SHADOW ELIMINATION AND INTERPOLATION FOR COMPUTER VISION AND GRAPHICS

A Dissertation Presented

by

YASUYUKI MATSUSHITA

Submitted to the Graduate School of the
University of Tokyo in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2002

E.E.Dept., School of Engineering

SHADOW ELIMINATION AND INTERPOLATION FOR COMPUTER VISION AND GRAPHICS

A Dissertation Presented

by

YASUYUKI MATSUSHITA

Approved as to style and content by:

Masao Sakauchi, Chair

Kiyoharu Aizawa, Member

Hiroshi Harashima, Member

Katsushi Ikeuchi, Member

Masaru Kitsuregawa, Member

Yoichi Sato, Member

Shinichiro Ogaki, Department Chair
E.E.Dept., School of Engineering

To my parents.

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

First of all I would like to thank my advisor, Professor Masao Sakauchi, for his encouragement and strong support on numerous occasions and allowing me to pursue my research interest. His incredibly broad knowledge-base and his organizational skills helped me to find out how to survive a long voyage toward Ph.D.

Next but not less I thank my mentor, Professor Katsushi Ikeuchi, for introducing me to a rich and beautiful area of research and for his competent guidance throughout my work. I have been honored to share his superlative experience in the field of computer vision and also his excellent way of finding and solving problems and enjoying life.

I would like to thank people in our laboratory and Ikeuchi's laboratory, especially I would thank Ko Nishino, recently moved to Columbia University, who collaborated with me throughout my Ph.D work. His relentless pursuit of excellence and constant thirst for knowledge is both exemplary and inspirational. Without his support this Ph.D. would not have been possible. Particular mention must be made of Suguru Sato in our laboratory who provided much needed advice and guidance on using computers with his deepest knowledge particularly on programming.

I wish to express my deepest gratitude to my supervisor at Microsoft Research Asia, Harry Shum for his wisdom, creativity and strong power to realize ideas which lead completion of my work. His enthusiastic attitude towards research and everyday life and excellent sense of humor have been encouragement for me. Special thanks goes to Sing Bing Kang of Microsoft Research, Steve Lin, Xing Ton of Microsoft Research Asia for their helpful comments and suggestions in completing my project at Microsoft Research Asia.

All the participants in the partial studies of the research deserve individual mention for their kindness and contribution shown to me during the research; limitation of the page prevents me from doing so, because it will need more pages than this thesis itself.

The final acknowledgment goes to my family, who has provided me with a level of emotional and financial support that can only be given, not expected. It is to them that I dedicate this thesis.

ABSTRACT

SHADOW ELIMINATION AND INTERPOLATION FOR COMPUTER VISION AND GRAPHICS

DECEMBER 2002

by

YASUYUKI MATSUSHITA

Ph.D., THE UNIVERSITY OF TOKYO

Directed by: Professor Masao Sakauchi

Shadowing and other illumination effects give human beings rich clues to understand visual scenery. However, for many computer vision algorithms that rely on visual appearance, such illumination effects generally become more harmful than effective. Proper handling of shadowing effect has been a hard problem especially when computer vision algorithms are taken to outdoor scenes. Though an extensive amount of work has been done for the case of parameterized light sources to solve the problem, in practical terms we cannot expect such preassumptions are valid in outdoor scenes. We are motivated by this background, and our focus is proper handling of scene illumination using a set of images captured using a fixed camera but under several different illumination conditions without knowing the scene geometry nor lighting conditions. This work has two important parts.

The first part is estimation of scene illumination from a set of images captured under various illumination conditions, and interpolation of captured illumination

conditions. In this part, we first introduce an existing method to derive scene illumination image, and propose our approach to enhance the estimates. In addition, to deal with the non-linearity in variation of scene illumination along the time axis, we propose two different approaches for non-linear interpolation of scene illumination which is used to estimate intermediate illumination conditions.

The second part is manipulation of scene illumination using obtained scene illumination images to enhance the performance of computer vision algorithms such as object tracking in outdoor scene. This part can lead to some application areas such as shadow removal from the scene for robust visual surveillance, image-based scene texture editing for computer graphics and enhancement of image segmentation. In this thesis, we investigate on those applications and confirm the effectiveness of scene illumination handling. We also integrate our shadow elimination technique to an existing road traffic monitoring system and confirm that it enhanced the accuracy of object detection and tracking.

論文要旨

コンピュータビジョン・グラフィックスのための影の消去と補間

2002 年 12 月

松下 康之

東京大学 工学系研究科 電子情報工学専攻

指導教官: 坂内 正夫 教授

影を含む照明による影響は，視覚的な環境の構造を理解するための大きな手がかりを人間に与える．しかしながら，これらの照明による影響は多くのコンピュータ・ビジョンの手法に対して，手がかりを与えるのではなく，むしろその精度を下げる要因として問題視されてきた．特に，光線が物体に遮られる際に生じるキャストシャドウは，屋外におけるコンピュータビジョンシステムの性能を低下させる要因として捉えられており，これを適切に処理するフレームワークが求められている．これまでに光源を既知としてこのような問題に対処する研究が多くなされてきたが，屋外環境下では光線の分布が複雑であるため，このような前提を用いることは一般に難しい．このような背景を配慮し，本研究ではシーンの構造・照明条件が未知である画像列中で，照明による影響を適切に処理する手法について検討をおこなう．本論文は，二つの構成からなる．

前半では，固定視点から撮影された様々な照明条件下の静的なシーンの画像列から，シーンへ入射する照明分布を示す照明画像を推定，およびその補完をおこなう手法を提案する．この中では，照明画像を推定する既存の手法を拡張し，より正確な照明画像を導く手法について述べる．さらに，得られた照明画像列から，それらの中間

的な照明画像を補完する手法について検討する．照明画像の時系列変化は非線形であり，これに対処するために二つの非線形補完手法を提案する．

後半では，得られたシーンの照明画像を用いて入力画像の照明成分を正規化することにより，屋外での移動物体追跡に代表される，いくつかのコンピュータビジョンアルゴリズムの精度向上に関する検討をおこなう．照明成分の正規化により，入力画像列から影の消去，照明成分に依存しない二次元画像編集，さらに照明による影響を排除した画像を用いた画像分割等のアプリケーションが実現する．本稿では，これらの応用に関する検討をおこない，さらに既存の交通監視システムに本手法を組み込むことで物体追跡の精度が向上したことを確認した．

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Rapidly increasing availability of video sensors and less expensive high performance computers accelerate research on video understanding problems. *Video understanding* has been a Computer Vision problem for a long time, and an extensive amount of methods has been proposed to solve its central problem, i.e. object detection and object tracking. Compared to *Image Understanding*, which uses only a single image, video understanding allows to use massive information to solve those problems. Though computational cost may be higher in video understanding, but the problem becomes easier because of its much richer information. The goal of video understanding is that automatically deriving a description understandable for human beings from an image sequence. The problem substantially involves Artificial Intelligence problems.

Apart from the pure goal of video understanding, more practical video understanding systems start emerging in industry. For example, Face Recognition has been widely studied [TP91, LGTB97, WFKM97, CTB92, LFG$^+$01], and some of practical face recognition systems are now commercialized in its potential application area such as video telephony and authentication (identifying an individual) which uses face as a biometric. Another example is video surveillance systems which aim to monitor wide range of the scenes to automatically find out what is

going on in its coverage area. One of the obvious application is a security application. The objective of those security applications is alerting security officers when detecting intruders, or suspicious individuals before they become the criminals. In fact, there already exist a lot of monitoring cameras mounted in stores and banks. In addition, there are already commercial products of home security packages [Sim, SV] composed of PC, cameras, and monitoring software.

While those face recognition systems and security applications are generally categorized into indoor applications, outdoor video surveillance systems which cover wider range of area have also been investigated. Especially, with the increasing need for a more efficient road transporting system that is both economically sound and environmentally preferable, research and development on road traffic monitoring systems are now a great deal of attention. Road traffic monitoring specifically aims to accomplish automatically counting the number of vehicles passing by, detecting traffic accidents, and recognizing vehicles' number-plate, for example. To archive those specific goals, the road traffic monitoring systems primitively involve the collection of data that describe the appearances of vehicles and their motion. Those data are then combined for the higher level processing, such as congestion and incident detections.

Current road traffic monitoring systems rely on the technology of spot sensors based on loop detectors or microwaves. In contrast to those spot sensors, vision sensor is potentially much more powerful than those spot sensors currently available. In the economical aspect, installation of video cameras for road monitoring becomes cheaper compared to installing other spot sensors densely, because one video camera can cover much wider range of areas. Another advantage is that vision sensor can gather plentiful variety of information from the scene as an image sequence, while the spot sensor can only detect whether or not a vehicle exists there. From the fertile information given as an input image sequence, the road traffic monitoring techniques are expected to extract useful traffic information automatically.

One of the major difficulties in outdoor video surveillance techniques is proper handling of variation of the scene illumination and the dynamic scene structure. As for the case of indoor scenes, more specifically in the dark room, we can rely on the preassumptions that illumination do not vary and objects out of interest do not move. However, methods developed under such preassumptions will not work in the outdoor scenes. Suppose we are going to develop a method to detect vehicles from the input image sequence of outdoor traffic scene. Trees lining a street are waving, that can be mis-detected as vehicles, and illumination suddenly changes, that can spoil the detection algorithm itself. To handle the dynamic background[1], several methods to absorb those changes have been proposed [KvB90, Kil92, SG99, FR97].

Compared to those methods handle the dynamic background, the less amount of study has been done to tackle the dynamic variation of scene illumination. In the area related to handling illumination changes, illumination invariants are well studied mostly in the color society. Especially, *Color Constancy* has been widely studied [WB82, FHH01, BF97, FFB95, RBK98] which gives the ability to a vision system by assigning a color description to an object that does not depend on the illumination environment. It allows the system to recognize objects under many different illumination conditions. Color constancy refers to the lack of change in the perceived color of a colored patch as the global illumination changes. Another strand of research has focused on photometric invariants under different types of illumination conditions, e.g. using Reflectance Ratio [NB93, NB96] as illumination invariants, choosing object features which is invariant to illumination changes, etc.

[1]Here, we refer to the background the area out of interest.

1.2 Thesis overview

In this dissertation, we focus on deriving illumination invariant images using *intrinsic images*[2] to enhance video surveillance technologies in the outdoor scene. By deriving illumination invariant images, we believe many computer vision methods which had been only applicable in indoor scenes can be brought to outdoor scenes. We start with estimating intrinsic images in Chapter 2. We formulate the problem of decoupling a set of reflectance images and the corresponding set of illumination images from an input image sequence as a problem of edge classification in derivative domain. After introducing an existing effective method [Wei01], we propose an approach to consider time-varying component of reflectance values to enhance the estimates of scene illumination images.

There are several applications which we consider our intrinsic images are capable of enhancing their output. In Chapter 3, we introduce three application areas where our intrinsic images is considered to be effectively applicable. The first one is shadow removal for robust video surveillance described in Section 3.1. We propose an approach to use illumination images to generate illumination invariant image sequence to reduce appearance variation of the moving objects, which potentially is expected to solve the problems caused by dynamic illumination and large static cast shadows. Unlike previously proposed shadow removal algorithms, our method does not explicitly treat cast shadow regions but use illumination image. Secondly, scene texture editing using the reflectance images is proposed in Section 3.2. Reflectance image essentially is an image containing only the reflectance properties of the scene, and the illumination image is composed of the illumination effects. Thus, if we want to change the texture of the object in the scene in a 2-D image, it is preferable to accomplish editing using reflectance images because the reflectance image is free from the illumination effects such as cast shadow on the ground, shading on the object surface, etc. After editing scene

[2]Detailed introduction of intrinsic images is found in Section 2.1.

textures in reflectance images, the final image is obtained by multiplying illumination images. The final one is the application to *image segmentation* described in Section 3.3. We found the reflectance image is also effective in enhancing image segmentation. Most image segmentation algorithms aim to put boundaries among objects ignoring illumination effects. In this sense, illumination effects such as cast shadows and shading are nothing but obstacles against image segmentation algorithms. Thus, we have investigated the potential of using reflectance images for image segmentation, and confirmed its effectiveness with some experiments.

It has been a difficult problem for a long time to estimate the scene appearance under unsampled intermediate illumination conditions. The problem can be formed like: given several photographs captured under the different illumination conditions, can we produce an image under the novel illumination condition? We simplified the problem to be reproduction of intermediate illumination images, but not arbitrary illumination images. Since the illumination effects observed in retinal images varies non-linearly, simple interpolation methods such as the linear intensity interpolation would not work. In Chapter 4, we propose two different methods to accomplish the non-linear interpolation of illumination images to estimate intermediate illumination images. One type of method uses *Shadow Hull*s to compute intermediate cast shadow shape from sampled illumination images associated with the sunlight angles. Using the estimated intermediate shadow shape, the intermediate illumination image is then computed. Details of this method is described in Section 4.1. The other type of method described in Section 4.2 uses roughly estimated scene geometry using a stereo algorithm to compute geometry-based cast shadows. The geometry-based shadows are not correct because the estimated scene geometry is not very accurate. However, they are still useful to indicate global shadow motion. Thus, we use the geometrically-based shadow as a guide to compute general shadow distortion between sampled illumination conditions.

Chapter 5 describes our approach to apply our illumination normalization method to outside video surveillance systems, which is the final goal of this work. Our focus here is to enhance the accuracy of object tracking by removing cast shadows from the input image sequences. We applied illumination normalization method described in Section 3.3 to a set of images captured from a fixed view point but under the different illumination conditions. In addition, we employed eigenspace method to create a database of illumination images which we refer to as *illumination eigenspace*. We utilize the illumination eigenspace in order to accomplish real-time search of similar illumination images, that consequently used to estimate intermediate illumination images. The effectiveness of our method is confirmed with experimental results on vehicle tracking in the urban scene.

The concluding chapter, Chapter 6, summarizes the contributions of the thesis in more detail and discuss a number of directions for future research.

CHAPTER 2

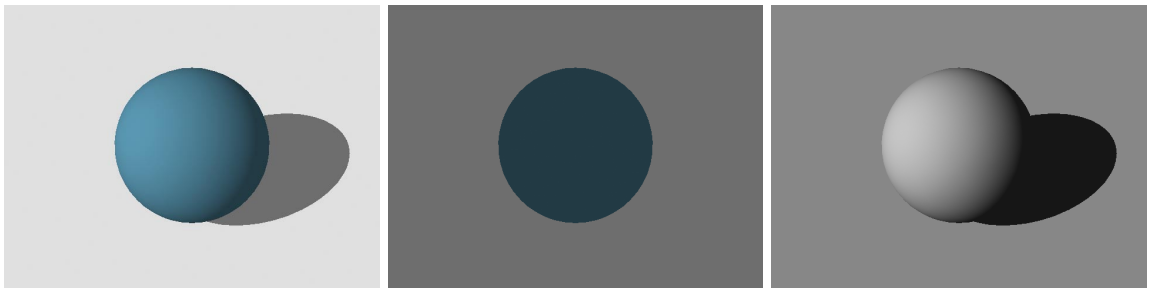
INTRINSIC IMAGES

This chapter first introduces the notion of *Intrinsic Images* in Section 2.1, then summarizes the related works on estimating intrinsic images in Section 2.2. After mathematical preparation for image filtering and reconstruction described in Section 2.3, we formulate the decomposition into intrinsic images as the edge classification problem in Section 2.4. Section 2.5 describes recently proposed ML estimation framework to derive intrinsic images, and subsequently we propose our approach to enhance the estimates obtained using ML estimation method in Section 2.6.

2.1 Introduction

The notion of *intrinsic images* was first proposed by Barrow and Tenenbaum [BT78] in 1978. They proposed to consider every retinal image as a composition of a set of latent images, i.e. images containing only the reflectance properties, illumination or depth of the scene. One type of the intrinsic images, a reflectance image, contains the reflectance values of the scene, while the other one type, an illumination image, contains the illumination intensities, and they are multiplied to produce a single retinal image. First, let us make some terminologies clear.

- ***Luminance*** L : the amount of visible light that comes to the eye from a surface



(a) (b) (c)

Figure 2.1. Intrinsic images. A luminance image (a) is composed of a reflectance image (b) and an illumination image (c).

- ***Illuminance*** E : the amount of incident lighting onto a surface per unit area
- ***Reflectance*** R : the proportion of incident light that is reflected from a surface

A luminance image is a retinal image of the scene that eyes can capture. As shown in Figure 2.1, a luminance image (a) is supposed to be decomposed into a reflectance image (b) and an illumination image (c). Without considering atmospheric attenuation, the relationship among those three can be described by the following Equation (2.1).

$$L = E \cdot R \tag{2.1}$$

It is worth denoting that the illuminance E is a function of incident lighting S and surface normal N of the scene, i.e. E is the product of $\vec{S}$ and $\vec{N}$. Because illuminance is measured by per unit area, it is proportional to cosine of the angle between the surface normal and the direction of incident lighting when the light type is directional. One special case is that by forming $E(x, y, t) = \vec{S}(t) \cdot \vec{N}(x, y)$ with an assumption that the scene is composed of Lambertian surfaces, the photometric stereo algorithm [Woo78, Hay94] estimates both $S(t)$ and $N(x, y)$ by singular value decomposition (SVD) technique.

2.2 Related work on estimating intrinsic images

As we immediately notice, Equation (2.1) is essentially ill-posed, and it is hard to decouple E and R without any prior knowledge about the scene, illumination or their statistics. While decomposing a single image into the intrinsic images, namely a reflectance image and an illumination image, remains a difficult problem [BT78, AP96, Lan77], deriving intrinsic images from a set of images has seen great success. Recently, Weiss developed an ML estimation framework [Wei01] to estimate a single reflectance image and multiple illumination images from a series of images captured from a fixed view point but under significant variation of lighting condition. Weiss's method to derive intrinsic images is useful for largely diffuse scenes, however, it has a problem when applied to scenes containing non-Lambertian surfaces. Weiss's method assumes a single reflectance value constant along the time-axis which implicitly presuppose the scene is composed of Lambertian surfaces. This assumption is inevitable from the assumption of the single reflectance image which has to be independent from illumination changes. For real world scene, we can't expect the assumption to hold. A typical example is white lines and traffic signs on the road surface, which show variable reflection with respect to illumination changes. To be more precise, because Weiss's method relies on the statistics, dense sampling of illumination conditions and properly unbiased sampling are required even for the Lambertian scenes. Finlayson *et al.* [FHD02] proposed a method which derives the scene texture edges from the lighting-invariant image, and by subtracting those edges from the raw input images, they successfully derive shadow-free images of the scene.

2.3 Preparation: Image filtering and reconstruction

When we are going to manipulate images in derivative domain, it is necessary to restore those modified images from derivative domain. In this section, we describe the mathematical preparation for image filtering and image reconstruction.

Suppose we have a filter h and apply it to an image f. Filtering operation using a 2-dimensional filter $h(x, y)$ applying to an input image $f(x, y)$ can be described as:

$$\begin{aligned} g(x,y) &= f(x,y) \otimes h(x,y) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')h(x'-x,y'-y)dx'dy' \end{aligned} \tag{2.2}$$

where $\otimes$ represents convolution[1] and $g(x, y)$ is the filtered image. In frequency domain, Equation (2.2) changes to the following Equation (2.3).

$$G(u,v) = F(u,v) \cdot H(u,v) \tag{2.3}$$

Upper scale letters in Equation (2.3) indicates corresponding variables in frequency domain. They have the relationship as followings:

$$\begin{aligned} g &\overset{FT}{\Longleftrightarrow} G \\ f &\overset{FT}{\Longleftrightarrow} F \\ h &\overset{FT}{\Longleftrightarrow} H \end{aligned} \tag{2.4}$$

where $\overset{FT}{\Longleftrightarrow}$ represents the Fourier transform $\mathcal{F}$, and inverse Fourier transform $\mathcal{F}^{-1}$. The definitions of the Fourier transform and the inverse Fourier transform respectively are:

$$F(u,v) = \mathcal{F}\{f(x,y)\}$$

[1] See Appendix A.

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi j(ux+vy)}dxdy \tag{2.5}$$

$$\begin{aligned} f(x,y) &= \mathcal{F}^{-1}\{f(x,y)\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{2\pi j(ux+vy)}dudv \end{aligned} \tag{2.6}$$

In practice, functions are sampled at equally spaced discrete points. The discrete Fourier transform and the discrete inverse Fourier transform respectively are:

$$F(u,v)_n = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-2\pi j(ux/M+vy/N)} \tag{2.7}$$

and

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{2\pi j(ux/M+vy/N)} \tag{2.8}$$

for an $M \times N$ grid in x and y. Usually, we set $M = N$ to make those equations more convenient, then Equation (2.7) (2.8) change to

$$F(u,v)_n = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-2\pi j(ux+vy)/N} \tag{2.9}$$

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{2\pi j(ux+vy)/N} \tag{2.10}$$

Let us go back to Equation (2.2). Given the filtered image $g(x,y)$ and the filter $h(x,y)$, we can derive the original image $f(x,y)$ as followings. Such reconstruction procedure is necessary for recovering images from the derivative domain after edge-based manipulation of the images.

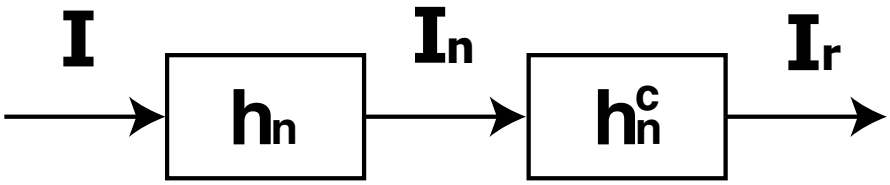


Figure 2.2. Process of image filtering and reconstruction.

$$f(x,y) = g(x,y) \otimes \mathcal{F}[H^{-1}(u,v)] \tag{2.11}$$

where $\mathcal{F}[h] = H$ and equally $\mathcal{F}[H^{-1}] \overset{FT}{\Longleftrightarrow} H^{-1}$. This reconstruction process is described by deconvolution. If we write $a(x,y) = \mathcal{F}[H^{-1}(u,v)]$, a is such that satisfies the following equation.

$$a \otimes h = \delta \overset{FT}{\Longleftrightarrow} A \cdot H = 1 \tag{2.12}$$

We look back the basis of image filtering and reconstruction in this section. Image filtering and reconstruction problem is well studied in the field of image and signal processing [GW87, Jai89, Mit98, Ant93, Str93].

2.4 Time-series analysis for estimating intrinsic images

Estimating intrinsic images from a single image is difficult without a strong pre-knowledges, because the visual rays are the convolution of reflectance and illuminance, namely the decomposition problem is the inverse problem. Estimation using multiple images can be less difficult because there is some change of relying on time-series statistics or physics-based models of temporal intensity variations. In this section, we first propose to treat the decomposition problem as the edge

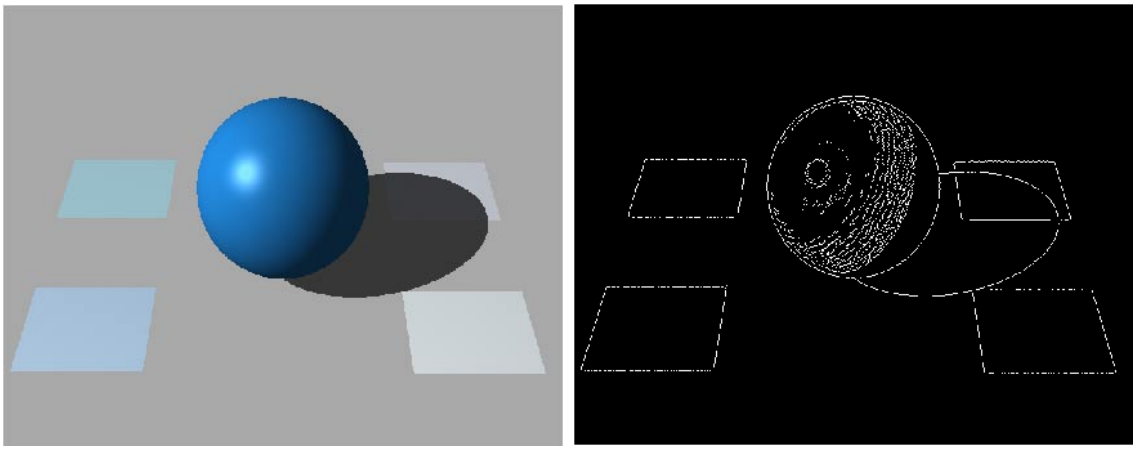
classification problem. After introducing ML estimation framework to derive intrinsic images, we enhance the method for practical use by deriving time-varying reflectance images in Section 2.6.

2.4.1 Problem formulation

We formulate the problem of estimating intrinsic images as the problem of edge classification of a derivative image sequence. Our interest is determining two of the intrinsic images, i.e. illumination images E and reflectance images R from a set of luminance images L captured from a fixed camera position but each under the different illumination condition. In derivative domain, we assume that every retinal image is composed of the mixture of various types of edges categorized into the following five types. In some cases, the edge is referred to the binarized line segment after thresholded, but we regard an edge as a real derivative value between a point and the neighboring point.

- Cast shadow boundary ... An edge appeared on the boundary of a cast shadow.
- Shading ... Reflected intensity is dependent of surface orientation.
- Highlights and Specularity ... Highlight on the object's surface shows the high brightness.
- Occluding boundary ... Boundary of structurally different objects.
- Pigment boundary ... Almost same as the occluding boundary. Boundary of different pigments, but structurally on the same plane.

Figure 2.3 shows a sample luminance image (a) and its derivative image (b). The derivative image (b) is generated by summing up the absolute values of horizontally and vertically derivative-filtered outputs. Figure 2.4 is the same image as Figure 2.3(b) but with the explanation of edge types.



(a) (b)

Figure 2.3. A retinal image (a) and its derivative image (b).

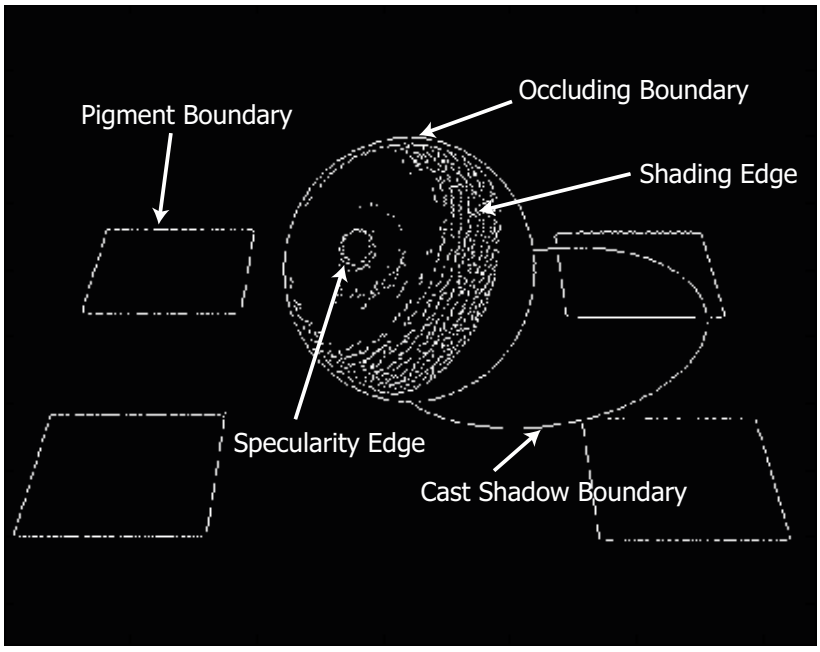


Figure 2.4. Categorization of edge types in the derivative image.

Our interest is to decouple the illumination component and the reflectance component from an input image sequence. We propose to use edge-based manipulation in derivative domain to accomplish this. This manipulation in spatial-derivative domain is quite useful because it originally holds the intensity relationship among neighboring pixels and the computational cost becomes lower because of its sparseness [HM99].

