Image dehazing from the perspective of environmental illumination

Sanchayan Santra

Electronics and Communication Sciences Unit

Indian Statistical Institute

A thesis submitted to Indian Statistical Institute for partial fulfillment of the requirements of the degree of

Doctor of Philosophy

July 2019

To Maa and Baba.

Acknowledgments

This thesis would not have taken the current form without the help and support of all the people who have contributed in their own in my journey of PhD. Although it is not possible to mention the names of all the people, still I would like to use this space to mention the names of a few. First I would like express my sincere gratitude to my guide Prof. Bhabatosh Chanda for being patient with me and giving me the freedom of exploration. I may not even realize in how many things he may have taught me in so many ways. The technical aspects constitute only a small part of it. It is a grace to have him as my advisor. Apart from that interactions with other professors in our department e.g. Dipti Sir, Aditya Sir, Pinak Sir and Nikhil Sir has always been enriching. My special thanks go to Ranjan, without whom I may not have started deep learning and my last three works would not have been a reality. I would like to thank Pulak Da for directing me in the right direction which resulted in my first published work. It was really necessary at that time. I would like to thank Anabik for making the travels enjoyable and hassle-free with his own way handling different situations. I would like to thank Shounak for always nudging me to work. Things would have been tough without the assurances given by Umer Da, Chintan Da. Soumitra Da has always helped whenever asked for. The experience of attending my first conference would have been boring without the company of Mrinmoy Da, Swapna Di and Arpan. Sankha has always been a good critique. My experience at our lab and the department could have never been so enriching and entertaining without my friends, colleagues and seniors - Jija Di, Soumitra Da, Mrinmoy Da, Umer Da, Ranjan, Shounak, Sankha, Avisek, Subhasis Da, Angshuman, Bikash Da, Swapna Di, Debapriya, Moumita, Koustuv Da, Subhrajyoti, Suchismita, Samriddha, Kingshuk and Yusuf Da. Not to forget about the summer interns - Mayank, Arnab, Aneek, Shrisendu, Pranoy, Nishant, Shubham, Satish and Pratiti. Special thanks should be given to Manjari, Aparajita and Avisek for making my stay at hostel memorable: celebrating almost anything, making me work, trips in different parts of Kolkata and the list goes on. Those will remain as a part of my cherished memories. I should mention about the storyteller, Abhisek, whose stories had made the meals, especially the dinners enjoyable. Among the people outside ISI, Vivek has always supported me and inspired me to keep moving on even when things did not look right. I should also mention the name of Nabarun and Koushik for helping me during their brief stay in Kolkata.

I should not forget to mention about the non-teaching staffs of our department: Dilip Da, Partha Da, Badal Da, Dipesh Da, and Shekhar Da, for working behind the scenes and making our stay at the department a pleasant experience. I also like to thank ISI for providing support in attending different conferences.

Abstract

Haze and fog are atmospheric phenomena where the particles suspended in the air obscure visibility by scattering the light propagating through the atmosphere. As a result only a part of the reflected light reaches the observer. So, the apparent intensity of the objects get reduced. Apart from that, the in-scatter of the atmospheric light creates a translucent veil, which is a common sight during haze. Image dehazing methods try to recover a haze-free version of a given image by removing the effects of haze. Although attempts have been made to accurately estimate the scene transmittance, the estimation of environmental illumination has largely been ignored. Only a few methods have been proposed for its estimation and the only the recently proposed methods have considered to estimate this when proposing an end-to-end method. So, that methods that we propose here mainly motivated by the how we can estimate the environmental illumination under different settings.

We start with relaxing the haze imaging model to account for the situations when the sky is not cloudy. Normally during fog and haze the sky remains cloudy. As a result the entire scene receives the same amount of light. But the sky may not always remain cloudy when a scene is being photographed in haze or fog condition. If we only consider daytime scenes, the direct sunlight plays a role in the illumination when the sky is clear. But, when this happens, the scene receives different amount of light in different portion of the image. The imaging model is relaxed to capture this situation. The method that is proposed here is based on the color line based dehazing, extended to work under this relaxed model. Since, the proposed relaxation is done with the assumption of daytime scenes, this model is not applicable for night-time scenes. So, in the next chapter, the imaging model is further relaxed to include the night-time haze situations. This is done by allowing the environmental illumination to vary spatially within the image. But this introduces a challenge. Given a hazy image the color and even the number of different illuminants present in the scene is not known. Moreover it can vary across the scene, especially in the night-time images. We have shown the construct of color line based dehazing to estimate both the possible illuminants present in the scene and the patches they affect, by the simple technique to Hough Transform. This has enable us to propose a method that works for both day and night-time images.

Although, these color line based methods works well in the default value of the parameters, its performance degrades if the parameter values are not well suited for the given image. But tuning the parameters, which are around 10 in number, is not straight forward. So Convolutional Neural Networks (CNN) are utilized in the subsequent chapters to automatically learn the haze-relevant features. In the initial attempt (Chapter 4), we work with the original imaging model (constant environmental illumination for the whole scene) and by taking small patches from the input image. The transmittance and environmental illumination is estimated from patches using a CNN based model. This CNN predicts transmittance and environmental illumination given a hazy patch as input. But it is seen, trying to estimate the environmental illumination from small patches is error prone. So, in the next chapter we work by taking bigger patches from images. We have utilized a Fully Convolutional Network to handle the big patches. This network is trained using our proposed loss, called the Bi-directional Consistency Loss. This loss requires only pair of hazy and haze-free images and favors only those transmittance and airlight by which the haze-free image can be obtained from the hazy image and vice-versa.

Instead of just directly regressing the parameter values using a CNN, in the last chapter a method has been proposed to estimate the transmittance by comparing various dehazed version of a hazy patch with the original hazy one. This is motivated by the fact that comparing which has patch has more haze is easier than estimating the level of haze from a single patch. To automate the process of comparison, we have designed a CNN based module, called the *patch quality comparator*. By finding the transmittance in this way, we only obtain its value in such a way so that it does not produce bad looking outputs. It is also seen that the quality of estimate of environmental illumination greatly affects the transmittance computation. A correct estimate of environmental illumination produces very good outputs.

Contents

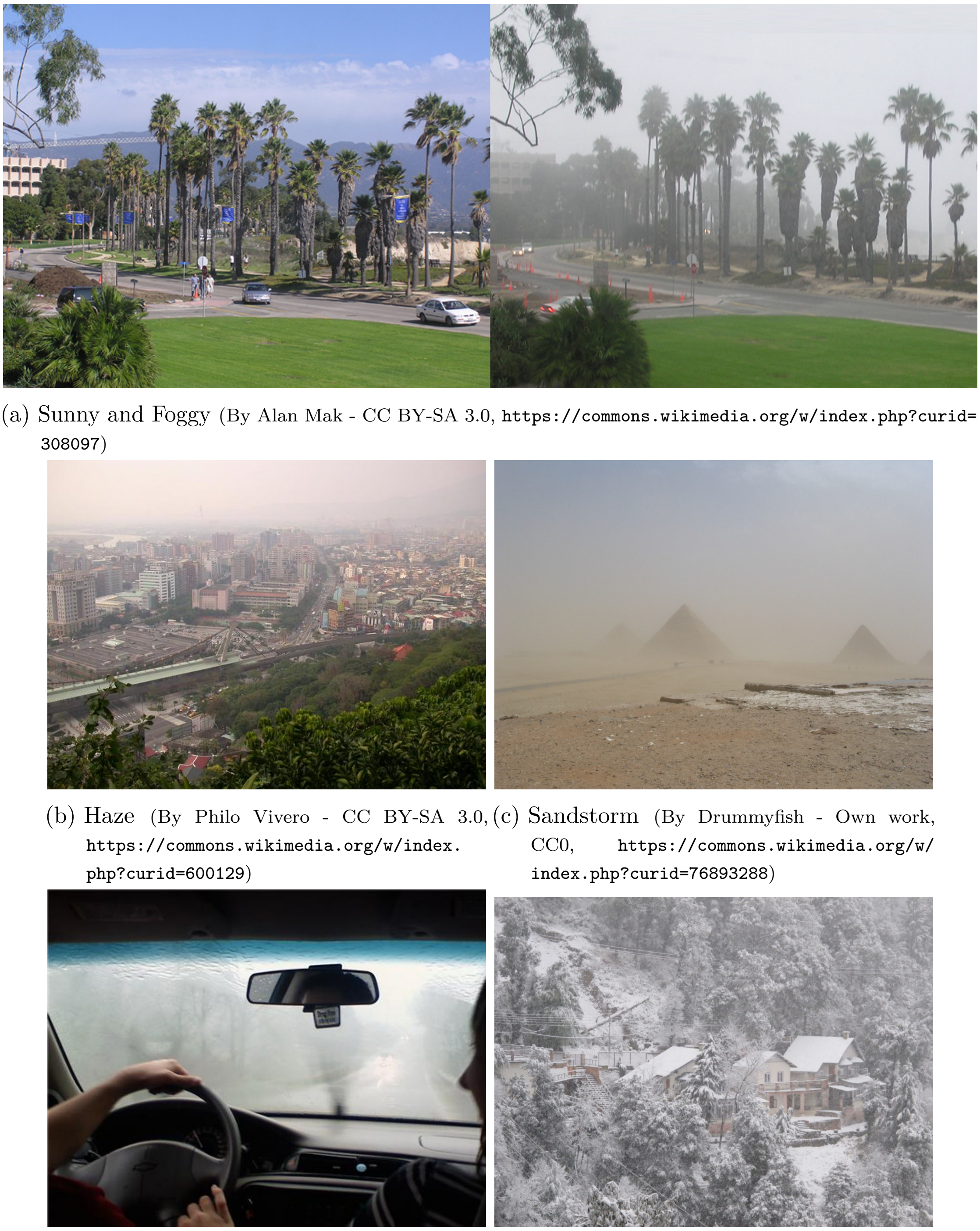
List of Figures

List of Tables

Chapter 1

Introduction

Our five senses provide us the interface to the physical world. We continuously collect data through these 'sensors' and try to respond accordingly. Among these five senses, our visual sensors, i.e., the eyes provide the most feature rich data. For this reason, visual data plays an important roles in our lives. This visual data is generated from the light that our eyes receive. This light can reach our eyes either directly from some light sources or after getting reflected or refracted by some object. Light may be attenuated or interfered by medium before reaching our eyes or some artificial visual sensors like cameras. In such cases, the visibility gets impaired. Therefore, the existence of obstacles in the path between our eyes and the object, through which the light propagates before reaching our eyes, diminishes or obstructs the visibility of that object. Now depending on the type of the obstruction, the visibility can vary in a broad spectrum. If the light is completely blocked the visibility becomes zero. This is termed as occlusion. If the light is attenuated partially, the visibility gets reduced and we receive only a partial information about the scene. The partial attenuation can happen in various ways. For example, if we look at an object through a colored glass, we won't be able to see the object in its true color because the glass is going to absorb some colors from the reflected light that is passing through it. Similar thing happens in bad weather conditions like fog, haze, sandstorm, rain and snow (Figure 1.1). The particles present in the atmosphere, i.e. haze and fog particles, raindrops, snowflakes obstruct the light and only a part of it reaches our eyes. As a result, it becomes difficult to distinguish the objects. Reduced visibility greatly increases the risk of accidents in all kind of transportation system. Navigation becomes hard in these situations. Therefore, from the point of visibility, these situations are, no doubt, undesirable. From the point of view of outdoor computer vision systems, the reduction in contrast and degradation in color greatly impacts the performance of the systems. This is because most of them have been proposed with assumption of "clear" scenes in mind. However these weather conditions are natural phenomena, and we have little control over them. So, if we are able to somehow design



(a) Sunny and Foggy (By Alan Mak - CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=308097`)

(b) Haze (By Philo Vivero - CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=600129`)

(c) Sandstorm (By Drummyfish - Own work, CC0, `https://commons.wikimedia.org/w/index.php?curid=76893288`)

(d) Rain (By Malinaccier - Own work, CC BY 3.0, `https://commons.wikimedia.org/w/index.php?curid=6675965`)

(e) Snow (By Amareshwara Sainadh - Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=25890658`)

Figure 1.1: Adverse weather conditions that reduces visibility

a method that can virtually “clean” these obstruction so as to increase the visibility, then it would be of immense help. However simple image processing techniques falls short in overcoming these situations. Therefore to be able to remove these obstructions, an understanding of how light undergoes changes in these situations becomes necessary. But the change depends on the weather condition. So, each weather condition is usually treated separately. The methods proposed in this thesis, *i.e.* image dehazing methods, focus on improving the visibility of images taken under fog and haze condition. In the following section we provide the theoretical basis of how images are formed during haze.

1.1 Image formation under haze

Haze and fog are atmospheric phenomena where the particles suspended in the air obscure visibility by scattering the light propagating through the atmosphere. Because of the scattering the scene radiance gets attenuated. The relationship between scattering and attenuation of a light beam is modeled by the following equation [40].

$$E_x = E_0 e^{-x\beta_\lambda}, \tag{1.1}$$

where, E_0 is the irradiance of the light at position $x = 0$, that is without the effect of scattering, while E_x denotes the irradiance of light after traveling a distance x in the scattering medium. β_λ is called the scattering coefficient that quantifies the amount of scattered flux per unit length of path. The amount of scattering in general depends on the wavelength of light. The subscript λ is used to denote this dependence. On the other hand, attenuation is not the only phenomena that occurs during haze. A simple observation reveals, that objects become lighter as their distance from the horizon decreases (Figure 1.1a). It seems as if the atmosphere acquires a certain luminance. This phenomenon is known as the **airlight** and it occurs due to the scattering of light from the sun, the sky and the ground, towards the observer by the particles present in the atmosphere between the observer and the object. In this situation, the apparent luminance of a black object at a distance x is given by the following [32, 42, 40].

$$B_x = L_h(1 - e^{-x\beta_\lambda}), \tag{1.2}$$

where, B_x is the apparent luminance of a black object at distance x. L_h is the luminance of the horizon sky and β_λ is the scattering coefficient of the medium. But objects are not always black. If we consider an object with an intrinsic luminance L_0, then their apparent luminance becomes the following, when they are observed from a distance x.

$$L_x = L_0 e^{-x\beta_\lambda} + L_h(1 - e^{-x\beta_\lambda}), \tag{1.3}$$

where, L_h is the luminance of the horizon sky and β_λ is the scattering coefficient of the medium. The first part of the equation, called the **direct transmission**, and the second part is known as the airlight. Since we are interested in working with images, we will be working with irradiance of objects as measured by some camera. If we assume the response of the camera is linear to the observed luminance, then the image captured during haze can be presented as, following equation 1.3 [45].

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

$$t(\mathbf{x}) = e^{\beta_\lambda d(\mathbf{x})}. \tag{1.5}$$

Here $I(\mathbf{x})$ is the intensity observed by the camera at pixel $\mathbf{x} = (x, y)$. $J(\mathbf{x})$ is the true intensity at the same position without the effect of haze. t models the attenuation due to scattering, and it is called the **scene transmittance**. $d(\mathbf{x})$ denotes the depth of pixel $\mathbf{x}$ from the camera or observer. Equation 1.5 shows that attenuation depends on the distance of the pixel from camera. The radiance at the horizon is denoted by A. As this quantity is directly affected by environmental illumination, this is usually considered to be the global environmental illumination. This stems from the fact that during foggy weather the sky tends to remain cloudy [44]. Now for RGB images we can use equation 1.4 for each channel. But it has been shown by Narasimhan and Nayar [45] that for fog and haze the transmittance does not vary much with wavelength of light within the visible spectrum. Therefore, for RGB images the following form of the imaging equation is generally used.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})). \tag{1.6}$$

Here $\mathbf{I}(\mathbf{x})$, $\mathbf{J}(\mathbf{x})$ and $\mathbf{A}$ are 3×1 vector and $t(\mathbf{x})$ is a scalar.

1.2 Motivation and Objective

For the purpose of image dehazing we are interested in recovering the $\mathbf{J}$, that is the scene radiance without the effect of haze, while having access to only the hazy image ($\mathbf{I}$) [see equation 1.6]. This makes the problem an ill-posed one, because here only $\mathbf{I}(\mathbf{x})$'s are known and all the other variables, including t and $\mathbf{A}$, are unknown. Another challenging aspect of the problem is its dependence on depth (equation 1.5). With increase in distance from the camera, the degradation also increases. Although the image dehazing methods aims to completely eradicate haze, it may not always be possible. The methods can only recover the information that is available in the image.

Although it is common to assume that the environmental illumination is constant over a scene, this assumption does not always hold true. In most of the foggy situation the sky remains cloudy and the light from the sun gets diffused by the clouds. As a result, the whole scene receives more or less the same amount of illumination. Only in this situation the constant environmental illumination holds. But during haze and fog the sky may not always remain cloudy. Then direct sunlight plays a role in illuminating part of the scene. In this situation the intensity of illumination may vary within the scene. Here the assumption of constant environmental illumination does not hold. Another thing that is implicitly assumed in the imaging model is that a single illuminant is illuminating the scene. But this does not hold true for hazy images taken during the night. Because during the night there may be multiple artificial light sources in the scene. These lights can be of different colors also. So, the constant environmental illumination assumption also does not hold in this situation. Therefore, relaxing of the equation [equation 1.6] becomes necessary. In this thesis we have proposed methods to recover the $\mathbf{J}$ under varied environmental illumination. This is more challenging because relaxation of imaging model increases the number of unknowns further.

1.3 Related Works

Till date a variety of image dehazing methods have been proposed. These methods can be categorized in various ways. Division can be made depending on the type of the image *e.g.* daytime or night-time. Another way of division is based on the number of required image *i.e.* multi-image method and single image method. Although there are different varieties of the problem, dehazing of daytime scenes using a single image has received the most attention. But these are only broad categories, not all methods can be categorized in this way. There exits a separate line of research that tries to restore images taken under water. The image formation process is in way similar to haze and fog, but this is a completely different problem and normal image dehazing methods does not work for these images. So, we don't go into the detail of those methods. For a bit more detailed overview of the daytime dehazing methods the reader may refer to the survey by Li et al. [37].

Since the reduction of visibility due to haze a very common problem, there have been attempts to solve the problem without considering image formation model. Oakley and Bu [47] have proposed to restore the contrast loss due to airlight. They have assumed that the airlight is constant throughout the image and have proposed a method to detect and remove its existence. Possibly the first work in image dehazing that has attempted to use the image formation model is by Narasimhan and Nayar. They are among the first people who have studied how haze affects the scene from the perspective of computer vision. In their work [45] they have first described how light undergoes changes under fog and haze by building upon the works of McCartney [40]. They have also shown how the scene structure may be extracted from two images of the same scene captured under different weather condition. They have also given the formulation of dichromatic atmospheric scattering model. Using this model they have shown various information of the scene can be extracted (*e.g.* the color of the haze, relative depth, "clear day" scene radiance) using two or more images. In their later work [44], they have further refined the model for outdoor scenes and homogeneous atmosphere. Using this model they have proposed method to extract scene structure and restore contrast using two images taken under different weather conditions. Since getting images of the same scene under different weather condition is difficult in practice, there have been attempts to use image of the same scene under different polarization states. The work by Schechner et al. [55] is based on the fact that scattered airlight is partially polarization. Although polarization filters cannot remove haze completely, they have shown using the imaging model and two images taken through a polarizer at different orientation. This relaxes the requirement of two images to be taken under different weather condition. The method of Shwartz et al. [57] uses the same idea of the Schechner et al. [55] but relaxes the requirement of sky to be present in the input image. The method uses independent component analysis (ICA) to estimate the haze parameters.

All the methods mentioned till now take the help of multiple images due to the ill-posed nature of imaging model. Another challenging aspect of the problem is its depth dependent degradation, that means with increasing depth, the amount of degradation increases. Estimating depth from a single image is an ill-posed problem. So, some methods try to dehaze an image when depth map of the scene is somehow available. Narashiman and Nayar [43] have proposed a method when the additional information of depth is provided interactively by the user. Using this provided depth and the imaging model the effects of weather is removed from a single image. Hautiére et al. [26] proposed a dehazing method for the specific case of in vehicle on-board camera. They have assumed the scene contains mainly road and the camera properties, height from ground is known. Kopf et al. [31] attempted to dehaze an image by using a exact 3D model of the scene.

Only more recent methods have focused on dehazing with only a single image as input. These methods achieve this by making stronger assumptions about the input and/or the output images. Tan [60] made an observation that haze-free images have more contrast than the hazy ones. So, in his method he tried to obtain a dehazed image by maximizing the local contrast in a Markov Random Field (MRF) based framework. Although, the resulting images attain more visibility, they tend to contain saturated colors and look unnatural. Fattal [21] tried to estimate scene transmittance with the assumption that surface shading and scene transmittance are locally statistically uncorrelated. This method fails in case of fog and dense haze when surface shading and scene transmittance does not vary sufficiently. Ancuti et al. [7] proposed a fast pixel level method using a 'semi-inverse' of the original image. Based on the hue disparity between the hazy image and its semi-inverse haze is detected and removed. He et al. [28] have proposed dark channel prior to estimate scene transmittance. Dark channel prior is based on the observation that in haze-free images, in most of the local regions not covering the sky, pixels often have low intensity in at least one color channel. In case of hazy images the intensity of those color channels is mainly contributed by the airlight. This information is utilized to estimate the transmittance. Nishino et al. [46] have used a Bayesian method to jointly estimate depth and albedo. They model the image using the framework of a factorial MRF assuming depth and albedo to be statistically independent. They enforce natural image and depth statistics as priors when estimating the latent albedo and depth from the image. Tarel et al. [62] has mainly focused to handle the problem of image dehazing from the perspective of driver assistance systems. The method they have proposed is focused towards better handling of road images. The planar road assumption introduces further constrains which results in a fast restoration algorithm. Ancuti and Ancuti [6] has proposed a image fusion based pixel level method. Their method work by fusing a white-balanced and global contrast enhanced input image. The fusion weights are computed in terms of luminance, chromaticity and saliency. Gibson and Nguyen [25] proposed a framework called the color ellipsoid framework. This is based on the observation made by Omer and Werman [48] that in a small patch the colors are usually distributed normally. Therefore they form an ellipsoidal structure and depending on the haze the form and the position of the ellipsoid changes. The estimated ellipsoid serves as the key to estimate the transmittance. They have also shown existing dehazing methods like Fattal [21] and He et al. [28] can be explained from the point of view of the proposed framework. In that sense this framework is a unification of these methods. Meng et al. [41] had proposed to estimate the scene transmittance by exploiting the inherent boundary constrain enforced by the radiance cube. This extends the idea of Dark Channel [28] in transmittance computation with the help of this boundary constrain using simple morphological closing operation. This estimate is regularized using a ℓ^1 norm based contextual regularization to obtain more robust estimate for the whole image. Yan et al. [65] proposed a method to remove the effect of dense scattering layer from images. In regular methods after removing the scattering layer (*e.g.* haze) originally unnoticable artifacts gets largely amplified. Their method proposed to solve this issue by using non-local structure aware regularization. They have also present a way to efficiently solve the proposed optimization. Fattal [22] adopted the idea of color line [48] to image dehazing. Omer and Werman [48] made an observation that colors in a small patch of a natural image ideally lie on a line (color line) passing through the origin in the RGB space. But due to sensors and other distortions they form elliptical color clusters. In hazy condition, this ideal color line gets shifted in the direction of airlight. From this shift the transmittance is estimated. Galdran et al. [24] has proposed a method from the point of view of contrast enhancement. This method is based on a perceptually inspired variational contrast enhancement framework. They have adapted the contrast enhancement method so that it conforms to the haze imaging model. The method of Sulami et al. [59] is solely dedicated to the estimation of environmental illumination. They have estimated the orientation of environmental illumination using the color line model and its magnitude using a global regularity that is observed in hazy images. Tang et al. [61] have tried to solve the problem of image dehazing in a learning framework. They took existing haze-aware features like dark channel, local max contrast, local max saturation, hue disparity to regress the transmittance from image patches. The training data for the regressor was generated by synthetically adding haze to patches of haze-free images. Choi et al. [15] have proposed a no-reference perceptual fog density predictor model (FADE) that works by only using fog aware statistical features. It measures the deviations from statistical regularities from natural foggy and fog-free images to predict fog density. They have utilized this prediction to propose a defogging algorithm. But Ma et al. [39] has made a perceptual study and have shown that FADE and other proposed metrics does not perform well in predicting the quality of dehazed images. [68] have proposed color attenuation prior to model the scene depth. This prior is based on the observation that difference between the brightness and saturation can approximately represent the concentration of haze. So, they have modeled depth as a linear function of brightness and saturation. The parameters of this function is learned in a supervised fashion. With the recovered depth information they dehaze the given image. Bahat and Irani [10] have utilized the patch recurrence property of images to estimate the haze parameters. The patch recurrence property says, a small image patch tend to repeat within a natural image, both within and across scale. In case of fog and haze the recurrence property diminishes because the patch can occur at different depths. This is utilized to recover the airlight color and transmittance of the patches. The method proposed by Berman et al. [11] is based on the observation that the colors of a haze-free image can be approximated by a few hundred colors and they form tight clusters in the RGB space. In case of haze, these cluster form lines (termed as haze-lines). These haze-lines are used to estimate the transmittance at different pixels. Later they have proposed another work [12] based on the haze line to estimate the airlight.

The recent success of Convolutional Neural Networks in the domain of computer vision [34, 18, 38] has encouraged its use in image dehazing. Cai et al. [14] have proposed a CNN based end-to-end learning framework to estimate medium transmittance. Instead of using handcrafted features, a CNN is utilized to learn the haze relevant features and predict the transmittance. Ren et al. [53] have also employed a CNN to the estimate scene transmittance. To be able to properly estimate the transmittance in the whole image they have used multi-scale CNN to capture both coarse and fine scale structures. Li et al. [35] works using a reformulated atmospheric scattering model that unified the transmittance and environmental illumination using a single parameter (named $K(\mathbf{x})$). They have proposed a CNN to estimate this $K(\mathbf{x})$ and generate the clean image directly. They have also shown their dehazing network improves the detection and recognition results when used in conjunction with Faster R-CNN [52].

The image dehazing methods proposed for daytime scenes does not work well for night time images. The common atmospheric model used by the daytime methods does not work for the night time images, mainly because of assumption of the constant environmental illumination. Although the initial attempt by Pei and Lee [50] the imaging model of the daytime methods. But to compensate for the night-time images a color transfer method is utilized. Then they have applied Dark Channel Prior [28] and Guided filter [27] to estimate the scene radiance. To increase the brightness and overall contrast, bilateral filter is applied as the post-processing filter. The method propose by Zhang et al. [67] works using a modified version of the imaging model to account for the changes in night-time images. Since in night-time images artificial lights are the only source of the illumination, overall image intensity can be low and colors can be biased by the color of the lights. Due to these reasons, the authors have compensated for the intensity loss by balancing the illumination and have corrected the possible color bias. Then the method of He et al. [28] is used to obtain the dehazed image while estimating the environmental illumination in local neighborhood. The night-time dehazing method proposed by Li et al. [36] is more focused on removing the glows caused by multiple scattering of the light near the light sources. For that they use a modified version of imaging model that incorporates this multiple scattering term. Then they separately estimate the glows in addition to estimating the transmittance and environmental illumination to obtain the dehazed image. The night-time dehazing method of Ancuti et al. [5] is build on the previously proposed multi-scale daytime dehazing method by the same authors [6]. For the fusion process the derived inputs are local airlight estimation at two scales and the Laplacian of the input. The fusion weights are computed based on local contrast, saturation and saliency. The derived input and the weights are blended in multi-scale using a Laplacian and Gaussian pyramid respectively.

1.4 Contribution

Although attempts have been made to accurately estimate the scene transmittance, the estimation of environmental illumination has largely been ignored. Only a few methods have been proposed for its estimation [59, 12] and the only the recently proposed methods have considered to estimate this properly when proposing an end-to-end method [35]. For this reason, the methods that we propose here is mainly motivated by the cause how we may estimate the environmental illumination more accurately under different settings. The contributions of the proposed methods can be summarized as follows.

- Commonly it is assumed that during fog, the sky remains cloudy and the whole scene is receiving the same amount of illumination. But during fog and haze sky does not always remain cloudy, especially during haze condition. In that situation, mainly the direct sunlight contributes to the environmental illumination. Since in this case the whole scene may not receive the same amount of light, we say in this situation the intensity of environmental illumination can vary within the scene, but not its color. We have used the idea of Fattal [22] i.e. color line prior and have extended it to dehaze images under the proposed relaxation.

- The relaxation proposed for the previous method does not work for night-time images. Since during the night artificial lights are the only light source, the environmental illumination can vary both in terms of intensity and color. So, the imaging model is further relaxed to account for the spatially variant environmental illumination. But this introduces a new challenge. The number of illuminants is not known. Moreover, it is not possible to know in advance which illuminant affected which pixels. This has been tackled by the simple technique of Hough Transform. This has helped us to propose a method that works for both day and night-time images. Methods proposed till now works for only one kind of image, either daytime or night-time, not both.

- The use of color line introduces many assumptions and subsequently thresholds (around 10) to check the validity of the assumptions. This also results in the use of only a part of all possible patches in the estimation step. Others are not considered due to failure in the validity test. The tuning of the threshold values can also get quite hard in practice. For that, we move our attention to Convolutional Neural Networks (CNN). The CNNs have proved to be quite effective in automatic feature extraction, and have enjoyed success in many application [34, 18, 38].

- In our first CNN based method, we work with the basic atmospheric scattering model (equation 1.6) i.e. with assumption of constant environmental illumination and try to estimate transmittance and environmental illumination jointly from patches. Since quality of estimated transmittance depend on the evironmental illumination, we have proposed to estimate them jointly. But, it is seen trying to estimate environmental illumination from a small patch is error prone. The network learns to report the average color of the patch as the environmental illumination.

- In the next method we work with bigger patches to fix the issue with environmental illumination. But when using bigger patches the assumption of constant transmittance within a patch gets violated. So, given a big patch we need to estimate the transmittance of the same size. For this reason we use a Fully Convolutional Network to estimate the transmittance and airlight of the same size as the input. Although our network predicts the transmittance and airlight, the network is trained using pair of hazy and haze-free images only. This is enabled by our newly proposed loss (Bi-directional Consistency Loss), that facilitates the training of the network without ground-truth transmittance and airlight while conforming to the imaging equation at the same time. A multi-level strategy is also proposed to deal with the problem of resolution arising from variation in input image size. This method has originally been proposed for NTIRE 2018 challenge on image dehazing, and it is placed 5th in the competition [8].

- In our last method, we have proposed to estimate transmittance in each patch by comparing the dehazed version of the input image with the input hazy one, instead of directly regressing using a CNN. The desired transmittance is obtained by finding the one that clears the haze but does not overdo and produce bad looking output. Now whether the dehazed patch looks good or bad is decided by our proposed CNN based module called the patch quality comparator. But note that using this comparator we are only able to estimate the transmittance. Obtaining the environmental illumination in this way is not this much straight forward and left as a future work.

1.5 Organization of thesis

The thesis proposes methods to dehaze hazy images with a focus on estimation of environmental illumination. The thesis contains, apart from the introduction, six chapters. Their organization is described in the following subsections.

1.5.1 Variable environmental illumination intensity

Image dehazing methods commonly assume that the environmental illumination constant within the whole scene. But this does not remain true in many situations. In Chapter 2, we propose method where the constrain of constant environmental illumination is relaxed. The relaxation is made to handle the case when the intensity of environmental illumination varies within the scene but its color remains the same. The proposed method is based on the idea of color line based dehazing by Fattal [22], but unlike the method of Fattal [22] it estimates both the environmental illumination and the airlight to dehaze an image.

Related publication: Sanchayan Santra, and Bhabatosh Chanda. "Single image dehazing with varying atmospheric light intensity." In Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015 Fifth National Conference on, pp. 1-4. IEEE, 2015.

1.5.2 Variable environmental illumination intensity and color

In the next chapter (Chapter 3), the imaging model is further relaxed to handle the night-time images. In night time situations, both the color and intensity of the environmental illumination may vary spatially due to the presence of artificial lights. So, the imaging model is relaxed to model a spatially variant environmental illumination. With the help of this relaxed version of imaging model a dehazing method is proposed that works for both day and night-time images. Methods proposed till now work exclusively for either day or night-time images. But the method proposed here works independent of this criterion. This method is also based on the color line, but the relaxation introduces new challenge. From a given image obtaining the different illuminants present in the scene is not straight forward. But this is easily tackled with the use of Hough Transform.

Related publication: Sanchayan Santra, and Bhabatosh Chanda. "Day/night unconstrained image dehazing." In Pattern Recognition (ICPR), 2016 23rd International Conference on, pp. 1406-1411. IEEE, 2016.

1.5.3 Supervised with transmittance and environmental illumination

In this chapter (Chapter 4) we propose a method of image dehazing that jointly estimates transmittance and environmental illumination from image patches using a CNN. Methods have been proposed that employ a CNN to estimate the transmittance, but they don't focus the estimation on the estimation of environmental illumination. But the quality of dehazed image depends on the estimated environmental illumination. So, in the method proposed in this chapter, we estimate them jointly.

Related publication: Sanchayan Santra, Ranjan Mondal, Pranoy Panda, Nishant Mohanty, and Shubham Bhuyan. "Image Dehazing via Joint Estimation of Transmittance Map and Environmental Illumination", In Advances In Pattern Recognition (ICAPR), 2017 Ninth International Conference On. IEEE 2017.

1.5.4 Supervised with haze-free image only

The method proposed in Chapter 5 also jointly estimates the transmittance and airlight, but it works with bigger sized image patches. We work with bigger patches because it is seen, estimating environmental illumination from small patches is error prone. Since we work with bigger patches, we can't assume the transmittance to be constant within a patch. So, given a patch transmittance and airlight is estimated at each pixel. For this reason, a Fully Convolutional Network is utilized here. The network is trained using a newly proposed loss, called Bi-directional Consistency Loss. It requires only pair of hazy and haze free images and directs the network to estimate the transmittance and airlight in such a way, so that the haze-free image may be obtained from the hazy image using those estimates and vice versa. The method also proposes to tackles the challenge of image resolution by utilizing a muti-level approach.

Related publication: Ranjan Mondal, Sanchayan Santra, and Bhabatosh Chanda. "Image Dehazing by Joint Estimation of Transmittance and Airlight using Bi-Directional Consistency Loss Minimized FCN." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 920-928. 2018.

1.5.5 Patch quality comparator

In this chapter (Chapter 6) we propose a method that dehazes a given image by comparing various dehzed version of a given hazy patch with the original hazy version and choosing the best one. The comparison is performed by our proposed Convolutional Neural Network (CNN) based module called patch quality comparator. To select the best dehazed patch we employ binary search to find a dehazed patch such that its haze is cleaned, but at the same time quality has not degraded. This is quite different from the existing methods where a network is utilized to directly regress the haze paremeters.

Related publication: Sanchayan Santra, Ranjan Mondal, and Bhabatosh Chanda. "Learning a Patch Quality Comparator for Single Image Dehazing." IEEE Transactions on Image Processing (TIP) 27, no. 9 (2018).

