UNIVERSITY OF CALIFORNIA
SANTA CRUZ

SINGLE IMAGE HAZE AND NOISE REMOVAL

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Erik Matlin

June 2011

The Thesis of Erik Matlin
is approved:

Professor Peyman Milanfar, Chair

Professor James Davis

Professor John Vesecky

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Single Image Haze and Noise Removal

by

Erik Matlin

One of the central problems in image processing is the restoration of images corrupted by various types of degradation. Images of outdoor scenes often contain atmospheric degradation, such as haze and fog, caused by particles in the atmospheric medium absorbing and scattering light as it travels to the observer. Although this effect may be desirable from an artistic standpoint, for a variety of reasons (such as computer vision or sensing algorithms, or even pure curiosity) one may need to restore an image corrupted by these effects, a process generally referred to as haze removal. Additionally, all images contain some corruption from noise, and if not considered in the haze removal process, noise can dominate the results.

Accordingly, the task of this thesis is to present an effective method for removing both haze and noise from a single digital image. First is an investigation on the effect of noise on an existing single image haze estimation method, with denoising proposed as a pre-processing step. Additionally, existing methods for refining the haze estimation for the purpose of reducing artifacts are reviewed. Next, two different methods for removing haze and noise are proposed. The first approach is the process of denoising the image using a state-of-the-art denoising algorithm prior to dehazing. The second approach involves simultaneously denoising and dehazing using an iterative non-parametric kernel regression based method. Experimental results for both methods are compared.

Findings show that when the noise level is known a priori, simply denoising prior to dehazing outperforms the iterative method in terms of mean squared error, although results are visually comparable. In contrast, when the precise noise level is not given, and the denoising algorithm must be tuned using an external quality metric, latent errors from either “under"-denoising or “over"-denoising are amplified in the dehazing process. In this situation, the iterative approach can yield superior results, both in terms of mean squared error and subjective visual quality, especially in low noise cases.

Dedicated to my Mom.

Acknowledgments

First of all, I would like to thank my advisor, Professor Peyman Milanfar. He has been present throughout my entire journey at UC Santa Cruz, as the professor in my very first class here, to the final signature on my thesis. He has provided me with tremendous insight and guidance throughout my graduate work and research, and I'm not sure where I would be without his wisdom.

I would also like to thank the members of my thesis reading committee, Professor James Davis and Professor John Vesecky, not only for their support and advice, but also for being so accommodating, putting up with all of my last-minuteness. Graduate advisor Carol Mullane also deserves much gratitude for this last point, and for all the help she has provided me.

I must also mention my fellow members in the MDSP lab: Hae Jong, Priyam, Xiang, Hossein, and Chelhwon. It has been a pleasure working with them, and I wish them the best in all their endeavors.

My girlfriend, Magnolia, deserves a special thanks for showing me how to take action in pursuing my goals. Without her, I may still be putting off graduate school applications.

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

Chapter 1

Introduction

This chapter introduces the haze removal problem including the imaging model most widely used, and an overview of previous work.

1.1 Background

Images of outdoor scenes often contain haze, fog, or other types of atmospheric degradation caused by particles in the atmospheric medium absorbing and scattering light as it travels from the source to the observer. While this effect may be desirable in an artistic setting, it is sometimes necessary to undo this degradation. For example, many computer vision algorithms rely on the assumption that the input image is exactly the scene radiance, i.e. there is no disturbance from haze. When this assumption is violated, algorithmic errors can be catastrophic. One could easily see how a car navigation system that did not take this effect into account could have dangerous consequences. Accordingly, finding effective methods for haze removal is an ongoing area of interest in the image processing and computer vision fields.

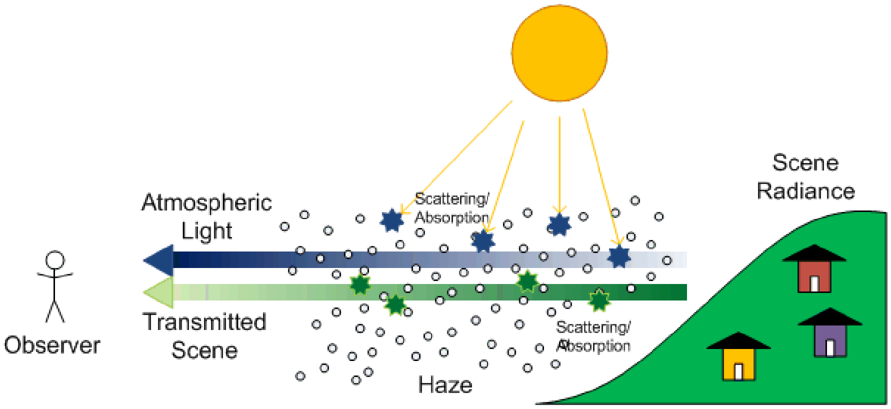


Figure 1.1: Haze Model

A widely used model for haze formation is:

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{a}_\infty (1 - t(\mathbf{x})) \tag{1.1}$$

where $\mathbf{x}$ is a pixel location, $\mathbf{I}$ is the observed image, $\mathbf{R}$ is the underlying scene radiance, $\mathbf{a}_\infty$ is the atmospheric light (or airlight), and t is the transmission coefficient. Intuitively, the image received by the observer is the convex combination of an attenuated version of the underlying scene with an additive haze layer, where the atmospheric light represents the color of the haze (figure **1.1**)[1]. The ultimate goal of haze removal is to find $\mathbf{R}$, which also requires knowledge of $\mathbf{a}_\infty$ and t. From this model, it is apparent that haze removal is an under-constrained problem. In a grayscale image, for each pixel there is only 1 constraint but 3 unknowns; for an RGB color image, there are 3 constraints but 7 unknowns (assuming t is the same for each color channel). Essentially, one must resolve the ambiguous question of whether an object's color is a result of it being far away and mixed with haze, or if the object is close to the observer and simply the correct color.

[1] For a thorough discussion on the physics of haze, the reader is referred to [1]

In order to make the problem easier, the atmosphere is generally assumed to be homogeneous. This has two simplifying consequences: the atmospheric light is constant throughout the image—meaning it only has to be estimated once, and transmission follows the Beer-Lambert law:

$$t(\mathbf{x}) = \exp(-\beta d(\mathbf{x}))$$

where β is the scattering coefficient of the atmosphere, and d is the scene depth. This allows recovery of the scaled scene depth if transmission is known and vice-versa.

Once t and $\mathbf{a}_\infty$ are known, a direct approach finds the underlying scene algebraically from equation 1.1:

$$\begin{aligned}\hat{\mathbf{R}}(\mathbf{x}) &= \frac{\mathbf{I}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\ &= \frac{\mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\ &= \mathbf{R}(\mathbf{x}) \qquad (1.2)\end{aligned}$$

Unfortunately, the image model in equation 1.1 is incomplete. All images, whether they are hazy or not, contain some amount of noise due to measurement (sensor) error. Including this yields a more complete image model for haze formation:

$$\begin{aligned}\mathbf{Y}(\mathbf{x}) &= \mathbf{I}(\mathbf{x}) + \mathbf{n}(x) \\ \mathbf{Y}(\mathbf{x}) &= \mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) + \mathbf{n}(x) \qquad (1.3)\end{aligned}$$

where the observed image is now $\mathbf{Y}$, and $\mathbf{n}$ is the noise contribution, assumed to be independent and identically distributed (I.I.D.), with zero mean and variance σ^2. In this case, even if we know t and $\mathbf{a}_\infty$ perfectly, if the scene radiance is then recovered directly, this more complete model shows that the noise term can dominate the results:

$$\begin{aligned}\hat{\mathbf{R}}(\mathbf{x}) &= \frac{\mathbf{Y}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\ &= \frac{\mathbf{R}(\mathbf{x})\,t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) + \mathbf{n}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\ &= \mathbf{R}(\mathbf{x}) + \frac{\mathbf{n}(\mathbf{x})}{t(\mathbf{x})} \end{aligned} \tag{1.4}$$

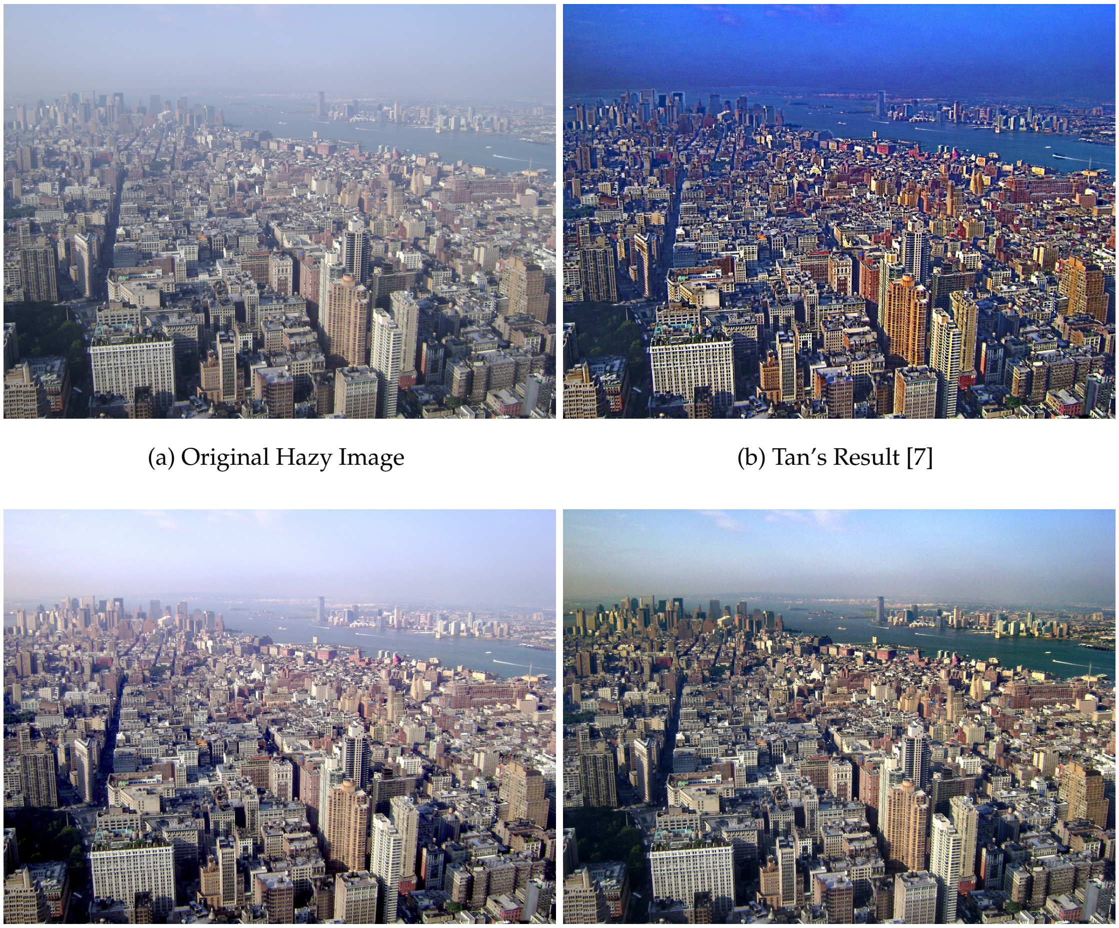
Note that t is in the range of 0 to 1, implying that the noise term is amplified in all cases except when no haze is present. Even when the amount of noise is small in the original image, in regions with dense haze, t is very small, making the noise contribution a significant factor.

1.2 Previous Work

Among current haze removal research, haze estimation methods can be divided into two broad categories of either relying on additional data or using a prior assumption.

Methods that rely on additional information include: taking multiple images of the same scene using different degrees of polarization [2, 3], multiple images taken during different weather conditions [4], and methods that require user supplied depth information [5] or a 3D model [6]. While these can achieve good results, the extra information required is often not available, and so a more flexible approach is preferable.

Significant progress in single image haze removal has been made in recent years. Tan [7] made the observation that a haze-free image has higher contrast than a hazy image, and was able to obtain good results by maximizing contrast in local regions of the input image. However, the final results obtained by this method are not based on a physical model and are often unnatural looking due to over-saturation. Fattal [8] was able to obtain good results by assuming that transmission and surface shading are locally uncorrelated. With this assumption, he obtains the transmission map



(a) Original Hazy Image

(b) Tan's Result [7]

(c) Fattal's Result [8]

(d) He et al.'s Result [9]

Figure 1.2: Single image haze removal methods. Tan's results appear oversaturated, while both Fattal's and He et al.'s results are more visually appealing, with the latter removing more haze. Images come from supplementary material for [9], found at http://personal.ie.cuhk.edu.hk/˜hkm007/.

through independent component analysis. This is a physically reasonable approach, but this method has trouble with very hazy regions where the different components are difficult to resolve. Lastly, a simple but powerful approach proposed by He et al. [9] uses dark pixels in local windows to obtain a coarse estimate of the transmission map followed by a refinement step using an image matting technique from [10]. Their method obtains results on par with or exceeding other state-of-the-art algorithms, and is even successful with very hazy scenes. For a comparison of these three methods, see figure **1.2**.

As illustrated in equation 1.4, noise can be a major problem when restoring hazy images. The works mentioned above largely avoid the issue, usually by assuming a noise free image or only dehazing images up to a point where noise is negligible. Existing literature has taken two basic approaches: denoising prior to dehazing, and denoising during dehazing. Recently, Joshi and Cohen [11] took the first approach, addressing noise through image fusion. By taking multiple images of the same hazy scene and using weighted averaging, they obtain a sharp, low noise hazy image that they then dehaze using a variation on the dark channel method. Schechner and Averbuch [12] use a polarization based method to estimate haze, and address noise by adding a local penalty term proportional to the transmission value as a regularization term in scene radiance recovery. Since the regularization is relatively simple, hazy regions are essentially blurred while non-hazy regions are left sharp. A more sophisticated variant on this theme is proposed by Kaftory and Schechner [13], which uses a total variation method based on Beltrami flow for regularization. This method is able to maintain sharpness in heavily denoised regions. Although effective, the use of complicated PDE methods, requiring minimization over the entire image, is a major drawback.

1.3 Summary of Thesis

This thesis addresses the problem of recovering the underlying scene radiance of a single noisy, hazy image. The main contributions are as follows. First is an investigation on the effect of noise on an existing single image haze estimation method, with denoising proposed as a pre-processing step. Next is the proposal of two different methods for scene radiance recovery from a single noisy hazy image once estimates for the haze content have been obtained. The first method is the process of denoising the image using a state-of-the-art denoising algorithm prior to dehazing. This can be interpreted as a single image adaptation of [11]. The second method is the process of simultaneously denoising and dehazing using an iterative non-parametric kernel regression based method. This can be seen both as an adaptation to a single image as well as a simplification of methods presented in [12] and [13].

1.3.1 Organization

Chapter 2 addresses the first step for restoration: haze estimation. Due to its simplicity and effectiveness, the dark channel prior serves as the basis for single image haze estimation in this thesis. First, a brief review of the original method is provided. Since the addition of noise was not originally considered, we explore its effects and find that due to the nature of the estimation method, the results are very sensitive to it. The problem of applying the dark channel prior to the noisy case is addressed by proposing the application of an existing state-of-the-art denoising method as a pre-processing step.

Chapter 3 provides a review of two techniques for further refining the transmission map obtained from the dark channel prior. Without the additional refinement step, the recovered scene radiance suffers from a halo effect around depth discontinuities. In [9], He et al. proposed a refinement approach adopted from the image matting literature [10], and while this provides a satisfactory solution, the result involves the inversion of a very large matrix. In the interests of avoiding this inversion, another refinement solution is obtained by using the "Guided Image Filter," proposed in [14]. Although this filter provides only an approximate solution to the image matting approach, it still captures details of the original image and sufficiently suppresses the halo effect, providing visually satisfying results. Previously, the effect on the transmission map from the addition of noise has not been addressed. We provide a brief discussion concluding that noise has little effect on this refinement step, and that in strong noise cases, applying a denoising method prior to refinement gives satisfying results.

Chapter 4 discusses recovery of the final scene radiance once the haze estimates are obtained. First we present the effect of disregarding noise in the input image, showing that even if noise is weak, it can be significantly amplified in the scene radiance when haze is strong. Two approaches to remedy this problem are proposed. The first approach is to denoise the image prior to dehazing, which proves convenient considering that the initial haze estimation already requires this step. The results are convincing when paired with a state-of-the-art denoising algorithm. The second approach is an iterative nonparametric method based on kernel regression to simultaneously denoise and dehaze. Results of the two approaches are then compared.

Finally, Chapter 5 concludes the thesis with a summary of findings, open questions, and potential topics for future research.

Chapter 2

Single Image Haze Estimation

There are several methods for estimating the haze contribution in a single image. One of the most successful of these methods is known as the dark channel prior, which is used as the basis for haze estimation in this thesis.

2.1 Dark Channel Prior

The dark channel prior was first introduced by He et al. in [9], and is essentially a localized version of an earlier haze removal method known as dark object subtraction [15]. The key to this approach is the observation that natural haze-free outdoor images are generally well textured, and contain a variety of different colored objects. As a consequence, most patches will contain one or more pixels with very low intensity in at least one of the color channels. These dark pixels can be attributed to dark objects, shadows, or objects that are primarily a combination of only one or two of the RGB color channels. With this observation in mind, one can construct the so called “dark channel" of an image, which can be expressed mathematically as a minimum value operation in patches around the target pixel:

$$I^{dark}(\mathbf{x}) = \min_{c \in \{r,g,b\}} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left(I^c(\mathbf{y}) \right) \right) \tag{2.1}$$

where $I^{dark}(\mathbf{x})$ represents the ″dark channel" of image I at pixel location $\mathbf{x}$, I^c is a color channel of image I, and $\mathbf{y} \in \Omega(\mathbf{x})$ signifies all pixels $\mathbf{y}$ in a local patch around $\mathbf{x}$. If applied to a haze-free image, the above observation yields:

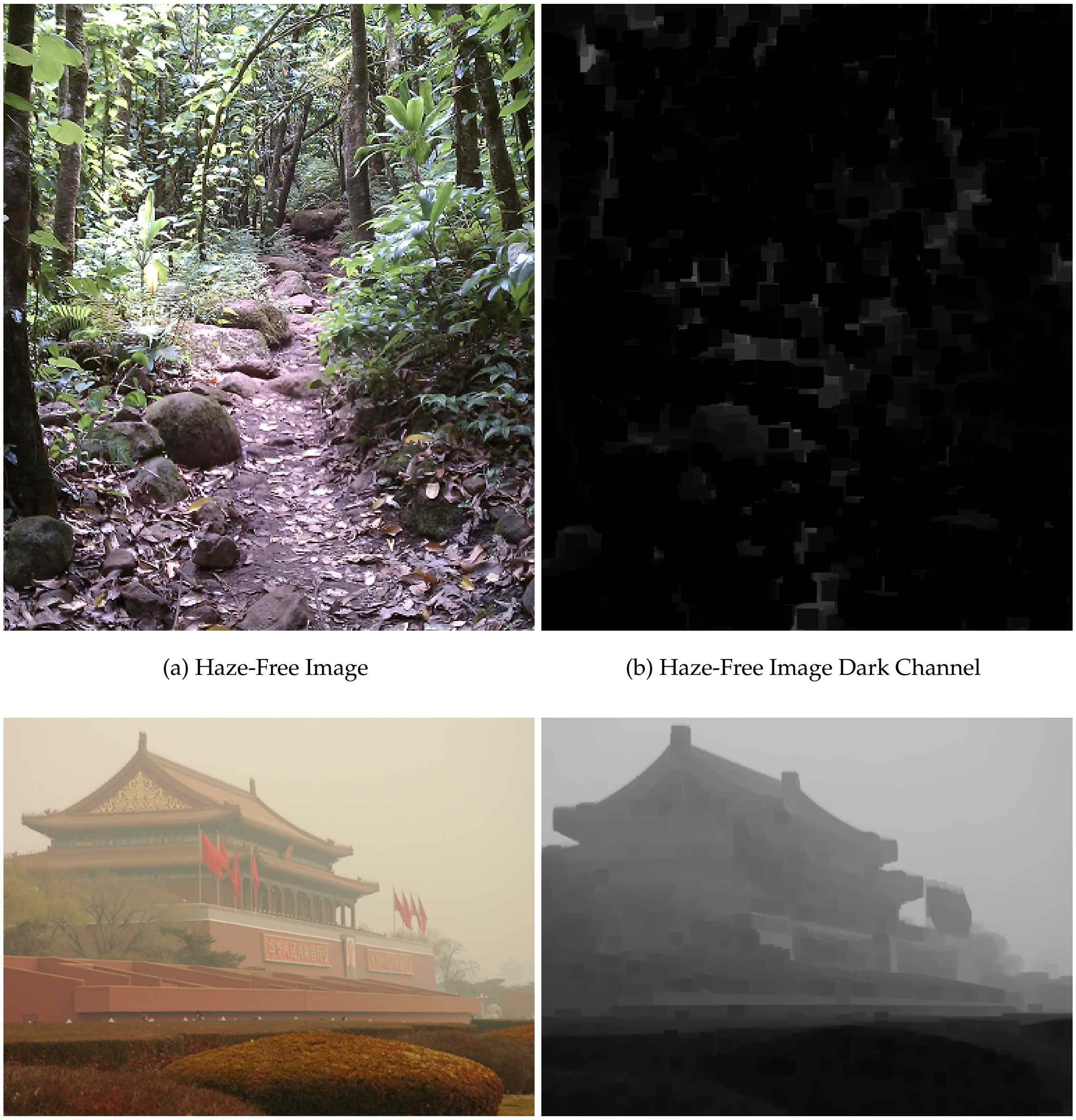
$$I^{dark}(\mathbf{x}) \rightarrow 0$$

This observation is confirmed in [9] through an analysis of 5,000 images collected from Flickr.com. He et al. found that, using a 15x15 patch size, "75% of the pixels in the dark channels have zero values, and the intensity of 90% of the pixels is below 25."

In contrast to a haze-free image, a hazy image contains an additive component due to the atmospheric light contribution, and since haze is typically close to white, this component is present in all color channels. The end effect is that pixels that would have been close to zero in a haze-free image, now take on a value that is approximately equal to the amount of haze. Consequently, the dark channel of a hazy image provides a good approximation of the amount of haze throughout the image. Fig. **2.1** shows a comparison between the dark channels of a haze-free image and a hazy image, confirming this effect.

2.1.1 Atmospheric Light Estimation

The dark channel prior effectively identifies the relative amount of haze in an image, and thus can aid in estimating the atmospheric light, $\mathbf{a}_\infty$. In the haziest regions of the image, transmission tends toward zero, and the atmospheric light contribution



(a) Haze-Free Image

(b) Haze-Free Image Dark Channel

(c) Hazy Image

(d) Hazy Image Dark Channel

Figure 2.1: The dark channel of the haze-free image is mostly zero, while the dark channel of the hazy image is proportional to the local amount of haze.

dominates the scene radiance. This is apparent from examining Eq. (1.1):

$$\begin{aligned}\lim_{t(\mathbf{x})\to 0} \mathbf{I}(\mathbf{x}) &= \lim_{t(\mathbf{x})\to 0} \mathbf{R}(\mathbf{x})\,t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) \\ &= \mathbf{a}_\infty\end{aligned}$$

Thus the haziest regions of **I** contain mostly atmospheric light. The approach adopted in this thesis, following [9], is estimating $\mathbf{a}_\infty$ as the brightest intensities in each color channel chosen from the 0.1% haziest pixels in **I**, which correspond to the top 0.1% brightest pixels in the dark channel. This is much different than simply choosing the brightest pixels from the entire image. If the selection were not limited, the estimate may come from a bright object in the foreground that contains little haze, and thus little information about the atmospheric light.

An important consequence of this realization is that patch size plays a significant role in finding good atmospheric light candidates. While a patch size of 15x15 is often sufficient for obtaining a good overall haze estimate, regions with bright white or gray objects larger than the patch size may be detected as hazy, when in fact they are not. This can lead to significant errors in atmospheric light estimates when pixels from these false regions are brighter than pixels in the actual hazy regions (Fig. **2.2**). To overcome this problem, the patch size must be sufficiently increased to minimize this type of error.

2.1.2 Transmission Estimate

Assuming now that $\mathbf{a}_\infty$ is given, an estimate of the transmission map is obtained by once again applying the dark channel prior. Noting that atmospheric light and transmission are positive quantities, and that transmission is approximately constant in a local window ($t(\mathbf{y}) \approx t(\mathbf{x}) \quad \forall\, \mathbf{y} \in \Omega(\mathbf{x})$), the following is derived in [9]:

$$\begin{aligned}
\frac{\mathbf{I}(\mathbf{x})}{\mathbf{a}_\infty} &= \frac{\mathbf{R}(\mathbf{x})\, t(\mathbf{x})}{\mathbf{a}_\infty} + (1 - t(\mathbf{x})) \\
\min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{I^c(\mathbf{y})}{a^c_\infty} \right) \right) &\approx \min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{R^c(\mathbf{y})\, \hat{t}(\mathbf{x})}{a^c_\infty} \right) \right) + (1 - \hat{t}(\mathbf{x})) \\
\min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{I^c(\mathbf{y})}{a^c_\infty} \right) \right) &\approx 0 + (1 - \hat{t}(\mathbf{x})) \\
\hat{t}(\mathbf{x}) &= 1 - \min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{I^c(\mathbf{y})}{a^c_\infty} \right) \right) \qquad (2.2)
\end{aligned}$$

where the superscript c signifies a specific color channel, i.e. the red, green, or blue color channel of the corresponding parameter. Note that $\mathbf{a}_\infty$ is a vector quantity containing a separate value for each color channel, and so a^c_∞ is a scalar quantity referring to one individual value. Since $1 - \hat{t}(\mathbf{x})$ is constant, it is taken out of the min operation. Furthermore, since t and a^c_∞ are both positive, the dark channel prior implies that $\min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{R^c(\mathbf{y})\hat{t}(\mathbf{x})}{a^c_\infty} \right) \right) \to 0$. Thus, an estimate for the transmission map is obtained by simply subtracting the dark channel of the normalized image from 1.

According to [16], humans require some presence of haze to perceive depth, a phenomenon called aerial perspective, and so removing the haze completely can lead to unnatural, flat looking images. A scaling parameter can be introduced into Eq. (2.2) to control the amount of remaining haze:

$$\hat{t}(\mathbf{x}) = 1 - w \min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{I^c(\mathbf{y})}{a^c_\infty} \right) \right) \qquad (2.3)$$

The scaling parameter, w, takes a value from 0 to 1, corresponding to the amount of haze left in the image. A typical value to preserve aerial perspective is 0.95. As in the atmospheric light estimation, patch size is also an important parameter to choose. If the patch size is too small, the dark channel prior may not be valid, and transmission will be underestimated, due to bright objects. If the patch size is too large, transmission may be overestimated, due to darker pixels in regions no longer local (i.e. at a closer depth to the observer) to the target pixel. For a 600×400 image, a 15×15 patch size usually strikes a good balance. Fig. **2.3** shows an example of transmission map estimates using Eq. (2.3) using $w = 0.95$, comparing different patch sizes [1].

2.2 Sensitivity to Noise

In the field of statistics, sample minima are well known to be extremely sensitive to noise, and while they may provide good information when the data is clean, they are inherently unreliable when the data is noisy. Consequently, since the dark channel relies on sample minima, it is not a robust statistical measure. Fig. **2.4** is an example of this sensitivity to noise. The dark channel of Fig. **2.3**a was computed after Gaussian noise was added with various standard deviations.

Various approaches can be taken to robustly estimate the dark channel, and by extension the transmission map, considering the presence of noise. A basic method from the field of descriptive statistics is to use quantiles, such as the 10th percentile as an estimate [17]. While this may give more accurate results, a better estimate can be made. A more sophisticated approach can be taken by using stochastic approximation [18] to locate local minima, followed by some type of point estimate. However, since other point estimates are needed (such as for the atmospheric light estimate) and a pilot estimate of the dehazed image is necessary for later haze removal stages, a still better approach may be to simply denoise the entire image as a first step. This is the method employed here, with the specific denoising algorithm being the color version

[1] The scaling parameter is presented here for the interest of the reader. In subsequent chapters, we set aside this parameter and let $w = 1$.

of BM3D [19] [20]. From the denoised hazy image, the dark channel, atmospheric light, and transmission map can be estimated.

2.2.1 Denoising with BM3D

An image model typically used for denoising is:

$$\mathbf{Y}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) + \mathbf{n}(\mathbf{x}) \tag{2.4}$$

where $\mathbf{x}$ is a pixel location, $\mathbf{Y}$ is the observed image, $\mathbf{I}$ is the clean image, and $\mathbf{n}$ is the noise contribution, which is typically assumed to be independent and identically distributed, zero-mean Gaussian noise with variance σ^2. Therefore the job of a denoising algorithm is to estimate $\mathbf{I}$ given this image model. In the case of haze estimation, this image estimate allows computation of the dark channel.

BM3D is a state-of-the-art denoising algorithm proposed for grayscale images by Dabov at al. in [19] and later extended to color images in [20]. While a detailed discussion on the BM3D algorithm is beyond the scope of this thesis, a brief summary is necessary. The algorithm works in two major denoising steps: pilot estimate and final estimate. The pilot estimate of the image is constructed by first grouping similar 2D image patches into 3D arrays—called block matching in [19], followed by “collaborative hard-thresholding" in a 3D transform domain , typically the Discrete Cosine Transform (DCT), and finally an aggregation step to compute the pilot image by weighted averaging of all overlapping block-wise estimates. For the final estimate, grouping is performed once again, but this time from the pilot estimate. Next is a “collaborative Wiener filtering" step in a 3D transform domain (DCT), using the new groups as an estimate for the true image energy spectrum. Finally, another aggregation step computes the final image estimate.

