SEPARATING LAYERS IN IMAGES AND ITS APPLICATIONS

LI YU

NATIONAL UNIVERSITY OF SINGAPORE

2015

SEPARATING LAYERS IN IMAGES AND ITS APPLICATIONS

LI YU
(B.Eng., Beijing University of Posts and Telecommunications)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Signature: Li Yu **Date:** 22 / 10 / 2015

To my beloved wife

Acknowledgements

I would never have been able to finish my thesis without the guidance of my supervisor and my committee members, help from my friends, and support from my family.

I would like to express my deepest gratitude to my advisor, Prof. Michael S. Brown, for his excellent guidance, caring, and patience. He guided me to grow from a young undergraduate to a rigorous researcher and he helped me conquer each difficulty along my study. Everything he taught me, in both research and life, would be a great treasure in my future career. I would also like to thank my committee: Prof. Roger Zimmermann, Prof. Ng Teck Khim and Prof. Tan Ping for their invaluable suggestions and feedbacks at different stages in my PhD study. Their criticisms, comments and advices were critical in making my thesis more accurate and solid. I am also lucky to work with talented people: Prof. Robby T. Tan, Dr. Lu Jiangbo, Dr. Min Dongbo, Dr. Gao Junhong and Guo fangfang on interesting topics, some of which are not included in this thesis [Gao et al. 2013; Guo et al. 2013; Li et al. 2015a].

I am grateful to Lu Zheng,Deng Fanbo, Lin Haiting, Cheng Dongliang, Nguyen Ho Man Rang and other members in vision lab, who as both labmates and friends, were always willing to help and gave their best suggestions. It would be an unforgettable experience to work and play with them all.

Special thanks to my beloved wife, Zhang Xiaoyi, for her love, support and contribution along the way. She was always there cheering me up and stood by me through all the good times and bad. At last, I would like to thank my parents. They were always supporting me and encouraging me with their best wishes.

Contents

Abstract

Image layer separation refers to one series of problems that try to decouple the input image into different component layers. While it has many practical applications, it is very challenging to solve due to its ill-posedness nature. Usually additional information and assumptions are required to make the problem tractable. In this thesis we focus on three types of specific layer separation problems – reflection separation, intrinsic images decomposition, and haze removal. We try to find our solutions to these problems by using different information or imposing new constraints. We first describe our approach for reflection removal, by using a small set of input image captured from different viewpoints. Next a layer separation method for single image input is presented with application to both reflection removal and intrinsic images estimation. The third work targets on the specific problem of recovering the visibility in nighttime haze scene. Besides these three layer separation works we also describe an application of layer separation that aims to remove compression artifacts raised in image contrast boosting. In all four works, our solutions show excellent performance in experimental results that are either on par with or superior to the current state-of-the-art. A summary chapter is included to summarize our contributions. In the end of the summary chapter, potential future directions are also presented.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Problem of Image Layer Separation

Figure 1.1: This image captured before a window contains two layers – background and reflection.

Some computer vision tasks can be considered as layer separation problems where the imaging formation is modeled as being composed of different layers. Figure 1.1 shows such an image which contains an additional layer of reflection as it is taken in front of a window. In this thesis, besides the window reflection problem, we focus on the other three specific layer separation tasks, namely intrinsic

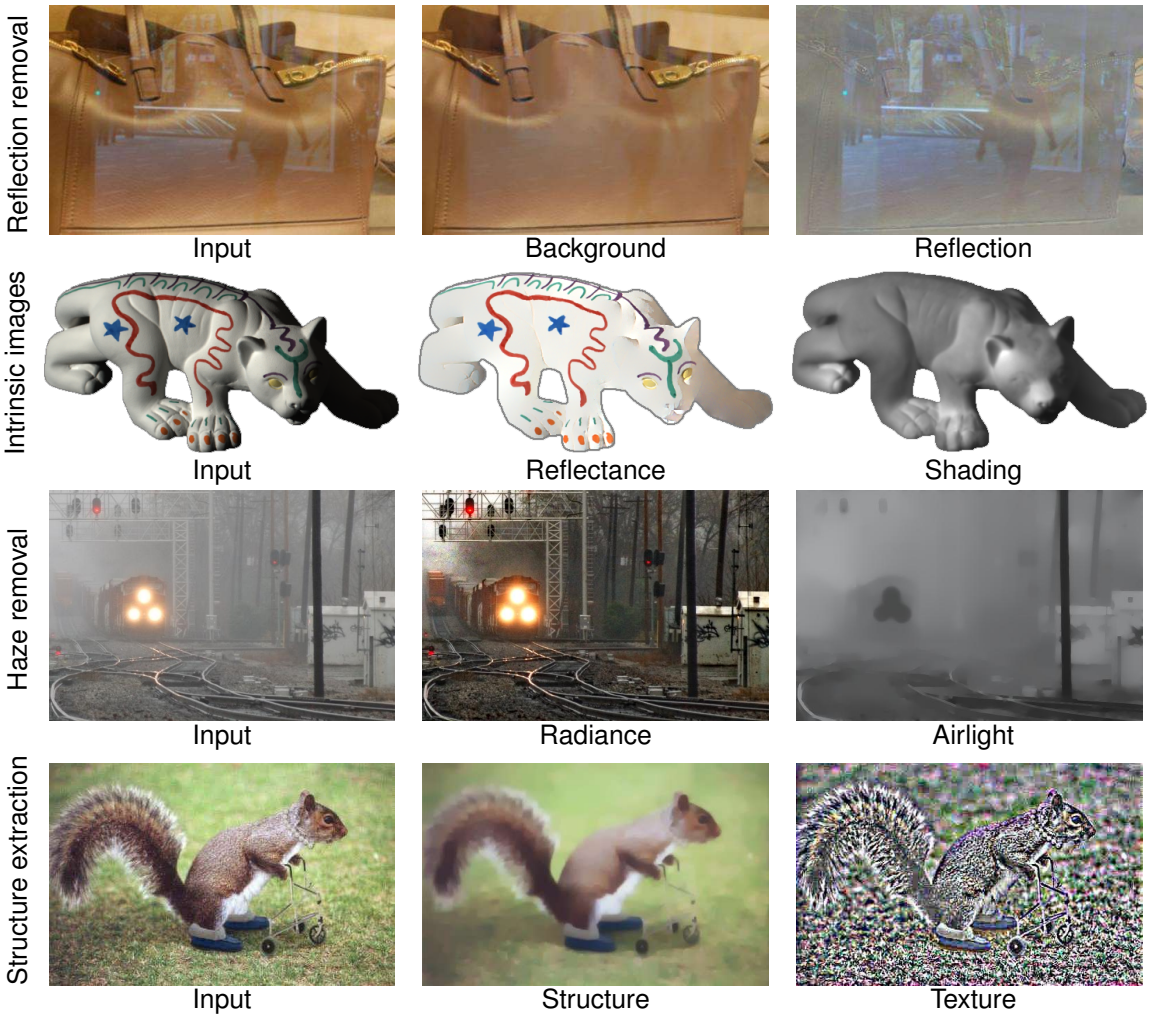


Figure 1.2: Examples of the four layer separations targeted in this thesis. The left column shows the inputs and the rest shows the separated layers from the input.

images, dehazing, and structure-texture decomposition. Given the input image, the task of layer separation tries to decompose it into multiple layers for different purposes. These goals can be briefly summarized as follows (examples are shown in figure 1.2):

Reflection removal tries to separate the reflection interference and recover the clear background scene when imaging a scene behind glass window (the first row in figure 1.2).

Intrinsic images estimation aims to separate a given input image into its material related properties known as reflectance or albedo, and its light-related properties, such as shading (the second row in figure 1.2).

Haze removal also referred to as dehazing, aims to recover the visibility of the scene from haze degradation. It usually models the haze image as a superposition of two layers– direct transmission, describing the scene's radiance with its decay, and airlight which results from scattered light and lead to the shift of scene color (the third row in figure 1.2).

Structure-texture decomposition assumes the image as composed of one structure layer, corresponding to the main large objects in the image, and one textural layer, corresponding to the fine details (the fourth row in figure 1.2).

Successful image layer separation is highly desired. In some cases, we want to remove the influence of one layer and get the clear recovery of another. Reflection removal and haze removal are two such examples where we want a clean image without the degradation by reflection or haze. Layer separation can also be used as a pre-processing step to facilitate other computer vision tasks and computer graphics applications. Take intrinsic image decomposition for example, there are many image editing applications like relighting, recoloring, rematerialing based on intrinsic images (*e.g.* [Barron and Malik 2012]). In addition, the shape-from-shading technique [Zhang et al. 1999], inferring object geometry from shading changes, requires a clean input image of shading which can also be obtained from intrinsic image decomposition. Another layer separation technique, the structure-texture decomposition, is shown to be useful for low level vision tasks like optical flow estimation [Sun et al. 2010], tone mapping [Durand and Dorsey 2002], JPEG artifacts removal [Li et al. 2014] *etc*.

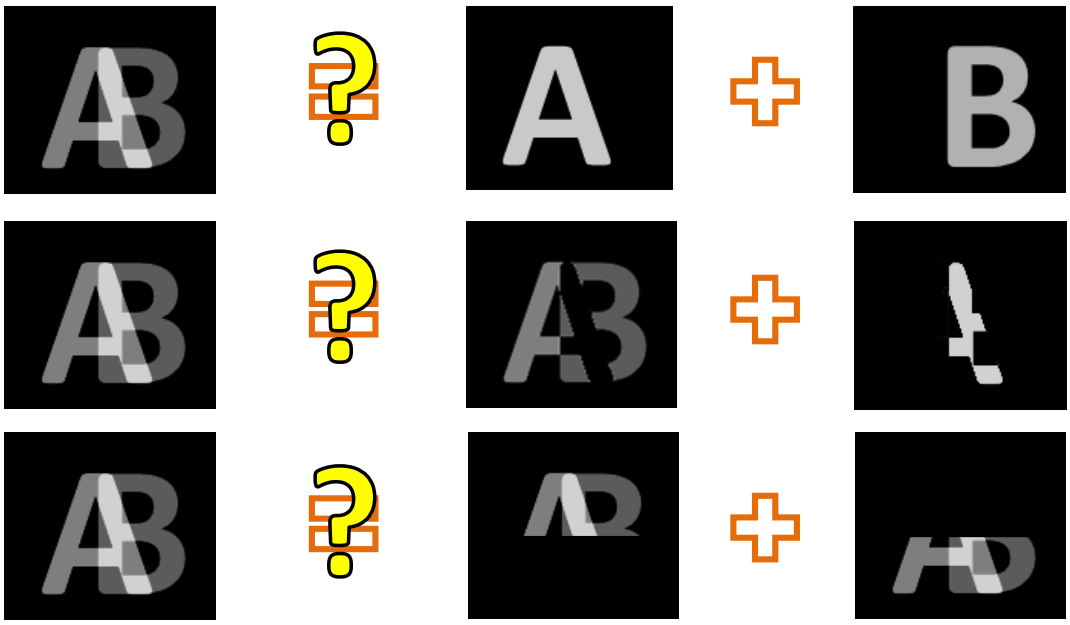


Figure 1.3: This figure shows the ill-posed nature of the layer separation problem. Given an single input image, there are multiple valid solutions as shown here.

Mathematically, many of the layer separation problems can directly take or can be derived into the following form:

$$I = L_1 + L_2, \tag{1.1}$$

where I is the observed image and it is modeled as a linear combination of the two layers L_1 and L_2. The detail of the two layers for different problems will be given in chapter 2. With two unknown layers to be solved and only one known input image, there are infinite number of possible decomposition solutions to the problem as shown in figure 1.3. Therefore the problem of layer separation is inherently an ill-posed one. To make this problem tractable, additional information is needed. With the additional information as constraint, the ill-posedness may be eased and

a most likely explanation of the two layers may be found.

There are many prior works addressing layer separation problems which propose to use information supplied either from user indication (*e.g*. [Levin and Weiss 2007]), from multiple images (*e.g*. [Gai et al. 2012]), or from learned statistics (*e.g*. [Bousseau et al. 2009]). Prior works have also made different assumptions for applying the methods. A complete review and analysis of these related works can be found in chapter 2. In short, while recent methods show the improvements on the performance of layer separation, there are still a few limitations in current methods which restrict their application to practical use. First of all, some methods make very rigid assumptions. For example, the reflection removal methods of [Szeliski et al. 2000; Gai et al. 2012; Guo et al. 2014] assume the scenes are planar. Secondly, some recent methods use complicated modelling and optimization methods, making the algorithm very slow in generating results. For instance, the state-of-the-art intrinsic image method of [Gehler et al. 2011] takes more than 10 minutes to generate a 400×500 decomposition result even on a modern machine. Moreover, there is still need to find better constraint to improve the quality of layer separation result. Layer separation remains a challenging problem and more efficient and robust solutions are needed for practical use.

1.2 Our Contributions

The goal of our work is to design robust and efficient methods for the layer separation problem. Since image layer separation is a broad topic, our study cannot address all layer separation problems at the same time. We just choose reflection separation, intrinsic image decomposition, and nighttime haze removal as our

main focus. We also present an application for the structure-texture decomposition. Specifically, we identified the following issues to address:

1. Reflection separation methods using motion cue require no additional equipment during the capturing, which is promising for casual imaging. However current motion-based approaches require the scene to be planar and the layers to be static. This is rarely met in real cases. We examine how to relax these constraints and make the technique more practical in use.

2. Most of the layer separation methods used multiple images or a single image with user markups. Automatic single image layer separation is the most difficult case due to the severe ill-posedness. We explore the possibility of separating layers in a single image automatically with high quality results. We also are interested in a fast solution to the layer separation problems.

3. Existing dehazing methods always assume a daytime haze scenario. These methods tend to fail for nighttime scenes since the nighttime scenes contain active light sources, such as street lights, car lights, building lights, *etc*. These lights will cause a spatial varying environmental light color and introduce a prominent glow to the scene due to the multiple scattering effect [Narasimhan and Nayar 2003]. These two factors are not accounted for in the current daytime haze model. We examine the daytime dehazing model and adjust it to nighttime scene by introducing new terms.

4. JPEG artifacts reduction is a classic problem in signal processing which has been addressed intensively for decades. Existing methods, however, are unsuitable for dealing with the artifacts amplified in contrast enhancement.

When applied as pre-processing, existing methods tend to over-smooth an image. When applied as post-processing, these are often ineffective at removing the boosted artifacts. We want to design a new pipeline which can suppress the compression artifacts in contrast enhancement. We consider a layered approach which requires layer separation as the first step.

In this thesis, four distinct works are proposed which aim to provide solutions to above issues. These four works correspond to chapters 3, 4, 5, 6 in the thesis. Specifically, the contributions of these works can be summarized as follows:

[Chapter 3] This work introduces an automatic method for removing reflection interference when imaging a scene behind a glass surface. Our approach exploits the subtle changes in the reflection with respect to the background in a small set of images taken at slightly different viewpoints. Unlike previous approaches that exploit motion, our approach does not make any assumptions regarding the background and reflected scenes geometry, nor requires the reflection to be static. This makes our approach practical for use in casual imaging scenarios. Our approach is straight forward and produces good results compared with existing methods. This work has been published in ICCV'2013 [Li and Brown 2013].

[Chapter 4] In this work, we address the problem of extracting two layers from an image where one layer is smoother than the other. This problem arises most notably in intrinsic image decomposition and reflection interference removal. We introduce a novel strategy that regularizes the gradients of the two layers such that one has a long tail distribution and the other a short tail distribution. We formulate our problem in a probabilistic framework and describe an optimization scheme to solve this regularization with only a few iterations. We have applied our

approach to the intrinsic image and reflection removal problems. It demonstrates high quality layer separation on par with other techniques, but being significantly faster than prevailing methods. This work has been published in CVPR'2014 [Li and Brown 2014].

[Chapter 5] To address nighttime dehazing, we introduce a new nighttime haze model that models glow in addition to the direct transmission and airlight. The basic idea is to incorporate a *glow term* into the standard haze model. This results in a new model that has three terms: the direct attenuation, airlight and glow. Working from this new model, we propose an algorithm to first decompose the glow from the input image. This results in a new haze image with reduced glow, but still containing haze and potentially multi-colored light sources. To address this, a spatially varying atmospheric light map which locally encodes different light colors is estimated. From this atmospheric map, we calculate the transmission, and finally obtain the nighttime scene radiance. Our estimated scene radiance has better visibility, with reduced glow and haze and does not suffer from color shifts due to the spatially varying lights. This work has been published in ICCV'2015 [Li et al. 2015b].

[Chapter 6] Unlike previous three works which focus on how to decouple layers in images, this work is an application of layer separation. It adapts layer separation as the key to address the JPEG artifacts which arise in contrast enhancement. While contrast enhancement boosts the image appearance, it can unintentionally boost unsightly image artifacts, especially artifacts from JPEG compression (see figure 1.4 for example). To resolve this problem, we propose a framework that suppresses compression artifacts as an integral part of the contrast enhancement procedure. We show that this approach can produce compelling results superior to those

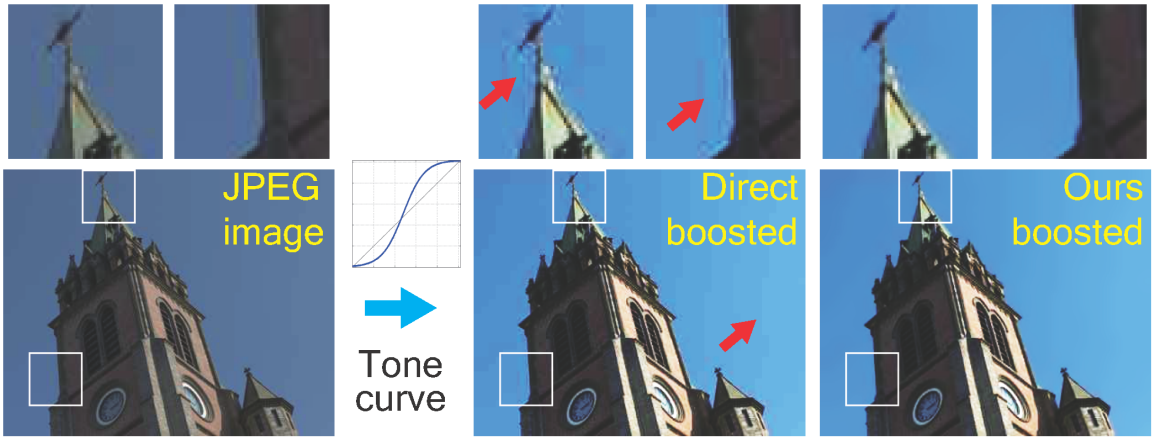


Figure 1.4: This figure shows the problem of arising artifacts in image contrast enhancement. Our method in chapter 6 can provide better results.

obtained by existing JPEG artifacts removal methods for several types of contrast enhancement problems. This work has been published in ECCV'2014 [Li et al. 2014].

1.3 Thesis Organization

This thesis is organized as seven chapters. This first chapter provides an introduction to the layer separation problems with our goals and our contributions presented. The following chapter 2 provides more background on some layer separation problems that are related to our works and gives details to prior works addressing these problems. The next four self-contained chapters (3, 4, 5, 6) describe our works addressing different layer separation problems or developing applications for existing layer separation techniques. The whole thesis is concluded in chapter 7 with future directions discussed.

Chapter 2

Background and Related Work

As we have mentioned in chapter 1, the problem of layer separation is an ill-posed one and needs additional constraints to make it tractable. In this chapter, we will provide more background for the three layer separation problems being presented in chapter 3, 4, 5 with their related prior works being reviewed. For the layer separation application presented in chapter 6, we leave the related work in its own chapter.

2.1 Reflection Removal

There are situations when a scene must be imaged behind a pane of glass (*e.g.* on street, in art museums, see figure 1.1 for example). This is not a conducive setup for imaging as the glass will produce an unwanted layer of reflection in the resulting image. This problem is a typical layer separation problem as described in chapter 1, where the captured image I is a linear combination of a reflection layer L_R and

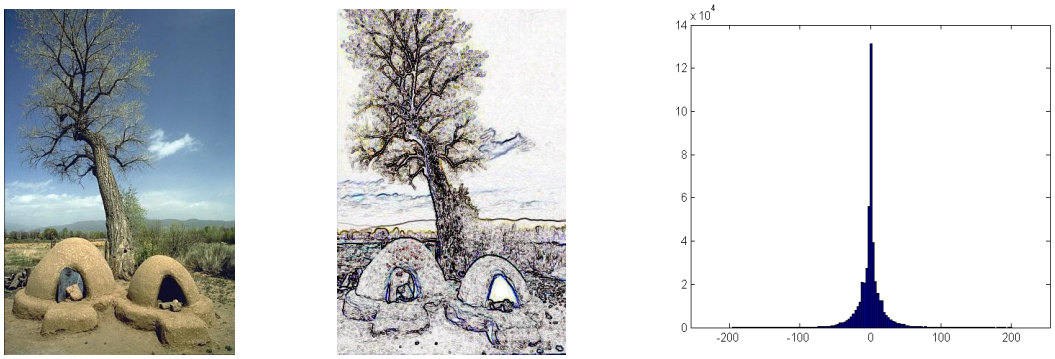


Figure 2.1: Natural image statistics illustration. This shows one natural image, its gradient map and the gradient histogram plot. The gradient histogram of a natural image centers at zero and drops fast which forms a long-tail shape.

the desired background scene, L_B, as follows [Levin and Weiss 2007]:

$$I = L_R + L_B. \tag{2.1}$$

The goal of reflection removal is to separate L_B and L_R from an input image I. In order to solve this ill-posed problem, different methods explore different additional information or constraints. These additional information and constraints come from different types of input used. Therefore according to the input we categorize the works in reflection removal into three classes:

Single image method Levin and Weiss [Levin and Weiss 2007] proposed a method where a user labelled image gradients as belonging to either background or reflection. In addition to make the agreement with the markup, it also employs a strategy that imposes a gradient sparsity prior on the recovered layers. The gradient sparsity prior, also called the natural image prior (see figure 2.1 for illustration), has been shown to be successful in many ill-posed low-level vision problems where

Figure 2.2: User-assisted single image reflection removal [Levin and Weiss 2007] illustration. An input image containing reflection is labelled by the user to indicate where are the background edges and reflection edges (yellow/blur colors). After an optimization, each individual layer can be recovered as shown in the middle and right column. Their associated gradient histograms are also plotted to show that both of them follow the long-tail distribution.

multiple solutions are possible (*e.g.* image deblurring [Fergus et al. 2006]). The basic idea is to require the image gradient histogram to have a long-tail distribution. With proper user labelling, their method produced compelling results as shown in figure 2.2. One limitation with their method is that the results heavily rely on the quality of user mark-ups. It needs intensive and careful user labelling to guarantee a high quality layer separation.

Statistics methods A more common strategy is to use multiple images. Some methods assume a stationary camera that is able to capture a set of images with different mixing of the layers through various means, *e.g.* rotating a polarized lens [Farid and Adelson 1999; Kong et al. 2012; Ohnishi et al. 1996; Shechner et al. 1999; Shechner et al. 2000], changing focus [Schechner et al. 2000], or applying a flash [Agrawal et al. 2005] (see figure 2.3). With some statistic techniques like independent component analysis (ICA), each independent layer can be recovered.

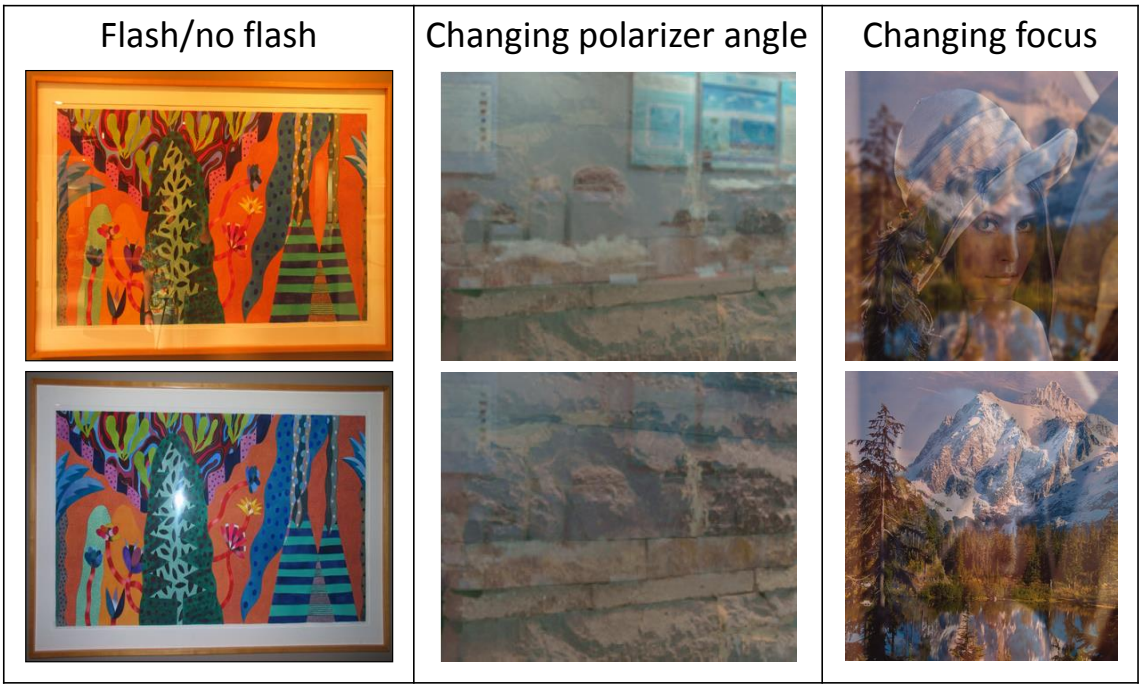


Figure 2.3: This shows some example inputs for statistics based methods that use flash/no flash pair [Agrawal et al. 2005], polarized lens [Kong et al. 2012], and focus [Schechner et al. 2000].

While these approaches demonstrate good results, the ability of controlling focal change, polarization, and flash may not always be possible. Sarel and Irani [Sarel and Irani 2004; Sarel and Irani 2005] proposed video based methods that work by assuming the two layers, reflection and background, to be statistically uncorrelated. These methods can handle complex geometry in the reflection layer, but require a long image sequence such that the reflection layer has significant changes in order for a median-based approach [Weiss 2001] to extract the intrinsic image from the sequence as the initial guess for one of the layers.

Motion based methods The third category of reflection removal techniques exploits motion between the layers present in multiple images. These approaches require no additional equipment as in statistics methods. The user just needs to

Figure 2.4: This shows three input samples from one input sequence containing 15 images for motion based method. Images are from [Guo et al. 2014].

move the camera and capture the same scene at different viewpoints. One issue with changing viewpoint is to handle alignment among the images. Szeliski *et al*. [Szeliski et al. 2000] proposed a method that could simultaneously recover the two layers by assuming they were both static scenes and related by homography transformations. Gai *et al*. [Gai et al. 2012] proposed a similar approach that aligned the images in the gradient domain using gradient sparsity, also assuming static scenes and homography motion. The most recent motion base method, [Guo et al. 2014], again assumed that the motion of each layer is a homography, and proposed a low-rank approximation formulation for the layer separation. One sample input sequence of motion based reflection separation method can be found in figure 2.4. The homography motion used in these methods lead to one major limitation that these methods work well only on planar scenes, like the case shown in figure 2.4. Tsin *et al*. [Tsin et al. 2006] relaxed the planar scene constraint in [Szeliski et al. 2000] and used dense stereo correspondence with stereo matching configuration which restricts the camera motion to unidirectional parallel motion. Therefore, even though motion based methods are more casual in use, the constraints on

scene geometry or restricted motion of the camera limit the type of scenes that can be applied. In chapter 3, we will present our motion based reflection removal solution without using the planar scene and static reflection constraint.

