Single Image Haze Removal Using Dark Channel Prior

HE, Kaiming

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Doctor of Philosophy

in

Information Engineering

The Chinese University of Hong Kong

August 2011

Abstract of thesis entitled:
Single Image Haze Removal Using Dark Channel Prior
Submitted by HE, Kaiming
supervised by Prof. TANG, Xiaoou
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in August 2011

Haze is a natural phenomenon that obscures scenes, reduces visibility, and changes colors. It is an annoying problem for photographers since it degrades image quality. It is also a threat to the reliability of many applications, like outdoor surveillance, object detection, and aerial imaging. So removing haze from images is important in computer vision/graphics.

But haze removal is highly challenging due to its mathematical ambiguity, typically when the input is merely a single image. In this thesis, we propose a simple but effective image prior, called *dark channel prior*, to remove haze from a single image. The dark channel prior is a statistical property of outdoor haze-free images: most patches in these images should contain pixels which are dark in at least one color channel. Using this prior with a haze imaging model, we can easily recover high quality haze-free images. Experiments demonstrate that this simple prior is powerful in various situations and outperforms many previous approaches.

Speed is an important issue in practice. Like many computer vision problems, the time-consuming step in haze removal is to combine pixel-wise constraints with spatial continuities. In this thesis, we propose two novel techniques to solve this problem efficiently. The first one is an unconventional large-kernel-based linear solver. The second one is a generic edge-aware filter which enables real-time performance. This filter is superior in various applications including haze removal, in terms of speed and quality.

The human visual system is able to perceive haze, but the underlying mechanism remains unknown. In this thesis, we present new illusions showing that the human visual system is possibly adopting a mechanism similar to the dark channel prior. Our discovery casts new insights into human vision research in psychology and physiology. It also reinforces the validity of the dark channel prior as a computer vision algorithm, because a good way for artificial intelligence is to mimic human brains.

摘要

霧霾是一種大氣遮蓋景物的自然現象。這種現象會降低景物能見度並改變景物顏色。霧霾會降低圖像質量，給攝影者們帶來困擾。霧霾也會影響許多實際應用的可靠性，例如戶外監控、物體檢測識別及航空拍攝。因此，圖像去霧在計算機視覺和計算機圖形學領域都是重要的課題。

但由於数學上的歧義性，圖像去霧是一個很困难的問題，尤其是當輸入僅為一張圖像的時候。在本論文中，我們提出一個簡單而有效的圖像先驗規律，實現單一圖像去霧。我們稱這個先驗規律為暗原色先驗，它是基於戶外無霧圖像的一個統計規律：這些圖像中的大多數區塊都包含有這樣的一些像素，它們的某一個原色會是很低的亮度。結合暗原色先驗和霧霾成像的物理規律，我們可以輕鬆地還原高質量的無霧圖像。實驗表明，這個簡單的先驗規律在多種情形下都非常有效，並且優於很多以前的方法。

速度是一個實際應用中的重要問題。和許多計算機視覺方法類似，我們的去霧算法花費了大量運算時間以結合單點約束和空間連續性。在本論文中，我們提出了兩種新方法來快速解決這個問題。第一種方法是一個非傳統的大核線性求解器。第二種方法是一種新穎的保邊濾波器，它能把我們的算法提升至實時的速度。我們還發現，無論是速度還是質量，這種濾波器在許多其它應用中也有優異的表現。

我們對霧霾的研究並不局限於計算機視覺領域。人類視覺系統擁有感知霧霾的能力，但相關機制還有待研究。在本論文中，我們提出了多個錯覺實驗，表明人類視覺系統很有可能也在使用類似暗原色先驗的機制。我們的發現為心理學和生理學中的人類視覺研究帶來了新的見解。我們的實驗也加強了暗原色先驗作為一種計算機視覺算法的可靠性，因為在人工智能領域，模仿人腦的算法通常會很有效。

Acknowledgement

It would not be possible for me to finish the PhD training without the help, guidance, and love from many people.

Firstly, I would like to thank my advisor, Prof. Xiaoou Tang. He led me to a new stage of life when I was struggling for a future in my undergraduate final year. He has offered me so many opportunities in the university and in Microsoft Research Asia. I would specially thank his patience, encouragement, and confidence before I could publish my first work until the second year. I have learned so much from him in the past four years, on how to do research, how to plan my career, and more importantly, how to be a man of integrity and kindness.

And it is my great fortune to work with Dr. Jian Sun, my mentor in Microsoft Research Asia. He is always enthusiastic, devoted, and full of brilliant ideas. With patience and broad knowledge, Jian has given me so much help in my research journey. I still remember how the seemingly insignificant ideas became insightful ones after countless discussions with him. He also taught me a lot on research attitudes.

I also want to thank all the members in the multimedia lab in the university and all my colleagues in the visual computing group in Microsoft Research Asia. They have shared with me their experience in research, coding, mathematics, paper writing, and daily life. I also acknowledge my collaborators Christoph Rhemann and Carsten Rother in Microsoft Research Cambridge. Their great efforts are essential in the last CVPR paper.

Last in this page but first in my heart, I would like to thank my parents and Miss Zitao Lu. It would not be possible for me to pass through all the obstacles without their love, support, and sacrifice.

To my family

Contents

List of Figures

Chapter 1

Introduction

Haze is an atmospheric phenomenon where turbid media obscure the scenes. Haze brings troubles to many computer vision/graphics applications. It reduces the visibility of the scenes and lowers the reliability of outdoor surveillance systems; it reduces the clarity of the satellite images; it also changes the colors and decreases the contrast of daily photos, which is an annoying problem to photographers (see Fig. 1.1 left[1]). Therefore, removing haze from images is an important and widely demanded topic in computer vision and computer graphics areas.

The main challenge lies in the *ambiguity* of the problem. Haze attenuates the light reflected from the scenes, and further blends it with some additive light in the atmosphere. The target of haze removal is to recover the reflected light (*i.e.*, the scene colors) from the blended light. This problem is mathematically ambiguous: there are an infinite number of solutions given the blended light. How can we know which solution is true? We need to answer this question in haze removal.

Ambiguity is a common challenge for many computer vision problems. In terms of mathematics, ambiguity is because the number of equations is smaller than the number of unknowns. The methods in computer vision to solve the ambiguity can roughly categorized into two strategies. The first one is to acquire more known variables, *e.g.*, some haze removal algorithms capture multiple images of the same scene under different settings (like polarizers). But it is not easy to obtain extra images in practice. The second strategy is to impose extra constraints using some knowledge or assumptions

[1] All the images in this thesis are best viewed in the electronic version.

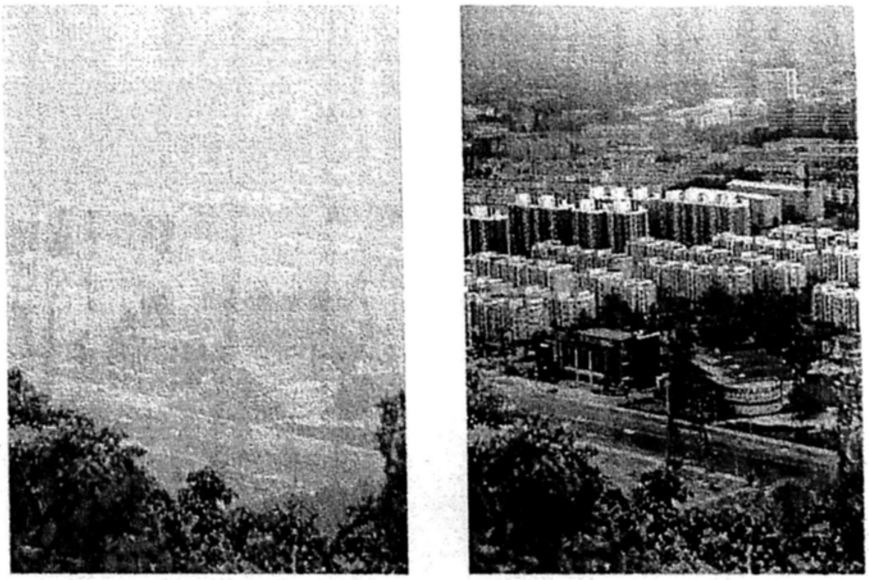


Figure 1.1: Haze removal from a single image. Left: input hazy image. Right: haze removal result of our approach.

known beforehand, namely, some "*priors*". This way is more practical since it requires as few as only one image. To this end, we focus on *single* image haze removal in this thesis. The key is to find a suitable prior.

Priors are important in many computer vision topics. A prior tells the algorithm "what can we know about the fact beforehand" when the fact is not directly available. In general, a prior can be some statistical/physical properties, rules, or heuristic assumptions. The performance of the algorithms is often determined by the extent to which the prior is valid. Some widely used priors in computer vision are the smoothness prior, sparsity prior, and symmetry prior.

In this thesis, we develop an effective but very simple prior, called the *dark channel prior*, to remove haze from a single image. The dark channel prior is a statistical property of outdoor haze-free images: most patches in these images should contain pixels which are dark in at least one color channel. These dark pixels can be due to shadows, colorfulness, geometry, or other factors. This prior provides a constraint for each pixel, and thus solves the ambiguity of the problem. Combining this prior with a physical haze imaging model, we can easily recover high quality haze-free images. Experiments demonstrate that our method is very successful in various situations (*e.g.*, Fig. 1.1 right) and outperforms many previous approaches.

Besides quality, speed is another concern in practical applications, typically in real-time video processing and interactive image editing. The time-

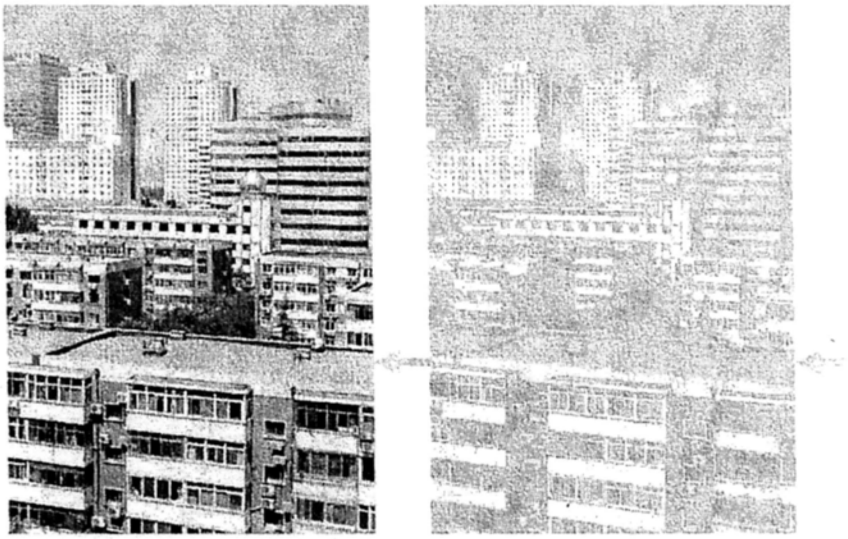


Figure 1.2: An illusion experiment. The roofs pointed by the arrows have exactly the same color, but they are perceived very differently by human eyes. This striking illusion suggests that the dark channel prior may be closely related with some human visual mechanisms. Details are given in Chapter 5.

consuming procedure in haze removal is to combine pixel-wise constraints and spatial continuities, which is a common problem in many computer vision topics. In this thesis, we propose two novel techniques to efficiently handle this problem. In the first technique, we achieve a faster speed by solving a large kernel linear system. This discovery is against conventional theories but we can prove its validity theoretically and experimentally. The second technique is a novel edge-aware filter. It is non-iterative and can be computed in real-time, but still exhibits very high quality. We find this filter superior to previous techniques in various edge-aware applications including haze removal. Thus, we advance the state-of-the-art in a broader area.

Our study on haze is not limited in computer vision. The human visual system (HVS) is also faced with the inherent ambiguities in many vision problems, including haze perception. But the HVS has good abilities to solve these ambiguities. People have long realized that the only way the HVS could do so is to use certain priors [25]. However, most of these priors remain mysterious in psychology and physiology. In this thesis, we design several psychologically based illusion experiments (*e.g.*, Fig. 1.2), which suggest that the dark channel prior is related to the mechanism used by the HVS to perceive haze. Our discovery casts new insights into human vision research area. It also reinforces the validity of the dark channel prior as a computer vision algorithm, because a good way for artificial intelligence is to mimic human brains.

1.1 Publications

This thesis involves the following publications during my PhD training period. The main concept and methods are published in:

- [30] Kaiming He, Jian Sun, and Xiaoou Tang. Single Image Haze Removal using Dark Channel Prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009 (accepted as oral).
- [33] Kaiming He, Jian Sun, and Xiaoou Tang. Single Image Haze Removal using Dark Channel Prior. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

Our work [30] receives the *CVPR Best Paper Award* in 2009. The two fast algorithms discussed in this thesis are published in:

- [31] Kaiming He, Jian Sun, and Xiaoou Tang. Fast Matting using Large Kernel Matting Laplacian Matrices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [32] Kaiming He, Jian Sun, and Xiaoou Tang. Guided Image Filtering. In *The 11th European Conference on Computer Vision (ECCV)*, 2010 (accepted as oral).

These two techniques are generic and applicable in haze removal, alpha matting, and edge-aware filtering. The above papers inspired a novel alpha matting method, which is not included in this thesis:

- [29] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A Global Sampling Method for Alpha Matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

1.2 Thesis Overview

This thesis is organized as follows. In Chapter 2 we introduce the physical model of haze and give a comprehensive study on existing haze removal methods. In Chapter 3 we introduce the dark channel prior and apply it to single image haze removal. In Chapter 4 we discuss two fast algorithms and their applications in haze removal and other cases. In Chapter 5 we study the relationship between the dark channel prior and the human visual system. We conclude in Chapter 6.

□ **End of chapter.**

Chapter 2

Background

In the first part of this chapter, we study the physical model of the haze effect, which is given by a concise *haze imaging equation.* This equation is the focus of the whole thesis. We also formulate the haze removal problem and point out the challenges in this problem.

In the second part of this chapter, we review the previous works on haze removal. We investigate both multiple-image and single image cases, on their advantages and limitations.

2.1 Haze Imaging Model

The *haze imaging equation* is given by [49, 18, 79]:

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

An example of the haze imaging equation is given in Fig. 2.1. The variables are explained in the following:

- $\mathbf{x} = (x, y)$ is a 2D vector representing the coordinates (x, y) of a pixel's position in the image.
- $\mathbf{I}$ represents the *hazy image* observed. $\mathbf{I}(\mathbf{x})$ is a 3D RGB vector of the color at a pixel.

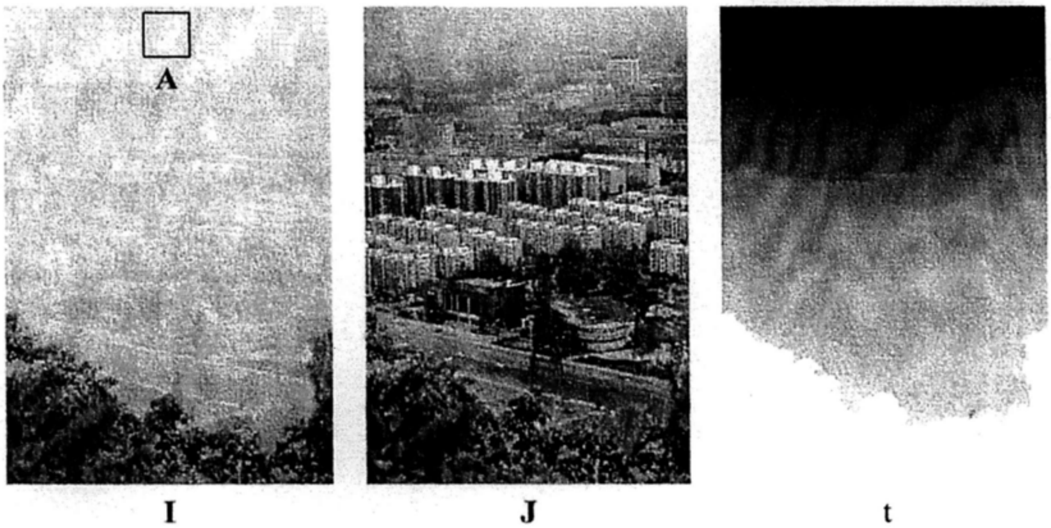


Figure 2.1: Variables in the haze imaging equation. The transmission map t is shown as white when t=1, and black when t=0.

- $\mathbf{J}$ represents the *scene radiance* image. $\mathbf{J}(\mathbf{x})$ is a 3D RGB vector of the color of the light reflected by the scene point at $\mathbf{x}$. It would be the light seen by the observer if this light were not through the haze. So we often refer to the scene radiance $\mathbf{J}$ as a *haze-free* image. See Fig. 2.1.

- t is a map called *transmission* or transparency of the haze. $t(\mathbf{x})$ is a scalar in $[0, 1]$. Intuitively, $t(\mathbf{x}) = 0$ means completely hazy and opaque, $t(\mathbf{x}) = 1$ means haze-free and completely clear, and $0 < t(\mathbf{x}) < 1$ means semi-transparent. See Fig. 2.1.

- $\mathbf{A}$ is the *atmospheric light*. It is a 3D RGB vector usually assumed to be spatially constant. It is often considered as "the color of the atmosphere, horizon, or sky" [49, 18, 79].

A physical view is shown in Fig. 2.2. The haze is formed by the particles in the atmosphere absorbing and scattering light. The term $\mathbf{J}(\mathbf{x})t(\mathbf{x})$ in (2.1) is called *direct attenuation*. The light reflected from an object is partially absorbed by the particles in the atmosphere and is attenuated. The transmission t is the ratio of "the light that is not attenuated and reaches the observer" to "the light reflected from the object". The term $\mathbf{A}(1 - t(\mathbf{x}))$ is called *airlight* [40, 18][1]. The particles scatter the light they absorb, playing as an infinite number of tiny light sources floating in the atmosphere. The airlight is due to these light sources. A detailed physical derivation of the

[1] Sometimes researchers call $\mathbf{A}$ "airlight" instead.

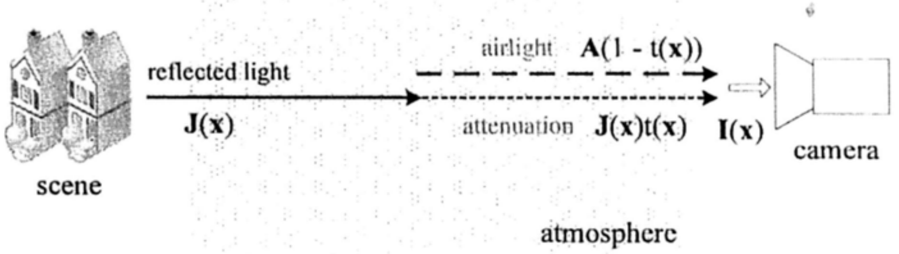


Figure 2.2: A macro physical picture of the haze imaging model.

haze imaging equation is in Appendix A. Note the direct attenuation is a multiplicative effect, whereas the airlight is an additive one.

The atmospheric phenomena of haze, fog, and mist are all due to the particles in the atmosphere, like dust, sand, water droplets, or ice crystals. In meteorology, all these phenomena mainly differ in their particle material, size, shape, and concentration[87], but their physical influences on imaging are similar[49]. The haze imaging equation (2.1) is valid for all these phenomena, and a haze removal algorithm should handle all of them. In this thesis, we do not distinguish their meteorological properties, and refer to them as "haze" for simplicity unless specified. But one should keep in mind that we are handling a more general concept including haze, fog, and mist.

2.1.1 Depth

Suppose that a scene point in the position $\mathbf{x}$ has the distance $d(\mathbf{x})$ from the observer. d is called the *depth* of the scene point. It is found that the haze transmission t is physically related to the depth d (see Appendix A for details):

$$t(\mathbf{x}) = \exp\left(-\int_0^{d(\mathbf{x})} \beta(z)dz\right). \tag{2.2}$$

Here, β is the *scattering coefficient* of the atmosphere. β is determined by the physical properties of the atmosphere, like particle material, size, shape, and concentration. The integral $\int_0^{d(\mathbf{x})}$ is on a line between a scene point and the observer.

If the physical properties of the atmosphere are homogenous, the scattering coefficient β is a spatial constant. Thus, we can rewrite (2.2) as

$$t(\mathbf{x}) = \exp\left(-\beta d(\mathbf{x})\right), \tag{2.3}$$

or equivalently:

$$d(\mathbf{x}) = -\frac{\ln t(\mathbf{x})}{\beta}. \tag{2.4}$$

Equation (2.4) says if we can estimate the transmission t, we can calculate the depth up to an unknown scale[2]. In computer vision, depth estimation is a very important yet difficult problem [64]. In this sense, the bad haze can be put to good use if we can separate the transmission t. This is another important motivation in haze removal.

The constant-β assumption is invalid when the atmosphere is inhomogeneous, *e.g.*, when the haze is concentrated in some regions and forms cloud-like media. In this case, we cannot use (2.4) to estimate the depth.

2.1.2 Image Degradation

Given the haze imaging equation $\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x}))$, let us see how haze degrades images.

The first degradation is the visibility reduction due to the direct attenuation. Visibility is a measure of how well the object can be discerned. In computer vision visibility is often described by the gradient of the image. From (2.1) we have:

$$\nabla\mathbf{I}(\mathbf{x}) = t(\mathbf{x})\nabla\mathbf{J}(\mathbf{x}), \tag{2.5}$$

where we consider t as uniform so the gradient of t is ignored. Because t is in the range $[0, 1]$, $\nabla\mathbf{I}$ has a smaller magnitude than $\nabla\mathbf{J}$. So the visibility is reduced and the objects are more difficult to discern (see Fig. 2.3). We can see that visibility reduction is due to the direct attenuation $\mathbf{J}(\mathbf{x})t(\mathbf{x})$, or say, due to the *multiplicative* t.

The second degradation is chrominance shift due to the airlight. The chrominance describes the colorfulness regardless of the luminance, which is represented by the direction of the color vector in the RGB color space. The haze imaging equation (2.1) suggests the vector $\mathbf{I}(\mathbf{x})$ is a linear combination of the two vectors $\mathbf{J}(\mathbf{x})$ and $\mathbf{A}$ (see Fig. 2.4). Due to the *additive* $\mathbf{A}$, the

[2] This scale is less important in many cases, because people are often interested in relative depth.

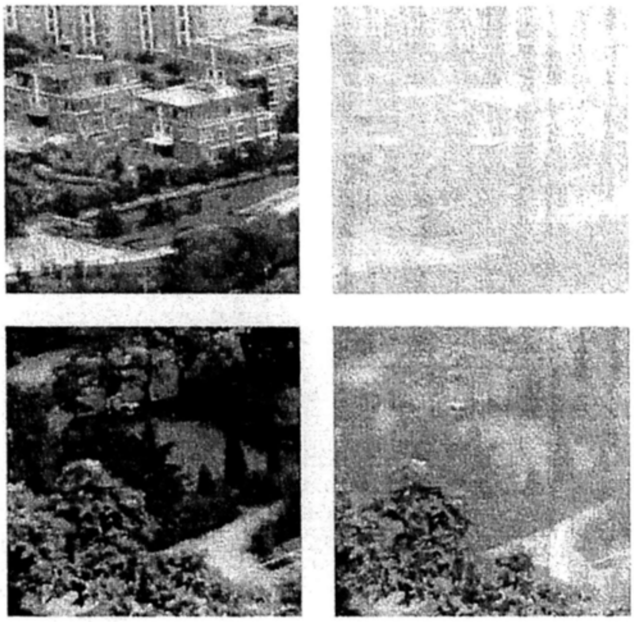


Figure 2.3: Images degraded by haze. On the left are the scene radiance **J**. On the right are the observed hazy images **I**.

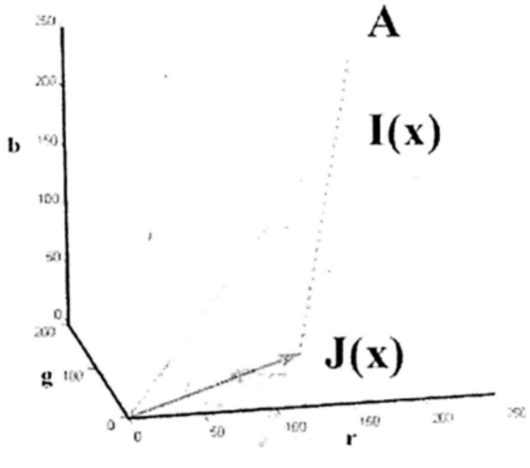


Figure 2.4: The color vector **I** is a linear combination of **J** and **A** in the RGB space.

vectors $\mathbf{I}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ are not in the same direction: the chrominance is shifted. Usually the atmosphere is white or gray, so a hazy image appears grayish and less vivid. See Fig. 2.3.

In sum, the *multiplicative* attenuation reduces the visibility, and the *additive* airlight changes the chrominance. Haze is troublesome in many computer vision/graphics applications. The reduced visibility impacts object detection and recognition, lowers the reliability of outdoor surveillance systems, and obscures the satellite images. In consumer-level photography, haze changes the colors and reduces the contrast of the photos. The degradation cannot be avoided by higher level cameras or better lens, because it happens in the atmosphere before reaching the apparatus. Therefore, removing the haze effects from images is demanded in computer vision/graphics.

2.1.3 Problem Formulation and Ambiguity

We have introduced the haze imaging equation:

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

with

$$t(\mathbf{x}) = e^{-\int_0^d \beta(z)dz}. \tag{2.2}$$

The target of haze removal is: given the input hazy image $\mathbf{I}$, recover the scene radiance image $\mathbf{J}$. Usually we also need to recover t and $\mathbf{A}$ (see Fig. 2.1). Next we explain why this problem is mathematically ambiguous.

Denote the number of pixels in an image as N. If the input $\mathbf{I}$ is an RGB color image, we have a set of $3N$ equations:

$$I_c(\mathbf{x}) = J_c(\mathbf{x})t(\mathbf{x}) + A_c(1 - t(\mathbf{x})), \tag{2.6}$$

where the scalars $I_c(\mathbf{x})$, $J_c(\mathbf{x})$, and A_c are the color components in the channel $c \in \{r, g, b\}$. However, we have $3N$ unknown reflections J_c, N unknown transmission t, and 3 unknown atmospheric light A_c. The total number of unknowns are $4N + 3$, much greater than the number $3N$ of equations. In computer vision, we refer to this problem as ambiguous, ill-posed, or under-constrained. The ambiguity is mainly due to the spatially variant t, which contributes N variables. So we require at least one extra constraint for each pixel to solve the ambiguity.

The physical meaning of the ambiguity can be understood in the following way. the haze plays a role like a semi-transparent glass filter. The color of the filter is $\mathbf{A}$, and the transparency is t. An object with a color $\mathbf{J}$ is seen through the filter (see Fig. 2.5). Objects in different colors can be observed

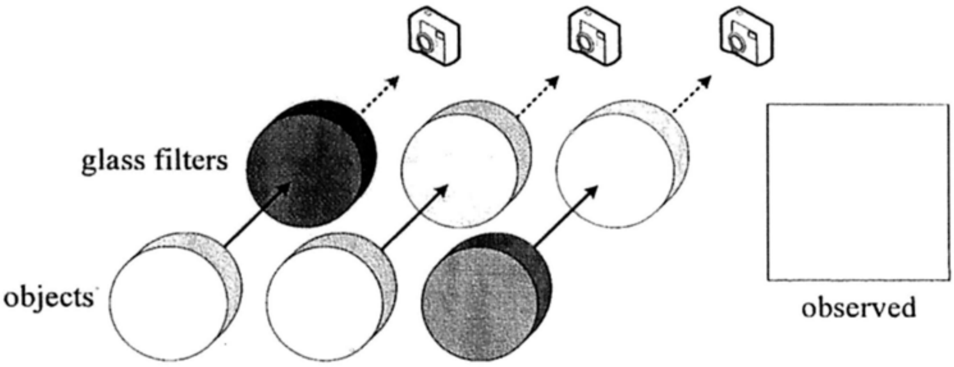


Figure 2.5: The same observed color can be explained by objects in different colors seen through filters in different colors.

as the same, if they are seen through filters in different colors with proper transparency. So given the observed color, how can we know the object color? We are faced with a similar question in haze removal.

2.2 Related Works

In computer vision, the methods to handle the ambiguity are roughly on two ways. The first way is to acquire more known variables, reducing the discrepancy between the number of equations and the number of unknowns. In haze removal, this is often by capturing two or more images of the scene. The second way is to use some knowledge or assumptions known beforehand, *i.e.*, *priors*. The priors impose extra constraints/dependency among the unknown variables.

In the following, we review the previous haze removal methods in both categories. We do not discuss the technical details of these methods. Instead, we are interested in how they introduce extra constraints. All the methods are reformulated in a same framework, though they may be expressed in different forms in the original works. We believe that this is helpful to better compare these methods.

2.2.1 Multiple-Image Haze Removal

Some earlier methods take two or more images of the same scene. Though this strategy increases the number of known variables, at the same time it

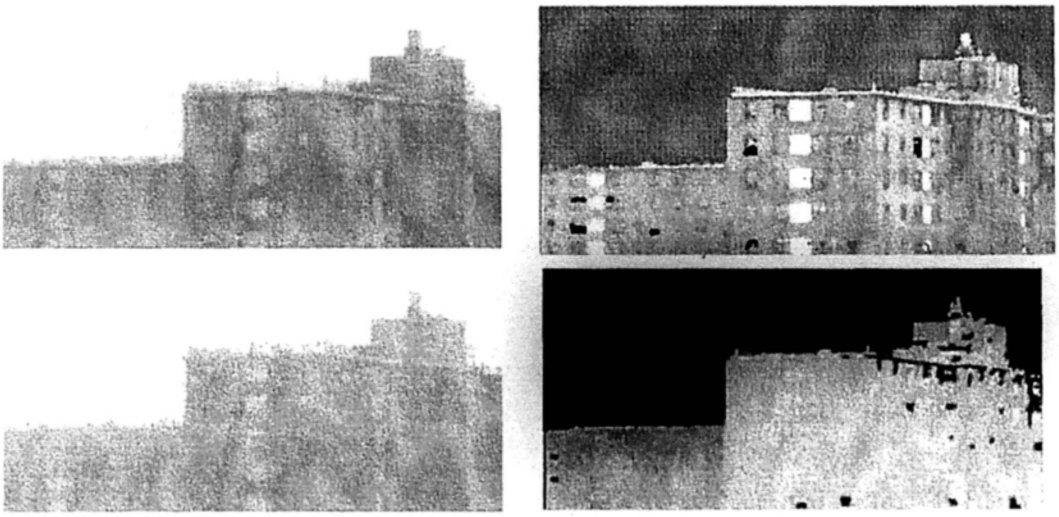


Figure 2.6: Haze removal based on varying atmospheric conditions. On the left are two images taken in two hazy conditions. On the right are the estimated scene radiance and depth (the sky is ignored). Images from [48].

brings in more unknowns. So the setting must under certain constraints to avoid too many unknowns being introduced.

Methods Based on Varying Atmospheric Conditions

The *dichromatic* method proposed in [52, 48, 50] takes at least two images of the same scene, under different atmospheric conditions (Fig. 2.6 left). The two images are strictly aligned. Thus the two images shares the same depth $d(\mathbf{x})$ and the same reflectance[3] $\rho_c(\mathbf{x})$.

It further assumes that the atmospheric light $\mathbf{A}$ is the only light source of the scene. Then the reflection $\mathbf{J}$ satisfies:

$$J_c(\mathbf{x}) = \rho_c(\mathbf{x})A_c. \tag{2.7}$$

Under the constant β assumption, the haze imaging equations of the two

[3] The term *reflectance* refers to the ratio of the reflected light to the incident light, whereas the term *reflection* refers to the reflected light alone.

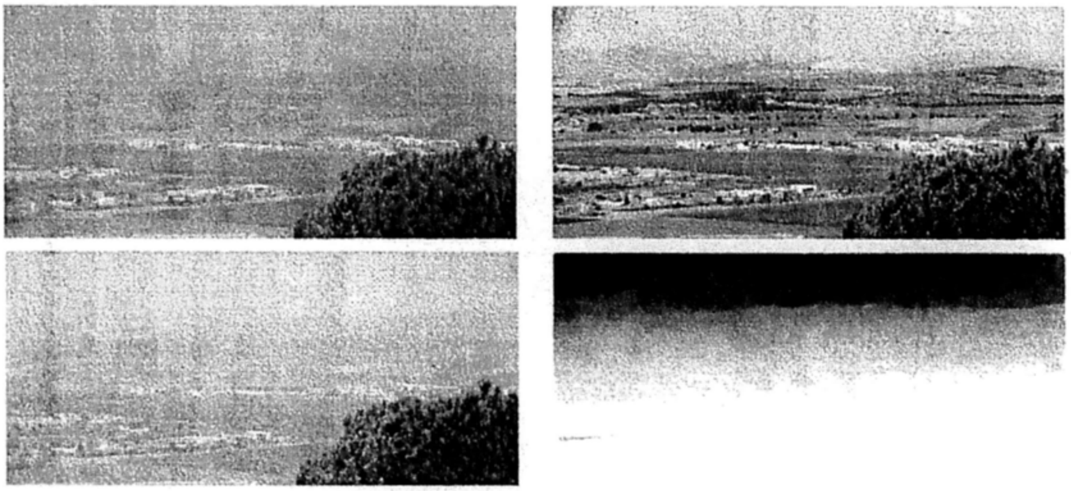


Figure 2.7: Polarization-based methods. On the left are two images taken in two polarizer states. On the right are the estimated scene radiance and depth. Images from [65].

images are:

$$\begin{aligned} I_c^1(\mathbf{x}) &= \rho_c(\mathbf{x})A_c^1 t^1(\mathbf{x}) + A_c^1(1 - t^1(\mathbf{x})) \\ I_c^2(\mathbf{x}) &= \rho_c(\mathbf{x})A_c^2 t^2(\mathbf{x}) + A_c^2(1 - t^2(\mathbf{x})), \end{aligned} \tag{2.8}$$

where the superscript ($\cdot^1$ or $\cdot^2$) is the image index. This is called the *dichromatic* model in [52, 48, 50].

The equations (2.8) provide $6N$ constraints, together with $3N$ unknown ρ_c, $2N$ unknown t, and 6 unknown A_c: in total $5N + 6$ unknown variables. The constraints outnumber the unknowns, and the problem becomes *over-constrained*. This problem can be solved by standard fitting algorithms. Fig. 2.6 show an example result.

The above analysis is valid only when $t^1(\mathbf{x}) \neq t^2(\mathbf{x})$. To ensure this condition, the two images must be taken under very different atmospheric conditions, *e.g.*, one in denser haze and the other in thinner haze. This is not an easy task: the weather may remain unchanged in several minutes or even hours. This is the main limitation of this method.

Polarization-based Methods

The methods in [65, 66] utilized a physical effect of the haze - polarization. The airlight is highly polarized, but the direct attenuation is much less.

Using a polarizer (a glass filter) attached in the camera lens, these methods take two images of the same scene under two polarization states (see Fig. 2.7). Assume the direct attenuation is completely unpolarized, the haze imaging equations of the two images are:

$$\begin{aligned} I_c^{\parallel}(\mathbf{x}) &= \frac{1}{2}J_c(\mathbf{x})t(\mathbf{x}) + A_c^{\parallel}(1 - t(\mathbf{x})) \\ I_c^{\perp}(\mathbf{x}) &= \frac{1}{2}J_c(\mathbf{x})t(\mathbf{x}) + A_c^{\perp}(1 - t(\mathbf{x})). \end{aligned} \tag{2.9}$$

Here $\parallel$ and $\perp$ denote two states, and the factor $\frac{1}{2}$ is the influence of a polarizer to unpolarized light.

The equations (2.9) provide $6N$ constraints, together with $3N$ unknown J_c, N unknown t, 3 unknown $A_c^{\parallel}$ and 3 unknown $A_c^{\perp}$: in total $4N+6$ unknown variables. The problem becomes over-constrained and can be solved. Fig. 2.7 is an example.

A limitation of the polarization-based methods is the settings. Capturing two strictly aligned polarized images is troublesome in practice. Another problem is that the direct transmission is not always completely unpolarized. Besides, a recent work [81] finds that polarization-based methods increase the noise and are not beneficial to visibility.

Methods Based on Given Depth

The methods in [51, 39] use some given depth information to remove the haze. As the depth d is a map describing the structure of the scene, we categorize these methods as multiple-image ones.

Under the constant-β assumption, the haze imaging equation becomes:

$$I_c(\mathbf{x}) = J_c(\mathbf{x})e^{-\beta d(\mathbf{x})} + A_c(1 - e^{-\beta d(\mathbf{x})}) \tag{2.10}$$

We have $3N$ equations in (2.10), together with $3N$ unknown J_c, 3 unknown A_c, and one unknown β: in total $3N + 4$ unknowns. The problem is almost well-posed. The extra unknown variables (A_c and β) can be estimated by

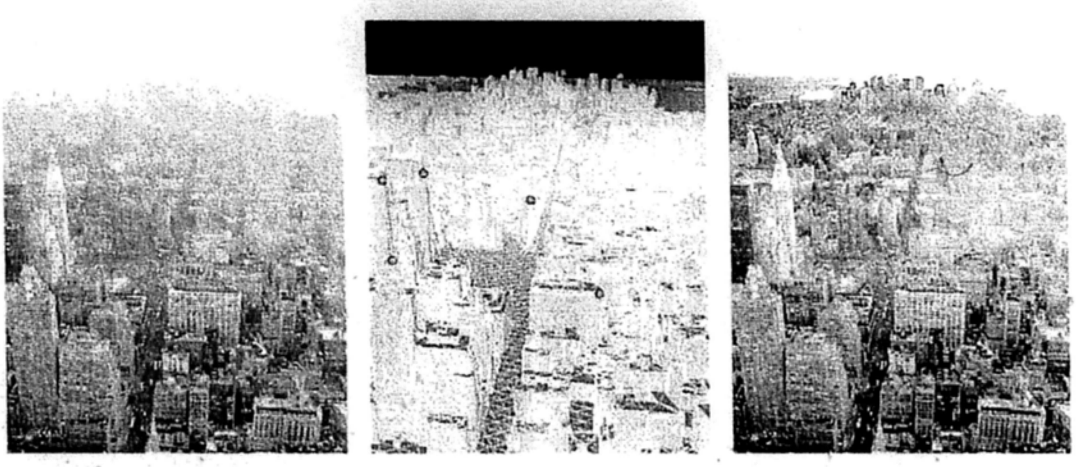


Figure 2.8: Methods Based on Given Depth. From left to right: hazy image, given 3D structure, haze removal result. Images from [39].

fitting [39].

But it is not trivial to obtain the depth information. In [51], the user is asked to roughly draw the depth. In [39], the 3D structural model of the scene is provided by some databases, such as Google Earth or NASA radar images. The 3D structures are then aligned to the hazy image and provide the depth. See Fig. 2.8 for an example. However, the 3D structure model is not available in most cases.

Summary

The multiple-image haze removal methods share some common advantages and limitations.

These methods turn the ill-posed problem into a well-posed or over-constrained one. One benefit is the fast running time. The computation is often pixel-wise and no complex optimization is needed. Another benefit is that they may handle some special situations. For example, the dichromatic methods can handle night images, and the polarization-based or depth-based methods allow t to vary across color channels.

The common limitation of these methods is that the extra images are not easily available. They all require special settings with carefully calibration. This is not practical in most cases, such as for hand-held cameras and outdoor surveillance systems. The limitations of these methods motivate the development of single image haze removal methods.

2.2.2 Single Image Haze Removal

Single image haze removal methods have to rely on some priors. The priors can be statistical/physical properties, heuristic assumptions, simplifications, and application-based rules. In the haze imaging model (2.1), the discrepancy between the number of equations ($3N$) and the number of unknowns ($4N+3$) is about N. So the priors are expected to introduce at least one constraint for each pixel.

Dark Object Subtraction

A straightforward assumption is that the transmission t can be treated as a constant independent of the position $\mathbf{x}$:

$$I_c(\mathbf{x}) = J_c(\mathbf{x})t + A_c(1-t) \tag{2.11}$$

Thus the number of unknowns about the transmission t reduces from N to 1.

