REMOVING SHADOWS FROM COLOR IMAGES

by

Cheng Lu

B.Sc., University of Science and Technology, Beijing 1996

M.Sc., Simon Fraser University, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2006

APPROVAL

Name:	Cheng Lu
Degree:	Doctor of Philosophy
Title of thesis:	Removing Shadows from Color Images

Examining Committee:

Dr. Brian Funt
Chair

Dr. Mark S. Drew, Senior Supervisor,

Dr. Ze-Nian Li, Supervisor

Dr. Ghassan Hamarneh, Supervisor,

Dr. Greg Mori, SFU Examiner

Dr. Michael H. Brill, External Examiner,
Datacolor Inc.

Date Approved: Aug. 4, 2006

Abstract

This thesis is concerned with the derivation of a shadow-free image representation. We propose several methods of automatically removing and detecting shadows in a color image.

Our methods stem from the illuminant invariant theory which requires a camera calibration step to find the direction of illumination changes. In our work, instead of a camera calibration we aim at finding this direction from evidence in the color image itself. Specifically, we recognize that producing a 1-d projection in the correct invariant direction which is orthogonal to the direction of changes of illumination, will result in a 1-d distribution of pixel values that have smaller entropy than projecting in the wrong direction. To be able to develop an effective description of the entropy-minimization task, we go over to the quadratic entropy, rather than Shannon's definition. Replacing the observed pixels with a kernel density probability distribution, the quadratic entropy can be written as a very simple formulation, and can be evaluated using the efficient Fast Gauss Transform. The entropy is more insensitive to quantization than is the usual definition. The shadow removal step produces good shadow-free color image results whenever strong shadow edges are present in the image. In almost every case studied, entropy has a strong minimum for the invariant direction, revealing a new property of image formation.

Shadow detection per se is an important step in image analysis. We propose a method of detecting not just strong shadow edges but indeed entire shadow regions in the image taken under ambient light. We argue that the difference in a log domain of the flash image and the ambient image gives a very simple feature space consisting of two components - one in an illuminant-change 3-vector direction, and one along the gray axis. This space provides excellent separation of the shadow and nonshadow areas.

We propose a method for efficient ambient illuminant estimation using the flash image. We design a novel white balance scheme which uses the white patch under the estimated illuminant as reference white color for balancing images.

Keyword: Shadow, color image processing, vision, illuminant, digital camera.

To my family.

Acknowledgments

I am truly grateful to my supervisor, Dr. Mark S. Drew, for his continuous support, guidance, encouragement and patience throughout my Ph.D. work. Thank you so much, Mark, for giving me so much help to both my study and my life in Canada.

Thanks go to Dr. Ze-Nian Li for his providing such good research environment and sharing his working and life experience with me. Thanks go to Dr. Michael Brill for his time and hard work on being careful enough to check up every detail of my thesis, and to Drs. Ghassan Hamarneh and Greg Mori for their comments and suggestion to my thesis. My thanks also go to Dr. Graham Finlayson for his great collaboration in these years from which I learned a lot.

I would like to thank all my colleagues and friends at Computing Science School whom I enjoyed working with.

Most of all, heartfelt thanks go to my mum, dad and sisters all those miles away in China for their encouraging and supporting me so many years since the first day of my education in Canada. I am heartily grateful to my Dad and mum-in-law, this thesis would not have been possible without their encouragement, support and especially taking care of my new born baby twins. Finally, I express my deepest love to my wife Xu, Jia and my twin daughters Amber and Breanna.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

One of the most fundamental tasks for any visual system is that of separating the changes in an image which are due to a change in the underlying imaged surfaces from changes which are due to the effects of the scene illumination. The interaction between light and surface is complex and introduces many unwanted artifacts into an image. For example, shading, shadows, specularities, and inter-reflections as well as changes due to local variation in the intensity or color of the illumination all make it more difficult to achieve basic visual tasks such as image segmentation [75], object recognition [109], and tracking [71]. The importance of being able to separate illumination effects from reflectance has been well understood for a long time. For example, Barrow and Tenenbaum [6] introduced the notion of "intrinsic images" to represent the idea of decomposing an image into two separate images: one which records variation in reflectance and another which represents the variation in the illumination across the image.

Barrow and Tenenbaum proposed methods for deriving such intrinsic images under certain simple models of image formation. In general however, the complex nature of image formation means that recovering intrinsic images is an ill-posed problem. More recently, Weiss [115] proposed a method to derive an intrinsic reflectance image of a scene given a sequence of images of the scene under a range of illumination conditions. Using many images ensures that the problem is well-posed but implies that the application of the method is quite restricted. The Retinex and Lightness algorithms of Land [77] and others [9, 12, 66, 69] can also be seen as an attempt to derive intrinsic reflectance images under certain restrictive scene assumptions. Specifically, those algorithms are founded on the premise that scenes are 2-d planar surfaces constructed from a tessellation of uniform reflectance patches. In addition, the intensity of illumination across the scene is assumed to vary only slowly and is assumed to be spectrally constant. Under these conditions it is possible to distinguish changes in reflectance from changes in illumination and to factor the latter out, thus deriving an intrinsic reflectance image referred to as a lightness image in those works.

Estimating and accounting for the color of the prevailing scene illumination is a related problem which has received much attention [10, 30, 45, 76, 83]. In this body of work the focus is not on deriving intrinsic reflectance images but rather on obtaining a rendering of a scene as it would appear when viewed under some standard illumination. Often these color constancy algorithms, as they are called, are derived under the same restrictive conditions as the lightness algorithms and factors such as specularities, shading, and shadows are ignored. A different approach to this problem is the so-called illuminant invariant approach [53,59]. Instead of attempting to estimate the color of the scene illuminant, illuminant invariant methods attempt simply to remove its effect from an image. This is achieved by deriving invariant quantities — algebraic transformations of the recorded image values — which remain constant under a change of illumination. Methods for deriving quantities which are invariant to one or more of illumination color, illumination intensity, shading, and specularities have all been proposed in the literature.

In this thesis we consider how we might account for shadows in an imaged scene: an illumination which has so far largely been ignored in the body of work briefly reviewed above. That accounting for the effect of shadows on color constancy in images has not received more attention is somewhat surprising since shadows are present in many images and can confound many visual tasks. As an example, consider that we wish to segment the image in Figure. 1.1 into distinct regions each of which corresponds to an underlying surface reflectance. While humans can solve this task easily, identifying two important regions corresponding to the grass and the path, such an image will cause problems for a segmentation algorithm, which will quite likely return at least three regions corresponding to shadow, grass, and path. In fact, identifying shadows and accounting for their effects is a difficult problem since a shadow is in effect a local change in both the color and intensity of the scene illumination. To see this, consider again Figure. 1.1. In this image, the non-shadow region is illuminated by light from the sky and also by direct sunlight, whereas in contrast the shadow region is lit only by light from the sky. It follows that to account for shadows we must be able in effect to locally solve the color constancy problem — that is, identify the color of the scene illuminant at each pixel in the scene.

In light of this problem, a method was devised [40] which showed that given certain assumptions about scene illumination and camera sensors it is possible to solve the color constancy problem at

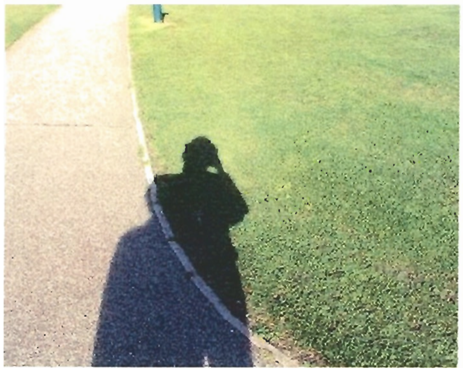


Figure 1.1: A color image which consists of shadows.

a single image pixel. That is, given a single triplet of sensor responses it is possible to derive a 1-d quantity invariant to both the color and intensity of the scene illuminant. This in effect provides a 1-d reflectance image which is therefore shadow-free. The method devised finds an intrinsic image, originally 1-d grayscale but subsequently 2-d chromaticity, based on assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrowband camera sensors. Nevertheless, the method works well when these assumptions do not hold. A crucial piece of information is the angle for an "invariant direction" in a log-chromaticity space. Originally, this information was gleaned via a preliminary calibration routine, using the camera involved to capture images of a color target under different lights. Subsequently, it was shown in principle [38] that we can in fact dispense with the calibration step, by recognizing a simple but important fact: the correct projection is *that which minimizes entropy* in the resulting invariant greyscale image. In this thesis, the entropy based method is examined in detail, and in order to carry out an efficient search over smooth values that are not subject to quantization problems, we replace the Shannon's entropy measure, used previously, by a Quadratic Entropy measure such that a Gaussian mixture model of the probability density function (pdf) produces an analytic formula. We show that such quadratic entropy values are much smoother and usually produce only a single minimum, making this approach the most efficient. The quadratic entropy can be evaluated in linear time using a Fast Gauss Transform, leading to a simple method for finding the invariant direction.

Given the derived intrinsic image, we recover a full-color 3-d image representation which is the same as the original image but with shadows removed. Here our approach is similar to that taken in lightness algorithms [9, 66, 69, 77]. In that work the effects of illumination are factored out by working with an edge representation of the image, with small edges assumed to correspond to the slowly changing illumination while large changes correspond to a change in reflectance. Under these assumptions, small changes are factored out and the resulting edge-map is re-integrated to yield an illumination-free lightness image. In our case we also work with an edge-map of the image but we are concerned with separating shadow edges from reflectance edges and factoring out the former. To do so we employ the 2-d shadow-free chromaticity. We reason that a shadow edge corresponds to any edge which is in the original image but absent from the invariant representation, and we can thus define a thresholding operation to identify the shadow edge. Edges are in-painted across the shadow edge, and re-integrating yields a color image, equal to the original save for the fact that it is shadow-free.

The problem existing in recovering a full-color shadow-free image is that it is difficult to accurately identify the shadow edges. Subtracting the edges in the original image from those in the invariant image is not always able to give complete shadow edge map. To solve this problem, we develop a new method to identify shadows in a color image via flash/no-flash image pairs. This is accomplished by employing a log difference color space in which a simple, diagonal, model of illumination change applies. In this space, in-shadow pixels will have a large illumination difference between the flash and no-flash image than the out-of-shadow pixels, and shadow region can be identified using Least Median of Squares method which automatically gives outliers as shadows.

Applications using flash/no-flash image pairs use the combination of the two images. For consumer-grade digital cameras, due to the different illumination conditions the two images usually have different camera settings such as exposure time and white balance. So prior to combining the two images, the difference caused by different camera settings should be compensated such that the difference between compensated images reflects only the difference in illumination.

Shadows are accounted for as a change of scene illuminants. Estimating the color of the scene illuminant is an important but yet difficult problem for color constancy. An immediate application of having a good estimate of the scene illuminant is being able to adjust the "white balance", i.e., adjusting each overall color channel so as to remove any noticeable color cast. In view of this need, we present a practical design for estimating the main image illuminant, again using flash/no-flash image pairs. In particular, the flash image is used as a reference light, and then the ambient illuminant in the no-flash image is classified into different illuminant temperature groups. We show that these are distributed along a straight line in a log-difference chromaticity space. Furthermore, in another novel approach, white balance can be easily performed using the color of the white patch in a lookup table of calibration color target images for each illuminant.

1.2 Thesis Arrangement

In Chapter 2, we present background material on previous work that relates to the central topic of this thesis. Specifically, we briefly describe research in color constancy which is the basis of the illuminant invariant theory, previous work in detecting and removing shadows from images, and finally related applications of flash/no-flash image pairs.

In Chapter 3, we outline the motivation for a projection-based definition of an illuminant invariant, and set out the relevant equations in the subsequent chapter.

In Chapter 4, we aim at finding the projection direction from evidence in the color image itself instead of a camera calibration. We seek that projection which minimizes entropy, and from there go on to remove shadows. To be able to develop an effective description of the entropy-minimization task, we go over to the quadratic entropy, rather than Shannon's definition.

In Chapter 5, we present an algorithm for deriving the 3-d shadow-free image. We develop a shadow edge map extraction algorithm and set out image re-integration, which carries out a solution of the Poisson equations using the Fourier transform.

In Chapter 6, we present a shadow-matte algorithm via flash/no-flash image pairs. We show that illumination change can be identified through a log color difference space. Furthermore, we develop a method to estimate the ambient illuminant using the flash image as a reference light.

In the last chapter, Chapter 7, the contributions of this thesis, along with a discussion of possible future work based on the this thesis, will be presented.

Chapter 2

Background and Related Work

2.1 Color, Color Constancy and Color Invariant

It is well known that the colors in a scene do not depend only on the surface properties of the objects present but also on the illumination conditions. Indeed, this dependence is so strong that computer vision techniques based on color, such as object recognition and tracking, may fail because of this circumstance. Common illuminants, such as daylight and artificial light, differ markedly in their spectral properties; and since the spectrum of the light impinging on a camera is the product of the illumination and spectral reflectance of the surface, the illumination must be accounted for and discounted for stable representation of surface color. Humans have a remarkable ability to discount the illumination when judging object appearance. This ability, called color constancy, demonstrates at least a subconscious ability to separate the illumination spectral-power distribution from the surface reflectance function within the color signal. Color constancy approaches have been widely used to cancel out the effect of illumination. These mainly proceed in one of two directions: estimating surface spectral reflectance, or estimating illuminant color. Another group of methods focus on a color invariant description, and many different approaches have been reported in the literature. Most of these techniques aim at finding features from color images that are robust to changes of illumination conditions, including lighting geometry, intensity of illumination, and illumination color.

2.1.1 The Properties of Color

Color is usually caused by the vision system responding differently to different wavelength of light. In this section we wish to describe typical properties of colored surfaces and colored light sources.

The Color of Light Sources

Light is the radiation we can see. Light can be described by its wavelength, for which the nanometer (nm) is a conventional unit of measure. By dispersing light into a spectrum, any source can be described in term of the *power* (amount of light) emitted at each wavelength. Plotting this power as a function of the wavelength gives the *spectral power distribution* (SPD) curve of the light source. Sources of light include: Black Body Radiators, sun, sky, and artificial illuminants.

One important light source is called a black body. When cold, a black body looks black and reflects no light. It emits electromagnetic radiation when heated. The spectral power distribution of a hot black body can be measured by heating hollow metal chambers [46]. On being heated, blackbodies glow like metals, first a dull red like a hot electric stove, then become progressively brighter and whiter, like the filaments of incandescent lamps. Planck's Law states that the spectral power distribution of this radiation depends only on the temperature of the body. If we write T for the temperature of the body in degrees Kelvin, h for Planck's constant, k for Boltzmann's constant, c for the speed of light, and λ for the wavelength, we have

$$E(\lambda) \propto \frac{1}{\lambda^5} \frac{1}{(exp(hc/kT\lambda) - 1)} \tag{2.1}$$

This defines the spectral concentration of radiant excitance, in watts per square meter per wavelength interval, as a function of wavelength λ (in meters) and temperature T (in degrees Kelvin).

This means that there is a one-parameter family of light color corresponding to black body radiators — the parameter being the temperature — and so we talk about the *color temperature* of a light color. Figure 2.1 shows *relative spectral power distributions* for three blackbodies, at 2865, 5000, and 10,000K respectively. (Spectral power distributions are usually normalized at 560nm, resulting in relative spectral power distribution.) It is evident (and very well-known) that as the temperature increases, the spectrum of light moves from reddish to whitish to bluish. The range 2500 to 10,000K is the most important region in terms of modeling typical illuminant colors.

The most important natural light source is the sun. The sun is usually modeled as a distant, bright point. The sky is another important natural light source. The sky is bright because light from the sun is scattered by the air, plus of course any interreflections. A patch of surface outdoors

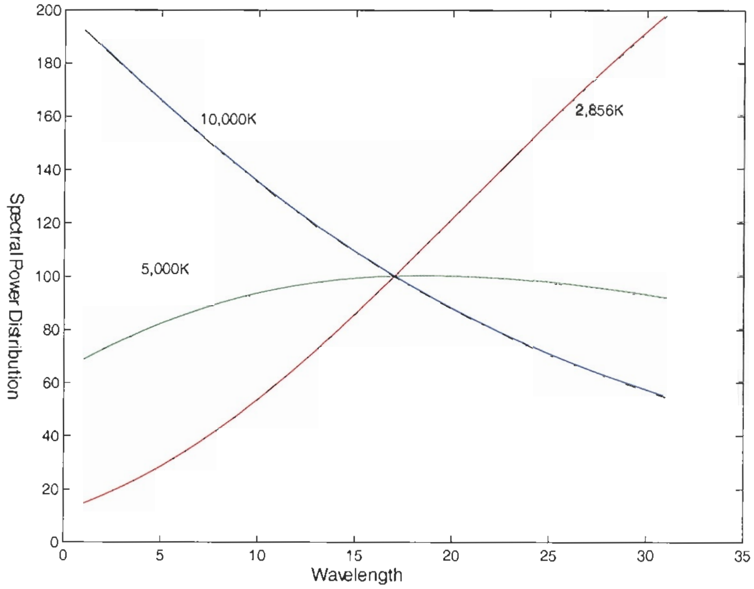


Figure 2.1: The relative spectral power distribution of blackbody (or Planckian) radiators with color temperature 2,856K, 5,000K, and 10,000K, representing the range of color temperatures of interest in the color problem: the curve for 2856K is a typical SPD of a filament lamp; the curve for 5000K is in the color range of sunlight, 10000K is the color of some computer monitors [120].

during the day is illuminated both by light that comes directly from the sun and by skylight from sun that has been scattered by the air, plus of course any interreflections amongst objects. A crude geometrical model of skylight is a hemisphere with constant radiance. For clear air, the intensity of radiation scattered by a unit volume depends on the fourth power of the frequency; this means that light with long wavelength can travel very much further before being scattered than light with short wavelength. It follows that, when the sun is high in the sky, blue light from sun is scattered out of the ray so that the sky looks blue and the sun looks yellow. The color of daylight varies with the time of day and the time of year (see Figure 2.2). These effects have been widely studied.

Typical artificial light sources are commonly of a few types [46]. *Incandescent light* arises from a metal filament which is heated to a high temperature. The spectrum roughly follows the blackbody law. *Fluorescent lights* work by generating high speed electrons that strike gas within the bulb; this in turn releases ultraviolet radiation which cause phosphors to fluoresce. In some bulbs, light is produced by electrons in metal atoms dropping from an excited state to a lower energy state. Two

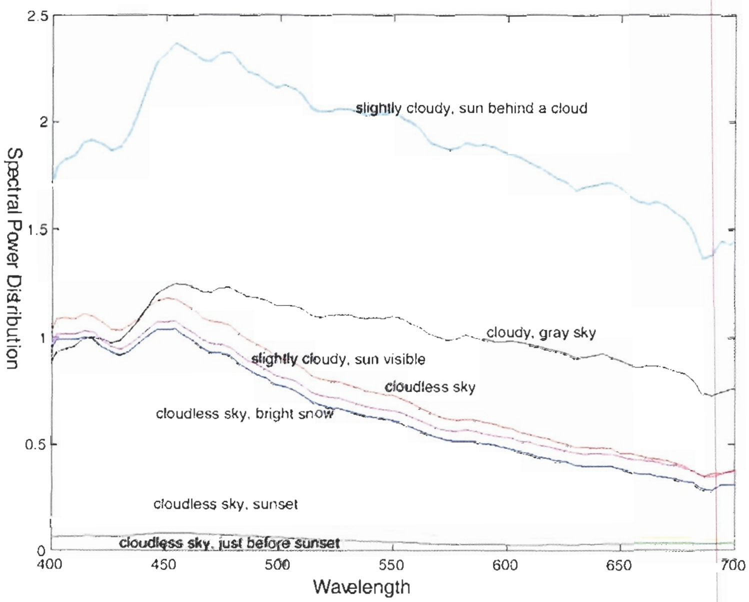


Figure 2.2: Spectral power distributions of seven different daylights measured at different times of day and under different conditions, due to Jussi Parkkinen and Pertti Silfsten. Plot from data obtainable at *http : //www.it.lut.fi/research/color/lutcs_database.html.*

common cases are *sodium arc lamps*, and *mercury arc lamps*. Sodium arc lamps produce a yellow-orange light, and are commonly used for freeway lighting. Mercury arc lamps produce a blue-white light, and are often used for security lighting.

Many nonblackbody light sources, such as sun and sky lights, can be described by the color temperature of the blackbody that they most nearly resemble visually. This is called their *correlated color temperature.*

A number of spectral power distributions have been defined by the International Commission on Illuminant (CIE) for use in describing color. These distributions are known as standard illuminants. Incandescent light is represented by illuminant A, equivalent to a blackbody radiator with a color temperature of 2856K. Natural daylight is defined by the D illuminants. D65 is the most commonly one adopted in industries to represent daylight with a correlated color temperature of 6500K. F series illuminants define a range of fluorescent lamps. Figure 2.3 shows spectral power distributions of illuminants A and D65.

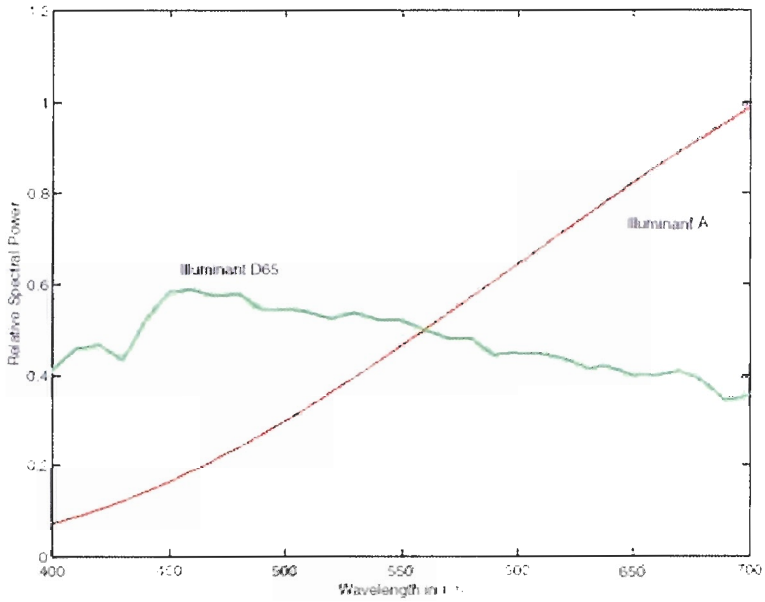


Figure 2.3: Spectral power distributions of two standard CIE models: illuminant A—which models incandescent lamps; illuminant D65—models daylight. Plot from data due to Andrew Stockman and Lindsy Sharpe, and available at $http : //cvision.ucsd.edu/cie.htm$

The Color of Surfaces

The color of surfaces is a result of a variety of mechanisms, including different absorption at different wavelengths, refraction, diffraction, and bulk scattering (for more detail, see [82]). These effects are bundled into a macroscopic BRDF model, which typically is a Lambertian plus specular approximation. The terms of surface color are now *spectral reflectance* (sometimes abbreviated *reflectance*) or (less commonly) *spectral albedo*. Figure 2.4 shows examples of spectral reflectances for a number of different natural objects.

BRDF The most general model of local reflection is the *bidirectional reflectance distribution function*, usually abbreviated BRDF. The BRDF is defined as the ratio of *spectral radiance* in the outgoing direction to the *spectral irradiance* in the incident direction.

Lambertian surface A Lambertian surface has a constant BRDF, which means the radiance leaving the surface is independent of outgoing direction. Such surfaces will look equally bright from any viewing direction. A Lambertian surface is also called an *ideal diffuse surface*.

Specular surface Specular surfaces are glossy or mirror-like surfaces. An ideal specular surface behaves like an ideal mirror. Radiance can leave only along *specular direction*, obtained by reflecting the direction of incoming radiation about the surface normal. If the view direction is not same as the specular direction, the camera will not see specular reflected light.

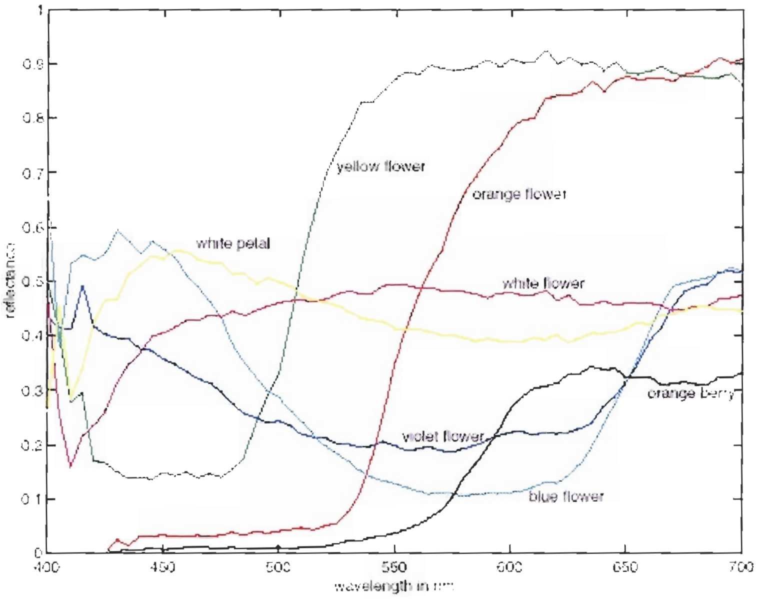


Figure 2.4: Spectral reflectances for several different leaves, a color name is given for each, measured by Esa Koivisto, data available at $http : //www.it.lut.fi/research/color/lutcs\_database.html$

Formation of Image Color

The interaction of light and reflection can be extremely complex. Factors such as interreflection and specularities are often difficult to model and deal with. To simplify matters we will initially consider a color image model which ignores specularities. In later discussion, we will relax the model and will allow specularities.

The color of light arriving at a camera is determined by two factors: first, the spectral reflectance of the surface; and second, the spectral radiance of illumination. If a patch of perfect Lambertian surface with spectral reflectance $S(\lambda)$ is illuminated by a light whose spectral power distribution is $E(\lambda)$, the spectrum of the reflected light is $S(\lambda)E(\lambda)$. Now we consider Lambertian shading. For a Lambertian surface, since its BRDF is constant and independent of outgoing directions, the image intensities of the surface only change with the illumination directions. Suppose n is the normal of the surface patch, and a is the illumination direction. Then, the shading term σ of such a patch is $n \cdot a$.

Thus, if a linear photoreceptor of the kth type sees this surface patch, its response is:

$$R_k = \sigma \int_{\omega} E(\lambda)S(\lambda)Q_k(\lambda)d\lambda \tag{2.2}$$

where ω is the range of all relevant wavelengths, and $Q_k(\lambda)$ is the sensitivity function of the kth photoreceptor.

The color of light falling on the surface can vary widely, from blue fluorescent light indoors to orange or even red light at sunset. Thus the color of the light arriving at the camera can be quite a poor representation of the surface color. Especially, as the illumination color changes, so too does the color of the light reflected from the surface and as a consequence different RGBs are recorded for the same surface viewed under different lights.

Generally, two approaches are reported in literature for dealing with these illumination dependencies. One the color constancy algorithms; the other is color invariant description approaches.

2.1.2 Color Constancy

The term color constancy has been coined to describe the ability to make statements about surface color properties which are independent of changes of illumination conditions. This means, given an image under unknown illuminants, an image processing system can assign colors to the scene despite changes in the illumination. This capacity is important for correcting color in photography, and computer vision applications.

Finite-dimensional Linear Model

The description of illuminant spectral power distribution functions and surface reflectance functions can be further simplified using finite-dimensional models. A surface reflectance $S(\lambda)$ can be approximated as:

$$S(\lambda) = \sum_{i=1}^{d_S} S_i(\lambda)r_i \tag{2.3}$$

where $S_i(\lambda)$ is the basis functions for the model of the reflectance, and the r is a d_S — component column vector of weights. The dimension required for large samples of measured surfaces lies somewhere between 3 and 8. Similarly, if a finite-dimensional linear model of the illuminant is a reasonable model, any illuminant can be written as

$$E(\lambda) = \sum_{j=1}^{d_E} E_j(\lambda)e_j \tag{2.4}$$

where the $E_j(\lambda)$ is the basis functions for the model of reflectance, and e is a d_E dimensional vector of weights. Judd [22] measured 605 daylight illuminants and showed that they are well modeled by a set of 3 basis functions.

Basis functions are generally chosen by performing a principal component analysis of each data set (reflectances and illuminants) in isolation [17,84]. However, this type of analysis is inappropriate in that it does not take into account how illuminant, reflectance and sensor interact. In [23] it is argued that the basis functions should be found relative to the sensors used. They propose using the sensor responses to find basis functions for surface reflectances or illuminants, which are referred to as one-mode analysis. They conclude that a 3-dimensional basis set for surface reflectance and a 3-dimensional basis set for illuminant is sufficient to model the Munsell chips [21] under a wide range of blackbody radiator illuminants.

If we realize that a population of spectra lie within such low dimensional linear model, then we can parameterize the spectra by specifying the model weights.

Substituting those two linear models into the measurement Eq. 2.2, the response of a camera sensor of the kth type is

$$R_k = \int_\omega \left(\sum_{j=1}^{d_E} E_j(\lambda)e_j\right)\left(\sum_{i=1}^{d_S} S_i(\lambda)r_i\right) Q_k(\lambda)d\lambda \tag{2.5}$$

We can see the measurement R_k at kth receptor is a combination of the coefficients of the illuminants and reflectance.

The responses from all the types of sensors (typically 3, like human cones) are put in one response vector p, the coefficients of the illuminant in e vector(dimension d_E) and the coefficients of reflectance in r vector(dimension d_S). There is a linear relation between p and r and Eq. 2.5 can be rewritten as a matrix transform. A lighting matrix $\Lambda(e)$ maps reflectances, defined by the r vector, onto a corresponding response vector:

$$p = \Lambda(e)r \tag{2.6}$$

where $\Lambda(e)_{ij} = \int_\omega Q_i(\lambda)E(\lambda)S_j(\lambda)d\lambda$ is the component at ith row, jth column in matrix Λ. The lighting matrix is dependent on the illuminant weight vector e. The roles of illumination and reflectance are symmetric. There is also a linear relation between p and e. We can write a sensor response vector as a surface matrix transform Ω and illumination weight vector e:

$$p = \Omega(r)e \tag{2.7}$$

where $\Omega(r)_{ij} = \int_\omega Q_i(\lambda)E_j(\lambda)S(\lambda)d\lambda$ is the component at ith row, jth column in matrix Ω. The surface matrix is dependent on the surface weight vector r.

Finite-dimensional linear models are an important tool for color constancy. Maloney and Wandell [78, 84] established a general result for recovering reflectance and illumination coefficients r and e: a trichromatic system that views surfaces under an unknown illuminant can recover two reflectance descriptors per surface. In short, if the illuminant is known, the solution is trivial, we just need to invert the lighting matrix. If the illumination is not known but we have sensor responses at s points, assuming that the illuminant does not vary substantially between points, then we have 3 equations at each point(3s total) for $d_S s + d_E$ unknowns. This can be solved only for $d_S = 2$ as long as $s > d_E$. It usually assumes that illumination is 3 dimensional, that is $d_E = 3$. This set of constraints is known as 3-2 model. Under this 3-2 model, the lighting matrices $\Lambda(e)$ are 3×2 injective maps (i.e., color vectors are linear combinations of the two column vectors of $\Lambda(e)$), and surfaces seen under a single illuminant span a plane in the 3-d receptor space. Maloney and Wandell exploit this plane constraint in their algorithms for color constancy. Maloney proves that each illuminant corresponds to a unique plane of response vectors. Given uniqueness, Maloney and Wandell present an algorithm that can determine the illuminant weight vector e and hence the pseudo-inverse $\Lambda(e)^{-1}$. Consequently the surface weight vector can be recovered as:

$$r = \Lambda(e)^{-1}p \tag{2.8}$$

Gamut Mapping and Diagonal Matrix Color Constancy

Another view of the color constancy problem is "gamut mapping" which is originally developed by Forsyth [45].

A reasonable approach to color constancy may be to not try to derive the reflectance vector but rather to compute the color that the object would have under a canonical light. Specially, if a surface is observed under two different illuminants 1 and 2, the corresponding sensor responses are related by a map which depends only on the illuminants. This mapping is usually considered linear: a matrix transformation is applied to color vectors. In the finite-dimensional basis for reflectance, we can write these two sensor responses as

$$p_1 = \Lambda(e_1)r, \;\; p_2 = \Lambda(e_2)r \tag{2.9}$$

and the mapping between them as

$$p_1 = \Lambda(e_1)\Lambda(e_2)^{-1}p_2 \tag{2.10}$$

If we transform the response vector from image 2 using this matrix$\Lambda(e_1)\Lambda(e_2)^{-1}$, we are effectively generating a new image 2 as it would look in the illumination of image 1. This is an attractive approach because it does not attempt to recover physical quantities. It merely attempts to convert the image under some illumination to a image seen under another illumination.

Given an illuminant, the gamut under this illuminant is the set of all possible observed colors for all possible surface colors. The total set of observed colors is the "gamut" of the illuminant. If we plot those colors in a chromaticity space, it can be shown that the gamut is a convex region. If an image gamut for an illuminant contains two surfaces, say p_1 and p_2, then the gamut must contain the value $\alpha p_1 + (1 - \alpha)p_2$ for $0 \leq \alpha \leq 1$. This also means that the convex hull of the image gamut contains illuminant information. Let us call the gamut of a reference canonical illuminant C (The canonical light is a single known light commonly chosen to be white light). If we observed images under a different illuminant, we get a different gamut which is also a convex region in chromaticity space. We here call this gamut G. We know that there exists a number of transformations T mapping the observed colors under the new illuminant to the corresponding observed colors under the canonical illuminant. Because C covers all possible observed colors under the canonical illuminant, we must have: $T(G) \subset C$. In fact, since both C and G are convex, we need to check only their convex hulls for the transformation maps T. It can be shown that a transform which maps all hull vertices of gamut G into canonical gamut C will also map the non-vertex points into canonical gamut. We here refer the set of possible T to a family of maps taking the gamut G into C.

In the case of finite-dimensional linear models, T depends linearly on e so that the family of maps that satisfy the constraint is itself also a convex set. This family can be constructed by intersecting a set of convex hull, each corresponding to the family of maps that take a hull vertex of G to some point inside C.

This suggests an algorithm for finding the transformation:

- Compute the gamut G under the unknown illuminant for an observed image.
- Compute the convex set of the linear transformations T such that the transformed convex hull of G is included in C.
- Choose one transformation map from this family.

- Transform all observed colors in the image by the map. This will produce the colors that would be seen under the canonical illuminant.

The goal of the Forsyth's gamut-mapping is to calculate transform matrices. In a trichromatic system, the 3×3 matrix is a possible solution. However, this general linear map has nine parameters which implies that solving this map is a nine-dimensional problem. Unfortunately this algorithm is extraordinarily complex and, as Forsyth suggests, "may not be suitable for machine vision." To address this computation problem, Forsyth proposes that mapping matrices should be restricted to three-parameter diagonal matrices. Restricting transforms to diagonal matrices results in Forsyth's simpler CRULE algorithm. Moreover, It has been shown that if sensors are sensitive to a single wavelength of light, the diagonal matrix is a perfect solution to color constancy [55].

Forsyth based his gamut mapping on restrictive assumptions that the scene is flat, the illumination is constant throughout, and that all reflectances are Lambertian surfaces. Finlayson [32] has shown that if we focus on solving only for surface chromaticity and forego estimating surface lightness, then the restriction to flat matte surfaces can be relaxed. Finlayson in his "Color in Perspective" uses a *perspective chromaticity space*: $r = R/B$, $g = G/B$, where R,G,B are sensor responses on three channels, and shows that convexity is maintained where required. One advantage to working in such a chromaticity space is that the algorithm is immediately robust with respect to illumination intensity variation. Such variation is due to the effects of shading and extended light sources. Another important constraint proposed by Finlayson is that illumination itself varies only within certain bounds. It follows that the solution set could be further constrained by considering the gamut of possible illuminants (e.g., purple illuminants do not occur in practice).

Once we have formed this family of maps, it remains to choose an appropriate map from the family of maps. There are a variety of possible strategies. The original method chose the solution which maximizes the volume of restored gamut. Finlayson [33] uses the same heuristic in chromaticity space. In [54], he takes the mean of all the possible maps as the appropriate one. However, both them are not appropriate in perspective space because of the distorted nature of this space. This leads Finlayson et al. [54] to propose finding a constraint set on the two-dimensional plane (r, g, and $b = 1$), and perform the average in a three-dimensional cone which is built by connecting the plane to the origin.

Because the gamut of the image is mapped by a diagonal matrix, the mapped image gamut volume must be proportional to the determinant of the map (note the determinant is the cube of the geometric mean of the three parameters). This observation leads to other two definitions of the optimal feasible map [56]: the map which maximizes the L_1 norm (sum of the diagonal matrix terms), and the map which maximizes the L_2 norm (Euclidean length of the diagonal matrix).

Clearly, for above gamut-mapping algorithms, the goal of color constancy becomes finding a linear transform that maps each color sensor response to a descriptor which is independent of the illuminant. This linear transform is usually in the form of a diagonal matrix. This diagonal linear transform leads to an important solution for color constancy: diagonal matrix transformation (DMT).

The diagonal transformation can be tracked back to Von Kries's law. He assumed that color constancy was, in essence, the result of independent lightness calculations in each channel, meaning that one can rectify an image by scaling each channel independently. This practice is known as Von Kries's law.

Diagonal matrix transformation is the main mechanism in several color constancy algorithms, in particular Von Kries law [62], retinex algorithm [9, 66, 77], Finlayson's spectral sharpening and the above gamut-mapping approach. All these algorithms respond to changing illumination by adjusting the response of each sensor channel independently.

Diagonal matrix transformation support of color constancy is expressed mathematically in Eq. 2.11.

$$p^{i,c} = D^{e,c} p^{i,e} \tag{2.11}$$

where $p^{i,e}$ denotes an observation (a 3-vector of sensor response in a trichromatic system), e and i index illumination and surface reflectance respectively, the vector $p^{i,c}$ represents a descriptor of the surface under the canonical illuminant. The diagonal transform $D^{e,c}$ maps the observation to the descriptor. However, there is no consistent definition for this transform. For example, both Maloney and Wandell's finding surface vector r (Eq.2.8) and Forsyth's mapping between observations under different illuminants (Eq. 2.10) can be expressed by Eq. 2.11.

