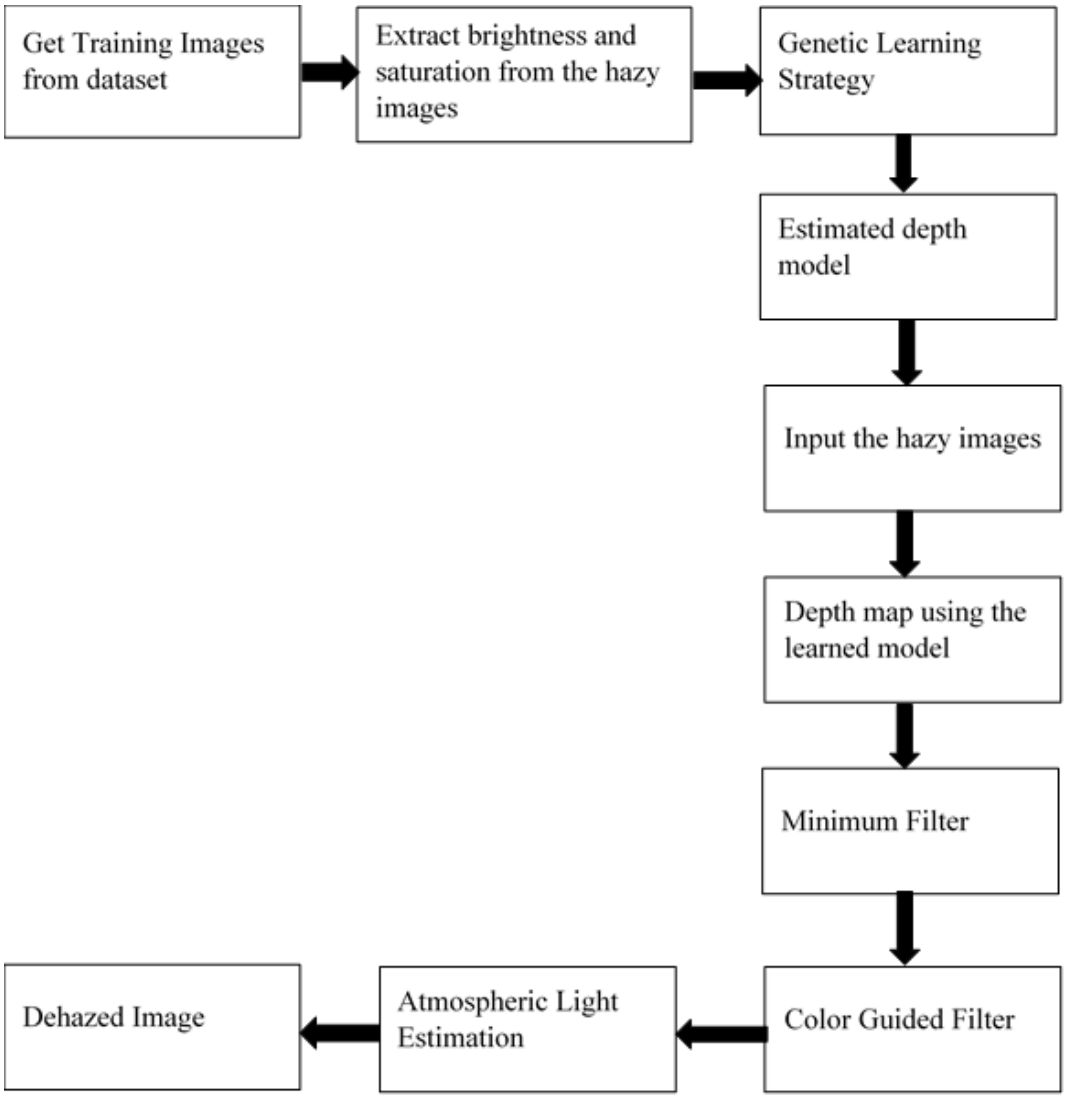


Figure 3.1 Single Image Dehazing model

The methodology adopted in this thesis work can be divided into two stages. First stage consists of training stage, in which hazy images and corresponding depth map are taken from the dataset. The aim of this stage is to establish a relationship between the saturation and the brightness component, the two features of a hazy image, with the depth. The depth

estimated employed in this thesis work is based upon the single hazy image. So, the saturation and brightness vectors of the hazy images are extracted and are input to the genetic learning algorithm.

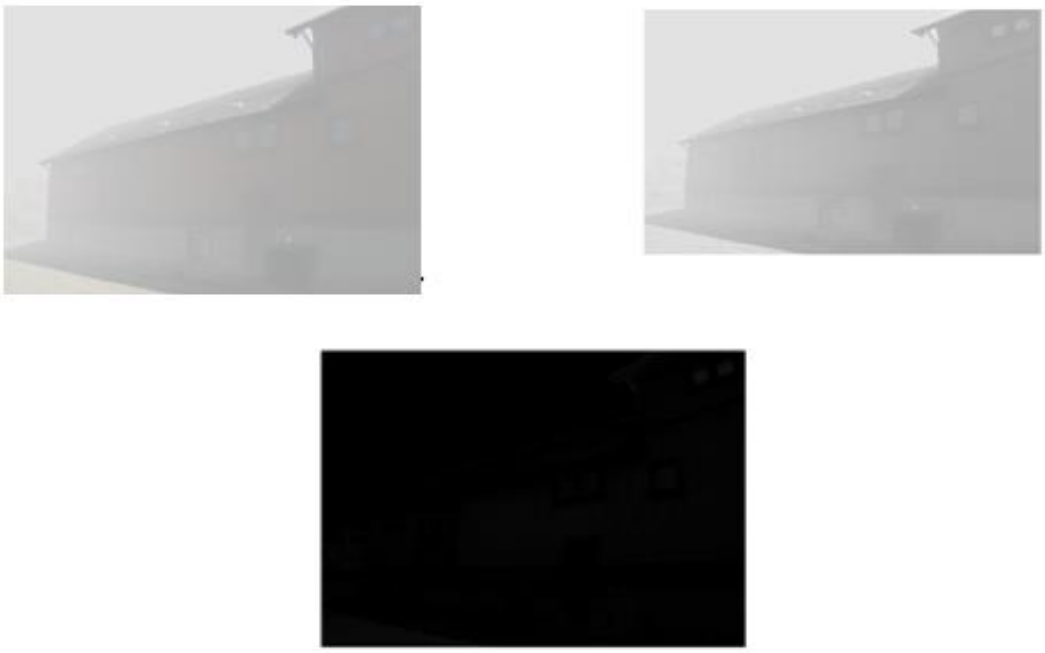


Figure 3.2 Hazy image, brightness component of hazy image and saturation component of the hazy image

Figure 3.2 shows the example of the hazy images, the corresponding extracted brightness values from the image and the saturation component of the hazy image used for training.

Basically, the evolutionary genetic learning algorithm is programmed to create a random initial solution consisting of 100 populations that consist of the possible solutions that describe or give the relationship between the depth, saturation and brightness of hazy images. The actual ground truth depth is also available. The saturation and brightness values of the images are input or processed by the algorithm on a pixel by pixel basis. 100 possible random solutions that relates the three discussed variables, are generated by the genetic algorithm during each generation. Then, using the saturation and brightness for each pixel in the hazy image, the depth value predicted by the learning algorithm is calculated. Then RMSE between the estimated depth value and actual ground truth value is calculated. This process is continued for every pixel in the hazy image and for each

generation until desired number of generations is reached, or the acceptable error is obtained. The objective function used by the algorithm is Root Mean Square Error between the estimated depth and the actual ground truth depth. So, the main objective is the reduce the RMSE at each generation to give the best possible relationship between the depth and the two haze features. Since single image dehazing is based on the depth estimation of the image from the available hazy image, the first step in the field of image dehazing is to estimate the depth of the image, and this work or task is performed by the learning stage.

The block diagram for haze removal based on single image is shown in the Figure 4.2. Equation 1 relates the hazy image J with corresponding scene radiance representing original image I, atmospheric light A and depth map d. The parameter β is known as scattering coefficient whose value depends upon the medium in which imaging is performed. Its value is large for dense mediums whereas low for less dense mediums. From the equation, it is clear that in order to recover the original image from the hazy image, we need to know the values of β, atmospheric light A and depth map d. The depth map d can be obtained from the learned model. The atmospheric light from a hazy image can be estimated using the procedure described in the section 3.1. Once all these values are known, the original dehazed image can be easily estimated.

3.2 Estimation of the Atmospheric Light

As the depth map of the input hazy image has been recovered, the distribution of the scene depth is known. Bright regions in the map stand for distant places. According to Equation (6), the top 0.1 percent brightest pixels in the depth map are picked, and the pixel with highest intensity in the corresponding hazy image I is selected among these brightest pixels as the atmospheric light A.0(**see** Figure 3.3**)**. The red patches in the following hazy images show the pixels for the estimation of the atmospheric light.

Figure 3.3 Estimation of atmospheric light

3.3 Scene Radiance Recovery

After the depth of the scene d and the atmospheric light are known, it is possible to estimate the medium transmission t easily according to Equation (2) and recover the scene radiance J in Equation (1). For convenience, Equation (1) can be written as follows:

$$J(x) = (I(x) - A)/t(x) + A = (I(x) - A)/\exp(-\beta d(x)) + A \qquad (7)$$

For avoiding producing too much noise, the value of the transmission t (x) is restricted between 0.1 and 0.9. So, the final function used for restoring the scene radiance J in the proposed method can be expressed by:

$$J(x) = (I(x) - A)/(\min\{\max\{\exp(-\beta d(x)), 0.1\}, 0.9\}) + A \qquad (8)$$

where J is the haze-free image that is wanted.

3.4 Estimation of the Depth Information

As the relationship among the scene depth *d*, the brightness *v* and the saturation *s* has been established and the model has been estimated, the depth map of a given input hazy image cane be established**.** However, the model may fail to work in some particular situations. For instance, the white objects in an image are usually with high values of the brightness and low values of the saturation. Therefore, the model tends to consider the scene objects with white color as being distant. Unfortunately, this misclassification will result in inaccurate estimation of the depth in some cases. As shown in Figure 3.4, the white geese in the first image are the regions for which the model can hardly handle, and these regions are wrongly estimated with high depth values in the depth map **(**see Figure 3.4(b)). To

overcome this problem, each pixel needs to be considered in the neighborhood. Based on the assumption that the scene depth is locally constant, the raw depth map is processed by:

$$d_r(x) = \min_{y \varepsilon \wedge_r(x)} d(y) \tag{9}$$

where $\wedge_r(x)$ is an $r \times r$ neighborhood centered at x, and d_r is the depth map with scale r. As shown in Figure 3.4(c), the new depth map d_{15} can well handle the geese regions. However, it is also obvious that the blocking artifacts appear in the image. To refine the depth map, the guided image filtering **[**21**]** was used to smooth the image (Details in APPENDIX B). Figure 3.4(d) shows the final restored depth map of the hazy image. As can be seen, the blocking artifacts are suppressed effectively.

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

Figure 3.4 Possible misclassification and its remedy

(a) hazy image (b)raw depth map (c)patched depth map (d)guided filter depth map

3.5 Learning Strategy

3.5.1 Probabilistic Learning (Color Attenuation Prior method)

What is interested in is the joint conditional concentration:

$$\mathrm{L} = \mathrm{p}(\mathrm{d}(\mathrm{x1}), \ldots, \mathrm{d}(\mathrm{xn}) | \mathrm{x1}, \ldots, \mathrm{xn}, \theta 0, \theta 1, \theta 2, \sigma 2) \tag{10}$$

where n is the total number of pixels within the training hazy images, $\mathrm{d}(\mathrm{x_n})$ is the depth of the nth scene point, and L is the likelihood. Assuming that the random error at each scene point is independent (i.e.$\mathrm{p}(\varepsilon 1, \ldots, \varepsilon \mathrm{n}) = \prod_{i=1}^{i=n} \mathrm{p}(\varepsilon \mathrm{i})$), Equation (11) can be re-written as:

$$\mathrm{L} = \prod_{i=1}^{i=n} \mathrm{p}(\mathrm{d}(\mathrm{x}_i) | x_i, \theta_0, \theta_1, \theta_2, \sigma^2) \tag{11}$$

According to Equation (10) and Equation (11),

$$L = \prod_{i=1}^{n} \frac{e^{-\frac{dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i)}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \tag{12}$$

where dg_i represents the ground truth depth of the ith scene point. So the problem is to find the optimal values of θ_0, θ_1, θ_2, and σ to maximum L. For convenience, instead of maximizing the likelihood directly, the natural logarithm of the likelihood is maximized ln L. Therefore, the problem can be expressed as follows:

$$arg \max_{\theta_0,\theta_1,\theta_2,\sigma} lnL = \prod_{i=1}^{n} \ln\left(\frac{e^{-\frac{dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i)}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\right. \tag{13}$$

To solve the problem, the partial derivative of ln L with respect to σ is calculated and is made equal to zero:

$$\frac{\partial lnL}{\partial \sigma} = -\frac{\pi}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i) = 0 \tag{14}$$

According to Equation (14), the maximum likelihood estimate for the variable σ^2 is

$$\sigma^2 = \sum_{i=1}^{n}(dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i)^2 \tag{15}$$

As for the linear coefficients θ_0, θ_1 and θ_2, gradient descent algorithm is used to estimate their values. By taking the partial derivatives of ln L with respect to θ_0, θ_1 and θ_2 respectively, following expressions can be obtained:

$$\frac{\partial lnL}{\partial \theta_0} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i) \tag{16}$$

$$\frac{\partial lnL}{\partial \theta_1} = \frac{1}{\sigma^2}\sum_{i=1}^{n} v(x_i)(dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i) \tag{17}$$

$$\frac{\partial lnL}{\partial \theta_2} = \frac{1}{\sigma^2}\sum_{i=1}^{n} s(x_i)(dg_i - (\theta_0 + \theta_1 V(x)_i + \theta_2 S(x)_i) \tag{18}$$

The expression for updating the linear coefficients can be concisely expressed by:

$$\theta_i := \theta_i + \frac{\partial \ln L}{\partial \theta_i} \text{ s.t. } i \in \{0,1,2\} \quad (19)$$

It is worth noting that the expression above is used for iterating dynamically, and the notation: = does not express the mathematical equality, but means that setting the value of θ_i in the left term to be the value of the right term.

Parameters Estimation Algorithm

Input: The training brightness vector v, the training saturation vector s, the depth vector d, and the number of iterations

Output: Linear coefficients θ_i, i= 0,1,2 and σ^2

Auxiliary functions: function for obtaining the size of the vector: n=size(in)

begin

1. n=size(v)
2. $\theta_0 = 0; \theta_1 = -1; \theta_1 = -1;$
3. sum=0; wSum=0; vSum=0; sSum=0;
4. for iteration from 1 to t **do**
5. for index from 1 to n **do**
6. temp=d[i]- θ_0-θ_1*v[i]- θ_2*s[i];
7. wSum=wSum+temp;
8. vSum=vSum+v[i]*temp;
9. sSum=sSum+s[i]*temp;
10. sum=sum+square(temp);
11. **end for**
12. σ^2=sum/n;
13. θ_0=θ_0+wSum; θ_1=θ_1+vSum; θ_1=θ_1+sSum;
14. **end for**

end

3.5.2 Genetic Learning

GP is an evolutionary computation technique that automatically solves problems without requiring the user to know or specify the form or structure of the solution in advance. The general procedure of GP is as follows. First, an initial population is randomly created that consists of individual programs, each of which is considered as a candidate solution. Then, GP stochastically evolves the population by genetic operations such as selection, crossover, and mutation.

Underlying randomness in these operations leads it to escape local minima which deterministic methods may be captured. As generation continues, the overall population improves their fitness to the ground-truth training data. Finally, the best-so-far individual is taken as the solution. It is worth noting that GP does not guarantee the optimal solution, but it can provide a near-optimal one in an acceptable amount of computation time.

In GP, an individual program in the population is represented as a tree structure, making mathematical expressions easy to evolve and evaluate. To make GP search the possible solution space efficiently, available nodes of the tree should be first defined. In this implementation of Genetic learning algorithm, a Genetic program solution was represented using tree structure using single gene; an individual GP solution consists of only one gene structure in single gene analysis, whereas, in the case of multigene analysis, more than one genes are used to represent the single GP solution.

Once a first generation of population has been generated, the genetic algorithm seeks to generate the new population in another generation using crossover and mutation. The objective is to generate a new fitter set of individual solutions and carry them to the next generation using the minimization of the root mean square error as the criteria. Our aim in the genetic programming based learning is minimization of the fitness function and selection of the best individual among the possible solutions.

Proposed Learning Strategy using Genetic Algorithm

1. Input training data of brightness, saturation and depth vectors to the program
2. Randomly create an initial population of programs from the available primitives
3. **repeat**
    a. Execute each program and ascertain its fitness.
    b. Select one or two program(s) from the population with a probability based on fitness to participate in genetic operations
    c. Create new individual program(s) by applying genetic operations with specified probabilities
4. **until** an acceptable solution is found or some other stopping condition is met (a maximum number of generations is reached).
5. **return** the best-so-far individual.