2.2 Intrinsic Image Decomposition

The intrinsic images problem was first raised by Barrow and Tenenbaum [Barrow and Tenenbaum 1978]. The intrinsic image model assumes that an image scene is the product of a scene's reflectance and illumination at each pixel, expressed as

$$I = RL, \tag{2.2}$$

where R is the reflective property or albedo at each pixel and L is the illumination falling on this pixel. The multiplication here is performed element-wise. Intrinsic image decomposition's aim is to estimate R and L given an input I. This problem can be reformulated into the form in equation (1.1) by taking the log at both side, *i.e.*

$$\log(I) = \log(R) + \log(L). \tag{2.3}$$

One of the earliest works addressing intrinsic image decomposition was the Retinex algorithm [Land and McCann 1971] that employed simple heuristics on local edges. Other intrinsic image decomposition methods using multiple images [Weiss 2001] or using user markup [Bousseau et al. 2009] have been proposed and shown to produce good results in some cases. More works focus on automatic single image intrinsic image estimation. Many later works [Grosse et al. 2009; Shen et al. 2008; Tappen et al. 2005] followed the idea of the Retinex method and focused

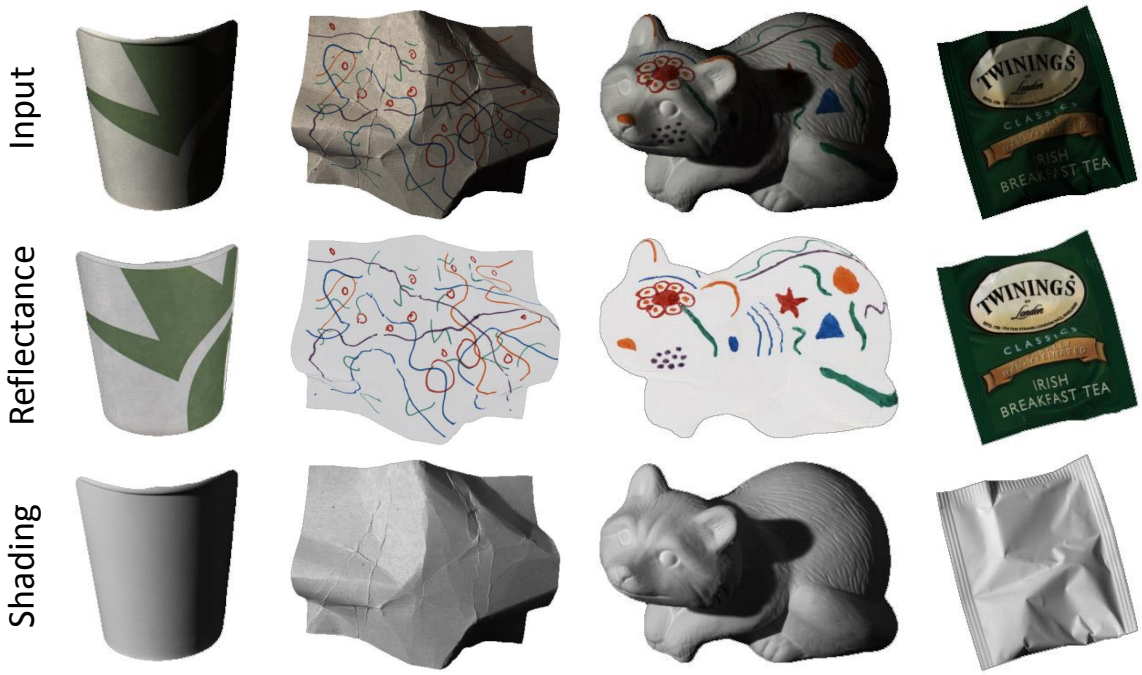


Figure 2.5: This shows four exemplar input and its associated ground truth reflectance and shading images in MIT dataset [Grosse et al. 2009].

on separating reflectance and illumination edges. These methods are referred to as *edge-based methods*. Among them, the work of [Tappen et al. 2005] proposed to use a learned classifier to label edges and then use a Markov Random Field to propagate the initial guesses with regularization. The authors of [Grosse et al. 2009] extended the original Retinex algorithm to work on full RGB color images. Their work also provided a ground-truth dataset (the MIT dataset, see 2.5 for sample images) for intrinsic images containing 16 real objects. Surprisingly, according to their survey [Grosse et al. 2009], just extending the Rentix approach to color images was the top performing algorithm at that time.

Latest methods adopt complicated modelling and optimization. Unlike edge-based methods which rely on local information, these new methods share the idea

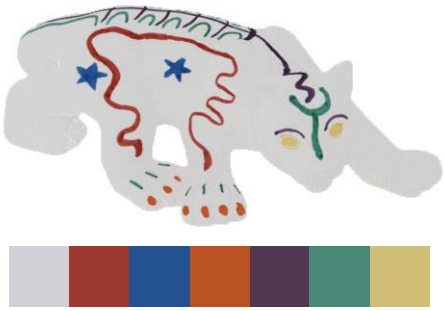


Figure 2.6: This image shows the notion of "global sparsity" in reflectance layer. This ground true reflectance image from MIT dataset [Grosse et al. 2009] contains only seven reflectance colors as listed below.

that there is a sparse set of reflectance value present in the scene (as illustrated in figure 2.6), which is usually referred to as global sparsity prior [Barron and Malik 2012; Gehler et al. 2011; Shen and Yeo 2011; Zhao et al. 2012]. For example, the work in [Gehler et al. 2011] used the Conditional Random Field (CRF) and modeled the global sparsity as a global potential. In the work of [Barron and Malik 2012] the authors forced the global sparse reflectance by minimizing a measure of global entropy on reflectance values. They also proposed to solve intrinsic image decomposition, and shape from shading together in a unified framework. These *global approaches* achieve excellent results on the MIT dataset. However, in order to obtain a global solution, they usually require huge computational complexity in optimization (except [Zhao et al. 2012] which derived a closed-form solution). In chapter 4, we will propose a new edge-based method which can get comparable results with those global methods, while being significantly faster in computation.

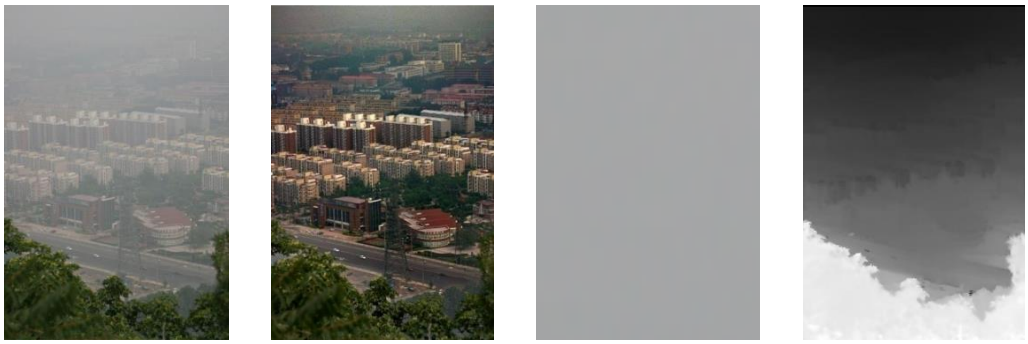


Figure 2.7: This shows the components in haze removal using standard haze model.

2.3 Haze Removal

The presence of haze significantly degrades the quality of an image, which is due to tiny particles floating in the air that adversely scatter the line of sight of lights rays entering the imaging sensor. In particular, light rays are *scattered-out* to directions other than the line of sight, while other light rays are *scattered-in* to the line of sight. The scattering-out process causes the scene reflection to be attenuated. The scattering-in process creates the appearance of a particles-veil (also known as airlight) that washes out the visibility of the scene.

A number of methods have been developed to address visibility enhancement for hazy or foggy scenes from a single image (*e.g.* [Tan 2008; Fattal 2008; He et al. 2011; Meng et al. 2013; Tang et al. 2014; Fattal 2014]). The key to their success relies on the optical model and the possible estimation of its parameters, particularly the atmospheric light and transmission. The standard haze model [Koschmieder 1925] describes a hazy scene as a linear combination of the direct transmission and airlight as following:

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x})t(\mathbf{x}) + \mathbf{L}(1 - t(\mathbf{x})), \tag{2.4}$$

where $\mathbf{I}(\mathbf{x})$ is the observed color at pixel $\mathbf{x}$, $\mathbf{R}(\mathbf{x})$ is the scene radiance when there is no haze or fog particles. There is one more fractional term t which is the transmission factor indicating the portion of scene reaching the camera. The transmission factor is related to the scene depth as $t(\mathbf{x}) = \exp(-\beta d(x))$, where the term β is the attenuation factor of the particles, and d is distance between the camera and the object or scene. The two terms $\mathbf{R}(\mathbf{x})$ and $t(\mathbf{x})$ multiply together to form the direct transmission. The last term $\mathbf{L}(1 - t(\mathbf{x}))$ is the airlight, representing the particle veil induced by the scattering-in process by the atmospheric light, $\mathbf{L}$, which is assumed to be globally uniform. Example of these components are illustrated in figure 2.7.

Given a color image $\mathbf{I}$, the main goal of single image dehazing is to recover the scene's reflection $\mathbf{R}$, or at least to enhance the visibility of $\mathbf{R}$. The most commonly employed steps to achieve this goal is to first estimate the globally uniform atmospheric light, $\mathbf{L}$, and then to estimate the transmission $t(\mathbf{x})$. Having obtained these two components, estimating $\mathbf{R}$ for every pixel becomes straightforward. Most of the methods assume that the atmospheric light is present in the input image and can be estimated by the brightest region in the image. Although, this estimation is a crude approximation, in most cases it works adequately. An exception applies to [Fattal 2014], which utilizes the atmospheric light estimation proposed in [Sulami et al. 2014]. The method [Sulami et al. 2014] estimates the globally uniformed color of the atmospheric light by using small patches of different reflections that form color lines in RGB space and estimates the magnitude of the atmospheric

light by minimizing the distance between the estimated shading and the estimated transmission for different levels of transmission. Having estimated the atmospheric light, these methods estimate the transmission using various cues, such as local contrast [Tan 2008], independence between shading and transmission [Fattal 2008], dark channel [He et al. 2011], boundary constraint [Meng et al. 2013], learned regressor [Tang et al. 2014] and more. The methods differentiate themselves from each other mainly on the cues used for estimating the transmission. All of these aforementioned methods use the standard dehaze model and work well on daytime haze scenes. We will show in chapter 5, for nighttime haze scenes, this standard model and the methods based on this model may not be suitable. We will also present our solution to this problem in chapter 5.

2.4 Summary

This chapter has provided detail background for the problems of reflection removal, intrinsic image decomposition, and haze removal. Related works for these problems have been reviewed with their limitations discussed. Our works in this thesis focus on solving these limitations as mentioned in section 1.2. In the next chapter, we begin to present our first work for motion based reflection removal .

Chapter 3

Exploiting Reflection Change for Automatic Reflection Removal

In this chapter we describe our multi-view approach for the problem of reflection removal. Key to our idea is to use dense correspondence to align the images from different viewpoints. This will remove the planar constraint underlying homography motion used in previous methods. After the alignment, a pixel-wise comparison can be made across the input set that gradients with variation across the image set are assumed to belong to the reflected scenes while constant gradients are assumed to belong to the desired background scene. By correctly labelling gradients belonging to reflection or background, the background scene can be separated from the reflection interference. Experimental comparisons with prior works and a discussion will also be presented at the end of this chapter.

3.1 Introduction

As described in chapter 2, an image I captured before glass window can be modelled as a linear combination of a reflection layer L_R and the desired background scene L_B as $I = L_R + L_B$. In order to solve this problem, one single image reflection removal method by Levin and Weiss [Levin and Weiss 2007] requires additional user mark-ups as constraint and force gradient sparsity prior to both of background and reflection layers. Other methods usually use multiple images. Techniques closer to ours exploit motion between the layers present in multiple images. However, as pointed out in chapter 2, their constraints on scene geometry or assumed motion of the camera limit the type of scenes that can be processed.

Our proposed method in this chapter builds on the single-image approach [Levin and Weiss 2007] , but removes the need for user markups by examining the relative motion in a small set (*e.g.* 3-5) of images to automatically label gradients as either reflection or background. This is done by first aligning the images using SIFT-flow and then examining the variation in the gradients over the image set. Gradients with more variation are assumed to be from reflection while constant gradients are assumed to be from the desired background. While a simple idea, this approach does not impose any restriction on the scene or reflection geometry. This allows a more practical imaging setup that is suitable for handheld cameras.

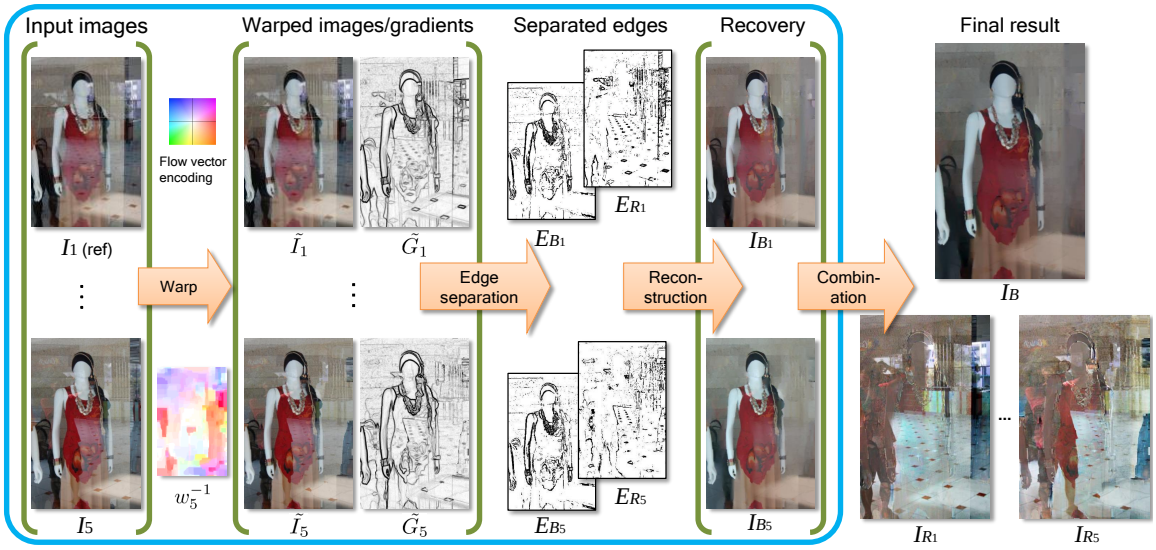


Figure 3.1: This figure shows the pipeline of our approach: 1) warping functions are estimated to align the inputs to a reference view; 2) the edges are labelled as either background or foreground based on gradient frequency; 3) a reconstruction step is used to separate the two layers; 4) all recovered background layers are combined together to get the final recovered background.

3.2 Method

3.2.1 Imaging Assumption and Procedure

Our approach takes as input a small set of k images taken of the scene from slightly varying viewpoints. Following the motion base method, we assume that when the background is captured from different points of view, the background and the reflection layers undergo different motions due to their different distance to the transparent layer. This configuration is shown in figure 3.2. These input images are assumed to be related by a warping, such that the background is constant and the reflection layer is changing. This relationship can be expressed as:

$$I_i = w_i(L_{R_i} + L_B), \tag{3.1}$$

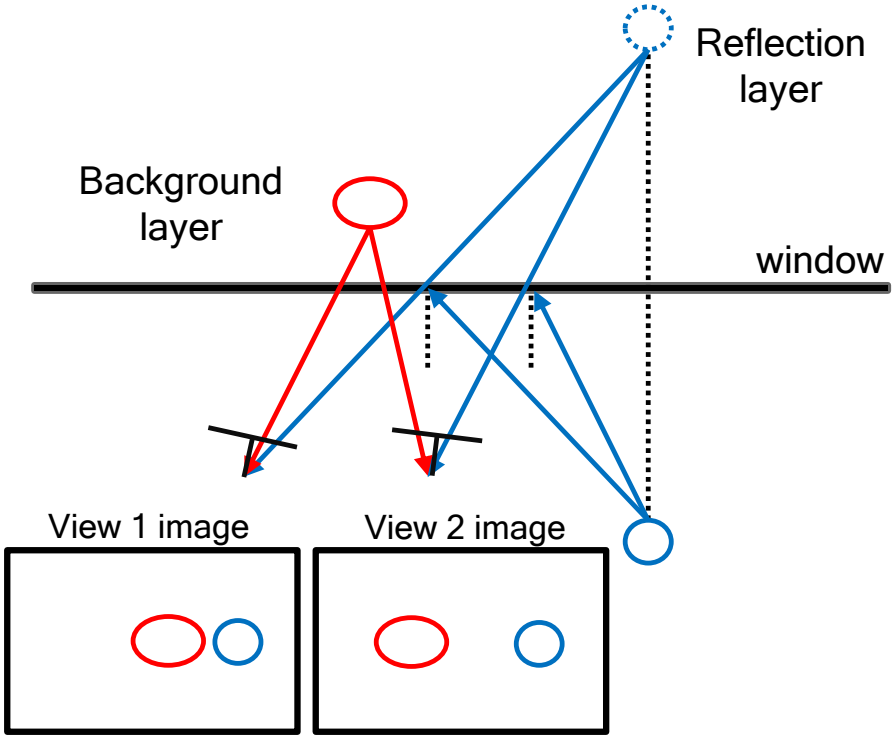


Figure 3.2: This shows the configuration of motion based reflection separation method. Different displacements for the two layers appear in two different views.

where I_i is the i-th mixture image, $\{w_i\}$, $i = 1, \ldots, k$ are warping functions caused by the camera viewpoint change with respect to a reference image (in our case I_1). Assuming we can estimate the inverse warps, w_i^{-1}, where w_1^{-1} is the identity, we get the following relationship:

$$w_i^{-1}(I_i) = L_{R_i} + L_B. \tag{3.2}$$

Even though L_B appears static in the mixture image, the problem is still ill-posed since we have an infinitely possible L_B. However, the presence of a static L_B in the image set makes it possible to identify gradient edges of the background layer L_B and edges of the changing reflection layers L_{R_i}. More specifically, edges in L_B are assumed to appear every time in the image set while the edges in the reflection

layer L_{R_i} are assumed to vary across the set. This means edges can be labeled based on the frequency of a gradient appearing at a particular pixel across the aligned input images. After labeling edges as either background or reflection, we can reconstruct the two layers using an optimization that imposes the sparsity prior on the separated layers as done by [Levin and Weiss 2004; Levin and Weiss 2007]. Figure 3.1 shows the processing pipeline of our approach. Each step is described in the following sections.

3.2.2 Warping

Our approach begins by estimating warping functions, w_i^{-1}, to register the input to the reference image. Previous approaches estimated these warps using homographies [Gai et al. 2008; Gai et al. 2012; Szeliski et al. 2000], however, this planarity constraint often leads to regions in the image with misalignments when the scene is not planar.

To remove the planar constraint behind global homography, we seek a dense correspondence estimation across different views. A commonly used dense correspondence technique is the optical flow estimation [Sun et al. 2014; Li et al. 2015a] which estimates at each point a 2D translation indicating the displacement of the pixel. However in our case the scene is interrupted by the reflection, leading to the brightness constancy assumption in the optical flow estimator to fail. This is demonstrated empirically as shown in figure 3.3 with a top performance optical flow estimator Classic+NLP [Sun et al. 2014]. Our observation is that even with moderate reflection interference, the structures of the images are mostly the same across the image set. This led us to try structural descriptor based dense corre-

Figure 3.3: We have compared optical flow with SIFT flow in aligning the input set. For the optical flow, we used the popular Classic+NL method [Sun et al. 2014].

spondence method. We adopt common-used SIFT-flow estimator [Liu et al. 2011] and found it is able to handle the moderate interference of reflection and provide a dense warp suitable to bring the images into alignment as also illustrated in figure 3.3.

Our implementation fixes I_1 as the reference, then uses SIFT-flow to estimate the inverse-warping functions $\{w_i^{-1}\}$, $i = 2, \ldots, k$ for each of the input images $I_2, \ldots, I_k$ against I_1. We also compute the gradient magnitudes G_i of the each input image

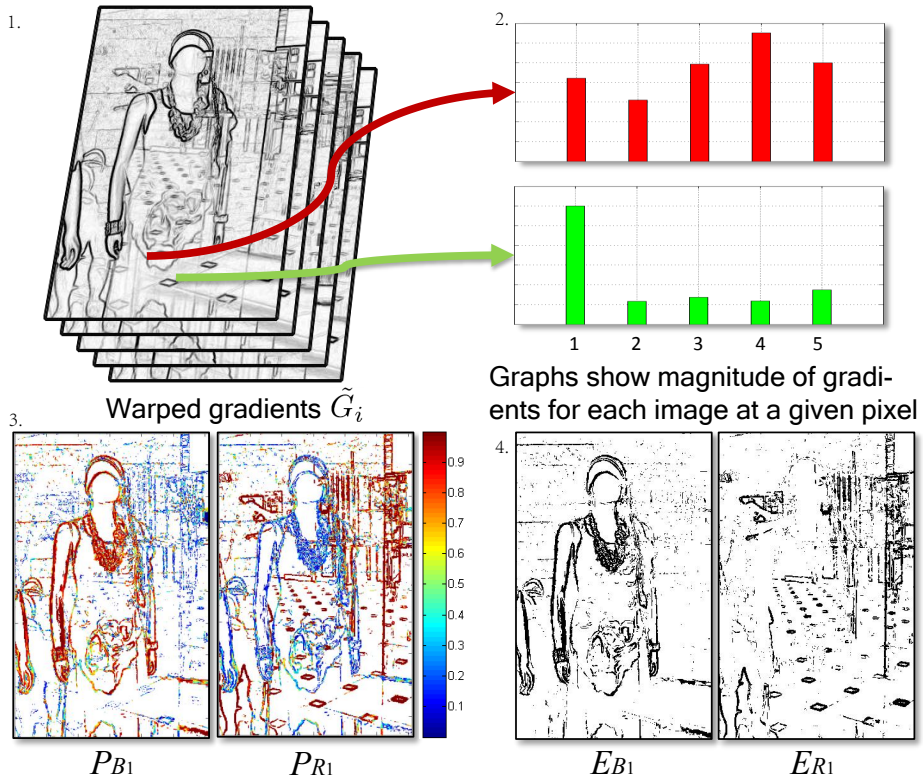


Figure 3.4: Edge separation illustration: 1) shows the all $\tilde{G}$ gradient maps – in this case we have five input images; 2) plots the gradient values at two position across the five images - top plot is a pixel on a background edge, bottom plot is a pixel on a reflection edge; 3) shows the probability map estimated for each layer; 4) Final edge separation after thresholding the probability maps.

and then warp the images I_i as well as the gradient magnitudes G_i using the same inverse-warping function w_i^{-1}, denoting the warped images and gradient magnitudes as $\tilde{I}_i$ and $\tilde{G}_i$.

3.2.3 Edge separation

Our approach first identifies salient edges using a simple threshold on the gradient magnitudes in $\tilde{G}_i$. The resulting binary edge map is denoted as E_i. After edge

detection, the edges need to be separated as either background or foreground in each aligned image $\tilde{I}_i$. As previously discussed, the edges of the background layer should appear frequently across all the warped images while the edges of the reflection layer would only have sparse presence. To examine the sparsity of the edge occurrence, we use the following measurement:

$$\Phi(\mathbf{y}) = \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{y}\|_1^2}, \tag{3.3}$$

where $\mathbf{y}$ is a vector containing the gradient magnitudes at a given pixel location. Since all elements in $\mathbf{y}$ are non-negative, we can rewrite equation (3.3) as $\Phi(\mathbf{y}) = \sum_{i=1}^{k} y_i^2 / (\sum_{i=1}^{k} y_i)^2$. This measurement can be considered as a L_1 normalized L_2 norm. It measures the sparsity of the vector which achieves its maximum value of 1 when only one non-zero item exists and achieves its minimum value of $\frac{1}{k}$ when all items are non-zero and have identical values (*i.e.* $y_1 = y_2 = \ldots = y_k > 0$). This measurement is used to assign two probabilities to each edge pixel as belonging to either background or reflection.

We estimate the reflection edge probability by examining the edge occurrence, as follows:

$$P_{R_i}(\mathbf{x}) = s\left(\frac{\sum_{i=1}^{k} \tilde{G}_i(\mathbf{x})^2}{(\sum_{i=1}^{k} \tilde{G}_i(\mathbf{x}))^2} - \frac{1}{k} \right), \tag{3.4}$$

where, $\tilde{G}_i(\mathbf{x})$ is the gradient magnitude at pixel $\mathbf{x}$ of $\tilde{I}_i$. We subtract $\frac{1}{k}$ to move the smallest value close to zero. The sparsity measurement is further stretched by a sigmoid function $s(t) = (1 + e^{-(t-0.05)/0.05})^{-1}$ to facilitate the separation. The background edge probability is then estimated by:

$$P_{B_i}(\mathbf{x}) = s\left(-\left(\frac{\sum_{i=1}^{k} \tilde{G}_i(\mathbf{x})^2}{(\sum_{i=1}^{k} \tilde{G}_i(\mathbf{x}))^2} - \frac{1}{k} \right) \right), \tag{3.5}$$

where $P_{B_i}(\mathbf{x}) + P_{R_i}(\mathbf{x}) = 1$. These probabilities are defined only at the pixels that are edges in the image. We consider only edge pixels with relatively high probability in either the background edge probability map or reflection edge probability map. The final edge separation is performed by thresholding the two probability maps as:

$$E_{B_i/R_i}(\mathbf{x}) = \begin{cases} 1, & E_i(\mathbf{x}) = 1 \text{ and } P_{B_i/R_i}(\mathbf{x}) > 0.6; \\ 0, & \text{otherwise.} \end{cases}$$

Figure 3.4 shows the edge separation procedure.