We consider that a problem of the decomposition into intrinsic images is the categorization problem of edges in filtered input images. In derivative domain, a reflectance image contains the following edges.

- Highlights and Specularity
- Occluding boundary
- Pigment boundary

On the other hand, an illumination image contains the following edges in derivative domain.

- Cast shadow boundary
- Shading
- Occluding boundary

In this way, we formulate the decoupling problem as the edge discrimination problem.

2.5 ML Estimation for determining reflectance edges

Recently, Weiss [Wei01] proposed an ML estimation framework to derive intrinsic images by formulating the problem as follows.

$$L(x, y, t) = R_w(x, y) * E_w(x, y, t) \tag{2.13}$$

where $*$ is the pixel-wise multiplication. Here we denote Weiss's reflectance image R_w and illumination image E_w with small w attached. In log domain, Equation (2.13) changes into the following equation.

$$l(x, y, t) = r_w(x, y) + e_w(x, y, t) \tag{2.14}$$

where $(l, r_w, e_w) = (\log L, \log R_w, \log E_w)$. With n-th derivative filter f_n, the method derives the filtered reflectance image r_{wn} by taking median along the time-axis as follows.

$$\hat{r}_{wn}(x, y) = \text{median}_t\{l(x, y, t) \otimes f_n\} \tag{2.15}$$

where $\hat{r}_{wn}$ is the estimate of the filtered reflectance image. Finally, the illumination images e_w are computed in unfiltered domain.

$$\hat{e}_w(x, y, t) = l(x, y, t) - \hat{r}_w(x, y) \tag{2.16}$$

2.6 Deriving Time-varying Reflectance Images

Weiss's method which is described in Section 2.5 to derive intrinsic images is useful for largely diffuse scenes, however, it has a problem when applied to scenes containing non-Lambertian surfaces. Since his method implicitly assumes the scene is composed of Lambertian surfaces, and this assumption is inevitable from the definition of the reflectance image which has to be independent from illumination changes. For real world scene, we can't expect the assumption to hold. A typical example is white lines on the road surface, which show variable reflection with respect to illumination changes. Therefore, while the time invariant reflectance image $R_w(x, y)$ derived by Weiss's framework reasonably describes the scene texture without lighting effects, the estimated illumination images $E(x, y, t)$ tend to contain considerable amount of scene texture. Those scene textures should

not really be a component of the "illumination" image, since illumination images should represent the distribution of incident lighting per unit area. These annoying scene textures in illumination images arise at scene regions where surfaces of different reflectance properties meet. Therefore it is necessary to assume a set of time-varying reflectance images $R(x, y, t)$ instead of a single one.

Our estimation method is based on Weiss's method. We first estimate Weiss's reflectance image to use it as a scene texture image. Again, we denote Weiss's reflectance image and illumination image with small w attached, i.e. R_w and E_w, and our reflectance image and illumination image, R and E respectively. First, we apply the ML estimation method to the image sequence to derive a single reflectance image $R_w(x, y)$, and a set of illumination images $E_w(x, y, t)$. Our goal is to derive time-varying, i.e. lighting condition dependent, reflectance images $R(x, y, t)$ and corresponding illumination images $E(x, y, t)$ that do not contain scene texture as written as the following equation.

$$L(x, y, t) = R(x, y, t) * E(x, y, t) \tag{2.17}$$

where $*$ is the pixel-wise multiplication.

Equation (2.17) changes into the following equation in log domain.

$$l(x, y, t) = r(x, y, t) + e(x, y, t) \tag{2.18}$$

We use lower-case letters to denote variables in log domain, e.g. r represents the logarithm of R. With n-th derivative filters f_n, a filtered reflectance image r_{wn} is computed by taking median along the time axis of $f_n \otimes i(x, y, t)$. We used two derivative filters, i.e. $f_0 = [0\ \ 1\ \ -1]$ and $f_1 = [0\ \ 1\ \ -1]^T$. With those filters, input images are decomposed into intrinsic images by Weiss's method as described in Equation (2.19). The method is based on the statistics of natural images [HM99].

$$\hat{r}_{wn}(x,y) = \text{median}_t\{f_n \otimes l(x,y,t)\} \tag{2.19}$$

The filtered illumination images $e_{wn}(x,y,t)$ are then computed by using estimated filtered reflectance image r_{wn}.

$$\hat{e}_{wn}(x,y,t) = f_n \otimes l(x,y,t) - \hat{r}_{wn}(x,y) \tag{2.20}$$

To be precise, e is computed by $e = l - r$ in the unfiltered domain in Weiss's original work while we estimate e in the derivative domain for the following edge-based manipulation.

We use the output of Weiss's method as initial values of our intrinsic image estimation. As mentioned above, the goal of our method is to derive time-dependent reflectance images $R(x,y,t)$ and their corresponding illumination images $E(x,y,t)$. The basic idea of the method is to estimate time-varying reflectance components by canceling the scene texture from Weiss's illumination images. To factor the scene textures out from the illumination images and associate them with reflectance images, we use the texture edges ofr_w. We take a straightforward way to remove texture edges from $e_w$and derive illumination images $e(x,y,t)$ with the following Equation (2.21) (2.22).

$$e_n(x,y,t) = \begin{cases} 0 & \text{if}|r_{wn}(x,y)| > T \\ e_{wn}(x,y,t) & \text{otherwise} \end{cases} \tag{2.21}$$

$$r_n(x,y,t) = \begin{cases} r_{wn}(x,y) + e_{wn}(x,y,t) & \text{if}|r_{wn}| > T \\ r_{wn}(x,y) & \text{otherwise} \end{cases} \tag{2.22}$$

where T represents a threshold value. While we currently manually set the threshold value T used to detect texture edges in r_{wn}, we found the procedure is not so sensitive to the threshold as long as it covers texture edges well. Since the operation is linear, the following equation is immediately confirmed.

$$\begin{aligned} f_n \otimes l(x,y,t) &= r_{wn}(x,y) + e_{wn}(x,y,t) \\ &= r_n(x,y,t) + e_n(x,y,t) \end{aligned} \tag{2.23}$$

Finally, time-varying reflectance images $r(x,y,t)$ and scene texture-free illumination images $e(x,y,t)$ are recovered from filtered reflectance images r_n and illumination images e_n through the following deconvolution process, which is same as described in Weiss's paper.

$$(\hat{r},\hat{e}) = g \otimes \Big(\sum_n f_n^r \otimes (\hat{r}_n,\hat{e}_n)\Big) \tag{2.24}$$

where f_n^r is the reversed filter of f_n, and g is the filter which satisfies the following equation.

$$g \otimes \Big(\sum_n f_n^r \otimes f_n\Big) = \delta \tag{2.25}$$

The pseudo code of the algorithm is as shown in Figure 2.6.

To demonstrate the effectiveness of our method for deriving time-dependent intrinsic images, we prepared a CG scene which contains cast shadows and surface patches with different reflectance properties, which is analogous to real road

```
DERIVE r_wn(x, y) AND l_wn BY WEISS'S METHOD
  FOREACH x,y DO
    IF |r_wn(x, y)| > THRESHOLD THEN
      e_n(x, y, t) = 0
      r_n(x, y, t) = r_wn(x, y) + e_wn(x, y, t)
    ELSE
      e_n(x, y, t) = e_wn(x, y)
      r_n(x, y, t) = r_wm(x, y)
    END IF
  END
RECOVER UNFILTERED IMAGES
```

Figure 2.5. Pseudo-code of deriving time-varying reflectance images

surfaces, e.g. white lines on a pedestrian crossing. Figure 2.6 shows a side-by-side comparison of the results applying Weiss's method and our method. The first two columns are the CG scenes, where each scene has the property that the histogram of derivative-filtered output is sparse, which is the required property of the ML estimation based decomposition method and also is the statistics usually found in natural images [HM99]. The last column is the real world scene. As can be seen clearly, texture edges are successfully removed from our illumination image while they obviously remain in Weiss's illumination image in each result. Considering an illumination image to be an image which represents the distribution of incident lighting, our illumination image is much better since incident lighting has nothing to do with the scene reflectance properties.

The proposed method is practical but contains important limitation. That is our method can only applied to the largely planar scenes, since it cannot discriminate occluding boundaries and pigment boundaries. However, it is still useful for those scenes where traffic monitoring systems are required because they mostly are planar on road surfaces.

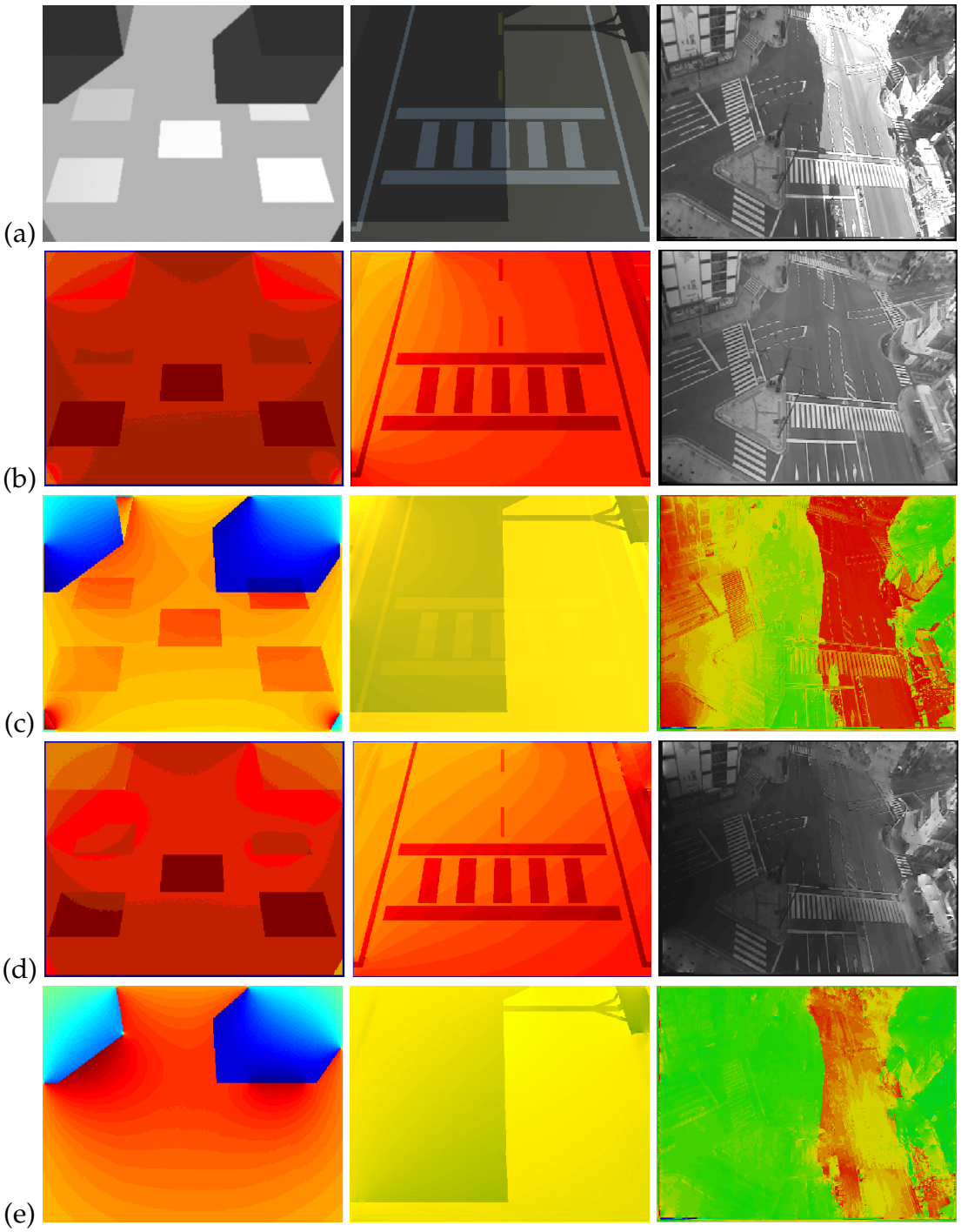


Figure 2.6. (a) an input image $i(x, y, t)$, (b) Weiss's reflectance image $r_w(x, y)$, (c) Weiss's illumination image $l_w(x, y, t)$, (d) our time-varying reflectance image $r(x, y, t)$, (e) our illumination image $l(x, y, t)$.

2.7 Summary and Future Work

Estimating intrinsic images has been a difficult problem since it essentially is an ill-posed problem. Although human beings immediately and unconsciously distinguish illumination effects from reflectance properties, it is difficult for machines to do this. Because the observed intensities are the convolution of incident lighting and the surface reflectance property, computationally it is the inverse problem. As we described in Section 2.4.1, solving the inverse problem naturally comes down to the edge discrimination problem in derivative domain. Our future work is to give a solution to this problem with large coverage of the properties of the scene.

In this chapter, we first introduced the notion of intrinsic images. Followed by mathematical preparation for image filtering and unfiltering, Weiss's ML estimation framework for deriving intrinsic images is introduced. Subsequently, we propose a method to improve Weiss's result with the notion of time-varying reflectance images. Edge-based manipulation is done to derive time-varying reflectance images, and we confirmed the improved illumination images and corresponding time-varying reflectance images are robustly derived. Though we haven't finished the estimation of intrinsic images starting from the problem formulation described in Section 2.4.1 yet, the work is on-going and we believe the edge-based classification yields one solution.

CHAPTER 3

ILLUMINATION NORMALIZATION USING INTRINSIC IMAGES

Representation of intrinsic images is useful to analyze and manipulate the scene illumination and reflectance properties, once a luminance image is decomposed into a reflectance image and an illumination image. We can think of a lot of avenues of applications using the intrinsic images with taking its advantage of efficient representation.

In this chapter, we first show the efficiency of shadow removal using illumination images in Section 3.1. This shadow removal technique is used in our real-time illumination-normalization framework described in Chapter 5, and more detailed evaluation is also found there.

We also describe a method utilizing intrinsic images to accomplish scene texture editing in Section 3.2. Modifying scene texture with preserving scene illumination has been a difficult and tough labor for CG creators because the scene texture and scene illumination are not separated in ordinary 2-D images. However, it becomes quite simple using intrinsic images because the scene texture and scene illumination are essentially separated in the representation of intrinsic images. We show the simplicity of the texture editing using real world images.

Finally, in Section 3.3, we propose an approach to use the reflectance image for image segmentation. The objective of image segmentation is generally to discrim-

inate foreground objects from the background, or to separate respective objects from each other. For those objectives, illumination effects such as cast shadows and shading on the scene surface are obstructive for the most of image segmentation algorithms. We consider the reflectance image is the ideal input to the most image segmentation algorithms, since it is free from confusing scene illumination effects. We confirmed that the better results are obtained using reflectance images with three different image segmentation algorithms. Though the performance of image segmentation algorithms cannot be measured by a single criterion, but our objective here is to decrease the number of erroneous subimage regions caused by shading and cast shadows.

3.1 Shadow removal for video surveillance systems

Video surveillance systems involving object detection and tracking require robustness against illumination changes caused by variation of, for instance, weather conditions. Annoying obstacles include not only the change of illumination conditions, but also the large shadows cast by surrounding structures such as large buildings and tall trees. Since most visual tracking algorithms rely on the appearance of the target object, typically using color, texture and feature points as cues, these shadowing effects degrade the accuracy of object tracking. In urban scenes where building robust traffic monitoring systems is of special interest, it is usual to have large shadows cast by tall buildings surrounding the road. Building a robust video surveillance system under such an environment is a challenging task. To make the system insensitive to dramatic changes of illumination conditions and robust against large static cast shadows, it would be valuable to cancel out those illumination effects from the input image sequence.

This section describes our method to "normalize" an input image sequence of traffic scene in terms of the distribution of incident lighting to remove illumination effects including shadowing effects. We should note that our method does not

cope with shadows cast by moving objects but those cast by static objects such as buildings and trees.

Using the scene illumination images obtained by our method described in Section 2.4, the input image sequence can be normalized in terms of illumination. To estimate the intrinsic images of the scene where video surveillance systems are to be applied, it is necessary to remove moving objects from the input image sequence because our method requires the scene to be static. Therefore we first create background images in each short time period (ΔT) in the input image sequence, assuming that the scene illumination does not vary in that short time period. We simply use the average image of the short input sequence as the background image, but of course the more complicated methods would give the better background images [TKBM99]. These background images $\mathbf{B}(x, y, t)$ are then used for the estimation of intrinsic images. Using the estimation method described in the former section, each image in the background image sequence is decomposed into corresponding reflectance images $\mathbf{R}(x, y, t)$ and illumination images $\mathbf{E}(x, y, t)$.

$$\mathbf{B}(x, y, t) = \mathbf{R}(x, y, t) * \mathbf{E}(x, y, t) \tag{3.1}$$

where $*$ is the pixel-wise multiplication.

Once decomposed into intrinsic images, any image whose illumination condition is captured in the series of $\mathbf{B}(x, y, t)$ can be normalized with regards to its illumination condition by simply dividing the input image $\mathbf{L}(x, y, t)$ by its corresponding estimated illumination images $\mathbf{E}(x, y, t)$. Through this normalization, cast shadows are also removed from the input images.

Since the incident lighting effect is fully captured in illumination images $\mathbf{E}(x, y, t)$, the normalization by dividing with $\mathbf{E}$ corresponds to removing the incident lighting effect from the input image sequence. Let us denote the resulting illuminant-invariant images $\mathbf{N}(x, y, t)$ that can be derived by the following equation.

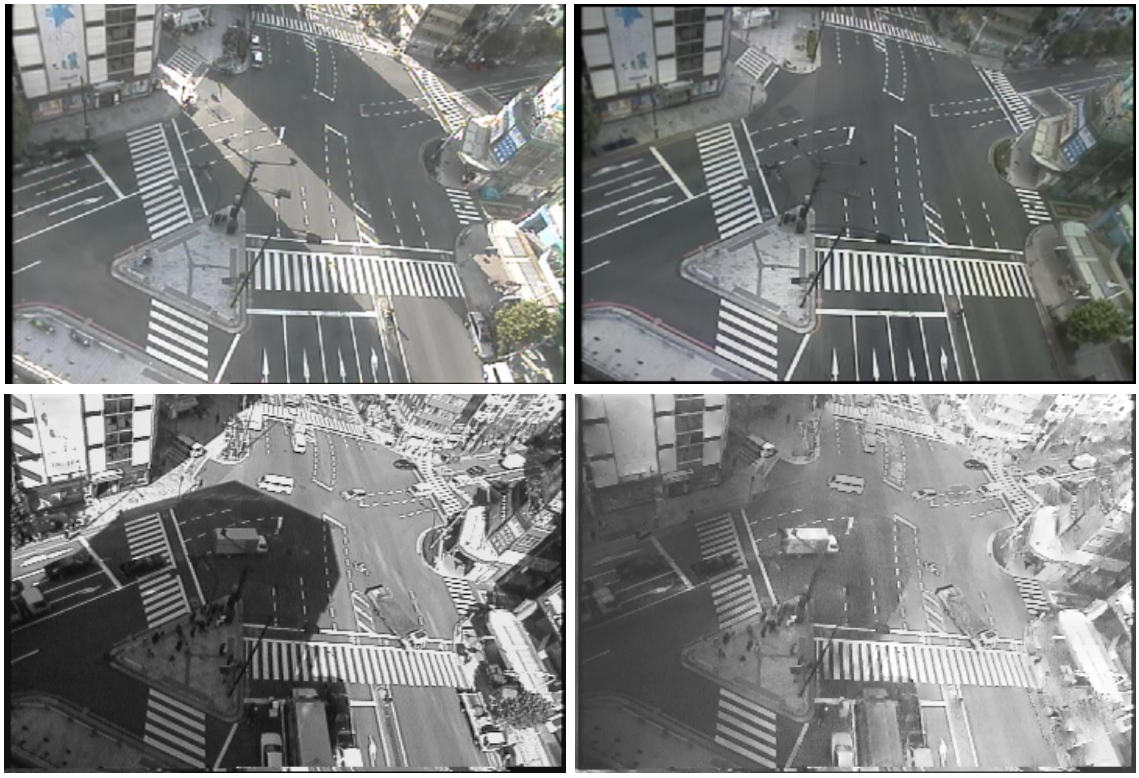


Figure 3.1. An input image L (left of each pair) and the illuminance-invariant image N (right or each pair).

$$\mathbf{N}(x, y, t) = \mathbf{L}(x, y, t) \oslash \mathbf{E}(x, y, t) \tag{3.2}$$

where $\oslash$ is the pixel-wise division.

Figure 3.1 shows the results of our normalization method. The left-hand side figure shows the input image $\mathbf{L}$ and the right figure represents the illuminance-invariant image $\mathbf{N}$. Notice that shadows of the buildings are removed and shadow boundaries become seamless in $\mathbf{N}$. We can see the shadowing effect are clearly removed by simple division operation once we obtain the corresponding illumination image.

The limitation of this method is that, when the scene structure changes dramatically, this method bears an unreasonable result since the illumination image $\mathbf{E}$ is thoroughly associated with the scene structure. This implies that the output value on the image around the moving object is locally not correct because the scene structure has changed around it. However, globally it gives the correct output in terms of illumination normalization.

3.2 Scene Texture Editing

One of the most difficult problem in image editing is preserving *consistency of illumination* through the image manipulation. Suppose we have two photographs taken under the different illumination conditions and want to move an object (actually it would be a subimage region) from one photo to another. Since the illumination condition is not parameterized in those two photos, adjusting the illumination condition when superimposing is hard to be accomplished. As can be seen from this example, preserving the consistency of scene illuminations throughout the photo editing has been a tough task when the illumination conditions are not parameterized.

Our method permits scene texture manipulation without considering the *true* distribution of illumination of the scene nor the scene geometry. It can modify texture of the scene, e.g. modifying wall paper of a room in the image. Our scene texture modification goes along the following steps.

1. **Estimation step :** Estimate intrinsic images of the scene using our method described in Chapter 2.

2. **Modification step :** Make a modification on scene texture in the reflectance image $\mathbf{R}$.

3. **Rendering step :** Take a product of the modified reflectance image $\mathbf{R}$ and illumination image $\mathbf{E}$ to get the final result.

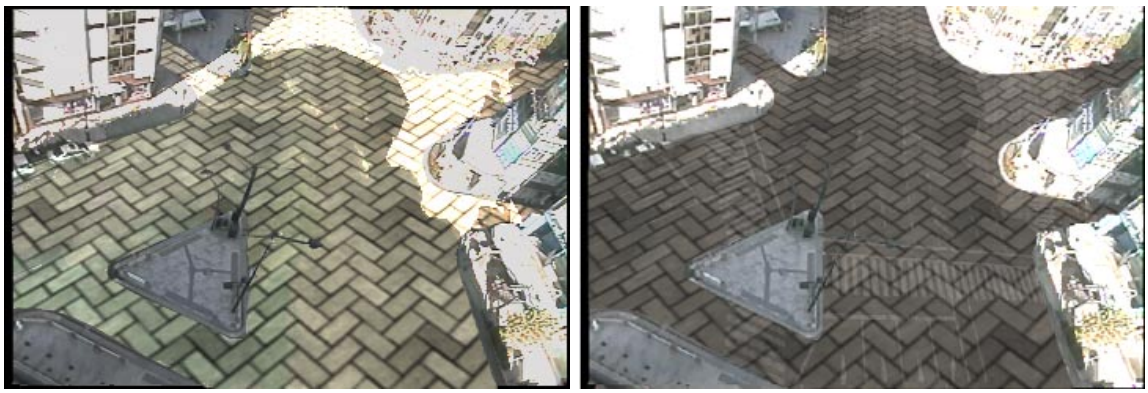
Since the reflectance image can be regarded as the scene-texture image which is free from illumination effects, the ideal editing of the scene texture is enabled using the reflectance image. During the texture modification, we don't need to take into account the illumination effects such as cast shadow. After the texture modification, the final result is obtained by taking the product of the edited reflectance image and the corresponding illumination image.

(a) (b) (c)

Figure 3.2. Decomposition into intrinsic images. (a) An original image, (b) the reflectance image, (c) the corresponding illumination image.

(a) (b)

Figure 3.3. Scene texture manipulation using the reflectance image. (a) Edited reflectance image, (b) brick texture used for the modification.



(a) (b)

Figure 3.4. Final result of scene texture modification. (a) our method, (b) resulting image with alpha blending.

We tested our scene texture editing approach to the cross road scene. Figure 3.2 shows estimation step, a single sample from the input image sequence (a) and resulting intrinsic images, i.e. the reflectance image (b) and the corresponding illumination image (c). In the estimation step, the input image sequence is decomposed into a set of reflectance images and illumination images using the method described in Chapter 2. Though only one input image is displayed in Figure 3.2, several input images under the different illumination conditions are required. Using the obtained reflectance image, the scene texture is edited in modification step. We changed the asphalt texture on the road surface to a brick texture by manually selecting the area. The interim output of the modified reflectance image is shown in Figure 3.3 (a), and (b) is the brick texture used for the modification. Finally, at rendering step, the modified reflectance image is multiplied together with the corresponding illumination image to produce the final result. Figure 3.4 depicts the final rendered result (a) in comparison with the resulting image with ordinary alpha blending (b). Notice that shadowed regions and well lighted regions are properly preserved with regard to their shape and contrast in the result of our method (a), while the resulting image (b) using alpha blending is much less natural.

Modification of the scene texture without intrinsic images is a tough task. Because the graphics designer first has to select shadowed area to create a shadow mask by clicking, and choose different alpha values associated with lit areas and shadowed areas to determine the blending function. By using intrinsic images, such terrible tasks are simplified to just multiplying the corresponding illumination images. Our method is fully in 2-D image domain, and since we don't have the scene geometry, the method cannot archive 3-D object insertion.

For the scene texture editing, another key point is that the reflectance image used for texture modification contains variation of the reflectance properties. The difference between using a constant reflectance image and the time-varying reflectance image is well observed when we apply texture modification over the pigment boundary. Figure 3.5 depicts the comparison of the scene texture editing us-

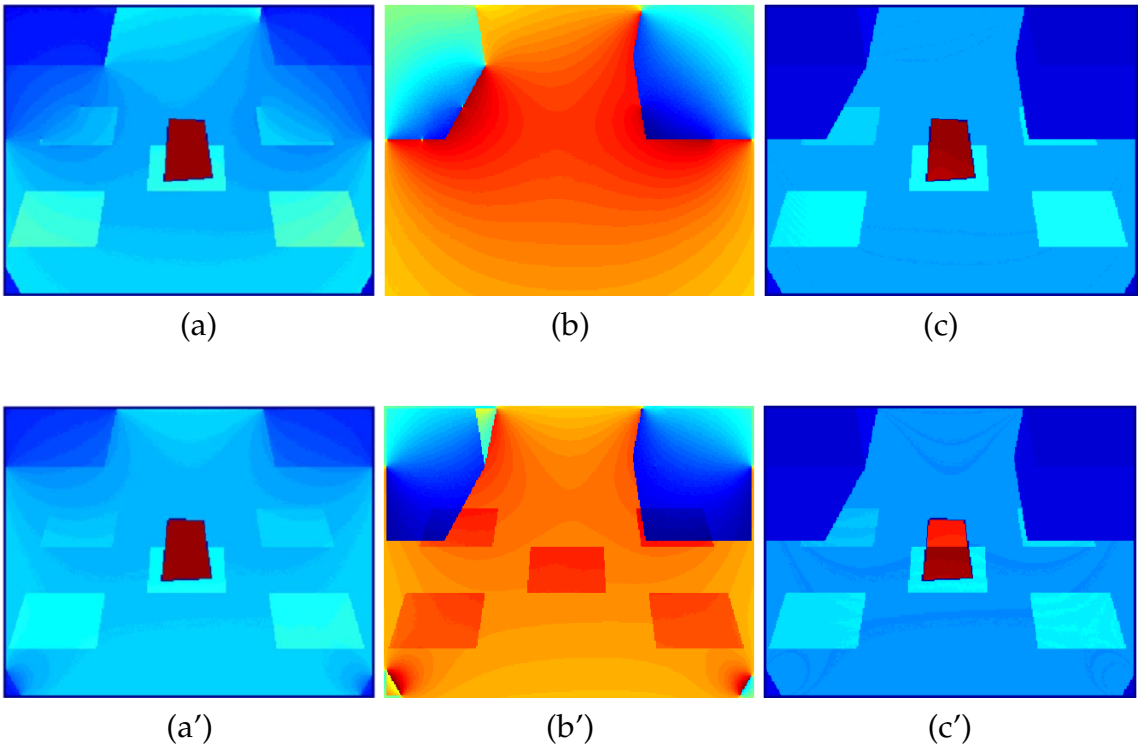


Figure 3.5. Texture manipulation. (a), (b) and (c) show the modified reflectance image, corresponding illumination image, and the resulting image respectively using our method. (a′), (b′) and (c′) show the same but using Weiss's method.

ing reflectance image derived by Weiss's method [Wei01] with our reflectance image. Results using our method and Weiss's method are shown in the first row and the second row respectively. From left to right, (a) reflectance image after modification, (b) corresponding illumination image and (c) final rendered result obtained by taking product of (a) and (b). The trapezoid appeared in (a) is the modified area replacing the original image by simple uniform colored texture. Looking at the final result (c′), we notice that an undesirable effect, i.e. a horizontal edge on the modified texture, is appeared in the final result. The edge is obviously produced by the illumination image (b′), since the illumination image contains the pigment boundary which primarily should not be included in the illumination

images. On the other hand, the undesirable edge is clearly removed using our method as shown in Figure 3.5 (c). This is because our time-varying reflectance image correctly absorbs the variation of reflectance properties, and as a result, the corresponding illumination image (b) becomes more accurate compared to (b').

