Estimating, Detecting and Removing Illumination in Images

Clément Fredembach

A thesis submitted for the Degree of
Doctor of Philosophy

University of East Anglia
School of Computing Sciences

November 1, 2006

Abstract

The response of an imaging device to a colour signal depends on the combination of the scene illuminant, surface reflectances and the device sensor sensitivities. If one is given only the device responses, it is not possible to subsequently separate the components of the colour signal, and this creates problems for many computer vision algorithms.

In this thesis, we focus on the interaction between illuminants and reflectances, and analyse the problems of estimating, detecting, and removing illumination in images.

Our starting point is to look at the chromagenic theory, wherein two images are taken of every scene: with and without a coloured filter, a similar situation as for the macular pigment present in the human retina. We propose that some reflectances perform better than others and show, through experimentation on synthetic and real images, that our bright-chromagenic algorithm outperforms current state of the art illuminant estimation methods.

Changing the focus from estimation to discrimination and assuming there are a fixed number of illuminants in a scene, we show that pixel-level illumination masks may be determined. Moreover, using segmentation techniques, we are able to obtain very accurate region-level masks.

Once multiple illuminants are detected, it becomes possible to remove them, so as to depict the scene under a single one. In this work, we focus on the problem of shadow removal and show that a simple, one-dimensional reintegration of gradient fields yields artifact-free, shadow-free images.

To adequately carry out the integration, we develop a method to output random Hamiltonian paths on grid graphs that are "complete by downsampling" with linear complexity. We further show that these paths can be efficiently used in other image-processing applications, such as image segmentation, denoising and texture classification.

Acknowledgements

A Ph.D. thesis is a personal work, or so the story goes. Of course, while the work itself is often done behind a computer there are many other factors outside of the computer that are part of the experience.

First of all I would like to thank my supervisor, Prof. Graham Finlayson for always stretching his “epsilon” (his word) available time to encompass all matters of listening to silly ideas and putting them in perspective, as well as providing insight into new directions and possibilities. This thesis would not be the same without him.

I am also indebted to Prof. Sabine Süsstrunk, my M.Sc. supervisor for her never ending interest in my work and well-being, and also for providing me with a desk at EPFL during the winters. My debts stretch as far as Vancouver and Dr. Mark Drew who generously allowed me to come and write my thesis there.

Making lists is customary in these first pages, though it is a dangerous exercise for a notoriously forgetful person. In both the colour lab, UEA and Norwich, I am grateful for the presence of Michal, Julia, Steve, Aurélien, Aurélie, Matthias, Kalliopi, C.J., Richard, Ross, Muriel, Peter and Dave for the nights out, the nights in, the discussions ranging from French grammar to imperialist Russia, and for trying to solve the great English mysteries of house insulation, water taps and bus systems (these questions have remained unanswered and will therefore be passed on).

Special thanks to Betty, my colleague/friend/partner and surrogate mother in time of illness to whom I cannot express my full gratitude in these few lines.

An often visited place on this campus is the graduate students association bar, where I would like to sent my thanks, building staff and customers alike for providing a safe haven for all the graduate students after a difficult day and for running always enjoyable pub quizzes.

Finally, my deepest thanks to my parents and siblings who, despite not being sure what this thesis is about, have never been less than fully supportive in all my endeavours, both at home and (mostly) abroad.

Contents

List of Figures

List of Tables

Publications

The following are publications related to this work by the author:

- G.D.Finlayson and C.Fredembach: "Fast Re-integration of Shadow Free Images," *Proc. of the IS& T SID 12th Color Imaging Conference (CIC), Scottsdale, 2004*

- C.Fredembach and G.D.Finlayson: "Path-based Shadow Removal," *Proc. of the 10th AIC Conference, Granada, 2005*

- C.Fredembach and G.D.Finlayson: "Hamiltonian path based shadow removal," *Proc. of the 16th British Machine Vision Conference (BMVC), Oxford, 2005*

- C.Fredembach and G.D.Finlayson: "Path-Based Colour Image Segmentation," *Proc. of the 3rd European Conference on Graphics Imaging and Vision (CGIV), Leeds, 2006*

- C.Fredembach and G.D.Finlayson: "Simple Shadow Removal," *Proc. of the International Conference on Pattern Recognition (ICPR), Hong Kong, 2006*

- C.Fredembach and G.D.Finlayson: "The 1.5-d Sieve," *To appear in Proc. of the 3rd European Conference on Visual Production Media (CVMP), London, 2006*

In Theory there is no difference between Theory and Practice; in Practice there is. That’s the Theory anyway.

Jan L. A. van Snepscheut

Chapter 1

Introduction

"Tis the First Sight and Second Thoughts ye have, and 'tis a wee gift an' a big curse to ye [...] First Sight is when you can see what's really there, not what your heid tells you ought to be there. Second sight is a dull sight, it's seeing only what you expect to see."

Terry Pratchett, The Wee Free Men.

The human visual system routinely uses second sight. For example: snow almost invariably appears white -it is one of its defining aspects- whatever the conditions are. Similarly, luminance edges, such as the ones created by shadows, are never mistaken for material changes: we do not see what is, but merely what we expect to see.

Artificial visual systems (such as cameras) on the other hand are blessed, or cursed, by "First Sight": they see the world as it is, without interpreting it. A signal resulting from a white reflectance (snow) lit by a blue light source (e.g., the sky) is, from a physical point of view, blue and will be seen as such by a camera.

As a consequence, computer-based applications designed to reproduce tasks from the human visual system are fragile, and so one needs to incorporate additional information: using "Second Thoughts" to look at a signal and analyze its meaning. This definition is, however, very general and would, in fact, encompass most of computer vision related research.

In this thesis, we restrict ourselves to the problem of interactions between light and surfaces in images. The problems addressed in turn are: 1) estimating the light incident upon a scene, 2) detecting the location of illuminants when several lights are present in the image and 3) removing the effect of a light source on an image. In this last instance, the primary focus will be on shadow removal, because of its prevalence and importance in both vision and photographic applications.

The contributions we make in this work are: a robust method for illuminant estimation: the bright-chromagenic algorithm. A novel algorithm for multiple illuminant detection based on the chromagenic theory. We then go on to develop an algorithm that allows us, under certain conditions, to output random Hamiltonian paths in linear time. We show how those paths can be used to remove shadows within a robust framework and provide additional “case studies” illustrating the possibilities of our Hamiltonian path-based approach.

This thesis is organized in the following way: In the second chapter, we look at the prior art and background of the image formation process. We review existing colour constancy theories and methods and vision-based methods for illuminant detection and removal.

The third chapter explores the problem of estimating illuminants. We present the chromagenic theory of illuminant estimation and analyze its behavior. From this analysis, we develop the bright-chromagenic algorithm for illuminant estimation and show that it remedies the weaknesses of the original formulation. We illustrate the performance of our algorithm using various experiments and show that it significantly outperforms other existing illuminant estimation algorithms.