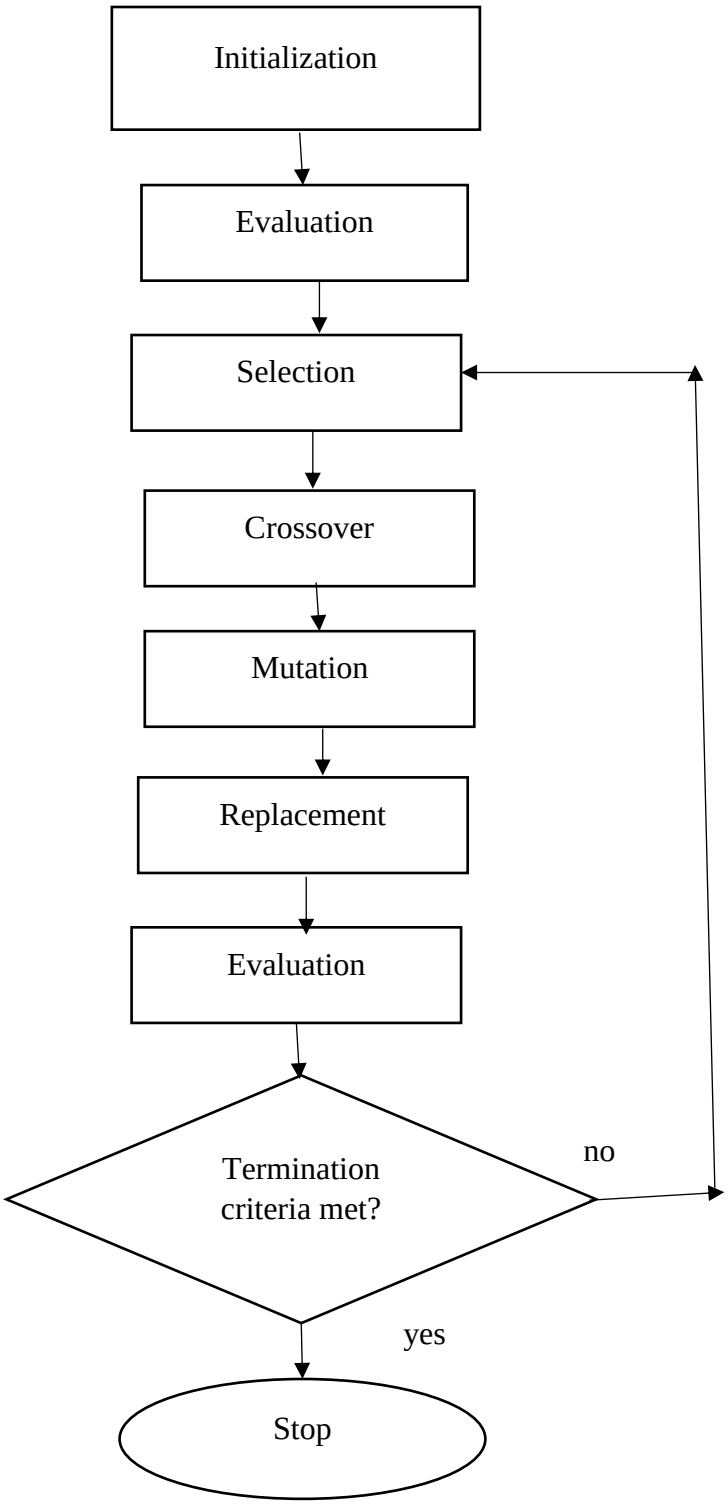


Figure 3.5 Genetic Algorithm Flowchart

3.5.3 Representation

In GP, programs are usually expressed as syntax trees rather than as lines of code. For example, Figure 3.6 shows the tree representation of the program max (x+x, x+3*y). The variables and constants in the program (x, y and 3) are leaves of the tree. In GP they are called terminals, whilst the arithmetic operations (+, * and max) are internal nodes called functions. The sets of allowed functions and terminals together form the primitive set of a GP system.

In high performance environments, the tree-based representation of programs may be too inefficient since it requires the storage and management of numerous pointers. In some cases, it may be desirable to use GP primitives which accept a variable number of arguments (a quantity we will call arity). Fortunately, it is now extremely common in GP applications for all functions to have a fixed number of arguments. If this is the case, then, the brackets in prefix-notation expressions are redundant, and trees can efficiently be represented as simple linear sequences. In effect, the function's name gives its arity and from the arities the brackets can be inferred. For example, the expression (max (+ xx) (+ x (* 3 y))) could be written unambiguously as the sequence max+ x x + x * 3 y.

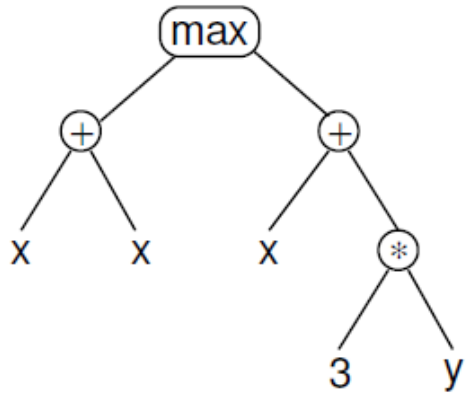


Figure 3.6 GP syntax tree representing max (x+x, x+3*y)

The choice of whether to use such a linear representation or an explicit tree representation is typically guided by questions of convenience, efficiency, the genetic operations being used (some may be more easily or more efficiently implemented in one representation), and other data one may wish to collect during runs. (It is sometimes useful to attach additional information to nodes, which may be easier to implement if they are explicitly

represented). These tree representations are the most common in GP, e.g., numerous high-quality, freely available GP implementations use them.

3.5.4 Initializing the Population

Like in other evolutionary algorithms, in GP the individuals in the initial population are typically randomly generated. There are many different approaches to generating this random initial population. Here, two of the simplest (and earliest) methods (the full and grow methods), and a widely used combination of the two known as Ramped half and-half will be described. In both the full and grow methods, the initial individuals are generated so that they do not exceed a user specified maximum depth. The depth of a node is the number of edges that need to be traversed to reach the node starting from the tree's root node (which is assumed to be at depth 0). The depth of a tree is the depth of its deepest leaf In the full method (so named because it generates full trees, i.e. all leaves are at the same depth) nodes are taken at random from the function set until the maximum tree depth is reached. (Beyond that depth, only terminals can be chosen.) Figure 3.7 shows a series of snapshots of the construction of a full tree of depth 2. The children of the * and / nodes must be leaves or otherwise the tree would be too deep. Thus, at both steps t = 3, t = 4, t = 6 and t = 7 a terminal must be chosen (x, y, 1 and 0, respectively).

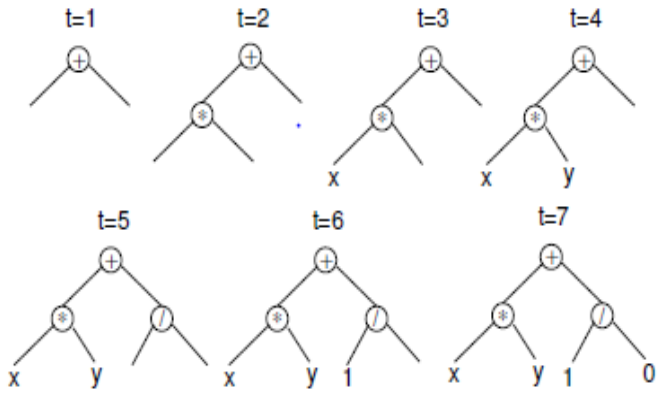


Figure 3.7 Creation of a full tree having maximum depth 2 using the full initialization method (t = time).

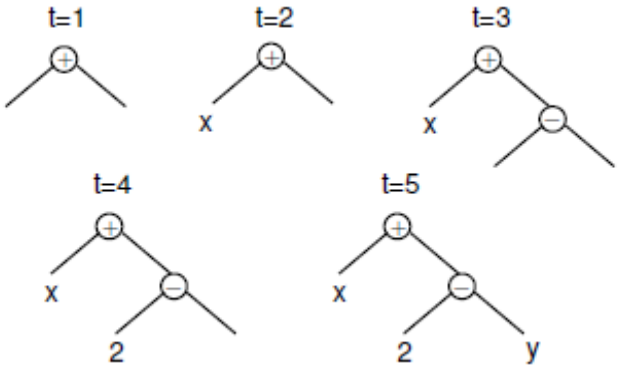


Figure 3.8 Creation of a full tree having maximum depth 2 using the grow method (t = time)

Although, the full method generates trees where all the leaves are at the same depth, this does not necessarily mean that all initial trees will have an identical number of nodes (often referred to as the size of a tree) or the same shape. This only happens, in fact, when all the functions in the primitive set have an equal arity. Nonetheless, even when mixed-arity primitive sets are used, the range of program sizes and shapes produced by the full method may be rather limited. The grow method, on the contrary, allows for the creation of trees of more varied sizes and shapes. Nodes are selected from the whole primitive set (i.e., functions and terminals) until the depth limit is reached. Once the depth limit is reached only terminals may be chosen (just as in the full method). Figure 3.8 illustrates this process for the construction of a tree with depth limit 2. Here the first argument of the + root node happens to be a terminal. This closes off that branch preventing it from growing any more before it reached the depth limit. The other argument is a function (-), but its arguments are forced to be terminals to ensure that the resulting tree does not exceed the depth limit.

Because neither the grow or full method provide a very wide array of sizes or shapes on their own, Koza (1992) proposed a combination called ramped half-and-half. Half the initial population is constructed using full and half is constructed using grow. This is done using a range of depth limits (hence the term "ramped") to help ensure that we generate trees having a variety of sizes and shapes. While these methods are easy to implement and use, they often make it difficult to control the statistical distributions of important

properties such as the sizes and shapes of the generated trees. For example, the sizes and shapes of the trees generated via the grow method are highly sensitive to the sizes of the function and terminal sets. If, for example, one has significantly more terminals than functions, the grow method will almost always generate very short trees regardless of the depth limit. Similarly, if the number of functions is considerably greater than the number of terminals, then the grow method will behave quite similarly to the full method. The arities of the functions in the primitive set also influence the size and shape of the trees produced by grow.

The initial population need not be entirely random. If something is known about likely properties of the desired solution, trees having these properties can be used to seed the initial population.

3.5.5 Selection

As with most evolutionary algorithms, genetic operators in GP are applied to individuals that are probabilistically selected based on fitness. That is, better individuals are more likely to have more child programs than inferior individuals. The most commonly employed method for selecting individuals in GP is tournament selection, which is discussed below, followed by fitness proportionate selection, but any standard evolutionary algorithm selection mechanism can be used.

In tournament selection many individuals are chosen at random from the population. These are compared with each other and the best of them is chosen to be the parent. When doing crossover, two parents are needed and, so, two selection tournaments are made. Note that tournament selection only looks at which program is better than another. It does not need to know how much better. This effectively automatically rescales fitness, so that the selection pressure[4] on the population remains constant.

Thus, a single extraordinarily good program cannot immediately swamp the next generation with its children; if it did, this would lead to a rapid loss of diversity with potentially disastrous consequences for a run. Conversely, tournament selection amplifies small differences in fitness to prefer the better program even if it is only marginally superior to the other individuals in a tournament.

3.5.6 Recombination and Mutation

GP departs significantly from other evolutionary algorithms in the implementation of the operators of crossover and mutation. The most commonly used form of crossover is subtree crossover. Given two parents, subtree crossover randomly (and independently) selects a crossover point (a node) in each parent tree. Then, it creates the offspring by replacing the subtree rooted at the crossover point in a copy of the first parent with a copy of the subtree rooted at the crossover point in the second parent, as illustrated in Figure 3.9. Copies are used to avoid disrupting the original individuals. This way, if selected multiple times, they can take part in the creation of multiple offspring programs. Often crossover points are not selected with uniform probability. Consequently, the uniform selection of crossover points leads to crossover operations frequently exchanging only very small amounts of genetic material (i.e., small subtrees); many crossovers may in fact reduce to simply swapping two leaves. To counter this, Koza (1992) suggested the widely used approach of choosing functions 90% of the time and leaves 10% of the time. Many other types of crossover and mutation of GP trees are possible.

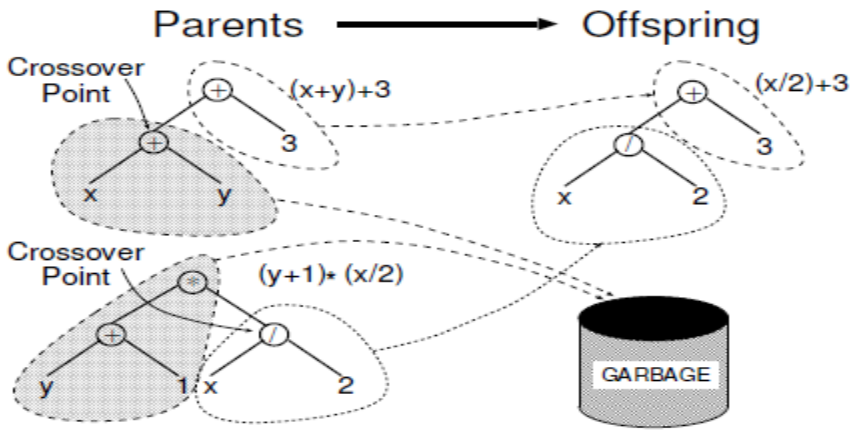


Figure 3.9 Example of subtree crossover.

The most commonly used form of mutation in GP (which will be called subtree mutation) randomly selects a mutation point in a tree and substitutes the subtree rooted there with a randomly generated subtree. This is illustrated in Figure 3.10.

Another common form of mutation is point mutation, which is GP's rough equivalent of the bit-flip mutation used in genetic algorithms (Goldberg,1989). In point mutation, a random node is selected, and the primitive stored there is replaced with a different random primitive of the same arity taken from the primitive set. If no other primitives with that arity exist, nothing happens to that node (but other nodes may still be mutated).

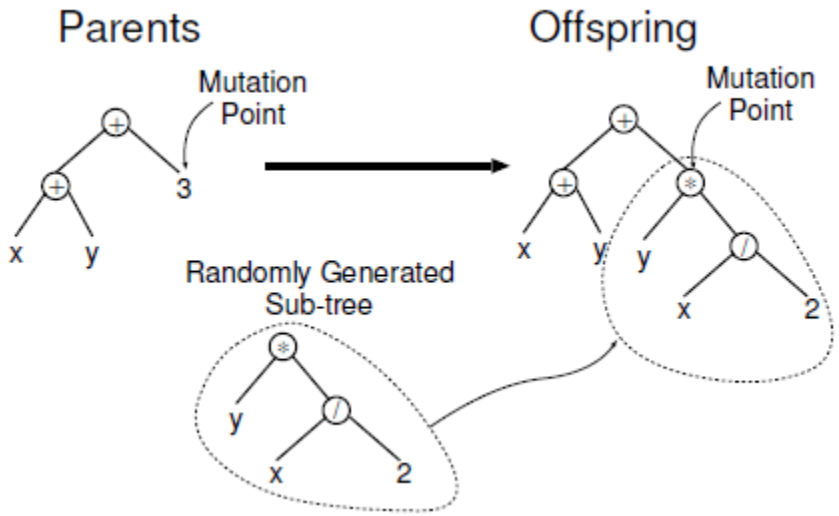


Figure 3.10 Example of subtree mutation.

When subtree mutation is applied, this involves the modification of exactly one subtree. Point mutation, on the other hand, is typically applied on a per-node basis. That is, each node is considered in turn and, with a certain probability, it is altered as explained above. This allows multiple nodes to be mutated independently in one application of point mutation.

The choice of which of the operators described above should be used to create an offspring is probabilistic. Operators in GP are normally mutually exclusive (unlike other evolutionary algorithms where offspring are sometimes obtained via a composition of operators). Typically, crossover is applied with the highest probability, the crossover rate often being 90% or higher. On the contrary, the mutation rate is much smaller, typically being in the region of 1%.

When the rates of crossover and mutation add up to a value p which is less than 100%, an operator called reproduction is also used, with a rate of 1 − p. Reproduction simply

involves the selection of an individual based on fitness and the insertion of a copy of it in the next generation.

3.5.7 Fitness Function for the genetic learning algorithm

The fitness function used for the algorithm is Root Mean Square Error (RMSE) between the actual depth values and predicted depth values. The basic idea behind the function is to calculate the error or deviation between each depth value and the corresponding predicted depth value and continue this calculation for all the data points in the generation. Then square of error is calculated for each data points for each possible solution in the generation and overall RMSE is calculated. The possible GP solutions will be those that have the best fitness, that is, the least value of RMSE. As a result, the solutions with the best fitness are carried to the next generation for further evolution.

3.5.8 Summary of Different parameters for the proposed genetic learning

Initial population size=100

Number of generations=70

Mutation probability=0.014

Tree grow method= ramp half and half method

Percent of elite individuals for next generation=5%

Selection type= Tournament Selection

Tournament size=20

Fitness function= Root Mean Square Error

3.6 Comparison metrics

3.6.1 MSE

In order to quantitatively assess and rate the algorithms, the mean squares error (MSE), rate of visible edges(e), gradient ratio(r) and SSM of the results are calculated for comparison. The MSE of each result can be calculated by the following equation:

$$MSE = \sqrt{\frac{1}{3N}} \sum_{c\epsilon[r,g,b]} ||J^c - G^c||^2 \quad (20)$$

where J is the dehazed image, G is the ground truth image, J^C represents a color channel of J, G^C represents a color channel of G, *N* is the number of pixels within the image G, and *e* is the MSE measuring the difference between the dehazed image J and the ground truth image G. Note that J and G have the same size since they are corresponding with the hazy image I. Given J and G, a low MSE represents that the dehazed result is satisfying while a high MSE means that the dehazing effect is not acceptable

3.6.2 SSIM

This metric maps two images to an interval [-1,1], where similar images have higher values. This metric is defined in Equation (4.4), where μ_A, μ_B, σ_A and σ_B are the values of mean and standard deviation for A and B, σ_{AB} is the covariance between A and B, $c_1=(k_1L)^2$ and $c_2=(k_2L)^2$, where L is the dynamic range of the pixel values ($2^{\text{bitsperpixel}}$-1) and $k_1=0.001$ and $k_2=0.03$ are constants.

$$\text{SSIM}(A, B) = \frac{(2\mu_A\mu_B + c_1)(2\sigma_{AB} + c_2)}{(\mu_A{}^2 + \mu_B{}^2 + c1)(\sigma_A{}^2 + \sigma_B{}^2 + c_2)} \quad (21)$$

3.6.3 Dehaze Descriptors

Among the few methods of dehazed image assessment, the most widely used is the indicator of visibility enhancement proposed by Hautiere et al[17] . The method includes two contrast descriptors namely the rate of new visible edges e and the gain of visibility level r. The value of e evaluates the ability of dehazing method to restore edges which were not visible in hazy image but are in dehazed image. The value of r denotes the mean ratio of gradient norms before and after dehazing. They are defined as follow:

$$e = \frac{n_r - n_o}{n_o} \quad (22)$$

$$r = \exp[\frac{\sum_{P_i\in\gamma} \log r_i}{n_r}] \quad (23)$$

Here, n_r and n_o denote respectively the numbers of visible edges in the dehazed image and in the original image. γ is the set of visible edge in the dehazed image. Pi denotes the

pixels that belong to γ . r_i is the ratio of the gradient at Pi in the dehazed image and in the original image.