1.5.6 Conclusion

In the last chapter we conclude this thesis by discussing the issues addressed in the previous chapter and also outline the possible future directions where the effort needs to be foucsed at to further progress the state-of-the art.

Chapter 2

Variable environmental illumination intensity

The commonly used atmospheric scattering model, what we have already described in chapter 1, is given by the following equation.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A}. \tag{2.1}$$

In this equation the environmental illumination ($\mathbf{A}$) is assumed to be constant throughout the scene. This is true if the sky is cloudy as the light from the sun gets diffused by the clouds and the whole scene receives more or less the same amount of light. In foggy weather the sky usually remains cloudy, and this assumption holds. But the sky does not always remain cloudy in these situations, specially in the haze conditions. Then the contribution of the direct sunlight in illumination becomes significant. So, different parts of the scene can receive light of different intensity. An example of such situation is shown in figure 2.1. That means, in this kind of situation the intensity of the environmental illumination can vary within the scene, but its color remains the same as sun is the only source of illumination. To model this scenario equation 2.1 needs to be modified as follows.

$$\begin{aligned} \mathbf{I}(\mathbf{x}) &= \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))m(\mathbf{x})\hat{\mathbf{A}}. \qquad (2.2) \\ &= \mathbf{J}(\mathbf{x})t(\mathbf{x}) + a(\mathbf{x})\hat{\mathbf{A}}. \qquad (2.3) \end{aligned}$$

This modification says that the color of environmental illumination (given by $\hat{\mathbf{A}}$) remains constant throughout the scene, but its intensity ($m(\mathbf{x})$) can vary. Now to recover haze free image using this equation, we need to estimate a single $\hat{\mathbf{A}}$, and the airlight component, denoted by $a(\mathbf{x})(= (1 - t(\mathbf{x}))m(\mathbf{x}))$, at each pixel. For this we take the idea of color line based dehazing by Fattal [22] and customize it to work under the modified imaging model. However, the method of Fattal [22] estimates transmittance only. It assumes

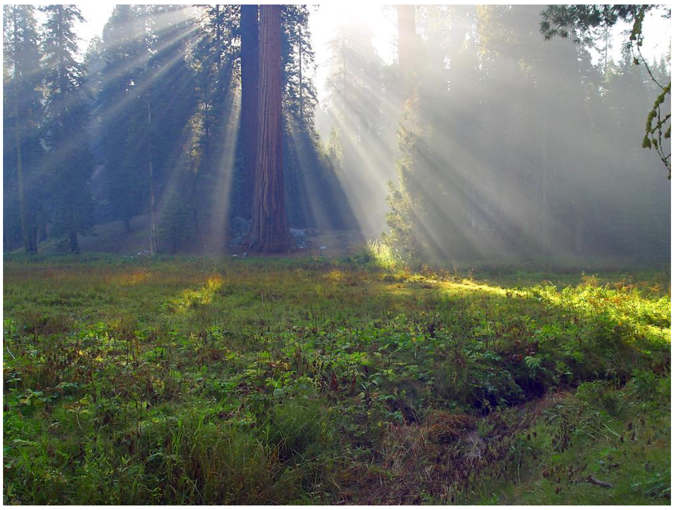


Figure 2.1: If the sunlight is dominant, the intensity of the environmental illumination can vary within the scene.

that the environmental illumination is known in advance. However, it is far from reality, especially when it is space variant. So, here we estimate the $\hat{\mathbf{A}}$ and $a(\mathbf{x})$. Unlike other methods, we do not estimate the transmittance of the medium directly.

2.1 Proposed Solution

The method that we propose here is based on color line [48]. So, in this section we first describe the color line model and how it is utilized in image dehazing. Then we outline how $\hat{\mathbf{A}}$ can be estimated with the help of the information obtained from the color line.

2.1.1 Color line and Hazy image

The color line model, as described in Omer and Werman [48], states that if we take a small patch of a natural image then the colors in that patch ideally lie on a line passing through the origin in the RGB space. But due to sensor and other camera related distortions, the colors spreads out and forms a cluster in the RGB space (figure 2.2[1]). This can be seen in the following way. Suppose for the colors within a patch we can write $\mathbf{I}(\mathbf{x}) = l(\mathbf{x})\mathbf{R}$, where $l(\mathbf{x})$ is the shading component and $\mathbf{R}$ is surface reflectance vector. Then we may say that, $\mathbf{R}$ provides the direction of the color line and $l(\mathbf{x})$ provides the position of the color points ($\mathbf{I}(\mathbf{x})$) in that direction. But this happens only if the patch contains a single object, that is, a constant surface reflectance R. Now, if we assume that

[1] Original image by Diego Delso, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=29956015`

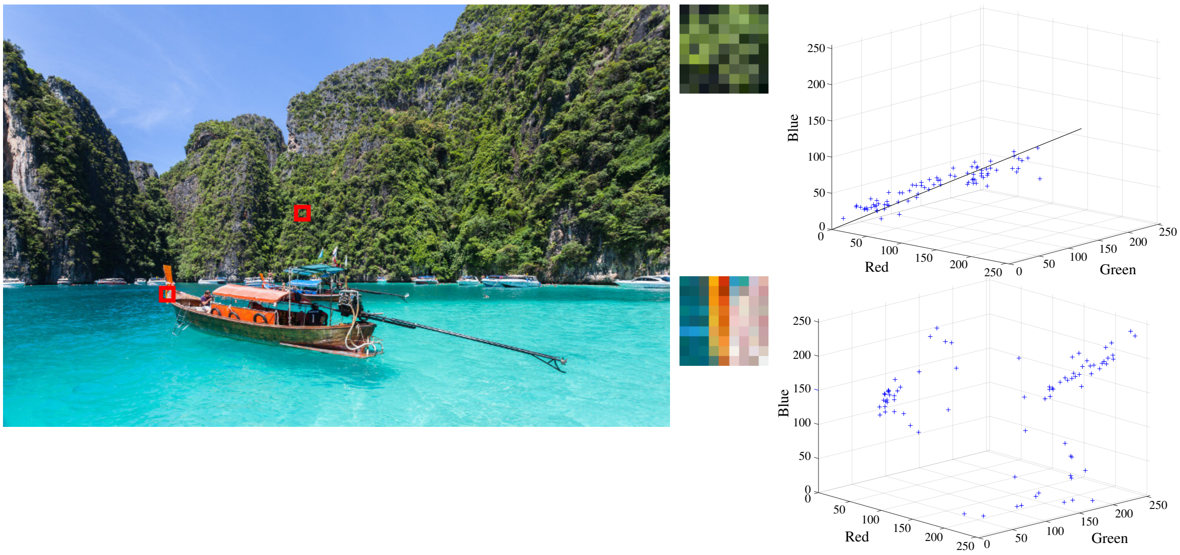


Figure 2.2: Colors of two different patches of an haze-free image plotted as points in the RGB space. As described by color line, these colors forms a cluster in the RGB space. The first patch contains a single object only, as a result the colors form a single elongated cluster. But the second patch contains more than one object (i.e. boat, water), so there are more than one cluster. Color line model fails in this case.

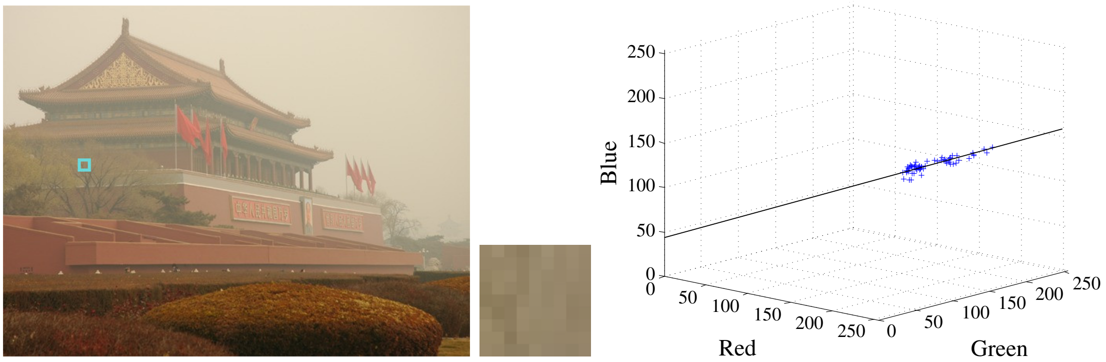


Figure 2.3: Colors in a patch of an hazy image plotted as points in the RGB space. Due to added haze the corresponding color line gets shifted and does not pass through origin.

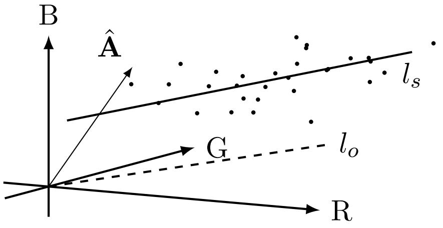


Figure 2.4: Colors in a patch plotted as points in RGB space and the corresponding fitted line l_s. The original line l_o got shifted due to haze in the direction given by $\hat{A}$ to form l_s.

all the pixels within the patch are affected by the same amount of haze (i.e. within a patch $t(\mathbf{x})$ and $m(\mathbf{x})$ is constant), then this line gets shifted by the amount given by the airlight component ($a(\mathbf{x})$) in the direction given by $\hat{\mathbf{A}}$ (figure 2.3 and 2.4). This occurs due to the additive airlight ($(1-t(\mathbf{x}))m(\mathbf{x})\hat{\mathbf{A}}$) present in equation 2.3. So, if we know $\hat{\mathbf{A}}$ and can determine this color line from small image patches, then the airlight component ($a(\mathbf{x})$) can easily be estimated by computing the magnitude of shift of the color line. But as mentioned before, for this to work we require following two information.

1. The patch should contain a single object only.
2. All the pixels in a patch should be affected by the same amount of haze.

These conditions hold, although not always, if the patches are sufficiently small. So one needs to use small patches to estimate the color line and the airlight component.

2.1.2 Estimation of $\hat{\mathbf{A}}$

In the previous section we have seen that from a patch of a natural image we may get a line passing through the origin formed by the RGB vector of the pixels in the patch. If the patch is affected by haze then this line (color line) gets shifted in the direction given by $\hat{\mathbf{A}}$. Note that the shifted line, the vector $\hat{\mathbf{A}}$ and the origin (of RGB space) lie on the same plane. In other words, the plane containing the shifted line and the origin, let us call them **patch planes**, also contains $\hat{\mathbf{A}}$. Since in our relaxed model (equation 2.3) we have assumed $\hat{\mathbf{A}}$ to be constant throughout the scene, each color line is shifted along the same $\hat{\mathbf{A}}$. So, a line depicting the direction $\hat{\mathbf{A}}$ is contained in all the patch planes. Therefore, if we get two patch planes that are not parallel then $\hat{\mathbf{A}}$ lies in the intersection of the patch planes (figure 2.5). Now instead of finding only a pair of non-parallel patch planes and computing their intersection, we compute the intersection of the all patch planes to obtain a more robust estimate of $\hat{\mathbf{A}}$. Note that each of the patches being considered for computing intersection should have a non-zero airlight component. Otherwise, the estimate will be error prone. Since, the normal to a plane is perpendicular to any line lying on the plane, the $\hat{\mathbf{A}}$ is also perpendicular to the normal of the patch plane. As, same direction vector $\hat{\mathbf{A}}$ lies in all the patch planes, we try to find a vector that is perpendicular to all the normals of the patch planes. Thus, we yield the desired $\hat{\mathbf{A}}$.

2.2 Dehazing Steps

Our proposed dehazing method takes the following 5 main steps to dehaze an image.

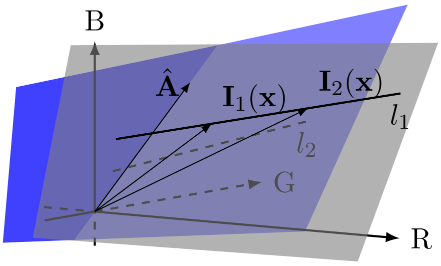


Figure 2.5: From two different patches we obtain two fitted line l_1 and l_2 and two patch planes. Since both the lines are shifted by the same $\hat{\mathbf{A}}$, both of the patch planes will contain the $\hat{\mathbf{A}}$. So, $\hat{\mathbf{A}}$ lies in the intersection of the two patch planes.

1. **Color line and patch plane estimation:** Since our method is based on color line, in the very first step we estimate it. We also estimate the patch planes in this step, as they are required for estimating the $\hat{\mathbf{A}}$.

2. **Estimation of $\hat{\mathbf{A}}$:** Once we obtain the patch planes, we can estimate the $\hat{\mathbf{A}}$ by determining their intersection as described previously.

3. **Estimation of airlight component ($a(\mathbf{x})$):** Once we have both the color line and $\hat{\mathbf{A}}$, we compute the amount of shift of color line from the origin in the direction of $\hat{\mathbf{A}}$.

4. **Aggregation and Interpolation of estimated $a(\mathbf{x})$:** The color line works only under certain assumptions and those assumptions may fail in many patches. In such cases, estimated $a(\mathbf{x})$'s are used to be erroneous and using those estimates in dehazing the image usually produces degraded output. So, we retain just the good estimates and then interpolate $a(\mathbf{x})$ at rest of the pixels.

5. **Haze free image recovery:** Once we have estimated $a(\mathbf{x})$ at all the pixels, we recover the haze-free image.

In the following subsections we describe each of the step in detail.

2.2.1 Color line and patch plane estimation

Our method hinges on the idea of color line. So, in the very first step we estimate the color line. For that we take the help of RANSAC [23]. We first divide the image into patches of size $\omega \times \omega$ with 50% overlap. Then on the RGB vectors of each patch we apply RANSAC to fit a line. After the fitting, RANSAC provides a set of points (inliers) that lies close to the estimated line and two points (say $\mathbf{I}_1$, $\mathbf{I}_2$) that lie on the line estimated using the reported inliers. Let's write the equation of the line in the following form:

$$\mathbf{L} = \rho\mathbf{D} + \mathbf{P}_0. \tag{2.4}$$

Here L denotes points on the line. D gives the direction of the line and P_0 is a point through which line passes. ρ is a free parameter. So, from the points provided by the RANSAC ($\mathbf{I}_1$ and $\mathbf{I}_2$), we can estimate the parameters of the color line as follows.

$$\mathbf{P}_0 = \mathbf{I}_1, \tag{2.5}$$

$$\mathbf{D} = \frac{\mathbf{I}_2 - \mathbf{I}_1}{||\mathbf{I}_2 - \mathbf{I}_1||}. \tag{2.6}$$

We also need to estimate the patch planes to be able to compute the $\hat{\mathbf{A}}$. Since the patch plane we are trying to estimate contains the origin and the estimated line, the normal to this plane can be computed using the vectors joining the origin and the points on the line. So, from the output of the RANSAC the normal ($\hat{\mathbf{A}}$) can be computed as follows,

$$\hat{\mathbf{n}} = \frac{\mathbf{I}_1 \times \mathbf{I}_2}{||\mathbf{I}_1 \times \mathbf{I}_2||}. \tag{2.7}$$

Note that in the subsequent steps of our method only the inlier points of the patch are used, because the outliers are not part of the color line. However, the color line estimated in this way can be erroneous, if the assumptions made in section 2.1.1 (i.e., the patch is a part of a single object and, thus, is affected by same amount of haze) are violated or the data is noisy. Thus, using those patches may lead to wrong estimation of the airlight component. So, the estimates need to be validated using the following tests.

- If the number of inliers reported by RANSAC is small, then the estimated color line is likely to be bad. In fact, an estimated color line that would be used in further computation only if the number of inliers is greater than a fraction (θ_r) of total number of points in the patch.

- Since color line direction $\mathbf{D}$ represents hue (in some form) of the patch, its all three components should be positive.

- If a patch contains more than one object then there exists a possibility of depth discontinuity. In that case, our assumption of color line is violated. The assumption of all pixels being affected by same amount of haze is also violated. So, we check for the existence of an edge in the patch by thresholding (θ_g) the gradient magnitude of the patch, and only the patches without edges are used to estimate the color line.

- If the estimated color line lies close to the origin, then the patch is not much affected by haze. Using the patch plane obtained from this line can affect the estimate of $\hat{\mathbf{A}}$. So, a patch plane is used for computing $\hat{\mathbf{A}}$ only if the corresponding estimated color line be at least d_0 distance away from the origin.

2.2.2 Estimation of $\hat{\mathbf{A}}$

In the previous step we have obtained normal ($\hat{\mathbf{n}}$) of the patch planes and we have already shown that $\hat{\mathbf{A}}$ is (ideally) perpendicular to all the normal of the patch planes. But, in reality we may only get a vector that is perpendicular to most but not all $\hat{\mathbf{n}}$. So, we compute $\hat{\mathbf{A}}$ by minimizing the following error

$$E(\hat{\mathbf{A}}) = \sum_i (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{A}})^2. \tag{2.8}$$

Here $\hat{\mathbf{n}}_i$ denote the normal of the i^{th} patch plane. Minimizing this equation boils down to solving the following equation,

$$\Big(\sum_i \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i^T\Big)\hat{\mathbf{A}} = 0. \tag{2.9}$$

As we know $\hat{\mathbf{A}}$ denote the color of the environmental illumination, it can't be a null vector. So, we need a non-trivial solution of this equation. Therefore, we compute the eigen vectors of $\sum_i \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i^T$. Ideally, the eigen vector with eigen value 0 gives the solution. But due to estimation errors we may not always get a 0 eigen value. So, we accept the eigen vector corresponding to the smallest eigen value as the solution.

Although all the normals together yields the result, to make the estimate more robust we discard some of the normals from our selection based on the dark channel value [28] of the corresponding patch. This is done to ensure that only the patches with high amount of haze contributes to the computation of $\hat{\mathbf{A}}$. The dark channel value of a patch Ω_i is given by

$$D_H(\Omega) = \min_{\mathbf{x}\in\Omega}\Big(\min_{c\in\{R,G,B\}} I_c(\mathbf{x})\Big). \tag{2.10}$$

Thus, the normal corresponding to patch Ω_i is discarded if the following condition holds.

$$D_H(\Omega_i) \leq \theta_D \max_{\Omega_j} D_H(\Omega_j). \tag{2.11}$$

2.2.3 Estimation of Airlight Component ($a(\mathbf{x})$)

In the previous steps we have estimated the color lines from the patches and also the vector $\hat{\mathbf{A}}$. So, the airlight component ($a(\mathbf{x})$) can be obtained by computing the amount of shift of the fitted line in the direction of $\hat{\mathbf{A}}$. If we use the form of equation of line given previously (equation 2.4), then the shift can be computed by solving the equation

$$\mathbf{P}_0 + \rho\mathbf{D} - \delta\hat{\mathbf{A}} = 0. \tag{2.12}$$

where δ gives the amount of shift in the direction of $\hat{\mathbf{A}}$. This equation basically says that the fitted color line ($\mathbf{P}_0 + \rho\mathbf{D}$) needs to be shifted by δ amount in the direction given by $-\hat{\mathbf{A}}$, so that the color line would pass through the origin under dehazed condition. But due to noise and error in estimation the color line and $\hat{\mathbf{A}}$ may not always intersect. So, it may not always be possible to make the fitted line pass through the origin by shifting it in the direction of $-\hat{\mathbf{A}}$. So, instead we estimate the shift (δ) in such a way that the distance of the fitted line from the origin is minimum when it is shifted by $-\delta\hat{\mathbf{A}}$. This is achieved by minimizing the following equation.

$$E_l(\rho, \delta) = ||\mathbf{P}_0 + \rho\mathbf{D} - \delta\hat{\mathbf{A}}||^2. \tag{2.13}$$

As shown in Fattal [22], the ρ and δ that minimizes this equation can be found by solving the following equation

$$\begin{bmatrix} ||\mathbf{D}||^2 & -(\hat{\mathbf{A}} \cdot \mathbf{D}) \\ -(\hat{\mathbf{A}} \cdot \mathbf{D}) & ||\hat{\mathbf{A}}||^2 \end{bmatrix} \begin{bmatrix} \rho \\ \delta \end{bmatrix} = \begin{bmatrix} -(\mathbf{D} \cdot \mathbf{P}_0) \\ \hat{\mathbf{A}} \cdot \mathbf{P}_0 \end{bmatrix}. \tag{2.14}$$

Here both $\mathbf{D}$ and $\hat{\mathbf{A}}$ provides direction only. So, we can say $||D||$ and $||A||$ is 1. Then solution is given by following.

$$\begin{bmatrix} \rho \\ \delta \end{bmatrix} = \frac{1}{1 - (\hat{\mathbf{A}} \cdot D)^2} \begin{bmatrix} 1 & (\hat{\mathbf{A}} \cdot \mathbf{D}) \\ (\hat{\mathbf{A}} \cdot \mathbf{D}) & 1 \end{bmatrix} \begin{bmatrix} -(\mathbf{D} \cdot \mathbf{P}_0) \\ \hat{\mathbf{A}} \cdot \mathbf{P}_0 \end{bmatrix}. \tag{2.15}$$

Since we are working under non-ideal conditions, the computed estimates needs to be validated before they are used in subsequent steps. After validation the estimated δ is assigned to all the inlier pixels of the patch (as reported by RANSAC) as the $a(\mathbf{x})$. For the validation the following checks are employed by our method.

- We are trying to compute the airlight component by finding the shift of the fitted color line in the direction given by $\hat{\mathbf{A}}$. But if the fitted line is parallel to $\hat{\mathbf{A}}$, then shift cannot be determined. In general, the estimate of shift (δ) becomes sensitive to noise as the angle between $\mathbf{D}$ and $\hat{\mathbf{A}}$ decreases. So, to retain only the good estimates we use the computed shift only if the angle between $\mathbf{D}$ and $\hat{\mathbf{A}}$ is more than some threshold θ_E.

- Ideally, the fitted line and $\hat{\mathbf{A}}$ should intersect in the RGB space. But as this may not happen always, we minimize the error function $E_l(\rho, \delta)$. If the obtained error after minimization is quite high (i.e., greater than some threshold E_θ), that means the fitted line remains quite far from the origin after being shifted by $-\delta\hat{\mathbf{A}}$. Use of such estimate of shift is likely to introduce error. So, these are not used in further computation.

- We are trying to estimate the airlight component ($a(\mathbf{x}) = (1 - t(\mathbf{x}))m(\mathbf{x})$) by computing the shift (δ). From the imaging model, we know $t(\mathbf{x})$ varies between 0 and 1. Apart from that we are working with pixel values between 0 and 1. That means, $m(\mathbf{x})$ should lie between 0 and $\sqrt{3}$ (when $m(\mathbf{x})\hat{A}$ is $[1, 1, 1]^T$). So, δ should also lie between 0 and $\sqrt{3}$. Any estimate of δ that goes beyond this range is invalid. But this bound sometimes allows overestimation of $a(\mathbf{x})$. So, we use the smallest intensity present in the patch as the upper limit. If the estimate of shift falls outside this range it is not considered.

- We are trying to estimate the color line by fitting a line to the pixel colors in the RGB space. If the colors do not spread much in the direction of the line, then the fitted line becomes sensitive to noise. To discard such badly fitted lines, we project the inlier points on the estimated line and compute the standard deviation. If this standard deviation falls below a certain threshold (θ_s) we discard the fitted line and the corresponding estimated shift as well.

2.2.4 Aggregation and Interpolation of Estimated $a(\mathbf{x})$

The estimates obtained till now are done on patches extracted from an image. For that we have extracted patches with 50% overlap. So, naturally a pixel may receive more than one estimate and these need to be aggregated. In this work, aggregating is implemented as *max* function. That means, we retain the value with the maximum airlight component. Thus we obtain a single estimate of $a(\mathbf{x})$ at the pixels. But, as mentioned earlier, in the process of estimating the airlight component ($a(\mathbf{x})$) we have discarded quite a few patches, where the estimates are potentially incorrect. So, at those pixels we don't have an estimate of $a(\mathbf{x})$, which is required to recover the haze free image at every pixel. So, we interpolate the $a(\mathbf{x})$ at those pixels by minimizing the following function [inspired from Fattal's [22] GMRF based regularizer].

$$\psi(a(\mathbf{x})) = \sum_{\Omega} \sum_{\mathbf{x} \in \Omega} \frac{(a(\mathbf{x}) - \tilde{a}(\mathbf{x}))^2}{(\sigma_a(\Omega))^2} + \alpha \sum_{\mathbf{x}} \sum_{\mathbf{y} \in L(\mathbf{x})} \frac{(a(\mathbf{x}) - a(\mathbf{y}))^2}{\|\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})\|^2} + \beta \sum_{\mathbf{x}} \frac{a(\mathbf{x})}{\|\mathbf{I}(\mathbf{x})\|}. \quad (2.16)$$

where $\tilde{a}(\mathbf{x})$ is the estimated airlight component. $\tilde{a}(\mathbf{x})$ is 0 where the estimates are not obtained. $L(\mathbf{x})$ denotes the neighborhood of $\mathbf{x}$, normally it is the 4-neighborhood. $a(\mathbf{x})$ is the airlight component to be interpolated. Here α and β are two scalars that controls the importance of the corresponding terms in this function. Ω denotes a patch and $\sigma_a(\Omega)$ is the error variance of the estimate within the patch. $\sigma_a(\Omega)$ is taken to be 0 where the estimates are not available; otherwise it is estimated assuming the colors in the patch is corrupted by Gaussian noise with 0 mean and variance σ^2. The estimate of the error is computed in the same way what is done by Fattal [22]. Let's assume $\boldsymbol{\mathcal{E}}$ is the error in the estimated line offset ($\mathbf{P}_0$). That means $\mathbf{P}_0 = \delta\hat{\mathbf{A}} + \boldsymbol{\mathcal{E}}$. Then the estimated shift becomes

$$\begin{aligned} \delta' &= \frac{(\hat{\mathbf{A}} \cdot (\delta\hat{\mathbf{A}} + \boldsymbol{\mathcal{E}})) - (\mathbf{D} \cdot (\delta\hat{\mathbf{A}} + \boldsymbol{\mathcal{E}}))(\hat{\mathbf{A}} \cdot \mathbf{D})}{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2} \\ &= \frac{\delta + (\hat{\mathbf{A}} \cdot \boldsymbol{\mathcal{E}}) - \delta(\mathbf{D} \cdot \hat{\mathbf{A}})^2 - (\mathbf{D} \cdot \boldsymbol{\mathcal{E}})(\mathbf{D} \cdot \hat{\mathbf{A}})}{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2} \\ &= \delta \frac{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2}{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2} + \frac{(\hat{\mathbf{A}} \cdot \boldsymbol{\mathcal{E}}) - (\mathbf{D} \cdot \boldsymbol{\mathcal{E}})(\mathbf{D} \cdot \hat{\mathbf{A}})}{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2} \\ &= \delta + \frac{(\hat{\mathbf{A}} - \mathbf{D}(\mathbf{D} \cdot \hat{\mathbf{A}})) \cdot \boldsymbol{\mathcal{E}}}{1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2} \end{aligned} \quad (2.17)$$

Therefore, the variance of the estimated shift is

$$\sigma^2 \frac{||\hat{\mathbf{A}} - \mathbf{D}(\mathbf{D} \cdot \hat{\mathbf{A}})||^2}{(1 - (\mathbf{D} \cdot \hat{\mathbf{A}})^2)^2}. \quad (2.18)$$

The first term of the function to be minimized (equation 2.16) ensures that the computed airlight component is close to the estimated one while considering the estimate to contain Gaussian noise with variance $\sigma_a(\Omega)$. The second term is the regularization term that penalizes the variation of $a(\mathbf{x})$ in the local neighborhood based on the smoothness of $\mathbf{I}(\mathbf{x})$. This also acts as an interpolator at pixels where estimates are not computed, because the first term becomes effectively 0 for those pixels. The last term ensures that the airlight component would be a small fraction of $\mathbf{I}(\mathbf{x})$. Since $\mathbf{I}(\mathbf{x})$ is the combination of both direct transmittance and airlight (equation 2.3), the airlight component should not be greater than $||\mathbf{I}(\mathbf{x})||$. Now instead of using the equation 2.16 directly, we use the following form of this optimization function to minimize it.

$$\Psi(\mathbf{a}) = (\mathbf{a} - \tilde{\mathbf{a}})^T \boldsymbol{\Sigma} (\mathbf{a} - \tilde{\mathbf{a}}) + \alpha \mathbf{a}^T \mathbf{L}_g \mathbf{a} + \beta \mathbf{b}^T \mathbf{a}. \tag{2.19}$$

Here $\mathbf{a}$ and $\tilde{\mathbf{a}}$ are the vector form of $a(\mathbf{x})$ and $\tilde{a}(\mathbf{x})$. $\boldsymbol{\Sigma}$ is a diagonal matrix with its entries given by $\sigma_a(\Omega)$. $\mathbf{L}_g$ is the Laplacian matrix of the graph constructed by taking each pixel as one vertex and connecting it with its neighbors as defined by $L(\mathbf{x})$. In this graph the weight of an edge between the vertices $\mathbf{x}$ and $\mathbf{y}$ is taken as $\frac{1}{\|\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})\|^2}$. Each element of $\mathbf{b}$ is $\frac{1}{\|\mathbf{I}(\mathbf{x})\|}$, and α and β are the same importance controlling scalars. The equation 2.19 is minimized by solving the following equation.

$$(\boldsymbol{\Sigma} + \alpha \mathbf{L}_g) a = (\boldsymbol{\Sigma} \tilde{\mathbf{a}} - \beta \mathbf{b}). \tag{2.20}$$

2.2.5 Haze-free Image Recovery

We have obtained $\hat{\mathbf{A}}$ and $a(\mathbf{x})$ at each pixel. So, airlight can now be obtained by computing $a(\mathbf{x})\hat{A}$. Subtracting this airlight from the input image, we get the airlight removed image.

$$\mathbf{I}_{\bar{a}}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) - a(\mathbf{x})\hat{\mathbf{A}}. \tag{2.21}$$

This removes the translucent veil of the haze. But it does not fix the reduction in intensity due to scattering. For that we require the transmittance ($t(\mathbf{x})$). But this can't obtained directly from the computed airlight. So, instead we enhance the contrast of the airlight removed image based on the amount of intensity that is removed from it as airlight. If we denote the recovered image as $\mathbf{J}'(\mathbf{x})$, then it is obtained as follows.

$$J'_c(\mathbf{x}) = \frac{\mathbf{I}_{\bar{a}}(\mathbf{x})}{1 - Y(a(\mathbf{x})\hat{\mathbf{A}})}, \text{ where, } c \in \{R, G, B\} \tag{2.22}$$

$$Y(a(\mathbf{x})\hat{\mathbf{A}}) = 0.2989 \times a(\mathbf{x})\hat{A}_R + 0.5870 \times a(\mathbf{x})\hat{A}_G + 0.1140 \times a(\mathbf{x})\hat{A}_B. \tag{2.23}$$

For many images the output stays dark even after this operation so we use gamma correction to improve the overall brightness.

2.3 Experimental Settings

The setting under which we generate the results are given as follows. To generate all the results we have used the MATLAB implementation of our method. For the line fitting part with RANSAC, we have used the code of Peter Kovesi [33]. The default values that we have used for the different parameters are given in Table 2.1. These values are kept

Table 2.1: Default parameter values

θ_r	θ_g	d_0	θ_E	E_θ	θ_s	θ_D
0.4	0.06	0.005	15°	0.05	0.006	0.45

same throughout our experiments and for all the images, although fine tuning them may improve the result in some cases. The parameters α and β usually take small values (typically around 2×10^{-4} and 1×10^{-5}) depending on the input image. Note that in our implementation pixel values lie between 0 and 1. The parameter values also depend on this range.

2.4 Results

To evaluate the performance of our method we have used a hazy image dataset provided by Fattal [22] in addition to using some real-world hazy images obtained from the internet. The dataset provided by Fattal [22] contains both synthetic and real-world hazy images. For comparative study, we have used the results of two methods. One is by He et al. [28] and the other by Fattal [22]. We have used our own implementation of the method of He et al. [28] to generate its results, while the results of Fattal's method are obtained from the author's website[2]. Note that in Fattal's method the environmental illumination is not computed. It is supplied manually; whereas in our method and method of He et al. [28] the environmental illumination is computed from the image.

2.4.1 Quantitative Results

For the real-world hazy images, most of the time, we don't have the corresponding ground-truth (i.e., haze-free) images. Image of the same location can be taken both during haze and haze-free situation. But this may not always be feasible. So, the common approach is to use synthetically generated hazy images generated by adding haze to haze-free images using the atmospheric scattering model (equation 1.6 and 1.5). Fattal has provided one such set of synthetic images to facilitate quantitative evaluation. This dataset contains 12 synthetically hazy images of both indoor and outdoor scenes. The hazy images has been generated assuming constant scattering coefficient and using the same environmental illumination ($[0.5, 0.6, 1]^T$) for all the images. As we have the ground truth haze-free images for each of the hazy images, we employ three full-reference metrics to quantitative evaluation of the results: Peak signal-to-noise raio (PSNR), Structural

[2] `http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/`

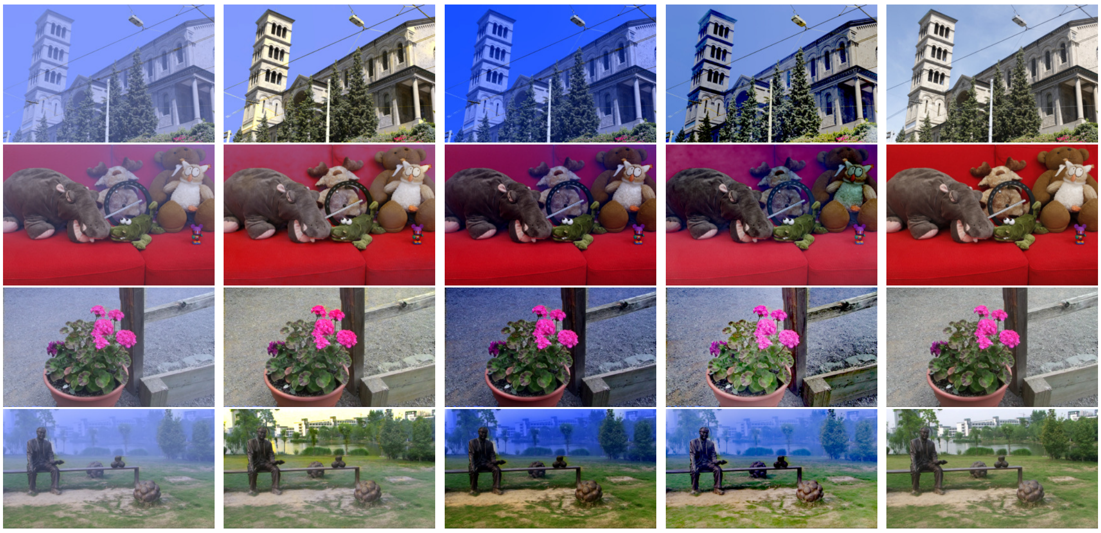
Table 2.2: Quantitative Comparison on the images of Fattal's dataset. High PSNR and SSIM indicates better results, while it is the opposite for ΔE_{00}. The best results are bold and the second best results are underlined. Note that in Fattal's method only $t(\mathbf{x})$ is computed and $\mathbf{A}$ is manually provided.