The pilot estimate in BM3D serves both to improve the patch grouping performance, as well as provide a better spectrum estimate (as opposed to hard thresholding in the standard image domain) for Wiener filtering. BM3D is currently among the best performing denoising algorithms, in terms of both mean squared error (MSE) and subjective image quality. Another advantage to BM3D compared to other denoising algorithms is that it takes only the one tuning parameter: an estimate for the noise standard deviation.

For color images, a simple constraint is added to the patch grouping step [20]. Instead of processing each color channel separately, the image is first transformed from RGB to a luminance-chrominance color space such as YCbCr or YUV. Grouping is performed only on the luminance channel, and then applied to the chrominance channels. The rest of the algorithm is performed independently on each channel, still in the luminance-chrominance space. Finally, the image is converted back to the RGB color space for display.

An important parameter (in fact the only parameter) needed in using the BM3D algorithm is an estimate for the noise standard deviation. Although in synthetic examples this parameter can be exactly known, this is not possible for real noisy images. An effective no-reference method for setting tuning parameters in denoising algorithms is proposed by Zhu et al. [21], called the Q metric. This metric is based on the singular values of local gradient matrices computed on anisotropic patches within the target image. The anisotropic patches are determined from the input noisy image, and Q is recalculated on the denoising results as the tuning parameter is varied. Zhu et al. find that the output with the highest Q corresponds well to the output with the lowest mean squared error. Again using the image from Fig. **2.3**a as a source, Fig. **2.5** shows that, in this case, Q indeed provides a reasonable estimate of the best tuning parameter.

2.2.2 Dark Channel for Noisy Images

As mentioned previously, to address noise when computing the dark channel, the image is first denoised using the BM3D algorithm, and then the dark channel is computed on the denoised image. Fig. **2.6** shows the dark channels of the image from Fig. **2.3**a following denoising after various levels of Gaussian noise were added. Compare these to the results of Fig. **2.4**, when no denoising was performed. Also for this image example, Fig. **2.7** shows the results of a Monte Carlo simulation comparing the MSE of dark channels computed for a noisy image to the dark channels of a denoised image for various levels of noise.

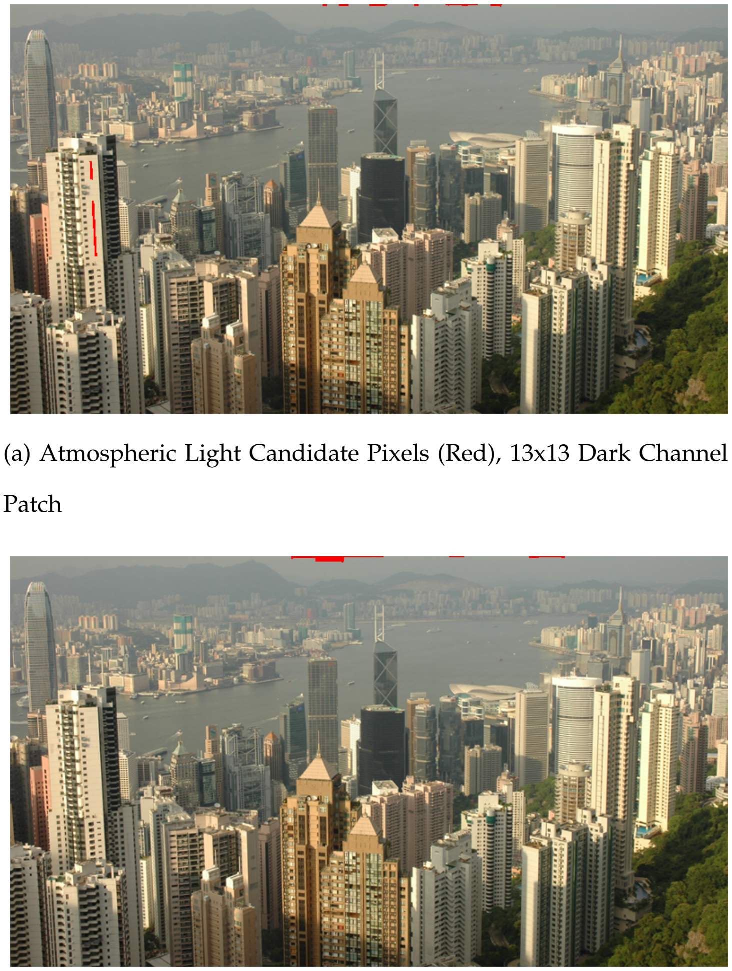
As mentioned above, the BM3D algorithm requires an estimate of the noise standard deviation. However, since this parameter is generally not known, the question arises of whether it is better to err on the side of over-estimating or under-estimating the amount of noise in the image. The result of over-denoising an image is generally that the image becomes smoother. Since haze is primarily a slowly varying component in the image, intuitively, one would conclude that it is better to over-estimate the noise, since a smoother image would not significantly affect the haze content. Fig. **2.8** supports this hypothesis, showing that, for an image with added Gaussian noise ($\sigma_n = 0.1$), although the estimates for the dark channel and atmospheric light reach an optimal point at the exact noise level, these estimates only worsen slightly as the denoising level increases, but are much worse if the image is “under"-denoised

When using the best Q as a stopping criterion, experiments have shown that it generally tends to err on an overestimate of the noise, rather than an underestimate of the noise. Thus when the noise level is unknown, this method is suitable for haze estimation.

2.3 Summary

The dark channel prior is a simple but effective tool for estimating the haze content and atmospheric light. However, complications arise when the image contains noise. Experiments show that first denoising the image effectively mitigates these difficulties. Furthermore, experimental evidence suggests that over-estimating the noise content of the image leads to more accurate haze and atmospheric light estimation compared to results from under-estimating the noise content.

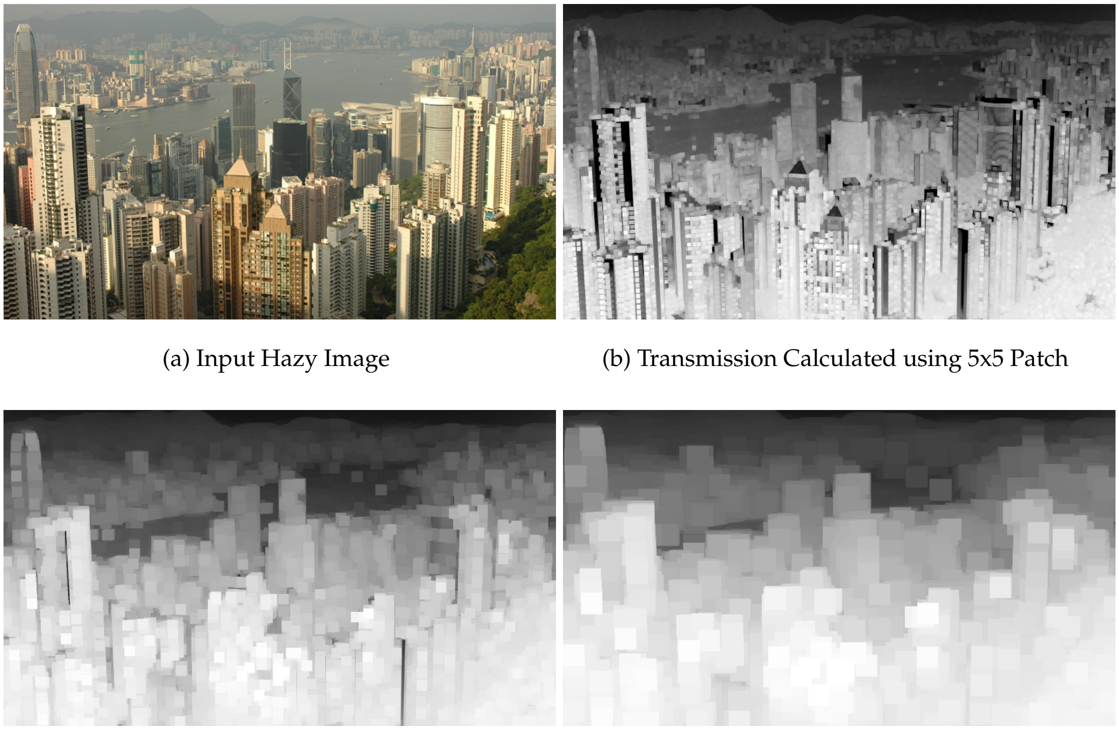
The next chapter will show that haze removal using the direct result from the dark channel prior leads to artifacts in the final image, since only a coarse estimate of the haze is provided. As a result, the transmission map estimate must be further refined before the scene radiance can be recovered. This refinement process can take several forms, which we discuss next.



(a) Atmospheric Light Candidate Pixels (Red), 13x13 Dark Channel Patch

(b) Atmospheric Light Candidate Pixels (Red), 25x25 Dark Channel Patch

Figure 2.2: A 13x13 patch size in Fig. **2.2**a results in the false detection of the foreground building as an atmospheric light candidate, leading to an erroneous atmospheric light estimate. Increasing the patch size to 25x25 remedies the problem, correctly determining the most distant region as the haziest.



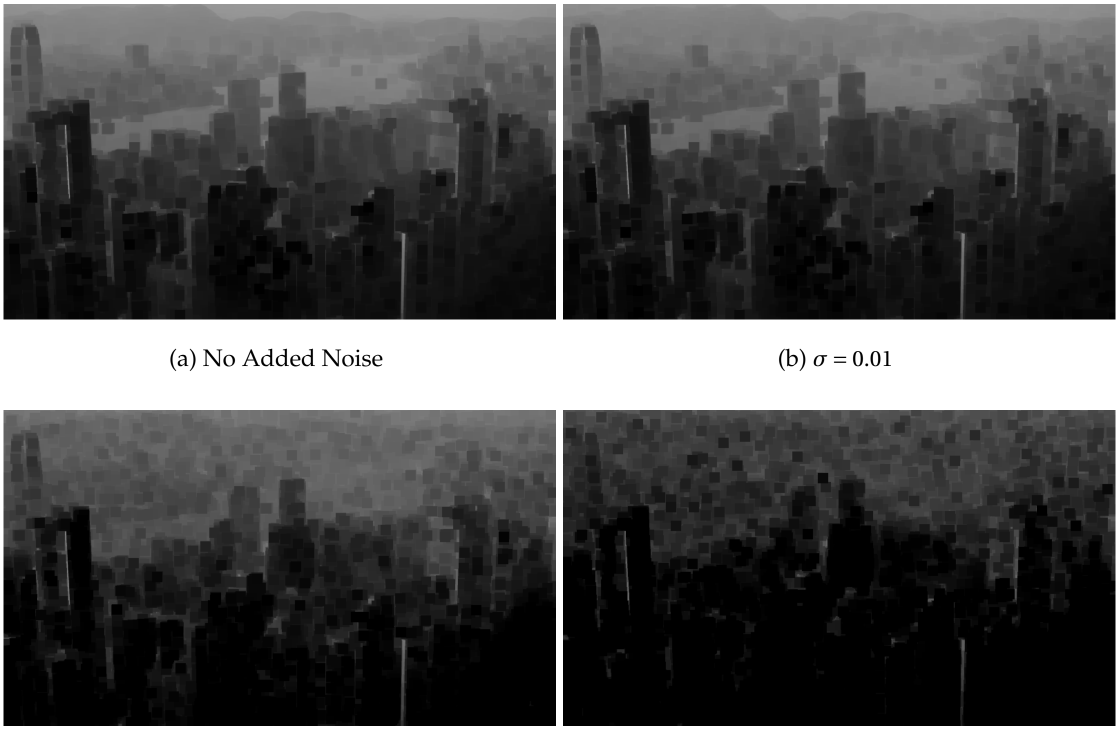
(a) Input Hazy Image

(b) Transmission Calculated using 5x5 Patch

(c) Transmission Calculated using 15x15 Patch

(d) Transmission Calculated using 31x31 Patch

Figure 2.3: With a 5x5 patch size, transmission is underestimated in many of the foreground buildings, while a 31x31 patch size leads to overestimated transmission at depth discontinuities. A 15x15 patch size leads to some underestimation in the foreground buildings, but reduces the over-estimation at depth discontinuities.



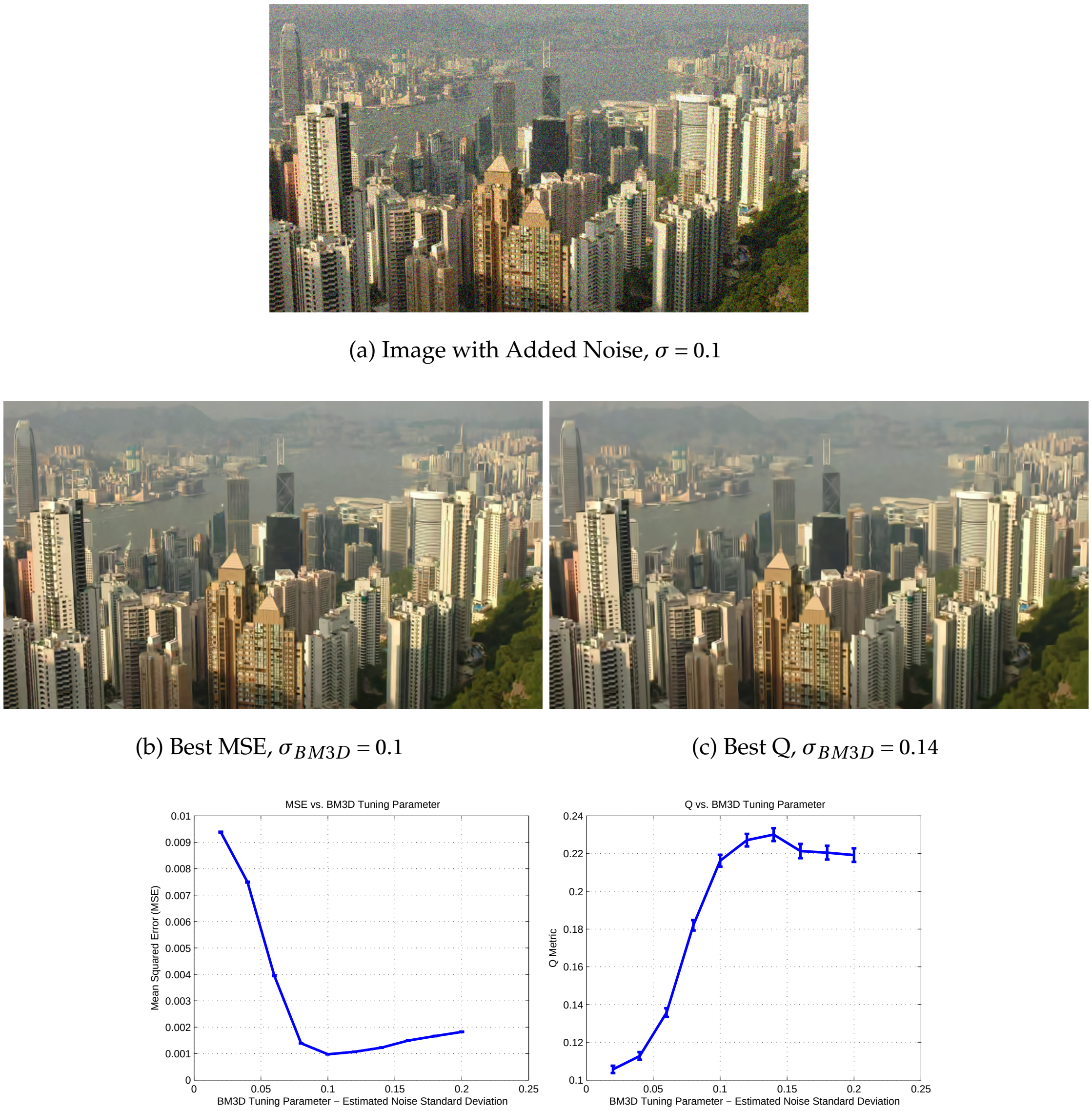
(a) No Added Noise

(b) $\sigma = 0.01$

(c) $\sigma = 0.05$

(d) $\sigma = 0.1$

Figure 2.4: The dark channels, using a patch size of 15x15, are computed for the image from Fig. **2.3**a with added white Gaussian noise corresponding to standard deviations from 0 to 0.1 for an image whose color values are normalized to 1. As can be observed, the dark channel estimate changes drastically as noise level is increased.



(a) Image with Added Noise, $\sigma = 0.1$

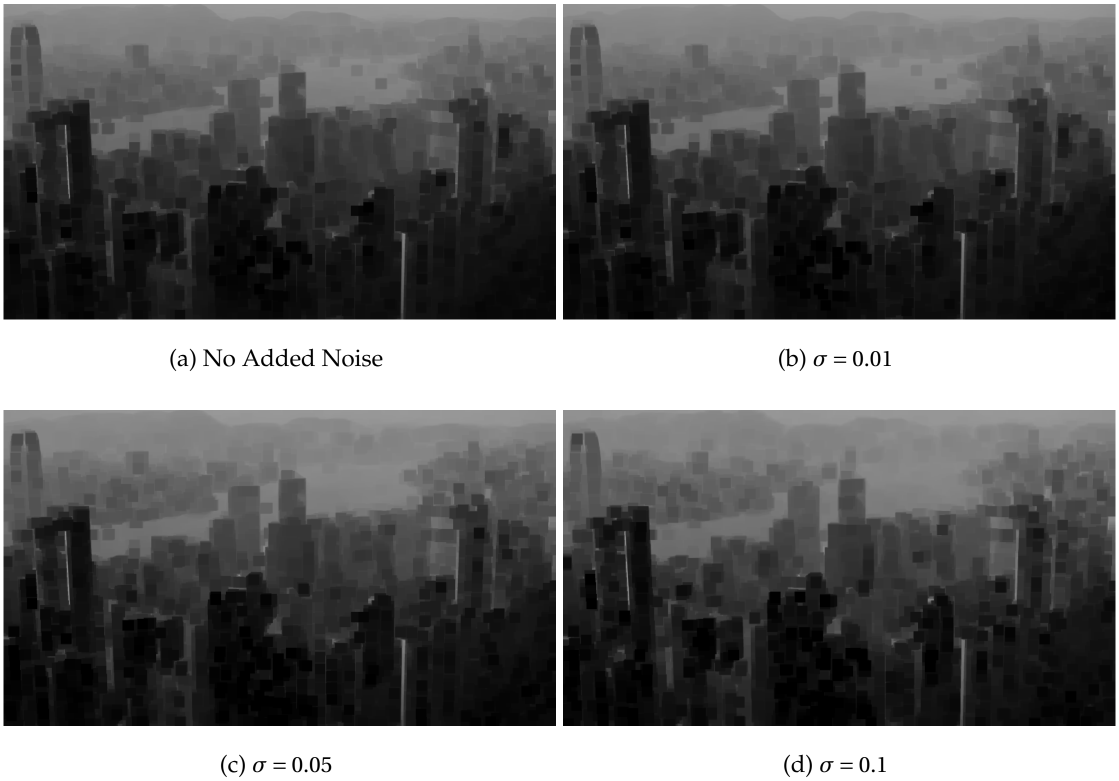
(b) Best MSE, $\sigma_{BM3D} = 0.1$

(c) Best Q, $\sigma_{BM3D} = 0.14$

(d) MSE vs σ_{BM3D}

(e) Q vs σ_{BM3D}

Figure 2.5: Noise with standard deviation 0.1 was added to image from Fig. **2.3**a, and then removed using BM3D. Results from a 5-trial Monte Carlo simulation show using the best Q as a stopping parameter leads to a slightly over-smoothed result in this case.



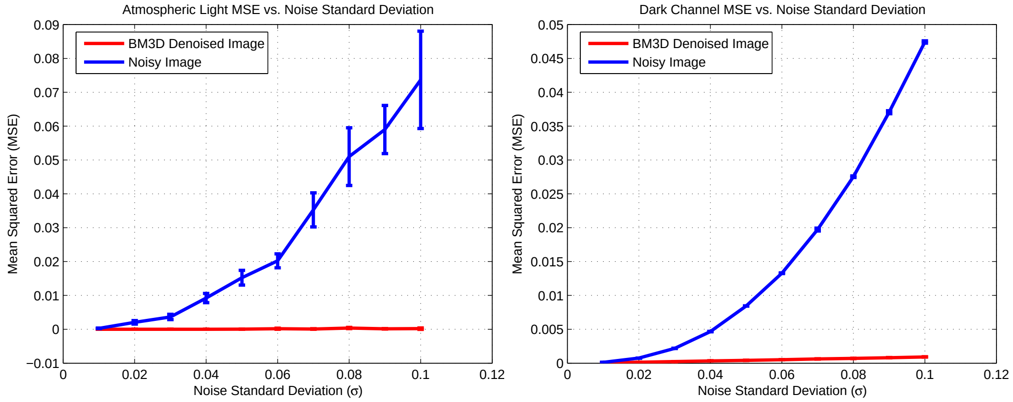
(a) No Added Noise

(b) $\sigma = 0.01$

(c) $\sigma = 0.05$

(d) $\sigma = 0.1$

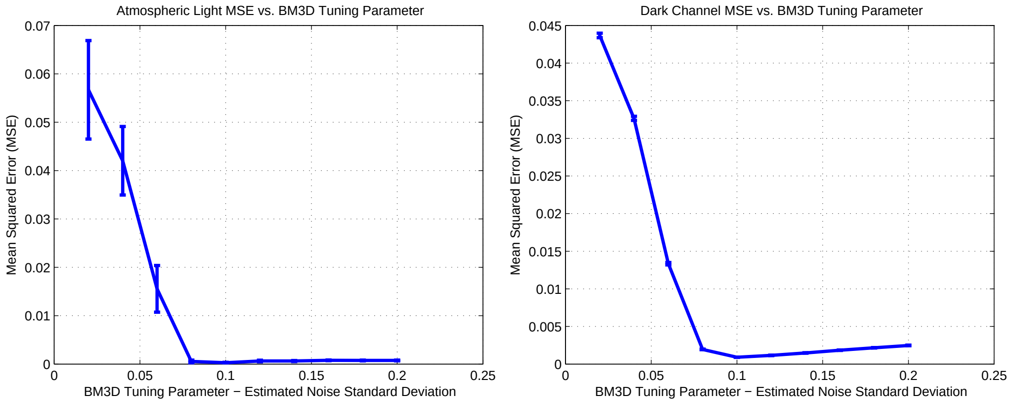
Figure 2.6: The dark channels, using a patch size of 15x15, are computed for the image from Fig. **2.3**a that was denoised using BM3D after Gaussian noise was added (compare to Fig. **2.4**). The BM3D algorithm was given the exact noise variance. Note that while the dark channels have some variation, the overall result changes very little.



(a) Atmospheric Light MSE vs. σ_n

(b) Dark Channel MSE vs. σ_n

Figure 2.7: Average Atmospheric Light and Dark Channel MSE vs Noise Standard Deviation, results of 5 trial Monte Carlo simulation for image from Fig. **2.3**a. The BM3D algorithm was given the exact noise variance. Atmospheric light and dark channel are significantly improved in denoised image.



(a) Atmospheric Light MSE vs σ_{BM3D}

(b) Dark Channel MSE vs σ_{BM3D}

Figure 2.8: Average Atmospheric Light and Dark Channel MSE vs BM3D Tuning Parameter, results of 5 trial Monte Carlo simulation for image from Fig. **2.3**a with added Gaussian noise, standard deviation of 0.1

Chapter 3

Transmission Map Refinement

Assuming for a moment that noise is not present in the hazy image, once the transmission map, t, and atmospheric light, $\mathbf{a}_\infty$, are known, the scene radiance can be solved for directly using Eq. (1.2), which is repeated here for convenience:

$$\begin{aligned}\hat{\mathbf{R}}(\mathbf{x}) &= \frac{\mathbf{I}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\ &= \frac{\mathbf{I}(\mathbf{x}) - \mathbf{a}_\infty}{t(\mathbf{x})} + \mathbf{a}_\infty \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is a pixel location, $\mathbf{I}$ is the input image, $\mathbf{R}$ is the scene radiance, $\mathbf{a}_\infty$ is the atmospheric light, and t is the transmission. When using Eq. (3.1), t is typically lower bounded to a small number, such as 0.1, to avoid instability. Chapter 2 provided a simple method for estimating the transmission and atmospheric light. However, if the scene radiance is recovered directly from this transmission estimate, the result contains block artifacts and halos around depth discontinuities, due to the nature of the estimation method (see Fig. **3.1**).

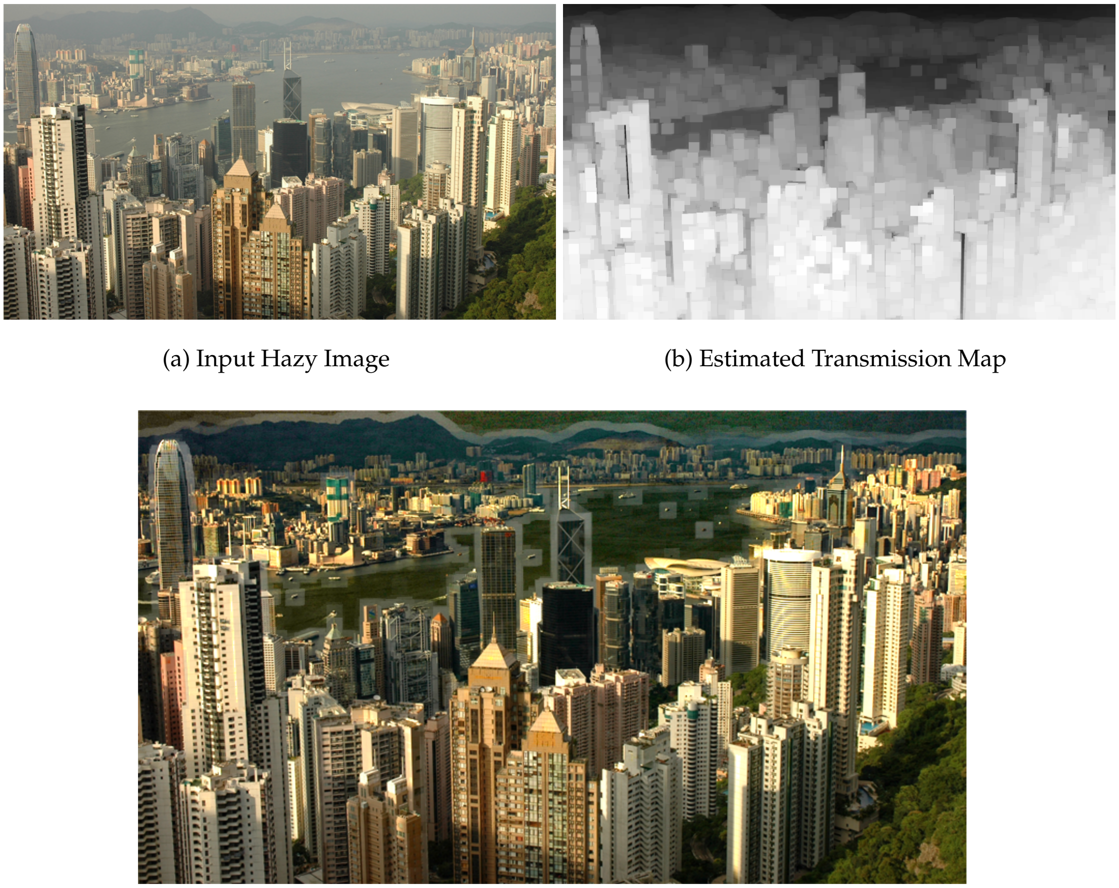
In order to remove these artifacts, it is necessary to further refine the transmission map. This chapter first provides a review of two existing, related methods that use the original hazy image as a guide to refine the transmission map. The first is a result from Levin et al. [10] originally intended for image matting that was initially adopted in [9]. The second is a much faster method called the “Guided Image Filter" [14] equivalent to a single Jacobi iteration in solving for the solution from [10]. Although in [14], the Guided Filter was briefly mentioned as applicable to transmission map refinement, here it is examined more closely for this application, with a direct comparison to the results using the first approach. Finally, noise in the guide image has not been previously considered for the application of transmission map refinement, and so this chapter ends with a discussion of this case.

3.1 Matting Laplacian

Interactive image matting is a problem in which one must separate a foreground object from a given image based on limited user input. For example, one may wish to separate a person from the surrounding scenery. User input generally involves the creation of a coarse image mask, called a trimap [10], where white pixels definitely belong to the foreground object, black pixels definitely belong to the background object, and gray pixels can belong to either. The job of an image matting algorithm is then to refine this coarse image mask, resolving whether the gray pixels belong to the foreground or background. Formally, the image matting problem can be described by the following equation:

$$\mathbf{I}(\mathbf{x}) = \mathbf{F}(\mathbf{x})\alpha(\mathbf{x}) + \mathbf{B}(\mathbf{x})(1 - \alpha(\mathbf{x})) \tag{3.2}$$

where $\mathbf{x}$ is a pixel location, $\mathbf{I}$ is the input image, $\mathbf{F}$ is the foreground image, $\mathbf{B}$ is the background image, and α is called the alpha matte, describing the opacity of the fore-



(a) Input Hazy Image

(b) Estimated Transmission Map

(c) Recovered Scene Radiance

Figure 3.1: Recovering the scene radiance using the transmission map obtained directly from the method presented in Chapter 2 results in undesirable artifacts.

ground. Looking closely, this is exactly the haze formation equation first presented in Chapter 1, repeated here for convenience:

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{a}_{\infty}(\mathbf{x})\, (1 - t(\mathbf{x})) \tag{3.3}$$

where **F** corresponds to the underlying scene radiance, **R**, **B** corresponds to the global atmospheric light, $\mathbf{a}_{\infty}$, and α corresponds to the transmission, t. Furthermore, the coarse transmission map estimated in Chapter 2 can be interpreted as the user-input trimap in interactive image matting.