The *dark object subtraction* proposed in [9] is to find this sole unknown variable. This method is first developed in remote sensing area where the images are often multi-spectral (often more than three channels). Consider a single channel c. It is assumed *there must be an object that is dark in this channel.* Specifically, there exists at least a pixel satisfying:

$$J_c(\mathbf{x}) = 0 \tag{2.12}$$

and therefore:

$$I_c(\mathbf{x}) = A_c(1-t). \tag{2.13}$$

The pixel satisfying $J_c(\mathbf{x}) = 0$ corresponds to the minimum value in $I_c(\mathbf{x})$. So we can obtain the constant t from (2.13) if A^c has been given.

However, the constant-t assumption limits its application. This method is often applied in satellite images where all the scene points have the same distance with the camera. But this is rarely true in other cases where the depth is not constant. Fig. 2.9 shows an example. We can see that the method cannot remove the distant haze.

The dark object subtraction partially inspires the *dark channel prior* proposed in this thesis.

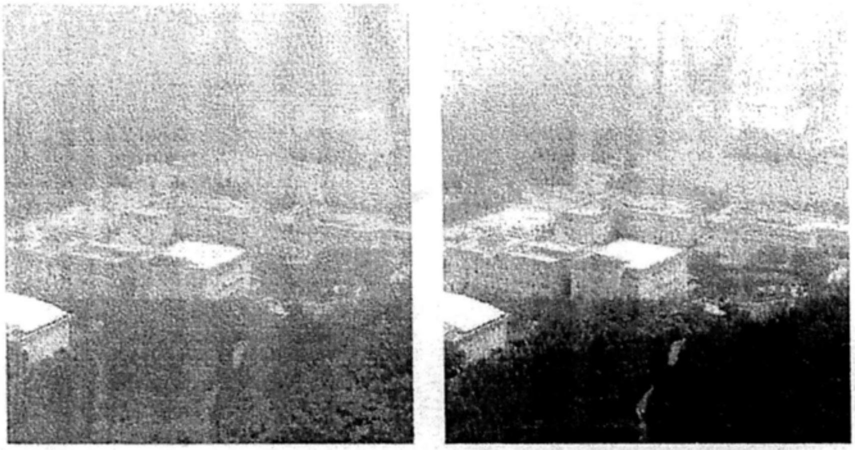


Figure 2.9: Haze removal by dark object subtraction [9]. Left: input hazy image. Right: haze removal result.

Visibility Maximization

Haze reduces visibility. Removing the haze will enhance the visibility of the image. In [79] Tan proposes a method to maximize the visibility under the constraint of the haze imaging model.

Consider a small local patch. The transmission t is treated to be constant inside this patch, so $\nabla I_c(\mathbf{x}) = t\nabla J_c(\mathbf{x})$. The total visibility inside the patch is defined as the sum of the gradient magnitude:

$$\sum_{c,\mathbf{x}} |\nabla J_c(\mathbf{x})| = \frac{1}{t}\sum_{c,\mathbf{x}} |\nabla I_c(\mathbf{x})|. \tag{2.14}$$

The visibility of the recovered image $\sum_{c,\mathbf{x}} |\nabla J_c(\mathbf{x})|$ will keep increasing when the transmission t is decreasing. Tan supposes the atmospheric light is the light source of the scene, so $J_c(\mathbf{x}) = \rho_c(\mathbf{x})A_c$ with $0 <= \rho_c(\mathbf{x}) <= 1$. This leads to the following constraint:

$$0 <= J_c(\mathbf{x}) <= A_c. \tag{2.15}$$

In Tan's method, the value $J_c(\mathbf{x})$ outside this range is truncated. This operation prevents the visibility of $\sum_{c,\mathbf{x}} |\nabla J_c(\mathbf{x})|$ from increasing because the truncated values provides zero gradients. The optimal t is the value that maximizes the visibility. This computation is performed in each patch, pro-

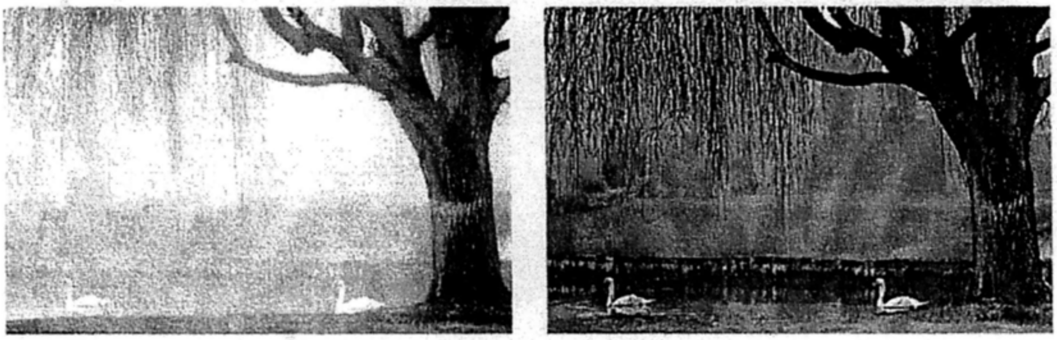


Figure 2.10: Haze removal by maximizing visibility [79]. Left: input hazy image. Right: haze removal result.

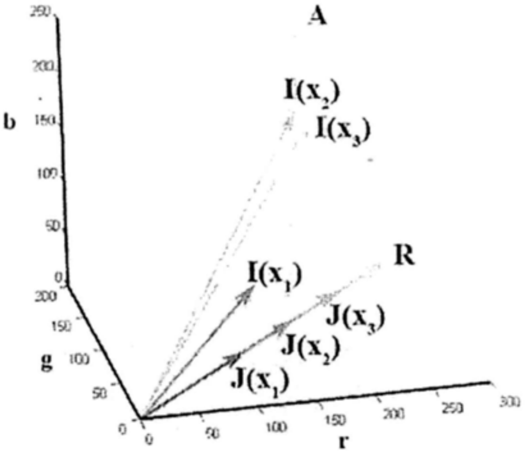


Figure 2.11: The color vectors in the RGB space. Three pixels at $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ have the same reflectance R, but different luminance and transmission.

viding a constraint for each pixel.

Tan's method has the advantages of enhancing the visibility. But it has some limitations. Although the scene radiance should have a better visibility than the hazy image, it does not necessarily have the maximum visibility. Moreover, the truncation operation is not physically valid. The resulting images often look over-saturated and unnatural, as in Fig. 2.10.

Independent Component Analysis

In [18] Fattal solves the haze removal problem by Independent Component Analysis (ICA). ICA is a statistical method to separate two additive components from a signal. Fattal adapts this method to the haze problem.

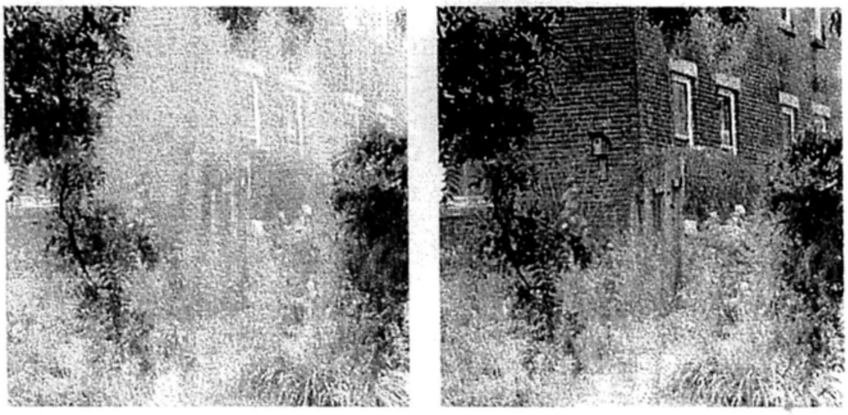


Figure 2.12: Haze removal using Independent Component Analysis. Left: input hazy image. Right: haze removal result

This method considers the scene points with the same reflectance in a local patch. Thus the colors $\mathbf{J}(\mathbf{x})$ of these scene points are in the same direction $\mathbf{R}$, as shown in Fig. 2.11. Denote the luminance by $l(\mathbf{x})$. Then $\mathbf{J}(\mathbf{x})$ is $\mathbf{R}l(\mathbf{x})$, and the haze imaging equation becomes:

$$I_c(\mathbf{x}) = R_c l(\mathbf{x}) t(\mathbf{x}) + A_c(1 - t(\mathbf{x})) \tag{2.16}$$

The two scalar components $l(\mathbf{x})$ and $t(\mathbf{x})$ are to be separated. The luminance $l(\mathbf{x})$ depends on the illumination, object reflectance, and the scene geometry, whereas the transmission $t(\mathbf{x})$ depends on the depth and the property of the haze. Fattal's assumption is that: these two components are due to unrelated sources and therefore statistically independent. Under this assumption the method can recover the reflectance of the pixels and provides extra constraints. We omit the technical details. Unlike Tan's method, this method is physically valid. The results often looks more natural and visually pleasing (see Fig. 2.12).

The main limitation of this method results from the locally based statistics. To ensure the statistics is reliable, the method requires the two components $l(\mathbf{x})$ and $t(\mathbf{x})$ to vary significantly in a local patch. This condition is not always satisfied. For example, a local patch of a distant scene usually exhibit negligible variance in $t(\mathbf{x})$. To handle this problem, the method only apply the ICA to some reliable patches. The constraints are thus missed in some pixels. This method uses a Markov Random Field (MRF) to extrapolate the missing values. But the results are not satisfactory when the reliable pixels are insufficient. This method does not work well for dense haze.

Summary

In single image haze removal, the difference between the number of unknowns and the number of equations is about the number of pixels N. This suggests the prior should impose at least one constraint for each pixel.

If each pixel is considered independently, it is perhaps impossible for us to make any assumption, just as the glass filter experiment in Fig. 2.5. But we can make assumptions or built priors for a group of pixels. Both Tan's method [79] and Fattal's method [18] consider small local patches. Tan's visibility maximization assumption can be applied in each patch, but is less physically valid. Fattal's ICA method is physically based, but it is only applied in some patches. Therefore, we expect a robust prior that is reliable in as many patches as possible. The dark channel prior proposed in this thesis is on this way.

□ **End of chapter.**

Chapter 3

Dark Channel Prior and Single Image Haze Removal

In this chapter, we propose a novel prior - *dark channel prior* - for single image haze removal. The dark channel prior is based on the statistics of outdoor haze-free images. We find that, in most of the local regions which do not cover the sky, it is very often that some pixels (called *dark pixels*) have very low intensity in at least one color (RGB) channel. In hazy images, the intensity of these dark pixels in that channel is mainly contributed by the airlight. Therefore, these dark pixels can directly provide an accurate estimation of the haze transmission. To improve the quality of the resulting transmission map, we develop a soft matting interpolation method. Various experiments show that we can recover a high quality haze-free image and produce a good depth map.

Our approach is physically valid and is able to handle distant objects in heavily hazy images. We do not rely on significant variance of transmission or surface shading. The result contains few artifacts.

3.1 Dark Channel Prior

A prior is the assumption or knowledge that can be built beforehand. Tan's method [79] is based on the prior that the scene radiance **J** should have a better visibility than the hazy image **I**. Fattal's method [18] is based on the prior that the object luminance l and the transmission t are statistically independent. We find that both priors talk about the *interaction* between the scene radiance (**J** or l) and the haze influence (**I** or t). But let us consider an image taken in a clear day in which no haze exists. Human beings are able to tell whether it is a *haze-free* image, even there is no interaction between the scene radiance and the haze at all. This motivates us to find a prior, which concerns the scene radiance **J** (an haze-free image) *alone*.

We propose the *dark channel prior* which is solely about haze-free image. In the following, we first propose our observation and give intuitive explanation. Then we introduce the dark channel prior in a mathematical form. We further design experiments to verify this prior.

3.1.1 Observation

Our observation is as following:

> *For outdoor haze-free images, in most patches that do not cover the sky, there exist some pixels whose intensity is very low and close to zero in at least one color channel.*

We refer to the pixels "whose intensity is very low and close to zero in at least one color channel" as *dark pixels*. To understand this observation, we explain what factors contribute to the dark pixels.

Factors contributing to dark pixels

First, the dark pixels can come from the shadows in the image. Outdoor images are full of shadows, *e.g.*, the shadows of trees, buildings, and cars. Objects with irregular geometry like rocks and plants are easily shaded. In most cityscape images, the windows of the buildings look dark from the outside, because the indoor illumination is often much weaker than the outdoor light. This can also be considered as a kind of shadows. See the first row in Fig. 3.1 for examples.

Second, the dark pixels can come from colorful objects. Any object with low reflectance in any color channel will result in dark pixels. For example,

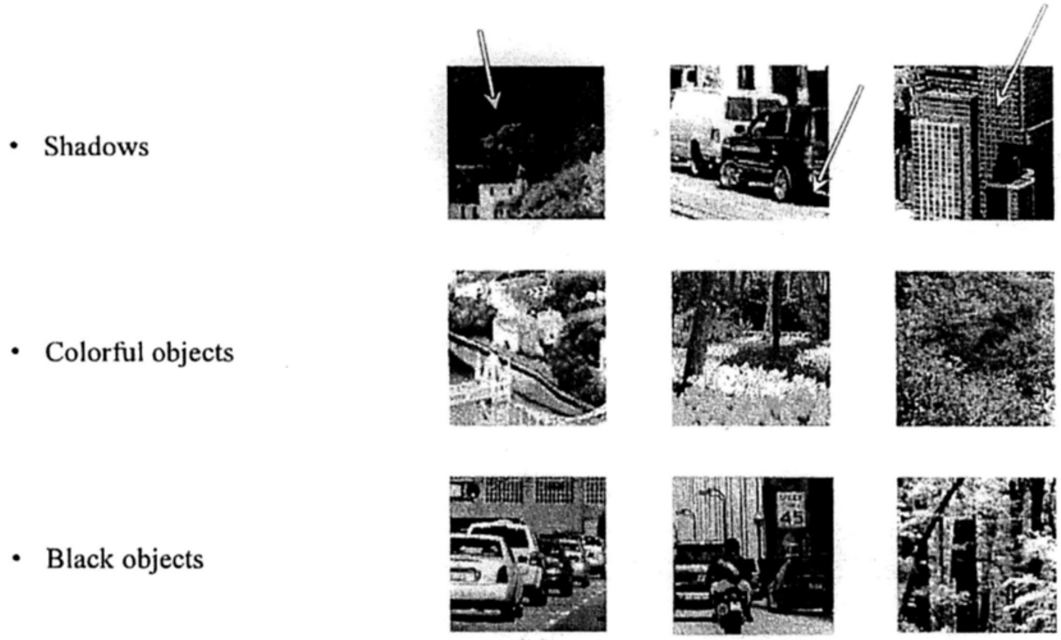


Figure 3.1: Shadows, colorful objects, and black objects contribute dark pixels.

a green color has low intensity in its red and blue channels, and a yellow color has low intensity in its blue channels. Outdoor images often contain objects in various colors, like flowers, leaves, cars, buildings, road signs, or pedestrians. See the second row in Fig. 3.1. The colorfulness of these objects generates many dark pixels. Notice that by our definition a dark pixel is not necessarily dark in terms of its *total* intensity; it is sufficient to be dark in only one color channel. So a bright red pixel can be a dark pixel if only its green/blue component is dark.

Third, the dark pixels can come from black objects, like vehicles tyres, road signs, and tree trunks. See the third row in Fig. 3.1. These dark pixels are particularly useful for in-vehicle camera which oversees the road conditions.

If an image patch includes at least one of these factors, this patch must have dark pixels. This is the intuitive explanation of our observation.

Remarks

It is worth mentioning some remarks on this observation.

(a) The observation only talks about *haze-free* images, *i.e.*, the scene radiance **J**. This is because the above four factors contributing the dark pixels concern the properties of the scene objects, such as shading, colors, reflectance, and geometry. These factors do not concern the reflected light traveling in the atmosphere. Actually, we shall see that this observation does not hold for hazy images (so we can use it to remove haze).

(b) The observation is on *most* patches but not all. By "most" we mean that there is a high probability that the observation is true. So this observation is in the sense of statistics. It is possible that some patches or images do not obey it, but such cases are rare. We shall verify this observation by statistical experiments.

(c) Currently we do not set the size of the patch. A larger patch has a better chance to contain a dark pixel. But we expect the observation to provide local constraints instead of global ones. So the patch size cannot be too large. We discuss this problem in the experiments.

(d) The observation is on *outdoor* images. Haze only occurs in outdoor images, so we are mainly interested in this case. Besides, a patch in an outdoor image often covers richer contents than an indoor image, typically when the scene objects are far away. So the probability that a patch covers dark pixels is higher in outdoor images.

(e) We ignore the sky regions in this observation. This is because the color of the sky is hard to predict. Fortunately, we shall see that our haze removal method can gracefully handle both non-sky and sky regions. So we do not have to treat the sky separately. More details are given in the algorithm section.

(f) We should also notice that the intensity of a pixel depends not only on the object colors or the light reflected, but also on the exposure settings of the camera (shutter speed, aperture, ISO, etc.). Black surfaces or shaded regions also reflect lights, and the corresponding pixels can be bright when the exposure value is high. We assume the images are taken under proper exposure settings so that a black color is correctly recorded by a low intensity value.

3.1.2 Mathematical Formulation

Next we discuss the mathematical formulation of the above observation.

Dark Pixel

Formally, we define a dark pixel as "the pixel whose minimum intensity among the three RGB channels is below a small threshold δ":

$$\mathbf{J}(\mathbf{x}) \text{ is a dark pixel} \iff \min_{c\in\{r,g,b\}}(J_c(\mathbf{x})) <= \delta. \tag{3.1}$$

Accordingly, if $\mathbf{J}(\mathbf{x})$ is a dark pixel, all the pixels $\mathbf{J}(\mathbf{x}')$ that satisfying the following condition are also dark pixels:

$$\min_{c\in\{r,g,b\}}(J_c(\mathbf{x}')) <= \min_{c\in\{r,g,b\}}(J_c(\mathbf{x})). \tag{3.2}$$

Here $\mathbf{x}'$ denote the coordinates of another pixel. This inequality suggests the following necessary and sufficient condition for a patch to contain at least one dark pixel:

$$\text{A patch } \Omega \text{ contains at least one dark pixel} \iff \min_{\mathbf{x}'\in\Omega}(\min_{c\in\{r,g,b\}}(J_c(\mathbf{x}'))) <= \delta. \tag{3.3}$$

In the following, we do not consider the small threshold δ explicitly, and simply write the inequality as:

$$\min_{\mathbf{x}'\in\Omega}(\min_{c\in\{r,g,b\}}(J_c(\mathbf{x}'))) \approx 0. \tag{3.4}$$

We can see we only need to concern the quantity $\min_{\mathbf{x}'\in\Omega}(\min_{c\in\{r,g,b\}}(J_c(\mathbf{x}')))$. This motives us to define a *dark channel*.

Dark Channel

Denote $\Omega(\mathbf{x})$ as the patch centered at the pixel $\mathbf{x}$. Given an image $\mathbf{J}$, the *dark channel* of $\mathbf{J}$ is defined as a map satisfying:

$$J^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{x}'\in\Omega(\mathbf{x})}(\min_{c\in\{r,g,b\}} J_c(\mathbf{x}')) \tag{3.5}$$

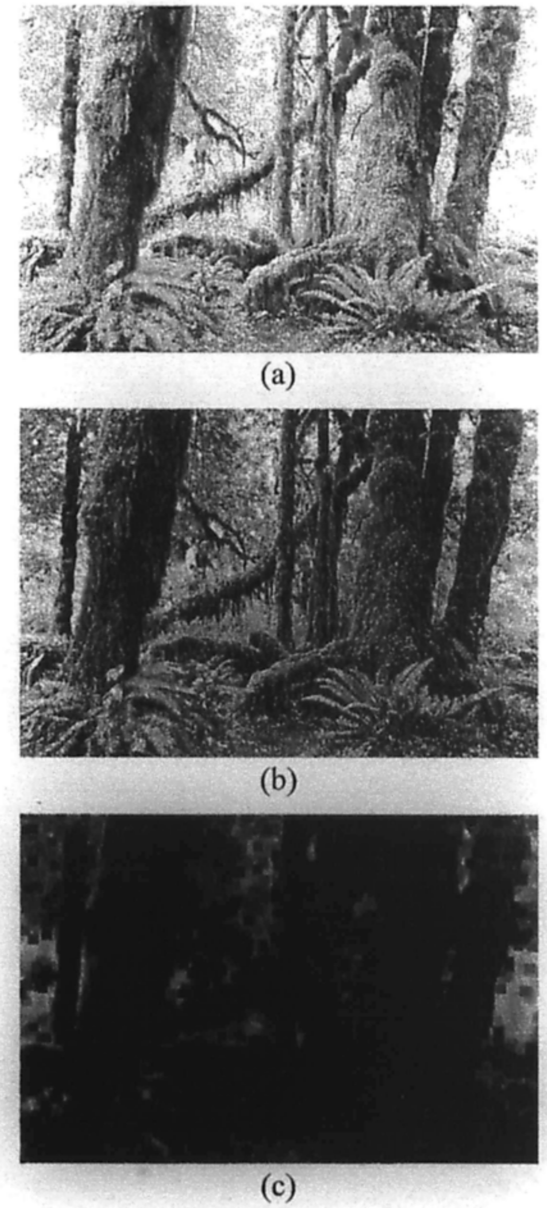


Figure 3.2: Computation of a dark channel. (a): An arbitrary image **J**. (b): For each pixel, we calculate the minimum of its (r, g, b) values. (c): A minimum filter is performed on (b). This is the dark channel of **J**. The image size is 800 × 551, and the patch size of Ω is 15 × 15.

Note each pixel in J^{dark} is a scalar. A dark channel is the outcome of two minimum operators. The operator "$\min_{c\in\{r,g,b\}}$" is performed on each pixel. See Fig. 3.2(b) for an example. The operator "$\min_{\mathbf{x}'\in\Omega(\mathbf{x})}$" is a *minimum filter* [27]. See Fig. 3.2(c).

It is worth mentioning that the dark channel is an operation on an image, no matter it is a haze-free or hazy one. We can also compute the dark channel $I^{\text{dark}}(\mathbf{x})$ of a hazy image $\mathbf{I}$ by:

$$I^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{x}'\in\Omega(\mathbf{x})}\big(\min_{c\in\{r,g,b\}} I^c(\mathbf{x}')\big). \tag{3.6}$$

We shall see the "dark channel" operation is very useful in haze removal.

Definition: Dark Channel Prior

Now we are ready to describe the mathematical formulation of the dark channel prior. Our observation says most patches should contain dark pixels. Formally, we have the following mathematical expression of the observation:

For outdoor haze-free images, most patches $\Omega(\mathbf{x})$ not covering the sky should satisfy:

$$\min_{c\in\{r,g,b\}}\big(\min_{\mathbf{x}'\in\Omega(\mathbf{x})} J^c(\mathbf{x}')\big) \approx 0, \tag{3.7}$$

or more simply:

$$J^{dark}(\mathbf{x}) \approx 0. \tag{3.8}$$

We call this observation *dark channel prior*.

The dark channel prior is the core idea of this thesis. It has a concise form as (3.8). It is a statistically and physically based law. Though it is very simple, we shall demonstrate it is very powerful in haze removal. We also discover a tight relationship between this prior and the human visual system, which is discussed in Chapter 5.

3.1.3 Experimental Verification

According to the prior, the dark channels of haze-free images should be mostly dark. Fig. 3.2 (c) shows an example. More visual examples are given in Fig. 3.3. Carefully inspecting these images, we can find out the sources of the dark pixels: shadows, colorful objects, and black objects. They exist almost everywhere in the images.

To verify how good the dark channel prior is, we should test a large collection of images and study their statistical properties. We download over 300,000 images from popular image search engines using 150 most popular tags annotated by the Flickr users. We are interested in outdoor landscape and cityscape scenes because they are the potential victims of haze. We only focus on daytime images. Among them, we manually label the haze-free ones. We randomly choose 5,000 images and cut out the sky regions. The images are resized so that the maximum of width and height is 500-pixel. Fig. 3.3 (left) shows several samples from the data set.

To compute the dark channels, we should set the patch size. As mentioned before, a larger patch has a better chance to contain a dark pixel. But we expect the prior to provide local constraints instead of more global ones. In this experiment, we set the patch as a rectangle of 15 × 15 pixels. We compute the dark channels of all the 5,000 images. Fig. 3.3 (right) shows some examples.

Fig. 3.4(a) is the distribution of the pixel intensity of all the 5,000 dark channels[1]. Each bin contains 16 intensity levels. We find that about 86% of the pixels fall in the first bin. Fig. 3.4(b) is the corresponding cumulative distribution. We can see that about 75% of the pixels in the dark channels have zero intensity, 90% of the pixels is below the intensity 25 (*i.e.*, about 0.1 in the [0, 1] gray scale). This statistic gives a very strong support to the dark channel prior.

We also concern whether some images have bright dark channels. We compute the average intensity of each dark channel. The distribution is shown in Fig. 3.4(c). Again, we find that most dark channels have very low average intensity, indicating that most images obey our prior.

[1] In this experiment, the color intensity of the input images (and the dark channels) is represented in [0, 255].

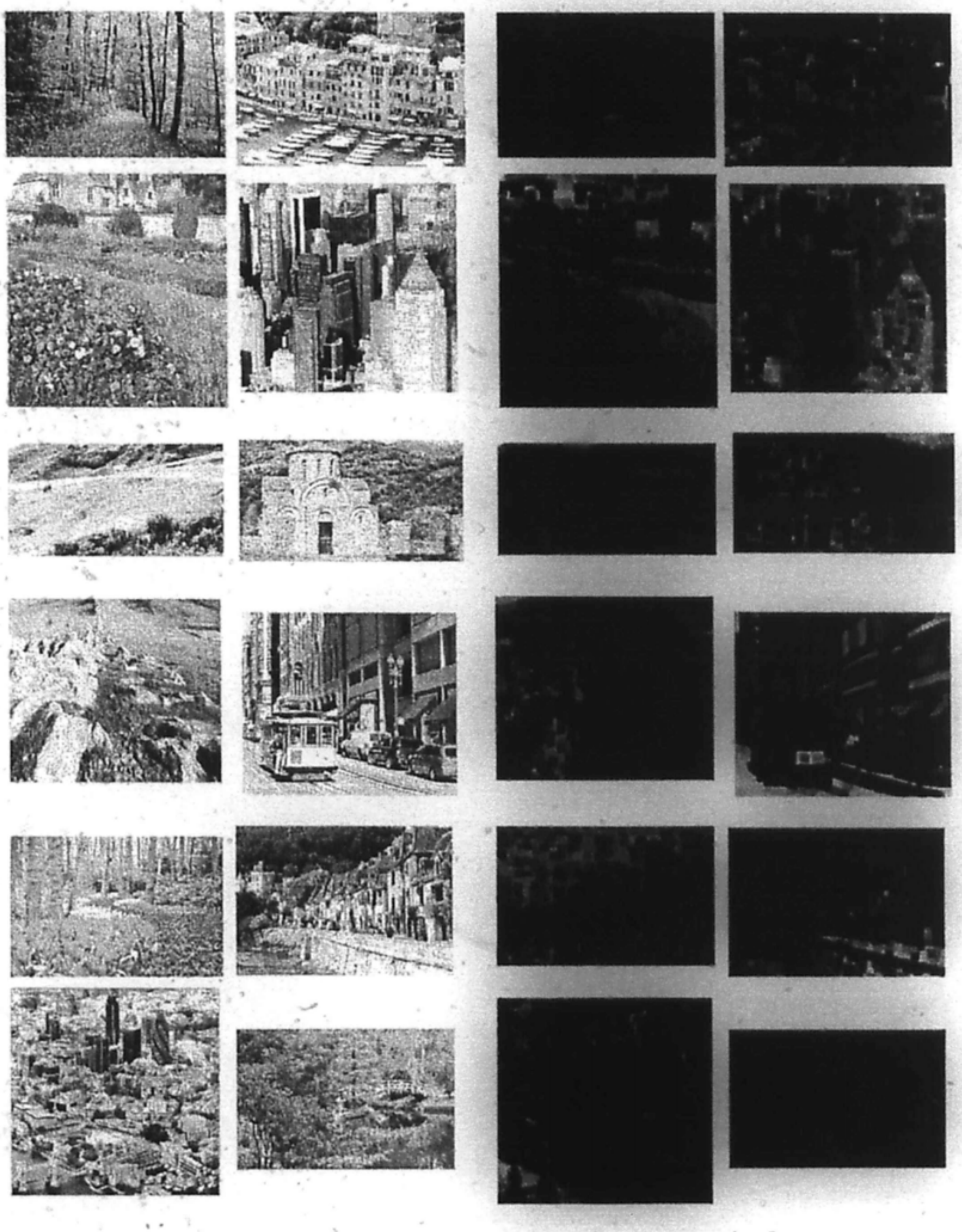
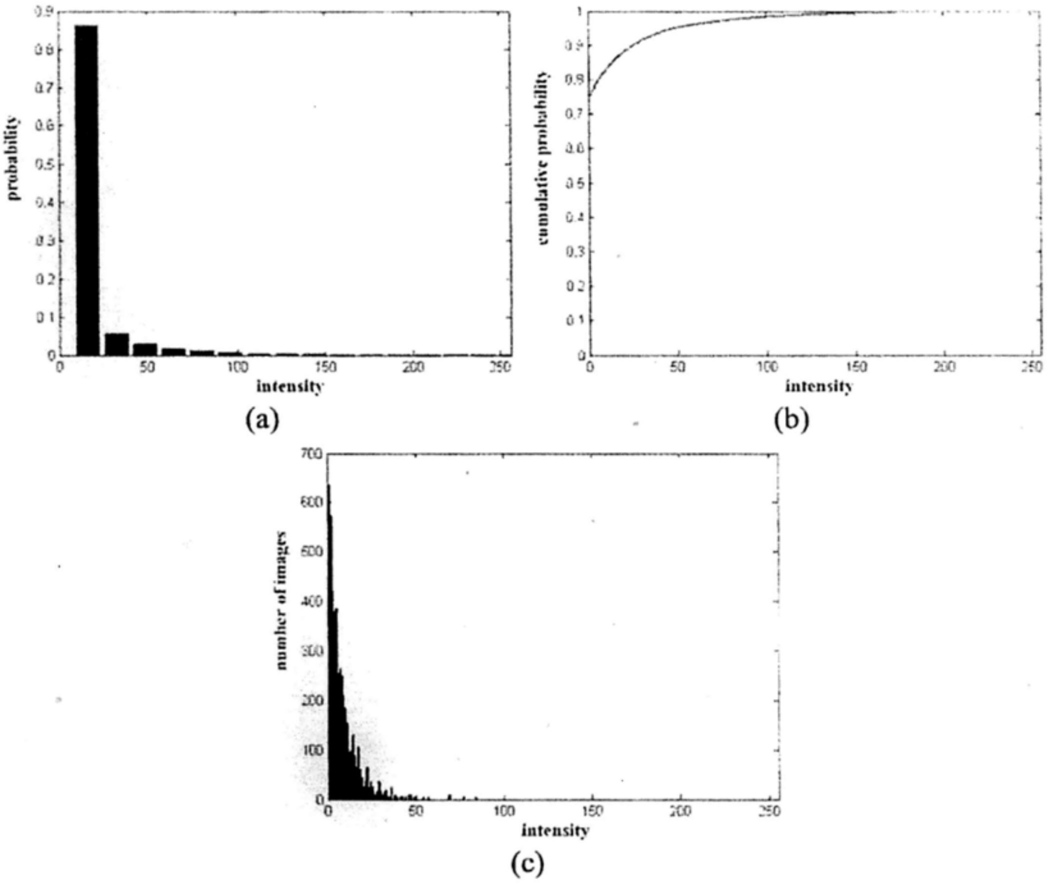


Figure 3.3: Dark channel prior. Left: outdoor haze-free images. Right: the corresponding dark channels. The dark channels are mostly dark. The images are resized so that the maximum of width and height is 500 pixels. Their dark channels are computed using a patch size 15 × 15.



(a)

(b)

(c)

Figure 3.4: Statistics of the dark channels. (a) Distribution of the pixel intensity of all of the 5,000 dark channels (each bin represents 16 intensity levels). (b) Cumulative distribution. (c) Distribution of the average intensity of each dark channel.

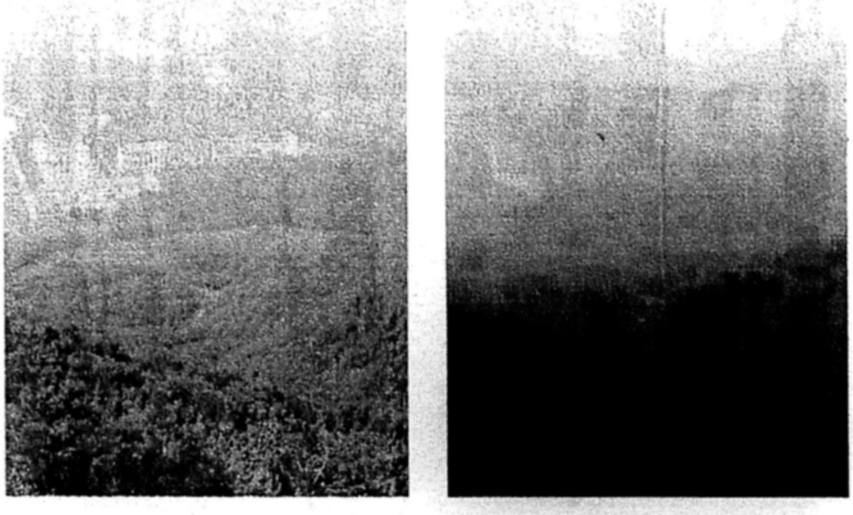


Figure 3.5: A hazy image and its dark channel.

3.2 A Novel Algorithm for Single Image Haze Removal

In this section, we propose a single image haze removal algorithm using the dark channel prior. We show that the dark channel prior introduces an extra constraint to each pixel, providing an estimated transmission value. Then we develop a method called "soft matting" to refine this transmission map. We further propose a robust method to estimate the atmospheric light $\mathbf{A}$, which is also based on the dark channel prior.

We have seen the dark channels of haze-free images. But what does the dark channel of a hazy image look like? Fig. 3.5 shows an example. We can see that the dark channel of a hazy image is not dark (though we still use the name "dark channel"). The reason is the additive airlight: the dark pixels are brightened when the airlight is added. From Fig. 3.5 we also see that the dark channel approximately tells the haze thickness: it is brighter in where the haze is thicker. We show that this effect can be explained by combining the haze imaging equation and the dark channel prior, and the transmission and atmospheric light are estimated accordingly.

3.2.1 Transmission Estimation

First we use the dark channel prior to estimate the transmission t. Recall the haze imaging equation (2.1):

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

Suppose the atmospheric light $\mathbf{A}$ has been estimated. We shall give an automatic method to estimate $\mathbf{A}$ in Section 3.2.3. We normalize the haze imaging equation (2.1) by $\mathbf{A}$:

$$\frac{I_c(\mathbf{x})}{A_c} = t(\mathbf{x})\frac{J_c(\mathbf{x})}{A_c} + 1 - t(\mathbf{x}) \tag{3.9}$$

Note that we normalize each color channel c independently. Then we compute the dark channel on both sides of this equation, *i.e.*, we insert the minimum operators:

$$\min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \frac{I_c(\mathbf{x}')}{A_c} = \min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \left(t(\mathbf{x}')\frac{J_c(\mathbf{x}')}{A_c} + 1 - t(\mathbf{x}') \right) \tag{3.10}$$

where we denote "$\min_{c\in\{r,g,b\}}$" by "$\min_c$" for simplicity.

We consider the transmission of a local patch $\Omega(\mathbf{x})$ as approximately uniform, and denote this transmission value as $\tilde{t}(\mathbf{x})$. The side effect of this approximation will be handled in Section 3.2.2. Thus we replace the term $t(\mathbf{x}')$ in (3.10) by $\tilde{t}(\mathbf{x})$. This allows us to move the transmission out of the minimum operators:

$$\min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \frac{I_c(\mathbf{x}')}{A_c} = \tilde{t}(\mathbf{x}) \min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \left(\frac{J_c(\mathbf{x}')}{A_c} \right) + 1 - \tilde{t}(\mathbf{x}). \tag{3.11}$$

The scene radiance $\mathbf{J}$ is a haze-free image. Due to the dark channel prior, we have:

$$J^{\mathrm{dark}}(\mathbf{x}) \equiv \min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c J_c(\mathbf{x}') \approx 0. \tag{3.12}$$

As A_c is always positive, this leads to:

$$\min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_{c} \frac{J_c(\mathbf{x}')}{A_c} \approx 0 \tag{3.13}$$

Putting (3.13) into (3.11), we can eliminate the term containing J and approximate the transmission by:

$$\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_{c} \frac{I_c(\mathbf{x}')}{A_c}. \tag{3.14}$$

In fact, the term "$\min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \frac{I_c(\mathbf{x}')}{A_c}$" is the dark channel of the normalized hazy image $\frac{I_c}{A_c}$. So (3.14) can be expressed in a more concise form:

$$\tilde{t}(\mathbf{x}) = 1 - \hat{I}^{\text{dark}}(\mathbf{x}), \tag{3.15}$$

where $\hat{\mathbf{I}}$ represents the normalized hazy image.

As we mentioned before, the dark channel prior does not concern the sky regions. Fortunately, the color of the sky in a hazy image $\mathbf{I}$ is usually very close to the atmospheric light $\mathbf{A}$ (because the depth $d \to \infty$ and the transmission $t \to 0$, see (2.2)). So in the sky region we have:

$$\min_{\mathbf{x}'\in\Omega(\mathbf{x})} (\min_{c} \frac{I_c(\mathbf{x}')}{A_c}) \to 1, \tag{3.16}$$

and equation (3.14) will estimate $\tilde{t}(\mathbf{x}) \to 0$, which is consistent with the true transmission (zero) of the sky. So (3.14) gracefully handles both sky and non-sky regions. We do not need to separate the sky beforehand.

Equation (3.14) is the core method of our single image haze removal algorithm. It provides an estimated transmission for each pixel, and thus solves the ambiguity of the problem. Its computation is very simple. Fig. 3.6 shows an example of the estimated transmission map by (3.14) and the recovered scene radiance $\mathbf{J}$. As we can see, this simple method is effective on recovering the vivid colors and unveiling low contrast objects. The transmission maps reasonably describe the thickness of the haze.

We can also see the artifacts from Fig. 3.6. They are mainly due to the locally-uniform-transmission assumption. This assumption is not true near the depth edges, *i.e.*, where the depth d (and t) changes abruptly. So halo artifacts appear near depth edges (*e.g.*, the white outlines of the leaves in Fig. 3.6). Besides, the transmission maps have block-like artifacts. To suppress these artifacts, we propose a soft matting method to refine the estimated transmission.

3.2.2 Soft Matting

The dark channel prior provides a constraint for each pixel, but we should also consider spatial continuities. Denote the refined transmission map by t. We hope to find a map t that: (i) approximates the transmission $\tilde{t}$ estimated by (3.14); (ii) has an edge when the depth is discontinuous; and (iii) is spatially smooth otherwise. The first condition is the pixel-wise constraint, and the other two are spatial continuity concerns. Since the depth is not known, we approximate the depth discontinuity by the sharp edges in the hazy image $\mathbf{I}$.

Combining pixel-wise constraints and spatial continuities is commonly desired in many computer vision/graphics applications, such as stereo vision [76], image denoising [21], surface interpolation [77], and alpha matting [42]. A popular solution to these problems is the Markov Random Fields (MRF) model.

We adopt the following MRF model in our algorithm.

$$E(t) = \lambda \sum_{\mathbf{x}} \|t(\mathbf{x}) - \tilde{t}(\mathbf{x})\|_2^2 + \sum_{\mathbf{x}} \sum_{\mathbf{x}' \in \mathrm{N}(\mathbf{x})} w(\mathbf{I}, \mathbf{x}', \mathbf{x}) \|t(\mathbf{x}) - t(\mathbf{x}')\|_2^2 \quad (3.17)$$

In this equation, the first term is a data term with a weight λ. It describes the error between $\mathbf{t}$ and $\tilde{\mathbf{t}}$. We set a small λ (10^{-4}) in the experiments, so $\mathbf{t}$ is softly constrained by $\tilde{\mathbf{t}}$. The second term is the smoothness term where $\mathrm{N}(\mathbf{x})$ is a small neighborhood around $\mathbf{x}$. The weight $w(\mathbf{I}, \mathbf{x}', \mathbf{x})$ imposes continuity conditions on $\mathbf{t}$: a large weight suggests smoothness, and a small weight suggests an edge. The smoothness is adjusted according to $\mathbf{I}$.