Image	Fattal [22]			He et al. [28]			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	**21.43**	**0.96**	**6.34**	11.16	0.78	28.74	14.84	0.76	15.54
couch	**20.8**	**0.9**	**6.71**	18.4	0.86	13.89	16.08	0.78	16.11
dolls	**21.29**	**0.77**	**6.1**	19.73	0.85	10.65	15.5	0.77	17.61
flower1	**30.01**	**0.98**	**3.91**	14.1	0.88	23.26	16.8	0.89	13.86
flower2	**31.94**	**0.99**	**2.92**	14.37	0.86	20.94	17.02	0.85	13.92
lawn1	**24.49**	**0.97**	**6.65**	13.84	0.8	22.38	14.71	0.82	20.01
lawn2	**24.94**	**0.97**	**6.46**	11.2	0.74	29.32	14.52	0.83	20.23
mansion	**26.96**	**0.97**	**4.04**	17.45	0.87	19.35	17.49	0.85	16.62
moebius	**19.01**	**0.9**	**10.61**	12.66	0.78	26.7	15.85	0.86	17.46
raindeer	**26.22**	**0.94**	**4.1**	18.12	0.83	14.22	13.59	0.73	22.54
road1	**25.74**	**0.96**	**5.24**	12.95	0.8	26.11	14.16	0.84	20.78
road2	**23.6**	**0.96**	**7.11**	15.84	0.84	22.13	15.88	0.86	20.13
Average	**24.7**	**0.94**	**5.85**	14.98	0.82	21.47	15.54	0.82	17.9

Similarity Index (SSIM) and CIEDE2000 (ΔE_{00}). PSNR is a measure of the quality of reconstruction of signals. In this context it measures how much the dehazed image is close to the original haze-free image. A higher PSNR value indicates that the output is closer to the haze-free image. SSIM [64] determines the similarity between two images based on how human perceive change in structural information. It gives scores between -1 and 1. An SSIM value of 1 indicates two images are structurally similar. ΔE_{00} [56] measures the perceptual dissimilarity between colors. So, a small value indicates the colors are similar. As restoration of colors is a crucial aspect of image dehazing, we use the average ΔE_{00} to measure how much the colors are restored to their original state. The quantitative results obtained on the three above mentioned metrics are reported in Table 2.2. We see the results of Fattal [22] is always leading with a significant margin mainly because the environmental illumination is supplied manually. The method of He et al. [28] fails in most of the cases because it has not been able to estimate $\mathbf{A}$ properly. Our method performs better than the method of He et al. [28] as we are estimating the $\hat{\mathbf{A}}$ from all parts of the image.

2.4.2 Qualitative Results

For qualitative evaluation of our method we have used both synthetic and real world hazy images. The four synthetic images on which we show the results are the part of the images that we have utilized for quantitative evaluation. The comparative results



(a) Input (b) Fattal [22] (c) He et al. [28] (d) Our (e) Ground Truth

Figure 2.6: Visual comparison of the results on four synthetic images: *church*, *couch*, *flower2*, and *lawn1*.

are given in figure 2.6. The results reflect what we have already seen in the quantitative scores. The method of Fattal [22] performs the best. The method of He et al. [28] performs the worst. It could not clear the color introduced by the haze. A blueish tint remains in all the images. Our method is not as good as the Fattal's method, but it performs well in many cases (e.g. *flower2*).

For comparing the performance on real-world hazy images, we have used five images: four benchmark images (figure 2.8) and one obtained from Wikimedia Commons[3] (figure 2.7). We see that the results obtained by Fattal has high contrast which may not always be visually pleasing. He et al. [28] has not been able to clear the haze in all the cases. The sky is not handled properly by our method in the first two images of figure 2.8 (*dubai* and *florence*); the noise has been amplified. The colors appear much more soothing in the results obtained by our method especially in the last two images (*tiananmen* and *ny12*). Fig. 2.7 shows an image where the intensity of environmental illumination varies within the scene, because sunlight is dominating the illumination. Here we see that he He et al. [28] has completely failed to clear the haze; Whereas our method is able to clear the haze, handling the variation in the sunlight within the scene.

The results of figure 2.7 shows a case of varying environmental illumination where not all parts of the image is receiving the same amount of light. This has been very well

[3] “Oberfallenberg4” by böhringer friedrich - Own work. Licensed under CC BY-SA 2.5 via Wikimedia Commons

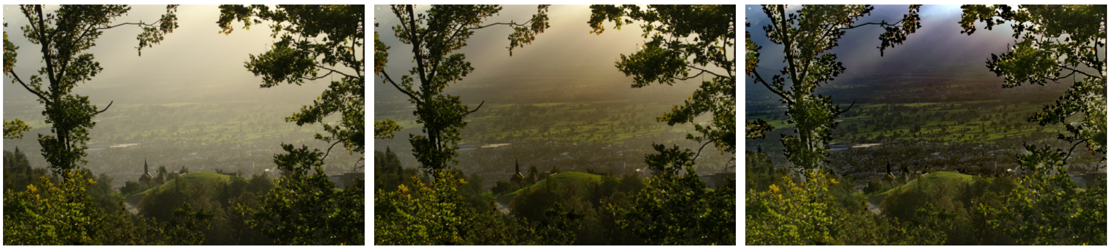
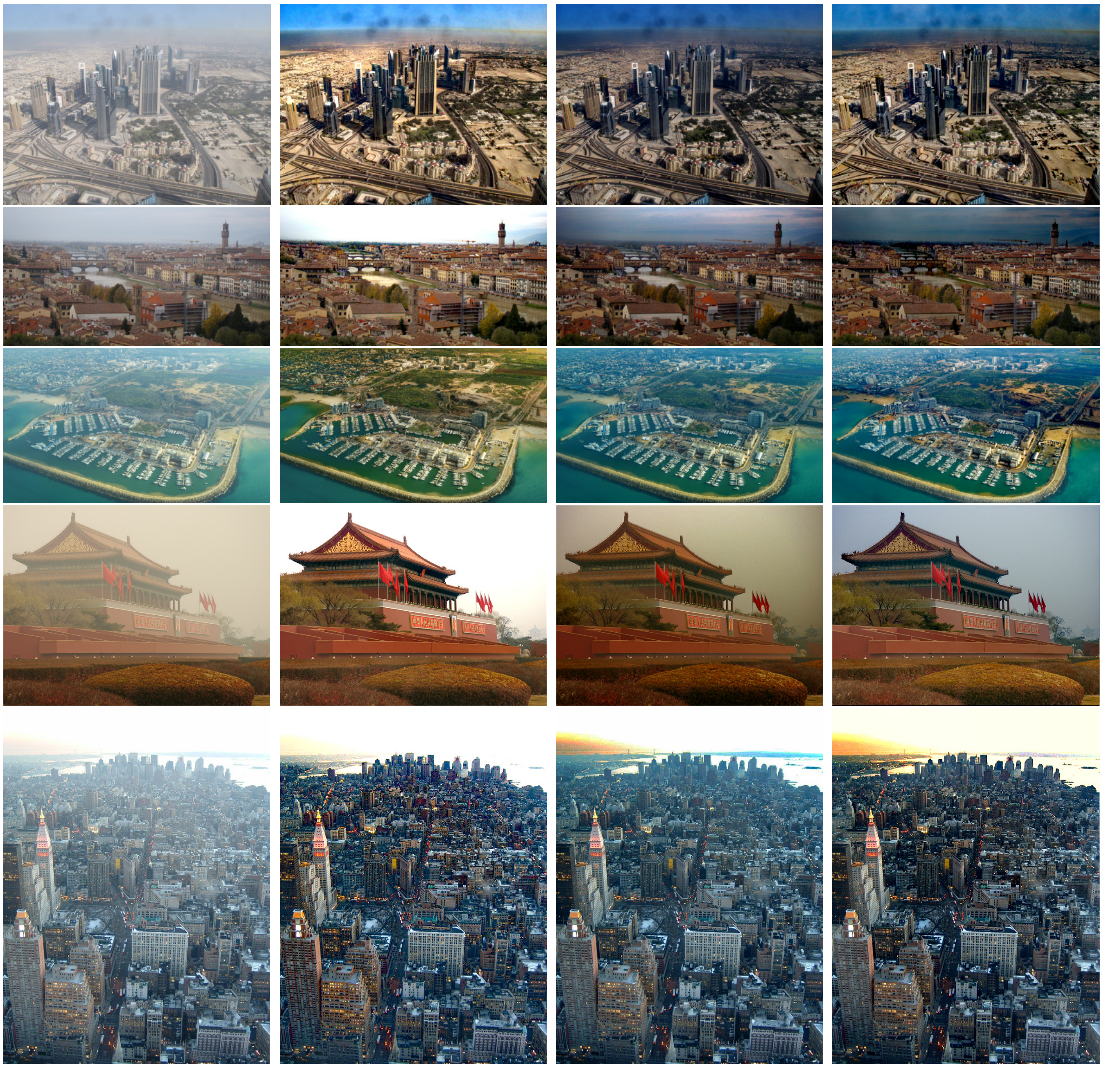


Figure 2.7: (from left to right) Input image, result of He et al. [28] and our method.

tackled by our method other than the 'god-rays' that are visible in the output of our method. Although the variation in illumination has been handled successfully, mainly in the top of the image some of the rays are visible. The estimate of color line using RANSAC becomes error prone if the patch being considered is smooth. The top of the image, where the 'god-rays' are visible, appears to be smooth because the airlight brightness in the area is oversaturated and no trace of obscured reflectance is available. As a result the color line is not estimated at those places and the transmittance is interpolated from the near-by locations. For this reason our method is not able to detect variations in those regions; while the variations present at other places of the image has been handled successfully. Another factor is that the color of the rays are different from the color of the airlight. So, those are not considered to be a part of the haze.

2.5 Summary

In this chapter we have proposed an image dehazing method using a relaxed haze image formation model. We use the color line model to estimate color line direction $\hat{\mathbf{A}}$ from multiple patches of the image, then we estimate the contribution of airlight in each patch. We interpolate airlight magnitude at pixels where the estimates are unreliable. Then the image forming equation is used to recover haze free image. Here we have estimated the airlight contribution $(a(\mathbf{x})\hat{A})$ at each patch but not the scene transmittance $t(\mathbf{x})$. So, unlike other methods we do not compute the depth map. Our method assumes that within an image, $\hat{A}$ is constant but the magnitude of airlight varies. So, our method may fail to give satisfactory results where this is violated, e.g. nighttime haze images. We discuss this problem next.



(a) Input (b) Fattal [22] (c) He et al. [28] (d) Our

Figure 2.8: Visual comparison of results on *dubai*, *florence*, *herzeliya*, *tiananmen*, and *ny12* image.

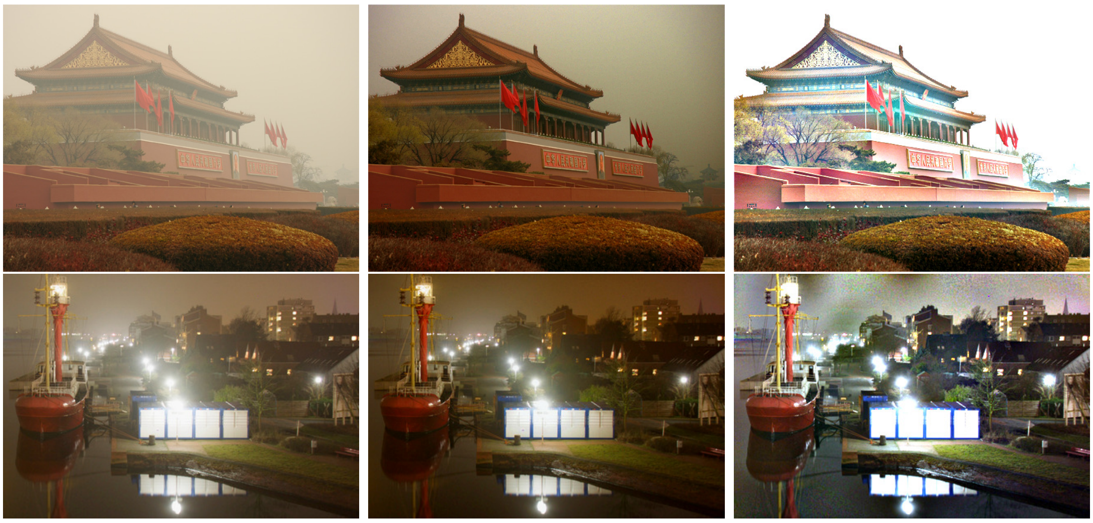
Chapter 3

Variable environmental illumination intensity and color

In the previous chapter we have considered the situations where only the intensity of environmental illumination varies within the scene but its color remains the same, and modified the imaging model accordingly. As said before, that model can't handle variations in color of the illumination within the scene, e.g. in the images taken during the night when artificial lights are the main illuminants. However, light undergoes the same change under foggy environment at night as it is in day time. The only change that occurs when we move from day-time imaging to night-time imaging is that instead of the sun we have artificial lights as illuminants. These artificial lights may not have the same color or the same intensity. So, the assumption of constant $\mathbf{A}$ (or $\hat{\mathbf{A}}$) fails in this situation. Therefore to dehaze night-time images one common approach is to modify the atmospheric scattering model to the following form [67].

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A}(\mathbf{x}). \tag{3.1}$$

Put it simply, the constant $\mathbf{A}$ is relaxed to $\mathbf{A}(\mathbf{x})$ to account for the spatially varying illumination (both intensity and colour). Although this relaxed model is being used for dehazing night-time images, this can also be used to dehaze day-time images. The model used for day-time imaging (equation 1.6) is just a special case of this model (equation 3.1). Therefore using this relaxed model (equation 3.1) it is possible to dehaze both day-time and night-time images. Usually, image dehazing methods focus on dehazing either day-time or night-time images. So day-time image dehazing methods fail on night-time images, and night-time image dehazing methods fail on day-time images (figure 3.1). In this chapter we propose a method that can dehaze both day-time and night-time images. In fact, this method extends the method proposed in the previous chapter so that it also works for night-time hazy images. Here the basic approach remains the same, only the method of estimating the environmental illumination is changed to account for the



(a) Input (b) He et al. [28] (c) Li et al. [36]

Figure 3.1: Day time dehazing methods [28] works well for daytime images, but does not work satisfactorily in night time images. Whereas night-time dehazing methods ([36]) works well for night-time images, but fails to work properly for daytime images.

relaxation made to the imaging model (equation 3.1). The method being proposed here is also based on the color line model and applies the same principle to compute the airlight component. That is, as before, for a patch of a natural image we may get a line formed by the RGB vectors of the patch and the line passes through the origin of the RGB-space. For a hazy image this line gets shifted in the direction of $\hat{\mathbf{A}}$. But, this has two inherent assumptions.

- All the pixels of the patch are affected by the same amount of haze.
- The color of environmental illumination ($\hat{\mathbf{A}}$) incident on the patch is same for all of its pixels.

The first assumption holds if the patch is sufficiently small so that it does not contain depth discontinuity. The second assumption did not have any effect in the previous method, because we have only tackled the case where $\hat{\mathbf{A}}$ remains constant throughout the image. But in the relaxed model we are considering here, $\mathbf{A}$ varies spatially. So, we need to make also the second assumption. Since illumination varies smoothly within the scene, this holds if the patch size is sufficiently small. So in addition to estimating the color line from the patches, we have to estimate the $\mathbf{A}$ it got affected with. For that we decompose the relaxed model (equation 3.1) to the following form similar to what is done in the previous chapter.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))m(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x}), \tag{3.2}$$

$$= \mathbf{J}(\mathbf{x})t(\mathbf{x}) + a(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x}). \tag{3.3}$$

That is, $\mathbf{A}(\mathbf{x})$ is decomposed into its magnitude $m(\mathbf{x})$ and direction $\hat{\mathbf{A}}(\mathbf{x})$ both of which are space variant. We denote the airlight component with $a(\mathbf{x})$. We first estimate $\hat{\mathbf{A}}(\mathbf{x})$ and $a(\mathbf{x})$ from patches, and then $\mathbf{I}(\mathbf{x})$ at each pixel in order to recover the dehazed image.

3.1 Proposed Method

The method we propose here is based on the same principle as the previous method [see Chapter 2]. That is why the five main steps remain almost the same as stated before. Only a few of them are modified to account for the space-variant environmental illumination. So, the set of updated steps is as follows.

1. **Color line and patch plane estimation:** We start with estimating the color line from patches, as the rest of the steps hinges on this estimation. Moreover, we estimate the patch planes (*i.e.* their normals) which are required for estimating the $\hat{\mathbf{A}}$'s.

2. **Estimation of $\hat{\mathbf{A}}$'s:** To estimate the amount of shift of the color line, we first need to know the direction along which it got shifted (here $\hat{\mathbf{A}}$). Since under the relaxed model (equation 3.1) the $\hat{\mathbf{A}}$ can vary spatially, different patches get affected by different $\hat{\mathbf{A}}$'s. So, in this step for each patch we estimate the corresponding $\hat{\mathbf{A}}$, and this association between $\hat{\mathbf{A}}$ and the patch is maintained through out.

3. **Estimation of airlight component ($a(\mathbf{x})$):** Once we have the color line and $\hat{\mathbf{A}}$ for a patch, we estimate the shift of the color line in the direction of $\hat{\mathbf{A}}$ to get the airlight component of that patch.

4. **Aggregation and Interpolation of estimated A's and $a(\mathbf{x})$:** The assumptions, under which the concept of color line holds, may not be valid for all the patches. So, we don't use the estimates (of both $\hat{\mathbf{A}}$ and $a(\mathbf{x})$) obtained at those patches where the assumptions can potentially fail. But, for image dehazing, we have to have $\hat{\mathbf{A}}$ and $a(\mathbf{x})$) at those places. So, instead of using unreliable estimated values, we interpolate the values at those patches (or pixels) based on neighbouring estimates.

5. **Haze free image recovery:** Once we have $\hat{\mathbf{A}}$ and $a(\mathbf{x})$) for all the patches we can recover the haze-free image by applying the imaging model (equation 3.2).

In the following subsections we describe each of the steps. Since steps 1, 3, and 5 remain almost the same, we describe them very briefly.

3.1.1 Color line and patch plane estimation

We want to estimate the color line from the patches. So, we first divide the image into patches of size $\omega \times \omega$ with 50% overlap. Then on the points in RGB space due to pixels of each patch we apply RANSAC to fit a line. Here also we compute the normal ($\hat{\mathbf{n}}$) (figure 3.2) to the patch plane (the plane containing this color line and the origin). We consider the equation of the line to be in the following form

$$\mathbf{L} = \rho\mathbf{D} + \mathbf{P}_0, \tag{3.4}$$

and estimate its parameters and the normal to the patch plane as follows.

$$\mathbf{P}_0 = \mathbf{I}_1, \tag{3.5}$$

$$\mathbf{D} = \frac{\mathbf{I}_2 - \mathbf{I}_1}{||\mathbf{I}_2 - \mathbf{I}_1||}, \tag{3.6}$$

$$\hat{\mathbf{n}} = \frac{\mathbf{I}_1 \times \mathbf{I}_2}{||\mathbf{I}_1 \times \mathbf{I}_2||}. \tag{3.7}$$

Here $\mathbf{I}_1$ and $\mathbf{I}_2$ are two points on the fitted line as estimated by RANSAC. Now the fitted line may be wrong or may not follow the properties we assume for the color line. So, the estimates need to be validated. For validation we do the following.

- The fitted line should have a high number ($> \theta_r\omega^2$) of inliers. Otherwise the fit is potentially wrong.
- The $\mathbf{D}$ denotes a color in the RGB space. So, all of the components of $\mathbf{D}$ should be positive.
- As per the assumption of color line, a patch should not contain more than one object. Therefore, sum pf gradient magnitude over the patch should be low ($< \theta_g$).
- If the fitted line is close ($< d_0$) to the origin then the effect of haze in this patch is quite less. So, the normal computed from this patch may not be reliable and, hence, should not be used for computing the $\hat{\mathbf{A}}$.
- If the colors don't spread out well in the RGB space, the estimated line becomes sensitive to noise. So to avoid this, the standard deviation of the inlier points should be high ($> \theta_s$) in the direction of $\mathbf{D}$.

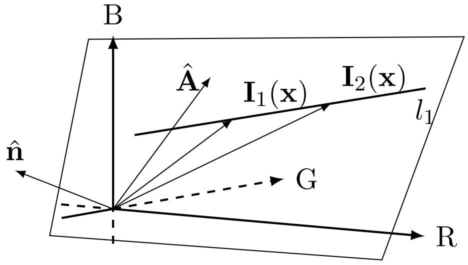


Figure 3.2: The plane containing the color line and origin will also contain the $I(\mathbf{x})$'s and $\hat{A}$.

3.1.2 Estimation of A's

To compute the airlight component we require, apart from the color line, the $\hat{\mathbf{A}}$ at each patch. We know that the color line gets shifted in the direction of $\hat{\mathbf{A}}$ by the amount given by the airlight component. So, the plane containing the fitted color line and the origin also contains $\hat{\mathbf{A}}$. If we have two patches such that they are affected by the same $\hat{\mathbf{A}}$ and their patch planes are not parallel, then this $\hat{\mathbf{A}}$ lies in the intersection of these two patch planes. But under our relaxed model each pixel can have a different $\hat{\mathbf{A}}$. On the other hand, we have assumed $\hat{\mathbf{A}}$ to be constant in a patch. So, these two assumptions together imply that $\hat{\mathbf{A}}$ may differ across patches, but not within a patch. Therefore, we can't just take two arbitrary patches and intersect their patch planes to get an $\hat{\mathbf{A}}$. The patches may not be affected by the same $\hat{\mathbf{A}}$. But we don't know the $\hat{\mathbf{A}}$'s or the patches they affect; we are trying to estimate them. So, the only way out is estimating both the patch plane and corresponding $\hat{\mathbf{A}}$ at the same time.

We know that a $\hat{\mathbf{A}}$ lies on the intersection of the patch planes corresponding to the patches it affects. So we can say that this $\hat{\mathbf{A}}$ is perpendicular to the normals of those patch planes. If we consider the normals as positional vectors in the RGB space, we get some points in the RGB space corresponding to the normals. Since we are trying to find a vector that is perpendicular to the normals, this can be achieved by a fitting a plane to those points and finding the normal of the fitted plane (figure 3.3). But we don't know the number of distinct illuminants in the scene and therefore the number of planes to fit. So, basically we want to fit multiple, but unknown number of planes. For this we utilize a Hough Transform based technique, because of its simplicity. Here each normal of the patch plane (now a point in the RGB space) votes for the plane it can reside in. The planes with high number of votes denote a good fit of the plane. Let's say the plane we are trying to fit to the normal of the patch planes is given by the following equation.

$$n_x a + n_y b + n_z c = 0. \tag{3.8}$$

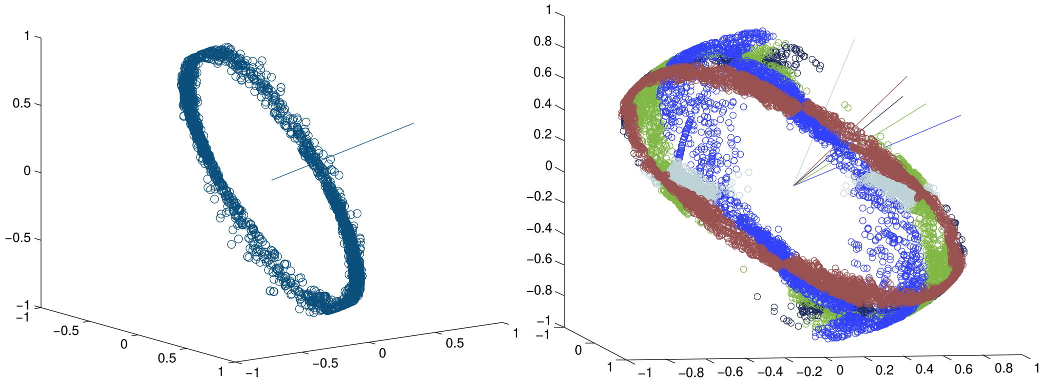


Figure 3.3: Normals obtained from image patches plotted as points in RGB space (colored circles) and their associated $\hat{A}$. Each color denotes a group of $\hat{n}$'s and the corresponding $\hat{A}$ is also colored the same. Left figure denotes the case when number of illuminants is only one. In the right figure the number of illuminants is more than one. So, we get many groups of the normals and their associated $\hat{\mathbf{A}}$.

Here $[n_x, n_y, n_z]^T = \hat{\mathbf{n}}$ and $[a, b, c]^T$ denote the normal to this plane (equation 3.8). Therefore this $[a, b, c]^T$ gives our $\hat{\mathbf{A}}$. Using different values of a, b and c, we can obtain any point, therefore a vector, in the RGB space. But we are interested in only the direction of the environmental illumination, not its magnitude. This information can be utilized to reduce the number of parameters of the plane we are trying to fit (equation 3.8). In 3D polar coordinate system a point is denoted by (r, θ, ϕ), where r denote the distance of the point from the origin and θ and ϕ denote the angles from the reference directions. Since we are trying to find a unit vector in the RGB space to get $\hat{\mathbf{A}}$, we can say in polar coordinate system we are trying to find points of the following form $(1, \theta, \phi)$ with θ and ϕ ranging from 0° to 90°. In the Cartesian coordinate its equivalent points are of the form $(\cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi)$. So, instead we fit a plane whose equation has the following form.

$$n_x \cos\theta \sin\phi + n_y \sin\theta \sin\phi + n_z \cos\phi = 0. \tag{3.9}$$

The fitting is done using the technique of Hough Transform. Each of the $\hat{\mathbf{n}}$ cast a vote for the combination of θ and ϕ such that equation 3.9 gets satisfied. In the computed Hough space a cell with high count denotes a good fit of the plane. We know that number of distinct $\hat{\mathbf{A}}$'s in the image can be more than one. So, taking only the cell with highest count won't suffice. We need to consider all good fits. But considering neighboring cells with high count would generate spurious $\hat{\mathbf{A}}$'s having similar values. So, in the computed Hough space we find local maximas that have count above a threshold (t_H) to get prospective $\hat{\mathbf{A}}$'s. To make this computation more robust we discard some patches based on its dark channel value [28], because dark channel value gives a rough estimate of the amount of haze present in a patch. The dark channel value of a patch Ω_i is computed as

$$D_H(\Omega_i) = \min_{\mathbf{x}\in\Omega_i}\Big(\min_{c\in\{R,G,B\}} I_c(\mathbf{x})\Big). \tag{3.10}$$

A patch (Ω_i) is kept if the following condition is satisfied

$$D_H(\Omega_i) > \theta_D \times \max_{\Omega_j} D_H(\Omega_j), \tag{3.11}$$

where $0 < \theta_D < 1$. From the obtained $\hat{\mathbf{A}}$'s we get the patches affected by it using the information of the patches that voted for the selected $\hat{\mathbf{A}}$. Now two problems arise if we use only the patches that voted for the selected maximas (*i.e.* $\hat{\mathbf{A}}$'s). First, one patch may vote for more than one of the selected $\hat{\mathbf{A}}$'s. Second, a patch may not vote for any of the selected $\hat{\mathbf{A}}$'s. So instead each patch is assigned to one of the $\hat{\mathbf{A}}$'s based on its cosine distance ($|\cos\psi|$ to be precise) from the corresponding patch plane's normal ($\hat{\mathbf{n}}$). We associate a patch to that $\hat{\mathbf{A}}$ with which the distance is minimum. Then we can group patches based on its associated $\hat{\mathbf{A}}$. We update $\hat{\mathbf{A}}$ of each group by computing the intersection of the patch planes of the patches in that group. Similar to our previous method, the eigenvector corresponding to the minimum eigenvalue of $\sum_i \hat{\mathbf{n}}_i\hat{\mathbf{n}}_i^T$ is used as the solution to the intersection. Now it may happen that for some groups the computed eigenvector has some of its components negative. But being a vector in the RGB space, we know all the components of $\hat{\mathbf{A}}$ should be positive. So, the eigenvectors with negative component are discarded and the normals belonging to the group of the discarded eigenvector, is assigned to one of the remaining $\hat{\mathbf{A}}$ based on the angular distance as done previously. Then we update the $\hat{\mathbf{A}}$ of the existing groups. This process is repeated till all the computed $\hat{\mathbf{A}}$'s is a valid RGB vector.

3.1.3 Estimating airlight component ($a(\mathbf{x})$)

In the previous steps we have computed from the patches their associated color line and $\hat{\mathbf{A}}$. We can now compute the airlight component by finding the amount of shift of the color line from the origin in the direction of $\hat{\mathbf{A}}$. Similar to our previous method, this is obtained by minimizing the following error.

$$E_l(\rho, \delta) = \|\mathbf{P}_0 + \rho\mathbf{D} - \delta\hat{\mathbf{A}}\|^2. \tag{3.12}$$

That is we try to shift the line ($\mathbf{P}_0 + \rho\mathbf{D}$) in the direction of $-\hat{\mathbf{A}}$, so that the line reaches as close to the origin as possible. Here δ provides the amount by which the color line needs to be shifted. Therefore we get the airlight component. In some specific situations this minimization may not give the desired value. So the estimated δ needs to be validated. Above mentioned situations and corresponding tests are briefly discussed below.

- If the color of the patch is similar to the environmental illumination, this shift computation fails. In general as the angle between the $\hat{\mathbf{A}}$ and the color line reduces, this shift computation becomes more sensitive to noise. So, we use the estimated shift in subsequent computations if the angle between the fitted line and the $\hat{\mathbf{A}}$ is sufficiently large ($> \theta_E$).

- The minimized value of equation 3.12 denotes how far the line is from the origin after being shifted in the direction of $-\hat{\mathbf{A}}$. If this value is large that means the line is far from the origin even after being shifted. This estimate is likely to be wrong. So, we accept the estimate of shift as correct, if the value of equation 3.12 after minimization is less than E_θ.

- We know that the estimated airlight component ($a(\mathbf{x})$) should have a value between 0 and $\sqrt{3}$. Naturally, if the estimated shift is beyond this range, then it is erroneous. But in practice the upper bound of $\sqrt{3}$ allows overestimation of $a(\mathbf{x})$ in some patches. So, the smallest intensity that is present in the patch is used as the upper limit.

3.1.4 Aggregation and Interpolation of estimated A's and $a(\mathbf{x})$

In the steps described till now we have worked on patches extracted from the image with 50% overlap. So, quite naturally for many pixels more than one shift value and $\hat{\mathbf{A}}$ are computed. So they need to be aggregated. We retain the maximum value of the shift and the corresponding $\hat{\mathbf{A}}$ as the aggregated data for a pixel. However, during estimation the estimated values are validated, and as a result, some estimates are discarded which are potentially incorrect. So, there are some pixels without any estimate of $\hat{\mathbf{A}}$ and $a(\mathbf{x})$. Since these values are required at every pixel to dehaze the input image, we need to interpolate the values at those pixels before applying the dehazing transformation. Here we need to interpolate bot the quantities: $\hat{\mathbf{A}}$ and $a(\mathbf{x})$, or sometimes $a(\mathbf{x})$ alone. Since, each $\hat{\mathbf{A}}$ is 3×1 vector, it may not be interpolated it directly. To each of the $\hat{\mathbf{A}}$'s a label is assigned and their influence at all the pixels is computed. The influence of each label is obtained by minimizing the following function.

$$E_A(F) = (F - P)^T (F - P) + \frac{\lambda}{2} F^T L F. \tag{3.13}$$

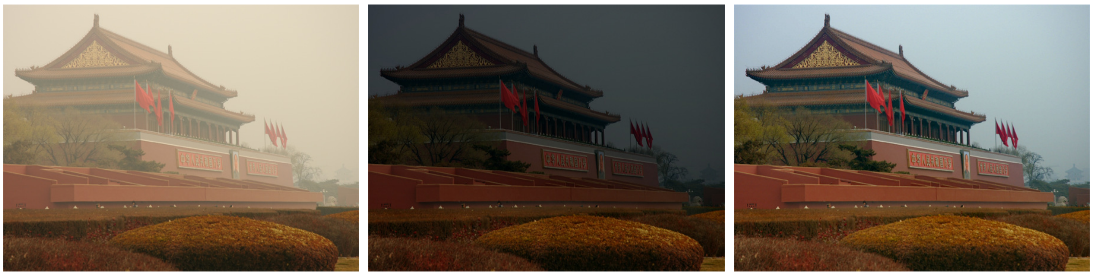


Figure 3.4: (From left to right) Input image, its airlight removed image and corresponding enhanced image.

where F is a matrix of size $n \times k$, and n k denote the total number of pixels in the image and the number of obtained $\hat{\mathbf{A}}$'s, respectively. $F(i, j)$ denotes the influence of j-th $\hat{\mathbf{A}}$ on i-th pixel. P is also a $n \times k$ matrix with $P(i, j) = 1$ if j-th $\hat{\mathbf{A}}$ is assigned to i-th pixel during aggregation; otherwise it is 0. The scalar λ controls the smoothness of the influence being compute here. L is the laplacian matrix of the graph constructed from the given image considering each pixel as a vertex and $1/||\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})||^2$ as the weight of the edge between pixel $\mathbf{x}$ and pixel $\mathbf{y}$. 4-connected neighborhood is considered here. The final interpolated $\hat{\mathbf{A}}(\mathbf{x})$ is a normalized weighted sum of the $\hat{\mathbf{A}}$'s where the weights are the influences obtained by minimizing equation 3.13.

The interpolation of $a(\mathbf{x})$ is done similar to our previous method by minimizing the following function.

$$\Psi(\mathbf{a}) = (\mathbf{a} - \tilde{\mathbf{a}})^T \boldsymbol{\Sigma} (\mathbf{a} - \tilde{\mathbf{a}}) + \alpha \mathbf{a}^T \mathbf{L}_g \mathbf{a} + \beta \mathbf{b}^T \mathbf{a}. \tag{3.14}$$

where $\tilde{\mathbf{a}}$ is the estimated airlight component after aggregation and $\mathbf{a}$ is its interpolated value, both in the vector form $(n \times 1)$. $\tilde{\mathbf{a}}$ is zero at the positions where the estimate is discarded. Σ is a diagonal matrix with its diagonal entries containing the error variance of $a(\mathbf{x})$ where it is estimated and 0 at other places. L_g is a laplacian matrix constructed similarly as before, but with a larger neighborhood. Each element of $\mathbf{b}$ is $1/||\mathbf{I}(\mathbf{x})||$. α and β are scalars that controls the importance of the corresponding terms.