3.3 Enhancing image segmentation

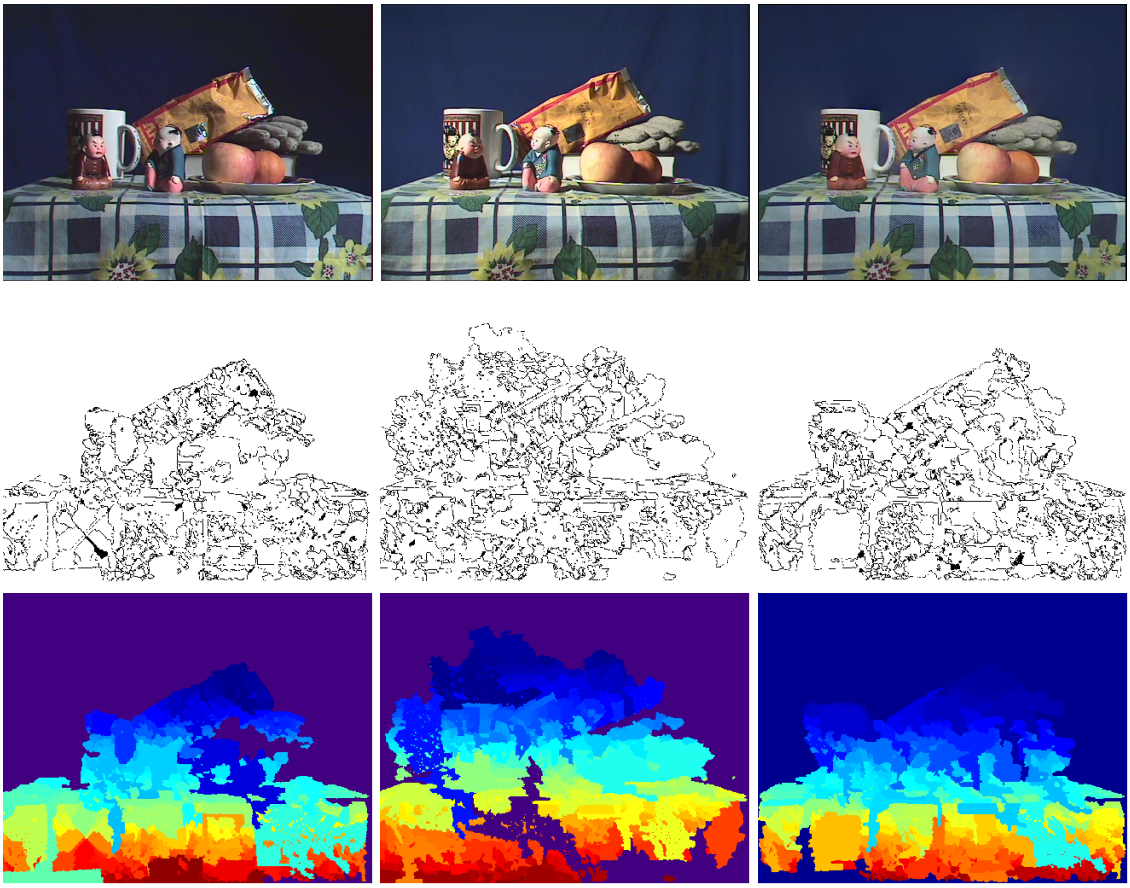
A technique that is used to find the area of interest is usually referred to as a segmentation technique, i.e. segmenting the foreground from the background or distinguishing objects from each other. In this section, we propose to use reflectance images as input to segmentation algorithms. Since the reflectance image is essentially illumination-free image, it is expected to reduce the number of undesirable subimage regions caused by illumination effects such as shadowing effects when the reflectance image is used as input. This section is composed of three parts. In each part, we introduce a major technique of image segmentation and evaluate the improvement of the segmentation result using our method.

[Thresholding-based method]

The thresholding method itself is too simple and cannot be used as an effective segmentation algorithm by itself, however, lots of algorithms are based on the thresholding method in fact. In this part we first introduce the idea of image thresholding and compare the output of using normal input images and using our reflectance images as input. Image thresholding is based upon the simplest idea. Parameters T_1, T_2 called *threshold*s are chosen and then applied to an image $\mathbf{I}$ as follows.

$$\begin{aligned} \mathbf{I}(x,y) &= \text{the object} \quad if \;\; T_1 > \mathbf{I}(x,y) > T_2 \\ &= \text{others} \qquad \text{otherwise} \end{aligned} \tag{3.3}$$

The output is usually labeled as a boolean variable '0' or '1' to indicate 'object' or 'the others'. A widely used method of thresholding, which is particularly useful for thresholding based on several features, is the method of *Clustering* [JMB91].



(a) (b) (c)

Figure 3.6. Results of image segmentation using watershed algorithm.

Once clustering is applied , a complete segmentation of an image $\mathbf{I}$ is a finite set of subregions $\{\mathbf{I}_1, ..., \mathbf{I}_N\}$.

$$\mathbf{I} = \cup_{n=1}^{N} \mathbf{I}_n \qquad \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \quad \text{if} \quad i \neq j \tag{3.4}$$

In the gray scale images, we frequently use the histogram technique to find the threshold. The threshold value may be global. But in most cases, a global threshold is not the best threshold to segmentation. Then we need the local self-adapting

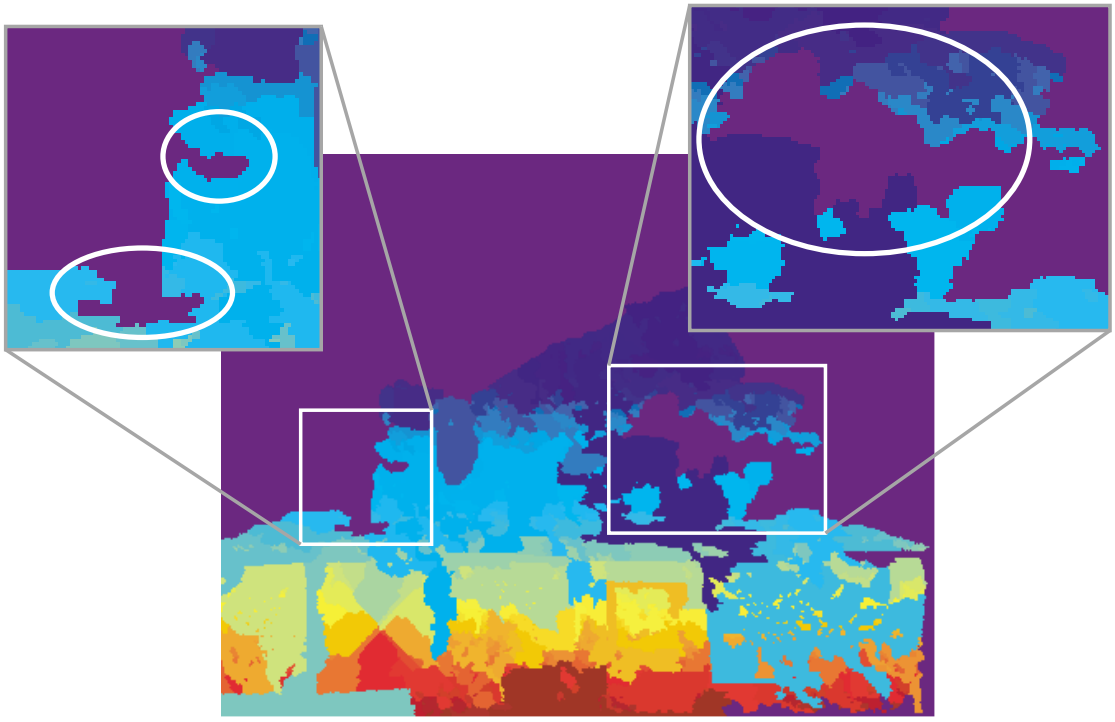


Figure 3.7. Erosion caused by shading and shadowing effects.

threshold. So far, lots of methods choosing locally self-adaptive threshold has been proposed. Salembier *et al.* proposed an approach based on area morphology operators [SS95]. The area morphology operators generate scaled images based on the area of connected components within the image level sets, i.e. thresholded versions of the gray scale image. In the context of scale space, Maragos [Mar89] presents a multi-scale shape description with morphological filters. One of the most widely used algorithm is *Watershed algorithm* [GP93, Vin91, Vin93, Mey93, BM93] which is also widely used in the area of mathematical morphology. In the framework of the watershed transformation, the image gradient magnitude is considered as a topographic surface which is formed by surrounding ridges. The areas surrounded by those ridges are the watersheds, that share the local minimum in gradient magnitude. In this way, the watersheds define a segmentation of the image.

One of the annoying factors complicating thresholding-based methods is shading effect caused by illumination. With shading, a single object shows appearance variation on its non-planar surface, which makes the problem difficult when choosing the threshold value. In this scene, using reflectance image for the thresholding based image segmentation is expected to give the better result since it does not contain shading nor shadowing effects.

We applied the watershed algorithm to several sets of images, a test set is composed of a reflectance image of the scene and the same scene under several illumination conditions. The result is shown in Figure 3.6. In the figure, (a) and (b) are the results of different illumination samples and (c) is the result obtained by using the reflectance image. Along the row from top to bottom, the input image, ridges computed by watershed algorithm and the output obtained by filling areas surrounded by the ridges. Since we used the most basic watershed algorithm which does not contain rich post-processing, the resulting images are not very sufficient throughout the data set. The output shown in Figure 3.6 is rather the intermediate output for succeeding post processes such as merging and splitting. However, we can see the effectiveness of using reflectance image in respect of reducing undesirable effects caused by shading and shadowing. Figure 3.7 is the closeup view of the image in the bottom of Figure 3.6 (a). We can notice the eroded parts in the figure, and those areas are then connected to the background. The eroded parts actually are the dark area with effect of shading and shadowing as can be seen in the original input image in Figure 3.6 (a). This is the typical undesirable effect caused by shading and shadowing, however, it is well reduced when reflectance images are used.

[Edge finding-based method]

Edge detection is one of the most common approach in discontinuity segmentation as well as thresholding. Edge detection is usually followed by edge relaxation

and edge following to archive image segmentation, since the image resulting from edge detection cannot be used as the image segmentation result by itself. Supplementary processing steps must follow to combine edges into edge chains that correspond better with borders in the image. The final aim is to reach at least a partial segmentation, i.e. to group local edges into an image where only edge chains with a correspondence to existing objects or image parts are present.

- Edge detection ... Finding edges by the mentioned edge detecting operators. The detection of discontinuity of an image is accomplished by using gradient operators. The gradient ∇ is the first order derivative operator described as follows.

$$\nabla \mathbf{I} = \begin{bmatrix} \frac{\delta \mathbf{I}}{\delta x} \\ \frac{\delta \mathbf{I}}{\delta y} \end{bmatrix} = \frac{\delta \mathbf{I}}{\delta x} + \frac{\delta \mathbf{I}}{\delta y} \tag{3.5}$$

The second order derivative operator is the Laplacian operator ∇^2 which is given by

$$\nabla^2 \mathbf{I} = \begin{bmatrix} \frac{\delta^2 \mathbf{I}}{\delta x^2} \\ \frac{\delta^2 \mathbf{I}}{\delta y^2} \end{bmatrix} = \frac{\delta^2 \mathbf{I}}{\delta x^2} + \frac{\delta^2 \mathbf{I}}{\delta y^2} \tag{3.6}$$

Those operators, such as gradient and Laplacian operators, are defined in many ways suitable for each applications. For example, the simplest gradient operator can be defined as:

$$\frac{\delta}{\delta x} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \ , \ \frac{\delta}{\delta y} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T \tag{3.7}$$

- Edge relaxation ... Process of considering edge properties in the context of their neighborings to increase the quality of the edge image. This process is required, since the resulting edge image is often strongly affected by noise which causes surplus edges or missing of the important edges. Edge relaxation considers not only the magnitude of the edges and adjacently but also the context of edges.

- Edge linking ... Linking adjacent edge pixels by seeing if they have the similar properties. One of the most common discrimination function is like :

$$\Big|\|\nabla f(x_1, y_1)\| - \|\nabla f(x_2, y_2)\|\Big| \leq T_m \tag{3.8}$$

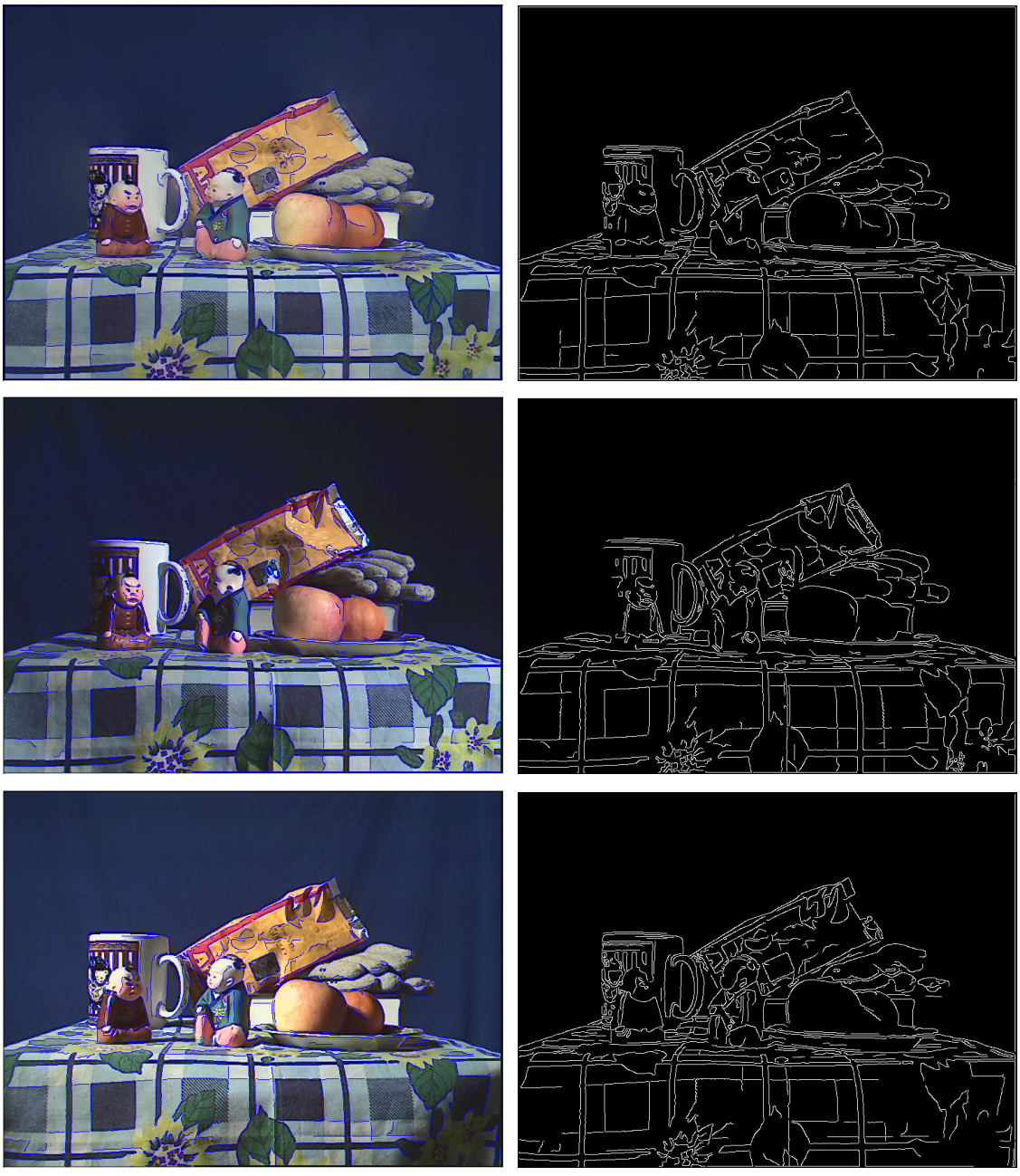
for some magnitude different threshold T_m to combine edges have similar magnitude properties. As for the edge orientation, the discrimination function can be described as :

$$\Big|\phi(\nabla f(x_1, y_1)) - \phi(\nabla f(x_2, y_2))\Big| \leq T_a \tag{3.9}$$

where T_a is the angular threshold.

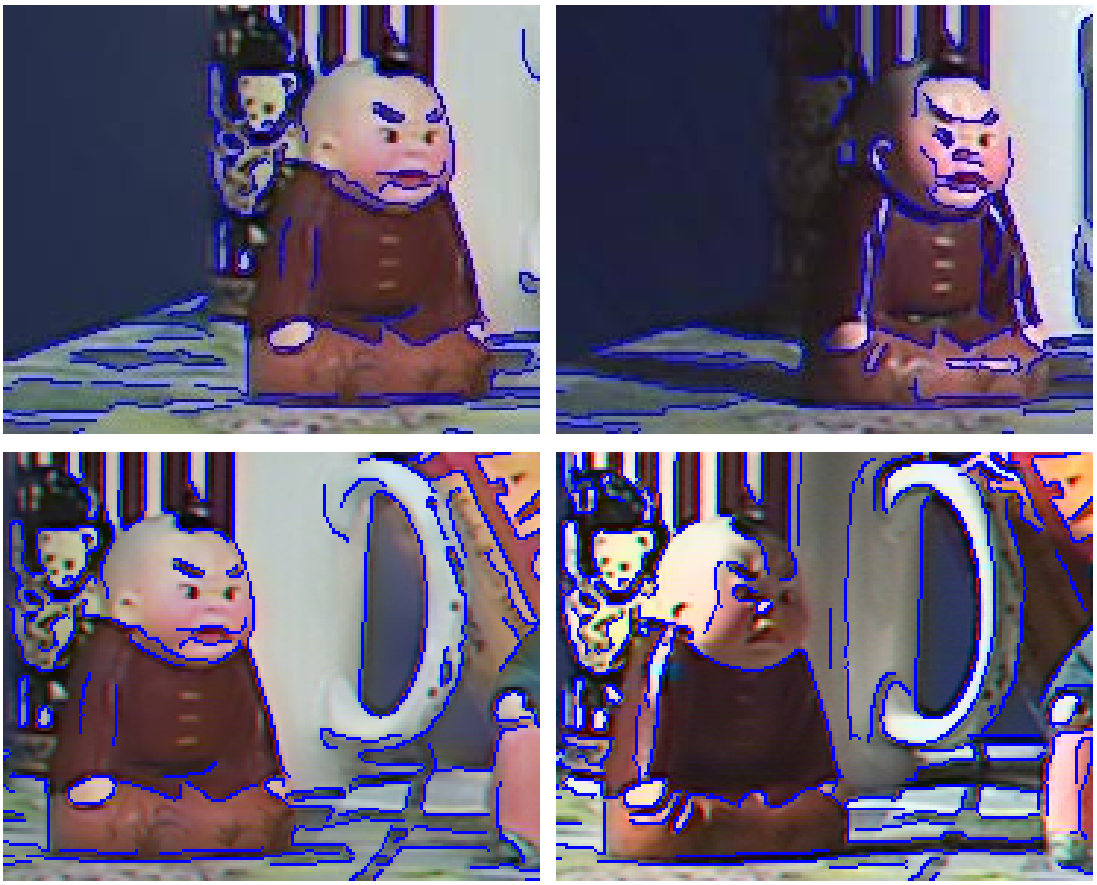
Once we get the linked edges, we can use them as the boundary of the region where we want to segment.

- Region construction from borders ... Region construction is the final step of the edge-based image segmentation. Connected borders are processed to form subimage regions to produce the final resulting image.



(a) (b)

Figure 3.8. Results of edge detection. Edges are overlaid to the original images in blue color (a), and edge mask images (b).



(a) (b)

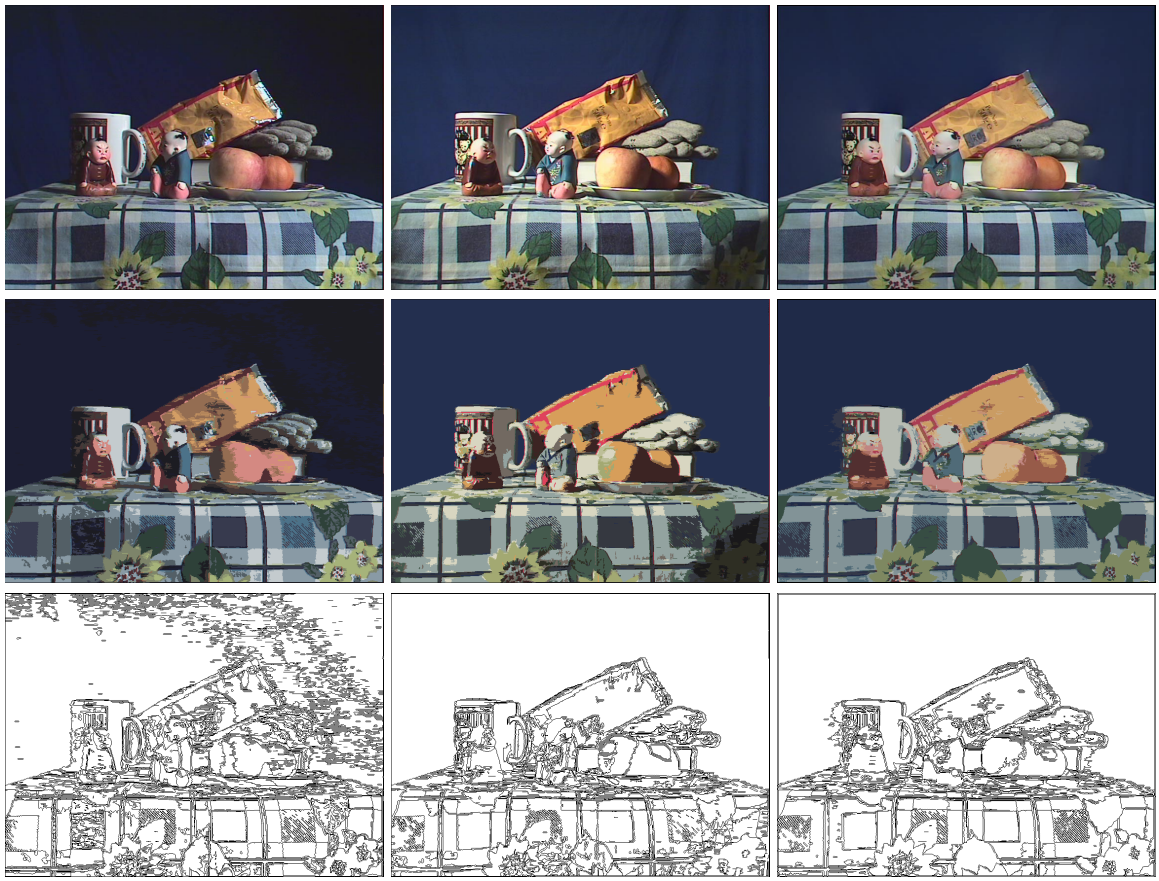
Figure 3.9. Closeup view for comparison of the results of edge detection. (a) Results using the reflectance image, (b) results using images under certain illumination conditions.

As we briefly reviewed the ordinary process of the edge-based image segmentation, the most important step is Edge Detection step because it has the most strong effects on the whole following steps. Since retinal images are textured by shading and cast shadows, those effects give undesirable effects to edge finding-based image segmentation. For those applications which aim to extract objects' boundary, shading effect is nothing but troublesome noise. By using reflectance images, since illumination effects are factored out from them, it is assumed that the more accurate segmentation results are obtained.

In this experiment, we focus on how the edge detection result can be improved using our reflectance images. We used a low-level feature extraction tool named EDISON (Edge Detection and Image SegmentatiON) developed at the Robust Image Understanding Laboratory at Rutgers University [RIU] which implements the work of confidence based edge detection [MG01] presented by Meer *et al.* We applied its edge detection algorithm to the reflectance image obtained by using our method and two images captured under different illumination conditions. Camera parameters when capturing and parameters used for the edge detection algorithm are totally the same. The results are shown in Figure 3.8. From top to bottom, resulting images using the reflectance image and two different illumination samples. In column direction, (a) is the results overlaid with detected edge segments, and the corresponding edge masks are in (b). We can see the detrimental edge segments appeared in the resulting images using original illumination samples (the second and third row), but they are much reduced in the result using our reflectance image (the top row). Let us have a closer look at those images. Figure 3.9 is the closeup view of parts of Figure 3.8. Note that though the boundary of cast shadow and shading are detected as edge segments in results using illumination samples (b), those excessive edge segments are clearly reduced in the result using our reflectance image (a).

[Richer method : Mean-Shift Algorithm-based method]

The essential challenge of image segmentation is its ambiguous objective. Suppose we have an image of zebra. Some applications may want to segment out the whole region of zebra, while the others may want to detect the white stripes on zebra. In this way, the objective is application-dependent and cannot be defined for general use. So, is the image segmentation useless? The answer is no. It is still useful for specific applications. Each segmentation algorithm has its own goal, and



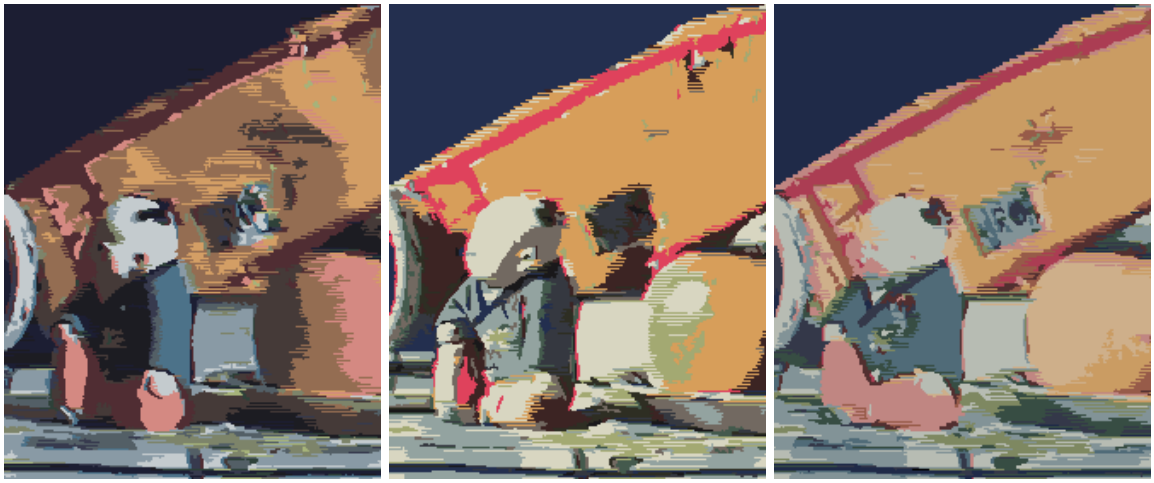
(a) (b) (c)

Figure 3.10. Results of the mean shift algorithm-based image segmentation.

for those specific goals a lot of heuristic algorithms have been presented. They use much richer techniques compared to simple thresholding or edge-based methods.