3.2.4 Layer Reconstruction

With the separated edges of the background and the reflection, we can reconstruct the two layers. Levin and Weiss [Levin and Weiss 2007] showed that the long tailed distribution of gradients in natural scenes is an effective prior in this problem. In our work, we use Laplacian approximation since the L_1 norm converges quickly with good results. For each image $\tilde{I}_i$, we try to maximize the probability $P(L_{B_i}, L_{R_i})$ in order to separate the two layers and this is equivalent to minimizing the cost $-\log P(L_{B_i}, L_{R_i})$. Following the same deduction in [Levin and Weiss 2004], with the independent assumption of the two layers (*i.e.* $P(L_{B_i}, L_{R_i}) = P(L_{B_i}) \cdot P(L_{R_i})$), the objective function becomes:

$$\begin{aligned} J(L_{B_i}) = & \sum_{\mathbf{x},n} |(L_{B_i} * f_n)(\mathbf{x})| + |((\tilde{I}_i - L_{B_i}) * f_n)(\mathbf{x})| \\ & + \lambda \sum_{\mathbf{x},n} E_{B_i}(\mathbf{x}) |((\tilde{I}_i - L_{B_i}) * f_n)(\mathbf{x})| \\ & + \lambda \sum_{\mathbf{x},n} E_{R_i}(\mathbf{x}) |(L_{B_i} * f_n)(\mathbf{x})|, \end{aligned} \tag{3.6}$$

where f_n denotes the derivative filters and $*$ is the 2D convolution operator. For f_n, we use two orientations and two degrees (first order and second order) derivative filters. While the first term in the objective function keeps the gradients of the two layer as sparse as possible, the last two terms force the gradients of L_{B_i} at edges positions in E_{B_i} to agree with the gradients of input image $\tilde{I}_i$ and gradients of L_{R_i} at edge positions in E_{R_i} agree with the gradients of $\tilde{I}_i$. Following the procedure of iterative reweighted least square (IRLS) [Meer 2004] optimization technique, this objective function can be further rewritten in the following form:

$$\begin{aligned} J(L_{B_i}) = & \sum_{\mathbf{x},n} \frac{1}{W_{B_i}(\mathbf{x})} \|(L_{B_i} * f_n)(\mathbf{x})\|^2 + \frac{1}{W_{R_i}(\mathbf{x})} \|((\tilde{I}_i - L_{B_i}) * f_n)(\mathbf{x})\|^2 \\ & + \lambda \sum_{\mathbf{x},n} E_{B_i}(\mathbf{x}) \frac{1}{W_{R_i}(\mathbf{x})} \|((\tilde{I}_i - L_{B_i}) * f_n)(\mathbf{x})\|^2 \\ & + \lambda \sum_{\mathbf{x},n} E_{R_i}(\mathbf{x}) \frac{1}{W_{B_i}(\mathbf{x})} \|(L_{B_i} * f_n)(\mathbf{x})\|^2, \end{aligned} \tag{3.7}$$

where the weights $W_{B_i}(\mathbf{x}) = |(L_{B_i} * f_n)(\mathbf{x})|$ and $W_{R_i}(\mathbf{x}) = |((\tilde{I}_i - L_{B_i}) * f_n)(\mathbf{x})|$. Next, iterating between solving the least square system with fixed weights and updating the weights can efficiently solve this optimization problem.

3.2.5 Combining the Results

Our approach processes each image in the input set independently. Due to the reflective glass surface, some of the images may contain saturated regions from specular highlights. When saturation occurs, we can not fully recover the structure in these saturated regions because the information about the two layers are lost.

In addition, sometimes the edges of the reflection in some regions are too weak to be correctly distinguished. This can lead to local regions in the background

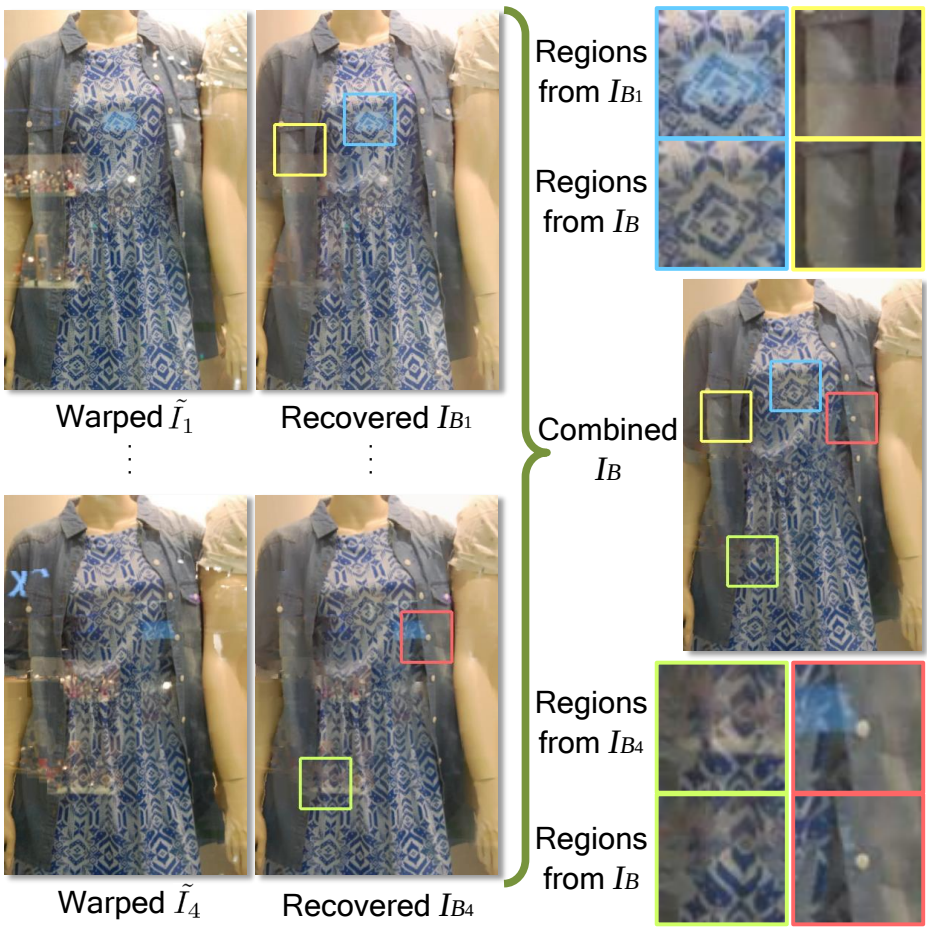


Figure 3.5: This figure shows our combination procedure.

where the reflection is still present. These erroneous regions are often in different places in each input image due to changes in the reflection. In such cases, it is reasonable to assume that the minimum value across all recovered background layers may be a proper approximation of the true background. As such, the last step of our method is to take the minimum of the pixel value of all reconstructed background images as the final recovered background, as follows:

$$L_B(\mathbf{x}) = min_i\, L_{B_i}(\mathbf{x}). \tag{3.8}$$

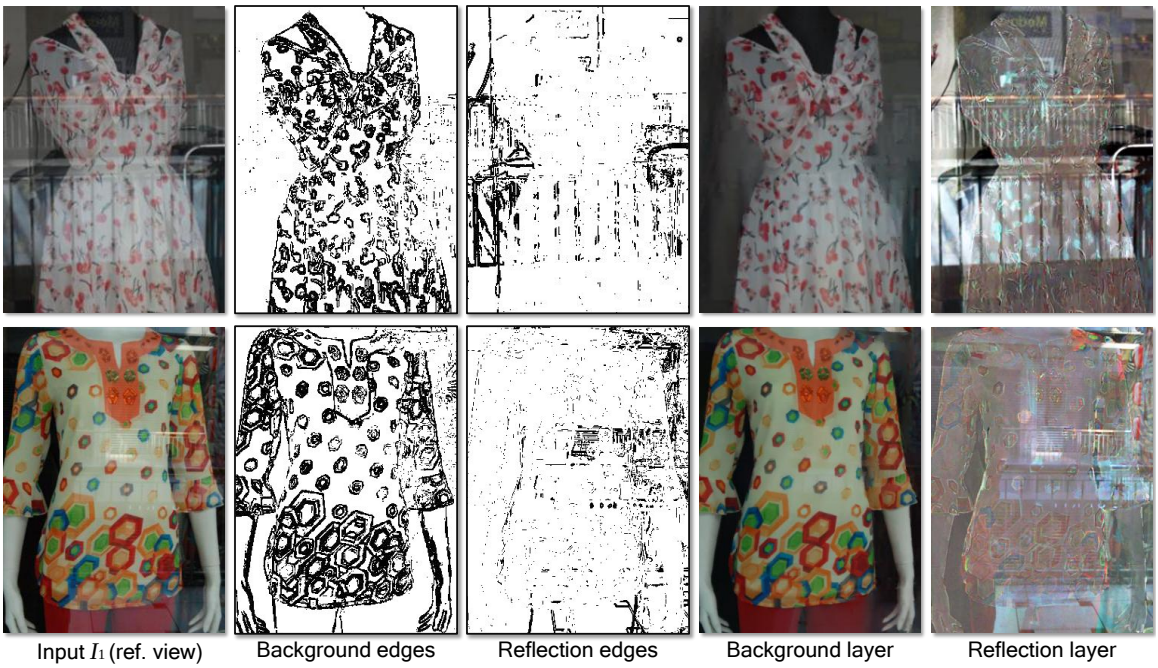


Figure 3.6: Example of edge separation results and recovered background and foreground layer using our method.

Based on this, the reflection layer of each input image can be computed by $L_{R_i} = \tilde{I}_i - L_B$. The effectiveness of this combination procedure is illustrated in figure 3.5. The recovered background on each single image is good at first glance but may have reflection remaining in local regions. A simple minimum operator combining all recovered images gives a better result in these regions.

3.3 Results

In this section, we present the experimental results of our proposed method. The effectiveness of our method is demonstrated by comparing ours with previous methods. All of the data captured are real scenes of objects behind glass windows

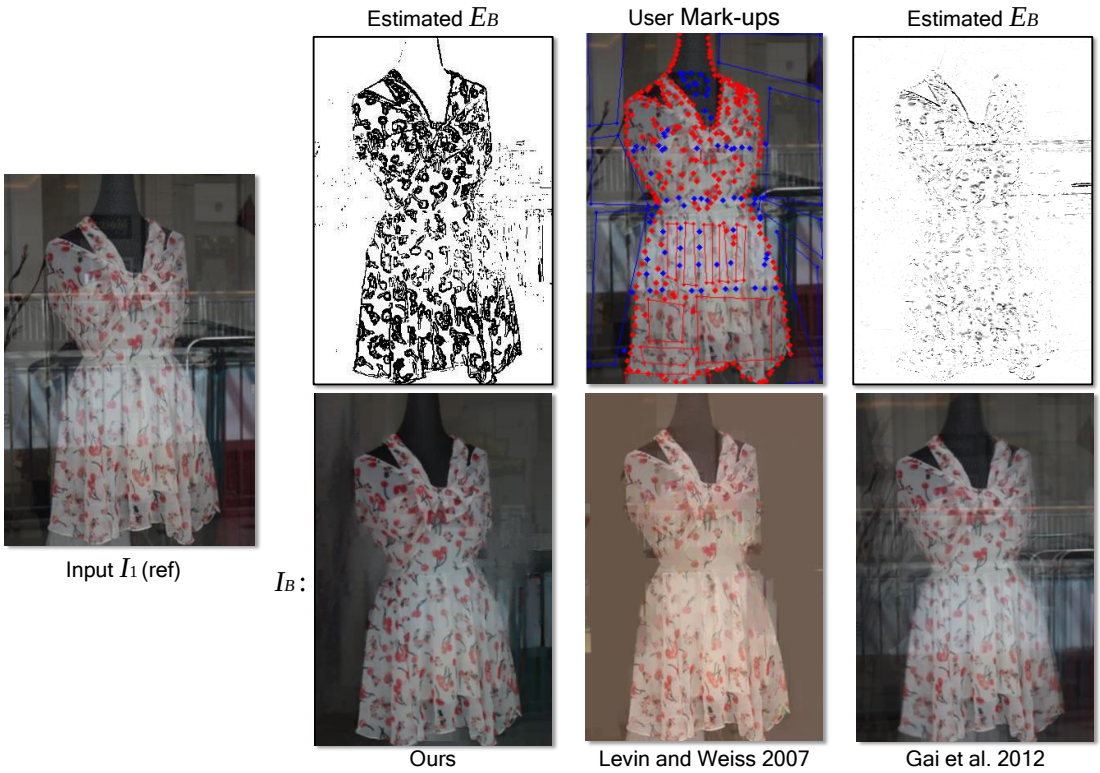


Figure 3.7: This figure shows a comparison between our method and those in [Levin and Weiss 2007] and [Gai et al. 2012].

and share the problem of reflection interference. They are taken under different lighting conditions from both indoor and outdoor, *e.g.* on shop streets, in museums *etc*. We have tested the reflection separation task using our method and previous methods. Input sequences range from three to five images of size 400*500. The experiments were conducted on an Intel i7 PC (3.4GHz CPU, 8.0GB RAM). The code was implemented in Matlab.

Figure 3.6 shows two examples of our edge separation results and final reconstructed background layers and reflection layers. Our method provides a clear separation of the edges of the two layers which is crucial in the reconstruction step. Figure 3.10 shows more reflection removal results of our method.

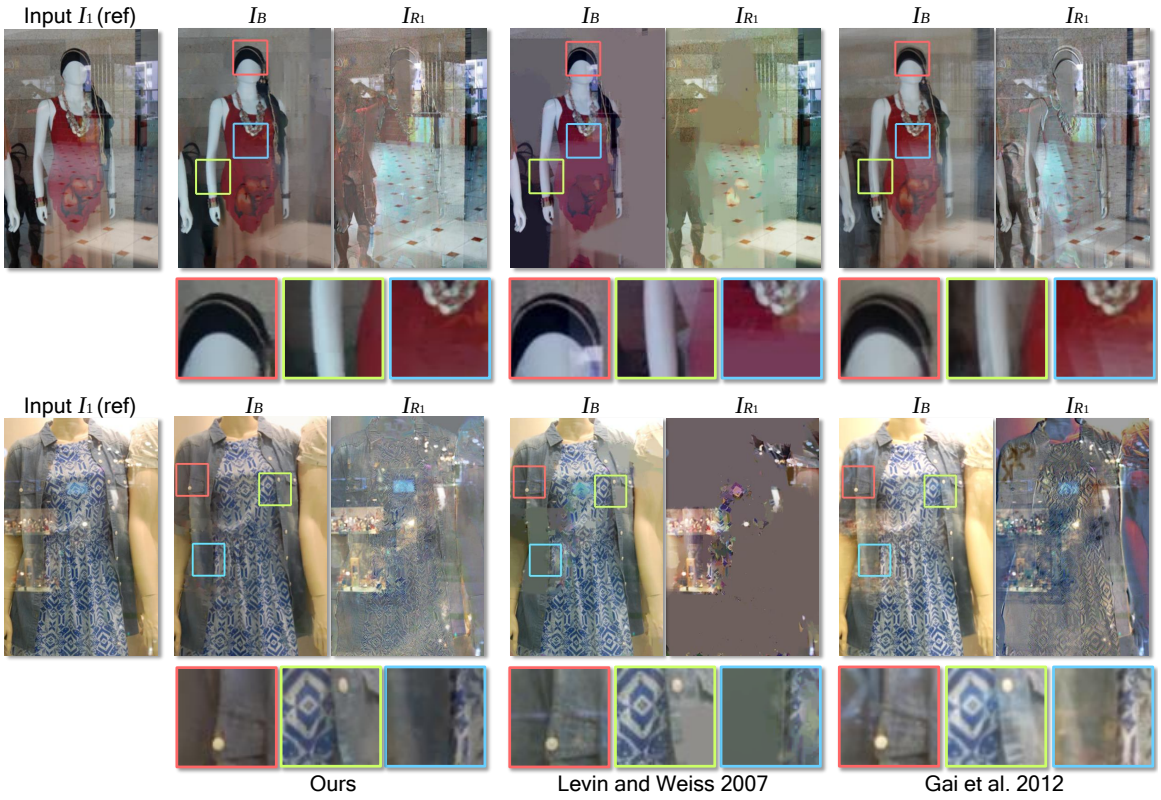


Figure 3.8: Two more examples of reflection removal results of our method and those in [Levin and Weiss 2007] and [Gai et al. 2012].

We also compare our methods with those in [Levin and Weiss 2007] and [Gai et al. 2012] as shown in figure 3.7. For these two methods, we use the source code from the authors to generate the results. In the upper row we show the crucial intermediate step of labelling out the background edges. Unlike the automatic labelling in [Gai et al. 2012] and our approach, in this step the method of [Levin and Weiss 2007] needs the user to provide markups. The bottom row shows the corresponding final reflection removal results from those edge labelling. As can be seen, our method can get a clearer and more accurate edge map of the background than the method in [Gai et al. 2012]. This high quality edge map contributes to better layer recovery. The method of [Gai et al. 2012] gives less edges labelling due

to the unsatisfactory alignment using the global transformation in their method. For the method in [Levin and Weiss 2007], even with these elaborate markups, the reflection removal result is not as satisfactory as ours. The comparisons between our and [Levin and Weiss 2007] are not entirely fair since [Levin and Weiss 2007] uses single image to generate the result, while we have the advantage of the entire set.

Figure 3.8 shows two more comparisons of the reflection removal results. Background and reflection layers are displayed side by side with zoomed in patches listed below for better visualization. Our result arguably provides the best result. The results of [Levin and Weiss 2007] still exhibited some edges from different layers even with the elaborate user mark-ups. This may be fixed by going back to further refine the user markups. But in the heavily overlapping edge regions, it is really challenging for users to indicate the edges. If not clearly indicated, then it tends to over smooth one layer in the result. For the method of [Gai et al. 2012], since it uses global transformations to align images, local misalignment effects often appear in the final recovered background image. Also, their approach uses all the input image into the optimization to recover the layers. This may lead to a result that has edges from different reflection layers of different images mixed and appear as ghosting effect in the recovered background image. For heavily saturated regions, none of the two previous methods can give visually plausible results like ours. This is because our final combination step is specifically designed for handling this situation.

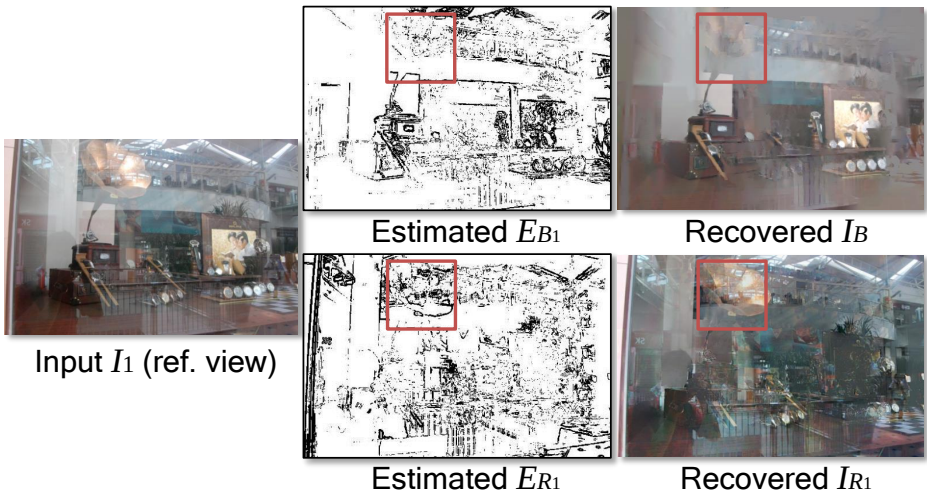


Figure 3.9: A failure case of our approach due to dominant reflection against the background in some regions (*i.e.* the upper part of the phonograph).

3.4 Discussion and Conclusion

We have presented our method to automatically remove reflection interference due to a glass surface. Our approach works by capturing a set of images of a scene from slightly varying viewpoints. The images are then aligned and edges are labeled as belonging to either background or reflection. This alignment was enabled by SIFT-flow, whose robustness to the reflection interference enabled our method. When using SIFT-flow, we assume that the background layer will be the most prominent and will provide proper SIFT features for matching in the SIFT-Flow algorithm. While we found this to work well in practice, images with very strong reflection can produce poor alignment as SIFT-flow may attempt to align to the foreground which is changing. This will cause problems in the subsequent edge separation and final reconstruction. Figure 3.9 shows such a case. While these failures may be handled by user input, it is a notable issue.

Another challenging issue is when the background scene has large homogeneous regions. In such cases there are no edges to be labeled as background. This makes subsequent separation challenging, especially when the reflection interference in these regions is weak but still visually noticeable. While this problem is not unique to our approach, it is an issue to consider. We also found that by combining all the background results of the input images we can overcome local regions with high saturation. While a simple idea, this combination strategy can be incorporated into other techniques to improve their results. Lastly, we believe reflection removal is an application that would be welcomed on many mobile devices, however, the current processing time is still too long for real world use. Exploring ways to speed up the processing pipeline is an area of interest for future work.

In the next chapter, we will present our second work in layer separation which can extract two layers from one image while one layer is smoother than the other.

Figure 3.10: More results of reflection removal using our method in varying scenes (*e.g.* art museum, street shop,*etc.*).

Chapter 4

Single Image Layer Separation using Relative Smoothness

Our layer separation method in chapter 3 requires multiple images as input. In this chapter, we present another method which takes only a single image input for layer separation. Layer decomposition from a single image is the most ill-posed case and solutions require additional constraints to be enforced. We show the separation is possible when one layer is smoother than the other. Two example problems are intrinsic image decomposition and reflection interference removal. We introduce a novel strategy that regularizes the gradients of the two layers to different distributions and describe an optimization scheme to find the most likely solution under this regularization with only a few iterations. The experiments on benchmark dataset and real images are presented with a discussion at the end.

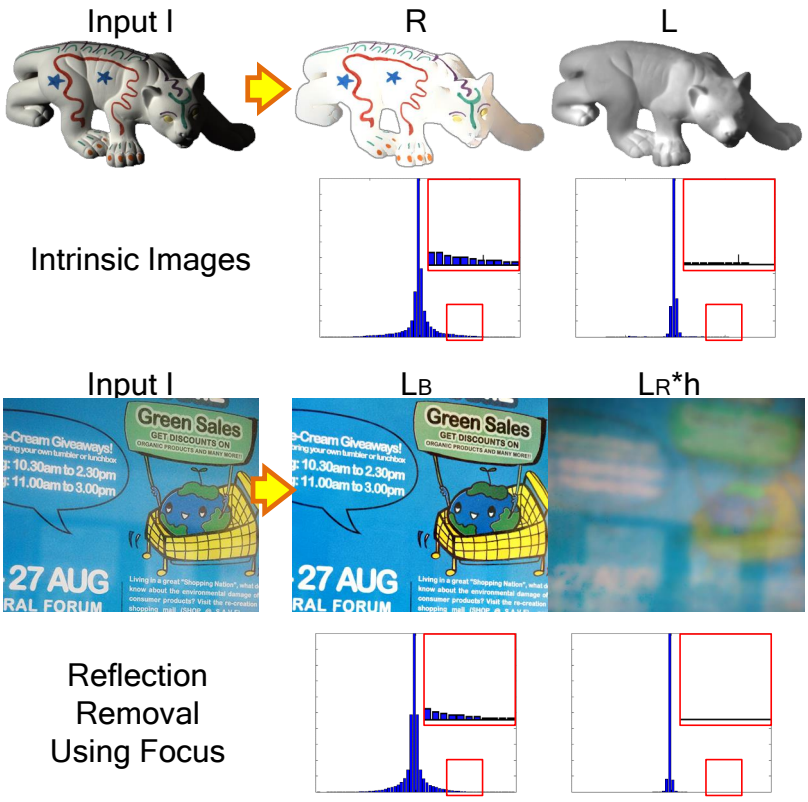


Figure 4.1: This figure shows the two problems our method is applied to: intrinsic image decomposition and single image reflection removal using focus. The corresponding gradient histograms of each layer is shown below. In both of these problems one layer has fewer large gradients than the other layer.

4.1 Introduction

The work in this chapter addresses the problem of layer separation from a single image with application to 1) intrinsic image decomposition and 2) single image reflection interference removal using focus. For intrinsic images, we take the log form that $\log(I) = \log(R) + \log(L)$. As mentioned in chapter 3, images with window reflection can be directly expressed as a linear combination of a reflection layer L_R and and the background scene L_B. In this chapter, we use a slightly modified version based on Schechner *et al.*'s [Schechner et al. 2000] proposition of using focus

such that the desired layer is more in focus while the reflection is blurred. This can be expressed as: $I = L_B + L_R * h$, where the reflection layer is convolved with the depth of field kernel h modelled as a Gaussian blur. Therefore, both of these problems take the form as in equation (1.1).

The work in this chapter is inspired by the success of imposing a gradient sparsity constraint on the images, however, in our problem, the two layers' gradients do not have the same distributions. Prevailing methods for intrinsic image decomposition (see [Gehler et al. 2011; Grosse et al. 2009; Shen and Yeo 2011; Tappen et al. 2005]) typically adapt the idea that natural images have piecewise constant reflectance while the illumination is smoothly varying. This means that the illumination layer L is smoother than the reflectance R. Therefore one of the layers is assumed to be smooth, *i.e.* illumination L and the defocused reflection $L_R * h$, and therefore should have very little large gradient. Figure 4.1 shows an example. This means we need an additional constraint on the smooth layer. Based on this intuition, we propose a novel method to solve the layer separation problem by building two likelihoods for each layer from the gradient histograms in which one layer is smoother than the other. To get the desired layer separation, the necessary objective function is formulated. An efficient scheme is described to optimize the objective function which is non-convex and has an inequality constraint. Our method provides high-quality results on megapixel images in a matter of seconds. This is much faster than existing intrinsic image and reflection separation methods.

As we have reviewed in chapter 2, there are mainly two categories of methods for single image intrinsic image estimation. The first one is the local approach which tries to separate reflectance and illumination edges before recovering the two layers. The other category of intrinsic image estimation is the global approach

which usually forces a global constraint that the reflectance values present in the scene are sparse. These global approaches achieve excellent results on the MIT dataset. Since our method does not make use of a prior on the reflectance values directly, we categorize our method as local one. We show, however, that our method can achieve much better performance than other local edge-based methods. Our results are close (sometimes even better in a few cases) to those obtained by the state-of-art global approaches that use sophisticated models and inference while being significantly faster.

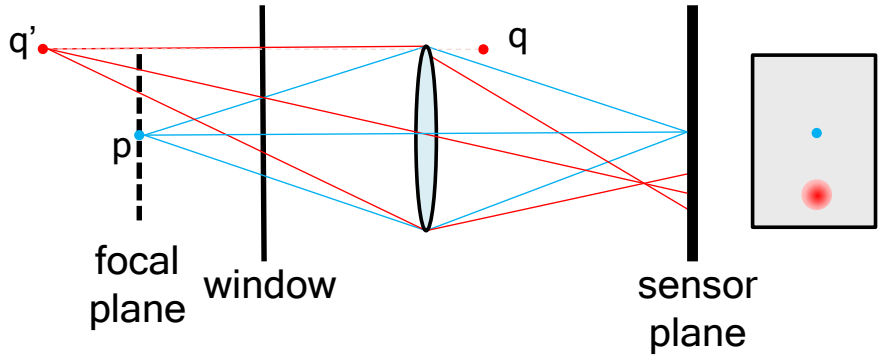


Figure 4.2: The optical geometry of camera defocus in reflection separation.

A complete review on the second problem – reflection removal was also given in chapter 2. The most close method to this work is the reflection removal using focus blur by Schechner *et al*. [Schechner et al. 2000]. However their method still requires two input images, specifically one where the reflection was in focus and one where the background was in focus. We show that using our method we can obtain a high-quality separation of reflection with only a single image focused on the background (See figure 4.2 for the configuration). Moreover, we do not need to explicitly estimate the blur point-spread-function h as done in [Schechner et al. 2000].