The optimization is better written in matrix forms. We reorder the pixels in t and $\tilde{t}$ to form column vectors $\mathbf{t}$ and $\tilde{\mathbf{t}}$ respectively[2]. The quadratic cost

[2] For example, the pixel with the coordinates (x, y) corresponds to the $(x + y * wid)$th entry in the vector, where wid is the image width.

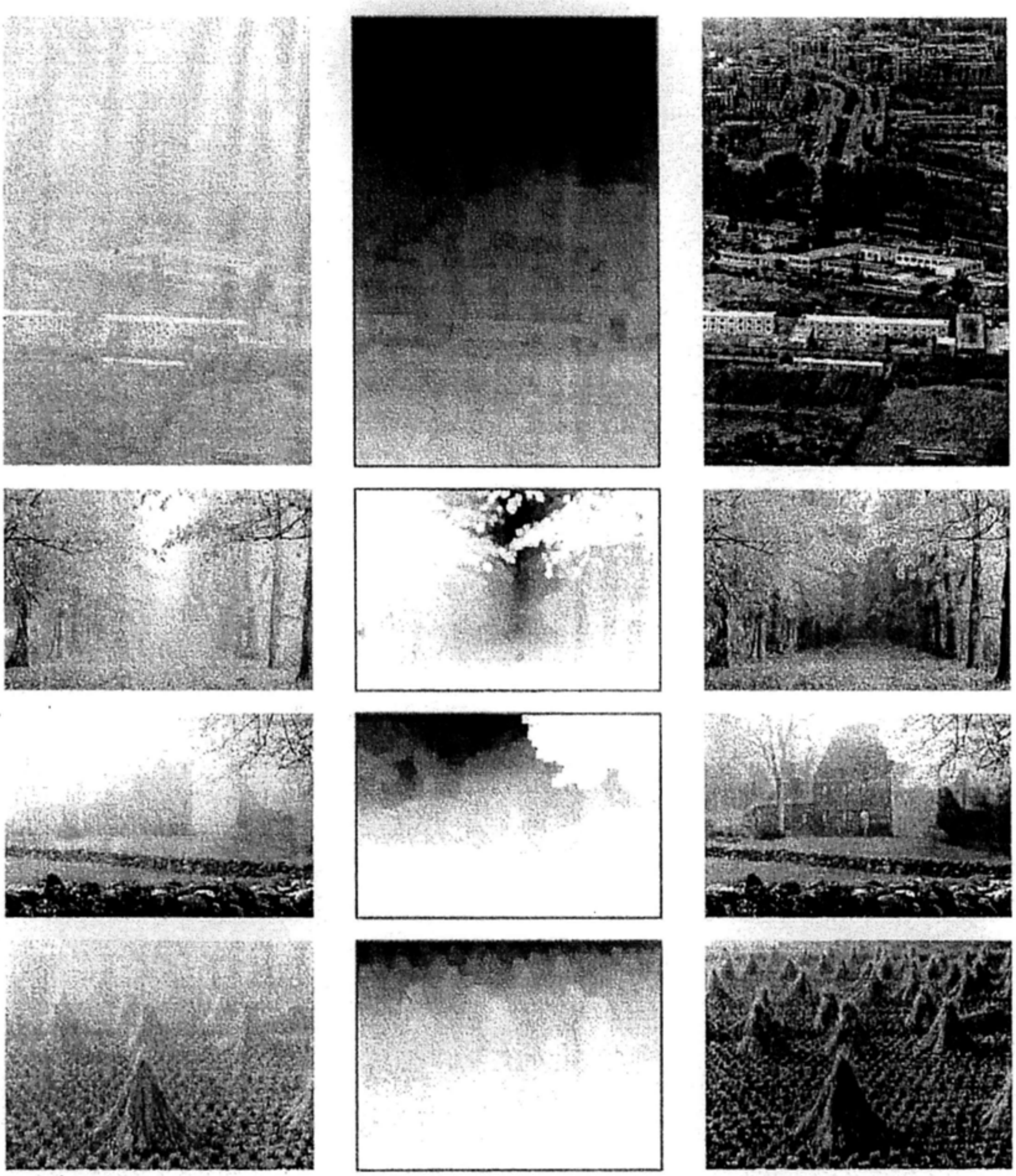


Figure 3.6: Haze removal using Equation (3.14). Left: input hazy images. Middle: transmission maps estimated by (3.14). Right: recovered scene radiance by these transmission maps.

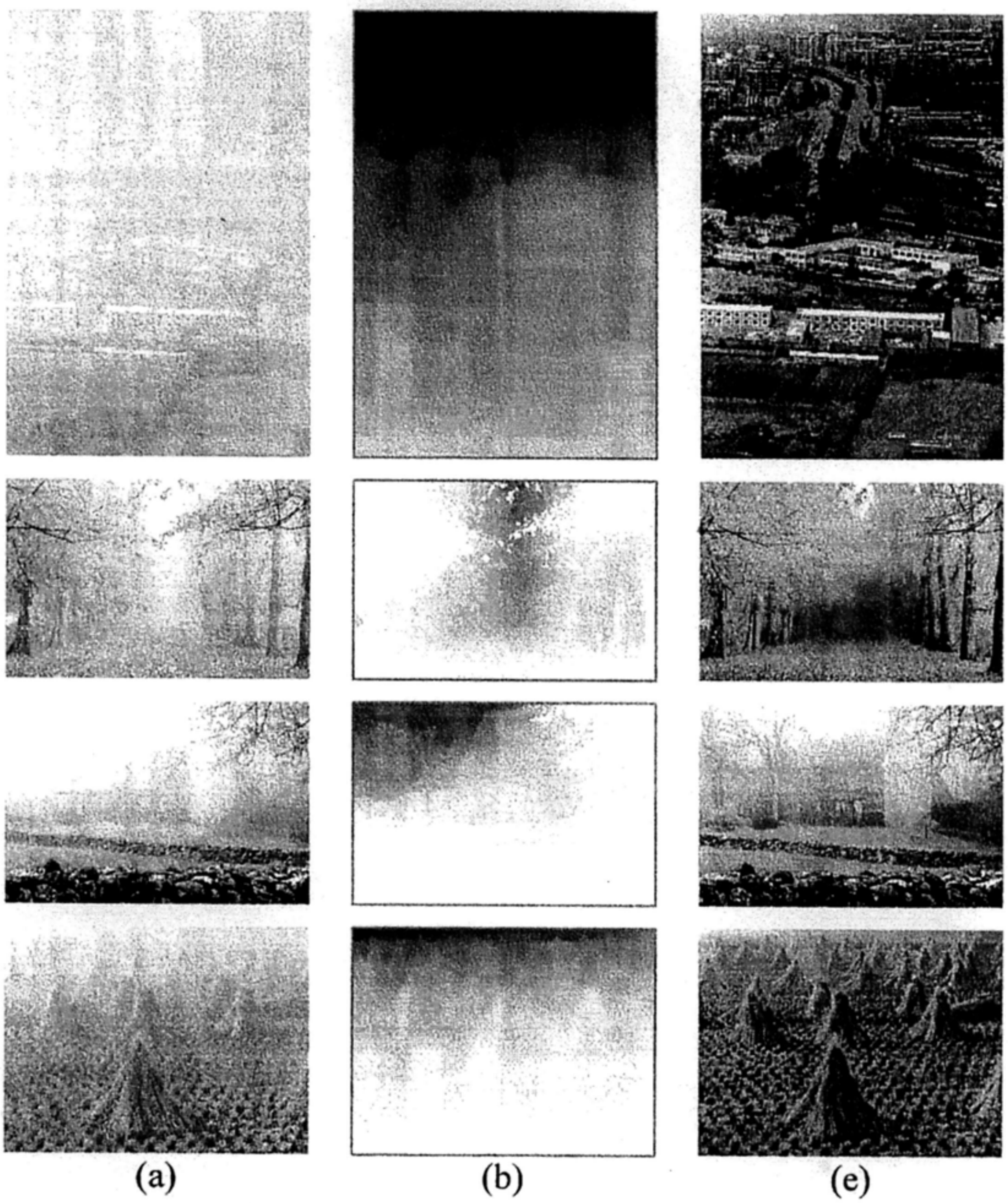


Figure 3.7: Soft matting results of Fig 3.6. (a): Input hazy images. (b): Refined transmission t after soft matting. (c) Recovered scene radiance images.

function (3.17) becomes:

$$E(\mathbf{t}) = \lambda\|\mathbf{t} - \tilde{\mathbf{t}}\|_2^2 + \mathbf{t}^{\mathrm{T}}\mathrm{L}\mathbf{t}, \tag{3.18}$$

where L is an $N \times N$ Laplacian matrix[3] containing the weights.

We adopt the matting Laplacian matrix [42] previously designed for alpha matting. We choose this matrix for two reasons. First, the haze imaging equation (2.1) is analogous to the alpha matting equation:

$$\mathbf{I}(\mathbf{x}) = \mathbf{F}(\mathbf{x})\alpha(\mathbf{x}) + \mathbf{B}(\mathbf{x})(1 - \alpha(\mathbf{x})), \tag{3.19}$$

where $\mathbf{F}$ is the foreground, $\mathbf{B}$ is the background, and α is the foreground transparency [70]. The map α plays a role like t in the haze equation. The matting Laplacian matrix has been proven successful in recovering the profile of α (*e.g.*, in [42, 84]). Second, the matting Laplacian matrix is derived from a local linear model. We can show that the haze imaging equation also leads to this model. Thus, the matting Laplacian matrix is a proper choice in haze removal. We discuss more details in Chapter 4.

The (i,j) element of the matting Laplacian matrix is given by [42]:

$$\sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + (\mathbf{I}_i - \mu_k)^T(\Sigma_k + \frac{\varepsilon}{|w_k|}\mathrm{U}_3)^{-1}(\mathbf{I}_j - \mu_k))), \tag{3.20}$$

Here, the subscript i, j, and k are the pixel indexes in a form like $(x+y*wid)$. $\mathbf{I}_i$ and $\mathbf{I}_j$ are the colors of the input image $\mathbf{I}$ at pixels i and j, δ_{ij} is the Kronecker delta, μ_k and Σ_k are the mean and covariance matrix of the colors in window w_k, U_3 is a 3×3 identity matrix, ε is a regularizing parameter, and $|w_k|$ is the number of pixels in the window w_k. Having defined the matrix L, the optimal $\mathbf{t}$ in (3.18) is obtained by solving the following sparse linear system:

$$(\mathrm{L} + \lambda\mathrm{U})\mathbf{t} = \lambda\tilde{\mathbf{t}}, \tag{3.21}$$

where U is an identity matrix of the same size as L. The linear system can be solved by standard linear solvers like Conjugate Gradient [63].

The matting Laplacian matrix has also been applied in [36] to deal with

[3] A Laplacian matrix is a matrix whose elements in each row sum to zero.

the spatially variant white balance problem. In [36] and in alpha matting [42], the data term are known in some sparse regions, and the MRF is used to extrapolate the values into the unknown regions. So the data constraints are strict or fixed. In our application, the data term have already filled the whole image, and the MRF is used to smooth and refine them. Our data constraints are soft and loose. So we call our method *soft* matting.

Figure 3.7 shows the soft matting results of Figure 3.6. As we can see, the halos and block artifacts are suppressed. The refined transmission maps manage to capture the sharp edge discontinuities and outline the profile of the objects. Besides, the transmission map is smooth in where no depth edge exists.

In Chapter 4 we discuss more on the matting Laplacian matrix and soft matting. We propose fast algorithms, alternatives, and generalizations.

3.2.3 Atmospheric Light Estimation

We have been assuming that the atmospheric light $\mathbf{A}$ is known. Next we propose a method to estimate $\mathbf{A}$. In previous works (*e.g.* [79, 18]), the color of the most haze-opaque (smallest t) regions is considered as $\mathbf{A}$. However, the detection of the "most haze-opaque" regions is not trivial, because the estimation of t is often after the estimation of $\mathbf{A}$. So we cannot find such regions by the criterion of "smallest t". Some methods (*e.g.* [65]) require the user to mark such regions. But in most applications automatic methods are required.

In Tan's work [79], the brightest pixels in the hazy image are considered to be the most haze-opaque. This is true only when the atmospheric light is the sole illumination source of the scene (see Fig. 3.8 top). Denote the reflectance of the scene point by ρ. The scene radiance of each color channel is given by:

$$J_c(\mathbf{x}) = \rho_c(\mathbf{x})A_c. \tag{3.22}$$

where $0 \leq \rho_c \leq 1$. The haze imaging equation (2.1) can be written as:

$$I_c(\mathbf{x}) = \rho_c(\mathbf{x})A_c t(\mathbf{x}) + (1 - t(\mathbf{x}))A_c. \tag{3.23}$$

Since $\rho_c(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x})) \leq 1$, we have:

$$I_c(\mathbf{x}) \leq A_c. \tag{3.24}$$

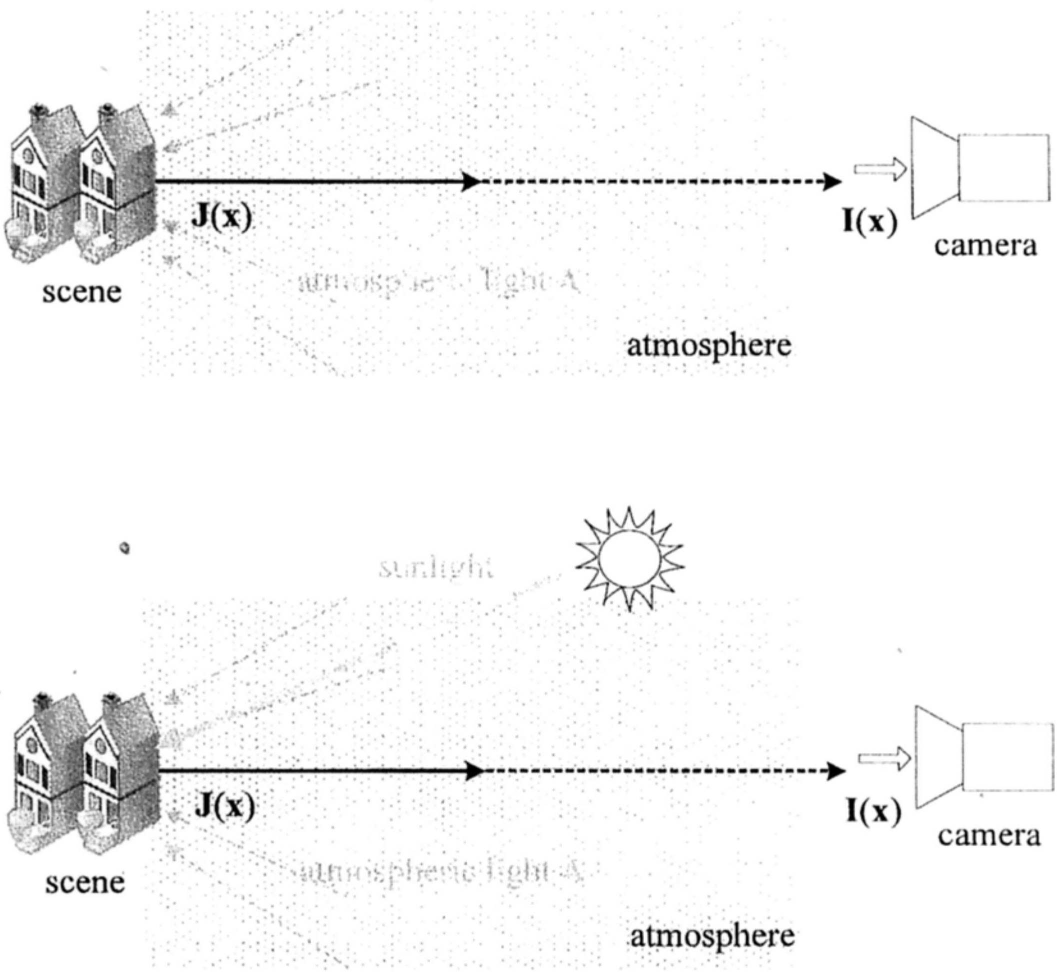


Figure 3.8: Illumination sources in hazy days. Top: The atmospheric light is the sole illumination source. Bottom: the sunlight is another illumination source.

for any pixel in the image. The brightest I_c is the closest to A_c. If pixels at infinite distance ($d \rightarrow \infty$ and $t \approx 0$) exist in the image, the brightest I_c equals to the atmospheric light A_c.

Unfortunately, the atmospheric light is rarely the sole illumination source. If the weather is not cloudy or overcast, the sunlight may go through the atmosphere and illuminate the scene objects (see Fig. 3.8 bottom). The light reflected or radiated by the clouds can also be another illumination source in hazy weather. Denote the sunlight (or the light from clouds) by $\mathbf{S}$. We modify (3.22) by:

$$J_c(\mathbf{x}) = \rho_c(\mathbf{x})(S_c + A_c), \tag{3.25}$$

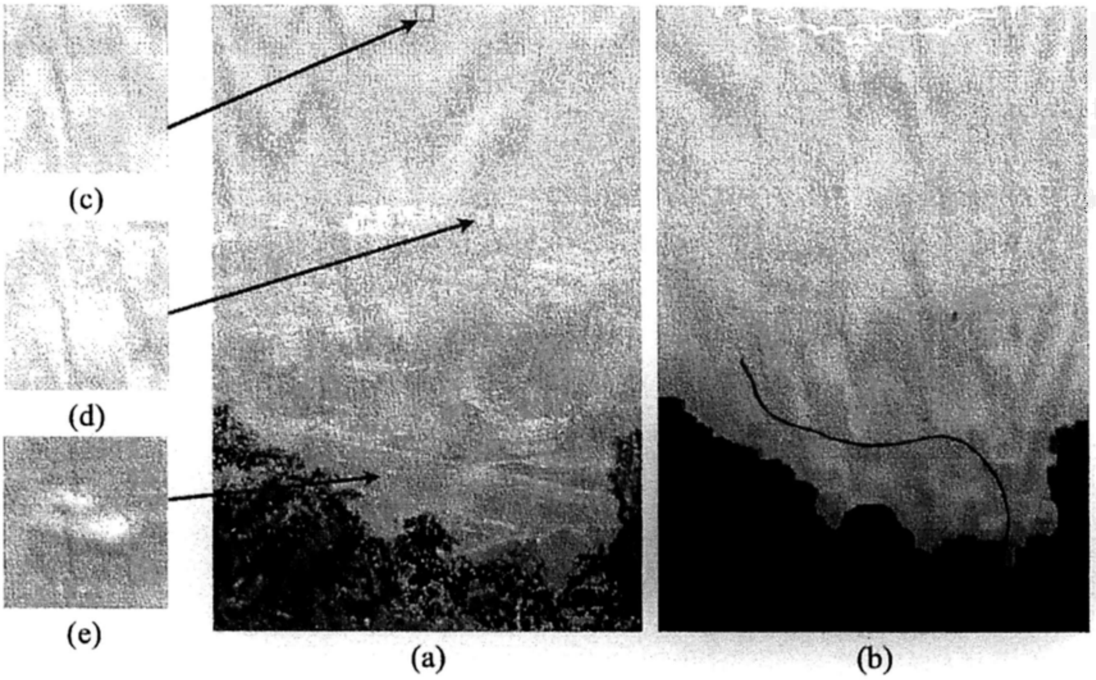


Figure 3.9: Estimation of the atmospheric light **A**. (a) Input hazy image. (b) Dark channel and the most haze-opaque region (bounded by the yellow lines). (c) The patch from where our method automatically obtains the atmospheric light. (d) and (e): Two patches that contain pixels brighter than the atmospheric light.

and (3.23) by:

$$I_c(\mathbf{x}) = \rho_c(\mathbf{x})S_c t(\mathbf{x}) + \rho_c(\mathbf{x})A_c t(\mathbf{x}) + (1 - t(\mathbf{x}))A_c. \tag{3.26}$$

The inequality $I_c(\mathbf{x}) \leq A_c$ no longer holds. The brightest $I_c(\mathbf{x})$ can be brighter than the atmospheric light A_c. It can be on a white car or a white building (Figure 3.9(d)(e)).

We develop a more robust method to estimate **A** based on the dark channel prior. We notice that in (3.26), the sunlight term $\rho_c(\mathbf{x})S_c t(\mathbf{x})$ is more negligible when $t(\mathbf{x})$ is smaller. In the most hazy-opaque region (smallest t), the impact of the sunlight is the smallest. Recall our transmission estimation (3.14):

$$\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \min_{c} \frac{I_c(\mathbf{x}')}{A_c}. \tag{3.14}$$

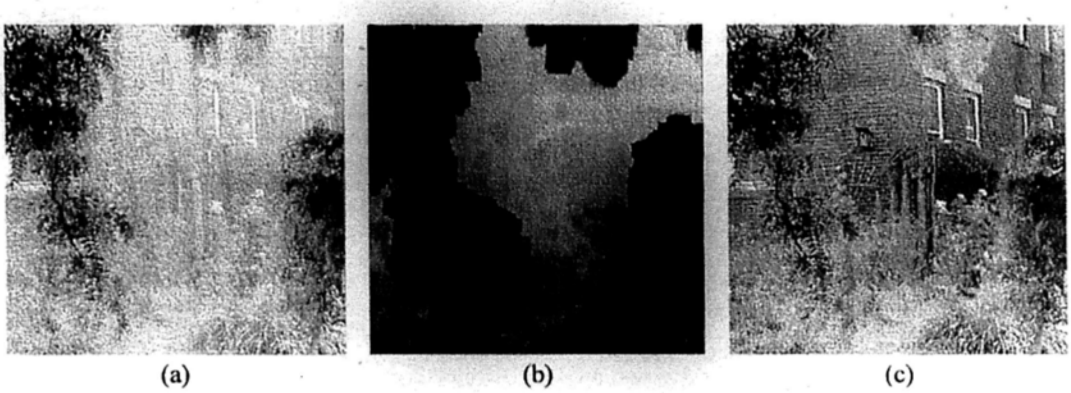


Figure 3.10: Estimation of the atmospheric light when there is no infinitely distant pixel. (a): Input image. (b): Dark channel. The red pixels on the top right corner are the most haze-opaque regions detected by our method. (c): Our haze removal result.

Assume the color of the atmosphere is grayish, *i.e.*, the three components of $\mathbf{A}$ are identical: $A_r = A_g = A_b \equiv A$. Then (3.14) becomes:

$$\begin{aligned}\tilde{t}(\mathbf{x}) &= 1 - \frac{1}{A}\min_{\mathbf{x}'\in\Omega(\mathbf{x})}\min_{c} I_c(\mathbf{x}') \\ &= 1 - \frac{1}{A} I^{\text{dark}}, \end{aligned} \tag{3.27}$$

where the dark channel I^{dark} is given in (3.6). This equation indicates the most haze-opaque region (smallest $\tilde{t}$) corresponds to the brightest values in I^{dark} (Figure 3.9(b)). This claim is true even if we do not know the value A in (3.27).

So we can use the dark channel to detect the most haze-opaque region and estimate the atmospheric light. We first pick the top 0.1% brightest pixels in the dark channel. These pixels are most haze-opaque in the image (bounded by yellow lines in Figure 3.9(b)). Among them, the pixels with highest intensity[4] in the *input* image I_c are selected as the atmospheric light A_c[5]. These pixels are in the red rectangle in Figure 3.9(a). Note that these pixels may not be the brightest in the whole image.

Theoretically, we can use the estimated $\mathbf{A}$ to fix the grayish-$\mathbf{A}$ assumption and improve its accuracy. But in experiments we find the first estimation

[4] We choose the top 1% highest intensity to avoid noise.

[5] This operation is on each channel independently.

is good enough even if **A** is colored (*e.g.*, blueish or reddish). So this improvement is not necessary.

This method works well even when the image does not contain pixels at infinite distance. It is usually sufficient if only the image contains a densely hazy region, where the sunlight term can be ignored. In Fig. 3.10(b), our method detects the most hazy region which is not infinitely distant. The estimated **A** is already a good approximation in our haze removal algorithm. The haze removal result is shown in Fig. 3.10(c).

This simple method based on the dark channel prior is more robust than the brightest-pixel method. We use it to automatically estimate the atmospheric lights for all images in this thesis.

3.2.4 Scene Radiance Recovery

With the atmospheric light **A** and the transmission t, we can recover the scene radiance **J** by inverting haze imaging equation:

$$J_c(\mathbf{x}) = \frac{I_c(\mathbf{x}) - A_c}{t(\mathbf{x})} + A_c. \tag{3.28}$$

However, the direct attenuation term $\mathbf{J}(\mathbf{x})t(\mathbf{x})$ in the haze imaging equation (2.1) is very close to zero when $t(\mathbf{x})$ is very small. The recovered scene radiance **J** from (3.28) is prone to noise. Therefore, we restrict the transmission $t(\mathbf{x})$ by a lower bound t_0, *i.e.*, we preserve a small amount of haze in very dense haze regions. The final scene radiance $\mathbf{J}(\mathbf{x})$ is recovered by:

$$J_c(\mathbf{x}) = \frac{I_c(\mathbf{x}) - A_c}{\max(t(\mathbf{x}), t_0)} + A_c. \tag{3.29}$$

A typical value of t_0 is 0.1.

Since the scene radiance is usually not as bright as the atmospheric light, removing the haze from an image may reduce the whole intensity. The image after haze removal will look dim. So we increase the intensity of **J** for display. The image **J** is multiply by a ratio C so that the average intensity of **J** equals to the average intensity of **I**. Some final recovered images are shown in Fig. 3.7.

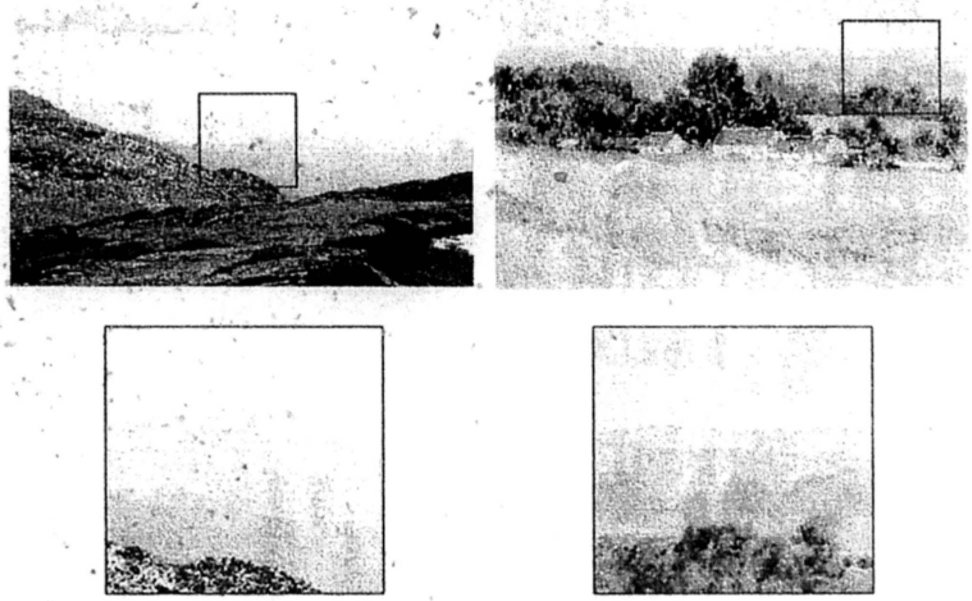


Figure 3.11: Aerial perspective: the atmospheric effects are important clues for depth. Left: a photograph from Wiki by J. A. Gaspar. Right: the painting *Haystack at Giverny* by Claude Monet (1840-1926).

Aerial Perspective

Haze is not always unwanted. They provide clues for human beings to perceive the distance. This is known as *aerial perspective* or *atmospheric perspective* [11]. See Fig. 3.11 (left) for an example. The texture and details of the mountains are hard to see. But the farther objects appear more whitish, so we can still tell their relative distance. The aerial perspective has been noticed by artists and used to represent depth in their works (*e.g.*, see Monet's painting in Fig. 3.11 (right)).

In Fig. 3.11 we also notice that haze does not always reduce contrast: they may increase contrast near the depth edges. In fact, we can compute the gradient on both size of the haze image equation (2.1):

$$\nabla \mathbf{I}(\mathbf{x}) = t(\mathbf{x})\nabla \mathbf{J}(\mathbf{x}) + (\mathbf{J}(\mathbf{x}) - \mathbf{A})\nabla t(\mathbf{x}). \tag{3.30}$$

The gradient of **I** can be mostly contributed by $\nabla t(\mathbf{x})$ near the sharp depth edges.

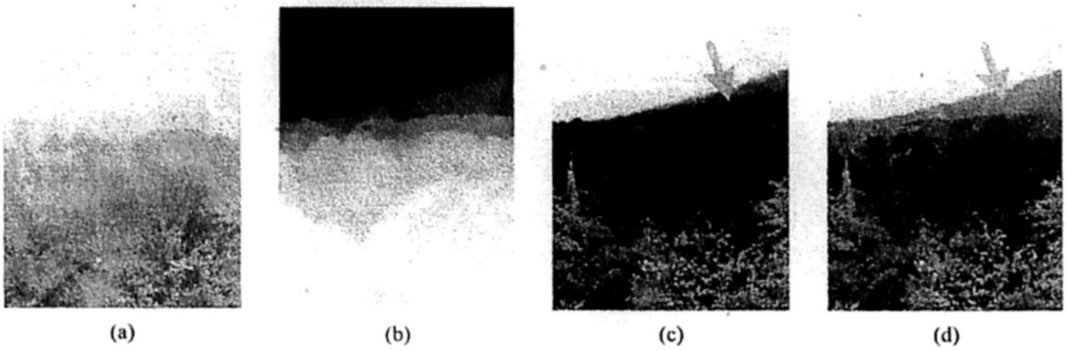


Figure 3.12: Haze removal with aerial perspective. (a) Input hazy image. (b) Recovered transmission. (c) Haze removal result without aerial perspective ($\kappa = 1$). (d) Haze removal result with aerial perspective ($\kappa = 0.95$). The depth is more perceivable because the remained haze.

If we remove the haze thoroughly, the feeling of depth may be greatly weakened and the image may appear unnatural. See Fig. 3.12(c) for an example: the mountains are hard to discern. To handle this problem, we can optionally keep a very small amount of haze for the distant objects by introducing a constant parameter κ ($0 < \kappa < 1$). After obtaining the transmission t, we modify its value by:

$$t(\mathbf{x}) := 1 - \kappa(1 - t(\mathbf{x})). \tag{3.31}$$

Here we use ":=" to denote assignment. The nice property of this modification is that we adaptively keep more haze for the distant objects. In haze-free regions, t is 1 on the right hand side so it is not changed by κ; in completely hazy regions, t is 0 on the right hand side so it will become $1 - \kappa > 0$, and thus a small amount of haze would remain in the recovered image. The value of κ is application-based. We set it to 0.95 in this thesis. We use this modified t in (3.29) to recover the scene radiance. In Fig. 3.12(d) shows an example. The depth is more perceivable because of the remained haze.

3.2.5 Implementation

Our single image haze removal algorithm is summarized in Algorithm 1.

In Step 1 and 2 the dark channel computation involves the minimum filter "$\min_{\mathbf{x}'\in\Omega(\mathbf{x})}$". A brute-force algorithm takes $O(r^2N)$ time where r is the patch Ω's radius. Instead, we adopt van Herk's fast algorithm [83] which is $O(N)$ time. The computational time is independent of the patch size and very fast ($<$ 10ms per megapixel).

In Step 3, the linear system is solved by Conjugate Gradient (CG) or Preconditioned Conjugate Gradient (PCG) [63]. This is the sole time-consuming step in the whole algorithm. The time complexity is $O(N^{\frac{3}{2}})$ in the number of pixels N. For a 600×400, it takes over 10s to solve the linear system. We shall propose fast algorithms in the Chapter 4.

Algorithm 1 A single image haze removal algorithm based on dark channel prior

```
1: Estimate the atmospheric light $\mathbf{A}$ as in Sec. 3.2.3.
2: Estimate the transmission by (3.14):
       $\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{x}'\in\Omega(\mathbf{x})} \min_c \frac{I_c(\mathbf{x}')}{A_c}.  \qquad (3.14)$
3: Refine the tranmssion by soft matting, i.e., solve the linear system (3.21) for $\mathbf{t}$:
       $(\mathrm{L} + \lambda\mathrm{U})\mathbf{t} = \lambda\tilde{\mathbf{t}}.  \qquad (3.21)$
4: Recover the scene radiance $\mathbf{J}$ by:
       $t(\mathbf{x}) := 1 - \kappa(1 - t(\mathbf{x})).  \qquad (3.32)$
       $J_c(\mathbf{x}) = \frac{I_c(\mathbf{x}) - A_c}{\max(t(\mathbf{x}), t_0)} + A_c.  \qquad (3.29)$
```

3.2.6 Relation to Previous Methods

The dark channel prior is partially inspired by the dark object subtraction [9]. Both methods concern dark intensity. However, the dark object subtraction is only valid for constant transmission. This method finds one dark object in the whole image. On the contrary, our method is valid for spatially variant transmission, and it is based on the observation that the dark pixels may appear everywhere. Besides, the dark object subtraction requires dark objects in each color channel, whereas our method only requires a pixel to be dark in only one channel.

Both our method and Tan's visibility maximization method [79] assume locally uniform transmission and search for this value. In Tan's method the visibility is maximized only when a portion of pixels have intensity < 0 or $> A$, which are then truncated. This operator is not physically based. This method often overestimates the thickness of the haze, and many pixels are actually brought below 0. On the contrary, our dark channel prior ensures that the pixel intensity is not below zero. Indeed, our method set the dark channel (the minimum intensity) to zero, so it will not violate the physical constraints.

Fattal's ICA method [18] assume a statistical independency of the scene luminance and the haze transmission. Our dark channel prior, at a broader view, indicates that the dark channel of the scene radiance is independent of the haze, because it is mostly a zero constant.

Multiplicative vs. Additive

In the haze image equation (2.1), the direct transmission $\mathbf{J}(\mathbf{x})t(\mathbf{x})$ is a *multiplicative* degradation of the scene radiance $\mathbf{J}$. It accounts for visibility/contrast reduction. The airlight $\mathbf{A}(1 - t(\mathbf{x}))$ is an *additive* degradation with respect to the scene radiance. It leads chrominance shift, making the scene whitish or grayish. The existing two single image methods (Tan's [79] and Fattal's [18]) solve the ambiguity from the multiplicative term. They are driven by the observations that the multiplicative term changes the image visibility [79] or the color variance [18].

On the contrary, our method is based on the additive term. It based on the fact that the dark channel is brightened by the additive airlight. In the derivation of the transmission estimation (3.14), the dark channel prior eliminates the multiplicative term (see Equations (3.11) to (3.14)) and leaves the additive term. We can see this fact more clearly if we generalize the haze image equation (2.1) by:

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t_1(\mathbf{x}) + \mathbf{A}(1 - t_2(\mathbf{x})), \tag{3.33}$$

where t_1 does not necessarily equal to t_2. Using the dark channel prior to eliminate the multiplicative term we can obtain:

$$\tilde{t}_2(\mathbf{x}) = 1 - \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \min_c \frac{I_c(\mathbf{x}')}{A_c}. \tag{3.34}$$

Thus, we can separate the additive term $\mathbf{A}(1 - t_2(\mathbf{x}))$ with the multiplicative term $\mathbf{J}(\mathbf{x})t_1(\mathbf{x})$. If $t_1 \neq t_2$, we need other priors to compute t_1, depending on the problem formulation. In the literature of human vision research [26], the additive term is called a *veiling luminance*, and (3.33) can model semi-transparent veils, glare from strong light sources, or specular reflection highlights. Our method has potential in solving this problem.

3.3 Experimental Results

In this section, we demonstrate the results of our method and compare with various previous works.

3.3.1 Patch Size

A key parameter in our algorithm is the patch size in the transmission estimation (3.14). On one hand, the dark channel prior becomes better for a larger patch size, because the probability that a patch contains a dark pixel is increased. We can see this in Fig. 3.13: the larger the patch size is, the darker a dark channel is. Consequently, (3.14) is more accurate for a larger patch. A patch that is too small will lead to over-saturated colors in the haze removal results (Fig. 3.14(b)). On the other hand, the assumption that the transmission is uniform in a patch becomes less appropriate if the patch size is getting larger. Halos near depth edges may become stronger (Fig. 3.14(c)).

Fig. 3.15 shows the haze removal results using different patch sizes. The image sizes are 600 × 400. In Fig. 3.15(b), the patch size is 3 × 3. The colors of some grayish surfaces look over-saturated (see the buildings in the first row, and the rectangles in the second and the third rows). In Fig. 3.15(c) and (d), the patch sizes are 15 × 15 and 30 × 30 respectively. The results appear more natural than those in (b). This shows that our method works well for sufficiently large patch sizes: the soft matting is able to reduce the artifacts introduced by large patches. In the remaining part of this thesis,

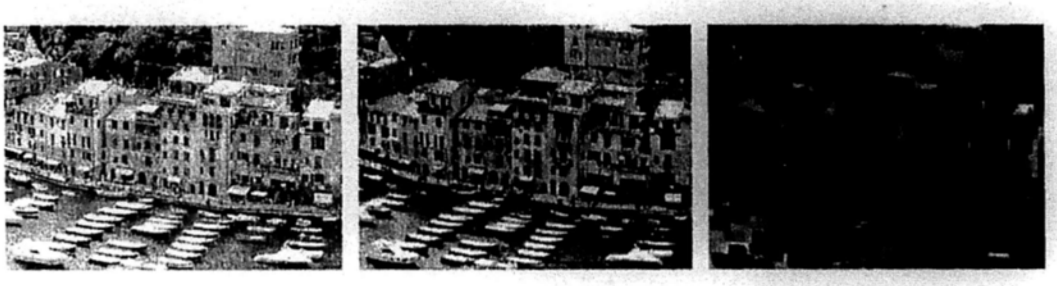


Figure 3.13: A haze-free image (600 × 400) and its dark channels using 3 × 3 and 15 × 15 patches respectively.

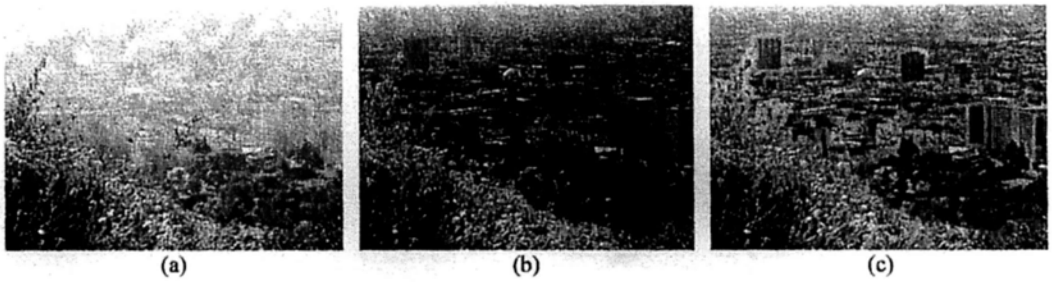


Figure 3.14: A 600 × 400 hazy image (a) and the recovered scene radiance using 3 × 3 (b) and 15 × 15 (c) patches respectively (without soft matting). The recovered scene radiance is over-saturated for a small patch size, while contains apparent halos for a large patch size.

we use a patch size of 15 × 15 for 600 × 400 images, and scale the patch size according to the image size.

3.3.2 Results of Our Method

We have shown some results in Fig. 3.7 and Fig. 3.15. We show more cityscape examples in Fig. 3.16 and Fig. 3.17, landscape examples in Fig. 3.18, and miscellaneous cases in Fig. 3.19. The atmospheric light estimated by our algorithm is indicated by the red rectangles in these figures. As we can see, our method can recover the details and vivid colors even in heavily hazy weather, and is robust in various cases. Our approach also works for gray-scale images if there are enough shadows. Cityscape images usually satisfy this condition. Fig. 3.20 shows an example.