3.1.5 Haze free image recovery

By following the steps described till now, we have obtained $\hat{\mathbf{A}}$'s and $a(\mathbf{x})$ at each pixel. So, we can compute the airlight of an image $(a(\mathbf{x})\hat{A}(\mathbf{x}))$. Subtracting this from the observed image we get airlight removed image.

$$\mathbf{I}_{\bar{a}}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) - a(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x}). \tag{3.15}$$

This removes the color cast introduced by the haze, but also reduces the overall intensity of the image (figure 3.4). For restoring the overall brightness of the image we need to divide $\mathbf{I}_{\bar{a}}(\mathbf{x})$ by $t(\mathbf{x})$. But we can't obtain $t(\mathbf{x})$ from the computed airlight. So, we enhance the contrast of the airlight removed image depending on the proportion of original intensity removed from it when the airlight is removed. Let's say the recovered image is $\mathbf{J}'(\mathbf{x})$, and each color channel of it is computed as follows.

$$J'_c(\mathbf{x}) = \frac{\mathbf{I}_{\bar{a}}(\mathbf{x})}{1 - Y(a(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x}))}, \text{ where, } c \in \{R, G, B\} \tag{3.16}$$

$$Y(a(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x})) = 0.2989a(\mathbf{x})\hat{A}_R(\mathbf{x}) + 0.5870a(\mathbf{x})\hat{A}_G(\mathbf{x}) + 0.1140a(\mathbf{x})\hat{A}_B(\mathbf{x}). \tag{3.17}$$

Though this transformation works well for some images, its good performance can't be guaranteed. Sometimes the dehazed image turns out dark, and further enhancement becomes necessary.

3.2 Experimental Settings

All the results that we have reported here have been generated using a MATLAB implementation of the proposed method. The parameters have been kept at their default values unless otherwise stated to generate the results. Their default values are reported in Table 3.1. For fitting the color line, we have used the code of RANSAC as provided by Peter Kovesi [33] while taking 0.02 as the inlier threshold. Patch size of 8×8 has been used throughout the experiment. θ_D has been varied between 0.1 and 0.45. The threshold used in the Hough space (t_H) is taken to be 30% of the maximum Hough space response. The step value parameter in the Hough space (h_s) is taken to be 3°. For α and β, small values are used depending on the input image. Their typical values are 2×10^{-4} and 1×10^{-5} respectively.

Table 3.1: Default parameter values

θ_r	θ_g	d_0	θ_E	E_θ	θ_s	λ
0.4	0.06	0.0005	15°	0.05	0.006	1

3.3 Results

We have run our method on a variety of day-time and night-time images, since our method works for both types of images. The image set includes benchmark images used for testing day-time and night-time dehazing methods. To evaluate the performance of day-time dehazing we have used both synthetically generated hazy images with ground-truth as well as real world hazy images. However, in case of night-time images, we have used only real world hazy images for evaluating the performance of night-time dehazing. We have compared our method with two day-time dehazing method [22, 28] and also two night-time dehazing methods [67, 36]. The results of Fattal [22] have been obtained from the author's website[1] while the results of Zhang et al. [67] and that of Li et al. [36] are generated using the code provided by the respective authors keeping the parameters at their default values. For generating the result of He et al. [28], we have used our own implementation. To quantitatively measure the quality of results obtained for synthetically generated hazy images, we have used three full reference metrics: PSNR, SSIM [64] and ΔE_{00} [56]. The real world images have been evaluated only qualitatively.

3.3.1 Daytime Images

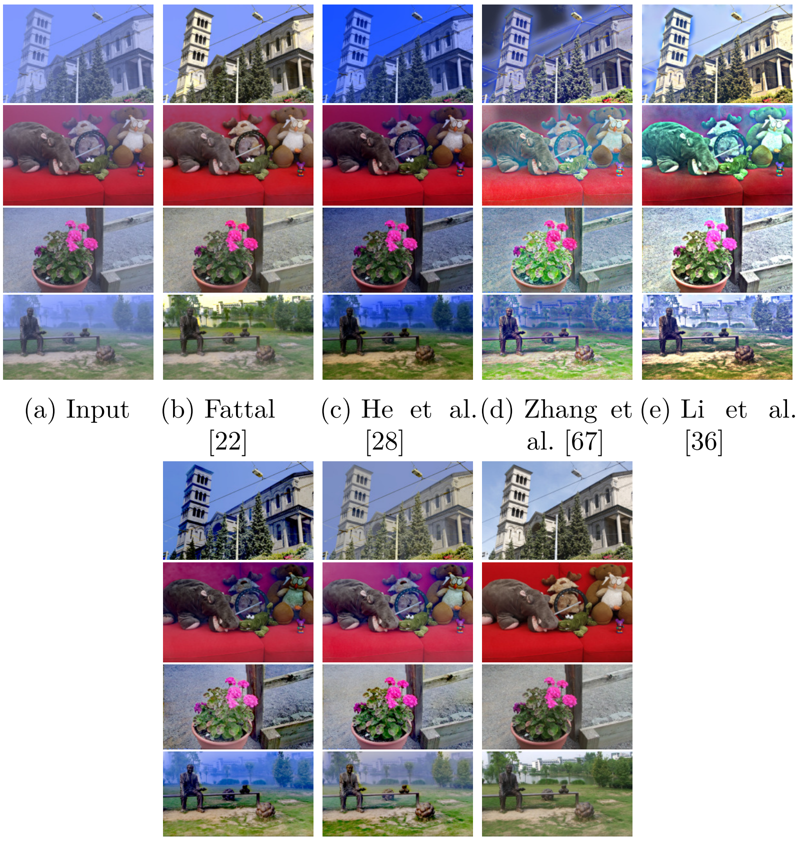
We have used the benchmark hazy images provided by Fattal [22] to evaluate the dehazing performance on day-time images. This benchmark image set contains both synthetic and real world hazy images. For the synthetic images we report both the quantitative and qualitative results. The scores (PSNR, SSIM, ΔE_{00}) obtained on the synthetic images is reported in Table 3.2. Figure 3.5 shows the qualitative results on a few of the synthetic images. The scores in Table 3.2 shows similar trend that we have seen in the previous chapter. The results of Fattal [22] are always the best and He et al. [28] performs comparatively well in a few cases only. Our current method has improved considerably, specially in terms of ΔE_{00}. This implies the estimate of the color of environmental illumination ($\hat{\mathbf{A}}$) is much more accurate. This can also be seen in figure 3.5: a bluish tint is present in the outputs of the method of Chapter 2, but it has been successfully neutralized by our current method. We have also included results of night-time dehazing methods [67, 36] on these day-time images to show that they perform poorly for such images. This is clearly seen in figure 3.5, where the colors are saturated and the enhancement has introduced noise and artifacts in the image. These attributes are not present in the results of any day-time dehazing methods.

Now if we turn our attention to the day-time real-world images, we see that the methods perform similarly like they have performed on synthetic images (figure 3.6). The results of Fattal [22] has the highest contrast but have a little bias towards yellow tint in the dehazed images. The method of He et al. [28] has cleared the haze to some extent, but it could not clear the color cast present in the image. Both of the night-time

[1] `http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/`

Table 3.2: Quantitative Comparison on the images of Fattal's dataset. The best results are bold and the second best results are underlined. Note that in Fattal's method only $t(\mathbf{x})$ is computed and $\mathbf{A}$ is manually provided.

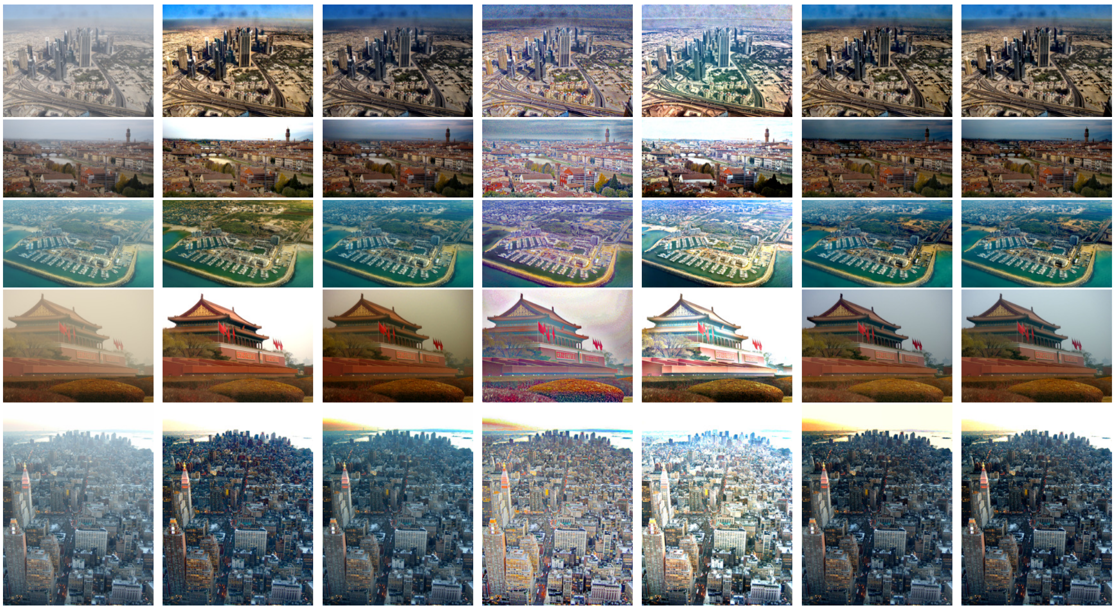
Image	Fattal [22]			He et al. [28]			Our (ch2)			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	**21.43**	**0.96**	**6.34**	11.16	0.78	28.74	14.84	0.76	15.54	16.99	0.9	11.59
couch	**20.8**	**0.9**	**6.71**	18.4	0.86	13.89	16.08	0.78	16.11	16.23	0.78	15.84
dolls	**21.29**	**0.77**	**6.1**	19.73	0.85	10.65	15.5	0.77	17.61	13.64	0.8	18.14
flower1	**30.01**	**0.98**	**3.91**	14.1	0.88	23.26	16.8	0.89	13.86	19.27	0.89	11.19
flower2	**31.94**	**0.99**	**2.92**	14.37	0.86	20.94	17.02	0.85	13.92	21.93	0.9	7.79
lawn1	**24.49**	**0.97**	**6.65**	13.84	0.8	22.38	14.71	0.82	20.01	17.14	0.84	13.78
lawn2	**24.94**	**0.97**	**6.46**	11.2	0.74	29.32	14.52	0.83	20.23	15.79	0.8	15.9
mansion	**26.96**	**0.97**	**4.04**	17.45	0.87	19.35	17.49	0.85	16.62	19.65	0.84	9.2
moebius	**19.01**	**0.9**	**10.61**	12.66	0.78	26.7	15.85	0.86	17.46	18.72	0.86	12.59
raindeer	**26.22**	**0.94**	**4.1**	18.12	0.83	14.22	13.59	0.73	22.54	15.47	0.74	17.23
road1	**25.74**	**0.96**	**5.24**	12.95	0.8	26.11	14.16	0.84	20.78	15.91	0.78	14.45
road2	**23.6**	**0.96**	**7.11**	15.84	0.84	22.13	15.88	0.86	20.13	15.02	0.82	15.74
Average	**24.7**	**0.94**	**5.85**	14.98	0.82	21.47	15.54	0.82	17.9	17.15	0.83	13.62



(a) Input (b) Fattal [22] (c) He et al. [28] (d) Zhang et al. [67] (e) Li et al. [36]

(f) Our (ch2) (g) Our (h) Ground Truth

Figure 3.5: Visual comparison of the results on four synthetic images: *church*, *couch*, *flower2*, and *lawn1*.



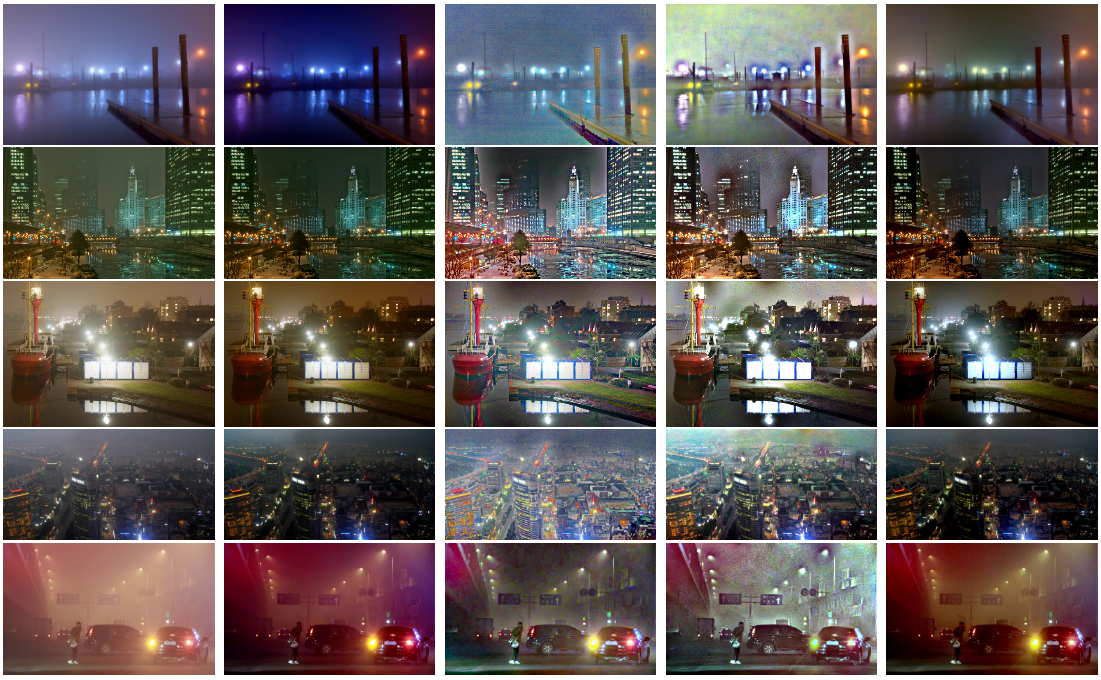
(a) Input (b) Fattal [22] (c) He et al. [28] (d) Zhang et al. [67] (e) Li et al. [36] (f) Our (ch2) (g) Our

Figure 3.6: Visual comparison of results on *dubai*, *florence*, *herzeliya*, *tiananmen*, and *ny12* image.

dehazing methods [67, 36] perform poorly. The images are over enhanced and the colors are saturated. The current proposed method and the method proposed in the previous chapter perform almost similarly as the present method is a general case of the previous method, and both of them are equally competent to handle day-time images.

3.3.2 Night-time images

We have only qualitatively evaluated our results on night-time hazy images, since we don't have the ground truth of the night-time hazy images. Second, it is very difficult and unrealistic too to synthesize night-time hazy images as that require set many parameters heuristically. Here we present the results of our method for dehazing night-time images. Our results are compared with that of night-time dehazing methods proposed by Zhang et al. [67] and Li et al. [36]. We have also included the results of the day-time dehazing method of He et al. [28] for comparison. This method, as it is exclusively for day-time hazy images, does not work well for night-time hazy images. This can clearly be seen in figure 3.7. The method of He et al. [28] is able to clear only the overall appearance of haze. It cannot clear the local color cast due to artificial lighting. But both of the night-time dehazing method is able to clear those features. In the results of Zhang et al. [67] we see that the method is trying to enhance the brightness of the whole image,



(a) Input (b) He et al. [28] (c) Zhang et al. [67] (d) Li et al. [36] (e) Our

Figure 3.7: Visual comparison of results on night-time images.

which makes the results look unnatural. The method proposed by Li et al. [36] controls this enhancement a little better and successfully removes the glows around the lights. However, it is not removed by our method as our model does not focus on the glow around the artificial lights. In our method, we only remove the airlight and enhance the image based on contrast intensification. Hence, the over enhancement does not occur.

3.4 Summary

We have proposed in this chapter a unified dehazing method that works for both night-time and day-time images. This is achieved by using a relaxed haze imaging model (equation 3.1) where traditional assumption of constant atmospheric light is relaxed to a spatially variant one. We determine possible directions of the atmospheric light vectors using color line and Hough transform. These directions are used to calculate airlight component in each patch of the image. We have computed the airlight component ($a(\mathbf{x}) = (1 - t(\mathbf{x}))m(\mathbf{x})$) as a whole, and explicit computation of $t(\mathbf{x})$ is not attempted. For this reason objects with low intensity and color similar to airlight becomes dark after dehazing, and we employ contrast enhancement in the last phase of our method to overcome this problm. This contrast enhancement procedure idoes not guarantee to work satisfactorily in all images. Apart from that, by discarding some patches and the discretizing Hough space, some of the $\hat{\mathbf{A}}$ cannot be computed. Moreover, the computed airlight at some pixels is not accurate one due to the interpolation. So, in this case the input image may not be dehazed properly. Another major problem is that we have used around 10 parameters in the method as thresholds and weights. Their default values work well in most of the images, but in some images they perform poorly. Manipulation of the parameters is not straight forward. Decreasing the thresholds allow estimates to be computed in more patches, consequently this may potentially allow incorrect estimates of the airlight component. This is an inherent problem of the color line based method, so exploration of other haze related features becomes necessary. So, in the next chapter we explore the use of Convolutional Neural Networks in this regard.

Chapter 4

Supervised estimation of transmittance and environmental illumination using CNN

Convolutional neural network (CNN) is a class of neural network that uses convolution operation for feature extraction. Convolution with particular kernels (*e.g.* Gaussian, Laplacian, Sobel) has traditionally been used for extracting features from images or signals in general. Instead of using predefined kernels, in a CNN the network learns the convolution kernels from the data so that the network outputs the desired target values. Additionally, when these convolution operations are stacked together, the network effectively learns a hierarchy of features. These two characteristics have proven to be very effective in various computer vision and image processing tasks like classification [34, 29], segmentation [38], object detection [51] and many more. This has motivated its use in the domain of image dehazing. However, the main hurdle of using a CNN in the problem of dehazing is the lack of training data. It is quite hard, if not impossible, to obtain the corresponding haze-free image (precisely pixel-level registered) of an hazy image. So, the common route is to use images with known depth maps and adding haze synthetically to those. So we have both haze-free image (original) and corresponding hazy image (synthesize). This strategy has been applied to estimate scene transmittance from images. But not much attention has been given to estimate the environmental illumination barring a few methods such as [49, 59]. Note that the quality of dehazed image depends on the accuracy of estimated environmental illumination [59]. Due to this interdependence of transmittance and environmental illumination, in this chapter we propose to estimate them jointly. For that we work with the basic version of the atmospheric scattering model, assuming space-invariant environmental illumination, that is

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})), \tag{4.1}$$

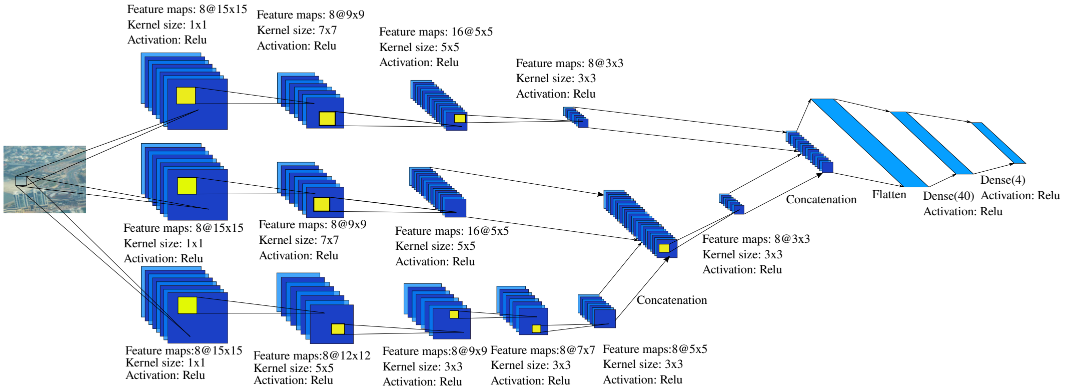


Figure 4.1: The architecture of our joint t-$\mathbf{A}$ estimator network

$$t(\mathbf{x}) = e^{-\beta d(\mathbf{x})}. \tag{4.2}$$

Where $\mathbf{I}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ are the observed hazy image and the clear scene radiance, $\mathbf{A}$ is the global environmental illumination, and $t(\mathbf{x})$ is the scene transmittance. Assuming that the haze is homogeneous, we can express $t(\mathbf{x})$ by equation 4.2, where β is the scattering coefficient and $d(\mathbf{x})$ is the scene depth. Since recovering the scene radiance from the input hazy image is ill-posed, we make a simplifying assumption that has also been utilized in the color line based dehazing. We assume t to be constant within a patch, as depth variation becomes negligible if the patch is sufficiently small. This does not hold true for patches with depth discontinuities and using those patches results in wrong estimate. So, these patches are not considered when estimating the haze parameters. For each eligible image patch, we try to estimate both t and A using the following equation.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t + (1 - t)\mathbf{A}. \tag{4.3}$$

Moreover, if a patch is smooth then estimation of its transmittance and environmental illumination becomes quite hard. Because in a smooth patch, where all pixel values are same, it becomes almost impossible to discriminate whether the pixel values are due to haze or object radiance. So, such patches are also not considered. Now, to estimate the haze parameters (t and $\mathbf{A}$) from patches, we learn the mapping from $\mathbf{I}(\mathbf{x})$'s of a patch to a t and $\mathbf{A}$ using a CNN. So, given a patch this CNN estimates both t and A. Although we obtain from the CNN a specific $\mathbf{A}$ for each patch, we use an aggregated global $\mathbf{A}$ while dehazing the input image.

4.1 Joint *t*-A Estimator Network

We are trying to estimate both t and $\mathbf{A}$ from patches using a CNN. So, we need to extract both coarse and fine scale features from a patch, because the transmittance depends on fine scale features while $\mathbf{A}$ is comparatively a more coarse quantity. We estimate them jointly maintaining their dependency. This suggest the architecture of our model (figure 4.1) for computing features in three different paths. The extracted features are then concatenated and fed to the dense layers for the joint estimation of t and $\mathbf{A}$. The convolution layers on the top two paths have bigger convolution kernels while the bottom one uses small kernels. The output feature maps of the bottom two paths are concatenated and combined using convolution. The output of the top most path is then concatenated with the combined output of the two lower paths and flattened. This flattened feature map is further processed using a dense layer of neurons before producing the output through 4 neurons (1 for t and 3 for $\mathbf{A}$). ReLU is used as the activation function throughout the network.

4.2 Dehazing Method

The proposed method consists the following steps in order to dehaze a given image.

1. **Estimation of** t **and A from patches:** In the very first step the input image is divided into overlapping patches. Then each patch is fed to the estimator network to estimate the corresponding haze parameters.

2. **Aggregation and interpolation of estimate:** Since overlapping patches are used, a pixel may get multiple estimates due to its belongingness to various patches. These estimates are aggregated. On the other hand, some patches are discarded during the estimation, some pixels may not get any estimate. The estimate at these pixels is interpolated based on estimates at neighbouring pixels.

3. **Recovering the scene radiance:** After estimating the necessary haze parameters at all pixels, we just invert the haze model to obtain the dehazed image.

Now, these steps are discussed in details in the following subsections.

4.2.1 Estimation of t and A from Patches

Input image is divided into 15 × 15 patches with a stride of 5 pixels both horizontally and vertically. As mentioned earlier, among these patches, we process only the patches that does not contain any edge and has intensity variance more than a threshold. All the patches that are selected are fed to our joint estimator network to obtain t and $\mathbf{A}$ for the patches. The t obtained for a patch is taken to be transmittance for all the pixels in that patch.

4.2.2 Aggregation and Interpolation of estimate

In the previous step for estimating t and $\mathbf{A}$, we have considered overlapping patches. So, multiple values of t and $\mathbf{A}$ may be assigned to a pixel if it belongs to multiple patches. For multiple values of transmittance t, we aggregate these to a single value per pixel by taking their average. For aggregating the environmental illumination, we take average of all the $\mathbf{A}$'s obtained from different patches to get the global $\mathbf{A}$.

On the other hand, since we discard many patches, even after aggregation it is quite likely that at some pixels t is not available. Note that we require transmittance value at each pixel to dehaze an image. So, we interpolate the t values at those places to create complete transmittance map. Similar to the previous chapters this is done by minimizing the following function which is inspired from the method of Fattal [22].

$$\psi(t(\mathbf{x})) = \sum_{\mathbf{x}} s(\mathbf{x})(t(\mathbf{x}) - \tilde{t}(\mathbf{x}))^2 + \lambda \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \frac{t(\mathbf{x}) - t(\mathbf{y})^2}{||\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})||^2}. \tag{4.4}$$

Here $\tilde{t}(\mathbf{x})$ denotes existing estimate of scene transmittance. $t(\mathbf{x})$ is the interpolated transmittance that we are trying to obtain. $s(\mathbf{x})$ is either 1 or 0 depending on whether an estimate of transmittance exists at pixel $\mathbf{x}$ or not. $N(\mathbf{x})$ denotes the four-connected neighborhood of pixel $\mathbf{x}$. λ is a scalar controlling the smoothness of the interpolated transmittance. After the minimization we get a transmittance value at all the pixels.

4.2.3 Recovering the scene radiance

From the above mentioned steps, we obtain $t(\mathbf{x})$ at each pixel and $\mathbf{A}$ for the whole image. So, we can compute the scene radiance ($\mathbf{J}(\mathbf{x})$) corresponding to the input hazy image ($\mathbf{I}(\mathbf{x})$) as follows:

$$\mathbf{J}(\mathbf{x}) = \mathbf{A} + \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{\max\{0.1, t(\mathbf{x})\}} \tag{4.5}$$

In the denominator the value of t is bounded from below to avoid arbitrary large values in the output ($\mathbf{J}(\mathbf{x})$).

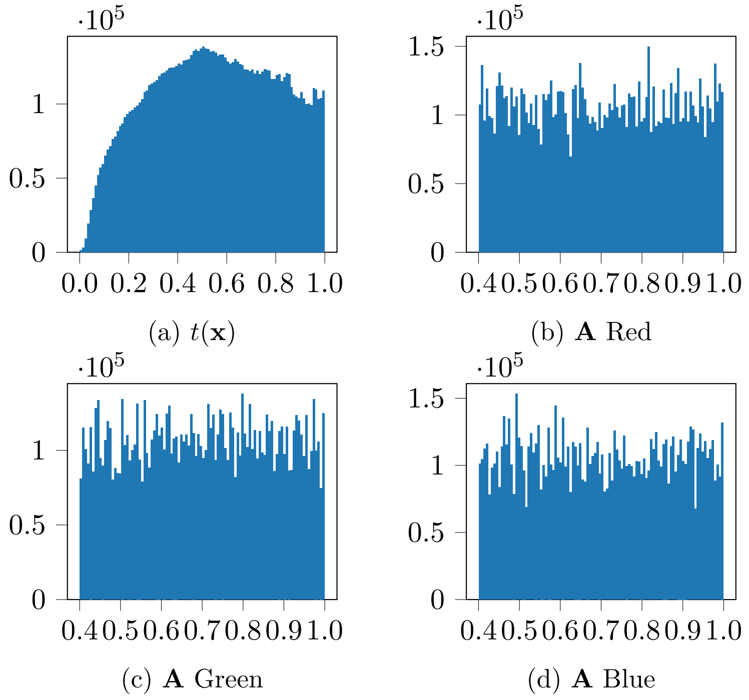
4.3 Experimental Details

4.3.1 Training Data Generation

We need hazy patches and their corresponding haze parameter values (t and $\mathbf{A}$) to train our joint estimator network. But acquiring that data from natural scenes is difficult in practice. So, this data is generated synthetically by adding haze to haze-free images. For that purpose we have used NYU Depth Dataset [58]. Hazy images are generated from the clean haze-free images by applying the atmospheric scattering model (equation 4.1 and equation 4.2) with different values of environmental illumination ($\mathbf{A}$) and scattering coefficient (β). The value of the transmittance t at pixel $\mathbf{x}$ depends both on the depth at $\mathbf{x}$ and as well as β (equation 4.2). Since the dataset we are using to generate the hazy images contains indoor images only, the variation of depth is not in the same order of magnitude as outdoor images. But we only need to estimate transmittance from patches, not the actual depth. So, we can compensate the lack of proper depth variation by varying β between 0.5 and 1. This ensures that in the training data we have transmittance values that encompasses the whole range (figure 4.2a). The values of $\mathbf{A}$ has been chosen in similar way so that the training data contains all possible values of $\mathbf{A}$ (figure 4.2b, figure 4.2c, figure 4.2d). After generating the hazy images, 15×15 patches are extracted out of them while taking a stride of 5 pixels. To ensure that the network gets only proper information as the training data, we discard a portion of the generated patches. As discussed earlier, estimating the parameters from the smooth patches is difficult. So, if the variance of the intensities of a patch lies below a threshold, that patch is discarded. In some of the depth maps, the depth information is not present in all the pixels. We discard patches with missing depth information. After these pre-processing we get around six hundred thousand patches as the training data.

4.3.2 Experimental Settings

The results that we report here are generated by keeping the parameters fixed to the mentioned values. Our joint t-$\mathbf{A}$ estimator network has been trained by minimizing the mean squared error between the estimated t and $\mathbf{A}$ and the corresponding ground truth. The network is trained using the Adadelta optimizer with a batch size of 1000. The network is trained for 90 epochs and it takes approximately 10 hours on a computer with a 2.4 GHz CPU.



(a) $t(\mathbf{x})$

(b) $\mathbf{A}$ Red

(c) $\mathbf{A}$ Green

(d) $\mathbf{A}$ Blue

Figure 4.2: Histogram of the transmittance values and each of the RGB channel of environmental illumination present in the training data.

4.4 Results

The method we propose here works with the assumption of a constant global $\mathbf{A}$. So, this method is not likely to work with night time images. For this reason we have applied this method on daytime images only. In this experiment we have used the synthetic hazy image dataset by Fattal [22] and also some real world hazy images. For the synthetic images we have used metrics like PSNR, SSIM, and ΔE_{00} to quantitatively evaluate the results. We have compared our results with that of the method by Fattal [22], He et al. [28] and Ren et al. [53] as well as with the method proposed in the previous chapter. Among these works, the method of Ren et al. [53] is based a CNN that estimates scene transmittance from the given image. The environmental illumination is computed separately from the estimated transmittance. The results of Fattal [22] are obtained from the author's website[1]. The results of He et al. [28] are generated using our own implementation of the method. For generating the results of the method of Ren et al. [53], we have executed the code provided by the authors' with its default setting.

[1] http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/

Table 4.1: Quantitative Comparison on the images of Fattal's dataset. High PSNR and SSIM indicates better results, while it is the opposite for ΔE_{00}. The best results are bold and the second best results are underlined. Note that in Fattal's method only $t(\mathbf{x})$ is computed and $\mathbf{A}$ is manually provided.

Image	Fattal [22]			He et al. [28]			Ren et al. [53]			ch3			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	**21.43**	**0.96**	**6.34**	11.16	0.78	28.74	14.17	0.87	20.25	16.99	$\underline{0.9}$	$\underline{11.59}$	$\underline{17.17}$	0.84	14.68
couch	**20.8**	**0.9**	**6.71**	18.4	0.86	13.89	17.99	$\underline{0.88}$	12.91	16.23	0.78	15.84	$\underline{19.96}$	0.79	$\underline{10}$
dolls	**21.29**	**0.77**	**6.1**	$\underline{19.73}$	0.85	$\underline{10.65}$	16.93	$\underline{0.86}$	12.37	13.64	0.8	18.14	12.55	0.82	16.6
flower1	**30.01**	**0.98**	**3.91**	14.1	0.88	23.26	9.08	0.43	24.64	$\underline{19.27}$	$\underline{0.89}$	$\underline{11.19}$	16.3	0.8	20.87
flower2	**31.94**	**0.99**	**2.92**	14.37	0.86	20.94	10.81	0.6	22.45	$\underline{21.93}$	$\underline{0.9}$	$\underline{7.79}$	18.08	0.82	16.43
lawn1	**24.49**	**0.97**	**6.65**	13.84	0.8	22.38	14.37	0.83	21	$\underline{17.14}$	$\underline{0.84}$	$\underline{13.78}$	14.26	$\underline{0.84}$	24.82
lawn2	**24.94**	**0.97**	**6.46**	11.2	0.74	29.32	13.29	0.77	22.27	$\underline{15.79}$	0.8	$\underline{15.9}$	14.64	$\underline{0.82}$	24.84
mansion	**26.96**	**0.97**	**4.04**	17.45	$\underline{0.87}$	19.35	17.69	0.89	17.52	$\underline{19.65}$	0.84	$\underline{9.2}$	19.64	0.82	14.5
moebius	**19.01**	**0.9**	**10.61**	12.66	0.78	26.7	16.36	$\underline{0.9}$	19.85	$\underline{18.72}$	0.86	$\underline{12.59}$	16.84	0.82	17
raindeer	**26.22**	**0.94**	**4.1**	$\underline{18.12}$	$\underline{0.83}$	$\underline{14.22}$	16.82	0.81	15.49	15.47	0.74	17.23	17.92	0.74	14.85
road1	**25.74**	**0.96**	**5.24**	12.95	0.8	26.11	14.11	0.84	22.22	15.91	0.78	$\underline{14.45}$	$\underline{16.83}$	$\underline{0.81}$	22.51
road2	**23.6**	**0.96**	**7.11**	15.84	0.84	22.13	$\underline{16.45}$	$\underline{0.88}$	20.17	15.02	0.82	$\underline{15.74}$	18.2	0.86	19.45
Average	**24.7**	**0.94**	**5.85**	14.98	0.82	21.47	14.84	0.8	19.26	$\underline{17.15}$	$\underline{0.83}$	$\underline{13.62}$	16.86	0.82	18.05

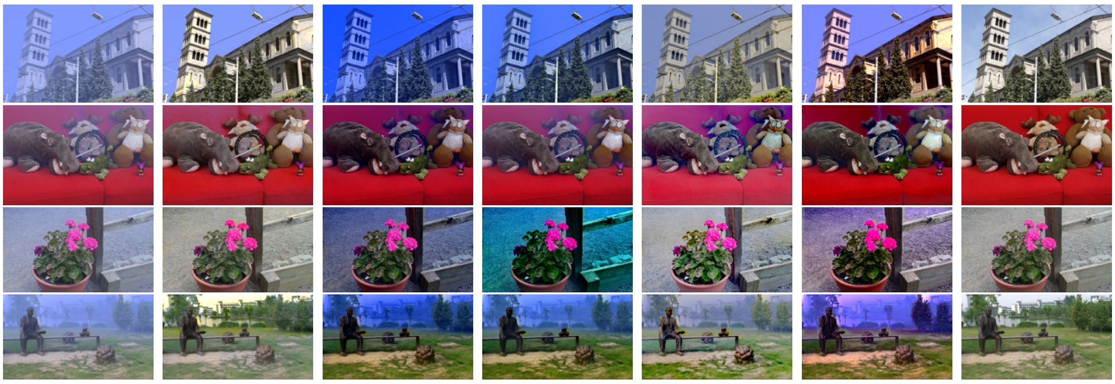
4.4.1 Quantitative Results

We have quantitatively evaluated the results obtained by the proposed method using the synthetic hazy image dataset of Fattal [22]. The results are reported in Table 4.1. From that it is seen the results of Fattal [22] is still at the top. The method of Ren et al. [53] performs a litter better than He et al. [28] in terms of SSIM and ΔE_{00} in many images. But its ΔE_{00} values are higher (a low value is better) in many images compared to both of our proposed methods. The table also shows this method performs comparably to our color line based method without the need of large number of user tunable parameters. This shows the potential of CNNs in this task. But it is observed that the value of ΔE_{00} varies quite a bit across images. This behavior is analyzed in Section 4.5.