Finlayson et al. [55] prove that diagonal matrix transforms can support perfect color constancy under small-dimensional model constraints — illuminant space linearly spanned by a 2-dimensional basis and the reflectance space by a 3-dimensional basis, this constraint is referred to as a 2-3 case — if sensors used are sharper (more narrow-band), which can be achieved using a linear combination of sensor sensitivity.

With the addition of sharpening Eq. 2.11 becomes

$$Tp^{i,c} = D^{e,c} T p^{i,e} \tag{2.12}$$

where T denotes the *sharpening transform* of the original sensor sensitivities. It is important to note that applying a linear transformation to response vectors has the same effect as applying the transformation to the sensor sensitivity function.

Also, Finlayson et al. proved that under the conditions imposed by the Maloney-Wandell color constancy algorithm, illuminants are 3-dimensional and reflectance are 2-dimensional (the 3-2 case), color constancy can be expressed in terms of a simple independent adjustment of the sensor responses — in other words color constancy can always be formulated as a generalized diagonal matrix transform — as long as the sensor space is first transformed to a new basis. A consequence of this result is that any color constancy algorithm which makes 3-2 assumptions, these includes the Maloney and Wandell subspace algorithm, Forsyth's MWEXT and Funt and Drew's Lightness algorithm [50], must effectively calculate a simple Von Kries law type scaling of sensor responses; that is diagonal matrix.

We can solve the color constancy problem if red, green, and blue sensor responses, or RGB's, for surfaces seen under an unknown illuminant can be mapped to corresponding RGB's under a known reference light. Despite significant effort, the general three-dimensional color constancy problem has yet to be solved. Maloney and Wandell argued that the three-dimensional color constancy problem was in fact too difficult to solve: There is an intrinsic ambiguity between the brightness of an illuminant and the lightness of a surface, and so dark surfaces viewed under bright lights reflect the same spectral power distribution as that of highly reflective surfaces under dimmer light. This argument is taken on board in almost all modern color constancy algorithms: Modern algorithms attempt only to recover reference chromaticities. The chromaticity constancy problem has proven to be much more tractable. Finlayson made two important observations. The first was that the gamut of possible image chromaticities depended on the illuminant color, and the second was that the illuminant color was itself quite limited. The chromaticities of real illuminants tend to be tightly clustered around the Planckian locus. In Finlayson's algorithm an image chromaticity is said to be consistent with a particular light if it is within the gamut of all possible chromaticities observable under that light. Usually, a single chromaticity will be consistent with many lights; but different chromaticities are consistent with different set of lights. Intersecting all these illuminant sets results in an overall set of feasible illuminants: illuminants that are consistent with all image chromaticities together and at the same time. Typically, the set of feasible illuminants is quite small, and selecting the mean or median illuminant from the feasible set leads to good color constancy. Unfortunately, when color diversity is small, the feasible set can be large. In this case it is quite possible that an incorrect illuminant will be selected.

Retinex Color Constancy

One of the most famous theories of color constancy is Land's Retinex theory. The term 'Retinex' is derived from 'retina' and 'cortex' and describes the biological mechanisms that convert luminous flux into patterns of lightness. The original aim of the theory is a computational model of human vision [76, 77], but it has also been used and extended for machine vision [9, 66, 69]. The details vary in the various of Retinex versions.

Land proposes that rather than our perception being based directly on the sensor responses it was based on a relative response — a relative measure of brightness in a single channel — which he called lightness. He further proposed that the lightness value of a pixel was computed by a series of comparisons of the pixel's intensity with that of many other pixels in the scene. In Land's theory a pixel's lightness value is computed by taking the average of the pixel's ratio to many other pixels in the image. To implement this lightness computation Land proposed a path based approach. Suppose we start at a random pixel which we denote A and follow a random path(a sequence of pixels) to an end pixel Z. Along the way we will visit a sequence of pixels which we denote B, C, and so on. Now, at pixel B we can compute the ratio of B to A by $\frac{B}{A}$ and we can store this value at B. Moving to pixel C, we can compute the ratio of C to A by taking the product of the value we have just stored at B: $\frac{B}{A}$ and the ratio of C to B: $\frac{C}{B}$. That is:

$$\frac{B}{A} \times \frac{C}{B} = \frac{C}{A} \tag{2.13}$$

We store this value at C and move on to the next pixel in the path. At the end of a path we have computed the ratio between the start pixel and each pixel along the path and have stored this ratio at the corresponding pixel. We repeat this process for many paths with different start and end points. Each time we visit a pixel and keep its ratio to the starting pixel. At the end of the process we will have a record of a pixel's ratio to many different pixels and to obtain an estimate of the lightness at a pixel then we average all these ratios.

Thus, the key elements in the retinex computation are to take a ratio followed by a product along a random path and to average the results over many different paths. For computational efficiency the algorithm is usually implemented in log space in which ratios become differences and products additions. We then express the lightness value at a given pixel j as:

$$L^j = \frac{1}{N}\sum_{i=1}^{N} \Lambda_{i,j} \tag{2.14}$$

where $\Lambda_{i,j}$ represents the lightness computed along a path p beginning at pixel i and passing through pixel j:

$$\Lambda_{i,j} = \sum_{k \in p, k<j} \left(\log(R^{k+1}) - \log(R^{k}) \right) \tag{2.15}$$

and N denotes the number of paths which pass through pixel j, and R^k denotes the sensor response at pixel k. This procedure is applied independently to each of the three channels and thus a triplet of lightness values: (L_R,L_G,L_B) are obtained at each pixel in the image. Land proposed that it was these lightness triplets rather than the raw cone responses, that are the basis of our color perception. A complete definition of the retinex algorithm requires that we specify certain parameters: path length, number of path, and how a path is calculated. It has been shown that the choice of these parameters can have a significant effect on the algorithm's output [11].

Land included a *thresholding* step in his original algorithm which was designed to remove the effects of an illumination whose intensity is varying across the extent of a scene. If the pixels are part of the same surfaces, then their ratio will be one. However, if pixels are from different surfaces their ratio will be something quite different to one. It supposes that in addition to a change in surface reflectance along a path there is also a gradual change in the intensity of illumination along the path. This implies that neighboring pixels which are from the same surface can also have a ratio different to one. Land suggested that because illumination typically changes more gradually than does surface reflectance, the effect of a changing illumination could be removed by thresholding ratio such that if a ratio is only slightly different to one it is set to one, but ratios which are quite different to one are left unchanged. By setting the ratios close to one to be one, the resulting lightness image is the same as would have been obtained were the illumination constant across the scene.

Some versions of retinex algorithm supplement ratio, threshold, and product with another step: a *reset* step. As each ratio is computed along a path, this step will check if the current value is greater than one. If it is, then the whole path is reset, such that the area with highest ratio is equal to one and all other compounded ratios are sub-unitary.

Gray-World Color Constancy and Statistical Estimation of Illumination

Since removing the effects of illumination and recovering the illumination are equivalent problems [51], recovering and identifying illumination could be a solution to color constancy. The simplest approaches attempt to identify the illuminant color using a simple statistical estimator. An obvious candidate for such a statistic is the mean, and this leads to the so called "gray-world" assumption. If the average surface color in a scene is constant, and is approximately some known reflectance which is referred to as gray, then the average RGB in an image be used to calculate the scene illumination [14, 114]. Of course the average scene color is rarely gray and so this grayworld approach does not work well in practice. Alternatively, the maximum R, G and B can be used as an illumination estimate [34]. This estimator is justified in a number of scenarios. If for example, there is a white reflectance in a scene then the maximum RGB present will correspond to the RGB of white and this will give a correct estimate of the illuminant color. Moreover, even if white is not present but say there is a bright blue and a bright yellow then the maximum RGB still gives a correct estimate (since Blue is indistinguishable from white for the blue channel and yellow is indistinguishable from white for the Red and Green channels). The max RGB assumption turns out to deliver much better illuminant estimation than the gray world algorithm: though, it is still wrong much of the time. Finlayson proposes another statistical method to identify the illumination, which is called 'Color by correlation' [57]. The essence is using a correlation matrix to establish a stochastic relationship between the chromaticities in a scene and a set of illuminants. This matrix is used as an associative memory to correlate the data from a scene with the set of possible illuminants. The rows of the matrix correspond to all perceivable chromaticities (empirically determined), while the columns correspond to the set of possible illuminants. This matrix is computed based on an a priori observation of the world, i.e. based on a large reference set of surfaces and illuminants. The illuminant of a scene is estimated by a simple voting scheme, that is based on the chromaticities existent in the scene.

Dichromatic Color Constancy

Most models of reflection assume that the surfaces are Lambertian, i.e. perfectly matte and appear equally bright from all directions (isotropic). In practice however, materials are rather inhomogeneous, being composed of a medium and a colorant, e.g. plastics, paints. Depending on the viewing angle, they might appear more or less glossy, which can not be explained by the Lambertian model.

The dichromatic model of reflection [74] assumes that materials are inhomogeneous. The incident light interacts first with the interface of the material, causing an interface reflection. The interface reflection is perceived as a highlight or specularity; this is why it is also called specular reflection.

The other part of the incident light is not reflected at the interface, is scattered inside the body of the material (by the colorants within) and is either absorbed, transmitted (if the material is not opaque), or re-emitted through the interface. This produces a body reflection(Lambertian).

Both of these components can further be decomposed into a term describing the spectral power distribution of illumination, reflectance and a scale factor depending on the geometry.

$$C(\theta, \lambda) = m_I(\theta)S_I(\lambda)E(\lambda) + m_B(\theta)S_B(\lambda)E(\lambda) \tag{2.16}$$

where $m_I(\theta)S_I(\lambda)E(\lambda)$ and $m_B(\theta)S_B(\lambda)E(\lambda)$ are the spectral power distribution of the interface and body reflectance respectively, and m_I and m_B are the corresponding weight factor depending on the geometry θ which includes the incident angle of the light and the viewing direction.

As for many materials, the color of the specularity is same as the color of the light, so independent of surface, and Eq. 2.17 becomes:

$$C(\theta, \lambda) = m_I(\theta)E(\lambda) + m_B(\theta)S_B(\lambda)E(\lambda) \tag{2.17}$$

Thus the sensor responses for dichromatic reflectances will be:

$$R_k = \int_\omega m_I(\theta)E(\lambda)Q_k(\lambda)d\lambda + \int_\omega m_B(\theta)S_B(\lambda)E(\lambda)Q_k(\lambda)d\lambda \tag{2.18}$$

which we rewrite as:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = m_I(\theta)\begin{pmatrix} R \\ G \\ B \end{pmatrix}_E + m_B(\theta)\begin{pmatrix} R \\ G \\ B \end{pmatrix}_B \tag{2.19}$$

where R,G,B are 3 channel outputs. We use E as subscript here because we know the RGB of the interface reflectance is equal to the RGB of the illuminant E.

As can be seen from the above equation the RGBs for a surface lie on a 2-dimensional plane, one component of which is the RGB of the illuminant. If we consider two objects within the same scene then we end up with two RGB planes. Both planes contain the same illuminant RGB. This means that their intersection must be the illuminant itself. Indeed, this is the essence of dichromatic color constancy [112].

Methods in dichromatic-based color constancy rely on the dichromaticity reflection model proposed by Shafer [74]. Among those methods, estimating illumination chromaticity is a main goal. Tan [113] found a direct correlation between illumination and image chromaticities. He described this correlation a two-dimensional "inverse-intensity chromaticity space". Finlayson et al. estimate the illuminant under the dichromatic model when there is just a single surface in the scene [31]. They use Planckian locus as an illuminant constraint based on the observation that, on the CIE xy chromaticity diagram, most natural illuminant chromaticities fall on a long thin "band" which is very close to the Planckian locus of black body radiator. Single-surface dichromatic color constancy proceeds in two simple steps. First, the dichromatic plane for a single surface is calculated and this plane is projected to a line in chromaticity space. In the second step the dichromatic line is intersected with the Planckian locus. The intersection point defines the chromaticity of the illuminant.

2.1.3 Color Invariant Description

Despite significant progress in computational color constancy, color constancy algorithms have not demonstrated the levels of performance and generality that allow them to applied systematically for a large number of object surfaces in color images.

Recent color-based recognition methods find color invariant descriptors from color distributions without using full-blown color constancy algorithm.

The so-called "Color Invariant" descriptions do not try to estimate the scene illuminant and surface reflectance directly. Rather they are features that do not change with a change of light.

There are many different ways how invariant features might be achieved. One possibility is to normalize each image location by some reference RGB such that a new color space is obtained in which pixels are invariant to lighting changes. It is clear that here the scaling factors that describe illumination change will be canceled out. Similarly color ratio between neighboring pixels stay constant across a change of light and have therefore been used for color invariant indexing [53]. It is also possible to derive global statistical features of the color distribution that do not depend on the color of the light [35, 64].

Various color invariant models have been proposed. Evaluating and analyzing them depend on following criteria:

- Robustness to change in surface orientation (i.e. the geometry of the object);
- Robustness to the change in intensity of the illumination:
- Robustness to the change in the spectral power distribution (SPD) of the illumination.

Discrete Reflectance Model

As explained in Section 1, an image taken with a device such as a digital color camera is composed of sensor responses (in the case of Lambertian surfaces) that can be described by:

$$R_k = \sigma \int_{\omega} E(\lambda)S(\lambda)Q_k(\lambda)d\lambda, \quad k = R, G, B, \tag{2.20}$$

Suppose that the sensor sensitivities of the color camera are narrow-band such that their spectral responses can be approximated by a Dirac delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, then Eq. 2.20 becomes simply

$$R_k = \sigma E(\lambda_k)S(\lambda_k)q_k. \tag{2.21}$$

The shading term σ is determined by the direction of the light source and surface normal. It follows that changing the relative position of the light source with respect to the surface introduces shading. We consider a trichromatic imaging device where the responses of three sensors at a single pixel are denoted (R_i,G_i,B_i), so a change in shading can also be described by:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} \rho_i R_i \\ \rho_i G_i \\ \rho_i B_i \end{pmatrix} \tag{2.22}$$

where ρ_i is a simple scalar describing the change of shading. Note that this scalar has a subscript i indicating that all pixels can have their own individual brightness factors. That is, brightness changes, or lighting geometry, is a local phenomenon.

Let us consider a change in illumination color (assuming lighting geometry is held fixed). As explained in Section 2, a diagonal matrix can accurately describe the change of color across different illuminations for the sensors that are narrow-band. From Eq. 2.21, illumination change is very simple to model: if (R, G, B) denote the camera responses of an object under one color of light, the $(\alpha R,\ \beta G,\ \gamma B)$ are the responses induced by the same object viewed under a different illumination. That is, the color channels are simply scaled by the a diagonal matrix:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \tag{2.23}$$

Furthermore, when Eq. 2.23 is not a good model it is possible to linearly transform RGB so as to ensure that the models holds [55].

All of above models, however, rely on the linearity property of the imaging process and hence on calibrated images where pixel values are assumed to have a linear relationship with the actual intensities in the scene. Actually, a power function is applied to raw sensor responses as part of the coding process and this function can be device and even image dependent [19]. That is, R_i, G_i and

B_i are raised to a fixed exponent which we refer to as *gamma* and denote γ:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} R_i^\gamma \\ G_i^\gamma \\ B_i^\gamma \end{pmatrix} \tag{2.24}$$

Now combining Eq. 2.22, 2.23 and 2.24 we can see that:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} \alpha' \rho_i' R_i^\gamma \\ \beta' \rho_i' G_i^\gamma \\ \gamma' \rho_i' B_i^\gamma \end{pmatrix} \tag{2.25}$$

where we incorporate γ into the scalars: $\alpha' \rightarrow \alpha^\gamma, \beta' \rightarrow \beta^\gamma, \gamma' \rightarrow \gamma^\gamma$.

Eq.2.25 tells us how RGB values change when one or more of lighting geometry, illumination color, and gamma changes.

In following sections, these models are used to study and analyze the different color invariant description model.

Color Normalization

Chromaticity space One general chromaticity representation is to divide RGBs by their respective intensity $R+G+B$:

$$(r,\ g,\ b) = (\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}) \tag{2.26}$$

By simply filling in R,G,B from expression 2.25 in the above equation, it can easily be seen that by factoring out intensity, thus, this chromaticity space is independent of intensity and hence independent of the imaging geometry (the position of the objects and the light source) and shading effects.

Gray-world normalization Gray-world normalization [14] divides each pixel by the average of the image:

$$(\frac{R}{mean(R)}, \frac{G}{mean(G)}, \frac{B}{mean(B)}). \tag{2.27}$$

We denote mean(R) the mean red pixel value for the image, mean(G) the mean green pixel value, and mean(B) the mean blue pixel value.

As for the grayworld normalization, the scaling factors that describe illumination change according to the diagonal model cancel out. Thus grayworld normalized images are independent of the color of the incident light.

MaxRGB normalization Similar to grayworld algorithm, MaxRGB [76] normalizes the image globally by a reference pixel. However, here this reference patch is chosen as the brightest one that exists in the scene

$$(\frac{R}{max(R)}, \frac{G}{max(G)}, \frac{B}{max(B)}) \tag{2.28}$$

Images normalized according to Eq. 2.28 are invariant to the scene illumination.

Comprehensive normalization As can we have seen from above, illumination independence may be found by applying grayworld normalization. On the other hand, chromaticity transforms provide invariant to intensity and hence scene geometry. Finlayson et al. proposed [43] that by recursively applying each of these normalizations in turn, a procedure called *comprehensive color image normalization*, it is possible to arrive at a representation that encapsulates both types of invariance.

The comprehensive normalization is defined as:

1. $I_0 = I$ initialization
2. $I_{i+1} = G(C(I_i))$ iteration step
3. $I_{i+1} = I_i$ termination condition

where I denotes an image, $G()$ denotes the grayworld normalization from Eq. 2.27, and $C()$ denotes the chromaticity normalization from Eq. 2.26. That is, chromaticity normalization and grayworld normalization are applied successively and repeatedly to an image until the resulting image converges to a fixed point.

Finlayson proved that this iterative progress always converges to an unique fixed point. Thus if I and I' denote images of the same scene where only the lighting condition have changed then $CN(I) = CN(I')$ where $CN()$ denotes the comprehensive normalization procedure. This method has been shown to outperform the individual grayworld or normalization functions.

However, this invariance is bought at the price of an iterative procedure. To address this shortcoming, Finlayson et al. presents another normalization scheme [58] which they called non-iterative comprehensive normalization procedure, by which iteration is avoided by working with logarithms of RGB images rather than the RGBs themselves. Eq. 2.25 makes it clear that in RGB space, as the lighting conditions change, the effect on a pixel is multiplicative. Taking logarithms turns the multiplications into addition so that the model 2.25 becomes:

$$\begin{pmatrix} \log(R_i) \\ \log(G_i) \\ \log(B_i) \end{pmatrix} \to \begin{pmatrix} \alpha" + \rho_i" + \gamma \log(R_i) \\ \beta" + \rho_i" + \gamma \log(G_i) \\ \gamma" + \rho_i" + \gamma \log(B_i) \end{pmatrix} \tag{2.29}$$

where $\alpha" = \log \alpha'$, $\beta" = \log \beta'$, $\gamma" = \log \gamma'$, and $\rho_i" = \log \rho_i'$.

By inspecting the structure of this model, the dependency on lighting geometry can be simply removed by subtracting the mean log response at a pixel from each pixel. The effect of illumination color can be removed in a similar way. However, rather than dealing with each $\log RGB$ vector, we must operate on the vectors of all $\log R$ (or G, or B) responses across the image. That is, subtracting the mean $\log R$ value from all $\log R$ pixel values and subtracting the mean $\log G$ and mean $\log B$ values from all $\log G$ and all $\log B$ values.

In addition, in this work Finlayson also address the problem of non-linear image formation. To remove the dependence of image color on gamma, they use a second order statistic. In brief, in log space, the power function (gamma) becomes multiplication by a scalar: x^γ maps to $\gamma \log x$. It follows that by dividing resulting values of each channel from the above normalization process by their respective standard divisions, the gamma will cancel.

Color Ratios

Color ratio approaches are mainly used in color image indexing, in which indexing is performed by means of color ratios, rather than the direct color values. By checking how the color response at a point varies with the response at its neighbors, color ratios emphasize the distribution of color changes — boundaries, edges, etc, which are intrinsic properties of the image, and thus largely invariant to external factors, such as illumination changes and image geometry changes.

Funt and Finlayson [53] suggest the use of color ratios by their Color Constant Color Indexing (CCCI)

$$\left(\frac{p_1^1}{p_1^2}, \frac{p_2^1}{p_2^2}, \frac{p_3^1}{p_3^2} \right) \tag{2.30}$$

for object recognition, where p_i^j denotes the pixel value of the ith channel at location j. From the diagonal model of illumination change in Eq. 2.23 it is straightforward to see that this color ratios stay constant across a change of light.

Gevers and Smeulders [59] extend the CCCI technique to account not only for varying illumination color but also for shading effects. Their m1m2m3 space is defined as

$$(m1, m2, m3) = (\frac{p_1^1 p_2^2}{p_1^2 p_2^1}, \frac{p_1^1 p_3^2}{p_1^2 p_3^1}, \frac{p_2^1 p_3^2}{p_2^2 p_3^1}) \tag{2.31}$$

again, p_i^j denotes the pixel value of the ith channel at location j.

Adjeroh and Lee propose another color ratio based invariant feature by observing how the color response at a point varies with response at its m-neighbors. They take the ratio of this variation to the actual color response at that point using the following formula:

$$\Phi = \frac{p_{x,y} - \frac{1}{m}\sum_{i=1}^{m}(p_{x,y} - p_{x,y}^i)}{p_{x,y}} \tag{2.32}$$

where $p_{x,y}$ denotes the color response at a point (x, y) in the image, and $p_{x,y}^i$ denote the response at the ith neighbor of point (x, y). The ratio is performed in each channel separately. The resulting ratio, while invariant to illumination changes, essentially encodes information about the spatial context at a small scale in the image.

Statistical Invariants

By studying the color distribution, Finlayson et al. provide a statistical invariant feature suitable for robust indexing, which they call *color angles* [35]. If denote $v_i = (p_{i1}, p_{i2}, \ldots, p_{iN})$, $i = 1, 2, 3$, as the vectors of responses for each channel, then the color angles of the image are defined as

$$\phi_{ij} = \cos^{-1}(\frac{v_i v_j}{|v_i||v_j|}) \tag{2.33}$$

The three angles between R, G, and B channels are invariant to color of the light source.

Another statistic invariant feature comes from Healey and Slater's *moment invariants* [64]. In this work, Healey and Slater derive invariant features from the moments of color histograms. Starting with the observation that the color histograms of the same scene under two lights are related by an affine transformation, a transformation matrix L is found (by the Cholesky decomposition of the covariance matrix of the color distribution). Using L the color responses can then be transformed into a representation where eigenvalues of centered moment matrices are invariant to illumination changes.

Hue as an Invariant

Hue corresponds to the common definition of color, e.g., red, orange, violet etc. Hue might be calculated using any of the color spaces: HSV, HLS and IHS [101]. Basically, each of these spaces codes RGB by three correlates: hue, saturation and brightness. Brightness correlates to magnitude: white is brighter than gray. Saturation measures the purity of color: a light red is more desaturated than a saturated red.

HSV, HLS and IHS define the hue in the same way [101]:

$$H = \cos^{-1} \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \tag{2.34}$$

We can see that hue cancels out the dependence of lighting geometry. That is, hue does not change with a change in brightness.

Finlayson et al. define a log hue that is invariant to both brightness and gamma [44]:

$$H = \tan^{-1} \frac{\log(R)-\log(G)}{\log(R)+\log(G)-2\log(B)} \tag{2.35}$$

Applying a log transform to RGB values moves the γ term from the exponent into multiplicative scalars, and then the ratio removes it in the log hue.

2.2 Illumination Invariant Image and Intrinsic Image

The above so-called "color invariant" descriptions have been shown to be able to deliver fairly illuminant-independent object recognition. Yet this approach suffers from some problems. First invariants can be calculated only by assuming that there is only single scene illuminant, i.e. the light impinging on every object in the scene should be same. We may consider an example for this situation: Figure 2.5 shows an image with a shadow region. There are typically two lights in the scene: sunlight and skylight. The skylight is everywhere, but the sunlight is blocked by the photographer so that only skylight hits the shadow region. In fact the skylight has a colder color temperation than the sunlight, so between the shadow and no-shadow regions, not only the brightness is different but the reflected light (or the spectral power distribution $E(\lambda)S(\lambda)$ has different color as well. Second, to cancel out variation from illuminants, color invariants have to exploit spatial context or band ratios; e.g., under reasonable conditions the ratio of adjacent RGB's is illuminant independent. The result is that the color of a pixel can be known only relative to the neighborhood of the invariant. The invariants may be interesting for computer vision tasks such as object recognition

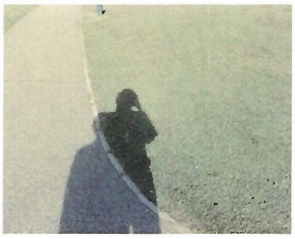


Figure 2.5: Color image with cast shadows

or tracking, but not so much for photography, which generally needs full color information (3 bands), perhaps with illuminant effect removal as developed here. In contrast, color constancy can, by definition, map the image color under one illuminant to corresponding colors under the reference illuminant: we can calculate the three-dimension full color at each point in an image. It is in this sense that color constancy computation delivers more information.

Finlayson et al. in the paper 'Colour Constancy at a Pixel' [40] proposed a new approach called an *illuminant invariant* to bridge the gap between the classical color constancy computation and the invariant approach sketched above. This research begins with the following question: Does there exist a one-dimensional color coordinate, expressed as a function of the RGB or the chromaticity, for which the color constancy problem can be solved? By construction the gray-scale image factors out all dependencies that are due to light intensity and light color. The construction builds on two insights. First, by assuming that the chromaticities of lights lie on the Planckian locus and that camera sensors sample light like Dirac delta functions (they have narrow-band sensitivities), we show that illumination change in log-chromaticity space is translational. It follows that the log chromaticities, which are intensity independent, also translate under a change in illumination color: $(logR/G,\ logB/G)$ becomes $(logR/G,\ logB/G) + (a, b)$ under a second light. It is important to remark that the translational term (a, b) must be the same for all RGBs. Second, and this is the key insight, the translational term for different illuminants can always be written as (a, b), where a and b are fixed constants and a depends on illumination. That is, illumination change translates log chromaticities in the same direction. It follows that the coordinate axis orthogonal to the direction of illumination variation, $y = -(a/b)x$, records only illuminant invariant information: There exist constants r_1 and r_2 such that the coordinate $r_1 logR/G + r_2 logB/G$ is independent of illumination. Of course, real lights will rarely lie exactly on the Planckian locus, nor will camera sensitivities be exactly narrow band. However, experiments demonstrate that the invariant computation is robust to departures from either of these conditions. Note that the invariant coordinates calculated under all lights, including the reference light, do not change. That is, invariant computation at a pixel and one-dimensional color constancy are two sides of the same coin.

Recovering *intrinsic images*, consisting of a shading image and a reflectance image, from a single color image was presented in [110] . The method develops an algorithm that uses multiple cues to recover shading and reflectance intrinsic images from a single image. First each image derivative is classified as being caused by shading or a change in the surfaces reflectance. The algorithm then decomposes an image into shading and reflectance images by integrating reflectance and shading derivative images using Fast Fourier Transform. The system uses two classifiers, one which uses color information to separate shading and reflectance derivatives and a second classifier that uses local image patterns to classify each derivative. For color information, changes in color between pixels indicate a reflectance change, and changes due to shading should affect all three color channels proportionally. For the areas where the color information is ambiguous for distinguishing the two types, the information from the areas where the classification is clear is propagated to the area where the evidence is ambiguous. In order to propagate evidence, each derivative is treated as a node in a Markov Random Field with two possible states: shading and reflectance. Generalized Belief Propagation algorithm [119] is used to infer the best label of each node in the MRF.

In a different direction, Weiss [115] proposed using multiple images where the reflectance is constant, but the illumination changes. This approach was able to create full frequency images, but required multiple input images of a fixed scene. He observed that for an outdoor scene, edges due to illumination often have as high a contrast as those due to reflectance changes, and showed that the output of derivative filtering the images can be well fit by a Laplacian distribution. He verified that a Maximum Likelihood estimate of the filtered reflectance image with respect to the Laplacian distribution can give the reflectance image by median filtering the sequence of the derivative filtered images.

2.3 Shadow Detection

What is a Shadow?

A shadow occurs when an object partially or totally occludes direct light from a source of illumination. From the structure point of view, shadows can be divided into two classes: cast and self shadows. A cast shadow is projected by the object in the direction of the light source, a self shadow is the part of the object which is not illuminated by direct light. A cast shadow consists of a center part without any light from the light source, called the umbra, and a soft transition from dark to bright, called the penumbra, where some light from the light source reaches the background [20] (see Figure 2.6).

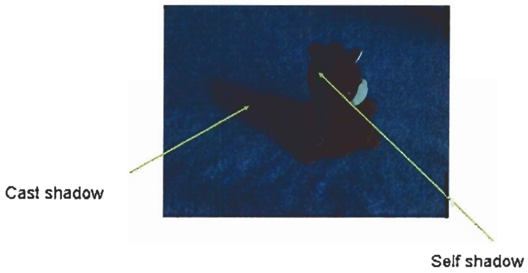


Figure 2.6: Cast shadows and self-shadows.

From the visibility point of view, shadows can be divided into two classes: hard and soft shadows. Hard shadows appear when the light source is a single point source: a surface point is either completely lit or completely shadowed, or in other words, the light source is either completely visible or completely occluded from the surface point. Perfectly hard shadows do not exist physically, but may suffice as an approximation for a relatively small or far away light source. Soft shadows emerge when the light source is not considered to be a point. Such light sources are usually referred to as area light sources. In this case it is possible that a portion of the light source is visible to the surface point, and a portion is occluded. This results in shadow boundaries with gradual falloff from fully shadowed to fully lit.

The presence of shadows in an image can modify the perceived object shape and color. In order to derive relevant shadow properties, the shadow segmentation problem has been increasingly studied in the past years [20, 105, 111]. Most of the existing shadow detection methods exploit information derived from pixel appearance changes in the presence of shadows. They can be classified into model-based and non-model-based methods [93]. Model-based techniques rely on models representing the a priori knowledge of the geometry of the scene, the objects, and the illumination. Non-model based methods, on the other hand, detect shadows using spectral, spatial and temporal properties [20, 111]. A comparative study of shadow detection methods can be found in [93].

2.3.1 Models used in Shadow detection

The model-based methods are designed to describe shadows using physical and statistical models. A physics-based approach to distinguish radiance changes from shadows is presented in [105], where the Dichromatic Reflection Model is exploited for learning background color during an off-line training phase and separating shadowed background from foreground regions in outdoor image sequences.

The same Dichromatic Reflection Model is used in [100] too. The ambient illumination term is added in the expression of the reflected light at a given surface point such that the radiance of the reflected light on shadow pixels depends only on the ambient illumination.

The methods in [88] use a linear transformation to describe the reduction of pixel intensity in shadow regions. Current frame pixels with illumination reductions that follow the linear model are then identified as probable shadow pixels. The coefficients of the linear transformation are statistically estimated over a spatial region in a video sequence. The approaches in [111] tackle the above problems by adaptively updating the linear transformation coefficients over time. However, the correct initialization of these coefficient values still requires human attention.

In [106], the object-shadow boundary can be approximately well modeled by one or more straight lines when the object is a vehicle or a polyhedron in general. The shadows can be then detected by locating the boundary lines.

In [88], the change in appearance of a pixel when shadowed and when illuminated is statistically modeled. The parameters of the model, which exploits a diagonal model of illumination change, require a time-consuming learning phase and are optimized for traffic monitoring. A mixture of Gaussians is used in [105] for modeling the pixels in objects, shadows and backgrounds. The algorithm is initialized by first collecting t initial frames, and then estimating the parameters of the mixture for each pixel by K-means clustering technique.

2.3.2 Properties used in Shadow detection

The image properties used in shadow detection can be classified into three groups: spectral properties, edge and texture properties and geometric properties. Among them, the spectral properties are most used and play a core role in almost all shadow detection works.

In [100], the proposed technique exploits spectral and geometrical properties of shadows in a scene to perform this task. The presence of a shadow is first hypothesized with an initial and simple evidence based on the fact that shadows darken the surface which they are cast upon. The validity of detected regions as shadows is further verified by making use of hypotheses on color invariance and geometric properties of shadows. Hue is used as the color invariance feature which is expected to be unchanged between shadow and object regions.

A method for real-time cast shadow detection for video conference applications is proposed in [102]. The algorithm uses color information in the YUV color space in order to avoid time consuming color transformations. The authors observed that a shadow reduces the YUV values of a point linearly with respect to the same point in light. This observation is used in the method to detect shadows and remove them from segmented objects. Intensity, hue and saturation are exploited in [20] to detect moving cast shadows. The detection is based on the observation that shadows change significantly the brightness of an area without significantly modifying the color information. On the basis of the same observation, a statistical background subtraction algorithm which exploits a computational color model that separates the brightness from the chromaticity components of a pixel is presented in [111].

Edge and texture information is exploited more recently in [107] to detect background regions which are covered or uncovered by a moving cast shadow. Static edges and uniform changes of shading in the background texture are searched for over time. A classification of color edges by means of photometric invariant features into shadow-geometry edges, highlight edges, and material changes is proposed in [60]. A color-ratio space is defined with two components $arctan\frac{R}{B}$ and $arctan\frac{G}{B}$. The shadow-material edges are identified at the points where the modulus of the gradients of the color-ratio space are very small but the gradients of the RGB space are bigger than a predetermined threshold.

Geometric properties of shadows are generally less exploited for shadow segmentation than spectral properties, but they can provide valuable information. In [100], geometry evidence such as shadow-background boundary and shadow-object boundary is checked to verify the existence of shadows in images of a constrained, simple environment. In [49], geometry information of light source direction is combined with color information to detect cast shadows. A limitation of the method is the active process that is required to determine the location of the light source. The approach proposed in [92] overcomes this limitation by presenting a method for the estimation of the projection of the light source direction in the 2-d image plane. The estimated direction is used to guide the cast shadow detection process.

2.3.3 Conclusion for Shadow Detection Approaches

Although many approaches have been developed for detecting shadows, there is not a generally accepted method to detect shadows in image sequences. As concluded by the review presented in [93], different approaches to shadow detection should be taken when addressing different kinds of scenes. As we surveyed above, most of the methods use spectral features, such as intensity, chromaticity, hue or saturation, to identify the shadow pixels. But those features over a shadow region may change under various circumstances. For example, the shadow cast by a big vehicle like a bus can be slightly darker than that of a small vehicle in real-world traffic scenes. The scene illumination also changes temporally when a cloud blocks the sun. Furthermore, many methods need to select ad-hoc thresholds, thus making the methods unpractical to use.

2.4 Flash/No-flash applications

Previous work on flash/noflash pairs, starting with [24] and continuing with [3,28,91], and related work on filling in information in nighttime imagery using much brighter, daytime images [96,97], do not explicitly use the flash information for finding shadows (finding a "shadow matte", in graphics terms — cf. [16]). Instead, these efforts are aimed at using the high-frequency information and detail from the lower-noise flash+ambient image to reduce noise in the ambient-only image, e.g. using a joint bilateral filter. But the flash image by itself suffers from several drawbacks that make using the ambient-only image appealing: illumination dropoff with distance (producing a "tunnel effect" [3]), pixel saturation, and strong interreflections. The ambient image is known to be warmer and more pleasing.

Dicarlo et al. use the camera flash to obtain an additional image for estimating the scene illuminant [24]. The flash/no-flash images are combined to produce a pure-flash image for the scene. This pure-flash image together with knowledge of the SPD of the flash are then used to estimated the surface reflectance in the scene. Finally, using the surface reflectance and the no-flash image, the most likely ambient illuminant can be determined. This approach provides a practical way to estimate ambient illuminant, though it has some limitations when applied to real world applications: it requires about camera sensor spectral characteristics and flash SPD. Also, estimation of scene surface reflectance is mandatory for estimating the illuminant, which uses a linear model approximation of the surface reflectance. The dimensionality of the linear model must be chosen to match the number of camera sensors, so a 3-d model is used, for a standard camera, which is usually not sufficient for representing surfaces in the natural world.

Chapter 3

Illumination Invariants

3.1 Overview

The interaction between light and surface is complex and introduces many unwanted artifacts into an image. Separating illumination effects from reflectance is a very important task and has been studied as a color constancy problem. Finlayson [40] showed that given certain assumptions about scene illumination and camera sensors it is possible to solve a restricted color constancy problem at a single image pixel. This is called *illuminant invariant* theory. Specifically, given a single triplet of sensor responses it is possible to derive a 1-d quantity invariant to both the color and intensity of the scene illuminant. This in effect provides a 1-d reflectance image which is, by construction, shadow-free. Importantly, results demonstrate that applying the theory to images captured under conditions which fail to satisfy one or more of the underlying assumptions, still results in gray-scale images which are, to a good approximation, shadow-free. The 1-d illumination invariant formation will hold so long as a camera is equipped with fairly narrow sensitivities. When sensitivities are significantly broader, the model does not hold. However, Finlayson, Drew and others found that new narrowed-band sensitivities by calculating an appropriate *sharpening* transform. Since we mean to apply logs, it is important to use a color space transform that produces positive. To do so, we proposed a "spectral sharpening with nonnegative constrain" transform, which is similar to the previous "spectral sharpening with positivity" method but replaces guessing objective function weights by actual nonlinear constrains.

In Section 3.2, we introduce the theory of illuminant invariant which forms a simple image representation using assumptions of Planckian lighting, Lambertian surfaces and narrowband camera.

In Section 3.3, we propose a modified spectral sharpening scheme which uses a nonnegative constraint so as to guarantee meaningful logarithms. In Section 3.4, we apply the theory of illuminant invariant to real images so as to derive 1-d grayscale shadow free images.