An assessment method dedicated to visibility restoration proposed in [17] is used here to measure the dehazing effect. The color level of an image is first transformed to the gray level image, and two indicators are used to compare the two gray level images: the input image and the haze removal image. The visible edges in the image before and after restoration are selected by a 5% contrast threshold according to the meteorological visibility distance proposed by the international commission of illumination (CIE). To implement this definition of contrast between two adjacent regions, the method of visible edges segmentation proposed in [18] has been used.

3.7 Tools

Matlab 2016b is used for the implementation of the proposed dehazing method based on Genetic Learning Strategy.

3.8 Data collection

Collecting good training data is a crucial task for most learning-based applications. In the proposed framework, pairs of observed hazy image and the corresponding depth map d are required to train the model. However, unfortunately, a large scale of such data is extremely difficult to obtain especially in the outdoor environment since long range 3D depth is inaccurate, which demands time.

The images from dataset: LIVE Color+3D database [12] including 12 scenes with real depth acquired using an advanced range scanner are used for training and validation. The LIVE Color+3D Database consists of accurately co-registered color images and ground-truth range maps at high-definition resolutions, and the synthetic hazy images. The image and range data in the LIVE Color+3D Database were collected using an advanced range scanner, RIEGL VZ-400, with a Nikon D700 digital camera mounted on top of it. Since there are inevitable translational and rotational shifts when mounting the camera onto the range scanner, calibration is performed before data acquisition. The mounting calibration is done manually using the RIEG RiSCAN PRO software, which is designed for scanner operation and data processing. Next, to acquire the image and range data in natural scenes, the device obtains distances by lidar reflection and waveform analysis as it rotates, and

then the digital camera takes an optical photograph with the same field of view. The acquired range data are exported from the range scanner as point clouds with three-dimensional coordinates and range values, while the image data are stored in the digital camera. Finally, to obtain the aligned 2D range map with the 2D image, the 3D point clouds are projected and transformed into the 2D range map by applying the pinhole camera model with lens distortion. The natural environments where the image and range data were collected cover areas around Austin, Texas, including the campus at The University of Texas and the Texas State Capitol. In this work, 300 images have been used for the genetic learning and the 100 images from the dataset are used for validation of this work

CHAPTER 4 RESULTS AND DISCUSSION

Processor: Intel(R) Core(TM) i5-2430M CPU @ 2.40 GHz (4 CPUs),~2.4GHz

Memory: 4096 MB RAM

The basic idea behind the single image dehazing is recovery of original scene depth and the first step for dehazing is estimation of the depth values as discussed in the previous sections. First, the genetic learning strategy was used to obtain the relationship between the depth and the haze features; brightness and saturation.

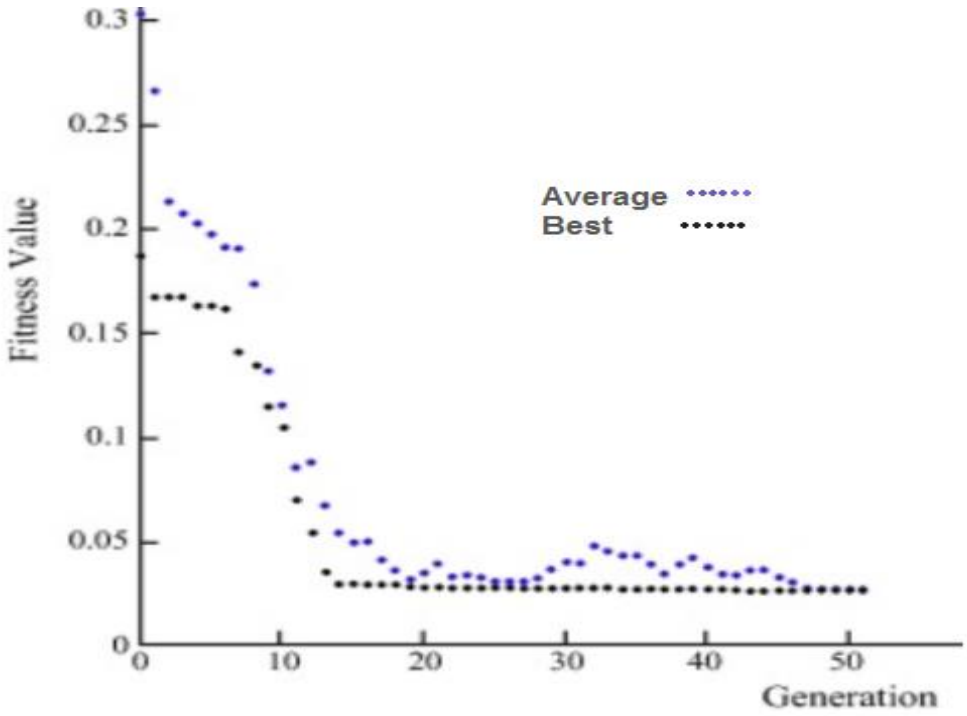


Figure 4.1 Fitness plot for genetic learning

The Figure 4.1 shows a fitness curve obtained during genetic learning stage. From the Figure 4.1, when analyzing average fitness, it's somewhat surprising to see the huge jump from the first generation to the second; however, this is the first trained generation to work with the fitness function; each time the algorithm selects a random parent it selects the more fit out of the two possibilities surrounding a randomly generated population. When this is performed for an entire generation, it is not surprising there would be substantial growth, especially in the first generation where lots of the worst candidates are thrown out.

The graph of the most fit individual seems expected as well; despite each generation starting at over 0.17 output for the fitness function, the most fit individual has a linear downward slope in reaching higher fitness.

The expression for the depth map obtained using the genetic learning strategy is

$$d(x) = \frac{1}{1 + e^{-(a1 + a2V(x) + a3 * S(x))}} \tag{24}$$

Where V(x) is the brightness vector and S(x) is the saturation vector

a1=1.44, a2=4.14, a3=5.29

The expression for the depth map obtained using the CAP method that uses probabilistic learning strategy is

$$d(x) = a0 + a1 * V(x) + a2 * S(x) \tag{25}$$

Where V(x) is the brightness vector and S(x) is the saturation vector

a0=0.122, a1=0.9597, a2=-0.7802

The above obtained results for depth were used to estimate the depth from a given input hazy image. First, the brightness and saturation component of the hazy image are extracted and placed in the expression that is defined by the depth equation to estimate the depth values. Once the depth values for each pixel are calculated, the original image can be easily recovered as discussed in the previous sections. The method was also tested for various values of β.

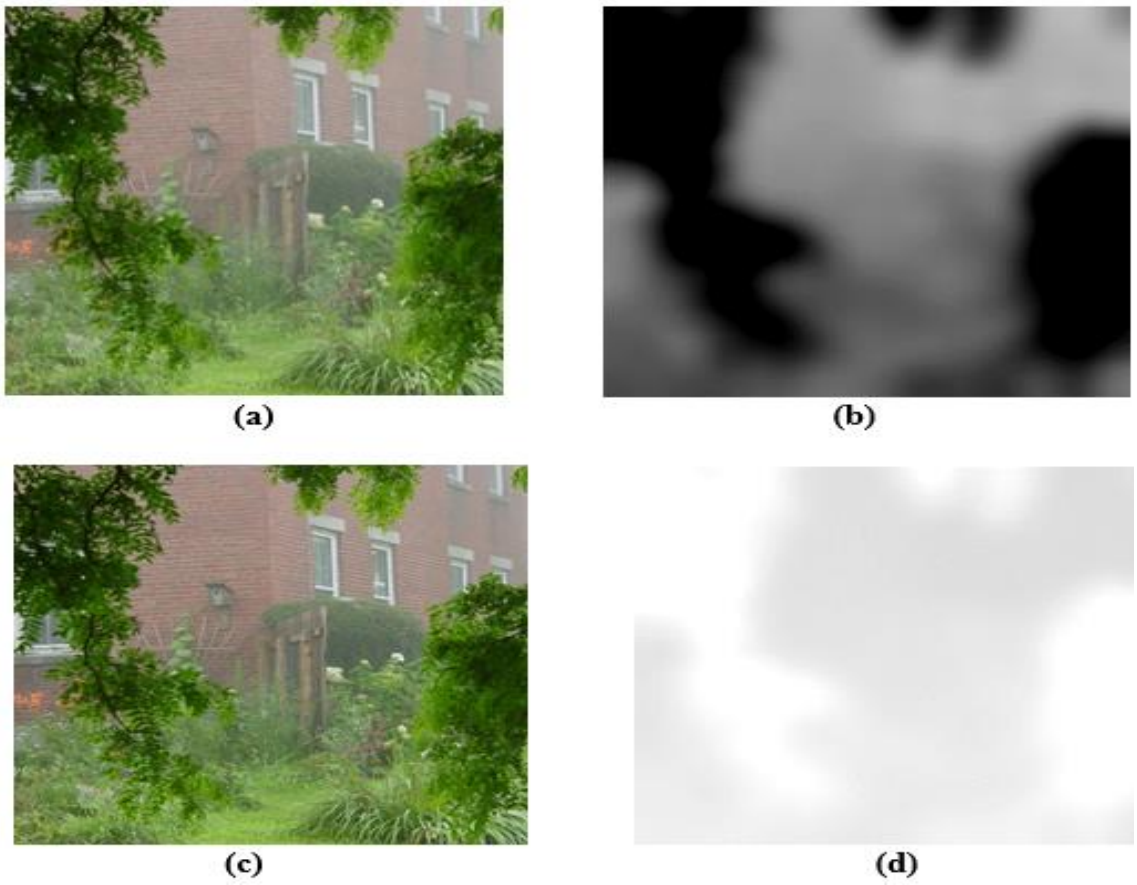


Figure 4.2 Example of dehazing using proposed genetic learning method with β=0.2

(a) Hazy image (b) Estimated depth map (c) Dehazed image (d) Estimated transmission map

Figure 4.3 shows the dehazing results obtained using β=0.2. Figure (a) represents the hazy image, figure (b) represents the estimated depth of the scene using the proposed learning strategy. Figure (c) shows the dehazed image and finally figure (d) shows the restore transmission obtained using the estimated depth.

Depth map is a kind of image which is composed of the gray pixels defined by 0 to 255 values. The 255 values of gray pixels stand for that 3D pixels are located most distant place in the 3D scene while the 0 value of gray pixels stand for the 3D pixels are located at the nearest place. In depth map, each depth pixel would define the position in Z-axis where its corresponding 2D pixel will be located. The two learning algorithms produce an approximate depth map and transmission map that are used to recover the original image

as described by the equation discussed before. However, it should be noted that the two learning models do not exactly estimate the depth map, but they give the approximate depth map, because it is a very challenging task in the field of computer vision to produce the accurate depth map of an image using only single image.

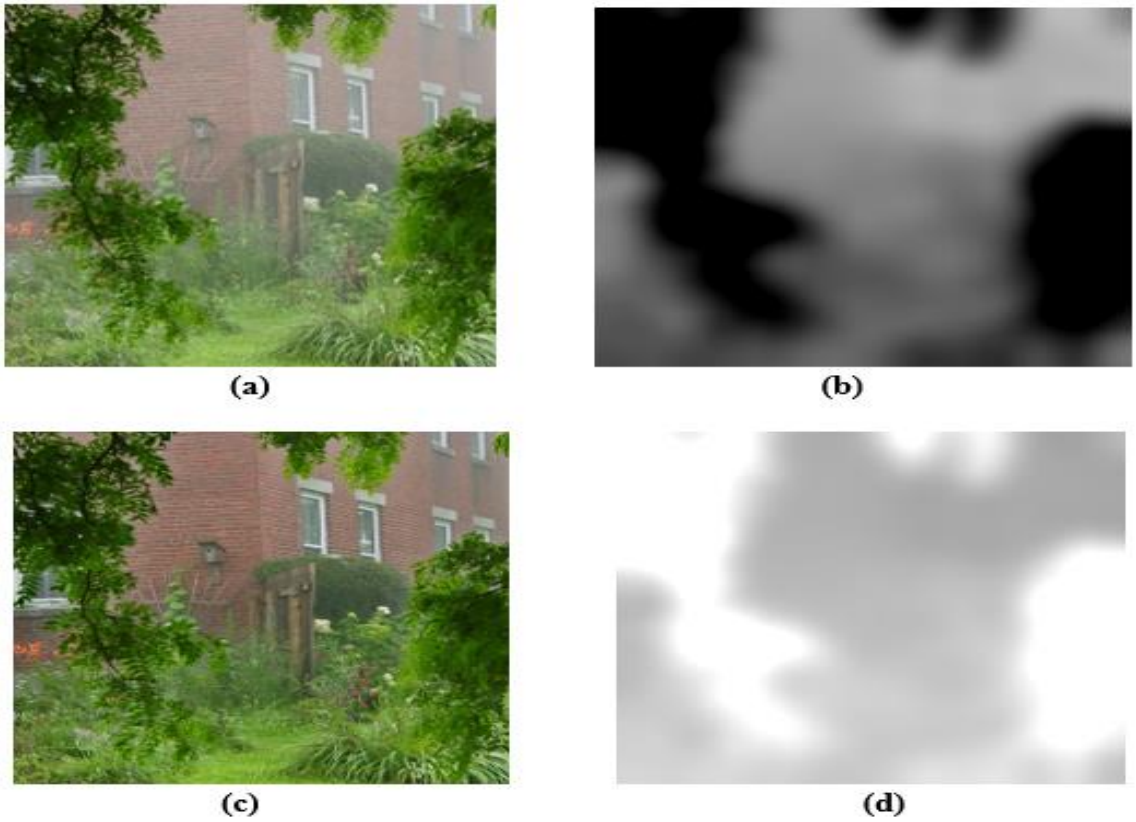


Figure 4.3 Example of dehazing using proposed genetic learning method with β=0.6

(a) Hazy image (b) Estimated depth map (c) Dehazed image (d) Estimated transmission map

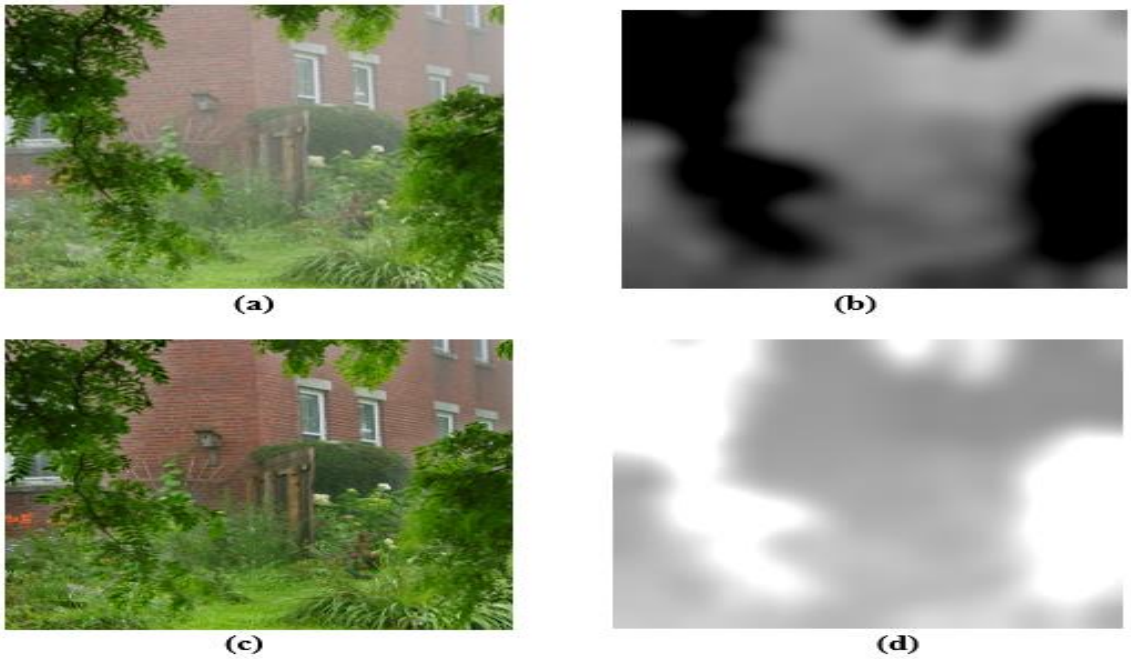


Figure 4.4 Example of dehazing using proposed genetic learning method with β=0.8

(a) Hazy image (b) Estimated depth map (c) Dehazed image (d) Estimated transmission map

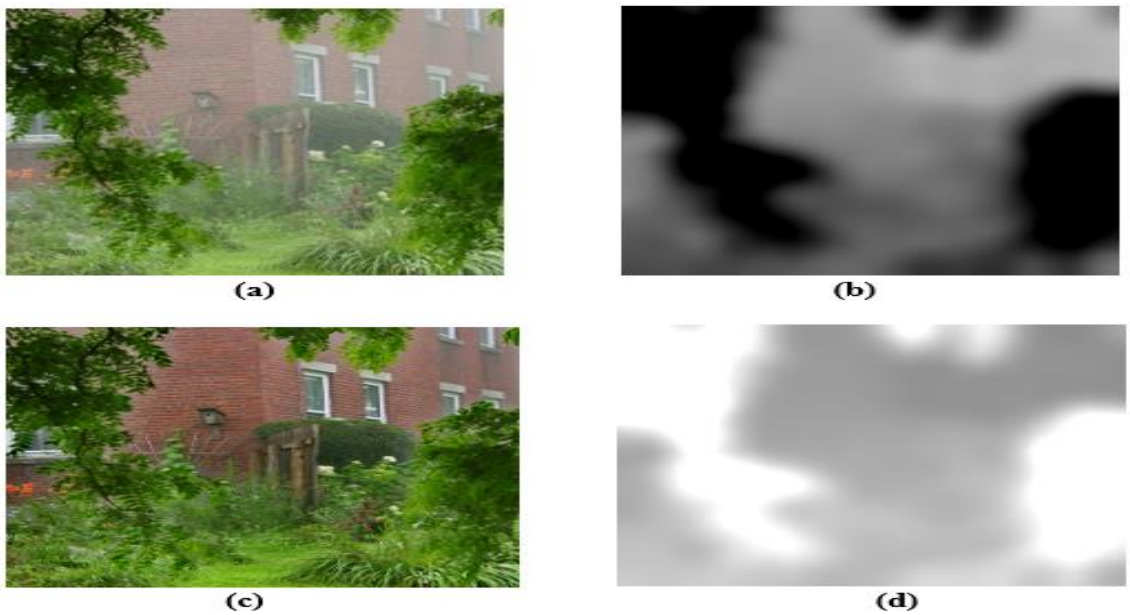


Figure 4.5 Example of dehazing using proposed genetic learning method with β=1

(a) Hazy image (b) Estimated depth map (c) Dehazed image (d) Estimated transmission map

Figure 4.4 ,4.5 and 4.6 show the results of dehazing using different values of β=0.6,0.8 and 1 along with their corresponding restored depth and transmission of the scenes.