In [10], Levin et al. proposed a state-of-the-art, closed-form solution to this problem. Their solution is based on the assumption that within a small window, pixels belonging to a single object are a linear mixture of two colors. In other words, these pixels belong to a line in RGB space. This assumption is supported by Omer and Werman in [22], who found that pixels from the same object form elongated clusters in RGB space. These clusters can be approximated locally by a linear model. From this assumption, Levin et. al derive an affinity matrix, L, known as the "Matting Laplacian."

The (i,j) element of L is defined as:

$$L(i,j) = \sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|}\left(1 + \left(\mathbf{I_i} - \mu_k\right)^T \left(\Sigma_k + \frac{\varepsilon}{|w_k|}\mathbf{U}_3\right)^{-1} \left(\mathbf{I_j} - \mu_k\right)\right)\right) \tag{3.4}$$

where δ_{ij} is the Kronecker delta, μ_k and Σ_k are the mean and covariance of the colors in window w_k centered around k, $|w_k|$ is the number of pixels in each window, and U_3 is a 3x3 identity matrix. ε is a small regularization parameter. The summation is for all windows shared by pixels i and j in the image I. Note that for an image of size MxN, L is a symmetric matrix with size MNxMN, and that i and j refer both to a location in L and to the ith and jth pixels in the vectorized image, I. Since window sizes that are too large may violate the color line model, the typical size used, and the size used for all examples shown here, is 3×3.

3.1.1 Transmission Map Refinement

Given the coarse transmission map estimate, the Matting Laplacian is used to find a refined transmission map by minimizing the following quadratic cost function:

$$E(\mathbf{t}) = \mathbf{t}^T L\mathbf{t} + \lambda(\mathbf{t} - \hat{\mathbf{t}})^T(\mathbf{t} - \hat{\mathbf{t}}), \tag{3.5}$$

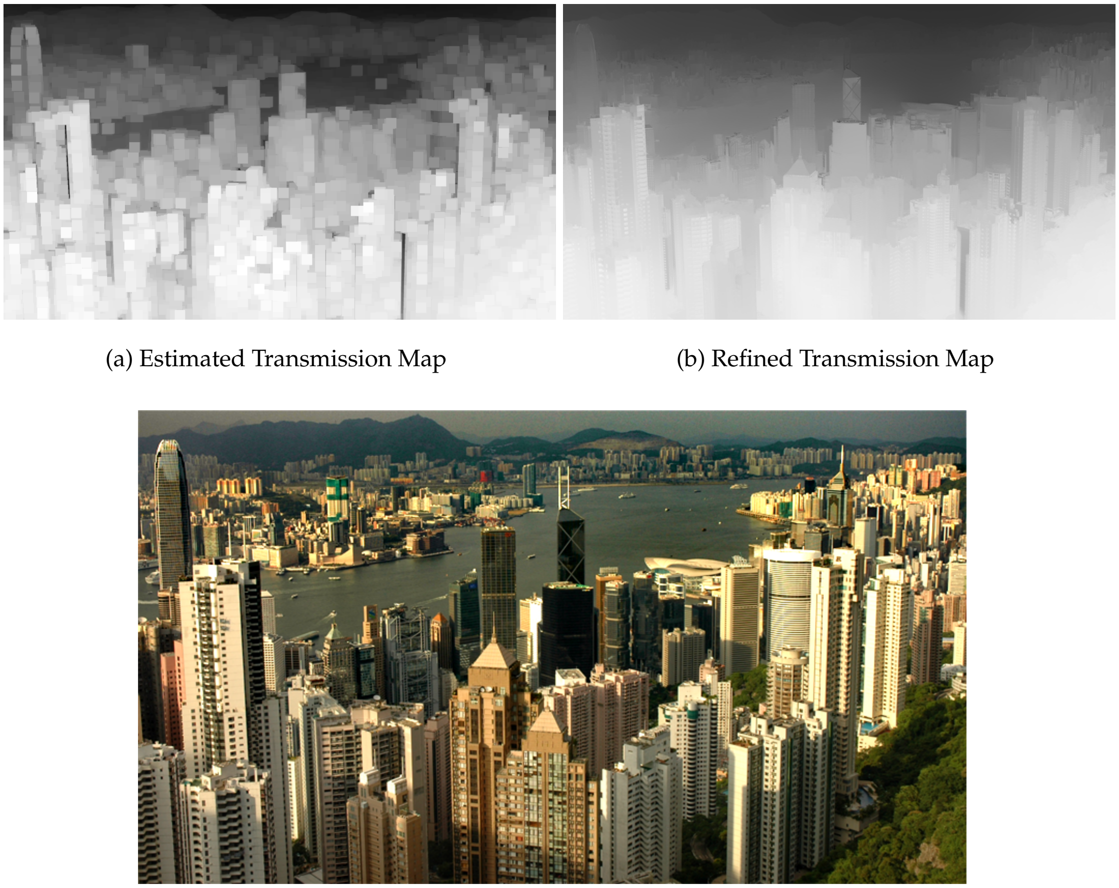
where $\mathbf{t}$ is the vector form of t, $\hat{\mathbf{t}}$ is the vector form of the coarse transmission estimate, and λ is a regularization parameter. The solution minimizing Eq. (3.5) is found by solving for $\mathbf{t}$ from the following:

$$(L + \lambda U)\mathbf{t} = \lambda \hat{\mathbf{t}} \tag{3.6}$$

where U is an identity matrix with the same size as L, and λ is a small value (10^{-3} to 10^{-4}) so that $\mathbf{t}$ is softly constrained by $\hat{\mathbf{t}}$. Although L is dimensionally large (MNxMN), due to the window constraint on i and j, it is sparse. In MATLAB, this linear system can be solved using the backslash operator. Iterative methods, such as the conjugate gradient method or the Jacobi method, can also be used. Fig. **3.2** shows the results of applying the Matting Laplacian to the results from Fig. **3.1**.

3.2 Guided Filter

Much of modern image processing involves some type of locally adaptive image filtering. These filters are typically defined via information from a guidance image, which is often the target image itself. The filtering process may be done in the vector domain, where the image is column stacked into a large vector, by first computing a large matrix that encodes information from the guidance image, and then solving for the filter output through the minimization of a quadratic cost function involving the target image. The most relevant example of this process is the refinement of the initial transmission map through the Matting Laplacian, discussed in the previous section. Alternatively, the filter kernel can be explicitly built using the guidance image, and then applied to the target image through a standard linear filtering process in the pixel domain. A notable example of this is the bilateral filter [23]. In general, the locally



(a) Estimated Transmission Map

(b) Refined Transmission Map

(c) Recovered Scene Radiance from Refined Transmission Map

Figure 3.2: Refining the transmission map using the Matting Laplacian results in a much smoother recovered scene radiance. For this example, $\lambda = 10^{-3}$ and $\epsilon = 10^{-4}$.

adapted filtering process can be described as a weighted sum [14]:

$$q_i = \sum_j W_{ij}(I) p_j, \tag{3.7}$$

where i and j are pixel indexes, q is the result, W is a weight dependent on the guidance image, I, and p is the target (or input) image. For example, the joint bilateral filter

[24] could be defined in this way with the filtering kernel W^{jbf} as:

$$W_{ij}^{jbf}(I) = \frac{1}{K_i} \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\sigma_s^2}\right) \exp\left(-\frac{|I_i - I_j|^2}{\sigma_r^2}\right), \tag{3.8}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are pixel locations, I_i and I_j are pixel intensities, σ_s and σ_r adjust the spatial similarity and range similarity respectively, and K_i is a scaling coefficient so that $\sum_j W_{ij}^{jbf}(I) = 1$. When I and p are identical, this becomes the original bilateral filter [23].

The Guided Filter was proposed by He et al. in [14] as a form of translational variant filtering, and here we present a review of its derivation as well as its application to transmission map refinement. Its weights result from the optimization of a quadratic cost function in a local window around the target pixel. Much like the Matting Laplacian, the cost function is based on the assumption that in a local window the pixel intensities or colors can be represented by a linear model. In fact, He et al. show that their filter is one Jacobi iteration in optimizing Eq. (3.5).

For a grayscale image, the derivation of the Guided Filter relies on a local linear assumption, which is expressed mathematically as:

$$q_i = a_k I_i + b_k, \quad \forall\, i \in w_k, \tag{3.9}$$

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

$$E(a_k, b_k) = \sum_{i \in w_k} \left(\left(a_k I + b_k - p_i\right)^2 + \epsilon a_k^2 \right), \tag{3.10}$$

where ϵ is a regularization parameter to prevent a_k from being too large. The solution to Eq. (3.10) is found to be:

$$a_k = \frac{\frac{1}{|w|}\sum_{i\in w_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \tag{3.11a}$$

$$b_k = \bar{p}_k - a_k \mu_k \tag{3.11b}$$

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

$$q_i = \frac{1}{|w|} \sum_{k:i\in w_k} (a_k I_i + b_k) \tag{3.12}$$

$$= \bar{a}_i I_i + \bar{b}_i \tag{3.13}$$

He et al. further show that the Guided Filter can be expressed as in Eq. (3.7), with the kernel weights expressed as:

$$W_{ij}(I) = \frac{1}{|w|^2} \sum_{k:(i,j)\in w_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}\right). \tag{3.14}$$

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

The advantage to computing the filter output from Eq. (3.12) as opposed to explicitly computing the filter weights from Eq. (3.14) is that all the summations are box filters ($\sum_{i\in w_k} f_i$), so by applying the Integral Image technique from [25] the output is calculated in O(N) time, independent on window size. If the filter output is computed from Eq. (3.14), then the calculation time is also proportional to the square of the window size.

The Guided Filter is also extended to color guidance images by rewriting Eq. (3.9) as:

$$q_i = \mathbf{a_k^T I_i} + b_k, \quad \forall \quad i \in w_k, \tag{3.15}$$

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

$$\mathbf{a_k} = (\Sigma_k + \epsilon U)^{-1} \left(\frac{1}{|w|} \sum_{i \in w_k} \mathbf{I_i} p_i - \mu_\mathbf{k} \bar{p}_k \right) \tag{3.16a}$$

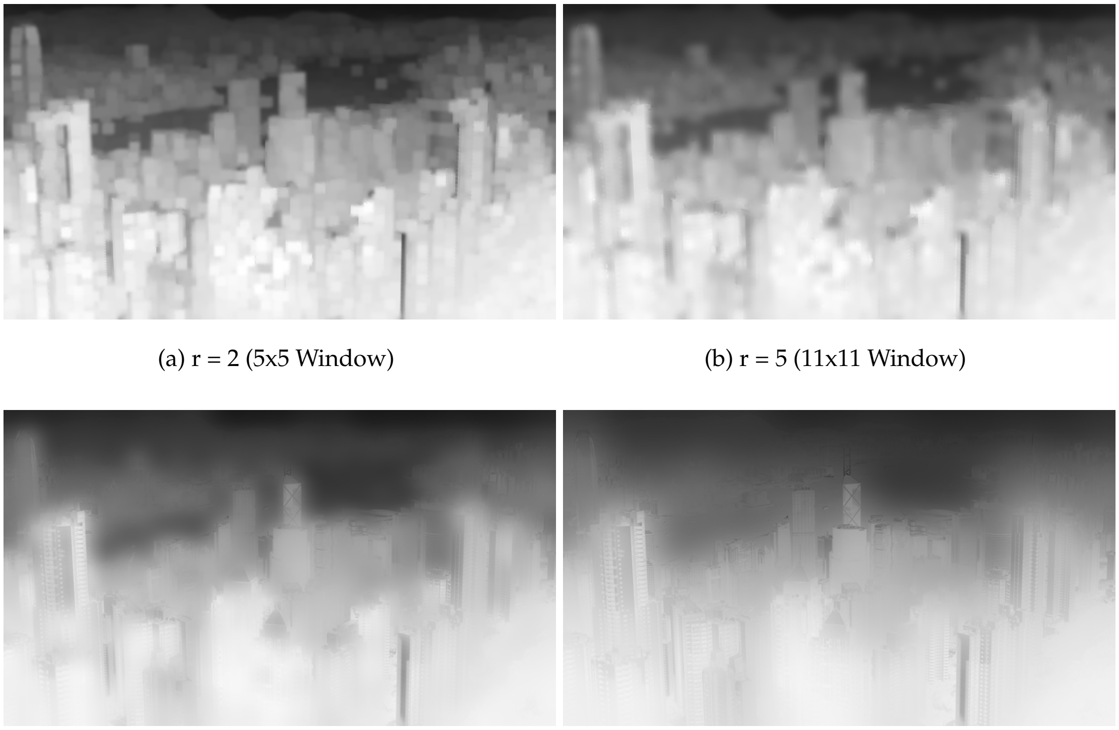
$$b_k = \bar{p}_k - \mathbf{a_k^T} \mu_\mathbf{k} \tag{3.16b}$$

$$q_i = \mathbf{\bar{a}_i^T I_i} + \bar{b}_i \tag{3.16c}$$

where Σ_k is a 3x3 covariance matrix of the colors in $\mathbf{I}$, U is a 3x3 identity matrix, and $\mu_\mathbf{k}$ is a 3x1 mean vector of the colors in $\mathbf{I}$. In the following section, we observe that the transmission map can be satisfactorily refined using the Guided Filter with either a color guidance image or a grayscale guidance image.

3.2.1 Transmission Map Refinement

The Guided Filter provides an approximation (first Jacobi iteration) to the result obtained from using the Matting Laplacian and solving Eq. (3.6), and reduces the problem of solving a large linear system of equations to a simple filtering process that can be computed in O(N) time. The Matting Laplacian typically uses 3x3 windows, which, although more accurate, tends to slow down the convergence of iterative solutions. He et al. show in [26] that using larger window sizes significantly improves the rate of convergence, and in most cases the accuracy of the result is close to that of using a small window size. Consequently, in using the Guided Filter, a large enough window size can be chosen to obtain a sufficiently refined transmission map without



(a) r = 2 (5x5 Window)

(b) r = 5 (11x11 Window)

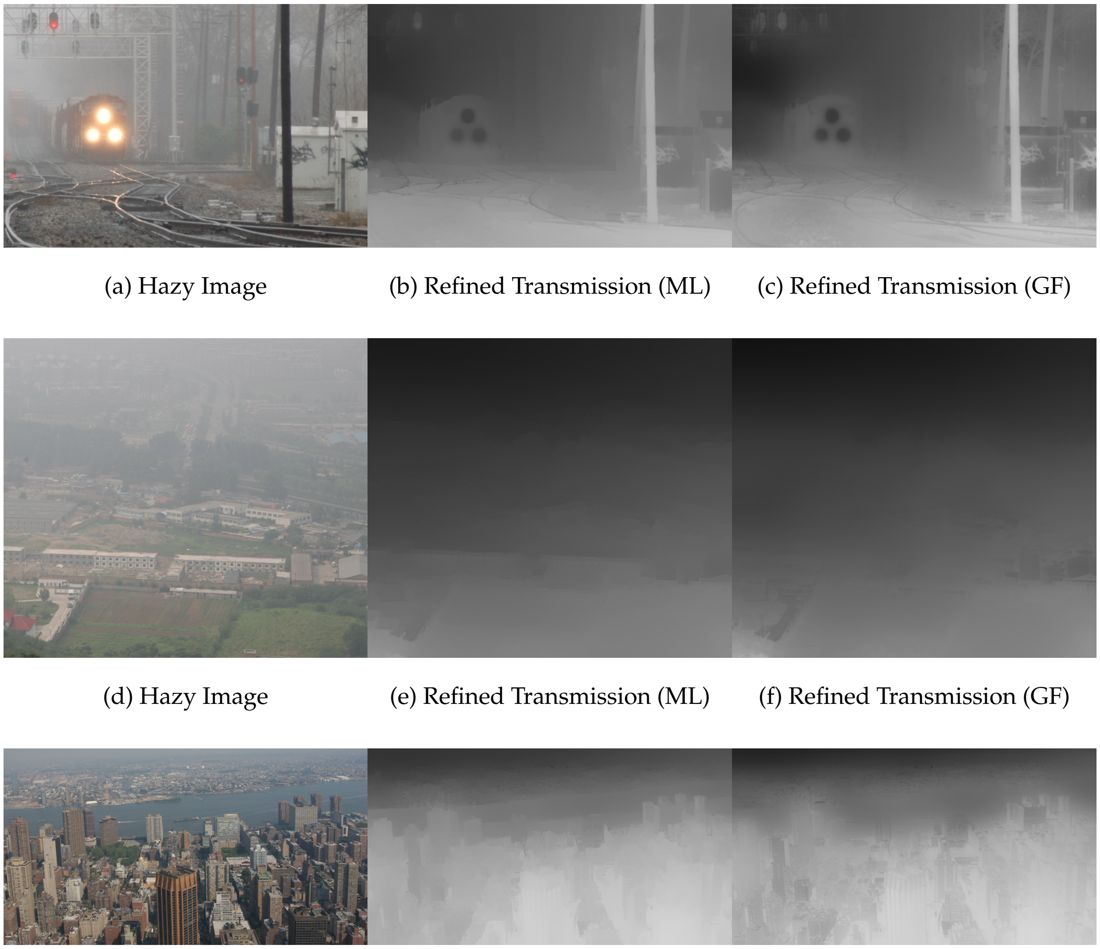
(c) r = 15 (31x31 Window)

(d) r = 30 (61x61 Window)

Figure 3.3: Refining the transmission map using the Guided Filter with various window sizes for the image from Fig. **3.1**. For this example, $\epsilon = 10^{-2}$.

the need for iterations. Fig. **3.3** shows a comparison of transmission refinement results using different window sizes.

We can see in Fig. **3.4** and Fig. **3.5** that the results obtained by the Guided Filter are not significantly visually different than the results obtained with the Matting Laplacian. Furthermore for a 600x400 image, using the color Guided Filter decreased computation time by a factor of 10, while using the grayscale Guided Filter decreased computation time by a factor of more than 200. A comparison between results using color and grayscale guidance images shows very little difference between the two approaches.

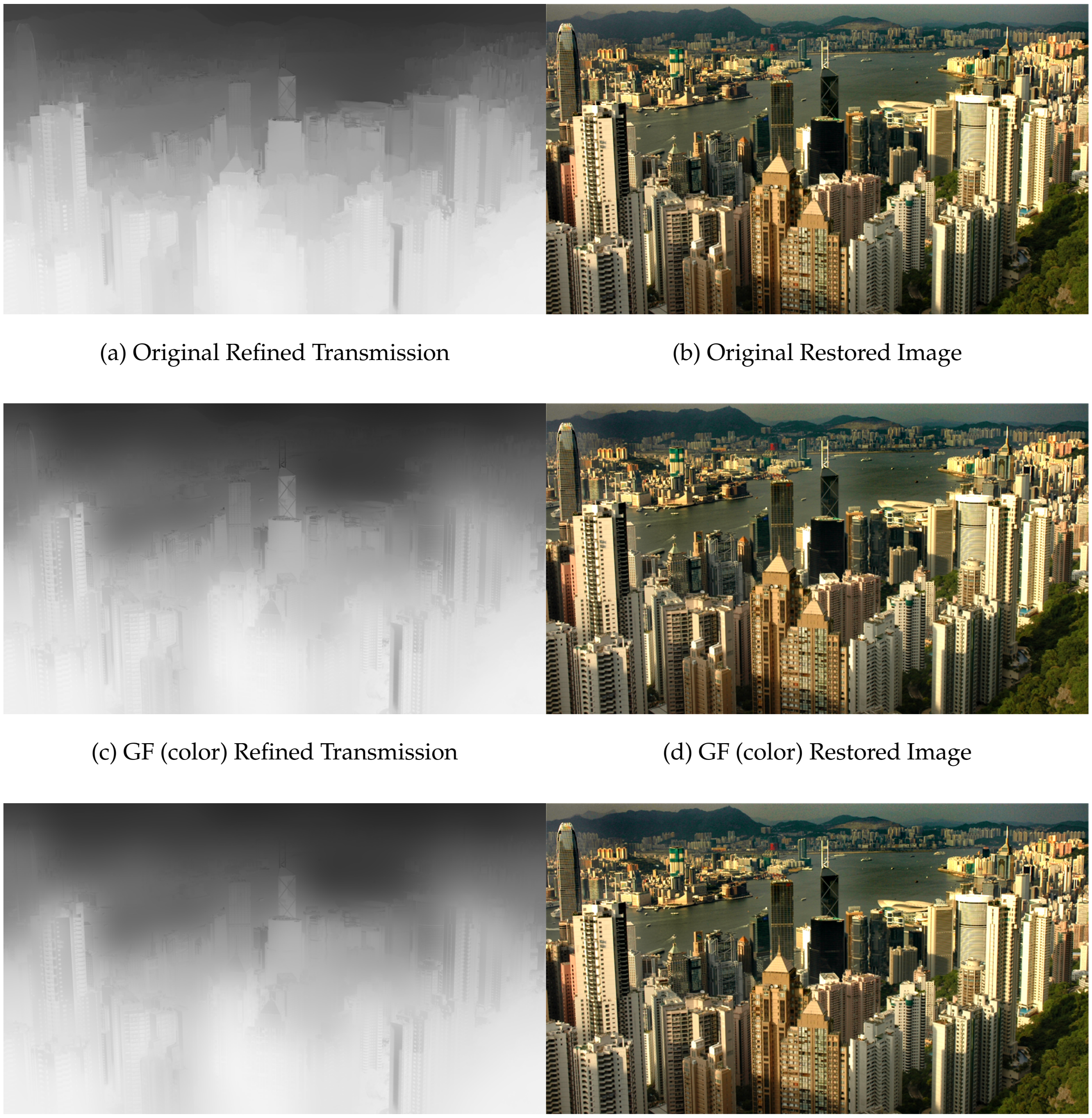


(a) Hazy Image (b) Refined Transmission (ML) (c) Refined Transmission (GF)

(d) Hazy Image (e) Refined Transmission (ML) (f) Refined Transmission (GF)

(g) Hazy Image (h) Refined Transmission (ML) (i) Refined Transmission (GF)

Figure 3.4: Comparison between results of the Matting Laplacian(ML) and Guided Filter(GF) for various images. Matting Laplacian parameters are $\epsilon = 10^{-4}$, and $\lambda = 10^{-3}$. Guided Filter parameters are $\epsilon = 10^{-2}$, and $r = 30$



(a) Original Refined Transmission

(b) Original Restored Image

(c) GF (color) Refined Transmission

(d) GF (color) Restored Image

(e) GF (gray) Refined Transmission

(f) GF (gray) Restored Image

Figure 3.5: Comparison Between Results from Matting Laplacian ($\epsilon = 10^{-4}, \lambda = 10^{-3}$) with Results from Guided Filter ($\epsilon = 10^{-2}, r = 30$)

3.3 Noisy Examples

Due to their formulations as regularized least squares problems, the Matting Laplacian and Guided Filter are generally robust to noise in the guide image. Furthermore, most of the actual transmission map estimation is performed using the dark channel prior (Chapter 2), with the refinement step simply serving to suppress halo effects, and not significantly alter the transmission values except around depth discontinuities. Thus, a low to moderate noise level in the guide image still results in an acceptably refined transmission map. Note, that only the transmission map refinement step is addressed here. Recovering the final scene radiance requires additional considerations, and is addressed in the following chapter.

Examples of refining the transmission map from a noisy guide image are seen in Fig. **3.6** for the Matting Laplacian and Fig. **3.7** for the Guided Filter. These examples show that the refined transmission maps are not significantly degraded by noisy guide images, with the refined transmission maps changing very little between noise levels. One area to note, however, is around the center building. When the noise level is strong some effects from the noise are observed on closer view. Some visual improvement can be gained by first using a denoising algorithm, such as BM3D, on the guide image. A close up view of this visual improvement is shown in Fig. **3.8**. Since the initial transmission map estimation requires a denoised image already (see Chapter 2), refinement using this preprocessed image adds no additional computation.

3.4 Summary

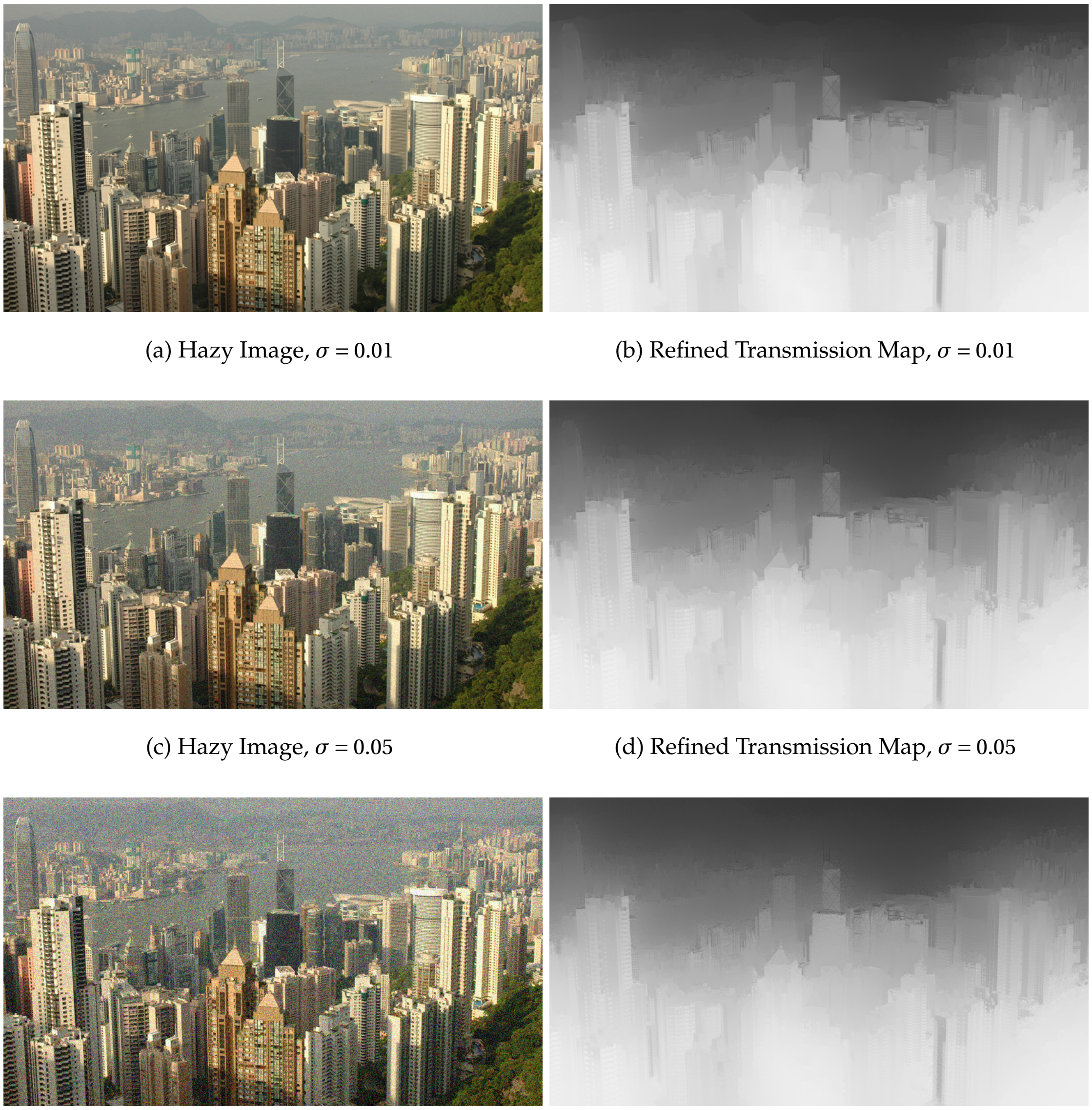
The initial transmission map estimate obtained using the dark channel prior is not suitable for direct scene radiance recovery because it results in undesirable artifacts, such as halos around depth discontinuities. Fortunately, this transmission map can be further refined to remove these artifacts.

The Matting Laplacian, which was originally derived for image matting, is based on the color line model, and provides a method to refine the transmission map, accurately capturing details in the source image. Results are visually satisfying, but the refinement process involves solving a large linear system, either by inverting a very large matrix, or using an iterative approach.

The Guided Filter provides an approximate solution to the Matting Laplacian, while also being much faster to compute. This filter takes advantage of an efficient method for performing simple box filtering to achieve an algorithm that is computed in O(N) time. While there are detectable differences between the refined transmission maps of the Guided Filter and the Matting Laplacian, final scene radiance results are visually comparable.

Both the Matting Laplacian and Guided Filter are robust to noise in the guide image; however, when the noise level is too high, artifacts can appear in the refined transmission map. Pre-filtering the guide image with a denoising algorithm reduces these artifacts. In the case of using BM3D the final results are visually similar to the results obtained using the original clean guide image.