We also show the resulting depth maps in Fig. 3.16 to 3.20. The depth

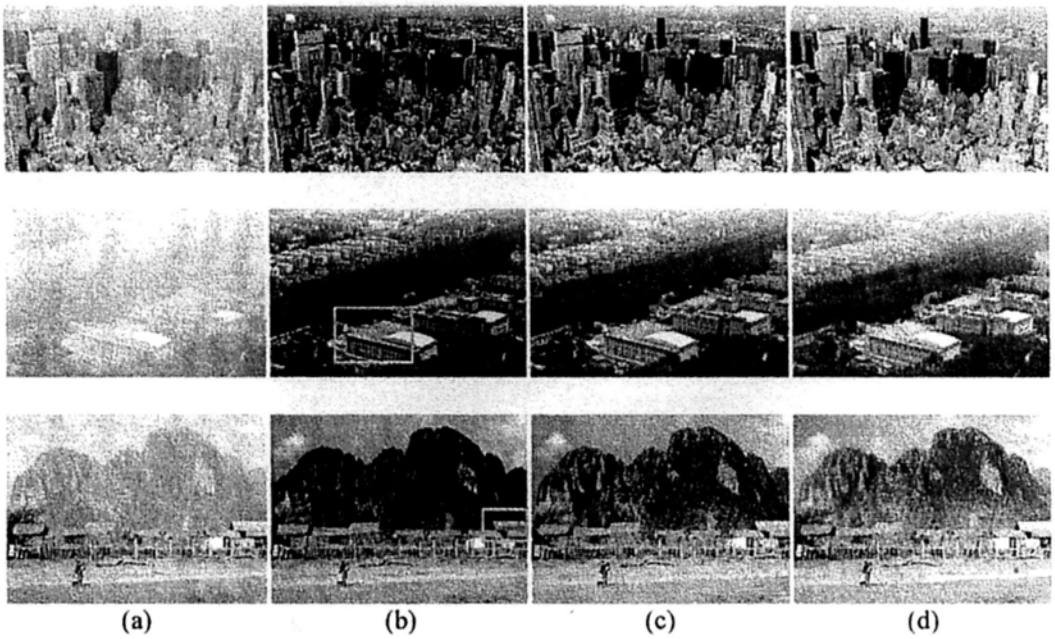


Figure 3.15: Recovering images using different patch sizes (after soft matting). (a): Input hazy images. (b): Using 3 × 3 patches. (c): Using 15 × 15 patches. (d): Using 30 × 30 patches.

maps are computed according to (2.4):

$$d(\mathbf{x}) = -\frac{\ln t(\mathbf{x})}{\beta}. \qquad (2.4)$$

Here we show relative depth where the scale $(-\frac{1}{\beta})$ is unknown[6]. We can see that our depth maps are visually reasonable: they have consistent edges with the input and are smooth otherwise. Notice that estimating depth from a single image is a very challenging problem in computer vision [64]. But it becomes much easier with the "help" of the haze. We can use the depth map to defocus the images (see Fig. 3.21 and Fig. 3.22). This technique can be used to simulate miniature scenes (Fig. 3.22). These interesting results are generated from "annoying" hazy images.

[6] The depth maps are shown by pseudo-color mapping.

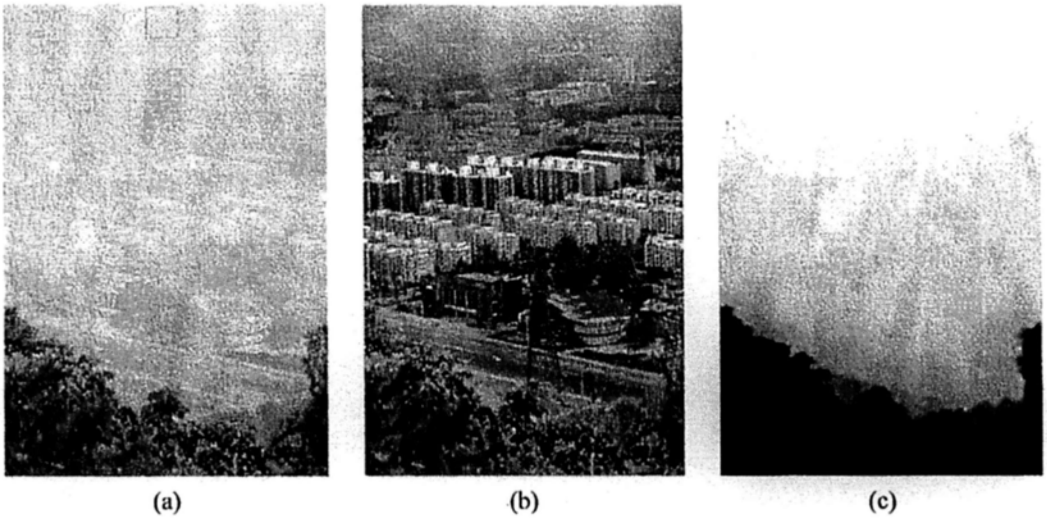


Figure 3.16: Haze removal from a single image. (a) Input hazy image. (b) Our haze removal result. (c) Our recovered depth map.

3.3.3 Comparisons with Previous Methods

Next we compare with all the state-of-the-art methods in previous works. We show that our method outperforms them in various situations.

In Fig. 3.23 we compare with the dichromatic methods [52, 48, 50] which requires two images taken in different atmospheric conditions. Our results is merely from the bottom left image. Though our method uses only one image, the recovered scene appears more natural and visually pleasing. Our depth map has no missing labels in the buildings.

In Fig. 3.24 we compare with the polarization-based method [65, 66] which requires two polarized hazy images. Our result is from only one of the two images. We can see that our method recovers comparable details and contrast with the polarization-based method. Our result also appear less blueish. Moreover, recent work [81] points out that the polarization-based method is not beneficial, because the polarizer reduces the exposure value and increases the noise level. The noise is further increased when the two images are combined. So if we use only one *unpolarized* image as input, the signal-to-noise ratio (SNR) would be much higher (see [81]).

In Fig. 3.25 we compare with Kopf *et al.*'s work [39] which based on given 3D models of the scene. Our result does not require any geometric information. We notice that even with the given depth, their method cannot handle inhomogeneous haze (like the cloud pointed by the red arrow in Fig. 3.25). Because when β is not a constant, the transmission can not be obtained by

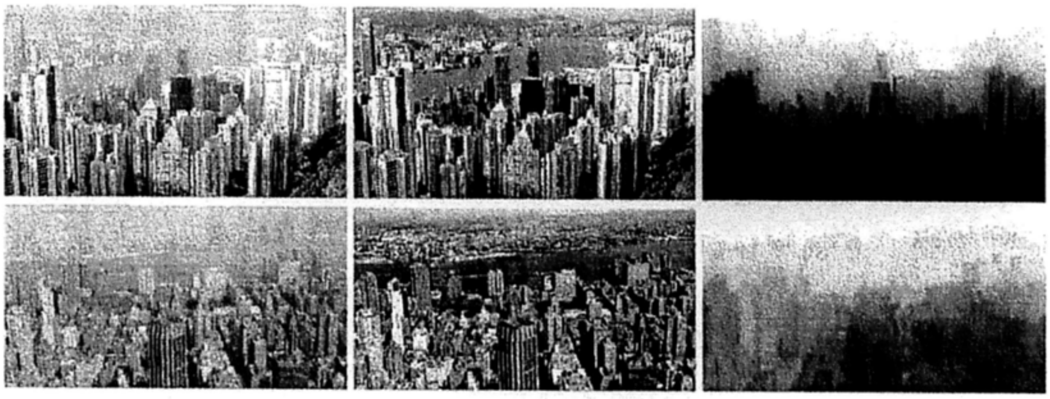


Figure 3.17: Haze removal results (cityscapes).

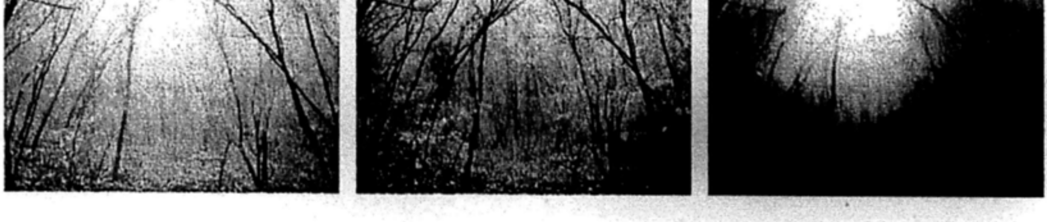


Figure 3.18: Haze removal results (landscapes).

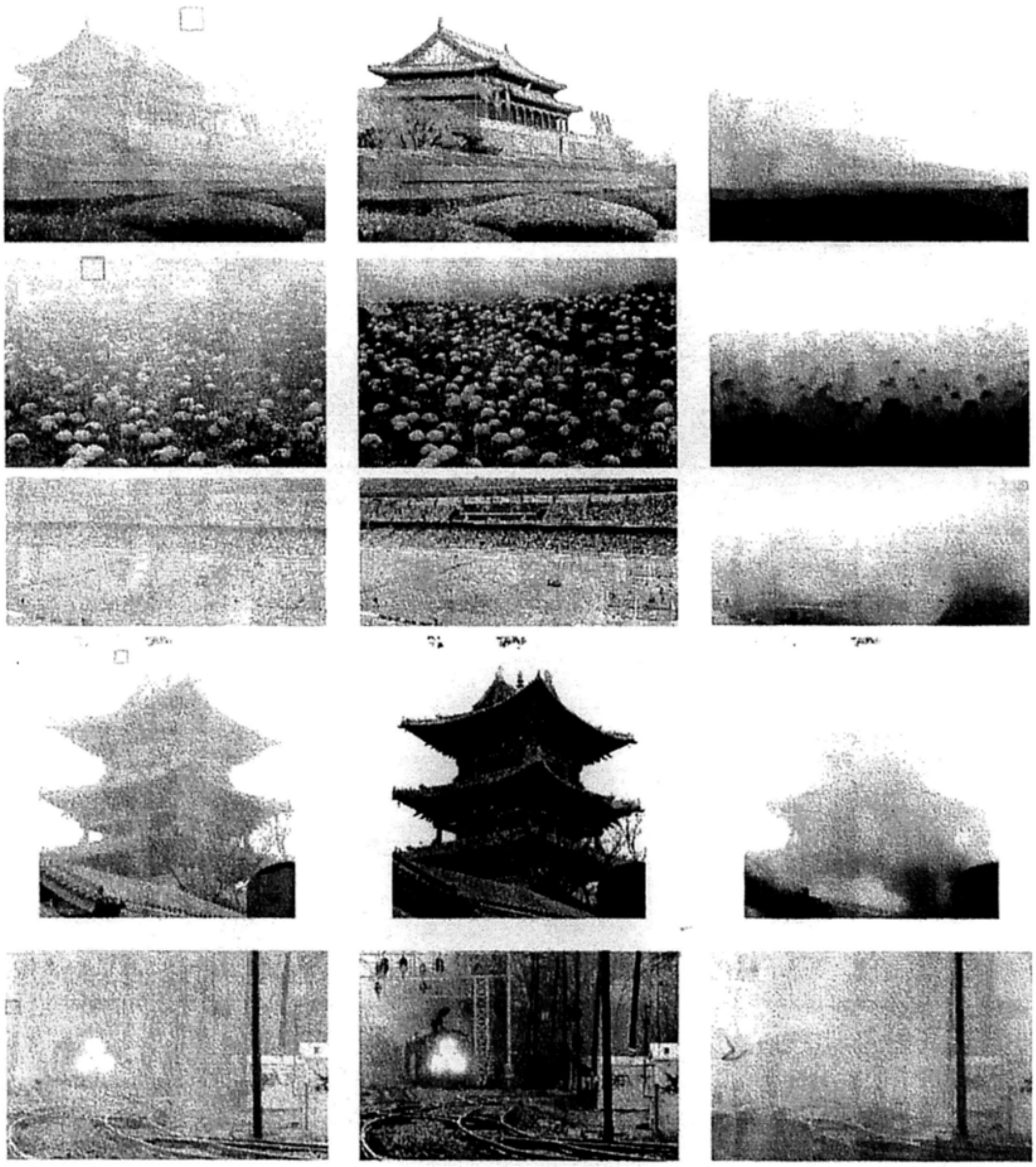


Figure 3.19: Haze removal results (miscellaneous).

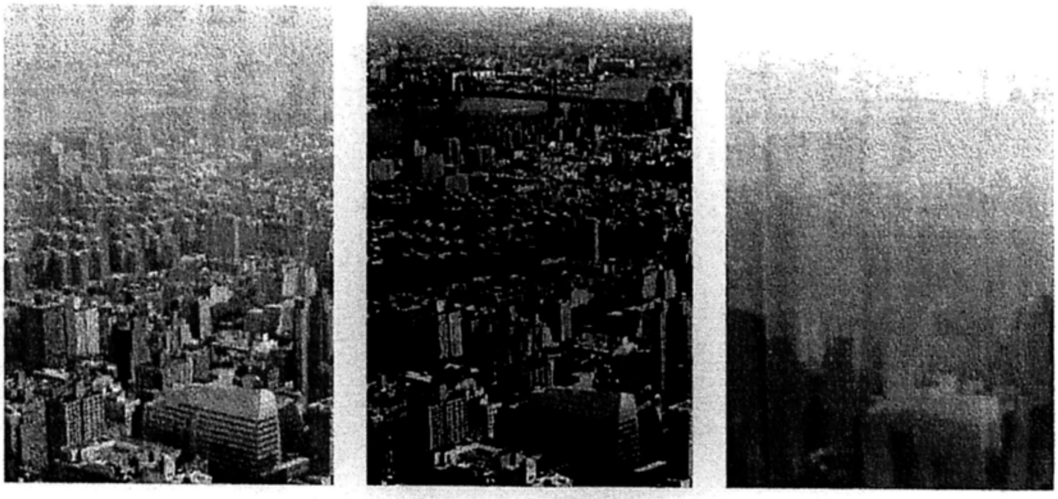


Figure 3.20: Haze removal result (gray-scale).

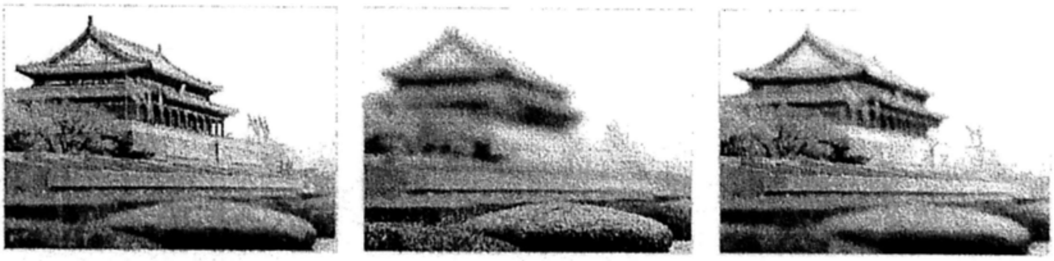


Figure 3.21: Defocusing on three different positions. The input image and depth map is in Fig. 3.19

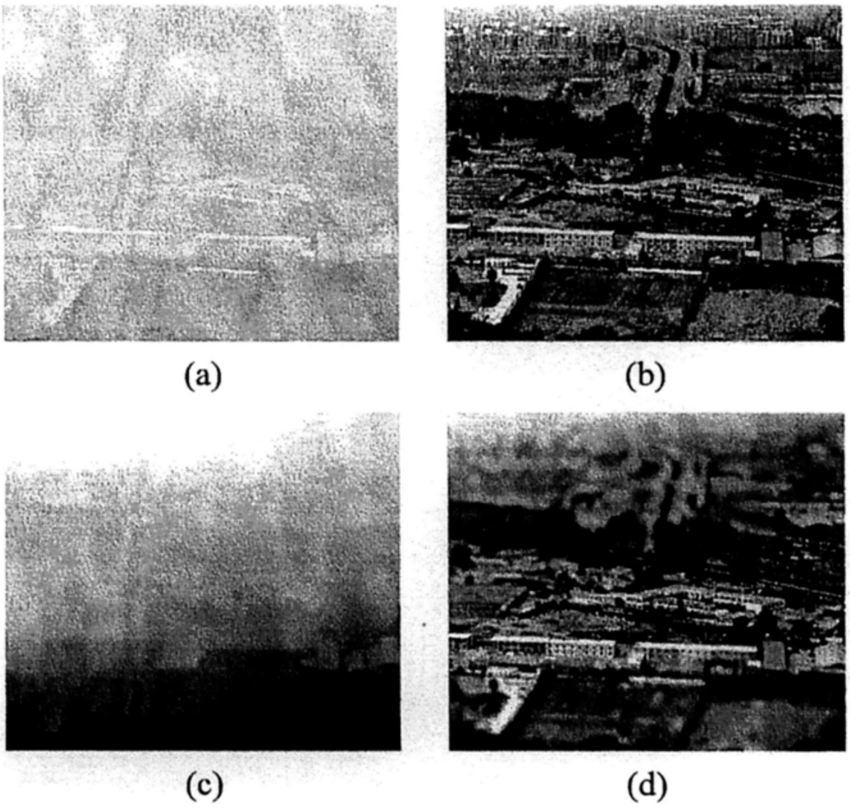


Figure 3.22: Defocusing. (a) Input hazy image. (b) Our haze removal result. (c) Our recovered depth map. (d) A defocused image using the depth map.

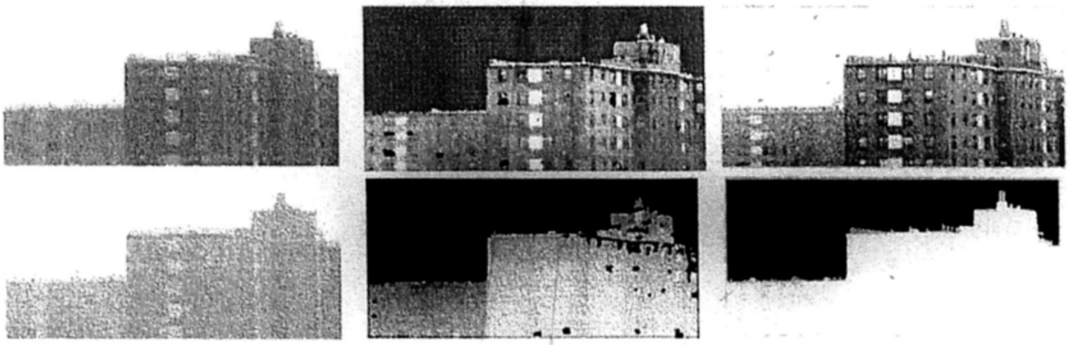


Figure 3.23: Comparison with the dichromatic methods [52, 48, 50]. Left: two input haze images in different weather conditions. Middle: the recovered scene radiance and depth by the dichromatic methods. Right: our results, only using the bottom left image as the input.

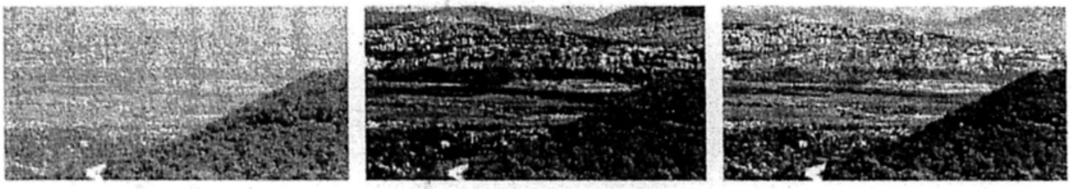


Figure 3.24: Comparison with the polarization-based methods [65, 66]. Left: one of the two input polarized images. Middle: the result in [65]. Right: our result from the single input image.

$t = e^{-\beta d}$ from the given depth. On the contrary, our method does not rely on the constant-β assumption. Our result appears clearer in the cloud-like area.

Next we compare with single image methods. In Fig. 3.26 we compare with the dark object subtraction method [9]. This method assumes a constant transmission, so it can only remove the haze of the nearest objects. See Fig. 3.26 (b): the haze effects in the town is not reduced at all because of the nearer trees. The result of this method the becomes better if the image is cropped (Fig. 3.26 (c)). But the distant haze still remains. On the contrary, our method can remove the haze in each position (Fig. 3.26 (d)).

In Fig. 3.27, we compare our approach with Tan's visibility maximization method [79]. The results of this method have over-saturated colors. This is because the visibility is maximized only when some pixels' intensity is brought below zero. On the contrary, our method recovers the scenes without severely changing the colors (*e.g.*, the swans and the buildings in Fig. 3.27).

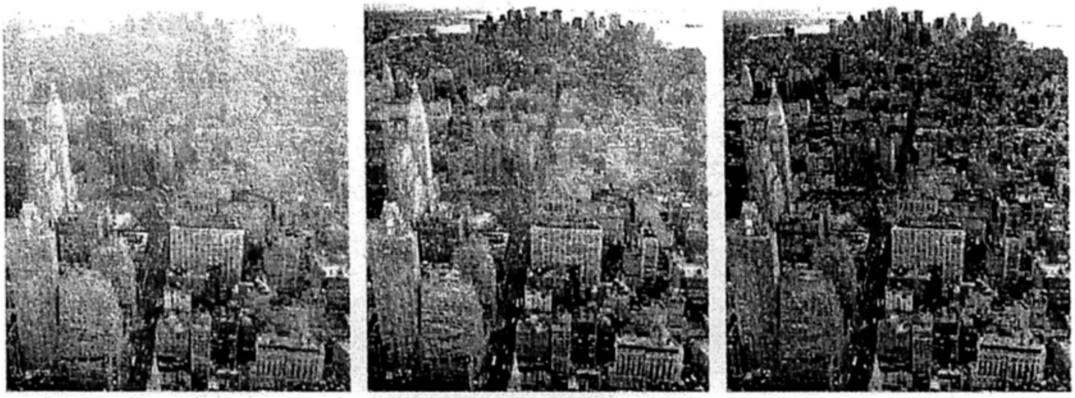


Figure 3.25: Comparison with the 3D-geometry-based method [39]. Left: input. Middle: the result of [39]. Right: our result.

We also notice the apparent halo artifacts in Tan's result. This is because the smoothness term in his MRF is not edge-aware.

Fattal's ICA method [18] is the most competitive one. In Fig. 3.28, we show that our result is comparable with Fattal's representative example in his paper. In Fig. 3.29, we show that our method outperforms Fattal's in dense haze. His method is based on local statistics and requires sufficient color information and variance. When the haze is dense, the color is faint and the variance is not high enough for estimating the transmission. Fig. 3.29 (b) shows Fattal's results using the reliable transmission values: the reliable regions are sparse. The transmission is then extrapolated by MRF. But the results after extrapolation (Fig. 3.29 (c)) are still not satisfactory: some regions are too dark (see the mountains) and some haze is not removed (see the cityscape). On the contrary, our approach is more successful in both cases (Fig. 3.29(d)). This is because our dark channel prior provides reliable estimation in much more regions than Fattal's method.

3.3.4 Limitations

The limitations of our method are mainly due to two reasons: the failure of the dark channel prior, and the invalidty of the haze imaging equation (2.1).

The dark channel prior is statistically based, so there is a chance that some patches or whole images does not obey this prior. The prior may fail when the scene objects are inherently gray or white, with no shadow cast

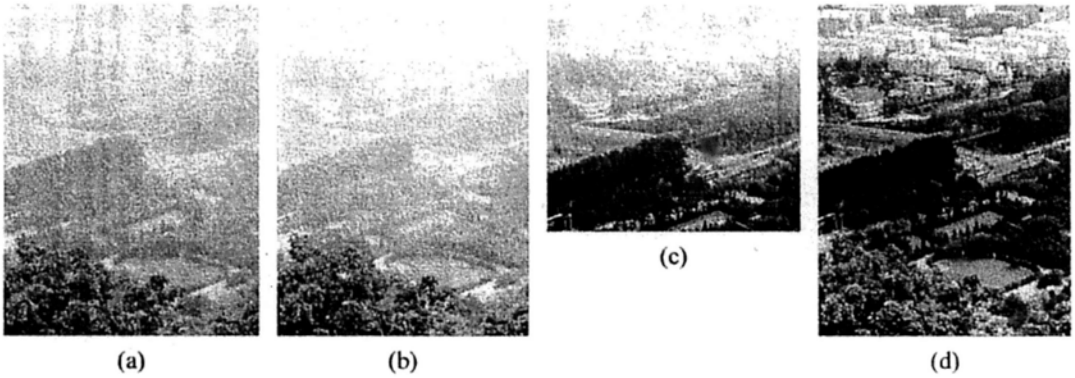


Figure 3.26: Comparison with dark object subtraction [9]. (a) Input. (b) Result of dark object subtraction. (c) Result of Dark Object Subtraction on a cropped image. (d) Our result.

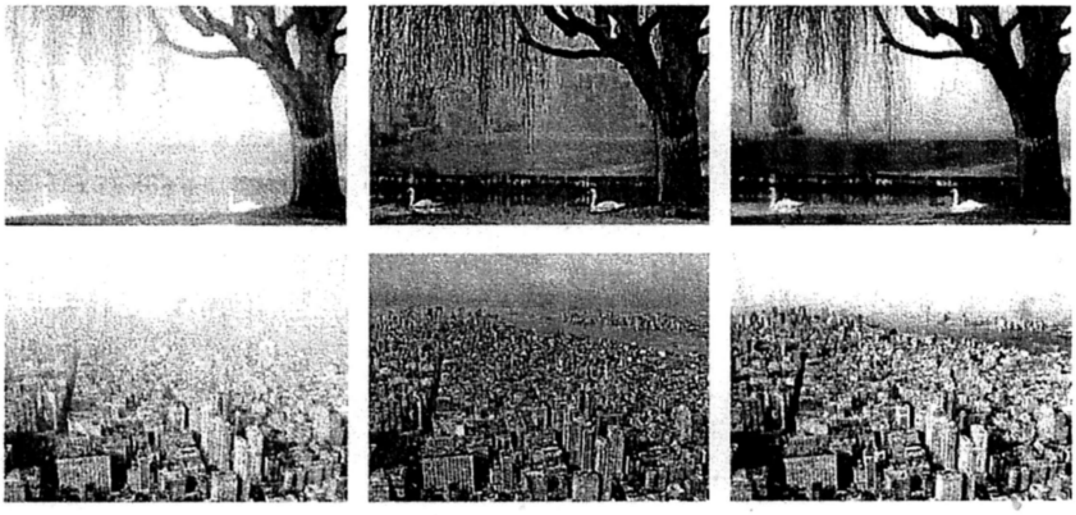


Figure 3.27: Comparison with Tan's Visibility Maximization method [79]. Left: input images. Middle: Tan's results. Right: our results.

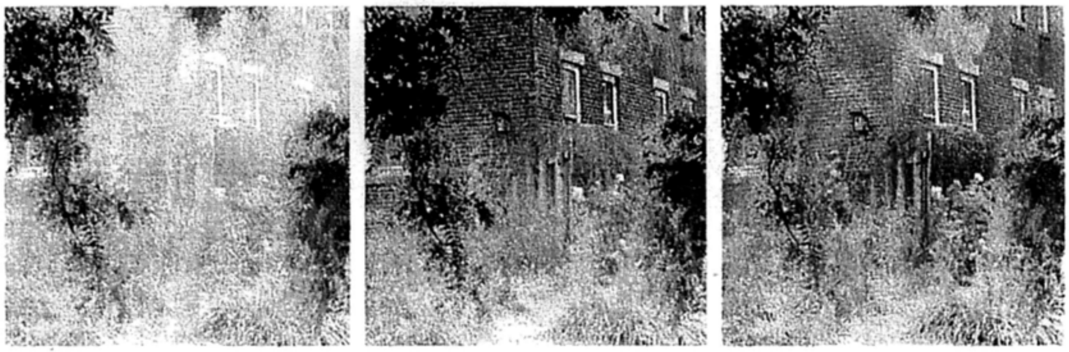


Figure 3.28: Comparison with Fattal's ICA method [18]. Left: input image. Middle: Fattal's result. Right: our result.

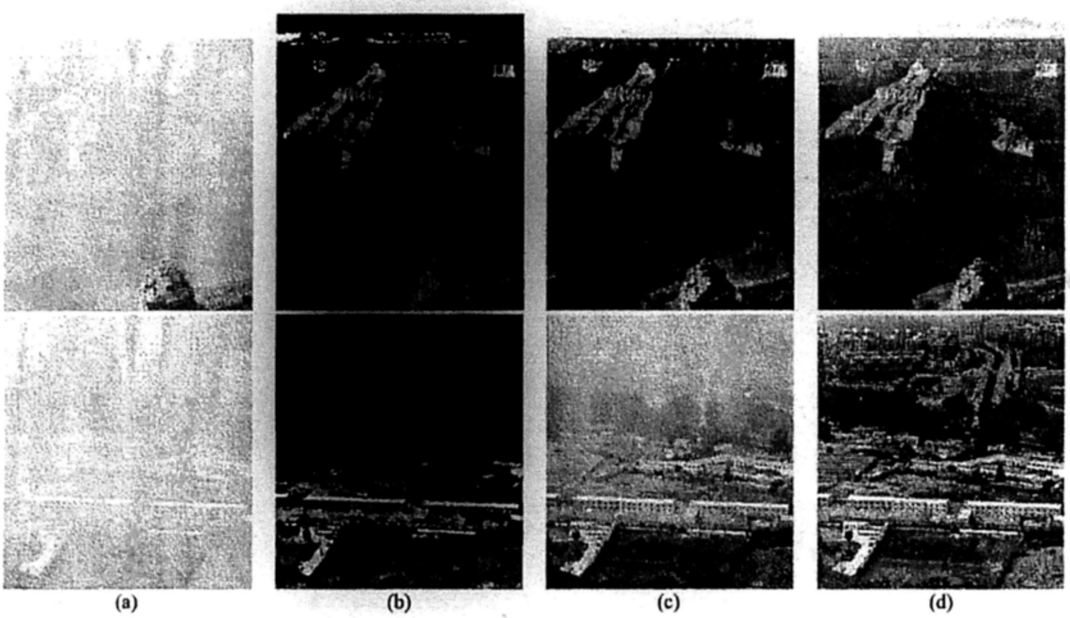


Figure 3.29: More comparisons with Fattal's method [18]. (a) Input images. (b) Fattal's results before extrapolation. The transmission is not estimated in the black regions. (c) Fattal's results after extrapolation. (d) Our results.

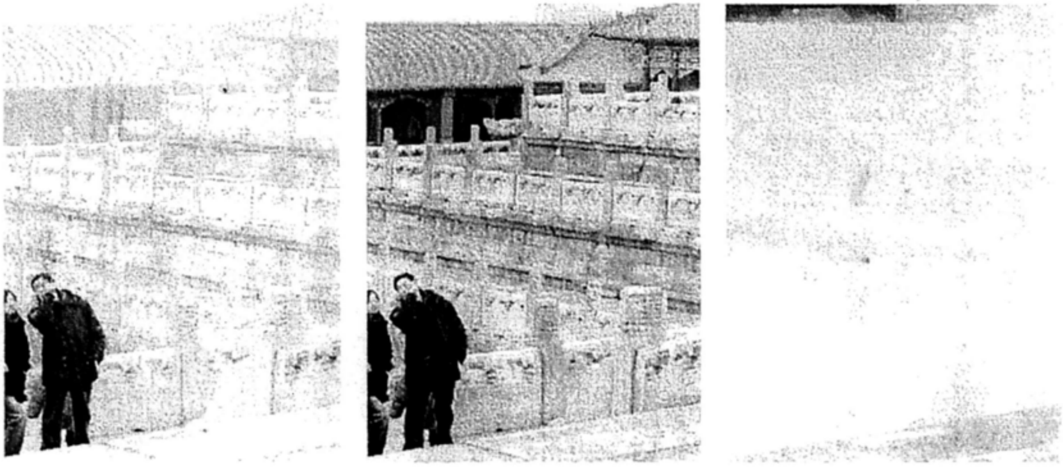


Figure 3.30: Failure of the dark channel prior. Left: input image. Middle: our result. Right: our transmission map. The transmission of the marble is underestimated.

on them, and with no colorful objects around them. The white marble in Fig. 3.30 is an example. In this case, our method may overestimate the thickness of the haze (underestimate the transmission). The recovered colors become more saturated than they should be.

Another limitation is that the haze imaging equation (2.1) may be invalid. It involves two situations: (i): channel-dependent transmission t, and (ii) nonconstant atmospheric light $\mathbf{A}$.

(i): Channel-dependent transmission. In the haze imaging equation (2.1), it is assumed that the scattering coefficient β is independent of the wavelength λ. But this is not true when the particles in atmosphere is too small. An example is the Rayleigh Scattering [23]: the particles are air molecules instead of dust or water droplet. In this case, the transmission t should be modified as channel-dependent (see [49, 65]):

$$t_c(\mathbf{x}) = e^{-\beta_c d(\mathbf{x})}, \tag{3.35}$$

where the scattering coefficients usually satisfy: $\beta_r < \beta_g < \beta_b$. And we can modify the haze imaging equation (2.1) by:

$$I_c(\mathbf{x}) = J_c(\mathbf{x})t_c(\mathbf{x}) + A_c(1 - t_c(\mathbf{x})). \tag{3.36}$$

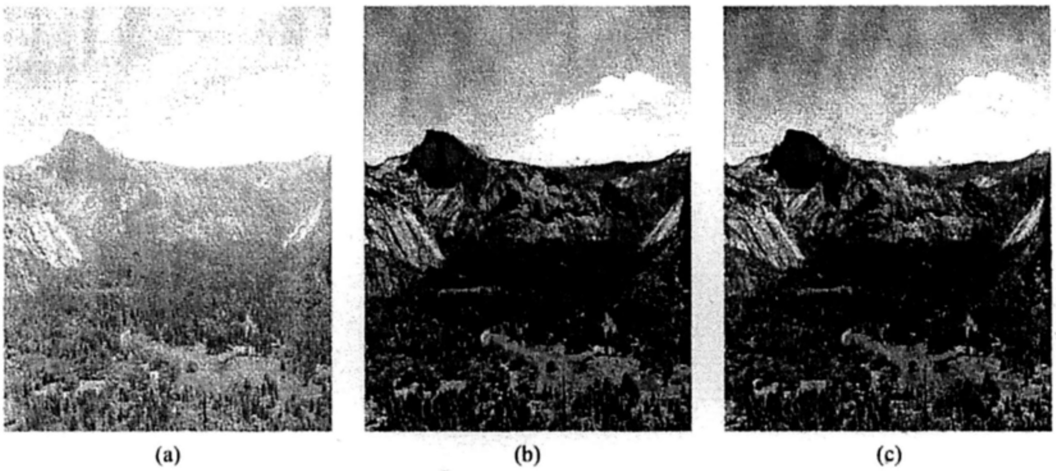


Figure 3.31: Channel-dependent transmission. Left: input image. Middle: our result using the original haze imaging model (2.1). Right: our result where each channel is processed independently.

In Rayleigh Scattering, the blue channel transmission t_b is often the smallest. So the distant objects often appears blush (see Fig. 3.31 (a)). Rayleigh Scattering is also the reason for the blue sky. If we follow the original haze imaging equation (2.1) to remove haze, the distant objects become bluish (see Fig. 3.31 (b)). Notice that the nearer objects is less influenced[7]. One way to fix this problem is to process each channel independently. The result is in Fig. 3.31 (c). We can see that the bluish artifacts is removed. But this modification requires that each channel has its own dark pixels, which is less valid than the dark channel prior in some cases.

(ii): nonconstant atmospheric light. The haze imaging equation (2.1) may also be invalid when the atmospheric light $\mathbf{A}$ is nonconstant. We should modify it by:

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(\mathbf{x})(1 - t(\mathbf{x})), \tag{3.37}$$

where $\mathbf{A}(\mathbf{x})$ depends on the position. This is often due to *point light sources* like the sun and the street lights at night. The atmospheric light is stronger near the center of a point light source (see Fig. 3.32). In this case our method may fail because our fixed atmospheric light is not correct in most positions of the image (see Fig. 3.33).

[7] This is because of the exponential dependency in (3.35).

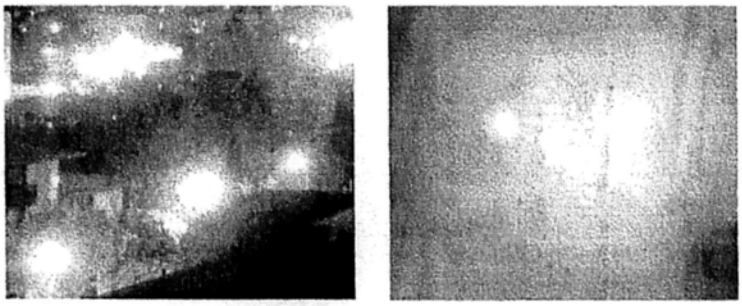


Figure 3.32: Point light sources lead to nonconstant atmospheric light.

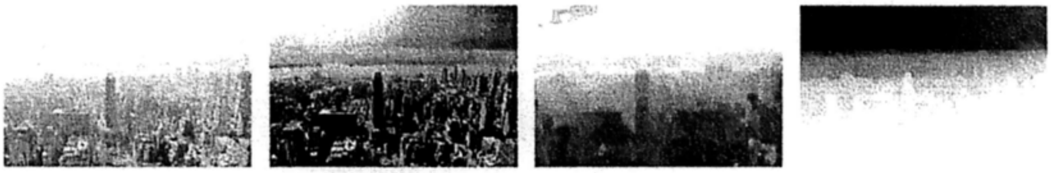


Figure 3.33: Nonconstant atmospheric light. (a): Input image. (b): Our result. (c): Dark Channel. Red pixels indicate where the atmospheric light is estimated. (d): Estimated transmission map.

Both the channel-dependent transmission model (3.36) and the nonconstant atmospheric light model (3.37) increase the number of unknowns from about $4N$ to about $6N$. So we require more prior knowledge or assumptions to handle these situations. We will study these problems in the future.

3.4 Conclusion

In this chapter, we propose the dark channel prior and a haze removal algorithm based on it. This novel prior is very simple but effective. We show experiments to support its validity. We also show that our algorithm outperforms previous works in various experiments.

We do not focus on the running time in this chapter. The only time consuming step is to solve the linear system in soft matting (3.21). In the next chapter, we propose two fast algorithms to address the speed issue.

We find that the dark channel prior may also be adopted by the human visual system. This is another strong support for our prior as a computer vision algorithm. We investigate this problem in Chapter 5.

Chapter 4

Fast Algorithms

In practical applications, the efficiency of a haze removal algorithm is important. For example, in outdoor surveillance system it is required to process the video in real-time; in an image editing system, instant feedback is essential for user experience. In this chapter, we study the speed-up of our haze removal algorithm.

All the operations of our algorithm except soft matting can be computed in real-time for mega-pixel images. The purpose of soft matting is to combine pixel-wise constraints with spatial continuities. This was achieved by solving a linear system. In the first section of this chapter, we propose a large-kernel-based algorithm[1] to increase the speed of the linear solver. In the second section, we propose a non-iterative filter called *guided filter*[2] to replace the soft matting step. This technique allows us to process mega-pixel images in real-time with almost no quality degradation.

The guided filter is an edge-aware filter. Edge-aware filtering is an important and widely applied technique in computer vision/graphics. The guided filter is also applicable in various applications besides haze removal. We show that this filter is a state-of-the-art technique for edge-aware filtering, in terms of both quality and efficiency.

[1] This method is published in our paper [31] as an alpha matting method.

[2] The guided filter is published in our paper [32] as a general edge-aware filter.

4.1 A Large-Kernel-Based Linear Solver

In the soft matting step, we need to solve a large linear system with a matting Laplacian matrix L:

$$(\mathrm{L} + \lambda \mathrm{U})\mathbf{t} = \lambda \tilde{\mathbf{t}}. \tag{3.21}$$

A popular and generally superior technique is the Conjugate Gradient solver (CG) [63], which takes $O(N^{\frac{3}{2}})$ time in terms of the number of pixels N. It is quite slow when N gets larger.