Here are well known image segmentation methods based on the mean shift algorithm [CF85, Fuk90, Che95, CM97, CM99, CRM00]. The mean shift algorithm basically is a method to find modes of distribution of data represented as arbitrary-dimensional vectors using a non-parametric procedure for estimating density gradients. The algorithm consists of the following steps.

1. Choose a radius for the search window in the vector space.



(a) (b) (c)

Figure 3.11. Closeup view of resulting images of the mean shift algorithm-based image segmentation.

2. Define the initial location of the window.

3. Compute the mean of the data points inside the window and set the center of the window at this point.

4. Repeat Step 3. until the translation distance of the window is converged.

When applied to image segmentation, it starts with mapping the image domain into the feature space. Then, an adequate number of search windows at random locations in the feature space is defined to find centers of high density regions. After that, regions in the image domain corresponding to high density regions in the feature space is obtained. Finally, some post-processing is applied to get the final segmentation results.

Figure 3.10 shows the image segmentation result using Comaniciu's method [CM97]. From left to right, the first and the second column represent input images and resulting images of illumination samples, and the last column corresponds to those of the scene reflectance image. From top to bottom, the original input image, re-

sulting image and the boundary mask image corresponding to the result image. The resulting images are colored in the mean color of the same subimage region for each segments.

Let's have a closer look at the results in Figure 3.11. In Figure 3.11, (a) and (b) are the results using ordinary illumination samples as input, while (c) is the result obtained by using the reflectance image as an input. We can notice that the number of subimage regions caused by illumination effects are well reduced in (c) compared to (a) and (b). Those subimage regions caused by scene illumination observed in (a) and (b) are regarded as the independent image regions though they really are connected to other subimage regions. In this point, (c) shows the more correct result in terms of distinguishing objects because (c) is not affected by the scene illumination.

3.4 Summary

Though the illumination effects make the scene the charm of variety for human beings, the variety is nothing but annoying effect for the most of computer vision algorithms. Our focus in this chapter is enhancing three different applications by handling illumination effects using our intrinsic images.

The first application is shadow removal from input image sequences described in Section 3.1. Our method permits removing shadowing effect from the input image sequence without explicit representation of shadowed regions. Large shadows cast on the road surface has been one of the most troublesome factor which lower the accuracy of moving object detection / tracking in traffic monitoring systems. We believe our shadow removal technique would immediately bring much better results for those systems. In addition, our method can be used as a preprocessing stage for video surveillance algorithms, and this directly means that our method can be integrated into existing video surveillance systems. As for the integration to existing video surveillance systems, we describe our approach in Chapter 5.

Secondly, we presented a method to use intrinsic images for scene texture editing in Section 3.2. Using reflectance images for scene texture modification, we don't need to be careful not to change the scene illumination when editing, since the scene illumination is utterly removed from the reflectance images. This advantage is quite needed in applications such as viewers of products like houses and cars for a user who wants to change the wall or floor texture of the house, for example. For those applications what we have to prepare is only photos from a fixed view point but under the several different illumination conditions. It doesn't require 3-D model of the scene for the editing. After scene texture editing using the reflectance image, the corresponding illumination image is then multiplied to generate the final result.

Finally, in Section 3.3, we investigated the use of reflectance images to improve results of image segmentation. For most of image segmentation algorithms, effects caused by scene illumination such as shading and cast shadows are nothing but harmful to decrease accuracy of the methods because illumination effects basically have no bearing on the scene structure. Illumination gives the scene the additional texture that consequently increase the number of subimage segments. However, using reflectance images, the segmentation algorithms do not suffer from the scene illumination effects and become capable of generating stable and reasonable results.

In this section, we focus on advantages of shadow elimination using illumination images. However, we should denote that there are several applications that take advantage of shadowing effects by contraries. Tzomakas *et al.* for example, model the intensity of the road and shadows under the vehicles and use moving shadows cast by those vehicles to estimate the possible presence of vehicles [TS98]. The method proposed by Stauder *et al.* [SMO99] explicitly detects shadow regions first to enhance object segmentation. We agree the illumination effect can be a clue for the image understanding for some applications, however, for most existing algorithms which do not use the scene illumination effects the scene illumination is

nothing but harmful to decrease the quality of output. We believe our method to handle scene illumination in 2-D images can be applied to many applications in addition to three applications described in this Chapter.

CHAPTER 4

NON-LINEAR INTERPOLATION OF ILLUMINATION IMAGES

Suppose we have relatively sparse sampled illumination images and want to estimate intermediate illumination images among them. The most intuitive and easiest way is the linear image interpolation using nearest neighbor illumination images. But we immediately notice that it doesn't work. Since the linear image interpolation is only an interpolation of intensities of each pixel among frames but the motion of cast shadows and specular reflections cannot be described by this scheme because of their non-linearity. In this chapter, we present two methods to estimate intermediate illumination images by non-linear interpolation.

One method is the interpolation using *Shadow Hull* described in Section 4.1. Shadow hull is analogous to *Visual Hull*, which is composed of the largest possible intersection of shadow volumes that are computed from sampled illumination images and light directions. We use the shadow hull to compute intermediate shadow shapes which cannot be represented by linear interpolation in 2-D images. Finally, we estimate intermediate illumination images using intermediate shadow shapes.

The other method uses rough scene geometry to compute the geometrically-based shadow motion as described in Section 4.2. The cast shadow computed using rough scene geometry is not precise but it gives the general motion of the

shadow. We describe the method to use the geometrically-based shadow motion represented by 2-D affine transformation to compute the shadow distortions. In the same way as the Shadow hull-based approach, intermediate illumination images are then estimated using intermediate shadow shapes computed using shadow distortions.

4.1 Shadow hull-based approach

In this section, we propose an approach to use *Shadow Hull*s for generating intermediate illumination images. We first derive shadow regions from illumination images and associate them to sunlight angles. Using computed shadow regions, shadow volumes are then pitched to construct *shadow hull* which is the largest possible intersection of shadow volumes.

4.1.1 Introduction

To explain the notion of *Shadow Hull*, we first introduce the idea of *Visual Hull*. Visual hull construction, or *Shape from Silhouette*, is a popular method in the topic of shape estimation. A visual hull is a geometric shape constructed using silhouettes of an object as seen from a number of view points. Each view volume from view points pitch a cone-like volume[1]. The intersection of these volumes results in a visual hull. As we add more cameras, the visual hull better approximates the shape of the object because it gives the more information of the shape.

Since the idea of the 'Shape from Silhouette' was first proposed by Baumgart [Bau74], a large amount of work has been done to explore better representations and more efficient algorithms.

For example, Aggarwal *et al.* [KA86, MA83] suggested using voxel as a representation of visual hulls. Almost at the same time, Potmesil [Pot87] used an oc-

[1]If the intrinsic parameter of the camera is approximated by perspective transformation. If it is like orthographic projection matrix, the view volume be like a pillar.

tree data structure to speed up the construction of visual hulls. As for the camera positioning problem, Shanmukh *et al.* [SP91] derived optimal positions and directions to take silhouette images for building 3D volume models. Szeliski built a non-invasive 3D digitizer using a turntable and a single camera with Shape from Silhouette as the reconstruction method [Sze93]. Laurentini studied the theoretical properties of visual hulls of 3D polyhedral objects [Lau94, Lau95] and curved objects [Lau99]. De Bonet and Viola extended the idea of voxel reconstruction to transparent objects by introducing the concept of Roxels [BV99]. Buehler *et al.* used visual hull as the geometric basis for image-based rendering [BMMG99]. Cipolla *et al.* recovered the camera positions and orientations from silhouettes under circular motions [MWC01, WC01]. Ponce *et al.* studied the exact Visual Hull of objects with smooth surfaces. In the past five years, due to advances in extracting silhouette (background subtraction) [EHD99, HHD99, EHD00] from images and video sequences, a large number of researchers have applied Shape from Silhouette to human related applications. In other words, Shape from Silhouette has become a standard and popular method of shape estimation. Estimating shape using Shape from Silhouette has many advantages. Silhouettes are readily and easily obtainable, especially in indoor environment where the cameras are static and the background subtraction is relatively effective. The implementation of most Shape from Silhouette methods is relatively straightforward, especially when compared to other shape estimation methods such as multi-baseline stereo [OK93] or space carving [KS00]. Visual hull constructed from Shape from Silhouette is upper bound of the object of interest. This inherently conservative property is particularly useful in applications such as obstacle avoidance in robot manipulation and visibility analysis in navigation where an upper bound on the shape of the object is preferred to a lower bound.

What we call ***Shadow Hull*** is analogous to Visual Hull, but it is composed of not the intersection of view volumes but that of shadow volumes. In the past, cast shadows have been used to determine the object surface and orientations [KS87,

BP98, SK83]. Our objective is different from them, i.e. to estimate intermediate shadow shapes with not accurate shadow hulls.

4.1.2 Shadow Hull Scenario

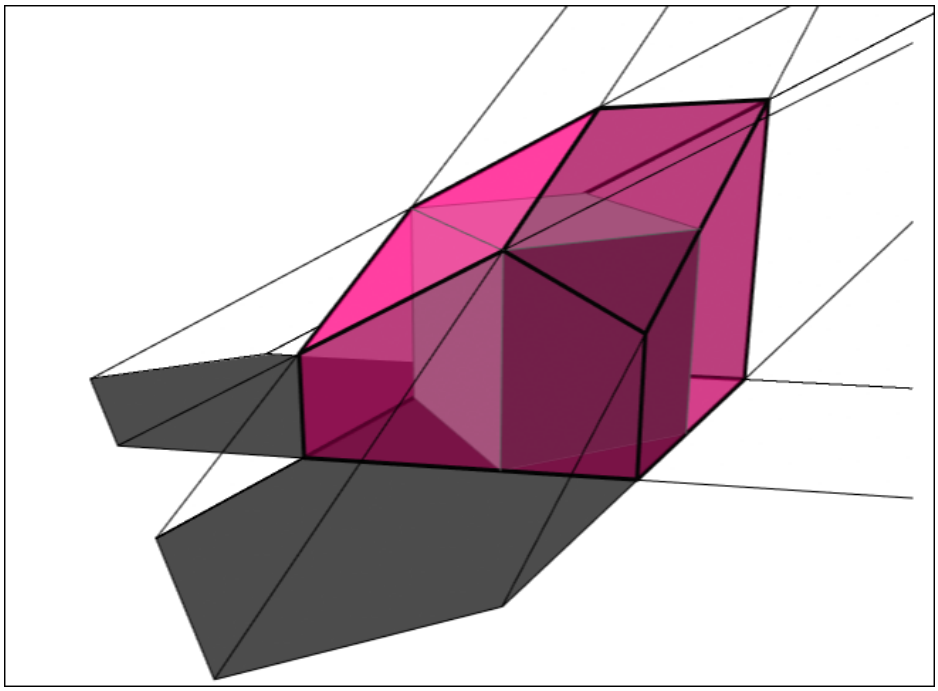


Figure 4.1. Computing shadow hulls using shadow regions associated with parameterized light sources.

Suppose we have N illumination images associated with parameters of the light source, and in those illumination images we have a set of cast shadow regions $\{\mathcal{S}_j^n; n = 1; ..., N\}$ of an object $\mathcal{O}$ with the light source $\mathcal{L}^n$. We can assume $\mathbf{\Psi}^n : \Re^3 \Rightarrow \Re^3$ is the projection function of cast shadows, i.e. $\mathbf{p} = \mathbf{\Psi}^n(\mathbf{P})$ where $\mathbf{P}$ is a point on a shadow -generating surface and$\mathbf{p}$ is the corresponding point on a receiver surface. If we assume the flat receiver surfaces, as is assumed in our experiments over the road scene, $\mathbf{p}$ becomes 2 dimensional and $\mathbf{\Psi}$ changes to $\mathbf{\Psi}^n : \Re^3 \Rightarrow \Re^2$. Here is a notion of *Shadow volume* $\mathcal{V}$. The shadow volume is

constructed from rays cast from the light source, intersecting the vertices of the shadowing object, then terminated at the vertices of the shadowed object. Defined in this way, the shadow volume is a pyramid /pillar when the light source is a point/directional light source. Thus the following equation is derived from the definition.

$$\mathbf{\Psi}^n(\mathcal{V}^n) = \mathcal{S}_j^n \tag{4.1}$$

Given a set of N cast shadow regions $\{\mathcal{S}_j^n\}$ and projection functions $\{\mathbf{\Psi}^n\}$, we immediately have shadow volumes $\{\mathcal{V}^n\}$ by

$$\mathcal{V}^n = \mathbf{\Psi}^{n-1}(\mathcal{S}_j^n) \tag{4.2}$$

Here, our interest is a volume $\mathcal{H}$ which satisfies

$$\mathbf{\Psi}^n(\mathcal{H}) = \mathcal{S}_j^n \quad \textit{for all} \quad n \in \{1, ..., N\} \tag{4.3}$$

If there exists at least one entity which satisfies Equation (4.3), we say the $\{\mathcal{S}_j^n\}$and projection functions $\{\mathbf{\Psi}^n\}$ are consistent. Otherwise, they are inconsistent. Thus the shadow hull is defined as followings.

Definition of Shadow Hull :

The shadow hull $\mathcal{H}_j$ of a set of consistent cast shadow regions $\{\mathcal{S}_j^n\}$ and projection functions $\{\mathbf{\Psi}^n\}$ is defined by the largest possible intersection volume of the volumes $\{\mathcal{V}^n\}$which satisfies Equation (4.3).

4.1.3 Computing shadow hulls

Practically, to compute a shadow hull in the outdoor scene, it is necessary to be provided the following conditions.

1. The camera is fixed and the camera parameters are known.
   Not necessarily all the camera parameters should be known. If we can assume the scene to be planar, only a plane-to-plane projection matrix, which is a projection matrix from image plane to the scene plane, is required. As for the case of plane-to-plane projection, corresponding points between the image plane and the real world are related by

$$\lambda \mathbf{X} = \mathbf{H}\mathbf{x}$$

$$\lambda \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.4}$$

   where λ is a scaling factor and $\mathbf{H}$ is a 3×3 matrix. The other notation used is that points on the world coordinate are represented by an upper case vector, $\mathbf{X}$, and their corresponding points in the image plane are denoted by a lower case vector, $\mathbf{x}$. In the case of the plane-to-plane projection, the camera model is completely determined once the matrix $\mathbf{H}$ is known. To determine projection matrix $\mathbf{H}$, at least 4 pairs of corresponding points are needed. Equation (4.4) can be rewritten as the following Equation (4.5).

$$\mathbf{A}\mathbf{h} = \mathbf{b} \tag{4.5}$$

   where

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -X_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1x_1 & -Y_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -X_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2x_2 & -Y_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -X_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3x_3 & -Y_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -X_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4x_4 & -Y_4y_4 \end{pmatrix} \tag{4.6}$$

$$\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T \tag{4.7}$$

$$\mathbf{b} = (X_1, Y_1, X_2, Y_2, X_3, Y_3, X_4, Y_4)^T \tag{4.8}$$

By choosing correspondences as $\mathbf{A}^{-1}$ could exist, $\mathbf{h}$ and projection matrix $\mathbf{H}$ are determined.

2. A number of images are captured, each under the different illumination conditions.

3. The geometry of shadow receiver is roughly known.

4. Sunlight angles are associated with the captured images.

We assume global intensity changes are linear as long as they are densely sampled, but the motion of cast shadows cannot be represented by linear image interpolation. Thus, we create shadow hulls from given shadow regions, that are derived from illumination images, and sunlight angles computed from time stamps

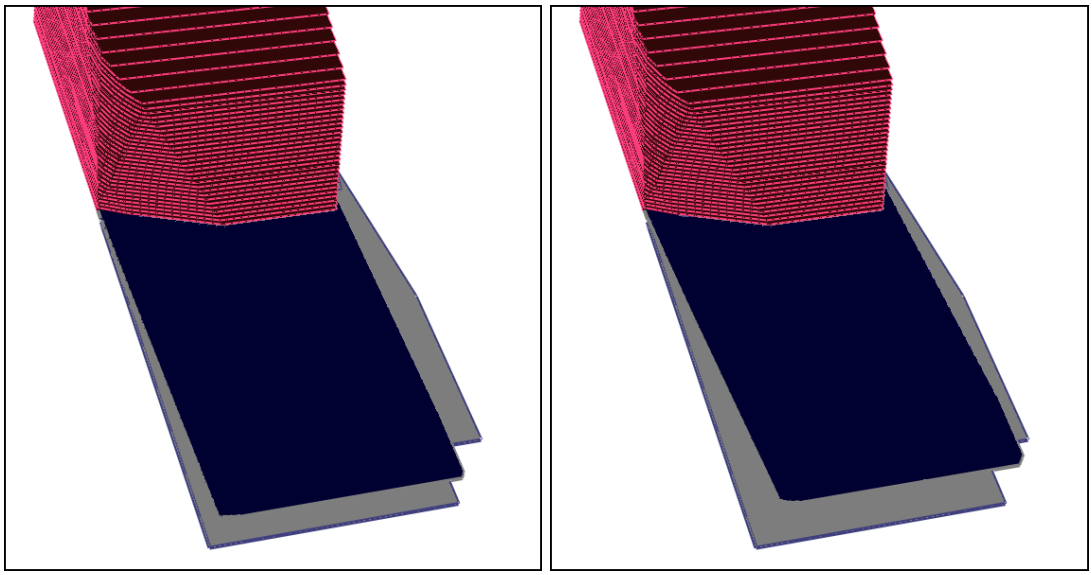


Figure 4.2. Result of interpolating cast shadow using a shadow hull.

of the input image sequence [2]. The resulting hull is not necessarily precise, but it gives *enough* information to compute intermediate cast shadow shapes between sampled illumination conditions.

In our approach, input images are decomposed into intrinsic images first. By thresholding, shadow regions are derived from the illumination images. We assume the intrinsic parameters of the camera is estimated beforehand. Shadows are cast on a plane in the real world and a projection matrix from the image plane to this scene plane can be computed by manually providing several correspondences between the two planes. Shadow regions are then mapped onto the world coordinate, and shadow volumes are computed using shadow regions associated with sunlight angles.

By taking the intersection of shadow volumes in the 3D space, we get the rough geometry of the objects casting the shadow, which has enough information for computing intermediate cast shadow (Figure 4.2 (a)). Figure 4.2 (b) shows the

[2]Sunlight angles can be computed precisely provided the latitude and longitude of the scene and the date and time.

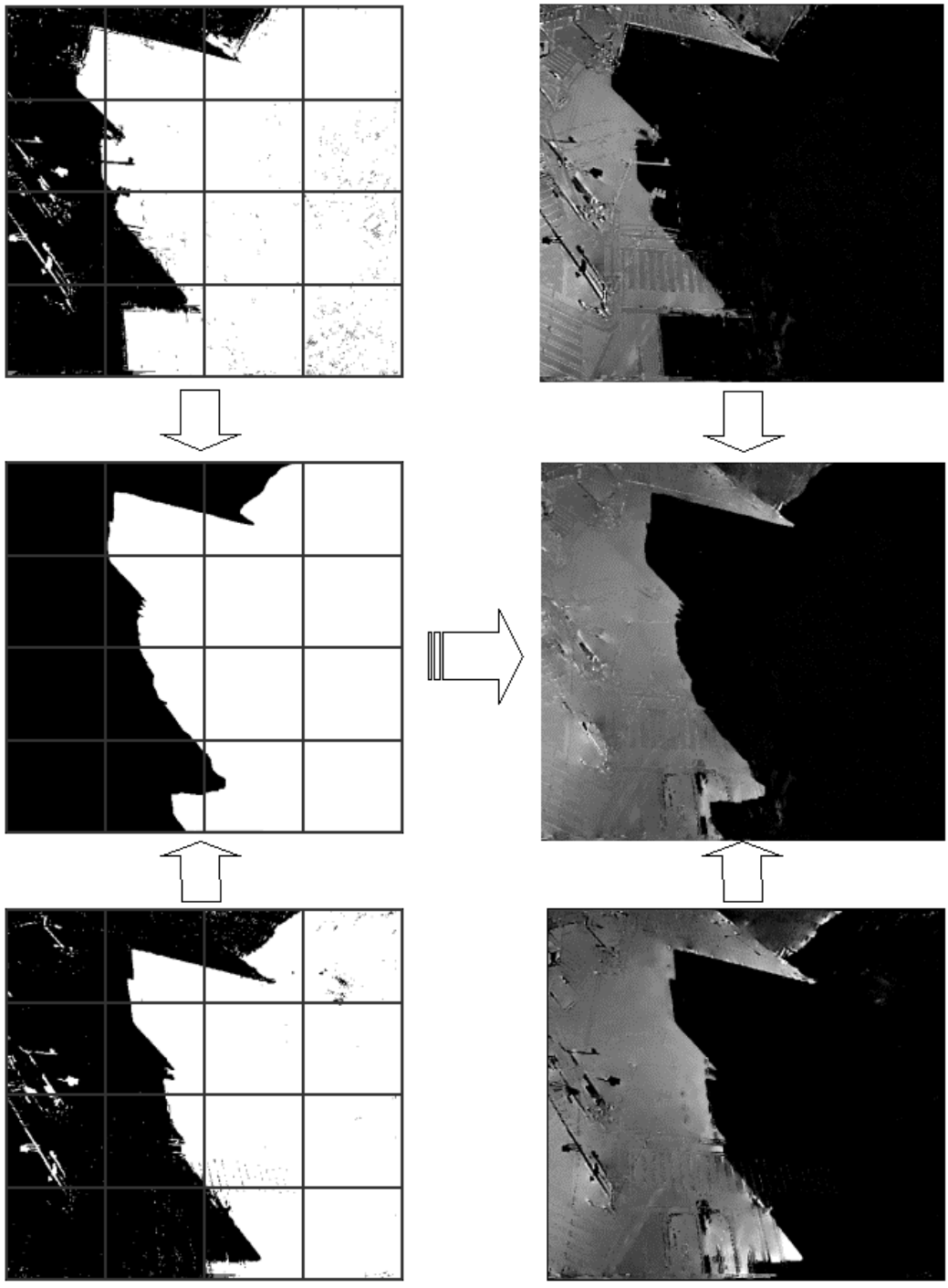


Figure 4.3. Shadow hull based shadow interpolation. Figures in top and bottom row are shadow regions and sampled illumination images. The middle row shows the interpolated results. The grid is overlaid for better visualization.

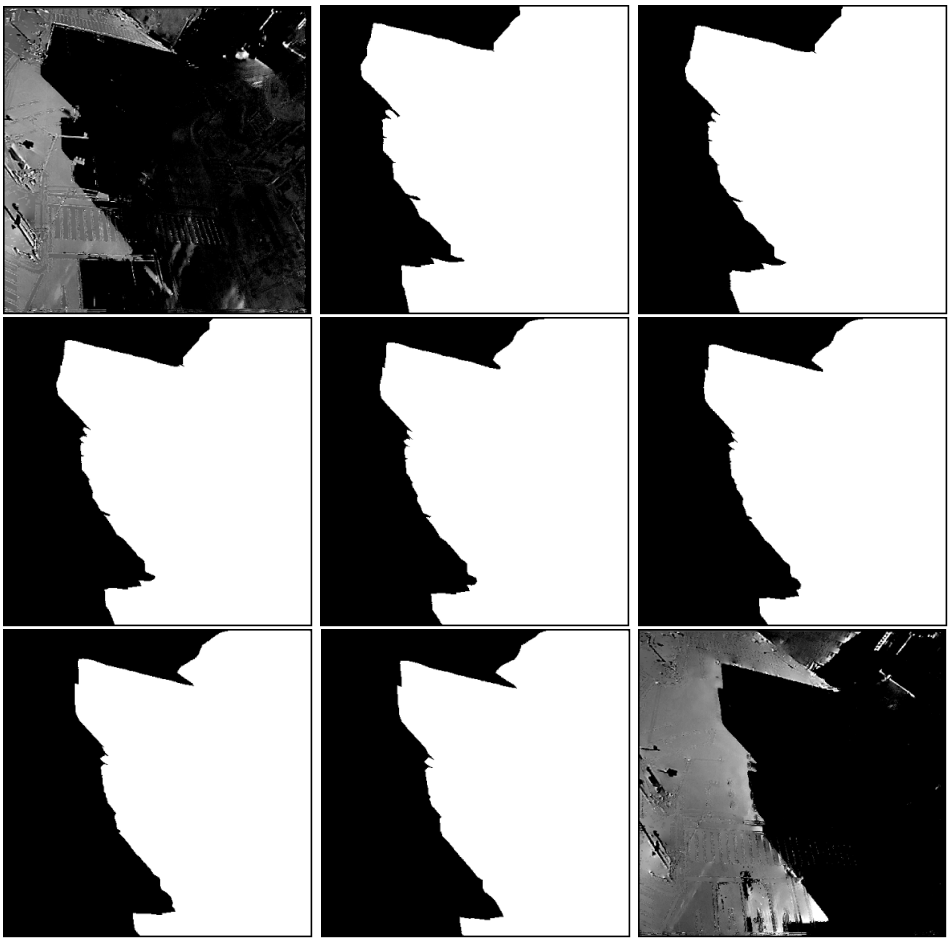
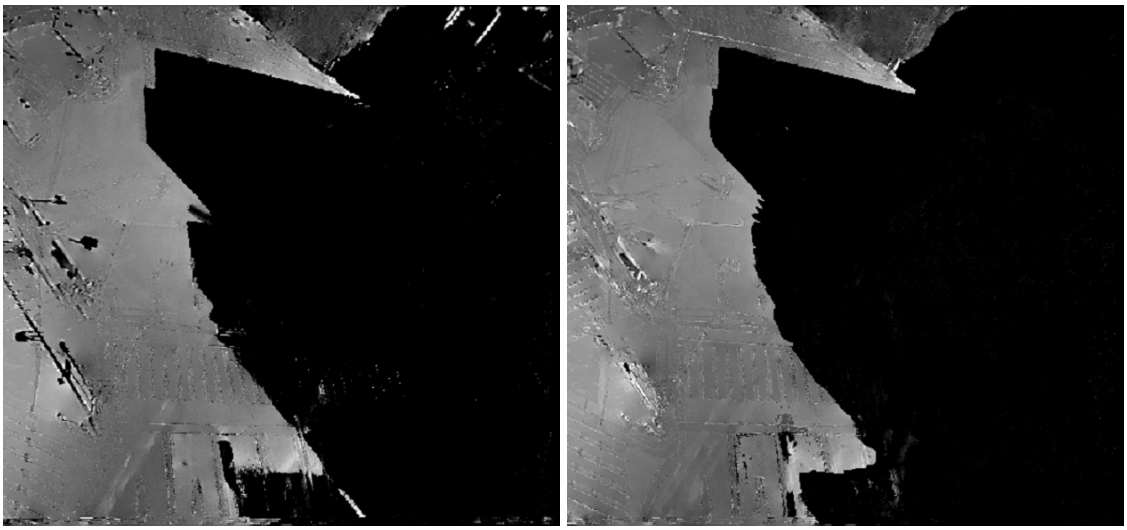


Figure 4.4. Interpolation of shadow region using shadow hull. Shadow region is deforming from left top shadow sample to right bottom shadow sample.

result of shadow interpolation in a CG scene using an estimated shadow hull. The dark regions show the interpolated shadow regions, while the lighter gray regions represent the sampled shadow regions.