This chapter also briefly discussed scene radiance recovery in the noise-free case. When noise is present in the hazy image, directly recovering the scene radiance results in significant noise amplification in hazy regions. The next chapter considers this problem and presents two methods for scene radiance recovery in the presence of noise.



(a) Hazy Image, $\sigma = 0.01$

(b) Refined Transmission Map, $\sigma = 0.01$

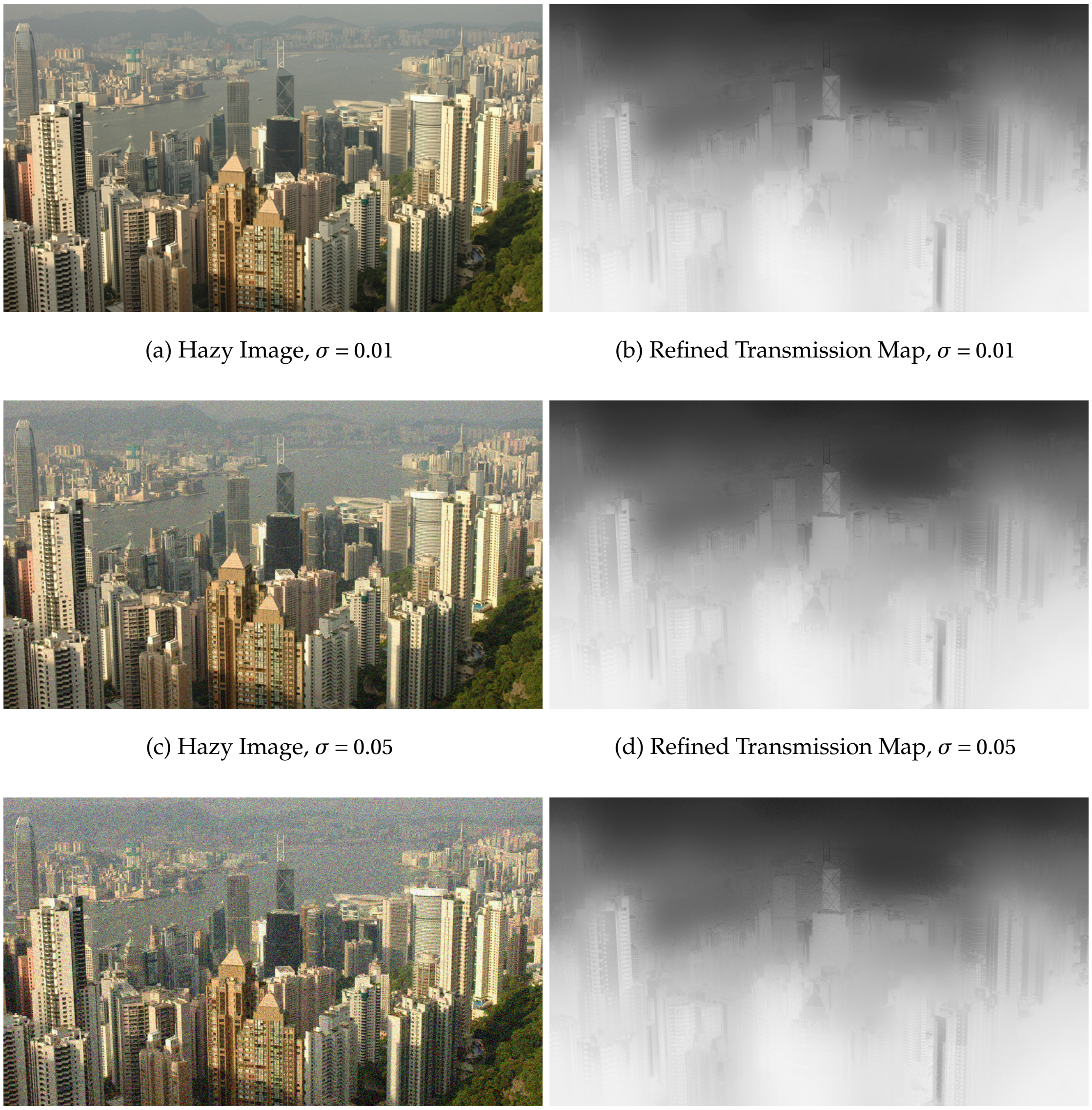
(c) Hazy Image, $\sigma = 0.05$

(d) Refined Transmission Map, $\sigma = 0.05$

(e) Hazy Image, $\sigma = 0.1$

(f) Refined Transmission Map, $\sigma = 0.1$

Figure 3.6: Results from refining the transmission map using the Matting Laplacian are robust to noise in the guide image. For this example, $\lambda = 10^{-3}$ and $\epsilon = 10^{-4}$. Gaussian noise with standard deviations ranging from 0.01 to 0.1 was added to the hazy image from Fig. **3.1**. The initial estimated transmission map was obtained from the clean image.



(a) Hazy Image, $\sigma = 0.01$

(b) Refined Transmission Map, $\sigma = 0.01$

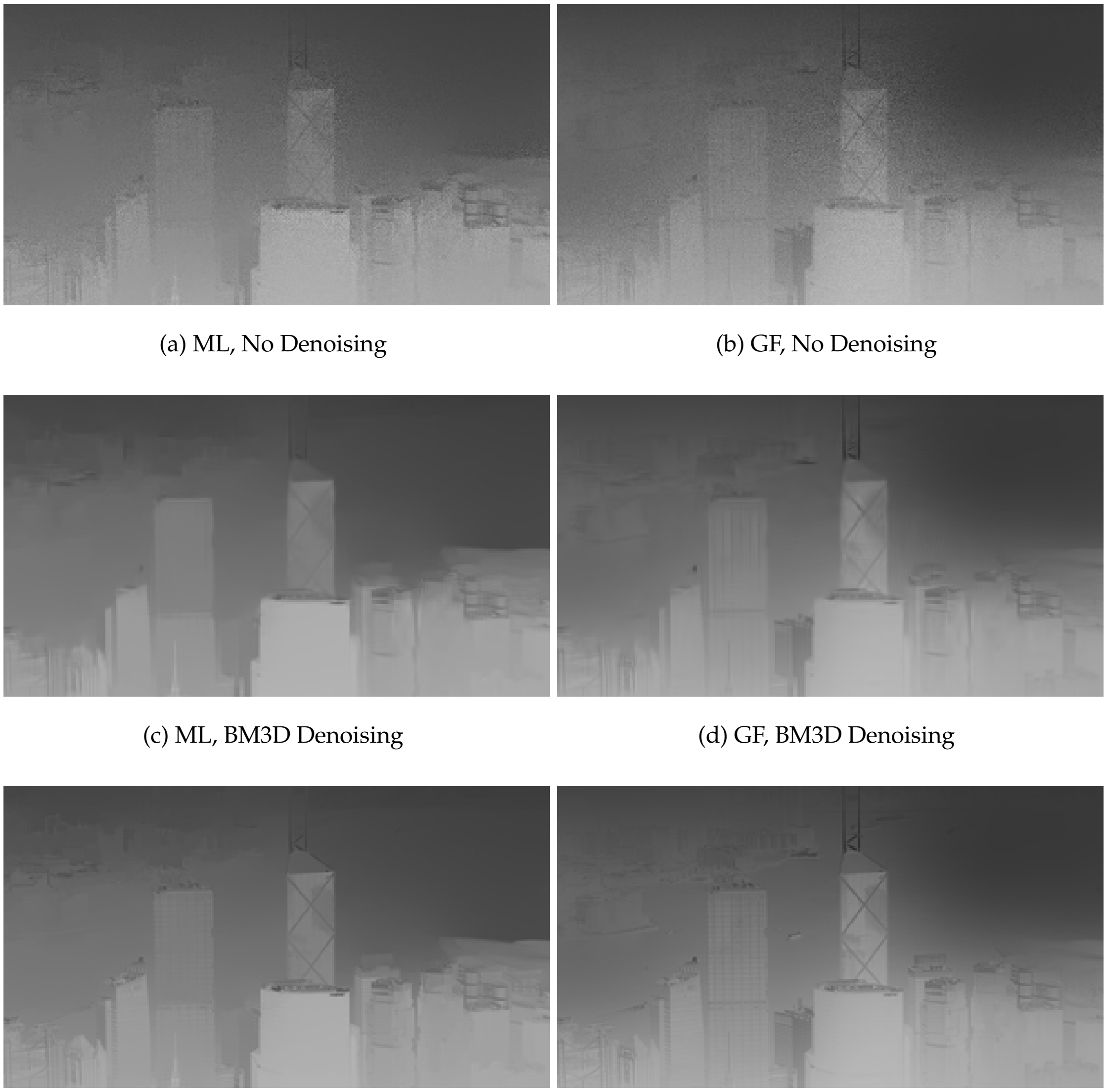
(c) Hazy Image, $\sigma = 0.05$

(d) Refined Transmission Map, $\sigma = 0.05$

(e) Hazy Image, $\sigma = 0.1$

(f) Refined Transmission Map, $\sigma = 0.1$

Figure 3.7: Results from refining the transmission map using the Guided Filter are robust to noise in the guide image. For this example, $\epsilon = 10^{-2}$, and $r = 30$. Gaussian noise with standard deviations ranging from 0.01 to 0.1 was added to the hazy image from Fig. **3.1**. The initial estimated transmission map was obtained from the clean image.


(a) ML, No Denoising

(b) GF, No Denoising

(c) ML, BM3D Denoising

(d) GF, BM3D Denoising

(e) ML, Clean Guide Image

(f) GF, Clean Guide Image

Figure 3.8: When noise is present in the guide image (Gaussian noise, standard deviation = 0.1), both the Matting Laplacian(ML) and Guided Filter(GF) show some artifacts in this close up view. After denoising the guide image with BM3D, the refinement results are visually similar to the results of using the original clean guide image. Matting Laplacian parameters are $\epsilon = 10^{-4}$, and $\lambda = 10^{-3}$. Guided Filter parameters are $\epsilon = 10^{-2}$, and $r = 30$

Chapter 4

Recovering Scene Radiance

The preceding chapters have discussed estimation of the two components necessary to recover the scene radiance: the atmospheric light and the transmission map. For these two cases, noise is simply additive and can be addressed by first preprocessing the image through a standard denoising algorithm. With these estimated components in hand, the final step in the dehazing process is to recover the underlying scene radiance. However, unlike the atmospheric light and transmission map, the noise contribution in the scene radiance is amplified proportionally to the amount of haze, and so is spatially varying according to the local haze content.

Thus the task of this chapter is to investigate scene radiance recovery in the presence of noise after estimates for the atmospheric light and transmission map are obtained [1]. First we provide a review of the exact relation of noise to haze content, including some experimental results of dehazing a noisy hazy image without denoising. Next we present two approaches to scene radiance recovery that have not been previ-

[1] Note that for all examples in this chapter, the Matting Laplacian is used for transmission map refinement with $\epsilon = 10^{-4}$ and $\lambda = 10^{-3}$.

ously considered. The first of these is the intuitive approach of treating haze and noise separately by first pre-processing the hazy image with the BM3D denoising algorithm followed by dehazing. Surprisingly, this has not been previously discussed in the literature, although [13] considers a standard denoising algorithm as a *post*-processing step. For the second approach we propose a novel iterative non-parametric method based on kernel regression to simultaneously perform denoising and dehazing. This approach is inspired as a simplification to the method proposed in [13]. Finally, this chapter concludes with a comparison of the two approaches. We will see that the first approach of denoising followed by dehazing achieves the best results when the estimated level of noise is accurate. However, due to the amplification of errors in scene radiance recovery, this approach is not robust to errors introduced in the denoising process from an inaccurate estimation of the noise level. For example, if it is over-estimated in the original image, the recovered scene radiance can suffer from significant oversmoothing in regions with dense haze. The kernel regression based approach, however, is more robust to this effect, and can achieve superior results in such cases, both in terms of mean squared error and subjective visual quality. In the case that the level of noise is known, it achieves results visually on par with the first method, although with somewhat higher mean squared error.

4.1 Naive Dehazing

When considering the effects of noise in the scene radiance recovery process, an important simplifying assumption is that the atmospheric light and transmission map are perfectly known. Although this is not generally the case, the conclusions drawn are still valid. Errors in the atmospheric light component will lead to some color biasing in the final image, which can be solved by performing white balancing as a post-processing step. Underestimating the transmission map results in some haze left in the image, while overestimating the transmission map leads to over-saturation. These effects can be suppressed by re-performing the dehazing process to remove additional haze, and by color post processing to reduce saturation.

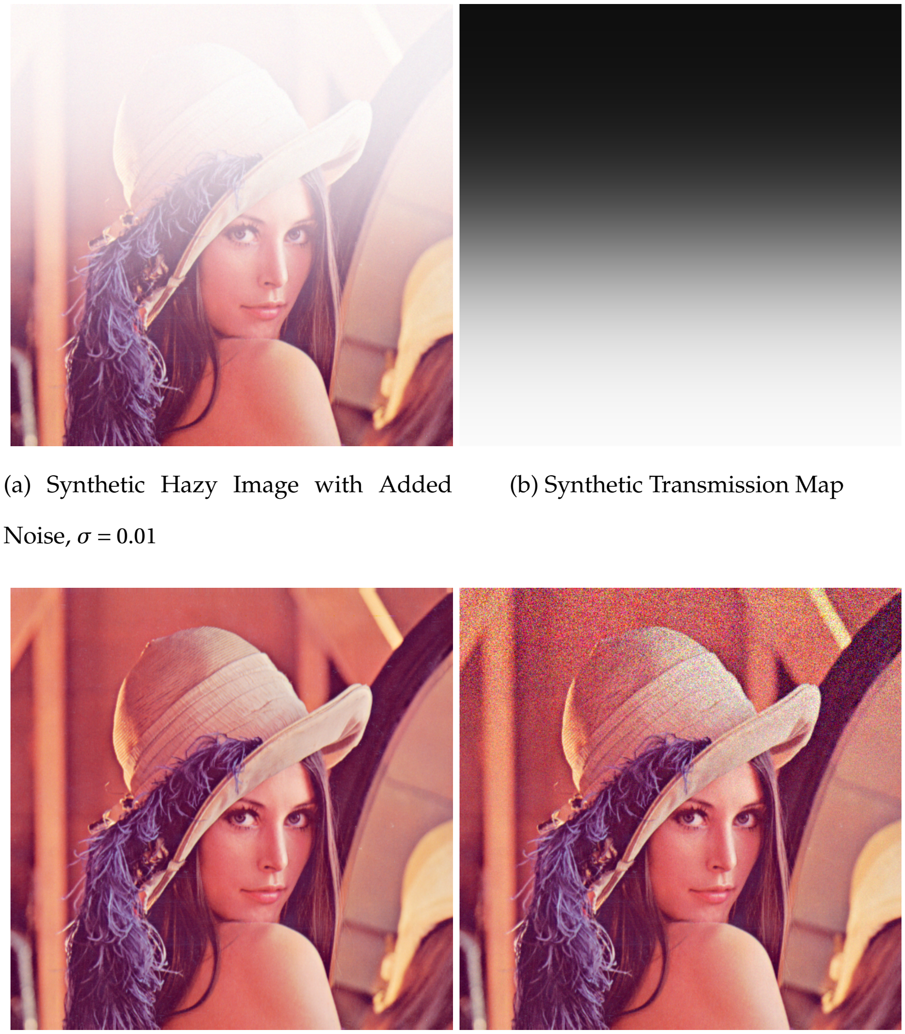
With the atmospheric light and transmission map perfectly known, Eq. (1.4) shows that when directly recovering the scene radiance, the noise contribution is amplified by $1/t(\mathbf{x})$, repeated here for convenience:

$$
\begin{aligned}
\mathbf{Y}(\mathbf{x}) &= \mathbf{I}(\mathbf{x}) + \mathbf{n}(\mathbf{x}) \\
\mathbf{Y}(\mathbf{x}) &= \mathbf{R}(\mathbf{x})\,t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) + \mathbf{n}(\mathbf{x}) \qquad (4.1)\\
\hat{\mathbf{R}}(\mathbf{x}) &= \frac{\mathbf{Y}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x}))}{t(\mathbf{x})} \\
&= \frac{\mathbf{R}(\mathbf{x})\,t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) + \mathbf{n}(\mathbf{x}) - \mathbf{a}_\infty(1 - t(\mathbf{x})))}{t(\mathbf{x})} \\
&= \mathbf{R}(\mathbf{x}) + \frac{\mathbf{n}(\mathbf{x})}{t(\mathbf{x})} \qquad (4.2)
\end{aligned}
$$

where $\mathbf{x}$ is a pixel location, $\mathbf{R}$ is the underlying scene radiance, $\hat{\mathbf{R}}$ is the estimated scene radiance, $\mathbf{n}$ is the noise contribution (assumed to be I.I.D. with mean 0 and variance σ^2), t is the transmission, and $\mathbf{a}_\infty$ is the atmospheric light.

Since t is a value between 0 and 1, Eq. (4.2) implies that except in the special case when haze is absent ($t = 1$), the noise contribution is amplified when recovering the scene radiance. Furthermore, in very hazy regions, where t is close to 0, the noise contribution can dominate the results. This implication is expressed in terms of noise variance by:

$$\sigma^2 \propto \frac{1}{t^2(\mathbf{x})} \qquad (4.3)$$



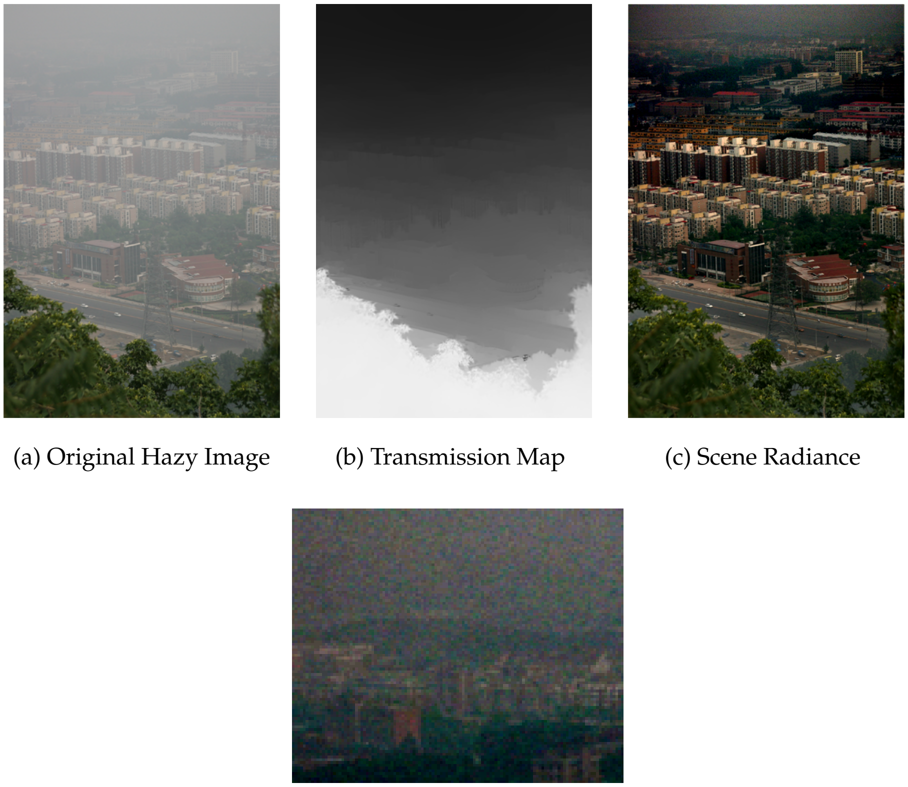
(a) Synthetic Hazy Image with Added Noise, $\sigma = 0.01$

(b) Synthetic Transmission Map

(c) Original Image

(d) Recovered Image

Figure 4.1: In this synthetic example, a small amount of noise is present in the hazy image (Gaussian, $\sigma = 0.01$). However, recovering the scene radiance using the ground truth transmission map Fig. **4.1**b and atmospheric light ($\mathbf{a}_\infty = [1, 1, 1]$) results in significant noise amplification in the hazier regions.



(a) Original Hazy Image (b) Transmission Map (c) Scene Radiance

(d) Close Up View

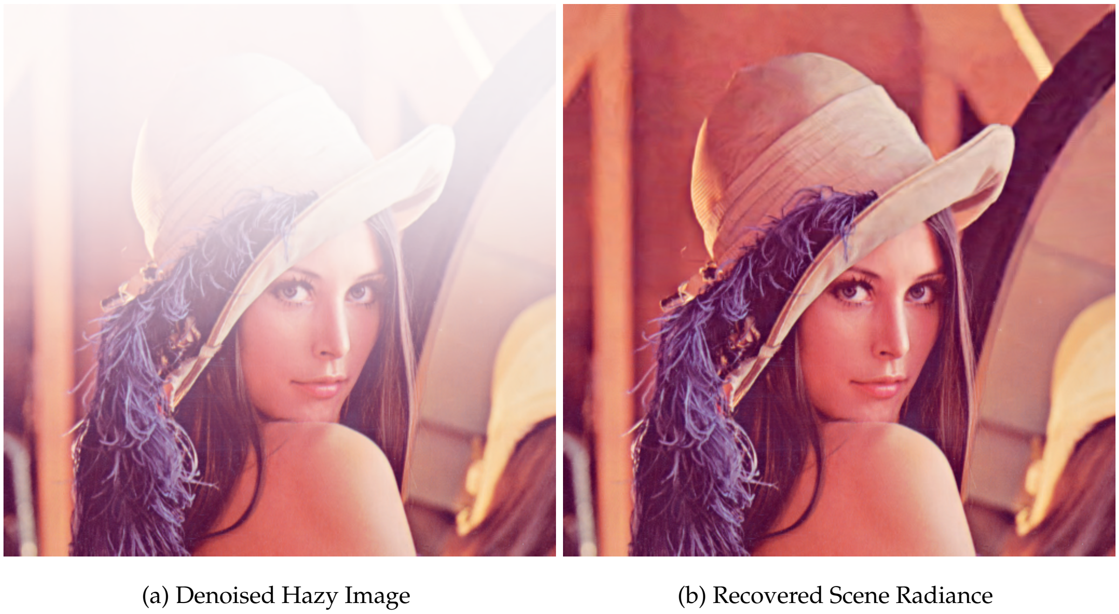
Figure 4.2: A close up view of the recovered scene radiance shows that in the most distant regions of the image (corresponding to the haziest), noise has become a significant contribution to the image, whereas in the original hazy image, the noise level is imperceptible. For this example, the transmission map was estimated after first denoising Fig. **4.2**a with BM3D using the best Q as a stopping parameter. The Matting Laplacian was used for transmission map refinement with the denoised image serving as the guide image ($\epsilon = 10^{-4}$, $\lambda = 10^{-3}$).

A synthetic example (Fig. **4.1**) with a small amount of added noise, shows that even when the transmission map and atmospheric light are perfectly known, the recovered scene radiance is significantly degraded by noise in hazy regions. It is remarkable that the added noise level is so small that it is imperceptible in the hazy image; yet, once the image is dehazed, the noise becomes quite apparent. A real example (Fig. **4.2**), shows that this is not simply a synthetic phenomenon. Using the methods outlined in Chapter 2 and Chapter 3, a convincing transmission map is obtained. Although the scene radiance is clear and contrast is restored in areas with little to moderate haze, in more distant regions, where haze is a dominant portion of the original image, there is significant noise amplification. Thus we have the motivation to suppress noise effects in the recovery process.

4.2 Denoising and Dehazing Separately

Simply denoising the hazy image as prior to performing the dehazing process is a natural approach to handling the problem of noise in scene radiance recovery. Prior to dehazing, we can treat our image model as: $\mathbf{Y} = \mathbf{I} + \mathbf{n}$, with the task of the denoising algorithm being only to estimate $\mathbf{I}$, which encapsulates the hazy image. In contrast, denoising as a *post*-processing step was considered in [13] using the Non-Local Means [27] denoising algorithm, with the authors finding that areas of the image containing little haze suffered from over-smoothing. This will be the case with most standard denoising algorithms, since they typically treat the noise level as homogeneous throughout the image, and we know that the noisy scene radiance requires adaptation to spatially varying noise.

Denoising as a first step is also convenient considering that accurately estimating the atmospheric light and transmission map already requires a denoised image. With this scheme in mind, we will examine some results using BM3D. As discussed in Chapter 2, BM3D [19] [20] is currently among the best performing denoising algorithms, and so should yield results that are about as good as one should expect for treating noise and haze separately. Returning to the synthetic example, the image in Fig. **4.1**a is first denoised (Fig. **4.3**a) using BM3D with the exact noise standard deviation, and then dehazed using the actual transmission map (Fig. **4.1**b). The resulting scene radiance is shown in Fig. **4.3**b. Comparing the result to Fig. **4.1**d shows a significant reduction in the noise level.



(a) Denoised Hazy Image

(b) Recovered Scene Radiance

Figure 4.3: The image shown in Fig. **4.1**a is first denoised using BM3D with the exact noise standard deviation, and then dehazed using the actual transmission map (Fig. **4.1**b). This process results in significant improvement in the recovered scene radiance.

The actual transmission map and atmospheric light are generally not given, and so Fig. **4.4** shows a simple block diagram for a more complete dehazing algorithm. Given a noisy hazy image, it is first denoised using BM3D. From this denoised version of the image, the atmospheric light, transmission map, and scene radiance are estimated.

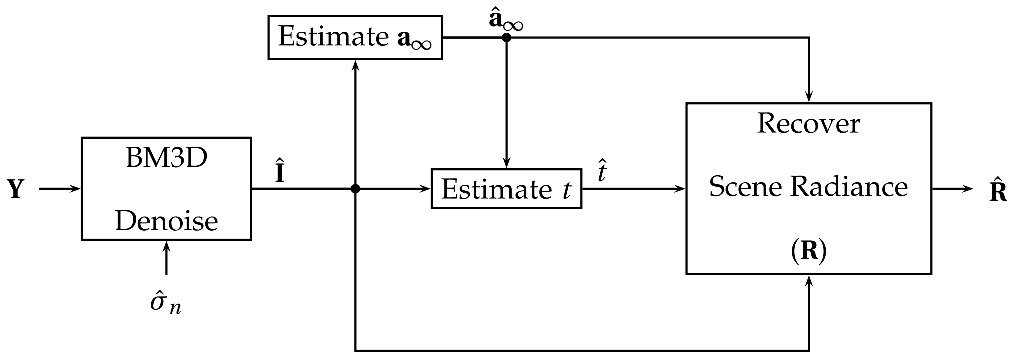
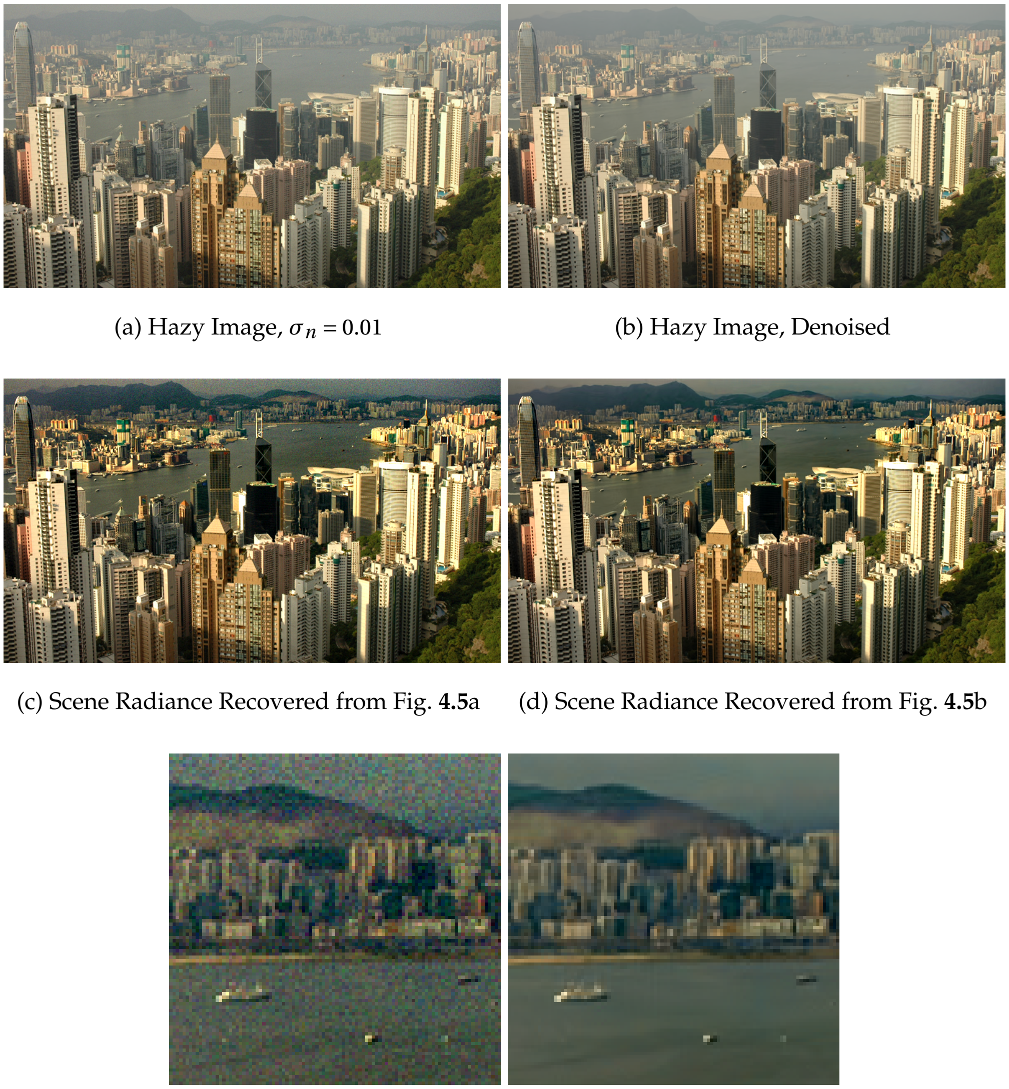


Figure 4.4: The noisy hazy image is first denoised using BM3D given the noise standard deviation, σ_n. From the denoised image, $\hat{\mathbf{I}}$, estimates for the atmospheric light ($\hat{\mathbf{a}}_\infty$) and transmission map ($\hat{t}$) are found. Using these estimates, direct dehazing is performed on $\hat{\mathbf{I}}$, yielding the estimated scene radiance, $\hat{\mathbf{R}}$

Example results of this process are shown in figure Fig. **4.5** using a real hazy image. A small amount of noise is added to the original image, and the known noise standard deviation is given to BM3D for denoising. Comparing the scene radiance recovered from the denoised image to the scene radiance recovered from the noisy image, a significant improvement is seen. A Monte Carlo simulation (Fig. **4.6**) shows that the MSE of the final result is much improved for this image over a range of noise levels for this image.