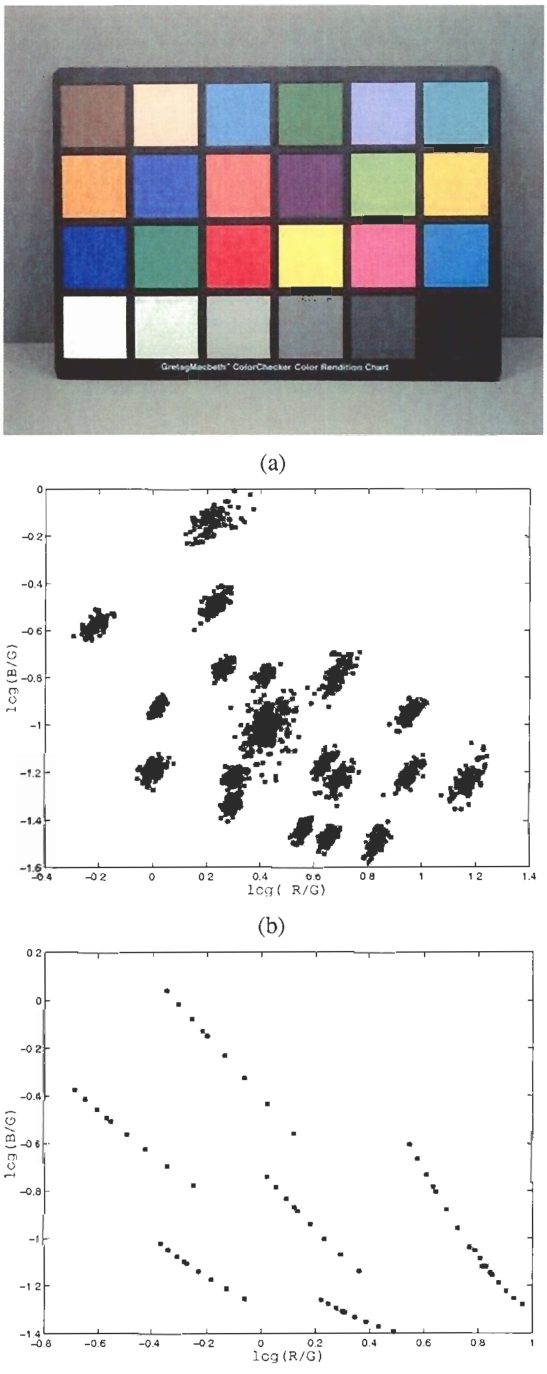
3.2 Theory of Illuminant Invariant Formation

Basically, the illuminant invariant method adopts certain assumptions about lights and cameras so as to lead to a 1-d (i.e., a scalar at each pixel), gray-scale image representation for a color image, which is illuminant invariant at each image pixel. The illuminant invariant image relies upon finding a special direction in a 2-d chromaticity feature space. This "invariant direction" is that for which particular color features, when projected into 1-d, produce a grayscale image which is approximately invariant to intensity and color of scene illumination. Thus shadows, which are in essence a particular type of lighting, are greatly attenuated. The first approach to finding this special angle is a camera calibration: a color target is imaged under many different lights, and the direction that best makes color patch images equal across illuminants is the invariant direction.

3.2.1 Planckian Lighting, Lambertian Surfaces, Narrowband Camera

Suppose we consider a fairly narrow-band camera, with three sensors, Red, Green, and Blue, as in Figure 3.7(a); these are sensor curves for the Sony DXC930 camera. Now if we image a set of colored Lambertian surfaces under a particular Planckian light, in a controlled light box, say, then for each pixel the log of the band-ratios $\{R/G, B/G\}$ appears as a dot in a 2-d plot. Chromaticity removes shading, for Lambertian reflectances under orthography, so every pixel in each patch is approximately collapsed into the same dot (no matter if the surface is curved). Figure 3.1(b) shows the log-chromaticities for the 24 surfaces of the Macbeth ColorChecker Chart shown in Figure 3.1(a) (the six neutral patches all belong to the same cluster). These images were captured using an experimental HP912 Digital Still Camera, modified to generate linear output.

For narrow-band sensors (or spectrally-sharpened ones [37]),and for Planckian lights (or lights such as Daylights which behave as if they were Planckian), as the illuminant temperature T changes, the log-chromaticity color 2-vector moves along an approximately straight line which is independent of the magnitude and direction of the lighting. Figure 3.1(c) illustrates this for six of the patches (with the six neutrals collectively shown as one): the plot shows the same six patches imaged under a range of different illuminants. In fact, the camera sensors are not exactly narrow-band and the log-chromaticity line is only approximate. Assuming that the change with illumination is indeed



(a)

(b)

(c)

Figure 3.1: (c): Median chromaticities for 7 patches, imaged under 14 different Planckian illuminants. (Note that some chromaticities overlap. The neutral patches, with data plotted at the bottom right, all belong to the same collective cluster.)

linear, projecting colors perpendicular to this "invariant direction" due to lighting change produces a 1-d grayscale image that is invariant to illumination. Note that the invariant direction is different for each camera; it can be recovered from a calibration with plots as in Figure 3.1(c). or, as shown later in Chapter 4, from a single image, and with no calibration.

Let's recapitulate how this linear behavior arises. If light with a spectral power distribution (SPD) denoted $E(\lambda, x, y)$ is incident upon a surface whose surface reflectance function is denoted $S(\lambda, x, y)$, then the response of the camera sensors can be expressed as:

$$R_k(x,y) = \sigma(x,y) \int E(\lambda, x, y) S(\lambda, x, y) Q_k(\lambda) d\lambda \tag{3.1}$$

where $Q_k(\lambda)$ denotes the spectral sensitivity of the kth camera sensor, $k = 1, 2, 3$, and $\sigma(x, y)$ is a constant factor which denotes the Lambertian shading term at a given pixel — the dot product of the surface normal with the illumination direction. We denote the triplet of sensor responses at a given pixel (x, y) location by $\boldsymbol{R}\ (x,y) = \{R_1(x,y),\ \ R_2(x,y),\ \ R_3(x,y)\}^T$.

Given Eq. (3.1) it is possible to derive a 1-d illuminant invariant (and hence shadow-free) representation at a single pixel given the following two assumptions. First, the camera sensors must be exact Dirac delta functions and second, illumination must be restricted to be Planckian [63]. If the camera sensitivities are Dirac delta functions, $Q_k(\lambda) = q_k \delta(\lambda - \lambda_k)$, then Eq. (3.1) becomes simply:

$$R_k = \sigma E(\lambda_k) S(\lambda_k) q_k \tag{3.2}$$

where we have dropped for the moment the dependence of R_k on spatial location. The assumption of camera sensors being Dirac delta functions has clearly simplified the image formation equation. Unfortunately, no camera could possibly, or usefully, be sensitive to only three narrow-band wavelengths of light so that this assumption does not really account for image formation in typical cameras. However, it has been shown that Eq. (3.2) models image formation fairly well for cameras whose response sensitivities are sufficiently narrow band [39]. Even when Eq. (3.2) does not hold, it can often be made to hold by applying an appropriate change of sensor basis [37]. In practice, however, Eq. (3.2) is, or can be made to be, a tolerable approximation for most real cameras. Henceforth we will assume that Eq. (3.2) is a good model of image formation (and this is verified by experiment in this section).

We modeled illumination as a blackbody radiator by using Planck's equation Eq. (2.1). More specifically, the illumination is modelled by Wien's approximation to Planck's law [63]. Thus an illuminant SPD can be parameterized by its color temperature T:

$$E(\lambda, T) = I k_1 \lambda^{-5} e^{-\frac{k_2}{T\lambda}} \tag{3.3}$$

where k_1 and k_2 are constants and I is a variable controlling the overall intensity of the light. This approximation is valid for the range of typical lights $T \in [2500, 10000]^o K$.

In Figure 3.2, we compare the normalized spectral power distribution of a typical daylight illuminant (CIE standard daylight source D55) with a 5500K blackbody radiator. It is clear that the shape of the two illuminant curves is broadly similar so as to induce similar RGB's when combining with surface reflectance and camera sensors.

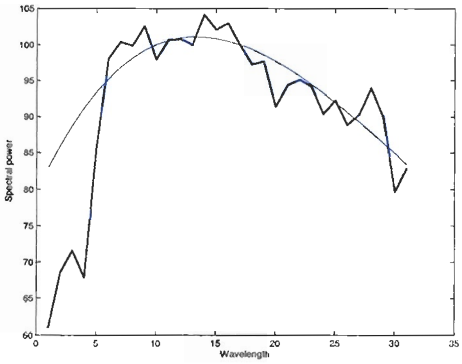


Figure 3.2: Spectral power distributions of one CIE light: D55, and a Planckian light with a temperature of 5500K.

In Figure 3.3 we have plotted the xy chromaticity diagram. The solid curve maps out the chromaticity locus of blackbody radiators from 1000 to 20,000 K (from right to left). We have also plotted the chromaticities for 102 typical illuminants (including daylights and artificial lights) measured around the Simon Fraser University campus [5]. It is evident that all of these lights fall very close to the Planckian locus.

In fact the Planckian constraint on the form of the illuminant can be relaxed to a much more general form, with CIE standard lights still represented very closely [85]. In one variant this is found by performing a principal component analysis on the log of a set of CIE daylights. The great majority of the variance is accounted for by the first eigenvector, so the log of CIE daylight can be represented by $L = L_m + pb$, where L_m is the mean of the log of illuminants, p is the first eigenvector, and b is a weighting factor. L_m and p are functions of spectral wavelength λ, whereas varying b produces different lights. The variation of b with λ is small compared with the variation with T, and so b may be considered a function of T. This leads to a new form for representing the log of illuminants: $L(\lambda) = a(\lambda) + u(\lambda)f(T)$. When going back to exponential form, we

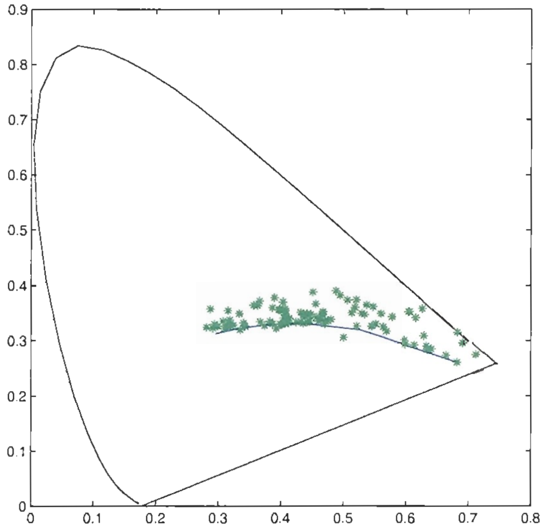


Figure 3.3: xy chromaticity diagram: Planck locus and real lights.

get a one-dimensional submanifold of illuminants $E(\lambda) = a'(\lambda)exp[u(\lambda)f(T)]$. Clearly, Wien's approximation of Planckian light is a special case of this form.

In general, Planck's equation does a good job of capturing the general shape of incandescent and daylight illuminants, but this is not true for fluorescents (which tend to have highly localized emission spikes). But Eq. 3.3 can still be used. This is because we are really interested not in spectra *per se* but rather in how they combine with sensor and surface in forming RGB's. For most daylights and man-made lights, there exists a blackbody radiator, modeled with a Wien's approximation of Planck's law, which can induce very similar RGB's for each of them [63].

With this approximation the sensor responses to a given surface can be expressed as:

$$R_k = \sigma I k_1 \lambda_k^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k. \tag{3.4}$$

Now let us form band-ratio 2-vector chromaticities $\underline{\rho}$:

$$c_j = \frac{R_k}{R_p}, \quad k \in \{1,2,3\}, \quad k \neq p, \quad j = 1,2, \tag{3.5}$$

e.g., for an RGB image, $p = 2$ means $R_p = G$, $c_1 = R/G$, $c_2 = B/G$. Substituting the expressions for R_k from Eq. (4) into Eq. (5) we see that forming the chromaticity co-ordinates removes intensity and shading information:

$$c_j = \frac{\lambda_k^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k}{\lambda_p^{-5} e^{-\frac{k_2}{T\lambda_p}} S(\lambda_p) q_p}, \quad k \in \{1,2,3\}, \; k \neq p, \; j = 1,2. \tag{3.6}$$

If we now form the logarithm $\underline{\rho}$ of $\underline{c}$ we obtain:

$$\rho_j = \log c_j = \log\left(s_k / s_p\right) + w_k + \left(e_k - e_p\right)\frac{1}{T}, \quad j = 1,2 \tag{3.7}$$

where

$$\begin{aligned} e_k &= -k_2/\lambda_k\,, & s_k &= S(\lambda_k)\,, \\ u_k &= \lambda_k^{-5} q_k\,, & w_k &= \log\left(u_k/u_p\right) \end{aligned} \tag{3.8}$$

Note here that, before we take logarithm, we first scale the pixel values to 0..1, from their original range of 0..255. We then replace 0 values with 1/255. Summarizing Eq. (3.7) in vector form we have:

$$\rho = s + \frac{1}{T} e \tag{3.9}$$

Notice that where s is a 2-vector which depends on surface and camera but is independent of the illuminant, the 2-vector direction e is *independent of the surface*, although the line for a particular surface has offset that depends on s_k. Every such line is parallel, with slope dictated by $(e_k - e_p)$.

Now the change of illumination simply amounts to movement along such a line. Then it is a straightforward matter to devise an illumination-invariant image by simply projecting the 2-d chromaticity points $\rho_k, k = 1, 2$, into the direction $e^{\perp}$ orthogonal to the vector $e \equiv (e_k - e_p)$.

3.2.2 Deriving illuminant invariant representation

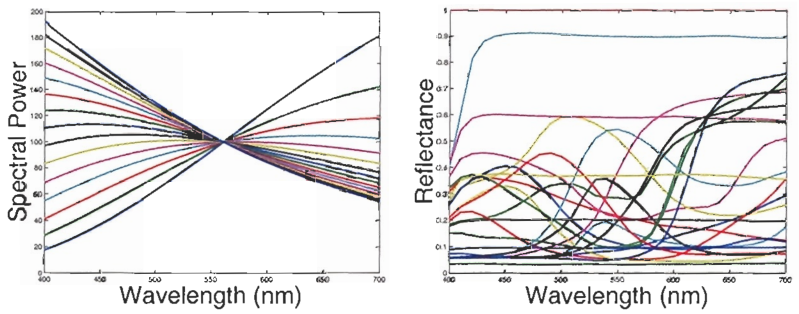
We have presented the illuminant invariant theory under quite restrictive conditions and it is therefore reasonable to ask: In practice, is the method at all useful? To answer this question, we must first calculate the orthogonal projection direction for a given camera. There are a number of ways to do this but the simplest approach is to image a set of reference surfaces under a series of n lights. Each surface produces n log-chromaticities which, ideally, will fall along straight lines. Moreover, the individual chromaticity lines will also be parallel to one another. Of course, because real lights may be non-Planckian and camera sensitivities are not Dirac delta functions, we expect there to be departures from these conditions. It is clear that the chromaticity coordinates do not fall precisely along straight lines in this case. Nevertheless, they do exhibit approximately linear behavior and, so, can we solve for the set of n parallel lines which best account for our data in a least-squares sense [40], that is they subtract the log-chromaticity values of each surface patch by their mean values, so that all the mean-subtracted log-chromaticity points are shifted to clustering around a single line through the origin, and then Singular Value Decomposition process is used to find the direction of the line. Once we know the orthogonal projection direction for our camera, we can calculate log-chromaticity values for any arbitrary image. The test of the method is then whether a resulting invariant quantity $\mathcal{I}$ is indeed illuminant invariant.

We now set up experiments with a synthetic image. In the experiments, the $E(\lambda)$, $S(\lambda)$ and $Q(\lambda)$ in Eq. (3.2) are represented in a vector format. If we measure spectral power distribution, surface reflectance, and sensor sensitivities using 31 samples from 400nm to 700nm at 10nm intervals, then $E(\lambda)$, $S(\lambda)$ are both 1×31 vectors, while $Q(\lambda)$ is a 31×3 matrix, each column representing the sensitivities of each sensor on different wavelengths.

Therefore, the color value for each pixel is a 1×3 vector $\boldsymbol{\rho}$ calculated by

$$\boldsymbol{\rho} = \boldsymbol{E}^T diag(\boldsymbol{S})\boldsymbol{Q} \quad . \tag{3.10}$$

We first generate 15 Planckian lights using Eq. (3.3) with $T = 3000..10000$ with interval 500. From the spectral power distribution (SPD) of these lights shown in Figure 3.4(a), we can see that these lights vary from reddish lights to bluish lights. The reflectance surfaces we use is the Macbeth ColorChecker target, which has 24 painted patches covering a wide gamut of colors. Figure 3.4(b) shows their reflectance curves.

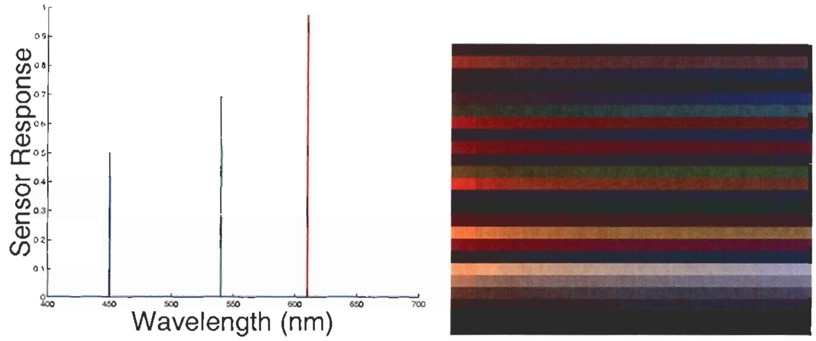


(a)

(b)

Figure 3.4: (a) SPDs of Planckian lights. (b) Surface reflectance curves of Macbeth ColorChecker.

The camera model used here has three ideally narrow-band sensitivities of delta-functions. The response curves of its sensors are shown in Figure 3.5(a), the three peaks from right to left representing the responses of R, G, B sensors respectively. With the illumination SPDs in Figure 3.4(a), the surface reflectance curves in Figure 3.4(b), and the camera sensitivity curves in Figure 3.5(a), we synthesize an image of 24×15 pixels as in Figure 3.5(b). Each row in the image shows the colors of one surface under the 15 different lights.

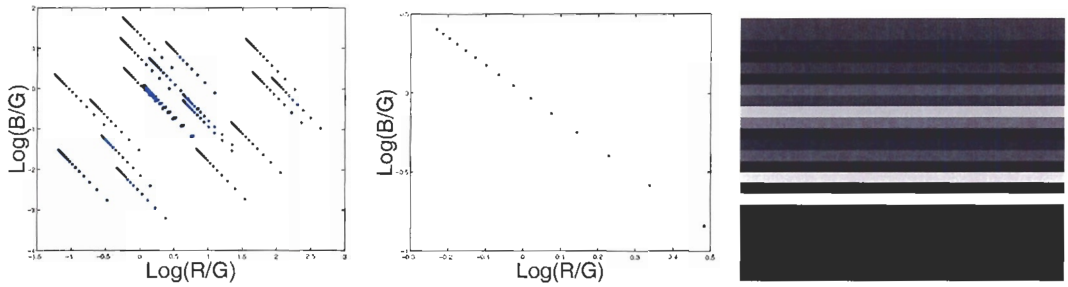


(a) (b)

Figure 3.5: Synthesized image with Planckian lights and delta-function camera. (a) Sensor response of delta-function camera. (b)Synthesized image.

Figure 3.6(a) is the log-chromaticity plot of this image. It is obvious that the points formed from each patch are forming a straight line, and all these lines are suggesting the same direction of lighting change. To compute this direction, we first obtain the mean-subtracted log-chromaticity plot. All the points from the same light are overlapped, so in Figure3.6(b) only 15 points can be seen. The direction is then computed with a linear least-square regression. We now project the log-chromaticity values in Figure 3.6(a) onto the direction orthogonal to lighting change, and form an invariant image as in Figure 3.6(c). Since the intensity values of this image are determined only by camera sensors and surface reflectance, we can see that the grayscale image has uniform intensity for each surface across different lights.

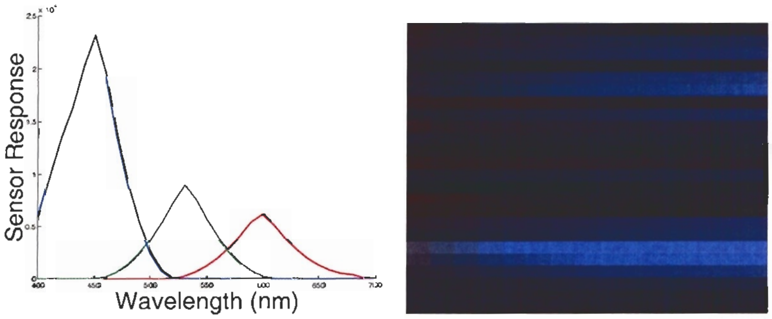
Now using measurements from real lights and camera, for the Macbeth ColorChecker Chart shown in Figure 3.1(a), If we now vary the lighting and plot median values for each patch, we see the approximately straight lines in Figure 3.1(c) which illustrates this for 7 of the patches (with six neutrals as one collective set of data): the plot shows the same 7 patches imaged under a range of different illuminants. We can see that in fact this straight line hypothesis is indeed essentially carried through in practice.



(a) (b) (c)

Figure 3.6: (a) Log-chromaticity plot of Figure 3.5(b). (b) Mean-subtracted log-chromaticity plot. (c) Invariant image of Figure 3.5(b).

To examine the camera sensors whose sensitivities are not delta functions, we used a SONY DXC-930 camera. Its sensors are fairly narrow-band, as shown in Figure 3.7(a). Using the Planckian lights in Figure 3.4(a) and the Macbeth ColorChecker reflectance curves as in Figure 3.4(b), a synthesized image is shown in Figure 3.7(b).

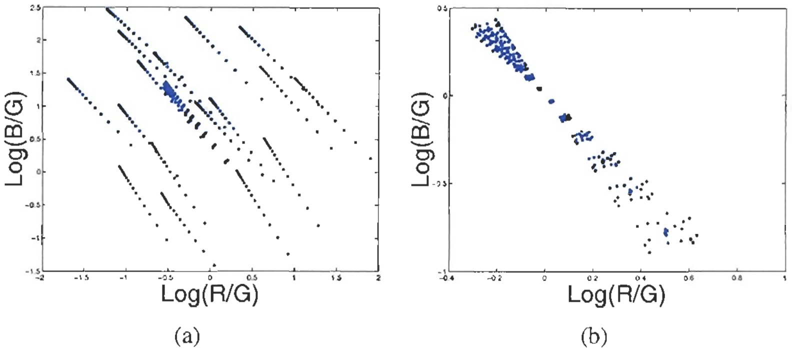


(a) (b)

Figure 3.7: Synthesized image with Planckian lights and SONY DXC-930 camera. (a) Sensor response of SONY DXC-930 camera. (b)* Synthesized image.

The log-chromaticity plot and mean-subtracted log-chromaticity plot are shown in Figure 3.8. Although the sensors of this SONY DXC-930 camera are quite narrow-band, it is already noticeable that the log-chromaticity plot lines are not formed in the same lighting-change direction. In this case, a linear least-square regression gives the principal direction of the data points.

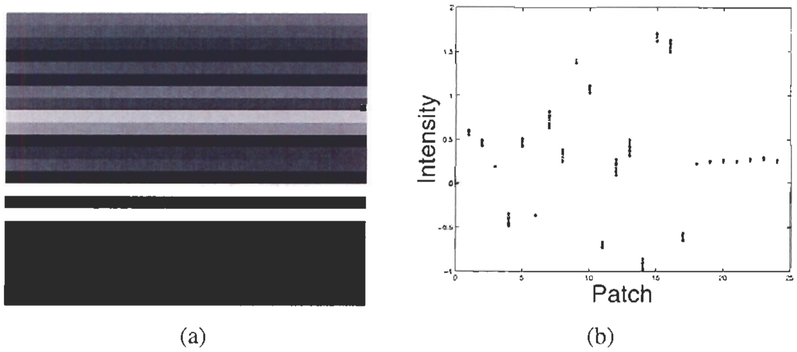
The invariant image obtained is shown in Figure 3.9(a). To examine this invariant image formed



(a) (b)

Figure 3.8: (a) Log-chromaticity plot of Figure 3.7(b) (SONY DXC-930 camera). (b) Mean-subtracted log-chromaticity plot.

with camera sensors that are not delta-functions, we plot out the intensities for all pixels in the invariant image as Figure 3.9(b). We can see that the intensities in each row of the invariant image diverge, but the intensity difference is hardly perceptible.

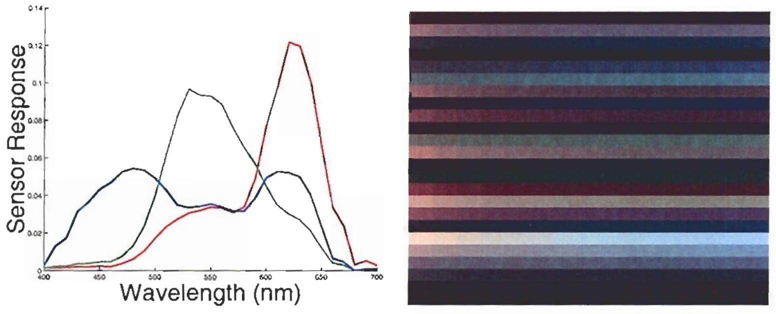


(a) (b)

Figure 3.9: (a) Invariant image of Figure 3.7(d) (SONY DXC-930 camera). (b) Intensity plot of (a).

Another real camera we used is a Kodak DCS420 camera with broader sensitivities as shown in Figure 3.10(a). This camera has a large overlapping range in red and blue response, in particular its blue response has almost the same sensitivity to long wavelengths as to short wavelengths. The image in Figure 3.10(b) is formed using these camera sensitivity curves.

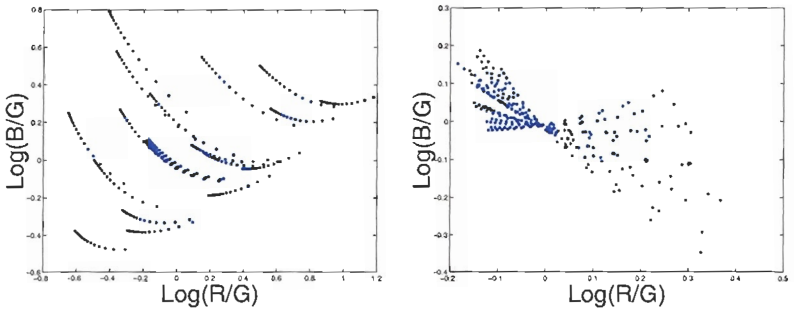
The log-chromaticity plots are shown in Figure 3.11(a) and (b). This time, the broad-band sensors have caused a divergence in log-chromaticity lines. Thus, the resulting invariant image, Figure 3.12(a), is not very smooth across the different lights although it attenuates the difference



(a) (b)

Figure 3.10: Synthesized image with Planckian lights and Kodak DCS420 camera. (a) Sensor response of Kodak DCS420 camera. (b)* Synthesized image.

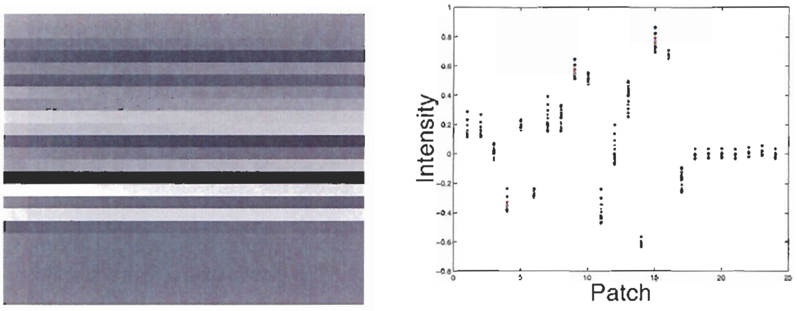
caused by different lights to a good degree. This can also be observed from the intensity plot in Figure 3.12(b).



(a) (b)

Figure 3.11: (a) Log-chromaticity plot of Figure 3.10(b) (Kodak DCS420 camera). (b) Mean-subtracted log-chromaticity plot.

For the above experiments, assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrowband camera sensors can lead to a illuminant invariant representation for color images. The model can be a good (at least tolerable) approximation for most real cameras, and even when these assumptions do not hold, such as for broad-band camera sensors, this model can attenuate the effect of illuminants to good degree.



(a) (b)

Figure 3.12: (a) Invariant image of Figure 3.10(b) (Kodak DCS420 camera). (b) Intensity plot of (a).

3.3 Spectral Sharpening with positivity constraint

From the above section, we see that cameras with broad-band sensors cause inaccuracies in forming invariant images. In this section we will investigate whether transforming these camera sensors to narrow band with a spectral sharpening matrix would lead to improved invariant images.

Clearly, the simplified image formation model (3.2) is more closely followed the more the camera sensor set $Q\ (\lambda)$ approximates a Dirac form. This is a better approximation in an intermediate color space in which the sensors are optimally combined so as to form new colors that better approximate color change induced by illuminant change via a diagonal model.

Spectral sharpening [55] is a method that applies a 3×3 transformation matrix M to the sensors, or directly to colors, so as to better enforce a diagonal model. Suppose $p^{E,S}$ denotes the camera triple RGB for a surface $S(\lambda)$ viewed under $E(\lambda)$. We would like to find the RGB 3-vector $p^{E',S}$ that would be produced under a second illuminant E'. The simplest approach to this problem involves a matrix transform:

$$p^{E',S} \ = \ M^{-1}\ D^{E,E'}\ M\ p^{E,S} \tag{3.11}$$

In Eq. 3.11, M is a fixed 3×3 matrix and $D^{E,E'}$ is an illuminant-dependent diagonal matrix. For our purposes, Eq. 3.11 can be usefully simplified by premultiplying both sides of it by M, yielding

$$M\ p^{E',S} \ = \ D^{E,E'}\ M\ p^{E,S} \tag{3.12}$$

In the above equation, the linear transform can be thought of as defining new sensor functions p':

$$p' \ = \ M\ p \tag{3.13}$$

and

$$p'^{E',S} = D^{E,E'} p'^{E,S} \tag{3.14}$$

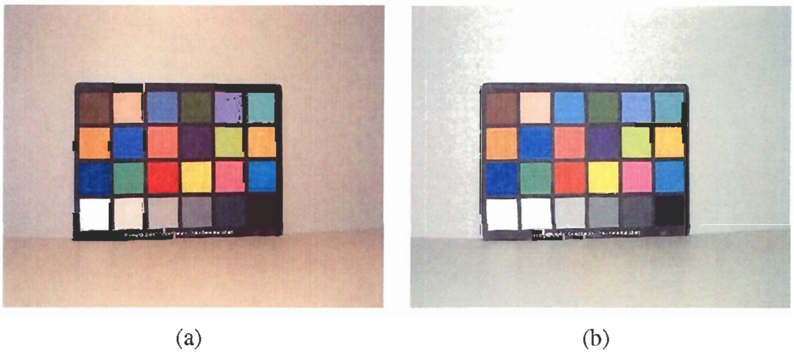
Thus the effect of the transformation can be modeled by simple scalar multipliers operating individually on each of the R, G, and B (the diagonal matrix has only three nonzero terms).

Since we mean to apply logs, it is important to use a color space transform that produces positive, or at least nonnegative, colors from the camera data (and zero values can be treated specially). To do so, we carry out a novel variant of a "spectral sharpening with positivity" transform [4,25]. Consider the pair of images C_1, C_2 in Figs. 6.2(a,b): here, a Macbeth ColorChecker chart [87] is imaged under two different lights, in a light-box. We can find the spectral sharpening transform $\boldsymbol{M}$ via the following novel optimization consisting of a constrained form of "database sharpening" [37]:

$$\overset{min}{\boldsymbol{M},\boldsymbol{D}} \ \sum[\boldsymbol{M}\,\boldsymbol{D}\,\boldsymbol{M}^{-1}C_1 \ - \ C_2]^2$$

$$\text{with constraints} \begin{cases} -\boldsymbol{M}\,C_1 \ \leq \ 0 \\ \text{non-negative sensor result,} \\ \\ \boldsymbol{D} \ = \ \text{diagonal matrix} \end{cases} \tag{3.15}$$

The Macbeth chart has 24 paint patches, so this optimization involves only 24 color pairs under the two lights; it is similar to that in [4] but replaces guessing objective function weights by actual nonlinear constraints [27].


(a)

(b)

Figure 3.13: (a) Macbeth ColorChecker under illuminant A. (b) Macbeth ColorChecker under daylight.

3.4 1-d shadow free image

Once we determine the direction of illuminant variation (the vector e) then we can determine a scalar, 1-d illuminant invariant representation by projecting the log-chromaticity vector $\underline{\rho}$ onto the vector orthogonal to $\underline{e}$, which we denote $\underline{e}^{\perp}$. That is, our illuminant invariant representation is given by a gray-scale image $\mathcal{I}$:

$$\mathcal{I}' = \rho^T e^{\perp}\,, \qquad \mathcal{I} = \exp(\mathcal{I}') \tag{3.16}$$

The resulting grayscale image is hence independent of lighting. In a sense, therefore, it is a intrinsic image that portrays only the inherent reflectance properties in the scene. Since shadows are mostly due to removal of some of the lighting, such an image also has shadows removed. Note that even in the situation in which shadows overlap, this physically-based invariant image can still remove the shadows, since the overlaped shadows are still caused by the changes of scene illumination. When several lights impinge on a surface point, the invariant-image formulation still applies since the light combination still sufficiently approximates a Planckian [36].

Figure 3.14 illustrates the process we have just described. The figure shows log-chromaticities for 4 different surfaces (open circles), for perfect narrow-band sensors under a range of Planckian illuminants. It is clear that the chromaticities for each surface fall along a line (dotted lines in the figure) in chromaticity space. These lines have direction $\underline{e}$. The direction orthogonal to $\underline{e}$ is shown by a solid line in Figure 3.14. Each log-chromaticity for a given surface projects to a single point along this line regardless of the illumination under which it is viewed. These points represent the illuminant invariant quantity $\mathcal{I}'$ as defined in Eq. (3.16).

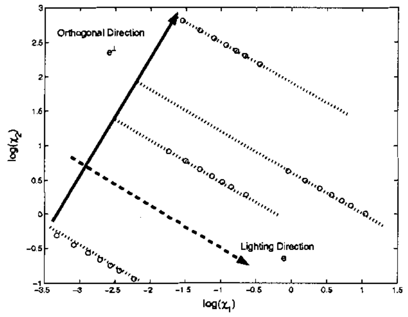


Figure 3.14: An illustration of the 1-d invariant representation.

From (3.7), we can remove σ and I via division by any color channel: but which channel should we use? If we divide by red, but red happens to be everywhere small, as in a photo of greenery, say, we are in trouble. A better solution is to divide by the geometric mean [36], $\sqrt[3]{R \times G \times B}$. Then we still retain our straight line in log space, but do not favor one particular channel.

Thus we amend our definition of Eq. (3.7) of chromaticity as follows:

$$c_k \;=\; R_k/\sqrt[3]{\Pi_{i=1}^{3} R_i}, \quad \equiv\; R_k/R_M, \tag{3.17}$$

and log version [36]

$$\begin{array}{l} \rho_k \;=\; \log(c_k) \;=\; log(s_k/s_M) \;+\; (e_k - e_M)/T,\; k = 1..3, \text{with} \\ s_k = k_1 \lambda_k^{-5} S(\lambda_k) q_k \;, s_M = \sqrt[3]{\Pi_{j=1}^{3} s_j} \;, e_k = -k_2/\lambda_k,\; e_M = -k_2/3 \sum_{j=1}^{p} \lambda_j, \end{array} \tag{3.18}$$

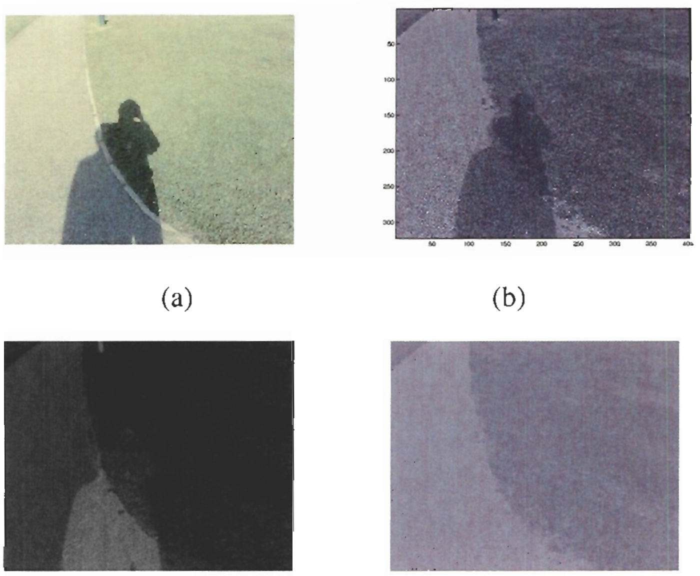
and for the moment we carry all three (thus nonindependent) components of chromaticity. Broad-band camera versions are stated in [36].

We have derived this 1-d illuminant invariant representation under quite restrictive conditions (though the conditions on the camera can be relaxed to broad-band sensors with the addition of some conditions on the reflectances [13]) and it is therefore reasonable to ask whether in practice the method is practical? To answer this question we must first calculate the orthogonal projection direction for a given camera. The test of the method is then whether the resulting invariant quantity $\mathcal{I}$ is indeed illuminant invariant.

Figure 3.15 illustrates the method for an image taken with a commercial Digital Still Camera (modified such that it returns linear output without any image post-processing). Figure 3.15a shows the color image as captured by the camera (for display purposes the image is mapped to sRGB [108] color space) — a shadow is very prominent. Figure 3.15b,c show the log-chromaticity representation of the image. Here, intensity and shading are removed but the shadow is still clearly visible, highlighting the fact that shadows represent a change in the color of the illumination and not just its intensity. Finally Figure 3.15d shows the invariant image representation defined by Eq. (3.16). It is clear that the method delivers very good illuminant invariance: the shadow is completely removed from the image. This image is typical of the level of performance achieved with the method. More examples will show in later sections, and in all cases shadows are completely removed or greatly attenuated.

3.5 Summary

In this chapter, we have presented an illuminant invariant theory, which was first proposed by Finlayson [40]. The theory was validated by first verifying the linear behavior for different illuminants



(a) (b)

(c) (d)

Figure 3.15: An example of the 1-d illuminant invariant representation. (a) The original image; (b) and (c) log-chromaticity representations (χ_1' and χ_2'); (d) the 1-d invariant $\mathcal{I}$.

under assumptions on lighting, surface and camera sensitivity. Then we modified the previous sharpening scheme by enforcing a nonnegative constraint. Finally, we show that applying the illuminant invariant theory to real images can derive 1-d grayscale shadow-free images.