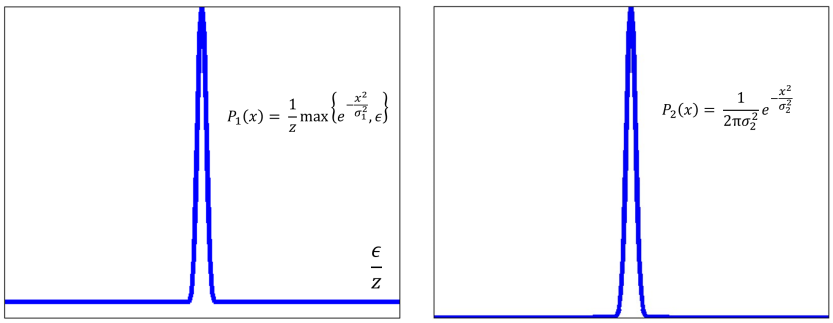


Figure 4.3: This shows the probability distribution of $P_1(x)$ and $P_2(x)$. $P_1(x)$ has the long-tail form whereas $P_2(x)$ is Gaussian with small standard deviation. The probability $P_2(x)$ drop to zero very fast.

4.2 Our Approach

4.2.1 Model

Inspired by the gradient sparsity prior used in [Levin and Weiss 2007], we introduce our priors on the two layers' gradients. Suppose L_2 is smoother than L_1, then large gradients are more likely to belong to L_1. We encode this into two probabilities as:

$$\begin{aligned} P_1(x) &= \frac{1}{z}\max\{e^{-\frac{x^2}{\sigma_1^2}}, \epsilon\}, \\ P_2(x) &= \frac{1}{2\pi\sigma_2^2}e^{-\frac{x^2}{\sigma_2^2}}, \end{aligned} \tag{4.1}$$

where x is the gradient value, z is a normalization factor, σ_1 and σ_2 are both small values making two narrow Gaussians which drop very fast. However by using the *max* operator with ϵ in P_1 we explicitly add a tail to prevent the probability from getting close to zero. $P_1(x)$ and $P_2(x)$ are illustrated in figure 4.3.

In order to solve the layer separation problem, we adapt a probabilistic model to seek the most likely explanation of the input image using the probabilities of the two

layers defined in equation (4.1). In essence, we are maximizing the joint probability $P(L_1, L_2)$. This can be achieved by minimizing the negative *log* probabilities. Taking the negative *log* to the probabilities in equation (4.1), we obtained:

$$\begin{aligned} -\log P_1(x) &\propto \min\{\frac{x^2}{\sigma_1^2(-\log \epsilon)}, 1\} + C_1, \\ -\log P_2(x) &\propto \tfrac{x^2}{\sigma_2^2} + C_2. \end{aligned} \tag{4.2}$$

Here, C_1 and C_2 are constants that we can drop later. While $-\log P_2(x)$ is in L_2 form, $-\log P_1(x)$ is in truncated L_2 form which we further simplify as $\rho(x) = \min\{x^2/k, 1\}$. The term k is still a small number fixed as a constant 10^{-4} in our method. The function ρ is similar to the sparse penalty used in [Xu et al. 2013]. With the assumption that the two layers are independent (*i.e.* $P(L_1, L_2) = P(L_1) \cdot P(L_2)$) and the derivative filter output are independent (*i.e.* $P(L_t) = \prod_i P_t(f_j * L)_i, t \in \{1, 2\}$), minimizing $-\log P(L_1, L_2)$ becomes:

$$\min_{L_1, L_2} \sum_{i,j} \Big(\rho(L_1 * f_j)_i + \lambda (L_2 * f_j)_i^2 \Big), \tag{4.3}$$

where i is the pixel index, f_j denotes different derivative filters. We used two directional first order derivative filters and a second order Laplacian filter, namely $f_1 = [-1\ 1]$, $f_2 = [-1\ 1]^T$, $f_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, and for simplicity we write $F_i^j L = (L * f_j)_i$ in the rest of this chapter. From the experiments we found using first order derivative filters for L_1 and a second order Laplacian filter for L_2 produced good results. The first order derivative filter helps to recover the significant edges in L_1, while the Laplacian filter encodes smooth variations in L_2. We integrate the weight between the two terms and the multiplier $\frac{1}{\sigma_2^2}$ together as one parameter λ which controls the

smoothness of the output L_2. The effect of different λ setting is shown in figure 4.4.

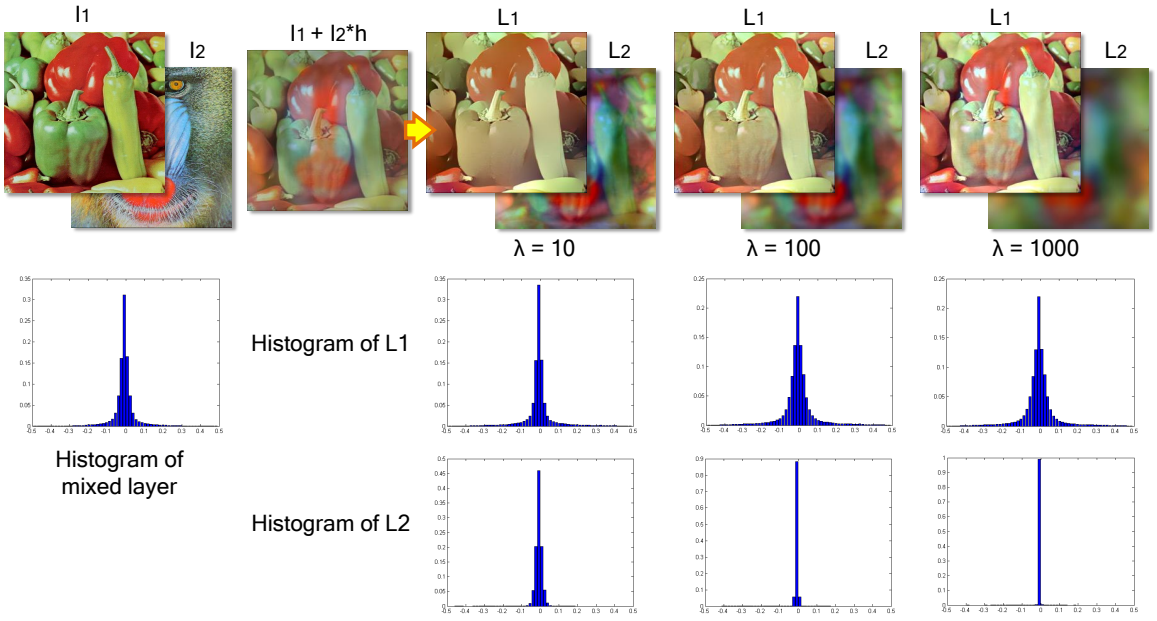


Figure 4.4: This figure shows the effect of different λ setting on the final separation results on a synthesized case. The corresponding gradient histograms are displayed below.

As can be seen, as λ increase, L_2 become more smooth (or more sparse in gradient histogram) and more images details move gradually from L_2 to L_1. Therefore, λ controls the detail transfer in the layer separation. When λ was small ($\lambda = 10$), L_1 lost many details which incorrectly appeared in L_2. When λ was large ($\lambda = 1000$), L_2 became over-smooth and the part of the detail appeared back in L_1. Setting $\lambda = 100$ is an appropriate choice as it gives the most pleasing result.

Our probabilities are defined on the gradients and to recover meaningful layers we have to bound the solution range *i.e.* $(L_1)_i \in [lb_i, ub_i]$. The ranges are set according to the application which will be discussed in section 4.3. Moreover, we can substitute L_2 with $I - L_1$ into the objective, making the final objective function on parameter L_1 as:

$$\begin{aligned} \min_{L_1} \sum_i \left(\rho(F_i L_1) + \lambda (F_i L_1 - F_i^3 I)^2 \right) \\ \text{s.t.} \quad lb_i \leq (L_1)_i \leq ub_i. \end{aligned} \tag{4.4}$$

4.2.2 Optimization

Our objective function is non-convex due to the non-convex $\rho(x)$ component. There is also an inequality constraint. Such problems require care when optimizing. We employ a two stage approach. First, we use the half-quadratic separation scheme [Geman and Yang 1995; Wang et al. 2008] to solve the non-convex problem without the inequality constraint and at the end of each iteration we perform a normalization step to force the solution to fall within the constrained range.

Using the half-quadratic method, auxiliary variables g_i^j are introduced at each pixel that allow us to move the $F_i^j L_1$ term outside the $\rho(\cdot)$ function, giving a new cost function:

$$\min_{L_1, g^j} \sum_i \Big(\sum_{j=1,2} (\beta (F_i^j L_1 - g_i^j)^2 + \rho(g_i^j)) + \lambda (F_i^3 L_1 - F_i^3 I)^2 \Big), \tag{4.5}$$

where β is a weight that we will increase during the optimization (in our implementation , starting from 10 or 20 and multiplied by $\eta = 2$ each time). As β gets larger the solution gets closer to that of equation (4.4). Minimizing equation (4.5) for a fixed β can be performed by alternating between computing L_1 and updating of g^j. The computation of L_1 and g^j-updates are described in the following paragraphs.

Update g^j To solve the problem of updating g^j with fixed L_1, we isolate each

g_i^j and try to find its minimum, which corresponds to:

$$\min_{g_i^j} \beta(F_i^j L_1 - g_i^j)^2 + \rho(g_i^j). \tag{4.6}$$

It is easy to find the two local minimum points of this equation, $F_i^j L_1$ and $\frac{\beta F_i^j L_1}{\beta + \frac{1}{k}}$. If $\beta \ll \frac{1}{k}$, $\frac{\beta F_i^j L_1}{\beta + \frac{1}{k}} \approx 0$. When $g_i^j = F_i^j L_1$, the function value is 1, when $g_i^j = 0$ the function value is $\beta(F_i^j L_1)^2$. Therefore when $1 < \beta(F_i^j L_1)^2$, namely $(F_i^j L_1)^2 > \frac{1}{\beta}$, the global minimum is at $g_i^j = F_i^j L_1$. Otherwise, the global minimum is at $g_i^j = 0$. Therefore the closed-form solution at each pixel is found to minimize equation (4.5) w.r.t. g^j as:

$$g_i^j = \begin{cases} F_i^j L_1, & (F_i^j L_1)^2 > \frac{1}{\beta} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

This simple thresholding rule holds when $\beta \ll \frac{1}{k}$.

Compute L_1 With g^j fixed, the function of equation (4.5) w.r.t. L_1 is as follows:

$$\min_{L_1} \sum_i \Big(\sum_{j=1,2} (\beta(F_i^j L_1 - g_i^j)^2 + \lambda(F_i^3 L_1 - F_i^3 I)^2 \Big). \tag{4.8}$$

This is quadratic w.r.t. L_1 and a global minimum can be obtained directly by taking the derivative over L_1 and setting it equal to zero. Rewriting this in matrix form gives the following equation:

$$\beta \sum_{j=1,2} F^{j^T} F^j L_1 + \lambda F^{3^T} F^3 L_1 = \beta \sum_{j=1,2} F^{j^T} g^j + \lambda F^{3^T} F^3 I. \tag{4.9}$$

Assuming circular boundary conditions, we can apply a 2D FFT $\mathcal{F}$ which diag-

onalizes the convolution matrices F^j's, allowing us to find the optimal L_1 directly:

$$L_1 = \mathcal{F}^{-1}\left(\frac{\beta \sum_j (\mathcal{F}(F^j)^\star \mathcal{F}(g^j)) + \lambda \mathcal{F}(F^3)^\star \mathcal{F}(F^3)\mathcal{F}(I)}{\beta \sum_j (\mathcal{F}(F^j)^\star \mathcal{F}(F^j)) + \lambda \mathcal{F}(F^3)^\star \mathcal{F}(F^3) + \tau}\right) , \tag{4.10}$$

where $\star$ is the complex conjugate, the parameter τ added to the denominator is a small number necessary to increase the stability of our algorithm ($\tau = 10^{-16}$ in our implementation). The multiplication and division are both performed element-wise. Solving equation (4.10) requires only two FFT for g^1 and g^2 and one IFFT at each iteration since the other terms can all be precomputed.

Normalize L_1 After getting L_1, we perform a normalization step to bring the solution to a meaningful range. This step is important since the solution to equation (4.5) is not unique and is related by a global constant. Therefore the goal of the normalization step is to make the solution fall in the range $[lb_i, ub_i]$. To find a suitable constant t we try to minimize the following objective function

$$\min_t \sum_i m_i((L_1)_i + t - lb_i)^2 + \sum_i n_i((L_1)_i + t - ub_i)^2, \tag{4.11}$$

where m_i, n_i are indicative functions such that m_i is equal to 1 only when $(L_1)_i + t < lb_i$, and n_i is equal to 1 only when $(L_1)_i + t > ub_i$, otherwise they all equal to 0. From this, L_1 is updated to $L_1 + t$. Simple gradient descent can be used for this step. After this, a few values may still fall outside the interval $[lb_i, ub_i]$. These values are clipped to lb_i or ub_i. We summarize the whole process in Algorithm 1. In all of our experiments, the optimization converges very quickly (within 5 iterations) and produces high-quality results. Convergence is empirically demonstrated in section 4.3.

Algorithm 1 Layer Separation using Relative Smoothness

Input: input image I; smoothness weight λ; initial β_0; iterations number $i_{\max}$; increasing rate η;

Initialization: $L_1 \leftarrow I; \beta \leftarrow \beta_0; i \leftarrow 0$.

  while $i < i_{\max}$ **do**

    update g_i^j using equation (4.7);

    compute L_1 using equation (4.10);

    normalize L_1 using equation (4.11);

    $\beta = \eta * \beta, i++$;

  end while

  $L_2 = I - L_1$;

Output: The estimation of two layers L_1 and L_2;

4.3 Experimental Results

Our experiments are done on a PC with Intel I7 CPU (3.4GHz) and 8GB RAM. The implementation is done using Matlab without any GPU acceleration.

4.3.1 Intrinsic Image Decomposition

We denote $\log(I)$, $\log(R)$, and $\log(L)$ as $\hat{I}$, $\hat{R}$, $\hat{L}$ respectively in the equations. In our implementation, the original images are normalized to $[1/256, 1]$. Therefore, after *log*, $\hat{R}$ should fall in the range $[\hat{I}, 0]$. Using our method, the objective function becomes:

$$\begin{aligned} \min_{\hat{R}} \sum_i \Big(\sum_{j=1,2} \rho(F_i^j \hat{R}) + \lambda (F_i^3 \hat{R} - F_i^3 \hat{I})^2 \Big) \\ \text{s.t.} \quad \hat{I}_i \le \hat{R}_i \le 0. \end{aligned} \tag{4.12}$$

If we set the smoothness weight λ to zero and just run our whole process once, meaning only threshold the gradient once to get the g^j and then recover $\hat{R}$, our method acts just like the Retinex algorithm [Grosse et al. 2009]. Our implementation of the Retinex algorithm can achieve better performance than the original one

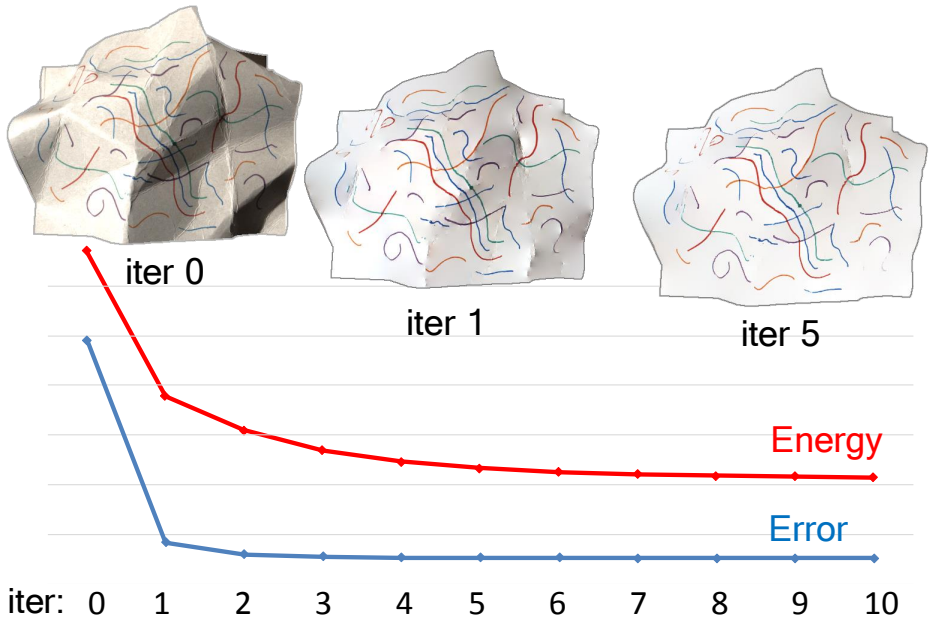


Figure 4.5: This figure illustrate the convergence of our algorithm. The red line and the blue line denote the energy defined by our objective function and the error between current estimation and ground truth using LMSE measurement respectively. Note that the scales of the energy and the error are different. We put them together here for illustration. The estimated reflectance of some of the steps are also plotted above.

described in [Grosse et al. 2009]. Therefore we report the Retinex result using our implementation.

Evaluation on the MIT dataset

We have tested our algorithm on the MIT intrinsic image dataset [Grosse et al. 2009].

Fast convergence We show here that our optimization framework can converge to a good solution very fast (no more than 5 iterations needed). We plot the energy values at each iteration for one intrinsic decomposition example in figure 4.5. At each iteration we measure the error between our layer estimation at the current

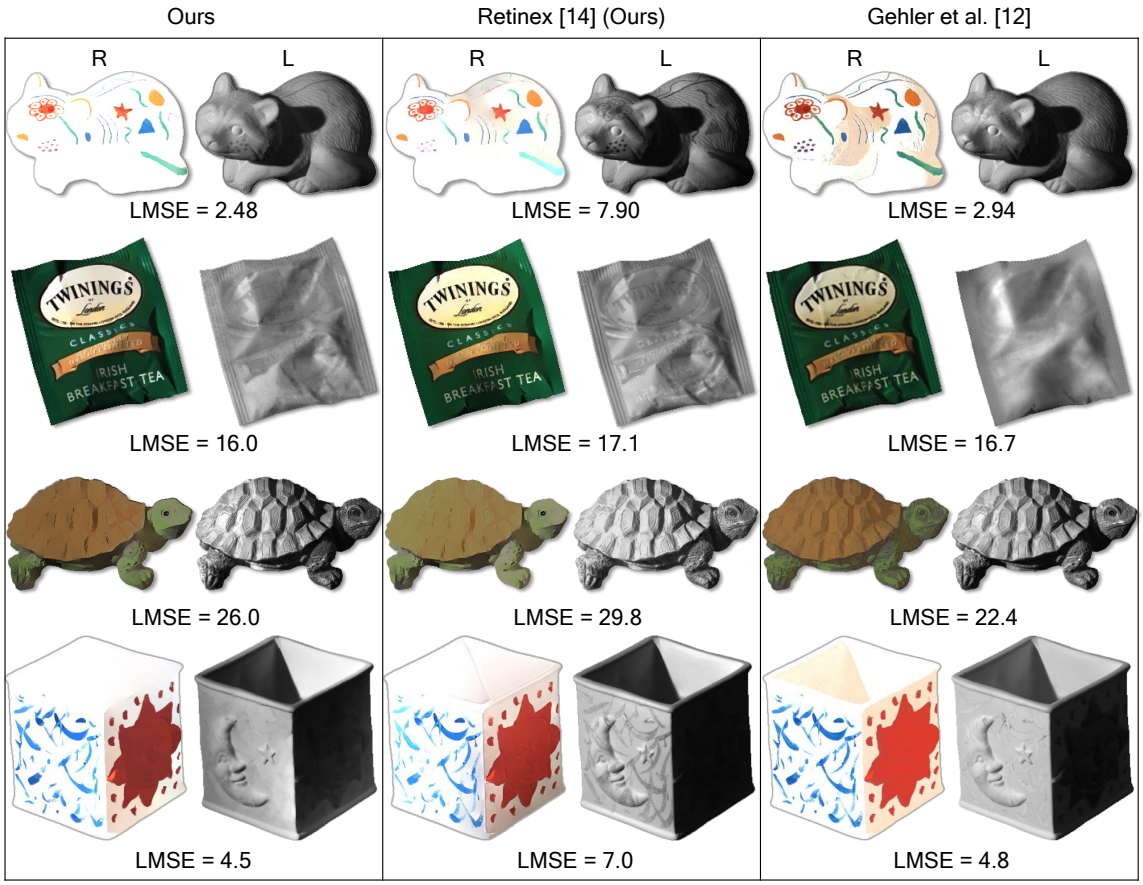


Figure 4.6: This figure shows the decomposition results by Retinex [Grosse et al. 2009], the method in [Gehler et al. 2011] and our approach on three images from the MIT intrinsic dataset. LMSE errors shown below are in 10^{-3}.

state w.r.t. the ground truth data using Local Mean Square Error (LMSE) [Grosse et al. 2009]. The curve is also plotted in figure 4.5 to show that our method can converge to high quality results quickly.

Comparison with previous methods We have compared the performance of our method with several representative intrinsic image estimation methods and reported the running time per image as well as the LMSE on the MIT dataset in table 4.1. Tappen *et al.*'s method [Tappen et al. 2005] is an edge-based method that

learns a classifier to distinguish reflectance edges and illumination edges. Methods in [Barron and Malik 2012; Gehler et al. 2011; Shen and Yeo 2011] use the global sparsity prior and the framework in [Barron and Malik 2012] uses more constraints to solve the shape from shading problem jointly with intrinsic images. These three methods are generally considered as state-of-art in terms of the performance on the MIT dataset. Note that for methods [Gehler et al. 2011] and [Barron and Malik 2012], we cannot get the LMSE as small as in the original paper. We report results provided by the authors that are considered to be their best performance.

Table 4.1: Quantitative Comparison with Previous Methods

Method	**Runtime**	**LMSE**
Tappen *et al*. 2005 [Tappen et al. 2005]	>200 s	0.0347
Shen & Yeo 2011 [Shen and Yeo 2011]	>300 s	0.0204
Gehler *et al*. 2011 [Gehler et al. 2011]	>600 s	0.0131*
Barron & Malik 2012 [Barron and Malik 2012]	>200 s	0.0133*
Retinex [Grosse et al. 2009]	<1 s	0.0217
Ours	1–3s	0.0149

As can be seen, our optimization with Matlab implementation is efficient compared with others due to the FFT acceleration. Even without using the global sparsity prior, our method can achieve high quality performance close to specially designed methods for intrinsic image (*e.g*. [Barron and Malik 2012; Gehler et al. 2011]).

We also show three example results in figure 4.6 and compare with the Retinex method [Grosse et al. 2009] and the best over-all performance method [Gehler et al. 2011]. Our method gives visually better results than Retinex [Grosse et al. 2009] since our results shows clearer edges and no bleeding artifacts. For the *raccoon* and *teabag* cases, our results are even better than [Gehler et al. 2011]. The results of [Gehler et al. 2011] has more regions with incorrect separations of the two layers,

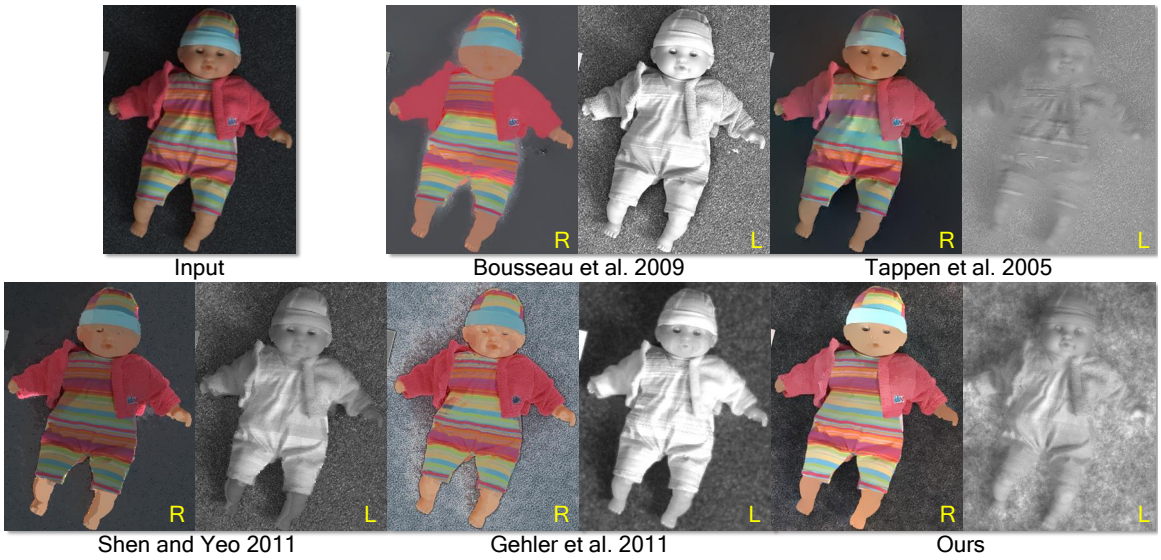


Figure 4.7: Comparison of decomposition results on a photo with the user-assisted approach [Bousseau et al. 2009] and other representative automatic approaches of [Tappen et al. 2005; Shen and Yeo 2011; Gehler et al. 2011]. All illumination images are shown in gray scale.

e.g. illumination components remaining in *raccoon's* reflectance and illumination details near the border of the *teabag* appears in the reflectance image.

Comparison on Real Input

We have also tested our method on the input image used in previous work in [Bousseau et al. 2009]. The method in [Bousseau et al. 2009] is a user-assisted one that can generate more piece-wise constant reflectance with user's labelling of regions sharing same reflectance or same illumination. However, their local 2D subspace model would fail on high contrast region (*e.g.* the border of the doll), resulting in artifacts in the reflectance image. Other three methods [Gehler et al. 2011; Shen and Yeo 2011; Tappen et al. 2005] more or less mixed the texture on the cloth into the illumination map. Our method shows arguably the the best reflectance

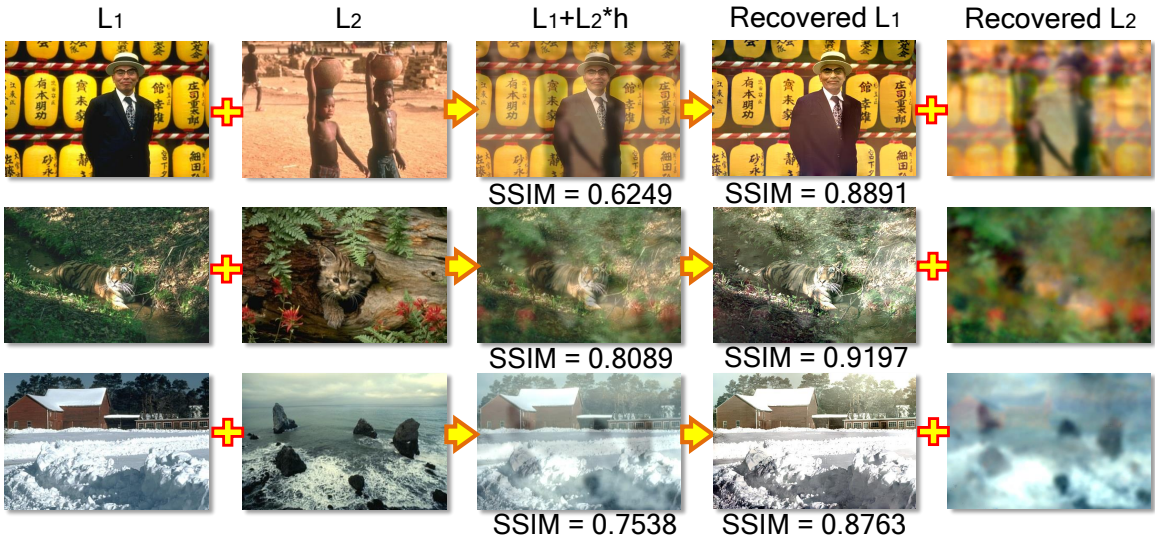


Figure 4.8: Three reflection removal examples on synthesized data. The corresponding SSIM with regard to the ground truth background layer are also listed below for quantitatively showing the effectiveness of our separation. Note that we just write the recovered reflection layer as L_R.

and illumination decomposition results, considering the piece-wise flat reflectance, clear edges and texture information.