In this section, we focus on the kernel size of the linear system. Considering the kernel radius r, we shall show that a conventional CG algorithm takes $O(rN^{\frac{3}{2}})$ time: it is slower when r gets larger. This is one reason why previous methods [41, 28, 42, 84, 36] often use the smallest kernel, no matter in what form the linear system is. However, we shall propose an $O(\frac{1}{r}N^{\frac{3}{2}})$ time algorithm particularly designed for the matting Laplacian matrix. It allows us to reduce the running time by choosing a larger kernel. This is an interesting discovery, because in conventional theories solving a less sparse (*e.g.*, larger kernel) linear system requires more time.

4.1.1 Related Works: Linear Solvers

Before introducing our algorithm, we briefly review the previous works on linear solvers. They are in two categories: direct (non-iterative) methods and iterative methods.

Direct methods find a solution in a finite number of operations. The Gaussian Elimination is a well-known strategy, which is in essential a matrix decomposition method. Other variations include LU, QR, and Cholesky decomposition [12]. These methods are effective for small linear systems. But they suffer from the "fill-in" problems [12, 63]: more and more entries in the matrix becomes non-zero during decomposition. The number of non-zero entries is $O(N^2)$ in the worst case. So these methods can easily run out of memory when N is the image size. For example, the memory for solving a 1000×1000 image can be in the order of 10^6 megabytes (1M MB or 1000 GB).

The Fourier transform can be used as a direct solver for some special linear systems, like the Poisson Equation [74] and the Screened Poisson Equation [6]. If the kernel of the matrix is spatially invariant, we can treat the linear system as a convolution and solve it by Fourier transform. But this is not the case in many problems, including any edge-aware matrices (*e.g.*the matting Laplacian matrix) [78] and any non-uniform data weights [77].

Due to the memory issue, practical linear solvers in computer vision/graphics are the iterative methods [63]. The Jacobi method, Gauss-Seidel method, Successive Over Relaxation (SOR), and Conjugate Gradient (CG) are some widely used ones. In general, these methods iteratively multiply a matrix and converge to the true solution. But their convergence speed is not fast. For example, it often takes hundreds to thousands of iterations for a CG solver to converge in many image problems [77, 78].

Preconditioning [63] is a technique to ensure fast convergence of the iterative solvers, often applied in Conjugate Gradient. The Preconditioned Conjugate Gradient (PCG) method multiplies a specially designed matrix (called preconditioner) in each iteration, which is expected to increase the convergence speed. Several preconditioners for general linear systems are given in [63], but their performance is less satisfactory in image problems [77, 78]. The hierarchical basis preconditioning (HBP) [77] exploits the multi-scale nature of image problems. It is particularly useful when the solution is mostly smooth. The locally adaptive hierarchical basis preconditioning (LAHBP) [78] improves this technique for various inhomogeneous problems like edge-aware smoothing. But the LAHBP is not applicable for the matting Laplacian matrix, because it requires the matrix to be first-order smooth[3].

The multigrid technique is another way to accelerate the iterative solvers [7]. It also exploits the multi-scale nature of image problems. It solves the problem at a coarser scale, and corrects the error at a finer scale. This process is often run recursively and iteratively. The multigrid method is optimal for homogenous problems like the Poisson Equation, but is less satisfactory in irregular cases.

4.1.2 Matting Laplacian Matrix

Our algorithm proposed in this section is particularly designed for the matting Laplacian matrix. We first introduce the derivation of this matrix. The derivation is mainly following Levin *et al.*'s alpha matting paper [42] in which the matrix is proposed. But we discuss in the scenario of haze removal. We shall also see why we choose this matrix in our MRF model (3.18): it is compatible with the haze image equation, and has nice edge-aware properties.

We still denote the estimated transmission in (3.14) by $\tilde{t}$, and the refined transmission by t. For simplicity, we denote the image as a one-channel image I (but the color image case is similar). Each image or map can be written

[3] This means that L_{ij} is non-zero only when j is in the 4-neighborhood of i

as an N×1 column vector. For example, the ith element of the vector $\mathbf{t}$ is:

$$t_i \equiv t(\mathbf{x}), \quad \text{with} \quad i = x + y * wid \quad \text{and} \quad \mathbf{x} = (x, y),$$

where wid is the image width. The index i is in the range $[0, N-1]$.

We assume that in a small local window w there is a linear transformation between the refined transmission t and the hazy image I:

$$t_i = aI_i + b, \quad \forall i \in w, \tag{4.1}$$

where a and b are the linear coefficients assumed to be constant in the window w. In fact, this model is compatible with the haze model. From the haze imaging equation $I_i = J_i t_i + A(1 - t_i)$ we can write t_i as:

$$t_i = \frac{1}{J_i - A} I_i - \frac{A}{J_i - A}. \tag{4.2}$$

If J_i is approximately constant in w, we let $a = \frac{1}{J-A}$ and $b = -\frac{A}{J-A}$ and obtain (4.1). In [42] it is shown that the constant-J assumption can be alleviated to a *color line* assumption. Later we shall further generalize it to a *color plane* assumption. This ensures that the local linear model (4.1) is valid in most situations.

The local linear model (4.1) has a good edge-aware property. In fact, if we take the gradient of (4.1) we have:

$$\nabla t_i = a \nabla I_i \quad \forall i \in w. \tag{4.3}$$

This indicates that t has an edge only if the image I has an edge, and t is smooth if I is smooth. This is the main purpose of soft matting - imposing continuities on t according to the edges of I.

We encourage the refined transmission t to obey the local linear model (4.1). So we minimize a cost function $\mathcal{E}(\mathbf{t}, \mathbf{a}, \mathbf{b})$:

$$\mathcal{E}(\mathbf{t}, \mathbf{a}, \mathbf{b}) = \sum_{k \in \mathcal{I}} \left(\sum_{i \in w_k} (t_i - a_k I_i - b_k)^2 + \varepsilon a_k^2 \right). \tag{4.4}$$

In this equation, k is another pixel index and $\mathcal{I}$ denotes the set of all the pixels in the image. w_k is a window centered at pixel k. We assume that each local window w_k has its own linear coefficients a_k and b_k. So the set of all a_k (and b_k) forms a map, denoted as a column vector $\mathbf{a}$ (and $\mathbf{b}$). ε is a regularization parameter.

If we further constrain that t should approximate the estimated $\tilde{t}$, we add a data term to the cost function:

$$\mathcal{E}(\mathbf{t},\mathbf{a},\mathbf{b}) = \sum_{k\in\mathcal{I}}\left(\sum_{i\in w_k}(t_i - a_k I_i - b_k)^2 + \varepsilon a_k^2\right) + \sum_{k\in\mathcal{I}}(t_k - \tilde{t}_k)^2. \tag{4.5}$$

Notice that this is a quadratic function $w.r.t.$ each a_k and b_k. The cost function is minimized by setting each partial derivative $\partial\mathcal{E}(\mathbf{t},\mathbf{a},\mathbf{b})/\partial a_k$ and $\partial\mathcal{E}(\mathbf{t},\mathbf{a},\mathbf{b})/\partial b_k$ to zero:

$$\begin{cases} \dfrac{\partial\mathcal{E}(\mathbf{t},\mathbf{a},\mathbf{b})}{\partial a_k} = \displaystyle\sum_{i\in w_k} 2I_i(a_k I_i + b_k - t_i) + 2\varepsilon a_k & = 0 \\ \dfrac{\partial\mathcal{E}(\mathbf{t},\mathbf{a},\mathbf{b})}{\partial b_k} = \displaystyle\sum_{i\in w_k} 2(a_k I_i + b_k - t_i) & = 0 \end{cases} \tag{4.6}$$

The solution is:

$$\begin{cases} a_k = \dfrac{\mathrm{cov}_k(I,t)}{\sigma_k^2 + \frac{1}{|w_k|}\varepsilon} \\ b_k = \bar{t}_k - a_k\mu_k. \end{cases} \tag{4.7}$$

Here, $|w_k|$ is the number of pixels in w_k, $\bar{t}_k = \sum_{i\in w_k} t_i$ and $\mu_k = \sum_{i\in w_k} I_i$ are the mean of t and I in the window w_k, $\mathrm{cov}_k(I,t)$ is the covariance of I and t in w_k, and σ_k^2 is the variance of I in the w_k. Notice that this is the solution to a simple linear regression problem [14].

We should point out that the solution (4.7) is written as an implicit matrix form in Levin $et\ al.$'s derivation [42]. But it is the explicit form (4.7) that inspires our large-kernel-based method [31] and the guided filter [32].

Putting (4.7) into (4.5) we can eliminate all the a_k and b_k, leaving us a quadratic function on t:

$$\mathcal{E}(\mathbf{t}) = \mathbf{t}^{\mathrm{T}}\mathrm{L}\mathbf{t} + \lambda\|\mathbf{t} - \tilde{\mathbf{t}}\|_2^2. \tag{4.8}$$

After a series of algebraic operations (see [42]), we can obtain the entries of the matting Laplacian matrix L:

$$\mathrm{L}_{ij} = \sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k + \frac{\varepsilon}{|w_k|}})), \tag{4.9}$$

where δ_{ij} is the Kronecker delta. For an RGB image $\mathbf{I}$, the formulation is in (3.20).

Equation (4.8) is exactly the cost function in (3.18). Its solution can be obtained by solving the linear system for $\mathbf{t}$ as in (3.21):

$$(\mathrm{L} + \lambda\mathrm{U})\mathbf{t} = \lambda\tilde{\mathbf{t}}. \tag{4.10}$$

where U is an identity matrix of the same size as L. The matrix L is an $N \times N$ matrix. Intuitively, the entry L_{ij} describes the relationship between any two pixels indexed by i and j. If an image has $N = 1000 \times 1000$ pixels, the matrix L has 10^6 entries in each row and $10^6 \times 10^6$ entries in total - which is a very huge number.

Fortunately, the matrix is *sparse*: only a small portion of its entries are non-zero. Notice the summation $\sum_{k|(i,j)\in w_k}$ in (4.9). Given a fixed pair of (i, j), the summation is over all the windows w_k that contain both pixels at i and j. The summation is not empty (so L_{ij} can be non-zero) only if i and j can be covered by at least one window. See Fig. 4.1 for an illustration where the pixels i and j are the furthest to ensure the summation is not empty. We can see that if the radiuses of the windows w is r, L_{ij} can be non-zero only when the pixel j is in a $(4r + 1) \times (4r + 1)$ neighborhood around the pixel i.

We define *kernel size* as the number of non-zero entries in each row of L. The i^{th} row of L has at most $(4r + 1) \times (4r + 1)$ non-zero entries, so the kernel size of L is $(4r + 1) \times (4r + 1)$. When the window w is getting larger, the matrix has more non-zero entries and becomes less sparse. In the previous methods [42, 36, 84] the radius r is always set to 1. One reason is that solving a less sparse linear system is slower in conventional theories.

4.1.3 Algorithm

Our observation is that a larger kernel actually reduces the iteration number. The cost is from the highly expensive in-iteration computation. But if we can

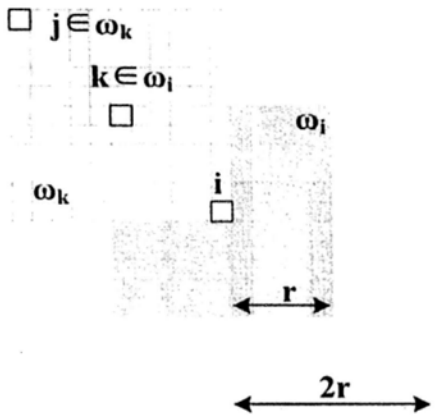


Figure 4.1: Kernel size. The kernel is a rectangle including all the gray pixels in this figure.

compute each iteration more efficiently, the total complexity can be reduced thanks to the faster convergence.

Analysis of Conjugate Gradient

In conventional theories, solving a less sparse matrix is often slower. But we point out it is not necessarily true. Here we analyze this issue in the Conjugate Gradient method; other iterative methods are essentially similar. We discuss a general linear system:

$$\mathrm{A}\mathbf{x} = \mathbf{b}, \tag{4.11}$$

where A is an $N \times N$ matrix, and $\mathbf{x}$ and $\mathbf{b}$ are $N \times 1$ vectors. In soft matting, we have $\mathrm{A} \equiv \mathrm{L} + \lambda \mathrm{U}$, $\mathbf{x} \equiv \mathbf{t}$, and $\mathbf{b} \equiv \lambda \tilde{\mathbf{t}}$.

The Conjugate Gradient algorithm is given in Algorithm 2, where we use ":=" to denote a variable assignment. In each iteration of CG, all the steps except Step 4 are vector addition, subtraction, or inner product, whose time complexity is $\mathrm{O}(N)$ regardless of the matrix A. The computational cost is dominated by the matrix multiplication in Step 4:

$$q_i = \sum_j \mathrm{A}_{ij} p_j. \tag{4.12}$$

Algorithm 2 Conjugate Gradient: solving $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$

```
Require:  Arbitrary initial guess $\mathbf{x}^0$
 1: Initial $\mathbf{r} := \mathbf{b} - \mathbf{A}\mathbf{x}^0$, $\mathbf{p} := \mathbf{r}$
 2: repeat
 3:     $\gamma := \mathbf{r}^{\mathrm{T}}\mathbf{r}$
 4:     $\mathbf{q} := \mathbf{A}\mathbf{p}$
 5:     $\alpha := \gamma / \mathbf{p}^{\mathrm{T}}\mathbf{q}$
 6:     $\mathbf{x} := \mathbf{x} + \alpha\mathbf{p}$
 7:     $\mathbf{r} := \mathbf{r} - \alpha\mathbf{p}$
 8:     $\beta := \frac{1}{\gamma}\mathbf{r}^{\mathrm{T}}\mathbf{r}$
 9:     $\mathbf{p} := \mathbf{r} + \beta\mathbf{p}$
10: until convergence
```

The number of operations for computing q_i is proportional to the number of non-zero elements in the i^{th} row of A, *i.e.*, the kernel size. So the time complexity of this step is $O(r^2N)$: it would be very high when the kernel is large.

But the running time of the solver also depends on the number of iterations. A pixel influences more pixels in each iteration when the kernel is larger. Intuitively, the "information" of a pixel is "propagated" at a speed of $O(r)$ per iteration. Suppose the image length is $N^{\frac{1}{2}}$. A pixel influences all the other pixels in $O(N^{\frac{1}{2}}/r)$ iterations: this is also expected to be the iteration number for convergence (we shall prove it theoretically). Since each iteration is $O(r^2N)$ complex, the total complexity of the solver is $O(r^2N) \times O(N^{\frac{1}{2}}/r) = O(rN^{\frac{3}{2}})$. This indicates the faster convergence cannot counteract the slower matrix multiplication, and solving a larger kernel matrix is slower.

However, we shall propose an $O(N)$ time algorithm to compute (4.12) particularly for the matting Laplacian matrix. $O(N)$ time indicates the complexity is independent of the kernel size r. Then the total speed of the solver becomes $O(N) \times O(N^{\frac{1}{2}}/r) = O(N^{\frac{3}{2}}/r)$. We can solve the linear system faster by using a larger r.

An O(1) time algorithm for matting Laplacian multiplication

The focus is on $\mathbf{q} := \mathbf{Ap}$ (Step 4, Algorithm 2). Notice that the matrix A is $\mathrm{L} + \lambda\mathrm{U}$ in our problem. So the computation of $\mathbf{q} := \mathbf{Ap}$ is in two steps: (i) $\mathbf{q} := \mathrm{L}\mathbf{p}$; and (ii) $\mathbf{q} := \mathbf{q} + \lambda\mathbf{p}$. The key is step (i).

Given any $\mathrm{N} \times 1$ vector $\mathbf{p}$, we show that $\mathrm{L}\mathbf{p}$ can be calculated by Algorithm 3. In the algorithm, a_k^* and b_k^* are intermediate variables in each window w_k.

Algorithm 3 O(N) time algorithm for computing $\mathbf{q} := \mathrm{L}\mathbf{p}$

```
1.
    $$a_k^* := \frac{\mathrm{cov}_k(I, p)}{\sigma_k^2 + \frac{\varepsilon}{|w|}} \tag{4.13}$$
2.
    $$b_k^* := \bar{p}_k - a_k^* \mu_k \tag{4.14}$$
3.
    $$q_i := |w| p_i - \left( (\sum_{k \in w_i} a_k^*) I_i + (\sum_{k \in w_i} b_k^*) \right) \tag{4.15}$$
```

Theorem 4.1.1. *The* $\mathbf{q}$ *computed by Algorithm 3 equals to* $\mathrm{L}\mathbf{p}$, *where* L *is the matting Laplacian matrix defined in (4.9).*

Proof. Equation (4.13) indicates a_k^* is a weighted combination of p_i in the window w_k: $a_k^* = \sum_{i \in w_k} \mathrm{W}_{ki}^a p_i$, where the weights W_{ki}^a depend on I. Written in a matrix form, this is a linear transform: $\mathbf{a}^* = \mathrm{W}^a \mathbf{p}$, where W^a is a coefficient matrix. If we put (4.13) into (4.14) and eliminate a_k^*, we can obtain another linear transform: $\mathbf{b}^* = \mathrm{W}^b \mathbf{p}$. Similarly, $\mathbf{q}$ is also $\mathbf{p}$'s linear transform: $\mathbf{q} = \mathrm{W}^q \mathbf{p}$. Consequently, to prove $\mathbf{q}$ equals to $\mathrm{L}\mathbf{p}$, we only need to prove $\partial q_i / \partial p_j = \mathrm{L}(i, j)$.

The calculation of $\partial q_i / \partial p_j$ from (4.13)-(4.15) is straightforward but technical. Please see Appendix B.1 for the details. □

We can show the time complexity of Algorithm 3 is $O(N)$. In fact, all the operations in (4.13) and (4.14) are mean, variance, or covariance of sliding windows. They can be written as a series of box filters. A box filter can be very efficiently computed in $O(N)$ time by integral images [22] or cumulative sums, regardless of the filter radius r (see Appendix B.2 for details). So the operation $\mathbf{q} := \mathbf{L}\mathbf{p}$ can be computed in $O(N)$ time instead of $O(r^2N)$.

We notice that (4.13) and (4.14) are analogous to (4.7) if we analogize $\mathbf{p}$ to $\mathbf{t}$. This is not surprising. The vector $\mathbf{p}$ in the CG algorithm is a modified residual (called *conjugate vector*) of the solution $\mathbf{x}$ in the current iteration, and $\mathbf{x}$ is $\mathbf{t}$ for which we solve in soft matting. Consequently, given the solution $\mathbf{t}$ of the current iteration, if we compute a_k and b_k as in (4.7), then a_k^* and b_k^* actually forms the conjugate vectors of a_k and b_k.

Convergence Speed

In this section, we theoretically analyze the convergence speed with respect to the kernel size. In the CG algorithm, the number of iterations needed for convergence is [77]:

$$n_{\text{iter}} \approx \frac{1}{2}\sqrt{\kappa}\ln(2/\epsilon), \tag{4.16}$$

where the error ϵ is a preset convergence criterion, and κ is the *condition number* of the matrix A. The condition number is defined as the ratio of the largest eigenvalue to the smallest eigenvalue:

$$\kappa = \frac{\xi^A_{\max}}{\xi^A_{\min}}, \tag{4.17}$$

where ξ^A represents the eigenvalue of the matrix A. Because in our problem $\mathrm{A} = \mathrm{L} + \lambda\mathrm{U}$, we have the relation:

$$\xi^A = \xi^L + \lambda. \tag{4.18}$$

So we have to find the eigenvalues ξ^L of the matting Laplacian matrix L.

The matrix L is dependent on the image I (see (4.9)), so it is difficult to obtain a general form of ξ^L for arbitrary images. Instead, we only consider the approximately constant regions of I. Usually many regions satisfies this condition. According to (4.9), for constant I the elements of L are given by:

$$\mathrm{L}_{ij} = \sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w|}). \tag{4.19}$$

In this equation L_{ij} does not depend on I's values. The eigenvalues of this matrix are approximately given by (see Appendix B.3 for the details):

$$\xi^L_{\max} = |w|, \tag{4.20}$$

and

$$\xi^L_{\min} \approx \frac{\pi^2}{6}\frac{1}{N}|w|^2. \tag{4.21}$$

Then the conditioned number κ is:

$$\kappa = \frac{\xi^A_{\max}}{\xi^A_{\min}} \approx \frac{\xi^L_{\max}}{\xi^L_{\min}} = O(\frac{N}{|w|}) = O(\frac{N}{r^2}). \tag{4.22}$$

Here $\frac{\xi^A_{\max}}{\xi^A_{\min}} \approx \frac{\xi^L_{\max}}{\xi^L_{\min}}$ is because λ is set as a small value in soft matting.

Having obtained κ, the number of iterations (4.16) is:

$$n_{\text{iter}} = O(\frac{1}{r}N^{\frac{1}{2}}). \tag{4.23}$$

This is consistent with our intuitive explanation.

Because the complexity of each iteration is $\mathrm{O}(N)$, so the total complexity of the linear solver is $\mathrm{O}(\frac{1}{r}N^{\frac{3}{2}})$. Thus we have proven that using a larger kernel is actually faster. We can also see that without the fast Algorithm 3 the total complexity is $\mathrm{O}(rN^{\frac{3}{2}})$. This is the complexity of the conventional CG method.

Color Plane Model

Our linear solver for the matting Laplacian matrix is *non-approximate* for the cost function with a *fixed* kernel radius r. But it is worth mentioning that if we change r, we also change the cost function that is to be optimized. In general, the solution obtained by r_1 is not identical to the one obtained by r_2 if $r_1 \neq r2$. Next we investigate whether the haze removal quality is influenced by a large kernel.

The basic motivation of the matting Laplacian matrix is the local linear model (4.1):

$$t_i = aI_i + b, \quad \forall i \in w. \tag{4.1}$$

The quality of the soft matting algorithm depends on the validity of this model. Previously (see (4.2)), we assume that J is nearly constant in the window w to obtain this model. Next we show that this assumption can be alleviated to a *color plane* assumption for color images. This is a more general assumption.

In the case of a color image $\mathbf{I}$, the model is extended as:

$$t_i = \mathbf{a} \cdot \mathbf{I}_i + b, \quad \forall i \in w. \tag{4.24}$$

where $\mathbf{a}$ is a 3×1 constant vector, and $\mathbf{I}_i$ is a 3×1 vector of the color at pixel i. In Levin *et al.*'s work [42], it is show that this local linear model is valid under the *color line* assumption: the foreground/background colors in a local window are in a line in the RGB color space. This assumption has been empirically verified in [42, 53].

But we show that we can further generalize the color line assumption to a *color plane* assumption: the colors of $\mathbf{J}$ in a local window are in a plane in the RGB color space (see Fig. 4.2). Denote the plane normal by a vector $\mathbf{n}$. The color plane model is:

$$\mathbf{n} \cdot \mathbf{J}_i = C, \quad \forall i \in w. \tag{4.25}$$

where C is a constant in the window w. Recall the haze imaging equation (2.1):

$$\mathbf{I}_i = \mathbf{J}_i t_i + \mathbf{A}(1 - t_i). \tag{4.26}$$

We project this equation on $\mathbf{n}$ and obtain: $\mathbf{n} \cdot \mathbf{I}_i = Ct_i + \mathbf{n} \cdot \mathbf{A}(1 - t_i)$, or equivalently:

$$t_i = \frac{\mathbf{n}}{C - \mathbf{n} \cdot \mathbf{A}} \cdot \mathbf{I} + \frac{\mathbf{n} \cdot \mathbf{A}}{C - \mathbf{n} \cdot \mathbf{A}}, \quad \forall i \in w. \tag{4.27}$$

This leads us to the local linear model (4.24).

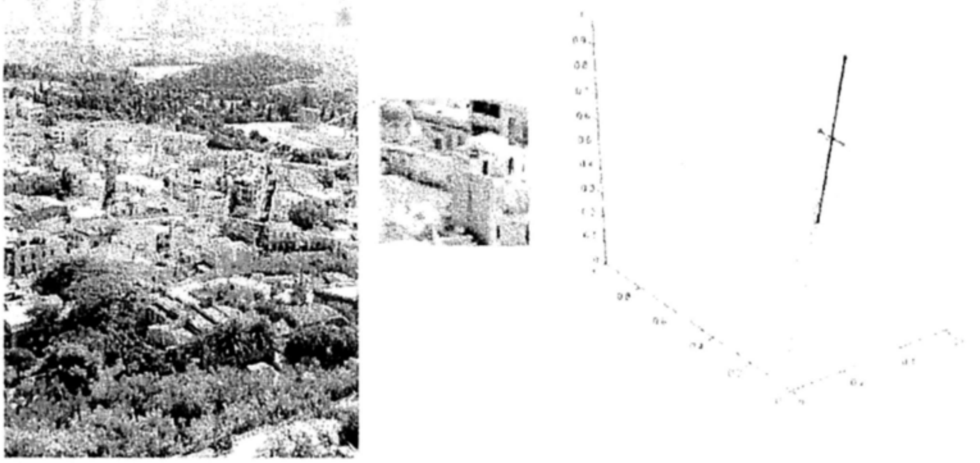


Figure 4.2: Color plane model.

The color plane model is more general than the color line model. Colors in a color line are certainly in a color plane, but not vice versa. The color plane assumption is easier to be satisfied, so we can use larger windows without breaking the assumption. In Fig. 4.2 we show an example. In the RGB space, we plot the colors in a 64×64 window. We also plot the principle components of these points. These colors can roughly be modeled by a plane.

If the color plane assumption is satisfied in large windows, the local linear model (4.24) is valid. In this case the quality of the soft matting step should be acceptable. Fig. 4.3 shows examples of haze removal results using different kernel sizes. We can see the recovered transmission maps t are slightly different: t contains more details and textures when the kernel is larger[4]. But the recovered scene radiance images $\mathbf{J}$ are almost visually identical. This is because the color plane model is valid for appropriately larger windows. In experiments, we find that $r = 8 \sim 16$ (for 600×400 images) introduces very little visible artifacts in most cases. This indicates that we can achieve about ×10 speed-up (compared with $r = 1$) without sacrificing quality.

4.1.4 Experiments

We verify the speed of our algorithm by experiments. We test many images and find similar conclusions, and here we take the image in Fig. 4.3 as an

[4] These details and textures can be safely smoothed by the bilateral filter [80].

example. In Fig. 4.4 we show the convergence speed of the CG algorithm with respect to kernel radius[5]. We can see that the algorithm converges much faster when the kernel gets larger. The number of iterations required to achieve err $= 10^{-10}$ are shown in Fig. 4.5. As we expect, the number of iterations is about $O(\frac{1}{r})$. As the running time of an iteration is constant regardless of r, the total running time is also shown in Fig. 4.5. The running time for this 600×400 image is about 10s for $r = 1$, 2s for $r = 8$, and 0.5s for $r = 32$.

The experimental results in Fig. 4.4 are difficult to obtain if we use conventional methods: the CG algorithm without our fast **Lp** algorithm would take over 300s to solve this linear system when $r = 8$, and 4000s when $r = 32$. This is one reason why previous methods [42, 84, 36] can only use $r = 1$.

4.1.5 Conclusion and Discussion

We have proposed a fast algorithm to solve the matting Laplacian matrix. Our algorithm is faster when the kernel is larger and the matrix is less sparse. which is against conventional theories.

Our algorithm is non-approximate with a given kernel size r. It provides us a chance to observe the results of larger kernels, which are almost unavailable in previous methods. The result of a large kernel (*e.g.*, $r > 1$) is different with the one of a small kernel (*e.g.*, $r = 1$), but is not necessarily degraded. In our alpha matting paper [31], we have found that an appropriately large kernel actually improves the quality.

Although our algorithm is particularly designed for the matting Laplacian matrix, a similar idea is expected to work in some other matrices used in computer vision/graphics problems. The focus is to reduce the running time in the matrix multiplication step (**Lp**) and to achieve speed-up by faster convergence.

The matrix-vector multiplication (**Lp**) is 2D image filtering. So our $O(N)$ time computation of **Lp** is actually a fast filtering algorithm. Inspired by this concept, we propose a novel filter called *guided filter* [32]. It is not only fast but also has very good quality, as we shall discuss below.

[5] We set $\lambda = 0.0001|w|^2$ in the linear system. A factor $|w|$ is to counteract the increased number of smoothness terms in the cost function (4.4). The other factor $|w|$ ensures the average intensity of the solution is almost unchanged, just as in many multigrid methods [7].

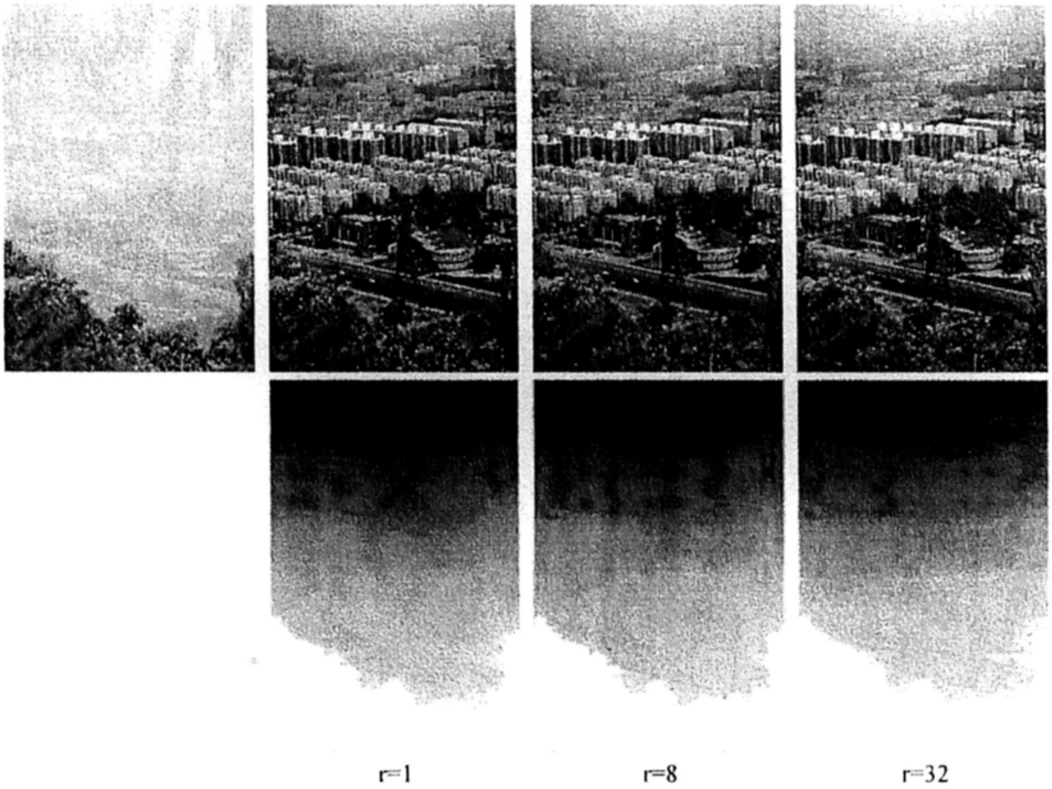


Figure 4.3: Haze removal results vs. kernel size. Top: input image and recovered scene radiance. Bottom: recovered transmission after soft matting.

4.2 Guided Image Filtering

We go back to the motivation of the soft matting: we expect to combine the pixel-wise constraints with spatial continuity concerns. Inspecting Fig. 3.7, we find that the refined transmission t has intensity like $\tilde{t}$, but has consistent edges with $\mathbf{I}$. It appears that the map $\tilde{t}$ undergoes a filtering process and becomes t, and the process appears to be "aware" of the edges in the image $\mathbf{I}$.

The solution to the soft matting (4.10) can be written as:

$$\mathrm{t} = \lambda(\mathrm{L} + \lambda\mathrm{U})^{-1}\tilde{\mathrm{t}}. \tag{4.28}$$

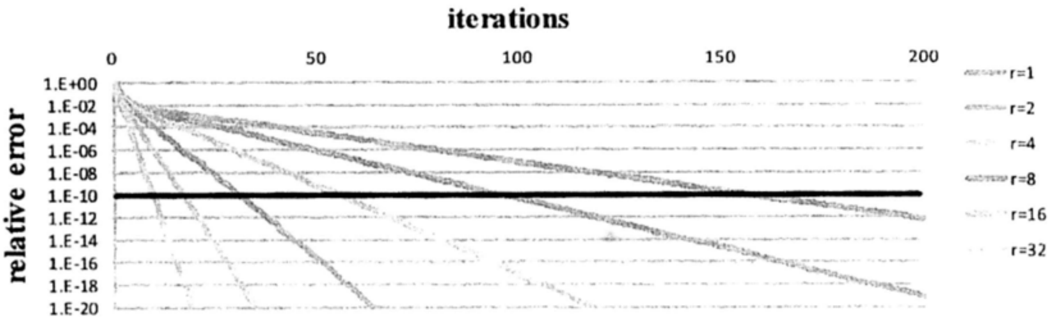


Figure 4.4: Convergence speed vs. kernel size. The errors are shown relative to the initial errors.

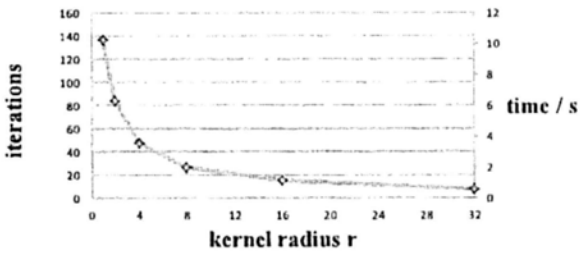


Figure 4.5: Iteration numbers and running time vs. kernel size. The iteration numbers correspond to the error $1e^{-10}$ in Fig. 4.4.

This is actually a *translation-variant* filtering process:

$$t_i = \sum_j W_{ij}\tilde{t}_j, \tag{4.29}$$

where the filtering kernel W is the matrix $\lambda(\mathrm{L} + \lambda\mathrm{U})^{-1}$ that is not explicitly computed. This equation indicates that the output t_i is a weighted average of $\tilde{t}_j$, where the weight W_{ij} is solely determined by the image **I**. The image **I** "guides" the filtering process by adjusting the weights (kernel), ensuring the edges of t are consistent with **I**.

In this section, we introduce a novel edge-aware filter, called *guided filter*, to replace the soft matting step. The weights of the filter are explicitly given, so we do not need to solve a linear system. We further propose a fast $O(N)$ time algorithm which is independent of the kernel size r. This enables real-time performance of the filter. Edge-aware filtering is an important and widely studied topic in computer vision/graphics. We show that the guided filter has many advantages over existing edge-aware filtering techniques, in terms of both speed and quality. We also demonstrate various applications of the guided filter including haze removal.

4.2.1 Related Works: Edge-aware Filtering

Filtering is one of the most important operations in computer vision/graphics. Simple linear translation-invariant (LTI) filters like Gaussian filter, Laplacian filter, and Sobel filter are widely used in image blurring/sharpening, edge detection, feature extraction and etc. [27]. The kernel of these filters are *explicitly* defined. LTI filters can also be designed implicitly: *e.g.*, solving a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to filtering the map $\mathbf{b}$ by $\mathbf{A}^{-1}$: $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, where the filtering kernel $\mathbf{A}^{-1}$ is not explicitly computed. An example of implicit LTI filters is the Poisson Equation used in high dynamic range (HDR) compression [20], image stitching [56], image matting [75], and image smoothing [6].

The kernels of LTI filters are spatially invariant and independent of any image content. But usually we hope the filtering process to "stop" so as to preserve edges. The bilateral filter, originated from [4, 71], named in [80], and generalized in [57], is perhaps the most widely used edge-aware filter. Its output at each pixel is a weighted average of the nearby input pixels, where the weights depend on the intensity/color similarities. It can smooth the image while preserving edges. Due to this nice property, it has been widely used in noise reduction [43], HDR compression [15], multi-scale detail decomposition [19], image abstraction [86], and etc. It is generalized to the joint bilateral filter in [57]: the weights are computed from another image rather than the filtering input. The joint bilateral filter is advantageous when the filtering input is not reliable to provide edge information, *e.g.*, when it is very noisy or is an intermediate map. The joint bilateral filter has been applied in flash/no-flash imaging [57], image unsampling [38], image deconvolution [90], and etc.

However, the bilateral filter[6] has some limitations. It has been noticed [15, 5, 17] that this filter may have the *gradient reversal* artifacts. The reason is that when a pixel (often on an edge) has very few similar pixels around it, the weighted average is unstable. Another problem of the bilateral filter is the speed. The brute-force implementation is in $O(Nr^2)$ time, which is very high when the kernel radius r is large. Paris *et al.*[55] propose an approximate

[6] We simply refer to bilateral/joint bilateral cases as the "bilateral filter" unless specified.

solution in a discretized space-color grid. This concept inspires $O(N)$ time algorithms [58, 88] based on histograms. Adams et al. [1] propose a tree-based algorithm for color images. All of these methods require quantization (approximation) to achieve satisfactory speed, but at the expense of quality degradation.

Edge-aware filtering can also be implicit, *e.g.*, through solving a linear system. The weighted least squares (WLS) filter in [17] adjusts the matrix affinities according to the image gradients. This method is able to produce a halo-free decomposition of the input image. But solving a linear system is relatively slow (compared with explicit filtering), as we discussed in the last section. The matting Laplcian matrix [42] and the Gaussian-weighted matrix [28] also exhibit edge-aware properties, although previously they were not developed for this purpose. In [8], a non-linear filtering process is proposed through L1-norm optimization. This filter is edge-aware, but the optimization is also time-consuming.

In sum, a fast and high quality explicit filter is still demanded in many edge-aware applications.

4.2.2 Algorithm

We propose the concept of "guided image filtering" before introducing our filter. *Guided image filtering* is a process that combines the information of two images, namely, a filtering input image (denoted as p) and a guide image (denoted as I), and generates one filtering output image (denoted as q). The filtering input p determines the colors, brightness, and tones of the filtering output q, whereas the guide image I determines the edges of q.

Guided image filtering involves the concept of edge-aware filtering. For example, in the case of flash/no-flash denoising (previously through the joint bilateral filter [57]), the output (q) is expected to have the tones of the no-flash noisy image (p), and be as sharp and clean as the flash image (I) (Fig. 4.6, 3rd row). In the case of single image texture smoothing (Fig. 4.6, 2nd row) or denoising, we can treat it as a special case of guided image filtering where the filtering input (p) and the guide (I) are identical, because the colors and the edge information are from the same image (but combined unequally). The guided image filtering also involves many other problems that are not viewed as edge-aware filtering before, like haze removal, image matting, and stereo vision [60]. For example, in our transmission refinement (previously through soft matting), the refined transmission map t has values close to the estimated $\tilde{t}$, and has edges consistent with the hazy image I. In this case the filtering input p is $\tilde{t}$, and the filtering output q is t (Fig. 4.6, 1st

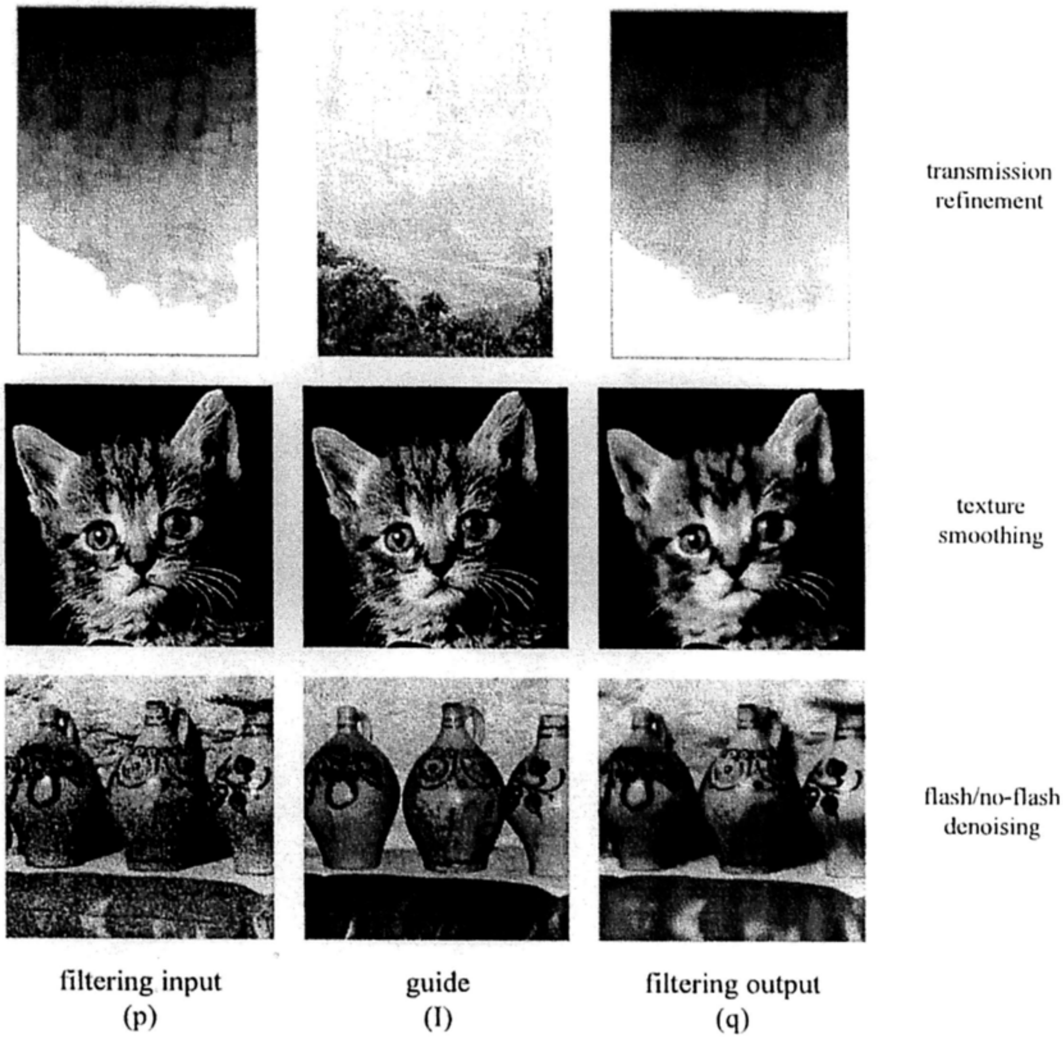


Figure 4.6: Guide image filtering.

row).