While BM3D is very effective at denoising when given the true noise standard deviation, for real noisy hazy images, this parameter is generally unknown. In addition, the presence of haze makes accurate estimation of the noise variance quite difficult. A truly complete denoising and dehazing scheme must include a method for



(a) Hazy Image, $\sigma_n = 0.01$

(b) Hazy Image, Denoised

(c) Scene Radiance Recovered from Fig. **4.5**a

(d) Scene Radiance Recovered from Fig. **4.5**b

(e) Close up of Fig. **4.5**c

(f) Close up of Fig. **4.5**d

Figure 4.5: The hazy image has a small amount of Gaussian noise added. It is denoised using BM3D given the exact σ_n, and the atmospheric light and transmission map are estimated from this. The scene radiance recovered from the denoised image is significantly improved in the distant image regions.

estimating the BM3D parameter. As discussed in Chapter 2, the Q metric [21] proposed by Zhu et al. is a suitable stopping criterion. A block diagram for a complete algorithm incorporating this metric is shown in Fig. **4.7**.

As outlined in [21], the anisotropic patch set is estimated from the noisy input image. This patch set is then used to compute the Q metric to evaluate the relative quality of the output of BM3D, while the BM3D denoising parameter, an estimate for the noise standard deviation, σ_n, is adjusted. The estimated σ_n that maximizes the Q metric is used for the final denoised image. This final denoised image is then used for all subsequent haze removal operations: estimating the atmospheric light, estimating

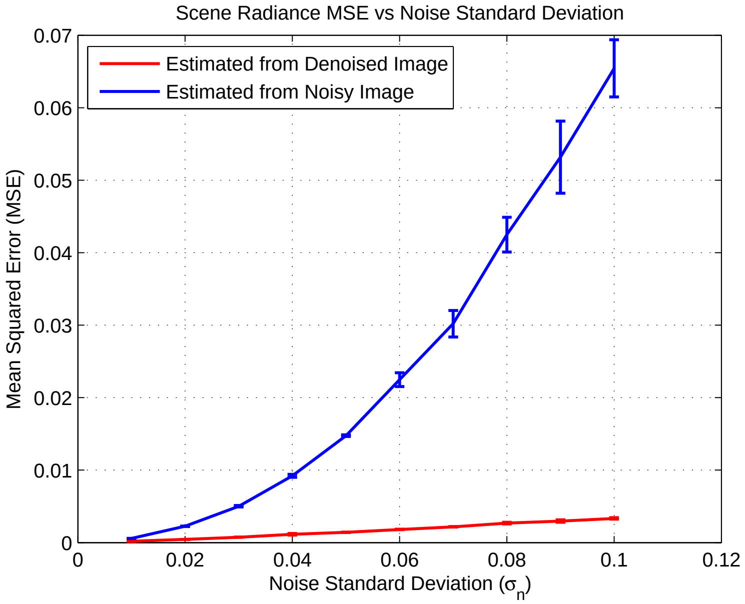


Figure 4.6: This 5-trial Monte Carlo Simulation for the image in **4.5** for various levels of noise, ranging from $\sigma_n = 0.01$ to $\sigma_n = 0.1$, shows that denoising using BM3D prior to performing any dehazing operations significantly reduces the MSE of the recovered scene radiance.

the transmission map, and finally recovering the scene radiance.

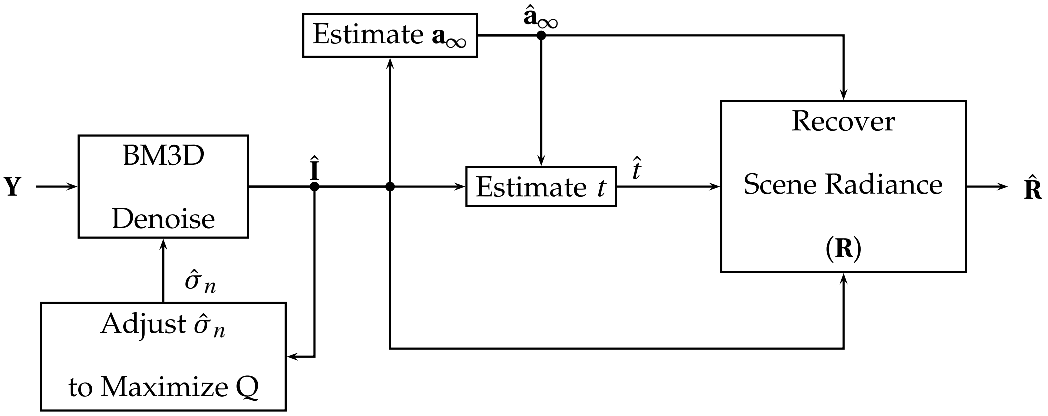


Figure 4.7: The noisy hazy image is denoised using BM3D while adjusting the estimated noise standard deviation, σ_n. The Q metric [21] measures the relative quality of the denoised images as σ_n is adjusted. The value of σ_n that maximizes Q is used for the final denoised image, $\hat{\mathbf{I}}$. From this denoised image, $\hat{\mathbf{I}}$, the atmospheric light ($\hat{\mathbf{a}}_\infty$) and transmission map ($\hat{t}$) are estimated. Using these estimates, direct dehazing is performed on $\hat{\mathbf{I}}$, yielding the estimated scene radiance, $\hat{\mathbf{R}}$

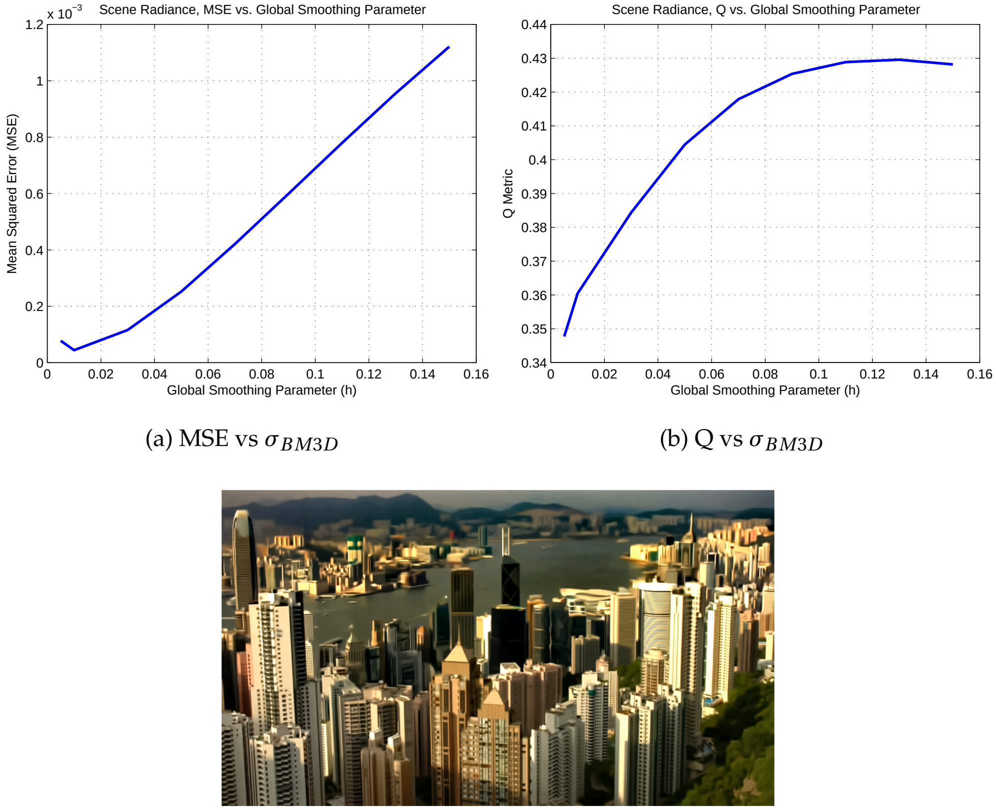
In the previous chapters it was noted that when estimating the atmospheric light and transmission map, it is far better to oversmooth than undersmooth. Since these parameters generally vary smoothly anyway, the results of estimating them from an oversmoothed image are similar to that of estimating them from an optimally smoothed image. However, since these parameters are sensitive to outliers, undersmoothing generally results in poor estimation. Unfortunately, this property does not hold for the scene radiance. There are typically well textured and detailed areas throughout the image that are severely degraded by oversmoothing. In contrast, if one does not smooth enough, any latent noise is amplified in the scene radiance. The experiment in

Fig. **4.8** shows this sensitivity for the hazy image from Fig. **4.5**a.

As mentioned previously, the Q metric typically offers good performance for tuning denoising algorithms when the level of noise is unknown, and is on par with or better than other techniques for this purpose [21]. Unfortunately, the introduction of haze makes distinguishing between whether an image is noisy or well textured more ambiguous, due to the attenuation of the scene radiance. This problem is exacerbated in low noise cases, where small quantitative errors in noise estimation may still result in a large relative difference. A particularly bad example of this is shown in Fig. **4.8**c, where maximizing Q in this case resulted in significant oversmoothing. One may speculate that since there are many regular structures in the foreground of this image, BM3D does well in avoiding oversmoothing these regions, which may lead the Q metric to under-penalize over-estimation of the noise level, thus allowing the quality of the background to suffer. A less extreme example is shown for a real noisy hazy image in Fig. **4.9**, where the smoothing level appears appropriate for most of the image, but upon examination of the background we can see some detail has been lost. So while the Q metric may perform well for a variety of images, we must acknowledge from these examples that it may not work equally well for every image.

4.3 Iterative Kernel Regression

In this section an iterative, non-parametric method based on kernel regression is proposed to simultaneously perform denoising and dehazing in order to recover the scene radiance. First we review both classical kernel regression [28][29] and locally adaptive kernel regression [30]. Next we propose an adaptation of the kernel regression model to the problems of both scene radiance estimation and atmospheric light



(a) MSE vs σ_{BM3D}

(b) Q vs σ_{BM3D}

(c) Scene Radiance from optimal (maximized) Q

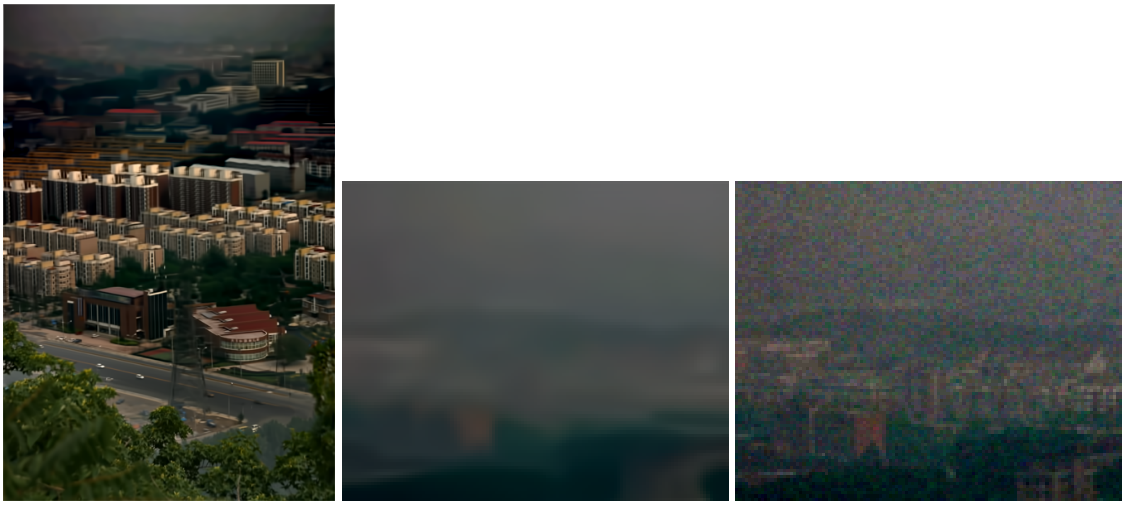
Figure 4.8: For this experiment the image in Fig. **4.5**a is used ($\sigma_n = 0.01$). The MSE of the final scene radiance is extremely sensitive to the BM3D Tuning Parameter. If Q is used as a stopping parameter for denoising then the resulting scene radiance (Fig. **4.8**c) is significantly oversmoothed. In this case, the estimated noise variance is $\hat{\sigma}_n = 0.13$. Compare this result to that of using the actual noise variance in figure Fig. **4.5**d.

estimation. Finally, the problem of selecting an adaptive smoothing parameter is addressed.

4.3.1 Review of Kernel Regression

4.3.1.1 Classic Kernel Regression

Classical parametric regression relies on estimating the parameters of a specific signal model given a set of data [31]. The final model given these estimated parameters is then used as an estimate of the underlying signal, which can be used for a variety of problems including upscaling and interpolation. One example commonly



(a) Recovered Scene Radiance (b) Close up, BM3D Recovery (c) Close up, Noisy Recovery

Figure 4.9: Using the real image example from figure Fig. **4.2**, the Q metric was used as a stopping parameter in denoising the image for the purposes of atmospheric light, transmission map, and scene radiance estimation. In this case, maximizing Q results in the estimated noise standard deviation: $\hat{\sigma}_n = 0.03$. The resulting scene radiance appears oversmoothed. Comparing the close up views of the hazy region for recovery from the denoised image to recovery from the noisy image reveals that some detail is lost in the background.

used is a first-order linear model, which can be described in vector notation by:

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x} + \boldsymbol{\varepsilon} \tag{4.4}$$

where **y** and **x** are column vectors of the observed data and their locations, β_0 and β_1 are parameters to be estimated, and $\boldsymbol{\varepsilon}$ is a vector of independent, zero-mean additive noise. In general, the parameters are often found using least squares. For the first-order linear example, this is expressed as:

$$\min_{\beta_0,\beta_1} \left\| \mathbf{y} - [\mathbf{1} \quad \mathbf{x}] \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \right\|^2 \tag{4.5}$$

A major limitation of this parametric approach is that the estimated signal is limited by the choice of global model. If the underlying signal is sufficiently complicated, natural images being a major example, it may be practically impossible to choose a model that fits all of the data well. Non-parametric methods overcome this limitation by allowing the data to ultimately dictate the structure of the model. In the case of kernel regression, this is known as a *regression function* [28]. Kernel regression can be thought of as locally weighted least squares, where the classic parametric regression approach is adapted to fit the data locally. More specifically, data at positions near the point of interest are given higher weights than data at positions far way from the point of interest. The general data model for kernel regression in 2-D is given by:

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad \forall\, \mathbf{x}_i \in w, \quad i = 1, \ldots, N \tag{4.6}$$

where y_i is a noisy measurement at spatial coordinate $\mathbf{x}_i = [x_{1i}, x_{2i}]^T$, $z(\,\cdot\,)$ is the unspecified regression function, ε_i is zero mean I.I.D. measurement noise, and w is a window around the point of interest at coordinate **x**, containing N samples. In general, the spatial coordinates may be randomly spaced, but for the sake of simplicity we will consider only fixed spacing in this thesis.

Although the exact form of $z(\,\cdot\,)$ may remain unspecified, a local Taylor expansion of this function is given by:

$$\begin{aligned} z(\mathbf{x}_i) &\approx z(\mathbf{x}) + \{\nabla z(\mathbf{x})\}^T (\mathbf{x}_i - \mathbf{x}) + \frac{1}{2}(\mathbf{x}_i - \mathbf{x})^T \{\mathscr{H} z(\mathbf{x})\}^T (\mathbf{x}_i - \mathbf{x}) + \cdots \\ &\approx \beta_0 + \boldsymbol{\beta}_1^T (\mathbf{x}_i - \mathbf{x}) + \boldsymbol{\beta}_2^T \text{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} + \cdots \end{aligned} \tag{4.7}$$

where ∇ is the gradient (2×1) operator, $\mathscr{H}$ is the Hessian (2×2) operator, and vech$(\,\cdot\,)$ is the half-vectorization operator, which lexicographically orders the lower triangular portion of a symmetric matrix in a vector, i.e.:

$$\text{vech}\left(\begin{bmatrix} a & b \\ b & d \end{bmatrix}\right) = \begin{bmatrix} a & b & d \end{bmatrix}^T \tag{4.8}$$

Therefore for a 2-D image, $\beta_0 = z(\mathbf{x})$, which is the pixel value at the coordinate of interest, and $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are vectors of partial derivatives, specifically:

$$\boldsymbol{\beta}_1 = \nabla z(\mathbf{x}) = \left[\frac{\partial z(\mathbf{x})}{\partial x_1}, \quad \frac{\partial z(\mathbf{x})}{\partial x_2}\right]^T \tag{4.9}$$

$$\boldsymbol{\beta}_2 = \frac{1}{2}\left[\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2}, \quad 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2}, \quad \frac{\partial^2 z(\mathbf{x})}{\partial x_2^2}\right]^T \tag{4.10}$$

Again, in contrast to classical parametric regression, where a global approach is taken, kernel regression involves a local approach. Whereas in Eq. (4.5), all of the data was given equal weight, kernel regression assigns higher weights to data near the point of interest. The least squares formulation of this idea is expressed in the following optimization problem:

$$\min_{\{\boldsymbol{\beta}_m\}_{m=0}^M} \sum_{i=1}^N \left[y_i - \beta_0 - \boldsymbol{\beta}_1^T (\mathbf{x}_i - \mathbf{x}) - \boldsymbol{\beta}_2^T \text{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \ldots\right]^2 K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) \tag{4.11}$$

and

$$K_{\mathbf{H}_i}(\mathbf{u}) = \frac{1}{\det(\mathbf{H}_i)} K(\mathbf{H}_i^{-1}\mathbf{u}) \tag{4.12}$$

where M is the regression order. $K(\,\cdot\,)$ is the so-called *kernel function* that penalizes pixels according to their distance from the point of interest, and $\mathbf{H}_i$ is the smoothing matrix (2×2), controlling the strength of this penalty, with the simplest choice being $\mathbf{H}_i = h\mathbf{I}$, where h is known as the *global smoothing parameter*. $K(\,\cdot\,)$ is a radially symmetric function with a maximum at zero. Although there are “optimal" kernel functions, in the case of classic kernel regression, it is known that the exact choice plays only a small role in estimation accuracy [32]. So generally, a kernel function is chosen given other considerations, such as ease of computation or having convenient properties. In this thesis a Gaussian function is used.

Smoothing parameter selection, however, does play an important role in estimation accuracy, and there are a variety methods described in the literature [32] [28] [29] for choosing one. Since these methods often involve dependence on unknown parameters, the smoothing parameter is sometimes chosen empirically such that it satisfies some quality measure, which is the approach taken in this thesis. Intuitively, the smoothing parameter is generally inversely proportional to the amount of texture in an image (e.g. a well textured image should have a relatively small h).

Now returning to the estimation problem, it is convenient to reformulate Eq. (4.11) as a weighted least squares problem in vector notation:

$$\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}} \,(\,\mathbf{y} - \mathbf{X}\mathbf{b}\,)^T\, \mathbf{K}\,(\,\mathbf{y} - \mathbf{X}\mathbf{b}\,) \tag{4.13}$$

where

$$\mathbf{y} = \left[y_1, y_2, \ldots, y_N\right]^T, \quad \mathbf{b} = \left[\beta_0, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_M^T\right]^T \tag{4.14}$$

$$\mathbf{K} = \text{diag}\left[K_{\mathbf{H}_1}(\mathbf{x}_1 - \mathbf{x}), K_{\mathbf{H}_2}(\mathbf{x}_2 - \mathbf{x}), \ldots, K_{\mathbf{H}_N}(\mathbf{x}_N - \mathbf{x})\right] \tag{4.15}$$

$$\mathbf{X} = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_1 - \mathbf{x})(\mathbf{x}_1 - \mathbf{x})^T\right\} & \cdots \\ 1 & (\mathbf{x}_2 - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_2 - \mathbf{x})(\mathbf{x}_2 - \mathbf{x})^T\right\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (\mathbf{x}_N - \mathbf{x})^T & \text{vech}^T\left\{(\mathbf{x}_N - \mathbf{x})(\mathbf{x}_N - \mathbf{x})^T\right\} & \cdots \end{bmatrix} \tag{4.16}$$

with “diag" defining a diagonal matrix. The solution to Eq. (4.13) is:

$$\widehat{\mathbf{b}} = (\mathbf{X}^T \mathbf{K} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{K} \mathbf{y} \tag{4.17}$$

and the estimate of the pixel value of interest is therefore:

$$\hat{z}(\mathbf{x}) = \widehat{\beta}_0 = \mathbf{e}_1^T \widehat{\mathbf{b}} = \sum_{i=1}^{N} W_i(K, \mathbf{H}_i, M, \mathbf{x}_i - \mathbf{x}) y_i \tag{4.18}$$

where $\mathbf{e}_1^T$ is a $[M \times 1]$ column vector with the first element equal to 1 and the rest equal to 0. W is called the *equivalent kernel* [28][30], and

$$\sum_{i=1}^{N} W_i(\cdot) = 1 \tag{4.19}$$

Thus the pixel estimate is simply a local linear combination of nearby samples. A 0th order regression model ($M = 0$) results in the well known *Nadaraya-Watson* estimator [33], for which Eq. (4.18) becomes:

$$\hat{z}(\mathbf{x}) = \widehat{\beta}_0 = \frac{\sum_{i=1}^{N} K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) y_i}{\sum_{i=1}^{N} K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x})} \tag{4.20}$$

“Classic" kernel regression always results in a local **linear** combination of nearby samples, and so is inherently limited. For highly complex data, such as images, this property is a significant drawback. This is the motivation behind “steering" kernel regression, proposed in [30] and summarized in the following section. The main idea is that the kernel function should adapt to the local structure of the data, resulting in a non-linear adaptive filtering operation.

4.3.1.2 Steering Kernel Regression

Takeda et al. proposed steering kernel regression in [30] as a form of data-adapted kernel regression, in which the shape of the kernel is adapted to the local structure of the data, i.e. the kernel not only depends on sample locations, but also on the sample values themselves. More specifically, the smoothing matrix is redefined as:

$$\mathbf{H}_i^s = h\mathbf{C}_i^{-1/2} \tag{4.21}$$

and is called the *steering* matrix, where h is a global smoothing parameter. The matrix, $\mathbf{C}_i$, is estimated from the covariance matrix of the local gradient vectors. A naive estimate is obtained by:

$$\hat{\mathbf{C}}_i = \mathbf{G}_i^T \mathbf{G}_i = \begin{bmatrix} \sum_i z_{x_1}^2(\mathbf{x}_i) & \sum_i z_{x_1}(\mathbf{x}_i) z_{x_2}(\mathbf{x}_i) \\ \sum_i z_{x_1}(\mathbf{x}_i) z_{x_2}(\mathbf{x}_i) & \sum_i z_{x_2}^2(\mathbf{x}_i) \end{bmatrix} \tag{4.22}$$

with

$$\mathbf{G}_i = \begin{bmatrix} z_{x_1}(\mathbf{x}_1) & z_{x_2}(\mathbf{x}_1) \\ \vdots & \vdots \\ z_{x_1}(\mathbf{x}_N) & z_{x_2}(\mathbf{x}_N) \end{bmatrix} \tag{4.23}$$

where $z_{x_1}(\cdot)$ and $z_{x_2}(\cdot)$ are the first derivatives measured along the x_1- and x_2-axes, and N is the number of samples in a local window around the position of interest, $\mathbf{x}_i$. Since this naive estimate can be rank deficient or unstable, Takeda et al. propose a regularization based on the singular value decomposition of $\mathbf{G}_i$ following work in

[34]. Using the truncated singular value decomposition, the local gradient matrix can be rewritten as:

$$\mathbf{G}_i = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T = \mathbf{U}_i \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} [\mathbf{v}_1 \quad \mathbf{v}_2]^T \tag{4.24}$$

where **U** and **V** are orthonormal matrices. The column vector $\mathbf{v}_1$ represents the dominant gradient orientation, while $\mathbf{v}_2$ represents the dominant "edge" orientation, with s_1 and s_2 representing the respective energies of these directions. From this singular value decomposition, the gradient covariance matrix can thus be expressed as:

$$\hat{\mathbf{C}}_i = s_1^2 \mathbf{v}_1 \mathbf{v}_1^T + s_2^2 \mathbf{v}_2 \mathbf{v}_2^T = s_1 s_2 \left(\frac{s_1}{s_2} \mathbf{v}_1 \mathbf{v}_1^T + \frac{s_2}{s_1} \mathbf{v}_2 \mathbf{v}_2^T \right) \tag{4.25}$$

From this form, $s_1 s_2$ is interpreted as an overall scaling parameter, while the ratio $\frac{s_1}{s_2}$ is interpreted as an elongation parameter, e.g. when $s_1 = s_2$ the patch is isotropic, whereas when $s_1 > s_2$ the patch has a dominant direction. Instability is reduced by regularizing Eq. (4.25) as follows:

$$\mathbf{C}_i = \gamma \left(\rho \mathbf{v}_1 \mathbf{v}_1^T + \frac{1}{\rho} \mathbf{v}_2 \mathbf{v}_2^T \right) \tag{4.26}$$

with

$$\rho = \frac{s_1 + \lambda'}{s_2 + \lambda'} \tag{4.27}$$

$$\gamma = \left(\frac{s_1 s_2 + \lambda''}{N} \right)^{1/2} \tag{4.28}$$

λ' restricts the ratio ρ from becoming degenerate, while keeping the kernel circular in flat areas ($s_1 \approx s_2 \approx 0$) and elongated near edge areas ($s_1 \gg s_2$). λ'' dampens the effect of noise and keeps γ from becoming zero [2]. Further intuition behind Eq. (4.28) is to reduce noise effects while keeping the footprint large in flat areas and small in textured areas

[2] In this thesis, λ' is fixed to 0.005, and λ'' is fixed to 10^{-7}.

in order to produce sharp images. This second point follows from the fact that the singular values are smaller in flat areas where gradients are small and large in textured areas where gradients are large.

Finally, the so-called locally adaptive regression kernel (LARK) is constructed from a Gaussian function and expressed as the following:

$$K_{\mathbf{H}_i^s}(\mathbf{x}_i - \mathbf{x}) = \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2}\right) \tag{4.29}$$

The expression $(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})$ essentially encodes the geodesic distance from the point of interest [35][36]. And so in contrast to classical regression, which penalizes distance only along spatial coordinates, the kernel in Eq. (4.29) penalizes distance along a 2-D manifold embedded in 3-D space. Fig. **4.10** shows how the shape of LARK adapts to the local structure of the image, and are quite robust to noise.