Chapter 4 is about detecting illuminants. We extend the invariant image methodology which removes shadows based on incomplete edges in two ways: first, we show

how discontinuities in edge maps can be completed and so how shadows can be identified. This insight then forms the basis of a region-based shadow removal algorithm. We then show how the chromagenic theory can be applied to the more general illuminant detection problem and demonstrate that we are able to obtain accurate illumination masks for both indoor and outdoor multiply-lit scenes.

In Chapter 5 we look at removing shadows. We provide a robust framework for shadow removal. Here we propose a graph-theoretical algorithm to generate a certain class of random Hamiltonian paths in linear time and go on to show that they fit the robust framework and deliver good results. We illustrate the performance of several shadow removal algorithms on a variety of images. We further show that if the conditions of shadow formation are known, then shadow regions can be removed without using paths by adding a constant whose value is found through constrained minimization.

Chapter 6 focuses on applications of Hamiltonian paths. In this chapter, we show the usefulness of said paths in other areas of image processing. We concentrate on image segmentation and scale-space. Namely, we exhibit algorithms that can effectively deal with noise removal, image segmentation and texture classification. All those methods stem from a theoretical proof that analyzing an image along one path is equivalent to a 1-dimensional processing, while analyzing the image according to a very large number of paths corresponds to a 2-dimensional process.

Finally, Chapter 7 concludes this thesis by summarizing our findings and offers insights into possible extensions of this work.

Chapter 2

Background

2.1 Colour Constancy & Illuminant Estimation

The image formation process within an imaging system, be it a human eye or a digital camera, is dependent on three factors: the physical properties of the imaged surfaces -referred to as reflectances-, the light incident upon those surfaces and the characteristics of the imaging system. In general, numerical measurements made by the imaging system do not allow for the subsequent dissociation of these factors: i.e., given an image without additional information, one cannot invert that process and explicitly separate reflectance, illumination and the characteristics of the system. The problem of separating illumination from reflectance is often referred to as the *colour constancy* problem.

The importance of solving for colour constancy is illustrated by the fact that a colour signal captured by an imaging system is the *combination* of surface reflectance properties and illumination conditions. The difficulty lies in the variety of possible illuminants, since even commonly encountered ones have very different spectral properties. Fig. 2.1 shows the same scene pictured under three standard illuminants: sky light, neon light and tungsten light -a common light bulb. One can see that, despite the reflectances in the scenes being identical, the colours in the images vary significantly and, as illustrated by the histograms underneath, so do the numerical values used to represent the images. These variations significantly weaken the performance of algorithms that rely on color information to detect [JR99] , recognize [TP91,MAU94] or track [JD03] objects within a scene, as well as frameworks that deal with large image databases for indexing [SB91] or scene analysis [KSK90].

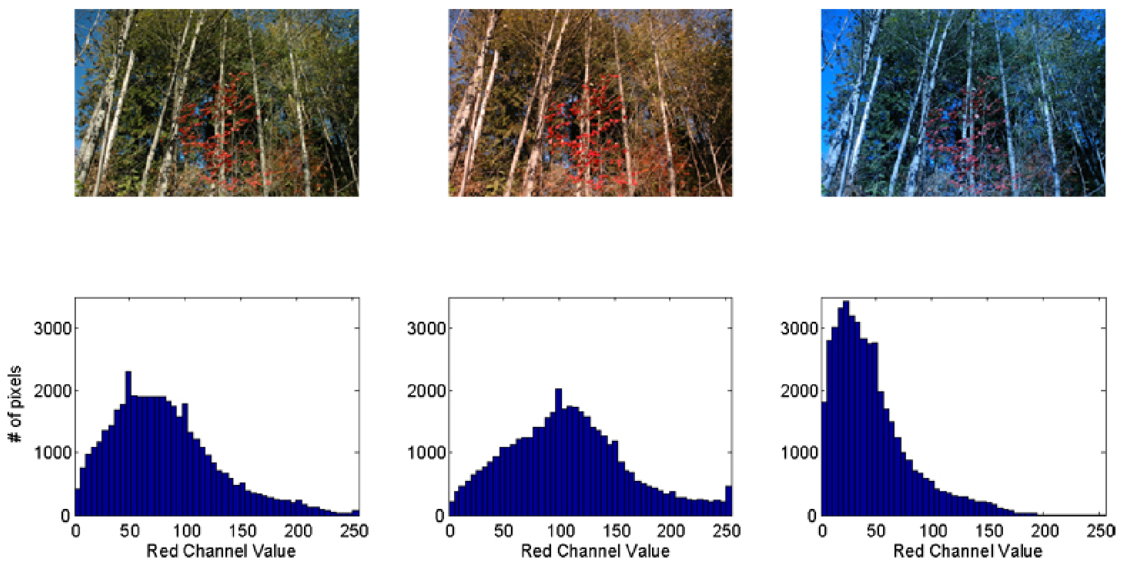


Figure 2.1: *Top row: the same scene under 3 different illuminants (a picture of a picture taken in a light booth). Bottom: the histogram (red channel only) of those images. The great variability of the colours despite it being the same scene create problems for many applications.*

The human visual system is however, to a certain extent, colour constant [BF97, Bra98, AR86]. For instance consider the case shown in Fig. 2.2 where snow, which is a white reflectance, is illuminated by the sky, a blue illuminant. The resulting colour signal is therefore blue and this blueness is what the camera sees. To our eyes however, the snow appears white, because our own visual system discounts the colour of the illumination.

Apart from changes in illumination, both image intensity and perceived colour can vary according to the shape of the objects, viewing and illumination geometry, thus the general colour constancy problem is very hard indeed. To simplify the problem, a

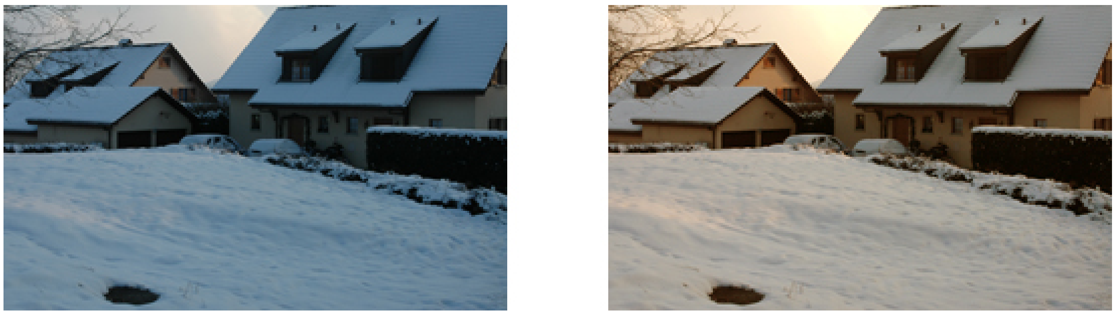


Figure 2.2: *An outdoor scene as seen by a digital camera (Left) and as it would be perceived by the HVS (Right)*