4.4.2 Qualitative Results

For qualitative evaluation, the results on both synthetic and real world images are used. However, only a subset of results on synthetic images is shown here as the qualitative results. From the figures containing results of synthetic images (figure 4.3), it is seen that the results reflect what is already seen in the quantitative results (Table 4.1). The result of Fattal [22] is the cleanest. The method of Ren et al. [53] has performed a little better in removing haze from the images, but it could not neutralize the color cast introduced by the haze. Both of our proposed methods are able to clear the color cast, but the currently proposed method has performed a little poorly in this regard. In terms of removing the haze, the currently proposed method has performed comparably.

Among the results of real-world images figure 4.4, we see that the output of Fattal [22] and our method proposed in the previous chapter is the cleanest in most of the



(a) Input (b) Fattal [22] (c) He et al. [28] (d) Ren et al. [53] (e) Our (ch3) (f) Our (g) Ground Truth

Figure 4.3: Visual comparison of the results on four synthetic images: *church*, *couch*, *flower2*, and *lawn1*.

images. But the output obtained by the method of Fattal [22] is a little yellowish, and the outputs due to our previously proposed method are a bit dark. He et al. [28] and Ren et al. [53] has performed more of less similarly for all the images. The proposed method has performed well in some of the images (*e.g.* dubai, florence, and herzeliya). However, it fails to remove the haze from the others. This is mostly due to improper estimation of environmental illumination. Note that the proposed method works without requiring any post-processing step that are required by our previous method.

4.5 Discussion

In the result section it is seen that our proposed network performs well in terms of PSNR and SSIM but falls short in ΔE_{00}. That means the method has failed to restore the original colors of the image, although it is able to uncover the structures. According to the imaging model (equation 4.1), the color cast during haze is introduced by the environmental illumination. So, an incorrect estimate of this quantity results in color shift of the output image (figure 4.5). The proposed method averages all the **A**'s estimated from the patches to obtain the global **A**. If some of the **A**'s estimated from the patches is wrong, there is a possibility that the estimated global **A** would be wrong. To avoid this, we assign the estimated **A** for a patch to all of its pixels and average **A** at a pixel if it receives more than one estimate. That means we aggregate **A** in the same way as done for transmittance. From figure 4.5 it is seen that the estimated **A** is varying quite a bit within the image depending on the content of the patch. For example, the network has estimated **A** to be red near the flag and a little greenish near the tree. Since the

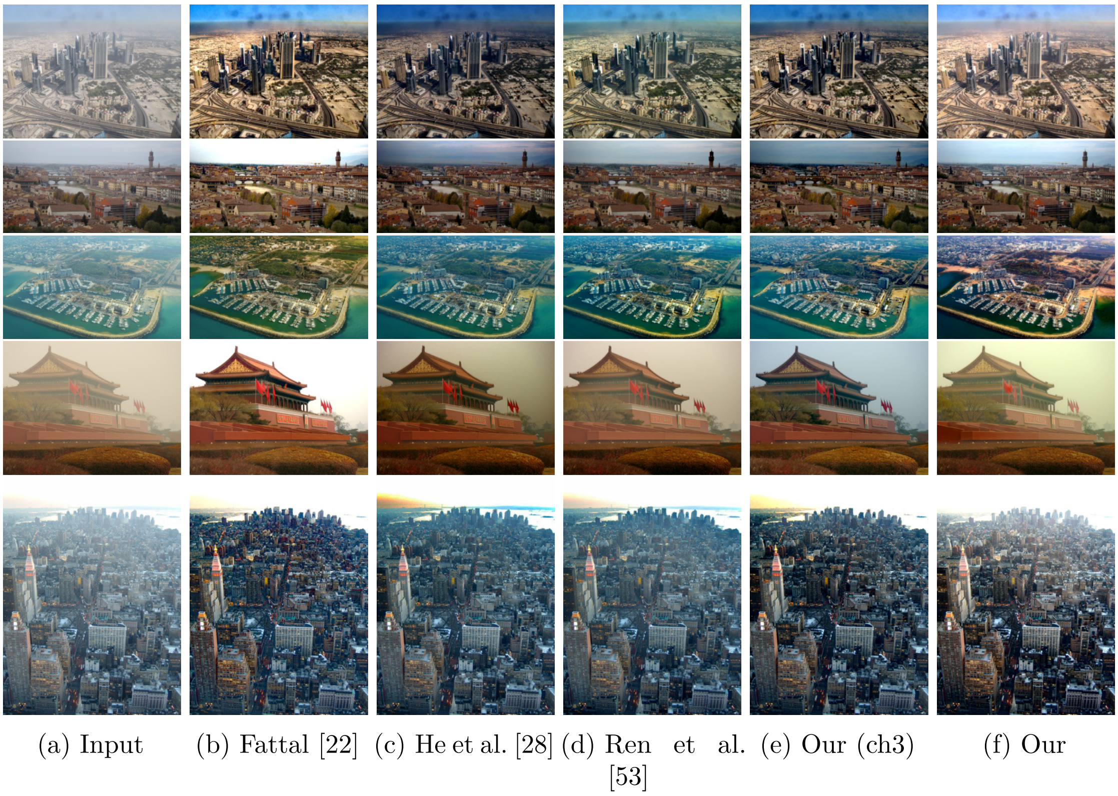
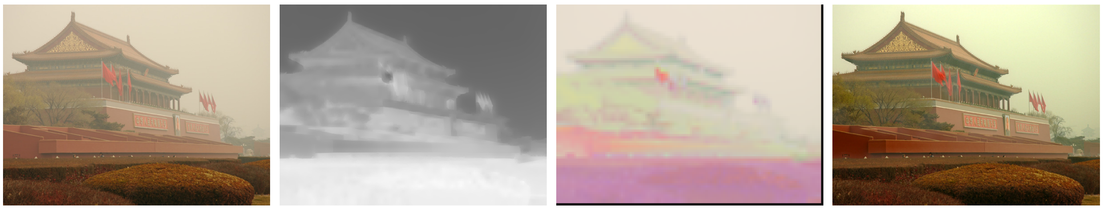

(a) Input (b) Fattal [22] (c) He et al. [28] (d) Ren et al. [53] (e) Our (ch3) (f) Our

Figure 4.4: Visual comparison of results on *dubai*, *florence*, *herzeliya*, *tiananmen*, and *ny12* image.



(a) Input (b) Transmittance (c) Patch aggregated **A** (d) Output

Figure 4.5: Input hazy image, estimated haze parameters and the output. The estimated **A**'s are aggregated and shown as image. It is observed that **A** is sensitive to patch content and at times taken as the average. Different parts of the image reports different **A**'s.

network is jointly estimating both t and $\mathbf{A}$, the wrong estimate of $\mathbf{A}$ has affected the estimate of t. As it can be seen that in the output the haze is not fully removed. Since $\mathbf{A}$ is more of a global quantity it should be estimated by looking at larger area of the image as far as possible.

4.6 Summary

Unlike previous chapters, here a method is proposed to estimate the scene transmittance and environmental illumination jointly from image patches using a CNN. The estimated values are then utilized to dehaze the given image. The results show the effectiveness of a CNN trained to estimate $t(\mathbf{x})$ and $\mathbf{A}$ while requiring only a few user tunable parameters to obtain obtain the final result. However, we note that trying to estimate the environmental illumination from small patches may produce wrong estimate at times and thereby affects the quality of the output. The network is also biased towards the average color of the patch as the estimated $\mathbf{A}$ if there is not much color variety in the patch. Working with bigger patches may solve this problem as the bigger patch is likely to contain diverse set of objects. But, in bigger patches the assumption of constant transmittance fails as the variation of depth may not remain insignificant. So, we need to estimate a spatially varying transmittance in this case.

We have used an indoor image dataset (NYU v2) to synthetically generate the training data. Indoor and outdoor images usually have different statistics. When using small patches this difference does not have much effect (as seen from the results), but when bigger patches are used difference becomes significant. In the next chapter we try to tackle these issues.

Chapter 5

Supervised estimation of transmittance and airlight using FCN

In the previous chapter, an attempt has been made to jointly estimate the scene transmittance and the environmental illumination. Although the method works well in many images, its performance varies with the goodness of the estimated of environmental illumination. Use of small patches is the main factor behind this, because in small patches it gets difficult to decide whether the colors in the patch are due to color of illumination or the color of the object present in it. So, working with bigger patches is inevitable. However, this only reduces the chances of confusion rather than completely eliminating them. But on the other hand when working with bigger patches, the assumption of constant transmittance within a patch gets violated. So, in this case estimating the transmittance for all the input pixels becomes necessary. In this type of problems that is where the size of the output is same as the input, Fully convolutional networks (FCN) [38] has shown promising results. For this reason in this chapter we propose to use a FCN based estimator network to estimate the haze parameters of a given patch. But instead of working with the model with constant environmental illumination, we work with the relaxed version of the imaging model.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A}(\mathbf{x}). \tag{5.1}$$

That is, in this model the environmental illumination can vary from pixel to pixel. So, given a hazy image if one is able to estimate both transmittance ($t(\mathbf{x})$) and environmental illumination ($\mathbf{A}(\mathbf{x})$), the dehazed image can be obtained using this model. But estimating $t(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$ independently can be hard because of the way they are related in the imaging equation. For example, when $t(\mathbf{x})$ is close to 1, the effect of $\mathbf{A}(\mathbf{x})$ becomes negligible in the hazy image. For this reason, we estimate $(1 - t(\mathbf{x}))\mathbf{A}(\mathbf{x})$ (denoted by $\mathbf{K}(\mathbf{x})$) as a whole. So, in our method we use the following form of the relaxed equation

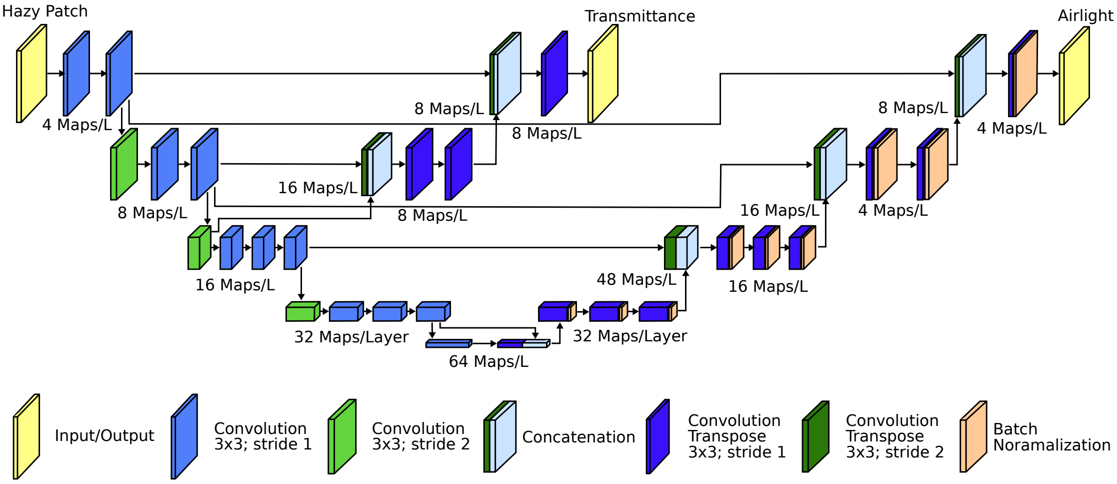


Figure 5.1: Proposed $t(\mathbf{x})$-$\mathbf{K}(\mathbf{x})$ estimator network

(equation 5.1).

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + \mathbf{K}(\mathbf{x}), \tag{5.2}$$

and try to estimate transmittance ($t(\mathbf{x})$) and airlight ($\mathbf{K}(\mathbf{x})$) using an FCN in order to recover the haze-free image. This method was originally proposed for NTIRE 2018 challenge on image dehazing [8].

5.1 $t(\mathbf{x})$-$\mathbf{K}(\mathbf{x})$ Estimator Network

As mentioned in the previous section, an FCN based neural network is utilized to estimate the haze parameters. The network is trained using a custom loss (named *Bi-directional Consistency Loss*) and a multi-scale training approach. All these are described in the following subsections.

5.1.1 Network Architecture

Our proposed estimator network is a two-way forked FCN that jointly estimates the scene transmittance and airlight (Fig. 5.1). There are two separate paths in the network to estimate the two parameters. One path for estimating the transmittance and the other one for estimating the airlight. The path to estimate airlight has more convolution layers than the transmittance estimation path. Success of the earlier methods point out the fact that transmittance can be well estimated from small patches. So, the receptive field can be kept small for computing transmittance. But airlight needs to be estimated from a large portion of the image. Therefore, to obtain an effectively large receptive field, the airlight estimation path contains more convolution layers. We have also taken convolutions with shift of 2 pixels instead of 1 to increase the size of the receptive field while keeping the number of layers, and therefore the number of training parameters less. Without this shift of 2 pixels, we would require more layers to get similar sized receptive field. The convolution layer are matched by same number of *convolution transpose* [20] layers in each path. Similar to the FCN of semantic segmentation [38], we have added some skip connections to combine small scale features with the upsampled feature maps. It helps in retaining fine details in the output. This also helps in the propagation of the gradient during the training. In the path for estimating the airlight, '*elu*' [17] is used as a activation function after each convolution layer except the last layer, whereas in the transmittance estimation path 'sigmoid' activation function is used. In the last layer of both the paths 'sigmoid' activation function is used. Batch-Normalization [30] layer is employed in the last few layers of the airlight estimation path to reduce the chance of over-fitting. Note that the network is designed in such a way that it can take input whose dimension is integer multiple of 128, e.g., 128 × 128, 256 × 256 and so on. This is possible due to the fully convolutional nature of the network.

5.1.2 Bi-directional Consistency Loss

In training neural networks the common approach is to use categorical cross-entropy or mean squared error (MSE) as the loss depending on the task at hand (e.g. classification or regression). But using MSE to train the estimator network can give rise to certain problems. The dehazed output that we obtain from the estimated $t(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$ is quite sensitive to the value of $t(\mathbf{x})$. This happens mainly because of the division by $t(\mathbf{x})$ when trying to obtain $\mathbf{J}(\mathbf{x})$ ($\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x})-\mathbf{K}(\mathbf{x})}{t(\mathbf{x})}$). This is specially true when the value of $t(\mathbf{x})$ is quite small. A small error in the estimate can produce large deviations in the output. So, it is better to use the corresponding haze-free image as the target output instead of targeting accurate haze parameter values. But it is also desirable that the atmospheric scattering model (equation 5.1) is not violated. So, we may simply use the following as the loss.

$$L = \frac{1}{N}|\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})t(\mathbf{x}) - \mathbf{K}(\mathbf{x})|. \tag{5.3}$$

But due to the ill-posed nature of the problem, the network may learn some wrong $t(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$ while still minimizing the loss. Also the network may get stuck at trivial solutions like $t(\mathbf{x}) = 0$ and $\mathbf{A}(\mathbf{x}) = \mathbf{I}(\mathbf{x})$. So, we propose a new loss to train our estimator network. We define the loss (L) as follows,

$$L = \frac{1}{N}\sum_{\mathbf{x}} \left(L_1(\mathbf{x}) + L_2(\mathbf{x})\right) \text{ where,} \tag{5.4}$$

$$L_1(\mathbf{x}) = |\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})t(\mathbf{x}) - \mathbf{K}(\mathbf{x})|, \text{ and} \tag{5.5}$$

$$L_2(\mathbf{x}) = \left|\mathbf{J}(\mathbf{x}) - \frac{\mathbf{I}(\mathbf{x}) - \mathbf{K}(\mathbf{x})}{\max\{t(\mathbf{x}), \epsilon\}}\right|. \tag{5.6}$$

Here $\mathbf{I}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ are Input hazy image and ground truth clean image respectively. N is the number of pixels in each image, $\mathbf{K}(\mathbf{x})$ is the estimated airlight and $t(\mathbf{x})$ is the estimated the transmittance we obtain using our estimator network. This design of the loss ensures that the haze parameters ($t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$) is estimated in such a way so that the following two conditions are satisfied.

- Using the estimated $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$, the hazy image can be generated from the ground-truth haze-free image.
- The hazy image can be dehazed using the estimated $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ so that the ground-truth haze-free image is obtained.

This imaging model inspired loss has certain advantages. First of all, this loss only requires a pair of hazy and haze-free images, apart from the network outputs. Ground-truth parameter values are not necessary. This design also helps in joint estimation of the parameters that conforms to the imaging model (equation 5.1).

5.1.3 Multi-level Strategy to Training

One of the weakness that is inherent in CNNs is that it works with a fixed image dimension and resolution. The dimension problem is usually tackled by resizing the input image. But re-sizing blindly may not maintain the aspect ratio and can cause the network to perform poorly due to the mismatch with the training data. An FCN may be used to solve this problem depending on the application the application at hand, but it does not solve the problem of resolution. If the resolution (physical area taken by a single pixel) of training and testing images does not match, the network performance can degrade. But we don't have any control over the dimension of an input image or its resolution. For these reasons we take a multi-level approach in both training and application phase. Here only the multilevel training part is described. The multilevel approach used in the dehazing step is described in the next section.

From the training data overlapping patches with 75% overlap from the both clear and corresponding hazy images. The extraction is done at multiple levels. In the first level, patches of size $P \times P$ are extracted, where $P = \min\{H, W\}$ for a given image of size $H \times W$. In the second level, we extract patches of size $\frac{P}{2} \times \frac{P}{2}$. In the third level patches of size $\frac{P}{4} \times \frac{P}{4}$ is extracted. This halving process is repeated until the patch size being considered falls below 128 × 128. Following this procedure, the extraction occurs at l levels where,

$$l = \lfloor(\log_2(\min(H, W)) - \log_2(128)) + 1\rfloor. \tag{5.7}$$

All the extracted patches are resized to 128 × 128 before they are used. These hazy and haze-free patch pairs are used to train our estimator network.

5.2 Dehazing Steps

The proposed method takes the following steps to dehaze an image.

1. **Multi-level estimation of transmittance ($t(\mathbf{x})$) and airlight ($\mathbf{K}(\mathbf{x})$) :** The method starts with extracting patches are multiple levels from the input image. Then $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ are estimated from the extracted patches with the help of the estimator network.

2. **Aggregation of $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$:** The obtained $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ maps are aggregated to form single full sized maps.

3. **Regularization using guided filter :** The obtained maps are further regularized to smooth out the artifacts arising from the patch based processing.

4. **Recovery of haze-free image :** At last the dehazed image is obtained by inverting the haze model.

Each of the step is described in detail in the following subsections.

5.2.1 Multi-level estimation of $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$

The method begins with downscaling the input image. Although this is not essential but it facilitates fast estimation on large images. An image is downscaled only if its number of both row and column is greater than L. The input image is scaled with a factor of $k = L/\min\{H, W\}$ for an image of size $H \times W$. In our experiments, we have taken $L = 850$, but this may be increased depending on the availability of resources. After scaling the image attains a dimension of $\lfloor kH \rfloor \times \lfloor kW \rfloor$. This scaled image is used in the subsequent steps.

Similar to the multilevel training strategy, the estimation of $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ is also done at multiple levels. The number of levels is restricted to maximum 3, because of the chosen downscaling factor. Now depending on the whether the image is downscaled or not the number of levels and the patch size at each level is decided. If an image is downscaled, then the image surely has dimensions bigger than L. For this reason these images are operated at three levels while working with patches of size 256 × 256, 384 × 384 and 512 × 512 in level one, two and three respectively. But the input image may not always be that big (number of rows or cols less than L). So, these needs to be handled separately. In this case the method starts working with patches of size 128 × 128. Increasing it to 256 × 256 in the second level and 384 × 384 in the third level. Note that this increment in the level, thereby increment in the patch size, is only done if the images are bigger than the corresponding patch size. If the images are not that big, the method may work at one or two levels only. This implicitly enforces that the images should be at least 128 × 128 for this method to work. So, after deciding the number of levels and the corresponding patch sizes, at each level, we take overlapping patches of the specified size, resize them to 128 × 128 and feed them to our estimator network. The network outputs $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$ maps of size 128 × 128. The obtained $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$-maps are then resized back to their actual sizes depending on the level the operation is being done. Then in each level we aggregate the patches to form $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$-maps of size $\lfloor kH \rfloor \times \lfloor kW \rfloor$, by averaging the estimates in the overlapping portions. After this step, we obtain transmittance and airlight maps for each level.

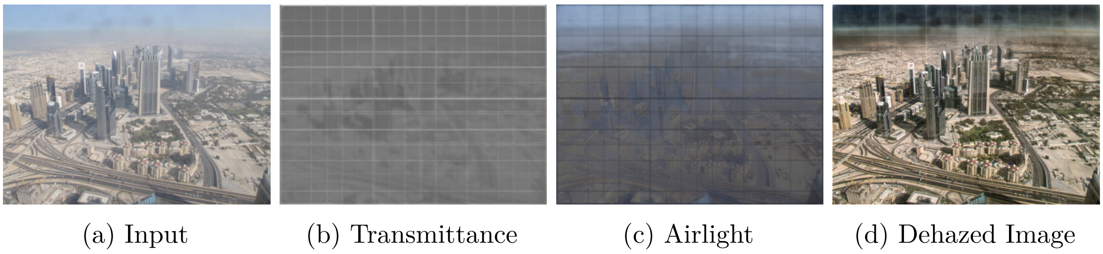
5.2.2 Aggregation of $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$

In the previous step, we have obtained transmittance and airlight map for each level. These needs to be aggregated to form single transmittance and airlight map. To aggregate them, we take weighted average of the estimates obtained at each level to generate $t(\mathbf{x})$ and $\mathbf{K}(\mathbf{x})$.

$$t(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(t)} t_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(t)}}, \tag{5.8}$$

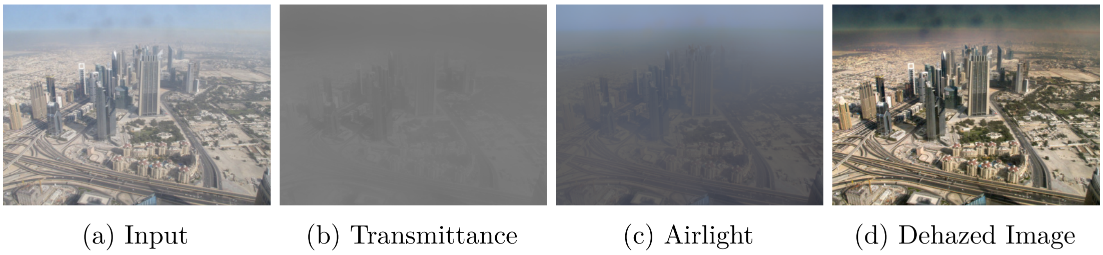
$$\mathbf{K}(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(K)} \mathbf{K}_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(K)}}. \tag{5.9}$$

Here $w_i^{(t)}$, $w_i^{(K)}$ are the weights that we use to aggregate $t(\mathbf{x})$'s and $\mathbf{K}(\mathbf{x})$'s respectively. $t_i(\mathbf{x})$ and $\mathbf{K}_i(\mathbf{x})$ denote the estimates we have obtained at level i and l denotes the number of levels the image is operated on in the previous step. In our experiment, we have taken all the weights to be 1.



(a) Input (b) Transmittance (c) Airlight (d) Dehazed Image

Figure 5.2: Halos appear due to patch based processing of the image. This affects the output.



(a) Input (b) Transmittance (c) Airlight (d) Dehazed Image

Figure 5.3: Use of guided filter successfully removes the halos.

5.2.3 Regularization using Guided Filter

We have obtained transmittance and airlight maps of size $\lfloor kH \rfloor \times \lfloor kW \rfloor$ after aggregation. But due to the patch based processing, these maps usually contain halos at the border of the patches (Figure 5.2). Using them directly results in artifacts in the output. So, these estimates needs to be refined before they are used in the recovery of the haze-free image. For this purpose, an edge-preserving smoothing filter is required that smooths the estimates but at the same time respects the object boundaries present in the image. We have used Guided Filter [27] for this purpose because of its efficiency (Figure 5.3). The Guided Filter filters a given input image while considering the content of a guidance image. We utilize this as an edge-preserving smoothing filter. For smoothing the airlight ($\mathbf{K}(\mathbf{x})$), each of its color channel is smoothed separately with corresponding channel of the hazy image as the guidance image. For smoothing the transmittance ($t(\mathbf{x})$), the gray-scale version of the hazy image is utilized as the guide.

5.2.4 Recovery of haze-free image

After obtaining the smooth transmittance and airlight maps, the dehazed image can be obtained by simply inverting the haze formation model (equation 5.2). But the bigger images has been downscaled before processing. For those images the obtained transmittance and airlight maps are of size $\lfloor kH \rfloor \times \lfloor kW \rfloor$. So, they are rescaled back to the original image size i.e. $H \times W$ to get the dehazed image of the same size as the input. After that, inverting equation 5.2, we obtain the dehazed image as follows,

$$\mathbf{J}'(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{K}(\mathbf{x})}{\max\{t(\mathbf{x}), \epsilon\}}. \tag{5.10}$$

Here $\mathbf{J}'(\mathbf{x})$ is the estimated dehazed image. Note that we have clipped the value of $\mathbf{J}'$ between 0 and 1 so that the output stays within the valid range of image intensity. ϵ is a small value used in the denominator to avoid division by zero.

5.3 Experimental Settings

All the results that is reported here is generated using a 3.6GHz quad core machine with 32 GB RAM and one Nvidia GeForce GTX 745 GPU, with Ubuntu 16.04 running on top of them. The estimator network is trained with the help of Keras [16] deep learning library with tensorflow backend [1]. The training data is generated using the training images of I-HAZE [2] and O-HAZE [3] dataset. The network is trained for 300 epochs with a batch size of 10 with the help of Adagrad optimizer [19]. For smoothing the transmittance and the airlight maps, the guided filter is applied a radius size of 60 and an epsilon value of 0.001.

5.4 Results

The model we have considered here is the relaxed version of the imaging model, where the environmental illumination varies from pixel to pixel. As shown in chapter 3, this relaxed version works for both day and night-time images. So, we have evaluated this method on both day and night-time images. The daytime images include synthetic hazy images of Fattal [22], the validation images of I-HAZE [2] and O-HAZE [3] dataset and some benchmark hazy images. The night-time images include only some benchmark images without any ground truth. For synthetic images and the images of I-HAZE and O-HAZE dataset we have used metrics like PSNR, SSIM and ΔE_{00} to quantitatively evaluate the results. For a comparative study the results of He et al. [28], Fattal [22], Ren et al. [53], Berman et al. [11] and Li et al. [35] has been included. The results of Fattal [22] has been obtained from the authors website[1]. The results of He et al. [28] has been generated using our own implementation of the method. The results of the remaining methods are generated using the code provided the respective authors in their default setting. For the images of I-HAZE [2] and O-HAZE [3] dataset, the results of He

[1] `http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/`

et al. [28] and Li et al. [35] could not be generated due to the resource requirement of the implementation on large images.

5.4.1 Quantitative Results

Table 5.1: Quantitative Comparison on the images of Fattal's dataset. High PSNR and SSIM indicates better results, while it is the opposite for ΔE_{00}. The best results are bold and the second best results are underlined. Note that in Fattal's method only $t(\mathbf{x})$ is computed and $\mathbf{A}$ is manually provided.

Image	He et al. [28]			Fattal [22]			Ren et al. [53]			Berman et al. [11]		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	11.16	0.78	28.74	**21.43**	**0.96**	**6.34**	14.17	0.87	20.25	15.68	0.89	16.9
couch	18.4	0.86	13.89	**20.8**	**0.9**	**6.71**	17.99	0.88	12.91	17.24	0.87	14.18
dolls	19.73	0.85	10.65	**21.29**	0.77	**6.1**	16.93	**0.86**	12.37	15.69	0.83	15.73
flower1	14.1	0.88	23.26	**30.01**	**0.98**	**3.91**	9.08	0.43	24.64	12.15	0.72	20.99
flower2	14.37	0.86	20.94	**31.94**	**0.99**	**2.92**	10.81	0.6	22.45	11.86	0.68	21.16
lawn1	13.84	0.8	22.38	**24.49**	**0.97**	**6.65**	14.37	0.83	21	14.78	0.86	17.92
lawn2	11.2	0.74	29.32	**24.94**	**0.97**	**6.46**	13.29	0.77	22.27	15.32	0.87	17.8
mansion	17.45	0.87	19.35	**26.96**	**0.97**	**4.04**	17.69	0.89	17.52	17.33	0.87	15.83
moebius	12.66	0.78	26.7	**19.01**	**0.9**	**10.61**	16.36	**0.9**	19.85	14.58	0.85	22.39
raindeer	18.12	0.83	14.22	**26.22**	**0.94**	**4.1**	16.82	0.81	15.49	16.59	0.82	15.28
road1	12.95	0.8	26.11	**25.74**	**0.96**	**5.24**	14.11	0.84	22.22	16.3	0.88	19.06
road2	15.84	0.84	22.13	**23.6**	**0.96**	**7.11**	16.45	0.88	20.17	18.22	0.9	16.82
Average	14.98	0.82	21.47	**24.7**	**0.94**	**5.85**	14.84	0.8	19.26	15.48	0.84	17.84

Image	Li et al. [35]			ch3			ch4			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	9.44	0.62	34.64	16.99	0.9	11.59	17.17	0.84	14.68	14.46	0.9	24.39
couch	16.77	0.83	17.32	16.23	0.78	15.84	19.96	0.79	10	19.51	0.85	12.94
dolls	17.21	0.85	10.88	13.64	0.8	18.14	12.55	0.82	16.6	14.9	0.84	13.51
flower1	12.22	0.79	29.41	19.27	0.89	11.19	16.3	0.8	20.87	21.31	0.95	14.72
flower2	13.13	0.79	25.26	21.93	0.9	7.79	18.08	0.82	16.43	22.7	0.95	11.38
lawn1	11.32	0.69	31.74	17.14	0.84	13.78	14.26	0.84	24.82	16.13	0.88	20.21
lawn2	10.98	0.68	31.7	15.79	0.8	15.9	14.64	0.82	24.84	14.89	0.89	20.92
mansion	14.24	0.7	24	19.65	0.84	9.2	19.64	0.82	14.5	21.89	0.93	13.64
moebius	13.22	0.77	27.61	18.72	0.86	12.59	16.84	0.82	17	18.22	**0.9**	15.28
raindeer	16.53	0.8	18.5	15.47	0.74	17.23	17.92	0.74	14.85	22.63	0.9	10.7
road1	11.75	0.66	29.31	15.91	0.78	14.45	16.83	0.81	22.51	16.14	0.9	18.41
road2	11.96	0.62	30.96	15.02	0.82	15.74	18.2	0.86	19.45	15.88	0.91	20.79
Average	13.23	0.73	25.94	17.15	0.83	13.62	16.86	0.82	18.05	18.22	0.9	16.41

We have quantitatively evaluated the results obtained by the proposed method on the synthetic hazy images given by Fattal [22] and I-HAZE [2] and O-HAZE [3]. The results obtained on Fattal [22], I-HAZE [2] and O-HAZE [3] dataset is reported in Table 5.1 and 5.2 respectively. For the synthetic hazy images Fattal [22] still remains as the top scoring method in all three metrics, except the *dolls* image. For our method it is seen in terms of PSNR and SSIM the scores have improved quite a bit than the previous methods, but the color line based method still has better ΔE_{00} scores by quite a big margin. So, we

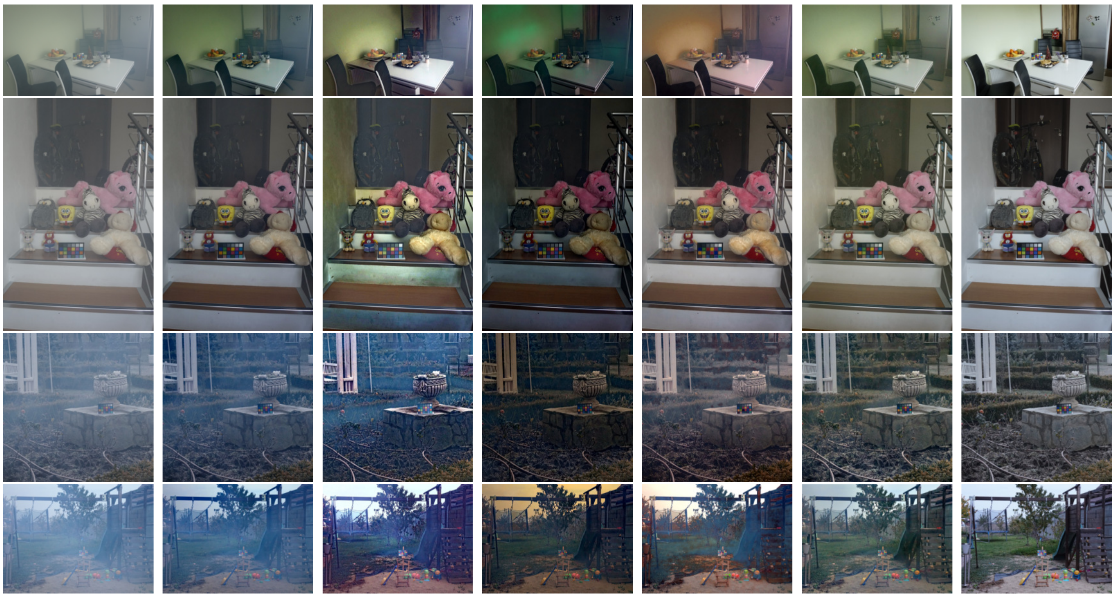
Table 5.2: Quantitative Comparison on the images of I-HAZE and O-HAZE dataset. The best results are bold and the second best results are underlined. (I) in the image column denotes indoor image whereas (O) denotes an outdoor image.