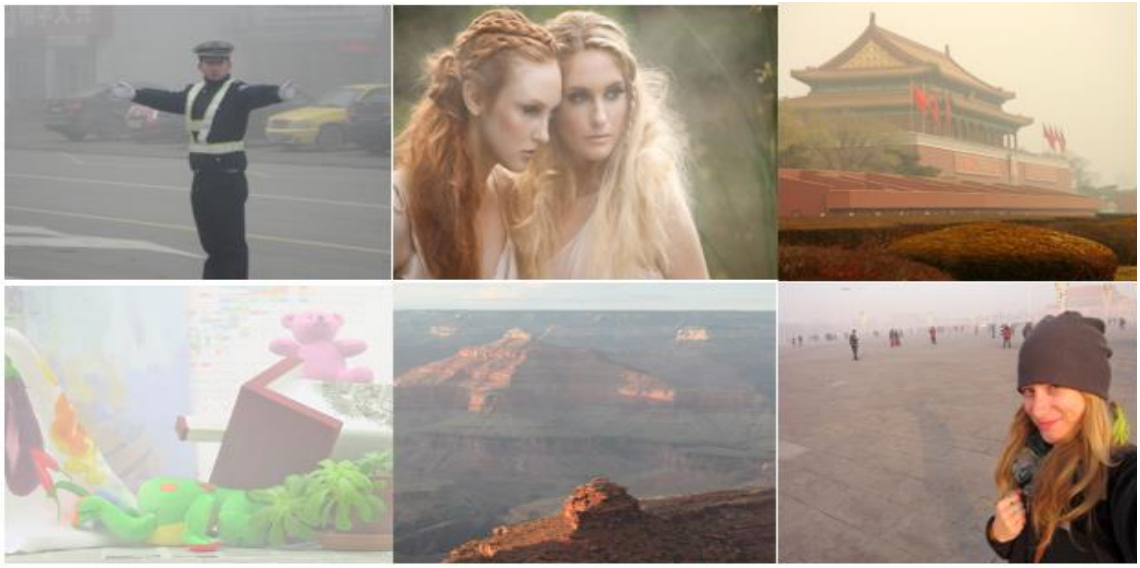


Figure 4.6 Hazy images examples

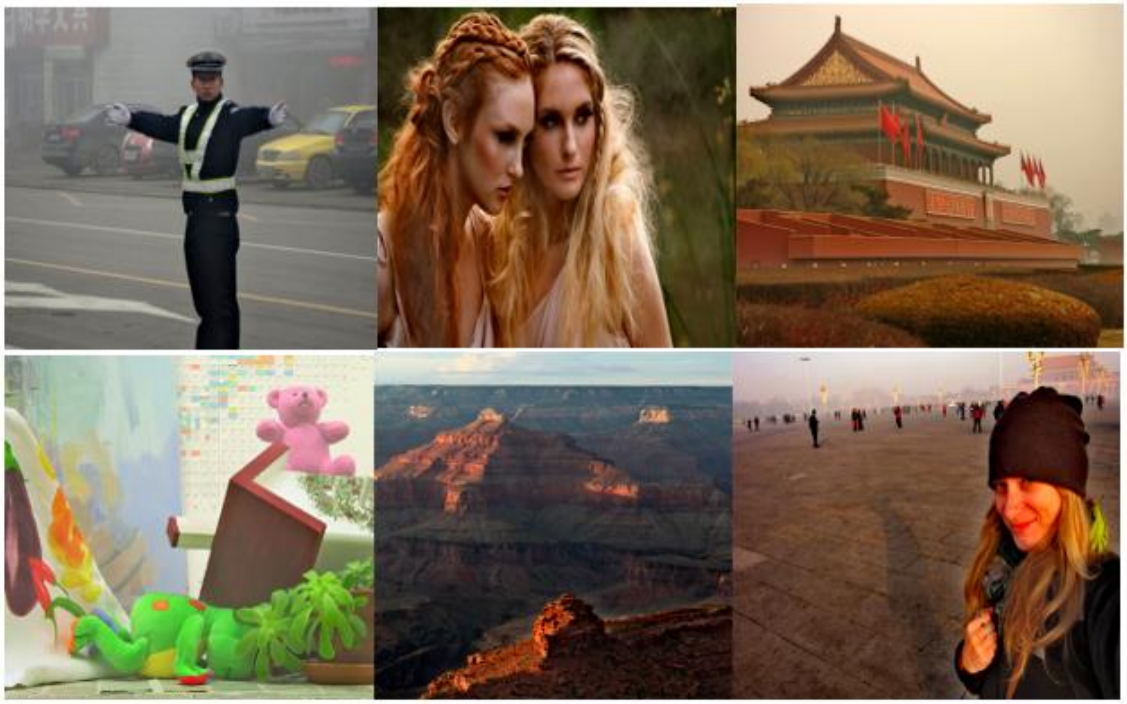


Figure 4.7 Dehazed images examples using proposed genetic learning

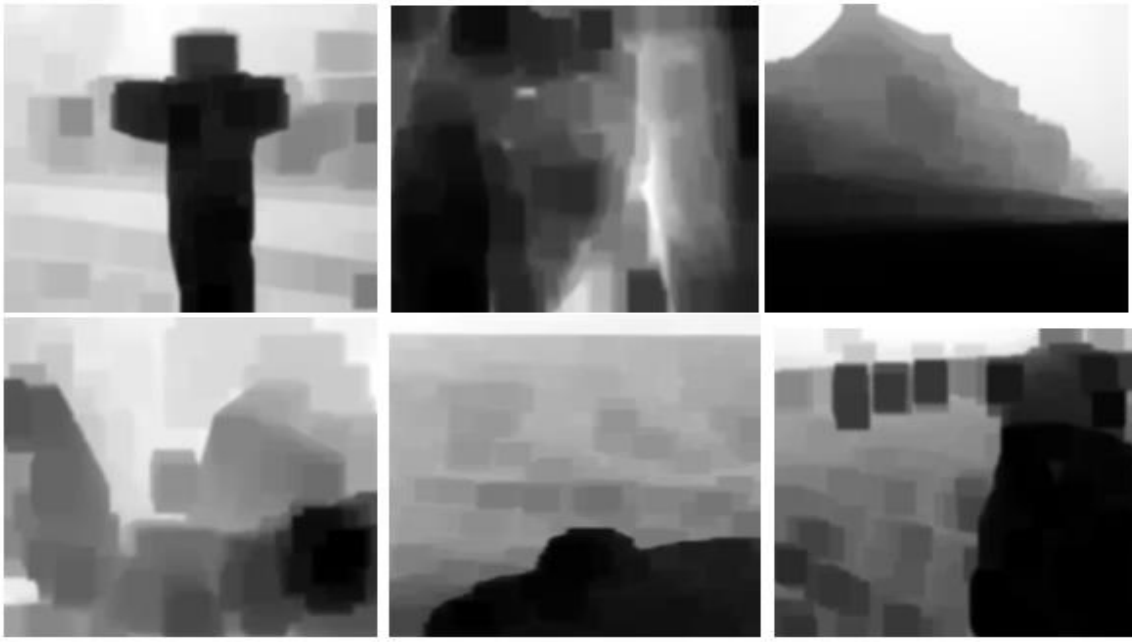


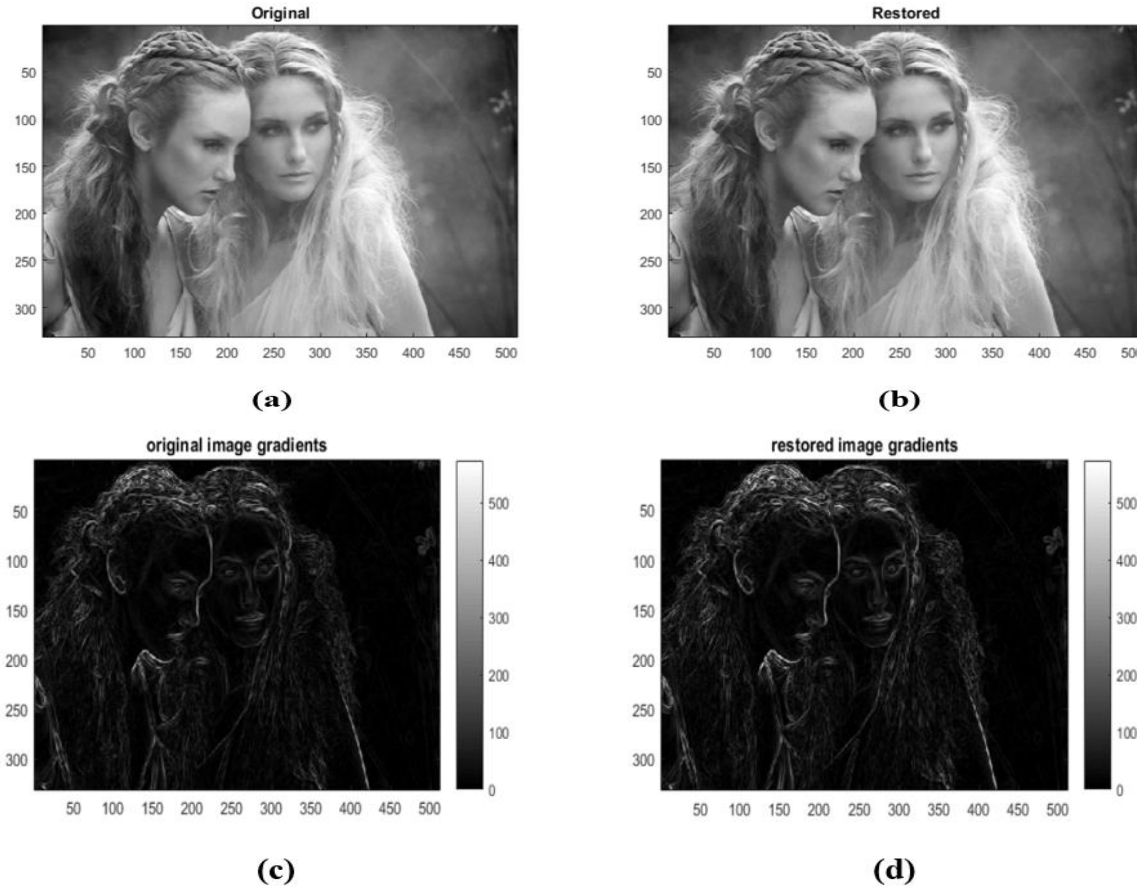
Figure 4.8 Estimated depth examples using proposed genetic learning method

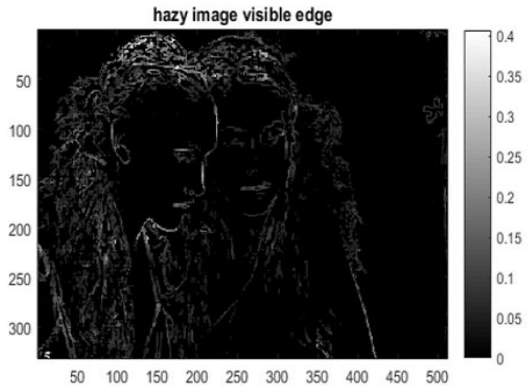
Figure 4.9 Restored transmission examples using the proposed genetic learning method

Figure 4.7 shows other different hazy images, figure 4.8 shows the corresponding dehazed image and figure 4.9 shows the estimated depth map and figure 4.10 shows the restored

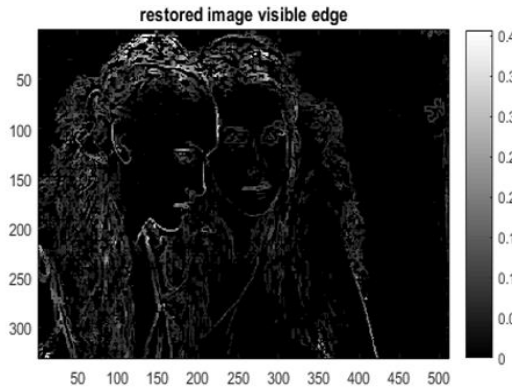
transmission using the proposed learning model. It can be observed from the depth images that where there is high value of points in the depth map, there is low value of point in the corresponding transmission map since the two maps are negative exponentially related as discussed in the previous section. In order words, bright values in the depth map correspond to points far from the view reference whereas, dark values mean points are near to the viewing reference. In the case of transmission map, dark values mean points far to the viewing reference and bright values means points near to the viewing reference.

4.1 Quantitative evaluation of the proposed dehazing method

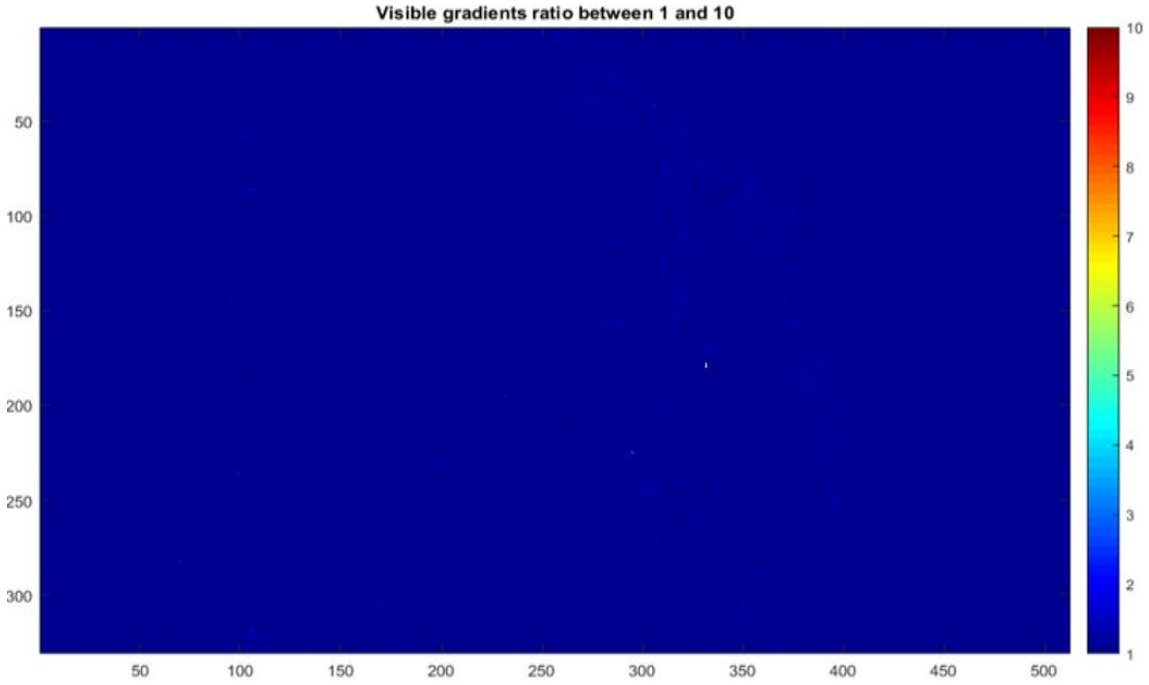



hazy image visible edge
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0

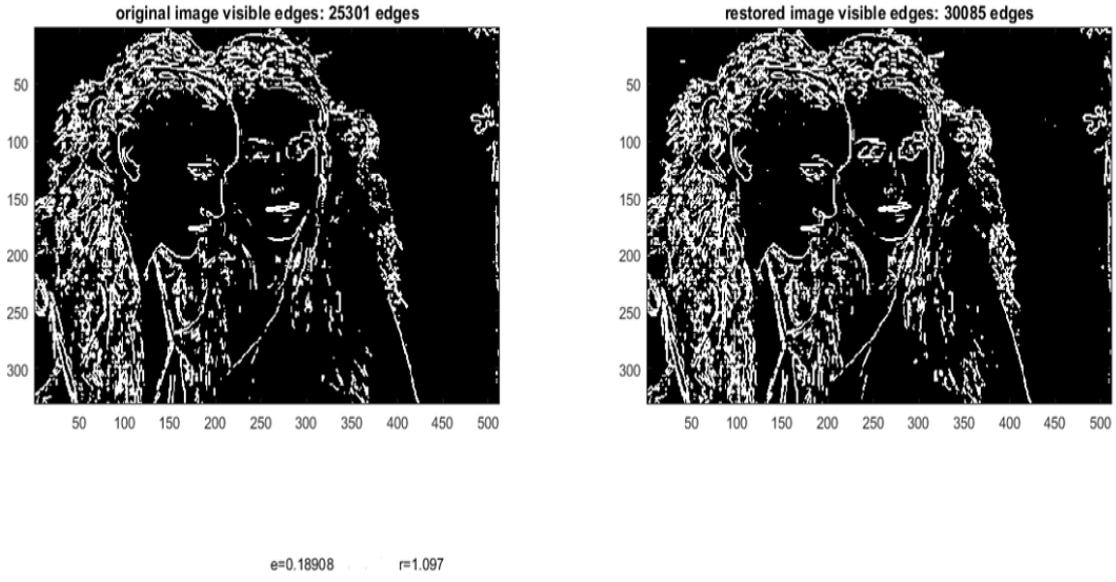
(e)


restored image visible edge
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
0

(f)


Visible gradients ratio between 1 and 10
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
10
9
8
7
6
5
4
3
2
1

(g)