Shadow interpolation using shadow hulls is useful to estimate the intermediate shadow shapes between sampled lighting conditions. Figure 4.3 shows the interpolated result of the real world scene. The left-hand side column represents the estimated shadow regions, while the right-hand side column shows the corresponding illumination images. The top and bottom row represent the images un-



(a) (b)

Figure 4.5. Comparison with the ground truth. (a) interpolated result using our method, (b) the ground truth

der sampled illumination conditions, and the middle row depicts the interpolated result. To obtain the intermediate illumination image, we first estimate shadow boundaries using the estimated shadow hull.

Once we obtain intermediate shadow boundaries, we then compute the intermediate illumination image $E_{int}(x, y)$ with the following equation.

$$E_{int}(x,y) = \begin{cases} \delta_{int}(x,y)\frac{\sum_k w_k \delta_k(x,y) E_k(x,y)}{\sum_k w_k \delta_k(x,y)} \\ \bar{\delta}_{int}(x,y)\frac{\sum_k w_k \bar{\delta}_k(x,y) E_k(x,y)}{\sum_k w_k \delta_k(x,y)} \end{cases} \tag{4.9}$$

where E_k is the k-th illumination image obtained by nearest neighbor search in the illumination eigenspace[3], and w_k is the weighting factor which is the distance from E_w^* to E_{w_k} (See Section 5.4) in the illumination eigenspace. $\delta_k(x, y)$ is the function which returns 1 if $L_k(x, y)$ is inside the shadow region, otherwise returns 0. $\bar{\delta}_k(x, y)$

[3]Illumination Eigenspace and the nearest neighbor search is described in detail in Section 5.3

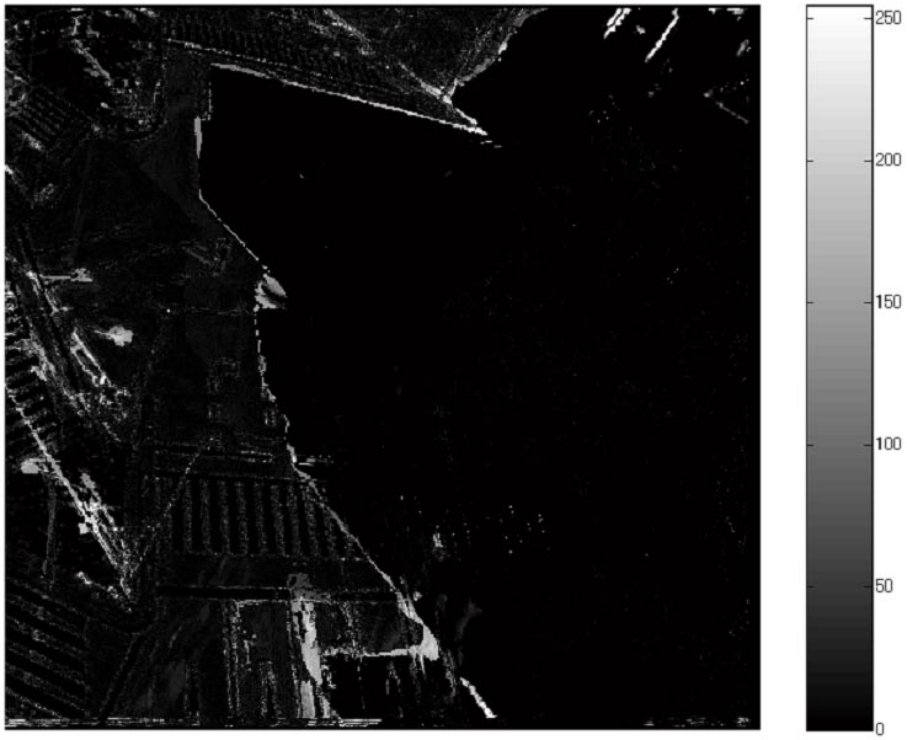


Figure 4.6. Image differencing between the ground truth and interpolated result using our method.

is the inverse of $\delta_k(x, y)$. The resulting intermediate image is shown in mid-right in Figure 4.3. Once we obtain the shadow hull, we can compute intermediate shadow shapes at arbitrary sampling rate. Figure 4.4 depicts the more intermediate shadow shapes at denser sampling rate between two illumination samples. It is more evident by comparing the resulting image with the ground truth. Figure 4.5 shows the comparison between the result of our method (a) and the ground truth (b). The output of pixel-wise differencing of the ground truth and the estimate using our method is shown in Figure 4.6. We can notice the slight difference between (a) and (b) in Figure 4.5 from Figure 4.6, however, it gives a globally correct shadow shape which is useful to remove shadowing effects from the input image.

4.2 Interpolation using Rough Scene Geometry

In this section, we propose an approach to interpolate cast shadows using rough scene geometry and parameterized light source. This work is also documented in [MKL+02].

4.2.1 Introduction

In the field of Computer Graphics, in the pursuit of photo-realism, methods for photo-realistic rendering generally forms two categories. One type requires accurate scene geometry and accurate physically-based rendering, which is called *model-based rendering* (MBR).

The other type of method requires densely sampled images to represents complex 3D environments with those sets of images, and the method is called *image-based rendering* (IBR). In recent years, much progress has been made in image-based rendering. One class of such methods relies on densely sampled images, such as the light field [LH96] and the Lumigraph [GGSC96]. Another class requires an accurate physically-based rendering algorithm and sufficiently detailed geometric and material properties of the scene and light sources [Deb98, SWI97, YDMH99]. Others require all of the above information [WAA+00].

Methods that rely on densely sampled images have the advantage that they do not require accurate geometry, which in practice requires a high-quality and expensive range finder. However, this advantage is achieved at the expense of a large database. In addition, it is not possible to relight the scene using these current image-based representations, with the exception of Wong *et al.* [WHON97], who use dense sampling of camera locations and illumination conditions (and hence may not be practical for real scenes). Methods that permit scene relighting typically need a detailed and accurate 3D geometric model in order to extract surface properties in the form of a Bidirectional Reflectance Distribution Function (BRDF). Usually, such models can only be acquired using expensive range find-

ers, and even then, the shapes used as examples tend to be simple. Nimeroff *et al.* proposed another approach [NSD94] to use steerable linear basis functions to accomplish re-rendering of a scene under a directional illuminant at an arbitrary orientation. One drawback of the method is that the method requires a huge basis set to handle narrow illuminants.

We are motivated by the need for a more *practical* approach to interpolate lighting appearance of a scene that has sparsely sampled lighting conditions. We require only images (light fields) as input, and assume that the camera positions associated with these images are known. The light fields are captured under a relatively small set of different lighting conditions. From these light fields, we can extract two separate datasets: view-dependent geometries using stereo, and *intrinsic images* using the mentioned method in Chapter 2.

4.2.2 Prior Work

Much of the work on realistic rendering relies on reflectance modeling and known 3D geometry. A representative approach in this area is presented by Sato *et al.* [SWI97], which merges multiple range datasets to yield a single 3D model. This shape is subsequently used for diffuse-specular separation and reflectance estimation. They showed results for single objects with no shadows. Wood *et al.* [WAA+00] also use color images and laser range scans. Their range datasets are merged manually to produce a global 3D model. Subsequently, a function that associates a color to every ray originating from a surface is constructed and compressed.

Yu *et al.* [YDMH99] compute surface BRDFs based on Ward's anisotropic BRDF model [War92] from multiple images and a 3D model. They assume that at least one specularity is observed per surface. On the other hand, Boivin and Gagalowicz [BG01] propose a technique for recovery of a BRDF approximation from a single image based on iterative analysis by synthesis (or *inverse rendering* [MG97]). The emittance of the light sources are assumed known. This is an extension of Fournier

et al.'s work [FGR93], which assumes perfectly diffuse surfaces, and Loscos *et al.* [LDR00], who additionally considered textured surfaces. Marschner and Greenberg [MG97] directly estimate the BRDF model of Lafortune *et al.* [LFTG97] from an image and a surface model. Malzbender *et al.* [MGW01] proposed a space and time efficient method for encoding an object's diffuse lighting response as the light position varies with respect to the surface by encoding a set of coefficients.

Debevec [Deb98] uses global illumination for augmented reality applications. He uses local geometry and manually computes reflectance parameters, with which objects can be inserted with realistic-looking inter-reflections. In a series of works geared for augmented reality, Sato *et al.* estimate the illumination distribution from shadows [SSI99b], and subsequently from the brightness distributions in shadows [SSI99a].

In our work, we rely on *intrinsic images* as a means for predicting shadows. Intrinsic images are a mid-level description of scenes first proposed by Barrow and Tenenbaum [BT78]. A given image of a scene can be decomposed into a *reflectance image* and an *illumination image*. Various methods have been proposed to compute this decomposition, with piecewise constant reflectances using the Retinex algorithm [LM71], with all-reflectance/all-illumination classification using wavelets [FV98], and with maximum-likelihood (ML) estimation assuming time-constant reflectance and time-varying illumination [Wei01].

4.2.3 Overview

An overview of our system is illustrated in Figure 4.7. The inputs to our method are a number of light fields, each captured under a different illumination condition. Once the light fields are acquired, view-dependent depth maps are computed at the sampled camera positions using a multi-view stereo algorithm.

In addition, we decompose the light fields into intrinsic images in a similar manner as [Wei01] (which handles a single image stream). For each camera and lighting position, the pair of intrinsic images consists of an illumination image that

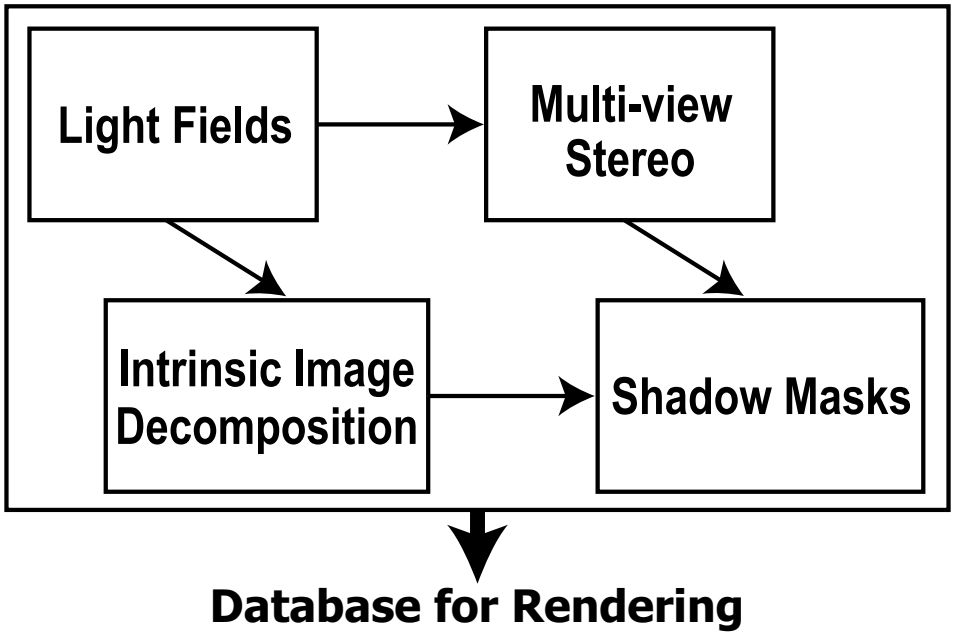


Figure 4.7. Block diagram of the affine transformation-based interpolation.

exhibits shading and shadowing effects, and a reflectance image that displays the unchanging reflectance property of the scene. The illumination images are used to identify pixels that contain cast shadows or attached shadows, which result when a surface area is occluded from the light source. These shadow masks are used in conjunction with shadows predicted by the scene geometry to estimate shadow appearance for novel lighting directions.

We call this new representation the *Intrinsic Lumigraph*, because it uses both geometry and intrinsic images for view reconstruction. When interpolating lighting condition of the scene, the diffuse reflection and shading can be well-approximated by interpolation of illumination images; however, shadows generally do not appear realistic when linearly combined. Our method for predicting shadow appearance enables us to synthesize images with much more accurate lighting interpolation.

4.2.4 Constructing the Intrinsic Lumigraph

In this section, we detail the process of constructing the Intrinsic Lumigraph. We first describe the capture of light fields under various illumination conditions, and then outline our algorithm for multi-view geometry. We next present our method for computing the intrinsic images, followed by the determination of shadow masks.

[Capturing light fields under various illuminations]

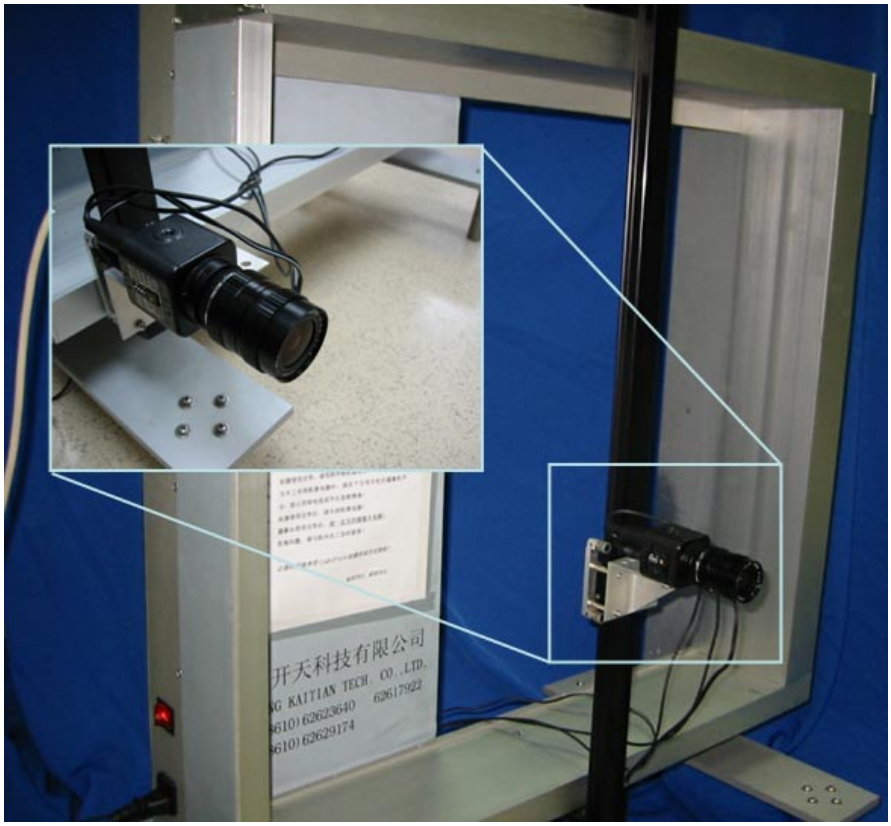


Figure 4.8. Light field capture device.

We capture our light fields using the imaging setup shown in Figure 4.8. The camera is digitally controlled to capture images at predefined positions on a 2D grid. Each light field consists of an image sequence along a linear path that is

captured under a fixed illumination condition, where the light source used is approximately a point light source.

[Generating view-dependent geometries]

Using the captured light fields, we compute depth maps at each camera position using a multi-view stereo algorithm. The stereo algorithm is based on the work of Kang *et al.* [KSC01]; it was chosen because it is very simple to implement and is very effective in handling occlusions. To improve the depth estimates, we linearly combine depth estimates from separate light fields taken under different lighting conditions. Depth estimates from areas that are more highly textured are favored.

From a sequence of N light fields, for each reference view we first obtain N estimated depth maps $D(n)$ and N confidence maps $C(n)$ using the multi-view stereo algorithm. The confidence map $C(n)$ is computed using the local matching error variance, which provides an indication of the reliability of the estimated depths. We use these confidence maps to refine the depth values through weighted averaging, i.e.,

$$D(x,y) = \frac{\sum_n^N D(n,x,y) \cdot C(n,x,y)}{\sum_n^N C(n,x,y)} \tag{4.10}$$

An alternative method for refining the estimated depth values is to use the local Hessian of the local brightness distribution. The eigenvalues of the local Hessian are correlated with the degree of local texturedness; the higher the amount of texture, the more reliable the depth estimates tend to be in general. To be conservative, we use the minimum eigenvalues as a measure of depth reliability and as a means for weighting the depth estimates.

Hessian is obtained from the differential method of SSD (sum of squared differences).

$$E(u,v) = \Sigma_{k,l}(I_1(x+u+k, y+v+l) - I_o(x+k, y+l))^2 \tag{4.11}$$

The differential method uses a local Taylor series expansion of the intensity function:

$$\begin{aligned} & E(u+\Delta u, v+\Delta v) \\ = \; & \Sigma_{k,l}(I_1(x+u+\Delta u+k, y+v+\Delta v+l) - I_0(x+k, y+l))^2 \\ \simeq \; & \Sigma_{k,l}(I_1(x+u+k, y+v+l) + \nabla I_1 \cdot (\Delta u, \Delta v)^T - I_0(x+k, y+l))^2 \\ = \; & \Sigma_{k,l}(\nabla I_1 \cdot (\Delta u, \Delta v)^T)^2 + \Sigma_{k,l}(\nabla I_1 \cdot (\Delta u, \Delta v)^T) e_{k,l} + E(u,v) \end{aligned} \tag{4.12}$$

where $\nabla I_1 = (I_x, I_y) = \nabla I_1(x+u+k, y+v+l)$ is the intensity gradient and $e_{k,l}$ is the term inside the brackets in 4.11. Minimizing w.r.t. $(\Delta u, \Delta v)$, we obtain a 2×2 system of equations

$$\begin{bmatrix} \Sigma_{k,l} I_x^2 & \Sigma_{k,l} I_x I_y \\ \Sigma_{k,l} I_x I_y & \Sigma_{k,l} I_y^2 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \Sigma_{k,l} I_x e_{k,l} \\ \Sigma_{k,l} I_y e_{k,l} \end{bmatrix} \tag{4.13}$$

The matrix on the left hand side is referred to as the *Hessian* of the system.

Both methods produce comparable results, which are significantly better than the depth maps generated from any one light field alone. We tested this on a synthetic light field with known 3D geometry, and compared our results that merge the depth estimates from all the light fields to one that uses only a single light field. The results can be seen in Figure 4.9. In this experiment, we used nine light fields of a synthetic scene under different illumination directions (left). Each light field has 9×9 images, and only the central image (used as the reference) is shown in Figure 4.9. In this work, the local matching error variance is used to improve accuracy of the depth values.

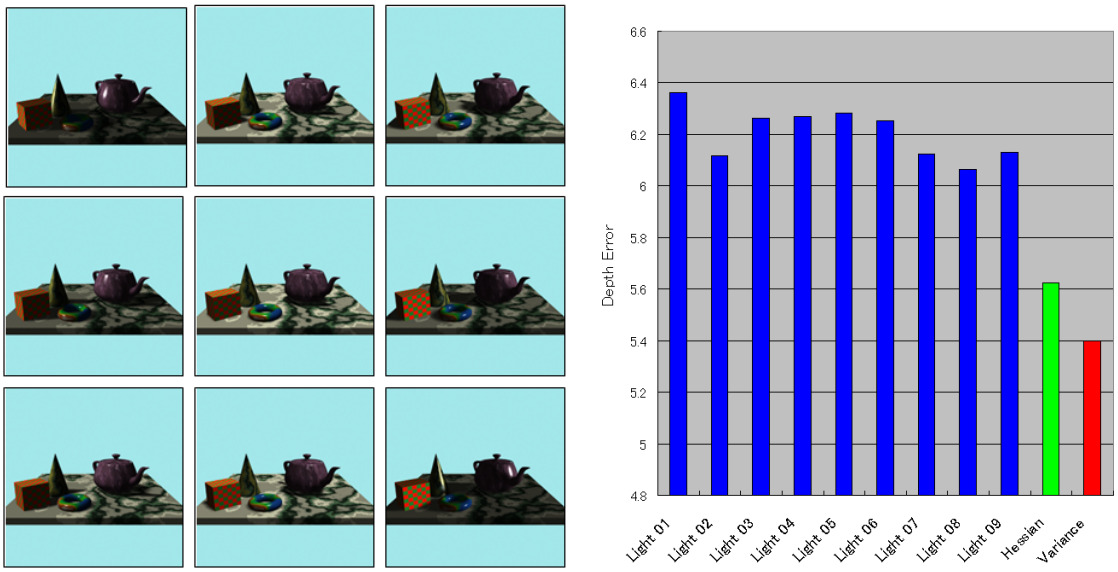


Figure 4.9. Illumination sampling (left) and comparison of mean depth errors (right). The nine blue bars correspond to mean depth errors for each of the light fields, the green bar is the error when the Hessian (5×5 window) is used, and the red bar is the error obtained when the matching error variance (5×5 window) is used.

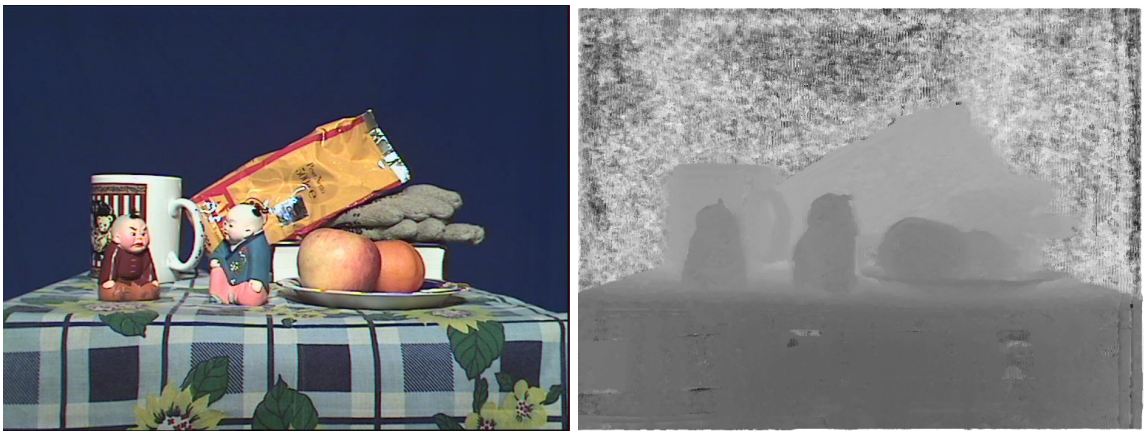


Figure 4.10. Illumination sampling (left) and obtained depth map using multi-view stereo (right).

We chose to compute the local view-dependent geometries because the stereo algorithm, while good, does not produce perfectly accurate geometry. In addition, some degree of photometric variation along the image sequence usually exists, making the direct production of a single accurate global 3D geometry from images very difficult. The local geometries encode such photometric variation, since they are *highly locally photo-consistent*. The stereo algorithm has the tendency to maximize this behavior.

Figure 4.2.4 (right) shows the obtained depth image (depthmap) using multi-view stereo algorithm. In the figure, the darker pixel represents the smaller depth value, while the brighter indicates the larger depth value. Though obtained depth estimates are usually smoothed to get final depth estimates, the figure shows pre-smoothed depth map to see the raw output.

[Extracting intrinsic images]

In this method, We applied Weiss's ML estimation method [Wei01] to derive intrinsic light fields. Given a sequence of N light fields with varying illumination, it is decomposed into a single reflectance light field and N illumination light fields. With images of $u \times v$ in size from $s \times t$ view points under n different illumination conditions, we can denote this decomposition as follows:

$$L(s,t,u,v,n) = R(s,t,u,v) \cdot E(s,t,u,v,n) \tag{4.14}$$

where $L(s,t,u,v,n)$, $R(s,t,u,v)$, and $E(s,t,u,v,n)$ are an input light field sequence, a reflectance light field, and an illumination light field sequence, respectively. In the log domain, (4.14) is written as (4.15):

$$l(s,t,u,v,n) = r(s,t,u,v) + e(s,t,u,v,n) \tag{4.15}$$

For each of M derivative filters $\{f_m\}$, a filtered reflectance light field $\hat{r}_m$ is estimated by taking the median of filtered input light fields:

$$\hat{r}_m(s,t,u,v) = median_n\{l(s,t,u,v,n) \otimes f_m\} \tag{4.16}$$

Finally, $R(s,t,u,v)$ is recovered by deconvolution of the estimated filtered reflectance light fields $\hat{r}_m$.

4.2.5 Computing shadow masks

A major difficulty in lighting interpolation is the realistic generation of shadows. To compute shadow masks for real scenes, our approach first infers shadow pixels from the illumination intrinsic image by simple thresholding, since image areas of lowest radiance can be taken as shadowed regions. A shadow mask computed in this manner is shown in Figure 4.13.

While this technique might allow us to estimate shadow regions for images at sampled illumination conditions, it cannot be employed for intermediate lighting directions, because we do not have the associated images. Since we are not able to predict the shape of intermediate shadow masks from intrinsic images, we instead predict the general *shadow distortion* between the sampled lighting conditions using the shadows cast from the view-dependent geometries. Although these geometries are not highly accurate, their shadows can be computed for arbitrary light directions, and the distortions in shadow shape as a light source moves from one sampled position to another can nevertheless be helpful in morphing the shadows computed from intrinsic images.

In this process, we first estimate light source type (point / directional) and lighting directions of captured images with some user interaction. By clicking on several pairs of corresponding shadow and object points in an image, the light source position can be determined by least-squares triangulation. With the light position

and the estimated geometry, the resulting shadows can be computed. We can also compute the geometric-based shadows for light positions between the sampled illumination directions.

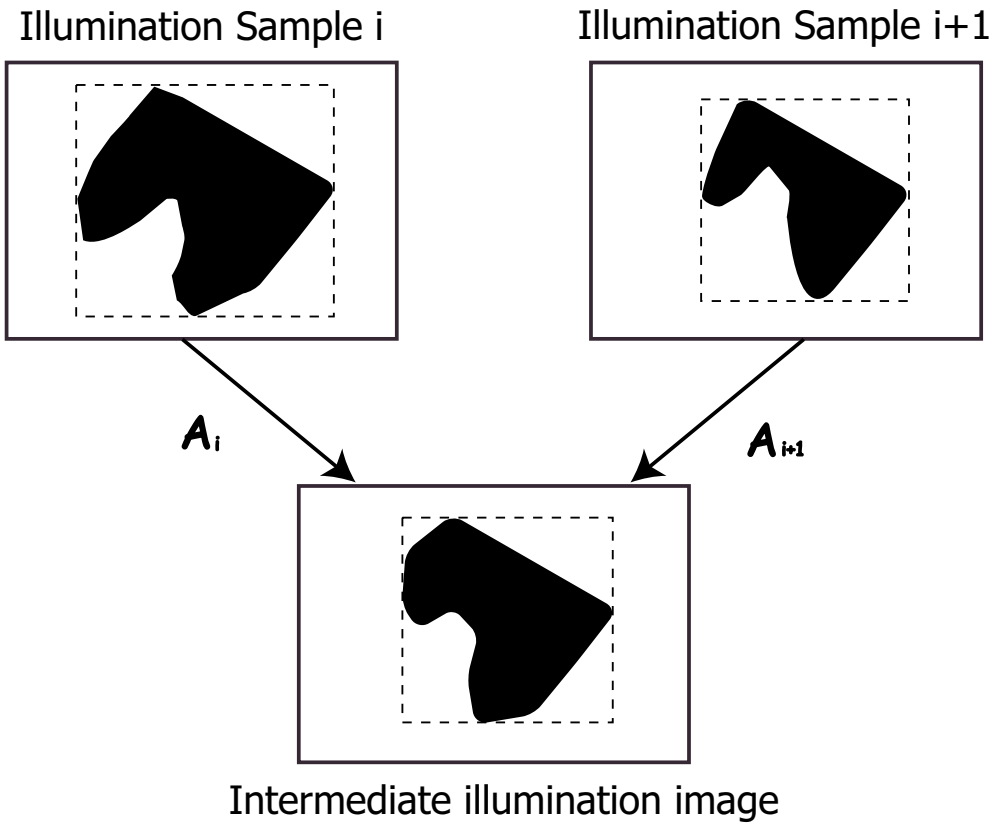


Figure 4.11. Illustration of subimage registration over the geometric-based shadow blobs. Changes of intermediate shadows' shape are represented by transformation matrices from neighboring bases, i.e. the geometric-based shadows under sampled illumination conditions.