4.3.2 Kernel Regression Model for Haze Removal

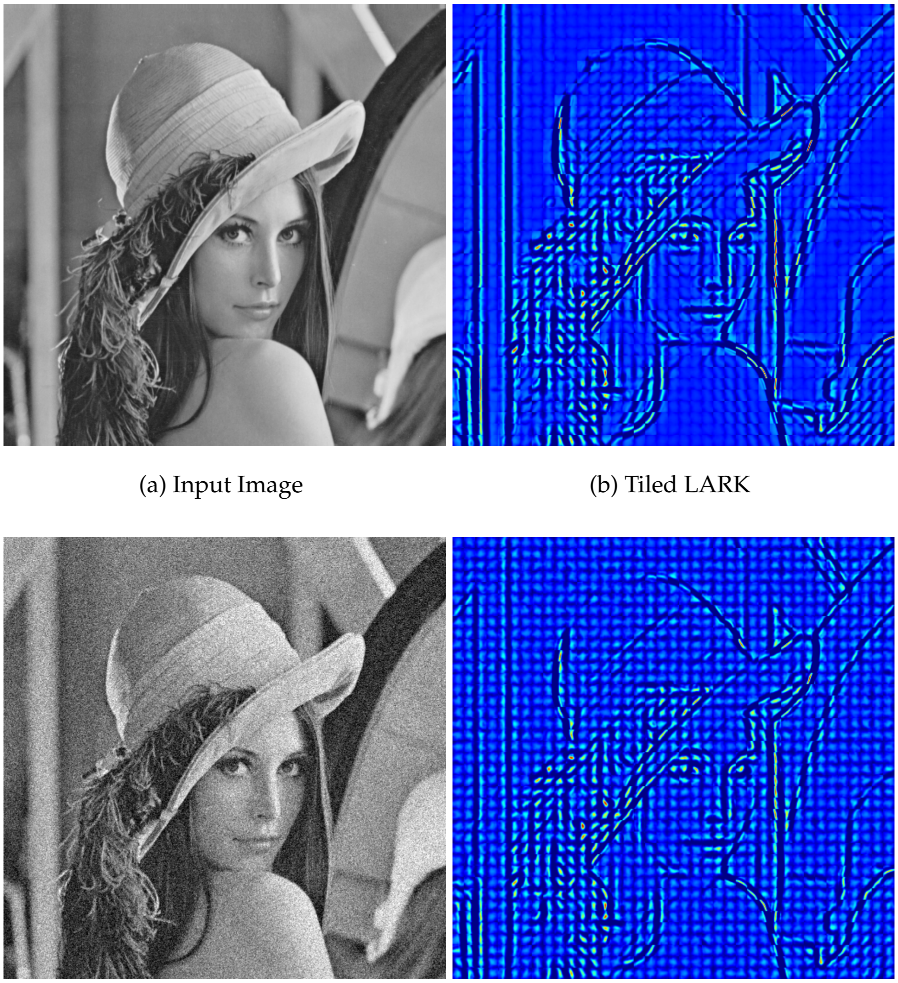
In the above discussion, the underlying signal is only corrupted by noise. However, in the problem of haze removal, the underlying signal is also attenuated according to its transmission value. Furthermore, the transmission map must be *estimated* from the noisy image. With this in mind, we propose an iterative method based on kernel regression for estimating the scene radiance and transmission map simultaneously. First, we recall the noisy hazy image model:

$$\mathbf{Y}(\mathbf{x}) = \mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{a}_\infty(1 - t(\mathbf{x})) + \mathbf{n}(\mathbf{x}) \tag{4.30}$$

In order to simplify some later notation, we rewrite this as:

$$\mathbf{Y}(\mathbf{x}) = \mathbf{R}(\mathbf{x})\, t(\mathbf{x}) + \mathbf{A}(\mathbf{x}) + \mathbf{n}(\mathbf{x}) \tag{4.31}$$

where $\mathbf{A}(\mathbf{x}) = \mathbf{a}_\infty(1 - t(\mathbf{x}))$. In contrast to $\mathbf{a}_\infty$, which is the *global* atmospheric light, $\mathbf{A}(\mathbf{x})$ can be considered the *local* atmospheric light, since it describes the atmospheric light contribution on a per pixel basis. Following the non-parametric kernel regression strat-



(a) Input Image

(b) Tiled LARK

(c) Input Image with Added Noise ($\sigma = 0.1$)

(d) Tiled LARK from Noisy Image

Figure 4.10: An image composed of non-overlapping locally adaptive regression kernels (LARK) shows how they capture the local image structure. Even in the presence of noise, the kernels maintain their basic structure.

egy, we can write our estimation problem for each color channel as:

$$\min_{A^c,R^c} \sum_{x_i \in \Omega(\mathbf{x})} \left[Y^c(\mathbf{x}_i) - R^c(\mathbf{x}_i)\, t(\mathbf{x}_i) - A^c(\mathbf{x}_i) \right]^2 K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) \tag{4.32}$$

where $\Omega(\mathbf{x})$ indicates a neighborhood around the coordinate of interest, $\mathbf{x}$, and the superscript, c, indicates a specific color channel of $\mathbf{Y}$, $\mathbf{R}$, or $\mathbf{A}$. $K_{H_i^s}$ indicates the locally adaptive regression kernel defined above. When considering a color image, the steering matrices are computed on the luminance channel, and are applied to all color channels. Note also that $t(\mathbf{x}) = 1 - \frac{A^c(\mathbf{x})}{a_\infty^c}$ where again c indicates a specific color channel. For simplification purposes, we assume that we have already estimated $\mathbf{a}_\infty$.

Since Eq. (4.32) is a minimization over two unknowns, the strategy is to find the solution iteratively, by decomposing it into two separate minimization problems, and alternating between fixing R^c and fixing A^c. Furthermore, we assume a 0th order regression model:

$$\min_{R^c} \sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} \left[Y'^c(\mathbf{x}_i) - t(\mathbf{x}_i) R^c(\mathbf{x}) \right]^2 K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) \tag{4.33}$$

$$\min_{A^c} \sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} \left[Y''^c(\mathbf{x}_i) - P^c(\mathbf{x}_i) A^c(\mathbf{x}) \right]^2 K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) \tag{4.34}$$

where $Y'^c = Y^c - A^c$, $Y''^c = Y^c - R^c$, and $P^c = 1 - \frac{1}{a_\infty^c} R^c$. Eq. (4.33) and Eq. (4.34) are simple weighted least-squares problems, and their solutions are:

$$\hat{R}^c(\mathbf{x}) = \frac{\sum\limits_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x})\, t(\mathbf{x}_i)\, Y'^c(\mathbf{x}_i)}{\sum\limits_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x})\, t(\mathbf{x}_i)^2} \tag{4.35}$$

$$\hat{A}^c(\mathbf{x}) = \frac{\sum\limits_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) P^c(\mathbf{x}_i)\, Y''^c(\mathbf{x}_i)}{\sum\limits_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) P^c(\mathbf{x}_i)^2} \tag{4.36}$$

So now our estimation problem is reduced to a set of linear filtering operations. Note that although the filtering operation itself appears linear, since the steering kernels are computed on the received data, the result is a non-linear filter.

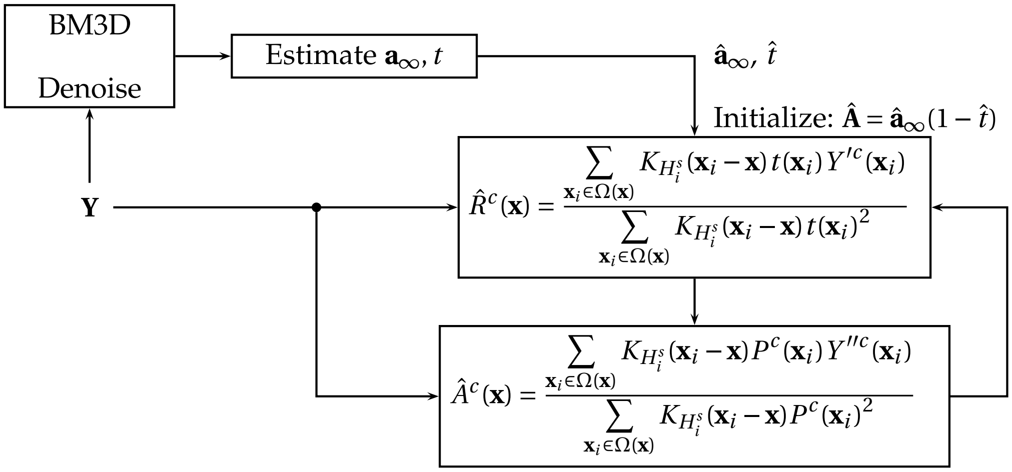
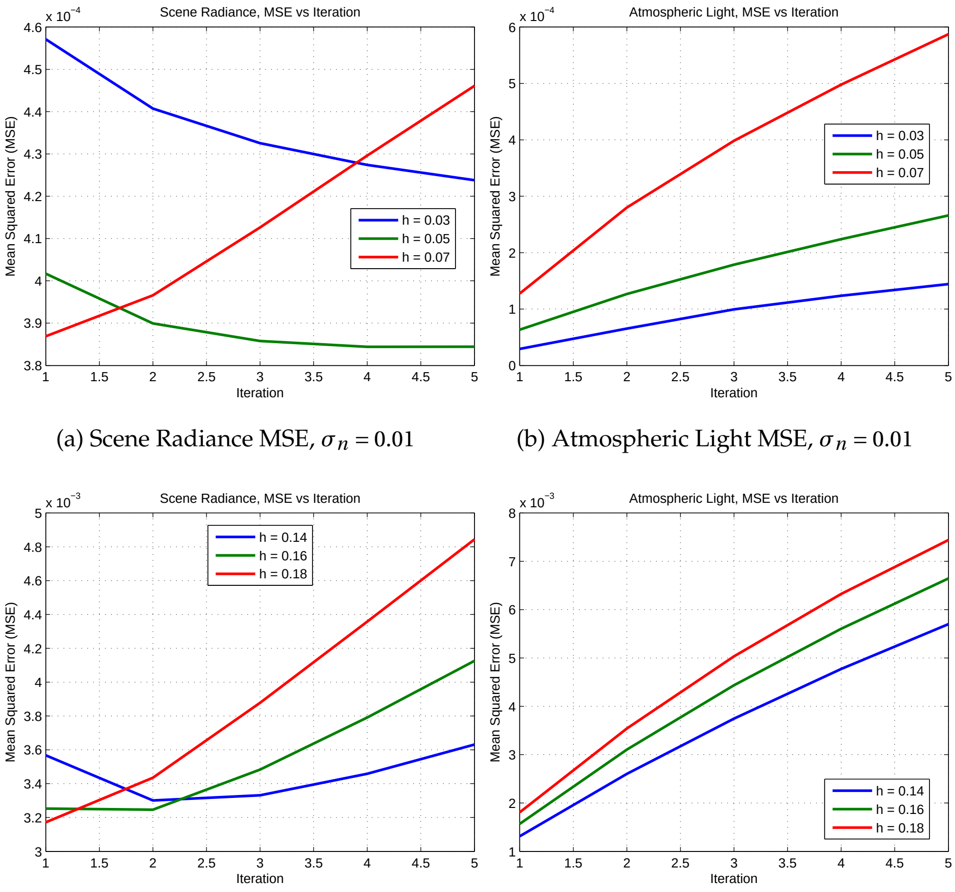

BM3D
Denoise
Estimate $\mathbf{a}_{\infty}, t$
$\hat{\mathbf{a}}_{\infty}, \hat{t}$
Initialize: $\hat{\mathbf{A}} = \hat{\mathbf{a}}_{\infty}(1 - \hat{t})$
$\mathbf{Y}$
$\hat{R}^c(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x})\, t(\mathbf{x}_i)\, Y'^c(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x})\, t(\mathbf{x}_i)^2}$
$\hat{A}^c(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) P^c(\mathbf{x}_i)\, Y''^c(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} K_{H_i^s}(\mathbf{x}_i - \mathbf{x}) P^c(\mathbf{x}_i)^2}$

Figure 4.11: The atmospheric light and transmission map are estimated from the denoised input image. These estimates are used to initialize the kernel regression iterations. The kernel regression iterations are performed on the noisy image, alternating between **A** and **R** until either MSE is minimized or Q is maximized for **R**.

The complete scene radiance estimation procedure is described as follows. Because Eq. (4.35) and Eq. (4.36) have an interdependence, in that an estimate for one requires an estimate of the other, the iterations are initialized with the atmospheric light and transmission map estimated from the BM3D denoised image, as in Sec. 4.2. For all examples shown in this thesis the locally adaptive kernel described in Eq. (4.29) is used. Image gradients for the steering kernels are computed via second order classic kernel regression with a simple Gaussian kernel on a pilot estimate of the scene radiance, which is from dehazing the denoised image as described in Section Sec. 4.2. Finally, since the ultimate goal is to estimate the scene radiance, the iterations are performed until the best $\hat{\mathbf{R}}(\mathbf{x})$ is found. In synthetic cases, since the ground truth is known, the best $\hat{\mathbf{R}}(\mathbf{x})$ is defined as the one that minimizes MSE; however, when access to the ground truth is not possible, maximizing the Q metric is used as a stopping criterion.

A simplified block diagram of the complete process is shown in Fig. **4.11**.



(a) Scene Radiance MSE, $\sigma_n = 0.01$

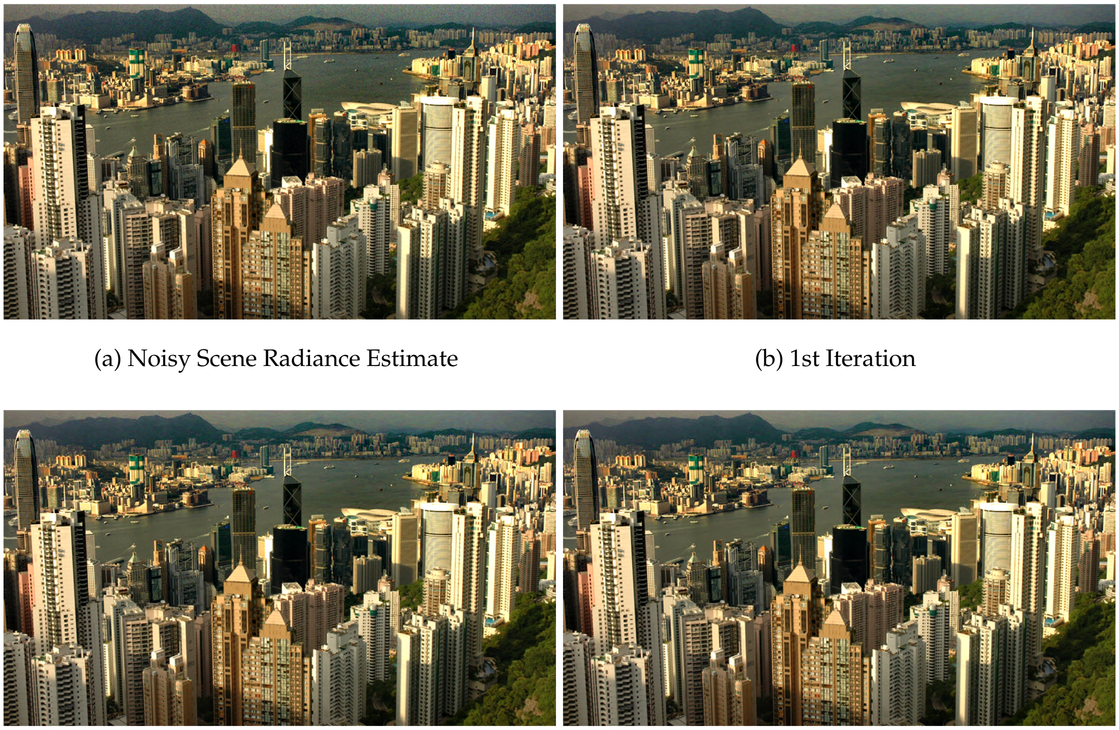
(b) Atmospheric Light MSE, $\sigma_n = 0.01$

(c) Scene Radiance MSE, $\sigma_n = 0.05$

(d) Atmospheric Light MSE, $\sigma_n = 0.05$

Figure 4.12: The iterative procedure was performed on the image from Fig. **4.5**a. Although the MSE of the local atmospheric light gets worse with each iteration, when using a sufficiently small smoothing parameter the MSE of the scene radiance is improved with iterations until a minimum is reached. Images are shown in Fig. **4.13**

The results in Fig. **4.12** through Fig. **4.15** show the effect of the above procedure using a constant smoothing parameter (h) throughout the image. In the following section, to make use of our knowledge that the noise variance in the scene radi-



(a) Noisy Scene Radiance Estimate

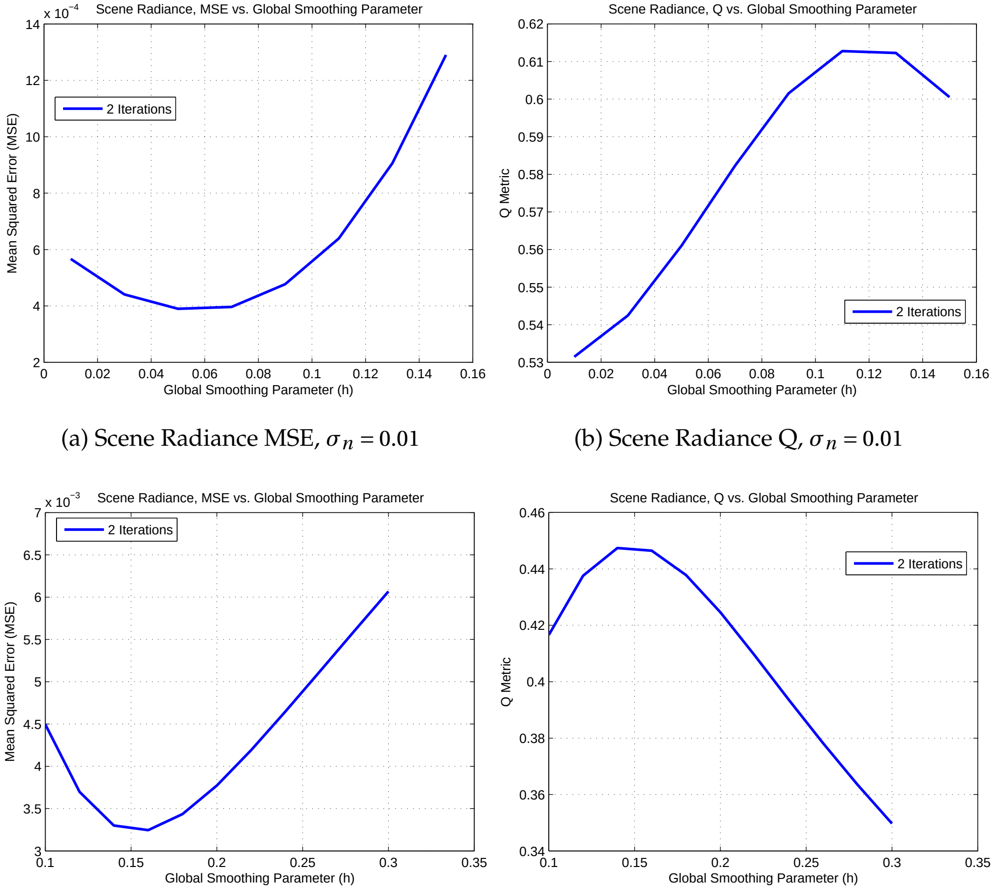
(b) 1st Iteration

(c) 2th Iteration

(d) 5th Iteration

Figure 4.13: Scene Radiance Estimation for Different Iterations for Fig. **4.5**a with $\sigma_n = 0.01$. For this example the global smoothing parameter, $h = 0.05$.

ance is dependent on the transmission, an adaptive smoothing parameter is proposed. Fig. **4.12** and Fig. **4.13** show the effect of iterations in recovering the scene radiance. Depending on the smoothing parameter, iterations may improve or worsen the MSE of the final result. When the image is undersmoothed in the first iteration, further iterations will improve the mean squared error of the estimate until a minimum is reached. In contrast, if the image is oversmoothed in the first iteration, further iterations will serve only to worsen the estimate. While it is possible to optimize the smoothing parameter for a single iteration, by introducing a further iterations we improve the probability



(a) Scene Radiance MSE, $\sigma_n = 0.01$

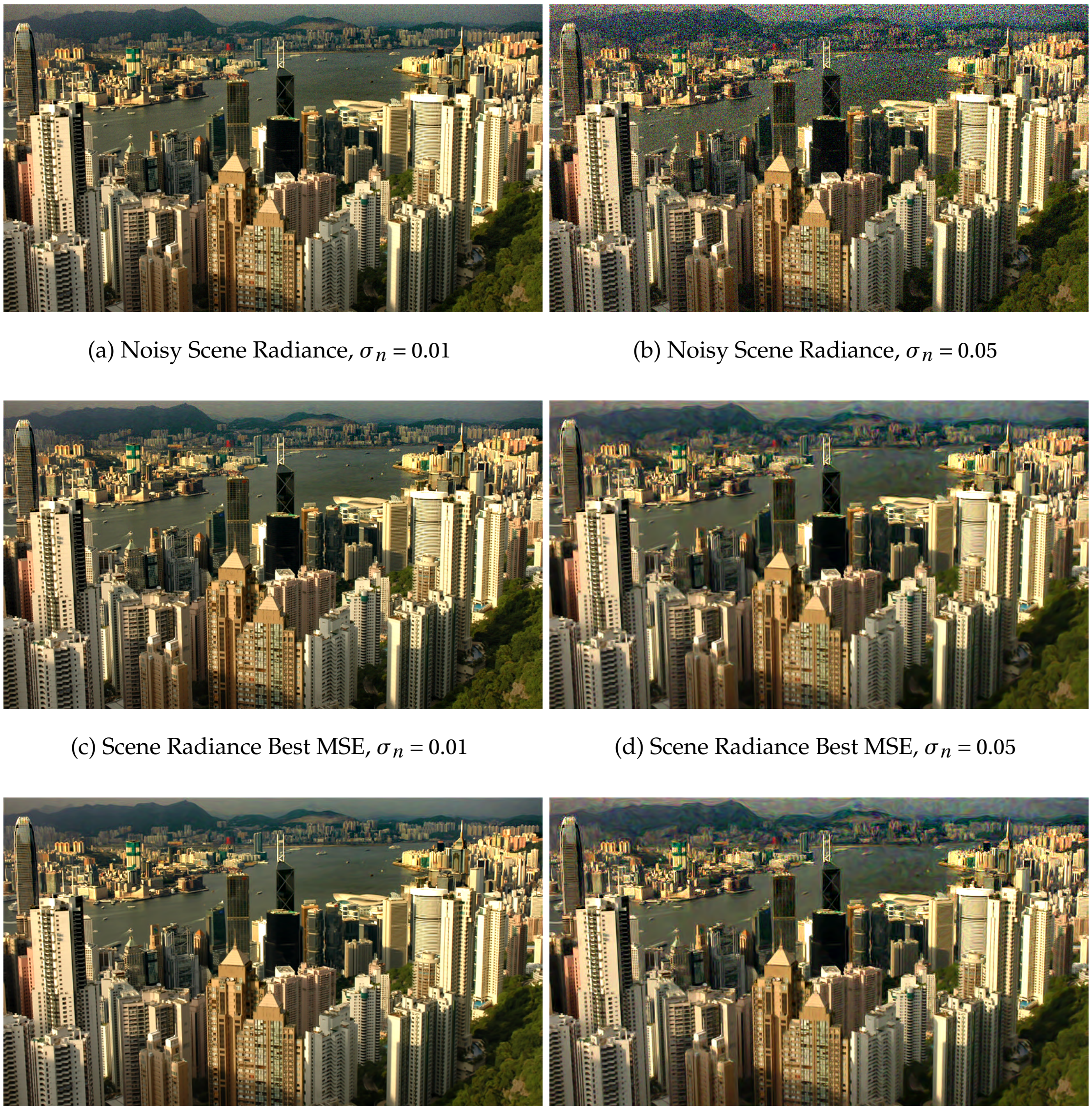
(b) Scene Radiance Q, $\sigma_n = 0.01$

(c) Scene Radiance MSE, $\sigma_n = 0.05$

(d) Scene Radiance Q, $\sigma_n = 0.05$

Figure 4.14: For this image(Fig. **4.5**a), with a low level of noise ($\sigma_n = 0.01$), Q metric slightly oversmooths but is visually comparable to the minimum MSE result. When the noise level is increased ($\sigma_n = 0.05$), the Q metric agrees well with the minimum MSE.

that the smoothing parameter we choose will yield a good estimate. For a reasonably sized smoothing parameter, a good estimate is typically found within 2 or 3 iterations, and so the procedure used in this thesis is to optimize the smoothing parameter for 2 iterations. Since the atmospheric light MSE is only worsened by iterations, limiting



(a) Noisy Scene Radiance, $\sigma_n = 0.01$

(b) Noisy Scene Radiance, $\sigma_n = 0.05$

(c) Scene Radiance Best MSE, $\sigma_n = 0.01$

(d) Scene Radiance Best MSE, $\sigma_n = 0.05$

(e) Scene Radiance Best Q, $\sigma_n = 0.01$

(f) Scene Radiance Best Q, $\sigma_n = 0.05$

Figure 4.15: Results from using Q as a stopping parameter are visually similar to the lowest MSE results

the number to 2 should also mitigate any visual artifacts that may be introduced by too many iterations. Furthermore, we have found that minimizing the MSE over many possible combinations of iterations and smoothing parameters does not result in significant visual improvement. Fig. **4.14** and Fig. **4.15** show the results of this 2-iteration optimization strategy. Using the Q metric as a stopping criterion generally results in a good approximation of the best h. Although in the low noise example this results in a slightly oversmoothed image, visually the result is similar to the image with the lowest MSE.

4.3.3 Adaptive Smoothing Parameter

By keeping the smoothing parameter, h, constant the kernel does not take into account the spatially varying noise variance in the scene radiance estimation. As such, we propose using a spatially adaptive h to take into account the noise variance amplification in hazy regions when recovering the scene radiance. In order to properly adapt h, we must first find the expression for h that minimizes the mean squared error of the kernel regression estimate. In the case of standard 0th order kernel regression, using LARK, an approximate expression for the optimal smoothing parameter is:

$$h_{opt} \approx \left(\frac{\sigma^2 (\det(\mathbf{C}))^{5/2}}{2\pi N(\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2} C_{22} - 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2} C_{12} + \frac{\partial^2 z(\mathbf{x})}{\partial x_2^2} C_{11})^2} \right)^{1/6} \tag{4.37}$$

where N is the number of elements in the local neighborhood, C_{ij} indicates the i,jth element in $\mathbf{C}$ and $z(\mathbf{x})$ is the underlying regression function (i.e. the noise free image). For a derivation, see Appendix A. From this expression, we can construct a spatially adaptive smoothing parameter. Note first that Eq. (4.37) relies on the unknown second derivatives of the noise free image. Although we can estimate these[3], these estimates may be prone to errors. Furthermore, some image patches may not even have second

[3] The approach taken in this thesis is to estimate these from the scene radiance recovered directly from the BM3D denoised image using second order kernel regression with a simple Gaussian kernel.

derivatives (e.g. a flat image patch), making an estimate for them nonsense. Thus in practice, we add an additional global smoothing parameter to suppress these errors. Next we recall from Eq. (4.3) that the noise variance, σ^2, is inversely proportional to the square of the transmission, t^2. We could replace σ^2 in Eq. (4.37) directly with $\frac{\sigma^2}{t^2(\mathbf{x})}$; however, since σ^2 is generally unknown, it can be absorbed into the global smoothing parameter. Furthermore since N is a constant, it can also be absorbed into the global smoothing parameter. And so we arrive finally at an expression for a spatially adaptive smoothing parameter:

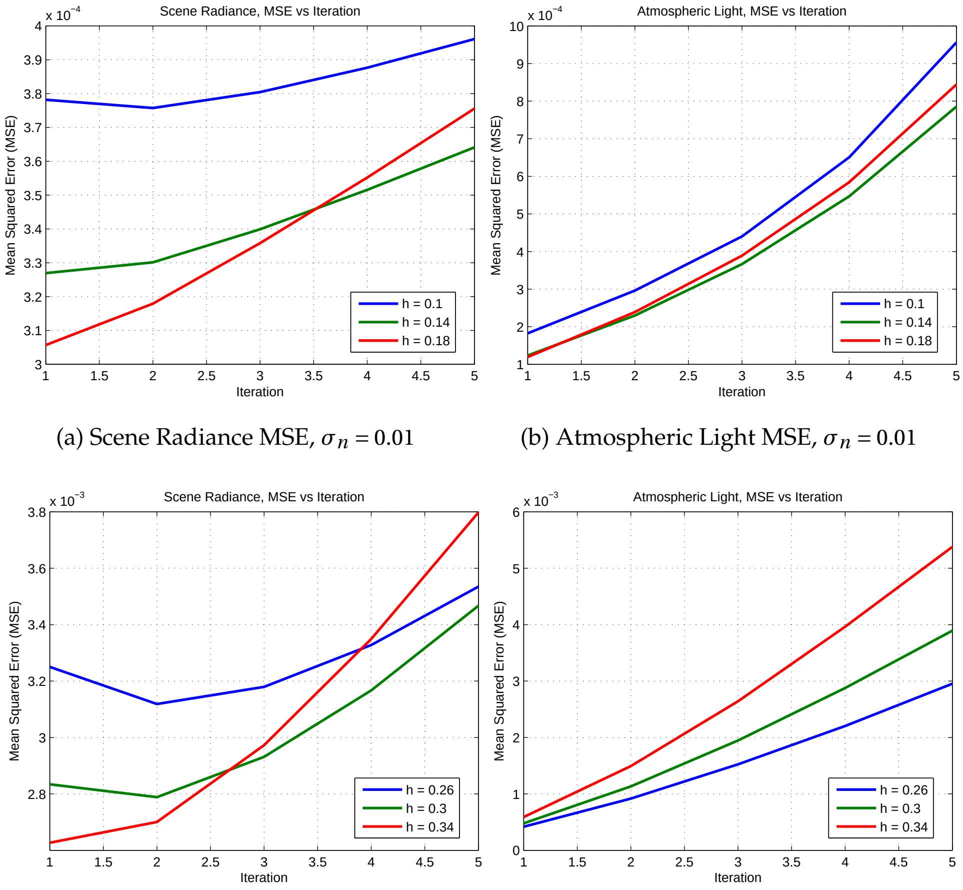
$$h_{adap}(\mathbf{x}) = h_{global}\left(\frac{(\det(\mathbf{C}))^{5/2}}{2\pi t(\mathbf{x})^2(\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2}C_{22} - 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2}C_{12} + \frac{\partial^2 z(\mathbf{x})}{\partial x_2^2}C_{11})^2}\right)^{1/6} \tag{4.38}$$

where h_{global} is the global smoothing parameter. If the 2nd derivatives cannot be reliably estimated, then Eq. (4.38) can be simplified to the following:

$$h_{adap}(\mathbf{x}) = h_{global}\left(\frac{1}{2\pi t(\mathbf{x})^2}\right)^{1/6} \tag{4.39}$$

Note that the adaptive smoothing parameter is necessary only in the scene radiance estimation (i.e. Eq. (4.35)). The smoothing parameter for the local atmospheric light estimation is fixed to h_{global}.