Lambertian model of surface reflectance has been used by many researchers [MW86, D'Z92, For90], where it is assumed that surfaces appear equally bright independently of the viewing direction, i.e., they are ideal diffuse surfaces. Using this model, the image formation process can be described as:

$$\rho_k = \int_\omega E(\lambda)S(\lambda)Q_k(\lambda)d\lambda \tag{2.1}$$

In this equation $S(\lambda)$ represents the surface reflectance. It defines the fraction of the incident light that is reflected on a per-wavelength basis. $E(\lambda)$ is the spectral power distribution of the illuminant, which defines the power emitted by the illuminant incident to $S(\lambda)$ at each wavelength. $Q_k(\lambda)$ is the spectral sensitivity of the imaging device's k^{th} sensor, specifying what proportion of the light incident at the sensor is absorbed at each wavelength. Multiplying these terms and integrating over ω, the range of wavelength to which the sensors have a non-zero response, gives ρ_k: the response of the imaging device's k^{th} sensor. In the case of the human visual system and digital cameras, the range of wavelengths ω is generally the visible spectrum: 380-700[nm]. For practical purposes, we concentrate on trichromatic imaging systems -that is, we have three sensors whose sensitivities are concentrated in the long (red), medium (green) and short (blue) part of the visible spectrum- and thus we will generally describe the sensor responses as R,G and B. Fig. 2.1, shows that changing the spectral power distribution of the incident illuminant changes the sensor responses: light change is a first order effect.

Another way to measure colour is to look at sensor responses in terms of their chromaticity, obtained by discarding the intensity information. Chromaticities can be a useful representation of sensor responses because changes in surface colour due to geometry and viewing angle typically depend on intensity, which is factored out in chromaticity space. There are many ways of discarding intensity information and we give here two possibilities of doing so. Three dimensional chromaticities can be obtained by:

$$c_k = \frac{\rho_k}{\rho_R + \rho_G + \rho_B}, \; k = \{R, G, B\} \tag{2.2}$$

That is, the sensor responses of each channel are normalized by the sum of the responses in all three channels. Used in some algorithms, two dimensional chromaticities are obtained by dividing two of the sensor responses by the response of the third, for example:

$$c_1 = \frac{\rho_R}{\rho_G}, \; c_2 = \frac{\rho_B}{\rho_G} \tag{2.3}$$

To solve for colour constancy, one needs to transform the ρ_k (or the c_k) so that they correlate with $S(\lambda)$, i.e., become independent of $E(\lambda)$. It has been shown in [HHFD97] that when the illumination is known it is relatively easy to recover an image that is independent of the illuminant. We will hence focus on solving for colour constancy by estimating the scene illuminant.

2.1.1 Discrete Model

The difficulty of the colour constancy problem is linked with the number of degrees of freedom it takes to determine lights and surfaces. We begin with the observation that spectral quantities can be sampled at m points and the image formation equation (2.1) can be rewritten as:

$$\rho_k = \sum_{i=1}^{m} E(\lambda_i)S(\lambda_i)Q_k(\lambda_i)\Delta\lambda \tag{2.4}$$

where the λ_i are the sample points and $\Delta\lambda$ is the sampling interval. Sampling at intervals $\Delta\lambda \simeq 10[nm]$ results in human colour responses which are visually indistinguishable (when assessed using colour difference formula) from the same spectrum at a finer sampling [Wan95]. From equation (2.4), we see that if there are n different surfaces in the image, we have $3n$ known values -the RGB sensor responses. However, if surfaces and illuminants are both described with the discrete model we have $m(n+1)$ unknown values to solve for, as each one of the m sampling points describes n surfaces plus a light component.

The number of parameters can, pragmatically, be reduced to $3n+3$ using the fact that light and surfaces, in a trichromatic imaging system, are described by 3 parameters each. That is, we can change the goal of colour constancy to recovering the RGBs of the surfaces and lights in a scene. Using 2D chromaticity vectors instead of 3D sensor outputs, one can further decrease the number of parameters to $2n+2$. However, the number of known values then stands at $2n$, so the problem remains under-constrained.

An alternate way to look at the discretisation of equation (2.1) is to describe the SPD of illuminants and surface reflectance functions as sums of basis functions. The surface reflectances $S(\lambda)$ can be approximated as:

$$S(\lambda) = \sum_{i=1}^{d_S} S_i(\lambda)s_i \tag{2.5}$$

where the $S_i(\lambda)$ are basis functions for reflectances and $\underline{s}$ is a $1 \times d_S$ vector of weights. To adequately model the space of reflectances one needs between 3 and 8 basis functions, depending on the desired accuracy [PJ89]. Similarly, $E(\lambda)$ can be rewritten as:

$$E(\lambda) = \sum_{i=1}^{d_E} E_i(\lambda) e_i \tag{2.6}$$

Judd in [JMW64] showed that daylight illuminants are well modelled by 3 basis functions.

The basis functions can be chosen by separately performing principal component analysis on reflectance and illuminant data [Coh64, MW86]. Marimont and Wandell in [MW92] however proposed that sensor responses themselves are to be used to find the basis functions. They concluded that 3 basis functions, $d_S = 3$ and $d_E = 3$, were sufficient to model most reflectances under a variety of illuminations (this work effectively links the idea of describing light and surface colour by the RGBs with the basis approach). Substituting equations (2.5) and (2.6) in (2.1) allows us to rewrite the image formation equation as a matrix transform, where a lighting matrix $\Lambda(\underline{e})$ maps reflectances from $\underline{s}$ onto the sensor responses $\underline{\rho}$.

$$\underline{\rho} = \Lambda(\underline{e})\underline{s} \tag{2.7}$$

where the kj^{th} term of this lighting matrix is equal to:

$$\Lambda(\underline{e})_{kj} = \int_\omega S_j(\lambda) Q_k(\lambda) \left[\sum_{i=1}^{d_E} e_i E_i(\lambda) \right] \tag{2.8}$$

In [MW86], Maloney and Wandell have shown that a trichromatic system that views surfaces under an unknown illuminant can, through algebraic means alone, recover two reflectance descriptors per surface, implying that if lights can be modelled as 3-dimensional, surface reflectances have to be 2-dimensional for perfect recovery to be possible. This result, while mathematically interesting, is not useful in practice. It would be useful if the world was composed of shades of yellows and blues, since then colour would be 2-dimensional. But we wish to have reds too: colour is always 3-dimensional.

This finite dimension model approach has also been investigated in the case of changing illumination -the same scene under different lights- by D'zmura and Iverson [DI93]. Their idea was to recover three reflectance descriptors per surface using multiple views. They show that recovery is possible when the same surfaces are rendered under different lights. Leaving aside the plausibility of this assumption, the method is numerically unstable and tends to work only when the assumed model holds exactly. This idea that constancy might be easier when we have more than a single RGB measurement per pixel is one we will return to in this thesis.