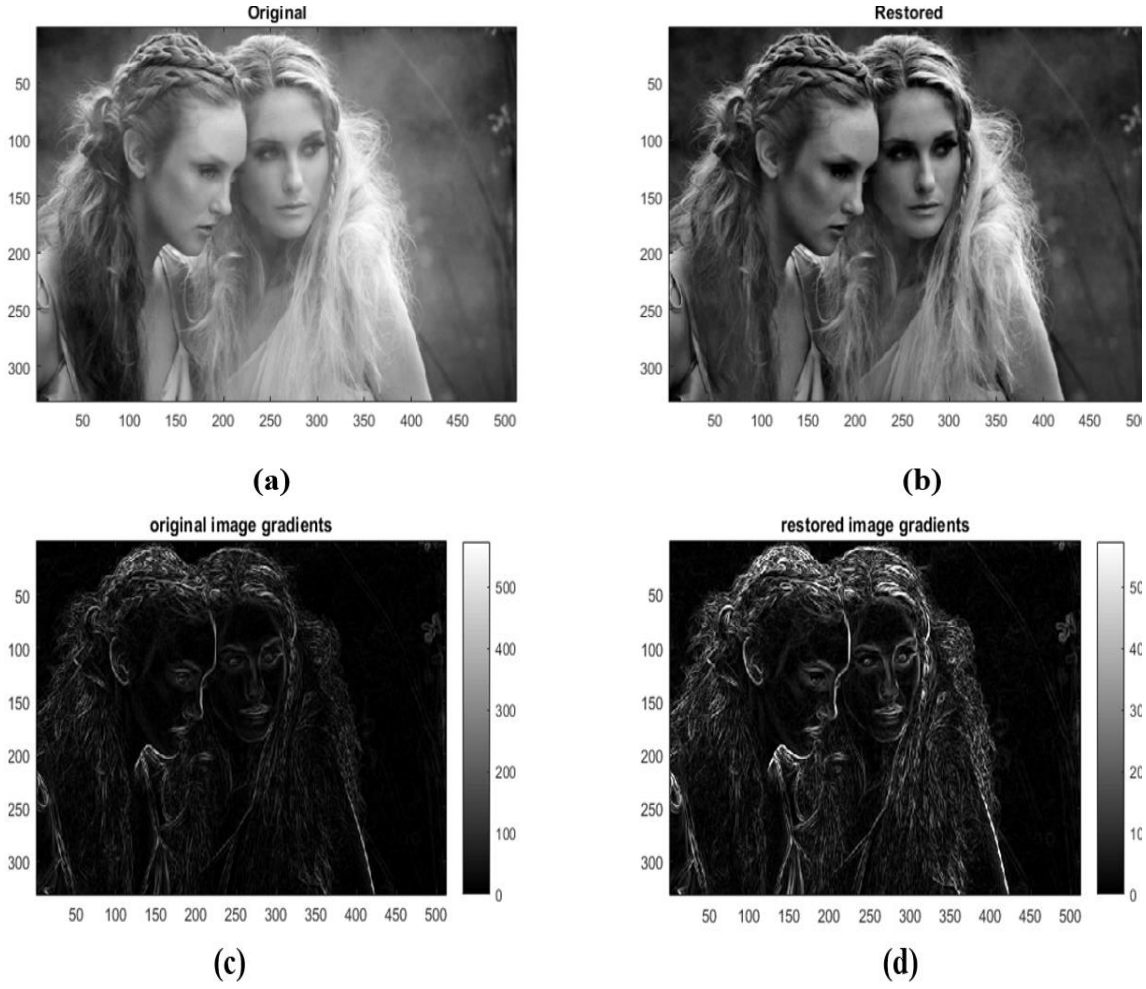


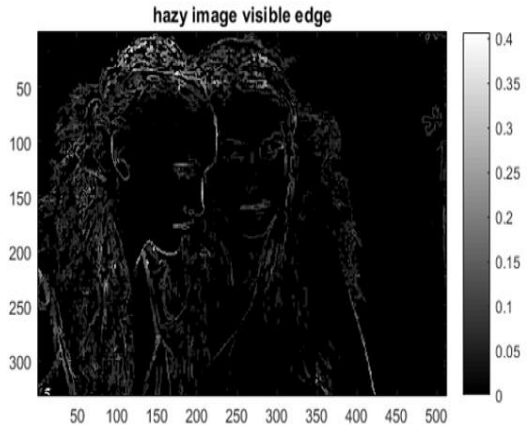
(h)

Figure 4.10 Quantitative evaluation of the proposed algorithm for β=0.2

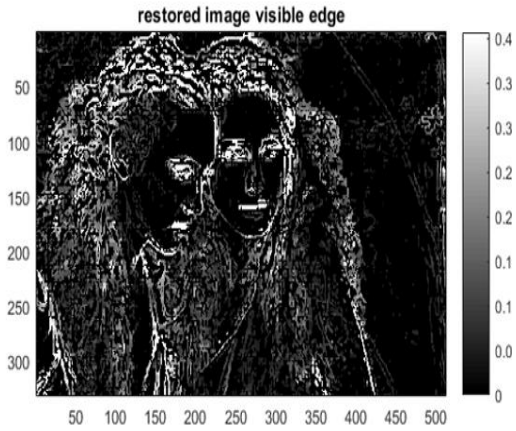
Figure 4.11 shows the steps involved in the calculation of the haze descriptors e and r. Firstly, the corresponding hazy and dehazed color image is converted into grayscale image as shown in figure (a) and (b). Figure (c) and (d) shows the gradient of original haze image and dehazed image. It should be noted that the descriptors e and r blind image assessment parameters and the ground truth image is not required for their calculation. Once the map of visible edges is obtained, the rate(e) of edges newly visible after restoration can be computed. Then, the mean r over these edges of the ratio of the gradient norms after and before restoration is computed. This indicator r estimates the average visibility enhancement obtained by the restoration algorithm. As seen in the figure (h), the number of visible edges after dehazing is larger than the number of visible edges in the original hazy image. This results in the value of e=0.18508 and r=1.097. The value of e=0 means that there is no improvement in the enhanced image in terms of number of visible edges by using the method to dehaze the image. In other words, the dehazed and original hazy image have same number of visible edges. The negative value of e indicates the result has degraded whereas the positive value of e indicates that the result is improved. In the given example of dehazing, the value of e=0.18508 means that the restored dehazed image has larger number of visible edges than the original hazy image, thus, the result is improved.

This indicator r estimates the average visibility enhancement obtained by the restoration algorithm and the value of r>1 indicates that the visibility enhancement is obtained by using the proposed algorithm. For no visibility enhancement, the value of r becomes equal to unity.

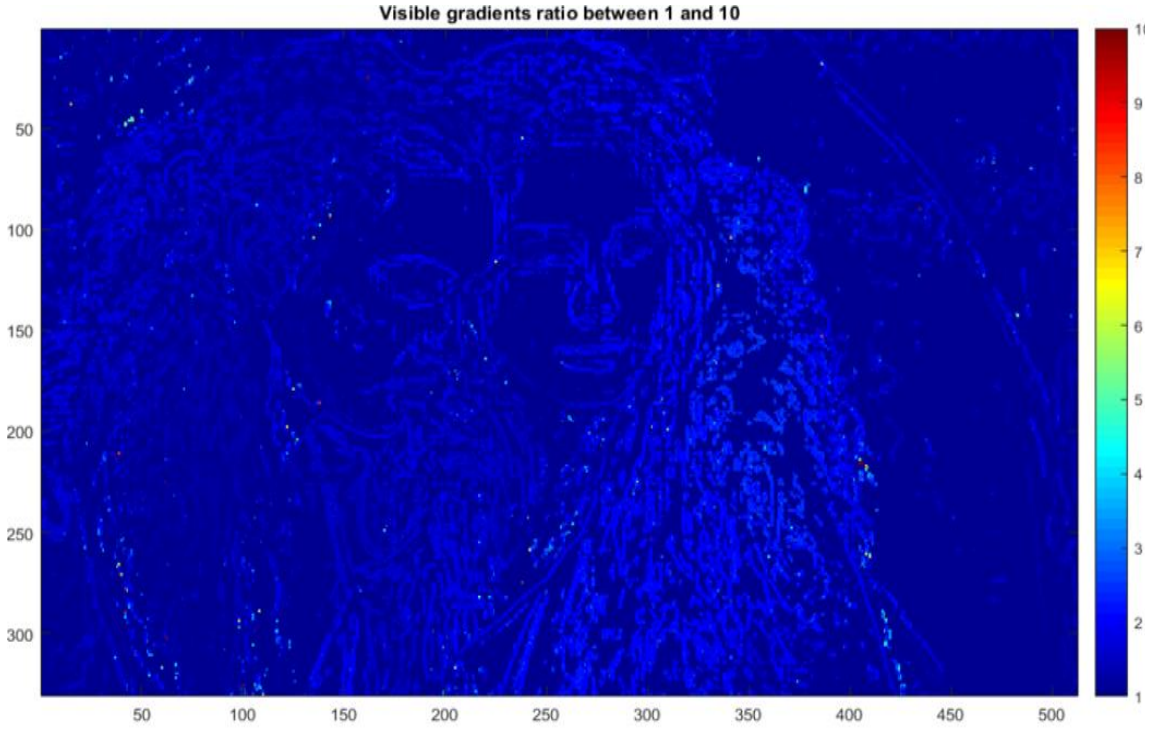



hazy image visible edge
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0

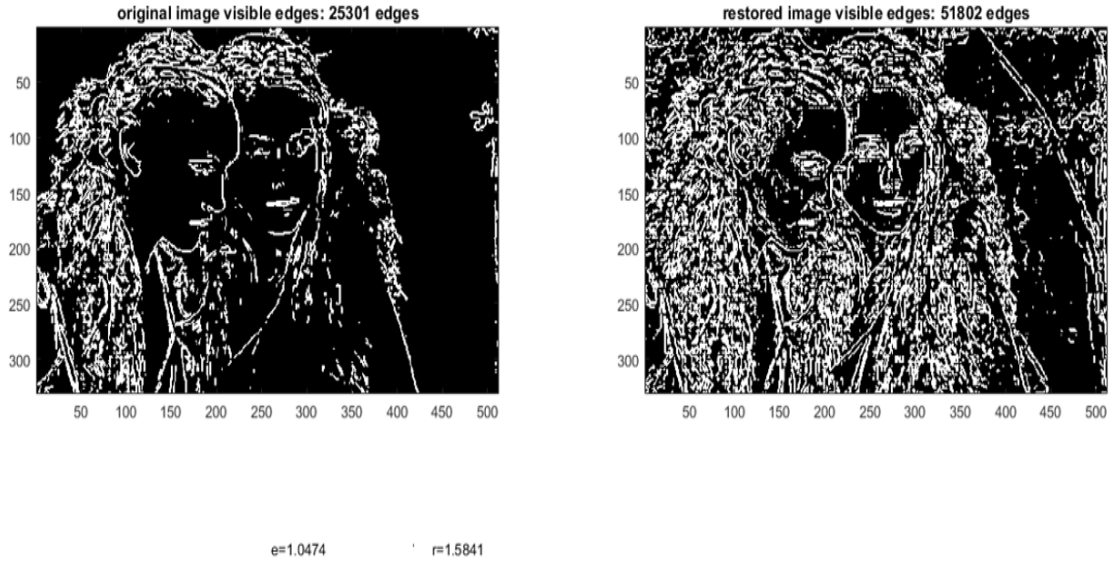
(e)


restored image visible edge
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
0.4
0.3
0.3
0.2
0.2
0.1
0.1
0.0
0

(f)


Visible gradients ratio between 1 and 10
50
100
150
200
250
300
50 100 150 200 250 300 350 400 450 500
9
8
7
6
5
4
3
2
1

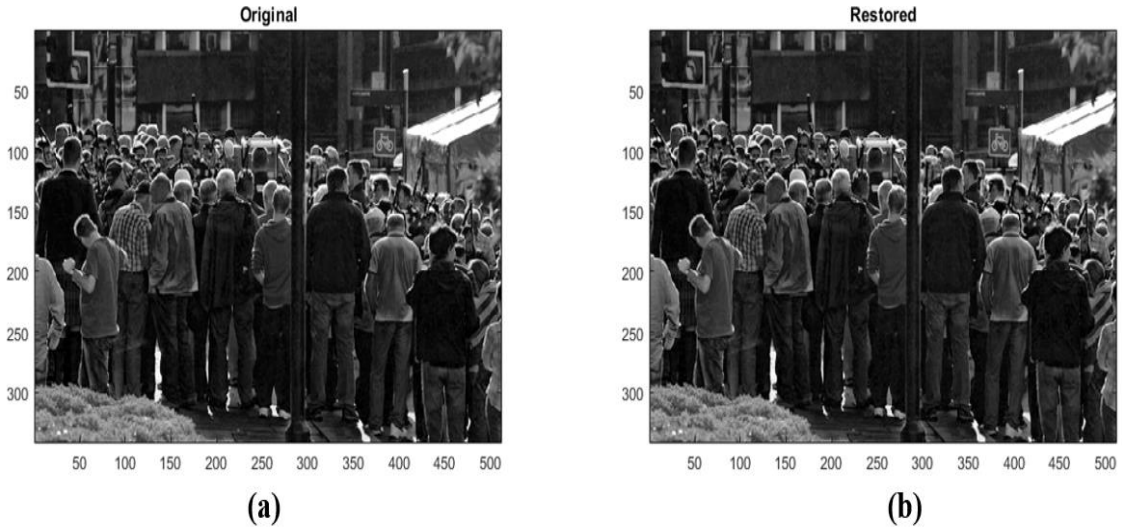
(g)



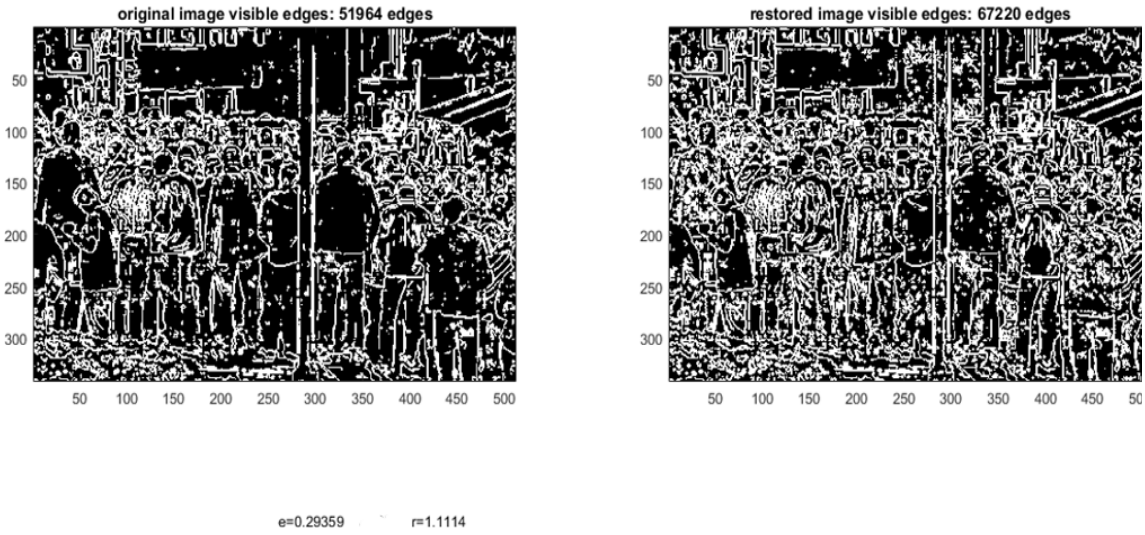
(h)

Figure 4.11 Quantitative evaluation of the proposed method for β=1

Figure 4.12 shows the evaluation result of the dehazing method for the same image of figure 4.11 using different value of scattering coefficient. Figure 4.13 and figure 4.14 shows the evaluation for the other images. The results obtained show that the e>0 and r>1 for all these images. Thus, the number of visible edges and visibility enhancement was possible for these images using the proposed algorithm.



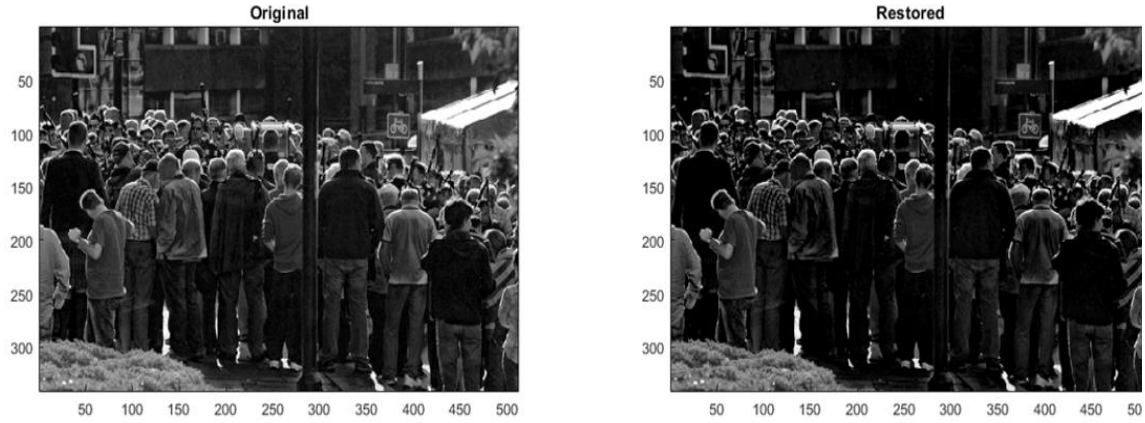
(a) (b)



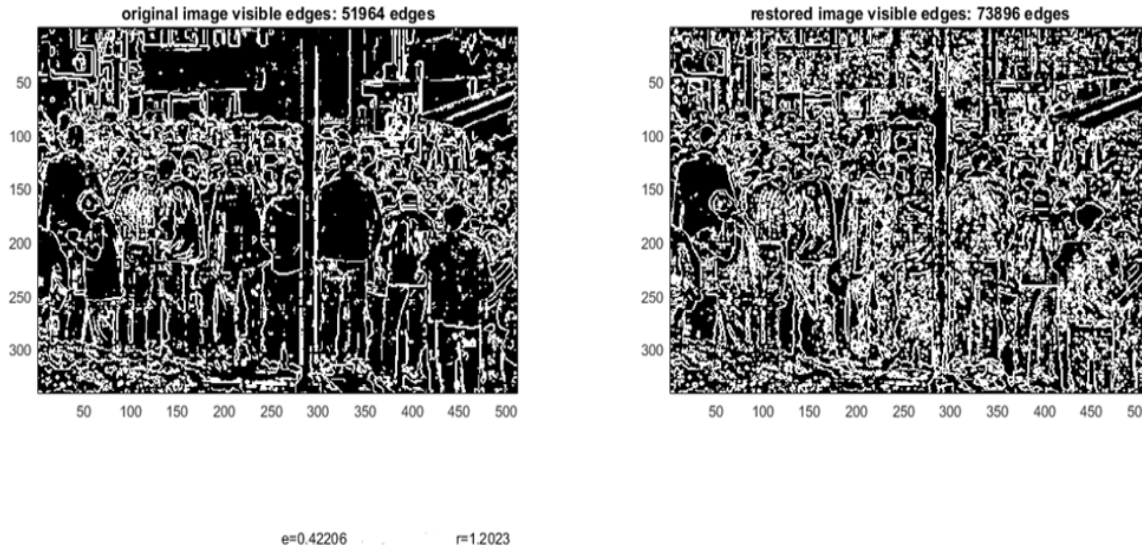
(c)

Figure 4.12 Quantitative evaluation of the proposed method for β =0.2

Figure 4.12 shows the dehazing performance of the proposed method for the given image in terms of e and r descriptors taking the value of scattering coefficient to be equal to 0.2. The value of e=0.295 and r=1.1154 proves that visual quality of the dehazed image is improved by using the proposed method for the given image.