Figures Fig. **4.16** to Fig. **4.19** show the results of using the adaptive smoothing parameter in equation Eq. (4.38). Again for the image from Fig. **4.5**a, performing the kernel regression iterations results in similar behavior to that of using the constant smoothing parameter (Fig. **4.16** and Fig. **4.17**). Although not necessarily optimal in the minimum MSE sense, typically two iterations provides a good estimate of the scene radiance. So as in the constant smoothing parameter case, we optimize for the best h_{global} after two iterations. As before, using the Q metric as a stopping criterion generally results in a good approximation of the best h_{global} (Fig. **4.18** and Fig. **4.19**).



(a) Scene Radiance MSE, $\sigma_n = 0.01$

(b) Atmospheric Light MSE, $\sigma_n = 0.01$

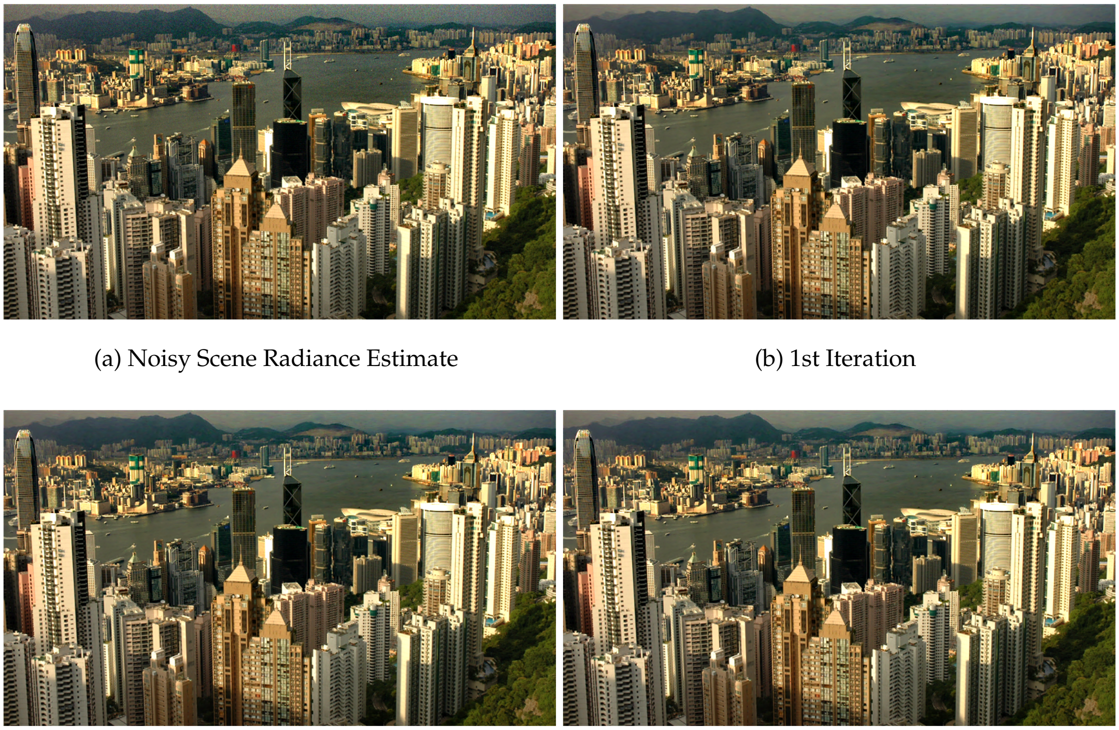
(c) Scene Radiance MSE, $\sigma_n = 0.05$

(d) Atmospheric Light MSE, $\sigma_n = 0.05$

Figure 4.16: The qualitative behavior over iterations of the MSE for the scene radiance and local atmospheric light with an adaptive smoothing parameter (Eq. (4.38)) is similar to that of using a constant smoothing parameter (Fig. **4.14**).

4.4 Experimental Results

In the following experiments we compare results of denoising followed by dehazing to the iterative method using Q as the only control parameter. Gaussian noise was added to each image, and the MSE of the results was computed as compared to



(a) Noisy Scene Radiance Estimate

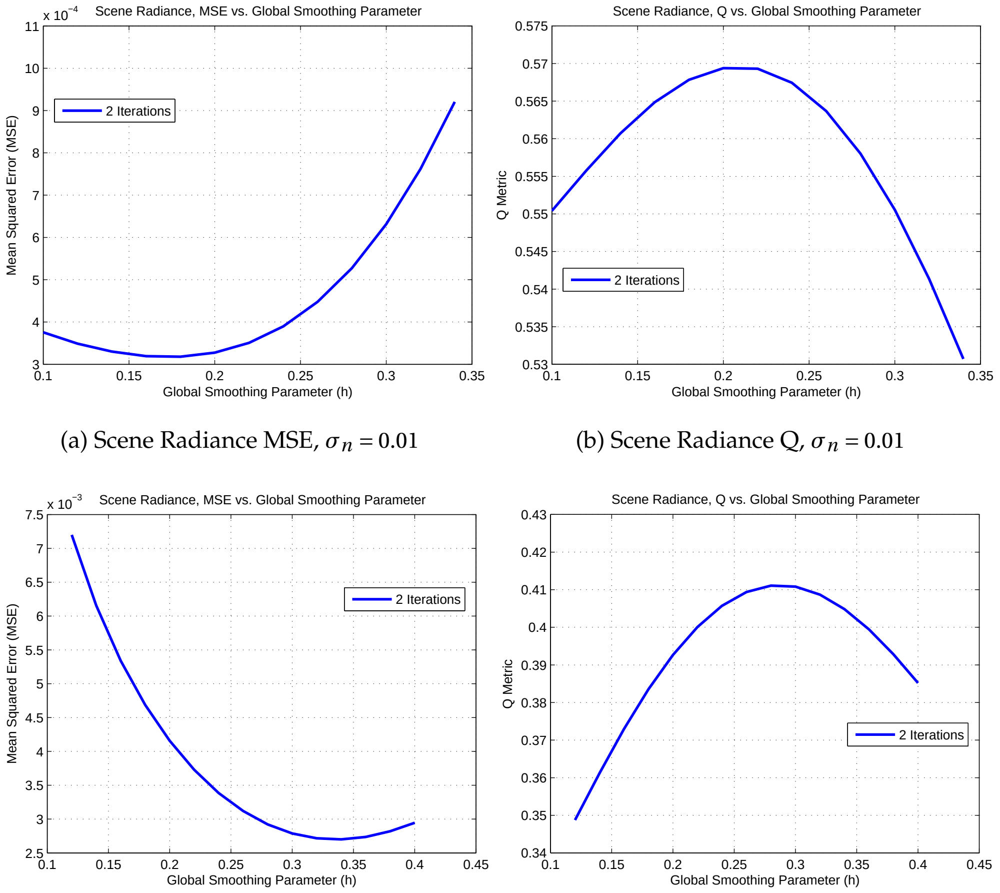
(b) 1st Iteration

(c) 2nd Iteration

(d) 5th Iteration

Figure 4.17: Scene Radiance Estimation for Different Iterations with $\sigma_n = 0.01$. This example uses the adaptive smoothing parameter from Eq. (4.38) with $h_{global} = 0.18$.

dehazing results on the noise free image. After noise is added, BM3D is used to estimate the hazy image using the Q metric to tune the denoising parameter. From this denoised image, the atmospheric light and transmission map are estimated. These estimates are then used to recover the scene radiance for both methods. For the iterative method, all three smoothing parameter strategies were tried for the scene radiance estimation. As shorthand for the following results, the process of denoising followed be dehazing is simply referred to as DD. The iterative method is referred to by the type of smoothing parameter used, with a constant smoothing parameter referred to as h_{const},



(a) Scene Radiance MSE, $\sigma_n = 0.01$

(b) Scene Radiance Q, $\sigma_n = 0.01$

(c) Scene Radiance MSE, $\sigma_n = 0.05$

(d) Scene Radiance Q, $\sigma_n = 0.05$

Figure 4.18: For this image(Fig. **4.5**a), using the adaptive smoothing parameter from Eq. (4.38), maximizing the Q metric tends to oversmooth for a low level of noise. When the noise level is increased, the Q metric agrees well with the minimum MSE.

Eq. (4.38) is referred to as h_{adap1}, and Eq. (4.39) is referred to as h_{adap2}. Reported MSE results are averaged over 5 trials.

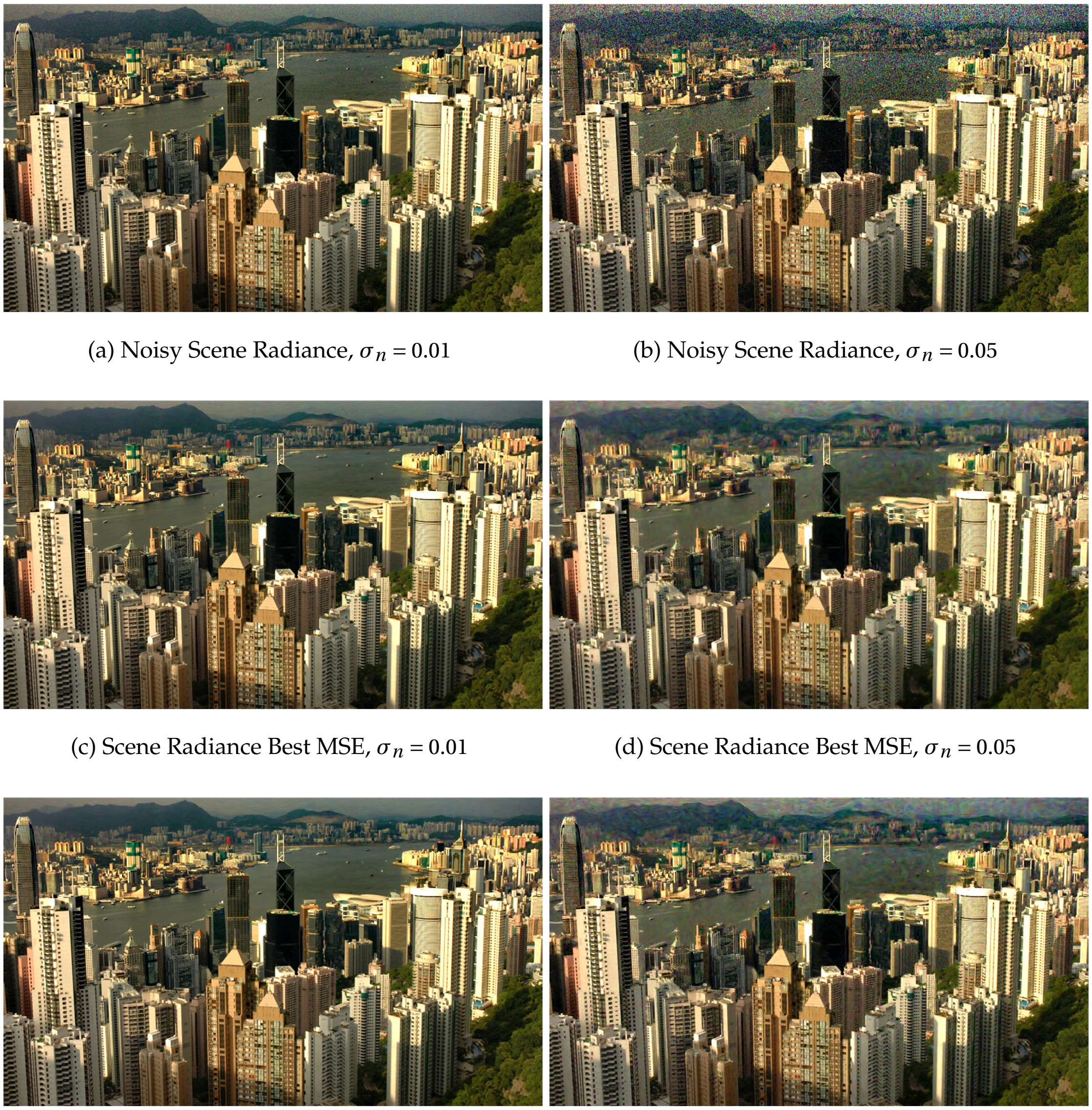
4.5 Summary

In this chapter we have discussed estimation of the underlying scene radiance from a noisy hazy image. Given estimates for the atmospheric light and transmission map, naively dehazing the image without regard to noise results in significant noise amplification in regions containing dense haze. Two methods were presented for addressing noise in this situation. The first of which was the use of a state-of-the-art denoising algorithm, BM3D, to pre-process the image prior to performing any dehazing operations. Since the strength of the underlying noise is not usually known, the Q Metric [21] can be used as a stopping criterion for tuning the algorithm. Although this method presented convincing results, if the denoising algorithm is not tuned precisely, the quality of the resulting scene radiance can suffer, sometimes significantly.

An alternative method for scene radiance estimation was based on non-parametric kernel regression. In this scheme, the scene radiance and local atmospheric light are alternately estimated over several iterations using a locally adaptive filtering operation. Furthermore, an adaptive smoothing parameter is proposed to compensate for the dependence of the noise level on the transmission. To initialize the iterations, pilot estimates for the atmospheric light and transmission map are required. These are estimated from the BM3D denoised image. Since the atmospheric light and transmission map are not sensitive to oversmoothing, this scheme serves well even when BM3D must be tuned using the Q Metric.

Comparing the two methods, the process of denoising followed by dehazing gives the best results when the exact amount of noise is known. However, when the noise must be estimated, the iterative procedure outperforms the first method when the amount of noise is low, and is competitive when the amount of noise is relatively high. When the results of processing on a real noisy hazy image are compared, the iterative method appears more visually appealing than denoising followed by dehazing.

In the following chapter, we conclude the thesis and present possibilities for future work.



(a) Noisy Scene Radiance, $\sigma_n = 0.01$

(b) Noisy Scene Radiance, $\sigma_n = 0.05$

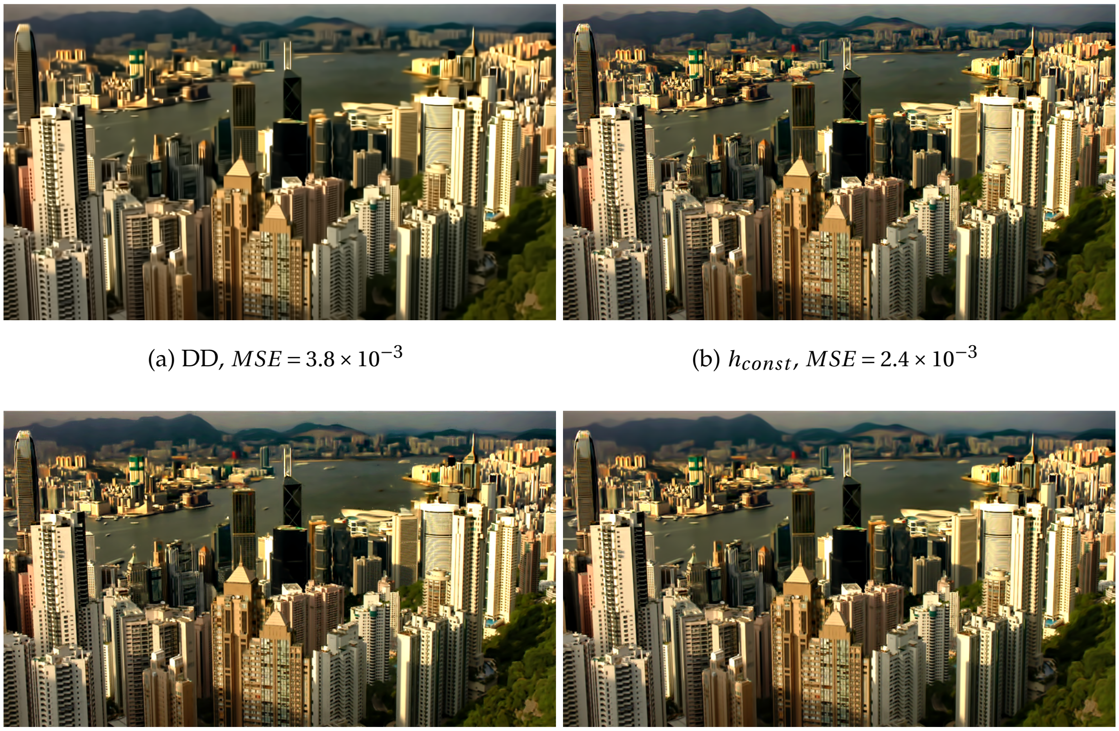
(c) Scene Radiance Best MSE, $\sigma_n = 0.01$

(d) Scene Radiance Best MSE, $\sigma_n = 0.05$

(e) Scene Radiance Best Q, $\sigma_n = 0.01$

(f) Scene Radiance Best Q, $\sigma_n = 0.05$

Figure 4.19: With the adaptive smoothing parameter (Eq. (4.38)) results from using the best Q are visually comparable to results from minimizing the MSE, with the low noise case ($\sigma_n = 0.01$) being slightly oversmoothed, and the higher noise case (σ_n) being slightly undersmoothed. In all cases the adaptive smoothing parameter improves the scene radiance versus the noisy case.



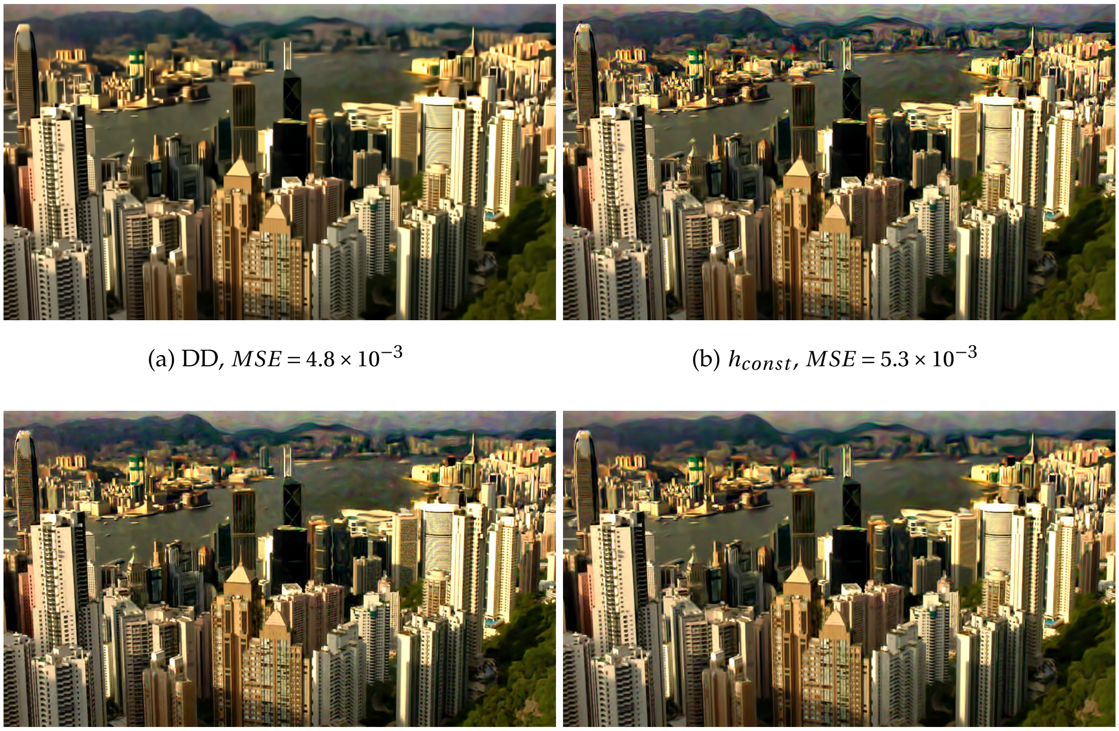
(a) DD, $MSE = 3.8 \times 10^{-3}$

(b) h_{const}, $MSE = 2.4 \times 10^{-3}$

(c) h_{adap1}, $MSE = 2.2 \times 10^{-3}$

(d) h_{adap2}, $MSE = 2.5 \times 10^{-3}$

Figure 4.20: Results for image in Fig. **4.5**a with $\sigma_n = 0.01$.



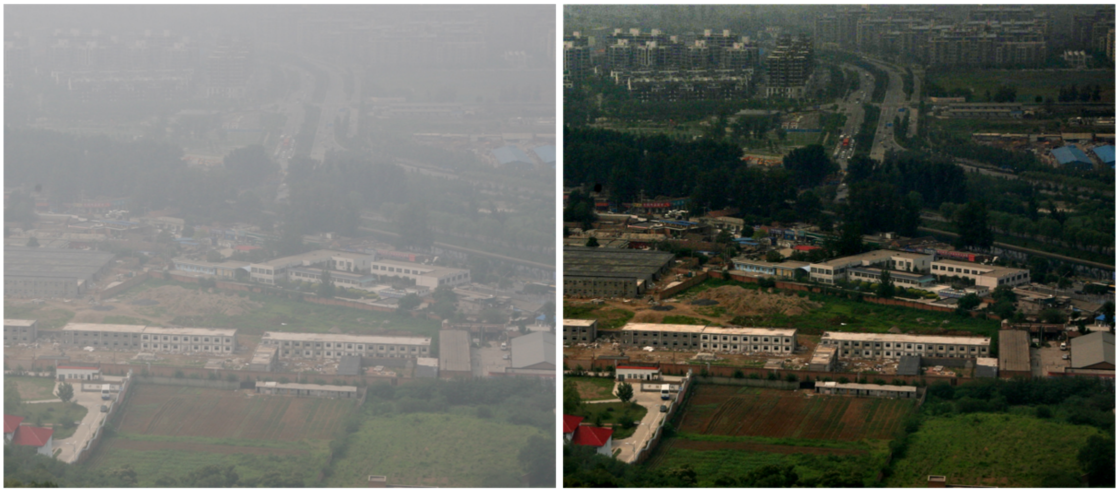
(a) DD, $MSE = 4.8 \times 10^{-3}$

(b) h_{const}, $MSE = 5.3 \times 10^{-3}$

(c) h_{adap1}, $MSE = 4.7 \times 10^{-3}$

(d) h_{adap2}, $MSE = 4.9 \times 10^{-3}$

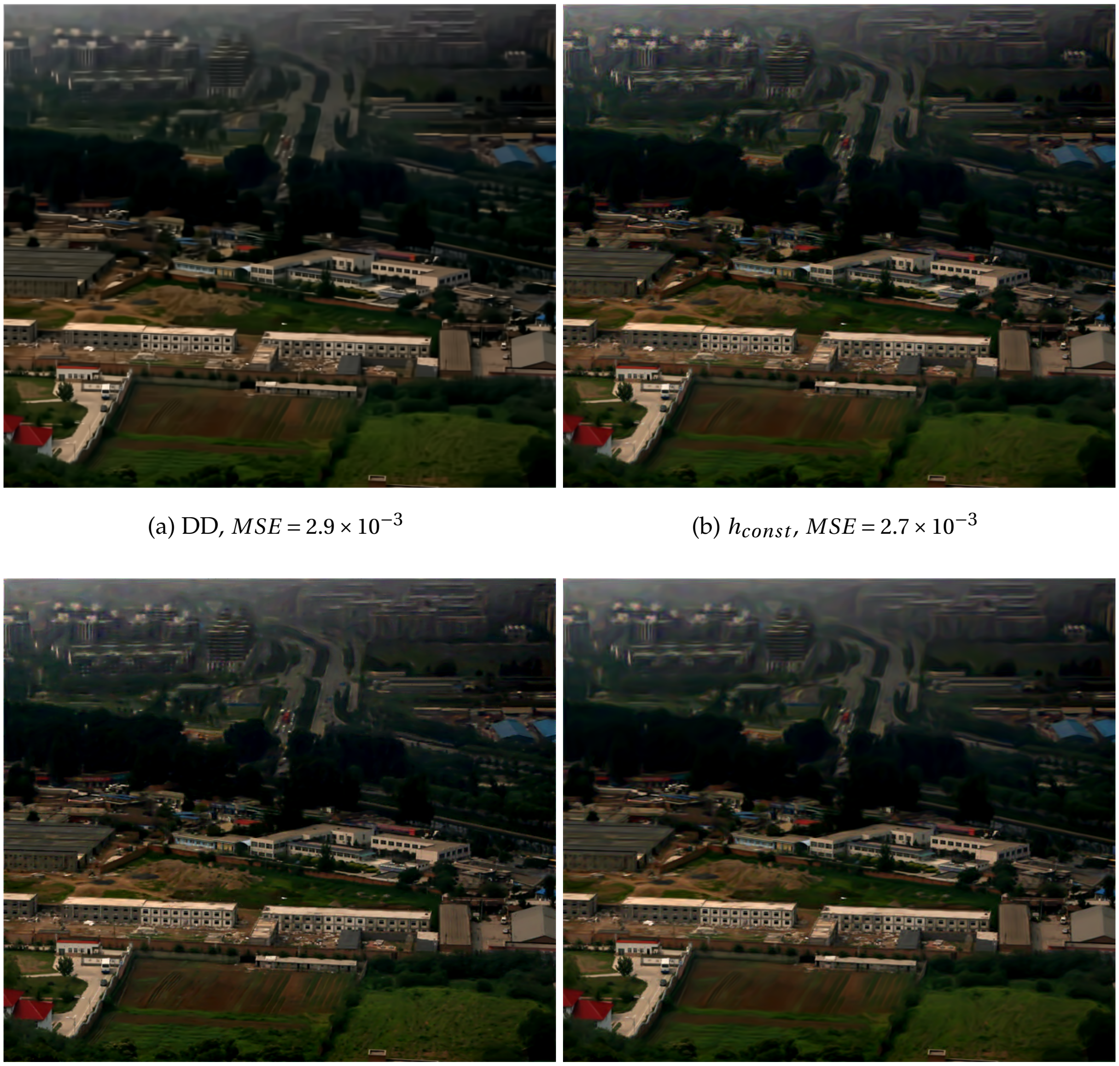
Figure 4.21: Results for image in Fig. **4.5**a with $\sigma_n = 0.05$.



(a) Hazy Image (Noise Free)

(b) Recovered Scene Radiance (Noise Free)

Figure 4.22: Source Image for examples in Fig. **4.23** and Fig. **4.24**.



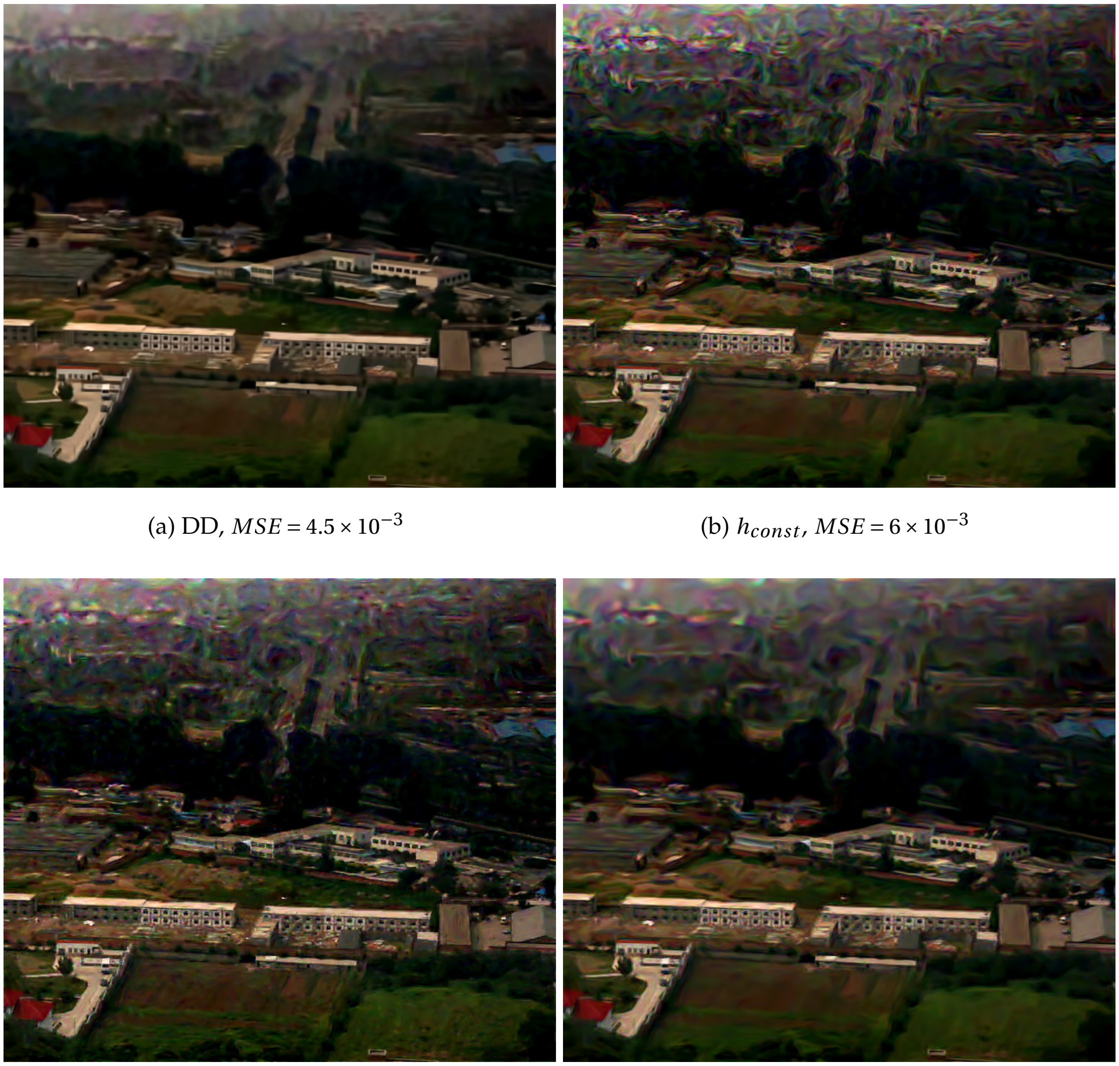
(a) DD, $MSE = 2.9 \times 10^{-3}$

(b) h_{const}, $MSE = 2.7 \times 10^{-3}$

(c) h_{adap1}, $MSE = 2.6 \times 10^{-3}$

(d) h_{adap2}, $MSE = 2.7 \times 10^{-3}$

Figure 4.23: Results for Fig. **4.22**a with $\sigma_n = 0.01$.