2.1.2 Assumptions About the World

The methods presented in the previous section use reduced dimensional models to deal with the under-constrained nature of the problem. Another, widely used, approach to solve for color constancy is to make assumptions about the world. These methods usually try to estimate the colour of the illuminant, i.e., the 3-dimensional vector $\underline{\rho}^{\mathrm{E}}$ corresponding to the RGB values of an achromatic reflectance observed under the scene illuminant.

The first approach, generally known as *Max-RGB* proposes that the colour of the illuminant can be estimated by taking the maximal sensor response in each colour channel, that is:

$$\underline{\rho}^{\mathrm{E}} = [\max(\rho_R), \max(\rho_G), \max(\rho_B)] \tag{2.9}$$

This estimation is accurate on the condition that the observed scene contains a white-like surface; that is, a surface that reflects light equally at all wavelengths and that is also maximally reflective. For that surface, equation (2.1) becomes:

$$\rho_k = \int_\omega E(\lambda)Q_k(\lambda)d\lambda,\ S(\lambda) = 1\ \forall\lambda \in \omega \tag{2.10}$$

and thus the sensor responses only depend on the light. We note that for this method to work, a white surface is not strictly necessary. Instead, one can imagine a blue and a yellow surface where max(ρ_B) will be based on the blue surface, while max(ρ_R) and max(ρ_G) will be calculated with the yellow reflectance.

This method was implicitly proposed in Land's Retinex algorithm [LM71]. The Retinex was originally intended to be a computational model of human vision that assumed human perception was based on relative responses. That is, colour perception of a surface does not depend on its intrinsic RGB values, but rather on the relation of its RGBs to the ones in its surroundings. Moreover, McCann has argued in a series of studies that the maximum plays a crucial role in colour perception [MMT76, McC04].

Another commonly used method to estimate illuminants is the so-called *Gray-World* algorithm. In Gray-world, one estimates the illuminant by calculating the average, over the whole image, of the sensors response in each channel:

$$\underline{\rho}^{\mathrm{E}} = [\mathrm{mean}(\rho_R), \mathrm{mean}(\rho_G), \mathrm{mean}(\rho_B)] \tag{2.11}$$

This algorithm has been proposed in various forms [Buc80, GJT88] and assumes that the average, over all the N *pixels* in the image, of the reflectances is constant over the visible spectrum:

$$\sum_{i=1}^{N} \frac{S_i(\lambda)}{N} = \alpha(\lambda) = \alpha,\ \alpha \in]0,1] \tag{2.12}$$

where α can take any value but is generally assumed to be 0.5 (true gray). If equation (2.12) holds, then the illuminant can readily be estimated. Since an achromatic surface reflects light at all wavelengths equally, the sensor responses therefore become:

$$\rho_k = \alpha \int_\omega E(\lambda) Q_k(\lambda) d\lambda \tag{2.13}$$

A problem with the original formulation [Buc80] is that, because the average is calculated over all the pixels, the algorithm is biased towards large areas. Gershon et al. [GJT88] have proposed a modified algorithm such that each reflectance in the image has an equal weight, independently of its size.

A different gray-world like method, named database grayworld, has been proposed in [BCF02] where instead of assuming the average of the reflectances to be gray, it is chosen to be the average of a reflectance database -if possible, the one on which the algorithm is tested. Let $\underline{\mu}_{DB}$ be the mean vector, over the database, of the reflectances and $\underline{\mu}$ be the observed mean of a scene. This method performs better than the simpler grayworld formulation as the assumption has a better chance of being verified. In mathematical terms, we have that:

$$\underline{\mu}_{DB} = [\mu_{DB}(\rho_R), \mu_{DB}(\rho_G), \mu_{DB}(\rho_B)] \tag{2.14}$$

and

$$\underline{\mu} = [\mu(\rho_R), \mu(\rho_G), \mu(\rho_B)] \tag{2.15}$$

The quantities $\underline{\mu}_{DB}$ and $\underline{\mu}$ are related by a 3×3 diagonal matrix, D, where:

$$\underline{\mu}_{DB} = D\underline{\mu} \tag{2.16}$$

with

$$D = \begin{pmatrix} \frac{\mu_{DB}(\rho_R)}{\mu(\rho_R)} & 0 & 0 \\ 0 & \frac{\mu_{DB}(\rho_G)}{\mu(\rho_G)} & 0 \\ 0 & 0 & \frac{\mu_{DB}(\rho_B)}{\mu(\rho_B)} \end{pmatrix} \tag{2.17}$$

The color of the estimated illuminant $\underline{\rho}^E$ is then equal to the diagonal terms of D. The idea of mapping RGBs using a 3×3 diagonal matrix is often referred to as the diagonal model. It is generalized by supposing that if a scene is imaged under two different lights, E_1 and E_2, the sensor responses can then be written as:

$$\underline{\rho}_1 = \Lambda(\underline{e}_1)\underline{s} \quad \text{and} \quad \underline{\rho}_2 = \Lambda(\underline{e}_2)\underline{s} \tag{2.18}$$

The mapping between the sensors responses is then defined by:

$$\underline{\rho}_1 = \Lambda(\underline{e}_1)\Lambda(\underline{e}_2)^{-1}\underline{\rho}_2 \tag{2.19}$$

In the database grayworld method, the average values of the scene are mapped onto the database average, which is what the matrix D does.

More recently, Finlayson and Trezzi developed a framework based on the Minkowski norm families [FT04]. They showed that both the gray-world and max-RGB assumptions were particular instances of Minkowski norms:

$$\mu_p(\underline{X}) = \left(\frac{\sum_{i=1}^{N} |X_i|^p}{N} \right)^{\frac{1}{p}} \tag{2.20}$$

Where μ_p is the *p-norm* of $\underline{X}$. Taking this norm on each channel (R,G,B) separately, one sees that $p = 1$ corresponds to taking the mean -like grayworld- and that $p = \infty$ is equivalent to taking the maximum value -like Max RGB. The authors investigated whether a "shade of gray" could be a more appropriate assumption and found that any $p \in [4, 6]$ was a better performing norm than either 1 or ∞.

The shade of gray method has been corroborated by Van de Weijer and Gevers in [vdWG05] where the norm of RGB differences is shown to work equally well, albeit with slightly different values of p.

2.1.3 Statistical Framework and Set of Solutions

We have seen in the beginning of this chapter that the colour constancy problem is underconstrained. This implies that, in general, there is no unique solution for the combination of lights and surfaces in a scene giving rise to an image. A common goal of most of previously presented algorithms, however, is to solve for a unique answer. With that in mind, more recent algorithms have adopted the approach of seeking the *set* of all possible solutions and finding, within this set, the best solution to the colour constancy problem. A different approach is to compute the likelihood of different solutions under some weak probabilistic assumptions, the scene illuminant estimate can then be chosen to be the most likely. So, why are there multiple solutions?