(a) **(b)**



(c)

Figure 4.13 Quantitative evaluation of the proposed method for β =1

4.2 Effect of variation of scattering coefficient on dehazed images

Figure 4.14 Test images for studying the effect of β variation

The 10 test haze images were dehazed by taking the different values of scattering coefficient β. The value of β was varied from 0 to 2 and the dehazed images were obtained for each corresponding value and each test image. The value of e, r and SSIM were noted and the plot was drawn between e, r and SSIM vs β to study how the β variation affects the different visual, visible edges and structural information of dehazed images.

Table 4.1 values of e for different values of β for 10 test images

β	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
test1	0	0.273	0.617	1.016	1.377	1.795	2.358	3.607	3.759	4.525	4.248
test2	0	0.164	0.321	0.509	0.735	1.133	1.856	2.298	2.179	1.689	1.519
Test3	0	0.042	0.088	0.138	0.197	0.258	0.332	0.323	0.264	0.157	0.073
Test4	0	0.084	0.209	0.356	0.582	0.912	1.253	1.911	2.127	2.459	2.483
Test5	0	0.219	0.501	0.889	1.466	2.202	3.004	3.602	4.444	3.732	2.559
Test6	0	0.152	0.326	0.546	0.823	1.401	2.150	3.147	3.237	3.159	3.015
Test7	0	0.164	0.367	0.592	0.822	1.086	1.358	1.983	2.775	3.118	3.389
Test8	0	0.124	0.283	0.446	0.690	1.129	1.511	1.539	1.853	1.765	1.791
Test9	0	0.159	0.357	0.585	0.853	1.224	1.978	2.371	2.704	3.135	3.085
Test10	0	0.120	0.250	0.381	0.591	0.893	1.314	2.047	3.016	2.988	1.984

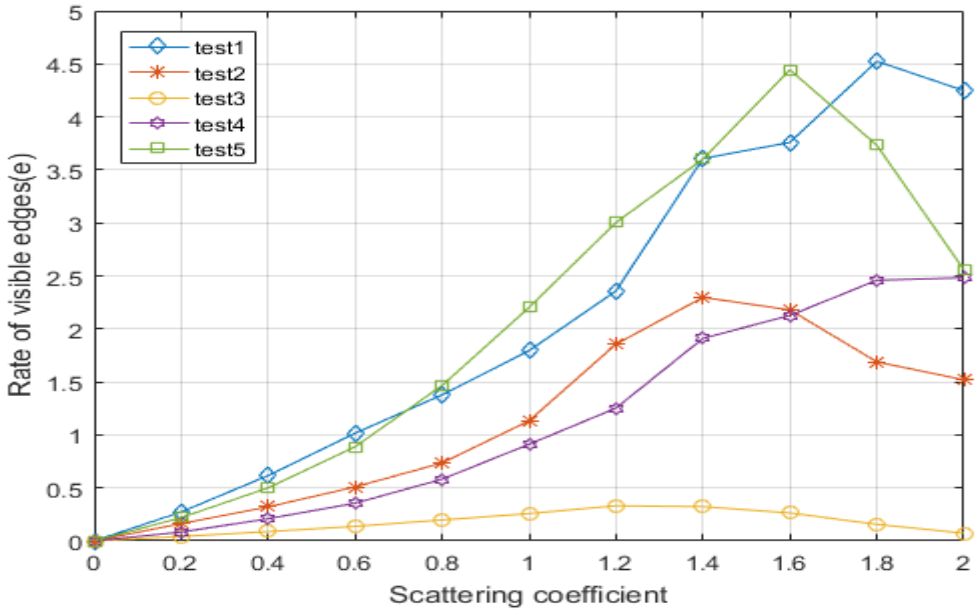


Figure 4.15 e vs β graph for first five test images

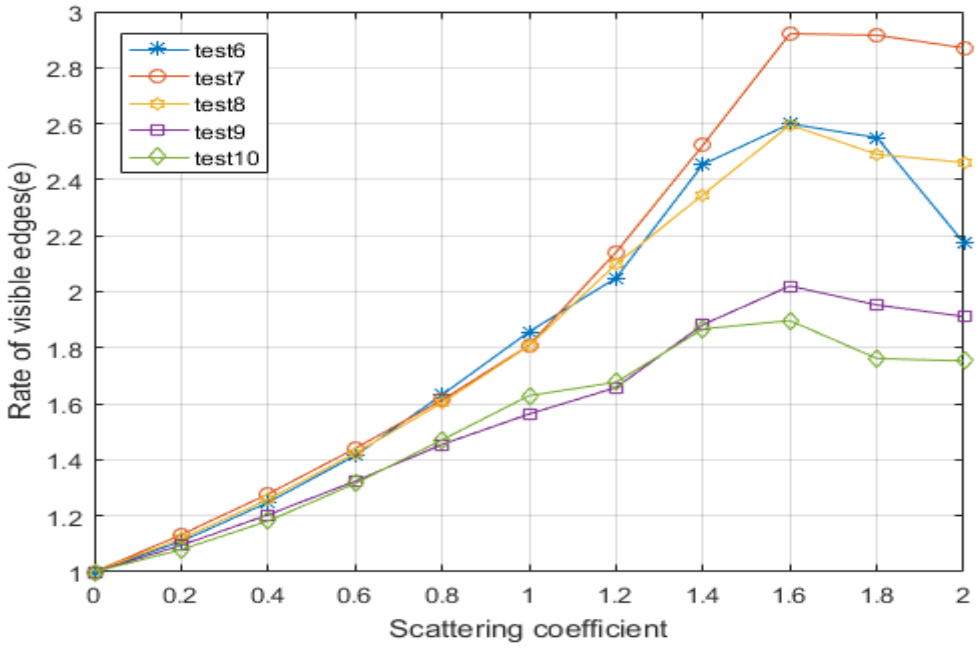


Figure 4.16 e vs β graph for second five test images

Figure 4.16 and Figure 4.17 show the results of varying scattering coefficient on the rate of visible edges (e) for 10 different test images. It can be observed from the above graph

that the value of e goes on increasing as the scattering coefficient increases and at a certain value of scattering coefficient, e started to decrease as the scattering coefficient is further increased. For example, for test image 1 and test image 2, the value of e reaches maximum when β=1.4, and its value decreases after this particular value of β.

Table 4.2 values of e for different values of β for 10 test images

β	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
test1	1	1.172	1.371	1.608	1.869	2.159	2.491	2.776	2.893	2.779	2.346
test2	1	1.101	1.21	1.315	1.443	1.575	1.685	1.729	1.581	1.414	1.339
test3	1	1.002	1.003	1.006	1.014	1.016	0.979	0.917	0.869	0.849	0.828
test4	1	1.092	1.199	1.320	1.456	1.577	1.717	1.884	1.911	1.864	1.707
test5	1	1.108	1.248	1.415	1.632	1.857	2.049	2.455	2.599	2.550	2.175
test6	1	1.133	1.278	1.441	1.613	1.809	2.141	2.523	2.921	2.916	2.871
test7	1	1.118	1.259	1.423	1.604	1.807	2.098	2.347	2.595	2.491	2.461
test8	1	1.095	1.204	1.323	1.455	1.564	1.658	1.883	2.020	1.953	1.912
test9	1	1.079	1.182	1.315	1.471	1.629	1.677	1.867	1.897	1.762	1.753
test10	1	1.057	1.131	1.233	1.340	1.488	1.710	1.914	1.977	1.777	1.726

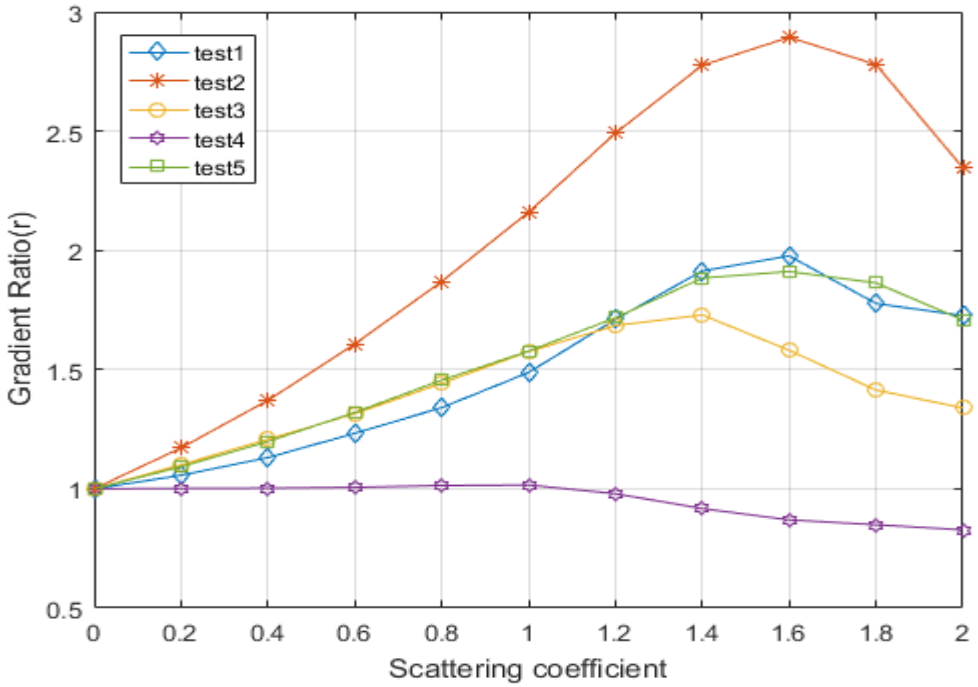


Figure 4.17 r vs β graph for first five test images

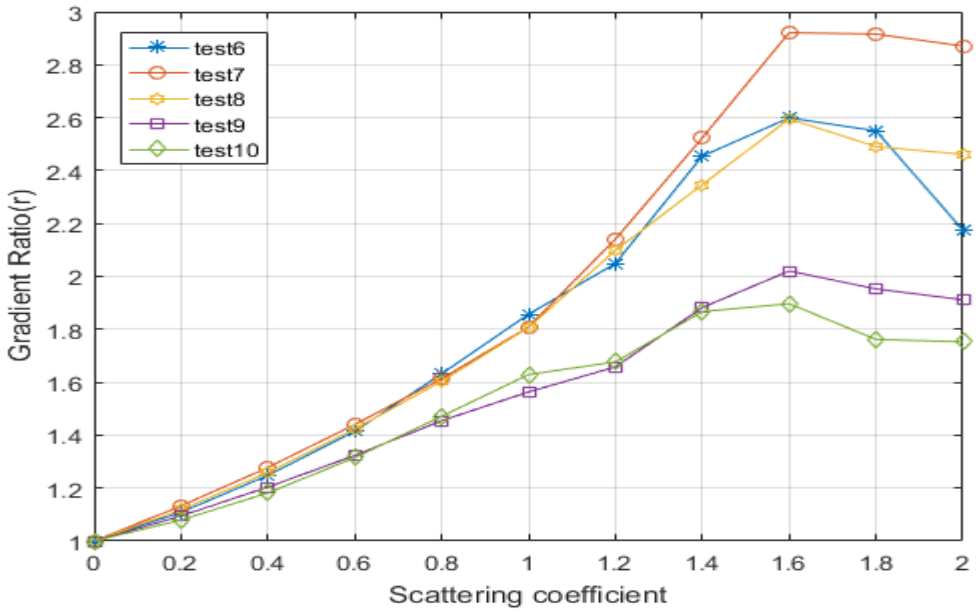


Figure 4.18 r vs β graph for second five test images

Figure 4.18 and Figure 4.19 show the results of varying scattering coefficient on the gradient ratio(r) for 10 different test images. It can be observed from the above graph that the value of e goes on increasing as the scattering coefficient increases and at a certain value of scattering coefficient, e started to decrease as the scattering coefficient is further increased. For example, for test image 2 and test image 6, the value of r reaches maximum when β=1.6, and its value decreases after this particular value of β.

Table 4.3 values of SSIMs for different values of β for 10 test images

β	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
test1	0.356	0.401	0.448	0.501	0.559	0.620	0.706	0.684	0.653	0.602	0.478
test2	0.638	0.692	0.744	0.791	0.826	0.842	0.755	0.567	0.435	0.349	0.282
test3	0.764	0.796	0.822	0.822	0.823	0.752	0.616	0.486	0.389	0.322	0.271
test4	0.588	0.616	0.647	0.68	0.712	0.738	0.740	0.716	0.649	0.559	0.454
test5	0.446	0.473	0.505	0.541	0.579	0.619	0.664	0.664	0.664	0.563	0.525
test6	0.505	0.534	0.567	0.605	0.649	0.696	0.706	0.679	0.646	0.626	0.616
test7	0.695	0.730	0.763	0.791	0.811	0.815	0.784	0.714	0.591	0.446	0.307
test8	0.487	0.537	0.592	0.652	0.715	0.778	0.792	0.738	0.632	0.494	0.381
test9	0.366	0.398	0.437	0.483	0.534	0.593	0.625	0.597	0.585	0.553	0.5
test10	0.493	0.534	0.579	0.630	0.684	0.721	0.72	0.697	0.578	0.451	0.374

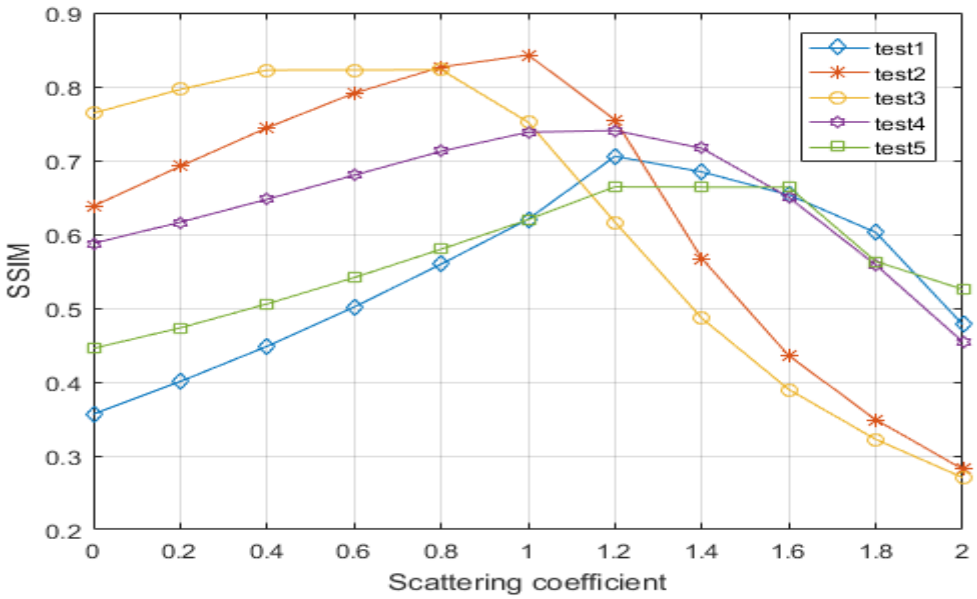


Figure 4.19 SSIM vs β graph for first five test images

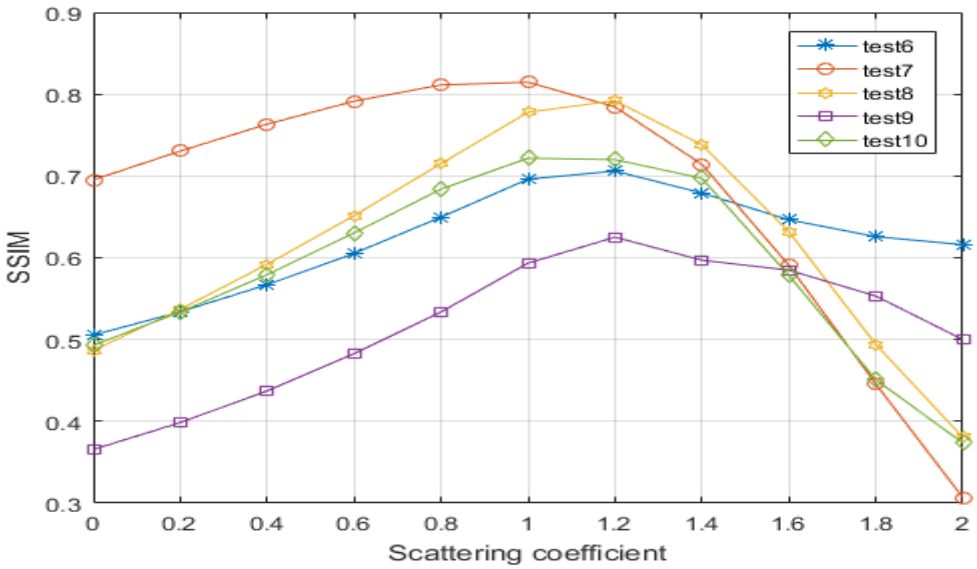


Figure 4.20 SSIM vs β graph for second five test images

Figure 4.20 and Figure 4.21 show the results of varying scattering coefficient on the r) SSIM for 10 different test images. It can be observed from the above graph that the value of SSIM goes on increasing as the scattering coefficient increases and at a certain value of

scattering coefficient, e started to decrease as the scattering coefficient is further increased. For example, for test image 2 and test image 7, the value of r reaches maximum when β=1, and its value decreases after this particular value of β.