Image	Ren et al. [53]			Berman et al. [11]			Our (ch3)			Our (ch4)			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
26 (I)	11.02	0.78	22.36	12.41	0.77	20.15	8.74	0.52	32.09	14.84	0.76	15.54	**15.7**	**0.87**	**13.85**
27 (I)	17.6	0.84	12.3	14.79	0.75	18.03	15.36	0.77	16.39	16.08	0.78	16.11	**21.84**	**0.87**	**8.25**
28 (I)	13.1	0.79	17.05	13.29	0.72	19.24	13.41	0.7	22.55	15.5	0.77	17.61	**16.11**	**0.82**	**13.7**
29 (I)	17.6	0.89	11.42	14.66	0.79	15.73	9.91	0.72	29.48	16.8	0.89	13.86	**21.82**	**0.91**	**9.33**
30 (I)	16.78	0.79	14.2	13.92	0.71	19.08	10.99	0.7	25.66	17.02	**0.85**	13.92	**20.62**	0.83	**12.19**
36 (O)	19.46	0.81	11.84	16.92	0.72	14.42	11.68	0.68	28.63	14.52	**0.83**	20.23	**23.11**	0.82	**7.6**
37 (O)	17.72	0.73	13.27	14.98	0.63	15.13	15.42	0.81	18.33	17.49	**0.85**	16.62	**21.32**	0.76	**8.52**
38 (O)	16.2	0.77	19.01	15.54	0.75	16.92	10.48	0.68	28.39	15.85	**0.86**	17.46	**22.25**	0.82	**8.51**
39 (O)	15.75	0.75	16.74	17.64	0.77	16.42	11.81	0.64	27.48	13.59	0.73	22.54	**19.9**	**0.8**	**10.84**
40 (O)	18.66	0.81	11.95	17.04	0.76	15.06	12.65	0.76	25.48	14.16	**0.84**	20.78	**22.1**	**0.84**	**7.85**
Average (I)	15.22	0.818	15.46	13.81	0.748	18.44	11.40	0.658	23.75	15.18	0.828	16.24	**19.21**	**0.86**	**11.46**
Average (O)	17.55	0.774	14.56	16.42	0.726	15.59	13.62	0.624	19.08	18.17	0.764	13.22	**21.73**	**0.81**	**8.66**

may conclude that the estimate of $\mathbf{A}$ is still not very good. For the results obtained on the images of I-HAZE and O-HAZE dataset, the scores have some similarities but due to the large size of the images (e.g. 4476×2882) there have been some changes in the trend. The current proposed method has gone to the top in most of the images in all three scores. The happens mainly due to the fact that the network has been trained using the training images of I-HAZE and O-HAZE dataset and it has been designed to handle large sized images. The scores of Ren et al. [53] was just below the scores of our method in the dataset of Fattal [22]. For I-HAZE and O-HAZE dataset we see scores obtained by the results of Ren et al. [53] are ranked second most of the time. It has even beaten our color line based method in terms of ΔE_{00}. This happens due to the multi-scale nature of their proposed method. Our color line based has performed poorly in most of the cases due to the resolution of the hazy images. As it works with a fixed patch size in these large images it had failed to perform satisfactorily.

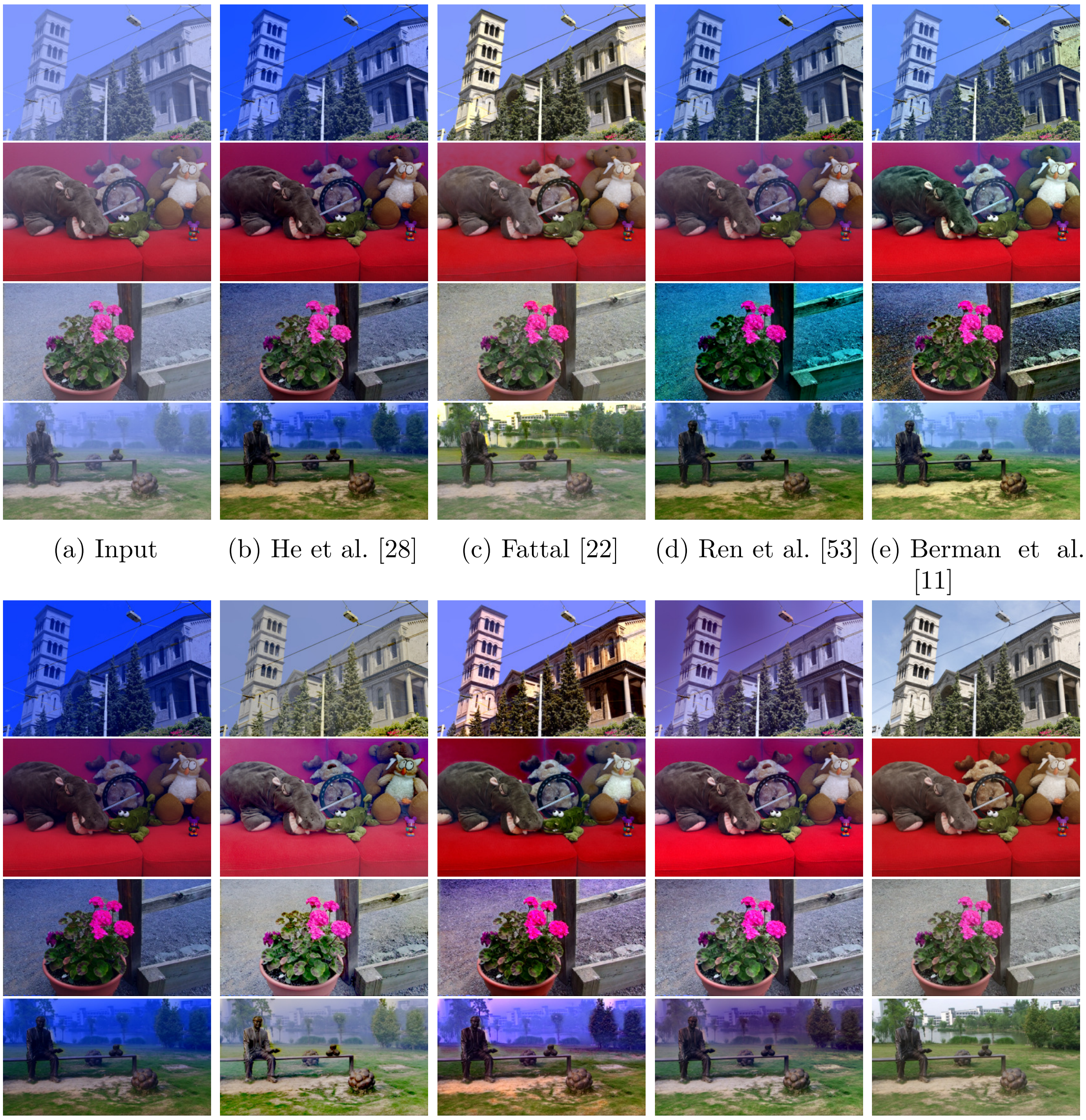
5.4.2 Qualitative Results

In this section we qualitatively compare results obtained on different kind of images. These include the some synthetic images of Fattal [22], the images from the validation set of I-HAZE and O-HAZE dataset, some real-world images. We have also included some night-time images as this method works with the relaxed version of the imaging model. The figure 5.5 shows the actual results obtained on the synthetic images given by Fattal [22]. It is seen that the results follow the pattern that we have seen in the quantitative results (Table 5.1). The method of Fattal [22] is performing the best. There is color distortions in all the contending method. The current method has performed a litte poorly in the *church* and *lawn1* image than our pervious method. The blue haze part has become dark. But the results does not have the yellowish tone that is present in the results obtained by our previous method. The figure 5.4 shows the results of the I-HAZE and O-HAZE datasets. Our method has performed well in cleaning the haze. The other methods have performed relatively well in the indoor images, but they have failed in the outdoor ones. The results on real world images is shown in figure 5.6. Apart from Fattal [22], the method of Berman et al. [11] is performing quite well, except the *ny12* image. The same thing can't be said for the other methods. The results obtained by our method is not good in the some images (*tiananmen* and *ny12*). Since the images of I-HAZE and O-HAZE is not like these real-world hazy images, our network has failed to properly estimate the haze parameters. Among the results of night-time images (figure 5.7) we see that the daytime dehazing methods ([28, 53, 11, 35]) are not able to clear the haze. The method of Berman et al. [11] is an exception in this regard. Despite being a daytime deahzing method it could clear haze to some extent all the images. Our method is performing poorly despite using a relaxed imaging model. The training data is to be blamed here. Since the training data did not contain night-time images, our method is performing poorly.



(a) Input (b) Ren et al. [53] (c) Berman et al. [11] (d) Our (ch3) (e) Our (ch4) (f) Our (g) Ground Truth

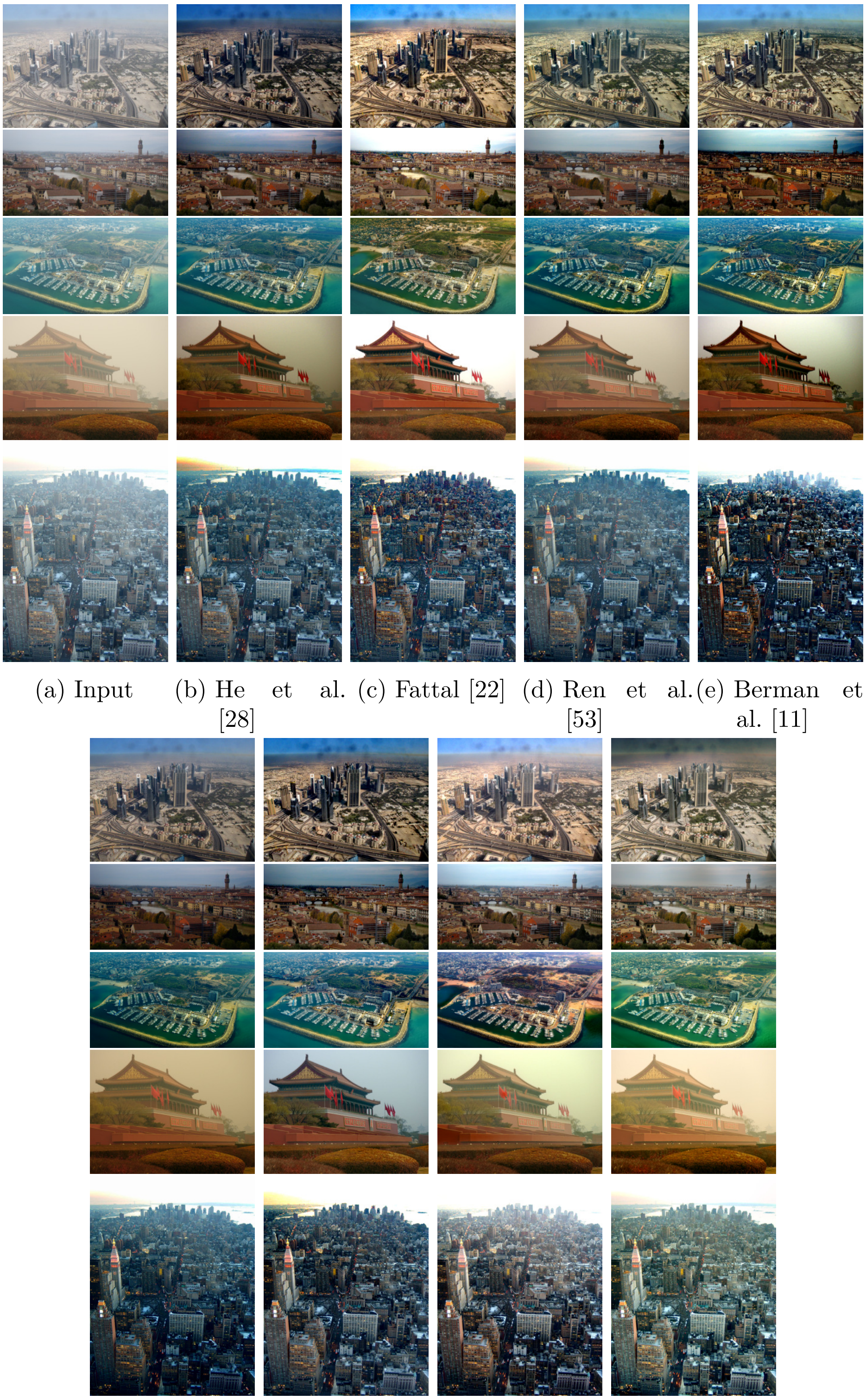
Figure 5.4: Visual comparison of the results on two images of I-HAZE and two images of O-HAZE dataset



(a) Input (b) He et al. [28] (c) Fattal [22] (d) Ren et al. [53] (e) Berman et al. [11]

(f) Li et al. [35] (g) Our (ch3) (h) Our (ch4) (i) Our (j) Ground Truth

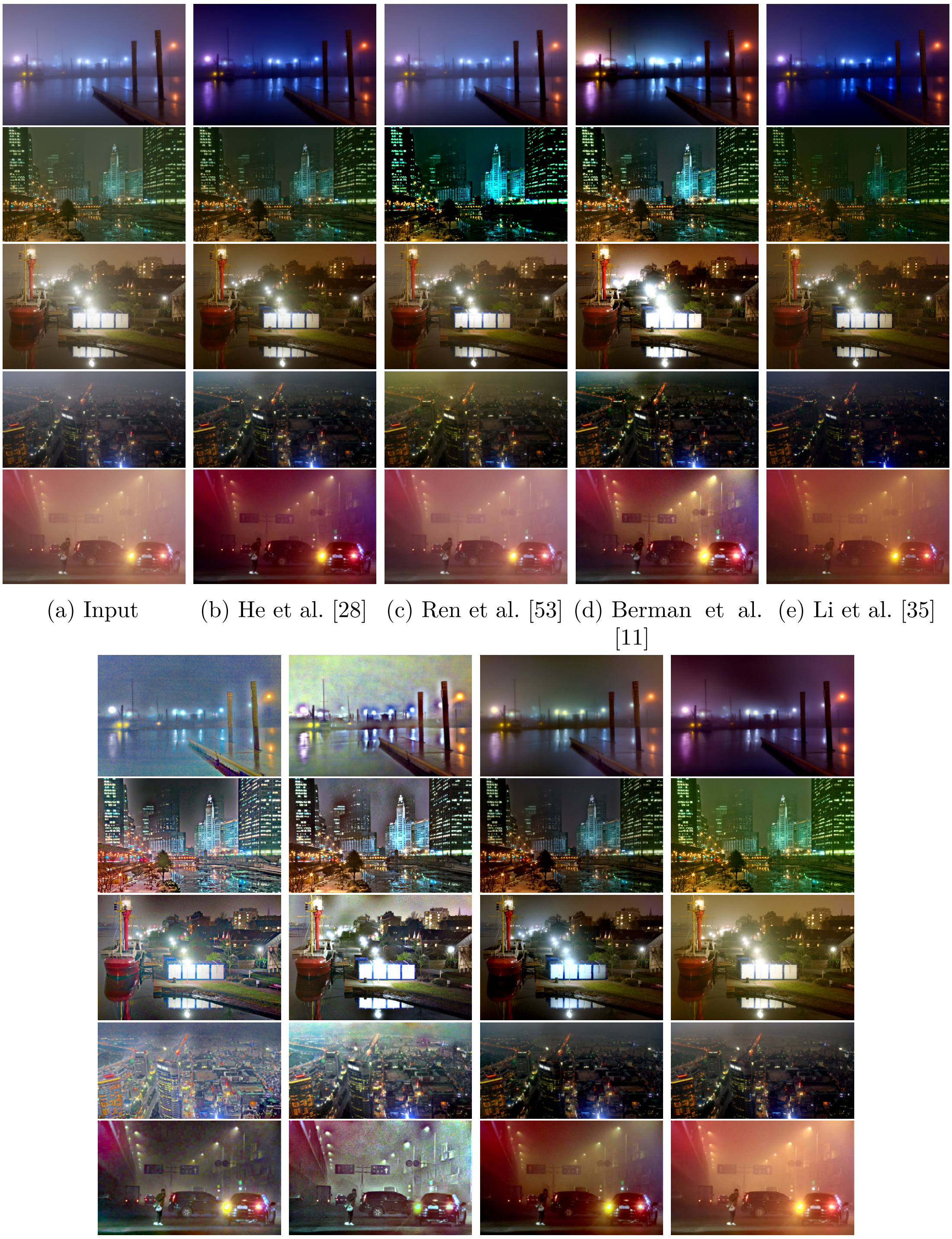
Figure 5.5: Visual comparison of the results on four synthetic images: *church*, *couch*, *flower2*, and *lawn1*.



(a) Input (b) He et al. [28] (c) Fattal [22] (d) Ren et al. [53] (e) Berman et al. [11]

(f) Li et al. [35] (g) Our (ch3) (h) Our (ch4) (i) Our

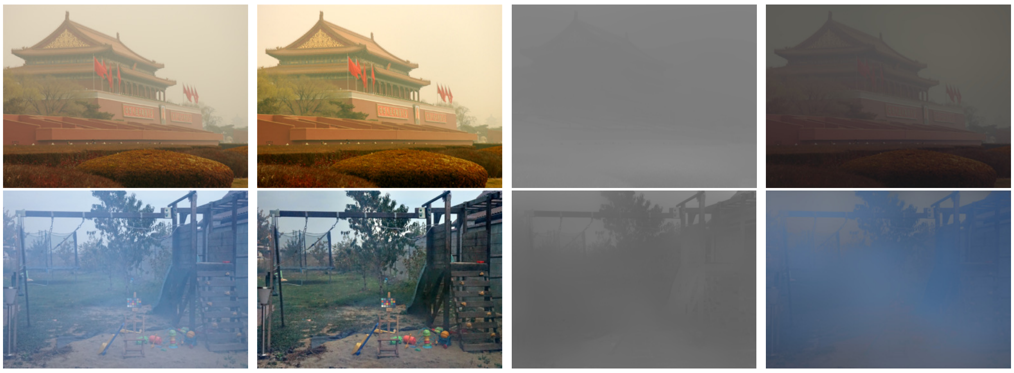
Figure 5.6: Visual comparison of results on *dubai*, *florence*, *herzeliya*, *tiananmen*, and *ny12* image.



(a) Input (b) He et al. [28] (c) Ren et al. [53] (d) Berman et al. [11] (e) Li et al. [35]

(f) Zhang et al. [67] (g) Li et al. [36] (h) Our (ch3) (i) Our

Figure 5.7: Visual comparison of results on night-time images.



(a) Input (b) Output (c) Transmittance (d) Airlight

Figure 5.8: Transmittance and airlight obtained by our method.

5.4.3 Discussion

From the quantitative and qualitative results it is seen that the proposed method works well in the synthetic images and on the images of I-HAZE and O-HAZE, but is not working satisfactorily in the real world images. This clearly shows that the training data is not a good representative of the real world hazy images. Since our method hinges on the trained model to estimate the haze parameters, the results of the method gets decided by the training data. On the other hand the transmittance and airlight estimated is not always accurate. This has been reflected in the dehazed results. If we see the results in figure 5.8, it is seen that the airlight is almost similar to the input image and the transmittance is almost same throughout the whole image. But the network has learned to predict this because this kind of prediction is giving good dehazed results (in terms of the loss) in the training data. This can be confirmed from the second image of figure 5.8, although this raises a question on the imaging model.

5.5 Summary

In this chapter we have proposed a image dehazing method that can work if hazy and haze-free image pairs are given as the training data. It does not require ground truth transmittance and environmental illumination. It has been enabled by our proposed loss (Bi-directional Consistency Loss). Using this loss we have trained an FCN to predict transmittance and airlight from patches. The utilization of FCN has facilitated the use of bigger patches. As this type of network can yield output that has size similar to the input, we are able to work with patches where transmittance can vary. We have also proposed a multi-level approach to the training and inference to mitigate the problem of scale that is faced when working with big images. But how it affects the performance needs to be evaluated. One problem that has negatively results is the training data. The network we have used was trained using the training images of I-HAZE [2] and O-HAZE [3] dataset, and those images are quite different from real world images. We hope to reduce this dependency in the method proposed in the next chapter.

Chapter 6

Dehazing based on patch quality comparator

In the previous chapter we have seen that the haze parameters predicted by the network heavily depends on the data used to train it. This issue of learning based systems is usually tackled by using a variety of samples in the training data. But collecting hazy and haze-free image pairs of the same scene can quite hard in practice. So, the usual way is synthesizing hazy images and training a regressor network to estimate the haze parameters. But here we take an altogether different approach. Instead of predicting the transmittance directly from a given hazy patch, we propose to find it by comparing different dehazed versions of the hazy patch with the original hazy one. This is motivated by the fact that comparing two patches and saying which one has more haze is easier than saying the haze level of a given patch. For this method we again start with the original version of the atmospheric scattering model.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A}. \tag{6.1}$$

Where $\mathbf{I}(\mathbf{x})$ is the observed intensity, $\mathbf{J}(\mathbf{x})$ is the intensity of light coming from the scene objects and before getting scattered, $t(\mathbf{x})$ is the scene transmittance denoting the amount of light that reaches the observer after getting scattered and $\mathbf{A}$ denotes the global environmental illumination. From this the aim is to recover the $\mathbf{J}(\mathbf{x})$ by estimating the remaining parameter values. Since the mapping $\mathbf{J}(\mathbf{x}) \rightarrow \mathbf{I}(\mathbf{x})$ is not one-to-one, as $t(\mathbf{x})$ varies from pixel to pixel, the estimation of $\mathbf{J}(\mathbf{x})$ independently at each pixel $\mathbf{x}$ can be inconclusive. The value of $\mathbf{I}(\mathbf{x})$ could be due to only $\mathbf{J}(\mathbf{x})$ when $t(\mathbf{x}) = 1$ or due to only $\mathbf{A}$ when $t(\mathbf{x}) = 0$. This confusion exists even if we know $\mathbf{A}$ for the given image. To surmount this hurdle a simple assumption is adopted which is common in the literature [22, 61]: within a small patch of the image the depth of the scene and consequently, the transmittance t is assumed to be constant. This assumption is valid because a patch of the image usually corresponds to a small part of a single surface in the scene, which may be approximated by a relatively smooth surface except at the places where the patch is on the boundary between two surfaces. Therefore the following equation is utilized to estimate transmittance within a patch.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t + (1-t)\mathbf{A} \tag{6.2}$$

But, if the patch that we consider is very smooth, i.e., $\mathbf{I}(\mathbf{x}) = \text{const}$ for all $\mathbf{x}$ and contrast is nil, the effect of haze is neither apparent nor measurable. So, we may argue that using equation 6.2 is not advantageous in two cases: very smooth patches and patches with strong depth discontinuity. Therefore, when we process patches for estimating the parameters, we discard these two kinds of patches, only the remaining ones are considered. In our method we concentrate only on the $t(\mathbf{x})$ estimation part an assume that $\mathbf{A}$ can computed by any one of the existing methods. Here $\mathbf{A}$ is computed as it described by He et al. [28].

6.1 Proposed Approach

The method that we propose here is a daytime image dehazing method that finds the transmittance at each patch (and subsequently at each pixel) by comparing the dehazed version of the patch with the hazy one. This comparison is performed by our proposed module called the *patch quality comparator*. This comparator, when given two patches as input, can indicate which one is of better quality in terms of haziness. We build the comparator in such a way, that the natural looking patches (e.g., without saturated colors, noise etc.) and the patches with less haze are declared as the better one. With the help of this comparator, we search for a transmittance value that can dehaze the given hazy patch and at the same time does not degrade the dehazed output by overdoing. This patch quality comparator is described in the following subsection, after discussing the principle behind this approach.

6.1.1 Principle

Our approach is built on the following principle. Given a hazy patch there is a $t = t'$ that properly dehazes this patch. Dehazing this patch with $t > t'$ retains some haze in the dehazed output, and on the other hand, using $t < t'$ produces over contrasted, bad looking, unnatural output. So, dehazing a given patch with $t = t'$, we get the actual $\mathbf{J}(\mathbf{x})$ by the following equation.

$$J^c_{t'}(\mathbf{x}) = A^c - \frac{A^c - I^c(\mathbf{x})}{t'}. \tag{6.3}$$

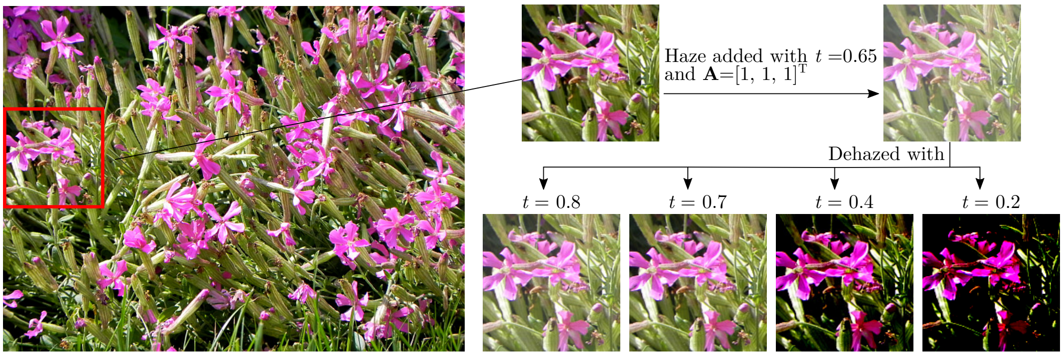


Figure 6.1: Haze is added in a patch. This haze patch is dehazed with t values less than 0.65 and greater than 0.65.

Where $c \in \{\text{R, G, B}\}$ is one of the color channels. If the same patch is dehazed with a t that is not equal to t', we can write the following.

$$J_t^c(\mathbf{x}) = A^c - \frac{A^c - I^c(\mathbf{x})}{t}. \tag{6.4}$$

From these two equation the following can be written

$$\Delta_{J^c}(\mathbf{x}) = J_t^c(\mathbf{x}) - J_{t'}^c(\mathbf{x}) = (A^c - I^c(\mathbf{x}))\Big(\frac{t - t'}{tt'}\Big). \tag{6.5}$$

So, depending on the value of $\Delta_{J^c}(\mathbf{x})$ we can say whether the dehazed output $J_t^c(\mathbf{x})$ is more than the actual $J^c(\mathbf{x})$ or less. Since $\mathbf{A}$ is the environmental illumination, everything in the scene is illuminated by it. So, the value of $I^c(\mathbf{x})$ $(= r(\mathbf{x})A^c$, where $r(\mathbf{x}) \in [0, 1]$ denotes the reflectance property of the scene object) can't be more than A^c. So, the first term of equation 6.5 always remains positive. Therefore the value of $\Delta_{J^c}(\mathbf{x})$ depends only on the relation between t and t'. If $t < t'$ then $\Delta(\mathbf{x})$ is negative. That means the dehazed output is less than the actual one, therefore darker. On the other hand, if $t > t'$ then the dehazed output is more than actual one: it can be further refined. This principle is also illustrated using an example in figure 6.1. We take a patch from an unhazed clear image. From this patch we generate a hazy patch using equation 6.2) with $t = 0.65$ and $\mathbf{A} = [1, 1, 1]^T$. This generated hazy patch is then dehazed using the same $\mathbf{A}$ but with different t's (e.g., 0.8, 0.7, 0.4, and 0.2). It is seen from the figure that dehazing the hazy patch with t less than the ideal (0.65) produces bad output and the patches dehazed with t greater than 0.65 are better than the original hazy patch. In these hazy patches some haze is still present and it can be further cleaned. We say these are good dehazed patches. Now if we dehaze a given patch with different values of t, we get some good dehazed patches and some bad dehazed patches (Figure 6.2). If we arrange these dehazed

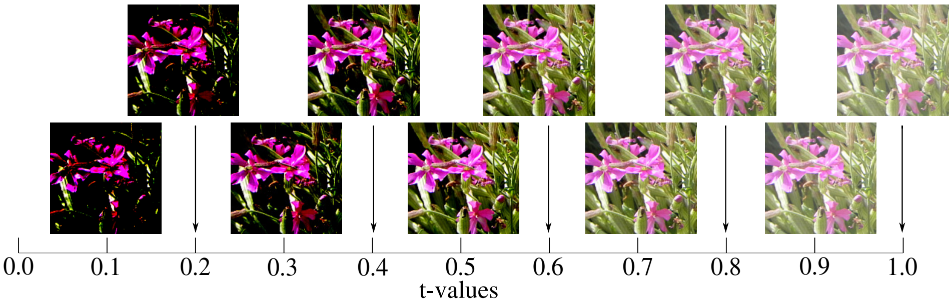


Figure 6.2: Same haze patch is dehazed with different t's. At $t = 1$ the dehazed patch is same as the original haze patch.

patches, say, in ascending order based on the value of t that has been used to obtain these patches, then starting from $t = 1$ down to $t = t'$, we get good dehazed patches, and bad dehazed patches for the remaining values of t. Thus the transition from good to bad dehazed patches occurs at $t = t'$ if we vary the value of t between 0 and 1. We use this fact to obtain the appropriate value of t for any given hazy patch. However, note that to be able to find the point of transition, we must be able to tell, without the knowledge of desired value of t, i.e. t', whether a dehazed patch is a good dehazed patch or a bad one. For this purpose we build the patch quality comparator.

6.1.2 Patch Quality Comparator

The patch quality comparator we propose here compares two patches, say, a given patch and its dehazed version, and reports whether the dehazed patch is good or bad. If we know beforehand whether a dehazed patch is good or bad, we can use this information to train a classifier to perform this comparison. Now instead of using some handcrafted features and employing a two-class (good and bad) classifier to do this task, we use a CNN to learn the features and do the classification simultaneously. The proposed network takes two patches as input and produces two outputs to denote which one of the input is better (figure 6.3). The ideal output is $[1, 0]^T$ if the first input is better and $[0, 1]^T$ otherwise. Here the assumption is that the two patches differ only in the amount of haze, and represent the same part of the same scene. The basic structure of the network is inspired from the common CNN based classifiers i.e. convolutional layers for feature extraction followed by dense layers for classification based on the extracted features. Our network is designed in the same way. As our network takes two patches as input, we process both of them by the same set of convolutional layers so as to extract the same features from each of them. Another advantage of extracting the same set of features from both the inputs is that, this reduces the dependence of the network on the ordering

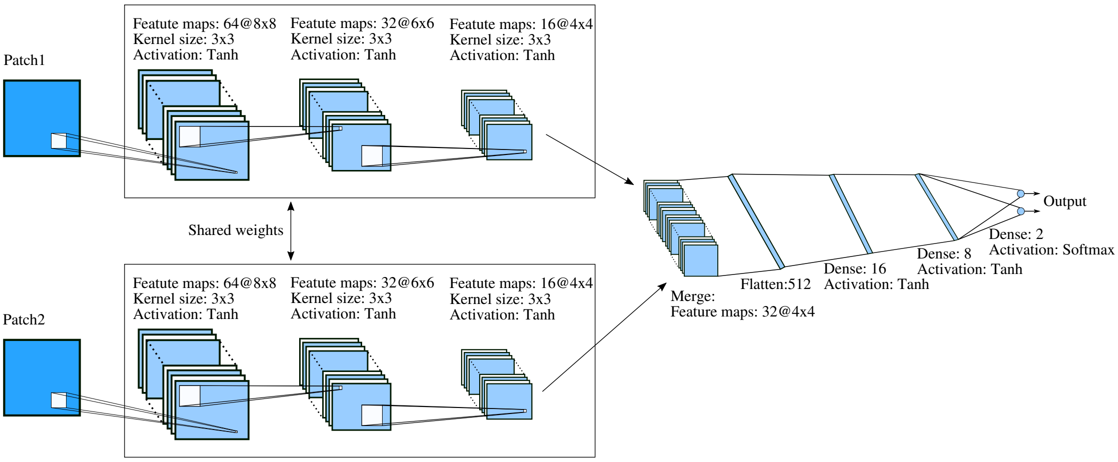


Figure 6.3: Architecture of our Patch Quality Comparator

of the training data. Now, the usual classification networks stack around 8 to 19 layers to classify an image. But in our case, we are processing small image patches (10×10). Therefore, we need only small number of convolutional layers to extract features. On the other hand, we have to take small number of dense layers to avoid overfitting of the classifier. We have used *tanh* function as the non-linear activation throughout the network except the last layer. In the last layer we use *softmax* activation function [13, p. 198] as we are training the network as a classifier. The use of *tanh* results in relatively large gradients compared to sigmoid during backpropagation which leads to speed up of the optimization process. Moreover, the problem of vanishing gradients associated with tanh is not likely to occur as the network is not very deep.

We train the comparator to distinguish the haziness of patches making sure the following conditions are met:

1. The haze patch is better than a bad dehazed patch.
2. A good dehazed patch is better than the haze patch.

Using this comparator we find at what value of t the transition from good dehazed patch to bad dehazed patch occurs, by repeatedly dehazing a given patch using different values of t. As we have already mentioned, this point of transition gives the desired value of t for this patch. Now, instead of arbitrarily searching for this point, we employ binary search to do this computation efficiently.

6.2 Implementation of the Method

The proposed dehazing method has the following 4 main steps

1. **Computation of environmental illumination:** When a hazy image is given as input, first $\mathbf{A}$ is computed.

2. **Transmittance finding using binary search:** From the input image, we take patches of size $\omega \times \omega$ with 50% overlap (both horizontally and vertically) and find t in each patch based on the response of the patch quality comparator, following binary search.

3. $t(\mathbf{x})$ **aggregation and interpolation:** Because of the overlap, a pixel receives more than one estimate of t. We take the average of these estimates while determining the value of t at that pixel. Now, it is quite likely that at some pixels the t is not estimated as they belong to either very smooth patches or to patches with strong edges, which are discarded from being processed as stated earlier. At those positions the value of t is interpolated.

4. **Haze-free image recovery:** Finally, the dehazed image is obtained by dehazing the image with the obtained $t(\mathbf{x})$ and already computed $\mathbf{A}$.

Details of each step is provided in the following subsections.

6.2.1 Computation of Environmental Illumination

The environmental illumination ($\mathbf{A}$) is computed as it is described by He et al. [28]. We describe it here briefly for the sake of completeness. Environmental illumination can be estimated from the color of the most haze opaque region, i.e., where value of the t is least or, in other words, depth is large. This region is detected with the help of the dark channel of the hazy image. Dark channel of an image $\mathbf{I}$ is given by

$$D(\mathbf{x}) = \min_{\mathbf{y} \in \Omega(\mathbf{x})} \Big(\min_{c \in \{R,G,B\}} I^c(\mathbf{y}) \Big), \tag{6.6}$$

where I^c is a color channel of $\mathbf{I}$ and $\Omega(\mathbf{x})$ is a local patch centered at $\mathbf{x}$. As dark channel approximates denseness of haze, the most haze opaque region selected by picking the top 0.1% brightest pixels in the dark channel. Within this region, the pixel with highest intensity in the input image is selected as the environmental illumination. This is utilized as the environmental illumination in the subsequent steps.

6.2.2 Transmittance finding using binary search

In this step we estimate t from a given patch. For that the image is first divided into patches of size $\omega \times \omega$ with 50% overlap. We find the ideal $t(= t')$ for each hazy patch using patch quality comparator following binary search strategy. The search is guided by the principle described in the Section 6.1.1, that is, we get a good dehazed patch when $t > t'$, and a bad dehazed patch when $t < t'$. We need to find the point of transition from good dehazed patch to bad dehazed one, and this good/bad decision is taken by the patch quality comparator.

We begin the process with $t_e = 1$ and $t_b = 0$. We compute $t_m = (t_b + t_e)/2$. Then the input patch ($\mathbf{I}_p$) is dehazed with the $t = t_m$ and $\mathbf{A}$ computed in the first step, and the input hazy patch and the dehazed patch is compared. If the dehazed patch is bad then we can say that $t_m < t'$. Therefore t' lies in the range (t_m, t_e). So, we set t_b to t_m. On the other hand, if the obtained dehazed patch is good, then $t_m > t'$. So, we set t_e to t_m as t' lies in the range (t_b, t_m). This process is repeated (i.e. computing t_m, dehazing with new t_m, comparing new dehazed patch with the hazy one, and finally updating t_e or t_b) until $(t_e - t_b)$ becomes small enough ($\leq \theta_t$). When the search stops, $t_m = (t_b + t_e)/2$ is declared as the desired t for this patch. The above mentioned steps are written in algorithmic form in Algorithm 1. Here function `dehaze()` dehazes a given patch with the provided t and $\mathbf{A}$ using equation 6.2). The `haze_patch_comparator()` is the function for Patch Quality Comparator described in Section 6.1.2. It takes two patches as input and produces output depicting which one of the input patches is better.