4.3.2 Single Image Reflection Removal with Defocus Blur

For the reflection removal problem, the estimated background value $(L_B)_i$ should fall in the range $[0, I_i]$, giving the objective function:

$$\min_{L_B} \sum_i \left(\sum_{j=1,2} \rho(F_i^j L_B) + \lambda (F_i^3 L_B - F_i^3 I)^2 \right) \quad \text{s.t.} \quad 0 \le (L_B)_i \le I_i. \tag{4.13}$$

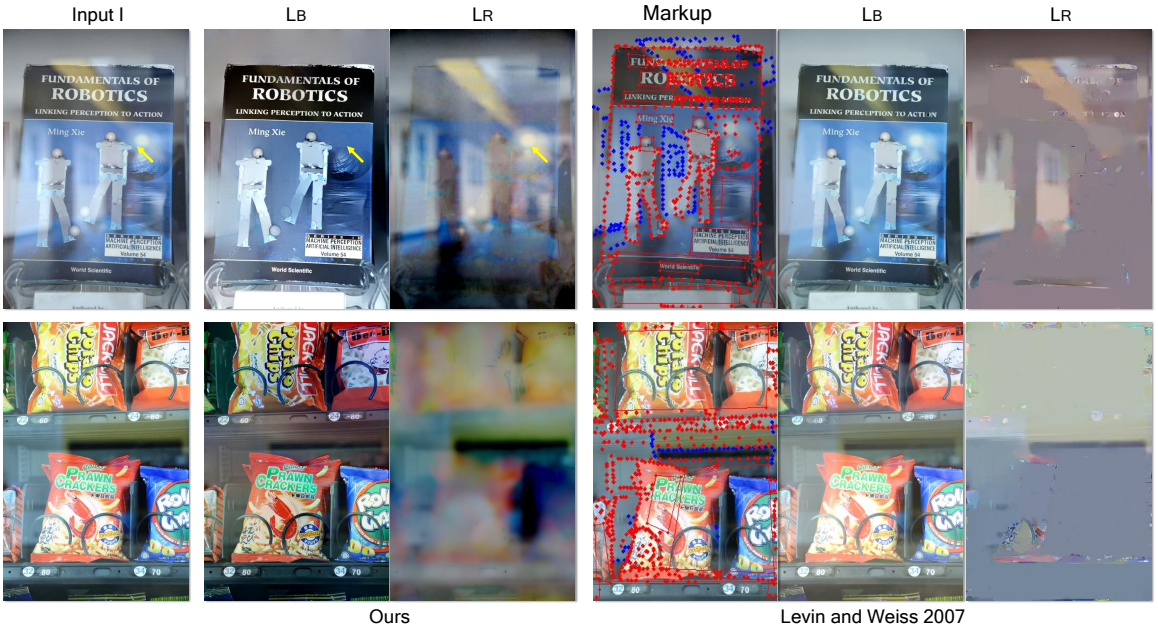


Figure 4.9: Two examples of reflection removal results of our method and prior single image approach in [Levin and Weiss 2007]. Our method provides visually clearer separation results. But in the top case, a small part of the background is smooth (pointed out by yellow arrow) which breaks our assumption, leading to incorrect separation at that small region (How to correct such cases is shown in figure 4.11).

Results on Synthetic Data

Based on the mixing process $I = L_B + L_R * h$, we have synthesised layer mixing data. A 2D Gaussian of standard deviation five is used as the defocus blur kernel h in our synthesis. The input mixing images as well as the final separation results are shown in figure 4.8. To quantitatively assess our algorithm, we have computed the the Structural Similarity Index (SSIM) [Wang et al. 2004] as the quality measure of the recovered background layers.

As can be seen, after separation on the synthesized images using our method, the SSIM is increased by at least 0.1 compared with the original mixed image;

visually, the background layer is much clearer after separation.

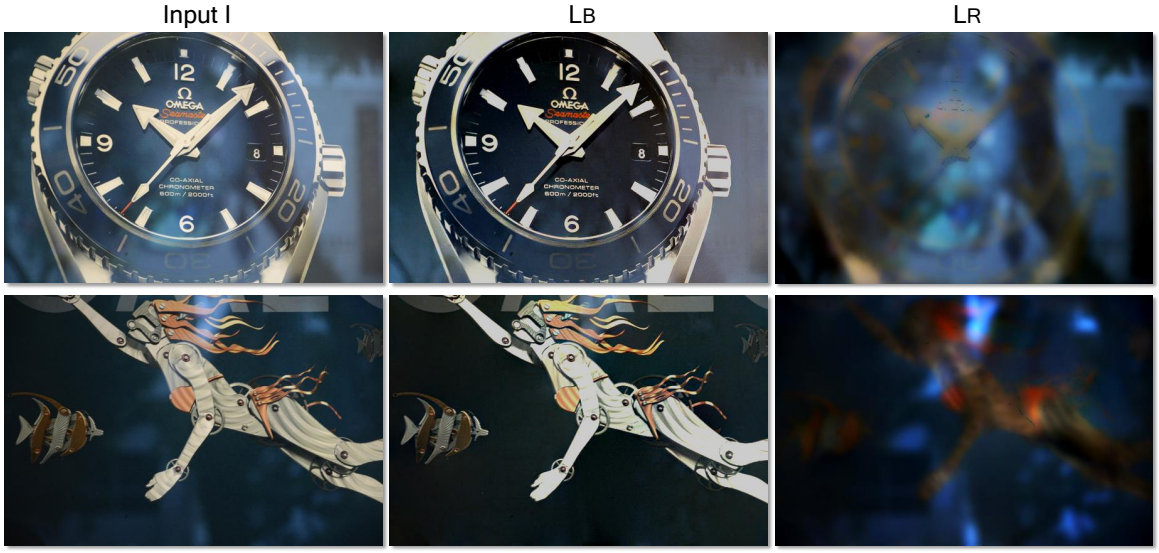


Figure 4.10: This shows two more examples of reflection removal results of our method.

Results on Real World Data

We have tested our method on reflection separation on real world cases and compared ours with the Levin and Weiss's user-assisted method in [Levin and Weiss 2007]. For the results produced by [Levin and Weiss 2007], large amount of user-markup is provided. However, at some locations, the background edges and reflection edges intersect, making it hard for the user to label the gradients, especially because the reflection layer has defocus blur. Our method can generate clearer separation of the background and the reflection layer than that of [Levin and Weiss 2007]. It is worth noting that the method in [Levin and Weiss 2007] is time consuming. Manually providing sufficient labelling can be challenging. In addition, this method solves the non-convex optimization using Iterative Reweighed

Figure 4.11: This is the previous example from figure 7 where part of the image is incorrectly separated. We show here that a simple user interaction (*e.g.* drawing a red rectangle indicating the region belongs to background) can help solve the problem.

Least Square that takes several minutes. Our method is automatic and requires less than two seconds to produce the results. However, the top image in figure 4.9 does reveal a limitation in our work. In particular, the specular highlight (pointed by the yellow arrow) on the ball pattern of the book cover is falsely categorized to reflection layer. This is due to the fact that the highlight is a smooth pattern which violates our assumption that the background layer is sharper than reflection.

4.4 Discussion and Conclusion

We have presented a method to automatically extract two layers from one image where one layer is smoother than the other. Our approach works by building two likelihoods for each layer from gradient histograms, that models this relative

smoothness. In order to solve the layer separation problem, the necessary objective function that finds the most likely explanation of the two layers is proposed. We also derived an efficient scheme to optimize the objective function which is non-convex and has an inequality constraint. We have tested our method on two layer separation problems of intrinsic image decomposition and reflection removal using defocus blur. Our method provides high-quality results in a manner that is significantly faster than prior work.

One challenging issue is that if our assumption that the two layer have different smoothness is violated, our method will fail to correctly separate the layers. An example was shown in section 4.3. If this happens, user intervention may be used to help. For example, we can simply have the user denote which layer a particular region should belong to as shown in figure 4.11.

In the next chapter, we will present our third work for nighttime dehazing that has seen less attention compared to daytime dehazing.

Chapter 5

Nighttime Haze Removal with Glow and Multiple Light Colors

The work presented in this chapter focuses on dehazing nighttime haze images. As we have mentioned in section 2.3, most existing dehazing methods use models that are formulated to describe haze in daylight. Daytime models assume a single uniform light color attributed to a light source not directly visible in the scene. Nighttime scenes, however, commonly include visible light sources with varying colors. Moreover, these light sources often introduce noticeable amounts of glow that is not present in daytime haze. To address these effects, we introduce a new nighttime haze model that accounts for the varying light sources and their glow. Based on the model, we propose a framework to recover the clear scene radiance. We demonstrate the effectiveness of our nighttime dehaze model and correction method on a number of examples and compare our results with existing daytime and nighttime dehazing methods' results.

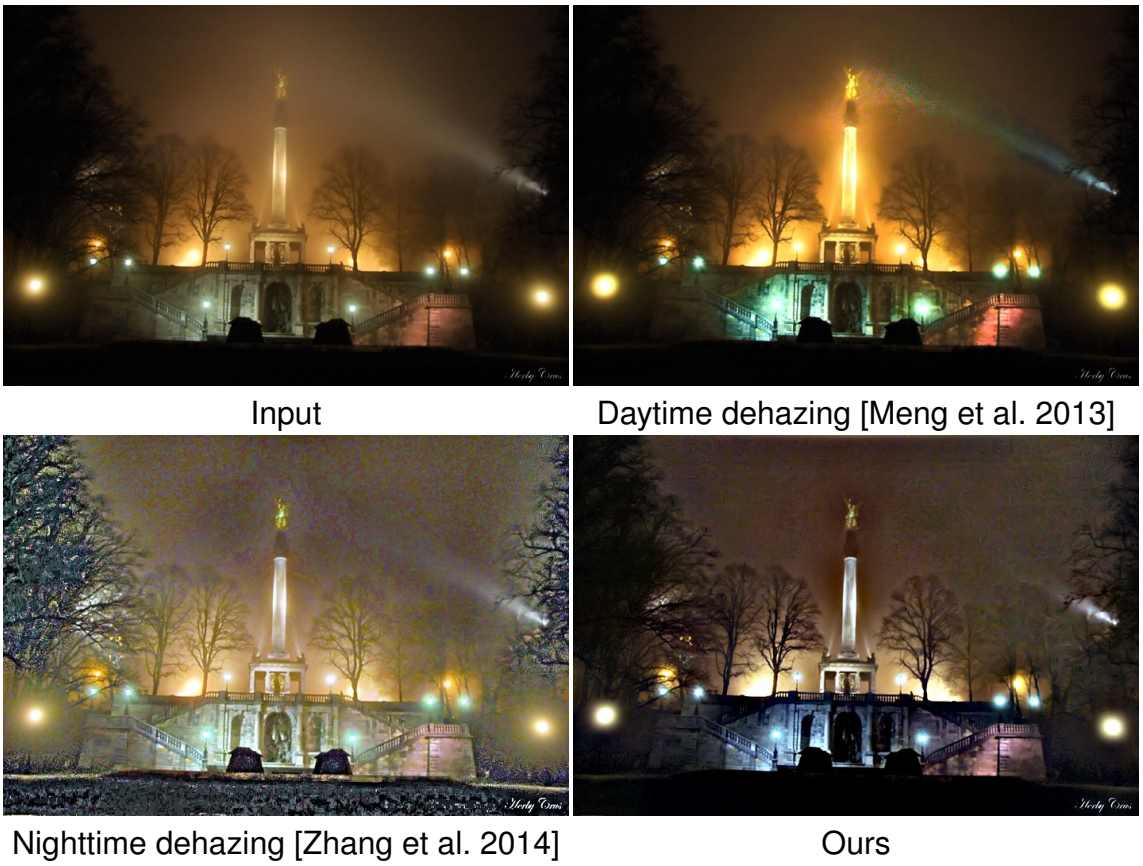


Figure 5.1: This shows a nighttime dehazing case. A daytime dehazing method [Meng et al. 2013] fails to handle glow and haze. A nighttime dehazing method [Zhang et al. 2014] is erroneous in dealing with glow and boosts the intensity unrealistically. Our result shows reduced haze and looks more natural.

5.1 Introduction

As mentioned in section 2.3, there are many methods dedicated to daytime dehazing for single images, such as [Tan 2008; Fattal 2008; He et al. 2011; Tarel and Hautiere 2009; Nishino et al. 2012; Ancuti and Ancuti 2013; Meng et al. 2013; Tang et al. 2014; Fattal 2014]. All methods employ a standard haze model [Koschmieder 1925] which describes a hazy scene as a linear combination of the direct transmission and airlight and assumes that the atmospheric light can be reasonably

approximated from the brightest region in the input image.

While these prior works are effective to handle daytime haze, they are not well equipped to correct nighttime scenes (see Fig 5.1). This is not too surprising, as the standard daytime haze model does not fit well with the conditions of most nighttime hazy scenes. Nighttime scenes generally have active light sources, such as street lights, car lights, building lights, *etc*. These lights add to the scattering-in process, giving more brightness to the existing natural atmospheric light. This implies that the airlight is brighter when the active lights are present in the scene. More importantly, nighttime light sources also introduce a prominent glow to the scene. This glow is a result from both strong lights directly travelling to the camera and light scattered around the light sources by haze particles [Narasimhan and Nayar 2003]. This noticeable glow is not accounted for in the standard haze model.

Furthermore, unlike daytime haze, the atmospheric light cannot be obtained from the brightest region in nighttime images. Due to the presence of active lights and their associated glow, the brightest intensity in the scene can differ significantly from the atmospheric light. Also, because of the multiple light sources, the atmospheric light cannot be assumed to be globally uniform. Consequently, normalizing the input image with the brightest region intensity will cause a noticeable color shift in the image.

There are significantly fewer methods that address nighttime haze. Pei and Lee [Pei and Lee 2012] propose a color transfer technique as a preprocessing step to map the colors of a nighttime haze image onto those of a daytime haze image. Subsequently, a modified dark channel prior method is applied to remove haze. While this approach produces results with improved visibility, the overall color in the final output looks unrealistic. This is due to the color transfer, which changes

colors without using a physically valid model. Zhang *et al.*'s [Zhang et al. 2014] introduce an imaging model for the nighttime haze that includes spatially varying illumination compensation, color correction and dehazing. The overall color of their results looks more realistic than those of [Pei and Lee 2012], however, the model does not account for glow effects, resulting in noticeable glow in the output. The method also involves a number of additional adhoc post processing steps such as gamma curve correction and histogram stretching to enhance the final result (see figure 5.1). In contrast to these methods, we model nighttime haze images by explicitly taking into account the glow of active light sources and their light colors. This new model introduces a unique set of new problems, such as how to decompose the glow from the rest of the image and how to deal with varying atmospheric light. By resolving these problems, we found our results are visually more compelling than both existing daytime and nighttime methods.

5.2 Nighttime Haze Model

For daytime haze scenes, as we have mentioned in section 2.3, the most commonly used optical model assumes that haze image is a linear combination of the direct transmission and airlight as

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x})t(\mathbf{x}) + \mathbf{L}(1 - t(\mathbf{x})).$$

As discussed in section. 2.3, nighttime scenes typically have active light sources that can generate glow when the presence of particles in the atmosphere is substantial. This glow has been analyzed by Narasimhan and Nayar [Narasimhan

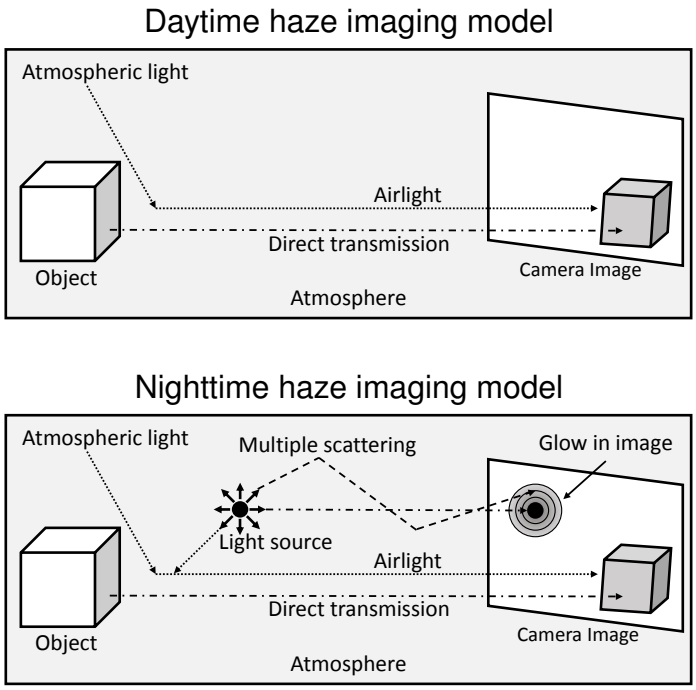


Figure 5.2: This figure compares the standard daytime haze model with our proposed nighttime haze model. The standard daytime haze model assumes that the atmospheric light is globally uniform and contributes to the brightness of the airlight. The model has another term called the direct transmission, which describes light travelling from the the object or scene radiance making its way to the image plane. The bottom shows a diagram of our proposed nighttime haze model. Aside from the airlight and direct transmission, the model also has a glow term, which represents light from sources that gets scattered multiple times and reaches the image plane from different directions. In our model, light sources potentially have different colors that contribute to the appearance of the airlight.

and Nayar 2003] who describe it as light from sources that gets scattered multiple times and reaches the observer from different directions. They model this glow as an atmospheric point spread function (APSF). Inspired by this, we model the entire nighttime hazy scenes by adding the glow model into the slightly modified standard haze model:

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x})t(\mathbf{x}) + \mathbf{L}(\mathbf{x})(1 - t(\mathbf{x})) + \mathbf{L}_a(\mathbf{x}) * APSF. \tag{5.1}$$

The new parameter $\mathbf{L}_a$ is the active light sources, where the light is convolved with the atmosphere point spread function, *APSF*, yielding a glow effect in the image [Narasimhan and Nayar 2003]. Besides this additional glow layer, we retain the other terms scene radiance **R**, atmospheric light **L**, and transmission factor t of the standard haze model. Although, the atmospheric light, **L**, in our model is no longer globally uniform. Our atmospheric light can change for different locations. This is due to the fact that various colors from different light sources can contribute to the atmospheric light as a result of the scattering-in process. While this represents a rather simple modification to the standard haze model, to the best of our knowledge this model is novel and offers a useful means to describe nighttime haze images with glow and active light sources.

For illustration, figure 5.2 shows diagrams of both the daytime haze and nighttime haze models. In the nighttime haze, aside from the natural atmospheric light, the airlight obtains its energy from active light sources, boosting the brightness in the image. The active light sources also creates its own presence in the image by having its direct light to the image and its scattered light that manages to reach the camera after multiple bounces inside the medium. In the image, these manifest themselves as glow, which is separate imagery from other objects in the scene. In the real world, the presence of glow can be significantly prominent in terms of the affected areas and the brightness. Also, due to the scattering, the brightness of the glow effect gradually decreases, making its appearance smooth.

Note that our model is different from the model proposed by Zhang *et al*. [Zhang et al. 2014]. Zhang *et al*.'s model is similar to the standard haze model that employs the two terms, yet adds a new parameter accounting for various light colors and brightness values. This varying light color and brightness is similar to the varying

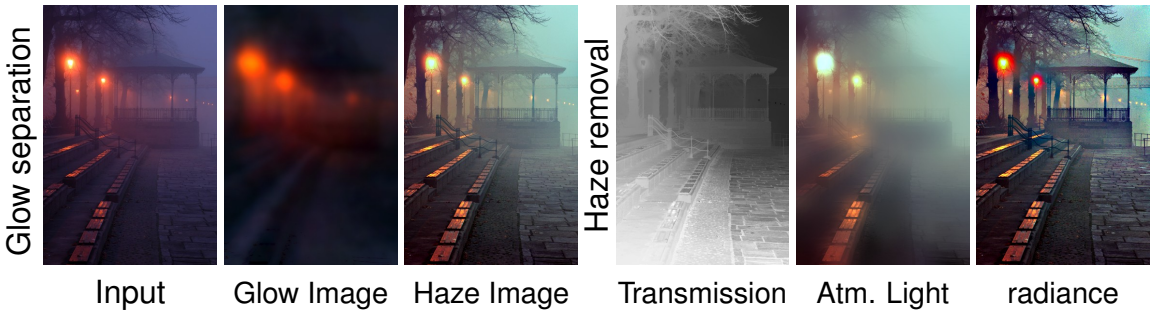


Figure 5.3: Our nighttime dehazing Pipeline: (column 1-3) given an input **I**, we decompose it into a glow image, **G**, and haze image **J**; (column 4-6) we further dehaze the haze image, **J**, yielding the transmission, t, atmospheric light, **L** and scene radiance, **R**.

atmosphere light, $\mathbf{L}(\mathbf{x})$ in our model in equation 5.1. We note that our model is also related to some extent to Schechner and Karpel's model [Schechner and Karpel 2004] for underwater images, which takes image blur into account by convolving the forward scattering with a Gaussian function. However, Schechner and Karpel do not intend to model glow, instead they want to model the scene blur caused by the significant amount of particles in underwater scenes.

5.3 Nighttime Haze Removal

Given an input image **I**, our goal is to estimate the scene radiance, **R**, for every pixel. Figure 5.3 shows the images involved in our pipeline. From the input image, **I**, we decompose the glow image **G** to obtain the nighttime haze image **J**. Having obtained the nighttime haze image that is ideally free from glow, we further dehaze it, and recover the transmission t, the varying atmospheric light **L**, and finally the scene radiance, **R**. Notice that, unlike in daytime dehazing, our atmospheric light is not achromatic due to the various colors of light sources. More details to our

nighttime dehazing process is provided in the following sections.

5.3.1 Glow Decomposition

Narasimhan and Nayar's method [Narasimhan and Nayar 2003] models glow by convolving a light source with the atmospheric point spread function (APSF) represented by a Legendre polynomial and the attenuation factor represented by the Lambert-Beer law. The model is then used to estimate the optical thickness (the distance of a light source to the camera) and the forward scattering parameter of the Henyey-Greenstein phase function, which represents the scattering degrees of different aerosols. Having estimated these two parameters, the deconvolution of the glow can be applied and as a result, the shapes of the light sources can be obtained. Since the optical thickness is known, the depth of the scene nearby the light sources can also be recovered. Although Narasimhan and Nayar's method can be used to estimate the glow's APSF parameters, it was neither meant to enhance visibility nor decompose glow from the input image. Moreover, it assumes that the locations and the areas of individual light sources are known, which is in general problematic to obtain automatically.

To resolve this, we take a different approach. We notice that appearance of glow can be dominant in nighttime haze scenes hindering the visibility of the scene behind. In some areas, the brightness of the glow can be so dominant that the nearby objects to the light sources cannot be seen at all. Consequently, to enhance visibility, we need first to remove the effects from glow. Our approach is to decompose this from the rest of the scene. To enable this decomposition process,

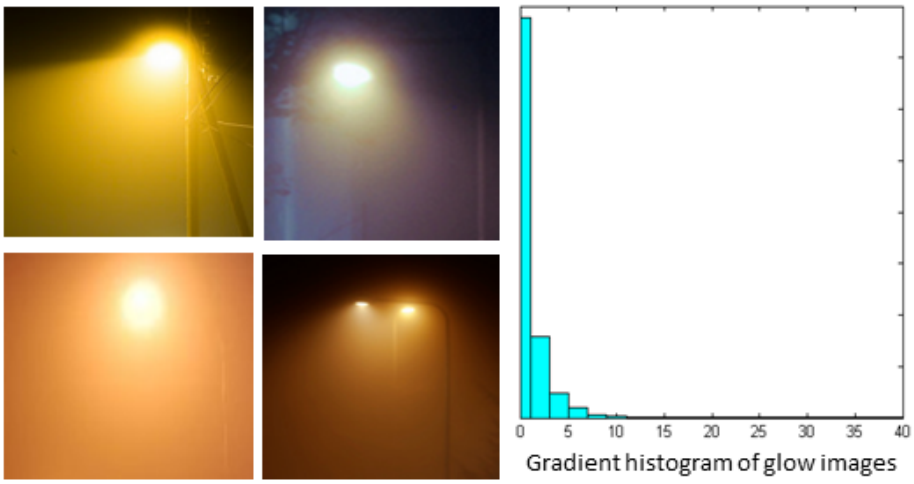


Figure 5.4: Some glow patches and their gradient histogram profile. Even though the color, shape, direction of the glow are different, the images gradient histogram are well modeled using a short tail distribution.

we first rewrite our model in equation (5.1) as:

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

where $\mathbf{J} = \mathbf{R}(\mathbf{x})t(\mathbf{x}) + \mathbf{L}(\mathbf{x})(1 - t(\mathbf{x}))$ and $\mathbf{G}(\mathbf{x}) = \mathbf{L}_a(\mathbf{x}) * APSF$. We call the former the nighttime haze image, and the latter the glow image. In this form, decoupling glow becomes a layer separation problem, with the two layers: $\mathbf{J}$ and $\mathbf{G}$, that need to be estimated from a single input image, $\mathbf{I}$.

As discussed in section 5.2, due to the multiple scattering surrounding light sources, the brightness of the glow decreases gradually and smoothly away from the light sources. We exploit this smoothness attribute of the glow image and employ our method in chapter 4, which targets layer separation for scenes where one layer is significantly smoother than the other. In particular, it exploited the distribution of the two layers noting that smooth layers have a "short tail" distri-

bution of the layer's gradient histogram. As shown in figure 5.4, the glow effect of nighttime haze also shares this characteristic of having very few large gradients, meaning we can also model it with a short tail distribution.

Following our work in chapter 4, we design our objective function for layer separation as:

$$\begin{aligned} E(\mathbf{J}) = \sum_{\mathbf{x}} \left(\rho(\mathbf{J}(\mathbf{x}) * f_{1,2}) + \lambda((\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})) * f_3)^2 \right) \\ \text{s.t.} \quad 0 \leq \mathbf{J}(\mathbf{x}) \leq \mathbf{I}(\mathbf{x}), \\ \textstyle\sum_{\mathbf{x}} \mathbf{J}_r(\mathbf{x}) = \sum_{\mathbf{x}} \mathbf{J}_g(\mathbf{x}) = \sum_{\mathbf{x}} \mathbf{J}_b(\mathbf{x}). \end{aligned} \tag{5.3}$$

where $f_{1,2}$ f_3 is the same filters used in chapter 4. In this case, the second term uses the L_2 norm regularization for the gradients of the glow layer, $\mathbf{G}$, where $\mathbf{G}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})$, will force a smooth output of the glow layer. As for the first term, the truncated quadratic function ρ will preserve the large gradients of $\mathbf{I}$ in the remaining layer $\mathbf{J}$. The parameter λ controls the smoothness of the glow layer.