4.3 Subjective Comparison with the CAP method

(a) (b) (c)

Figure 4.21 Subjective comparison of dehazing methods

(a) Hazy Image (b) Proposed method using genetic learning (c) CAP method

(a) (b) (c)

Figure 4.22 Subjective comparison of dehazing methods

(a) Hazy Image (b) Proposed method using genetic learning (c) CAP method

(a) (b) (c)

Figure 4.23 Subjective comparison of dehazing methods

(a) Hazy Image (b) Proposed method using genetic learning (c) CAP method

(a) (b) (c)

Figure 4.24 Subjective comparison of dehazing methods

(a) Hazy Image (b) Proposed method using genetic learning (c) CAP method

Figure 4.22, Figure 4.23, Figure 4.24 and Figure 4.25 show the results of the dehazing using the proposed and CAP (Color Attenuation Prior) method. The quantitative comparison is performed for the two methods in the next section.

4.4 Quantitative Comparison with CAP method

The performance evaluation metrics of any method can be divided into types: blind and non-blind evaluation metrics. Non-blind evaluation metrics require the ground truth images to check the validity of the method, whereas, blind evaluation method does not require ground truth original image for checking validity. The haze descriptors discussed previously developed for the evaluation are used to quantitatively compare the two methods in the hazy images shown in the Figure 4.26.

Figure 4.25 Hazy images dataset for assessment using e and r

Table 4.4 Comparison of two methods in terms of e and r descriptor

Images	CAP method		Proposed genetic learning method	
	e	r	e	r
1	0.4540	**1.2288**	1.2817	1.0044
2	0.25685	1.2437	0.55611	1.7765
3	0.32111	1.2161	1.1249	2.1417
4	**0.0861**	**0.8174**	0.0679	0.7938
5	0.3962	1.3472	1.4611	1.9863
6	**0.4205**	**1.1036**	0.4091	1.016
7	**1.5952**	1.8036	0.8952	1.8873
8	0.3161	1.3354	0.80968	2.0534
9	**0.0498**	**1.1102**	0.0356	0.7726
10	**0.6124**	**1.2896**	0.2594	1.1748

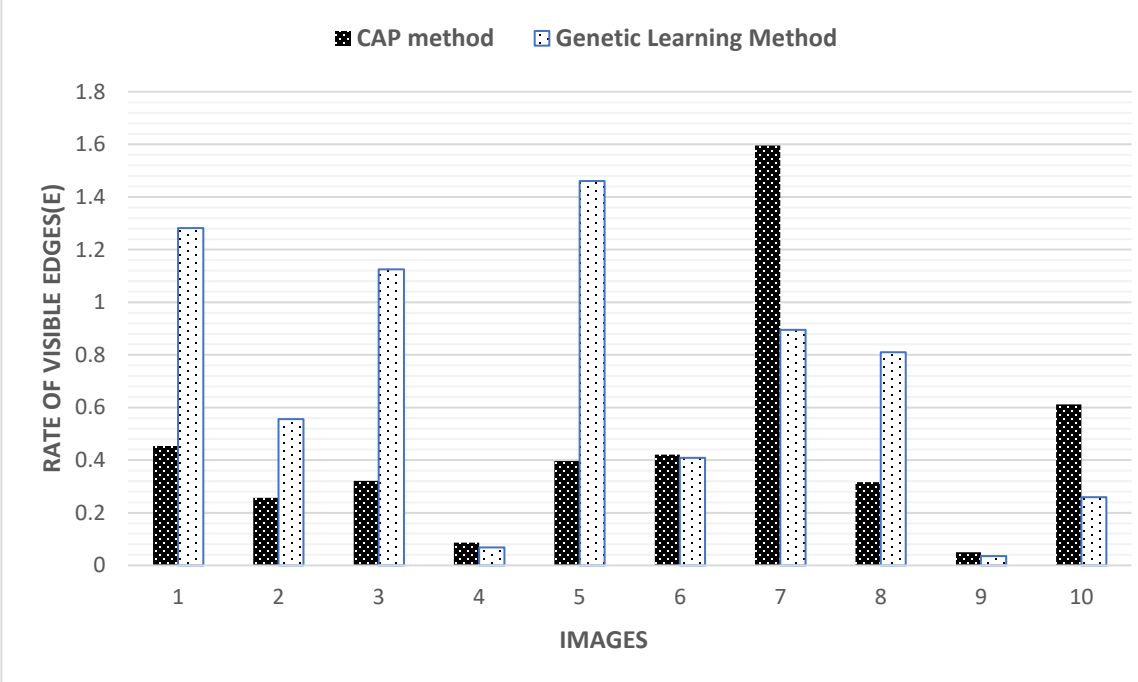


Figure 4.26 Dehazing assessment using e

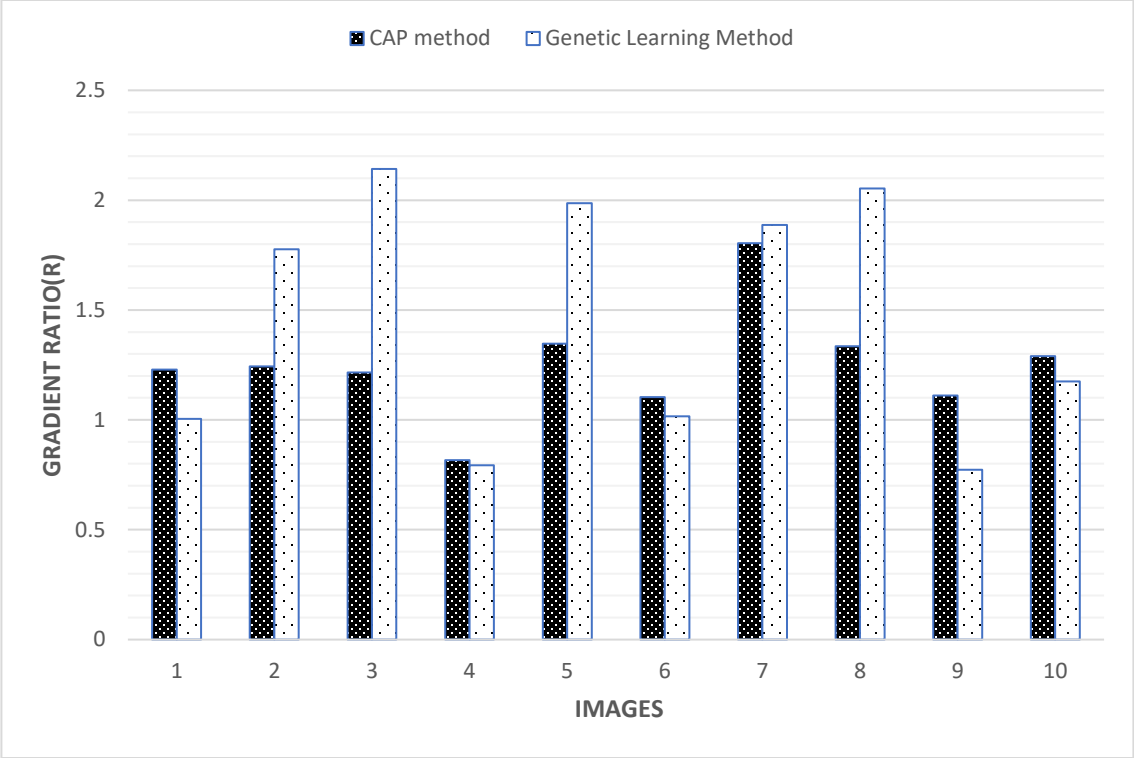


Figure 4.27 Dehazing assessment using r

The comparison results of Table 4.4 show that the proposed method outperforms the CAP method in terms of descriptor e for the 5 out of the 10 images, whereas in terms of descriptor, it outperforms the CAP method for 5 out of the 10 images tested. The assessment using e and r is blind assessment method since the ground truth image is not required for the evaluation of dehazing performance of the dehazing algorithm.

4.5 Validation

The performance of dehazing algorithms is evaluated on dataset: LIVE Color+3D database [12] including 12 scenes with real depth acquired using an advanced range scanner. The performance is evaluated on the 100 images from the dataset. The performance of the method was evaluated on the dataset images using the metrics MSE and SSIM.

Figure 4.28 Ground truth image and synthetic hazy images examples from dataset

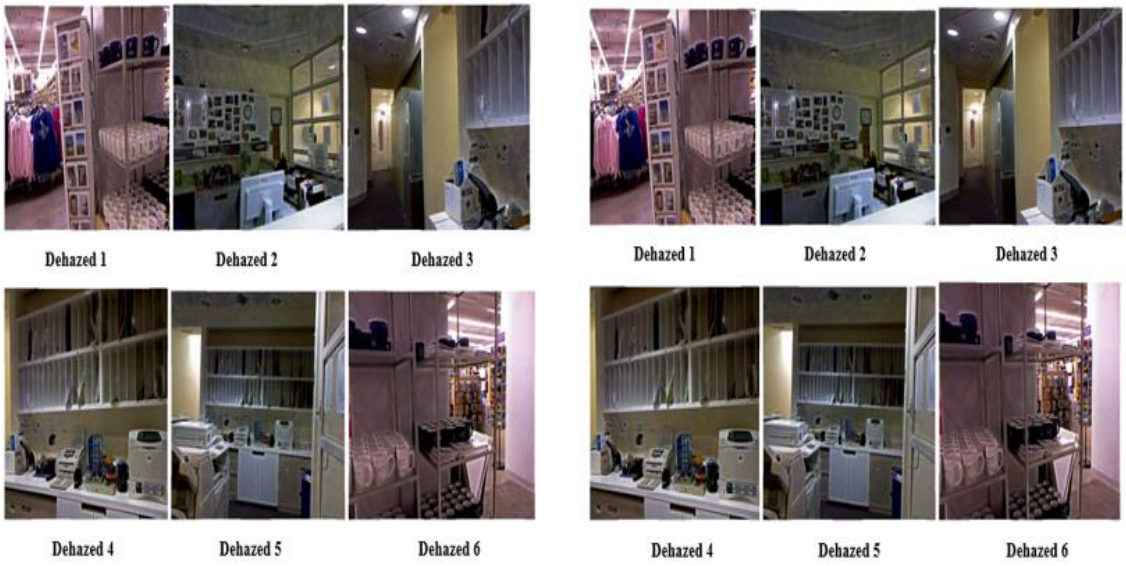


Figure 4.29 Dehazed dataset images using CAP and proposed genetic learning method examples

Table 4.5 MSE of the dehazed images for the two methods

Dehazing methods	MSE
Genetic learning method	**0.024571**
CAP	0.050529

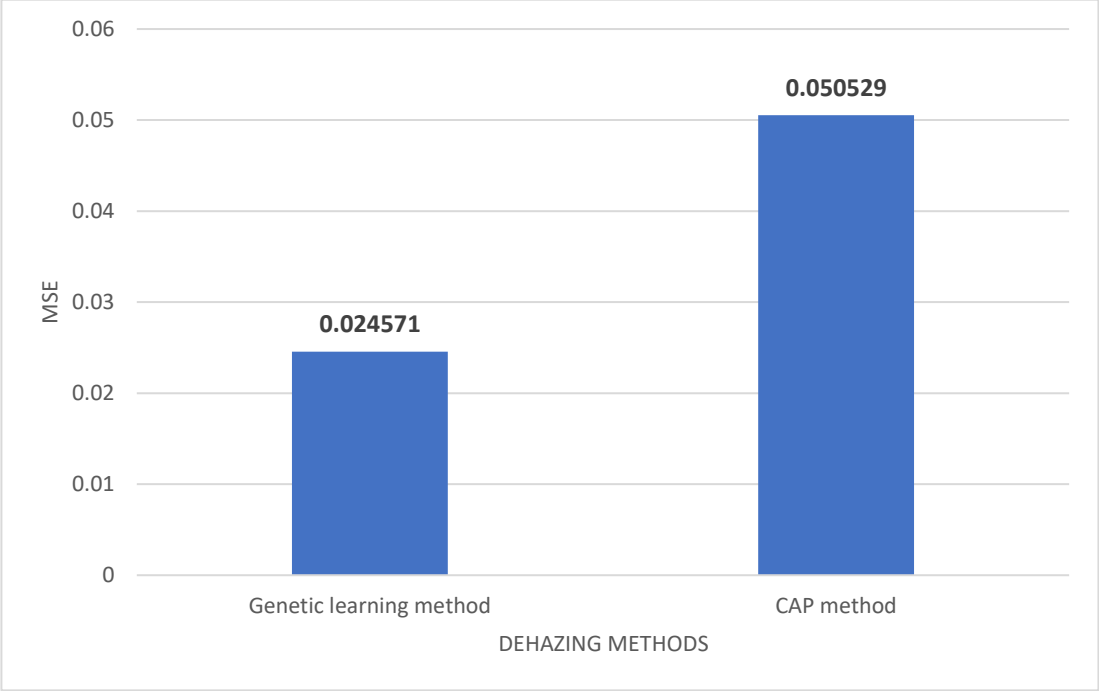


Figure 4.30 MSE results for dehazed image using two methods

It is seen from the Table 4.5 that the MSEs of the results produced by the two methods, the average MSE of proposed learning method for the 100 images from the dataset is 0.024571, which is lower than average MSE of probabilistic learning which is 0.050529. It can be observed from the table that the proposed learning based method is marginally better than probabilistic learning approach in terms of mean square error for the 100 images from the dataset.

Figure 4.31 Ground truth image examples from dataset

Figure 4.32 Synthetic hazy images examples

Figure 4.33 Estimated depth examples using proposed method based on genetic learning

Figure 4.34 Estimated depth examples using CAP method

Figure 4.32 shows the three ground truth images taken from the dataset and the synthetic hazy images are shown in the figure 4.33. The predicted depth map using the proposed method is shown in the Figure 4.34 whereas Figure 4.35 shows the recovered depth using the CAP method.

Table 4.6 MSE of the estimated depth using the two methods

Dehazing methods	MSE
Genetic learning method	**0.067601**
CAP	0.072044

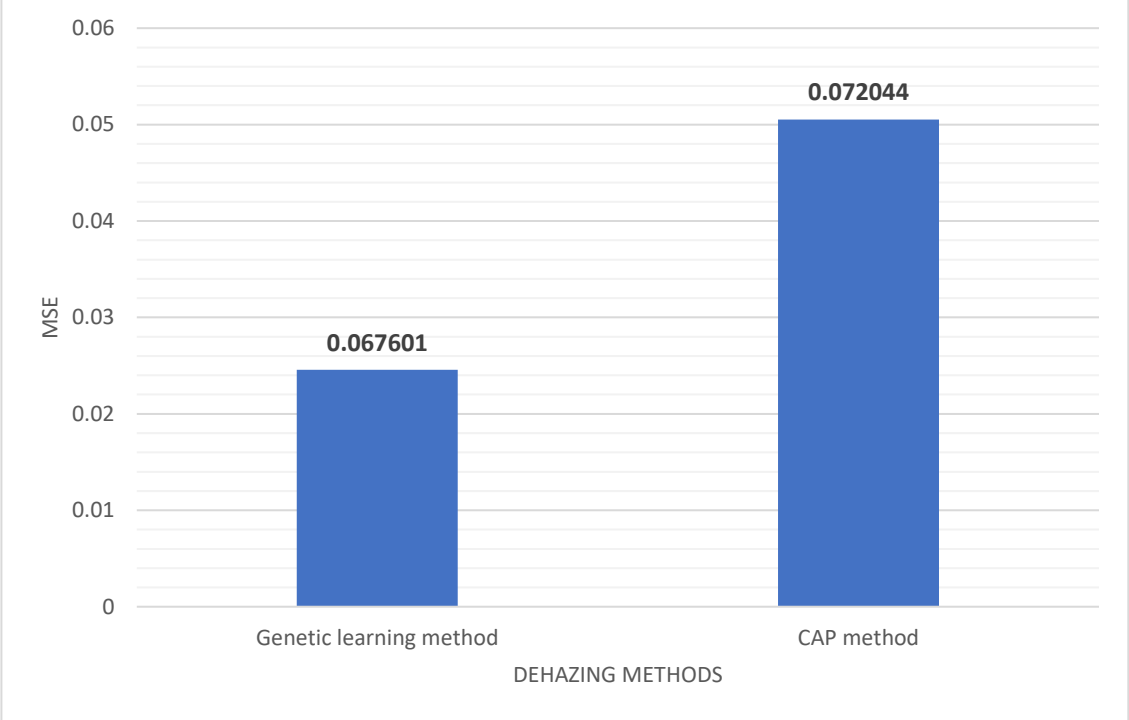


Figure 4.35 MSE of estimated depths using two methods

It can be observed from the Table 4.6 that the MSE of the predicted depth of the results produced by different algorithms, the average SSIM of proposed learning method for the 100 images from the dataset is 0. 0.067601, which is lower than average MSE of CAP which is 0.072044. It can be observed from the table that the proposed learning based method is marginally better than CAP method in terms of predicted depth for the 100 images from the dataset.