After computing the geometric-based shadows, the changes in the geometric-based shadows are represented by the region-based transformation matrices. Assuming each shadow blob to be a subimage region, we employed subimage registration to compute the region-based shadow transformation matrices.

By computing those matrices, the changes in geometric-based shadow shape from one sampled light position to another can be used to guide the transformations of shadows computed from intrinsic images. In Figure 4.11, transformation

matrix $\mathbf{A}_i^{+j}$ corresponds to warping of the shadow blob from base image i to intermediate image j. Since those shadow blobs do not have texture in them, nearest shadow blobs are assumed to be the corresponding shadow blobs. We assumed geometric distortion of the geometric-based shadow blobs can be described by linear 2-D geometric transformations as long as they are densely computed. Thus, we model the transform as 2-D affine and the transform $\mathbf{A}$ is described by 3×3 matrix.

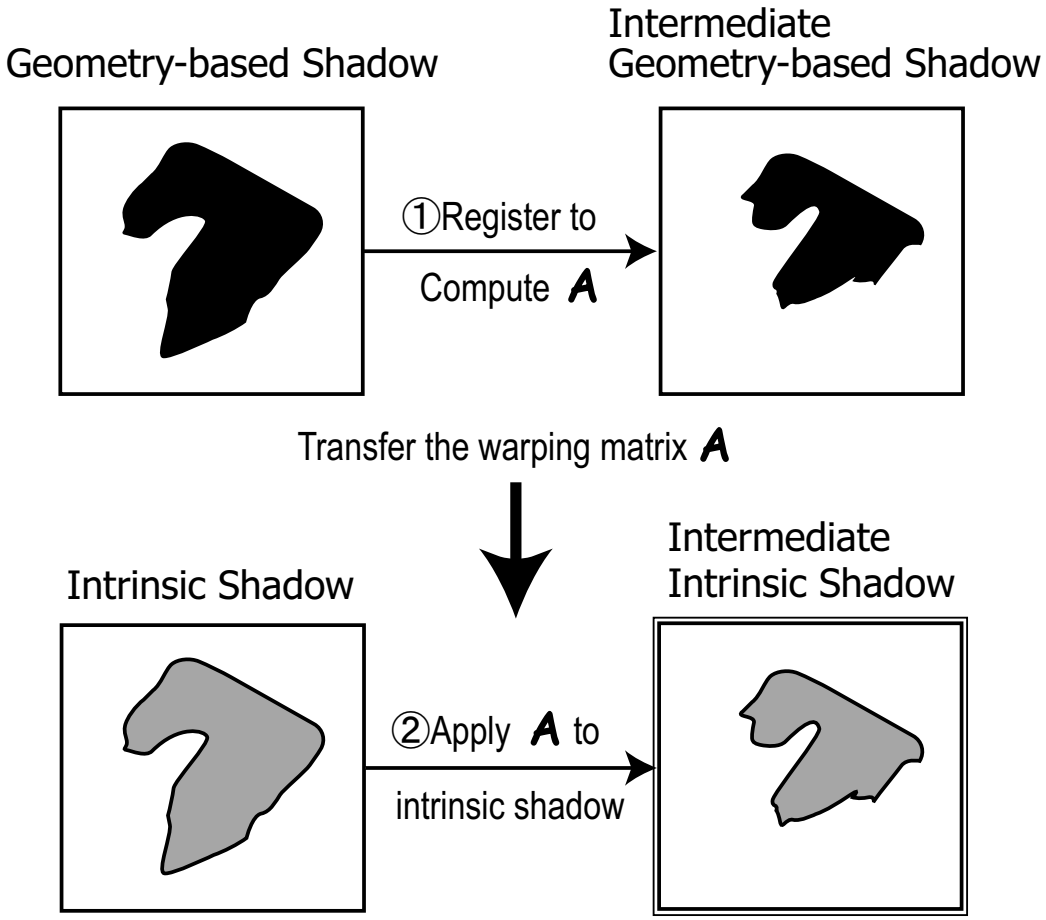


Figure 4.12. After computing transformation matrix $\mathbf{A}$ of the geometric-based shadow by subimage registration, the transform $\mathbf{A}$ is then applied to the corresponding intrinsic shadow to generate intermediate shadow.

This is done by attaching the intrinsic image shadows to the geometric-based shadows, and as the geometric shadows are morphed from one sampled light-

(a) (b) (c)

Figure 4.13. Example of an illumination image and its shadow mask counterpart. (a) Original image, (b) Illumination image, (c) Shadow masks.

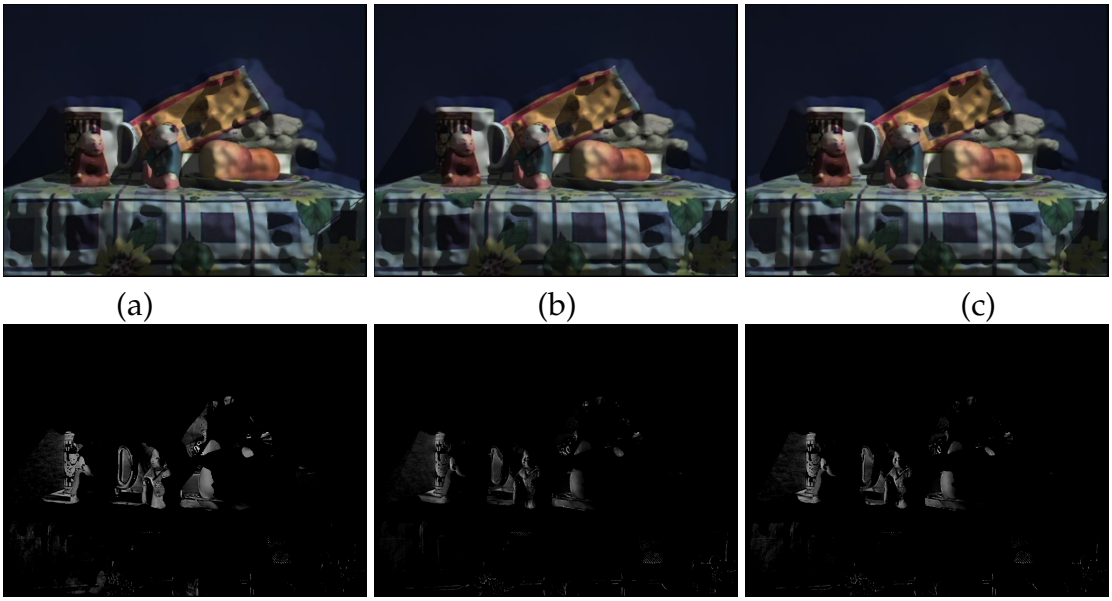
ing to another, the intrinsic shadows are morphed correspondingly as shown in Figure 4.12. Attachment is done by simply taking an *AND* operation on the corresponding shadow regions. In this operation, correspondences between the intrinsic shadows and the geometric-based shadows are estimated by checking overlapping regions of them. Figure 4.14 shows an example of geometric-based shadow warping applied to intrinsic shadow masks. Once the shadow masks are predicted at intermediate illumination conditions, the view can then be synthesized, as shown in Figure 4.15. This synthesis is computed by removing shadows in the sampled images via intrinsic images, linearly interpolating diffuse reflections, then computing shadows from the intrinsic lumigraphs.

4.2.6 Results

In this section, we show results of lighting interpolation for two real scenes.

[Toy Scene]

Figures 4.2.6 shows examples of interpolating lighting condition of a toy scene with our shadow warping technique (a) and direct linear interpolation (b). For this scene, we captured seven light fields with different lighting conditions, where each light field is composed by 17×17 images. We can clearly see the difference between the results of our method and those of linear interpolation, especially



(a) (b) (c)

(d) (e) (f)

Figure 4.14. Illustration of applying the transformation of geometric shadows to intrinsic shadows. (a) Geometric shadows at L1, (b) Geometric shadows between L1 and L2, (c) Geometric shadows at L2, (d) Shadow masks at L1, (e) Shadow masks between L1 and L2 (after applying the geometric-based warping), (f) Shadow masks at L2. L1 and L2 are sampled illumination conditions.

(a) (b)

Figure 4.15. Example of view synthesis at intermediate illumination conditions: Warped intrinsic shadow masks (left), Synthesized view (right).

on the cast shadow of left-hand side toy. This is more evident by comparing the leftmost two images in Figure 4.2.6. The direct linear interpolation resulted in significantly softer shadows, which is less consistent with the original sampled images. And furthermore, a comparison among the ground truth, the result of our method, and that of linear interpolation is shown in Figure 4.19. As we can see clearly, our method successfully produces a realistic shadow while the result of linear interpolation is quite unlike the ground truth.

To quantitatively compare the results of our method and simple interpolation method, the difference between the results and the ground truth is pixel-wisely computed. In Figure 4.19, (a) is the result of our method, (b) is the ground truth, and (c) is the result of simple interpolation. The image difference between (a) and (b) is shown in (d), and between (b) and (c) is shown in (e). The image differencing is done by summing up the RGB components' distance. As is shown in a color bar in the figure, the larger difference is colored by red while the smaller differences are colored in blue. We can clearly see the better result is obtained by our method.

[Portrait scene]

Figures 4.2.6 (a) shows the results of our lighting interpolation of a scene containing a portrait. We captured ten light fields under different illumination conditions for this scene. Each light field is composed of 16×16 images. While cast shadows in Figure 4.2.6 (b) are blurred and exhibit jumpy movements in video for linear interpolation, cast shadows warped by our method look more natural in Figure 4.2.6 (a) and move smoothly in video. This is more evident by comparing the rightmost two images in Figure 4.2.6. Again the direct linear interpolation method resulted in softer shadows, unlike those in the original input images.

4.2.7 Conclusions and Future Work

We have described an approach for lighting interpolation of a scene without the need for accurate physically-based rendering or detailed 3D geometry. It uses only

Figure 4.16. Closeup views. The left of each pair is generated using our method while the other is computed using direct interpolation.

light fields captured under different, sparsely sampled, illumination conditions. Our approach uses intrinsic images and local view-dependent depths computed from stereo in order to predict shadows at intermediate illumination conditions, which add significantly to the realism of the synthesized view. The limitation of the method is that the method requires the scene to be largely diffuse scene, since the reflectance image $R(x, y)$ in the Weiss's framework of intrinsic images is lighting-invariant which basically can not handle the changes of reflectance property. We are working on to derive time-varying reflectance image $R(x, y, t)$ and corresponding illumination images $L(x, y, t)$ to overcome the limitation.

(a) (b)

Figure 4.17. Interpolation results of the toy scene. (a) Lighting interpolation examples for the toy indoor scene. (b) Lighting interpolation using direct interpolation for the toy indoor scene.

(a) (b)

Figure 4.18. Interpolation results of the portrait scene. (a) Lighting interpolation examples for the portrait indoor scene. (b) Lighting interpolation using direct interpolation for the portrait indoor scene.

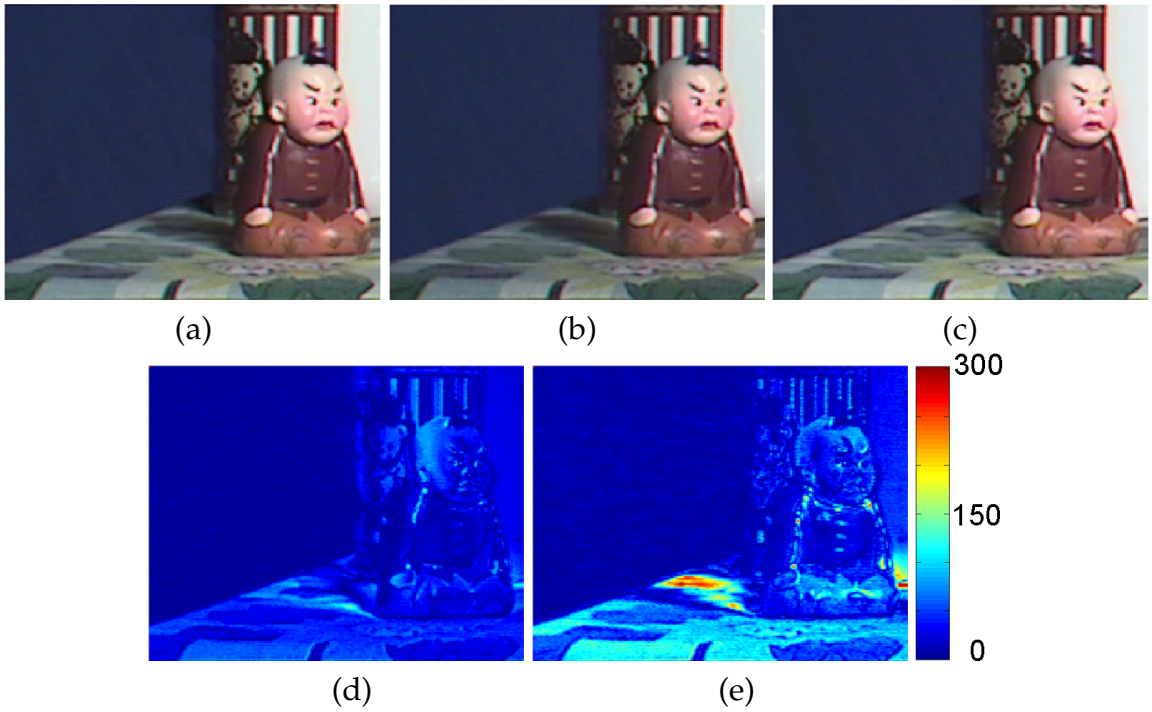


Figure 4.19. Comparison with the ground truth. (a) Interpolated result of our method, (b) the ground truth, (c) simple pixel-wise interpolation, (d) difference between our result (a) and the ground truth (b), (e) difference between simple interpolation (c) and the ground truth (b).

In future work, we would like to be able to perform object manipulation such as object insertion and removal, all while enabling realistic scene lighting interpolation. Moreover, we would like to address the more difficult issue of lighting interpolation of outdoor scenes. This has the added difficulty of not being able to capture light fields with a set of consistent illumination conditions, because of the time elapsed between successive camera snapshots within a light field capture.

4.3 Summary

Scene illumination gives the scene complex and rich tone which helps us to understand the scene geometry and surface materials. Physically, the illumination

effects can be totally described by a set of ray emitted from light sources including reflected ray on surfaces. Using the physically-based method with accurate scene geometry, physically-based surface reflection model, i.e. BRDF (Bidirectional Reflectance Distribution Function), and completely parameterized light source distribution, we can compute the exact appearance of the scene. However, it is well known that none of them is easy to obtain. Considering the interpolation of scene illumination using only 2-D images captured under the different illumination conditions, we immediately notice that it cannot be accomplished by simple image interpolation because illumination effects concern complicated physics as mentioned.

We proposed two different methods to accomplish the non-linear interpolation of scene illumination. Both of them rely on some kind of scene geometry to accomplish non-linear interpolation. The shadow hull-based approach uses the largest possible intersection of shadow volumes, and the intrinsic lumigraph uses rough scene geometry recovered using multi-view stereo algorithm. Those two methods especially focus on the motion of cast shadows. Representing motion of cast shadows is mathematically very complicated without knowing exact scene geometry and light source parameters, therefore it has been a difficult problem in the field of computer vision and graphics.

The first method described in Section 4.1 uses shadow hull to compute shadow regions with the additional information of sunlight angle and ground plane assumption. Using shadow regions estimated from sampled illumination images associated with sunlight angles, first shadow volumes are constructed. Taking the largest possible intersection of those shadow volumes, a shadow hull, which represents the minimally precise shape which is enough to cast sampled shadows, is obtained. We assume the intermediate shadow shape is approximated by the shadow shape computed using the shadow hull and the intermediate light source position as long as illumination conditions are densely sampled. Finally, estimated intermediate shadow regions are used to compute the intermediate illumination

images, and we confirmed the resulting illumination images are much closer to the ground truth.

The other method takes a different way from the shadow hull to use rough scene geometry as described in Section 4.2. The scene geometry recovered using multi-view stereo algorithm is not accurate, but it gives reasonably good indication of general shadow distortion. We assumed that the shadow distortion can be represented by 2-D affine transformation on the image plane as long as they are densely computed. We applied the geometrically-based shadow interpolation framework to lumigraphs, more precisely to the intrinsic lumigraphs which we refer to the new representation, to accomplish lighting interpolation of the scene.

CHAPTER 5

APPLICATION TO REAL-TIME VIDEO SURVEILLANCE SYSTEMS

For those outdoor vision applications, it is much preferred that the scene illumination condition is static and stable. Under the unstable, or dynamic, scene illumination condition, algorithms that rely on the visual appearance of objects suddenly become precarious and this results in a drastic increase in the number of errors. Even if the scene illumination condition is stable, illumination effects such as large cast shadows lie in the scene give a bad effect to object detection and object tracking. Suppose an object is crossing the boundary of a large cast shadow. While the object is inside the shadow, it looks darker compared to when it appears in a better lit area. This appearance variation causes tracking error when the object moves across the boundary of cast shadows. We are motivated by this fact and propose to normalize the illumination effect on the scene.

One of the most widely spread outdoor vision system is a road traffic monitoring system. The road traffic monitoring using video cameras is expected to become replacement for the human resources because of its faster and cheaper potential to gather various kinds of information such as traffic accidents, congestion and etc. We are going to integrate our illumination normalization technique described in Section 3.1 to those existing road traffic monitoring systems. Supposed constraints

are only two, i.e. 1) the camera is fixed and 2) various scene illumination conditions are observed. Basically our illumination normalization method can be used as a preprocessing stage for the subsequent processing, and can be integrated into existing systems without changing the core system design. One may think of using normalized correlation for illumination-free object tracking. However, normalized correlation is known to fail when the object comes to the shadow boundary, and the computational cost is more expensive compared to the basic block matching algorithm. In addition, using normalized correlation may yield the need to change the algorithm design itself. In this sense, we believe our method can be seamlessly integrated into the existing systems without dependency on their algorithm design.

In this chapter, we first give an overview in Section 5.1 of our illumination normalization framework for video surveillance systems and the following sections go along as described in the overview.

5.1 System overview

Our method is composed of two parts as shown in Figure 5.1. The first part is the estimation of intrinsic images, which is an off-line process, depicted in Figure 5.1 A. In this part, the scene background image sequence is first estimated to remove moving objects from the input image sequence using a method described in Section 5.2. Using this background image sequence, we then derive intrinsic images using our estimation method described in Chapter 2. Using estimated illumination images, which is one part of intrinsic images, we are able to robustly cancel out the illumination effects from input images of the same scene. It enables many vision algorithms such as tracking to run robustly. After the derivation, we construct a database using principle component analysis (PCA), which we refer to as *illumination eigenspace* [MNIS02a, MNIS02b], which captures the variation of lighting conditions in the illumination images. The database is used for the follow-

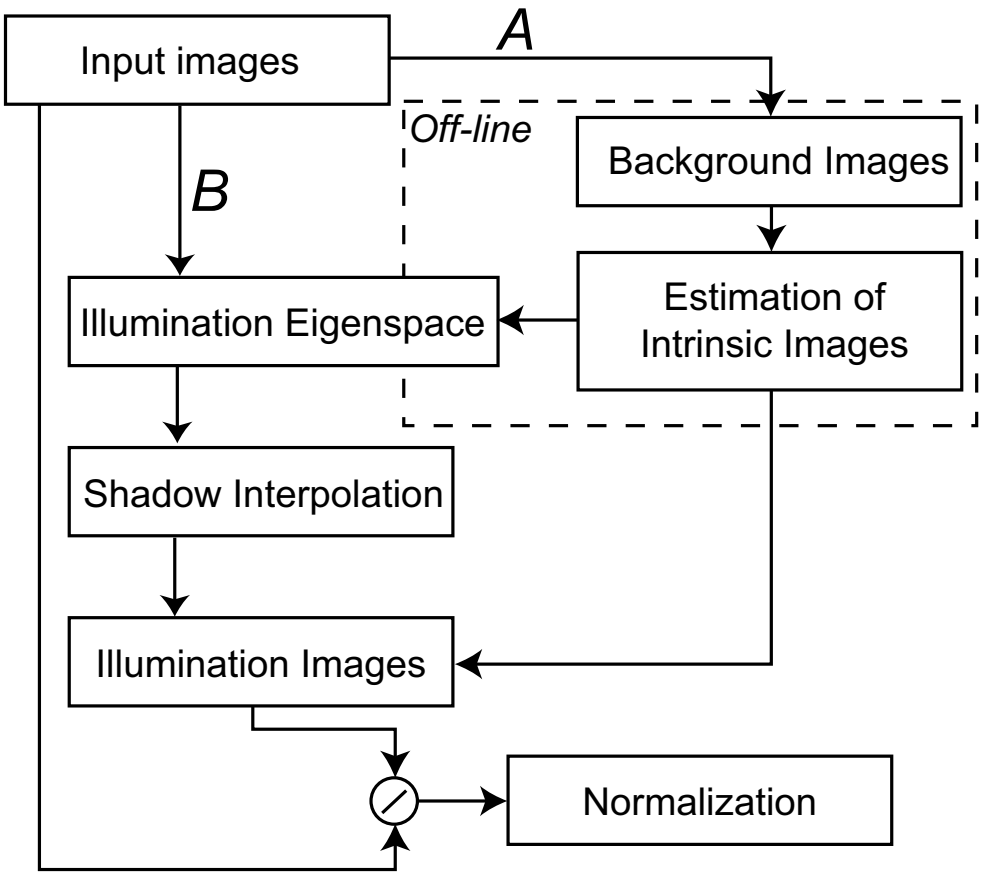


Figure 5.1. System diagram for illumination-normalization.

ing direct estimation method. Details of constructing the illumination eigenspace is depicted in Section 5.3.

The second part is direct estimation of illumination images, shown in Figure 5.1 B. Using the pre-constructed illumination eigenspace, we estimate an illumination image directly from an input image. In addition, to obtain more accurate illumination images, the method for shadow interpolation using shadow hulls described in Section 4.1 is used to estimate final illumination images.

5.2 Estimating background images

Background image is an image which does not contain moving foreground objects. The definition varies with the purpose of applications and extensive amount of work have been done so far. Separating foreground moving objects such as vehicles from a relatively static background scene is considered to be an important preprocessing stage in many computer vision algorithms [DGWH00, WADP97].

In many visual surveillance applications that work in outdoor scenes, background of the scene contains many non-static objects such as tree branches and bushes whose movement depends on the wind in the scene. This kind of background motion causes the pixel intensity values to vary significantly with time. To model the dynamic background instead of the static background image, Elgammal *et al.* present a non-parametric color modeling approach [EDD01b, EDD01a] based on kernel density estimation. Fitting a mixture of Gaussians using the EM algorithm provides the other way to model color blobs with a mixture of colors [SG99, RMG98, RMG99]. Ridder *et al.* [RMK95] proposed to use Kalman-Filtering for adaptively update the background image.

For the case of most road traffic monitoring systems, the background is the road surface and considered to be roughly static. Our purpose to create background images is not for the object detection but for estimating intrinsic images. To estimate the intrinsic images, it is necessary to remove foreground moving objects from the image sequence. Therefore we took an approach to estimate a set of background images for each short time period to generate an image sequence which does not contain moving objects but captures the illumination variations. Preassumptions here is that the scene illumination does not vary dramatically in the short time periods. To obtain background images, we took an approach to use color histogram. We first create the color histogram by accumulating the observed pixel colors along the time axis for each pixel. Subsequently, by taking the median of the color histogram, a static background image of the corresponding time period is obtained.

Of course the richer method such as Wallflower [TKBM99] would give the better results, but the simple method like color histogram reasonably yields good background images for our case.

5.3 Illumination eigenspace

If the background image is determined by a method based on the color histogram in the short time period as mentioned in the former section, it can be frequently updated to capture the slowly moving shadows in an up-to-date position in the background image. However, especially on partially cloudy days, shadows will come and go unpredictably and rapidly as the sun is obscured by clouds. This will cause shadows to appear or disappear in the scene, depending on whether the background was made during the overcast period or during the sunny period. In either case, the state transitions of the shadows will be detected as movement and not as the part of the background. This can cause problems, since we assume that illumination would not change during the short time period for background creation.

Therefore it is preferable to estimate illumination images $e(x, y, t)$ directly from the input image sequence without creating a background image: given one input image i obtain its corresponding illumination image l. We accomplish this by preparing a database of illumination images a priori. As a preliminary framework, we propose to use principle component analysis (PCA) [NMN96] to construct an illumination space of a target scene. PCA is widely used in signal processing, statistics and neural computing. The basic idea of PCA is to find the basic components $[s_1, s_2, \ldots, s_n]$ that explain the maximum amount of variance possible by n linearly transformed components.

To analyze key characteristics of the illumination variation of real world scenes, we have stored image sequences for 120 days from 1 year, from 7:00 a.m. to 15:00 of an crossroad from a fixed view point. For each 1 hour image sequence, a back-

ground image is estimated using 7 minutes image sequence, then corresponding intrinsic images are estimated. We store E_w in the database and keep the mapping from E_w to R and E. The reason why we apply PCA to E_w but not L is that we assume that the modeled eigenspace would represent a rough tendency of the variation of illumination.

$$E_w = \frac{R \cdot E}{R_w} \tag{5.1}$$

First, an illumination space matrix is constructed by subtracting $\bar{E}_w$, which is the average of all E_w, i.e. $\bar{E}_w = \frac{1}{n}\sum_n E_w$, from each E_w and stacked column-wise.

$$\mathbf{P} = \{E_{w_1} - \bar{E}_w, E_{w_2} - \bar{E}_w, \ldots, E_{w_n} - \bar{E}_w\} \tag{5.2}$$

$\mathbf{P}$ is a $N \times M$ matrix, where N is the number of pixels in the illumination image and M is the number of illumination images E_w. We made the covariance matrix $\mathbf{Q}$ of $\mathbf{P}$ as following Equation (5.3).

$$\mathbf{Q} = \mathbf{P}\mathbf{P}^{\mathbf{T}} \tag{5.3}$$

Then, the eigenvectors $\mathbf{e}_i$ and the corresponding eigenvalues λ_i of $\mathbf{Q}$ are determined by solving Equation (5.4). We implemented Turk and Pentland's method [TP91], which is useful to solve the equation when $\mathbf{Q}$ is high dimension, to get the eigenvectors of $\mathbf{Q}$.