A reddish RGB sensor response is consistent with both the image of a white surface observed under a red light and a red surface observed under a white light. In [For90], Forsyth developed an algorithm, named CRULE, to exploit this fact. CRULE is based on colour gamuts: the set of all RGBs that can be observed under a given light. Forsyth showed that these gamuts are convex, bounded and are subsets of the possible image colours. The colour gamut of a light can therefore be constructed by calculating the convex hull of RGB responses given by all surfaces observed under that light.

Forsyth's gamut mapping theory shows how these gamuts can be used in solving for colour constancy. When presented with a test image, the candidate solutions for colour constancy are the lights for which all of the image colours fall within their likely gamut.

Once the set of feasible illuminants has been determined, the second step is to select a single illuminant from that set as an estimate of the scene illuminant.

In Forsyth's original formulation, the illuminants themselves are not explicitly represented. They are, instead, defined by the 3×3 diagonal matrix that transforms the sensor responses to what their values would be if the scene was observed under a known, *canonical*, illuminant -it is the same diagonal mapping approach as shown in equations (2.18) and (2.19). With respect to this methodology, any map (a 3×3 diagonal matrix) that takes the gamut of the image inside the canonical gamut represents a candidate light. The selected illuminant map is the one which leads to the largest gamut volume (image colours are effectively made as colourful as possible).

In [Fin96] Finlayson proposed two additional ideas to the gamut mapping theory. The first one is that features like shape and shading affect the magnitude of the recovered light but not its colour. Thus, he proposed to perform gamut mapping in a 2D chromaticity space, using $c_1 = \frac{R}{B}$ and $c_2 = \frac{G}{B}$. With respect to this chromaticity space, illuminant change is a 2D diagonal map (a 2×2 diagonal matrix). As such, Forsyth's theory is directly applicable with the advantage of a simpler and faster implementation. The second idea is that the mappings themselves can be constrained by restricting them to expected illuminants, for instance purple illuminants do not occur in practice.

Barnard proposed in [Bar99] that, instead of selecting the estimate of the scene illuminant as one light among the set of plausible illuminants, one could take the numerical average of the set as the estimate of the scene illuminant. This method to select an estimate from the set was found to yield better estimations than Forsyth's or Finlayson's methods, although Finlayson and Hordley reported in [FH99] that taking the median instead of the mean gave the best results overall.

A complementary way to deal with the ill-posedness of the colour constancy problem is to use a probabilistic framework. In colour by correlation [FHH01], the scene illuminant is estimated by first comparing the chromaticities of the test image -in this work, lights are represented by 2D chromaticity distributions- to the 2D chromaticity distributions of expected illuminants in the set. Each image chromaticity has a certain likelihood of occurring under a given illuminant (e.g., the bluest chromaticity cannot be observed under the reddest light) and the sum of those likelihood values defines the plausibility of each test illuminant being the scene illuminant, i.e.,

$$l(E|C_{im}) = \sum_{\forall \underline{c} \in C_{im}} \log(p(\underline{c}|E)) \tag{2.21}$$

where l is the likelihood function of E being the scene illuminant given the chromaticities of the image C_{im} and $p(\underline{c}|E)$ is the probability of observing the 2D chromaticity vector $\underline{c}$ under the illuminant E. An implementation of colour by correlation in a 3D RGB color space instead of a 2D chromaticity has been proposed in [BMF00].

Brainard and Freeman in [BBS97] have formulated colour constancy as a Bayesian problem but used the finite linear model to represent light and surfaces as a weighted sum of basis functions, as represented in equations (2.5) and (2.6). In their work, basis functions and their weights were found using principal component analysis. Bayesian decision theory is then used to recover the combined vector of weights of surfaces and light in the image. The number of dimensions to recover -three for each reflectance in the image plus three for the light- is however too large for the algorithm to be used in practice (in [BBS97], only up to eight reflectances could be analytically recovered in a reasonable time).

Other probabilistic approaches that use either voting, soft probabilities or neural networks have been investigated. In [DI94], D'Zmura and Iverson used a linear model of surface reflectance and illumination -equations (2.5) and (2.6)- to derive the probability of observing a chromaticity coordinate (in the CIE-xy sense [WS82]) under a given illuminant. By selecting a large number of surfaces, a good estimate of the probability can be obtained.

An algorithm where the illuminant is selected by voting has been proposed by Sapiro in [Sap85, Sap98]. As in [DI94], lights and surfaces are represented using linear models that define a probability distribution. For each RGB triplet, this distribution is used to select a reflectance from sensor responses and then to estimate the illuminant. If the illuminant is deemed realistic -i.e., it falls on or near the expected lights- a vote is cast in favor of that illuminant. This process is repeated for all RGBs in the image and the selected estimate of the illuminant will be the candidate that has the most votes.

Finally, a neural network approach to colour constancy has been proposed in [CFB02].

This method takes as an input to the neural net a binary chromaticity histogram and outputs the estimated 2D chromaticity of the illuminant white point (that is, the colour observed when the illuminant is reflected by a white surface). While this approach can deliver good estimates, results from [BMCF02] and [HF06] suggest that its performance is strongly dependent on the training set being similar to the testing set.

2.1.4 Illuminant Estimation: The Current Level of Performance

At this point, we might want to look back at the presented algorithms available for illuminant estimation and assess their performance. The accuracy of the illuminant estimation is generally assessed using the angular error, an intensity independent measure between the sensor responses of a white reflectance under both the estimated and actual scene illuminant. If we denote those responses by $\underline{\rho}_{\text{est}}$ and $\underline{\rho}_E$ respectively, the angular error e_{Ang} is calculated as:

$$e_{\text{Ang}} = \operatorname{acos}\left(\frac{\underline{\rho}_E^{\mathrm{T}}\underline{\rho}_{\text{est}}}{\|\underline{\rho}_E\|\ \|\underline{\rho}_{\text{est}}\|}\right) \tag{2.22}$$

Our own experiments find that an angular error lower than three degrees is necessary for pleasing image reproduction. In other work, Funt et al. in [FBM98] found that the current angular error delivered by most algorithms is not sufficient to support colour-based object recognition.

More recently, Barnard et al [BMCF02] have devised a framework to compare different algorithms based on their freely available dataset [Bar02]. They measured the error between the estimated and actual illuminant of a variety of algorithms on both synthetic and real data.

In both these studies, the accuracy of the illuminant estimation algorithm is summarized as either the mean or Root Mean Square (RMS) angular error over the entire dataset. However, in [HF06] Hordley and Finlayson showed that, if one wants to summarize the performance of an illuminant estimation algorithm over a dataset, one should use the median angular error instead of the mean or RMS. The use of a median statistic also permits to assess if the difference of performance between two algorithms is statistically significant at chosen confidence level using, for example, the Wilcoxon sign test [HT01].