Algorithm 1 t searching algorithm

```
Input: $\mathbf{I}_p$, $\mathbf{A}$, $\theta_t$
Output: $t_m$
$t_e \leftarrow 1$
$t_b \leftarrow 0$
while $(t_e - t_b) > \theta_t$ and $t_e > t_b$ do
    $t_m \leftarrow (t_e + t_b)/2$
    $\mathbf{I}_d$ = dehaze($\mathbf{I}_p$, $t_m$, $\mathbf{A}$)
    $(a, b)$ = haze_patch_comparator($\mathbf{I}_p$, $\mathbf{I}_d$)
    if $a > b$ then {dehazed patch is bad}
        $t_b \leftarrow t_m$
    else {dehazed patch is good}
        $t_e \leftarrow t_m$
    end if
end while
$t_m \leftarrow (t_e + t_b)/2$
```

6.2.3 $t(\mathbf{x})$ aggregation and interpolation

Thus the transmittance parameter t is estimated from the patches as described in the previous step. If we consider overlapping patches, a pixel is likely to receive more than one estimate of t. These values are combined to obtain a single value of t at each pixel. Apart from that, during processing we discard some patches that are either very smooth or have strong depth discontinuities depicted by presence of edge. Therefore even after aggregation it is likely that there are pixels without any estimate of t. Value of t at those pixels needs to be interpolated. Now, we can't employ any generic interpolation technique (e.g. bilinear) as they won't be able to preserve the underlying image structure that the transmittance map is expected to follow. Therefore, we apply a Laplacian based interpolation similar to Fattal [22]. The interpolated result is obtained by minimizing the following function.

$$\phi(t(\mathbf{x})) = \sum_{\mathbf{x}} s(\mathbf{x})(t(\mathbf{x}) - \tilde{t}(\mathbf{x}))^2 + \lambda \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N_{\mathbf{x}}} \frac{(t(\mathbf{x}) - t(\mathbf{y}))^2}{||I(\mathbf{x}) - I(\mathbf{y})||^2} \tag{6.7}$$

where $\tilde{t}(\mathbf{x})$ is the aggregated estimate obtained after processing the patches. $t(\mathbf{x})$ is the transmittance obtained through interpolation. $N_{\mathbf{x}}$ denotes the neighborhood of the pixel location $\mathbf{x}$. The presence map $s(\mathbf{x})$ is taken as 1 if the estimate of transmittance is available at $\mathbf{x}$ and 0 if it is not. The regularization parameter λ controls the importance between the two terms. The first term is the error term that enforces the interpolated solution to be as close as the aggregated estimate. The second term is responsible for maintaining the smoothness in the transmittance map while interpolating $t(\mathbf{x})$ from the estimates available in the neighborhood. The smoothing is performed based on $||I(\mathbf{x}) - I(\mathbf{y})||^2$. The lower its value, the more similar are the neighboring $t(\mathbf{x})$ values. So, this term ensures that transmittance map becomes smooth where the input image is smooth, while allowing it to retain sharp profile near the edges. Now for the whole image, the equation 6.7 can be written as

$$\Phi(t_o) = (t_o - \tilde{t}_a)^T S(t_o - \tilde{t}_a) + \lambda t_o^T L t_o. \tag{6.8}$$

Here $\tilde{t}_a$ is $\tilde{t}(\mathbf{x})$ in vector form. Similarly, t_o is the vector form of $t(\mathbf{x})$. S is a diagonal matrix with $s(\mathbf{x})$ as its diagonal entries. L is a laplacian matrix of the graph constructed from the input image considering each pixel as a vertex and $1/||I(\mathbf{x}) - I(\mathbf{y})||^2$ as the edge weights between pixels $\mathbf{x}$ and $\mathbf{y}$. Each vertex is connected to their neighbors. Now, minimizing equation 6.8 we obtain $t(\mathbf{x})$ for the whole image. The vector t_o that minimizes equation 6.8, is uniquely defined by the solution of the following linear equation,

$$(S + \lambda L)t_o = S\tilde{t_a}. \tag{6.9}$$

6.2.4 Haze-free image recovery

Once we obtain $t(\mathbf{x})$ at every pixel, we can dehaze the input image. We use this computed $t(\mathbf{x})$ along with the environmental illumination ($\mathbf{A}$) obtained at the first step to get the dehazed result. Using the following equation we calculate the estimated dehazed image as follows.

$$J_e^c(\mathbf{x}) = A^c - \frac{A^c - I^c(\mathbf{x})}{\max\{t(\mathbf{x}), 0.0001\}}. \tag{6.10}$$

Note that, $J_e^c(\mathbf{x})$ values lying beyond the valid intensity range are clipped to the valid range. Second, we assume $t(\mathbf{x})$ should be greater than zero, otherwise no scene information would reach the observer or the sensor (camera). To ensure this, we clip the value of t(x) arbitrarily at 0.0001 from lower. Third, unlike many other methods we do not employ any kind of post-processing technique.

6.3 Experimental Details

In this section we describe in detail the setup we have used to train our comparator and to generate the dehazed images that we report in the next section.

6.3.1 Training Data Generation

To train our patch comparator we synthesize hazy patches from clean haze-free patches. These clean patches are taken from 421 natural haze-free images. These images are member of a subset of the 500 fog-free images used by Choi et al. [15]. We have discarded some images where haze is present, specially at distant objects. We have extracted patches from these images while discarding very smooth ones and the ones with strong edges. This generates around 2.5 million patches. However, using the training data generated using all of the extracted patches, may bias the comparator. Since all kind of patches may not have similar number of samples among the extracted ones. As a result the trained network is likely to perform well for the type of patches that contains more samples than the ones with less number of samples. To alleviate this situation, we have clustered the patches taking their RGB values and their gradient in horizontal and vertical directions as the feature. The patches are clustered using k-means [9] with 1×10^5 cluster centers. Then the patches closest to the cluster centers are used as clean patches (only the RGB part) to generate the training data. Each training datum contains a patch pair and corresponding labels indicating the better patch as defined earlier. Each pair consists of a hazy patch and its dehazed version. To generate these patch pairs, we first generate a hazy patch from a clean patch using the haze imaging model (equation 6.2) with a random $t(= t')$ between 0 and 1 and two random $\mathbf{A}$. As $\mathbf{A}$ is a 3×1 vector, we take 3 random values between 0 to 1 to get a single $\mathbf{A}$. We dehaze this generated hazy patch with the corresponding $\mathbf{A}$'s and 30 different values of t. Half of the t's are greater than t' and half of them are less than it. The values of ts' are taken in such a way that the values are concentrated near t' and are sparse as we go far from it. To achieve this we first divide each range ((0, t') and (t', 1)) into 5 varying length bins with smaller bins near t' and larger bins at distant. For example, we take bins of size $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ on either half of the length. From each of the bins, however, we sample equal number of t's. As stated earlier we say the dehazed patches obtained with $t < t'$ are bad dehazed patches and dehazed patches obtained with $t > t'$ are good dehazed patches and label them accordingly. This process is done for each patch obtained from the haze-free images to generate the training data to train the comparator.

6.3.2 Parameter Settings

We have taken patches of size 10×10 to train our comparator and also to dehaze a given image. The patches with standard deviation less than 0.02 are discarded as smooth patch. To discard patches with depth discontinuity (i.e., having strong edge), we check if it contains pixels with gradient magnitude greater than 0.5. The comparator is trained with mean squared error loss for 50 epochs with batch size of 500 using the Adadelta optimizer [66]. This setup is build on Keras 1.2.2 [16] with Theano 0.9.0 [63] and libgpuarray backend. With this setup the training of the comparator is done on a machine with Intel® Core$^{\text{TM}}$ i7-4790 CPU @ 3.60GHz and Nvidia GTX 745. For computing the environmental illumination using the method of dark channel, we have taken patches of size 15×15. But for determining t from the patches, we have used 10×10 patches. For t-searching algorithm θ_t is taken to be 5×10^{-4}. Lastly for interpolation, the regularization parameter λ (see equation 6.7 and 6.8) is taken to be 0.1.

6.4 Results

We have evaluated the results obtained by our method on a variety of images using both $\mathbf{A}$ computed using the method of He et al. [28] and ground-truth $\mathbf{A}$ values when available (denoted with GT $\mathbf{A}$). The images we have used include synthetic hazy images from the dataset of [22] and D-HAZY [4], images from I-HAZE [2] and O-HAZE [3] dataset, and some real world hazy images that have been used in the previous chapters. We have compared our results with the results obtained by the methods of He et al. [28], Fattal [22], Ren et al. [53], Berman et al. [11] and Li et al. [35], as well as with the method of last chapter and Chapter 3. To compare with the method of Fattal [22] we have used the dehazed images provided by the author. The results of He et al. [28] has been generated using our own implementation of the method. For the remain methods we have used the code as provided by the respective authors in their default setting. For the images of I-HAZE [2] and O-HAZE [3] dataset, the results of He et al. [28] and Li et al. [35] could not be generated due to the resource requirement of the implementation on large images.

6.4.1 Quantitative Results

We quantitatively evaluate the results we obtain for synthetic hazy images with known ground-truth. For this purpose we have used synthetic images provided Fattal [22] and D-Hazy dataset [4]. D-Hazy dataset [4] is synthesized from Middlebury [54] dataset and NYU Depth [58] dataset. The images given by Fattal [22] contains both indoor and outdoor images, while the D-Hazy dataset [4] contains images of various indoor scenes. We quantitatively evaluate the results obtained on these images using PSNR, SSIM and CIEDE2000 metric. Here the results are reported separately for synthetic images of Fattal dataset, and Middlebury and NYU section of D-Hazy dataset.

For Fattal dataset the quantitative results are given in Table 6.2. For the first time the method of Fattal [22] has been beaten in terms of score in a few images by our proposed method when the ground truth $\mathbf{A}$ is supplied. We also see the scores takes a big leap when the ground truth $\mathbf{A}$ value is used in the dehazing process (compare the score of our and our (GT $\mathbf{A}$)). This suggests the importance of quality of the estimated $\mathbf{A}$. This has been also reflected by the ΔE_{00} score of Our (GT $\mathbf{A}$) method. So, we can say wrong estimate of $\mathbf{A}$ can make the dehazing method fail.

For the NYU section of D-Hazy dataset we only report the average values of the metrics in Table 6.1, as the dataset contains 1449 images. Surprisingly the method of He et al. [28] is performing the best in terms of the score and our method becomes a close second. The value of $\mathbf{A}$ is taken to be $[1, 1, 1]^T$ when generating the dataset and in most of the images an haze opaque region is present. This features have facilitated easy computation of $\mathbf{A}$ and transmittance using the Dark Channel Prior [28]. Hence we see the high score. Our color line based method proposed in Chapter 3 is giving the worst results, mainly due to the indoor environment of the scene and dense haze.

For the Middlebury section of the D-Hazy dataset [4] we see varied results. Table 6.4

Table 6.1: Average metrics obtained on NYU portion of D-Hazy dataset. GT **A** denotes ground truth **A** is supplied to the method.

	PSNR	SSIM	ΔE_{00}
He et al. [28]	**15.07**	**0.8**	**11.7**
Ren et al. [53]	13.03	0.76	15.45
Berman et al. [11]	13.58	0.73	15.22
Li et al. [35]	12.38	0.76	16.61
Our (ch3)	11.36	0.67	21.24
Our (ch5)	11.87	0.74	19.63
Our	14.02	0.73	13.77
Our (GT **A**)	14.17	0.78	13.10

reveals that on an average our method performs quite well, but it can perform poorly if the computed **A** is wrong. Here also, the method of He et al. [28] is performing quite well due to the fact that these hazy images are constructed in the same way as the NYU section of the dataset. The only difference is that these images are quite big (3000 × 2000 approx). The method of Ren et al. [53] has also performed well in many images. The methods proposed in Chapter 3 and 5 also has not performed well.

The scores of the images of I-HAZE [2] and O-HAZE [3] dataset as reported in Table 6.3 tell a little different story. The method proposed in Chapter 5 performs the best while the method of Ren et al. [53] becoming the second. The performance of the proposed method is not good at all. One possible reason is the huge size of hazy images. A visual inspection can reveal the issues.

Table 6.3: Quantitative Comparison on the images of I-HAZE and O-HAZE dataset. High PSNR and SSIM indicates better results, while it is the opposite for ΔE_{00}. The best results are bold and the second best results are underlined. (I) in the image column denotes indoor image whereas (O) denotes an outdoor image.

Image	Ren et al. [53]			Berman et al. [11]			Our (ch3)			Our (ch5)			Our		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
26 (I)	11.02	0.78	22.36	12.41	0.77	20.15	8.52	0.61	30.43	**15.7**	**0.87**	**13.85**	8.01	0.6	32.74
27 (I)	17.6	0.84	12.3	14.79	0.75	18.03	12.94	0.65	21.27	**21.84**	**0.87**	**8.25**	12.23	0.65	22.5
28 (I)	13.1	0.79	17.05	13.29	0.72	19.24	11.98	0.72	23.51	**16.11**	**0.82**	**13.7**	12.89	0.69	20.53
29 (I)	17.6	0.89	11.42	14.66	0.79	15.73	10.62	0.63	24.02	**21.82**	**0.91**	**9.33**	10.81	0.59	23.42
30 (I)	16.78	0.79	14.2	13.92	0.71	19.08	12.96	0.68	19.52	**20.62**	**0.83**	**12.19**	14	0.72	19.55
36 (O)	19.46	0.81	11.84	16.92	0.72	14.42	14.05	0.59	15.97	**23.11**	**0.82**	**7.6**	15.86	0.7	16.29
37 (O)	17.72	0.73	13.27	14.98	0.63	15.13	14.4	0.62	17.41	**21.32**	**0.76**	**8.52**	15.3	0.68	16.77
38 (O)	16.2	0.77	19.01	15.54	0.75	16.92	14.62	0.63	21.14	**22.25**	**0.82**	**8.51**	14.21	0.69	23.21
39 (O)	15.75	0.75	16.74	17.64	0.77	16.42	13.73	0.7	18.16	**19.9**	**0.8**	**10.84**	13.65	0.69	21
40 (O)	18.66	0.81	11.95	17.04	0.76	15.06	11.32	0.58	22.73	**22.1**	**0.84**	**7.85**	13.18	0.68	22.31
Average (I)	15.22	0.818	15.46	13.81	0.748	18.44	11.40	0.658	23.75	**19.21**	**0.86**	**11.46**	11.59	0.65	23.74
Average (O)	17.55	0.774	14.56	16.42	0.726	15.59	13.62	0.624	19.08	**21.73**	**0.81**	**8.66**	14.44	0.69	19.91

Table 6.2: Quantitative results obtained on Fattal dataset in terms of SSIM (higher the better) and CIEDE2000 (lower the better) metric. GT **A** denotes ground truth **A** is supplied to the method.

Image	He et al. [28]			Fattal [22] (GT **A**)			Ren et al. [53]		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	11.16	0.78	28.74	21.43	**0.96**	**6.34**	14.17	0.87	20.25
couch	18.4	0.86	13.89	20.8	0.9	6.71	17.99	0.88	12.91
dolls	19.73	0.85	10.65	**21.29**	0.77	**6.1**	16.93	**0.86**	12.37
flower1	14.1	0.88	23.26	**30.01**	**0.98**	**3.91**	9.08	0.43	24.64
flower2	14.37	0.86	20.94	**31.94**	**0.99**	**2.92**	10.81	0.6	22.45
lawn1	13.84	0.8	22.38	**24.49**	**0.97**	6.65	14.37	0.83	21
lawn2	11.2	0.74	29.32	**24.94**	**0.97**	6.46	13.29	0.77	22.27
mansion	17.45	0.87	19.35	**26.96**	**0.97**	**4.04**	17.69	0.89	17.52
moebius	12.66	0.78	26.7	19.01	0.9	10.61	16.36	0.9	19.85
raindeer	18.12	0.83	14.22	26.22	0.94	4.1	16.82	0.81	15.49
road1	12.95	0.8	26.11	**25.74**	**0.96**	5.24	14.11	0.84	22.22
road2	15.84	0.84	22.13	**23.6**	**0.96**	7.11	16.45	0.88	20.17
Average	14.98	0.82	21.47	**24.7**	**0.94**	**5.85**	14.84	0.8	19.26

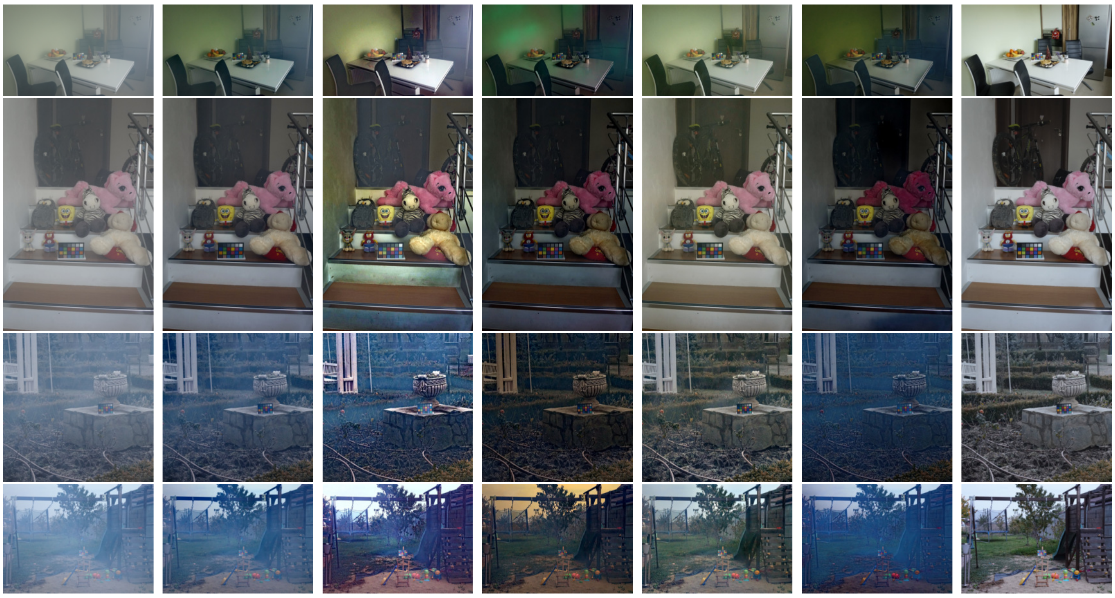
Image	Berman et al. [11]			Li et al. [35]			Our (ch3)		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	15.68	0.89	16.9	9.44	0.62	34.64	16.99	0.9	11.59
couch	17.24	0.87	14.18	16.77	0.83	17.32	16.23	0.78	15.84
dolls	15.69	0.83	15.73	17.21	0.85	10.88	13.64	0.8	18.14
flower1	12.15	0.72	20.99	12.22	0.79	29.41	19.27	0.89	11.19
flower2	11.86	0.68	21.16	13.13	0.79	25.26	21.93	0.9	7.79
lawn1	14.78	0.86	17.92	11.32	0.69	31.74	17.14	0.84	13.78
lawn2	15.32	0.87	17.8	10.98	0.68	31.7	15.79	0.8	15.9
mansion	17.33	0.87	15.83	14.24	0.7	24	19.65	0.84	9.2
moebius	14.58	0.85	22.39	13.22	0.77	27.61	18.72	0.86	12.59
raindeer	16.59	0.82	15.28	16.53	0.8	18.5	15.47	0.74	17.23
road1	16.3	0.88	19.06	11.75	0.66	29.31	15.91	0.78	14.45
road2	18.22	0.9	16.82	11.96	0.62	30.96	15.02	0.82	15.74
Average	15.48	0.84	17.84	13.23	0.73	25.94	17.15	0.83	13.62

Image	Our (ch5)			Our			Our (GT **A**)		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
church	14.46	0.9	24.39	12.46	0.84	25.42	**21.54**	0.93	7.07
couch	19.51	0.85	12.94	17.33	0.86	14.43	**27.36**	**0.95**	**3.4**
dolls	14.9	0.84	13.51	19.2	**0.86**	10.71	18.96	0.85	9.05
flower1	21.31	0.95	14.72	15.33	0.89	21.04	21.56	0.94	11.63
flower2	22.7	0.95	11.38	15.23	0.86	18.9	21.92	0.94	10.91
lawn1	16.13	0.88	20.21	14.19	0.84	21.45	24.18	0.96	**6.19**
lawn2	14.89	0.89	20.92	13.14	0.84	25.49	24.35	0.96	**6.17**
mansion	21.89	0.93	13.64	17.51	0.89	18.13	24.55	0.93	5.73
moebius	18.22	0.9	15.28	16.33	0.9	20.59	**21.11**	**0.93**	**8.9**
raindeer	22.63	0.9	10.7	18.76	0.86	13.34	**27.53**	**0.96**	**2.92**
road1	16.14	0.9	18.41	13.96	0.84	23.61	25.71	**0.96**	**4.68**
road2	15.88	0.91	20.79	15.3	0.83	22.65	23.04	0.95	**6.33**
Average	18.22	0.9	16.41	15.73	0.86	19.65	23.48	**0.94**	6.91

Table 6.4: Quantitative results obtained on Middlebury portion of D-Hazy dataset. GT **A** denotes ground truth **A** is supplied to the method.

Image	He et al. [28]			Ren et al. [53]			Berman			Berman et al. [11]		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
Adirondack	16.02	0.82	11.22	14.39	0.89	12.41	16.74	0.88	11.03	14.18	0.89	12.31
Backpack	14.4	0.85	11.32	**16.21**	0.87	9.81	12.24	0.82	14.02	16.1	**0.91**	9.67
Bicycle1	12.39	0.81	17.79	20.66	0.93	4.94	12.61	0.82	16.01	**23.21**	**0.96**	**3.84**
Cable	**12.95**	**0.7**	**16.32**	7.65	0.64	29.43	9.93	0.63	24.11	6.95	0.64	32.64
Classroom1	20.17	0.87	**6.98**	10.91	0.74	22.33	**20.95**	**0.89**	7.12	10.02	0.72	24.15
Couch	**18.68**	**0.81**	**6.1**	10.13	0.61	23.16	13.76	0.7	16.5	10.56	0.63	21.11
Flowers	**17.73**	0.89	**8.5**	10.47	0.78	21.32	17.45	**0.9**	11.74	9.25	0.76	24.24
Jadeplant	**13.48**	**0.69**	**11.79**	7.78	0.6	27.65	7.06	0.65	28.93	7.65	0.59	26.81
Mask	15.88	0.89	**9.39**	14.15	0.85	13.31	14.18	0.84	13.1	14.3	**0.91**	11.13
Motorcycle	13.81	0.79	14.29	13.2	0.81	14.89	11.6	0.62	19.15	12.25	0.82	16.52
Piano	18.66	**0.86**	6.68	12.4	0.71	17.34	15.08	0.78	15.14	13.89	0.75	13.93
Pipes	**15.52**	**0.79**	**10.57**	10.9	0.68	21.62	13.81	0.74	17.78	10.34	0.69	22.84
Playroom	17.7	**0.85**	7.89	13.42	0.77	15.07	17.64	0.83	10.1	13.24	0.78	14.24
Playtable	18.58	0.9	9.17	15.09	0.86	13.04	16.63	0.88	11.08	14.73	0.86	11.38
Recycle	12.5	0.82	17.17	**18.3**	**0.95**	7.8	13.43	0.88	14.63	16.62	0.9	8.82
Shelves	15.47	0.83	13.31	**20.43**	**0.94**	7.7	16.9	0.88	12.91	19.07	0.92	7.25
Shopvac	**13.87**	**0.8**	**13.59**	7.62	0.66	32.43	11.58	0.78	19.25	6.89	0.64	35.22
Sticks	16.96	0.9	8.87	**20.5**	**0.96**	**5.39**	20.41	0.93	7.54	19.13	**0.96**	6.3
Storage	**17.38**	**0.88**	**9.92**	11.23	0.82	18.97	16.36	**0.88**	11.51	10.24	0.79	21.48
Sword1	15.06	0.87	11.08	**15.48**	**0.91**	**10.19**	12.57	0.83	22.47	14.29	**0.91**	10.94
Sword2	**15.66**	0.89	**7.78**	12.89	0.88	13.99	14.89	0.88	14.47	12.8	0.9	13.49
Umbrella	10.4	0.8	20.73	**14.92**	0.9	12.2	9.63	0.72	27.81	14.58	**0.91**	**11.66**
Vintage	14.63	0.86	12.38	**19.27**	**0.96**	**5.31**	14.09	0.83	14.44	16.82	0.94	7.11
Average	15.56	0.83	11.43	13.82	0.81	15.66	14.33	0.81	15.69	13.35	0.82	15.96

Image	Our (ch3)			Our (ch5)			Our			Our (GT **A**)		
	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}	PSNR	SSIM	ΔE_{00}
Adirondack	11.63	0.82	17.95	13.04	0.87	16.75	15.88	0.9	10.38	**16.88**	**0.91**	**9.43**
Backpack	12.57	0.76	19.62	14.95	**0.91**	10.98	14.73	0.87	10.27	14.87	0.87	10.06
Bicycle1	14.6	0.82	12.91	18.31	0.94	8.34	20.6	0.95	5.41	20.62	0.95	5.38
Cable	5.96	0.59	37.26	6.28	0.61	37.01	8.89	0.65	25.44	8.9	0.65	25.44
Classroom1	10.1	0.74	26.62	7.03	0.64	34.16	15.98	0.75	12.01	16.01	0.75	11.99
Couch	10.77	0.66	28.62	7.76	0.55	32.78	15.63	0.75	9.34	15.65	0.75	9.3
Flowers	8.76	0.75	25.66	9.38	0.77	23.98	13.27	0.84	14.12	13.29	0.84	14.07
Jadeplant	6.3	0.55	37.42	6.37	0.54	35.85	7.36	0.64	27	7.85	0.66	23.86
Mask	11.78	0.81	17.87	13.71	0.9	13.65	**16.72**	0.89	9.72	16.68	0.89	9.78
Motorcycle	11	0.72	27.99	10.12	0.78	21.48	15.22	**0.83**	**11.79**	**15.23**	**0.83**	**11.79**
Piano	14.1	0.79	20.71	10.15	0.67	24.47	**19.75**	**0.86**	**6.26**	19.72	**0.86**	6.27
Pipes	12.82	0.69	18.65	8.65	0.65	29.49	14.38	0.77	12	14.39	0.77	11.98
Playroom	13.19	0.75	21.23	10.45	0.72	21.75	**18.22**	**0.85**	**7.6**	18.19	**0.85**	7.63
Playtable	14.56	0.84	16.7	12.05	0.81	19.25	16.58	0.87	11.97	**18.93**	**0.91**	**7.17**
Recycle	15.47	0.93	14.94	15.05	0.9	14.28	17.81	0.94	8.32	17.89	0.94	**7.78**
Shelves	16.65	0.85	15.29	13.13	0.87	16.52	19.47	0.92	7.3	19.78	0.92	**6.82**
Shopvac	6.59	0.65	35.49	8.6	0.67	29.82	10.82	0.76	19.99	10.8	0.76	20.06
Sticks	12.18	0.77	20.98	13.56	0.91	13.85	19.59	0.94	6.13	19.59	0.94	6.12
Storage	9.21	0.76	25.33	11.68	0.81	19.92	13.72	0.85	13.62	14.26	0.85	12.93
Sword1	10.14	0.73	23.99	12.57	0.88	14.01	14.04	0.85	13.45	14.76	0.87	11.52
Sword2	9.87	0.79	23.02	11.29	0.88	16.94	14.68	**0.91**	11.27	14.68	**0.91**	11.27
Umbrella	12.81	0.87	18.77	12.8	0.88	15.51	13.32	0.88	13.37	13.54	0.89	12.88
Vintage	14.23	0.84	11.49	16.93	0.94	8.98	16.92	0.92	8.33	16.96	0.92	8.3
Average	11.54	0.76	22.54	11.47	0.79	20.86	15.37	0.84	11.96	**15.63**	**0.85**	**11.38**



(a) Input (b) Ren et al. [53] (c) Berman et al. [11] (d) Our (ch3) (e) Our (ch5) (f) Our (g) Ground Truth

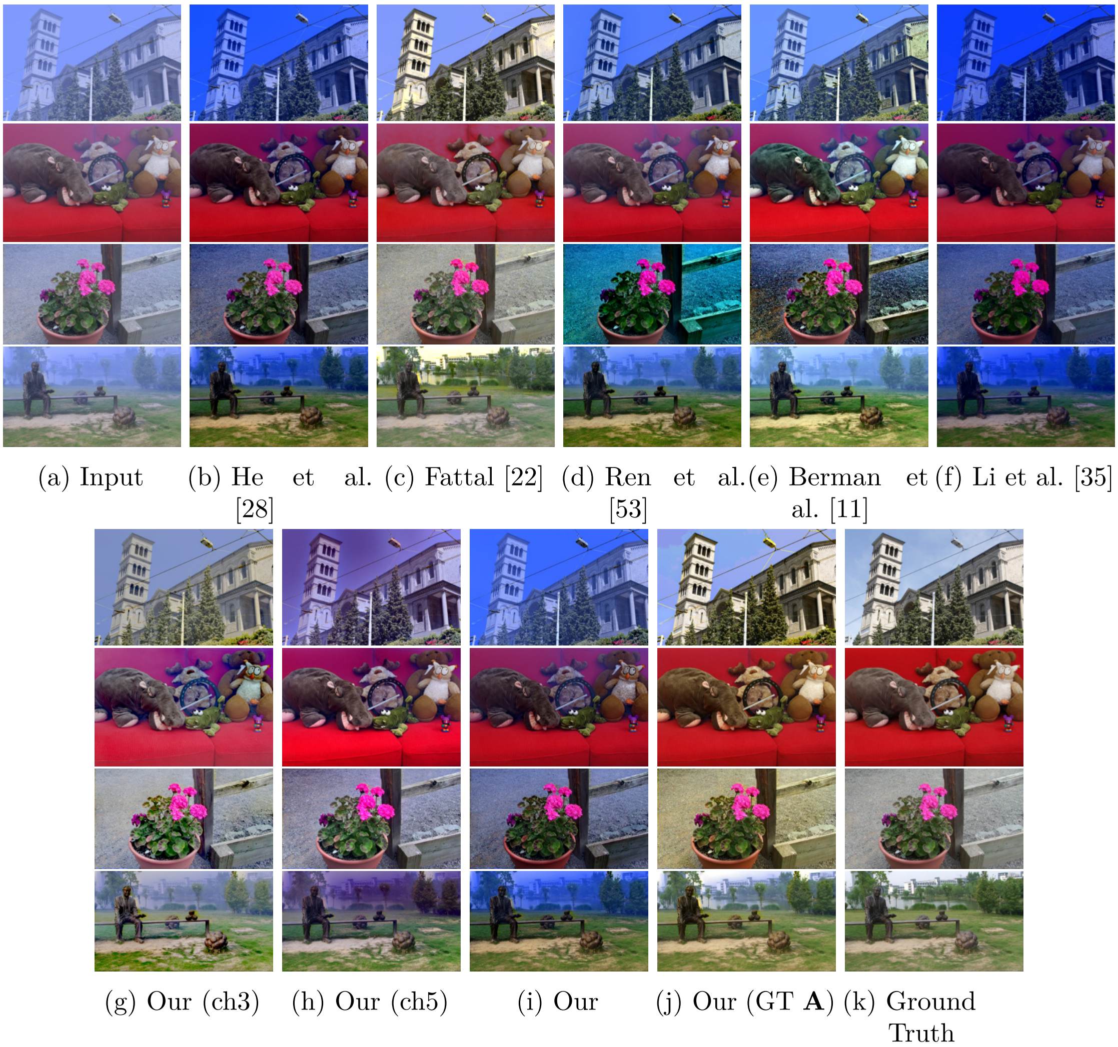
Figure 6.4: Visual comparison of the results on two images of I-HAZE and two images of O-HAZE dataset

6.4.2 Qualitative Results

In this section now we visually compare the results obtained by different methods on different images. Starting off with the four synthetic images of Fattal [22] (figure 6.5), it is seen that color bias exists in the dehazed images obtained by He et al. [28], Ren et al. [53], Berman et al. [11], and Li et al. [35] and also in our method (with **A** estimated using Dark Channel Prior). This suggests that wrong estimation of atmospheric light may introduce color bias in the dehazed output. This does not happen if the actual A is supplied to the dehazing method (e.g. in the method of Fattal [22] and our (GT **A**)). This is reflected in the CIEDE2000 values for the corresponding output images. In some cases the results of our (GT **A**) is a bit yellowish than that of Fattal [22]. For the NYU part of the D-Hazy dataset [4], we show 4 results in figure 6.7. As indicated by the quantitative scores, the results of He et al. [28] is the cleanest followed by our proposed method. The method of Ren et al. [53] and Li et al. [35] could clear the haze properly. The method of Berman et al. [11] has performed better than these two. On the other hand our method proposed in Chapter 3 has wrongly estimated the **A** which has resulted in color distortions. For the Middlebury section of D-Hazy dataset [4], we have illustrated the results with four images (figure 6.6): *Piano*, *Bicycle1*, *Motorcycle*, and *Flowers*. The results show that the method of Ren et al. [53] and Li et al. [35] have failed to dehaze completely, particularly when the value of t is relatively low. For *Motorcycle* and *Bicycle1* images, the floor is dehazed more by our method because of it's similarity in color with airlight. The results also show that the method of Berman et al. [11] tends to over-enhance the results a bit. Similar to what happens in the NYU section, the method proposed in Chapter 3 has distorted the colors in the output. Now if we move our attention to the images of I-HAZE Ancuti et al. [2] and O-HAZE Ancuti et al. [3] dataset, we only the method proposed in the last chapter is doing well. The currently proposed method is able to clear the haze in the indoor images, but not in the outdoor images. Apart from that the intensity of the images has reduced quite a bit. This has negatively affected the scores, even though the method has cleaned the haze to some extend. The method of Chapter 3 has again distorted the colors. Among the remaining the methods Berman et al. [11] has performed well in terms of clearing the haze, but the method of Ren et al. [53] is not able to do that. Lastly, in figure 6.8 we have shown the results of the dehazing methods on some real world images. From the results it is seen that the method of Berman et al. [11] has a tendency to over-enhance the results. This enhancement is more controlled in the results of Fattal [22]. The method of Ren et al. [53] is able to clear the haze partially, especially in *tiananmen* and *ny12* images. Li et al. [35] has performed a little better except in the *tiananmen* image. Among our methods the results of the color line based method (Chapter 3) looks more pleasing. The method proposed in the previous chapter has retained the brightness of the results but also some haze. The results of our method is a bit similar to results of He et al. [28], because we have utilized the technique proposed by He et al. [28] to compute the environmental illumination.