Since the regularization is all in gradient values, we do not have the information for 0-*th* order offset information of the layer colors. To solve this problem, our work in chapter 4 proposes to add one inequality constraint to ensure the solution is in a proper range. However, since this constraint is applied to each color channel (*i.e.* r, g, b) independently, it may still lead to color shift problem for the nighttime images are usually dark and very unstable for layer separation. An example of such a case is shown in figure 5.5.

To address this problem, we add a second constraint: $\sum_{\mathbf{x}} \mathbf{J}_r(\mathbf{x}) = \sum_{\mathbf{x}} \mathbf{J}_g(\mathbf{x}) = \sum_{\mathbf{x}} \mathbf{J}_b(\mathbf{x})$. With the two constraints combined together, we can obtain a glow separation result with less overall color shift. The second constraint forces the range of

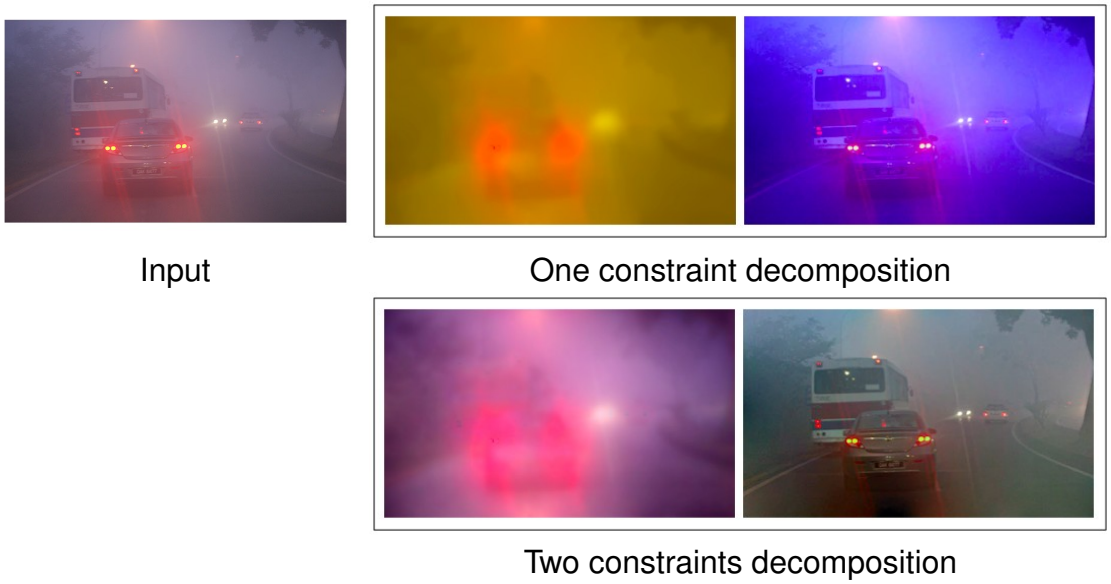


Figure 5.5: Effect of our first and second constraints for the glow decomposition. From the input image **I**, we decompose the glow by using solely the first constraint, resulting in the color shift in the estimated glow image and the estimated haze image. Based on the same input, we add the second constraint, and now the estimated glow image and haze image are more balanced in terms of their colors.

the intensity values for difference color channels to be balanced. This effectiveness of this additional constraint is shown in figure 5.5. The objective function in equation (5.3) can be solved efficiently using the half-quadratic splitting technique as shown in chapter 4.

5.3.2 Haze Removal

Having decomposed the glow image, **G**, from the nighttime haze image **J**, we still need to estimate the scene radiance, **R**. Presumably, since the glow has been significantly reduced from the image **J**, we should be able to enhance the visibility by using any existing daytime dehazing method. However, as previously mentioned, daytime dehazing algorithms assume the atmospheric light is globally uniform,

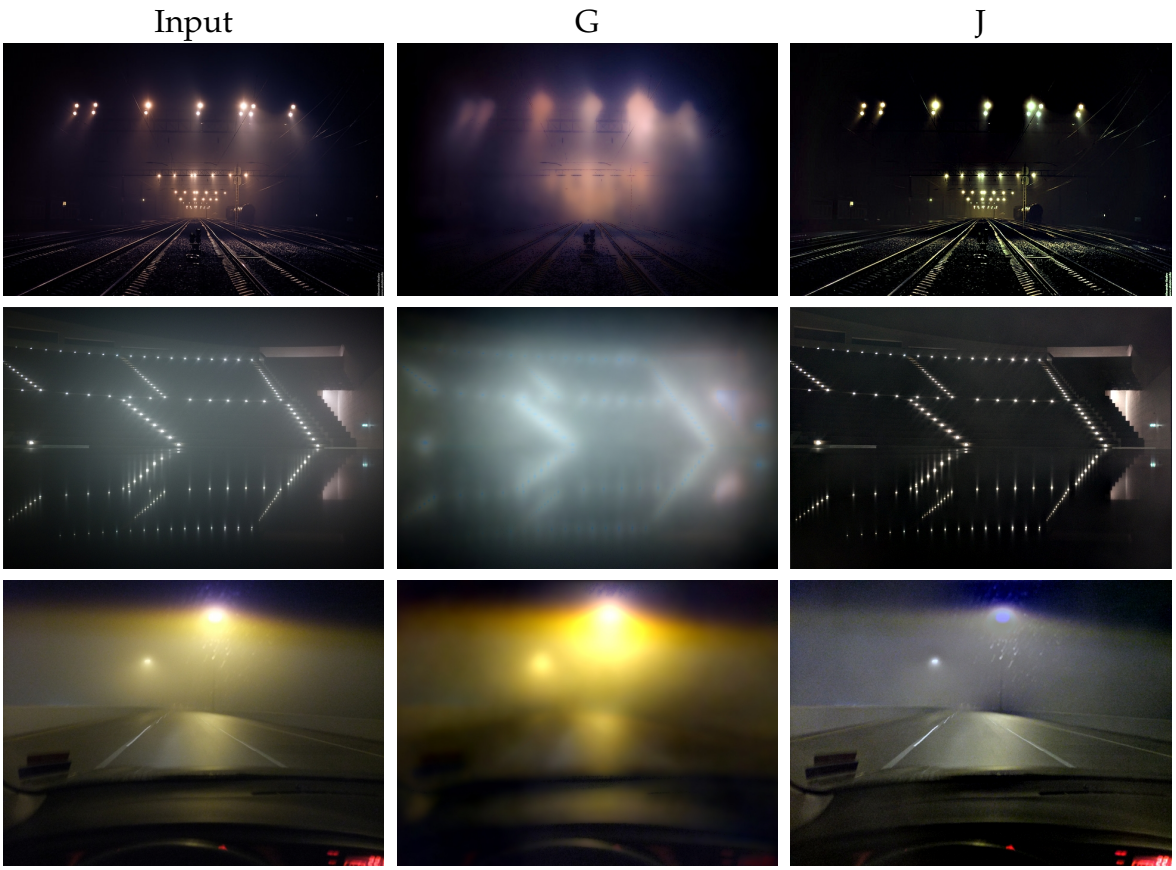


Figure 5.6: Glow decomposition results. (Left column) shows the input images. (Middle column) shows the estimated glow images. (Right column) shows the estimated haze images. As one can notice, the presence of glow in the haze images is much reduced.

which is not valid for nighttime scenes due to the presence of active lights.

To address this issue, we assume that atmospheric light is locally constant and the brightest intensity in a local area is the atmospheric light of that area. This brightest intensity assumption is similar to that used in color constancy that assumes the color represents the illumination [Joze et al. 2012]. To implement this idea, we split the image **J** into a grid of small square areas and find the brightest pixel in each area. We then apply a content-aware smoothness technique, such as the guided image filter [He et al. 2010] on the grid to obtain our varying atmospheric

light map.

Using the atmospheric light map, we estimate the transmission. If we employ the dark channel prior [He et al. 2011], the estimation is done by:

$$t(\mathbf{x}) = 1 - \min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c}\frac{J^c(\mathbf{y})}{L^c(\mathbf{y})}\right), \tag{5.4}$$

where Ω is a small patch, and $\mathbf{y}$ is the location index inside the patch. Unlike the original dark channel prior, the atmospheric light spatially varies.

Figure 5.3 shows the examples of our estimation on the atmospheric light, $\mathbf{L}$, the transmission, t, and the scene radiance, $\mathbf{R}$. As one can see in the figure, the estimated scene radiance shows better visibility than the original input image.

5.4 Experimental Results

We have gathered hazy and foggy nighttime images from the Internet, with various quality and file formats. Based on these images, we evaluated our method and compared the results with those of daytime dehazing methods of [Meng et al. 2013], [He et al. 2011] and nighttime method [Zhang et al. 2014].

We have two comparison scenarios. First, given an input of hazy nighttime image, we process it directly using our method, two daytime dehazing methods of [Meng et al. 2013], [He et al. 2011] and a nighttime method [Zhang et al. 2014]. Second, given an input of a hazy nighttime image, we decompose the glow from the haze image, and further process the haze image with varying atmospheric light using our method and using the method of [Meng et al. 2013]. The main purpose of the first scenario is to show the importance of the glow-haze decomposition, and

the main purpose of the second scenario is to show the importance of addressing the varying atmospheric light. Note that after decomposing the glow and estimating the varying light, our method uses the dark channel prior to obtain the final scene radiance; however, as other dehazing methods could also be used.

Figure 5.10, 5.11 shows results for scenario 1. As can be observed, for nighttime scenes with the presence of glow, the daytime dehazing methods [Meng et al. 2013] [He et al. 2011] tend to fail (the first and second rows of the figure). As for the nighttime dehazing method [Zhang et al. 2014] (the third row), the glow is not handled properly, and due to the additional adhoc post processing, the intensity and colors of some areas are visible exaggerated. Our results are shown in the fourth row in the figure, which we consider to look better in terms of visibility and with more natural colors.

Input **J** [Meng et al. 2013] Ours

Figure 5.7: The left column shows the haze images, **J**, after decomposing it from the glow images. The middle column shows the dehazing results using an existing daytime dehazing method [Meng et al. 2013]. The colors are noticeably shifted due to the varying atmospheric light. Right column shows our results, where the color shift is less significant since varying atmospheric light is used.

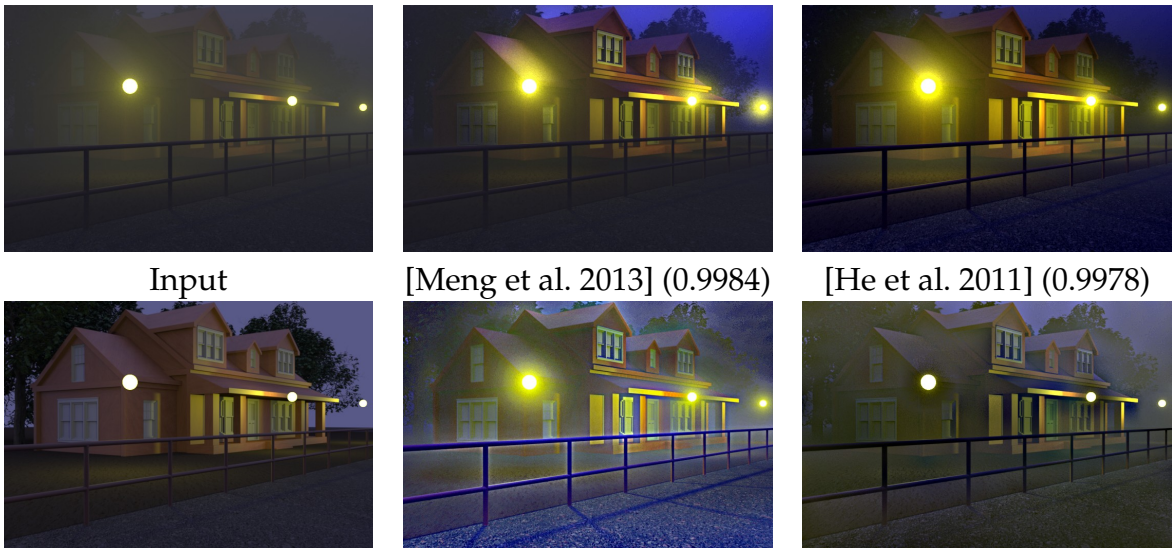


Figure 5.8: Quantitative evaluation using SSIM [Wang et al. 2004] on a synthetic image. The numbers indicate the SSIM index to the groundtruth. Our result has the largest SSIM index, implying that it is more similar to the groundtruth than the other. The synthetic data is generated using PBRT [Pharr and Humphreys 2010].

Figure 5.7 shows two results for scenario 2. Having decomposed the glow, the haze image was processed using [Meng et al. 2013], a daytime dehazing method. In comparison to our results, for less varying colors of the atmospheric light, they are similar to our results in terms of the dehazing quality. However, when the varying colors of the atmospheric light are significantly visible, the color shift problem becomes more apparent. In the middle column, Meng et al.'s method [Meng et al. 2013] shows visible color shift. The blue sky in the first row becomes reddish, and the white wall in the second row becomes bluish. Our results, shown in the right column, retain the colors of the scenes.

We also quantitatively evaluated our result by measuring its structural similarity index (SSIM index [Wang et al. 2004]) to a ground truth image. For this, we used a synthetic image generated using PBRT [Pharr and Humphreys 2010], since it is very

Input [Zhang et al. 2014] Our Result [Meng et al. 2013]

Figure 5.9: Evaluation on a nighttime image with globally uniform atmospheric light. These results show that our method's result is similar to that of Meng et al.'s [Meng et al. 2013], which is a daytime dehazing method.

difficult to obtain groundtruth image for real nighttime scenes. Figure 5.8 shows our result and the SSIM indexs against the other methods' results. Our SSIM value is larger than that of the other methods, implying that our result is more similar to the groundtruth.

Figure 5.9 shows an example of applying our method to a nighttime image with no active light sources (no glow), where we can assume a globally uniform atmospheric light. The result shows that our method behaves like an existing daytime dehazing method [Meng et al. 2013], while [Zhang et al. 2014] over-boosts the contrast such that in the bottom area of the image (red rectangle), the green channel gets boosted more than the other channels.

5.5 Discussion and Conclusion

The work in this chapter has focused on nighttime haze removal in the presence of glow and multiple scene light sources. To deal with these problems, we have introduced a new haze model that incorporates the presence of glow and allows for spatially varying atmospheric light. While our model represents a straight-forward

departure from the standard daylight haze model, we have shown its effectiveness for use in nighttime dehazing.

In particular, we detailed a framework to first decompose the glow image from the nighttime haze image, by assuming that the brightness of the glow changes smoothly across the input image. Having obtained the nighttime haze image a spatially varying atmospheric light map was introduced to deal with the problem of multiple light colors. Using the normalized nighttime haze image, we estimated the transmission and finally the scene radiance. Our approach was compared with a number of examples against several competing methods and was shown to produce favourable results.

There are a few remaining problems, however, that need further attention. First, our estimation on varying atmospheric light is admittedly an approximation. In the method we assume it is locally constant and obtained from the brightest intensity in each of the local area. Although the brightest intensity is used in color constancy [Joze et al. 2012], optically it is not always true, since the intensity value is dependent on other various parameters, such as reflectance and particle properties. This is a challenging problem, even in the color constancy community and requires additional work.

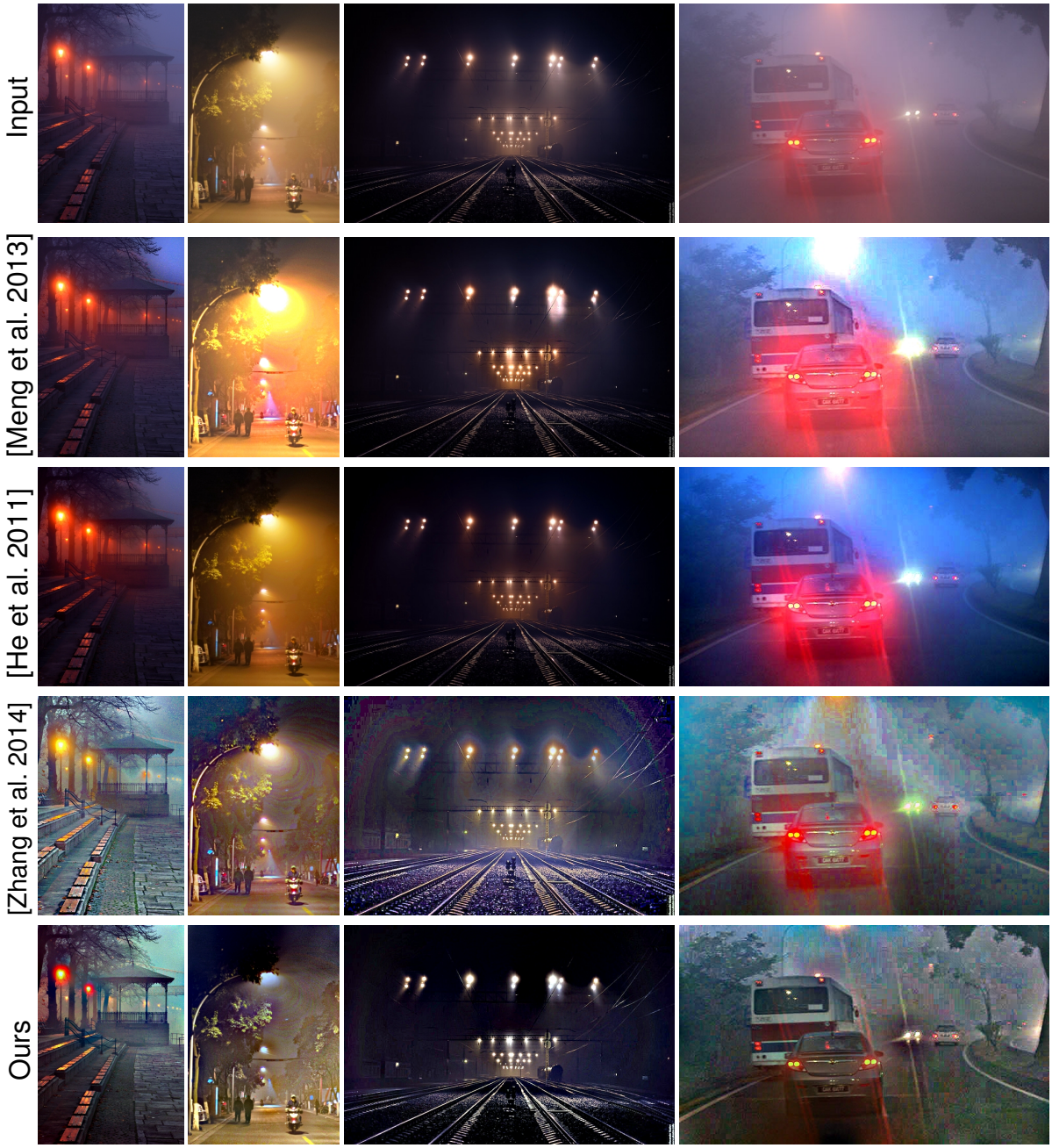


Figure 5.10: The qualitative comparisons of Meng et al.'s method [Meng et al. 2013], He et al.'s method [He et al. 2011], Zhang et al.'s method [Zhang et al. 2014], and ours using various nighttime images.

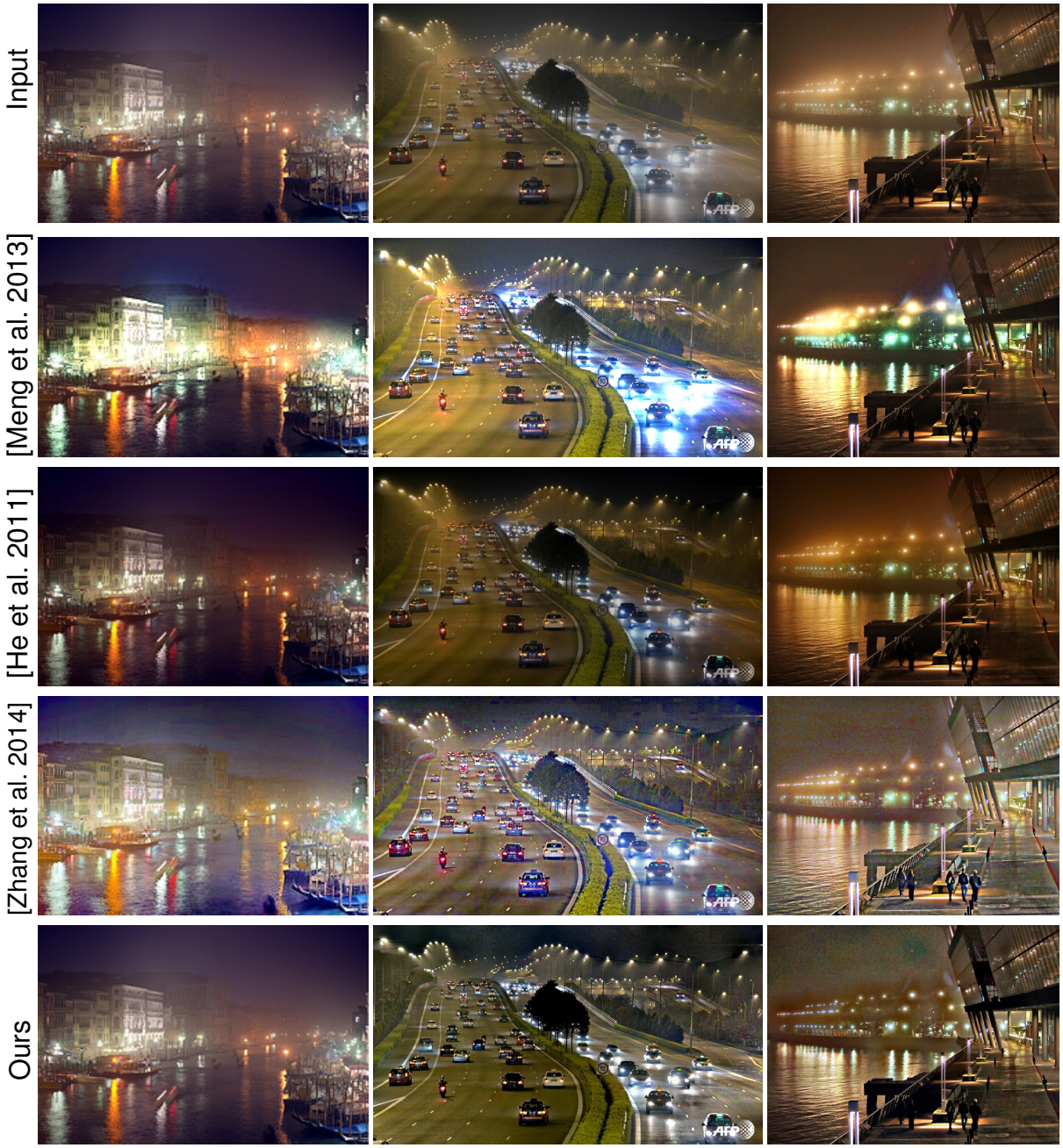


Figure 5.11: More qualitative comparisons of Meng et al.'s method [Meng et al. 2013], He et al.'s method [He et al. 2011], Zhang et al.'s method [Zhang et al. 2014], and ours using various nighttime images.

Chapter 6

A Contrast Enhancement Framework with JPEG Artifacts Suppression

In the previous three chapters, we described three works for separating layers in images. In this chapter, we present an application for another type of layer separation, namely the structure-texture decomposition. We use it in solving the problem of compression artifacts arised in contrast enhancement. Experimental comparisons shows the effectiveness of our method.

6.1 Introduction

A commonly applied procedure in low-level computer vision is contrast enhancement. This encompasses techniques that boost an image's global contrast through manipulations such as tone-curve adjustment, histogram equalization, and gradient-based enhancement. Such enhancement is beneficial for color segmentation, edge detection, image sharpening, image visualization, and many other tasks. In ad-

dition, spatially varying contrast enhancement is used to dramatically improve visibility in turbid media, such as haze, fog, rain, and underwater imaging.

Virtually all contrast enhancement algorithms operated on the assumption that the input image is uncompressed and free from significant noise. The reality, however, is that the vast majority of images available today on the internet or from commodity imaging devices are compressed. Moreover, images coming from sources that would require contrast enhancement, *e.g*. surveillance cameras, often have notable amounts of image compression [Jacobs et al. 2009]. The most common compression scheme is by far JPEG and its extension to video, MPEG. The JPEG compression scheme breaks an input image into 8×8 pixel blocks and applies a discrete cosine transformation (DCT) to each block individually. To reduce storage space, the DCT coefficients are quantized at various levels – more quantization gives higher compression but lowers image quality (for more details see [Watson 1993]). Lower-quality images exhibit what is termed collectively as "compression artifacts" that consist of the characteristic blocking artifacts resulting in discontinuities at the 8×8 borders, and oscillations or ringing artifacts next to strong edges.

Early JPEG compression methods use fixed quantization tables for different quality settings, however, most JPEG schemes now use what is referred to as *optimized JPEG* where quantization tables are customized based on the image's content [Wang and Lee 2001]. This allows relatively high compression rates with little noticeable visual artifacts. However, when contrast boosting operations are applied, blocking and ringing artifacts become prominently visible as shown in figure 6.1. [1]

[1]JPEG assigns a quality factor, *Q*X, to indicate the subjective quality from 0 to 100 (from low quality to high quality)

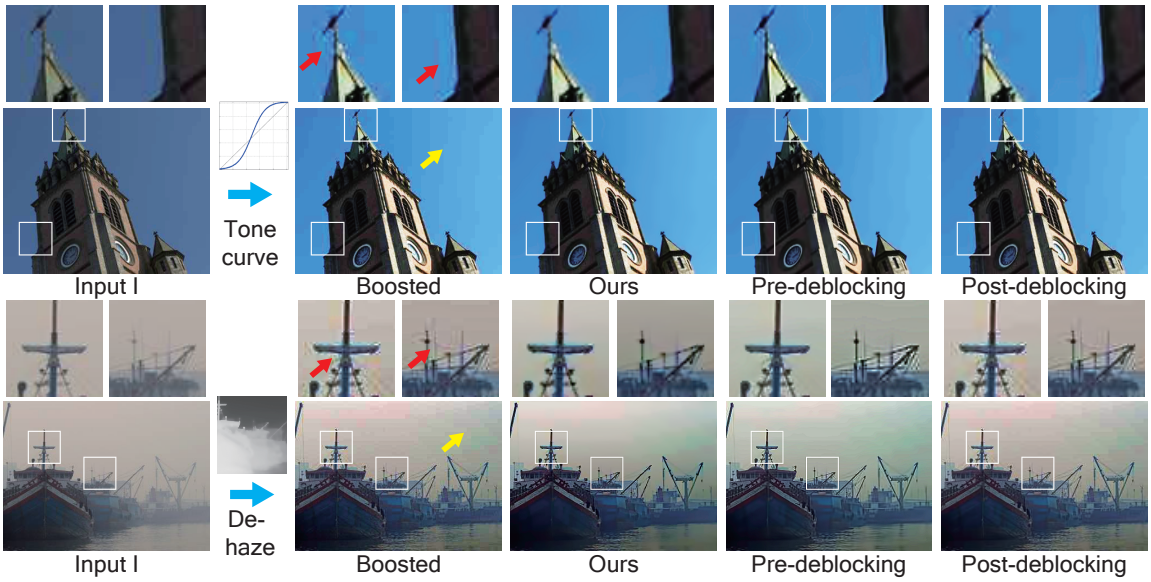


Figure 6.1: This shows the noticeable compression artifacts after contrast enhancement. Top two rows are a tone-curve adjustment case ($Q40$) and the bottom two rows are a dehazing case ($Q70$). The zoomed-in regions are listed above to show the details. The characteristics of the blocking artifacts are distinctive in smooth regions (pointed out by the yellow arrows), while the ringing artifacts are along strong edges (pointed out by the red arrows). Comparison of our results with those of the deblocking method [Foi et al. 2007] applied before or after contrast enhancement results are shown. Note, our method produces more compelling results for reducing both blocking and ringing artifacts.