Table 4.7 SSIM for the two methods

Dehazing methods	SSIM
Genetic learning method	**0.749860**
CAP	0.547907

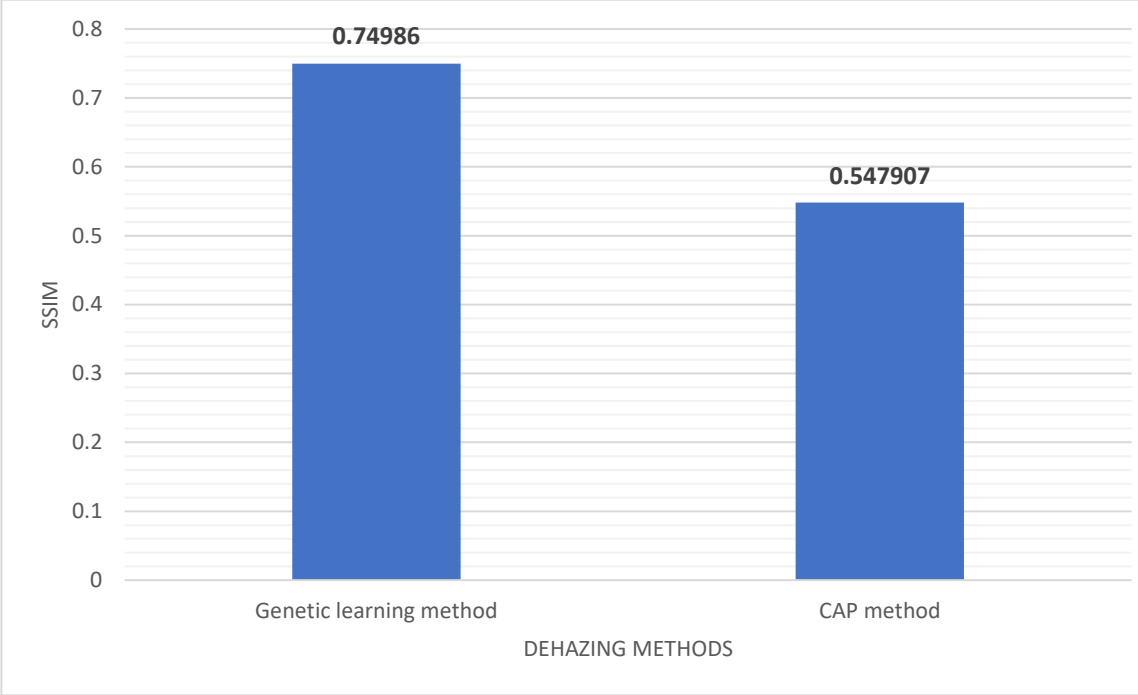


Figure 4.36 SSIM for the dehazed images using two methods

It is seen that the SSIMs of the results produced by different algorithms in the Table 4.7, the average SSIM of proposed learning method for the 100 images from the dataset is 0.749860, which is lower than average MSE of CAP method which is 0.547907. It can be observed from the table that the proposed learning based method is better than probabilistic learning approach in terms of SSIM for the 100 images from the dataset.

CHAPTER 5 CONCLUSION

In the thesis, the single image dehazing based on genetic learning approach was proposed and implemented. In this work, the genetic learning strategy was used to estimate the relationship between depth, saturation and brightness of the hazy image. Once the relationship is established, the model is used to estimate the depth of the given input hazy image, and finally, the original dehazed scene is obtained using the atmospheric model.

The range of β for which dehazing results is optimum was evaluated using 10 test images from the dataset. It was observed that within the range β=1.4 to 1.8, better results were obtained for e. In the range, β =1.2 to 1.6, the results demonstrated better performance in terms of descriptor r, whereas, for SSIM, the dehazing results were better with the range β=0.8 to 1.2.

The method was also compared with the CAP (Color Attenuation Prior) method on the images from the LIVE dataset in terms of MSE and SSIM. The methods were also compared in terms of e and r on other outdoor hazy images. It was found from the results that the performance of the proposed method is superior to the CAP method in terms of the mentioned evaluation measures.

On the 100 images from the dataset, the average MSE of the dehazed images using genetic learning method was 0.024571 and the average MSE that obtained using the CAP method was 0.050529. The depth estimation ability of the two methods was also compared by comparing the depth acquired using the two methods with the ground truth depth from the dataset and calculating the MSE. The average MSE of the depths obtained using genetic learning method was 0.067601 and that using CAP method was 0.072044. For the same set of images, the average SSIM of the dehazed images using genetic learning method was found to be 0.74986 and that using the CAP method was 0.547907. The results show that the proposed method using genetic learning strategy outperforms CAP method in terms of MSE and SSIM.

CHAPTER 6 LIMITATION AND FUTURE WORK

While dehazing, optimum value of scattering coefficient β is taken. In other words, β is assumed to constant and the assumption that β is constant is valid for the case of homogenous atmosphere. The scattering coefficient β is not constant for inhomogeneous weather condition. All the current image dehazing algorithms are based on constant β assumption, so a flexible method is required that is based on the advanced atmospheric scattering models. The future research work can be aimed towards this objective with an aim to make the coefficient β dynamic and extract β directly from the hazy image which is a very challenging task.

In this work, only two features, brightness and saturation of hazy images are taken to establish their relationship to the depth using genetically learned model. However, other features like dark channel of the hazy image can added along with these features to possible increase the accuracy of the estimated depth, and hence dehazing result could be improved as a result.

The current work is only applicable to image dehazing problems. The work can be easily extended to the video dehazing problems.

REFERENCES

[1] Qingsong Zhu, Member, IEEE, Jiaming Mai, and Ling Shao, Senior Member, IEEE, "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior," *IEEE Trans. On Image Processing"*, vol. 24, no. 11, Nov. 2015.

[2] T. K. Kim, J. K. Paik, and B. S. Kang, "Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering," *IEEE Trans. Consum. Electron.*, vol. 44, no. 1, pp. 82–87, Feb. 1998.

[3] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2001, pp. I-325–I-332.

[4] S. G. Narasimhan and S. K. Nayar, "Chromatic framework for vision in bad weather," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2000, pp. 598–605.

[5] S. G. Narasimhan and S. K. Nayar, "Interactive (de) weathering of an image using physical models," in *Proc. IEEE Workshop Color Photometric Methods Comput. Vis.*, vol. 6. France, 2003, p. 1.

[6] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[7] R. A. Davis, A. J. Charlton, S. Oehlschlager, and J. C. Wilson, "Novel feature selection method for genetic programming using metabolomic 1 H NMR data,", *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 50–59, 2006.

[8] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatial-temporal representations for action recognition: A genetic programming approach," *IEEE Trans. on Cybern.*, vol. 46, no. 1, pp. 158–170, 2016.

[9] K. B. Gibson, S. J. Belongie, and T. Q. Nguyen, "Example based depth from fog," in *Proc. IEEE ICIP*, Sep. 2013, pp. 728–732.

[10] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar. 2002.

[11] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Proc. IEEE CVPR*, 2014, pp. 2995–3002.

[12] C. C. Su, L. K. Cormack, and A. C. Bovik, "Color and depth priors in natural images*,"* *IEEE Trans. Image Process*., vol. 22, no. 6, pp. 2259–2274, June 2013.

[13] S. Gould, R. Fulton, and D. Koller, “Decomposing a scene into geometric and semantically consistent regions” in *Proc. IEEE ICCV*, 2009, pp.1-8.

[14] R. Poli, W.B. Langdon, and N.F. McPhee, *A field guide to genetic programming*, 2008.

[15] R. T. Tan, “Visibility in bad weather from a single image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[16] D.P.Searson, “GPTIPS 2: an open-source software platform for symbolic data mining” in *Handbook of Genetic Programming Applications*. Springer, 2015, pp. 551-573.

[17] Hautiere, Nicolas & Tarel, Jean-Philippe & DIDIER, AUBERT & Dumont, Eric. (2008) “Blind Contrast Enhancement Assessment by Gradient Ratioing at Visible Edges” *Image Analysis and Stereology*. 27.10.5566/ias. v27.p87-95.

[18] R. Kohler. “A segmentation system based on thresholding *“Computer Graphics and Image Processing*, vol. 15, no. 4, pp. 319–338, 1981.

[19] William E. Vargas “Two-flux radiative transfer model under non-isotropic propagating diffuse radiation” *Appl. Opt.38,* 1077-1085 (1999)

[20] William E. Vargas, "Generalized four-flux radiative transfer model," *Appl. Opt. 37,* 2615-2623 (1998)

[21] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans.Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

APPENDICES

APPENDIX A

Gradient Rationg at Visible Edges

Visibility Model

For non-periodic targets, visibility can be related to the (Weber) luminous contrast C, which is defined as:

$$C = \frac{\Delta L}{Lb} = \frac{Lt - Lb}{Lb} \tag{A.1}$$

where ΔL is the difference in luminance between target and background, L_t is the luminance of the target, L_b is the luminance of the background.

For suprathreshold contrasts, the visibility level (VL) of a target can be quantified by the coefficient:

$$VL = \frac{Cactual}{Cthreshold} \tag{A.2}$$

At threshold, the visibility level equals one and above threshold it is greater than one. Combining Eq. A.1 and Eq. A.2**,** we have:

$$VL = \frac{\frac{\Delta L}{L_{b\,actual}}}{\frac{\Delta L}{L_{b\,threshold}}} \tag{A.3}$$

As the background luminance L_b is the same for both conditions, then this equation reduces to:

In any given situation, it might be possible to measure the luminance of the target and its background, which gives ΔL_{actual}. But to estimate VL, we also need to know the value of $\Delta L_{threshold}$

Visible Edges Rationg

For each pixel belonging to a visible edge in the restored image, the following coefficient r:

$$r = f^{-1}(\Delta I_r)/f^{-1}(\Delta I_o) \tag{A.4}$$

where ΔI_r denotes the gradient in the restored image, ΔI_o the gradient in the original image and f the camera response function. Then, if the camera response function is assumed to be linear, which is generally the case for CCD sensors, Eq. A.4 becomes simply:

$$r = (\Delta I_r)/(\Delta I_o) = (\Delta L_r)/(\Delta L_o) \tag{A.5}$$

Here, r is mathematically defined because only the gradients of visible edges in the restored image are considered. Hence, only pixels having a minimum contrast can be restored, which ensures that ΔIo is different from zero. Thereafter, assuming that an object in the image is composed of edges, Eq. A.5 can thus be rewritten as:

$$VL = \frac{\frac{\Delta L_r}{\Delta L_{\ threshold}}}{\frac{\Delta L_o}{\Delta L_{\ threshold}}} \tag{A.6}$$

where $\Delta L_{threshold}$ would be given by Adrian's model. Finally, Eq. A.6 becomes:

$$VL = \frac{VL_r}{VL_o} \tag{A.7}$$

where VLr denotes the visibility level of the considered object in the restored image and VLo the visibility level of the considered object in the original image. Consequently, the computation of r enables to compute the gain of visibility level produced by a contrast restoration method. The remaining difficulty is in detecting the visible edges in the images, and it depends on the type of images under consideration. In the following sections, this methodology is applied to images altered by daytime fog acquired using invehicle cameras. On the basis of the equation (1) discussed in the previous section, Duntley developed a contrast attenuation law (Middleton, 1952), stating that a nearby object exhibiting contrast C_0 with the background will be perceived at distance d with the following contrast:

$$C = [(L - L_\infty)/L_\infty]e^{-\beta d} = C_O e^{-\beta d} \tag{A.8}$$

This expression serves to base the definition of a standard magnitude called "meteorological visibility distance" V_{met}, i.e., the greatest distance at which a black object ($C_0 = -1$) with a suitable size can be seen in the sky on the horizon. With the threshold contrast set to 5% (CIE, 1987), this definition yields the following expression:

$$V_{met} = -\frac{log 0.05}{\beta} \cong \frac{3}{\beta} \tag{A.9}$$

Blind Assessment

The local contrast estimator initially presented can be used to access the quality of the contrast restoration method.

Visible edges Segmentation

In order to be consistent with the definition of the meteorological visibility distance proposed by (CIE, 1987), it is enough to consider the set of edges which have a local contrast above 5% so as to obtain the visible edges under daytime foggy weather.

The LIP model (Jourlin and Pinoli, 2001) has introduced a definition of contrast well suited to digital images. In this definition, the contrast between two pixels x and y of an image f is given by:

$$C_{(x,y)} = max\left|f(x), f(y)\right| \triangle min\left|f(x), f(y)\right| \tag{A.10}$$

where $\triangle$ denotes LIP subtraction. Naturally, this definition of contrast is consistent with the definition of contrast used in visual perception (Eq. A.1).

Then, the contrast associated to a border F which separates two adjacent regions follows:

$$C_F(f) = \frac{1}{cardV} \text{⍟⍟}_{(x,y)\in V}\, C_{(x,y)}(f) \tag{A.11}$$

where ⍟ and ⍟ denote LIP multiplication and addition.

Implementation: To implement this definition of contrast between two adjacent regions, Kohler's segmentation method has been used (Kohler, 1981). Let f be a gray level image. A couple of pixels (x,y) is said to be separated by the threshold s if two conditions are met. First, y ∈ V4(x). Secondly, the condition (Eq. 20) is respected:

$$\min[f(x), f(y)] \leq s < \max[f(x), f(y)] \tag{A.12}$$

Let F(s) be the set of all couples (x,y)separated by s. With these definitions, for every value of s belonging to [0,255], F(s) is built. For every couple belonging to F(s), the contrast $C_{x,y}(s)$ is computed:

$$C_{x,y}(s) = min[\frac{\left|s - f(x)\right|}{\max\left(s, f(x)\right)}, \frac{\left|s - f(y)\right|}{\max(s, f(y))}] \tag{A.13}$$

The mean contrast (Eq. A.13) associated to F(s) is then computed:

$$C(s) = \frac{1}{cardF(s)} \sum_{(x,y)\in F(s)} C_{x,y}(s) \tag{A.14}$$

The best threshold s_0 verifies the following condition:

$$s_o = \underset{s\in[0,255]}{\operatorname{argmax}}\, C(s) \tag{A.15}$$

It is the threshold which delivers the best mean contrast along the associated border $F(s_0)$. Instead of using this method to binarize images, we use it to measure the contrast locally. The evaluated contrast equals $2C(s_0)$ along the associated border $F(s_0)$. Finally, if $2C(s_0) > 5\%$, $F(s_0)$ is considered to be a visible edge.

Descriptors

n_o and n_r denote respectively the cardinal numbers of the set of visible edges in the original image Io and in the contrast-restored image I_r . The latter set is denoted $\wp_r$. First of all, to compute e, the rate of new visible edges in I_r :

$$e = \frac{nr - no}{no} \tag{A.16}$$

The value of e evaluates the ability of the method to restore edges which were not visible in Io but are in I_r .

In complement, to compute r, the geometric mean of the ratios of VL defined by Eq. A.8. The value of r expresses the quality of the contrast restoration by the proposed method. This descriptor takes into account both invisible and visible edges in I_o:

$$\bar{r} = \exp[\frac{1}{n_r} \sum_{P_i\in P_r} logr_i] \tag{A.17}$$

APPENDIX B

Color Guided Filtering

In general, the locally adapted filtering process can be described as a weighted sum:

$$qi = \sum_{i} \text{Wij(I)pj} \tag{B.1}$$

where i and j are pixel indexes, q is the result, W is a weight dependent on the guidance image, I, and p is the target (or input) image. For example, the joint bilateral filter could be defined in this way with the filtering kernel W^{jbf} as

$$W_{ij}^{jbf}(I) = \frac{1}{K_i}\exp\left(-\frac{\left|x_i - x_j\right|^2}{\sigma_s^2}\right)\exp\left(-\frac{\left|I_i - I_j\right|^2}{\sigma_r^2}\right) \tag{B.2}$$

where x_i and x_j are pixel locations, I_i and I_j are pixel intensities, σ_s and σ_r adjust the spatial similarity and range similarity respectively, and K_i is a scaling coefficient so that $\sum_j \text{Wjb fij (I)} = 1$. When I and p are identical, this becomes the original bilateral filter. For a grayscale image, the derivation of the Guided Filter relies on a local linear assumption, which is expressed mathematically as:

$$q_i = a_k I_i + b_k, \forall i \in W_k \tag{B.3}$$

where q_i is the filter output, I_i is the guidance image, and a_k and b_k are linear coefficients assumed constant in window w_k . Window sizes are typically defined by their radius, r, which is the pixel distance from the center pixel to an outer pixel. Since square windows are used, the total window size is therefore (2r + 1) × (2r + 1). The linear coefficients are then determined by minimizing the following cost function:

$$E(a_k, b_k) = \sum_{i \in W_k} \left(\left((a_k I) + b_k - p_i\right)\right)^2 + \in a_k^2) \tag{B.4}$$

where ε is a regularization parameter to prevent a_k from being too large. The solution to Eq. (B.4) is found to be:

$$a_k = \frac{\frac{1}{\left|w\right|}\sum_{i \in w_k} I_i p_i - \mu_k \overline{p_k}}{\sigma_k^2 + \in} \tag{B.5}$$

$$b_k = \widetilde{p_k} - a_k \mu_k \tag{B.6}$$

where μ_k and $\sigma^2{}_k$ are the mean and variance of I in window w_k. $|w|$ is the number of pixels in w_k , and p_k is the mean of p in w_k . Since a pixel i belongs to many windows, the final filter output qi is averaged over all possible windows. So after computing all filter coefficients in the image, the filter output is:

$$q_i = \frac{1}{|w|} \sum_{k:i \in w_k} (a_k I_i + b_k) \tag{B.7}$$

$$= \overline{a_i} I_i + \overline{b_i} \tag{B.8}$$

The Guided Filter can be expressed as in Eq. (B.3), with the kernel weights expressed as:

$$W_{ij}(I) = \frac{1}{|W|^2} \sum_{k:(i,j) \in w_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \in}) \tag{B.9}$$

Thus, the Guided Filter simply measures the normalized correlation between two pixels. Spatial distance is taken into account by the fact that when pixels i and j are close together, they share more windows compared to when they are far apart.

The Guided Filter is also extended to color guidance images by rewriting Eq. (3) as:

$$q_i = a_k^T I_i + b_k, \forall\, i \in w_k \tag{B.10}$$

where I_i is a 3x1 color vector, a_k is a 3x1 coefficient vector, and q_i and b_k are scalars. The Guided Filter then becomes:

$$a_k = (\sum_k + \in U)^{-1} (\frac{1}{|w|} \sum_{i \in w_k} I_i p_i - \mu_k \tilde{p}_k) \tag{B.11}$$

$$b_k = \tilde{p}_k - a_k^T \mu_k \tag{B.12}$$

$$q_i = \tilde{a}_i^T + \tilde{b}_i \tag{B.13}$$

where Σ_k is a 3x3 covariance matrix of the colors in I, U is a 3x3 identity matrix, and μ_k is a 3x1 mean vector of the colors in I.