Chapter 4

Intrinsic Image by Entropy Minimization

4.1 Overview

Finding the invariant direction is the essential issue for the application of removing shadows. In this chapter, instead of a camera calibration[1] we aim at finding the invariant direction from evidence in the color image itself. Specifically, we recognize that producing a 1-d projection in the correct invariant direction will result in a 1-d distribution of pixel values that have smaller entropy than projecting in the wrong direction. Hence we seek that projection which minimizes entropy. Using entropy minimization to find the invariant direction was proposed in our joint work [38]. To be able to develop an effective description of the entropy-minimization task, we go over to the quadratic entropy, rather than Shannon's definition. Replacing the observed pixels with a kernel density probability distribution, the quadratic entropy can be written as a very simple formulation, and can be evaluated using the efficient Fast Gauss Transform.

In Section 4.2, we make the observation that using entropy minimization can obtain the invariant projection from a single color image. In Section 4.3, we formalize an entropy minimization algorithm, and show that entropy is a strong and reasonable indicator for finding the correction projection direction, by testing on synthetic and real images. In Section 4.4, we use quadratic entropy

[1] The term "calibration" is not used here in the usual sense of spatial or color calibration, but instead pertains to finding the invariant direction, by the use of a color target imaged under several illuminants.

instead of Shannon's entropy to make the minimization more efficient. We also include explanations for how to set the bin width for Shannon's entropy, and include a description of the Fast Gauss Transform in an Appendix.

4.2 Motivation

The problem we consider, and solve, is the determination of the invariant image from *unsourced* imagery — images that arise from cameras that are *not calibrated.* The input is a color image with unknown provenance, one that includes shadows, and the output is the invariant version, with shading and shadows removed.

To see how we do this let us remember how we find the intrinsic image for the calibrated case. This is achieved by plotting 2-d log-chromaticities as lighting is changed and observing the direction in which the resulting straight lines point and then projecting in this direction. The key requirement, in the calibration case, is to find the direction of straight lines — the invariant direction — as lighting is changed. The fundamental idea of finding the direction for a single unsourced image is the observation that, without having to image a scene under more than a single illuminant, projecting in the correct direction minimizes the entropy in the resulting grayscale image. The intuition behind this statement is evident if one thinks of a set of color patches under changing lighting. As lighting changes, for each color patch, pixels occupy an approximately straight line in a 2-d log-chromaticity space. If we project all these pixels onto a line perpendicular to the set of straight lines, we end up with a set of 1-d points, as in Figure 4.1(a). In a set of real images of color patches, we would expect a set of peaks, each well separated from the others and corresponding to a single color patch. On the other hand, if we instead project in some other direction, as in Figure 4.1(b), then instead of pixels located in sharp peaks of occurrence we expect the distribution of pixels along our 1-d projection line to be spread out. In terms of histograms, in the first instance, in which we guess the correct direction and then project, we see a distribution with a set of sharp peaks, with resulting low entropy. In the second instance we instead see a broader histogram, with resulting higher entropy.

Hence the idea in this presentation is to recover the correct direction in which to project by examining the entropy of a grayscale image that results from projection and identifying as the correct "invariant direction" that which minimizes the entropy of the resulting image. Changing lighting is automatically provided by the shadows in the image themselves.

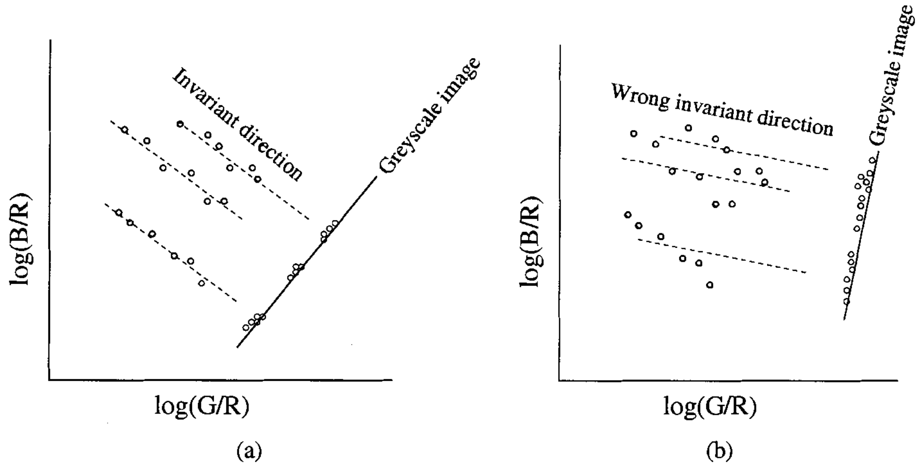


Figure 4.1: Intuition for finding best direction via minimizing the entropy.

4.3 Intrinsic Images by Entropy Minimization

Here, without any calibration or foreknowledge of the invariant direction, we begin by creating a large synthetic "image" that consists of a great many color patches. Since the image is synthetic, we in fact do know the ground truth invariant direction. Examining the question of how to recover this direction from a single image, with no prior information, we show that minimizing the entropy provides a very strong indicator for determining the correct projection. For a synthetic image, results are very good indeed. This result provides a proof in principle for the entropy-minimizing method.

But how do we fare with a real camera? We consider a set of captured color-patch images, taken with a known camera. Since we control the camera, and the target, we can establish the invariant direction. Then comparing to the direction recovered using entropy minimization, we find that not only is the direction of projection recovered correct (within 3 degrees), but also the minimum is global and is a very strong signal — essentially, nature is telling us that this is indeed the way to go: entropy minimization is a new and salient indicator for the projection that removes the shadows.

4.3.1 Synthetic Images vs. Entropy Minimization

If we wished to find the minimum-variance direction for lines in Figure 4.1, we would need to know which points fall on which lines. But what if we did not have that information? Entropy minimization is the key to finding the right invariant direction.

To test the idea that entropy minimization gives an intrinsic image, suppose we start with a theoretical Dirac-delta sensor camera, as in Figure 4.2(a). Now let us synthesize an "image" that consists of many measured natural surface reflectance functions interacting with many lights, in turn, and then imaged by our theoretical camera. As a test, we use the reflectance data $S(\lambda)$ for 170 natural objects, measured by Vrhel et al. [114]. For lights, we use the 9 Planckian illuminants $E(\lambda)$ with T from 2,500° to 10,500° Kelvin with interval of 1,000°. Thus we have an image composed of 1,530 different illuminant-reflectance color signal products.

If we form chromaticities (actually we use geometric mean chromaticities defined in Eq. (3.17) above), then taking logarithms and plotting we have 9 points (for our 9 lights) for every color patch. Subtracting the mean from each 9-point set, all lines go through the origin. Then it is trivial to find the best direction describing all 170 lines via applying the Singular Value Decomposition method to this data. The best direction line is found at angle 68.89°. And in fact we know from theory that this angle is correct, for this camera. This verifies the straight-line equation (3.9), in this situation where the camera and surfaces exactly obey our assumptions. This exercise amounts, then, to a calibration of our theoretical camera in terms of the invariant direction.

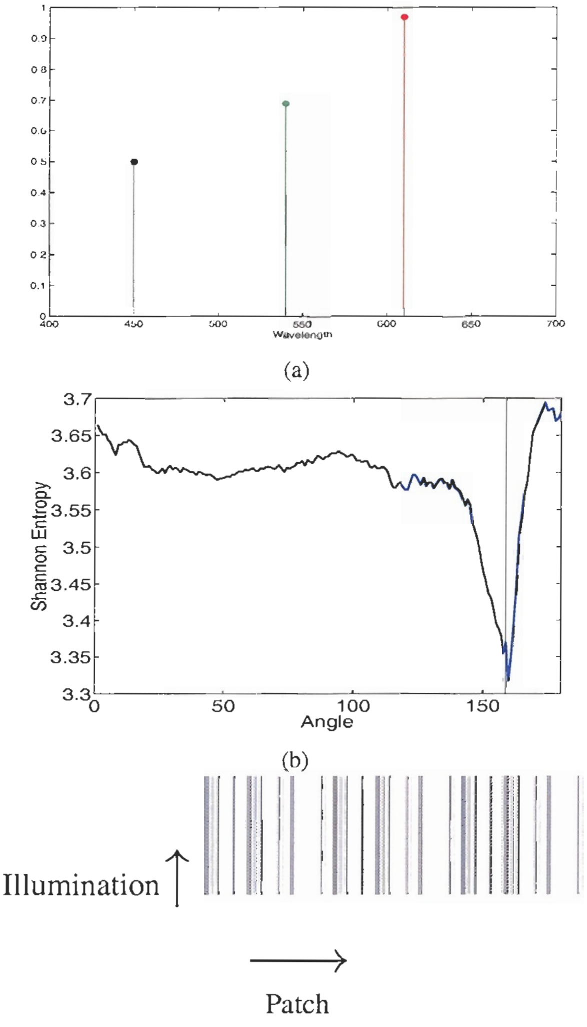
But now suppose we do not know that the best angle at which to project our theoretical data is orthogonal to about 69° — how can we recover this information? Clearly, in this theoretical situation, the intuition displayed in Figure 4.1 can be brought into play by simply traversing all possible projection angles that produce a projection direction $e^{\perp}$: *the direction that generates an invariant image with minimum entropy is the correct angle.*

To carry out such a comparison, we simply rotate from 0° to 180° and project the log - chromaticity image ρ 2-vector into that direction. A histogram is then formed, and finally the entropy is calculated: the histogram is divided by the sum of the bin counts to form probabilities p_i and, for bins that are occupied, the sum of $-p_i \log_2 p_i$ is formed.

Figure 4.2(b) shows a plot of angle versus this entropy measure, for the synthetic image. As can be seen, the correct angle of $159 = 90 + 69°$ is accurately determined (within a degree).

As we go from left to right across Figure 4.2(c) we change reflectance. From top to bottom we have pixels calculated with respect to different lights. Because the figure shows the invariant image coded as a gray scale there is very little variation from top to bottom. Yet the grayscale value does change from left to right. So, in summary, Figure 4.2(c) tells us that the same surface has the same invariant across lights but different surfaces have different invariants (and so the intrinsic image conveys useful reflectance information).

Next, we consider an "image" formed from measured calibration values of a color target, as in



(a)

(b)

(d)

Figure 4.2: (a): Theoretical narrowband RGB camera sensors. (b): Minimum Shannon's entropy invariant direction gives same angle as calibration test. (c): Invariant image for theoretical synthetic image — same graylevels across illuminants.

Figure 3.1.

4.3.2 Calibration Images vs. Entropy Minimization

Now let us investigate how this theoretical method can be used for real, non-synthetic images. We already have acquired calibration images, such as Figure 3.1(a), over 14 phases of daylight. These images are taken with an experimental HP 912 digital camera with the normal nonlinear processing software disabled.

Geometric Mean 2-D Chromaticity Space

We use the geometric mean 2-d chromaticity space ρ that is introduced in Section 3.4. We note that, in log space, ρ is orthogonal to $\boldsymbol{u} = 1/\sqrt{3}(1,1,1)^T$. I.e., ρ lives on a plane orthogonal to $\boldsymbol{u}$, as in Figure 4.3, $\rho \cdot \boldsymbol{u} = 0$. To characterize the 2-d space, we can consider the projector $\boldsymbol{P}_{u}^{\perp}$ onto the plane. $\boldsymbol{P}_{u}^{\perp}$ has two non-zero eigenvalues, so its decomposition reads

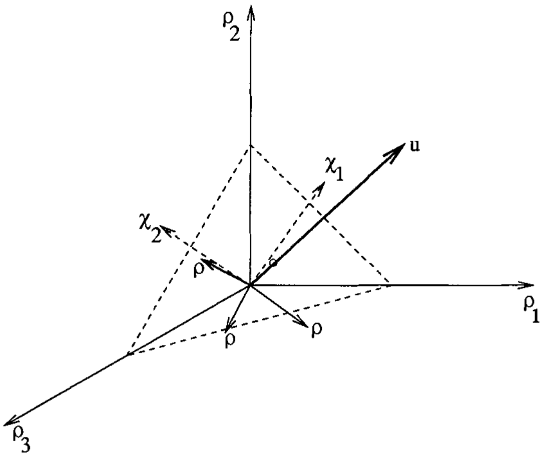


Figure 4.3: Geometric mean divisor means every ρ is orthogonal to $\boldsymbol{u}$. Basis in plane is $\{\chi_1, \chi_2\}$.

$$P_u^{\perp} = I - \boldsymbol{u}\,\boldsymbol{u}^T = \boldsymbol{U}^T\boldsymbol{U}\,, \tag{4.1}$$

where $\boldsymbol{U}$ is a 2×3 orthogonal matrix. $\boldsymbol{U}$ rotates 3-vectors ρ into a coordinate system *in* the plane:

$$\chi \equiv \boldsymbol{U}\,\rho\,, \quad \chi \text{ is } 2 \times 1. \tag{4.2}$$

Straight lines in ρ are still straight in χ .

In the $\{\chi_1, \chi_2\}$ plane, we are now back to a situation similar to that in Figure 4.1: we must find the correct direction θ in which to project, in the plane, such that the entropy for the marginal distribution along a 1-d projection line orthogonal to the lighting direction is minimized. The grayscale image $\mathcal{I}$ along this line is formed via

$$\mathcal{I} \;=\; \chi_1 \cos\theta \,+\, \chi_2 \sin\theta \tag{4.3}$$

and the entropy is given by

$$\eta \;=\; -\sum_i p_i(\mathcal{I})\ \log(p_i(\mathcal{I})). \tag{4.4}$$

Test of Main Idea

Thus the heart of this test of the entropy-minimization idea is as follows:

(a) Form a 2-d log-chromaticity representation of the image.

(b) for $\theta \;=\; 1..180$

  (i) Form grayscale image $\mathcal{I}$: the projection onto 1-d direction.

  (ii) Calculate entropy.

  (iii) Min-entropy direction is correct projection for shadow removal.

An actual algorithm would need to proceed faster than this type of brute force search, of course, and that issue is addressed in Section 4.4.2.

From the calibration technique described in Section 4.3.1 we in fact already know the *correct* characteristic direction in which to project to attenuate illumination effects: for the HP-912 camera, this angle turns out to be 158.5°. We find that entropy minimization gives a close approximation of this result: 161°.

First, transforming to 2-d chromaticity coordinates χ, the color patches of the target do form a scatterplot with approximately parallel lines, as in Figure 4.4(a). We compose an "image" consisting of a montage of median pixels for all 24 color patches and 14 lights. The calculation of entropy carried out for this image gives a very strong extremum, shown in Figure 4.4(b,c), and excellent grayscale $\mathcal{I}$ invariant to lighting in Figure 4.4(d).

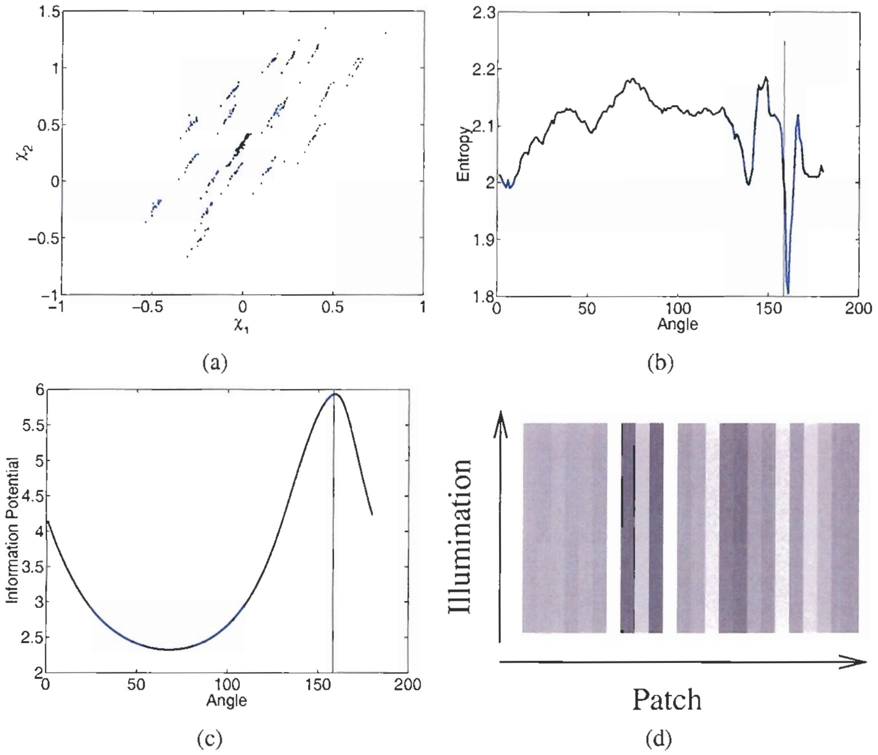


Figure 4.4: (a): 2-d chromaticity for measured color patches, HP 912 camera. (b): Minimum entropy invariant direction gives angle close to that of calibration method. (c): Same angle is found by quadratic entropy. (d): Grayscale invariant image for measured patch values — projected graylevels same for different illuminants.

4.3.3 Entropy Minimization — Strong Indicator

From the above sections, it is evident that the use of entropy can reveal how the projected data concentrates or spreads out around a set of peaks and how each peak separates from the others. There are also several ways to measure the dispersion in the data distribution. The most common one is the variance, which provides an average distance for each element from the mean. If a distribution has a large variance, then the data points are more likely to be far from the mean. Another statistical measure, the kurtosis, measures the departure from normality by indicating the extent of the peak (or the degree of flatness near its center) in a distribution. The kurtosis is given by the ratio of the fourth central moment divided by the square of the variance:

$$\gamma = \frac{\mu_4}{\mu_2^2} \tag{4.5}$$

where the μ_r is defined as $r - th$ central moment as

$$\mu_r = E[(X - \mu)^r] \tag{4.6}$$

Here, X corresponds to the data, μ corresponds to the mean and the variance is given by μ_2^2. From the definition, a greater kurtosis indicates a more peaked distribution than the normal distribution, and smaller kurtosis indicates the curve is flatter than the normal.

Both variance and kurtosis measure the dispersion with respect to a single normal distribution; but our case is different in that we are dealing with the distribution of a set of peaks. Therefore here we define a new measurement for the data dispersion, similar to the above but tailored to our problem. We call this more appropriate measure the *clusters' dispersion*, defined by the square error of the cluster dispersion:

$$E = \sum_{k=1}^{K} \sum_{x \in C_k} \| X - m_k \|^2 \tag{4.7}$$

where K is the number of clusters, and m_k is the center of cluster C_k. This is indeed the objective function for the K-means clustering algorithm. The difficulty of implementing the dispersion is that the number of clusters must be decided first. For this purpose, we use the Adaptive Mixture method [94] to estimate the number of component densities in a mixture model. The basic idea behind adaptive mixtures is to take one point at a time and determine the distance from the observation to each component density in the model. If the distance to each component is larger than some threshold, then a new item is created. If the distance is less than the threshold for all terms, then the parameter estimates are updated based on the recursive Expectation-Maximization algorithm.

To compare the performance of entropy minimization with variance, kurtosis and clusters' dispersion, we calculate the three over all projection angles for both calibration images and for the 'path' image, which includes shadows (Figure 2.5).

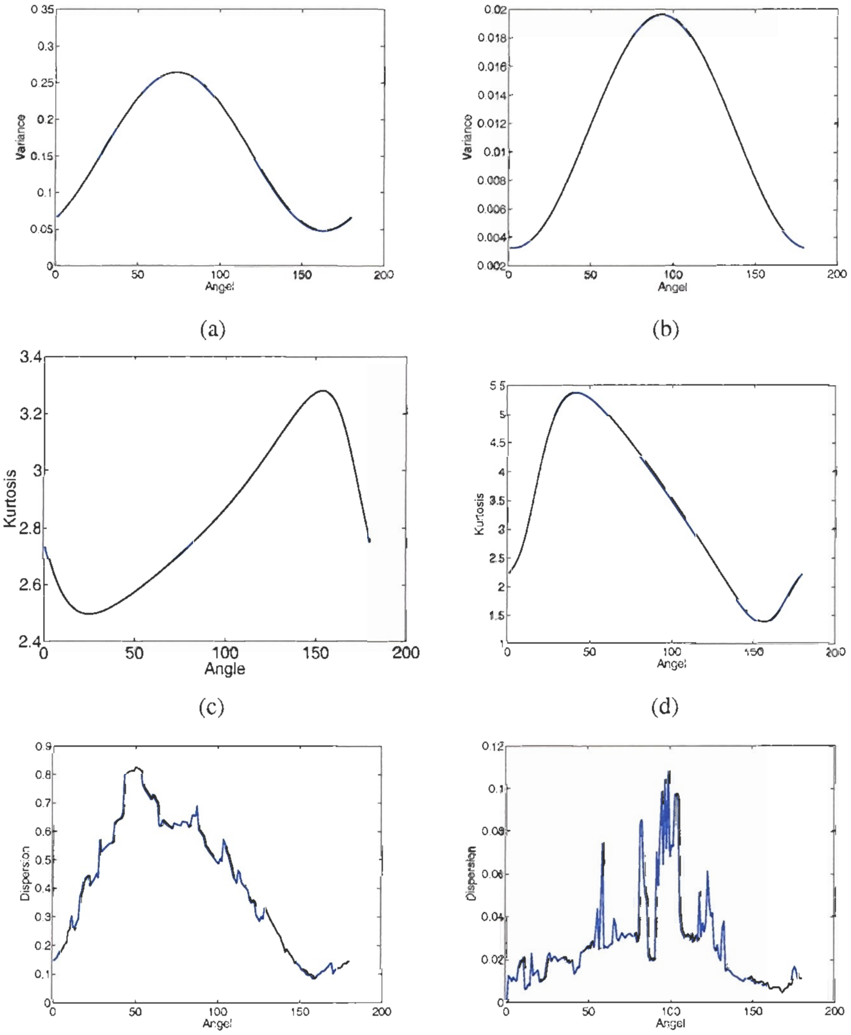
From the results in Figure 4.5, we can see that both variance and kurtosis give very smooth curves as the angle changes. This is because they regard the data as a single normal distribution so that they do not show a big difference between two neighboring angles. Variance has a minimum point at angle 158° which is very close to the calibrated angle 159°, but fails to find the angle for the 'path' image – the minimum falls at angle 2°. Kurtosis works well on the 'path' image (with minimum point at angle 155°) but does not work on the calibration images at all. Clusters' dispersion, on the other hand, gives a minimum angle at 156° for calibration images and 170° for the 'path' image. The angles are acceptable in term of the calibrated angle 159°, but the curves are too noisy. In fact, the intuition of our definition of clusters' dispersion is similar to the use of entropy in that they both pay attention to multiple peaks and the distribution for each peak. But a clustering step is necessary for measuring the clusters' dispersion, which is difficult to achieve accurately and is time consuming. From these results, we see that the entropy minimization method indeed outperforms other dispersion measures, both theoretically and effectively, for finding the invariant direction since the use of entropy not only can give a efficient and accurate direction detection but also provides a reasonable explanation for the distribution of the illumination invariant projection. This completes the theoretical justification of the main idea — finding the invariant projection direction by entropy minimization.

4.4 Intrinsic Image Recovery Algorithm

In this section, we examine the issues involved when we extend this theoretical success to the realm of real, non-calibration images. To dispense with a brute-force search over all angles, we also need a disciplined search mechanism, and we see that this is provided by the quadratic entropy measure, with implementation by Fast Gauss Transform.

4.4.1 Shannon's Entropy and Quantization

Real, non-synthesized and not staged in a laboratory, images are noisy and might not provide such a clean picture as in our theoretical tests above. As well, we must decide on a quantization procedure if we wish to utilize Shannon's definition of entropy.



(a) (b)

(c) (d)

(e) (f)

Figure 4.5: Full-angular plot for calibration images (left column) and the 'path' image (right column): (a,b): variance; (c,d): Kurtosis; (e,f): clusters' dispersion.

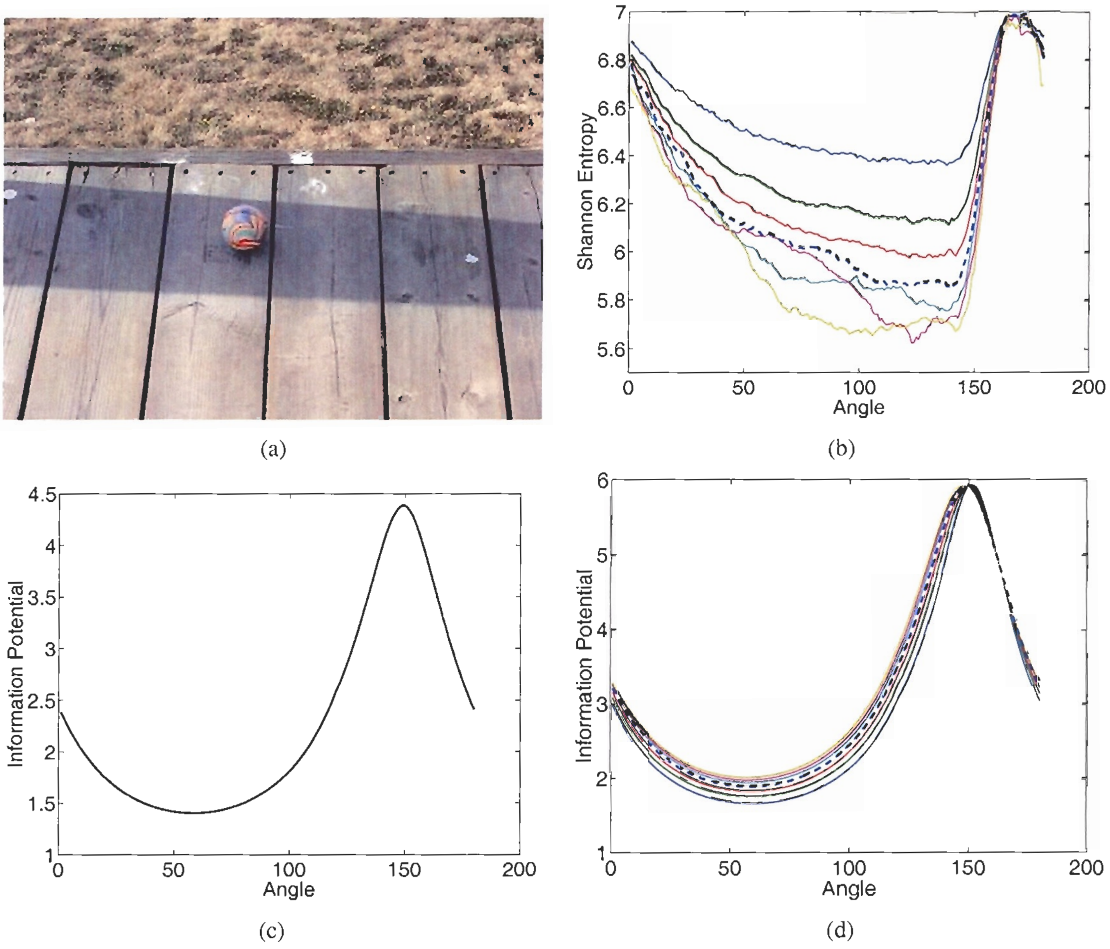


Figure 4.6: Effect of quantization on Shannon's entropy: (a): Original image; (b): Shannon's entropy plot, with changing bin-width (we seek the minimum) – the normative bin-width value given below by Eq. 4.8 is shown dashed, and the other curves are for multipliers of this width from 0.1 to 2.0, mapped to equal maxima; (c): Quadratic entropy plot with bandwidth calculated by minimizing AMISE (we seek the maximum of the quantity plotted). (d): Quadratic entropy plot with bandwidth multiplied by factors from 0.1 to 2.0.

Quantization Problem

Consider the color image in Figure 4.6(a): a colorful ball on a wooden deck is in a shadow cast by strong sunlight. To find the minimum entropy, we again examine projections $\mathcal{I}$ over angles 0° to 180°, for log-chromaticities χ formed according to Eq. (4.2). For each angle, we project the log-chromaticity, and then determine the entropy (4.4). However, the nature of the data, for real images, presents an inherent problem. Since we are considering ratios, we can expect noise to possibly be enhanced, especially for low values. To begin with, therefore, we apply Gaussian smoothing to the original image color channels. But even so, we expect that some ratios may be large. So the question remains as to what we should use as the range, and number of bins, in a histogram of a projected grayscale image $\mathcal{I}$. Using the usual, Shannon, definition of entropy, we cannot escape this quantization issue. However, the alternative Quadratic Entropy measure, used below, largely circumvents this issue by utilizing a different, kernel density driven non-parameteric estimate of the pdf that automatically incorporates smoothness. We still have to choose a bandwidth parameter, but the resulting quantity is much more impervious to our choice.

We calculate Shannon's entropy by approximating the pdf with a histogram over projected 1-d grayscale values. To form an appropriate bin width, we utilize Scott's Rule [103] (discussed below in Appendix 1 on page 74):

$$\text{bin\_width} \; = \; 3.5\, std(\text{projected data})\, N^{1/3} \tag{4.8}$$

where N is the number of invariant image data, for the current angle. Since there may be outlier ratios, we use the middle values only, i.e., the middle 90% of the data, to form a histogram. Note that the size N is therefore different for each angle, since we exclude outliers differently for each projection.

The entropy calculated is shown in Figure 4.6(b); but we know from varying the bin width in Figure 4.6(b) around the value in Eq. 4.8 that this entropy is sensitive to the bin-size. Note here that the image shown in fact actually portrays the worst behaviour of the algorithm, but nonetheless shows up how quantization can lead the algorithm astray. We would like to develop a smoother version of the entropy, retaining a clear indication of the minimum. As well, we would like to dispense with an exhaustive search over angles, for a practical algorithm, and go over to a smoother curve that facilitates efficient search for the minimum.

The deficiencies in using Shannon's entropy can be remedied by going over to a kernel density description of the pdf, using Gaussian kernels to produce a smoother pdf estimate. If we utilize a Quadratic Entropy, rather than Shannon's definition, we can generate and carry out a very fast search for the minimum entropy, since the curve is smooth and also generally has a single extremum. And a Fast Gauss Transform can produce each entropy evaluation in linear time.

Figure 4.6(c) shows the Information Potential, derived from the Quadratic Entropy in the next Section. We see that in this case there is a much simpler curve shape, and local quantization effects are eliminated, promoting a simple maximum search.

4.4.2 Quadratic Entropy and Gauss Transform

We can smooth the plot of entropy versus angle by going over to a kernel-density expansion form: we go over to a probability density formulation by expanding each data point into a region of influence, weighted by a kernel — we use a Guassian kernel. The passage from probability to probability density is sometimes expressed, for the Shannon entropy, by a continuous entropy, or "differential entropy":

$$\eta = -\int p(x)\log p(x)dx \tag{4.9}$$

By the mean-value theorem there exists a value x_i in each bin such that [1]

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)\,dx \tag{4.10}$$

and thus the integral of the function f can be approximated (in the Riemannian sense) by

$$\int_{-\infty}^{\infty} f(x)\,dx = \lim_{\Delta\to 0}\sum_{i=-\infty}^{\infty} f(x_i)\Delta \tag{4.11}$$

where this limit and bin size goes to zero are equivalent.

We will denote

$$H^{\Delta} := -\sum_{i=-\infty}^{\infty} \Delta f(x_i)\log\Delta f(x_i) \tag{4.12}$$

and expanding the logarithm, we have

$$H^{\Delta} = -\sum_{i=-\infty}^{\infty} \Delta f(x_i)\log\Delta f(x_i) = -\sum_{i=-\infty}^{\infty} \Delta f(x_i)\log f(x_i) - \sum_{i=-\infty}^{\infty} f(x_i)\Delta\log\Delta \tag{4.13}$$

As $\Delta \to 0$, we have

$$\sum_{i=-\infty}^{\infty} f(x_i)\Delta \to \int f(x)\,dx = 1 \tag{4.14}$$

and so

$$\sum_{i=-\infty}^{\infty} \Delta f(x_i)\log f(x_i) \to \int f(x)\log f(x)\,dx \tag{4.15}$$

But note that $\Delta \to 0$ as , therefore we need a special definition of the differential or continuous entropy:

$$\lim_{\Delta\to 0}\left[H^{\Delta}+\log\Delta\right] = -\int_{-\infty}^{\infty} f(x)\log f(x)\,dx \tag{4.16}$$

which is, as said before, referred to as the differential entropy. This means that the differential entropy is not a limit of the Shannon entropy for $n \to \infty$. Nevertheless, we easily could include a kernel density formulation in the differential entropy, simply by substituting a kernel expansion for $p(x)$. But we could not then integrate the result. Therefore we go over to a quadratic entropy formulation — this allows us to successfully analytically integrate the expansion, yielding a simple and expressive resulting expression.

Firstly we should replace our pdf over 2-d chromaticity coordinates by a Gaussian kernel density mixture to ensure that entropy is calculated over smooth values. If we go over to a Quadratic Entropy measure (a special case of Renyi's entropy [98]), then the entropy takes on a very simple form.

In 1-d, Renyi's entropy reads

$$\eta_\alpha \;=\; \frac{1}{1-\alpha}\log\int p^{\alpha}(x)dx, \qquad \alpha \geq 0,\ \alpha \neq 1 \tag{4.17}$$

where $p(x)$ is the pdf. This measure is known to approach Shannon's entropy as α goes to 1.

For the special case of $\alpha = 2$ we have

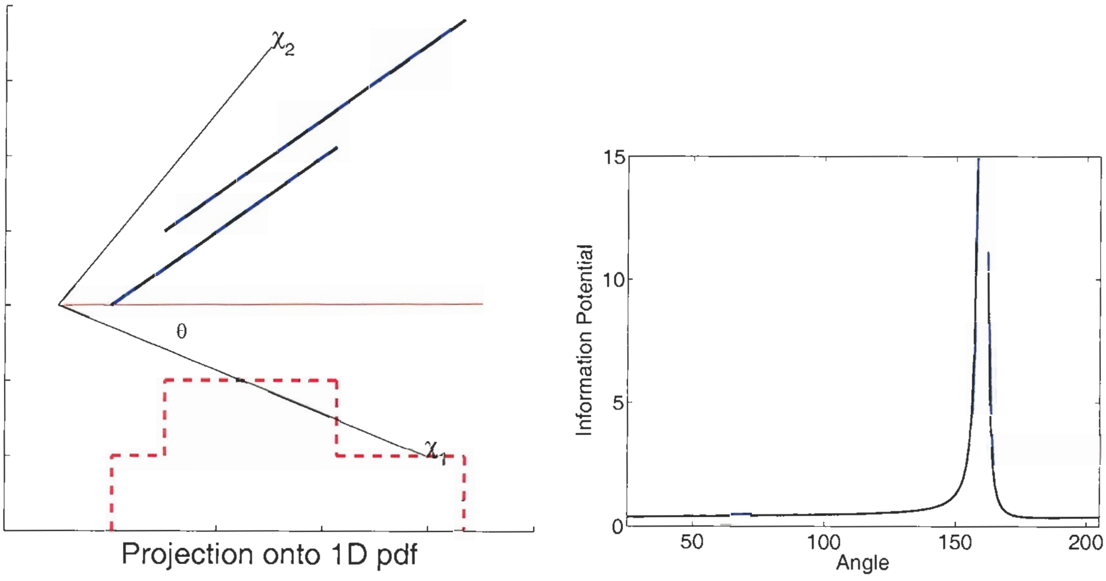
$$\eta_{quadratic} \;=\; -\log\int_{-\infty}^{\infty} p^2(x)dx \tag{4.18}$$

Notice that the log is outside the integral, making for a much simpler evaluation.

For the purposes of optimization, we can simply drop the log, giving the so-called *information potential* V,

$$V = \int p^2(x)dx \tag{4.19}$$

To see how the information potential is indeed related to minimizing the entropy, consider the parallel lines in (χ_1, χ_2) space formed for two paint patches, as illumination changes, in Figure 4.7(a). Suppose data points are uniformly distributed along each line, so that the projected, marginal pdf is proportional to 1 except when the line projections overlap, when the pdf is proportional to 2. Clearly, there is a singularity when the lines project to zero length along the projection axis, at the minimum-entropy angle. Figure 4.7(b) shows the theoretical value of V, as the projection angle changes. The information potential for real data usually also has a similar strong, single-maximum structure: the curve in Figure 4.4(c) showing the information potential for the measured patch data in Figure 4.4(a) has a maximum at 160.5°, whereas the correct angle is 158.5°.



(a) (b)

Figure 4.7: (a): Projection of two illumination-variation lines into 1-d marginal pdf by projection in θ direction. (b): Resulting (continuous) information potential for the quadratic entropy shows strong, single maximum at correct angle (where the singularity occurs).

Compared to Figure 4.4(b), the quadratic entropy has a much cleaner structure that facilitates a fast search by successive evaluation of the quadratic entropy over a few angles.

The quadratic entropy is explicitly evaluated using the Parzen window technique, below.

Parzen Window

The utility of the qudratic-entropy approach lies in approximating the pdf $p(x)$ from its N samples a_i by a Parzen window estimator [89] using Gaussian kernels G, with mean a_i and variance s^2:

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} G(a_i, s^2) \tag{4.20}$$

Since a convolution of two Gaussians is a Guassian with variance equal to the sum of variances of the constituent Gaussians and mean given by the difference of individual means, the information potential V becomes simply [117]

$$V = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(a_i - a_j, 2s^2) = \frac{1}{N^2}\frac{1}{\sqrt{2\pi(2s^2)}}\sum_i\sum_j e^{-\frac{(a_i-a_j)^2}{4s^2}} \tag{4.21}$$

Notice that now we can evaluate the entropy directly from the data, without the need to create a pdf first.