6.5 Summary

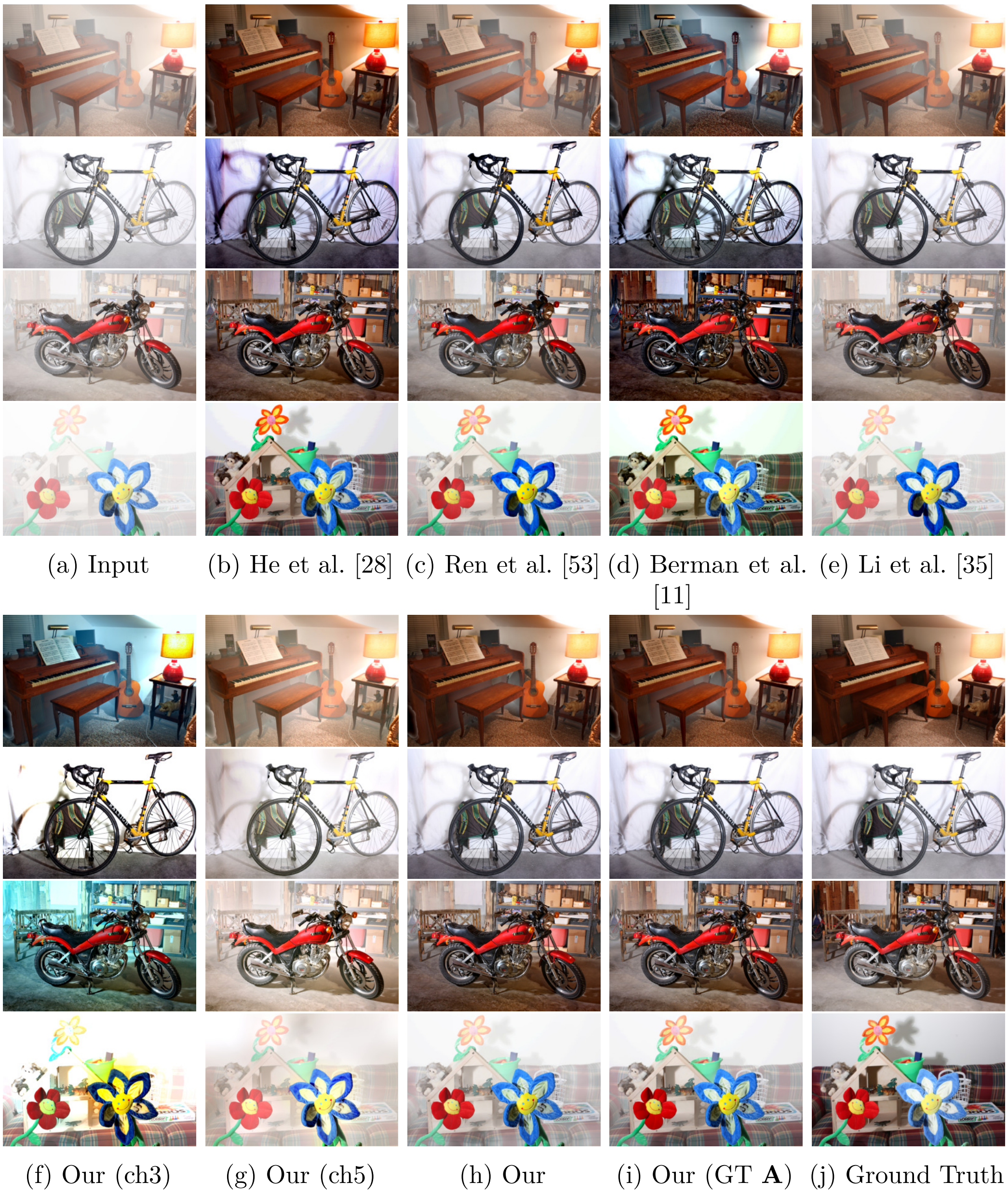
In this chapter we have proposed an image dehazing method that tries to estimate transmittance in each patch by comparing the dehazed version of the input image with the input hazy one. The comparison is done by our proposed *patch quality comparator*. With this CNN based comparator in our hand, we employ binary search to find transmittance in each patch. Although we have used the method of Dark Channel Prior [28] to compute environmental illumination, the results show it is not always accurate. The output greatly improves with correct environmental illumination. This shows that the environmental illumination is crucial in dehazing an image, although it has not received the required attention. The future work could be focused on accurate estimation of environmental illumination for both day and night time cases.



(a) Input (b) He et al. [28] (c) Fattal [22] (d) Ren et al. [53] (e) Berman et al. [11] (f) Li et al. [35]

(g) Our (ch3) (h) Our (ch5) (i) Our (j) Our (GT **A**) (k) Ground Truth

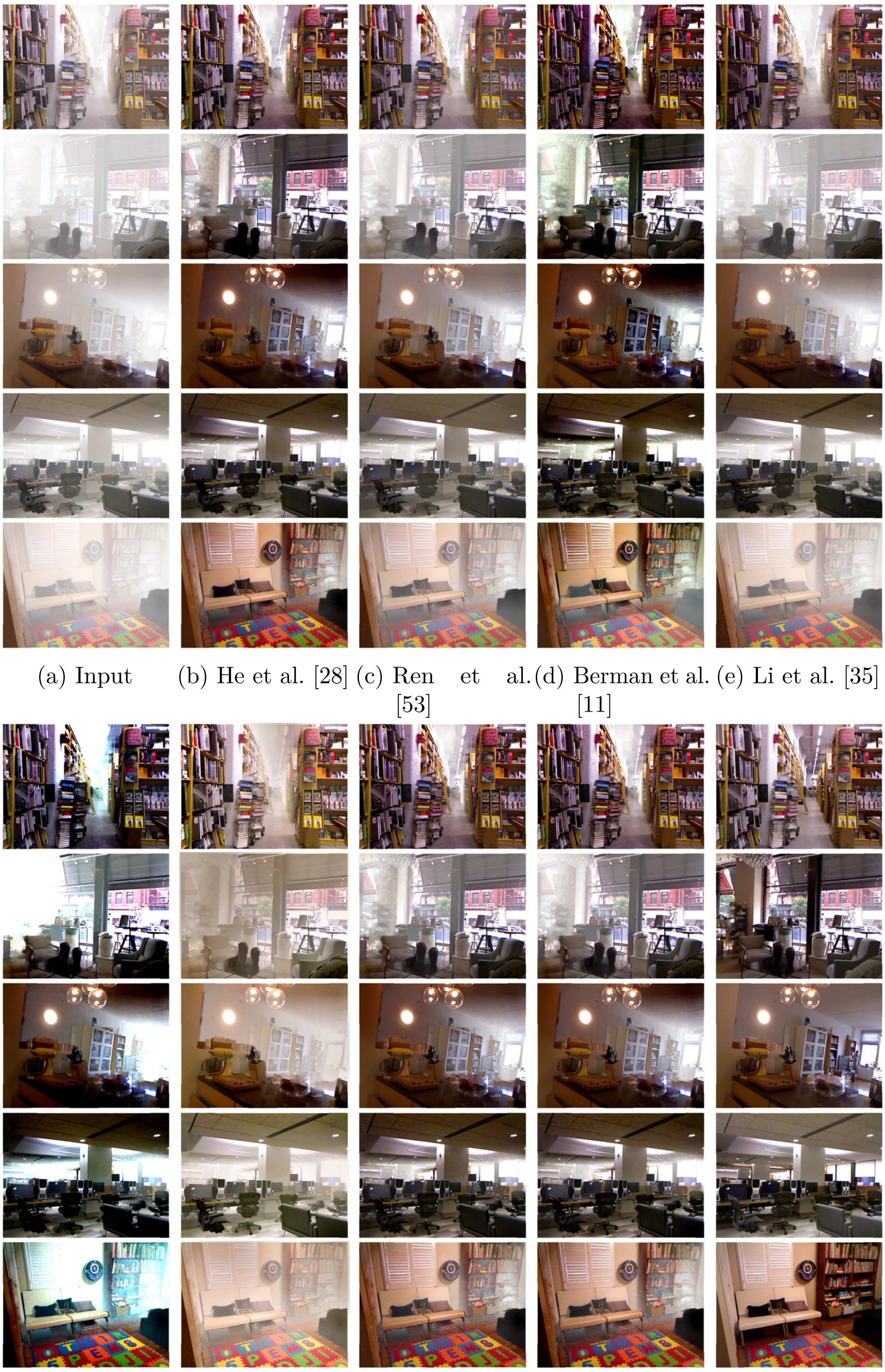
Figure 6.5: Visual comparison of the results on four synthetic images: *church*, *couch*, *flower2*, and *lawn1*.



(a) Input (b) He et al. [28] (c) Ren et al. [53] (d) Berman et al. [11] (e) Li et al. [35]

(f) Our (ch3) (g) Our (ch5) (h) Our (i) Our (GT **A**) (j) Ground Truth

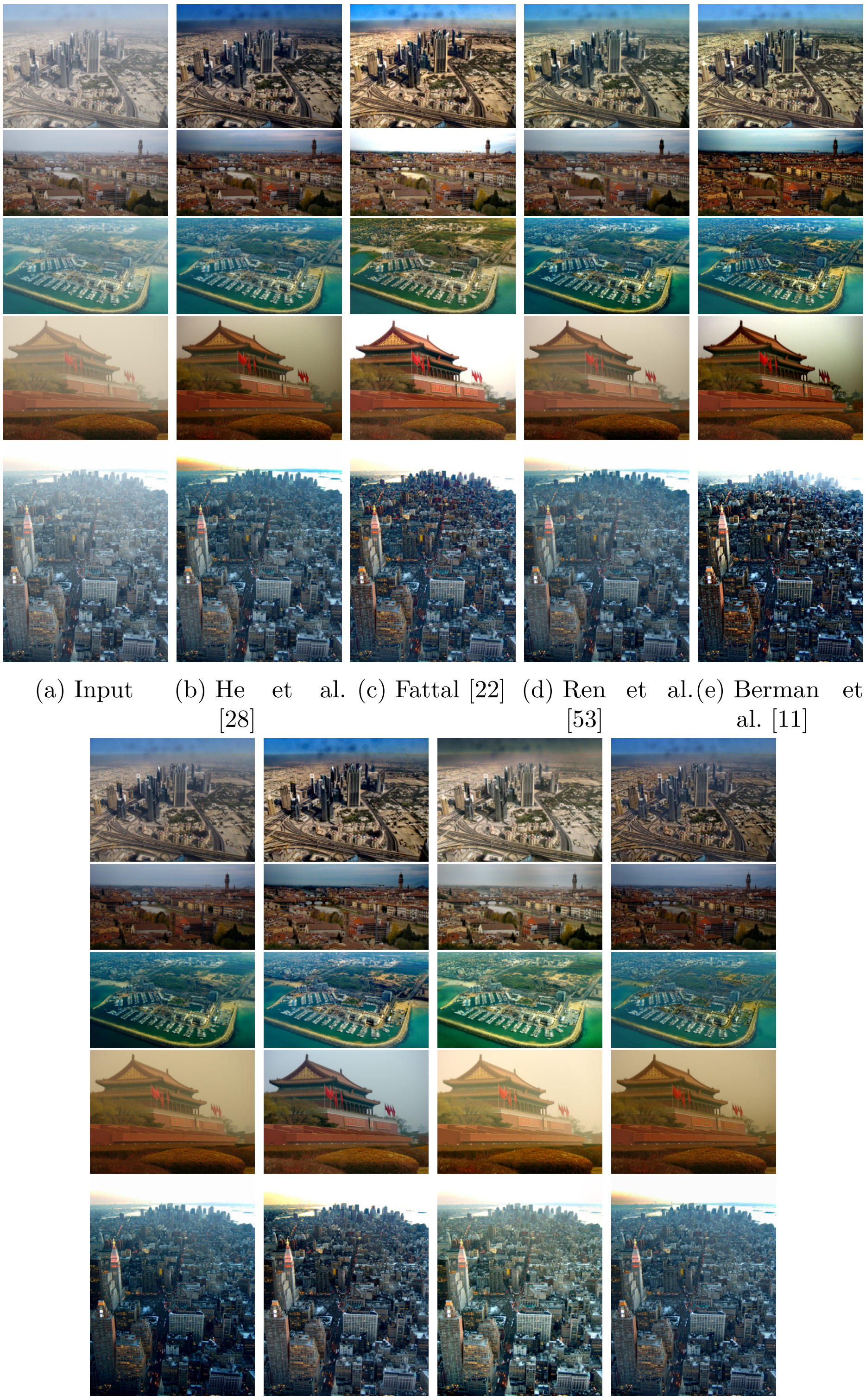
Figure 6.6: Visual comparison of the results of Middlebury portion of the D-Hazy dataset on *Piano*, *Bicycle1*, *Motorcycle*, and *Flowers*.



(a) Input (b) He et al. [28] (c) Ren et al. [53] (d) Berman et al. [11] (e) Li et al. [35]

(f) Our (ch3) (g) Our (ch5) (h) Our (i) Our (GT **A**) (j) Ground Truth

Figure 6.7: Visual comparison of some Results of NYU portion of D-Hazy dataset



(a) Input (b) He et al. [28] (c) Fattal [22] (d) Ren et al. [53] (e) Berman et al. [11]

(f) Li et al. [35] (g) Our (ch3) (h) Our (ch5) (i) Our

Figure 6.8: Visual comparison of results on *dubai*, *florence*, *herzeliya*, *tiananmen*, and *ny12* image.

Chapter 7

Conclusion

In this thesis the problem of image dehazing has been explored with an emphasis on removing the effects of environmental illumination using a variety of tool and techniques. In Chapter 1, we begin with describing the problem of image dehazing and the atmospheric scattering model that presents how images are formed under fog/haze condition. We have also discussed different state-of-the-art methods that exists in the literature.

In chapter 2, the assumption of constant environmental illumination is relaxed, so that we can handle the cases when the intensity of the illumination changes over a scene. This situation may occur if the sky is not cloudy and direct sunlight is illuminating some part of the scene. We have shown that a simple extension of the color line based dehazing can handle such cases very well. For that we have utilized the color line model to estimate the intensity of environmental illumination locally from multiple patches of the image, in addition to estimating its color. Though the transmittance is not computed explicitly, the computed airlight along with local contrast enhancement is able to recover a dehazed version of the input image. Since the proposed method assumes that within an image the color of environmental illumination is constant but its magnitude may vary, the proposed method is likely to fail in producing satisfactory results where this assumption is violated, e.g., night-time hazy images. So, in the next chapter (Chapter 3) the imaging model is further relaxed. The assumption of uniform color of environmental illumination is relaxed to a spatially variant one. That means over an image both the intensity and color of the environmental illumination may vary. This relaxed model has enabled the proposal of a image dehazing method that works for both night-time and daytime images. Although the relaxed version suggests that $\mathbf{A}(x)$ can vary from pixel to pixel, in reality it varies very slowly and may remain constant over significant large regions. So the number of unique colors of environmental illumination becomes much less than the total number of patches. Thus we can find the possible colors of the environmental illumination using color line model and Hough transform. These colors are then utilized to calculate the airlight component in the patches of the image. We did not attempt explicit computation of $t(\mathbf{x})$, since after the relaxation, the number of unknowns (parameters) becomes even more than the number of linear equations. It is observed that simple removal of the airlight reduces the pixel intensity significantly, and the objects with color similar to airlight becomes dark after dehazing. For this reason applying contrast enhancement as a post-processing step becomes necessary. It is noted that this enhancement may not work satisfactorily in all images and gamma correction may be necessary in isolated cases. The resolution of the Hough space and the inaccuracies introduced by the interpolation also impacts the dehazed output. The major problem of these two color line based methods is the total number of user tunable parameters. The method utilizes around 10 parameters as thresholds and weights. The proposed method works well in many images in the default values of the parameters, but the default values may not be ideal for all kind of images. But, choosing right parameter values can potentially improve the results. However, making adjustments to these parameters is not straight forward and needs lots of tedious effort. For example, lowering the threshold allows more patches in the estimation step, but can potentially allow incorrect estimates. So, from chapter 4 onward we have explored the use of Convolutional Neural Networks (CNN) for automated learning of hazing related parameters.

In Chapter 4, as the first step, we have estimated scene transmittance and environmental illumination jointly from small image patches using a convolutiona neural network (CNN). The estimate obtained from the patches are then aggregated and utilized to obtain the dehazed image. The results have shown that the CNN is able to estimate both $t(\mathbf{x})$ and $\mathbf{A}$ effectively from patches based on the given training data. Major advantage is it does not require any user specified sensitive explicit parameters during run time. But the method has its own share of pitfalls. It is observed that the average color of the patch is getting predicted as the environmental illumination in many cases. This error is much more prominent in those patches where the color does not vary much. The issue lies with the size of the patches. Given a small patch it is always difficult to distinguish whether the colors are due to an object or the illumination. On the other hand using bigger patches give rise to another issue. Let us describe it briefly. We have used an indoor image dataset with known depth maps (NYU v2) to synthetically generate the training data. The trained network is applied on outdoor images, although the image statistics of indoor and outdoor images vary quite a bit. In smaller patches (scale) it did not had much effect. But when using bigger patches this can become significant. So, in the next chapter we use two separate datasets (I-HAZE and O-HAZE) to train our network. The said image dataset is quite big and contains only hazy and haze-free image pairs. It does not contain any groundtruth transmittance or environmental illumination. So, the network is trained using these hazy and haze-free image pairs. To achieve that we have proposed a new loss (called *Bi-directional Consistency Loss*), that makes the network produce haze-free images that conforms to generate hazy image using the imaging model. Here a Fully Convolutional Neural Network (FCN) is utilized to predict the transmittance and airlight from bigger patches. The use of bigger patches has been enabled by the FCN as this type of network can produce output that has same size as the input. Apart from that, it is also observed that multi-scale approach is able to overcome the problems of fixed scale in CNNs in dehazing. However, effectiveness of this multi-scale approach needs to be validated in other situations. Although the network performs well in the validation images of the I-HAZE [2] and O-HAZE [3] datasets, it does not perform well in the other images. That means the network trained on this data is not well generalized. All the methods of dehazing described so far relies on the quality evaluation of dehazed image. We argue that given an image (hazy or dehazed) it is difficult to evaluate the quality of the image independently. On the other hand, given two versions of the same image it is much easier as well as reliable to decide which one is better in terms haze content and thus leads to an image dehazing method. The idea is implemented and validated through a patch comparator using CNN in Chapter 6. Here small patches are taken so that assumption of constant transmittance within a patch remains valid. Thus we search out the value of transmittance that properly dehaze the given patch with the assumption that correct **A** is known before hand. Hence, instead of training a network to predict the transmittance given a hazy patch, we are utilizing the network to compare two patches in terms of haziness. This comparison is relatively easier and using this we obtain a transmittance that does not overdo the dehazing. The results have shown that a wrong estimate of environmental illumination can result in wrong estimate of the transmittance. So, the environmental illumination is crucial in image dehazing and it should be estimated properly.

7.1 Future scope of work

From this thesis, it is seen that hand crafted features work well, but they can be restrictive in practice. The CNNs have an advantage in this regard due to their automated feature computation. But these networks feed on data. If the data is not proper the results can deviate badly from the desired ones.

- So, the design of data agnostic methods, to the extent possible, could be a possible future direction of work.

Although this problem is common for neural networks, but availability of large datasets for other computer vision tasks have remedied the problem to some extent. But in the context of image dehazing, collecting pairs of hazy and haze-free image of the same scene is quite hard if not impossible, because the environmental conditions can change between the two situations. Nonetheless, there have been efforts to make such datasets in real conditions (e.g. I-HAZE [2] and O-HAZE [3]) apart from the synthetic datasets that are available. But creating such a dataset for night-time hazy scenes is much more harder. As a result, currently there are no known night-time hazy image dataset, synthetic or otherwise. Hence, the CNN based methods focus on the dehazing of daytime images only. Other than that, the importance of environmental illumination in the estimation of transmittance is observed.

- So, research efforts should be made in the direction of reliable and systematic estimation of environmental illumination.
- Another possible direction is questioning the imaging model itself for hazy image formation.

This model has been formulated by the physicists [32], and then taken up and have been analyzed by the pioneers of this field. But after that this model has been used almost blindly followed without questioning the validity of the model and the assumptions made to reach at the current version of the model. As, it has been shown it is not valid in all kind of scenarios. So, the model should be looked at to have a better understanding of the problem and to really improve the results. Finally, it is seen in the previous chapter (Chapter 6) that the results are best among all the proposed methods. However, it assumes that environmental illumination is uniform over the whole scene and has unique color. So the patch comparator based dehazing method should be upgraded to accommodate non-uniform illumination intensity and variable colors over scene.

References

[1] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[2] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. "I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images". In: *ArXiv e-prints* (Apr. 2018). arXiv: `1804.05091 [cs.CV]`.

[3] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. "O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images". In: *ArXiv e-prints* (Apr. 2018). arXiv: `1804.05101 [cs.CV]`.

[4] C. Ancuti, C. O. Ancuti, and C. D. Vleeschouwer. "D-HAZY: A dataset to evaluate quantitatively dehazing algorithms". In: *2016 IEEE International Conference on Image Processing (ICIP)*. Sept. 2016, pp. 2226–2230. DOI: `10.1109/ICIP.2016.7532754`.

[5] C. Ancuti, C. O. Ancuti, C. D. Vleeschouwer, and A. C. Bovik. "Night-time dehazing by fusion". In: *2016 IEEE International Conference on Image Processing (ICIP)*. Sept. 2016, pp. 2256–2260. DOI: `10.1109/ICIP.2016.7532760`.

[6] C. Ancuti and C. Ancuti. "Single Image Dehazing by Multi-Scale Fusion". In: *IEEE Transactions on Image Processing* 22.8 (Aug. 2013), pp. 3271–3282. ISSN: 1057-7149. DOI: `10.1109/TIP.2013.2262284`.

[7] C. O. Ancuti, C. Ancuti, C. Hermans, and P. Bekaert. "A Fast Semi-inverse Approach to Detect and Remove the Haze from a Single Image". en. In: *Computer Vision – ACCV 2010*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Nov. 2010, pp. 501–514. ISBN: 978-3-642-19308-8. DOI: `10.1007/978-3-642-19309-5_39`.

[8] C. Ancuti et al. "NTIRE 2018 Challenge on Image Dehazing: Methods and Results". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2018, pp. 1004–100410. DOI: `10.1109/CVPRW.2018.00134`.

[9] D. Arthur and S. Vassilvitskii. “K-means++: The Advantages of Careful Seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’07. event-place: New Orleans, Louisiana. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 978-0-89871-624-5.

[10] Y. Bahat and M. Irani. “Blind dehazing using internal patch recurrence”. In: *2016 IEEE International Conference on Computational Photography (ICCP)*. May 2016, pp. 1–9. DOI: `10.1109/ICCPHOT.2016.7492870`.

[11] D. Berman, T. Treibitz, and S. Avidan. “Non-local Image Dehazing”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 1674–1682. DOI: `10.1109/CVPR.2016.185`.

[12] D. Berman, T. Treibitz, and S. Avidan. “Air-light estimation using haze-lines”. In: *2017 IEEE International Conference on Computational Photography (ICCP)*. Apr. 2017, pp. 1–9. DOI: `10.1109/ICCPHOT.2017.7951489`.

[13] C. Bishop. *Pattern Recognition and Machine Learning*. en. Information Science and Statistics. New York: Springer-Verlag, 2006. ISBN: 978-0-387-31073-2.

[14] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. “DehazeNet: An End-to-End System for Single Image Haze Removal”. In: *IEEE Transactions on Image Processing* 25.11 (Nov. 2016), pp. 5187–5198. ISSN: 1057-7149. DOI: `10.1109/TIP.2016.2598681`.

[15] L. K. Choi, J. You, and A. Bovik. “Referenceless Prediction of Perceptual Fog Density and Perceptual Image Defogging”. In: *IEEE Transactions on Image Processing* 24.11 (Nov. 2015), pp. 3888–3901. ISSN: 1057-7149. DOI: `10.1109/TIP.2015.2456502`.

[16] F. Chollet et al. *Keras*. `https://keras.io`. 2015.

[17] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)”. In: *arXiv e-prints*, arXiv:1511.07289 (Nov. 2015), arXiv:1511.07289. arXiv: `1511.07289 [cs.LG]`.

[18] C. Dong, C. C. Loy, K. He, and X. Tang. “Learning a Deep Convolutional Network for Image Super-Resolution”. en. In: *Computer Vision – ECCV 2014*. Lecture Notes in Computer Science. Springer, Cham, Sept. 2014, pp. 184–199. ISBN: 978-3-319-10592-5. DOI: `10.1007/978-3-319-10593-2_13`.

[19] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159. ISSN: ISSN 1533-7928.

[20] V. Dumoulin and F. Visin. “A guide to convolution arithmetic for deep learning”. In: *arXiv e-prints*, arXiv:1603.07285 (Mar. 2016), arXiv:1603.07285. arXiv: `1603.07285 [stat.ML]`.

[21] R. Fattal. “Single Image Dehazing”. In: *ACM SIGGRAPH 2008 Papers*. SIGGRAPH ’08. New York, NY, USA: ACM, 2008, 72:1–72:9. ISBN: 978-1-4503-0112-1. DOI: `10.1145/1399504.1360671`.

[22] R. Fattal. “Dehazing Using Color-Lines”. In: *ACM Trans. Graph.* 34.1 (Dec. 2014), 13:1–13:14. ISSN: 0730-0301. DOI: `10.1145/2651362`.

[23] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: `10.1145/358669.358692`.

[24] A. Galdran, J. Vazquez-Corral, D. Pardo, and M. Bertalmío. “A Variational Framework for Single Image Dehazing”. en. In: *Computer Vision - ECCV 2014 Workshops*. Ed. by L. Agapito, M. M. Bronstein, and C. Rother. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 259–270. ISBN: 978-3-319-16199-0.

[25] K. B. Gibson and T. Q. Nguyen. “An analysis of single image defogging methods using a color ellipsoid framework”. en. In: *EURASIP Journal on Image and Video Processing* 2013.1 (Dec. 2013), p. 37. ISSN: 1687-5281. DOI: `10.1186/1687-5281-2013-37`.

[26] N. Hautiére, J.-P. Tarel, J. Lavenant, and D. Aubert. “Automatic fog detection and estimation of visibility distance through use of an onboard camera”. en. In: *Machine Vision and Applications* 17.1 (Apr. 2006), pp. 8–20. ISSN: 1432-1769. DOI: `10.1007/s00138-005-0011-1`.

[27] K. He, J. Sun, and X. Tang. “Guided Image Filtering”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (June 2013), pp. 1397–1409. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2012.213`.

[28] K. He, J. Sun, and X. Tang. “Single Image Haze Removal Using Dark Channel Prior”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (Dec. 2011), pp. 2341–2353. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2010.168`.

[29] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[30] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *arXiv e-prints*, arXiv:1502.03167 (Feb. 2015), arXiv:1502.03167. arXiv: `1502.03167 [cs.LG]`.

[31] J. Kopf et al. “Deep Photo: Model-based Photograph Enhancement and Viewing”. In: *ACM SIGGRAPH Asia 2008 Papers*. SIGGRAPH Asia ’08. New York, NY, USA: ACM, 2008, 116:1–116:10. ISBN: 978-1-4503-1831-0. DOI: `10.1145/1457515.1409069`.

[32] H. Koschmieder. “Theorie der horizontalen Sichtweite”. In: *Beitrage zur Physik der freien Atmosphare* (1924), pp. 33–53.

[33] P. D. Kovesi. *MATLAB and Octave Functions for Computer Vision and Image Processing*. Available from: <http://www.peterkovesi.com/matlabfns/>. 2000.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1097–1105.

[35] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. “AOD-Net: All-In-One Dehazing Network”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4770–4778.

[36] Y. Li, R. T. Tan, and M. S. Brown. “Nighttime Haze Removal with Glow and Multiple Light Colors”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 226–234. DOI: `10.1109/ICCV.2015.34`.

[37] Y. Li, S. You, M. S. Brown, and R. T. Tan. “Haze visibility enhancement: A Survey and quantitative benchmarking”. In: *Computer Vision and Image Understanding* 165 (Dec. 2017), pp. 1–16. ISSN: 1077-3142. DOI: `10.1016/j.cviu.2017.09.003`.

[38] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2015, pp. 3431–3440. DOI: `10.1109/CVPR.2015.7298965`.

[39] K. Ma, W. Liu, and Z. Wang. “Perceptual evaluation of single image dehazing algorithms”. In: *2015 IEEE International Conference on Image Processing (ICIP)*. Sept. 2015, pp. 3600–3604. DOI: `10.1109/ICIP.2015.7351475`.

[40] E. J. McCartney. *Optics of the atmosphere: Scattering by molecules and particles*. New York: John Wiley and Sons, Inc., 1976.

[41] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. “Efficient Image Dehazing with Boundary Constraint and Contextual Regularization”. In: *2013 IEEE International Conference on Computer Vision*. Dec. 2013, pp. 617–624. DOI: 10.1109/ICCV.2013.82.

[42] W. E. K. Middleton. “Vision through the Atmosphere”. en. In: *Geophysik II / Geophysics II*. Ed. by J. Bartels. Handbuch der Physik / Encyclopedia of Physics. Berlin, Heidelberg: Springer, 1957, pp. 254–287. ISBN: 978-3-642-45881-1. DOI: 10.1007/978-3-642-45881-1_3.

[43] S. G. Narashiman and S. K. Nayar. “Interactive deweathering of an image using physical model”. In: *IEEE Workshop on color and photometric Methods in computer Vision*. 2003.

[44] S. G. Narasimhan and S. K. Nayar. “Contrast restoration of weather degraded images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.6 (June 2003), pp. 713–724. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2003.1201821.

[45] S. G. Narasimhan and S. K. Nayar. “Vision and the Atmosphere”. en. In: *International Journal of Computer Vision* 48.3 (July 2002), pp. 233–254. ISSN: 0920-5691, 1573-1405. DOI: 10.1023/A:1016328200723o.

[46] K. Nishino, L. Kratz, and S. Lombardi. “Bayesian Defogging”. en. In: *International Journal of Computer Vision* 98.3 (July 2012), pp. 263–278. ISSN: 1573-1405. DOI: 10.1007/s11263-011-0508-1.

[47] J. Oakley and H. Bu. “Correction of Simple Contrast Loss in Color Images”. In: *IEEE Transactions on Image Processing* 16.2 (Feb. 2007), pp. 511–522. ISSN: 1057-7149. DOI: 10.1109/TIP.2006.887736.

[48] I. Omer and M. Werman. “Color lines: image specific color representation”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. Vol. 2. Washington, DC, USA, June 2004. DOI: 10.1109/CVPR.2004.1315267.

[49] M. Pedone and J. Heikkilä. “Robust airlight estimation for haze removal from a single image”. In: *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2011, pp. 90–96. DOI: 10.1109/CVPRW.2011.5981822.

[50] S. Pei and T. Lee. “Nighttime haze removal using color transfer pre-processing and Dark Channel Prior”. In: *2012 19th IEEE International Conference on Image Processing*. Sept. 2012, pp. 957–960. DOI: 10.1109/ICIP.2012.6467020.

[51] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 779–788. DOI: `10.1109/CVPR.2016.91`.

[52] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 91–99.

[53] W. Ren et al. "Single Image Dehazing via Multi-scale Convolutional Neural Networks". en. In: *Computer Vision - ECCV 2016*. Lecture Notes in Computer Science. Springer, Cham, Oct. 2016, pp. 154–169. DOI: `10.1007/978-3-319-46475-6_10`.

[54] D. Scharstein et al. "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth". en. In: *Pattern Recognition*. Lecture Notes in Computer Science. Springer, Cham, Sept. 2014, pp. 31–42. ISBN: 978-3-319-11751-5. DOI: `10.1007/978-3-319-11752-2_3`.

[55] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar. "Instant dehazing of images using polarization". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. Dec. 2001, pp. I–I. DOI: `10.1109/CVPR.2001.990493`.

[56] G. Sharma, W. Wencheng, and E. N. Dalal. "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations". In: *Color Research & Application* 30.1 (Dec. 2004), pp. 21–30. ISSN: 0361-2317. DOI: `10.1002/col.20070`.

[57] S. Shwartz, E. Namer, and Y. Schechner. "Blind Haze Separation". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 1984–1991. DOI: `10.1109/CVPR.2006.71`.

[58] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. "Indoor Segmentation and Support Inference from RGBD Images". en. In: *Computer Vision - ECCV 2012*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Oct. 2012, pp. 746–760. DOI: `10.1007/978-3-642-33715-4_54`.

[59] M. Sulami, I. Glatzer, R. Fattal, and M. Werman. "Automatic recovery of the atmospheric light in hazy images". In: *2014 IEEE International Conference on Computational Photography (ICCP)*. May 2014, pp. 1–11. DOI: `10.1109/ICCPHOT.2014.6831817`.

[60] R. T. Tan. "Visibility in bad weather from a single image". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. June 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587643.

[61] K. Tang, J. Yang, and J. Wang. "Investigating Haze-Relevant Features in a Learning Framework for Image Dehazing". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014, pp. 2995–3002. DOI: 10.1109/CVPR.2014.383.

[62] J.-P. Tarel et al. "Vision Enhancement in Homogeneous and Heterogeneous Fog". In: *IEEE Intelligent Transportation Systems Magazine* 4.2 (2012), pp. 6–20. ISSN: 1939-1390. DOI: 10.1109/MITS.2012.2189969.

[63] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016).

[64] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[65] Q. Yan, L. Xu, and J. Jia. "Dense Scattering Layer Removal". In: *SIGGRAPH Asia 2013 Technical Briefs*. SA '13. New York, NY, USA: ACM, 2013, 14:1–14:4. ISBN: 978-1-4503-2629-2. DOI: 10.1145/2542355.2542373.

[66] M. D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: *arXiv e-prints*, arXiv:1212.5701 (Dec. 2012), arXiv:1212.5701. arXiv: 1212.5701 [cs.LG].

[67] J. Zhang, Y. Cao, and Z. Wang. "Nighttime haze removal based on a new imaging model". In: *2014 IEEE International Conference on Image Processing (ICIP)*. Oct. 2014, pp. 4557–4561. DOI: 10.1109/ICIP.2014.7025924.

[68] Q. Zhu, J. Mai, and L. Shao. "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior". In: *IEEE Transactions on Image Processing* 24.11 (Nov. 2015), pp. 3522–3533. ISSN: 1057-7149. DOI: 10.1109/TIP.2015.2446191.

List of Publications

[1] S. Santra and B. Chanda. "Single image dehazing with varying atmospheric light intensity". In: *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*. Dec. 2015, pp. 1–4. DOI: 10.1109/NCVPRIPG.2015.7490015.

[2] S. Santra and B. Chanda. "Day/night unconstrained image dehazing". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. Dec. 2016, pp. 1406–1411. DOI: 10.1109/ICPR.2016.7899834.

[3] S. Santra, R. Mondal, P. Panda, N. Mohanty, and S. Bhuyan. "Image Dehazing via Joint Estimation of Transmittance Map and Environmental Illumination". In: *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*. Dec. 2017, pp. 1–6. DOI: 10.1109/ICAPR.2017.8593161.

[4] R. Mondal, S. Santra, and B. Chanda. "Image Dehazing by Joint Estimation of Transmittance and Airlight Using Bi-Directional Consistency Loss Minimized FCN". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2018, pp. 1033–10338. DOI: 10.1109/CVPRW.2018.00137.

[5] S. Santra, R. Mondal, and B. Chanda. "Learning a Patch Quality Comparator for Single Image Dehazing". In: *IEEE Transactions on Image Processing* 27.9 (Sept. 2018), pp. 4598–4607. ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2841198.