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \tag{5.4}$$

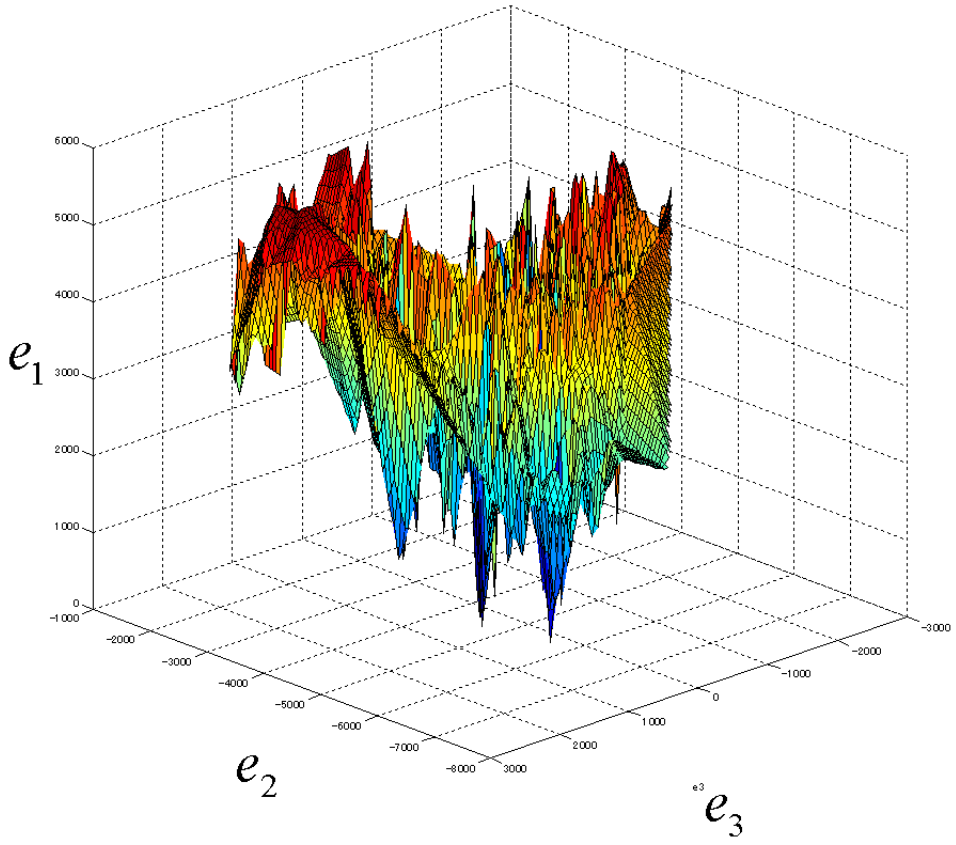


Figure 5.2. Illumination eigenspace constructed using 120 days data of a crossroad.

Figure 5.2 shows the manifold constructed by mapping all E_w onto the eigenspace using all eigenvectors. Figure 5.3 shows the hyper-plane constructed by mapping all illumination images onto the eigenspace using all eigenvectors. For display, only the first three eigenvectors are used in Figure 5.2 and 5.3.

In Figure 5.3, while the three axes represent the first three eigenvectors, the graph is transformed so that the variation along different days is aligned to the vertical axis which is the first eigenvector (the eigenvector with the largest eigenvalue). Also the variation along time-line is shown as the parabolic curve when the graph is sliced orthogonal to the vertical axis. For example, the upper part rep-

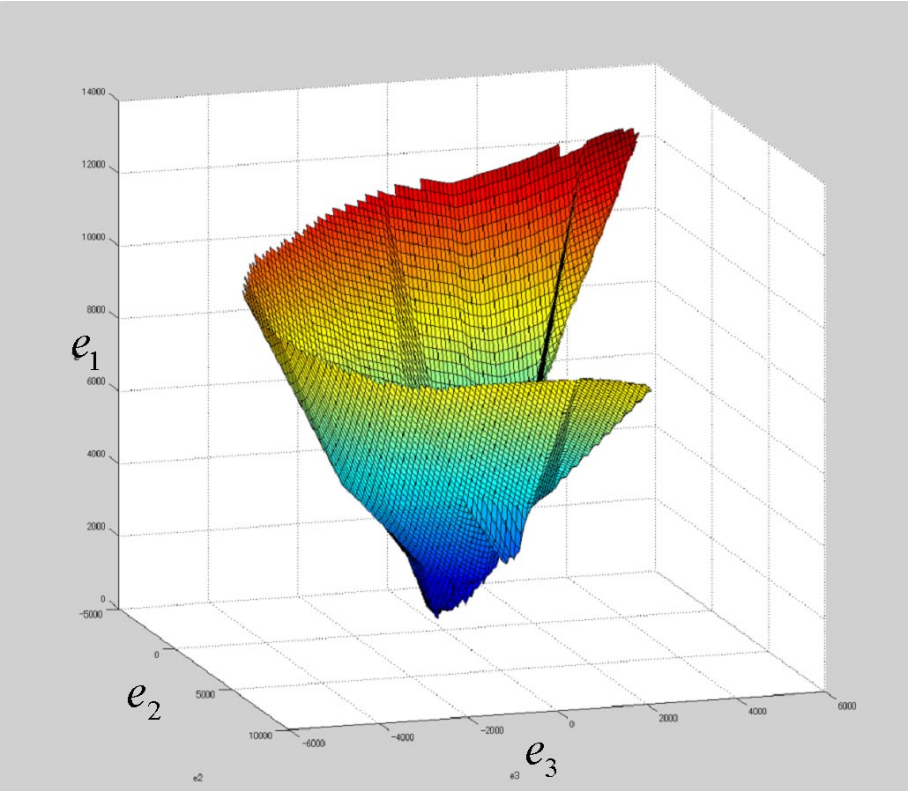


Figure 5.3. Illumination hyper plane in the eigenspace.

resents illumination variation along the time-line of a sunny day, and lower part represents rainy and cloudy days.

As can be seen clearly, the most significant variation caused by illumination and time in the illumination images can be captured with the first few eigenvectors. So that, by constructing an eigenspace of the illumination image sequence with the first k significant eigenvectors and mapping all illumination images onto the eigenspace, we obtain an efficient representation of the variation of illuminance in the input image sequence.

The number of stored images for this experiment was 2048 and the contribution ratio was 84.5% at 13 dimensions, 90.0% at 23 dimensions, and 99.0% at 120 dimensions. The graph of the cumulative contribution ratio is shown in Figure 5.4.

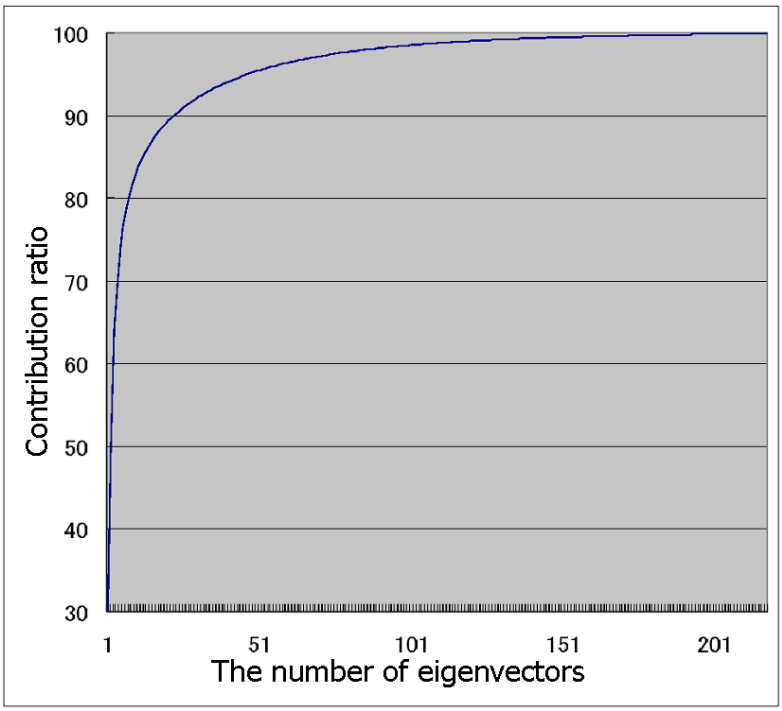


Figure 5.4. Contribution ratio of Illumination eigenvectors.

We choose to use 99.0% of eigenratio for this experiments. Thus the compression ratio is about 17:1, and the space needed to store the subspace is about 120MBytes.

5.4 Direct estimation of illumination images

Computational cost of deriving illumination images using the method described in Chapter 2 is high, and it is not yet possible for integration into real-time applications. For real-time applications, it is preferred if we could directly estimate the illumination image given an input image. To accomplish the direct estimation of illumination images, we take a straight forward way to store a lot of illumination images into the illumination eigenspace as a database of illumination images [MNIS02d, MNIS02c]. Though we construct the illumination eigenspace using E_w in the previous section, what we actually need is not E_w modeled in this

illumination eigenspace but E, the totally scene-texture-free illumination image. However, it is useful to construct an illumination eigenspace with E_w because the corresponding reflectance image R_w, which is the constant image along the time axis, can be used to derive *pseudo illumination images*, i.e. $E_w^* = L \oslash R_w$. Pseudo illumination images contain the foreground moving objects, but the global illumination effects are reduced in the images. Thus, we choose to use the pseudo illumination images as a query to search the best approximation of the scene illumination images. To estimate E directly from L, we take the following approach.

1. Construct illumination eigenspace using E_w with keeping the mapping from E_w to E and R.

2. Given an input image L, divide L by R_w to get pseudo illumination image E_w^*.

3. Accomplish nearest neighbor search in the illumination eigenspace using E_w^* as the query and obtain $\hat{E}_w$.

4. Get $\hat{E}$ and $\hat{R}$ from $\hat{E}_w$ using the mapping table.

We employ the SR-tree search algorithm [KS97] for the nearest neighbor search which is featured by the combined utilization of bounding spheres and bounding rectangles to improve the performance on nearest neighbor searches by reducing both the volume and the diameter of regions.

Figure 5.5 and 5.6 show the results of estimating E_w by the nearest neighbor search using E_w^* as a query. Each input image is taken under different illumination condition, i.e. rainy, cloudy, and sunny scene from top to bottom. Along the row, (a) shows original input images that contain moving objects in the scene. (b) is the pseudo-illumination image E_w^* obtained by simply dividing L by R_w. The column (c) has the estimated illumination image $\hat{E}_w$ by nearest neighbor search in illumination eigenspace. (d) is the background image corresponding to the estimated illumination image $\hat{E}_w$ in (c). The nearest neighbor search in PCA is very robust to

Dimension	13	23	48	120
Contribution ratio (%)	84.5	90.0	95.0	99.0
NN Search time (μs)	6.7	6.8	7.9	12.0

Table 5.1. Dimension of the illumination eigenspace, Contribution ratio and NN search cost.

estimate the most similar illumination image E_w from noisy query E_w^*. However, there are slight differences in shadow shapes because the database is sparse. It is possible to acquire the exactly correct illumination image E_w when the database is dense enough, but it is not easy to prepare such a database. To solve this problem, we used our method to interpolate illumination images which is described in Section 4.1. Using k illumination images that are obtained by nearest neighbor search in the illumination eigenspace, we generate an appropriate illumination image using our shadow interpolation method based on shadow hull.

As for the computational cost, the average time of the nearest neighbor search is shown in Table 5.1 with MIPS R12000 300MHz, when the number of stored illumination images is 2048 and the image size is 360×243. Since the input image is obtained at the interval of $33ms$ (at 30 frames/sec), the estimation time is fast enough for the real-time processing. The average time of computing an intermediate shadow shape using shadow hull is also quite short. Once we prepared shadow hulls for a scene, the process is only computing a silhouette of the shadow hulls on the ground plane using the projection matrix which is determined by time when the image is captured. The average processing time to compute intermediate shadow shapes under one illumination condition in the intersection scene using our current research code is about $0.1sec$, and it would be undoubtedly improved when those matrix operations are handed off to graphics hardwares that have specialized matrix operation units.

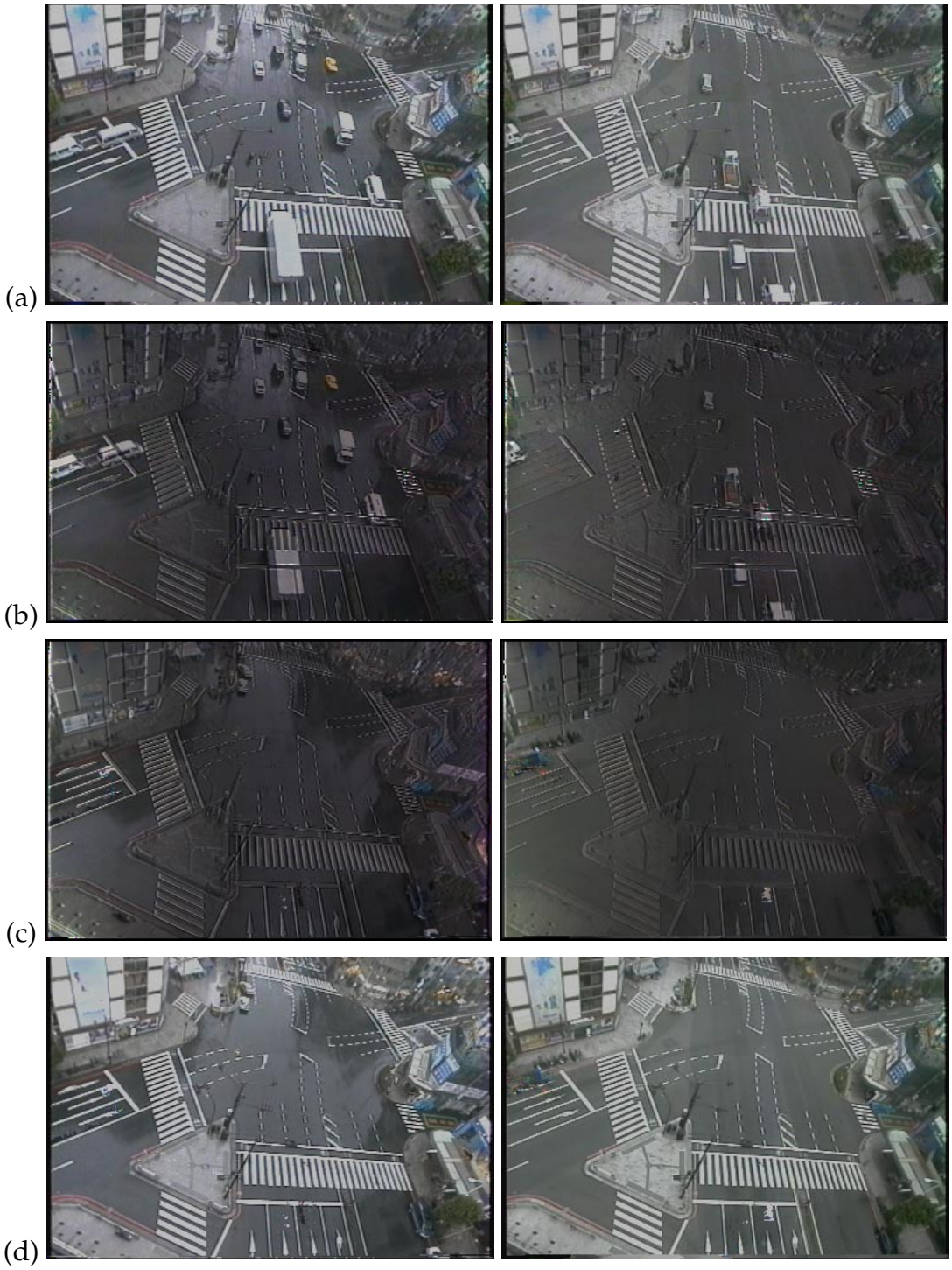


Figure 5.5. Direct estimation of intrinsic images (result 1). (a) An input image L, (b) the pseudo illumination image E_w^*, (c) the estimated illumination image $\hat{E}_w$ by the nearest neighbor search in the illumination eigenspace, (d) the corresponding background image B to (c).

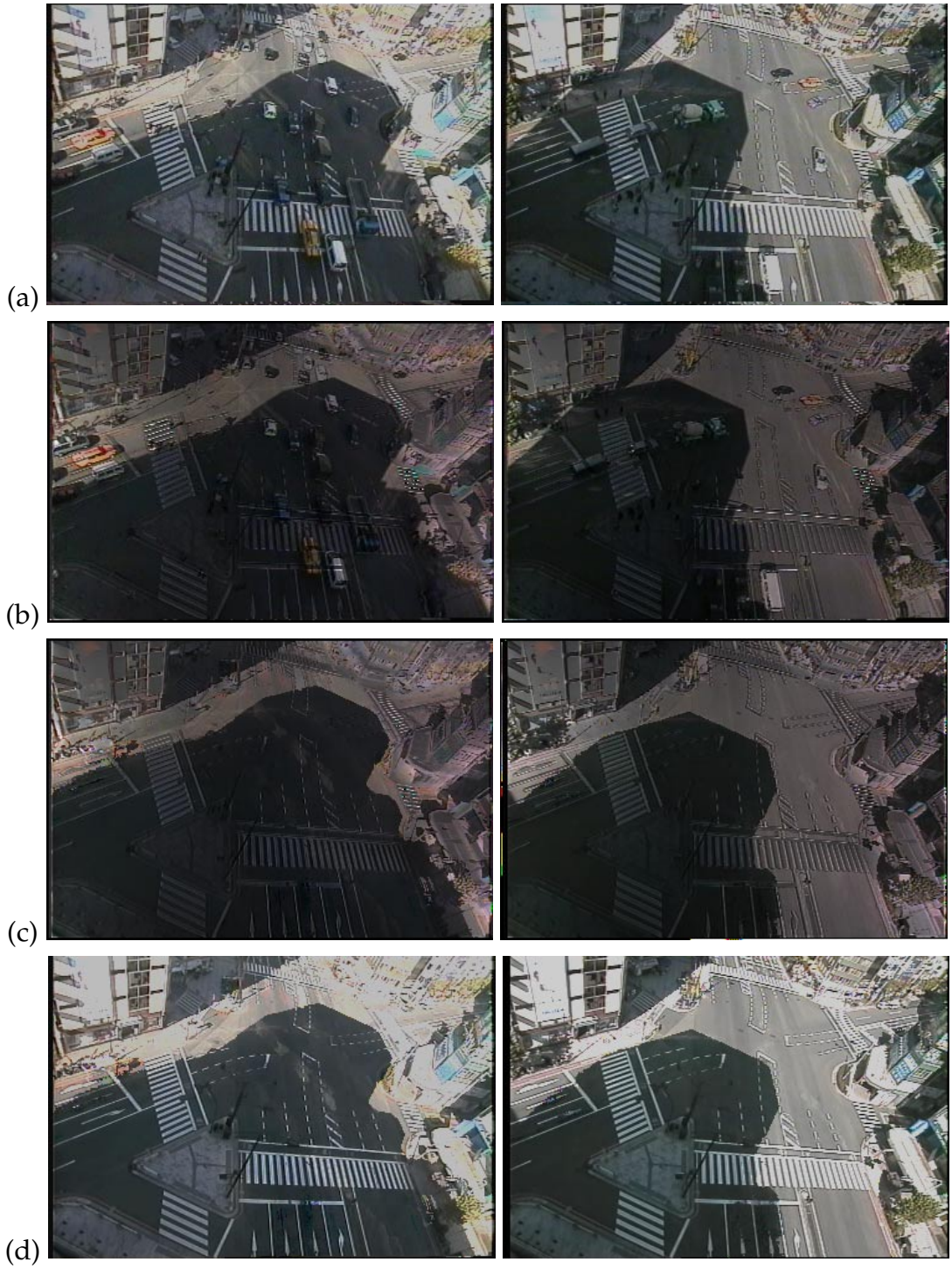


Figure 5.6. Direct estimation of intrinsic images (result 2). (a) An input image L, (b) the pseudo illumination image E_w^*, (c) the estimated illumination image $\hat{E}_w$ by the nearest neighbor search in the illumination eigenspace, (d) the corresponding background image B to (c).

5.5 Experimental results

To evaluate the effectiveness of our method, we preprocessed image sequences with our method and ran object tracking based on a simple block matching approach. The reason why we chose the block matching algorithm is to show even the simplest widely used tracking method can achieve good results after our illumination normalization preprocess. The block matching based tracking is done by pursuing the most similar window in the neighboring frame evaluated by Equation (5.5).

$$e_B(x,y) = min_{i,j}\Bigg\{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|f_t(x+m,y+n) - f_{t-1}(x+m+i,y+n+j)|\Bigg\} \tag{5.5}$$

In our tests, the maximal search distance of 10 pixels and window-size of 10×10 pixels were used. Since we focus especially on the advantage of our shadow elimination, we chose image sequences containing images of vehicles crossing boundaries of cast-shadows. The result is shown in Figure 5.7. In Figure 5.7, the row direction corresponds to the time axis from top to bottom. The first column of each pair and the second column of that represent results of the block matching based tracking applied to the original image sequence and image sequence with our preprocessing of illuminance normalization, respectively. Using the original image sequence (a), (b), (c), we get results where the block matching fails at shadow boundaries, because there is large intensity variation between inside and outside the shadow. On the other hand, after proper illuminance normalization using our method, the shadow boundary becomes seamless and the appearances of vehicles are preserved both inside and outside the cast shadow.

The outcome of the experiments over 502 sequences of vehicle tracking is shown in Table 5.2. In Table 5.2 $\mathrm{O}_{\mathrm{correct}}$ and $\mathrm{O}_{\mathrm{error}}$ indicate the number of correct tracking

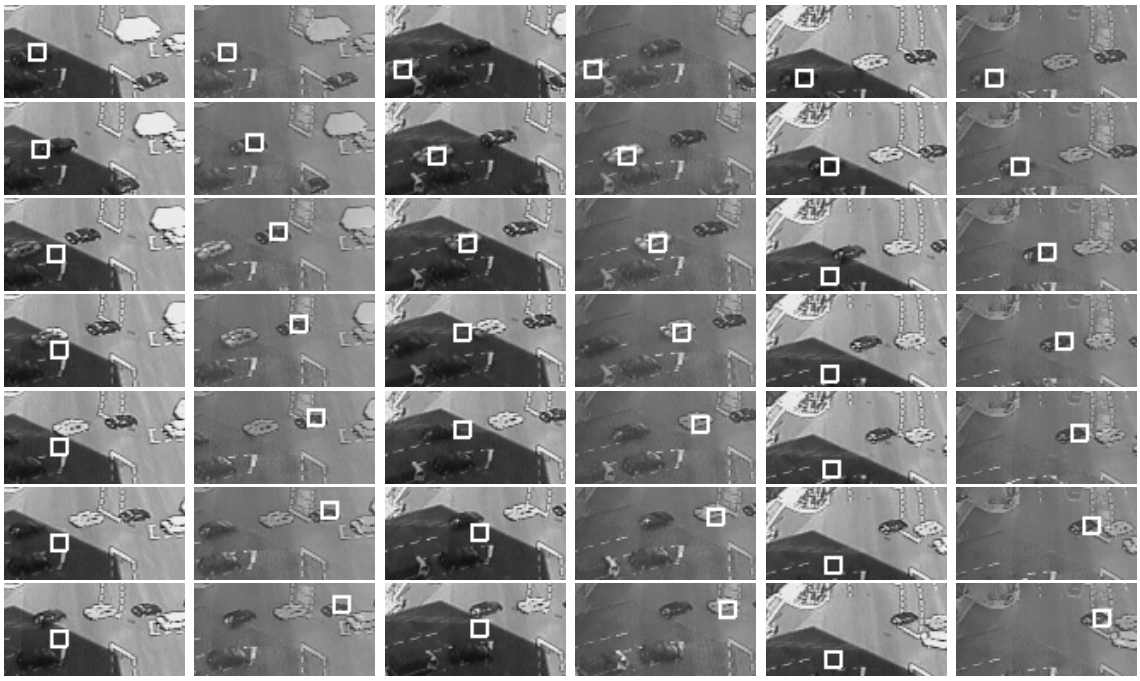


Figure 5.7. Result of tracking based on block matching. Along row from top to bottom it shows the frame sequence. The first column of each pair, (a), (b), (c), shows the tracking result over the original image sequence, and the second column of each pair, (a′), (b′), (c′), shows the corresponding result after our preprocessing.

results and error results with original input image sequences respectively. $\mathrm{N}_{\mathrm{correct}}$ and $\mathrm{N}_{\mathrm{error}}$represent the same but with normalized image sequences. The tracking performance with original input image sequences is 55.6%, while with normalized input it improved to 69.3%. The effectiveness of our method is clearly confirmed by looking at the $\mathrm{O}_{\mathrm{error}}$ column in the $\mathrm{N}_{\mathrm{correct}}$ row, which indicates the number of failure results with original image sequences while those were succeeded with normalized image sequences. It says 45.3% (101 sequences out of 223) were rescued by our method. On the other hand, 11.5% (32 sequences out of 279) got worse after applying our method. This bad effect happens typically when the shadow-edge cast on the vehicle surface largely differs from the shadow-edge in the illumination image. It happens because our method currently can handle only two-dimensional

shadow on the image plane, but the actual shadow is cast three-dimensionally on the scene. When the gap of the shadow-edge position is large, the error of the normalization is getting large, as a result, the block matching fails. Our system currently does not handle this problem since the error rate is small compared to the improved correct rate, but we are investigating on handling cast-shadows 3-dimensionally using the information of sunlight angle.

	$O_{correct}$	O_{error}	sum
$N_{correct}$	247	101	348
N_{error}	32	122	154
sum	279	223	502

Table 5.2. Tracking result over 502 sequences.

5.6 Summary

In this chapter, we have described a framework for normalizing illumination effects of real world scenes, which can be effectively used as a preprocess for robust video surveillance. We believe it provides a firm basis to improve the existing monitoring systems. We integrated the framework to the existing traffic monitoring system. We first described a method to normalize illumination effects from the input image sequence using intrinsic images which is basically an off-line process. Subsequently, we proposed to utilize *illumination eigenspace* as a key component of our framework, a pre-constructed database which captures the illumination variation of the target scene, to directly estimate illumination images for elimination of lighting effects of the scene including elimination of cast shadows. As for the intermediate illumination images that cannot be represented by linear combination of sampled illumination images, we combined our approach to use shadow hulls to accomplish non-linear interpolation of cast shadow regions using sunlight angles and camera parameters. The effectiveness of the proposed method is confirmed

by comparing the tracking results between the original image sequence and the image sequence preprocessed with our method. Since our method is used as a preprocessing stage, we believe this method can be applied to many video surveillance systems to increase robustness against lighting variations. Also, we have investigated direct estimation of illumination images corresponding to real scene images using the illumination eigenspace. Though our current implementation of the direct estimation in research code is not fast enough for real-time processing, we believe the framework has the potential to be processed in real-time.

CHAPTER 6

CONCLUSIONS

6.1 Summary

Elimination and interpolation of cast shadows have been one of the most difficult problems in the field of computer vision and graphics. Shadow elimination involves the problem of determining shadowed regions and its darkness. It has been difficult when the illumination condition is unknown. Separated from this problem, shadow interpolation or shadow morphing has also been difficult because it is hard to find correspondence between sampled shadows. In this dissertation, we have contributed a collection of concepts and algorithms to cope with these problems using intrinsic images.

- **Time-varying Intrinsic Images**

  We have investigated the time-varying reflectance images to properly handle the time-dependent characteristics of surface reflectance properties. We started from Weiss's ML estimation method and expand it to deal with the incorrect estimates that were originally inevitable with its single reflectance assumption. As a result, we successfully obtained the improved illumination images and time-varying reflectance images from an image sequence captured from a fixed view point but under several different illumination conditions.

- **Shadow Interpolation Framework**

  Shadow interpolation is a difficult problem because of two reasons. The first reason is that the motion of the shadow is unpredictable only from a set of sparsely sampled images. The second reason is that it is difficult to find correspondence between shadow blobs among images because shadows basically have no feature points. To overcome these difficulties, we proposed two different methods.

  One idea is to use shadow hulls constructed using shadowed regions in sampled illumination images associated with the sunlight angles in the outdoor scenes. Using the shadow hulls, the intermediate shadow shapes can be successfully computed and consequently the intermediate illumination images are robustly predicted. The framework is used to estimate the scene illumination images directly from an input image for robust video surveillance.

  The other approach is for the arena of computer graphics, specifically for image-based rendering techniques. Given a set of light fields captured under the different illumination conditions, we first compute rough scene geometry. The scene geometry is then used for computing global distortion of shadow to estimate the appearance of the scene under the intermediate illumination conditions.

- **Illumination Normalization Framework with Intrinsic Images**

  We proposed an idea to normalize the input image sequence in terms of illumination effects using illumination images. Normalization of illumination effects yields a lot of benefits as described in this thesis, e.g. illumination-free 2-D image editing for computer graphics, enhancing the accuracy of object segmentation and preserving appearance of moving objects inside and outside the cast shadow for robust video surveillance.