Now let us show how the 1D model above comes out of projecting 2D, (χ_1, χ_2) chromaticity-space data in a direction θ. For convenience let us write $x = \chi_1$, $y = \chi_2$. where X_i, Y_i is the 2D data and i indexes the image treated as a vector. If the 2D pdf is approximated as

$$p(x,y) = \frac{1}{N}\frac{1}{\sqrt{(2\pi)^2 s_1^2 s_2^2}}\sum_{i=1}^{N} \exp\left[-\frac{(x-X_i)^2}{2s_1^2}\right]\exp\left[-\frac{(y-Y_i)^2}{2s_2^2}\right], \qquad (4.22)$$

then to find the marginal probability density for this function along an axis μ, projected in the θ direction, we substitute $x = \mu\cos\theta + \nu\sin\theta$, $y = -\mu\sin\theta + \nu\cos\theta$. Also defining $M_i = X_i\cos\theta - Y_i\sin\theta$, after some algebra the projected marginal probability comes out to be

$$p_\theta(\mu) = \int_{\nu=-\infty}^{\infty} p\left(x(\mu,\nu),\, y(\mu,\nu)\right) d\nu = \frac{1}{N}\frac{1}{\sqrt{2\pi\tilde{s}^2}}\sum_i \exp\left[-\frac{(\mu-M_i)^2}{2\tilde{s}^2}\right] \qquad (4.23)$$

with

$$\tilde{s}(\theta)^2 = s_1^2\cos\theta + s_2^2\sin\theta$$

Thus the information potential is

$$V = \int_{\mu=-\infty}^{\infty} \{f_\theta(\mu)\}^2 d\mu = \frac{1}{N^2}\frac{1}{\sqrt{2\pi(2\tilde{s}^2)}}\sum_i\sum_j \exp\left[-\frac{(M_i-M_j)^2}{4\tilde{s}^2}\right] \qquad (4.24)$$

Therefore, the information potential is given by a simple sum, along the projected axis. For values s_1, s_2 we use the the value that minimizes the asymptotic mean integrated squared error (AMISE) [103], given by

$$s_1 = 1.06\, std(X)\, N^{-1/5}, \quad s_2 = 1.06\, std(Y)\, N^{-1/5} \qquad (4.25)$$

The information potential V can be regarded as the total potential energy of the data set, with the Gaussians in the role of potential energy of data point M_i in the potential field of data point M_j. The information potential plot calculated is shown in Figure 4.6(c); and we know from varying the values s_1, s_2 in Figure 4.6(d) around the value in Eq. 4.25 that this information is robust to the values s_1, s_2. In our application, the data is fixed, so we simply evaluate Eq. (4.24).

The sum V can be calculated in linear time, using the Fast Gauss Transform. Since all quadratic entropy curves found are simple and smooth, finding the maximum via a simple search routine is a fast, straightforward task. We have found that generally, real image data generates a maximum in just a few search steps.

4.5 Summary

In this chapter, we have presented an automatic invariant direction estimation algorithm. In this joint work, we proposed using entropy minimization to determine the invariant direction from images that arise from cameras, without calibration. We begin by adopting the Shannon definition of entropy to evaluate the entropy for each 1-d projection. We verify that the entropy is a strong indicator for finding the invariant. To facilitate efficient and stable search for the minimum, we have proposed using the quadratic entropy definition instead of Shannon entropy. Since the calculation of the Shannon entropy has been shown to be sensitive to the choice of bin width, we consider as future work using a crossvalidation scheme to obtain an optimum bin width. Moreover, our algorithm applies the Fast Gauss Transform to calculate the quadratic entropy in linear running time. Figure 5.4 shows invariant images resulted from projecting the original image to the invariant direction found by the minimum quadratic entropy. We see that the method is effective.

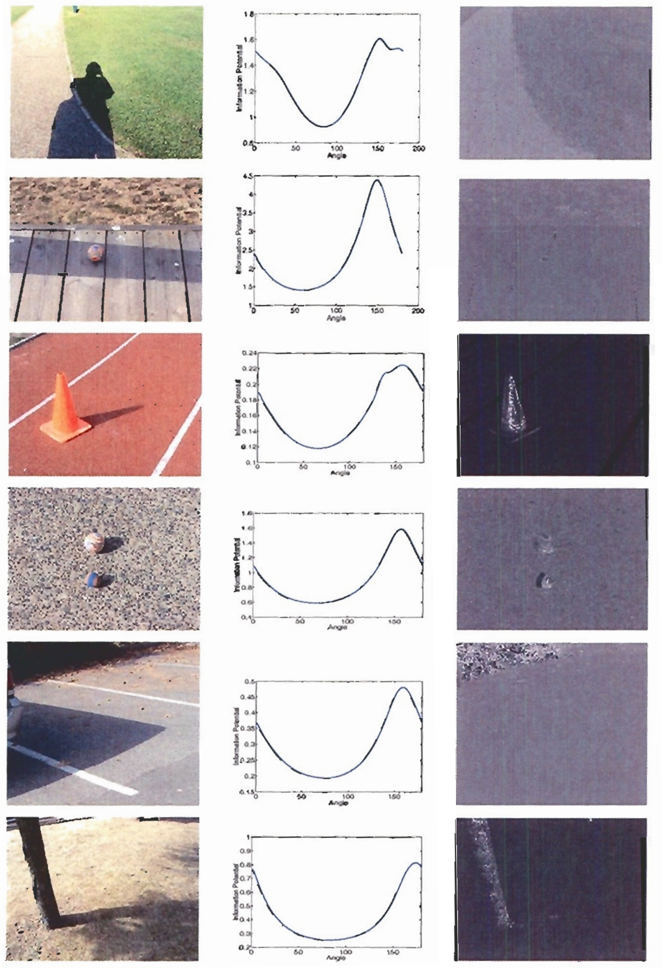


Figure 4.8: Quadratic entropy minimization: columns show original image, quadratic entropy plot and invariant image.

Appendix 1: Scott Rule

The histogram is calculated using the set of data points $X_1, X_2, ..., X_n$. We must choose a bin width h which determines the smoothness of the histogram. We denote the $k-$th bin as B_k and represent the number of data points that fall into the $k-$th bin by v_k. The 1-d histogram at a point x is defined as

$$\hat{f}_{Hist}(x) \;=\; \frac{v_k}{nh} = \frac{1}{nh}\sum_{i=1}^{n} I_{B_k}(X_i); \quad x \;\in\; B_k, \tag{4.26}$$

where $I_{B_k}(X_i)$ is the indicator function

$$I_{B_k}(X_i) \;=\; \begin{cases} 1 & X_i \text{ in } B_k \\ 0 & X_i \text{ not in } B_k \end{cases} \tag{4.27}$$

We now look at how to choose the bin width h. We first describe the mean squared error at a given point in the domain of the function. We can find the mean squared error (MSE) of the estimate $\hat{f}(x)$ at a point x from the following

$$MSE[\hat{f}(x)] \;=\; E[(\hat{f}(x) - f(x))^2] \tag{4.28}$$

where $f(x)$ represents the probability density function. Also, we can determine the error over the domain for x by integrating. This gives us the mean integrated squared error (MISE):

$$MISE \;=\; E[\int(\hat{f}(x) - f(x))^2 dx] \tag{4.29}$$

Using some assumptions, Scott [103] provides the following upper bound for the MSE

$$MSE(\hat{f}(x)) \;\leq\; \frac{f(\xi_k)}{nh} + \xi_k^2 h^2 \tag{4.30}$$

where

$$hf(\xi_k) = \int_{B_k} f(t)dt; \quad \text{for some } \xi_k \;\in\; B_k. \tag{4.31}$$

This upper bound shows what happens to the density estimate when the bin width h is varied. We can try to minimize the MSE by varying the bin width. If we want to measure the error in our estimate for the entire function, then we can integrate over all values of x. Let's assume $f(x)$ has an absolutely continuous and square-integrable first derivative. If we let n get very large, then the *asymptotic MISE* is:

$$AMISE_{Hist}(h) \;=\; \frac{1}{nh} \;+\; \frac{1}{12}h^2 R(f'), \tag{4.32}$$

where $R(g) = \int g^2(x)dx$ is used as a measure of the roughness of the function, and f' is the first derivative of $f(x)$. The first term of AMISE indicates the asymptotic integrated variance, and the second term refers to the asymptotic integrated squared bias. These are obtained as approximations to the integrated squared bias and integrated variance [103]. Thus AMISE indicates a trade-off between bias and variance, as the smoothing parameter h changes.

The optimal bin width h^* for the histogram is obtained by minimizing the AMISE, and is given by

$$h^* = \left(\frac{6}{nR(f')}\right)^{1/3} \tag{4.33}$$

where

$$R(f') = \frac{1}{4\sigma^3\sqrt{\pi}} \tag{4.34}$$

Combining these two expression, we obtain the following expression for the optimal bin width

$$h^* = \left(\frac{24\sigma^3\sqrt{\pi}}{n}\right)^{1/3} \approx 3.5\sigma n^{-1/3} \tag{4.35}$$

Scott [103] proposed the sample standard deviation s as an estimate of σ, resulting in the following Scott bin width rule:

$$\hat{h}^* = 3.5sn^{-1/3}. \tag{4.36}$$

Appendix 2: Fast Gauss Transform applied to Quadratic Entropy

In practice, computation of the information potential can be expensive. When computed naively, computation of V has complexity $O(N^2)$, where N is the number of pixels. This cost may be prohibitive when the image consists of a large number of pixels.

The Fast Gauss Transform (FGT) was introduced by Greengard and Strain [61] for efficient evaluation of a weighted sum of Gaussians. It has proved to be a very efficient algorithm in a variety of applications [8, 29, 118]. The discrete Gauss transform, here discussed as the approximation of a 1-d pdf, is to be evaluated on T grid points:

$$G(x_i) = \sum_{j=1}^{N} w_j \, e^{-\left(\frac{x_i - s_j}{\sigma}\right)^2} \qquad i = 1, \ldots, T. \tag{4.37}$$

Here, w_j are weight coefficients, $\{s_j\}$, $j = 1 \ldots N$ are the data point centers of the sum of Gaussians (the *sources*), and σ is a bandwidth parameter. The sum of Gaussians is evaluated only at a set of grid points $\{x_i\}$, $i = 1 \ldots T$ (the *targets*). A direct computation evaluating the sum of N source points at T targets, requires $O(TN)$ exponential evaluation operations.

The FGT algorithm speeds up the computation by approximation of the Gaussian function to achieve a desired precision. The basis of the fast algorithm is the expansion of the Gaussian in terms of the Hermite functions $h_n(x)$:

$$G(x) = e^{-(x-s)^2} = \sum_{n=1}^{p} \frac{s^n}{n!} h_n(x) + \epsilon(p), \tag{4.38}$$

where $h_n(x)$ is defined by

$$h_n(x) = (-1)^n \frac{d^n}{dx^n} e^{-x^2},$$

and ϵ is the error introduced by truncating the Hermite series after p terms. This is a rephrasing of the Taylor series about $s = 0$.

The FGT starts by dividing the feature space (the sources) into uniform boxes with side length σ. Then the Hermite expansion is applied such that the influence of sources and targets separates. For each source s, the Gaussian can be expanded using a shifted and scaled version of Hermite functions which are located at the center s_B of the box in which the source lies.

$$\begin{aligned} e^{-\left(\frac{x-s}{\sigma}\right)^2} &= e^{-\left(\frac{x-s_B-(s-s_B)}{\sigma}\right)^2} \\ &\cong \sum_{n<p} \frac{1}{n!} \left(\frac{s-s_B}{\sigma}\right)^n h_n\left(\frac{x-s_B}{\sigma}\right) \end{aligned} \tag{4.39}$$

This is a so-called far-field expansion, in that it is an approximation not dependent on the distance between source and target being small.

In a similar manner, the target Gaussian field can be approximated by a Hermite expansion about the center of the target box x_B:

$$\begin{aligned} e^{-\left(\frac{x-s}{\sigma}\right)^2} &= e^{-\left(\frac{x-x_B-(s-x_B)}{\sigma}\right)^2} \\ &\cong \sum_{n<p} \frac{1}{n!} h_n\left(\frac{s-x_B}{\sigma}\right) \left(\frac{x-x_B}{\sigma}\right)^n \end{aligned} \tag{4.40}$$

The two expansions are identical, except that the role of sources and targets are interchanged. Eq. (4.40) is a so-called near-field expansion, in that it expresses a function with target x as a Taylor series about a *nearby* target box center x_B.

The FGT first calculates the expansion coefficients in Eq. (4.39) (the monomials in s) and adds them for each source box, yielding a single expansion for each source box. These series are then shifted to the centers of target boxes using Eq. (4.40), for the Hermite series in each target box and its nearby source box. Thus each target point has only one Taylor expansion with monomials $((x - x_B)/\sigma)^n$. In this way, a sum of Gaussians can be computed in $O(T + N)$ operations.

The FGT is typically applied to Gaussian kernel evaluations where the targets are *not well-behaved* near the sources, making it necessary to use the far field Hermite expansion and the translation to a local Taylor expansion. However, here we wish to use the FGT specifically for calculating the information potential, as in Eq. (4.24). Here, sources and targets are identical — M_i are sources, with the term for each i of the form (4.37) with $x_j \equiv M_j$. In this case, the Hermite expansion is equivalent to the Taylor expansion, with no need to perform the conversion from the Hermite expansion to the local Taylor series. Therefore, *a simpler* evaluation is possible for Eq. (4.24): all points are transformed into a Hermite expansion about the centers of the boxes, and these expansions are directly evaluated at each point.

Formally, the kernel in V can be expressed as a Hermite series:

$$e^{-\frac{(M_i-M_j)^2}{4\tilde{s}^2}} \cong \sum_{n<p} \frac{1}{n!}\left(\frac{M_j - M_B}{2\tilde{s}}\right)^n h_n\left(\frac{M_i - M_B}{2\tilde{s}}\right) \tag{4.41}$$

where point M_j is located in a box B with center M_B and side length $\tilde{s}$. The Fast Gauss Transform for computing the information potential (4.24) thus consists of the following steps:

Step 1 Assign the N data points into uniform boxes with length $\tilde{s}$.

Step 2 Choose p sufficiently large to enforce a desired error precision. The error due to the truncation of the series Eq. (4.39) after p terms satisfies the following bound in this 1-d case [7,8,61]:

$$\left| e^{-\left(\frac{M_i-M_j}{2\tilde{s}}\right)^2} - \sum_{n<p} \frac{1}{n!}\left(\frac{M_j - M_B}{2\tilde{s}}\right)^n h_n\left(\frac{M_i - M_B}{2\tilde{s}}\right) \right| \le \left(\frac{1}{p!}\right)^{\frac{1}{2}} \left(\frac{\sqrt{2}}{4}\right)^p . \tag{4.42}$$

Step 3 For each box B, with center M_B, sum the Hermite polynomials, i.e. add corresponding coefficients:

$$A_n(B) = \frac{1}{n!} \sum_{M_j \in B} \left(\frac{M_j - M_B}{2\tilde{s}}\right)^n \tag{4.43}$$

Step 4 For each point M_i, compute the influence of all points M_j by adding the Hermite expansion for each box B.

$$\begin{aligned}\sum_j e^{-\left(\frac{M_i-M_j}{2\tilde{s}}\right)^2} &= \sum_B \sum_{M_j\in B} e^{-\left(\frac{M_i-M_j}{2\tilde{s}}\right)^2} \\ &\cong \sum_B \sum_{n\leq p} A_n(B) h_n\left(\frac{M_i-M_B}{2\tilde{s}}\right) \qquad (4.44)\end{aligned}$$

Because of the exponential decay of the Gaussian, points in a given box will have no effect (given a particular accuracy) on far-away targets. Thus it is reasonable to compute the influence of only a range of nearby boxes for each target point, where the range is determined by the desired error bound. If we take only the r closest boxes for a point in each direction (i.e., a neighborhood of $2r+1$ boxes centered at the point), it can be shown [61] that we incur an error bounded by $e^{-r^2/4}$. Denoting the $2r+1$ nearby boxes by $IR(B)$, the summation can be approximated by

$$\sum_B \sum_{M_j\in B} e^{-\left(\frac{M_i-M_j}{2\tilde{s}}\right)^2} \cong \sum_{IR(B)} \sum_{n\leq p} A_n(B) h_n\left(\frac{M_i-M_B}{2\tilde{s}}\right) \qquad (4.45)$$

Step 5 Finally, compute the information potential by adding all the Gaussian approximations obtained in step 4.

$$V \cong \sum_i \sum_{IR(B)} \sum_{n\leq p} A_n(B) h_n\left(\frac{M_i-M_B}{2\tilde{s}}\right). \qquad (4.46)$$

In step 3, each point contributes to exactly one expansion, so that the amount of work required to calculate the coefficients for all boxes is $O(Np)$. The amount of work required in step 4 is $O(p(2r+1))$ for each point, and $O(Np(2r+1))$ in total for all points. The desired precision ϵ dictates our choice of r and p. For calculating the information potential, the precision required is moderate, so that we can have small r and p. In this paper we use $r=6$ and $p=6$. Overall, the FGT algorithm achieves linear running time $O(N)$.

Chapter 5

Chromaticity and full color shadow-free images

5.1 Overview

We consider how to put some of the color back in to the 1-d shadow-free representation in this chapter. That is, we can re-light all image pixels uniformly (using, e.g., the illumination in the non-shadow region of the original image) so that the image remains shadow-free but is closer in color to a 2-d representation of the original image (e.g., a chromaticity image, with a 2-vector at each pixel).

We show how to recover a full-color 3-d image representation which is the same as the original image but with shadows removed [42]. To do so, we reason that a shadow edge corresponds to any edge which is in the original image but absent from the invariant representation, and we can thus define a thresholding operation to identify the shadow edge. Edges are in-painted across the shadow edge, and re-integrating yields a color image, equal to the original save for the fact that it is shadow-free. We guarantee that the edge map we are integrating satisfies the integrability condition.

In Section 5.2, we introduce the Drew and Finlayson's work [26] for producing 2-d shadow-free chromaticity images, which aims to put the chromaticity into the invariant grayscale images so as to form shadow-free chromaticity images. In Section 5.3, we present an intrinsic image algorithm which solves the Poisson equation on a modified gradient space for each color band such that 3-d color shadow-free images can be recovered. This step follows the work [38], but we propose a shadow edge-map location scheme and guarantee that the edge map we are integrating satisfies the integrability condition; this will be explained in Section 5.3.2 and 5.3.3, respectively.

5.2 2-d shadow-free chromaticity images

In the 1-d invariant representation described above we removed shadows but at a cost: we have also removed the color information from the image. In the rest of this section we investigate how we can put this color information "back in" to the image. Our aim is to derive an image representation which is shadow-free but which also has some color information. We begin by observing that the 1-d invariant can equally well be expressed as a 2-d log-chromaticity. We derived an invariant quantity $\underline{\chi}$ by projecting 2-d log-chromaticities onto the line in the direction $\underline{e}^{\perp}$. Equally, we can represent the point to which a pixel is projected by its 2-d co-ordinates in the log-chromaticity space, thus retaining some color information. That is, we derive a 2-d color illumination invariant as:

$$\underline{\tilde{\chi}} = P_{\underline{e}^{\perp}} \underline{\chi} \tag{5.1}$$

where $P_{\underline{e}^{\perp}}$ is the 2×2 projector matrix:

$$P_{\underline{e}^{\perp}} = \frac{\underline{e}^{\perp} \underline{e}^{\perp T}}{\|\underline{e}^{\perp}\|^2} \tag{5.2}$$

$P_{\underline{e}^{\perp}}$ takes log-chromaticity values onto the direction orthogonal to $\underline{e}$ but preserves the resulting quantity as a 2-vector $\underline{\tilde{\chi}}$. The original 1-d invariant quantity $\mathcal{I}$ is related to $\underline{\tilde{\chi}}$ by:

$$\mathcal{I} = \underline{\tilde{\chi}} \cdot \underline{e}^{\perp} \tag{5.3}$$

To visualize the 2-d invariant image it is useful to express the 2-d chromaticity information in a 3-d form. To do so, we write the projected chromaticity 2-vector $\underline{\tilde{\chi}}$ that lies in a plane orthogonal to $\underline{u} = (1,1,1)^T$ in its equivalent 3-space co-ordinates $\underline{\tilde{\rho}}$. We do this by multiplying by the 3×2 matrix U^T which decomposes the projector onto that plane:

$$\underline{\tilde{\rho}} = U^T \underline{\tilde{\chi}} \tag{5.4}$$

where $UU^T = I - \underline{u}\underline{u}^T / \|\underline{u}\|^2$ and the resulting $\underline{\tilde{\rho}}$ is a 3-vector. Exponentiating, we recover an approximation of color:

$$\underline{\tilde{p}} = \exp(\underline{\tilde{\rho}}) \tag{5.5}$$

Note that Eq. (5.5) is a 3-dimensional representation of 2-d information: $\underline{\tilde{\rho}}$ contains no brightness or shading information and so is still effectively a chromaticity representation. The usual way to derive an intensity independent representation of 3-d color is to normalize a 3-d sensor response $\underline{\rho}$

(a) (b) (c)

by the sum of its elements [68]. We take our 3-d representation over to this form by applying an L_1 normalization:

$$\underline{p} = \{\tilde{p_1}, \tilde{p_2}, \tilde{p_3}\}^T / (\tilde{p_1} + \tilde{p_2} + \tilde{p_3}) \tag{5.6}$$

We have found that this representation has the advantage over the grayscale invariant $\mathcal{I}$ of greater stability, and it is bounded in [0,1].

An illustration of the method is shown in Figure 5.1 (All three images in Figure 5.1 are derived from Figure 1.1. Figure 5.1a shows the L_1 chromaticity representation $\underline{r}$ of an image, with intensity and shading information factored out: $\underline{r} = \{R, G, B\}/(R + G + B)$. It is important to note that in this representation the shadow is still visible — it represents a change in the color of the illumination and not just its intensity. Figure 5.1b shows the illumination invariant chromaticity representation derived in Eq. (5.1)-(5.6) above. Now the shadow is no longer visible, indicating that the method has successfully removed the shadow whilst still maintaining some color information.

Comparing Figures 5.1a and 3b we see that the colors in the two images are quite different. This is because the representation in Figure 5.1c has had all its illumination removed and thus it is in effect an intrinsic reflectance image. To recover a color representation closer to that in Figure 5.1b we must put the illumination back into the representation [26]. Of course, we don't want to add illumination back on a pixel-by-pixel basis since this would simply reverse what we have just done and result in an image representation which once again contains shadows. To avoid this we want to re-light each pixel uniformly by "adding back" illumination. To see how to do this, consider again the 2-d chromaticity representation defined in Eq. (5.1). In this representation illumination is represented by a vector of arbitrary magnitude in the direction $\underline{e}$:

$$\text{illumination } = \underline{\chi}_E = a_E \underline{e} \tag{5.7}$$

We can put this light back into the illuminant invariant representation defined in Eq. (5.1) by simply adding the chromaticity of the light to the invariant chromaticities:

$$\tilde{\underline{\chi}} \rightarrow \tilde{\underline{\chi}} + \underline{\chi}_E = \tilde{\underline{\chi}} + a_E \underline{e} \tag{5.8}$$

The color of the light we put back in is controlled by the value of a_E. To determine which light to add back in we observe that the pixels in the original image that are brightest, correspond to surfaces that are not in shadow. It follows then that if we base our light on these bright pixels then we can use this light to re-light all pixels. That is, we find a suitable value of a_E by minimising

$$||\underline{\chi}_b - (\tilde{\underline{\chi}}_b + a_E \underline{e})|| \tag{5.9}$$

where $\underline{\chi}_b$ and $\tilde{\underline{\chi}}_b$ correspond to the log-chromaticity and the invariant log-chromaticity of bright (non-shadow) image pixels. Once we have added the lighting back in this way we can represent the resulting chromaticity information in 3-d by applying Eq. (5.6).

Figure 5.1c shows the resulting chromaticity representation with lighting added back in. Here we found a_E by minimizing the term in Eq.5.9 for the brightest 1% of pixels in the image. The colors are now much closer to those in the conventional chromaticity image (Figure 5.1b) but are still not identical. The remaining difference is due to the fact that when we project chromaticities orthogonally to the illuminant direction we remove illumination as well as any part of a surface's color which is in this direction. This part of the object color is not easily put back into the image. Nevertheless, for many surfaces the resulting chromaticity image is close to the original, with the advantage that the representation is shadow-free. Figure (5.3) shows this shadow-free chromaticity representation for a variety of different images. In all cases, shadows are successfully removed.

5.3 3-d shadow-free images

The 2-d chromaticity representation of images is often very useful. By additionally removing shadows from this representation we have gained a further advantage and increased the value of a chromaticity representation. For example, it has been shown in [71] that a shadow-free representation can help in object tracking. However, there is still room for improvement. Chromaticity images lack shading and intensity information and are also unnaturally colored. In some applications an image which is free of shadows but which is otherwise the same as a conventional color image would be very useful. In this section we consider how such an image might be obtained.

5.3.1 The Recovery Algorithm

Our method for obtaining full color shadow removal has its roots in methods of lightness recovery [9,66,69,77]. Lightness algorithms take as their input a 3-d color image and return two intrinsic images: one based on reflectance (the lightness image) and the other based on illumination. Lightness computation proceeds by making the assumption that illumination varies slowly across an image whereas changes in reflectance are rapid. It follows then that by thresholding a derivative image to remove small derivatives, slow changes (due, by assumption, to illumination) can be removed. Integrating the thresholded derivative image results in the lightness intrinsic image. An alternative set of algorithms for lightness recovery would be the Multi-scale Retinex (MSR) [72,73] which considered a multi-scale, pyramid model of Retinex.

Importantly, a lightness scheme will not remove shadows since although they are a change in illumination, at a shadow edge the illumination change is fast, not slow. Given their assumptions, lightness algorithms are unable to distinguish shadow edges from material edges. However, in our case we have the original image which contains shadows and we are able to derive from it 1-d or 2-d images which are shadow-free. Thus by comparing edges in the original and the shadow-free images we can identify those edges which correspond to a shadow. Modifying the thresholding step in the lightness algorithm leads to an algorithm which can recover full-color shadow-free images. There are two important steps which must be carefully considered if the algorithm is to work in practice. First, the algorithm is limited by the accuracy with which we can identify shadow edges. Second, given the location of the shadow edges we must give proper consideration to how this can be used in a lightness type algorithm to recover the shadow-free image.

Let us begin by defining the recovery algorithm. We use the notation $\rho_k(x,y)$ to denote the grey-scale image corresponding to a single band of the 3-d color image. Lightness algorithms work by recovering an intrinsic image from each of these three bands separately and combining the three intrinsic images to form a color image. We observe in Chapter 3 that under the assumption of Dirac delta function sensors, sensor response is a multiplication of light and surface. Let us transform sensor responses into log space so that the multiplication becomes an addition:

$$\rho_k'(x,y) = \sigma'(x,y) + E'(\lambda_k,x,y) + S'(\lambda_k,x,y) + q_k' \tag{5.10}$$

In the original lightness algorithm the goal is to remove illumination and, as a first step towards this, directional derivatives are calculated for the log-image:

$$\nabla_x \rho_k'(x,y) = \frac{\partial}{\partial x}\rho_k'(x,y)$$

$$\nabla_y \rho'_k(x,y) = \frac{\partial}{\partial y}\rho'_k(x,y) \tag{5.11}$$

These directional derivatives define edge maps for the log image. Next, a threshold operator $T(\cdot)$ is defined to remove gradients of small magnitude:

$$T(\nabla_i \rho'_k(x,y)) = \begin{cases} 0 & \text{if } \|\nabla_i \rho'_k(x,y)\| < \tau \\ \\ \nabla_i \rho'_k(x,y) & \text{otherwise} \end{cases} \tag{5.12}$$

where $i \in \{x, y\}$ and τ is the chosen threshold value.

In our case the goal is not to remove illumination *per se* (the small values in (5.12) above) but rather we wish only to remove shadows. In fact we actually want to keep the illuminant field and re-render the scene as if it were captured under the same single non-shadow illuminant. To do this we must factor out changes in the gradient at shadow edges. We can do this by modifying the threshold operator defined in (5.12). In principle, identifying shadows is easy: we look for edges in the original image which are not present in the invariant representation. However, in practice the procedure is somewhat more complicated than this. For now, let us assume that we have identified the shadow edge and leave a discussion of how we find it to the next section. Let us define a function $q_s(x,y)$ which defines the shadow edge:

$$q_s(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ is a shadow edge} \\ \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

We can then remove shadows in the directional derivatives of the log image using the threshold function $T_S(\cdot)$:

$$T_S(\nabla_i \rho'_k, q_s(x,y)) = \begin{cases} 0 & \text{if } q_s(x,y) = 1 \\ \\ \nabla_i \rho'_k & \text{otherwise} \end{cases} \tag{5.14}$$

where again $i \in \{x, y\}$. That is, wherever we have identified that there is a shadow edge we set the directional derivatives in the log-image to zero, indicating that there is no change at this point (which is true for the underlying reflectance). After thresholding we obtain directional derivatives where sharp changes are indicative only of material changes: there are no sharp changes due to illumination and so shadows have been removed.

We now wish to integrate edge information in order to recover a log-image which does not have shadows. We do this by first taking directional derivatives of the thresholded edge maps we have just defined to form a modified (by the threshold operator) Laplacian of the log-image:

$$\nabla^2_{T_S}\rho'_k(x,y) = \begin{array}{c} \nabla_x T_S\left(\nabla_x \rho'_k(x,y), q_s(x,y)\right) \\ +\nabla_y T_S\left(\nabla_y \rho'_k(x,y), q_s(x,y)\right) \end{array} \tag{5.15}$$

Now, let us denote the shadow-free log-image which we wish to recover as $\tilde{\rho}'(x,y)$ and equate its Laplacian to the modified Laplacian we have just defined:

$$\nabla^2 \tilde{\rho}'_k(x,y) \;= \nabla^2_{T_S}\rho'_k(x,y) \tag{5.16}$$

i.e., a Poisson equation. Thus, the shadow-free log-image can be calculated via:

$$\tilde{\rho_k}'(x,y) = \left(\nabla^2\right)^{-1}\nabla^2_{T_S}\rho'_k(x,y) \tag{5.17}$$

However, since the Laplacian is not defined at the image boundary without boundary conditions, we must specify these for uniqueness. Blake [9] made use of Neumann boundary conditions, in which the normal derivative of the image is specified at its boundary. Here we use homogeneous Neumann conditions: the directional derivative at the boundary is set to zero.

There are two additional problems with recovering $\tilde{\rho_k}'(x,y)$ according to Eq. (5.17) caused by the fact that we have removed shadow edges from the image. First, because we have modified the edge maps by setting shadow edges to zero we can no longer guarantee that the edge map we are integrating satisfies the integrability condition. That is, the edge map should be integrable (cf. [52]):

$$\nabla_y \nabla_x \rho'_k(x,y) = \nabla_x \nabla_y \rho'_k(x,y) \tag{5.18}$$

The second problem is caused by the fact that to ensure shadows are effectively removed, we must set to zero edges in quite a large neighborhood of the actual shadow edge. As a result edge information pertaining to local texture in the neighborhood of the shadow edge is lost and the resulting (shadow-free) image is unrealistically smooth in this region. To avoid this problem, rather than simply setting shadow edges to zero in the thresholding step we apply an iterative diffusion process which *fills in* the derivatives across shadow edges, bridging values obtained from neighboring non-shadow edge pixels. We also deal with the problem of integrability at this stage by including a step at each iteration to enforce integrability, as proposed in [48]. We will present the process of projecting onto integrable gradients in Section 5.3.2.

This iterative process is detailed below where t denotes artificial time:

1. Initialization, $t = 0$, calculate:

$$(\nabla_x \rho'_k(x,y))^t \rightarrow T_S(\nabla_x \rho'_k(x,y), q_s(x,y))$$

$$(\nabla_y \rho'_k(x,y))^t \rightarrow T_S(\nabla_y \rho'_k(x,y), q_s(x,y))$$

2. Update shadow edge pixels (i,j):

$$(\nabla_x \rho'_k(i,j))^t \rightarrow (\nabla_x \rho'_k(i-1,j))^{t-1} + (\nabla_x \rho'_k(i,j-1))^{t-1}$$

$$+(\nabla_x \rho'_k(i+1,j))^{t-1} + (\nabla_x \rho'_k(i,j+1))^{t-1}$$

$$(\nabla_y \rho'_k(i,j))^t \rightarrow (\nabla_y \rho'_k(i-1,j))^{t-1} + (\nabla_y \rho'_k(i,j-1))^{t-1}$$

$$+(\nabla_y \rho'_k(i+1,j))^{t-1} + (\nabla_y \rho'_k(i,j+1))^{t-1}$$

3. Enforce integrability by projection onto integrable edge map [48], and integrate:

$$F_x(u,v) = \mathcal{F}[\nabla_x \rho'_k], \quad F_y(u,v) = \mathcal{F}[\nabla_y \rho'_k],$$

$$a_x \;=\; e^{2\pi i u/N} - 1\,, \;\; a_y \;=\; e^{2\pi i v/M} - 1,$$

$$Z(u,v) = \frac{a_x^* F_x(u,v) + a_y^* F_y(u,v)}{|a_x|^2 + |a_y|^2}\,, \quad \rho'(0,0) = 0,$$

$$(\nabla_x \rho')^t = \mathcal{F}^{-1}\left[a_x Z\right], \quad (\nabla_y \rho')^t = \mathcal{F}^{-1}\left[a_y Z\right]$$

where image size is $M \times N$ and $\mathcal{F}[\cdot]$ denotes the Fourier Transform. Here we use a forward-difference derivative $\{-1, 1\}$, corresponding to the a_x, a_y above in the Fourier domain: i.e., the Fourier transform of a derivative $\nabla_x Z$ in the spatial domain corresponds to multiplication by $a_x(u)$ in the Fourier domain — this result simply follows by writing $\rho'(n+1) - \rho'(n)$ in terms of Fourier sums in the Discrete Fourier Transform (DFT). The projection step follows [48], but for a forward-difference operator. We will present how to derive $Z(u,v)$ in Section 5.3.2.

4. if $\|(\nabla_x \rho')^t - (\nabla_x \rho')^{t-1}\| + \|(\nabla_y \rho')^t - (\nabla_y \rho')^{t-1}\| \geq \epsilon$,
$t \rightarrow t+1$, goto 2.

where ϵ defines the stopping criterion.

Finally, we then solve the Poisson equation (5.16) using a final round of enforcing integrability by projection as above, with the re-integrated image given by

$$\tilde{\rho}'_k(x,y) = \mathcal{F}^{-1}\left[Z(u,v)\right] \tag{5.19}$$

We actually operate on an image four times the original size, formed by symmetric replication in x and y, so as to enforce periodicity of the data for the DFT and homogeneous Neumann boundary conditions.

Eq. (5.17) recovers $\tilde{\rho}'_k(x,y)$ up to an unknown constant of integration. Exponentiating $\tilde{\rho}'_k(x,y)$, we arrive at the reconstructed grey-scale image $\tilde{\rho}_k(x,y)$ (up to an unknown *multiplicative* constant). Solving (5.17) for each of the three color bands results in a full color image $\underline{\tilde{\rho}} = \{\tilde{\rho}_1 \quad \tilde{\rho}_2 \quad \tilde{\rho}_3\}^T$ where the shadows are removed.

To fix the unknown multiplicative factors, we apply a mapping to each pixel which maps the brightest pixels (specifically, the median of the top 1-percentile of pixels ordered by brightness) in the recovered image to the corresponding pixels in the original image.

5.3.2 Projection onto integrable gradient

The gradient is formed as a thresholded version of the correct gradient or is the result of a gradient-diffusion process. Therefore the ostensible gradient is not actually the gradient of a real image, but is only an approximation. Therefore the gradient is not integrable.

Suppose we have available the gradient $\nabla_x I$, $\nabla_y I$ of a grayscale image I. Let the Fourier domain version of the gradient be called $F_x(u,v)$, $F_y(u,v)$. Also, denote the Fourier transform of the image we seek by $Z(u,v)$:

$$Z(u,v) \;\equiv\; \mathcal{F}(I) \tag{5.20}$$

In order for the gradient (calculated here in the Fourier domain) to be integrable, it must be formed as the transform of the image, multiplied by $a_x(u,v)$ and $a_y(u,v)$ to effect the derivative operation, carried out in Fourier space.

So we would like to find an image transform $Z(u,v)$ such that the available gradient transform $\{F_x,\ F_y\}$ is as close as possible to $\{a_x Z(u,v),\ a_y Z(u,v)\}$. Therefore we should minimize a Least-Squares objective function:

$$\overset{min}{Z} \int\!\!\int [a_x^* Z^* \;-\; F_x^*][a_x Z \;-\; F_x] \;+\; [a_y^* Z^* \;-\; F_y^*][a_y Z \;-\; F_y] dx\, dy \tag{5.21}$$

Taking a variational derivative with respect to Z^* and setting this to zero, we have the Euler-Lagrange equation

$$a_x^*[a_x Z \;-\; F_x] \;+\; a_y^*[a_y Z \;-\; F_y] \;=\; 0 \tag{5.22}$$

so that a solution for $Z(u, v)$ must solve

$$(a_x^* a_x \;+\; a_y^* a_y)Z = (a_x^* F_x \;+\; a_y^* F_y) \tag{5.23}$$

or, rewriting, the solution for $Z(u, v)$ is

$$Z(u, v) \;=\; \frac{a_x^* F_x \;+\; a_y^* F_y}{a_x^* a_x \;+\; a_y^* a_y} \tag{5.24}$$

Then the gradient itself is given by

$$F_x(u, v) \;\Rightarrow\; a_x(u, v)\, Z \;,\; F_y(u, v) \;\rightarrow\; a_y(u, v)\, Z \tag{5.25}$$

This solution is in fact a *projection* onto the nearest integrable gradient, since substituting the new expressions for $F_x(u, v)$ and $F_y(u, v)$ in Eq. (5.25) into the image solution Eq. (5.24) we arrive at the same solution $Z(u, v)$: the projection operation is idempotent in that projecting twice (or more) times gives the same result.

5.3.3 Locating shadow edges

To complete the definition of the recovery algorithm we must specify how to identify shadow edges. The essential idea is to compare edge maps of the original image to those derived from an invariant image and to define a shadow edge to be any edge in the original which is not in the invariant image. We could start by calculating edge maps as simple finite difference approximations to directional derivatives,

$$\begin{aligned} \nabla_x \rho_I(x, y) &= \rho_I(x, y) \otimes \{-1,\ 0,\ 1\}/2 \\ \nabla_y \rho_I(x, y) &= \rho_I(x, y) \otimes \{-1,\ 0,\ 1\}^T/2 \end{aligned} \tag{5.26}$$

where $\rho_I(x, y)$ is the intensity image, taken here as the L_1 norm of the original image: $\rho_I = (1/3)(\rho_1 + \rho_2 + \rho_3)$, and $\otimes$ is the convolution operation. Unfortunately, as Figure 5.2(a) illustrates, finite differencing produce non-zero values at more locations than those at which there are true edges. Thus, while in the example in Figure 5.2(a) the edges of the road and the shadow are clear, so too are many edges due to the texture of the imaged surfaces as well as noise in the image. Obtaining the true edges in which we are interested from these edge maps is non-trivial, as evidenced by the large literature on edge detection (see [70] for a review).