Irrespectively of the chosen metric, all these studies concluded that the more complex methods such as colour by correlation and gamut mapping performed significantly better than simpler, assumption based, methods like max-RGB and gray-world. In practice, however, the simpler methods are still commonly used because their lower complexity enables the images to be processed at a very high speed.

2.1.5 Non-Lambertian Algorithms

An aspect not really treated in this chapter is that the vast majorities of the methods presented assume a Lambertian model of surface reflectances and flat surfaces. There exist, however, some algorithms that use non-Lambertian constraints in solving for colour constancy. For dielectric materials, it is known that the highlight colour is the same as the prevailing light, and so numerous algorithms have been developed for finding the specular colour, see [KSK88, TW89, FS99, TNI03].

The presence of mutual illumination [FDH91] or shadows [FF94] has also been used as a cue to solve for colour constancy.

These algorithms have the weakness that they all require their respective physics-based assumptions to be present within the scene. Additionally, one also needs to know the location of features such as highlights, shadows or mutual illumination in this image; this is a difficult task. The physical assumptions made by these algorithms are often violated in real images and thus, the performance of these methods can be unreliable on general data.

2.2 Multiple Illumination & Illuminant Detection

Almost all illumination estimation algorithms previously introduced have something important in common, they aim to recover the supposed *unique* scene illuminant. Images bearing multiple illuminants are however frequent in natural scenes: the use of a flash, the presence of shadows or indoor images comporting both office light and daylight are all instances where more than one illuminant is present in the resulting image; an example of a typical scene with multiple illuminants is shown in Fig. 2.3a. Since colour constancy is generally an underconstrained problem, adding another unknown illuminant to a scene will further complicate the image analysis. Moreover, if an algorithm estimates a single illuminant but two or more are present in an image, the estimation will necessarily be wrong.

In fact, the multiple light problem is even more complicated than one might think. For example: in a scene lit by two distinct illuminants E_1 and E_2, a given pixel can be illuminated by either E_1, E_2 or a combination of both $E_{\text{mix}} = \alpha E_1 + \beta E_2$. An illustration of the multiple illumination of images is shown in Fig. 2.3b where the prevailing illuminant is highlighted.

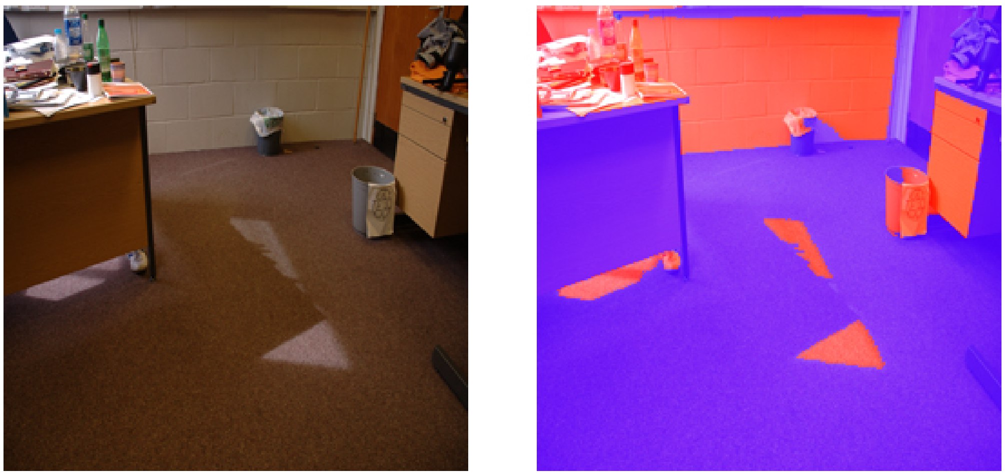


Figure 2.3: *An indoor scene with 2 ambient lights: sunlight through the windows and ceiling lighting. Scene geometry plays a role in how the pixels are illuminated. Right: regions of the image predominantly illuminated by E_1 (red) and E_2 (blue)*

For the sake of simplicity, we will focus on the case where two distinct illuminants are present in the image. Considering the image formation process at a pixel level, equation (2.1) can be rewritten as:

$$\rho_k^X = \int_\omega E^X(\lambda) S^X(\lambda) Q_k(\lambda) d\lambda \tag{2.23}$$

where ρ_k^X is the response of the sensor at pixel location X, $E^X(\lambda)$ is the incident light on pixel X and $S^X(\lambda)$ is the scene reflectance at location X. In this case, even if we know both illuminants, E_1 and E_2, and the sensor sensitivities, the problem is still ill-posed. This is because the sensor responses ρ_k^X can be described by either

$$\rho_k^X = \begin{cases} \int_\omega E_1^X(\lambda) S_1^X(\lambda) Q_k(\lambda) d\lambda & \text{if the light incident to } X \text{ is } E_1 \quad (2.24a) \\ \int_\omega E_2^X(\lambda) S_2^X(\lambda) Q_k(\lambda) d\lambda & \text{if the light incident to } X \text{ is } E_2 \quad (2.24b) \end{cases}$$

Knowing the illuminants is therefore not sufficient to recover the scene reflectances. One also needs to know which part of the image is illuminated by which light. Because of the difficulty of the multiple illumination problem, most of the research is focused not on estimating these illuminants but on *detecting* the prevailing light incident at a pixel; once detected, the illuminants can then be estimated separately.

The bulk of both prior art and current research in illuminant detection focuses on finding shadows in images, by far the commonest occurrence of multiply illuminated scenes.

2.2.1 Shadow Detection

A shadow is cast in a scene when an object lies in the path of the direct illumination source. If a scene is illuminated by two or more sources, then the shadow and non-shadow regions of an object may differ not just in terms of their relative brightness, but also in terms of their relative colour. For example, in a typical outdoor scene, the non-shadow parts of the image are illuminated by a mixture of direct sunlight and skylight. In contrast, shadow regions are lit mostly by skylight. These two illumination sources differ significantly both in brightness and colour -see Fig. 2.4 for an illustration- and, as a result, so do the image pixel values corresponding to shadow and non-shadow regions.