There are several existing methods to reduce JPEG compression artifacts. These methods are often referred to as "deblocking" or "deringing". In the context of contrast enhancement, these methods would be applied either before or after the enhancement process. When applied before the enhancement process, the algorithms can smooth image details that have small contrast. When applied as a post-processing step, the effectiveness can be diminished due to the compression artifacts that were boosted by the contrast enhancement process. Figure 6.1 shows an example.

In this chapter, we propose a framework based on structure-texture decompo-

sition to remove the compression artifacts that are amplified in the image contrast enhancement operation. After the decomposition, contrast enhancement is directly applied to the structure layer, which is devoid of compression artifacts. Meanwhile, the texture layer, containing both image details and compress artifacts, is carefully processed to suppress only the artifacts. After proper scaling, the cleaned texture component is added back to the enhanced structure layer to generate the artifacts free output. Experimental results on various contrast enhancement task (*e.g.* figure 6.1) demonstrate that our strategy can produce more compelling results (both qualitatively and quantitatively) than those of using general deblocking algorithms in either a pre- or post-processing manner. The details of our algorithm as well as comparisons with other methods are discussed in the following sections.

6.2 Related Work

We discuss relevant related work in the area of JPEG artifacts removal, contrast enhancement and multi-band image decomposition.

JPEG Artifacts Removal. JPEG artifacts, particularly blocking artifacts, have long been recognized in the image processing community (*e.g.* [Lee et al. 1998; Zakhor 1992]). Despite this, they remain unsolved and it is still an active area of research (*e.g.* [Dong et al. 2011; Yim and Bovik 2011]). Various methods have been used, which can be broadly categorized into three different approaches. The first approach treats the compression artifacts as non-Gaussian noise and attempts to remove them by adaptive local filtering which adjusts the filter kernel to remove block edges and preserve image edges (*e.g.* [Foi et al. 2007]). The second approach

is a reconstruction based approach that incorporates knowledge on natural images and encodes it into an energy function as a prior. Commonly used priors include spatial smoothness [Yang et al. 1995], quantization constraints, total variation (*e.g.* [Goto et al. 2011]), and gradient constraints (*e.g.* Field of Experts [Sun and Cham 2007]). The third approach for reducing compression artifacts relies on machine learning techniques to learn a mapping from compressed images to their uncompressed version [Lee et al. 2005; Burger et al. 2012]. While these approaches can reduce JPEG artifacts in images, their application as either a pre- or post-processing step can rarely outperform our method, which is designed explicitly for contrast enhancement.

Image Contrast Enhancement. Contrast enhancement can be performed in many ways. The most direct way is to apply a function f to the original pixel intensity value, i.e. $I^e = f(I)$. This strategy is known as tone-curve adjustment. The function can be determined either manually or by selecting from pre-defined curves functions. Alternatively, the function can also be based on automatic histogram equalization, which obtains f by considering the input image's histogram. Aside from applying a certain function, local image gradients can also be used as a cost function that is optimized to boost contrast [Majumder and Irani 2007].

Recovering visibility in bad weather or underwater is, in fact, a specific contrast enhancement problem [Tan 2008; He et al. 2011; Chiang and Chen 2012; Ancuti et al. 2012]. Optically, poor visibility in bad weather or underwater is due to the substantial presence of medium particles that have significant size and distribution [Tan 2008]. Light from the atmosphere and light reflected from an object are absorbed and scattered by those particles, leading to contrast reduction and thus

to the degraded images. Most current dehazing algorithms try to estimate either airlight or transmission map (see [Tan 2008; Fattal 2008; He et al. 2011]). Regardless the algorithms, the outputs of visibility enhancement show clear increase of contrast.

Multi-band Image Decomposition A common practice in solving computer vision and computational photography problems is to decompose images into different layers (or scales) and recombine them (*e.g*. multi-band image blending [Burt and Adelson 1983], optical flow estimation [Wedel et al. 2009], *etc*.). The most related works to ours in this direction are tone-mapping methods (*e.g*. [Durand and Dorsey 2002]), which attempt to reduce the contrast of a high dynamic range image to a limited range while preserving its details. This is usually achieved by reducing the contrast to the coarse layer and adding back the initial detail layer. Unlike these tone-mapping methods' problem, we want to increase the contrast but not the noise/artifacts. As a result, we need to put more effort on processing the detail layer.

6.3 Proposed Method

Our basic pipeline is illustrated in figure 6.2. It starts by decomposing the original input image into two layers: structure and texture layers. This takes the form of the layer decomposition problem proposed in chapter 1. The input image can be considered as the superimposition of the two layers:

$$I = I_S + I_T, \tag{6.1}$$

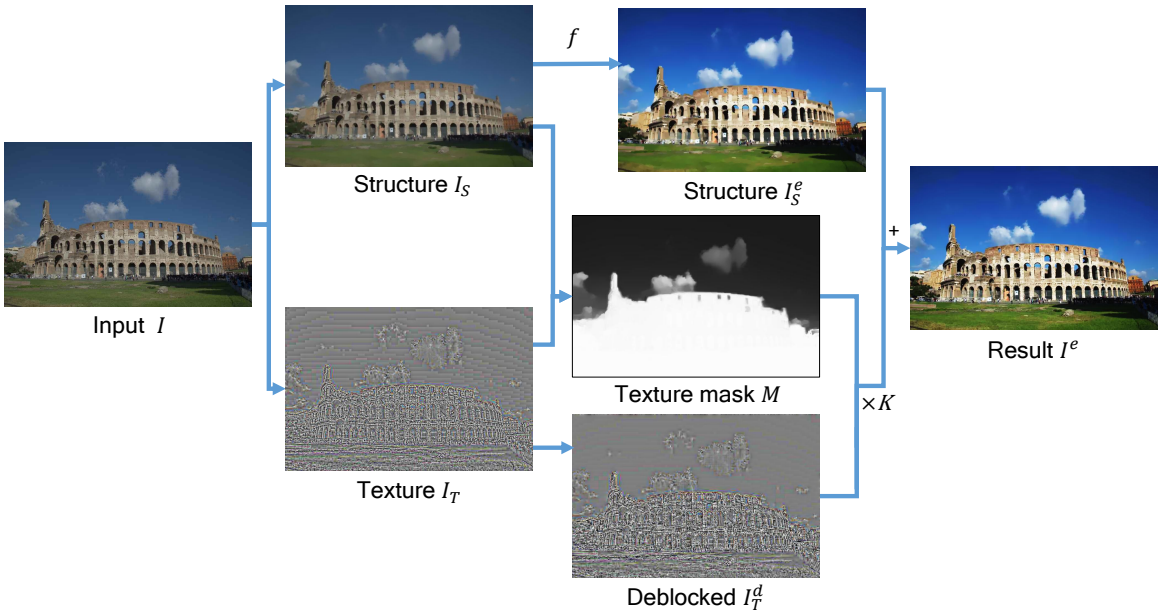


Figure 6.2: The overview of our proposed method. The input image is decomposed into structure and texture components. The contrast of the structure component is then boosted directly; the texture component that contains the JPEG artifacts is processed to reduce compression artifact. The two components are recombined at the last step to render the final result.

where I_S is the structure layer corresponding to the main large objects in the image, and I_T is the texture layer corresponding to the fine details [Aujol et al. 2006]. The contrast of the structure layer is then enhanced according to our task (*e.g.*, tone-curve adjustment or dehazing). The texture layer is processed through a combination of image matting and deblocking to remove compression artifacts. Finally the two layers are recombined to produce the final output. In the following, the details of each step are discussed.

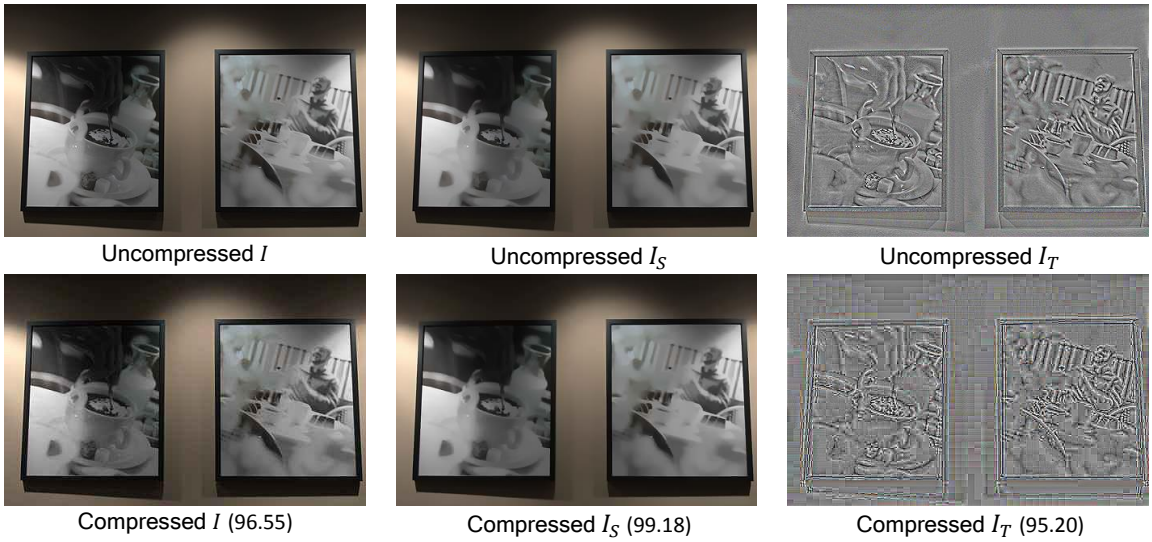


Figure 6.3: This shows two examples of structure-texture decomposition in uncompressed and compressed image ($Q40$) pairs. The structure similarity index measurement (SSIM) [Wang et al. 2004] values (in $\times 100$ scale in this chapter) between each pair are shown in the brackets. Notice that most of the characteristic compression artifacts exist in the texture layer, while the structure layer of the compressed image resembles that of the uncompressed image.

6.3.1 Structure-Texture Decomposition

To decompose the input image into a structure layer and texture (high-frequency) layer, any edge-aware smoothing operation (*e.g.* bilateral filter [Tomasi and Manduchi 1998], weight least square filter [Farbman et al. 2008]) can be applied. This procedure produces an image that retains strong structure and over-smooths out details. We take this image as the structure layer I_S, and obtain the texture layer by calculating the difference between the input image and its structure layer, $I_{T_i} = I_i - I_{S_i}$.

In our problem we applied the the total-variation (TV) image-reconstruction formulation based on Rudin-Osher-Fatemi method [Rudin et al. 1992]. Based on the TV regularization, the structure layer I_S is obtained by minimizing the following

objective function:

$$\min_{I_S} \sum_i (I_{S_i} - I_i)^2 + \lambda |\nabla I_{S_i}|, \tag{6.2}$$

where i is the pixel index, λ is the regulation parameter and ∇ is the gradient operator. An efficient half-quadratic splitting scheme to solve equation (6.2) is described in [Wang et al. 2008].

This structure-texture decomposition exploits the fact that most of the structure layer is related to larger gradient magnitudes, while the texture layer captures both fine image details and compression artifacts that exhibit smaller gradient magnitudes. The parameter λ is important for controlling this separation and needs to be adjusted according to the compression factor, i.e., more compression requires λ to be increased. We show the values of λ used for different compression levels in the experiments section. There are methods for deblocking using TV regularization (*e.g.* [Goto et al. 2011]). The main difference here is that they do not explicitly process the texture layer, while our method put significant effort on processing the texture layer as will be described later. As a result, TV-based deblocking methods tend to suffer from over-smoothing, while ours preserves more details.

Figure 6.3 shows two examples of the structure-texture decomposition results for the same images: one image is compressed and the other is not. As can be observed, unlike the texture layers that contain different information due to the artifacts, the structure layers are almost identical (both from the visual quality perspective and from the structure similarity index measurement, SSIM, perspective [Wang et al. 2004]). This shows the effectiveness of the TV regularization in producing a structure layer that significantly filter out any compression artifacts. As such,

this image layer is considered to be artifacts free and suitable to be boosted using the desired enhancement operation directly, resulting in the enhanced version of the structure, I_S^e.

6.3.2 Reducing Artifacts in the Texture Layer

Since the texture layer contains both scene details and compression artifacts, it needs further refinement to be able to remove artifacts and to keep scene details. To do this, we create a mask M that separates regions, where the most scene details are presence, from the remaining regions. Having created the mask, for the regions inside the mask, we refine them further to remove potential ringing and blocking artifacts. For the remaining regions, which are those outside the mask, we remove the content altogether, since the content is most likely compression artifacts.

Scene Detail Extraction

To create the image mask, M, we apply the discrete cosine transform (DCT) to each 8×8 patches in the texture layer. We use the DCT high-frequency layer to serve as a likelihood of the scene details, i.e. stronger high-frequency DCT coefficients means more details. Denoting the 8×8 DCT of one block as matrix B, then the likelihood of this block to be part of the scene details can be expressed as:

$$t = \sum_{u,v} B_{u,v}^2 - B_{1,1}^2 - B_{1,2}^2 - B_{2,1}^2, \tag{6.3}$$

where u, v denotes the position in the DCT. We take the sum of squares of all DCT coefficients except $B_{1,1}, B_{1,2}$ and $B_{2,1}$, and apply a threshold to the likelihood to make a binary indication of each block. The threshold we use is empirically set to

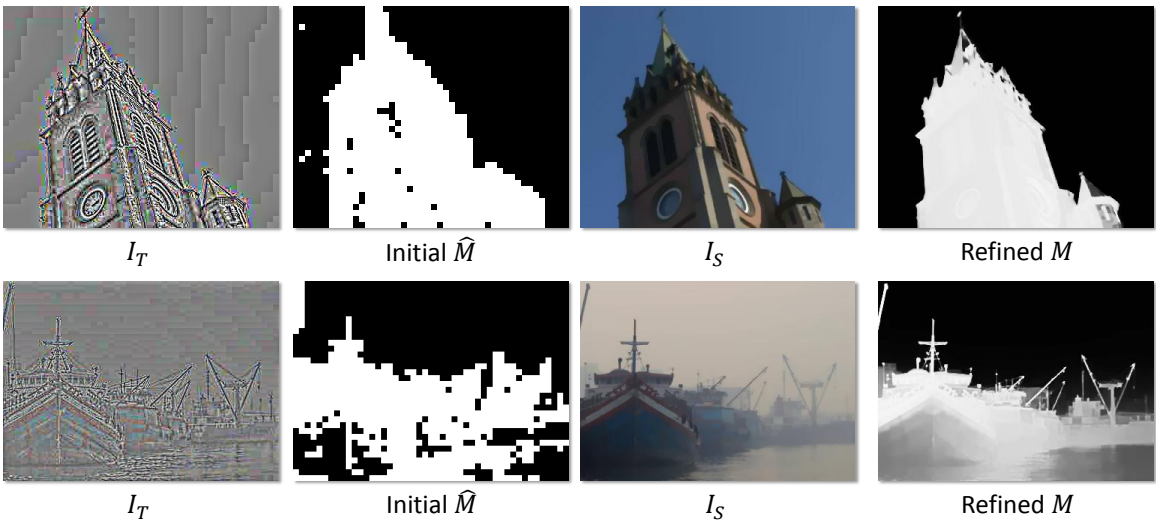


Figure 6.4: This shows two examples of the scene detail map generation. The initial results obtained by checking DCT coefficients are rough estimations. A soft matting technique can help refine the map by applying it to the structure layer, and the result is well aligned with the objects in the images

0.1. This initial block-wise estimation of texture region, denoted as $\hat{M}$, is a coarse estimate, as shown in the second column of figure 6.4.

This initial mask provides the regions of image details, but is too coarse for practical use. Thus, we apply a refinement step to better align the texture region with the structure layer. For this, we use a soft matting technique (inspired by [He et al. 2011]) by minimizing the following function on the scene detail map M:

$$\min_{\mathbf{m}}(\mathbf{m}-\hat{\mathbf{m}})^{\top}(\mathbf{m}-\hat{\mathbf{m}})+\alpha\mathbf{m}^{\top}L_s\mathbf{m}, \tag{6.4}$$

where $\mathbf{m}$ and $\hat{\mathbf{m}}$ are the vector forms of matrix M and $\hat{M}$, respectively. L_s is Levin's [Levin et al. 2008] matting Laplacian matrix generated from I_S. The smallest eigenvectors of the matting Laplacian correspond to the partitioning of images [Levin et al. 2008]. The first term forces the agreement with the initial estimation

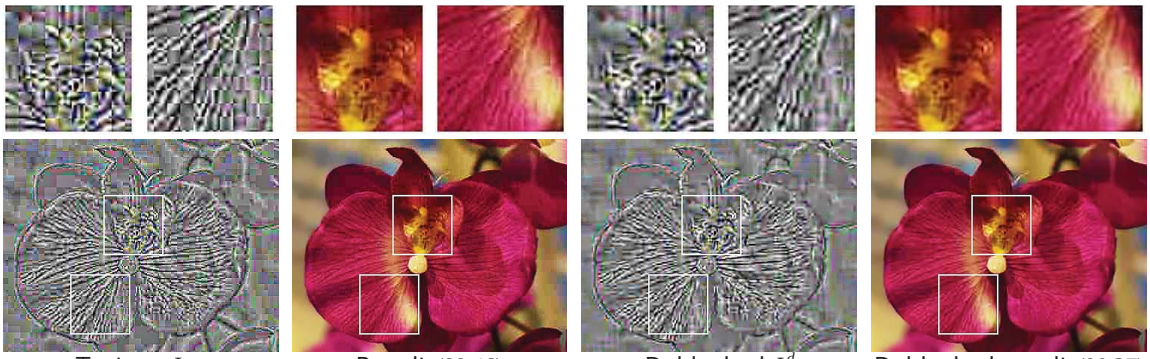


Figure 6.5: This shows the effect of blocking artifacts reduction. The left side shows the textural layers and its corresponding final composition results without the blocking artifacts reduction step. The right side shows the same pair but with the effect of blocking artifacts reduction. As can be seen both in texture and final results that the block is less noticeable when we apply the block artifacts reduction. The similarity against ground truth using SSIM for with and without deblocking are also shown in the bracket.

$\hat{M}$, while the second term forces the output to be aligned with the structure layer I_S. We set the regularization parameter α a large value (10^5 in our implementation), since it will provide clearer edges in the mask M. The last column of figure 6.4 shows the texture region map after refinement using the structure I_S. Most of the values in the map are near 0 or 1 (close to binary), but some values are between the two.

The result is a mask M whose edges have been refined. Another benefit of aligning the mask to the structure layer is that small amounts of textures around edges, which are indicative to ringing artifacts, are removed.

Block Artifacts Reduction

Having created the mask indicating the regions of scene details, we now try to reduce the potential blocking artifacts in the regions. Denoting the texture image

after blocking artifacts suppression as $I^d_{T_i}$, an objective function is defined as follows:

$$\min_{I^d_{T_i}} \sum_i (I^d_{T_i} - I_{T_i})^2 + \beta \sum_{i \in \eta} (\nabla I^d_{T_i})^2, \tag{6.5}$$

where i is the pixel index, and η are the locations at the 8×8 block borders. The first term forces the output to be similar to the input, while the second term smooths the edges at the 8×8 block borders, since they are more likely to be block artifacts. The smoothness level is controlled by the weight term β. We empirically set it 0.5 to achieve a proper compromise between oversmoothness and noticeable artifacts. This is effective in reducing the blockings in the texture map and result in a higher quantitative score as can be seen in figure 6.5.

6.3.3 Layer Recomposition

Having removed the artifacts in the texture layer, we now need to apply an enhancement operation to the texture layer before adding it back to I^e_S. However, since most contrast functions f are not linear and thus $f(I_S + I_T) \neq f(I_S) + f(I_T)$, we cannot simply apply the same process and then sum them up. As a consequence, we have to approximate the enhancement function adjustment by finding a scale multiplication factor K, which should obey the following condition as much as possible: $f(I) = f(I_S) + KI_T$, where I is the original input image. By denoting the enhanced texture layer as I^e_T, we intend to find the scale factor K:

$$I^e_T = K \circ M \circ I^d_T, \tag{6.6}$$

where $\circ$ is the element-wise multiplication operator. $M \circ I_T^d$ combines the steps in the previous section that generates the masked texture layer with reduced artifacts.

Like in the case of enhancing contrast for the structure layer I_S, the scale factor depends on the applications. For the application of image tone-curve adjustment, the tone-curve function f is applied to the intensity values of the input image, I. Taylor series $f(t + \Delta t) \approx f(t) + f'(t)\Delta t$ allows us to write:

$$f(I_{S_i} + I_{T_i}) = f(I_{S_i}) + f'(I_{S_i})I_{T_i}. \tag{6.7}$$

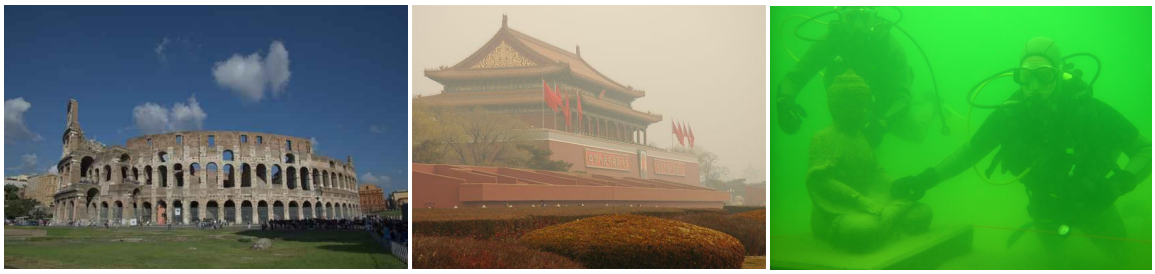
Hence, from the last equation, we have the scale factor for the tone adjustment $K_i = f'(I_{S_i})$.

In the dehazing or underwater application, the enhancement should consider the optical model of scattering media, which according to [He et al. 2011], the output of the enhancement should follow the following equation:

$$I_i^e = \frac{I_i - A}{t_i} + A, \tag{6.8}$$

where the I_i is the input image, A is the atmospheric light, t_i is the transmission, and i is pixel index. Therefore, the scale factor, K_i, should be approximately equal to $\frac{1}{t_i}$, since A is a constant and I_i^e is in $\frac{I_i}{t_i} + k$ form. Following [He et al. 2011], t is obtained from dark channel prior and A is obtained from the patch with the brightest intensity in dark channel.

Having recovered both the structure and texture layers, the final result can be achieved by simply summing up the two layers: $I^e = I_S^e + I_T^e$.



Low contrast (Q 40) Haze (Q 80) Underwater (Q 30)

Figure 6.6: This figure shows the inputs in the section which require contrast boosting.

Table 6.1: Average Runtime Comparison of FoE [Sun and Cham 2007], NN [Burger et al. 2012], SA-DCT [Foi et al. 2007] and ours

Method	SA-DCT	FoE	NN	Ours
Runtime(s)	20	287	25	15

6.4 Results

We evaluated our proposed framework by applying it to various contrast enhancement tasks: image tone-curve adjustment, dehazing and underwater visibility enhancement. Experiments were performed on a PC with Intel I7 CPU (3.4GHz) with 8GB RAM. The test images were either self-taken or downloaded from the Internet. Three examples are shown in figure 6.6. Note that, in these input images, there are often no noticeable artifacts. The artifacts become apparent after the contrast enhancement is applied.

The entire process for an image (approximately 500×600 in size) using our

Table 6.2: Quantitative Comparison of FoE [Sun and Cham 2007], NN [Burger et al. 2012], SA-DCT [Foi et al. 2007] and ours

Method	simple boosted	FoE	NN	SA-DCT (Pre)	SA-DCT (Post)	Ours
SSIM avg.	90.79	91.14	91.88	92.03	91.79	**92.05**
PSNR avg.	29.17	29.69	29.94	**30.12**	29.42	29.76

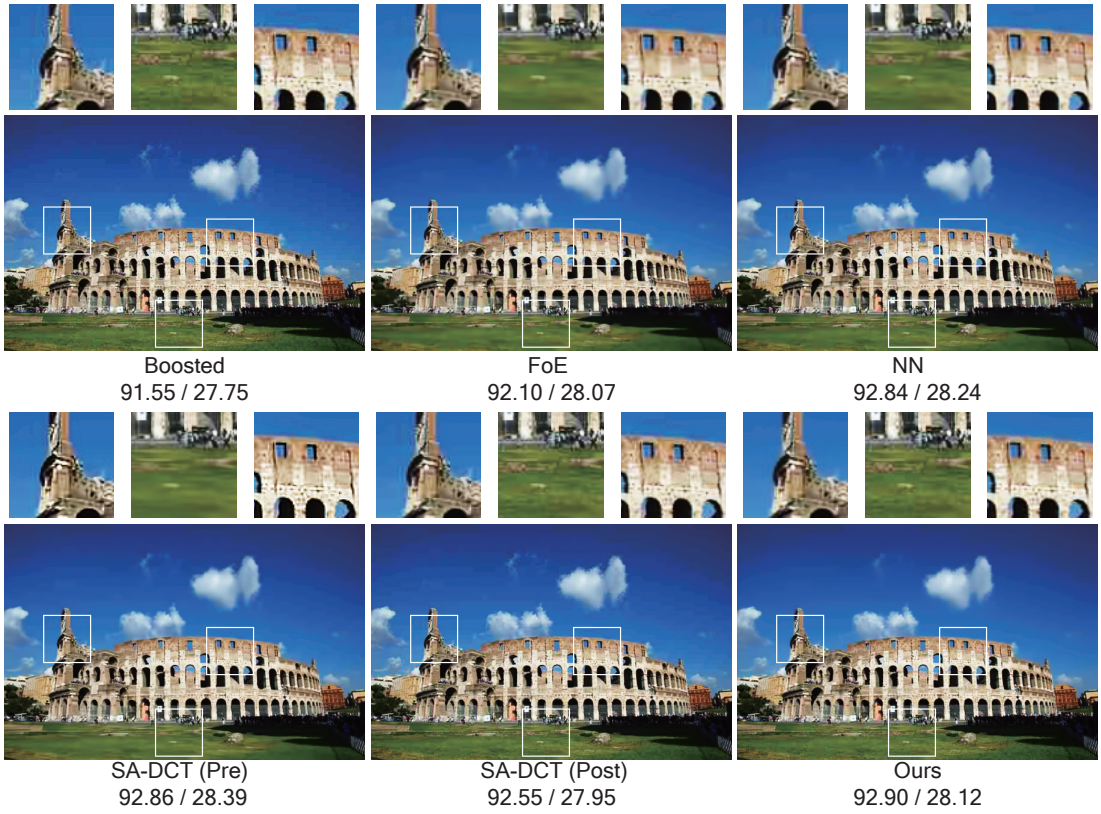


Figure 6.7: This figure shows an example in image tone-curve adjustment using FoE [Sun and Cham 2007], NN [Burger et al. 2012], SA-DCT [Foi et al. 2007]] and our approach. Shown below the images are the comparison SSIM/PSNR(dB) with respect to the groundtruth.

current un-optimized matlab implementation took about 15 seconds with the main bottleneck being the image matting which took more than 10 seconds. The only parameter that needs to be changed was the regulation term λ in the structure-texture decomposition in equation (6.2). This parameter was set according to the compression level. Higher compression requires larger λ for the decomposition. $\lambda = 0.02, 0.03, 0.04, 0.05$ is used for $> Q70$, $Q50 - Q70$, $Q30 - Q50$ and $< Q30$, respectively.