For a more careful approach, we begin by applying a smoothing filter (specifically the Mean-Shift algorithm proposed in [18]) to both the original image and the 2-d invariant image derived by exponentiating the invariant log image. This has the effect of suppressing features such as noise and high frequency textures so that in subsequent processing fewer spurious edges are detected. Then, we replace simple differencing by the Canny edge detector [15], returning estimates for the strength of horizontal and vertical edges at each image location:

$$\begin{aligned} \|\tilde{\nabla}_x \rho_i(x,y)\| &= C_x\left[\rho_i(x,y)\right] \\ \|\tilde{\nabla}_y \rho_i(x,y)\| &= C_y\left[\rho_i(x,y)\right] \end{aligned} \tag{5.27}$$

with $C_x[\cdot]$ and $C_x[\cdot]$ denote the Canny (or any other well-behaved) operators for determining horizontal and vertical edges respectively.

We determine an edge map for the invariant image in a similar way, first calculating horizontal and vertical edge strengths for each channel of the 2-d invariant image:

$$\begin{aligned} \|\tilde{\nabla}_x \chi_k(x,y)\| &= C_x\left[\chi_k(x,y)\right] \\ \|\tilde{\nabla}_y \chi_k(x,y)\| &= C_y\left[\chi_k(x,y)\right] \end{aligned} \tag{5.28}$$

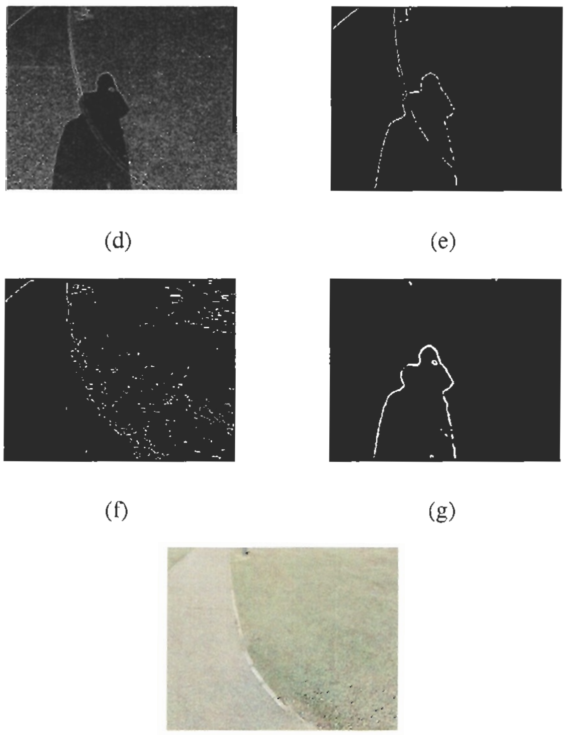
The edge maps from the two channels are then combined by a max operation:

$$\begin{aligned} \|\tilde{\nabla}_x \tilde{\underline{\chi}}(x,y)\| &= \max\left(C_x[\tilde{\chi}_1(x,y)], C_x[\tilde{\chi}_2(x,y)]\right) \\ \|\tilde{\nabla}_y \tilde{\underline{\chi}}(x,y)\| &= \max\left(C_y[\tilde{\chi}_1(x,y)], C_y[\tilde{\chi}_2(x,y)]\right) \end{aligned} \tag{5.29}$$

where $\max(\cdot,\cdot)$ returns the maximum of its two arguments at each location (x,y).

We use two criteria to determine whether or not a given edge corresponds to a shadow. First, if at a given location the original image has a strong edge but the invariant image has a weak edge, we classify that edge as a shadow edge. Second, if both the original image and the invariant image have a strong edge, but the orientation of these edges is different, then we also classify the edge as a shadow edge. Thus our shadow edge map is defined as:

$$q_s(x,y) = \begin{cases} 1 & \text{if } \|\tilde{\nabla}\rho_i\| > \tau_1 \;\;\&\;\; \|\tilde{\nabla}\underline{\chi}\| < \tau_2 \\ & \text{or } \left|\frac{\|\tilde{\nabla}_x\rho_i\|}{\|\tilde{\nabla}_y\rho_i\|} - \frac{\|\tilde{\nabla}_x\underline{\chi}\|}{\|\tilde{\nabla}_y\underline{\chi}\|}\right| > \tau_3 \\ 0 & \text{otherwise} \end{cases} \tag{5.30}$$



(d)

(e)

(f)

(g)

(h)

Figure 5.1: (a) An edge-map obtained using simple finite differencing operators. (b) Edges obtained using the Canny operator on the Mean-Shifted original image. (c) Edges obtained using the Canny operator on the Mean-Shifted 2-d invariant image. (d) The final shadow edge. (e) The recovered shadow-free color image.

where τ_1, τ_2, and τ_3 are thresholds whose values are parameters in the recovery algorithm. As a final step, we employ a morphological operation (specifically, two dilations) on the binary edge map to "thicken" the shadow edges:

$$q_s(x, y) \rightarrow (q_s(x, y) \oplus D) \oplus D \tag{5.31}$$

where $\oplus$ denotes the dilation operation and D denotes the structural element, in this case the 8-connected set. This dilation has the effect of filling in some of the gaps in the shadow edge. Figure 5.2(d) illustrates a typical example of a recovered shadow edge map $q_s(x, y)$. It is clear that even after the processing described the definition of the shadow edge is imperfect: there are a number of spurious edges not removed. However, this map is sufficiently accurate to allow recovery of the shadow-free image shown in Figure 5.2(e) based on the integration procedure described above.

5.4 Summary

We introduced two different shadow-free image representations in this chapter: a 2-d chromaticity representation which is equivalent to the 1-d representation but with some color information retained (which is Finlayson and Drew's work in [26]) and a 3-d full color image for which we proposed a shadow edge-map location scheme and guarantee that the edge map we are integrating satisfies the integrability condition. Figure 5.3 shows some examples of these different representations for a number of different images. In each example all the representations are shadow-free. In all cases we need to determine the direction of illumination change by using the entropy minimization outlined in Chapter 4. In summary, we conclude that the approach to shadow removal yields very good performance. In both cases (2-d and 3-d) the recovered images are of a good quality.

However, recovering the 3-d representation is more complex and there are a number of free parameters in the recovery algorithm. We must choose suitable thresholds in the edge detection step using the Canny operator (Eqs. (5.27) and (5.28)) and also when determining $q_s(x, y)$ in Eq. (5.30). The images in Figure 5.3 were obtained using a fixed set of parameters and, although the recovered shadow edge is not always perfect, the resulting shadow-free image is in all cases of good quality. We note however, that the algorithm in its current form will not deliver perfect shadow-free images in all cases. In particular, images with complex shadows, soft or diffuse shadows with poorly defined edges will likely cause problems for the algorithm. However, the current algorithm is robust when shadow edges are clear and we are currently investigating ways to improve the algorithm's performance on the more difficult cases. In addition, it is possible for the method to misclassify some edges in the original image as shadow edges. For example, if two adjacent surfaces differ in intensity, an edge detector will find an edge at the border of these two surfaces. However, in the 1-d invariant image intensity differences are absent, and so no edge will be found in this case. Thus, the edge between the two surfaces will wrongly be classified as a shadow edge. Indeed, the fifth example in Figure 5.3 exhibits such behavior: The boundary between the painted white line on the road surface, and the road surface itself, is not fully recovered, because the two surfaces (paint and road) differ mainly in intensity. A similar problem can arise if adjacent surfaces are related by a color change in the direction in which illumination changes. Here again, an edge will be found in the original image, but will be absent from the invariant images. The examples in Figure 5.3 (and the many other images we have processed) suggest that such problems arise only infrequently in practice. However, in future work, we intend to investigate ways to overcome these problems. In summary, we conclude that the approach to shadow removal proposed in this thesis yields very good performance. In all three cases (1-d, 2-d, and 3-d), the recovered images are of a good quality and we envisage that they will be of practical use in a variety of visual tasks such as segmentation, image retrieval, and tracking. As well, the method raises the possibility of enhancing commercial photography such as portraiture.

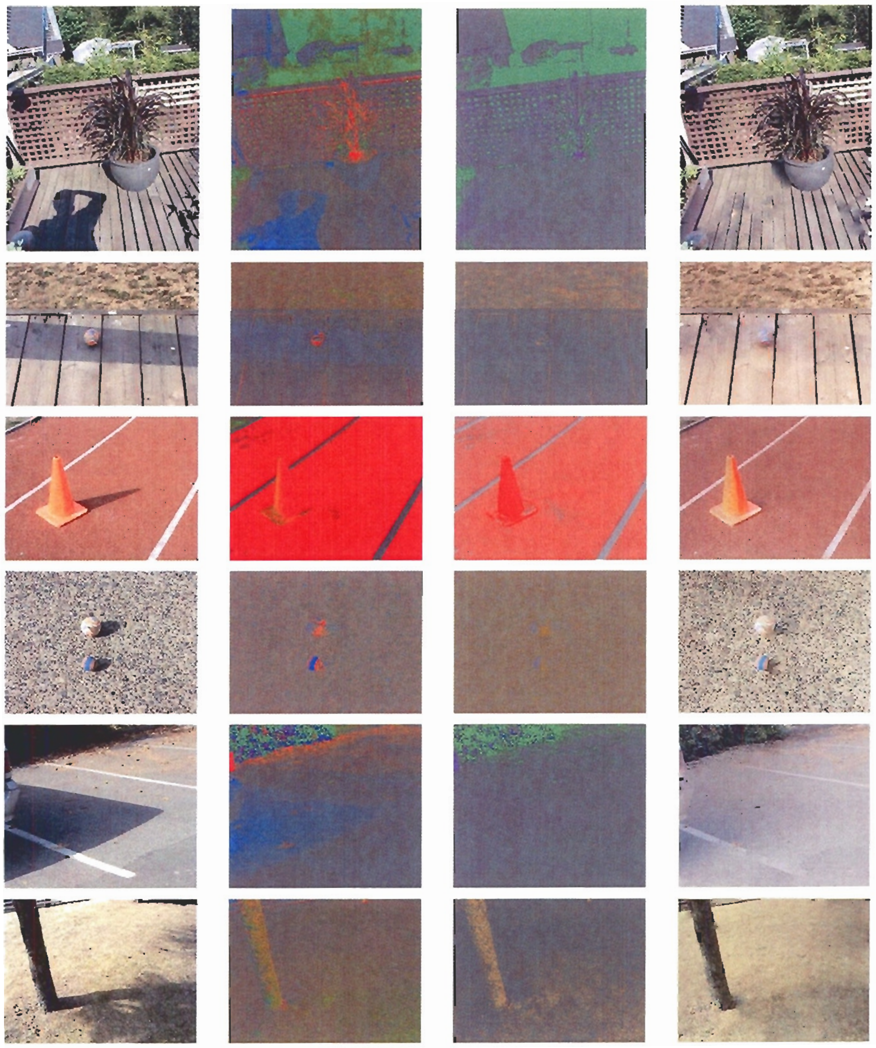


Figure 5.2: Additional invariant images, using entropy minimization: columns show original image, L_1 chromaticity image, invariant L_1 chromaticity, and re-integrated color image.

Chapter 6

Shadows and Illuminants from Flash/No-flash Pairs

6.1 Overview – Shadow Detection and Illuminant Estimation via Flash/No-flash Image Pairs

Flash/no-flash image pairs have been studied as a simple mechanism for supplying less-noisy information for dark areas of an ambient-light image. So far in this thesis, we consider the problem of detecting shadows in the image taken under ambient light (i.e., room lighting, window light, interreflected light), given extra information from a flash image registered with the first. We first examines the log-difference image for flash/no-flash pairs, showing that an intrinsic illumination difference image can be generated with no surface reflectance term confounding lighting change. We then recover the shadow mask for the ambient image, via projecting log-difference values onto the plane they mostly occupy and then detecting large differences by a robust procedure. Finally, we shows how to copy flash information over to the ambient image, inside the shadow mask, and then re-integrate by combining gradients to guarantee integrability. The result is an ambient image with no or greatly attenuated ambient shadowing. In this way, we arrive at an image with the advantages of the ambient-only image — warmth, no flash effects such as disturbing illumination dropoff with distance and pixel saturation etc. — but no shadows [81].

As shadows represent changes of scene illuminant, shadow detection indeed has a tight relation with illuminant estimation. We therefore present a practical method that can estimate the ambient illuminant using the flash image as a reference illuminant image. Once the ambient illuminant is recovered, we also carry out a simple white balance, using a pre-determined white patch under the ambient illuminant as a reference white, removing the effects of any automatic camera white balance procedure.

Both shadow detection and illuminant estimation use the combination of flash and no-flash images. However, for consumer-grade digital cameras, for different illumination conditions the two images usually have different camera settings automatically applied when they are taken, such as exposure time and white balance. We propose a camera-setting compensation method which can adjust the image pairs such that such that the difference between compensated images reflects only the difference in illumination.

In Section 6.2, we describe an approach to log image formation under flash/ambient lighting which uses the three assumptions of Planckian lighting, Lambertian surfaces and narrow-band camera sensors, as proposed in Chapter 3. In Section 6.3 we propose an algorithm to detect ambient shadows with the aid of the flash image. In Section 6.4, we present an ambient illuminant estimation method by which the ambient illuminant can be associated with one of a set of different illuminants. Finally, in Section 6.5, we propose a camera-setting compensation scheme such that the adjusted ambient/flash images properly have differences from illumination only.

6.2 Image Formation under Flash/Ambient Lighting

The light impinging on a surface point is of course quite different in an image taken under ambient lighting and under a combination of both ambient plus a flash. For clarity, let us refer to the first image as "Ambient" and the second as "Both" (A and B) (see Figure 6.1). If we control the camera settings, or at least know them, and assuming there are no saturated pixels, blooming effects etc. [95], then $(B - A)$ should yield an image as if it were taken under the flash only (assuming one adjusts overall pixel magnitudes to compensate for camera settings, as suggested in [91]). This is due to the fact that, assuming camera linearity, the B image consists of reflected light from the ambient sources plus from the flash. Of course, the effective direction for ambient lighting (at each pixel) is from a different direction than from the camera flash. That is, there is a different visibility function for the flash image, and this produces "flash-shadows". Aside from this issue, however, since the pure-flash image as in Figure 6.1(c), which we denote as "Flash" (F), sees reflected light from only the flash illumination, image F has no ambient shadows — no shadows that derive from the ambient (and are usually quite visible in both B as well as in A).

For the image A and F, we consider the simple image formation (Eq. 3.4) introduced in Chapter

3 which uses the three assumptions of Planckian lighting, Lambertian surfaces, and a narrowband camera.

In this approximation, recall that the RGB color R_k, $k = 1 \ldots 3$, is simply given by

$$R_k \;=\; \sigma \; I \; k_1 \lambda_k^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k \; . \tag{6.1}$$

with constants k_1 and k_2. Temperature T characterizes the lighting color, I gives the overall light intensity and σ is Lambertian shading — surface normal dotted into illumination direction — along with visibility.

Let us define the following short-hand notations:

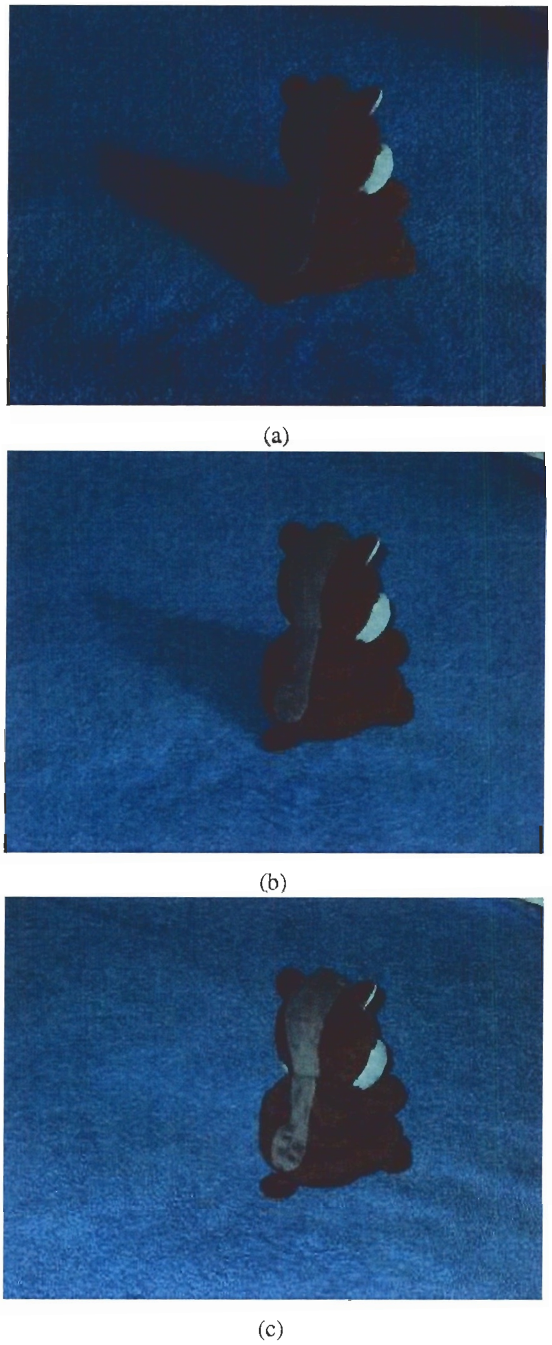
$$\begin{aligned} K &= \log(I k_1 \sigma); \quad & s_k &= \log(S(\lambda)); \\ w_k &= \log(k_1 \lambda_k^{-5} q_k); \quad & e_k &= -k_2/\lambda_k \end{aligned} \tag{6.2}$$

Taking logarithms, Eq. (6.1) becomes

$$\log R_k(\boldsymbol{x}\,) \;=\; w_k \;+\; K(\boldsymbol{x}\,) \;+\; s_k(\boldsymbol{x}\,) \;+\; (1/T(\boldsymbol{x}\,))e_k \tag{6.3}$$

Here, we have explicitly indicated dependence on 2-d pixel location $\boldsymbol{x}$: the 3-vector w_k is a characteristic 3-vector for the camera, as is e_k, and so does not depend on image location. However, the intensity and shading, encapsulated in K, do depend on location, as does the surface term s_k. Lighting color is dependent on the correlated color temperature T, which depends on what lighting the surface point sees and adds up. In shadows, both K and T are different. E.g., in Figure 6.1(a), we certainly would conceive that the intensity term K is different inside the shadow. But so is the lighting temperature T — the shadow region sees a different color of light than does the nonshadowed area. The temperature of the black-body radiator depends on pixel location, but the only way the illuminant can vary with location is if there is an admixture of lights with different color temperatures. Although admixing lights with different color temperatures does not produce a black-body spectrum corresponding to a single color-temperature, the flatness of the Planckian locus in the chromaticity diagram means that the Planckian locus is effectively approximately linear, and so the assumption of a blackbody spectrum for the admixed illuminants is sufficiently satisfied. Here, we assume that lighting is effectively independent of location $\boldsymbol{x}$ in each region: shadow and nonshadow.

A similar analysis, involving these assumptions, was discussed in Chapter 3 to remove shadows from a single image, but using 2-vector band-ratio versions of the quantities in (3.7). Here we consider actual 3-vector color quantities instead of such chromaticities. As we shall see, the advantage of using two images is that whereas the previous method loses the intensity information, the method presented here does not.



(a)

(b)

(c)

Figure 6.1: Sample images: (a): Ambient A, (b): Both B, and (c): Flash F. These images are 12-bit per channel linear consumer digital color camera images, in raw format with no gamma or other processing applied. However, images are shown in the sRGB color space, for display [67].

Using the above simplifying assumptions, we mean to determine how the fact that the ambient shadow is removed from the pure-flash image F can be used to find the ambient-shadows in A.

Log Image Difference

How does the model (6.3) translate into information that can be derived about shadowing and, as well, what can be gleaned from calibration images such as Figure 6.2(a,b)?

From (6.3), we notice that a difference image in log space space can remove both a camera term w_k as well as the surface term $s_k(\boldsymbol{x})$, and this is what we now study.

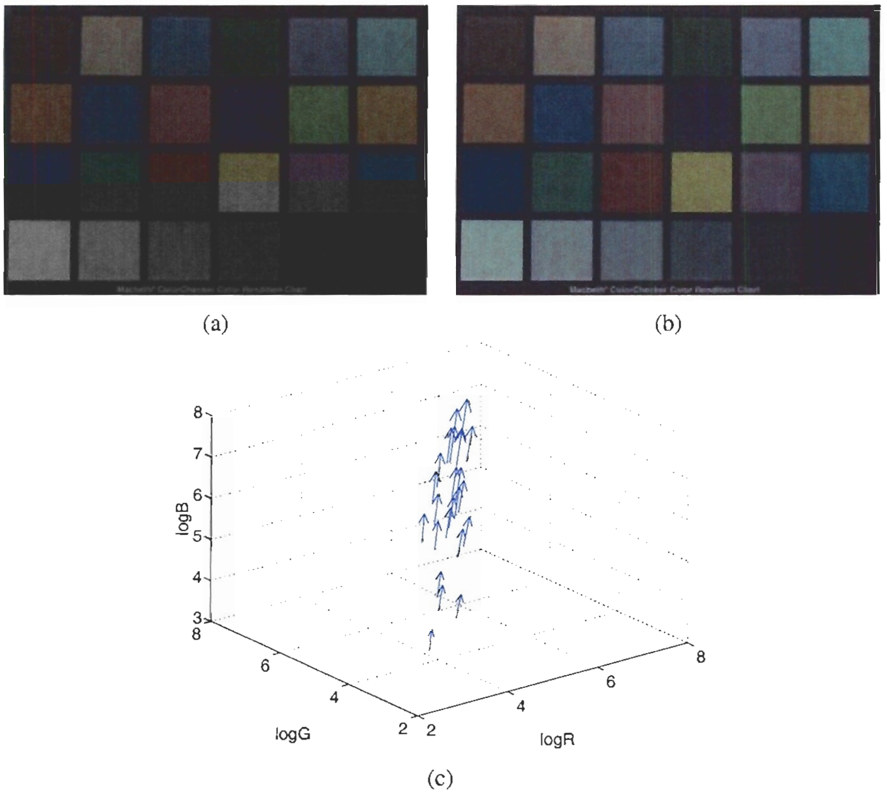
Firstly, note that in general, for any ambient/flash pair the intensity I and hence the scalar K can be different for each image of the pair. Let us form a *ratio image* (cf. [65]) by a difference image in log space: subtracting Eq. (6.3) for two images, A and F, we have

$$\begin{aligned}\log R_k^A(\boldsymbol{x}) - \log R_k^F(\boldsymbol{x}) &= [K^A(\boldsymbol{x}) - K^F(\boldsymbol{x})] \\ + [1/T^A(\boldsymbol{x}) - 1/T^F(\boldsymbol{x})]e_k &\end{aligned} \tag{6.4}$$

for the difference between log pixel values under light A and light F, at pixel indexed by $\boldsymbol{x}$. Notice that the *surface term s_k is entirely removed.* This is the key part of our algorithm: **the log difference image codes illumination difference only** — it is a type of intrinsic image. Consequently we expect to see the difference in illumination due to shadowing to stand out much better.

The ratio image consists of just two parts: (1) the intensity difference (and shading/visibility if that is pertinent — just intensity for a flat calibration pair); and (2) a term proportional to the camera dependent lighting-change 3-vector e_k. Notice that Eq. (6.4) is pixel-wise: the two parts would be different inside a shadow, if one is present, since the lighting color temperature would be different there than in the rest of the image, as would the intensities. This situation is illustrated in Figure 6.3, which shows the basic direction $\boldsymbol{u} = 1/\sqrt(3)(1,1,1)^T$, and the lighting-change direction $\boldsymbol{e}$: all log-difference image pixels live on the plane formed by these two directions, and shadow pixels have a different (constant) value of $(1/T^A - 1/T^F)$ than do pixels in ambient shadow regions. Pixel vectors are formed as the resultant of a vector in the $\boldsymbol{e}$ direction added to a vector in the $\boldsymbol{u}$ direction. Since the $\boldsymbol{e}$ component is constant outside the shadow, we expect non-shadow (i.e., usually most) pixels to approximately inhabit a line parallel to $\boldsymbol{u}$.

The Macbeth chart is flat, and Figures 6.2(a,b) have no shadowing. For this special calibration


(a)

(b)

(c)

Figure 6.2: (a,b): Color target under two different illuminants. (c): Vector differences between paint patches for log colors (in sharpened color space).

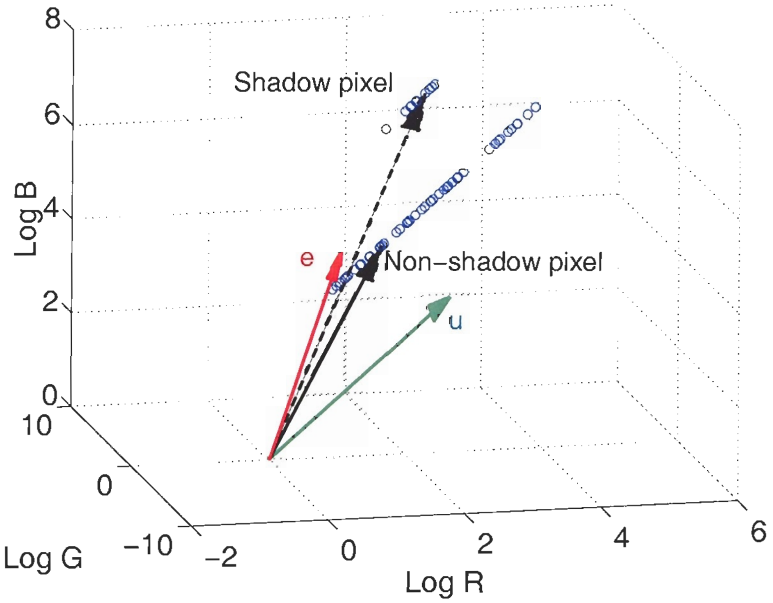


Figure 6.3: Log-difference image pixels consist of a constant times the lighting-change direction, plus an intensity/shading difference term along the $(1, 1, 1)$ direction.

case, Eq. (6.4) reduces to the simpler form

$$\begin{aligned} &\log R_k^{C_1,Macbeth}(\boldsymbol{x}) - \log R_k^{C_2,Macbeth}(\boldsymbol{x}) \\ &= [K^{C_1} - K^{C_2}] + [1/T^{C_1} - 1/T^{C_2}]e_k \end{aligned} \tag{6.5}$$

Thus if we plot the vectors from $\log R_k^{C_1}(\boldsymbol{x})$ to $\log R_k^{C_2}(\boldsymbol{x})$ in a sharpened color space, as in Figure 6.2(c), we expect to see *constant* vectors, given by the inverse color temperature difference $[1/T^{C_1} - 1/T^{C_2}]$ times the lighting-change vector e_k, plus a constant intensity difference $[K^{C_1} - K^{C_2}]$ along the $(1, 1, 1)$ unit vector, as in Eq. (6.5); and indeed the vector differences are nearly equal, verifying the suitability of the model (once sharpening is applied). The vectors in Figure 6.2(c) each start from a different point due to the surface term s_k in Eq.(6.3).

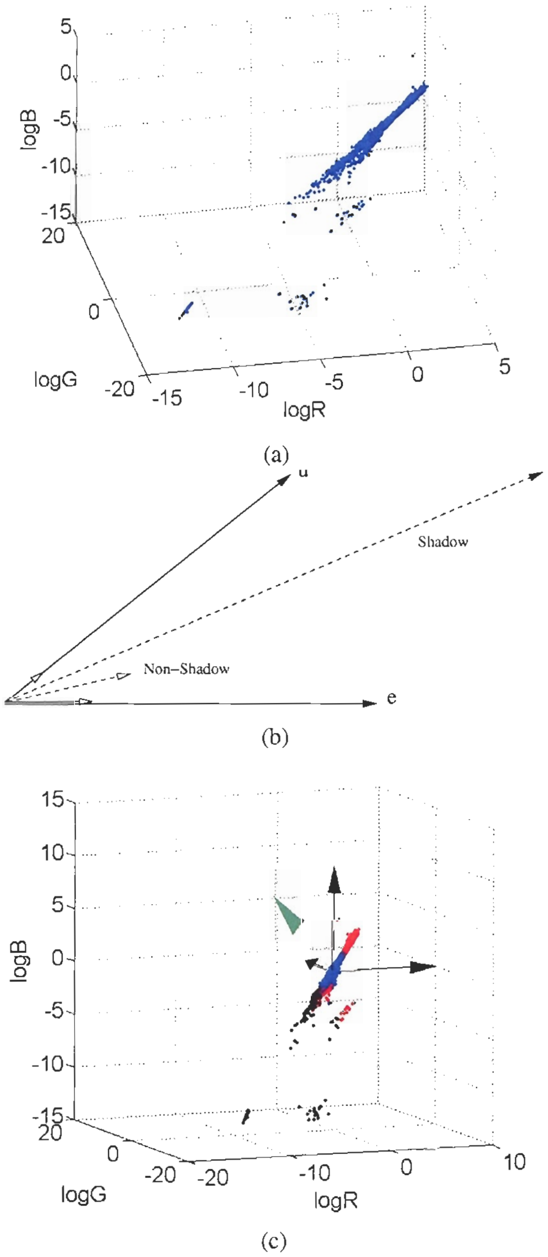
Notice that if we attempt to recover the value of 3-vector e_k from this difference, we are stymied by the intensity term, which multiplies the 3-vector $(1, 1, 1)^T$. So in fact, from the calibration images we can recover only that part of the lighting-change direction e_k which is orthogonal to $(1, 1, 1)$. For a log-difference vector $\boldsymbol{v}$, this part is given by $v_k - (1/3)(v_1 + v_2 + v_3)$. In non-log space, then, this part corresponds to dividing by the geometric mean $(v_1 * v_2 * v_3)^{1/3}$.

6.3 Ambient Shadow Detection

Since the flash image F has no ambient-shadows, we expect to be able to combine F with A, the ambient image, to be able to detect and thence eliminate the ambient-shadows. But we have found that a simple differencing scheme does not work well — while the difference between F and A is indeed large within the ambient-shadow region, this effect is confounded to some degree by the change in color of the two illuminants, ambient and flash illuminations. We wish to address the issue of how to re-cast the difference image, $F - A$, such that ambient-shadows are easily detected.

6.3.1 Shadow-Matte Algorithm

The ambient shadow-finding idea is to find a plane in which most (non-shadow) pixels live, with basis vectors given by the $(1, 1, 1)$ direction $\boldsymbol{u}$ and the lighting-change direction $\boldsymbol{e}$, as in Eq. (6.4). But how well does this play out in real imagery? Figure 6.4(a) shows a 3-d scatterplot of the log-difference data (Flash – Ambient) for the images in Figure 6.1. The singular values of the data are in proportion 1, 0.019, 0.003, showing that the data does roughly fall on a plane. We shall use image Figure 6.1 to illustrate the method in the following.



(a)

(b)

(c)

Figure 6.4: (a): Log-difference image pixels for a typical image (Figure 6.1). (b): Pixels identified as ambient-shadow result from a large log-difference between F and A in either the $\boldsymbol{u}$ or the $\boldsymbol{e}$ directions. (c): Pixels identified as ambient-shadow (red) and flash-shadow (black); the coordinate axes are also shown, along with the perpendicular to the plane which the data inhabit (green vector).

We can best model the log-difference by projecting onto this plane:

$$\boldsymbol{d} \;=\; \boldsymbol{V}\,(\log \boldsymbol{R}^{\,A} - \log \boldsymbol{R}^{\,F}) \tag{6.6}$$

where $\boldsymbol{V}$ is the 2×3 pair of principal component vectors in the plane: $\boldsymbol{d}$ is 2-dimensional. Projecting has the disadvantage of discarding some information. However, if lights are on the Planckian locus in the simplified model, then both nonshadow and shadow pixels inhabit the plane and projecting presumably discards noise.

In fact from Figure 6.4(a), we note that most of the variation in log-difference magnitude is along the $\boldsymbol{u}$ direction (in the sharpened color space). Therefore we apprehend shadow formation as a process that generates large values in the log difference between Flash and Ambient (i.e., the $\boldsymbol{u}$ direction), as well as differences in lighting color (the $\boldsymbol{e}$ direction). The situation is illustrated in Figure 6.4(b). To take both into account we simply use the Euclidean magnitude in the plane. Since the logarithm is monotonic, and we are seeking large differences, this will find outlier values from both causes.

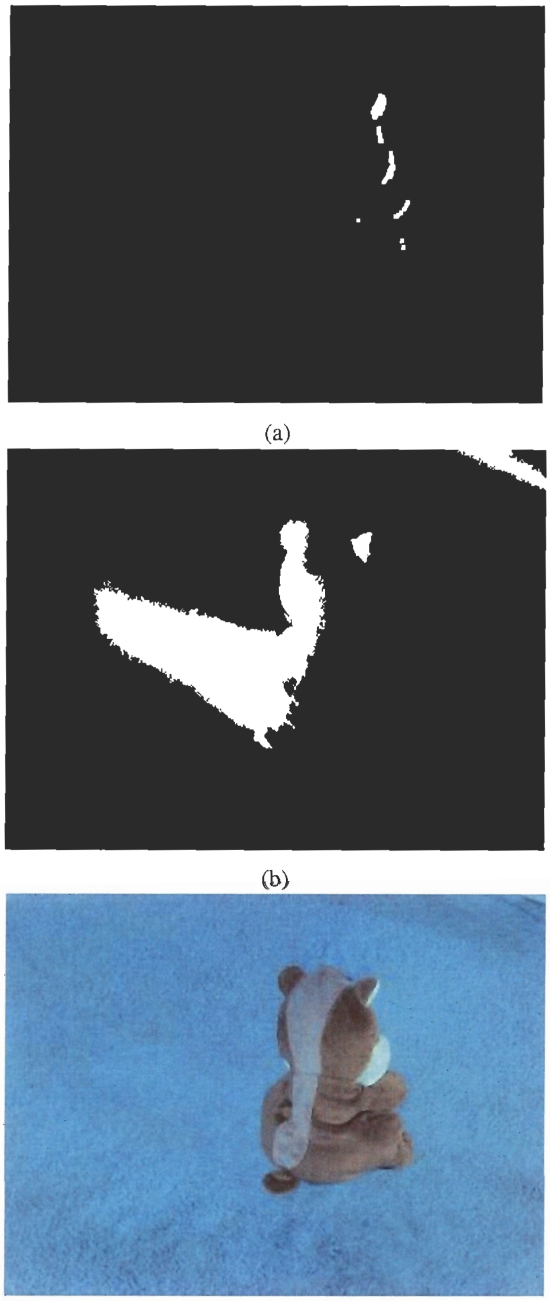
But first we must consider pixels that are in fact in shadows generated by the flash itself. In image F, these pixels are entirely black (or at least very dark). We use a robust method (Least Median of Squares — LMS [99]) to find these pixels, considering a 1-d location mode-finder. The LMS gives outliers automatically, and these are shown in Figure 6.5(a). In general, outliers will be either in flash-shadows, or in specularities, with outliers on the dark side of the mode and specularities on the bright side. We include both kinds of outliers in a "flash-shadow/specularity" mask since we wish to exclude them both. Notice especially that using this robust method means that we need use no thresholds: outlier detection is automatic.

Then from the diagram in Figure 6.4(b), we derive a magnitude $\mathcal{I}$ in the plane as $\mathcal{I} = \|\boldsymbol{d}\,\|$. Again we use the LMS, and the resulting outlier ambient-shadow mask (excluding the flash-shadows) is shown in Figure 6.5(b). As an auxiliary step, we additionally produce a cleaner shadow mask[1] by applying a Mean-Shift segmentation [18] to the $\mathcal{I}$ image, before finding outliers, because the mean shift segmentation gives regions with more complete shapes.

We see by visual inspection that indeed the method finds the shadow pixels in the ambient image Figure 6.1(a).

To see how this image maps back to feature space, consider Figure 6.4(c). Here, pixels identified as ambient-shadow are shown in red, and flash-shadow pixels are in black. Indeed, ambient-shadows

[1] The term "shadow mask" is not that used in the television industry, but instead is a binary map showing where in the image the shadow lies



(a)

(b)

(c)

Figure 6.5: (a): Flash-shadow mask. (b): Shadow mask for ambient shadows. (c): Re-integrated image without ambient or flash shadows.

have large values in both the $\boldsymbol{u}$ and $\boldsymbol{e}$ directions (and are not flash-shadows).

6.3.2 Shadow-Free Ambient Image Recovery

Finding the ambient-shadow mask takes us most of the way toward recovery of a *shadowless* ambient image. To proceed, we first map flash-image pixels to pixels in the ambient image, inside the shadow mask. To do so, we first regress on pixel values not in either the ambient shadow or flash shadow/specularities, from F to A. Then we use the resulting 3×3 matrix $\boldsymbol{T}$ to take flash pixels over to ambient ones. Then a straightforward approach would join the regressed F(in ambient shadow) and A(out of ambient shadow).

However, this does not entirely eliminate the ambient shadows. This is because (i) the regression cannot give perfect color mapping from F to A; and (ii) A has illumination discontinuities that do not usually exactly coincide with the mask boundary. But we may note that *edges* are not discontinuous in each image F and A inside the ambient-shadow mask. Therefore, to seamlessly join the two parts, we can blend the image edges from F and A by combining their gradient fields: if S is the shadow mask, then we define a new field $\log \boldsymbol{C}$ via

$$\begin{aligned} \nabla \log \boldsymbol{C}[S] &= \nabla(\log \boldsymbol{F}[S]\boldsymbol{T}\,) \\ \nabla \log \boldsymbol{C}[\neg S] &= \nabla(\log \boldsymbol{A}[\neg S]) \end{aligned} \qquad . \tag{6.7}$$

Taking another gradient and inverting Poisson's equation to recover $\log \boldsymbol{C}$ will blend the two gradient fields, as Chapter 4, and lead to a full color shadow free image.

Note that when copying over gradients, we can be quite inaccurate with the shadow mask, so long as we at least cover the mask; we found that simply dilating the mask several times helps the re-integration by copying over more of the shadow-free F edges. The re-integrated ambient image for our example image is shown in Figure 6.5(c). Additional results are shown in Figure 6.6.