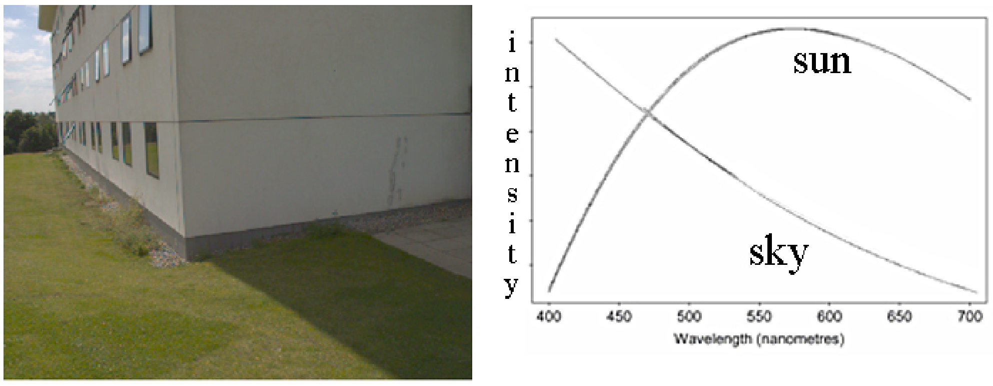


Figure 2.4: *An outdoor image containing a shadow. And the SPD of both illuminants: sun+sky light and sky-light only. Note the difference across the visible spectrum.*

In photography, shadows are often accidental and/or unwanted artifacts that in some conditions (e.g., cityscapes, flash) cannot be avoided. Finally, when working with images that have a large bit depth, the presence of a shadow can indicate a High Dynamic Range (HDR) image. HDR images cannot be displayed on typical CRT monitors, however, if one can remove or attenuate the shadow, the dynamic range can then be compressed and the image properly displayed.

Detecting shadows is a difficult problem since shadows are created in diverse ways and can greatly vary in intensity, colour, shape and sharpness. As a result, additional information is often needed for an accurate detection.

In essence, what distinguishes the shadow problem from the general multiple illumination one is that shadows are generally darker than their surroundings. The problem caused by shadows in computer vision applications is, however, essentially the same, that is a lot of algorithms for a variety of tasks can fail in cases of changing illumination conditions.

We can categorize detection methods in broadly two different approaches: video-based, where extra information is provided in the form of an image sequence, and single image methods where one recovers shadows either via user supplied hints or through a physical model of shadow formation.

We give here an overview of the most recent methods of shadow detection. Older methods have been reviewed in a comparative study by Prati et al. in [PTMC03] whose conclusion was that specific methods are desirable for better performance, for most of them are not readily “portable” between different applications.

Video-based methods

In almost all video methods, the input is a sequence of images taken from a fixed camera and the desired output is a background image that is free of shadows.

Weiss in [Wei01] observed that given an outdoor video sequence over a long period of time: cast shadows (due to objects occluding the sun) move. It follows that the edges which are constant throughout the frames are related to the scene structure and not to the shadows. Weiss showed that a shadow free background could be obtained by taking the median edges of the sequence, effectively removing the shadows without extracting them.

Weiss approach has been extended by Matsushita et al. in [MNIS02] and [MNIS04] using several light sources and multiple cameras to recover, using multiview stereo algorithms, a view-dependant model of a reflectance only background image. They were also able, with a thresolding operator, to recover shadow masks from the illumination images.

In [LDB06], Leone et al. looked at frame differences from video surveillance data. Their goal was to distinguish textured objects from shadows using a Matching Pursuit algorithm [MZ93]. The idea of using texture classifiers has also been used in [HHD99] and [SMO99] based on the assumption that, while shadows alter both brightness and colour, the texture of a shadow region is essentially unchanged from its non-shadow equivalent and so, texture information can therefore be used in finding out whether a moving element in a sequence is a shadow cast by an object or the object proper.

In an approach based on computer graphics natural image matting methods, Chuang et al. [CGC$^+$03] have been able to extract both shadow and lit images from a video. Their method however assumes a that primary point light source illuminates the scene and that fairly strong shadows are cast.

A multi-image method, as opposed to a video proper, has been proposed by Yoon et al. [YKE02]. The starting point is two -or more- images of the same scene illuminated by different point light sources. Using the difference between those images, they could both determine the shadows in the scene and render the object without shadows. While providing good results, this method requires a fairly controlled environment and is not practical outside of the lab.

Other multi-image methods are popular in remote sensing. In this framework, both images and potential shadows are constrained to a specific class, detecting cloud shadows in earth-like images, and are therefore not readily applicable to a more general problem (see [SF90] and [WHR91] for examples).

Single Image Methods

The single image problem of detecting shadows from the information present in only one image is more general but also more difficult than its video counterpart since less information is readily available.

Jiang et al. in [JW94] detected shadows by segmenting the input image and classifying its regions as either shadow or non-shadow. To do so, they proposed that the darkest image regions are possible shadows. These candidates were then evaluated based on their geometry, assuming that the shape of shadow regions differed from the one of actual objects. This approach suffers from its simplicity and is only accurate in scenes with simple object shapes and non-textured backgrounds.

Following a gradient-based approach, Tappen et al. [TFA03] classified image derivatives, depending on their direction and amplitude, as either shadow induced, material induced or undetermined. This three-way labelling is obtained with a classifier trained on a variety of reflectance and shadow transitions. The undetermined derivatives are then, in a second step, assigned a shadow or material label using a belief propagation algorithm that propagates information from reliably classified pixels.

In [LB05], Levine et al. also classified image edges as either shadow or material changes. They used a support vector machine, trained on colour and luminance ratios, for classification. In their method, it is assumed that a shadow transition occurs when there is an important change in luminance coupled with a weaker change in colour. Based on their detection results, they also proceeded to remove shadow regions from the image by assigning them the average brightness and colour values of their neighboring regions.

Both the Levine and Tappen approaches often work well. However, there are significant failures which manifest themselves in images: not all shadows are removed and there can be edge artifacts. In part these methods fail because some pixel derivatives can indicate *both* a shadow and a material edge, a common occurrence in the case of occlusion shadows.

More recently, Wu and Tang [WT05] used a Bayesian approach to extract shadows where they opted for user supplied hints to disambiguate regions that could be wrongly estimated by an automatic method. Their method delivers good results but cannot be ported to an automatic framework.

To alleviate the problem of training classifiers or having to rely on user supplied data, we propose to expand the physics-based method of Finlayson et al, first proposed in [FH01] that we discuss in more detail in section 2.2.2

2.2.2 Invariant Images for Shadow Detection

In this section, we look at the construction of shadow free images based on physical constraints proposed in [FH01], which takes a regular colour image as input and gives 1D, illumination invariant, representation of the image that is obtained by projecting 2D log-chromaticities in the "correct" direction.

Main Idea and Assumptions

We start by reviewing here the two main assumptions made in [FH01] that are necessary for the existence of an invariant image: the lights present in the image can be modelled by Planck's black-body radiators law and the camera sensors are narrow-band.

A Planckian illuminator is a light source that behaves in accordance with Planck's law of black-body radiators [WS82]. The law states that a perfect spherical radiator, when heated at a temperature T emits electromagnetic radiations at specific wavelengths. Examples of -approximate- Planckian light sources include the human body (emission peak in the infrared part of the spectrum), light bulbs (orange/red), the Sun (yellow/white) and the sky (blue). The Spectral Power Density of some Planckian illuminants can be seen in Fig. 2.5 along with their correlated colour temperature. Planck's formula of black-body radiation is:

$$E(\lambda, T) = 2\pi hc^2\lambda^{-5}\left(e^{-\frac{hc}{k\lambda T}} - 1\right) \tag{2.25}$$

Where λ is the wavelength, T the correlated temperature in Kelvin, h is Planck constant, c the speed of light and k is Boltzmann constant.