The concept of guided image filtering allows us to treat edge-aware filtering and other problems in a broader framework. In the following, we propose a novel filter called *guided filter* which was originally designed in our haze removal study but is later applied in many problems.

We are interested in a general linear *translation-variant* filtering process, which involves a guide image I, an input image p, and an output image q. Both I and p are given according to the application, and they can be identical. The filtering output at a pixel i is expressed as a weighted average:

$$q_i = \sum_j W_{ij}(I) p_j, \tag{4.30}$$

where i and j are pixel indexes. The filter kernel W_{ij} is a function of the guide image I, which is independent of p. This filter is linear with respect to p for a fixed I.

In the case of the bilateral filter [80] and the joint bilateral filter [57], the kernel W^{bf} is given by:

$$W^{\mathrm{bf}}_{ij}(I) = \frac{1}{K_i} \exp(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\sigma_{\mathrm{s}}^2}) \exp(-\frac{|I_i - I_j|^2}{\sigma_{\mathrm{r}}^2}). \tag{4.31}$$

where $\mathbf{x}$ is the pixel coordinate, and K_i is a normalizing parameter to ensure that $\sum_j W^{\mathrm{bf}}_{ij} = 1$. The parameters σ_{s} and σ_{r} adjust the spatial similarity and the range (intensity/color) similarity respectively. The images I and p are identical in the bilateral case [80], and are different in the joint bilateral case [57].

Definition: Guided Filter

Now we are ready to define the guided filter and its kernel. The key assumption of the guided filter is a local linear model between the guide I and the filtering output q. We assume that q is a linear transform of I in a window w_k centered at the pixel k:

$$q_i = a_k I_i + b_k, \forall i \in w_k, \tag{4.32}$$

where (a_k, b_k) are linear coefficients assumed to be constant in w_k. We use a square window of a radius r. This model is analogous to (4.4). We have seen this model can be generated for the haze imaging equation, but here we consider it as a generic case.

This local linear model ensures that q has an edge only if I has an edge, because $\nabla q = a\nabla I$. But we also hope the output q has similar colors or tones with the input p, so we seek a solution to (4.32) that minimizes the difference between q and p. Formally, we minimize the following cost function in the window w_k:

$$\begin{aligned} E(a_k, b_k) &= \sum_{i\in w_k} \left((p_i - q_i)^2 + \epsilon a_k^2\right) && (4.33) \\ &= \sum_{i\in w_k} \left((p_i - a_k I_i - b_k)^2 + \epsilon a_k^2\right). && (4.34) \end{aligned}$$

Here ϵ is for regularization, but we shall show that it is an important parameter controlling the smooth degree.

Though this cost function is similar with the one in (4.4), they are different in two aspects: (i) Eqn. (4.34) only concerns *one* window w_k, while Eqn. (4.4) concerns *all* windows simultaneously; and (ii) in (4.34) inside the quadratic term is the filtering *input* image p , while in Eqn. (4.4) inside this term is the *output* image (the transmission t to be solved). These two differences leads to two benefits: (i) the optimization here is *locally* based instead of *globally*, enabling a very fast algorithm; and (ii) the output (q) is not contained in the optimization process (note a_k and b_k are the optimization targets), avoiding a huge linear system.

The solution to (4.34) can be given by linear regression [14]:

$$\begin{aligned} a_k &= \frac{\mathrm{cov}_k(I, p)}{\sigma_k^2 + \epsilon} && (4.35) \\ b_k &= \bar{p}_k - a_k \mu_k. && (4.36) \end{aligned}$$

Here, μ_k and σ_k^2 are the mean and variance of I in w_k, $\bar{p}_k$ is the mean of p in w_k, and $\mathrm{cov}_k(I, p)$ is the covariance of I and p inside the window k. These two equations appear similar with (4.13) and (4.14).

Next we apply the linear model to all local windows in the entire image. Ideally, the filtering output is given by $q_i = a_k I_i + b_k$ (4.32) when (a_k, b_k) has been computed. But a pixel i is covered by many local windows w_k, so the value of q_i given by (4.32) may change when it is computed in different windows. A simple strategy is to average all the possible values of q_i. So after computing (a_k, b_k) for all the windows w_k in the image, we compute the filter output by:

$$q_i = \frac{1}{|w|} \sum_{k:i \in w_k} (a_k I_i + b_k) \tag{4.37}$$

$$= \bar{a}_i I_i + \bar{b}_i \tag{4.38}$$

where $\bar{a}_i = \frac{1}{|w|} \sum_{k \in w_i} a_k$ and $\bar{b}_i = \frac{1}{|w|} \sum_{k \in w_i} b_k$. This equation is analogous to (4.15) (but without subtraction).

With this modification, the relation $\nabla q = a \nabla I$ no longer holds, because the linear coefficients $(\bar{a}_i, \bar{b}_i)$ vary spatially. But since $(\bar{a}_i, \bar{b}_i)$ are the output of an average filter, their gradients should be much smaller than that of I near strong edges. In this sense we can still have $\nabla q \approx \bar{a} \nabla I$, meaning that strong edges in I can still be maintained in q.

Algorithm 4 O(N) time algorithm for guided filter

```
1.
    $a_k := \frac{\text{cov}_k(I, p)}{\sigma_k^2 + \epsilon}$        (4.35)
2.
    $b_k := \bar{p}_k - a_k \mu_k$        (4.36)
3.
    $q_i := \frac{1}{|w|} \left( (\sum_{k \in w_i} a_k) I_i + (\sum_{k \in w_i} b_k) \right)$        (4.37)
```

The whole algorithm of the guided filter is in Algorithm 4. We point out that the relationship among I, p, and q given by (4.35) to (4.37) are indeed in the form of "a filter" (4.30). In fact, a_k in (4.35) can be rewritten as a weighted sum of p: $a_k = \sum_j W^a_{kj}(I) p_j$. For the same reason, we also have $b_k = \sum_j W^b_{kj}(I) p_j$ from (4.36) and $q_i = \sum_j W^q_{ij}(I) p_j$ from (4.37). It can be proven (similar to the proof in Appendix B.1) that the kernel weights can be *explicitly* expressed by:

$$W_{ij}(I) = \frac{1}{|w|^2} \sum_{k:(i,j)\in w_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}). \tag{4.39}$$

Some further computations show that $\sum_j W_{ij}(I) = 1$. No extra effort is needed to normalize the weights.

As the guided filter is an explicit filter, we may first compute the kernel by (4.39) and then compute the output by the filtering process (4.30), without solving any linear system. This leads to an $O(Nr^2)$ algorithm where r is the window radius. However, the guided filter can be directly computed from the Algorithm 4. All the operations here can be computed in $O(N)$ time regardless of the kernel size, just as in Algorithm 3. This fast filtering algorithm is non-approximate.

We can generalize the guided filter to color/multichannel images. The generalization to a multichannel input image $\mathbf{p}$ is straightforward: we can simply treat each channel separately. So we focus on the case when the guide $\mathbf{I}$ is a multichannel image. The local linear model (4.32) becomes:

$$q_i = \mathbf{a}_k \cdot \mathbf{I}_i + b_k, \forall i \in w_k, \tag{4.40}$$

where $\mathbf{a}_k$ is a 3×1 vector. The computation of the resulting guided filter is shown in Algorithm 5. Here Σ_k is a 3×3 covariance matrix of $\mathbf{I}$ in the window k, and $\text{cov}_k(\mathbf{I}, p)$ is a 3×1 covariance vector. This algorithm is still $O(N)$ time. The state-of-the-art method for multichannel bilateral filtering is $O(N \log N)$ time [1]. So the guided filter is expected to be faster than the bilateral filter in multichannel cases. We shall show this in the experiment.

Using a multichannel image as a guide $\mathbf{I}$ is necessary in some applications, including haze removal and alpha matting. For example, the hazy image $\mathbf{I}$ is a color image and the estimated transmission $\tilde{t}$ is a gray-scale image, so we have to filter a single-channel image under the guidance of a three-channel image. We cannot treat each channel separately.

Algorithm 5 O(N) time algorithm for guided filter (color image **I**)

```
1.
    $\mathbf{a}_k := (\Sigma_k + \epsilon \mathrm{U})^{-1} \mathrm{cov}_k(\mathbf{I}, p)$    (4.35)
2.
    $b_k := \bar{p}_k - \mathbf{a}_k \cdot \mu_k$    (4.36)
3.
    $q_i := \frac{1}{|w|} \left( (\sum_{k \in w_i} \mathbf{a}_k) \cdot \mathbf{I}_i + (\sum_{k \in w_i} b_k) \right)$    (4.37)
```

4.2.3 Properties

Edge-aware Filtering

The guided filter is originated from a local linear model. We explain why it is edge-aware.

Fig. 4.7 (top) shows an example of the guided filter with various sets of parameters. We can see its edge-aware filtering behavior. The reasons are as following. Consider the case that $I = p$. It is clear that if $\epsilon = 0$, then the solution to (4.34) is $a_k = 1$ and $b_k = 0$. If $\epsilon > 0$, we can consider two cases:

Case 1: "Flat patch". If the image I is almost constant in w_k, then (4.34) is solved by $a_k = 0$ and $b_k = \bar{p}_k$;

Case 2: "Edge / high variance". If the image I changes greatly within w_k and we have:

$$\mathrm{cov}_k(I, p) = \mathrm{var}_k(I, I) = \sigma_k^2 \gg \epsilon, \tag{4.41}$$

then a_k is about 1 and b_k is about 0.

When (a_k, b_k) are averaged to get $(\bar{a}_i, \bar{b}_i)$ and then combined in (4.37) to get the output, we see that if a pixel is in the middle of a "flat patch" area, its value becomes the average of the pixels nearby ($a \approx 0$, $b \approx \bar{p}$, $q \approx \bar{p}$); if it is in the middle of an "edge/high variance" area, then its value is unchanged ($a \approx 1$, $b \approx 0$, $q \approx I = p$).

More specifically, the criterion of a "flat patch" or an "edge / high variance" one is given by the parameter ϵ. The patches with variance (σ^2) much smaller than ϵ are smoothed, whereas those with variance much larger than ϵ are preserved. The effect of ϵ in the guided filter is similar with the range variance σ_r^2 in the bilateral filter (4.31). Both parameters determine "what is

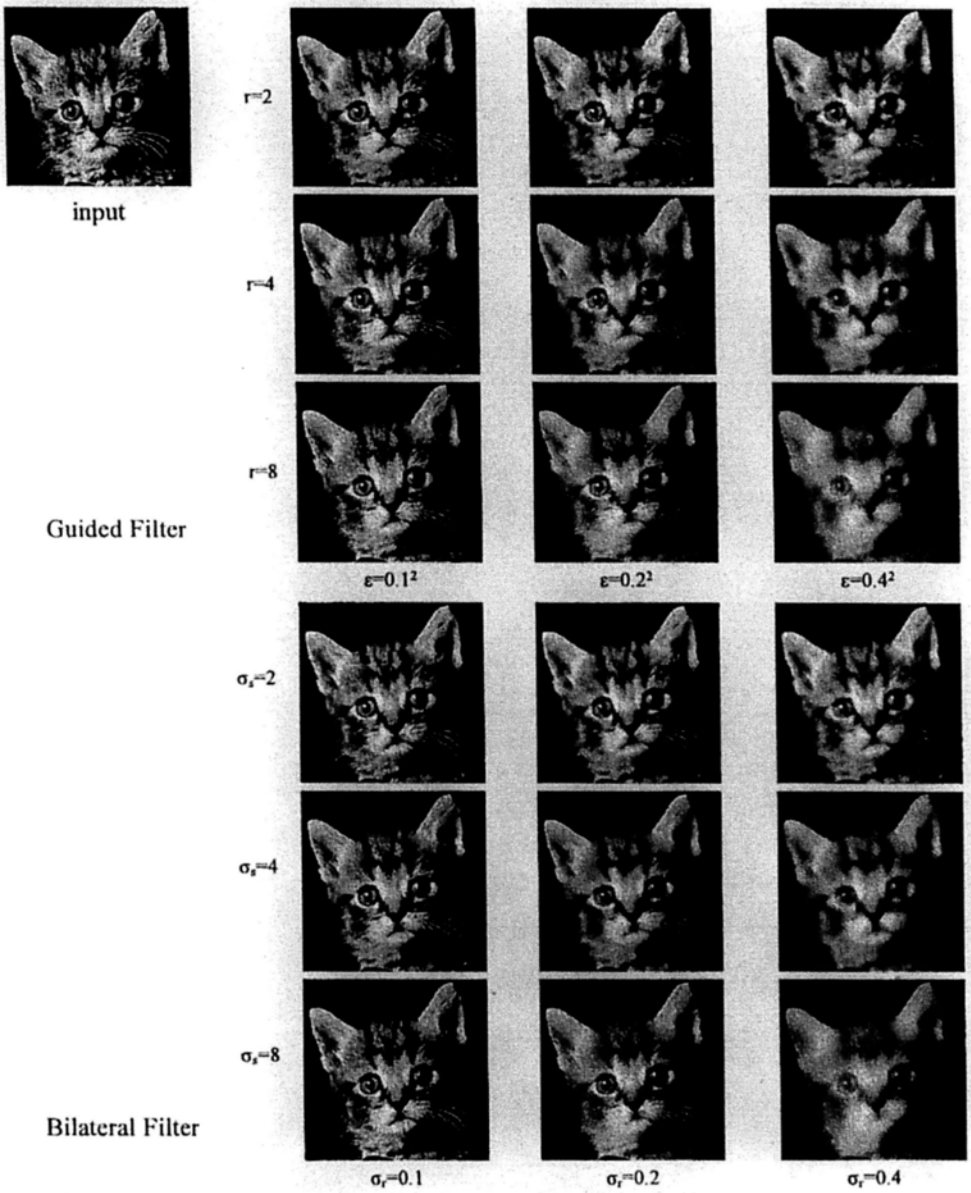


Figure 4.7: Filtered images of a gray-scale input. In this example the guidance I is identical to the input p. The input image has intensity in [0, 1]. The input image is from [80].

an edge or what is a high variance patch that should be preserved". Fig. 4.7 (bottom) shows the bilateral filter results as a comparison. We see that the edge-aware filtering results are visually similar with corresponding parameters.

The edge-aware smoothing property can also be understood by investigating the filter kernel (4.39):

$$W_{ij}(I) = \frac{1}{|w|^2} \sum_{k:(i,j)\in w_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}). \tag{4.39}$$

We find the weight is large when the pixels are on the same side of an edge, and is small otherwise. The property is similar with the bilateral filter (4.31). Take an ideal step edge of a 1-D signal as an example (Fig. 4.8). The terms $I_i - \mu_k$ and $I_j - \mu_k$ have the same sign (+/-) when I_i and I_j are on the same side of an edge, and have opposite signs when the two pixels are on different sides. Suppose $\sigma_k^2 \gg \epsilon$ ("edge / high variance"). In (4.39) the term $1 + \frac{(I_i-\mu_k)(I_j-\mu_k)}{\sigma_k^2+\epsilon}$ is about zero when the two pixels (i and j) are on different sides; this term increases to about 2 when the two pixels are on the same sides[7]. This means that the pixels across an edge are almost not averaged together.

We can also understand the smoothness controlling of the parameter ϵ from the kernel (4.39). When $\sigma_k^2 \ll \epsilon$ ("flat patch"), the kernel becomes $W_{ij}(I) = \frac{1}{|w|^2} \sum_{k:(i,j)\in w_k} 1$: this is a low-pass filter that biases neither side of an edge. So the edge-aware filtering ability is weakened when ϵ increases.

Fig. 4.9 shows two examples of the kernel shapes in real images. In the top row are the kernels near a step edge. Like the bilateral kernel, the guided filter's kernel assigns negligible weights to the pixels on the opposite side of the edge. In the bottom row are the kernels in a patch with small scale textures. Both filters average almost all the nearby pixels together, and behave like low-pass filters.

Gradient-preserving Property

Though the guided filter is an edge-aware filter like the bilateral filter, it avoids the "gradient reversal" artifacts [15, 5, 17]. These artifacts of the bilateral filter may appear when the filtering process is for detail enhancement/extraction.

Fig. 4.10 shows a 1-D illustration for detail enhancement. Given the input signal (black), its edge-aware smoothed version is used as a *base layer* (red). The difference between the input signal and the base layer is treated as the *detail layer* (blue). It is magnified to boost the details. The enhanced signal

[7] This is because $(I_i - \mu_k)(I_j - \mu_k) \approx -\sigma_k^2$ when the two pixels are on different sides, and $\approx \sigma_k^2$ when the two pixels are on the same side. See Fig. 4.8

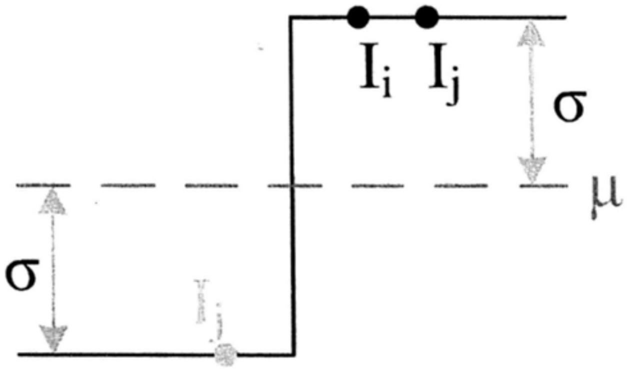


Figure 4.8: 1-D example of an ideal step edge. For a window that exactly center on the edge, the variables μ and σ are as indicated.

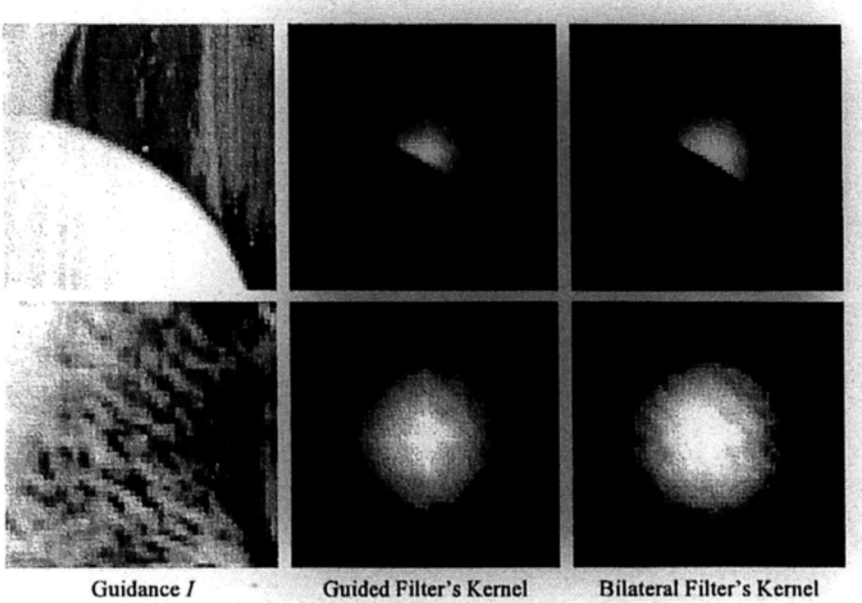


Figure 4.9: Filter kernels. Top: a step edge (guided filter: $r = 7, \epsilon = 0.1^2$, bilateral filter: $\sigma_s = 7, \sigma_r = 0.1$). Bottom: a textured patch (guided filter: $r = 8, \epsilon = 0.2^2$, bilateral filter: $\sigma_s = 8, \sigma_r = 0.2$). The kernels are centered at the pixels denote by the red dots.

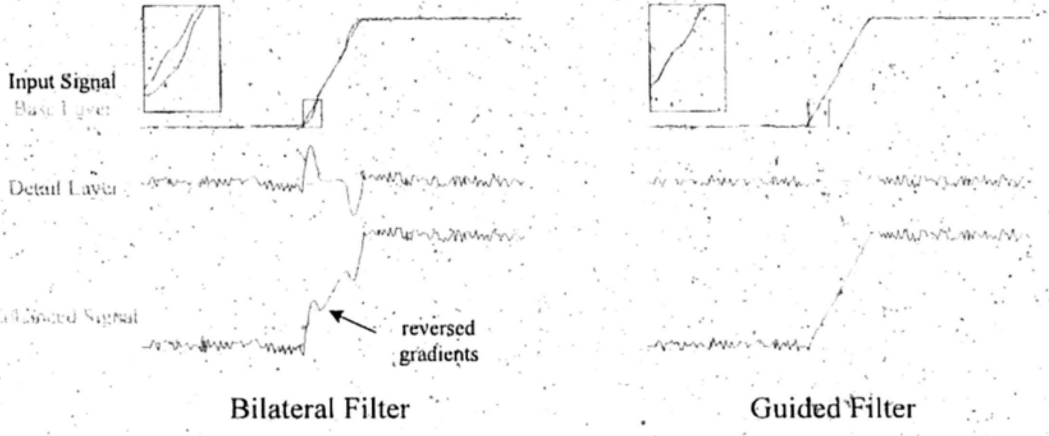


Figure 4.10: 1-D illustration for detail enhancement.

(green) is the combination of the boosted detail layer and the base layer. An comprehensive description can be found in [15].

For the bilateral filter (Fig. 4.10 left), the base layer is not consistent with input signal at the edge pixels (see the zoom-in). This is because when a pixel has very few similar pixels nearby, and the Gaussian weights (4.31) are all very small and thus unstable. As a result, the detail layer has great fluctuations, and the recombined signal has *reversed* gradients as shown in Fig. 4.10.

On the other hand, the guided filter (Fig. 4.10 right) better preserves the gradient: The gradient of the base layer is $\nabla q \approx \bar{a}\nabla I$ near the edge, and the gradient of the detail layer is about $(1-\bar{a})\nabla I$. Only the magnitude of the edge is scaled; the profile of the edge is almost unchanged. In the case of detail enhancement ($I = p$), the guided filter never reverses the gradient, because $\bar{a}$ and $1 - \bar{a}$ are always non-negative (see (4.35)) when $I = p$.

Relation to the Matting Laplacian Matrix

Both the guided filter and the matting Laplacian matrix are motivated by a local linear model, although their optimization processes are very different. Next we show the mathematical relation between them. This discovery inspires new applications of the guided filter.

Comparing the guided filter's kernel (4.39) with the matting Laplacian matrix (4.9), we find following relation between them:

$$\mathrm{L}_{ij} = |w|(\delta_{ij} - \mathrm{W}_{ij}), \tag{4.42}$$

Ignoring the scaling $|w|$, we find that the matting Laplacian matrix L is actually a "high-pass" version of the "low-pass" guided filter. Indeed, for any image p we have:

$$\mathbf{L}\mathbf{p} \propto (\mathbf{p} - \mathbf{W}\mathbf{p}), \tag{4.43}$$

in which the right hand side is the detail layer of p.

Elad [16] has noticed a similar relation between the bilateral filter and a Gaussian-weighted Laplacian matrix [85, 41, 28], and shown that filter is a one-iteration approximate solution to the linear system. Following this way, we find this conclusion is also valid for the guided filter and the matting Laplacian matrix.

The matrix form of (4.42) is:

$$\mathrm{L} = |w|(\mathrm{U} - \mathrm{W}), \tag{4.44}$$

where U is a unit matrix of the same size as L. In our soft matting, the transmission t is obtained by solving the linear system in (4.10):

$$(\mathrm{L} + \lambda\mathrm{U})\mathbf{t} = \lambda\tilde{\mathbf{t}}. \tag{4.10}$$

We rewrite this equation as

$$(\mathrm{L} + \Lambda)\mathbf{t} = \Lambda\tilde{\mathbf{t}}, \tag{4.45}$$

where Λ is a diagonal matrix. This equation suggests we can assign a variant λ_i to each pixel.

Now we approximate the solution by one iteration of the Jacobi linear solver [63]. We decompose W into a diagonal part W_d and an off-diagonal part W_o: $\mathrm{W} = \mathrm{W}_d + \mathrm{W}_o$. According to (4.44) and (4.45), we have:

$$(|w|\mathrm{U} - |w|\mathrm{W}_d - |w|\mathrm{W}_o + \Lambda)\mathbf{t} = \Lambda\tilde{\mathbf{t}} \tag{4.46}$$

Notice that only W_o is off-diagonal in this equation. Using $\tilde{t}$ as the initial guess of t, we compute one Jacobi iteration (see [63]):

$$\begin{aligned}
\mathbf{t} &\approx (|w|\mathrm{U} - |w|\mathrm{W}_d + \Lambda)^{-1}(|w|\mathrm{W}_o + \Lambda)\tilde{\mathbf{t}} && (4.47)\\
&= (\mathrm{U} - \mathrm{W}_d + \frac{\Lambda}{|w|})^{-1}(\mathrm{W} - \mathrm{W}_d + \frac{\Lambda}{|w|})\tilde{\mathbf{t}} && (4.48)\\
&= (\mathrm{U} - \mathrm{V})^{-1}(\mathrm{W} - \mathrm{V})\tilde{\mathbf{t}} && (4.49)
\end{aligned}$$

where $\mathrm{V} = \mathrm{W}_d - \frac{\Lambda}{|w|}$ is a diagonal matrix. In (4.49), all the matrices except W are diagonal. Since a diagonal matrix is actually a point-wise operation, the only matrix multiplication we need to compute is $\mathrm{W}\tilde{\mathbf{t}}$. The multiplication $\mathrm{W}\tilde{\mathbf{t}}$ is indeed a guided filter operation:

$$(\mathrm{W}\tilde{t})_i = \sum_j W_{ij}\tilde{t}_j \tag{4.50}$$

To further simplify (4.49), we can let the matrix $\mathrm{V} = \mathrm{W}_d - \frac{\Lambda}{|w|} = 0$, or equivalently:

$$\Lambda_{ii} = |w|W_{ii} = \frac{1}{|w|}\sum_{k\in w_i}(1 + \frac{(I_i - \mu_k)^2}{\sigma_k^2 + \epsilon}). \tag{4.51}$$

The expectation value of Λ_{ii} in (4.51) is about 2. This implies that the data term is loosely constrained. Using this equation, (4.49) is simplified as:

$$\mathbf{t} \approx \mathrm{W}\tilde{\mathbf{t}}. \tag{4.52}$$

Only the guided filter is remained.

Eqn.(4.52) means the guided filter is a one-iteration approximate solution to the linear system. To obtain an accurate result, we expect the initial guess $\tilde{t}$ to be reasonably good. We find that this condition is satisfied in our haze removal problem and a modified alpha matting problem, which we shall discuss in the experiment section.

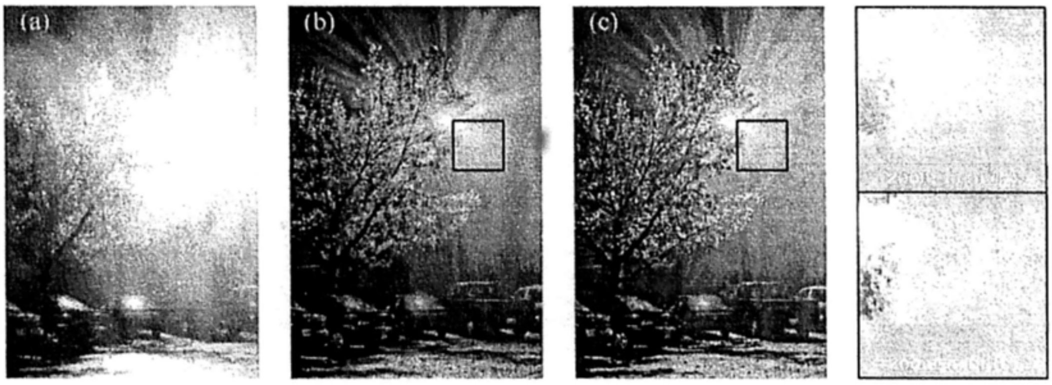


Figure 4.11: Comparison between O(N) guided filter and O(N) bilateral filter. (a) Input HDR image (32-bit floating number, displayed by linear scaling). (b) HDR-Compressed image using the O(N) bilateral filter in [58] (64 bins). (c) HDR-Compressed image using the guided filter. *Please view this figure in the electronic version.*

4.2.4 Experiments and Applications

In the following, we first study the running time of the guided filter by experiments. Then we show various applications of this filter, including the haze removal problem.

Running Time

We experiment the running time in a laptop with a 2.0Hz Intel Core 2 Duo CPU. The time complexity of the filter is O(N) in the number of the pixels N, so we only report the running time per megapixel (Mp) image. For the gray-scale guided filter (Algorithm 4), the running time is 80 milliseconds (ms) per megapixel. As a comparison, Porikli's [58] O(N) time approximate bilateral filter takes 40ms/Mp (using 32-bin histograms), and 80ms/Mp (64-bin). A 64-bin histogram means that the images are only quantized into 64 different values. On the contrary, the guided filter is non-approximate and is suitable for continuous values (float/double). In Fig. 4.11, we show an example of HDR compression (we shall explain the application later). Porikli's O(N) bilateral filter exhibits apparent quantization artifacts, while the guided filter does not. Yang *et al.*[88] improve Porikli's algorithm and reduce the quantization artifacts without greatly increasing the number of bins, but at the price of more running time. This algorithm takes 1.2s/Mp (8-bin, code from the website of [88]).

For multichannel cases (Algorithm 5), the guided filter takes 0.3s to process per megapixel RGB image. For comparison, the state-of-the-art method for multichannel bilateral filter [1] takes about 100s to process a 10Mp image as reported, *i.e.*, 10s/Mp on average.

The algorithms for the guided filter are mostly point-wise operations, so it can be readily adapted in GPU implementation. Though we have not yet made this attempt, a recent paper [60] reports that their GPU implementation for our guided filter achieves 5ms/Mp (multichannel cases).

Both the CPU and GPU implementations of the guided filter are very fast, enabling real-time performance in many cases. Next we see the quality and the application of this filter in various problems.

Haze Removal

We use the guided filter to replace the soft matting step in haze removal. To obtain a good result through filtering, we need the initial estimation $\tilde{t}$ to be accurate enough. Recall that $\tilde{t}$ is computed by (3.14) based on the dark channel prior:

$$\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \min_{c} \frac{I_c(\mathbf{x}')}{A_c}. \tag{3.14}$$

We notice that the minimum filter ($\min_{\mathbf{x}' \in \Omega(\mathbf{x})}$) *dilates* the nearer objects (see Fig.4.12 (b)). This effect is known as morphological filtering in image processing [27]. To counteract this effect, we may *erose* the map by an extra maximum filter. Formally, we compute $\tilde{t}$ by:

$$\begin{aligned} \varphi(\mathbf{x}) &= \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \min_{c} \frac{I_c(\mathbf{x}')}{A_c} \\ \tilde{t}(\mathbf{x}) &= 1 - \max_{\mathbf{x}' \in \Omega(\mathbf{x})} \varphi(\mathbf{x}'). \end{aligned} \tag{4.53}$$

An example is in Fig.4.12 (c). The estimated transmission has a more consistent profile with the input image.

Now we are ready to applied the guided filter. Written in a matrix form, the refined transmission t is simple given by (4.52):

$$\mathbf{t} \approx \mathbf{W}\tilde{\mathbf{t}}. \tag{4.52}$$

The computation is given in Algorithm 5, because the hazy image $\mathbf{I}$ is mul-

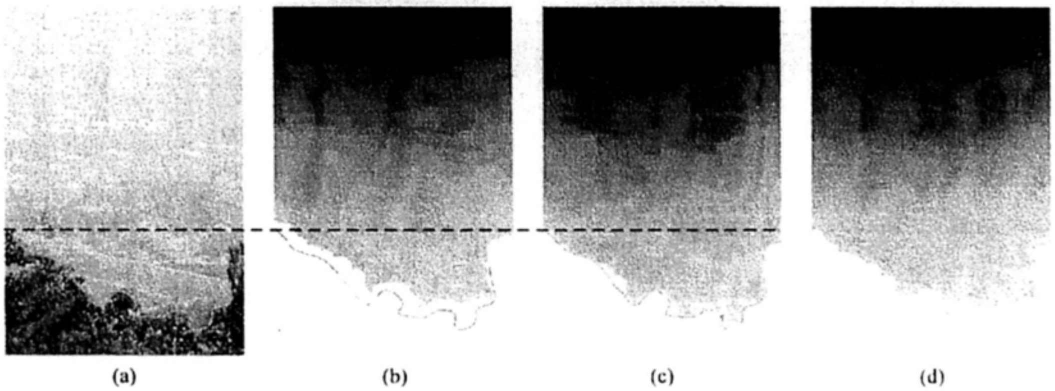


Figure 4.12: Dilation and erosion. (a) Input hazy image. The red curve approximately outlines the profile of the trees. (b) Estimated transmission by (3.14). The trees appear "dilated". (c) Estimated transmission after erosion (4.53). The profile is more consistent. (d) Guided-filtered result of (c).

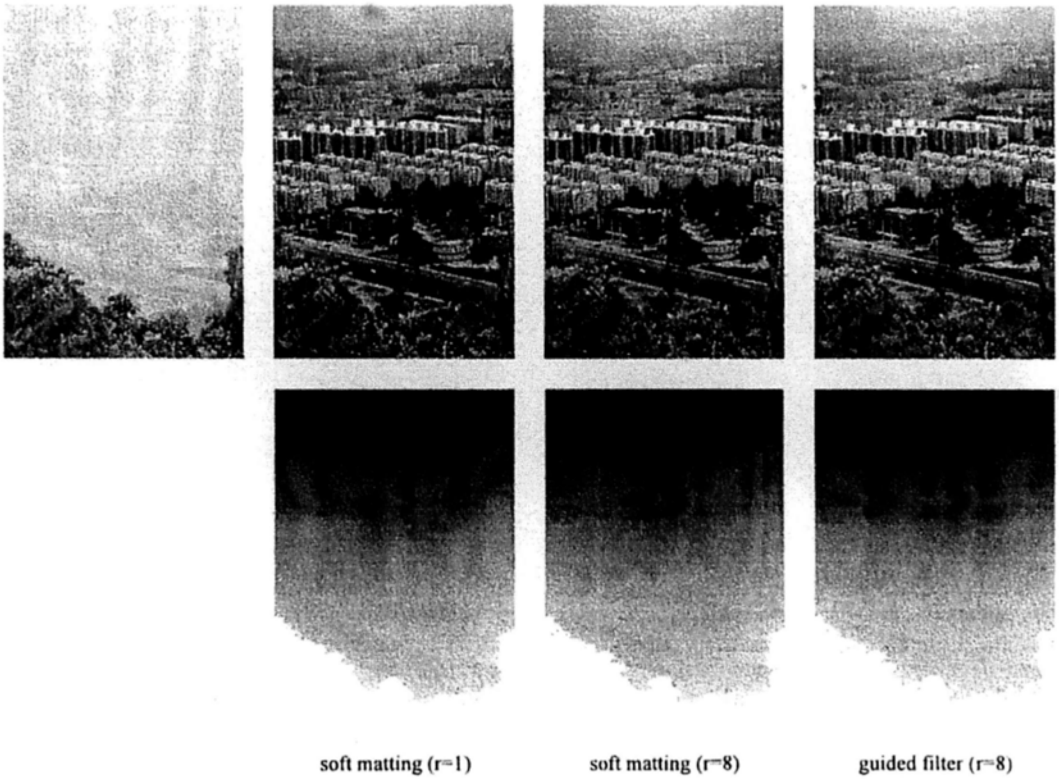


Figure 4.13: Comparisons of soft matting (small/large kernel) and guided filter in haze removal.

tichannel. In all the following haze removal experiments, the radius r of the window w in the guided filter is set as $\min(wid, hei)/50$, where wid and hei are the width and height of the image. For example, we set $r = 8$ for 600×400 images. The parameter ϵ is 0.0001.

Fig.4.12 (d) shows the resulting transmission t obtained by the guided filter. In Fig.4.13 we compare the soft matting (small/large kernel) and the guided filter. As we can see, the recovered scene radiance images (Fig.4.13 top) are almost visually identical. On the refined transmission maps, the soft matting (large kernel) and the guided filter generate visually similar results. This is as we expect, because the guided filter is an approximate solution to the linear system. More examples of haze removal using the guided filter are in Fig.4.14.

For this 600×400 image, the soft matting takes 10s using a small kernel ($r = 1$), and 2s using a large kernel ($r = 8$). The guided filter takes only 0.080s, which is much faster with no obvious quality degradation. In general, our algorithm takes about 0.350s/Mp in haze removal, with 0.3s for the guided filter, and 0.05s for the min/max filters and other point-wise operations.

The efficiency of our algorithm greatly facilitates haze removal in videos. We process each frame individually (except that we smooth the atmospheric light in the temporal domain to avoid sudden color changes). Fig. 4.15 shows some frames in two video sequences. The entire videos are available in my website [8].

Detail Enhancement and HDR Compression

We have discussed the usage of an edge-aware filter for detail enhancement. Fig. 4.16 shows an example for real images. We compare between the guided filter and the bilateral filter. Though both filters can enhance the details, the bilateral filter exhibit the gradient reversal artifacts (see the zoom-in patches).

The method for high dynamic range (HDR) compression [15] is similar to detail enhancement. But we need to scale both the base layer and the detail layer and then recombine them. Fig. 4.17 shows an example for HDR Compression. Again, we see gradient reversal artifacts in the result of the bilateral filter.