Altogether, the algorithm for ambient-shadow removal is as follows:

Algorithm:

```
Find sharpening matrix $\boldsymbol{M}$ from camera calibration.
Transform to sharpened color space:
    $A^\sharp = \boldsymbol{M}\ A$; $B^\sharp = \boldsymbol{M}\ B$.
$F^\sharp = B^\sharp - A^\sharp$.
Form logs, and log-difference: $\log F^\sharp - \log A^\sharp$.
Find best 2-d plane for log-difference feature: basis $\boldsymbol{V}$ .
```

```
Project log-difference into 2-d $\boldsymbol{d}$ : Eq. (6.6).
Find flash-shadow/specularity mask $S_F$=
    outliers LMS($||F||$).
Apply mean-shift filtering to $||\boldsymbol{d}\ ||$.
Find ambient-shadow mask $S_A$= outliers LMS($||\boldsymbol{d}\ ||$).
Combine masks: $S = S_A \cap (\neg S_F)$.
Copy color from $A$ into $F$, to set scale for flash image
    edge map:
    $\boldsymbol{P}$ =LS($F[\neg(S_A \cup S_F)], A[\neg(S_A \cup S_F)]$),
    where LS is least-squares regression; then $F' = \boldsymbol{P}\ F$.
Dilate $S$.
Form combined log-image edge map $\log \boldsymbol{C}$ via Eq. (6.7).
In each color channel, we effectively take another
    derivative in the Fourier domain via a phase shift,
    and then project onto an integrable edge
    map whilst undoing the Laplacian.
The inverted Poisson equation is unique only up to an
    additive constant, which we fix by regressing back to
    the ambient image $\log A$, in $(\neg S)$.
Exponentiate to recover non-log image.
```

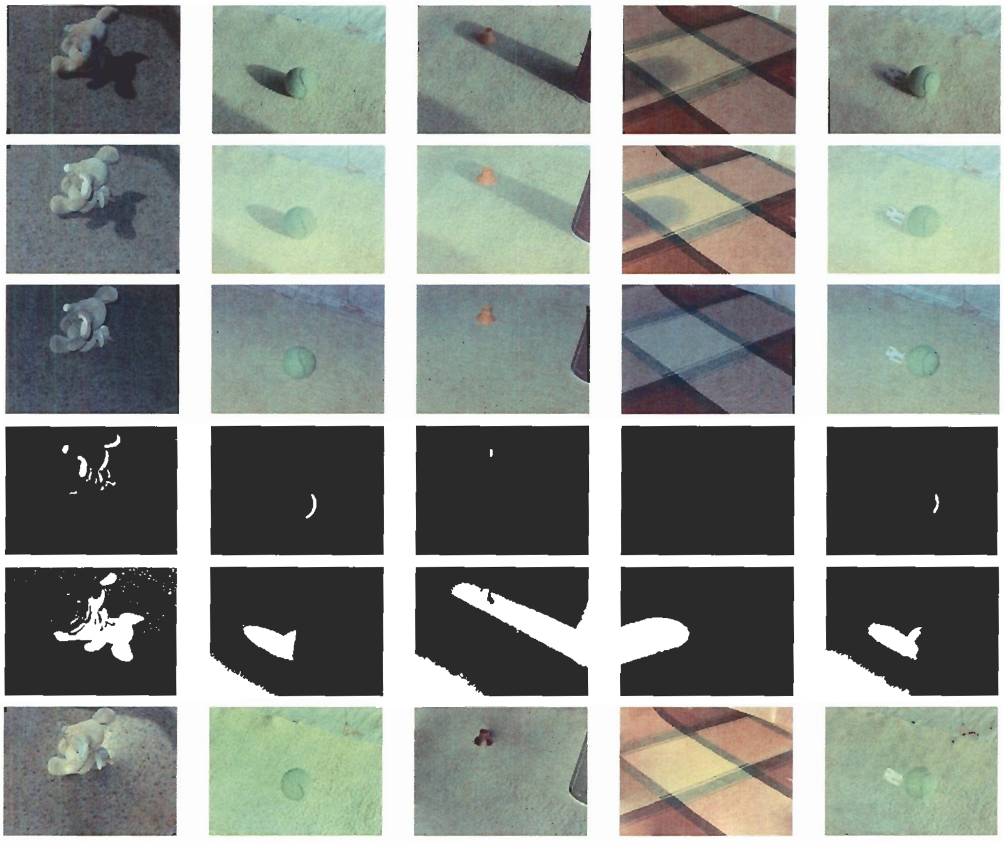


Figure 6.6: Further results: Top row: Ambient image A. 2nd row: Both image B. 3rd row: Flash image F. 4th row: Flash-shadow/specularity mask. 5th row: Ambient-shadow mask. 6th row: Shadowless ambient image.

6.4 Ambient Illuminant Estimation

Accurate estimation of ambient illuminant is useful for imaging applications. In most applications, however, this task is difficult because of the complicated combination of illuminants, surfaces, and camera characteristics during the imaging process. Here we consider the problem of estimating the ambient illuminant directly, with only flash/no-flash pairs, without information on surface reflectance and camera sensors. We show that projecting the difference in a log domain of the pure-flash image and the no-flash image into a geometric-mean chromaticity space, gives the chromaticity of the ambient illuminant. We verify that the chromaticities corresponding to illuminants with different temperatures fall along a line on a plane in the log geometric mean chromaticity space. This line coincides with the Planckian locus (the Planckian locus has a linear behavior in the log chromaticity space). By associating the chromaticity of the difference image to the nearest color temperature along the Planckian locus, we arrive at an estimate of the ambient illuminant [80]. Once the ambient illuminant is recovered, we also carry out a simple white balance, using a pre-determined white patch under the ambient illuminant as a reference white, removing the effects of any automatic camera white balance procedure.

6.4.1 Log-difference Geometric Chromaticity

From (6.4), a log-difference image which is formed by subtracting F from A in log space which removes both the camera term w_k as well as the surface term $s_k(\boldsymbol{x})$. Let us denote the log-difference image as D:

$$D_k^{A-F}(\boldsymbol{x}) = \log \rho_k^A(\boldsymbol{x}) - \log \rho_k^F(\boldsymbol{x}) \tag{6.8}$$

As we had stated, the surface term is entirely removed, leaving a type of *intrinsic illumination* difference image which arises from: (i) the intensity difference (with shading/visibility), (ii) a term proportional to the camera-dependent lighting-change 3-vector e_k. We focus our attention on the illumination temperature only. To remove the intensity difference component, we go over to the geometric mean log chromaticity space according to Eq.(3.18). Notice that Eq.(6.8) is pixel-wise: it would be different for each pixel. Here, we assume that the scene contains a single ambient illuminant temperature (for one or several sources), so the log-difference chromaticity reduces to a simple form:

$$\begin{aligned} r_k^{A-F} &= D_k^{A-F} - 1/3 \textstyle\sum_{j=1}^3 D_j^{A-F} \\ &= [1/T^A - 1/T^F](e_k - e_M); \end{aligned} \tag{6.9}$$

The above equation explicitly gives the log-difference vector as a function of illuminant temperature. For now, we carry all three components of chromaticity. We note that, in log space, r^{A-F} is orthogonal to $\boldsymbol{u} = 1/\sqrt{3}(1,1,1)^T$. I.e., $\boldsymbol{r}$ lives on a plane orthogonal to $\boldsymbol{u}$.

To characterize the 2-d space, we can consider the projector $\boldsymbol{P}_u^\perp$ onto the plane. $\boldsymbol{P}_u^\perp$ has two non-zero eigenvalues, so its decomposition reads

$$P_u^\perp = I - \boldsymbol{u}\,\boldsymbol{u}^T = \boldsymbol{U}^T\boldsymbol{U}, \tag{6.10}$$

where $\boldsymbol{U}$ is a 2×3 orthogonal matrix. $\boldsymbol{U}$ rotates 3-vectors $\boldsymbol{r}$ into a coordinate system *in* the plane:

$$\chi \equiv \boldsymbol{U}\boldsymbol{r}, \quad \chi \text{ is } 2 \times 1. \tag{6.11}$$

Straight lines for illuminant changes for r are still straight in χ, that is, in the $\{\chi_1, \chi_2\}$ plane, we expect to see log-difference images r^{A-F} with different ambient illuminations fall along a straight line through the origin. Recall that the vector $(e_k - e_M)$ is dependent on camera properties and captures the direction of changes of illumination. The scalar $[1/T^A - 1/T^F]$ locates a color temperature position on the line. Figure 6.7 illustrates a log geometric-mean chromaticity diagram for 1931 CIE color matching functions, where the dots aligned along a line are the log-difference geometric-mean chromaticity for a set of flash/no-flash pairs. These image pairs were formed synthetically using 9 Planckian lights, from 2500K to 14500K with interval 1500K, Macbeth ColorChecker 24 surfaces, simple sensors with single impulse responsivities, and a xenon flash.

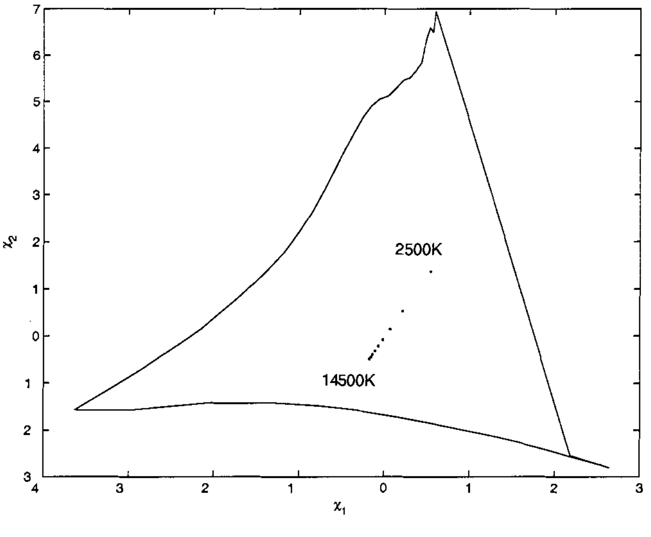


Figure 6.7: CIE log geometric-mean chromaticity diagram: log-difference geometric-mean chromaticity of flash/no-flash pairs under 9 Planckian lights are shown with blue dots.

Note that this color temperature is *not* the correlated color temperature of the ambient illuminant, as it corresponds to the inverse-temperature difference between ambient and flash lights. We know that the flash illuminant is fixed for any camera, so that the temperature position on the line is fixed and can be a *reference temperature* for the ambient illuminant, i.e. the temperature uses the flash light as a reference. Here, we call this line the *reference illuminant temperature locus*. Thus, we can estimate the ambient light in an image by classifying it into one of a set of candidate reference color temperatures. We also carried out this log-difference chromaticity procedure for the Sony DXC930 digital camera, using flash/no-flash color patches created synthetically with Macbeth chart data under illuminants A, C, F2, and the 5 Judd daylights. The result is shown in Figure 6.8(b). This camera has quite narrowband sensors (Figure 6.8(a)) There are visibly 8 clusters, each of them corresponding to one of the 8 illuminants. Among them, illuminants A, C, and the 5 daylights approximately align along the line of Planckian lights (the line is from Figure 6.7); fluorescent illuminant F2 is off the Planckian lights line.

6.4.2 Estimating the ambient illumination

Figure 6.9 illustrates the algorithm flow. First, in the sharpening phase, the constrained spectral sharpening process is performed and all camera responses RGB are transformed to a sharpened space. In the training step, using the sharpened camera the pairs of images of a reflectance database (e.g. the Macbeth ColorChecker) under a set of sample illuminants are taken with flash turned on and turned off; the flash/no-flash pair is registered and the pure-flash image F is calculated; in log space, the difference log image $log(A) - log(F)$ projected to the $\{\chi_1, \chi_2\}$ plane in the geometric-mean chromaticity space gives each illuminant a reference temperature along the reference temperature locus. In the estimating step for a new, unknown, image pair, we carry out the same process as in the training phase, then recover the temperature for the ambient light along the reference locus.

To assign an illuminant to the test image, we compute the error between the the log-difference chromaticity of the test image and each illuminant cluster along the locus. Here, we use Euclidean distance between the mean of the cluster for each sample illuminant and the mean of the log-difference chromaticity of the test image as an error metric:

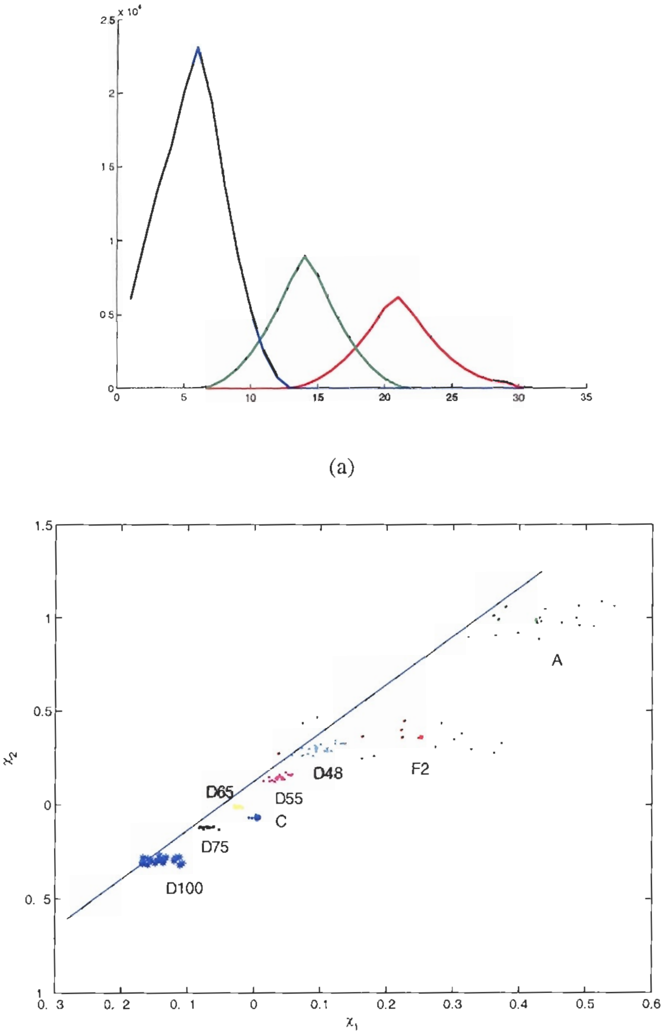
$$E_{ej} = \left(\sum_{i=1}^{2} (\chi_i^{mean(ej)} - \chi_i^{mean(et)})^2 \right)^{1/2} \tag{6.12}$$

where ej denotes the mean of the cluster for the jth illuminant, and et for the test image. Thus, the color temperature of jth illuminant is chosen if it provides the minimum distance to the test image.

6.4.3 Experiments and results

Spectral sharpening

Our illuminant estimation algorithm is based on the assumption that camera sensors are quite narrowband. If this is not the case we wish to carry out a transform on the sensor responses which sharpens the sensors. The spectral sharpening algorithm leads to a transform matrix M by which sensors can be optimally combined so as to form new colors that better approximate color changes



(a)

(b)

Figure 6.8: SONY DXC930 camera (a): Sensors, (b): log-difference geometric-mean chromaticity for Macbeth chart under eight illuminants.

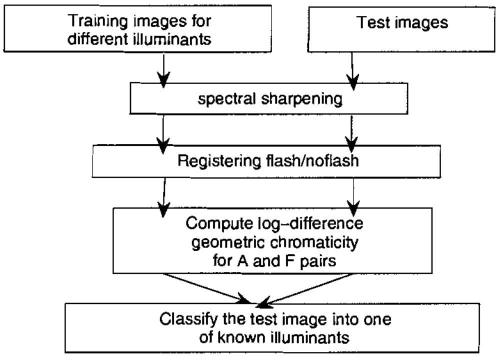

Training images for different illuminants
Test images
spectral sharpening
Registering flash/noflash
Compute log-difference geometric chromaticity for A and F pairs
Classify the test image into one of known illuminants

Figure 6.9: Algorithm flow for estimating ambient illuminants

induced by illuminant change, i.e. after matrixing the RGB color values, the log-difference image pixels for each illuminant temperature are more separate from the image pixels under other illuminants.

For cameras like the Sony DXC930 whose sensors are quite narrowband, the clusters for different illuminants are considerably separated so that classifying amongst these illuminants can be accurately achieved. However, when we carry out the log-difference chromaticity procedure for images which are taken using Kodak DCS420 which has broader sensors, the image pixel chromaticity for different illuminants are quite mixed; this would certainly lead to a failure in illuminant estimation. We used the method of spectral sharpening with positivity constraint (Section 3.3) to find the sharpening matrix for the Koday DCS420 such that the RGB values were transformed as if they were taken with a sharpened camera. Note that in our algorithm, the knowledge of camera sensors is not needed.

Again we used Macbeth surfaces under 8 illuminants to generate flash/no-flash pairs using Kodak DCS420 sensor curves. The log-difference chromaticity is shown in Figure 6.10(a): the points for each illuminant are not separated. This is not surprising because broadband sensors make the

RGB values more correlated. Figure 6.10(b) plots the chromaticity after sharpening the RGB values. We see that the effect on separating illuminants is dramatic: the clusters for the 8 illuminants are much better separated.

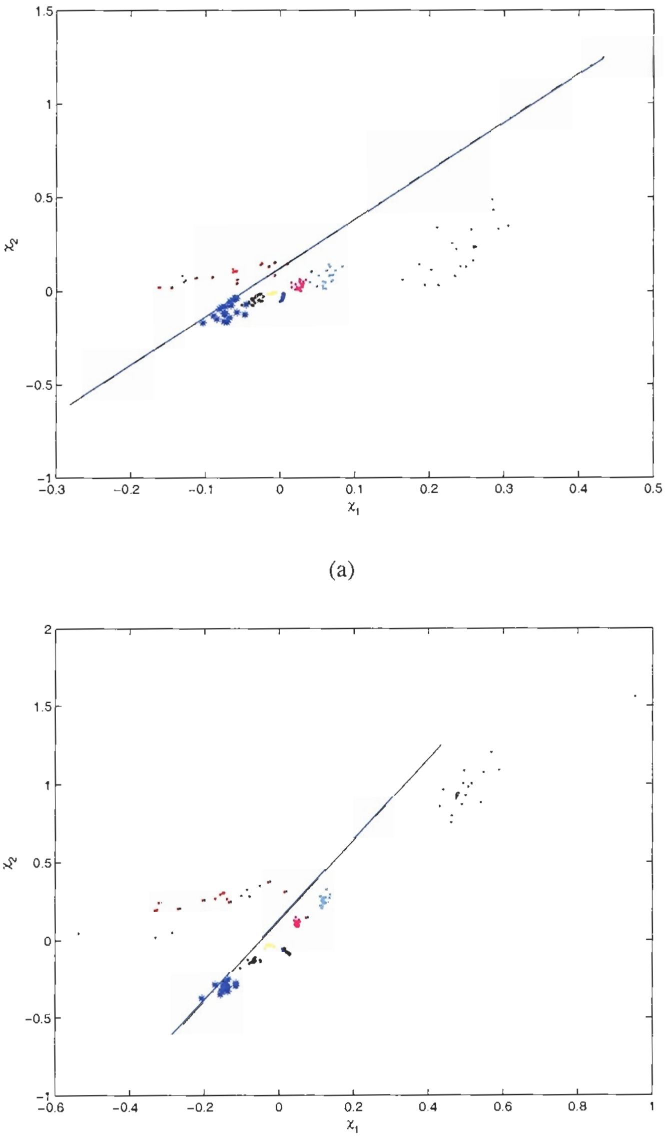
Estimating ambient illuminants

In order to evaluate our algorithm, we first use synthetic images. The Sony DXC930 sensors were used to generate image pairs. In the training phase, we image 461 Munsell color patches with 102 measured light sources [5]. The means of the log-difference chromaticity of these images for each illuminant was computed, plotted in Figure 6.11(a). In the test phase, we collect images of Macbeth ColorChecker 24 surfaces under the 102 light sources, as test images. Figure 6.11(b) shows the mean point of the log-difference chromaticity of the test images for each illuminant. Figure 6.11(c) shows the estimate result: the 45^0 straight line represents a perfect estimation of the 102 illuminants. The blue dots correspond to the estimate results. Here, the distance from one light to another is calculated by Euclidean distance of χ_1 and χ_2, and then the best match to each light with the smallest distance, order from 1 to 102 on the abscissa, is associated with the best matching light index, from 1 to 102, on the ordinate axis. We see most of the dots fall on or close to the straight line, except one point which associates the 93th illuminant with the 6th illuminant. Overall, results are excellent.

White balance

Estimating ambient illuminant can guide color balance for digital imaging. To further demonstrate the performance of the algorithm, we conducted experiments for carrying out the white balance for real images based on the estimated ambient illuminant temperature. The white balance algorithm looks for a white patch in the image, the chromaticity of which will be then be the chromaticity of the illuminant. For automatic white balance, the white patch is usually evaluated as the maximum or average found in each of the three image bands separately. The scaling coefficients are then obtained by comparing the chosen white patch with the values of the three channels of reference white. The difficulty is that maximum values (or average) of the three color bands are not necessarily white in the scene.

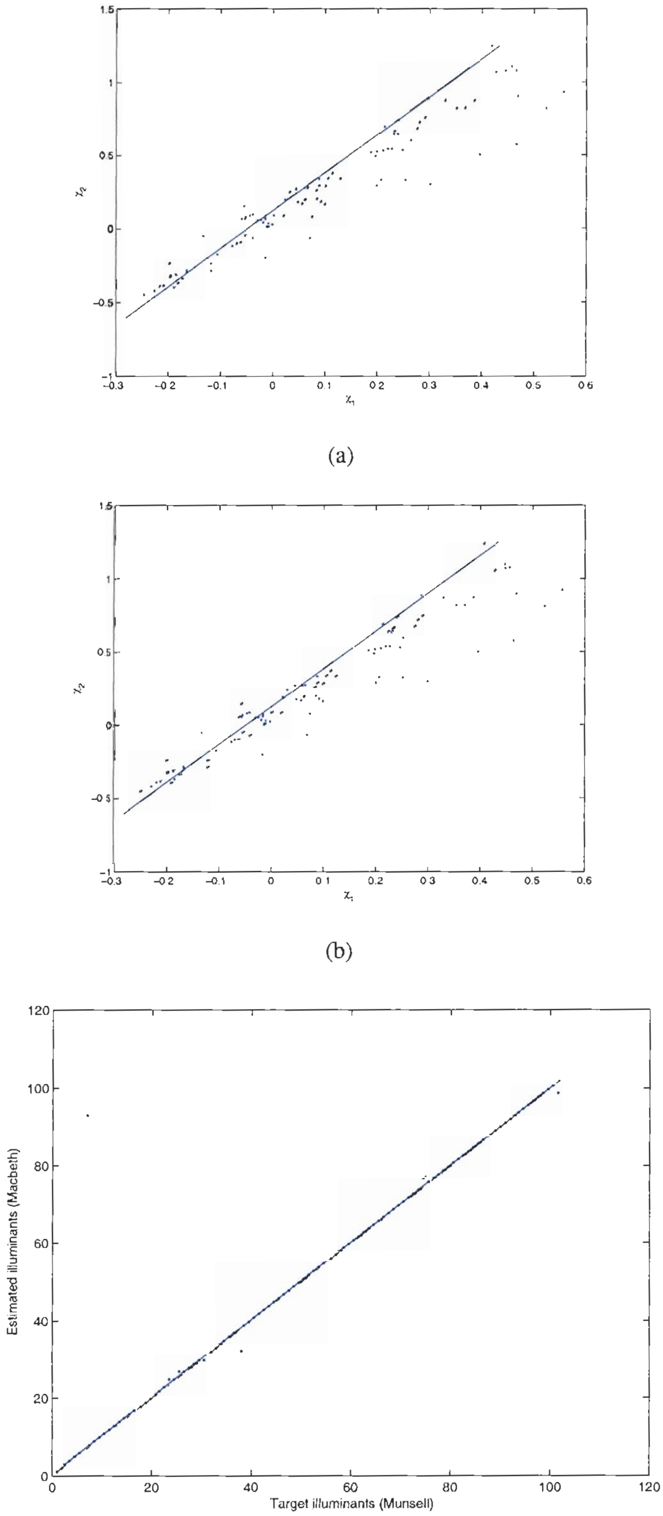
This problem can be solved using our illuminant estimate approach. The key is that once we find an ambient illuminant temperature for an image, i.e. we actually classify this illuminant into one of the known illuminant clusters along the reference illuminant locus, we can explicitly know which point in the cluster corresponds to the white patch of the Macbeth ColorChecker (supposing that we



(a)

(b)

Figure 6.10: Spectral sharpening (a): No sharpening, (b): With sharpening.



(a)

(b)

(c)

Figure 6.11: Illuminant estimation. (a): Mean points of log-difference chromaticity of Munsell patches for 102 illuminants, (b): Mean points of log-difference chromaticity of Macbeth patches for 102 illuminants, (c): Estimate result.

obtained the illuminant clusters using Macbeth patches). Thus, the white patch in each illuminant cluster can be the reference white color for this illuminant. In the training phase, we simply store the RGB values of the reference white patch for each known illuminant. In the testing phase,'once we assign an illuminant to the test image, the corresponding reference white patch will be used to carry out white balance for the test image, as follows:

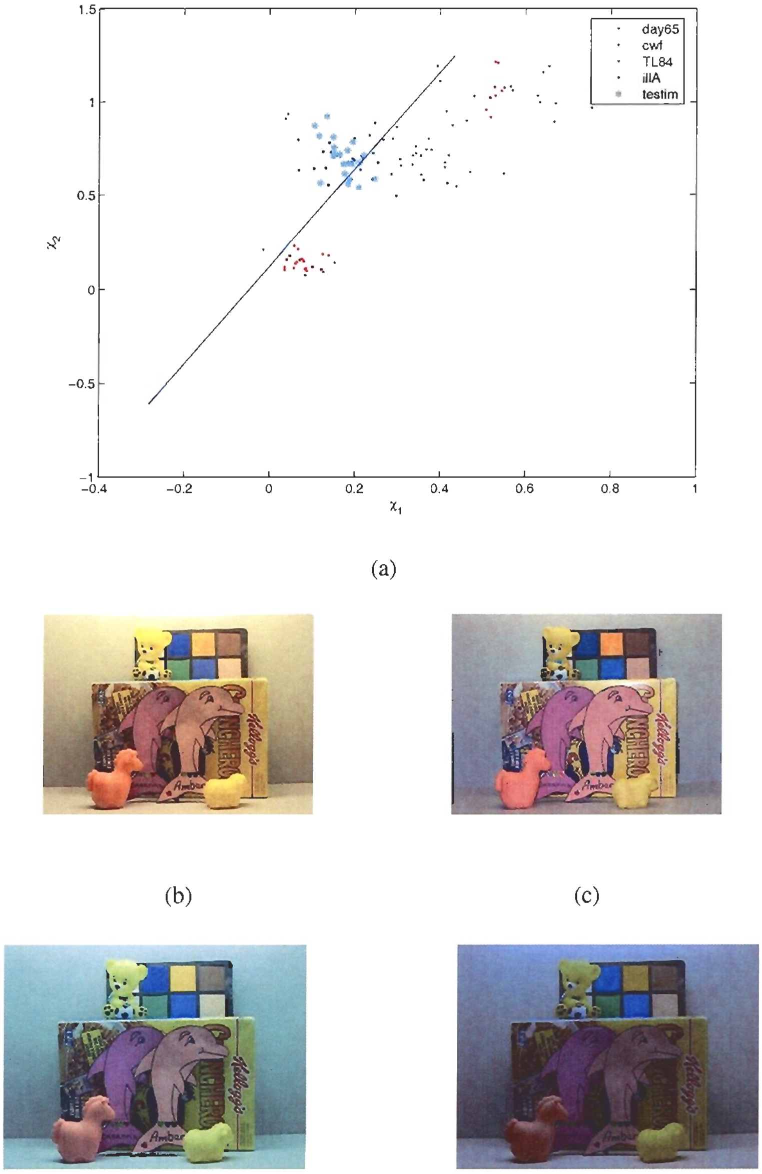
$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1/R_w & 0 & 0 \\ 0 & 1/G_w & 0 \\ 0 & 0 & 1/B_w \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (6.13)$$

R_w, G_w and B_w denote the RGB values of the reference white patch, which should be normalized by the maximum value among the three values.

The advantage of this white balance scheme is that it can get around the difficulty of evaluating the white patch using color information within the image. Also it saves computation for computing the maximum or average color values.

We captured real images using a consumer HP618 camera as the imaging device. We collected image pairs with the 24 patches of the Macbeth ColorChecker target under four lighting conditions: illuminant A, cool white fluorescent (CWF), the daylight D65, and the Tri-phosphor lamp TL84. The sharpened log-difference geometric-mean chromaticities of the images are plotted in Figure 6.12(a), where four clusters corresponding to the four illuminants are shown in different colors, and each cluster has 24 dots. We then captured a test image pair for a scene with multiple objects under illuminant CWF, in Figure 6.12(b,c). For saving computation time and for display, we samples the test image at 24 locations, evenly distributed on the images. We plot the chromaticities of these 24 sample pixels in Figure 6.12(a), marked with a cyan star. It is obvious that these sample points mostly overlap with the CWF cluster and so the white patch of the CWF is used for white balancing the test image.

This camera has four preset white balance settings: Auto, Daylight, Fluorescent, and Tungsten. In our training and testing phases, we manually choose Daylight white balance for both flash and no-flash images, so as to effectively eliminate the white balance for our illuminant estimate because the scaling factors for the white balance can be removed in the log difference process. For comparing our white balance result, we take another image for this scene under illuminant CWF using the 'Auto' white balance function (Figure 6.12(d)). The image contains subjects which are warm predominantly in color, and so the camera mistakes this for a color cast induced by a warm light source and creates a greenish color cast on the image. In contrast, our white balance result, shown in Figure 6.12(e), removes the greenish effect and is more close to the Fluorescent white balance.



(a)

(b)

(c)

(d)

(e)

Figure 6.12: White balance. (a): Ambient illuminant estimate for the test image; (b): no-flash test image; (c): flashed test image; (d): Auto white balance; (e): Our white balance result based on the correct estimate of ambient illuminant.

6.5 Automatic Compensation for Camera Settings for Images Taken under Different Illuminants

The combination of two images shot for the same scene but under different illumination has been used in wide applications ranging from estimating scene illumination, enhancing photographs shot in dark environments, to shadow removal. However, for consumer-grade digital cameras, due to the different illumination conditions, the two images usually have different camera settings when they are taken, such as exposure time and white balance [79]. Thus registering the two images becomes a necessary step prior to combining the two images. Unfortunately, how to register these two images has not been investigated fully. In this section, we propose a method which can parametrically adjust the two images so as to compensate for the difference in exposure speed, ISO, aperture size, and white balance. This is accomplished by training a 2nd-order masking model on a set of image pairs to predict the model parameters. This trained model can then be used to register two images. By adjusting RGB pixel values of one image so as to remove the shadow in the difference image, the compensating magnitude can be computed and used to train a masking model. This masking model can accurately compensate for camera setting such that the difference between compensated images reflects only the difference in illumination.

6.5.1 Motivation

Work on image pairs all analyze the difference image: (flash image – no-flash image) to infer the contribution of the flash light to the scene. To make this computation meaningful, the images must be in the same camera settings. These include: exposure time, ISO, aperture and white balance. Since different lighting conditions usually cause changes of camera settings, an effective registering method to compensate for the difference, for the image pair, is necessary. Unfortunately, this problem has never been investigated carefully. This is partly because the difference between the two images is a composite of the difference of camera settings and of the light arriving at the camera, and they are difficult to separate.

We present a method which uses a masking model to compute the compensation between the two images given the camera settings for the two images. This model assumes additivity and proportionality of the different factors involved. Here we use a 2nd-order masking model, with 9 parameters. To train the model for a digital camera, we collect a set of image pairs. For each image pair, two images are taken for the same scene, under different lighting conditions.

In the training phase, we restrict our attention to Lambertian surfaces. For such a reflectance map, at the surface point corresponding to each pixel, all lighting is added up into a single effective light [104]. A problem arising in the training phase is that we have to know how to adjust the magnitude in each color channel of one image to register with the other image, for each image pair. The problem is difficult because the difference between the two images is a composite of both camera settings and illumination. Here, we use the simple fact that *a shadow effect should be caused purely by the changes of illumination*. Suppose we have a pair of images A and B. Both of them are taken for the same scene but under a different lighting situation: only one light source, illuminant 1, illuminates the scene for A; there are two light sources, illuminant 1 and illuminant 2, illuminating the scene for B. An example of this situation would be taking the first image under sunlight, and the second image with flash added to the sunlight.

This additivity property leads to the fact that for the two images, A and B, subtracting the first from the second leads to any shadow caused by illuminant 1 disappearing in the difference image, i.e. as if it were taken under illuminant 2 only. For the above situation, the difference image will be a pure flash image, and the shadow caused by sunlight will disappear. This will be true for image pairs under any ambient lighting plus another image that includes light from a flash.

The use of a shadow helps us to be able to separate the contributions from the camera settings from illumination difference for the two images. That is, the shadow should be caused purely by the changes in illumination. We thus can use this fact to compute by what magnitude we have to adjust A so that no illuminant 1 causes shadows to appear in $B - A$. This magnitude will then be used to train the parameters for the masking model. In the training phase, we first collect image pairs for which we set up an imaging environment which has two light sources and there are shadow regions caused by one of the illuminants in each pair of images. For each image pair, we first adjust the magnitude for each color channel of image A, so that the difference image, which is obtained by subtracting the adjusted image A' from B, has the shadow removed. Then using the adjusted images, the parameters of the masking model can be computed given the camera settings of the two images. Once we obtain the parameters of the model, we can use this masking model to adjust new image pairs as if they are taken under the same camera settings, such that the difference between the two images will be totally controlled only by the light arriving at camera.

6.5.2 Camera Settings and Image Acquisition

We have designed an algorithm to work with images acquired using only consumer-grade digital cameras, with specialized equipment not required. All of the images were acquired in a RAW format and then conversion software was used to convert them into 12-bit linear TIFF images.

Consumer-grade cameras typically have the following settings which can be adjusted automatically or by user control:

1. Focal length.
2. Exposure time (shutter speed).
3. Aperture (f-number).
4. ISO (film speed).
5. White balance.

We fix the focal length so that the cameras focus remains constant. We also use a tripod when taking images to ensure that the image pair capture exactly the same points in the scene. For other settings, we turn on the 'auto' function and let the camera decide how to set the exposure time, aperture, ISO and white balance for different lighting situations.

The size of the aperture and the brightness of the scene control the amount of light that enters the camera during a period of time, and the shutter controls the length of time that the light hits the recording surface. In photography, exposure value (EV) is a value given to all combinations of camera shutter speed and aperture that give the same exposure. By the Additive Photographic Exposure System (APEX) [2], the exposure value is the sum of the Aperture Value (AV) and the Time Value (TV):

$$EV = AV + TV \tag{6.14}$$

If N is the f-number, the Aperture Value (AV) is

$$AV = log_2 N^2 = 2 log_2 N \tag{6.15}$$

If t is the shutter time in seconds, the Time Value (TV) is

$$TV = log_2 \frac{1}{t} = -log_2 t \tag{6.16}$$

Film speed (ISO) is the measure of a photographic film's sensitivity to light. Our test camera has two ISO settings: 100 and 200. All of the values, shutter speed, aperture, and ISO can be read out from digital photo conversion software the also outputs metadata.

Our test camera has four preset white balance settings: Auto, Daylight, Fluorescent, and Tungsten. The white balance algorithm looks for a white patch in the image, the chromaticity of which will be then taken to be the chromaticity of the illuminant. For automatic white balance, the white patch is usually evaluated as the maximum or average found in each of the three image bands separately. Scaling coefficients are then obtained by comparing the chosen white patch with the values of the three channels of a reference white. For the captured images, the scaling coefficients are not known. Here we use the mean value for each RGB channel to reflect the effect of white balancing.

6.5.3 A Masking Model for Compensating for Camera Settings

Our goal now is to find a model that can accurately describe the relation between the difference of the two images and the camera settings. As described above, the problem of compensating for camera settings for two images taken under different illuminations reduces to finding a 3-vector of scaling coefficients such that one of the two images A is transformed to A' such that the shadows caused by illuminant 1 will be removed in the difference image $B - A'$. Suppose we have a shadow region s in image A and B which is caused by illuminant 1, and an out-of-shadow region ns. So the light reaches the regions s and ns as follows:

- In A, neither illuminant 1 nor illuminant 2 reach region s,
- In A, illuminant 1 reaches region ns,
- In B, illuminant 2 reaches region s,
- In B, both illuminant 1 and 2 reach region ns.

Now we want to transform A to A' with a 3-coefficient vector M, a coefficient for each color channel, to compensate for the different camera settings so that the difference in the shadow region s between A' and B should be equal to the difference in the out-of-shadow region ns, i.e. the shadow disappears in $B - A'$.:

$$\begin{aligned}
B(ns) \ - \ A'(ns) \ &= \ B(s) \ - \ A'(s) \\
B(ns) \ - \ B(s) \ &= \ A'(ns) \ - \ A'(s) \\
B(ns) \ - \ B(s) \ &= \ M \ (\ A(ns) \ - \ A(s) \ ) \\
M &= \tfrac{B(ns) \ - \ B(s)}{A(ns) \ - \ A(s)}
\end{aligned} \tag{6.17}$$

As M is used to compensate the camera settings of the two images, according to the above equation, M should be a function of the ratios of exposure value, ISO, and white balance between the two images.

Here we choose a 2nd-order masking model to describe the difference brought about by camera settings [47]. This model, originally proposed for characterizing color printers, uses logarithms and assumes additivity and proportionality of variables. This is used to establish the amount of variables required to match the difference between the two images. The form of the 2nd-order model for our application is given as:

$$\begin{aligned} & log\left(\frac{B(ns)_i - B(s)_i}{A(ns)_i - A(s)_i}\right) \\ = & \ a_1\, log\left(\frac{EV_B}{EV_A}\right) + a_2\, log\left(\frac{ISO_B}{ISO_A}\right) + a_3\, log\left(\frac{mean_{Bi}}{mean_{Ai}}\right) \\ & + b_1\, log\left(\frac{EV_B}{EV_A}\right)^2 + b_2\, log\left(\frac{ISO_B}{ISO_A}\right)^2 + b_3\, log\left(\frac{mean_{Bi}}{mean_{Ai}}\right)^2 \\ & + c_1\, log\left(\frac{EV_B}{EV_A}\right)\, log\left(\frac{ISO_B}{ISO_A}\right) \\ & + c_2\, log\left(\frac{ISO_B}{ISO_A}\right)\, log\left(\frac{mean_{Bi}}{mean_{Ai}}\right) \\ & + c_3\, log\left(\frac{mean_{Bi}}{mean_{Ai}}\right)\, log\left(\frac{EV_B}{EV_A}\right) \end{aligned} \tag{6.18}$$

where the subscript A and B represent image A and B. The estimation of parameters a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2 and c_3 is accomplished by a least-squares procedure.