We compared our approach with several state-of-the-art deblocking methods:

a local filtering based method - shape adaptive DCT(SA-DCT) [Foi et al. 2007], a reconstruction based using Field of Experts (FoE) prior [Sun and Cham 2007] as well as a learning based method using Neural Network (NN) [Burger et al. 2012]. These methods were all used as both a pre-processing and post-processing step for the contrast enhancement methods. We note that the comparison with NN is not fair since it is a more general algorithm targeting on any kinds of noise (i.e. not just JPEG artifacts). The average run-times of these algorithms are summarized in table 6.1. Interestingly, even though we apply layer decomposition and matting as parts of our procedure, our method has the fastest performance among all.

For experiments involving tone-curve manipulation, we can also provide a quantitative comparison with the groundtruth. The groundtruth image is obtained by enhancing the uncompressed image using the same tone-curve. Quantitative results are reported using the perceptually-based quality measurement-structure similarity index (SSIM) [Wang et al. 2004] (in $\times 100$ scale) as well as the peak signal-to-noise ratio (PSNR). Table 6.2 summarizes the average SSIM and average PSNR on all our 15 test cases and at different compression levels (from $Q20$ to $Q90$). Our approach achieves the highest SSIM but not the highest PSNR. As sometimes the case with PSNR, we believe it does not properly reflect the qualitative results. On visual inspection of the images, it is clear our approach is qualitatively better than the other methods.

Figure 6.7 shows a tone-curve adjustment comparison. As can be seen, FoE and NN successfully removed block artifacts which resulted in overall improvements in both PSNR and SSIM. However, they tended to smooth sharp edges and details in the image. SA-DCT lost its effectiveness in deblocking when used after the enhancement, but when used before the enhancement, SA-DCT did a good job

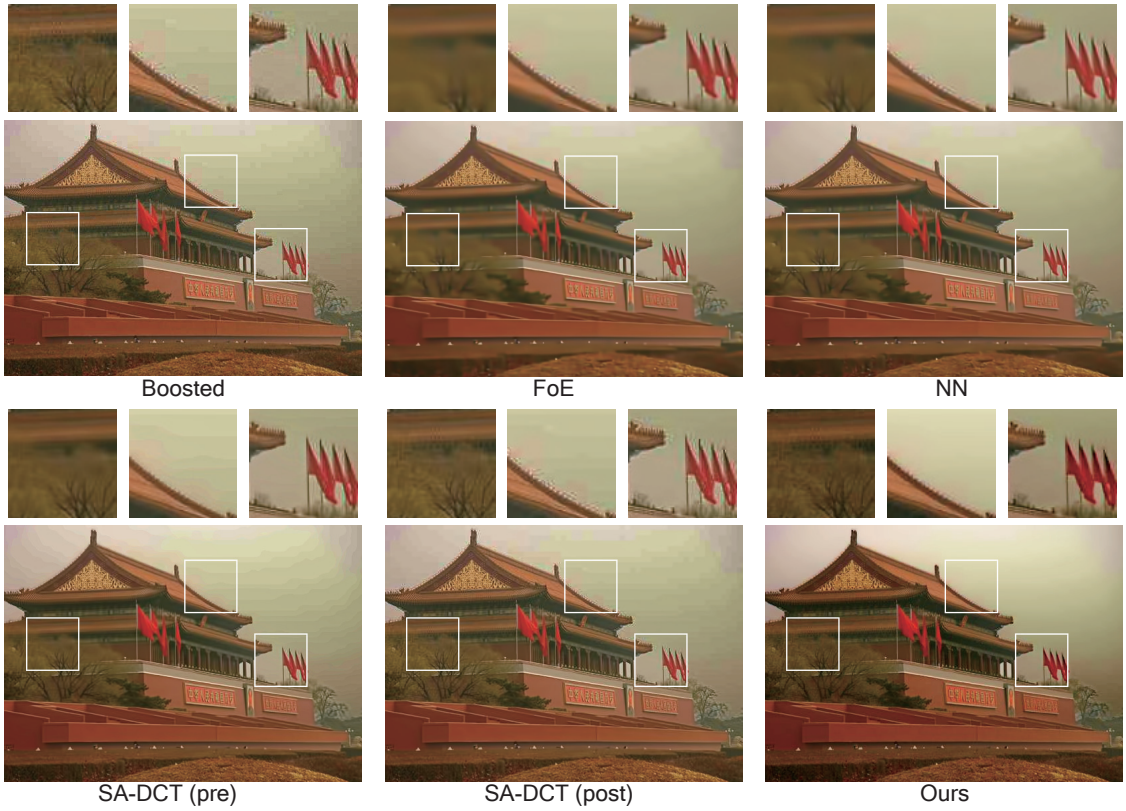


Figure 6.8: This figure shows an example of dehazing using of FoE [Sun and Cham 2007], NN [Burger et al. 2012], SA-DCT [Foi et al. 2007] and our approach.

and achieved the highest PSNR. However, upon close visual inspection, the results of our method are much cleaner (less ringing artifacts) and more image details preserved, resulting the highest SSIM value.

Figures 6.8 and 6.9 show examples of applying our method to dehazing and underwater visibility enhancement. Here, since we do not have the groundtruth recovered image, we can only show qualitative visual comparisons. In these applications, the advantage of our method becomes more observable. The results of using FoE and NN are over smoothed, causing them to lose details. SA-DCT slightly outperformed FoE and NN in reducing the compression artifacts. Ours is

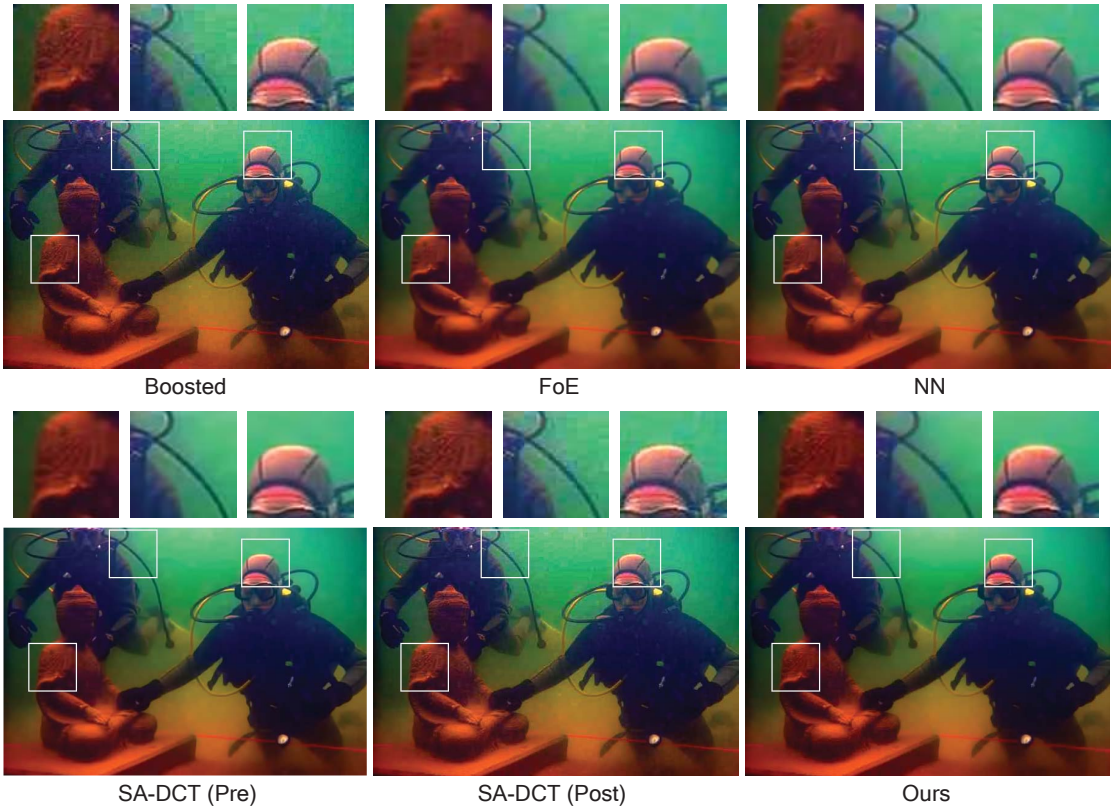


Figure 6.9: This figure shows an example of underwater image enhancement using FoE [Sun and Cham 2007], NN [Burger et al. 2012], SA-DCT [Foi et al. 2007] and our approach.

better in terms of removing artifacts (particularly with much less ringings) as well as preserving image details.

6.5 Discussion and Conclusion

We have introduced a framework to suppress artifacts appearing in JPEG images that becomes prominently visible when applying contrast enhancement. While the proposed framework is admittedly engineering in nature, our strategy of using structure and texture layer decomposition enables us to reduce the compression

artifacts in parallel with contrast enhancement, and to process them independently to each other. With this integrated framework, the key benefit is that we can process two tasks that are opposite to each other in terms of functionality. On one hand, we have a task to suppress noise as much as possible; on the other hand, within the same image, we have a task to enhance the content as much as possible. If these two tasks are processed sequentially, as pre- or post-processing, the results are not likely to be optimum. Since, the process of artifacts removal as pre-processing will remove the image content that have low contrast, and as post-processing will be affected by the enhanced artifacts. As shown in our experiments, we have demonstrated the effectiveness of the proposed framework using qualitative and quantitative measures.

While our approach targets suppressing JPEG compression artifacts for the task of contrast enhancement, our framework is suitable to other applications that have the same nature of problem. We consider JPEG compression artifacts to be an important problem because these are commonly troublesome for many computer vision and image processing algorithms that assume the input images have little noise. We also consider contrast enhancement, since it is one of the core operations in the low-level computer vision and image processing. Among other applications, it is crucially used to deal with turbid media, such as haze, fog, rain, and underwater, which has been addressed considerably in computer vision community recently.

Regarding our framework, the remaining question is whether our structure, texture, and masked texture layers can effectively distill JPEG images into a layer that is mostly image content and also into another layer that is mostly affected by compression artifacts. While our practical findings discussed in this chapter have

given us a positive answer (and we consider as a contribution that can be improved further), rigorous evaluation is still needed, and we will consider this in our future work.

Chapter 7

Conclusion

This chapter provides a summary of the works presented in the previous four chapters in this thesis. While each previously mentioned chapter has a self-contained summary and discussion, this chapter serves to re-iterate those summaries and discussions. Meanwhile, we also describe potential directions for future work.

7.1 Summary

This thesis has developed three distinct works for layer separation problems and one application based on layer separation. Improvements over previous methods have been demonstrated through experiments. Specifically we note the following findings:

- In chapter 3, we have presented a method to automatically remove reflection interference due to a glass surface. Our approach works by capturing a set of images of a scene from slightly different viewpoints. By aligning all the input images and labelling out the two kind of edges from statistics, a

subsequent layer separation can be achieved. Our approach can produce clearer layer separation results compared with existing methods. Moreover, unlike previous motion based methods, our approach does not need any constraints on scenes geometry nor require the reflection to be static. This makes our method a practical one for use in casual imaging scenarios.

- In chapter 4, we have presented a method to automatically extract two layers from one image where one layer is smoother than the other. Our approach works by building two likelihoods for each layer from gradient histograms, which models this relative smoothness. In order to solve the layer separation problem, an objective function that finds the most likely separation has been proposed. We have also derived an efficient scheme to optimize our objective function which is non-convex and has an inequality constraint. Tests on two layer separation problems of intrinsic image decomposition and reflection removal using defocus blur show our method can provide high-quality results that are comparable to the results by state-of-the-art methods. But our method is significantly faster than previous works.

- In chapter 5, we have developed a method for nighttime haze removal in the presence of glow and multiple scene light sources. We have introduced a new haze model specific for nighttime scene that incorporates the presence of glow and allows for spatially varying atmospheric light. Based on our nighttime haze model, we have also proposed a framework to first decompose the glow image from the nighttime haze image, by using our method in chapter 4. After that, a spatially varying atmospheric light map has been introduced to deal with the problem of multiple light colors. Compared with both previous

daytime dehazing and nighttime dehazing approaches, our method is shown to produce more favourable results.

- In chapter 6, as an application of layer separation, we have introduced a framework to suppress artifacts appearing in JPEG images which become prominently visible when applying contrast enhancement. The used structure and texture layer decomposition enables us to reduce the compression artifacts in parallel with contrast enhancement, and to process them independently to each other. As shown in our experiments, we have demonstrated the effectiveness of the proposed framework using qualitative and quantitative measures.

7.2 Future Directions

Future directions of research mentioned in prior chapters are re-iterated here. In particular, the following issues remain open and require further investigations:

- Our reflection removal method in chapter 3 only considers the dense correspondence with regards to the background to align the input images. One extension would be joint estimation of the two layers (reflection and background) motion and layer recovery. It is hard to recover two individual motions from the mixture image and recover the two layers simultaneously. Therefore alternating motion estimation and layer recovery may be the correct way. With proper regularizations (*e.g.* pairwise smoothness) on the motions of the two layers, it may help reduce the ambiguity when the reflection is prominent or the background has large textureless regions. As we

have mentioned in chapter 3, these two cases are the most challenging ones for our current approach.

- Our single image layer separation using relative smoothness proposed in chapter 4 is a general method. Our approach was demonstrated on two applications: intrinsic images and reflection separation in chapter 4. The work in chapter 5 also apply this method for separating the glow layer. Exploiting other layer separation problems that may benefit from our method is an interesting future direction. One example under consideration is the problem of removing adherent raindrops in images [You et al. 2013]. There are situations that our lens get raindrop/waterdrop adherent and results in another layer added to the clear image. If our focus is on the the scene at far distance, the raindrop layer would be smooth. In such a case, our method may help remove the raindrop layer.

- In our experiment part in chapter 5, we have used one synthetic data to quantitatively compare different methods. This is rarely done in prior published works. As we have mentioned in chapter 2, there are a lot of recent works addressing the problem of dehazing. Unfortunately, although their results look good for human visual perception, there is no quantitative measurement of all these methods. One of the reasons is that there is no ground truth data used in the community as a benchmark. This hinders the further progress of the field; as a new proposed method is difficult to be measured, and new problems are difficult to be identified. The MIT dataset [Grosse et al. 2009] is one successful case to demonstrate the use of ground truth dataset. The presence of the MIT dataset spurs a lot of modern approaches for intrinsic

image decompositions, *e.g.* [Gehler et al. 2011; Barron and Malik 2012]. Inspired by this, we can try to introduce a new data set of degraded images along with their ground truths. This dataset may be obtained using artificial ones or using captured images from a controlled indoor environment.

- While our approach in chapter 6 targets suppressing JPEG compression artifacts for the task of contrast enhancement, our framework is suitable to other applications that have the same nature of problem. One problem under consideration is to apply image segmentation for JPEG compressed images. Another interesting extension of our current approach is to study how to adjust our framework to solve the same problem in compressed videos. Simply applying our method to each frame may result in flickering in video. Temporal smoothness may need to be taken into consideration to stabilize the video.

Bibliography

Agrawal, A. K., Raskar, R., Nayar, S. K., and Li, Y. 2005. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics 24*, 3, 828–835.

Ancuti, C. O., and Ancuti, C. 2013. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing 22*, 8, 3271–3282.

Ancuti, C., Ancuti, C. O., Haber, T., and Bekaert, P. 2012. Enhancing underwater images and videos by fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Aujol, J.-F., Gilboa, G., Chan, T., and Osher, S. 2006. Structure-texture image decompositionmodeling, algorithms, and parameter selection. *International Journal of Computer Vision 67*, 1, 111–136.

Barron, J. T., and Malik, J. 2012. Color constancy, intrinsic images, and shape estimation. In *European Conference on Computer Vision*.

Barrow, H. G., and Tenenbaum, J. M. 1978. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*.

Bousseau, A., Paris, S., and Durand, F. 2009. User-assisted intrinsic images. *ACM Transactions on Graphics 28*, 5, 130:1–130:10.

Burger, H. C., Schuler, C. J., and Harmeling, S. 2012. Image denoising: Can plain neural networks compete with bm3d? In *IEEE Conference on Computer Vision and Pattern Recognition*.

Burt, P. J., and Adelson, E. H. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4, 532–540.

Chiang, J. Y., and Chen, Y.-C. 2012. Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing 21*, 4, 1756–1769.

Dong, W., Zhang, L., and Shi, G. 2011. Centralized sparse representation for image restoration. In *IEEE International Conference on Computer Vision*.

Durand, F., and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics 21*, 3, 257–266.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics 27*, 3, 67.

Farid, H., and Adelson, E. H. 1999. Separating reflections from images by use of independent component analysis. *Journal of the Optical Society of America A 16*, 9, 2136–2145.

Fattal, R. 2008. Single image dehazing. *ACM Transactions on Graphics 27*, 3, 72.

Fattal, R. 2014. Dehazing using color-lines. *ACM Transaction on Graphics 34*, 1, 13:1–13:14.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. 2006. Removing camera shake from a single photograph. *ACM Transactions on Graphics 25*, 3, 787–794.

Foi, A., Katkovnik, V., and Egiazarian, K. 2007. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing 16*, 5, 1395–1411.

Gai, K., Shi, Z., and Zhang, C. 2008. Blindly separating mixtures of multiple layers with spatial shifts. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Gai, K., Shi, Z., and Zhang, C. 2012. Blind separation of superimposed moving images using image statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 1, 19–32.

Gao, J., Li, Y., Chin, T.-J., and Brown, M. S. 2013. Seam-driven image stitching. In *Eurographics 2013*.

Gehler, P. V., Rother, C., Kiefel, M., L. Zhang, L., and Schölkopf, B. 2011. Recovering intrinsic images with a global sparsity prior on reflectance. In *Neural Information Processing Systems*.

Geman, D., and Yang, C. 1995. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing 4*, 7, 932–946.

Goto, T., Kato, Y., Hirano, S., Sakurai, M., and Nguyen, T. Q. 2011. Compression artifact reduction based on total variation regularization method for mpeg-2. *IEEE Transactions on Consumer Electronics 57*, 1, 253–259.

Grosse, R., Johnson, M. K., Adelson, E. H., and Freeman, W. T. 2009. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *IEEE International Conference on Computer Vision*.

Guo, F., Li, Y., Kankanhalli, M. S., and Brown, M. S. 2013. An evaluation of wearable activity monitoring devices. In *ACM Multimedia workshop on Personal data meets distributed multimedia*.

Guo, X., Cao, X., and Ma, Y. 2014. Robust separation of reflection from multiple images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

He, K., Sun, J., and Tang, X. 2010. Guided image filtering. In *European Conference on Computer Vision*.

He, K., Sun, J., and Tang, X. 2011. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 12, 2341–2353.

Jacobs, N., Burgin, W., Fridrich, N., Abrams, A., Miskell, K., Braswell, B. H., Richardson, A. D., and Pless, R. 2009. The global network of outdoor webcams: Properties and applications. In *ACM International Conference on Advances in Geographic Information Systems*.

Joze, H. R. V., Drew, M. S., Finlayson, G. D., and Rey, P. A. T. 2012. The role of bright pixels in illumination estimation. In *Color and Imaging Conference*, 41–46.

Kong, N., Tai, Y.-W., and Shin, S. Y. 2012. A physically-based approach to reflection separation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Koschmieder, H. 1925. *Theorie der horizontalen Sichtweite: Kontrast und Sichtweite*. Keim & Nemnich.

Land, E. H., and McCann, J. J. 1971. Lightness and retinex theory. *Journal of the Optical Society of America 61*, 1, 1–11.

Lee, Y., Kim, H., and Park, H. 1998. Blocking effect reduction of jpeg images by signal adaptive filtering. *IEEE Transactions on Image Processing 7*, 2, 229–234.

Lee, K., Kim, D. S., and Kim, T. 2005. Regression-based prediction for blocking artifact reduction in jpeg-compressed images. *IEEE Transactions on Image Processing 14*, 1, 36–48.

Levin, A., and Weiss, Y. 2004. User assisted separation of reflections from a single image using a sparsity prior. In *European Conference on Computer Vision*.

Levin, A., and Weiss, Y. 2007. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 9, 1647–1654.

Levin, A., Lischinski, D., and Weiss, Y. 2008. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 2, 228–242.

Li, Y., and Brown, M. S. 2013. Exploiting reflection change for automatic reflection removal. In *IEEE International Conference on Computer Vision*.

Li, Y., and Brown, M. S. 2014. Single image layer separation using relative smoothness. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Y., Guo, F., Tan, R. T., and Brown, M. S. 2014. A contrast enhancement framework with jpeg artifacts suppression. In *European Conference on Computer Vision*.

Li, Y., Min, D., Brown, M. S., Do, M. N., and Lu, J. 2015. Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. In *IEEE International Conference on Computer Vision*.

Li, Y., Tan, R. T., and Brown, M. S. 2015. Nighttime haze removal with glow and multiple light colors. In *IEEE International Conference on Computer Vision*.

Liu, C., Yuen, J., and Torralba, A. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 5, 978–994.

Majumder, A., and Irani, S. 2007. Perception-based contrast enhancement of images. *ACM Transactions on Applied Perception 4*, 3, 17.

Meer, P. 2004. Robust techniques for computer vision. *Emerging Topics in Computer Vision*.

Meng, G., Wang, Y., Duan, J., Xiang, S., and Pan, C. 2013. Efficient image dehazing with boundary constraint and contextual regularization. In *IEEE International Conference on Computer Vision*.

Narasimhan, S. G., and Nayar, S. K. 2003. Shedding light on the weather. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Nishino, K., Kratz, L., and Lombardi, S. 2012. Bayesian defogging. *Intl J. Computer Vision 98*, 3, 263–278.

Ohnishi, N., Kumaki, K., Yamamura, T., and Tanaka, T. 1996. Separating real and virtual objects from their overlapping images. In *European Conference on Computer Vision*.

Pei, S.-C., and Lee, T.-Y. 2012. Nighttime haze removal using color transfer pre-processing and dark channel prior. In *IEEE International Conference on Image Processing*.

Pharr, M., and Humphreys, G. 2010. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Rudin, L. I., Osher, S., and Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60*, 1, 259–268.

Sarel, B., and Irani, M. 2004. Separating transparent layers through layer information exchange. In *European Conference on Computer Vision*.

Sarel, B., and Irani, M. 2005. Separating transparent layers of repetitive dynamic behaviors. In *IEEE International Conference on Computer Vision*.

Schechner, Y. Y., and Karpel, N. 2004. Clear underwater vision. In *IEEE Conference on on Computer Vision and Pattern Recognition*.

Schechner, Y. Y., Kiryati, N., and Basri, R. 2000. Separation of transparent layers using focus. *International Journal of Computer Vision 39*, 1, 25–39.

Shechner, Y. Y., Shamir, J., and Kiryati, N. 1999. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *IEEE International Conference on Computer Vision*.

Shechner, Y. Y., Shamir, J., and Kiryati, N. 2000. Polarization and statistical analysis of scenes containing a semireflector. *Journal of the Optical Society of America A 17*, 2, 276–284.

Shen, L., and Yeo, C. 2011. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shen, L., Tan, P., and Lin, S. 2008. Intrinsic image decomposition with non-local texture cues. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sulami, M., Geltzer, I., Fattal, R., and Werman, M. 2014. Automatic recovery of the atmospheric light in hazy images. In *IEEE International Conference on Computational Photography*.

Sun, D., and Cham, W.-K. 2007. Postprocessing of low bit-rate block dct coded images based on a fields of experts prior. *IEEE Transactions on Image Processing 16*, 11, 2743–2751.

Sun, D., S.Roth, and Black, M. 2010. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sun, D., Roth, S., and Black, M. J. 2014. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision 106*, 2, 115–137.

Szeliski, R., Avidan, S., and Anandan, P. 2000. Layer Extraction from Multiple Images Containing Reflections and Transparency. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tan, R. T. 2008. Visibility in bad weather from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tang, K., Yang, J., and Wang, J. 2014. Investigating haze-relevant features in a learning framework for image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tappen, M. F., Freeman, W. T., and Adelson, E. H. 2005. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 9, 1459–1472.

Tarel, J.-P., and Hautiere, N. 2009. Fast visibility restoration from a single color or gray level image. In *IEEE International Conference on Computer Vision*.

Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*.

Tsin, Y., Kang, S. B., and Szeliski, R. 2006. Stereo matching with linear superposition of layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 2, 290–301.

Wang, C. Y., and Lee, S. M. Chang, L. W. 2001. Designing jpeg quantization tables based on human visual system. *Image Communication 16*, 5, 501–506.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4, 600–612.

Wang, Y., Yang, J., Yin, W., and Zhang, Y. 2008. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences 1*, 3, 248–272.

Watson, A. 1993. Dct quantization matrices visually optimized for individual images. In *Proceedings of the International Society for Optics and Photonics*, vol. 1913, 202–216.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. 2009. An improved algorithm for tv-l 1 optical flow. *Statistical and Geometrical Approaches to Visual Motion Analysis*, 23–45.

Weiss, Y. 2001. Deriving intrinsic images from image sequences. In *IEEE International Conference on Computer Vision*.

Xu, L., Zheng, S., and Jia, J. 2013. Unnatural L_0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yang, Y., Galatsanos, N. P., and Katsaggelos, A. K. 1995. Projection-based spatially adaptive reconstruction of block-transform compressed images. *IEEE Transactions on Image Processing 4*, 7, 896–908.

Yim, C., and Bovik, A. 2011. Quality assessment of deblocked images. *IEEE Transactions on Image Processing 20*, 1, 88–98.

You, S., Tan, R., Kawakami, R., and Ikeuchi, K. 2013. Adherent raindrop detection and removal in video. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zakhor, A. 1992. Iterative procedures for reduction of blocking effects in transform image coding. *IEEE Transactions on Circuits and Systems for Video Technology 2*, 1, 91–95.

Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. 1999. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 8, 690–706.

Zhang, J., Cao, Y., and Wang, Z. 2014. Nighttime haze removal based on a new imaging model. In *IEEE International Conference on Image Processing*.

Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., and Lin, S. 2012. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 7, 1437–1444.