[8] http://personal.ie.cuhk.edu.hk/~hkm007/cvpr09/video.rar

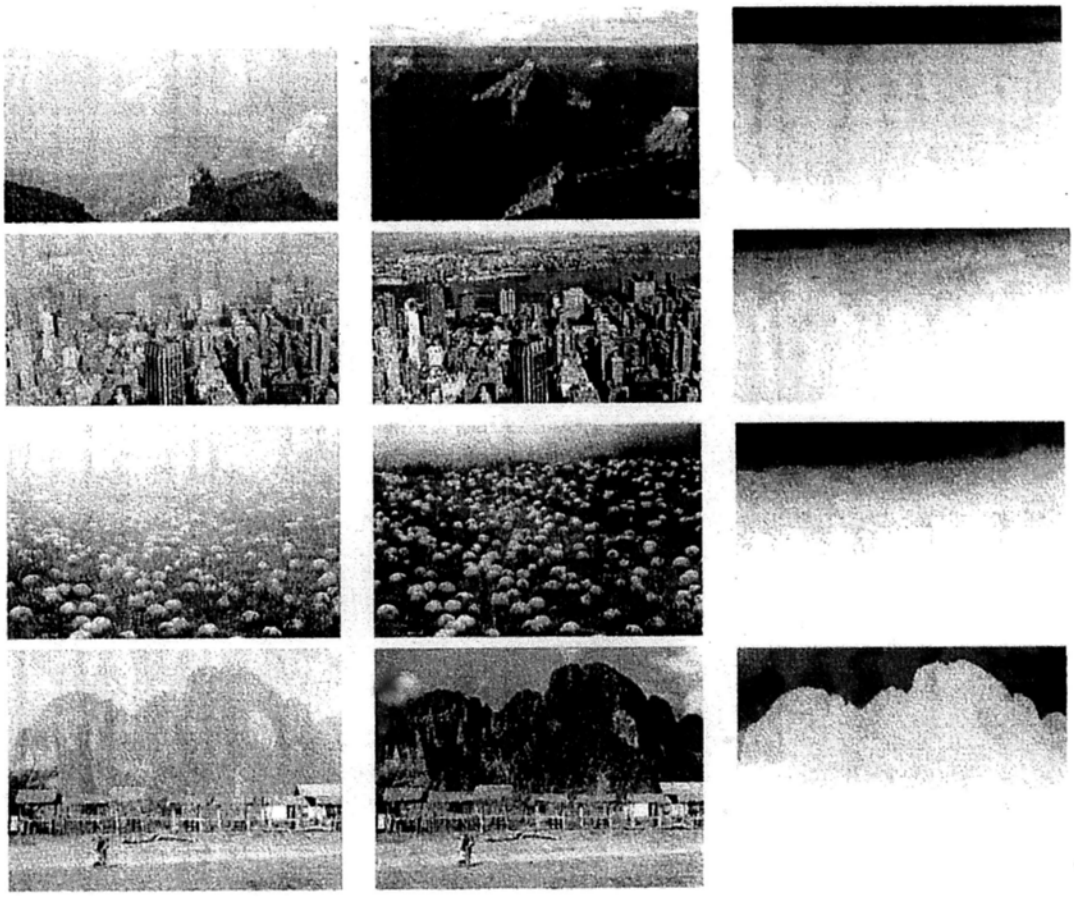


Figure 4.14: More examples of haze removal using the guided filter.

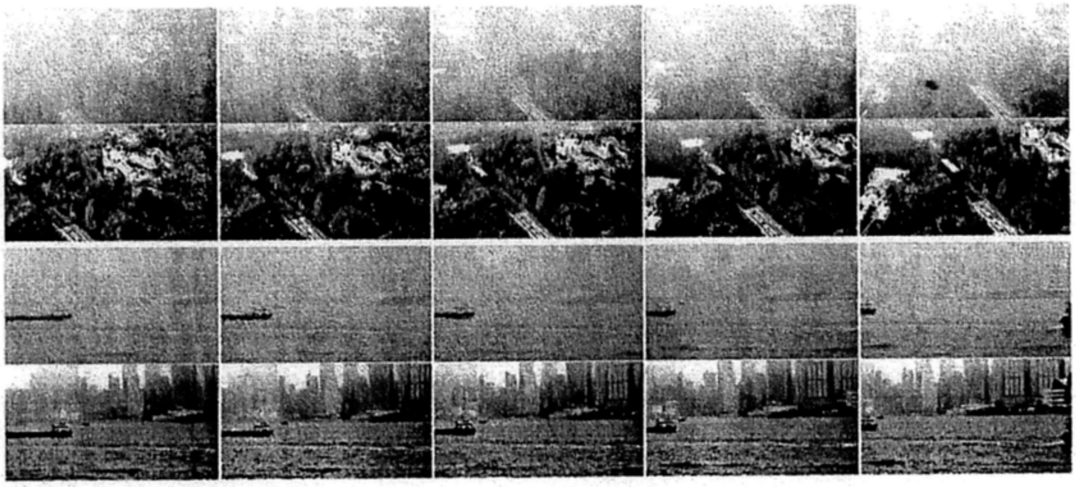


Figure 4.15: Video examples of haze removal using the guided filter.

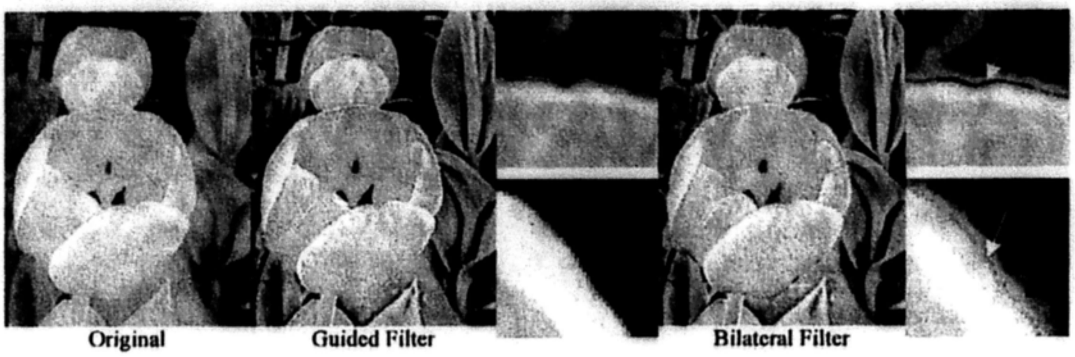


Figure 4.16: Detail enhancement. The parameters are $r = 16$, $\epsilon = 0.1^2$ for the guided filter, and $\sigma_s = 16$, $\sigma_r = 0.1$ for the bilateral filter. The detail layer is boosted ×5.

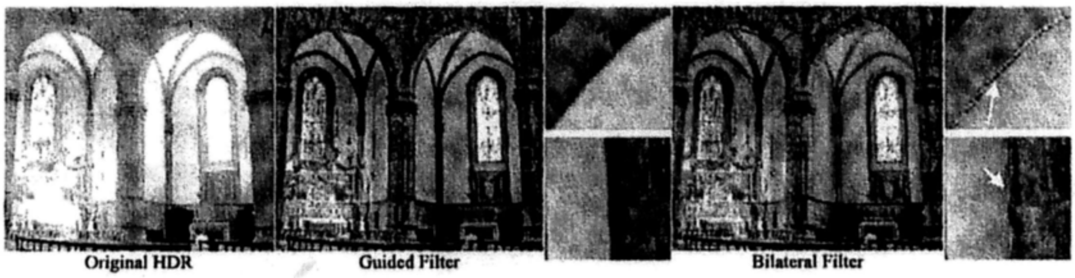


Figure 4.17: HDR compression. The parameters are $r = 15$, $\epsilon = 0.12^2$ for the guided filter, and $\sigma_s = 15$, $\sigma_r = 0.12$ for the bilateral filter.

Flash/no-flash denoising

In [57] it is proposed to denoise a no-flash image (p) under the guidance of its flash version (I). Fig. 4.18 shows a comparison between using the joint bilateral filter [57] and our guided filter. Both filters can suppress the noise, but the gradient reversal artifacts are noticeable in the joint bilateral case.

Alpha Matting/Guided Feathering

We apply the guided filter in a modified alpha matting problem, which we name *guided feathering*: given a binary mask of the foreground object, we refine the hairy/blurry boundaries and make it appear an alpha matte (Fig. 4.19). The binary mask, obtained by any segmentation method (*e.g.*[62]), is used as the filtering input p. The guide $\mathbf{I}$ is the image of the object.

A similar function called "Refine Edge" can be found in the commercial

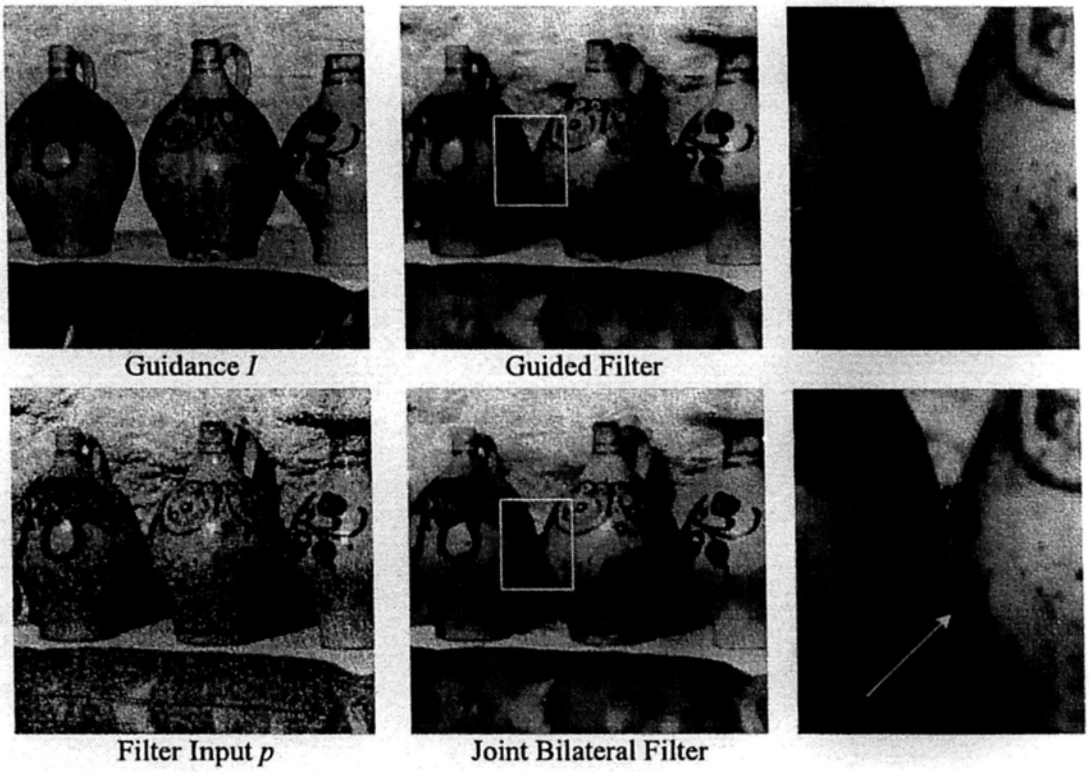


Figure 4.18: Flash/no-flash denoising. The parameters are $r = 8$, $\epsilon = 0.2^2$ for the guided filter, and $\sigma_s = 8$, $\sigma_r = 0.2$ for the joint bilateral filter.

software Adobe Photoshop CS4. We can also compute an accurate matte by solving the matting Laplacian matrix as in the "closed-form matting" method [42]. In Fig. 4.19 we compare our result with the Photoshop Refine Edge function and the closed-form method. Our result is visually comparable with the closed-form method and better than the Photoshop's. Both our method and Photoshop provide fast feedback (<1s) for this 6-mega-pixel image, while the closed-form solution takes about two minutes to solve the linear system.

Joint Upsampling

The application called joint upsampling [38] is to upsample an image under the guidance of another image. Taking the application of colorization [41] as an example. A gray-scale image is colorized through an optimization process. To reduce the running time, the chrominance channels are computed at a coarse resolution and upsampled under the guidance of the full resolution

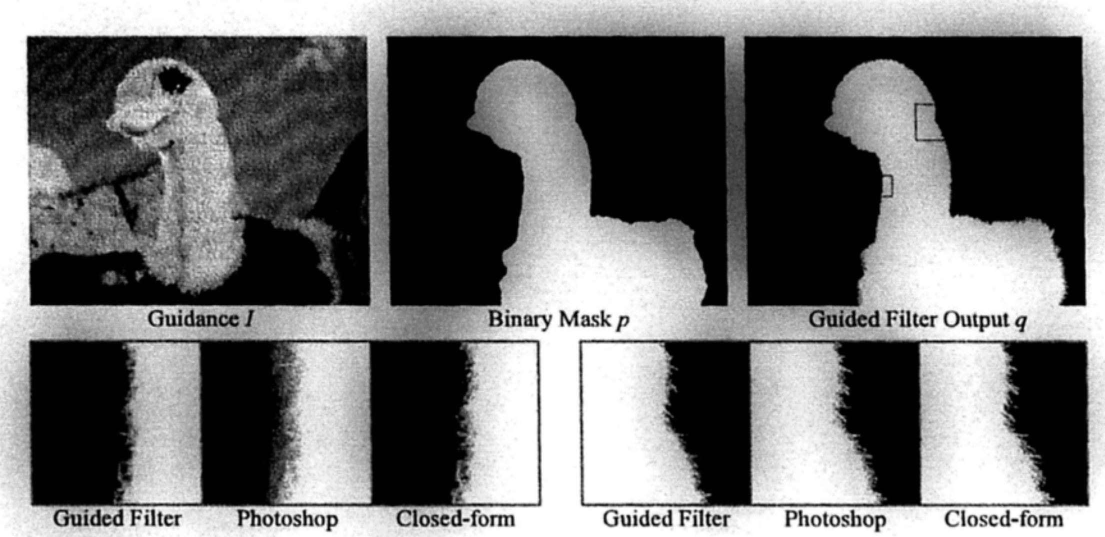


Figure 4.19: Alpha Matting/Guided Feathering. A binary mask p is filtered under the guidance of I. In the zoom-in patches, we compare with the Photoshop Refine Edge function and the closed-form matting. For closed-form matting, we erode and dilate the mask to obtain a trimap. The parameters for the guided filter are $r = 60$, $\epsilon = 10^{-6}$.

gray-scale image. This upsampling process is through the joint bilateral filter in [38], but we also test the guided filter. The results are visually comparable (Fig. 4.20), but our guided filter is faster than the non-approximate joint bilateral filter.

4.2.5 Conclusion

In this section we propose a novel guided filter. Originated from a local linear model, this filter can be used to replace the soft matting step and lead to real-time performance. We also show the edge-aware and gradient-preserving properties of this filter. The guided filter is a faster and better technique than the traditional bilateral filter in various applications.

Edge-aware techniques have more applications in computer vision/graphics than what we have introduced in this section. In many applications, we assign each pixel an estimated value, which can be a cost, confidence, a vote, or any other data. Then we need to account for spatial continuity, which is achieved by edge-aware techniques. A great many works achieve spatial continuity based on Markov Random Fields. But more and more works (like [15, 16]) attempt to use a simple filter instead. After our publication [32],

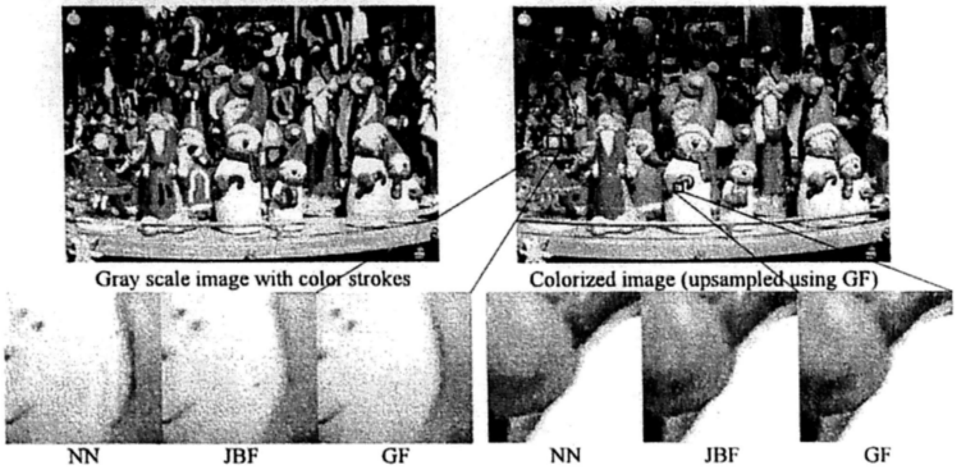

Gray scale image with color strokes
Colorized image (upsampled using GF)
NN
JBF
GF
NN
JBF
GF

Figure 4.20: Joint Upsampling. The upsampling methods includes: nearest-neighbor (NN), joint bilateral filter (JBF) [38], and guided filter (GF).

the guided filter has further been applied in stereo vision [60] and importance filtering [13]. We believe this filter is a potential technique in the future.

Chapter 5

Dark Channel Prior and Human Vision

Studies on computer vision and those on the human visual system (HVS) in psychology/physiology are mutually beneficial. The discoveries about the HVS and the human brain may inspire effective algorithms in computer vision; successful computer vision techniques suggest similar rules may be adopted in the HVS. In this chapter, we discuss the relation between the dark channel prior and the HVS. We find that the HVS possibly adopts a mechanism similar to the dark channel prior when perceiving haze. This discovery casts new insights into the area of human vision research. It also supports the validity of the dark channel prior as a computer vision algorithm, because a good way for artificial intelligence is to mimic human brains.

5.1 Introduction

Most visual problems, no matter for computers or for the HVS, are inherently ambiguous. For example, the 2D projection of the 3D world can be mathematically explained by an infinite number of 3D possibilities. A shadowed white card can reflect the same amount of light to a shadowless gray card. A hazy scene can be mathematically explained as being inherently faint and of low contrast but not covered by haze. The human beings are able to handle these ambiguities with no difficulty in most cases. Since many problems still remain challenging in computer vision, researchers always raise such a question: how can the human brains solve the problems so successfully?

The human visual system (HVS) could solve the ambiguities only through certain knowledge or assumptions known beforehand, *i.e.*, *priors*. This fact has long been realized by scientists as early as the 19th century [25]. But it remain arguable how the HVS obtains the priors. Some researchers (*e.g.*, Hering [35], see also [82]) suggest the priors are built in the genes and "present at birth". This was proved experimentally in the case of stereo disparity [59]; but in most other cases it is still unclear. Some other researchers (*e.g.*, Helmholtz [34], see also [82]) attribute the acquisition of priors to learning. The built-in-genes priors are shared by common people, while the learned priors can be more or less different among individuals.

It is also unclear where the visual mechanisms take effect in the HVS. The light reaching the retina in the eyes is converted into neuronal signals. The conversion is determined by the pattern of the photoreceptor cells, their sensitivity (*e.g.*, the R, G, and B receptors), and any other properties of the system. The signals are passed into the brain (*e.g.*, the visual cortex [37]) and undergo further processing. The visual mechanisms (that introduce priors) can be built in the signal delivery system or in the human brain as a physiological or intellectual process. The former mechanism (signal delivery) is unconscious because it is a mechanical process (just like a circuit board), while the latter mechanism (in the brain) can be unconscious or conscious.

Despite much debate on the mechanisms of the HVS, many studies (*e.g.*, [67, 24, 89, 44, 47]) share the hypothesis that the underlying priors are about the statistical properties of the natural world. The neurons, the physiological structures, or the intellectual behavior of the HVS are adapted to the natural world in the evolutionary processes (built-in-genes) or developmental processes (learned). In the terminology of computer vision, the HVS is "trained" by a huge data set which is "sampled" from the natural world in a long period. According to such statistics, the HVS is able to preclude the improbable solutions to the visual problems and solve the ambiguity. This scenario is somewhat similar to how we develop the haze removal algorithm. We collect a data set of outdoor haze-free images, and the dark channel prior is a statistical property of this set. Using this prior, we compute a most probable transmission value for each pixel and thus solve the ambiguity. In this sense, it is reasonable for us to ask: is it possible that the HVS uses a similar prior when perceiving haze?

But it is nontrivial to investigate the underlying mechanisms of the HVS. The system behaves like a black box: only the input image and the per-

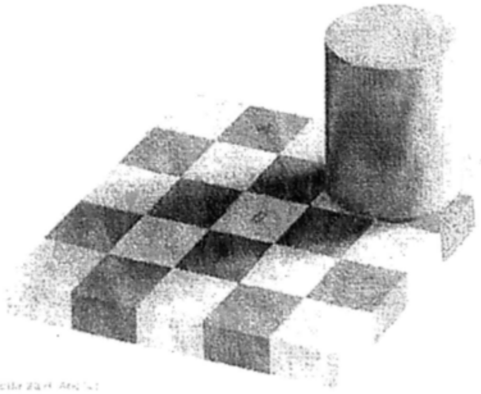


Figure 5.1: Checkershadow Illusion presented by [2]. The squares marked A and B have exactly the same intensity in the image (r=g=b=120), but they appear very different.

ception are available. The attempt to "disassemble" the black box through *physiological* experiments is one way to investigate it [37]. Another important and practical way is through illusions, which is often used in *psychology* [25].

An illusion is the distortion of the *perceived* image. The HVS is supposed [25] to use some mechanisms to process the visual signal, so the perceived signal (after processing) can be different with the physical input signal that just reaches the eyes. An illusions is presented when the perceived signal is severely distorted. In general the perceived illusion is systematic rather than random, and is shared by common people [73]. So illusions reflect the mechanisms of the HVS: they provide us opportunities to understand what is going on in the black box.

In this chapter we shall demonstrate several illusions, which are carefully designed based on the dark channel prior. We find that the dark channel of the image impacts the HVS's perception of haze. We also find that the HVS may partially rely on the dark channel to reduce (but not remove) the haze. The illusion experiments suggest that the HVS may adopt a mechanism like the dark channel prior. We hope that our experiments cast new insights into the area of HVS research.

5.2 Related Works

We briefly review some related works on human vision research.

As early as in the 19th century, people have found that lightness, contrast, colors, object shapes, depth, and many other attributes are not always perceived just as their physical properties [34, 35]. A great many illusion experiments are proposed by scientists and artists (*e.g.*, [2, 3, 72, 73, 25]). Fig 5.1 is one of the most famous lightness illusions: the squares marked A and B have exactly the same intensity in the image (r=g=b=120), but they appear very different. This and many other experiments unequivocally show that the physical signal must be processed by the HVS before perceived. Various theories and models are proposed to explain when the illusions appear and how the HVS processes the signal (*e.g.*, [25]).

The works partially related to haze are the ones about transparency (transmission) perception. A series of studies [46, 68, 61, 3, 69] find that the HVS has the ability to perceive the transparency of the covering media, like veils, glass filters, and haze. In [46] the author assumes the HVS is following the physical imaging model of the semi-transparent media (like the haze imaging equation) and deduces the perceived transparency accordingly. But in [68, 61] it is shown by experiments that the perceived transparency has significant and systematic differences with the physical value. In [69] new experiments show that the perceived transparency are asymmetric between brightening and darkening cases. In [3] some striking illusions show that the intensity of the objects covered by haze can appear very different with different context, even though it is actually unchanged. This experiment suggests the lightness perception is with an image segmentation procedure.

These works also attempt to explain the experimental results by modeling the HVS's mechanisms. In [68] it is supposed that the perceived transparency is determined by the relative contrast (Michelson contrast) of the luminance. But in [61] the authors find neither the Michelson contrast nor other simple contrast metrics can explain the experiments. In [69] the model is modified: Michelson contrast is only applied in darkening transparency, but not in brightening cases. Our following experiments show that the dark channel can be another possible determinant of transparency perception.

Many studies suppose that the human vision mechanisms (and the priors used) are adapted to the statistics of the natural world. In [89] it is found that the perceived depth can be explained by the statistics of distances in the natural world. In [47] the authors find that the perception of surface quality is determined by image statistical properties. In [44] it is shown that color contrast, constancy, and assimilation can be predicted by the natural spectral statistics. In [69], the authors assume that the perceived transparency in darkening cases are due to the adaptation to shadows (mathematically equivalent to black veils), and in brightening cases are due to the adaptation to spotlights. Our following experiments suggest the brightening cases can

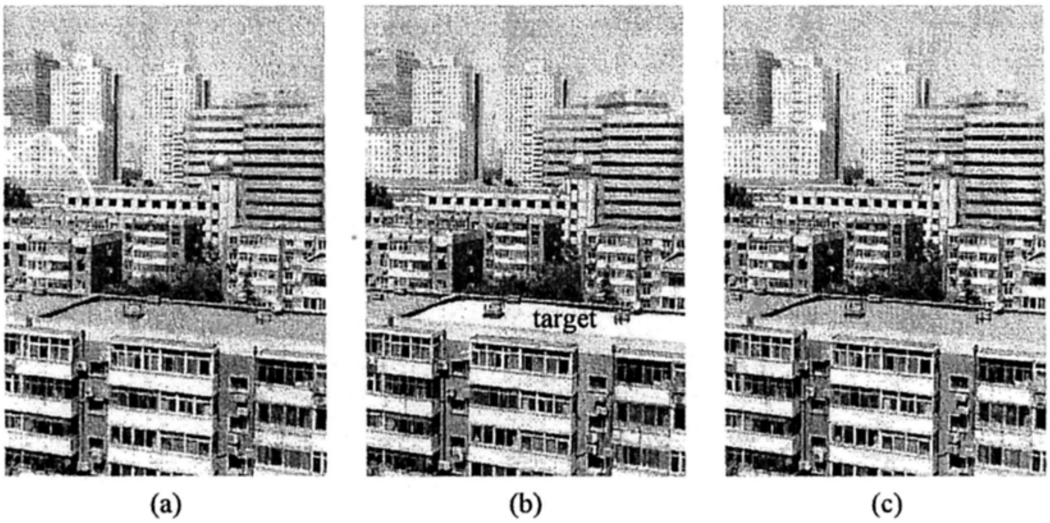


Figure 5.2: Image preparation. (a) A real image in a clear day. (b) The target region is manually segmented. (c) The target region is filled with a uniform color ((r, g, b) = (152, 140, 130)).

also be due to the adaptation to haze.

5.3 Illusion Experiments

Preparation

The image used in our experiments is prepared as following. In Fig. 5.2 (a) is a real image in a clear day. We manually segment the roof of the front building, as shown in Fig. 5.2 (b). We shall call this region the "target region". Then we fill this region with a uniform color ((r, g, b) = (152, 140, 130), we use 256-level in this chapter), as shown in Fig. 5.2 (c). We shall call Fig. 5.2 (c) the "source image" in the experiments. This operation is to completely remove the textures/noise in the target region, precluding the possibility that the following illusions can be due to these factors.

We also observe the following illusions in other images. But since illusions are systematic rather than random, we believe this example is representative and convincing.

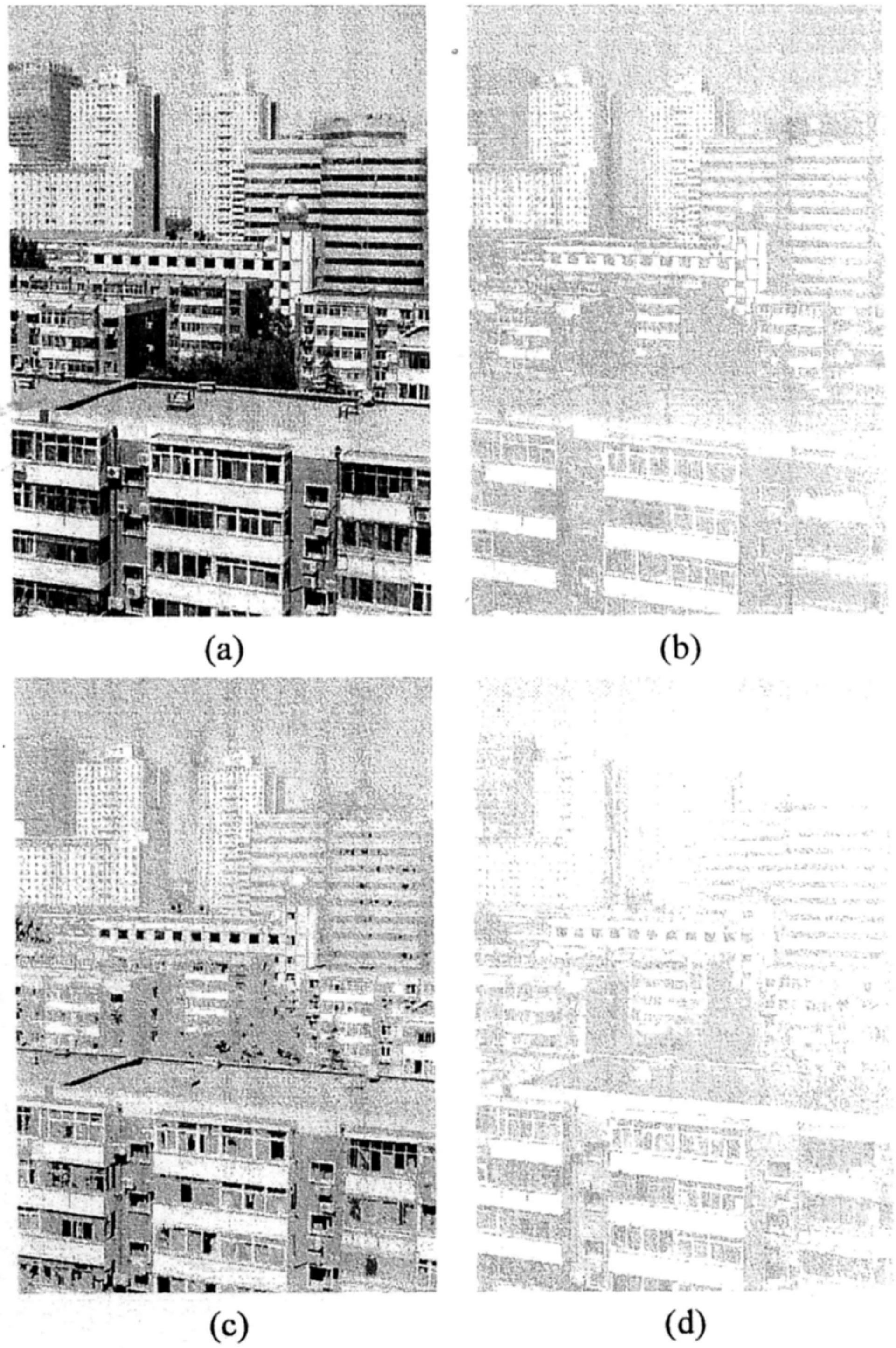
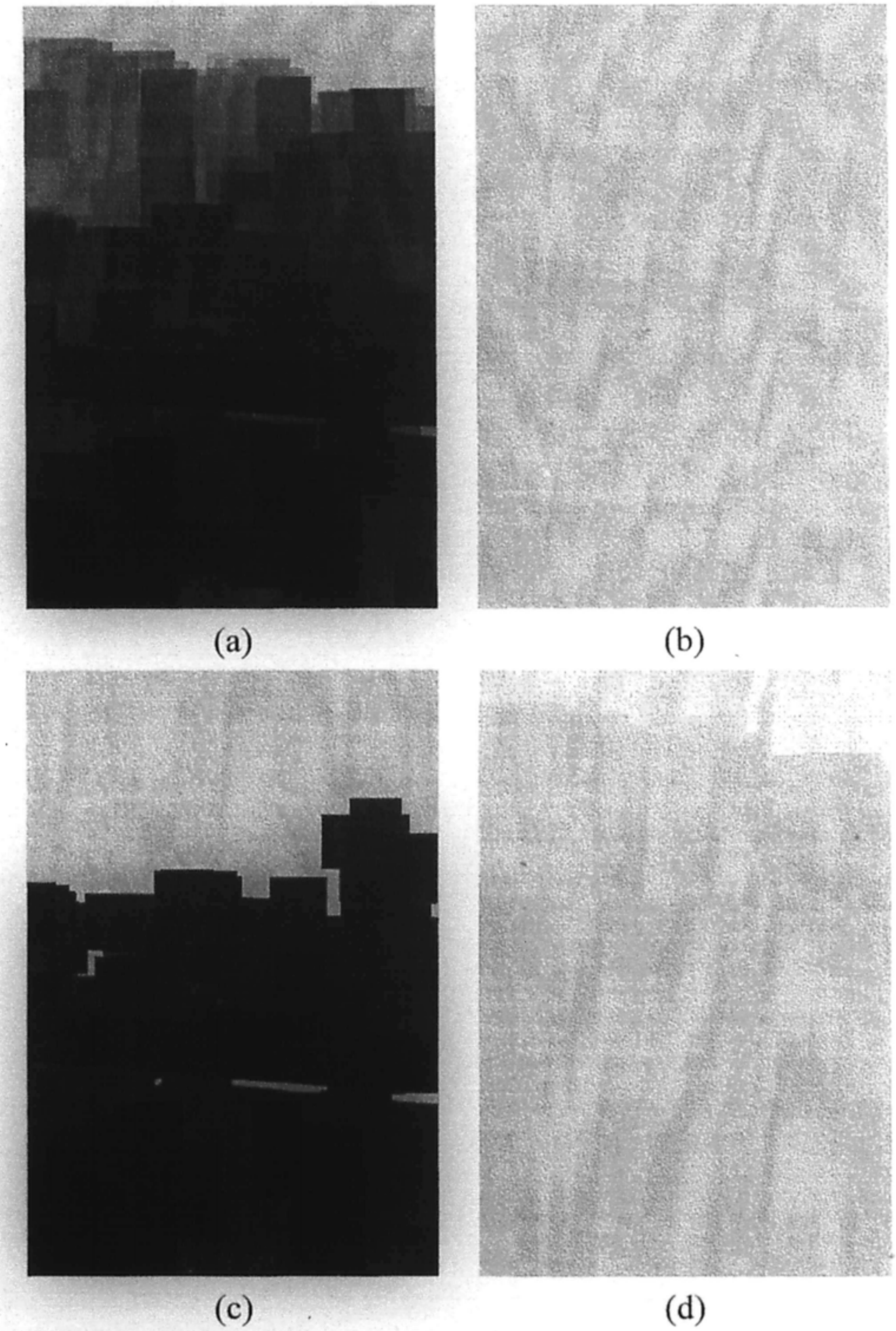

(a)
(b)
(c)
(d)

Figure 5.3: Illusion Experiment I-III.



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

Figure 5.4: Dark channels of Fig. 5.3, computed using a window radius 50 (image size 800×1200).

Experiment I

Experiment I is shown in Fig. 5.3 (a) and (b). Fig. 5.3 (a) is the source image. Fig. 5.3 (b) is obtained from the source image through the following way: each color channel of each pixel is increased to 128 if they are smaller than 128. Denote the source image by $\mathbf{I}$ and the manipulated image by $\check{\mathbf{I}}$. This operation means for any pixel $\mathbf{x}$ we have:

$$\check{I}_c(\mathbf{x}) = \max(I_c(\mathbf{x}), 128), \tag{5.1}$$

where $c \in \{r, g, b\}$ is the channel index.

Notice that the color of the target region is (152, 140, 130), so it is unchanged by this operation. Thus, the target regions in the two images in Fig. 5.3 (a) and (b) are actually identical. However, they appear very different: the target region in (b) looks much darker. This experiment indicates that the perceived color of a region depends on its context (surroundings). This fact has been verified in many other illusions (*e.g.*, [3, 25]). But it remains arguable that what properties of the context influence the perception.

In fact, the intensity of the dark channel is increased to at least 128:

$$\check{I}^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{x}' \in \Omega} \min_{c} \check{I}_c(\mathbf{x}') \geq 128, \tag{5.2}$$

because all $\check{I}_c$ are not smaller than 128. So the operation in (5.1) changes the dark channel of the images (see Fig. 5.4 (a) and (b)). We may reasonably assume it is the dark channel that influences the perceived color. But in this experiment, other factors like contrast and colors are also changed by the operation. So we need other experiments to test whether the dark channel is the most possible factor.

Experiment II

In Experiment II, we study the image in Fig. 5.3 (c). This image is synthesized in this way:

$$\check{I}_c(\mathbf{x}) = \begin{cases} \max(I_c(\mathbf{x}), 128) & \text{if } I_c(\mathbf{x}) > 36 \\ I_c(\mathbf{x}) & \text{if } I_c(\mathbf{x}) \leq 36 \end{cases} \tag{5.3}$$

This means that we keep the darkest pixels ($I_c(\mathbf{x}) \leq 36$) while increasing the other ones as in Fig. 5.3 (b).

Interestingly, we find that the illusion is greatly weakened: the target region in Fig. 5.3 (c) appears less dark as in Fig. 5.3 (b), although a substantial portion of the pixels are identical in both images. We note they have very different dark channels (see Fig. 5.4 (b) and (c)). This experiment is consistent with our conjecture that the dark channel influences the perception. This experiment also shows that the illusion can be determined merely by a few pixels.

Experiment III

In Experiment III, we further generate an image as Fig. 5.3 (d). We still keep the color of the target region unchanged ((r, g, b) = (152, 140, 130)). The remaining regions are synthesized from the source image (Fig. 5.3 (a)) in this way:

$$\tilde{I}_c(\mathbf{x}) = 0.5 * I_c(\mathbf{x}) + 128. \tag{5.4}$$

This operation also increases the intensity to at least 128. But it changes all the pixels' intensity (except the target region), whereas the operation in (5.1) only changes those below 128.

The illusion appears in this case again. The color of the target region in Fig. 5.3 (d) appears very similar with Fig. 5.3 (b), but much darker than Fig. 5.3 (a). However, the contexts in Fig. 5.3 (b) and (d) are very different in colors and variance. Besides, Fig. 5.3 (d) has a physical meaning: the image (except the target region) is covered by a haze layer with $t = 0.5$ and $A_c = 256$ as in equation (5.4). On the contrary, Fig. 5.3 (b) has no physical explanation in real world. One common property of Fig. 5.3 (b) and (d) is that they have similar dark channels (see Fig. 5.4 (b) and (d)). This experiment further supports our conjecture that the dark channel influences the perception.

Notice that all the four target regions in Fig. 5.3 are completely identical, but their perceptions are not. And the perceptions are consistent with the corresponding dark channels (Fig. 5.4).

Experiment IV

If the HVS really uses a prior like the dark channel prior, it is reasonable for us to assume that this prior is from the statistics of the haze phenomena in the natural world. The above illusions undoubtedly show that the HVS

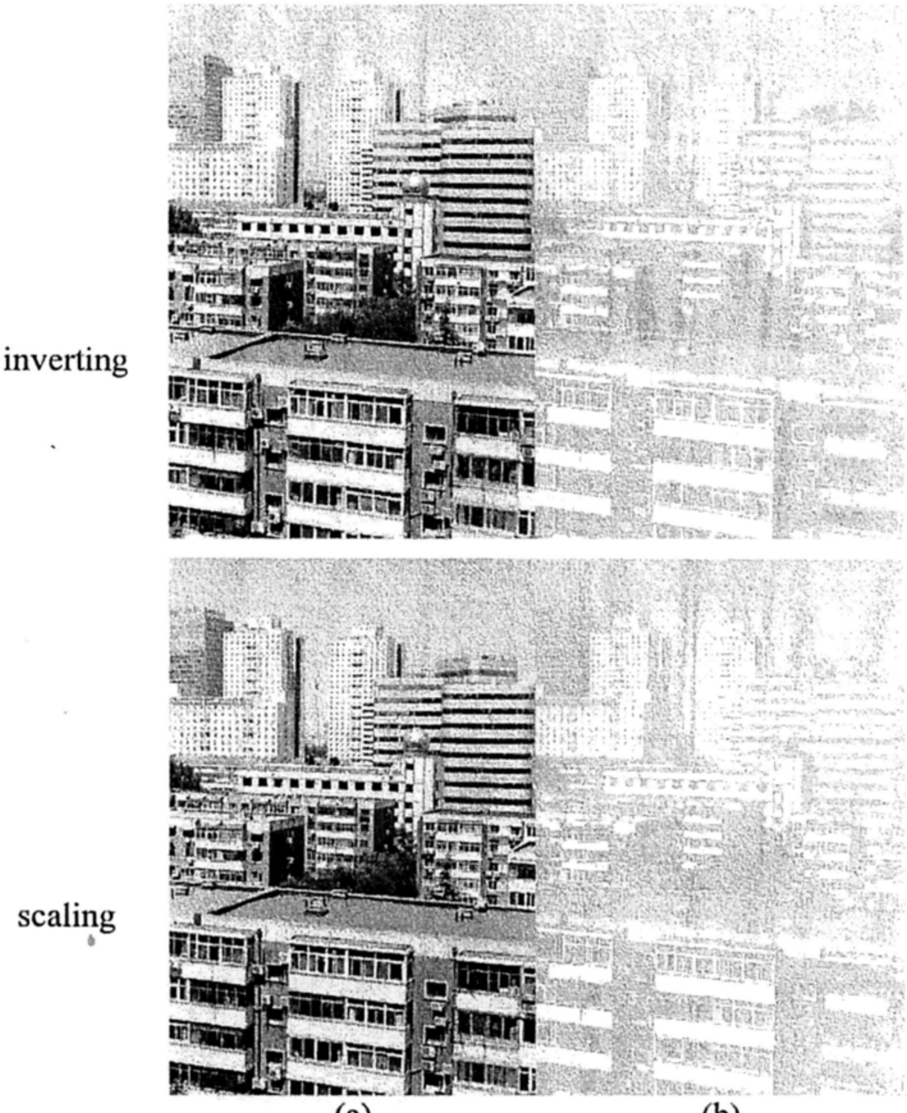


Figure 5.5: Illusion Experiment IV.

processes the visual signal. If the HVS is adapted to the haze, it is very possible that HVS attempts to "reduces" the haze in this process [1]

In Experiment IV, we focus on the images in Fig. 5.3 (a) and (b). We assume the HVS processes the visual signal in Fig. 5.3 (b) more than Fig. 5.3 (a). So we want to add extra operation on Fig. 5.3 (a) to counteract the discrepancy: we adjust color of the target region in Fig. 5.3 (a) to make it appear like Fig. 5.3 (b).