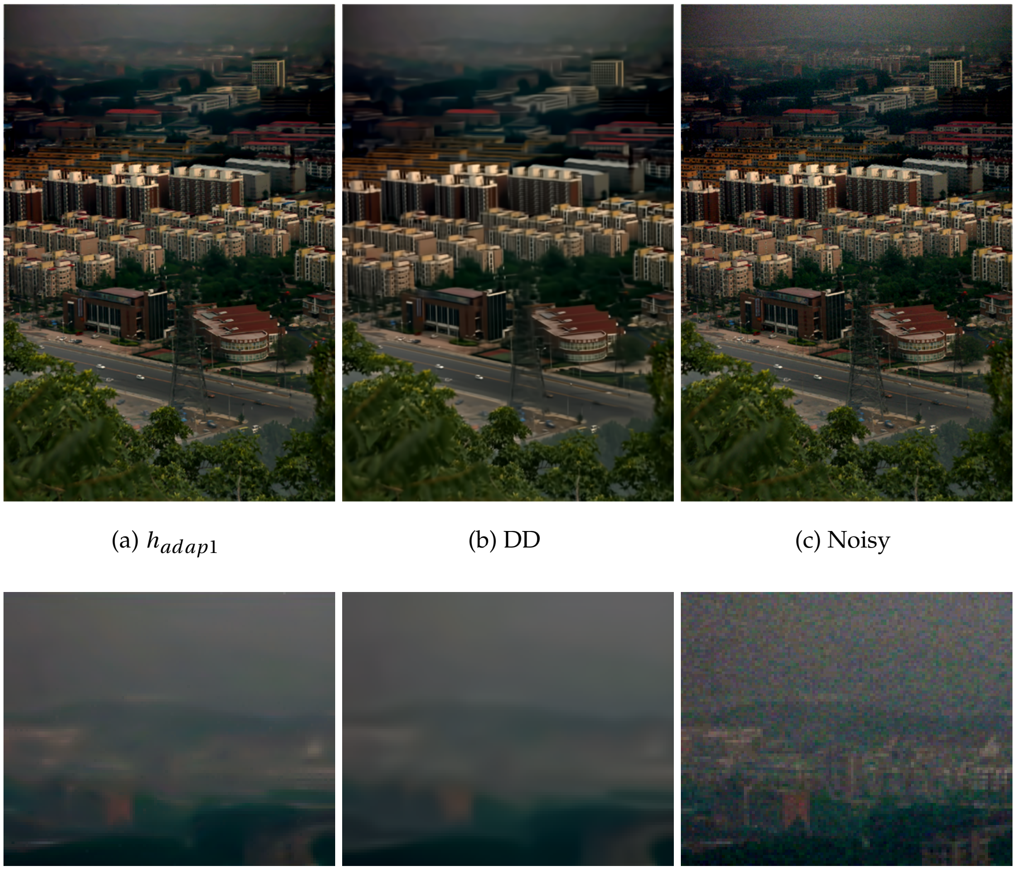
(a) DD, $MSE = 4.5 \times 10^{-3}$

(b) h_{const}, $MSE = 6 \times 10^{-3}$

(c) h_{adap1}, $MSE = 5.9 \times 10^{-3}$

(d) h_{adap2}, $MSE = 4.9 \times 10^{-3}$

Figure 4.24: Results for Fig. **4.22**a with $\sigma_n = 0.05$.



(a) h_{adap1}

(b) DD

(c) Noisy

(d) Close up, h_{adap1}

(e) Close up, DD

(f) Close up, Noisy

Figure 4.25: Scene radiance results from the real noisy example from figure Fig. **4.2**. Using the iterative method with h_{adap1} results in a slightly more detailed image.

Chapter 5

Conclusion

In this thesis we have addressed the problem of simultaneously removing haze and noise from a single image, and have made several novel contributions. The first is the adaptation of an existing technique, the dark channel prior [9], for estimating haze from a single **clean** hazy image to the case of a single **noisy** hazy image. The dark channel prior proved to be sensitive to noise, and so denoising the input image as a pre-processing step was proposed. Furthermore, we have directly compared the results of two existing transmission map refinement methods and considered their applications in the presence of noise. Both methods were found to be robust to low levels of noise; however, artifacts became apparent when the noise level was increased. Denoising the guide image prior to refinement is sufficient to suppress these artifacts, while maintaining a visually satisfying refined transmission map. Finally, we have presented two effective methods for final scene radiance recovery. The first method treats haze and noise separately by denoising the image as a pre-processing step to dehazing. The second method considers removing haze and noise simultaneously through an iterative kernel regression based approach using an adaptive kernel. In comparing these two methods, we found that when the noise level is known or can be precisely estimated, simply denoising prior to dehazing provides the best results when paired with a state-of-the-art denoising algorithm. However, this approach is sensitive to both under- and over-smoothing in the denoising step. This is especially true in low noise cases, when a small quantitative error in noise estimation can mean a large relative error, with any degradation remaining after denoising being further amplified in the dehazing process. The iterative approach proves to be more robust, offering visually comparable results to the first method when the noise level is known, and matching or exceeding performance of the first method both in terms of MSE and subjective visual quality when the noise level must be estimated.

Chapter 2 addressed the problem of haze estimation using the “dark channel prior," proposed in [9]. We found that although effective for clean images, the accuracy of the method is extremely sensitive to noise. In order to improve estimation of the dark channel, the input image must be denoised first. Furthermore, in the case that the exact level of noise is unknown, we found that it is far better to “over-"denoise than it is to “under-"denoise. From this denoised image, both the atmospheric light and a coarse transmission map can be accurately estimated.

The transmission map estimated directly using the dark channel prior suffers from block artifacts, and so it is necessary to further refine it before recovering the scene radiance. Accordingly, Chapter 3 reviewed two related methods for performing this refinement that use the hazy image as a guide for filtering the transmission map. The first method, called the “Matting Laplacian," in [9] was originally proposed for image matting in [10]. Although this provides visually satisfying results, it involves the inversion of large matrix, and so can be computationally expensive. An alternative refinement method, called the “Guided Filter," was proposed in [14] and is equivalent to one Jacobi iteration in solving the cost function for the Matting Laplacian. Computation of the Guided Filter is independent of filter size, and so can quickly achieve visually satisfying results by using a large filter radius. Both methods are robust to low to moderate amounts of noise, and in the case that the noise level is high, the guide image can be pre-processed with a denoising algorithm to suppress artifacts.

Chapter 4 discussed the problem of scene radiance recovery once the atmospheric light and transmission map have been estimated. First we considered the case of directly dehazing the image, disregarding the noise. We showed that any noise present in the hazy image is amplified in proportion to the haze content, and so even when the amount of noise appears negligible in the input image, it can become a significant factor in the recovered scene radiance. Two methods were proposed to remedy this problem. The first method treated haze and noise separately by denoising the image prior to dehazing. In contrast, the second method performed dehazing and denoising simultaneously through an iterative, kernel regression based approach. Furthermore, for this method, an approximately optimal adaptive smoothing parameter was proposed. When paired with a state-of-the-art denoising algorithm, the first method performed well, providing visually satisfying results. In the case that the noise level was known a priori, denoising prior to dehazing provided the best results in terms of MSE. Although worse in terms of MSE, the iterative approach provided results visually comparable to the first approach when the noise was known. When the noise level had to be estimated, the first method proved to be sensitive to errors in choosing the amount of denoising, due to amplification of errors in the dehazing process. In the case of “under"-denoising, even small amounts of noise left in the input image can become apparent in the dehazed result. In the case of “over"-denoising, significant details are lost in hazy regions. In this situation, the iterative approach achieved superior results both in terms of MSE and subjective visual quality in low noise cases, with comparable results in the high noise cases.

5.1 Directions for Future Work

Beyond the work presented in this thesis, there are several areas that deserve further research.

5.1.1 Parameter Estimation

There are a number of parameters that must be chosen throughout the dehazing process. For haze estimation, this includes dark channel patch size and refinement parameters. Although some “good" parameters have been found empirically, these may only work well for some images. It is desirable to automatically choose these parameters based on some measured features from the input image. This may be possible using the Q metric [21] that was used for tuning the denoising algorithms in this thesis.

5.1.2 Improved Dehazing Algorithm

For this thesis, we used the dark channel prior for haze estimation, and although this method provides good results for colorful images, it tends to over-estimate the haze content in images containing many gray or white objects. We also consider only a simple pixel-wise degradation model, which may not be true in cases where the haze is dense and light from the scene radiance may suffer from blurring effects due to heavy scatter.

5.1.3 Video

Another potential topic of research is expanding the presented haze removal and denoising process to video. He et al. have supplied video results in the supplementary material for [9]. The restored videos are compelling; however, it appears that these sequences are simply an application of the algorithm independently on individual video frames. More recently, Zhang et al. [37] found additional performance gains by considering the video frames together, using the dark channel prior and Guided Filter for haze estimation and refinement, and Markov random fields to increase spatial and temporal coherence. Noise, however, was not addressed in their haze removal process, and so is an open problem.

Appendix A

Optimal Smoothing Parameter Derivation

In this section we present a derivation of an approximate expression for the local mean squared error of a zeroth order kernel regression estimate using the locally adaptive kernel proposed in [30]. From this expression, we further derive a locally optimal smoothing parameter that approximately minimizes the mean squared error for each window where the assumptions hold true.

A.1 Zeroth Order Kernel Regression Estimate

The general data model for kernel regression in 2-D is given by:

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad \forall\, \mathbf{x}_i \in w, \quad i = 1, \ldots, N \tag{A.1}$$

where y_i is a noisy measurement at spatial coordinate $\mathbf{x}_i = [x_{1i}, x_{2i}]^T$, $z(\,\cdot\,)$ is the unspecified regression function, ε_i is zero mean I.I.D. measurement noise with variance σ^2, and w is a window around the point of interest at coordinate $\mathbf{x}$, containing N samples. Here we assume that the sample positions are non-random and equally spaced.

A local Taylor expansion of y_i is given by:

$$z(\mathbf{x}_i) = z(\mathbf{x}) + \{\nabla z(\mathbf{x})\}^T (\mathbf{x}_i - \mathbf{x}) + \frac{1}{2}(\mathbf{x}_i - \mathbf{x})^T \{\mathscr{H} z(\mathbf{x})\}^T (\mathbf{x}_i - \mathbf{x}) + \cdots \tag{A.2}$$

where $\mathbf{x}$ is a spatial coordinate, ∇ is the gradient (2×1) operator, and $\mathscr{H}$ is the Hessian (2×2) operator, i.e.:

$$\nabla z(\mathbf{x}) = \left[\frac{\partial z(\mathbf{x})}{\partial x_1}, \quad \frac{\partial z(\mathbf{x})}{\partial x_2}\right]^T, \quad \mathscr{H} z(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 z(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2} \\ \frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 z(\mathbf{x})}{\partial x_2^2} \end{bmatrix} \tag{A.3}$$

Eq. (A.2) is true in the limit as the number of terms goes to infinity. For zeroth order kernel regression, we fit a zeroth order model to the data, i.e.:

$$z(\mathbf{x}_i) \approx z(\mathbf{x}) \quad \forall\, \mathbf{x}_i \in w, \quad i = 1, \ldots, N \tag{A.4}$$

where $\mathbf{x}$ is the coordinate of interest. Thus our kernel regression estimate is found from the following optimization problem:

$$\min_{\widehat{z}(\mathbf{x})} \sum_{i=1}^{N} \left[y_i - \widehat{z}(\mathbf{x})\right]^2 K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x}) \tag{A.5}$$

where $K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x})$ is the adaptive kernel proposed in [30] and reviewed in Chapter 4:

$$K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x}) = \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2}\right) \tag{A.6}$$

where h is a global smoothing parameter and $\mathbf{C}_i$ is the regularized gradient covariance matrix explained in Chapter 4. This is a simple weighted least-squares problem, with the caveat that the weights are a function of the given data. The solution to this can be written as:

$$\widehat{z}(\mathbf{x}) = \frac{\sum_{i=1}^{N} K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x}) y_i}{\sum_{i=1}^{N} K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x})} \tag{A.7}$$

A.2 Mean Squared Error of Estimate

A typical measure for the performance of an estimator is the mean squared error between the estimate and the true signal. Since the true signal is not restricted to any specific expression, we compare our estimate to the local Taylor expansion of the signal (Eq. (A.2)). Furthermore, since our regression kernel is non-linear, because it depends on the measured data, finding an exact expression for the mean squared error is quite difficult. Therefore we make some simplifying assumptions, making the derived mean squared error only approximate, but also giving us a more intuitive expression.

The first assumption is that $\mathbf{C}_i$ does not vary significantly in a local window. This should hold true if the window size is sufficiently small, and the gradients change in a smooth manner. From this we have:

$$\mathbf{C}_i \approx \mathbf{C} \quad \forall\, i \in w, \quad i = 1, \ldots, N \tag{A.8}$$

where $\mathbf{C}$ is the gradient covariance matrix at the coordinate of interest:

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} \\ C_{12} & C_{22} \end{bmatrix}$$

Therefore:

$$\begin{aligned} K_{\mathbf{C}_i}(\mathbf{x}_i - \mathbf{x}) &= \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2}\right) \\ &\approx \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C} (\mathbf{x}_i - \mathbf{x})}{2h^2}\right) \end{aligned} \tag{A.9}$$

So we can now refer to $K_{\mathbf{C}_i}$ as $K_{\mathbf{C}}$ to reflect that its form is approximately constant locally. We notice now that this simplified kernel has the same form as a bivariate Gaussian distribution, which for a random variable $\mathbf{y} = [y_1, y_2]^T$ with mean $\boldsymbol{\mu}_{\boldsymbol{y}} = E(\mathbf{y}) =$

$[\mu_1, \mu_2]^T$ and covariance $\mathbf{\Sigma} = E((\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{y}})(\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{y}})^T)$ is described by the probability density function[38]:

$$p(\mathbf{y}) = \frac{1}{2\pi\sqrt{\det(\mathbf{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{y}})^T \mathbf{\Sigma^{-1}} (\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{y}})\right) \tag{A.10}$$

This is exactly our kernel with $(\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{y}}) \to (\mathbf{x}_i - \mathbf{x})$ and $\mathbf{\Sigma} \to (\frac{1}{h^2}\mathbf{C})^{-1}$. Note that:

$$\left(\frac{1}{h^2}\mathbf{C}\right)^{-1} = \frac{h^2}{\det(\mathbf{C})} \begin{bmatrix} C_{22} & -C_{12} \\ -C_{12} & C_{11} \end{bmatrix} \tag{A.11}$$

Our second assumption is that we have a sufficiently large window with a sufficient number of samples, such that:

$$N^{-1} \sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{u}_i)\mathbf{u}_i \approx \int \mathbf{u} K_{\mathbf{C}}(\mathbf{u}) d\mathbf{u} \tag{A.12}$$

where we make the substitution $\mathbf{u}_i = \mathbf{x}_i - \mathbf{x}$, and let $\int$ be shorthand for $\int_{-\infty}^{\infty}$. Without loss of generality, we can let the position of interest at the center of the window be $\mathbf{x} = [0, 0]^T$. Now recalling that the definition for expected value is $E(\mathbf{u}) = \int \mathbf{y}\, p(\mathbf{y})\, d\mathbf{y}$, and relating our kernel function to the probability density function in Eq. (A.10), we can write the following [38]:

$$\int \mathbf{u} K_{\mathbf{C}}(\mathbf{u}) d\mathbf{u} \approx 0 \tag{A.13}$$

and

$$\int u_1^2 K_{\mathbf{C}}(\mathbf{u}) d\mathbf{u} \approx \frac{h^2}{\det(\mathbf{C})} C_{22} \tag{A.14}$$

$$\int u_2^2 K_{\mathbf{C}}(\mathbf{u}) d\mathbf{u} \approx \frac{h^2}{\det(\mathbf{C})} C_{11} \tag{A.15}$$

$$\int u_{12}^2 K_{\mathbf{C}}(\mathbf{u}) d\mathbf{u} \approx -\frac{h^2}{\det(\mathbf{C})} C_{12} \tag{A.16}$$

With these assumptions, we can now proceed with finding the approximate mean squared error of our estimate. First recall that the mean squared error can be expressed in terms of variance and bias as the following [31]:

$$MSE(\widehat{z}(\mathbf{x})) = var(\widehat{z}(\mathbf{x})) + b^2(\widehat{z}(\mathbf{x})) \tag{A.17}$$

where

$$var(\widehat{z}(\mathbf{x})) = E(\widehat{z}^2(\mathbf{x})) - E^2(\widehat{z}(\mathbf{x})) \tag{A.18}$$

and

$$b^2(\widehat{z}(\mathbf{x})) = (E(\widehat{z}(\mathbf{x})) - z(\mathbf{x}))^2 \tag{A.19}$$

Recalling Eq. (A.7), the variance of our estimate is:

$$var(\widehat{z}(\mathbf{x})) = E\left(\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})y_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right)^2\right) - E^2\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})y_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) \tag{A.20}$$

$$= E\left(\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})(z(\mathbf{x}_i) + \varepsilon_i)}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right)^2\right) - E^2\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})(z(\mathbf{x}_i) + \varepsilon_i)}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) \tag{A.21}$$

$$\begin{aligned} = {} & E\left(\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})z(\mathbf{x})}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right)^2\right) + 2E\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})z(\mathbf{x_i})\varepsilon_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) + \\ & E\left(\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})\varepsilon_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right)^2\right) - E^2\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})z(\mathbf{x}_i)}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) - \\ & E^2\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})\varepsilon_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) \end{aligned} \tag{A.22}$$

$$= E^2\left(\frac{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})\varepsilon_i}{\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})}\right) \tag{A.23}$$

Since the ε_i's are uncorrelated with $E(\varepsilon) = 0$, we can eliminate cross terms and simplify this expression to:

$$var(\widehat{z}(\mathbf{x})) = \frac{\sigma^2}{N}\left(\frac{N^{-1}\sum_{i=1}^N K_{\mathbf{C}}^2(\mathbf{x}_i - \mathbf{x})}{\left(N^{-1}\sum_{i=1}^N K_{\mathbf{C}}(\mathbf{x}_i - \mathbf{x})\right)^2}\right) \tag{A.24}$$

$$\approx \frac{\sigma^2}{N}\frac{\int\left(\frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\right)^2 \exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{h^2}d\mathbf{u}\right)}{\int\frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}d\mathbf{u}\right)} \tag{A.25}$$

$$\approx \frac{\sigma^2}{N}\frac{\sqrt{\det(\mathbf{C})}}{4\pi h^2} \tag{A.26}$$

Recalling the Taylor expansion of our underlying signal (Eq. (A.2)), the bias of our estimate is:

$$
\begin{aligned}
b(\widehat{z}(\mathbf{x})) &= E(\widehat{z}(\mathbf{x})) - z(\mathbf{x}) && \text{(A.27)}\\
&= E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})z(\mathbf{x})}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) + E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})(\nabla z(\mathbf{x}))^T(\mathbf{x}_i-\mathbf{x})}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) + \\
&\quad E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})\frac{1}{2}(\mathbf{x}_i-\mathbf{x})^T(\mathscr{H}z(\mathbf{x}))^T(\mathbf{x}_i-\mathbf{x})}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) + \\
&\quad \cdots + E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})\varepsilon_i}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) - z(\mathbf{x}) && \text{(A.28)}\\
&= E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})(\nabla z(\mathbf{x}))^T(\mathbf{x}_i-\mathbf{x})}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) + \\
&\quad E\left(\frac{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})\frac{1}{2}(\mathbf{x}_i-\mathbf{x})^T(\mathscr{H}z(\mathbf{x}))^T(\mathbf{x}_i-\mathbf{x})}{\sum_{i=1}^{N} K_{\mathbf{C}}(\mathbf{x}_i-\mathbf{x})}\right) + \cdots && \text{(A.29)}
\end{aligned}
$$

We can neglect higher order terms, since they will have a small effect relative to the first and second order terms, and recalling Eq. (A.12) to Eq. (A.16) we have:

$$
\begin{aligned}
b(\widehat{z}(\mathbf{x})) &\approx \frac{\int \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}\right)\mathbf{u}^T\nabla z(\mathbf{x})d\mathbf{u}}{\int \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}\right)d\mathbf{u}} + \frac{\int \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}\right)\frac{1}{2}\mathbf{u}^T\mathscr{H}z(\mathbf{x})\mathbf{u}d\mathbf{u}}{\int \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}\right)d\mathbf{u}} && \text{(A.30)}\\
&\approx 0 + \int \frac{\sqrt{\det(\mathbf{C})}}{2\pi h^2}\exp\left(-\frac{\mathbf{u}^T\mathbf{C}\mathbf{u}}{2h^2}\right)\frac{1}{2}\left(\mathscr{H}_{11}u_1 + 2\mathscr{H}_{12}u_1u_2 + \mathscr{H}_{22}u_2^2\right)d\mathbf{u} && \text{(A.31)}\\
&\approx \frac{h^2}{2\det(\mathbf{C})}\left(\mathscr{H}_{11}C_{22} - 2\mathscr{H}_{12}C_{12} + \mathscr{H}_{22}C_{11}\right) && \text{(A.32)}\\
b^2(\widehat{z}(\mathbf{x})) &\approx \frac{h^4}{4(\det(\mathbf{C}))^2}\left(\mathscr{H}_{11}C_{22} - 2\mathscr{H}_{12}C_{12} + \mathscr{H}_{22}C_{11}\right)^2 && \text{(A.33)}
\end{aligned}
$$

Thus we have our final expression for the MSE within a window where **C** is assumed to be constant:

$$
MSE(\widehat{z}(\mathbf{x})) \approx \frac{\sigma^2}{N}\frac{\sqrt{\det(\mathbf{C})}}{4\pi h^2} + \frac{h^4}{4(\det(\mathbf{C}))^2}\left(\mathscr{H}_{11}C_{22} - 2\mathscr{H}_{12}C_{12} + \mathscr{H}_{22}C_{11}\right)^2 \qquad \text{(A.34)}
$$

A.3 Optimal Smoothing Parameter

To find the optimal smoothing parameter we minimize Eq. (A.34) with respect to h:

$$\frac{\partial}{\partial h}\left(\frac{\sigma^2}{N}\frac{\sqrt{\det(\mathbf{C})}}{4\pi h^2}+\frac{h^4}{4(\det(\mathbf{C}))^2}(\mathscr{H}_{11}C_{22}-2\mathscr{H}_{12}C_{12}+\mathscr{H}_{22}C_{11})^2\right)=0 \tag{A.35}$$

$$-\frac{\sigma^2}{N}\frac{\sqrt{\det(\mathbf{C})}}{2\pi h^3}+\frac{h^3}{(\det(\mathbf{C}))^2}(\mathscr{H}_{11}C_{22}-2\mathscr{H}_{12}C_{12}+\mathscr{H}_{22}C_{11})^2=0 \tag{A.36}$$

$$h=\left(\frac{\sigma^2(\det(\mathbf{C}))^{5/2}}{2\pi N(\mathscr{H}_{11}C_{22}-2\mathscr{H}_{12}C_{12}+\mathscr{H}_{22}C_{11})^2}\right)^{1/6} \tag{A.37}$$

Thus we arrive at an expression for our optimal smoothing parameter:

$$h_{opt}\approx\left(\frac{\sigma^2(\det(\mathbf{C}))^{5/2}}{2\pi N(\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2}C_{22}-2\frac{\partial^2 z(\mathbf{x})}{\partial x_1\partial x_2}C_{12}+\frac{\partial^2 z(\mathbf{x})}{\partial x_2^2}C_{11})^2}\right)^{1/6} \tag{A.38}$$

Bibliography

[1] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” in *ACM SIGGRAPH ASIA 2008 courses*, ser. SIGGRAPH Asia ’08. New York, NY, USA: ACM, 2008, pp. 69:1–69:22. [Online]. Available: http://doi.acm.org/10.1145/1508044.1508113

[2] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, “Instant dehazing of images using polarization,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2001, p. 325âĂŞ332.

[3] S. Shwartz, E. Namer, and Y. Y. Schechner, “Blind haze separation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,*, vol. 1, 2006, pp. 1984–1991.

[4] S. Nayar and S. Narasimhan, “Vision in bad weather,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 820 –827 vol.2.

[5] S. G. Narasimhan and S. K. Nayar, “Interactive deweathering of an image using physical models,” in *IEEE IEEE Workshop on Color and Photometric Methods in Computer Vision, In Conjunction with ICCV*, October 2003.

[6] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski, “Deep photo: Model-based photograph enhancement and viewing,” in *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, vol. 27, no. 5, 2008, pp. 116:1–116:10.

[7] R. Tan, “Visibility in bad weather from a single image,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, Jun. 2008, pp. 1 –8.

[8] R. Fattal, “Single image dehazing,” *ACM Transactions on Graphics*, vol. 27, no. 3, August 2008.

[9] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1956–1963, 2009.

[10] A. Levin, D. Lischinski, and Y. Weiss, “A closed-form solution to natural image matting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 228–242, 2008.

[11] N. Joshi and M. Cohen, “Seeing mt. rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal,” in *Computational Photography (ICCP), 2010 IEEE International Conference on*, Mar. 2010, pp. 1 –8.

[12] Y. Y. Schechner and Y. Averbuch, “Regularized image recovery in scattering media,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1655–1660, 2007.

[13] R. Kaftory, Y. Schechner, and Y. Zeevi, “Variational distance-dependent image restoration,” in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007, pp. 1 –8.

[14] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *Computer Vision âĂŞ ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6311, pp. 1–14.

[15] P. S. C. Jr., “An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data,” *Remote Sensing of Environment*, vol. 24, no. 3, pp. 459 – 479, 1988. [Online].

Available: http://www.sciencedirect.com/science/article/B6V6V-48BKDWT-H/2/b0ffa0d35a8c8c87d68cc7c08a3eb3d3

[16] A. J. Preetham, P. Shirley, and B. Smits, "A practical analytic model for daylight," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 91–100. [Online]. Available: http://dx.doi.org/10.1145/311535.311545

[17] A. Abebe, J. Daniels, J. McKean, and J. Kapenga, *Statistics and Data Analysis*. Michigan: Department of Statistics, Wester Michigan University, 2001.

[18] H. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*. New York: Springer-Verlag, 2003.

[19] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080 –2095, aug. 2007.

[20] ——, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, 16 2007-oct. 19 2007, pp. I –313 –I –316.

[21] X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3116 –3132, dec. 2010.

[22] I. Omer and M. Werman, "Color lines: Image specific color representation," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 946–953, 2004.

[23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, Jan. 1998, pp. 839 –846.

[24] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.*, vol. 23, pp. 664–672, August 2004. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015777

[25] F. C. Crow, "Summed-area tables for texture mapping," *SIGGRAPH Comput. Graph.*, vol. 18, pp. 207–212, January 1984. [Online]. Available: http://doi.acm.org/10.1145/964965.808600

[26] K. He, J. Sun, and X. Tang, "Fast matting using large kernel matting laplacian matrices," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, june 2010, pp. 2165 –2172.

[27] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation (SIAM)*, vol. 4, pp. 490–530, 2005.

[28] M. Wand and M.Jones, *Kernel Smoothing*. Chapman and Hall, 1995.

[29] W. Hardle, M.Muller, S. Sperlich, and A.Werwatz, *Nonparametric and Semiparametric Models*. Springer, 2004.

[30] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *Image Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 349 –366, feb. 2007.

[31] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[32] W. Hardle., *Applied Nonparametric Regression*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[33] E. A. Nadaraya, "On estimating regression," *Theory of Prob. and Appl.*, vol. 9, pp. 141–142, 1964.

[34] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, 2002, pp. 478–482.

[35] R. Kimmel, *Numerical Geometry of Images*. Springer, 2003.

[36] G. Peyré, "Geodesic methods for shape and surface processing," *In Advances in Computational Vision and Medical Image Processing: Methods and Applications (springer)*, vol. 13, pp. 29–56, 2008.

[37] J. Zhang, L. Li, Y. Zhang, G. Yang, X. Cao, and J. Sun, "Video dehazing with spatial and temporal coherence," *The Visual Computer*, vol. 27, pp. 749–757, 2011, 10.1007/s00371-011-0569-8. [Online]. Available: http://dx.doi.org/10.1007/s00371-011-0569-8

[38] A. Gut, *An intermediate course in probability*. New York: Springer, 2009.