Once we obtain the 9 parameters, this masking model can be used to register two new images, given their camera settings.

6.5.4 Experiments and Results

We first set up an imaging environment. In this environment five lighting sources are used: direct sunlight, cloudy daylight, a tungsten light lamp, an incandescent light lamp and a xenon flash light. We used five objects with different colors to create shadows on five different tablecloths. The environment is shown as Figure 6.13.

We capture images under the following five situations:

- Using direct sunlight as illuminant 1 to create shadows and adding flash as illuminant 2.

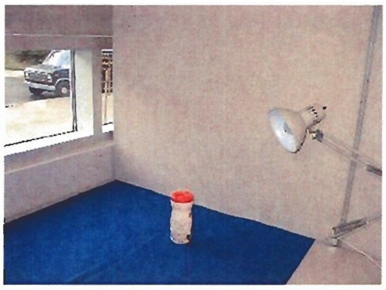


Figure 6.13: Experimental imaging environment.

- Using tungsten light as illuminant 1 to create shadows and adding flash as illuminant 2.
- Using tungsten light as illuminant 1 to create shadows and adding cloudy daylight as illuminant 2.
- Using incandescent light as illuminant 1 to create shadows and adding flash as illuminant 2.
- Using incandescent light as illuminant 1 to create shadows and adding cloudy daylight as illuminant 2.

In each situation, we captured 25 image pairs by different combinations of objects and tablecloths. Overall, we collected 125 image pairs. All of the images were acquired in a RAW format by a consumer-grade digital camera and conversion software was used to convert them into 12-bit linear TIFF images. We fixed the focal length and recovered from the image metadata the exposure time, aperture, and ISO values, using the conversion software, for each image.

To train and test our model, we picked one image pair as the test image and trained the remaining 124 pairs to compute the model. The training phase gives the model parameters, and then the transformation 3-vector M is computed for the test pair given the camera settings of the test pair. To evaluate compensation results, the difference between shadow regions and out-of-shadow regions is used again. Here, we use CIE ΔE_{94} color difference values to calculate the difference. This process was iterated, until we had compensation errors for all 125 image pairs. Figures 6.14,6.15 show the results of this compensation. The first image has shadows caused by direct sunlight, and we add a flash in the second image. After compensation, the difference image has no sunlight (ambient) shadows at all.

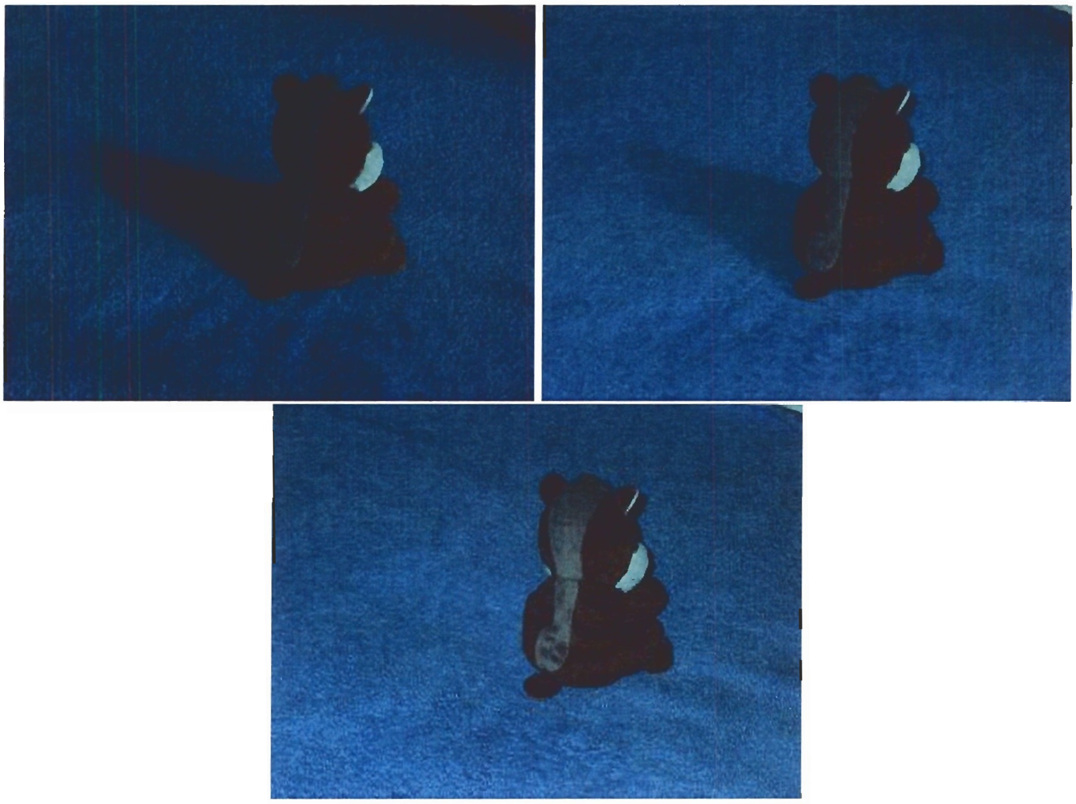


Figure 6.14: Image A, B and $B - A'$.

6.6 Summary

We have presented a simple approach to removing shadows from ambient images for ambient / ambient plus flash pairs. The method is pixel-based, in that no shadow edge need be found: the method simply relies on determining an illumination field image from the log-difference of the pure flash and ambient images. In this image, surface reflectance is approximately removed, leaving only illumination and shading information. In an ambient shadow region, the difference between the two is large, from an intensity difference or a lighting-color difference or both. Creating a shadow mask area is hence straightforward. Copying flash-image information over to the ambient image, flash and ambient edges are blended such that the resulting integrable edge field generates a shadowless, ambient-lighting result. This shadow removal algorithm using ambient/ambient plus flash pairs outperforms the previous invariant-based scheme in three respects: First, no invariant image is needed, and thus the entropy minimization or calibration step can be avoided; Second, the use of ambient/flash pairs enables us to find shadow *regions* instead of only shadow edges which are

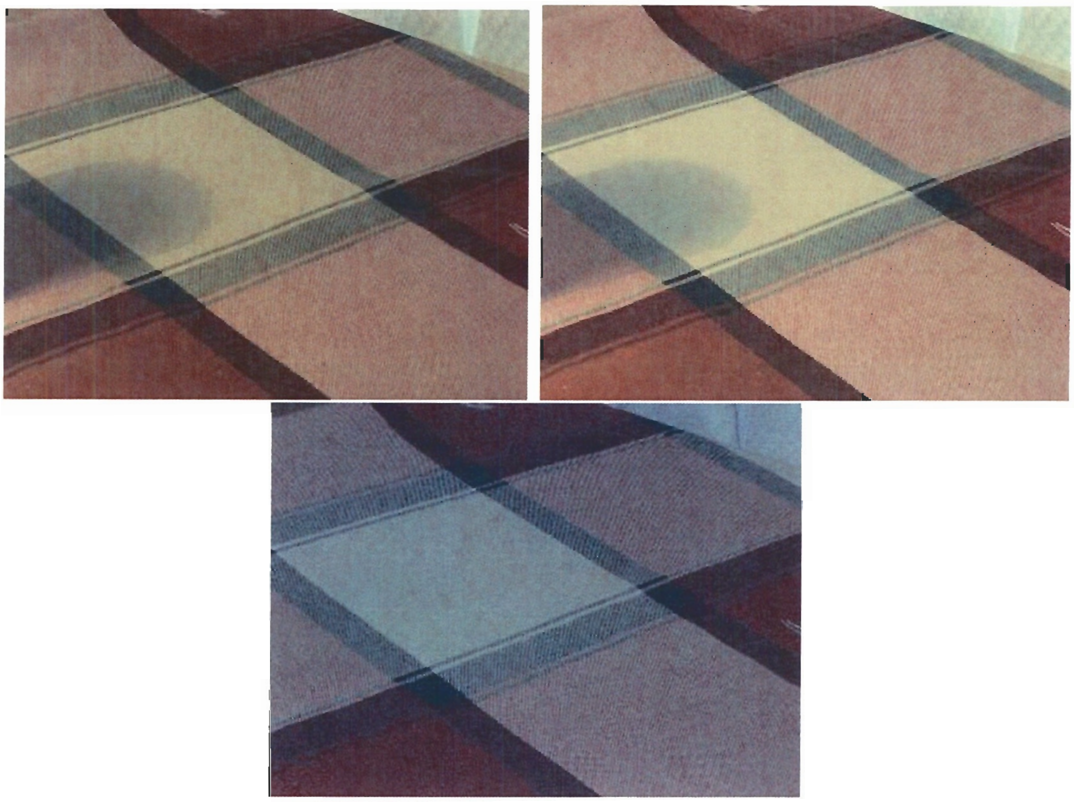


Figure 6.15: Image A, B and $B - A'$.

hard to find; Third, clearly removing shadows using ambient/ambient plus flash pairs is much faster.

We also presented a practical approach for estimating ambient illuminant in a scene. The method has a number of novel features. First, it is based on a simple image formation model and obviates using complicated physical constraints on surfaces and knowledge of camera sensors and flash spectra. In our method, surfaces in the scene need not be known, and information about the camera and flash is not required. Second, in our method, a novel reference illuminant temperature locus is proposed, which specifies the path for changes of illuminant temperatures; it can be used to estimate the scene illuminant. Third, based on our accurate estimate of ambient illuminant, an easy but more accurate white balance scheme can be carried out for automatically balancing images.

To make the combination of flash and no-flash images meaningful we address the problem of compensation for camera settings for image pairs which are taken for the same scene but under different lighting conditions. The difficulty of this problem is that the difference between the two images is a composite of camera settings and scene illumination. We solve this problem by using the fact that shadow effects in images should be entirely caused by illumination. By removing shadows in the difference image, we achieve the separation between camera settings and illumination. We propose using a masking model to describe camera settings. Via this model, the effect of camera settings can be easily eliminated, make algorithms that use difference-images much more dependable.

Chapter 7

Conclusion and Future Work

In this thesis, a set of methods for removing shadows from color images was proposed. Shadows in a color image are caused by changes of illumination in both intensity and color. This property is exploited in a previous work [40] which showed that given certain assumptions about scene illumination, surface reflectance and camera sensors it is possible to solve the color constancy problem at a single image pixel. That is, given a single triplet of sensor responses it is possible to derive a 1-d quantity invariant to both the color and intensity of the scene illuminant. This in effect provides a 1-d grayscale reflectance image which is therefore shadow-free. Deriving such illuminant invariant images relies on finding an invariant direction in a log-chromaticity space. Originally, the approach to finding this special angle was a camera calibration: a color target is imaged under many different lights, and the direction that best makes color patch images equal across illuminants is the invariant direction. Here, we proposed a different approach. In this thesis, instead of a camera calibration we aim at finding the invariant direction from evidence in the color image itself. Specifically, we recognize that producing a 1-d projection in the correct invariant direction will result in a 1-d distribution of pixel values that have smaller entropy than projecting in the wrong direction. The reason is that the correct projection results in a probability distribution spike, for pixels all the same except differing by the lighting that produced their observed RGB values and therefore lying along a line with orientation equal to the invariant direction. Hence we seek that projection which minimizes entropy, and from there go on to remove shadows from images. To be able to develop an effective description of the entropy-minimization task, we proposed to use the quadratic entropy, rather than Shannon's definition. Replacing the observed pixels with a kernel density probability distribution, the quadratic entropy can be written as a very simple formulation, and can be evaluated using the efficient Fast Gauss Transform.

Once I obtained the grayscale shadow-free invariant image, this 1-d representation is extended to an equivalent 2-d, chromaticity representation. Using this 2-d representation, it is possible to re-light all the image pixels in the same way, effectively deriving a 2-d image representation which is additionally shadow-free. Using the shadow-free chromaticity image, we proposed to recover a full-color 3-d image representation which is the same as the original image but with shadows removed. To do so, we worked with an edge-map of the image but we are concerned with separating shadow edges from reflectance edges and factoring out the former. We reasoned that a shadow edge corresponds to any edge which is in the original image but absent from the invariant representation, and we thus proposed a thresholding operation to identify the shadow edge. Since our shadow removal algorithm depends on finding the shadow edges, it can only deal with the hard shadows which have sharp shadow edges. Edges are in-painted across the shadow edge, and re-integrating yields a color image, equal to the original save for the fact that it is shadow-free.

Although the above algorithm can remove shadows from color images, detail information in dark shadow regions is difficult to recover. Also, the question of how to detect the shadow still remains. In this thesis, we proposed to use flash/no-flash image pairs to detect shadows in the image taken under ambient light (no flash), given extra information from a flash image registered with the first. Clearly, the pure-flash image is the difference between the two — and has no ambient-shadow component. This property is exploited to detect shadows in the ambient image. We argue that first going to a "spectrally sharpened" color space, and then focusing on the difference in a log domain of the flash image and the ambient image, gives a very simple feature space consisting of two components — one in an illuminant-change 3-vector direction, and one along the gray axis. This space provides excellent separation of the shadow and nonshadow areas. Using both gradient fields of flash/ambient image pairs, we do the image reconstruction by inverting the Poisson equation. In this way, we arrived at an image with the advantages of the ambient-only image — warmth, no flash effects such as disturbing illumination dropoff with distance, pixel saturation etc. — but no shadows.

A crucial piece for the applications using flash/no-flash image pairs is deriving the pure-flash image from the combination of the two images. For consumer-grade digital cameras, due to the different illumination conditions the two images usually have different camera settings such as exposure time and white balance. We proposed a camera setting compensation method to remove the difference caused by different camera settings such that the difference between compensated images reflects only the difference in illumination. This is accomplished by training a 2nd-order masking model sets of image pairs to predict the model parameters. This trained model can then be used to register two images.

Shadows are accounted for as a change of scene illuminants. We proposed a practical approach for estimating ambient illuminant in a scene using flash/no-flash image pairs without the need for any information on surface reflectance and camera sensors. The no-flash ambient and pure-flash images were first represented by a simple image formation model using our assumptions of Lambertian surfaces, Planckian lights, and narrowband camera sensors. We then projected the difference in a log domain of the pure-flash image and the no-flash image into a geometric-mean chromaticity space, gave the chromaticity of the ambient illuminant. We showed that the chromaticities corresponding to illuminants with different temperatures fall along a line. Simply by taking the nearest color temperature along this illuminant line, or classifying into one of potential illuminants, our algorithm arrives at an estimate of illuminant.

7.1 Contributions of the Thesis

The most important contribution of the this thesis is that it is the best shadow-removal system for color images to date. In the main, other successful methods of removing shadows rely on background subtraction in video, or on simple rules that are not physically-based. The following subsections detail the contribution of this thesis.

7.1.1 Intrinsic Images by Entropy Minimization

We have developed an intrinsic image algorithm that can find the invariant direction for a single unsourced image using entropy minimization. This is joint work of Finlayson, Drew and myself. In this joint work, we proposed using entropy minimization to determine the invariant direction from images that arise from cameras, without calibration. We begin by adopting the Shannon definition to evaluate the entropy for each 1-d projection. We verify that the entropy is a strong indicator for finding the invariant. To facilitate efficient and stable search for the minimum, we have proposed to use a quadratic entropy definition instead of the Shannon entropy. Moreover, our algorithm applies the Fast Gauss Transform to calculate the quadratic entropy in linear running time. We recover shadow-free RGB color images by solving the Poisson equation on a modified gradient space, which follows a previous work [41] but has three enhancements. First we guarantee that the edge map we are integrating satisfies integrability conditions. This is accomplished by projecting the modified edges onto an integrable edge map in Fourier space. Second I proposed a new shadow edge map estimation method. The advantage is that the method takes both the direction and intensity of edges into account when comparing edge maps of the original image to those derived from an invariant image. Third, when performing the integration, rather than simply setting shadow edges to zero, we apply an iterative diffusion process which fills in the derivatives across shadow edges, bridging values obtained from neighboring nonshadow edge pixels. Thus the edge information pertaining to local texture in the neighborhood of the shadow edges can be retained to some degree.

7.1.2 Spectral Sharpening with Nonnegative Constraint

The 1-d illumination invariant formation will hold so long as a camera is equipped with fairly narrow sensitivities. To effectively create narrowed camera sensors, a sharpening transform for camera sensors is commonly used. Since we mean to apply logs, it is important to use a color space transform that produces positive values (or at least nonnegative ones, with zeros treated specially). To do so, I proposed a "spectral sharpening with nonnegative constraint" transform, which is similar to a previous "spectral sharpening with positivity" method in which Finlayson and Drew added a term for nonnegative transform in the objective function and to the method by Barnard et al. In this thesis we replaced guessing objective function weights in the latter method by actual nonlinear nonnegative constraints. This is individual, not collaborative, work.

7.1.3 Ambient Shadow Detection via Flash/No-flash Image Pairs

I have presented a simple approach to removing shadows from ambient images via ambient/flash pairs. The method is pixel-based, in that no shadow edge need be found, and this is the advantage of this approach: the method simply relies on determining an illumination field image from the log-difference of the pure flash and ambient images, when such image pairs are indeed available. In the log-difference image, surface reflectance is approximately removed, leaving only illumination and shading information. In an ambient shadow region, the difference between the two is large, from an intensity difference or a lighting-color difference or both. Creating a shadow mask area is hence straightforward. Copying flash-image information over to the ambient image, flash and ambient information is blended such that the resulting integrable edge field generates a shadowless, ambient-lighting, result. This work was discussed with Mark Drew; its development was brought into being by myself.

7.1.4 Compensation for Camera Settings for Images Taken under Different Illuminants

I proposed a method for compensation for camera settings for image pairs which are taken for the same scene but under different lighting conditions. The difficulty of this problem is that the difference between the two images is a composite of camera settings and scene illumination. We solve this problem by using the fact that shadow effects in images should be entirely caused by illumination. By removing shadow in the difference image, we achieve the separation between camera settings and illumination. We propose using a masking model to describe camera settings. Via this model, the effect of camera settings can be easily eliminated, make algorithms that use difference-images much more dependable. The idea for this work was developed together with Mark Drew, and its form and actual embodiment were entirely created by myself.

7.1.5 Ambient Illuminant Estimation

I have presented a practical approach for estimating ambient illuminant in a scene. With the the aid of a flash image, the ambient illuminant can be classified into one of a collection of different, predetermined reference lights. The method has a number of novel features. First, it is based on a simple image formation model and obviates using complicated physical constraints on surfaces or knowledge of camera sensors or flash spectra. In our method, surfaces in the scene need not be known, and information about the camera and flash is not required. Second, in our method, a novel reference illuminant temperature locus is proposed, which specifies the path for changes of illuminant temperatures; it can be used to estimate the scene illuminant. Third, based on our accurate estimate of ambient illuminant, an easy but more accurate white balance scheme can be carried out for automatically balancing images. All this work was entirely created by myself.

7.2 Future Work

The most important contribution of the this thesis is that it is the most successful work for removing shadows from color images to date. Nevertheless, although encouraging results have been obtained in this research effort it still leaves open the possibility for future developments. The following is a list of some extensions and improvements that deserve further exploration.

- The current shadow removal algorithm using flash/no-flash consists of a gradient fields merging process in which the gradients in the detected ambient shadow regions are copied from the pure-flash image into the ambient image. This merging step works well if no flash shadow falls in ambient shadows, but would fail to remove ambient shadows in case some flash and ambient shadows coincide. A promising approach would involve blending rather than copying gradients in a variational scheme to combine the gradients of F and A, especially in areas where flash- and ambient-shadow masks coincide, and application to specular layers, such as window reflections, and daytime/nighttime surveillance camera pairs. Also this variational scheme would be appropriate for filling edges into shadow boundaries in the algorithm for removing shadows from a single image.

- The current shadow detection and removal methods work on raw image data or linear images. But cameras ordinarily supply images that are not only compressed, but also greatly processed away from being linear images. We will consider the issues involved in linearizing images, and in removing artifacts from compression (especially for JPEG images) so as to return to an original, physically-based image as much as possible so that the color science applied here is closest to the reality of the images.

- The current shadow-free image recovery algorithm experiences difficulties when part of shadow edges can not be identified to some degree. In Chapter 5 we discussed several cases in which our scheme for locating shadow edges fails to detect shadow edges or detects wrong edges. An alternative way is to find shadow regions rather than shadow edges: this could provide complete shadow boundaries. We consider as future work the exploitation of the illuminant invariant relation for two neighboring regions (supposing that we can perform segmentation first) which would result in shadow candidates; then shadows could be identified by merging or dividing those candidates using spatial constraints.

- We devised a scheme for estimating ambient illuminant using flash/no-flash pairs. So far this method only detects a single illuminant in the scene. In fact, our method is able to estimate multiple illuminants in the scene. As mentioned, shadows are caused by the changes of illuminants; thus it should be possible to use the knowledge of estimated scene illuminants to detect or remove shadows.

- A problem for our shadow removal algorithm can arise if adjacent surfaces are related by a color change in the direction in which illumination changes, or they are both grey surfaces but have different intensity. Here, an edge will be found in the original image, but will be absent from the invariant images. Thus, this edge will be removed by mistake in the resulting shadow-free image. In future work, we intend to investigate ways to overcome this problem. One thought is to take the information of objects and 3-d geometry of the scene into account such that the shadow removal algorithm can only consider the shadows belonging to the same surfaces.

Bibliography

[1] http://en.wikipedia.org/wiki/differential_entropy.

[2] http://en.wikipedia.org/wiki/exposure_value.

[3] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.*, 24:828–835, 2005.

[4] K. Barnard, F. Ciurea, , and B. Funt. Sensor sharpening for computational color constancy. *J. Opt. Soc. Am. A*, 18:2728–2743, 2001.

[5] K. Barnard, L. Martin, B. Funt, and A. Coath. A data set for colour research. *Color Research and Applications*, 27:147–151, 2002.

[6] H.G. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, 1978.

[7] B.J.C. Baxter and G. Roussos. A new error estimate of the fast Gauss transform. *SIAM J. on Sci. Comput.*, 24(1):257–259, 2002.

[8] R. Beatson and L. Greengard. A short course on fast multipole methods. In M. Ainsworth, J. Levesley, W. Light, and M. Marletta, editors, *Wavelets, Multilevel Methods and Elliptic PDEs*. Oxford Univ. Press, 1997.

[9] A. Blake. Boundary conditions for lightness computation in Mondrian world. *Comp. Vision, Graphics, and Image Proc.*, 32:314–327, 1985.

[10] D.H. Brainard and W.T. Freeman. Bayesian color constancy. *J. Opt. Soc. Am. A*, 14:1393–1411, 1997.

[11] D.H. Brainard and B.A. Wandell. Analysis of the retinex theory of color vision. In *J. Opt. Soc. Am*, volume A3, pages 1651–1661, 1986.

[12] G. Brelstaff and A. Blake. Detecting specular reflections using Lambertian constraints. In *Proc. Second Int. Conf. on Comp. Vision*, pages 297–302. IEEE, 1988.

[13] M.H. Brill and G. Finlayson. Illuminant invariance from a single reflected light. In *Color Research and Application*, volume 27, pages 45–48, 2002.

[14] G. Buchsbaum. Color signal coding: Color vision and color television. *Color Resarch and Application*, 12:266–269, 1987.

[15] John Canny. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 8:679–698, 1986.

[16] Y.-Y. Chuang, D. B Goldman, B. Curless, D. H. Salesin, and R. Szeliski. Shadow matting and compositing. *ACM Trans. Graph.*, 22:494–500, 2003.

[17] J. Cohen. Dependency of the spectral reflectance curves of the munsel color chips psychon. *Science*, pages 369–370, 1964.

[18] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24:603–619, 2002.

[19] C.Poynton. The rehabilitation of gamma. In *Human Vision and Electronic Imaging III*, volume 3299, pages 232–249, 1998.

[20] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. In *PAMI*, volume 25:10, pages 1337–1342, 2003.

[21] Spectrophotometric data for a colection of munsel samples. *Color and Light in Nature.* 1957.

[22] D.B.Judd, D.L.MacAdam, and G.Wyszecki. Spectral distribution of typical daylight as a function ofcorelated color temperature. In *In Advances in Neural Information Processing Systems*, volume 54, pages 1031–1040, 1964.

[23] D.H.Marimont and B.A.Wandel. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am. A*, pages 1905–1913, 1992.

[24] J. M. DiCarlo, F. Xiao, and B. A. Wandell. Illuminating illumination. In *Color Imaging Conf.*, pages 27–34, 2001.

[25] M.S. Drew and G.D. Finlayson. Spectral sharpening with positivity. *J. Opt. Soc. Am. A*, 17:1361–1370, 2000.

[26] M.S. Drew, G.D. Finlayson, and S.D. Hordley. Recovery of chromaticity image free from shadows via illumination invariance. In *IEEE Workshop on Color and Photometric Methods in Computer Vision, ICCV'03*, pages 32–39, 2003.

[27] M.S. Drew, C. Lu, and G.D. Finlayson. Removing shadows using flash/noflash image edges. In *IEEE Conf. on Multimedia and Expo: ICME*, 2006.

[28] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, 2004.

[29] A. Elgammal, R. Duraiswami, and L.S.Davis. Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. *IEEETRans. Patt. Anal. Mach. Intell.*, 25(11):1499–1504, 2003.

[30] G. Finlayson, S. Hordley, and P. Hubel. Colour by correlation: A simple, unifying approach to colour constancy. In *ICCV99*, pages 835–842, 1999.

[31] G. Finlayson and G. Schaefer. Single surface colour constancy. In *7th Color Imaging Conference: Color Science, Systems, and Applications*, pages 106–113, 1999.

[32] G.D. Finlayson. Color constancy in diagonal chromaticity space. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 218–223. IEEE Computer Society, June 1995.

[33] G.D. Finlayson. Color in perspective. *PAMI*, 18:1034–1038, 1996.

[34] G.D Finlayson. Retinex viewed as a gamut mapping theory of color constancy. In *8th Congress of the Internatinal Color Association*, 1997.

[35] G.D. Finlayson, S.S. Chatterjee, and B.V. Funt. Color angle invariants for object recognition. In *3rd Color Imaging Conference: Color, Science, Systems and Applications*, pages 44–47. Society for Imaging Science & Technology (IS&T)/Society for Information Display (SID) joint conference, 1995.

[36] G.D. Finlayson and M.S. Drew. 4-sensor camera calibration for image representation invariant to shading, shadows, lighting, and specularities. In *ICCV'01: International Conference on Computer Vision*, pages II: 473–480. IEEE, 2001.

[37] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *J. Opt. Soc. Am. A*, 11(5):1553–1563, May 1994.

[38] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *ECCV 2004: European Conference on Computer Vision*, pages 582–595, 2004. Lecture Notes in Computer Science Vol. 3023.

[39] G.D. Finlayson, M.S. Drew, and C. Lu. Shadow removal by entropy minimization. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 2006. Submitted for publication.

[40] G.D. Finlayson and S.D. Hordley. Colour constancy at a pixel. *J. Opt. Soc. Am. A*, 18(2):253–264, Feb. 2001. Also, UK Patent #2360660, "Colour signal processing which removes illuminant colour temperature dependency".

[41] G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images. In *ECCV 2002: European Conference on Computer Vision*, pages 4:823–836, 2002. Lecture Notes in Computer Science Vol. 2353.

[42] G.D. Finlayson, S.D. Hordley, C. Lu, and M.S. Drew. On the removal of shadows from images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28:59–68, 2006. http://www.cs.sfu.ca/ mark/ftp/Pami06/pami06.pdf.

[43] G.D. Finlayson, B. Schiele, and J.L. Crowley. Comprehensive colour image normalization. In *ECCV98*, 1998.

[44] G.D. Finlayson and S. Ssstrunk. Optimization for hue constant rgb sensors. In *10th Color Imaging Conference*, pages 348–343, 2002.

[45] D.A. Forsyth. A novel approach to color constancy. In *Int. Conf. on Computer Vision '88*, pages 9–18, 1988.

[46] David A. Forsyth and Jean Ponce. *Computer vision: a modern approach.* Prentice Hall, 2002.

[47] Clapper F.R. An empirical determination of half-tone colour reproduction requirements. In *TAGA proc.*, page 31, 1961.

[48] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 10:439–451, 1988.

[49] G. Funka-Lea and R. Bajcsy. Combining color and geometry for the active, visual recognition of shadows. In *IEEE Int. Conf. on Computer Vision*, pages 203–209, 1995.

[50] B. V. Funt and M. S. Drew. Color constancy computation in near-mondrian scenes using a finite dimensional linear model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 553–563, 1988.

[51] B. V. Funt, M. S. Drew, and M. Brockington. Recovering shading from color images. In G. Sandini, editor, *ECCV-92: Second European Conference on Computer Vision*, pages 124–132. Springer-Verlag, May 1992.

[52] B. V. Funt, M. S. Drew, and M. Brockington. Recovering shading from color images. In G. Sandini, editor, *ECCV-92: Second European Conference on Computer Vision*, pages 124–132. Springer-Verlag, May 1992.

[53] B.V. Funt and G.D. Finlayson. Color constant color indexing. *IEEE PAMI*, 17:522–529, 1995.

[54] G.D.Finlayson. A theory of selection for gamut mapping color constancy. In *Proc.IEEE Conference o Computer Vision and Pattern Recognition*, 1998.

[55] G.D.Finlayson, M.S.Drew, and B.V.Funt. Spectral sharpening: sensor transformations for improved color constancy. In *J.Opt.Soc.Am.A*, volume 11(5), pages 1553–1563, 1994.

[56] G.D.Finlayson and R.Xu. Convex programming color constancy. In *The IEEE Workshop on Color and Photometric Methods in Computer Vision*, 2003.

[57] G.D.Finlayson, S.D.Hordley, and P.M.Hubel. Color by correlation: A simple, unifying frameworlk for color constancy. In *IEEE Transactions on Pattern Analysis and Machine inteligence*, volume 23(11), pages 1209–1221, 2001.

[58] G.D.Finlayson and RuiXia Xu. Non-interative comprehensive normalization. In *First European Conference on Color in Graphics, Image and Vision*, pages 134–139, 2002.

[59] T. Gevers and A.W.M. Smeulders. Color-based object recognition. *Patt. Rec.*, 32:453–464, 1999.

[60] T. Gevers and H. Stokman. Classifying color edges in video into shadow-geometry, highlight, or material transitions. In *IEEE Trans. Multimedia*, volume 5, pages 237–243, 2003.

[61] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.*, 12(1):79–94, 1991.

[62] G.West and M.H.Brill. Necessary and sufficient conditions for von kries chromatic adaption to give color constancy. In *J.Math.Biol.*, volume 15, pages 249–258, 1982.

[63] G.Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*, volume New York. 1982.

[64] G. Healey and D. Slater. Global color constancy: Recognition of objects by use of illumination invariant properties of color distributions. *J. Opt. Soc. Am. A*, 11:3003–3010, 1994.

[65] G. Healey and L. Wang. Three–dimensional surface segmentation using multicolored illumination. *Optical Eng.*, 37:1553–1562, 1998.

[66] B. K. P. Horn. Determining lightness from an image. *Comp. Vision, Graphics, and Image Proc.*, 3:277–299, 1974.

[67] HP/Microsoft. http://www.srgb.com.

[68] R.W.G. Hunt. *The Reproduction of Color*. Fountain Press, England, 5th edition, 1995.

[69] A. Hurlbert. Formal connections between lightness algorithms. *J. Opt. Soc. Am. A*, 3:1684–1692, 1986.

[70] R. Jain, R. Kasturi, and B.G. Schunck. *Machine Vision*. McGraw-Hill, 1995.

[71] H. Jiang and M.S. Drew. Shadow-resistant tracking in video. In *ICME'03: Intl. Conf. on Multimedia and Expo*, pages III 77–80, 2003. http://www.cs.sfu.ca/mark/ftp/Icme03/icme03.pdf.

[72] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multi-scale retinex for bridging the gap between color images and the human observation of scenes. In *IEEE Transactions on Image Processing*, 1997.

[73] K.Barnard and B.V. Funt. Analyzing and improving multi-scale retinex. In *Fifth Color Imaging Conference*, 1997.

[74] G.J. Klinker, S.A. Shafer, and T. Kanade. Color image analysis with an intrinsic reflection model. In *Proc. Second Int. Conf. on Comp. Vision*, pages 292–296. IEEE, 1988.

[75] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *Int. J. Comput. Vision*, 2:7–32, 1988.

[76] E.H. Land. Recent advances in retinex theory. *Vision Res.*, 26:7–21, 1986.

[77] E.H. Land and J.J. McCann. Lightness and retinex theory. *J. Opt. Soc. Amer.*, 61:1–11, 1971.

[78] L.T.Maloney. Computational approaches to color constancy. In *Stanford Applied Psychology Lab, Tech. Re*, page 01, 1985.

[79] C. Lu and M.S. Drew. Automatic compensation for camera settings for images taken under different illuminants. In *Color Imaging Conference*, 2006.

[80] C. Lu and M.S. Drew. Practical scene illuminant estimation via flash/No-flash pairs. In *Color Imaging Conference*, 2006.

[81] C. Lu and M.S. Drew. Shadow removal via flash/noflash illumination. In *MMSP'04: IEEE Workshop of Multimedia Signal Processing*, 2006.

[82] D.K. Lynch and W. Livingston. *Color and Light in Nature.* Cambridge University Press, 2001.

[83] L. T. Maloney and B. A. Wandell. Patent: Color Imaging Process, 1987. U.S. Patent # 4648051.

[84] L.T. Maloney and B.A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *J. Opt. Soc. Am. A*, 3:29–33, 1986.

[85] J.A. Marchant and C.M. Onyango. A color invariant for daylight changes: relaxing the constraints on illuminants. *J. Opt. Soc. Am. A*, 18:2704–2706, 2001.

[86] Wendy L. Martinez and Angel R. Martinez. *Computational Statistics Handbook with Matlab*, volume New York. 2002.

[87] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart. *J. App. Photog. Eng.*, 2:95–99, 1976.

[88] I. Mikic, P. Cosman, G. Kogut, and M. Trivedi. Moving shadow and object detection in traffic scenes. In *ICPR2000*, pages vol. 1:321–324, 2000.

[89] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[90] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2005.

[91] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004.

[92] J. Pinel and H. Nicolas. Estimation 2d illuminant direction and shadow segmentation in natural video sequences. In *Proceedings of VLBV*, page 197:202, 2001.

[93] A. Prati, M.M. Trivedi I. Mikic, and R. Cucchiara. Detecting moving shadows: algorithms and evaluation. In *PAMI*, volume 25, pages 918–923, 2003.

[94] C.E. Priebe. Adaptive mixture density estimation. pages 796–806. Journal of the Amerrican Statistical Association, 1994.

[95] R. Ramanath, W.E. Snyder, Y.F. Yoo, and Mark S. Drew. Color image processing pipeline in digital still cameras. *IEEE Signal Processing*, 22(1):34–43, 2005.

[96] R. Raskar, A. Ilie, and J. Yu. Gradient domain context enhancement for fixed cameras. In *ACCV 2004*, pages 27–30, 2004.

[97] R. Raskar, A. Ilie, and J. Yu. Image fusion for context enhancement and video surrealism. In *NPAR '04: 3rd Int. Symp. on Non-photorealistic Animation and Rendering*, pages 85–152, 2004.

[98] A. Renyi. *A Diary on Information Theory*. Wiley, 1987.

[99] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[100] Elena Salvador, Andrea Cavallaro, and Touradj Ebrahimi. Cast shadow segmentation using invariant color features. In *Computer Vision and Image Understanding*, volume 95, pages 238–259, 2004.

[101] J. Sangwine and R.E.N. Horne. The colour image processing handbook. In *ChapmanHall*, 1998.

[102] O. Schreer, I. Feldmann, U. Goelz, and P. Kauff. Fast and robust shadow detection videoconference applications. In *VIPromCom 2002, 4th EURASIP IEEE International Symposium on Video Processing and Multimedia Communications*, pages 371–375, 2002.

[103] D.W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley and Kegan Paul, 1992.

[104] W. Silver. *Determining Shape and Reflectance Using Multiple Images*. PhD thesis, MIT, 1980.

[105] S.Nadimi and B.Bhanu. Physical models for moving shadow and object detection in video. In *PAMI*, volume 28:8, pages 1079–1087, 2004.

[106] A.W.K. So, K.-Y.K. Wong, R.H.Y. Chung, and F.Y.L. Shadow detection for vehicles by locating the object-shadow boundary. In *Signal and Image Processing*, pages 479–483, 2005.

[107] J. Stauder, R. Melch, and J. Ostermann. Detection of moving cast shadows for object segmentation. In *IEEE Trans. Multimedia*, volume 1:1, pages 65–77, 1999.

[108] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet – srgb. http://www.w3.org/Graphics/Color/sRGB.

[109] M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.

[110] M.F. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[111] T.Horprasert, D.Harwood, , and L.S.Davis. A statistical approach for real-time robus background subtraction and shadow detection. In *ICCV Frame-Rate Workshop*, 1999.

[112] S. Tominaga and B.A. Wandell. Standard surface-reflectance model and illuminanat estimation. *J. Opt. Soc. Am. A*, 6:576–584, 1989.

[113] Robby T.Tan, Ko Nishino, and Katsushi Ikeuch. Illumination chromaticity estimation using inverse-intensity chromaticity space. In *Computer Vision and pattern Recognition*, volume 25(7), pages 417–421, 2003.

[114] M.J. Vrhel, R. Gershon, and L.S. Iwan. Measurement and analysis of object reflectance spectra. *Color Research and Application*, 19:4–9, 1994.

[115] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV01*, pages II: 68–75. IEEE, 2001.

[116] G. West and M.H. Brill. Necessary and sufficient conditions for von Kries chromatic adaption to give color constancy. *J. Math. Biol.*, 15:249–258, 1982.

[117] D. Xu and J.C. Principe. Learning from examples with quadratic mutual information. In *Neural Networks for Signal Processing*, pages 155–164. IEEE, 1998.

[118] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *Int. Conf. on Comp. Vision*, pages 464–471, 2003.

[119] J. Yedidia, W. T. Freeman, and Y.Weiss. Generalized belief propagation. In *In Advances in Neural Information Processing Systems*, volume 13, pages 689–695, 2001.

[120] Heinrich Zollinger. *Color : a multidisciplinary approach*, volume Weinheim, New York. 1999.