We try to adjust in two ways. If we invert the haze imaging equation, we have:

$$J_c(\mathbf{x}) = \frac{I_c(\mathbf{x}) - A_c}{t(\mathbf{x})} + A_c. \tag{5.5}$$

The haze effect can be reduced given any $t < 1$. So in the first way, we adjust the color of the target region in Fig. 5.3 (a) by:

$$\acute{I}_c = \frac{I_c - 255}{t} + 255. \tag{5.6}$$

We slide a bar to adjust t (we assume a global t). We stay in the value that makes the target regions in (a) and (b) are most similar. The result is in Fig. 5.5 (top).

Another way is to scale the intensity of the target region, making it shadowed:

$$\acute{I}_c = k * I_c. \tag{5.7}$$

Similarly, we slide a bar to adjust k. However, an apparat visual difference exists no matter how we adjust k. We find that if we darken the target region in (a) by scaling, the chrominance still appears different. Fig. 5.5 (bottom) shows the case.

This experiment indicates that the illusion in Fig. 5.3 (a) and (b) is probably because the HVS is using a haze reducing model instead of a shadowing model, although both models can darken the region. It is worth mentioning that the image (b) is actually not a hazy image: it is a synthetic image that does not exist in the real world. But the HVS treats it as a hazy image and uses a model of haze. So the dark channel is perhaps the main factor on which the HVS bases to trigger the haze reducing process, regardless of the image content.

[1] The HVS may "reduce" instead of completely "remove" the haze, because a hazy image is still perceived as hazy, although it may be perceived less hazy than it is.

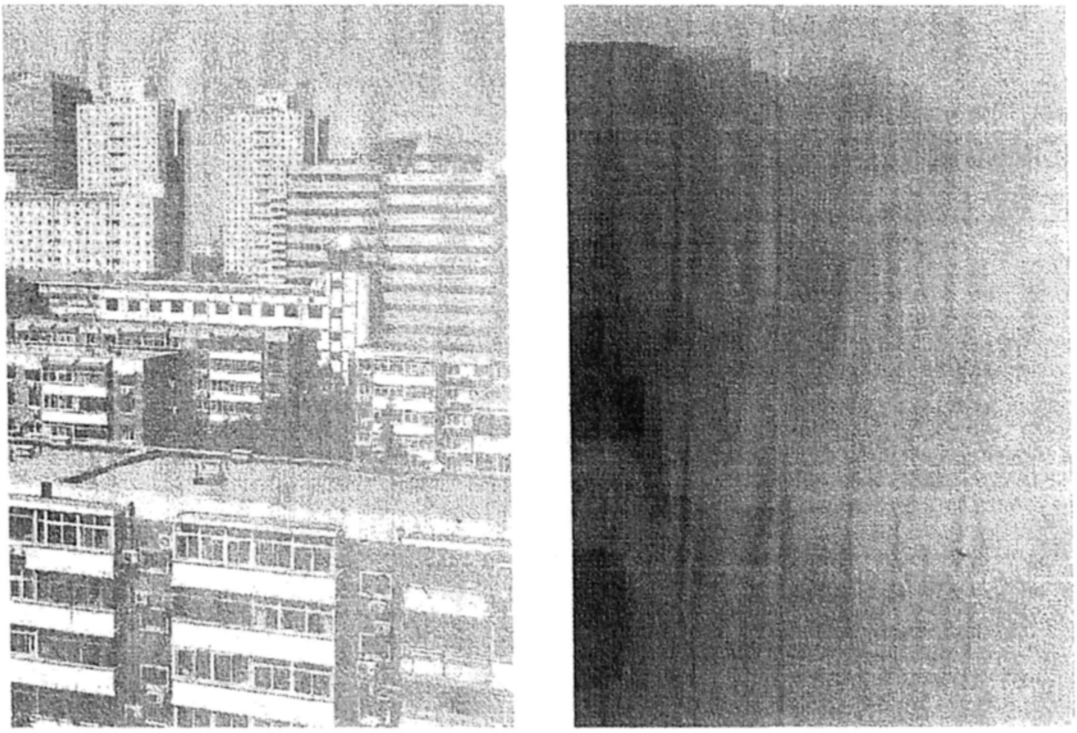


Figure 5.6: Illusion Experiment V.

Experiment V

In Experiment V, we process the source image in this way:

$$\tilde{I}_c(\mathbf{x}) = \max(I_c(\mathbf{x}), \theta). \tag{5.8}$$

Unlike the previous operation, the threshold θ is non-uniform: it is 80 on the left and gradually increases to 128 on the right. See Fig.5.6 (left) for the image after manipulation. Notice that target region ((r, g, b) = (152, 140, 130)) is unchanged and remains uniform. However, it appears non-uniform and darker on the right.

This can also be explained by the assumption that the HVS relies on the dark channel (Fig.5.6 right) to perceive haze. The HVS reduces more haze effect when the dark channel is brighter. This experiment also indicates that the operation of the HVS on the visual signal is locally based, just as our haze removal algorithm.

5.4 Proposed Model

We propose the following model to explain the HVS's mechanism for perceiving haze: given the photometric visual image (the physical signal reaching the eyes), the HVS computes the dark channel of the image. Based on the values in the dark channel, the HVS adopts the inverse haze imaging model (5.5) to reduce the haze effect.

Remarks:

- This procedure in the HVS is mostly unconscious and passive - the signal has been processed before any active thinking behavior happens in the brain. Actually, it is very difficult for one to avoid these illusions by his/her active thinking, even when he/she is told that the target regions in Fig.5.3 (a) and (b) are identical.

- The dark channel can undergo another process before the inverse haze imaging model. This process plays a role like soft matting or guided filtering: for spatial continuities. Moreover, the image can be segmented into sub-regions before or after the dark channel is computed. But these assumptions have not yet been supported by experiments.

- The transmission value t used in the inverse haze imaging model (5.5) is determined by the dark channel. But the concrete mathematical form is still unclear, because we hold that quantitatively measuring a perceived signal in our images is subject to many unstable factors and thus not reliable. But we are sure that the HVS does not completely remove the haze by setting $t = 1 - \hat{I}^{\text{dark}}$ (3.15): if the HVS does so, some pixels in the perceived image would be purely dark. Therefore, the HVS only *reduces* the haze but not removes it.

- The HVS does not judge whether the image is really formed by a physical haze model (Fig.5.3 (d)) or physically invalid synthesis (Fig.5.3 (b) and (c)). Even though the synthesis images are unnatural and perhaps never exist in the real world, the HVS still adopts the mechanism to the signal.

It is worth mentioning that we can only claim that our model is consistent with all the above experimental phenomena, but we cannot *prove* that the model is really adopted by the HVS, just as in most human vision studies. However, the illusions shown in this chapter are undoubted. Any new model or theory in future studies is supposed to be valid only when they are compatible with the illusions here.

5.5 Discussion and Conclusion

In this chapter, we demonstrate several illusion experiments which indicate the HVS may adopt a mechanism like the dark channel prior to perceive haze.

Though the experiments strongly support our conjecture that the dark channel prior is related with the HVS, we hold that the HVS must also combine other priors or knowledge in the process. For example, the HVS may segment the image into semantic regions before adopting the prior. It may also use content-adaptive window sizes instead of a fixed one when computing a dark channel. The HVS reduces the haze effect in the perceived image rather than remove it. But to what extent is the reduction? Besides, there has been no evidence that the HVS is using a guided filter or similar mechanisms. The entire "algorithm" running in the HVS remains unknown.

The haze is a kind of semi-transparent phenomena. Some studies [46, 68, 61, 69] find that the HVS has the ability to perceive the transparency of a layer, but the perceived transparency can be different from the true value. Because these studies show that the underlying mechanism is similar among common people, it is reasonable for us to suppose that this ability is obtained in an evolutionary process and coded by genes. In the evolutionary period, there are mainly two natural semi-transparent phenomena[2]: haze and shadow. While haze usually veils the scene by a white or gray color, a shadow is equivalent to covering an object by a purely black semi-transparent layer (*i.e.*, $A = 0$ if we analogize the haze imaging equation):

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}). \tag{5.9}$$

While a shadows preserves the luminance ratio[3] ($I(\mathbf{x}_1)/I(\mathbf{x}_2) = J(\mathbf{x}_1)/J(\mathbf{x}_2)$ if t is uniform), the haze does not ($I(\mathbf{x}_1)/I(\mathbf{x}_2) \neq J(\mathbf{x}_1)/J(\mathbf{x}_2)$ even when t is uniform). If the visual mechanisms are evolved from these two kinds of semi-transparent phenomena, it is entirely possible that the HVS uses

[2] We assume the evolutionary period is much longer than the modern era, so artificial semi-transparent objects are almost unavailable.

[3] Luminance ratio is a kind of definitions on "contrast" [25].

different ways to perceive different kinds of semi-transparent media. It has been found that the perceived transparency follows different rules in the cases of brightening/darkening semi-transparent layers [69]. This can also be explained by the dark channel prior: the ability of the HVS to perceive a bright semi-transparent layer is from the statistics of the haze phenomenon, whereas the ability to perceive a dark layer is from the statistics of shadows. Although the dark channel prior is effective in the case of haze, it is not useful in the case of shadow. The HVS adopts a different prior in the darkening case, so the perceptions of brightening/darkening transparency follow different rules.

In sum, much evidence suggests that the dark channel prior is related to human vision. We expect our study stemmed from a computer vision algorithm will cast new insights into the human vision area.

Chapter 6

Conclusion

In this thesis, we have studied the haze removal problems and related issues. We summarize the main contributions of this thesis in the following.

In Chapter 3 we propose the dark channel prior and a single image haze removal algorithm. The dark channel prior comes from an intuitive observation on outdoor haze-free images. Unlike the heuristic assumptions in previous methods, this prior is based on the image statistics, which is the result of inherent physical properties (illumination, colors, and geometry). This prior provides a robust estimation for each pixel, and thus solves the ambiguity of the problem. Despite of its simplicity, our haze removal algorithm based on this prior is very effective in various situations. Experiments show that our method outperforms most of the previous works.

In Chapter 4 we study fast algorithms for haze removal. The challenge is that we need to combine the pixel-wise constraints with spatial continuities, which is usually time-consuming. In the first part of this chapter, we develop an algorithm which reduces the time complex of a linear solver from $O(rN^{\frac{3}{2}})$ to $O(\frac{1}{r}N^{\frac{3}{2}})$ with a kernel radius r. This discovery is contrary to conventional theories, but we prove it true both theoretically and experimentally. This algorithms allows us to increase the speed by choosing a larger kernel. In the second part of this chapter, we treat the problem as a general edge-aware filtering process and propose a novel guided filter accordingly. This filter voids solving the linear system, and can be computed in $O(N)$ time regardless of the kernel size. It enables a real-time performance of our haze removal algorithm. This filter also exhibits many nice properties. Experiments show that it is advantageous in various applications including haze removal, in terms of both quality and efficiency.

In Chapter 5 we study the relation between the dark channel prior and the human visual system (HVS). We demonstrate several striking illusion experiments to show that the HVS probably adopts a similar mechanism like the dark channel prior to perceive haze. This study casts new insights into the area of human vision research in psychology/phisiology. It also further supports the validity of the dark channel prior as a computer vision algorithm, because we are possibly simulating a human visual mechanism.

In the future, we plan to study the problem under more general haze imaging situations, *e.g.*, spatially variant atmospheric light or channel-dependent transmission. The problem becomes more ill-posed and new priors are needed. We are also interested in applying the fast guided filter in more computer vision problems. On the human vision study, we expect to build a model to quantitatively explain the haze perception.

□ **End of chapter.**

Appendix A

Physical Model

The haze imaging model is:

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

In this appendix, we provide the physical derivation of this equation. The derivation is mainly following the method in [49].

A.1 Scattering

We begin with the micro picture of the phenomenon. The particles in the atmosphere scatter light. "Scattering" means that a particle absorbs a portion of the incident light and radiates the absorbed light as a light source (Fig. A.1).

Consider a small volume in atmosphere (see Fig. A.2). According to [45, 49], the total light flux Φ scattered by a small volume is proportional to the incident flux E:

$$\Phi(\lambda) = \beta(\lambda)E(\lambda)\Delta V. \qquad (A.1)$$

Here, E is the intensity of the incident light, Φ is the total scattered light flux, ΔV is the volume, β is the *total scattering coefficient* [49], and λ is the wavelength of the light, indicating the variables are wavelength-dependent.

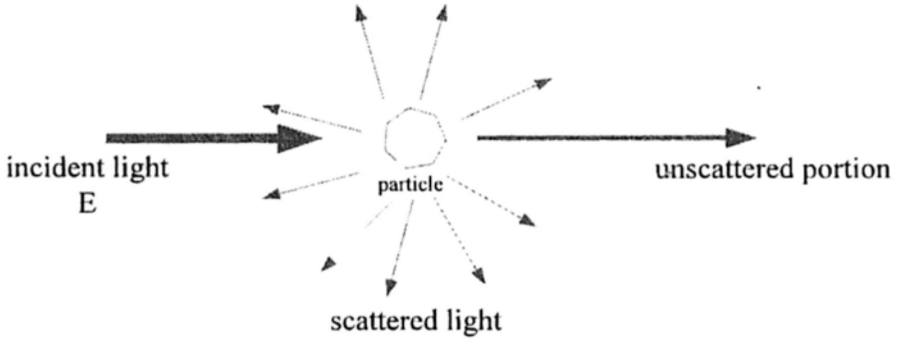


Figure A.1: A particle scattering light.

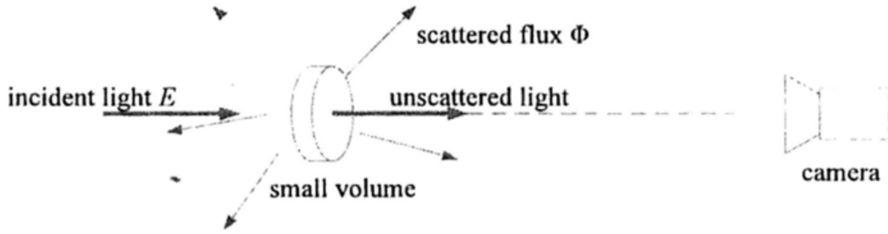


Figure A.2: Scattering model in a small volume of atmosphere.

The total scattering coefficient $\beta(\lambda)$ is determined by the particle material, size, shape, and concentration.

Scattering is the basic reason for both *direct attenuation* and *airlight*.

A.2 Direct Attenuation

The first effect of scattering is that it attenuates the incident light. Consider a beam have a unit cross-sectional area (*e.g.*, one pixel) as in Fig. A.3. Denote as z the direction of the observer to a scene point. According to (A.1), the incident flux passing through an infinitesimally small sheet of thickness dz is changed by dE:

$$dE(z, \lambda) = -\beta(z, \lambda)E(z, \lambda)dz. \tag{A.2}$$

where the volume ΔV becomes $1 \cdot dz$, and "-" indicates the light is weakened. Integrating this equation between $z = 0$ and $z = d$ we obtain:

$$E(d, \lambda) = E(0, \lambda)e^{-\int_0^d \beta(z,\lambda)dz}. \tag{A.3}$$

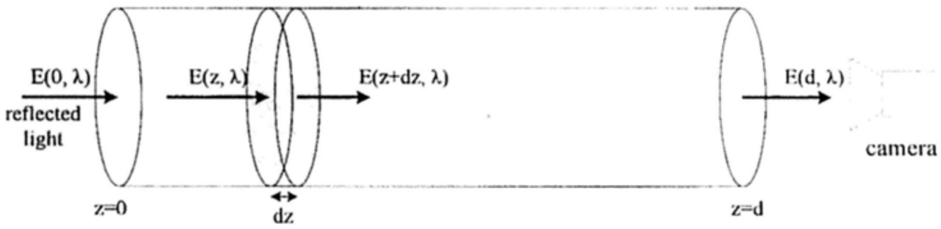


Figure A.3: Light traveling in the atmosphere is attenuated.

where $E(0,\lambda)$ is the light reflected by the scene point. $E(d,\lambda)$ is actually the *direction attenuation*. We shall represent this portion of light in colors.

A.3 Airlight

The second effect of scattering is that it blends the reflected light with additive *airlight*.

A particle radiates the light it absorbs, behaving as a light source suspending in the air. All these particles generate the *atmospheric light*: the light scattered a great many times by a huge number of particles. Statistically, the atmospheric light is spatially homogenous[1] and isotropic. If we place a small disk of a unit area in the atmosphere, the flux passing through this disk is constant regardless of its position and direction. We denote the intensity of this flux per unit area by $\varepsilon(\lambda)$.

Consider an infinitesimally thin cylinder volume with a unit cross-sectional area in the atmosphere (Fig. A.4). This volume is lit by the atmospheric light from all directions. The total incident flux contributed by the atmospheric light to this volume is:

$$\hat{E}(\lambda) = 2\varepsilon(\lambda), \tag{A.4}$$

where the number 2 is because the volume has two sides. According to (A.1), the total scattered flux is $d\hat{\Phi}(\lambda) = \beta(\lambda)2\varepsilon(\lambda)dz$. But because the cylinder has two sides, the flux dR radiated from the side facing the observer is:

$$dR(z,\lambda) = \beta(z,\lambda)\varepsilon(\lambda)dz. \tag{A.5}$$

This radiated light also has to pass through the haze between the volume

[1] in the sense of a large scale like a scene

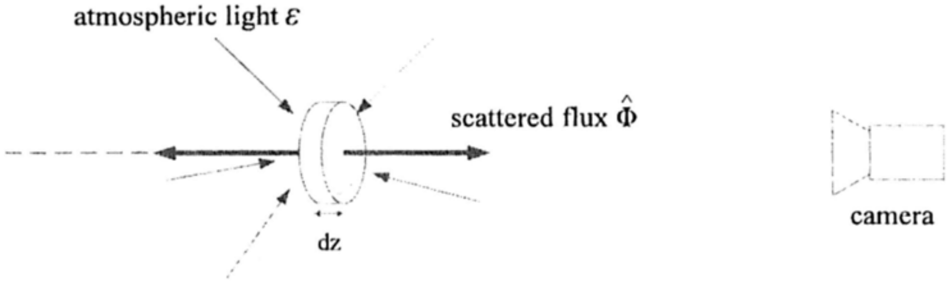


Figure A.4: A unit volume scatters atmospheric light coming from all directions.

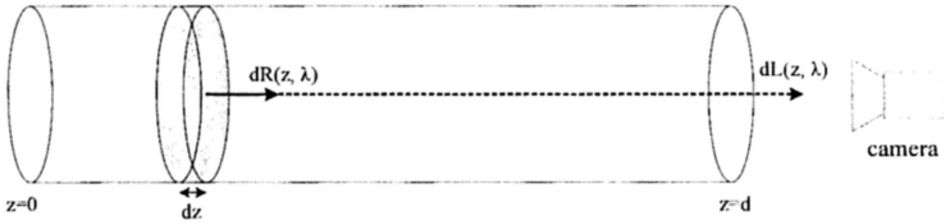


Figure A.5: The radiated light by a volume has to pass through the haze before reaching the observer.

and the observer, and is attenuated (see Fig. A.5). Following the attenuation equation (A.3), the light dL after attenuation that reaches the observer is:

$$dL(z,\lambda) = e^{-\int_0^z \beta(z',\lambda)dz'} \cdot dR(z,\lambda). \tag{A.6}$$

Integrating all the infinitesimal light sources between z=0 and z=d, we obtain the total light L:

$$L(d,\lambda) = \varepsilon(\lambda)(1 - e^{-\int_0^d \beta(z,\lambda)dz}). \tag{A.7}$$

This is the light that reaches the observer due to the atmospheric light. $L(d,\lambda)$ is actually the *airlight*.

A.4 Colorimetry

We have obtained the direction attenuation in (A.3) and the airlight in (A.7). So the total light reaching the observer is:

$$E(d,\lambda) + L(d,\lambda) = E(0,\lambda)e^{-\int_0^d \beta(z,\lambda)dz} + \varepsilon(\lambda)(1 - e^{-\int_0^d \beta(z,\lambda)dz}). \qquad \text{(A.8)}$$

Next we represent the two terms in colors. In colorimetry, the color seen by human eyes or captured by a camera is a weighted integral over the light spectrum [10]. Denote the weighting function as $s_c(\lambda)$ where c represents an RGB channel. The color of the direct attenuation in (A.3) is:

$$\int s_c(\lambda)E(d,\lambda)d\lambda = \int (s_c(\lambda)E(0,\lambda)e^{-\int_0^d \beta(z,\lambda)dz})d\lambda, \qquad \text{(A.9)}$$

and the color of the airlight in (A.7) is

$$\int s_c(\lambda)L(d,\lambda)d\lambda = \int s_c(\lambda)\varepsilon(\lambda)(1 - e^{-\int_0^d \beta(z,\lambda)dz})d\lambda. \qquad \text{(A.10)}$$

To further simplify these equations, we have to make the following assumption: *the scattering coefficient β is independent of the wavelength.*

This assumption is true when the particles in the atmosphere have a proper size[2]. Fortunately, this is satisfied in haze, fog and mist [45]. Under this assumption, we can write the right hand sides of (A.9) and (A.10) as:

$$\left(\int s_c(\lambda)E(0,\lambda)d\lambda\right) \cdot e^{-\int_0^d \beta(z)dz}, \qquad \text{(A.11)}$$

and

$$\left(\int s_c(\lambda)\varepsilon(\lambda)d\lambda\right) \cdot (1 - e^{-\int_0^d \beta(z)dz}). \qquad \text{(A.12)}$$

[2] radius in the order between $10^{-2}\mu$m and $10^{1}\mu$m [45]

By the definition of **J** and **A** in the haze image equation (2.1) we have:

$$J_c = \int s_c(\lambda)E(0,\lambda)d\lambda \tag{A.13}$$

$$A_c = \int s_c(\lambda)\varepsilon(\lambda)d\lambda, \tag{A.14}$$

where we use J_c to denote the value of **J** in channel c (likewise A_c). We further define the transmission t as:

$$t = e^{-\int_0^d \beta(z)dz}. \tag{A.15}$$

Therefore, (A.11) is the *direct attenuation* term $\mathbf{J}t$, and (A.12) is the *airlight* term $\mathbf{A}(1-t)$. The color of the total light (A.8) is $\mathbf{J}t + \mathbf{A}(1-t)$. This explains the haze imaging equation (2.1).

Appendix B

Mathematical Derivations

B.1 Theorem 4.1.1: proof

Proof. To prove $\mathbf{q}$ equals to $\mathrm{L}\mathbf{p}$, we need to prove $\partial q_i/\partial p_j = \mathrm{L}(i,j)$.

Putting (4.14) into (4.15) and eliminating b^*, we obtain:

$$\frac{\partial q_i}{\partial p_j} = |w|\delta_{ij} - \sum_{k\in w_i}(\frac{\partial \bar{p}_k}{\partial p_j} + \frac{\partial a_k^*}{\partial p_j}(I_i - \mu_k)) \tag{B.1}$$

Here, we have:

$$\frac{\partial \bar{p}_k}{\partial p_j} = \frac{1}{|w|}\sum_{n\in w_k}\frac{\partial p_n}{\partial p_j} = \frac{1}{|w|}\delta_{j\in w_k} = \frac{1}{|w|}\delta_{k\in w_j} \tag{B.2}$$

where $\delta_{j\in w_k}$ is 1 if $j \in w_k$, and is 0 otherwise. According to (4.13) we also have:

$$\begin{aligned}\frac{\partial a_k^*}{\partial p_j} &= \frac{1}{\sigma_k^2+\frac{\varepsilon}{|w|}}(\frac{1}{|w|}\sum_{i\in w_k}\frac{\partial p_i}{\partial p_j}I_i - \frac{\partial \bar{p}_k}{\partial p_j}\mu_k) \\ &= \frac{1}{\sigma_k^2+\frac{\varepsilon}{|w|}}(\frac{1}{|w|}I_j - \frac{1}{|w|}\mu_k)\delta_{k\in w_j} \qquad \text{(B.3)}\end{aligned}$$

Putting (B.2) and (B.3) into (B.1), we obtain:

$$\frac{\partial q_i}{\partial p_j} = |w|\delta_{ij} - \frac{1}{|w|}\sum_{k\in w_i, k\in w_j}(1+\frac{1}{\sigma_k^2+\frac{\varepsilon}{|w|}}(I_j-\mu_k)(I_i-\mu_k)) \qquad \text{(B.4)}$$

This equals to L(i,j) in (4.9). □

B.2 O(N) algorithms for mean, variance, and covariance in sliding windows

Suppose the window w is square and has the size $(2r+1)\times(2r+1)$. A box filter performed on any image I is defined as the sum on each window (sliding windows):

$$\begin{aligned}\mathrm{B}^I(x,y) &= \sum_{(x',y')\in w(x,y)} I(x',y') \qquad \text{(B.5)} \\ &= \sum_{x'=x-r}^{x+r}\sum_{y'=y-r}^{y+r} I(x',y'), \qquad \text{(B.6)}\end{aligned}$$

where $w(x,y)$ is the window centered at (x,y). A brute-force algorithm takes O(Nr^2) time to compute the sums in all windows. But we can reduce it to O(N) time by Integral Image technique [22] or by cumulative sums.

Here we adopt cumulative sums. Notice that the box filter is *separable*, meaning that we can compute a 1-D box filter along x-axis and then along y-axis. A 1-D box filter along x-axis is defined as:

$$\mathrm{B}^I(x) = \sum_{x'=x-r}^{x+r} I(x'). \tag{B.7}$$

The 1-D *cumulative sum* is defined as

$$\mathrm{C}^I(x) = \sum_{x'=0}^{x} I(x'). \tag{B.8}$$

Obviously, the cumulative sums at all pixels $x \in [0, N_x]$ of a 1-D signal can be computed in $\mathrm{O}(N_x)$ time through a single scan. So the 1-D box filter can be computed by:

$$\mathrm{B}^I(x) = \mathrm{C}^I(x+r) - \mathrm{C}^I(x-r-1) \tag{B.9}$$

in as few as one operation, regardless of the kernel radius r. The 1-D box filter along y-axis can be computed similarly. So the 2-D box filter can be computed in O(N) time for any r.

The mean, variance, and covariance in sliding windows can be expressed as a series of box filters. For simplicity, we denote $\mathrm{B}^I(x,y)$ by B^I_k where k is the index of the pixel (x,y). The mean in sliding windows is:

$$\bar{I}_k = \frac{1}{|w_k|}\sum_{i\in w_k} I_i = \frac{B^I_k}{B^U_k}, \tag{B.10}$$

where U is an image whose pixels are all 1. The variance in sliding windows is:

$$\sigma_k^2 = \frac{1}{|w_k|}\sum_{i\in w_k}(I_i - \bar{I}_k)^2 = \frac{1}{|w_k|}\sum_{i\in w_k} I_i^2 - \bar{I}_k^2 \quad (B.11)$$

$$= \frac{1}{|w_k|}B_k^{I^2} - \bar{I}_k^2, \quad (B.12)$$

and the covariance in sliding windows is:

$$\mathrm{cov}_k(I,p) = \frac{1}{|w_k|}\sum_{i\in w_k}(I_i - \bar{I}_k)(p_i - \bar{p}_k) = \frac{1}{|w_k|}\sum_{i\in w_k} I_i p_i - \bar{I}_k\bar{p}_k \quad (B.13)$$

$$= \frac{1}{|w_k|}B_k^{Ip} - \bar{I}_k\bar{p}_k. \quad (B.14)$$

Thus, the mean, variance, and covariance in sliding windows can all be computed in O(N) time, regardless of the kernel radius r.

B.3 Approximate eigenvalues of L

We only consider the approximately constant regions of I. According to (4.9), the elements of L in these region are given by:

$$\mathrm{L}_{ij} = \sum_{k|(i,j)\in w_k}(\delta_{ij} - \frac{1}{|w|}). \quad (B.15)$$

To compute the eigenvalues of L, we are interested in the product $\mathbf{Lp}$ where $\mathbf{p}$ is an eigenvector. We can show that (B.15) leads to:

$$\mathbf{Lp} = |w|\mathbf{p} - \frac{1}{|w|}\mathrm{BB}\mathbf{p} \quad (B.16)$$

$$= |w|\mathbf{p} - \frac{1}{|w|}\mathrm{B_yB_xB_yB_x}\mathbf{p}. \quad (B.17)$$

Here B denotes the matrix form of a 2D box filter. A 2-D box filter can be separated into two 1-D box filters $\mathrm{B_x}$ and $\mathrm{B_y}$ along the x-axis and the y-axis respectively. To find the eigenvalues of L, we only need to find the eigenvalues of $\mathrm{B_x}$ and $\mathrm{B_y}$.

We apply the discrete-time Fourier Transform (DTFT) [54] to approximately compute the eigenvalues of $\mathrm{B_x}$. The solution is:

$$\xi_{k_x} = \frac{\sin((2r+1)\frac{k_x\pi}{2N_x})}{\sin(\frac{k_x\pi}{2N_x})}, \quad \text{for} \quad k_x = 1, 2, ...N_x \tag{B.18}$$

with the eigenfunctions $\sin(\frac{k_x\pi}{N_x}x)$. Here N_x is the width of the image. The eigenvalues ξ_{k_y} of $\mathrm{B_y}$ is similar. From (B.16) the eigenvalues of L are given by:

$$\xi^L = |w| - \frac{1}{|w|}\xi^2_{k_x}\xi^2_{k_y}. \tag{B.19}$$

Its maximum is:

$$\xi^L_{\max} = |w|, \tag{B.20}$$

and its minimum (using Taylor expansion) is:

$$\begin{aligned} \xi^L_{\min} &\approx \frac{1}{3}((\frac{\pi}{2N_x})^2 + (\frac{\pi}{2N_y})^2)|w|^2 && \text{(B.21)} \\ &\approx \frac{\pi^2}{6}\frac{1}{N}|w|^2, && \text{(B.22)} \end{aligned}$$

where we assume $N_x \approx N_y \approx N^{\frac{1}{2}}$.

Bibliography

[1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Transactions on Graphics (SIGGRAPH 2009)*, 28:21:1–21:12, July 2009. 81, 86, 95

[2] E. H. Adelson. *The New Cognitive Neurosciences, 2nd ed*, chapter 24, Lightness Perception and Lightness Illusions, pages 339–351. MIT Press, 2000. 105, 106

[3] B. L. Anderson and J. Winawer. Image segmentation and lightness perception. *Nature*, 434:79–83, 2005. 106, 110

[4] V. Aurich and J. Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Mustererkennung 1995, 17. DAGM-Symposium*, pages 538–545. Springer-Verlag, 1995. 80

[5] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25:637–645, July 2006. 80, 89

[6] P. Bhat, B. Curless, M. Cohen, and C. L. Zitnick. Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 114–128, 2008. 65, 80

[7] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial (2nd)*. SIAM, 2000. 66, 77

[8] T. F. Chan, S. Osher, and J. Shen. The digital tv filter and nonlinear denoising. *IEEE Transactions on Image Processing (TIP)*, 10:231–241, 2001. 81

[9] P. Chavez. An improved dark-object substraction technique for atmospheric scattering correction of multispectral data. *Remote Sensing of Environment*, 24:450–479, 1988. 17, 18, 46, 56, 58

[10] CIE. *Commission internationale de l'Eclairage proceedings*. Cambridge University Press, Cambridge, 1931. 124

[11] D. Coon. *Psychology: A Modular Approach To Mind And Behavior*. Wadsworth Pub Co, July 2005. 44

[12] T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006. 65

[13] Y. Ding, J. Xiao, and J. Yu. Importance filtering for image retargeting. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 102

[14] N. Draper and H. Smith. *Applied Regression Analysis*. John Wiley, 2 edition, 1981. 68, 84

[15] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (SIGGRAPH 2002)*, pages 257–266, 2002. 80, 89, 91, 97, 101

[16] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing (TIP)*, 11:1141–1151, 2002. 92, 101

[17] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (SIGGRAPH 2008)*, pages 67:1–67:10, 2008. 80, 81, 89

[18] R. Fattal. Single image dehazing. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27:72:1–72:9, August 2008. 6, 7, 19, 21, 23, 39, 47, 57, 59

[19] R. Fattal, M. Agrawala, and S. Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 2007. 80

[20] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (SIGGRAPH 2002)*, 21:249–256, July 2002. 80

[21] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 261–268, 2004. 35

[22] C. Franklin. Summed-area tables for texture mapping. *ACM Transactions on Graphics (SIGGRAPH 1984)*, 18:207–212, January 1984. 73, 127

[23] M. H. Freeman and C. C. Hull. *Optics (11 edition)*. Butterworth-Heinemann, 2003. 60

[24] W. S. Geisler and D. Kersten. Illusions, perception and bayes. *Nature Neuroscience*, 5:508–510, 2002. 104

[25] A. L. Gilchrist. *Seeing Black and White*. Oxford University Press, 2006. 3, 104, 105, 106, 110, 116

[26] A. L. Gilchrist and A. Jacobsen. Lightness constancy through a veiling luminance. *Journal of Experimental Psychology: Human Perception and Performance*, 9(6):936–944, 1983. 48

[27] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2 edition, 2002. 28, 80, 95

[28] L. Grady, T. Schiwietz, and S. Aharon. Random walks for interactive alpha-matting. In *Visualization, Imaging and Image Processing (VIIP)*, 2005. 65, 81, 92

[29] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 4

[30] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1956–1963, 2009. 4

[31] K. He, J. Sun, and X. Tang. Fast matting using large kernel matting laplacian matrices. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2165–2172, 2010. 4, 64, 68, 77

[32] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–8, 2010. 4, 64, 68, 77, 101

[33] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, August 2010. 4

[34] H. Helmholtz. *Helmholtzs Treatise on Physiological Optics*. Optical Society of America, 1866/1924. 104, 106

[35] E. Hering. *Outlines of a Theory of the Light Sense*. Harvard University Press, 1874/1964. 104, 106

[36] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. *ACM Transactions on Graphics (SIGGRAPH 2008)*, pages 70:1–70:7, August 2008. 38, 39, 65, 69, 77

[37] E. R. Kandel, T. M. Jessell, and J. R. Sanes. Chapter 27: Central visual pathways. *Principles of Neural Science (4 ed.)*, 2000. 104, 105

[38] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26, July 2007. 80, 100, 101, 102

[39] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, 27(5):116:1–116:10, 2008. 15, 16, 51, 57

[40] H. Koschmieder. Theorie der horizontalen sichtweite. *Beitr. Phys. Freien Atm.*, 12:171–181, 1924. 7

[41] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23:689–694, August 2004. 65, 92, 100

[42] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 61–68, 2006. 35, 38, 39, 65, 66, 67, 68, 69, 75, 77, 81, 100

[43] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 901–908, 2006. 80

[44] F. Long and D. Purves. Natural scene statistics as the universal basis of color context effects. In *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, volume 100, pages 15190–15193, 2003. 104, 106

[45] E. McCartney. *Optics of the Atmosphere: Scattering by Molecules and Particles*. John Wiley and Sons: New York, 1975. 120, 124

[46] F. Metelli. Stimulation and perception of transparency. *Psychological Research*, 47:185–202, 1985. 106, 116

[47] I. Motoyoshi, S. Nishida, L. Sharan, and E. H. Adelson. Image statistics and the perception of surface qualities. *Nature*, 447:206–209, 2007. 104, 106

[48] S. G. Narasimhan and S. K. Nayar. Chromatic framework for vision in bad weather. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 598–605, June 2000. 13, 14, 51, 56

[49] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision (IJCV)*, 48:233–254, 2002. 6, 7, 8, 60, 120

[50] S. G. Narasimhan and S. K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25:713–724, 2003. 13, 14, 51, 56

[51] S. G. Narasimhan and S. K. Nayar. Interactive deweathering of an image using physical models. In *IEEE Workshop on Color and Photometric Methods in Computer Vision, in conjunction with ICCV*, October 2003. 15, 16

[52] S. K. Nayar and S. G. Narasimhan. Vision in bad weather. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 820–827, 1999. 13, 14, 51, 56

[53] I. Omer and M. Werman. Color lines: Image specific color representation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 946–953, June 2004. 75

[54] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing (2nd)*. Prentice Hall, 1999. 130

[55] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 24–52, 2006. 80

[56] P. Pérez. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH 2003)*, pages 313–318, 2003. 80

[57] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics (SIGGRAPH 2004)*, pages 664–672, 2004. 80, 81, 83, 99

[58] F. Porikli. Constant time o(1) bilateral filtering. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 81, 94

[59] V. S. Ramachandran, P. G. Clarke, and D. Whitteridge. Cells selective to binocular disparity in the cortex of newborn lambs. *Nature*, 1977. 104

[60] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 81, 95, 102

[61] R. Robilotto, B. Khang, and Q. Zaidi. Sensory and physical determinants of perceived achromatic transparency. *Journal of Vision*, 2:388–403, 2002. 106, 116

[62] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23:309–314, 2004. 99

[63] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003. 38, 46, 65, 66, 92, 93

[64] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems (NIPS)*, 2005. 9, 50

[65] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar. Instant dehazing of images using polarization. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 325–332, 2001. 14, 15, 39, 51, 56, 60

[66] S. Shwartz, E. Namer, and Y. Y. Schechner. Blind haze separation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1984–1991, 2006. 15, 51, 56

[67] E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001. 104

[68] M. Singh and B. L. Anderson. Toward a perceptual theory of transparency. *Psychological Review*, 109(3):492–519, 2002. 106, 116

[69] M. Singh and B. L. Anderson. Photometric determinants of perceived transparency. *Vision Research*, 46:879–894, 2006. 106, 116, 117

[70] A. R. Smith and J. F. Blinn. Blue screen matting. *ACM Transactions on Graphics (SIGGRAPH 1996)*, pages 259–268, 1996. 38

[71] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision (IJCV)*, 23:45–78, May 1997. 80

[72] R. L. Solso. *Cognition and the Visual Arts.* MIT Press, 1996. 106

[73] R. L. Solso. *Cognitive psychology (6th ed.).* Pearson Education, 2001. 105, 106

[74] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Box 157, Wellesley, Massachusets 02181, 1986. 65

[75] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23:315–321, August 2004. 80

[76] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(7). 35

[77] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12:513–528, 1990. 35, 65, 66, 73

[78] R. Szeliski. Locally adapted hierarchical basis preconditioning. *ACM Transactions on Graphics (SIGGRAPH 2006)*, pages 1135–1143, 2006. 65, 66

[79] R. Tan. Visibility in bad weather from a single image. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008. 6, 7, 18, 19, 21, 23, 39, 47, 56, 58

[80] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, page 839, 1998. 76, 80, 83, 88

[81] T. Treibitz and Y. Y. Schechner. Polarization: Beneficial for visibility enhancement? In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 15, 51

[82] R. Turner. Vision studies in Germany: Helmholtz versus Hering. *Osiris*, 1993. 104

[83] M. van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letter*, 13:517–521, 1992. 46

[84] J. Wang and M. Cohen. Optimized color sampling for robust matting. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 38, 65, 69, 77

[85] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pages 975–982, 1999. 92

[86] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 2006. 80

[87] World Meteorological Organization. *Manual on Codes*, volume I.2. 2010. 8

[88] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time o(1) bilateral filtering. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 557–564, 2009. 81, 94

[89] Z. Yang and D. Purves. A statistical explanation of visual space. *Nature Neuroscience*, 6:632–640, 2003. 104, 106

[90] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics (SIGGRAPH 2008)*, pages 74:1–74:10, 2008. 80