- **Practical Application to Road Traffic Monitoring Systems**

  As a practical application, we integrated the illumination normalization

framework into the existing road traffic monitoring system. To accomplish the real-time estimation of illumination images, our idea of illumination eigenspace is proposed. We evaluated the effectiveness by measuring error ratio of vehicle tracking based on a basic block matching technique and confirmed the illumination normalization framework remarkably contributed to increase the accuracy of vehicle tracking.

6.2 Future Directions

6.2.1 Modeling Illumination images

One potential application of this work is to construct meaningful model of illumination images using the illumination eigenspace. Although we have confirmed that the illumination eigenspace captures the key features of weather conditions and time-line variation, it is not clear whether we can create a novel scene illumination image using the illumination eigenspace. This is what we have to investigate, and for instance, the resulting application would enable the weather and time manipulation of the scene in terms of illumination conditions using only 2-D images.

Another interest is to investigate the generic model of the illumination variation of arbitrary scenes in 2-D images. Our current illumination normalization framework is scene specific, and it is necessary to start with estimating new intrinsic images when we apply our method to the new scene. It is still unclear whether there is the generic illumination model in 2-D images, however, extensive amounts of work remain to be done before this question is completely answered.

6.2.2 Estimating Intrinsic Images

Deriving intrinsic images from a set of images has not been solved yet. We are now investigating the estimation of intrinsic images starting with the formulation described in Section 2.4.1 and believe it would be solved in the near future.

6.2.3 Shadow Distortion Model

In this work, we investigated lighting interpolation for computer graphics in Section 4.2. To describe the shadow distortion, our approach used 2-D affine transformation on the image plane, however, it would clearly yield the better results using a more complex distortion model. For example, as proposed by Bregler *et al.* [BLCD02] in the different context, a combination of affine transformation to describe the global motion and another interpolation such as key-shape interpolation to represent the local motion would enhance the accuracy of resulting intermediate shadow shapes.

In addition, it is not clear yet how dense sampling of illumination images are required to archive shadow interpolation. Though it undoubtedly depends on the complexity of the scene, it should be confirmed at least with empirical study to give an idea on the parameter of sampling rate.

APPENDIX

CONVOLUTION AND REFLECTION

Convolution defines a way of combining two functions. In the most general form it is defined as a continuous integral. In one-dimension :

$$g(x) = \int_{-\infty}^{\infty} f(u)h(u-x)du \tag{A.1}$$

or in two-dimensions:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)h(u-x,v-y)dudv \tag{A.2}$$

The function $h(\bullet)$ in the above equations is usually called a filter. If both $f(\bullet)$ and $h(\bullet)$ are discrete, the convolution integral is simplified to:

$$g(i,j) = \sum_{m\in\Omega}\sum_{n\in\Omega} f(m,n)h(i-m,j-n) \tag{A.3}$$

Usually the convolution kernel $h(\bullet)$ only has non-zero value in a small neighborhood, and is also called a convolution mask. Rectangular neighborhoods are often used with an odd number of pixels in rows and columns, enabling the specification of the central pixel of the neighborhood.

There are some important mathematical properties of convolution. In the following explanation, $\otimes$ represents the convolution.

1. Commutative law $c = a \otimes b = b \otimes a$
2. Associative law $d = a \otimes (b \otimes c) = (a \otimes b) \otimes c = a \otimes b \otimes c$
3. Distributive law $d = a \otimes (b + c) = (a \otimes b) + (a \otimes c)$

where a, b, c and d are either continuous or discrete.

Recently, Ramamoorthi *et al.* [RH01] have shown that reflection is a unique type of convolution. They derive a convolution theorem for reflection, where the BRDF and the environment are represented by coefficients of Spherical Harmonics. This allows us to interpret reflection from a signal processing point of view, where the BRDF is a filter with a given frequency response, and the lighting is the input signal. This framework is particularly useful for inverse rendering, which may be formulated as a problem of deconvolution.

REFERENCES

[Ant93] A. Antoniou, *Digital filters: Analysis, design and applications*, McGraw Hill, New York, 1993.

[AP96] E. H. Adelson and A. P. Pentland, *The perception of shading and reflectance*, Perception as Bayesian Inference (D. Knill and W. Richards, eds.), Cambridge University Press, 1996, pp. 409–423.

[Bau74] B. G. Baumgart, *Geometric modelling for computer vision*, Ph.D. thesis, Stanford University, 1974.

[BF97] D. H. Brainard and W. T. Freeman, *Bayesian color constancy*, Journal of the Optical Society of America, vol. 14, 1997, pp. 1393–1411.

[BG01] S. Boivin and A. Gagalowicz, *Image-based rendering of diffuse, specular and glossy surfaces from a single image*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 2001, pp. 107–116.

[BLCD02] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande, *Turning to the masters: motion capturing cartoons*, Proceedings of Computer Graphics (SIGGRAPH), 2002, pp. 399–407.

[BM93] S. Beucher and F. Meyer, *The morphological approach to segmentation: The watershed transformation*, Mathematical Morphology in Image Processing (New York) (E. R. Dougherty, ed.), Marcel Dekker Inc., 1993, pp. 433–481.

[BMMG99] C. Buehler, W. Matusik, L. McMillan, and S. Gortler, *Creating and rendering image-based visual hulls*, Technical Report MIT-LCS-TR-780, MIT, 1999.

[BP98] J. Bouguet and P. Perona, *3d photography on your desk*, Proceedings of IEEE International Conference on Computer Vision, 1998, pp. 43–50.

[BT78] H. G. Barrow and J. M. Tenenbaum, *Recovering intrinsic scene characteristics from images*, Computer Vision Systems (A. Hanson and E. Riseman, eds.), Academic Press, 1978, pp. 3–26.

[BV99] J. S. De Bonet and Paul Viola, *Roxels : Responsibility weighted 3d volume reconstruction*, Proceedings of IEEE International Conference on Computer Vision, 1999, pp. 418–425.

[CF85] Y. Cheng and K. Fu, *Conceptual clustering in knowledge organization*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 7, 1985, pp. 592–598.

[Che95] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 17, 1995, pp. 790–799.

[CM97] D. Comaniciu and P. Meer, *Robust analysis of feature spaces: Color image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (San Juan, Puerto Rico), Jun. 1997, pp. 750–755.

[CM99] D. Comaniciu and P. Meer, *Mean shift analysis and applications*, IEEE 7th International Conference on Computer Vision, vol. 2, Sep. 1999, pp. 1197–1203.

[CRM00] D. Comaniciu, V. Ramesh, , and P. Meer, *Real-time tracking of non-rigid objects using mean shift*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, Jun. 2000, pp. 142–149.

[CTB92] Ian Craw, David Tock, and Alan Bennett, *Finding face features*, European Conference on Computer Vision, 1992, pp. 92–96.

[Deb98] P. E. Debevec, *Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography*, Proceedings of Computer Graphics (SIGGRAPH), Jul 1998, pp. 189–198.

[DGWH00] T. Darrell, G. Gordon, J. Woodfill, and M. Harville, *Integrated person tracking using stereo, color, and pattern detection*, International Journal of Computer Vision, vol. 2(37), Jun. 2000, pp. 175–185.

[EDD01a] Ahmed Elgammal, Ramani Duraiswami, and Larry S. Davis, *Efficient computation of kernel density estimation using fast gauss transform with applications for segmentation and tracking*, Second International Workshop on Statistical and Computational Theories of Vision, with The Eighth IEEE International Conference on Computer Vision (Vancouver, Canada), Jul. 2001.

[EDD01b] Ahmed Elgammal, Ramani Duraiswami, and Larry S. Davis, *Efficient non-parametric adaptive color modeling using fast gauss transform*, IEEE conference on Computer Vision and Pattern Recognition (Hawaii), 2001, pp. 563–570.

[EHD99] A. Elgammal, D. Harwood, and L. S. Davis, *Non-parametric model for background subtraction*, Proceedings of ICCV Frame-rate Workshop, Sep. 1999.

[EHD00] A. Elgammal, D. Harwood, and L. S. Davis, *Nonparametric background model for background subtraction*, Proceedings of the 6th European Conference on Computer Vision, vol. 2, 2000, pp. 751–767.

[FFB95] G. Finlayson, B. Funt, and J. Barnard, *Color constancy under a varying illumination*, Proceedings of the Fifth International Conference on Computer Vision, 1995.

[FGR93] A. Fournier, A. S. Gunawan, and C. Romanzin, *Common illumination between real and computer generated scenes*, Graphics Interface, May 1993, pp. 254–262.

[FHD02] Graham D. Finlayson, Steven D. Hordley, and Mark S. Drew, *Removing shadows from images*, Proceedings of European Conference on Computer Vision, vol. 4, 2002, pp. 823–836.

[FHH01] Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel, *Color by correlation: A simple, unifying framework for color constancy*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23(11), 2001, pp. 1209–1221.

[FR97] Nir Friedman and Stuart Russell, *Image segmentation in video sequences: A probabilistic approach*, Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence(UAI), Aug. 1997, pp. 175–181.

[Fuk90] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, Boston, 1990.

[FV98] W. T. Freeman and P. A. Viola, *Bayesian model of surface perception*, Advances in Neural Information Processing Systems(NIPS), vol. 10, Dec. 1998, pp. 787–793.

[GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, *The lumigraph*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 1996, pp. 43–54.

[GP93] J. M. Gauch and S. M. Pizer, *Multiresolution analysis of ridges and valleys in grey-scale images*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 15, Jun. 1993, pp. 635–646.

[GW87] R. C. Gonzalez and P. Wintz, *Digital image processing*, vol. 2nd Edition, AddisonWesley Publishing Co., Reading MA, 1987.

[Hay94] K. Hayakawa, *Photometric stereo under a light source with arbitrary motions*, Journal of the optical society of America, vol. 11(11), 1994, pp. 3079–3089.

[HHD99] T. Horprasert, D. Harwood, and L. S. Davis, *A statistical approach for real-time robust background subtraction and shadow detection*, Proceedings of ICCV Frame-rate Workshop, Sep. 1999.

[HM99] J. Huang and D. Mumford, *Statistics of natural images and models*, Proceedings of Computer Vision and Pattern Recognition, IEEE, 1999, pp. 541–547.

[Jai89] A. K. Jain, *Fundamentals of digital image processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[JMB91] J. M. Jolion, P. Meer, and S. Bataouche, *Robust clustering with applications in computer vision*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 13, 1991, pp. 791–802.

[KA86] Y. C. Kim and J. K. Aggarwal, *Rectangular parallelepiped coding: A volumetric representation of three dimensional objects*, IEEE Journal of Robotics and Automation, vol. RA-2, 1986, pp. 127–134.

[Kil92] Michael Kilger, *A shadow handler in a video-based real-time traffic monitoring system*, IEEE Workshop on Application of Computer Vision (Palm Springs, CA), 1992, pp. 1060–1066.

[KS87] J. R. Kender and E. M. Smith, *Shape from darkness: Deriving surface information from dynamic shadows*, Proceedings of IEEE First International Conference on Computer Vision, 1987, pp. 539–546.

[KS97] Norio Katayama and Shin'ichi Satoh, *The sr-tree: An index structure for high-dimensional nearest neighbor queries*, Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, 1997, pp. 369–380.

[KS00] K. N. Kutulakos and S. Seitz, *A theory of shape by space carving*, International Journal of Computer Vision, vol. 38(3), 2000, pp. 199–218.

[KSC01] S. B. Kang, R. Szeliski, and J. Chai, *Handling occlusions in multi-view stereo*, Conference on Computer Vision and Pattern Recognition, vol. 1, Dec. 2001, pp. 103–110.

[KvB90] K. P. Karmann and A. von Brandt, *Moving object recognition using an adaptive background memory*, Time-varying Image Processing and Moving Object Recognition (Elsevier, Amsterdam) (V. Cappellini, ed.), 1990, pp. 297–307.

[Lan77] E. H. Land, *The retinex theory of color vision*, Scientific American, vol. 237(6), Dec. 1977, pp. 108–128.

[Lau94] A. Laurentini, *The visual hull concept for silhouette-based image understanding*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 16(2), Feb. 1994, pp. 150–162.

[Lau95] A. Laurentini, *How far 3d shapes can be understood from 2d silhouettes*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 17(2), Feb. 1995, pp. 188–195.

[Lau99] A. Laurentini, *The visual hull of curved objects*, Proceedings of IEEE International Conference on Computer Vision, 1999, pp. 356–361.

[LDR00] C. Loscos, G. Drettakis, and L. Robert, *Interactive virtual relighting of real scenes*, IEEE Transaction on Visualization and Computer Graphics, vol. 6(4), 2000, pp. 289–305.

[LFG+01] S. Z. Li, Q. D. Fu, L. Gu, B. Scholkopf, Y. M. Cheng, and H. J. Zhang, *Kernel machine based learning for multi-view face detection and pose estimation*, Proceedings of 8th IEEE International Conference on Computer Vision (Vancouver, Canada), Jul. 2001, pp. 674–679.

[LFTG97] E. P. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg, *Non-linear approximation of reflectance functions*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 1997, pp. 117–126.

[LGTB97] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, *Face recognition: A convolutional neural network approach*, IEEE Transactions on Neural Networks, vol. 8(1), 1997, pp. 98–113.

[LH96] M. Levoy and P. Hanrahan, *Light field rendering*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 1996, pp. 31–42.

[LM71] E. H. Land and J. J. McCann, *Lightness and retinex theory*, Journal of Optic Society of America, vol. 61(1), 1971, pp. 1–11.

[MA83] W. N. Martin and J. K. Aggarwal, *Volumetric descriptions of objects from multiple views*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 5(2), Mar. 1983, pp. 150–174.

[Mar89] P. Maragos, *Pattern spectrum and multiscale shape representation*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 11, Jul. 1989, pp. 701–716.

[Mey93] F. Meyer, *Integrals, gradients, and watershed lines*, Proceedings of Workshop on Mathematical Morphology and its Applications to Signal Processing (Barcelona, Spain), May 1993, pp. 70–75.

[MG97] S. R. Marschner and D. P. Greenberg, *Inverse lighting for photography*, IS&T/SID 5th Color Imaging Conference, Nov. 1997, pp. 262–265.

[MG01] P. Meer and B. Georgescu, *Edge detection with embedded confidence*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, Dec. 2001, pp. 1351–1365.

[MGW01] T. Malzbender, D. Gelb, and H. Wolters, *Polynomial texture maps*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 2001, pp. 519–528.

[Mit98] S. K. Mitra, *Digital signal processing*, McGraw Hill, New York, 1998.

[MKL+02] Y. Matsushita, S. B. Kang, S. Lin, H. Y. Shum, and X. Tong, *Lighting interpolation by shadow morphing using intrinsic lumigraph*, Proceedings of the 10th Pacific Conference on Computer Graphics and Applications (Beijing, China), Oct. 2002, pp. 58–65.

[MNIS02a] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, *Handling shadow and illumination for video surveillance*, Proceedings of the First European Conf. on Color in Graphics, Image and Vision (CGIV '02) (Poitier, France), Apr. 2002, pp. 153–158.

[MNIS02b] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, *Realtime estimation of illumination images using illumination eigenspace*, Proceedings of IAPR Workshop on Machine Vision Applications (Nara, Japan), Dec. 2002, pp. 447–450.

[MNIS02c] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, *Robust object tracking with normalizing illumination*, Proceedings of IPSJ SIG-ICII (in Japanese) (Tokyo, Japan), 2002, pp. 43–50.

[MNIS02d] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, *Shadow removal for robust video surveillance*, Proceedings of IEEE Workshop on Motion and Video Computing(WMVC) (Orlando, FL), Dec. 2002, pp. 15–21.

[MWC01] P. Mendonca, K. Y. K. Wong, and R. Cipolla, *Epipolar geometry from profiles under circular motion*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23(6), Jun. 2001, pp. 604–616.

[NB93] S. K. Nayer and R. M. Bolle, *Computing reflectance ratios from an image*, Pattern Recognition, vol. 26(10), Oct. 1993, pp. 1529–1542.

[NB96] S. K. Nayar and R. M. Bolle, *Reflectance based object recognition*, International Journal of Computer Vision, vol. 17(3), 1996, pp. 219–240.

[NMN96] Shree K. Nayar, Hiroshi Murase, and Sameer A. Nene, *Parametric appearance representation*, Early Visual Learning (1996), 131–160.

[NSD94] J. Nimeroff, E. Simoncelli, and J. Dorsey, *Efficient re-rendering of naturally illuminated environments*, 5th Eurographics Workshop on Rendering, Jun. 1994, pp. 359–373.

[OK93] M. Okutomi and T. Kanade, *A multiple-baseline stereo*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 15(4), 1993, pp. 353–363.

[Pot87] M. Potmesil, *Generating octree models of 3d objects from their silhouettes in a sequence of images*, Computer Vision, Graphics and Image Processing, vol. 40, 1987, pp. 1–20.

[RBK98] H. A. Rowley, S. Baluja, and T. Kanade, *The retinex theory of color vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20(1), Jan. 1998, pp. 23–28.

[RH01] R. Ramamoorthi and P. Hanrahan, *A signal-processing framework for inverse rendering*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 2001, pp. 117–128.

[RIU] RIUL, *http://www.caip.rutgers.edu/riul/*.

[RMG98] Y. Raja, S. J. Mckenna, and S. Gong, *Colour model selection and adaptation in dynamic scenes*, Proceedings of 5th European Conference of Computer Vision, 1998, pp. 460–474.

[RMG99] Y. Raja, S. J. Mckenna, and S. Gong, *Tracking colour objects using adaptive mixture models*, Image Vision Computing, vol. 17, 1999, pp. 225–231.

[RMK95] C. Ridder, O. Munkelt, and H. Kirchner, *Adaptive background estimation and foreground detection using kalman-filtering*, Proceedings of International Conference on recent Advances in Mechatronics (ICRAM), 1995, pp. 193–199.

[SG99] Chris Stauffer and W. E. L Grimson, *Adaptive background mixture models for real-time tracking*, Proceedings of Computer Vision and Pattern Recognition (Fort Colins, CO), Jun. 1999.

[Sim] SimpleCCTV.com, *http://www.simplecctv.com/*.

[SK83] S. A. Shafer and T. Kanade, *Using shadows in finding surface orientations*, Computer Vision, Graphics, and Image Processing, vol. 22(1), 1983, pp. 145–176.

[SMO99] J. Stauder, R. Mech, and J. Ostermann, *Detection of moving cast shadows for object segmentation*, IEEE Transaction on Multi-Media, vol. 1(1), Mar. 1999, pp. 65–76.

[SP91] K. Shanmukh and A. K. Pujari, *Volume intersection with optimal set of directions*, Pattern Recognition Letter, vol. 12, 1991, pp. 165–170.

[SS95] P. Salembier and J. Serra, *Flat zones filtering, connected operators, and filters by reconstruction*, IEEE Transaction on Image Processing, vol. 4, Aug. 1995, pp. 1153–1160.

[SSI99a] I. Sato, Y. Sato, and K. Ikeuchi, *Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions*, International Conference on Computer Vision, Sep. 1999, pp. 875–883.

[SSI99b] I. Sato, Y. Sato, and K. Ikeuchi, *Illumination distribution from shadows*, Conf. on Computer Vision and Pattern Recognition, vol. 1, Nov. 1999, pp. 306–312.

[Str93] G. Strang, *Wavelet transforms versus fourier transforms*, Bulletin of the American Mathematical Society, vol. 28(2), 1993, pp. 288–305.

[SV] Surveillance-Video.com, *http://www.surveillance-video.com/*.

[SWI97] Y. Sato, M. Wheeler, and K. Ikeuchi, *Object shape and reflectance modeling from observation*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 1997, pp. 379–387.

[Sze93] R. Szeliski, *Rapid octree construction from image sequences*, CVGIP : Image Understanding, vol. 58(1), Jul. 1993, pp. 23–32.

[TKBM99] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, *Wallflower: Principles and practice of background maintenance*, Proceedings of International Conference on Computer Vision, 1999, pp. 255–261.

[TP91] M. Turk and A. Pentland, *Eigenfaces for recognition*, The Journal of Cognitive Neuroscience **3** (1991), no. 1, 71–86.

[TS98] C. Tzomakas and W. V. Seelen, *Vehicle detection in traffic secens using shadows*, Internal Report 98-06, Institut fur Neuroinformatik, 1998.

[Vin91] L. Vincent, *Watersheds in digital spaces: An efficient algorithm based on immersion simulations*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 13, Jun. 1991, pp. 583–598.

[Vin93] Luc Vincent, *Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms*, IEEE Transactions on Image Processing, vol. 2(2), Feb. 1993, pp. 176–201.

[WAA+00] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, *Surface light fields for 3D photography*, Proceedings of Computer Graphics (SIGGRAPH), Jul. 2000, pp. 287–296.

[WADP97] C. R. Wern, A. Azarbayejani, T. Darrell, and A. P. Pentland, *Pfinder: Real-time tracking of human body*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19(7), 1997, pp. 780–785.

[War92] G. J. Ward, *Measuring and modeling anisotropic reflection*, Proceedings of Computer Graphics (SIGGRAPH), Jul. 1992, pp. 265–272.

[WB82] G. West and M. H. Brill, *Necessary and sufficient conditions for von kries chromatic adaptation to give colour constancy*, Journal of Mathematical Biology, vol. 15, 1982, pp. 249–258.

[WC01] K. Y. K. Wong and R. Cipolla, *Structure and motion from silhouettes*, Proceedings of IEEE International Conference on Computer Vision, Jun. 2001, pp. 217–222.

[Wei01] Yair Weiss, *Deriving intrinsic images from image sequences*, Proceedings of the 9th IEEE International Conference on Computer Vision, IEEE, Jul. 2001, pp. 68–75.

[WFKM97] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph Malsburg, *Face recognition by elastic bunch graph matching*, Proceedings of 7th International Conference on Computer Analysis of Images and Patterns (Heidelberg) (Gerald Sommer, Kostas Daniilidis, and Josef Pauli, eds.), no. 1296, Springer-Verlag, 1997, pp. 456–463.

[WHON97] T. T. Wong, P. A. Heng, S. H. Or, and W. Y. Ng, *Image-based rendering with controllable illumination*, 8th Eurographics Workshop on Rendering, Jun. 1997, pp. 13–22.

[Woo78] R. J. Woodham, *Photometric stereo: A reflectance map technique for determining surface orientation from a single view*, Proceedings of Image Understanding Systems and Industrial Applications, SPIE 22nd Annual Technical Symposium, vol. 155, Aug. 1978, pp. 136–143.

[YDMH99] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, *Inverse global illumination: Recovering reflectance models of real scenes from photographs*, Proceedings of Computer Graphics (SIGGRAPH), Aug. 1999, pp. 215–224.

LIST OF PUBLICATIONS

Journals

[1] 松下 康之, 西野 恒, 池内 克史, 坂内 正夫, “時変イントリンシック画像とビデオサーベイランスへのその応用”, 電子情報通信学会論文誌（採録決定）.
[2] 上條 俊介, 松下 康之, 池内 克史, 坂内 正夫, “時空間 Markov Random Filed モデルによる隠れにロバストな車両トラッキング”, 電子情報通信学会論文誌 D-II, Vol.J83-D-II , No.12, pages 2597-2609, Dec., 2000.
[3] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, “Traffic Monitoring and Accident Detection at Intersections”, IEEE Transaction on ITS, Vol.1 No.2, pages 108-118, Jun., 2000.

Refereed Conferences

[1] 松下 康之, 西野 恒, 池内 克史, 坂内 正夫, “頑健なビデオサーベイランスのための照度不変画像列の生成”, 第一回 ITS シンポジウム, pages 451–458, Dec., 2002.
[2] Yasuyuki Matsushita, Ko Nishino, Katsushi Ikeuchi and Masao Sakauchi, “Realtime Estimation of Illumination Images Using Illumination Eigenspace”, IAPR Workshop on Machine Vision Applications, pages 447–450, Dec., 2002.
[3] Yasuyuki Matsushita, Ko Nishino, Katsushi Ikeuchi and Masao Sakauchi, “Shadow Elimination for Robust Video Surveillance”, IEEE Workshop on Motion and Video Computing, pages 15–21, Dec., 2002.
[4] Yasuyuki Matsushita, Sing Bing Kang, Stephen Lin, Heung-Yeung Shum and

Xin Tong "Lighting interpolation by shadow morphing using Intrinsic Lumigraphs", Pacific Graphics 2002, Beijing, China, pages 58–65, Oct., 2002.
[5] 松下 康之，西野 恒，池内 克史, 坂内 正夫, "イントリンシック画像を用いたイルミネーション画像のモデリングとその応用", 画像の認識, 理解シンポジウム (MIRU'02), vol. I, pages 253–260, Jul., 2002.
[6] Y. Matsushita, K. Nishino, K. Ikeuchi and M. Sakauchi "Handling Illumination and Shadow for Video Surveillance" Proc. of First European Conference on Color in Graphics, Imaging, and Vision(CGIV'02), pages 153–158, Apr., 2002.
[7] Y. Matsushita, M. Murao, S. Kamijo, K. Ikeuchi and M. Sakauchi, "Visualization of Traffic Activities", 8-th World Congress on Intelligent Transporting Systems, Sydney, Sep., 2001.
[8] M. Murao, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Visualization of Traffic Conditions for Drivers", International Workshop on Urban 3D/Multi-Media Mapping 2000(UM3 2000), Tokyo, CD-ROM, Sep., 2000.
[9] Yasuyuki Matsushita, Shunsuke Kamijo, Katsushi Ikeuchi and Masao Sakauchi, "Image Processing based Incident Detection at Intersections", In Proceedings of the Fourth Asian Conference on Computer Vision (ACCV 2000) , Taipei, Taiwan, pages 520-527, Jan., 2000.
[10] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Occlusion Robust Vehicle Tracking utilizing Spatio-Temporal Markov Random Field Model", 7th World Congress on ITS, Torino, CD-ROM, Nov., 2000.
[11] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Occlusion Robust Vehicle Tracking for Behavior Analysis utilizing Spatio-Temporal Markov Random Field Model", ITSC 2000, CD-ROM, Oct., 2000.
[12] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Traffic Monitoring at an Intersection utilizing Occlusion Robust Vehicle Tracking Method", The 3rd International Workshop on Urban 3D and Multi-Media Mapping(UM3'2000), Tokyo, Sep., 2000.
[13] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Occlusion Robust

Tracking utilizing Spatio-Temporal Markov Random Field Model", International Conference on Pattern Recognition(ICPR), Barcelona, Vol.1 pages 142–147, Sep., 2000.
[14] 上條 俊介, 松下 康之, 池内 克史, 坂内 正夫, "時空間 Markov Random Filed モデルによる隠れにロバストな車両トラッキング", 画像の認識, 理解シンポジウム (MIRU), 長野, Vol.II, pages 379–384, Jul., 2000.

Others

[1] 松下 康之, 西野 恒, 池内 克史, 坂内 正夫, "照度の正規化によるロバストな移動物体追跡", 第 3 回 知的都市基盤研究グループ研究発表会, Tokyo, pages 43–50, Jun., 2002.
[2] Y. Matsushita, M. Murao, S. Kamijo, K. Ikeuchi and M. Sakauchi, "Visualization of vehicle activities for traffic monitoring", The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Jul., 2001.
[3] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Research for Surveillance Technology on ITS", International Symposium on Multimedia Mediation Mechanism, Tokyo, Feb., 2000.
[4] 松下 康之, 上條 俊介, 池内 克史, 坂内 正夫, "移動物体が存在する環境下での背景画像の合成", 電子情報通信学会総合大会講演論文集, 広島, A–17–22, Mar., 2000.
[5] 上條 俊介, 松下 康之, 池内 克史, 坂内 正夫, "隠れマルコフモデルを応用した交差点における事故検出", 情報処理学会, CVIM 研究会, 99-CVIM-118, pages 45–52, Sep., 1999.
[6] 松下 康之, 上條 俊介, 池内 克史, 坂内 正夫, "交差点における交通事象把握", 情報処理学会第 59 回全国大会講演論文集, 5M–11, Sep., 1999.