For typical lights, one can specify illuminants SPDs using Wien's approximation of Planck's law:

$$E(\lambda, T) = Ik_1\lambda^{-5}e^{-\frac{k_2}{T\lambda}} \tag{2.26}$$

where k_1 and k_2 are constants, I is the overall intensity of the light and T is the temperature of the black-body.

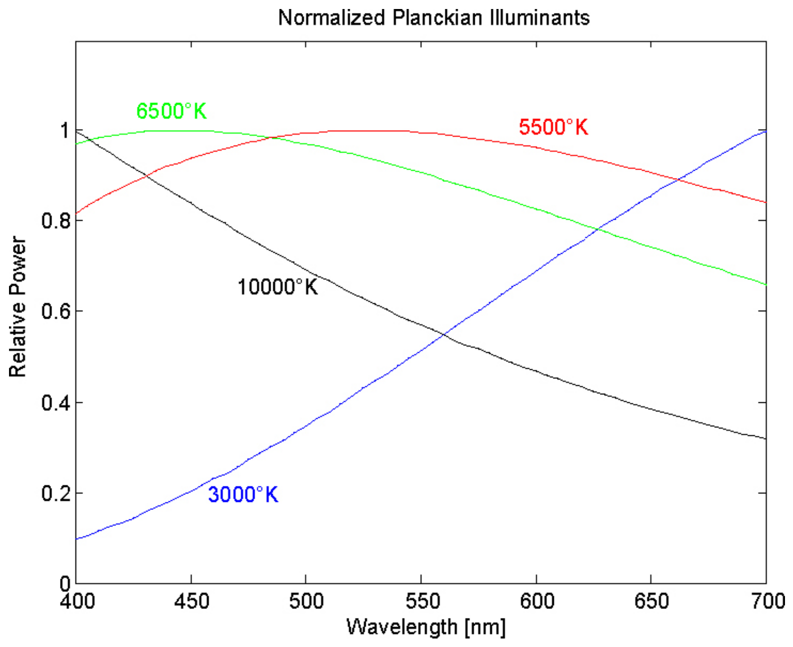


Figure 2.5: *Normalized Spectral Power Densities of some Planckian Illuminants*

If we allow all our illuminants to be modelled as such, we can substitute equation (2.26) in (2.1). This gives:

$$\rho_k = \int_\omega I k_1 \lambda^{-5} e^{-\frac{k_2}{T\lambda}} S(\lambda) Q_k(\lambda) d\lambda \tag{2.27}$$

We now assume that the camera sensors behave like Dirac delta functions, i.e., they have a non-null response at a single wavelength, λ_k (see Fig. 2.6 for an illustration), and can then rewrite the camera sensitivities as:

$$Q_k(\lambda) = q_k \delta(\lambda - \lambda_k) \tag{2.28}$$

where λ_k is the *only* wavelength at which Q_k has a non-null response.

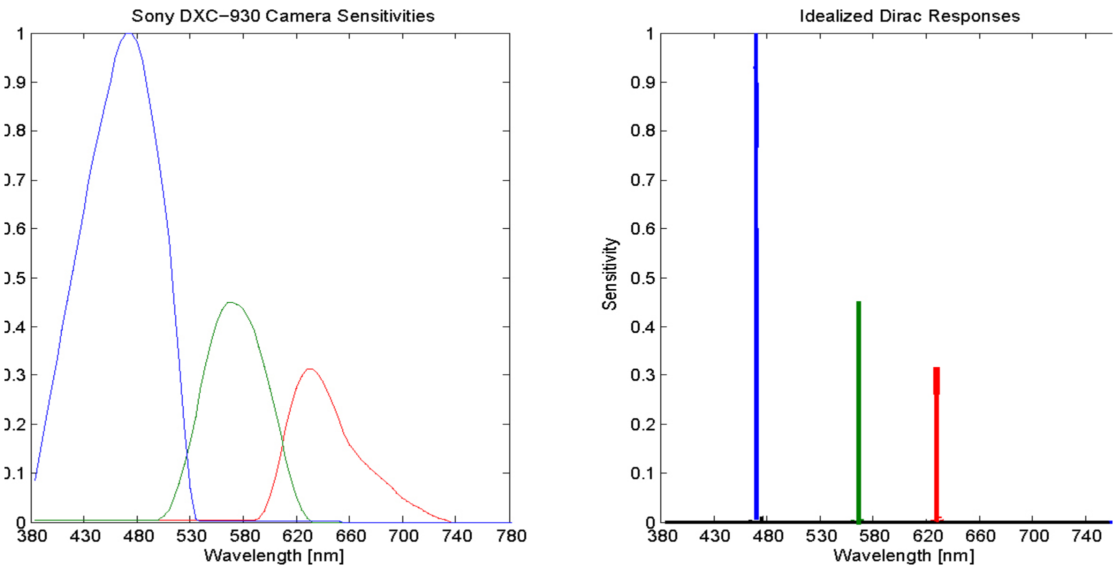


Figure 2.6: *Sony DXC-930 Camera sensors (Left); Idealized Dirac sensors (right)*

Using (2.28) in (2.27), we obtain:

$$\rho_k = I k_1 \lambda^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k \tag{2.29}$$

Let us transform the RGB sensor responses into a 2D chromaticity vector $\underline{c}$, with $c_1 = \frac{R}{G}$ and $c_2 = \frac{B}{G}$.

Then, by substituting (2.29) in the chromaticities formation, we have:

$$c_{1,2} = \frac{\lambda^{-5} e^{-\frac{k_2}{T\lambda_{R,B}}} S(\lambda_{R,B}) q_{R,B}}{\lambda^{-5} e^{-\frac{k_2}{T\lambda_G}} S(\lambda_G) q_G} \tag{2.30}$$

where we observe that the c_i do not depend on intensity and shading information. We now take the logarithms of (2.30) to obtain:

$$\log(c_{1,2}) = \log(\frac{S(\lambda_{R,B})}{S(\lambda_G)}) + \log(\frac{\lambda_{R,B}^{-5} q_{R,B}}{\lambda_G^{-5} q_G}) + k_2(\frac{1}{\lambda_G} - \frac{1}{\lambda_{R,B}})\frac{1}{T} \tag{2.31}$$

We see from this equation that using log chromaticities can provide us with images that are independent of illumination when the Plankian and Dirac assumptions are used. Equation (2.31) effectively shows that, for the same surface under various Planckian illuminants, its chromaticities fall on a line where, in log-chromaticity space, movement due to illumination is made along the slope $(\frac{1}{\lambda_j} - \frac{1}{\lambda_k})$. By projecting those log-chromaticity values on a direction perpendicular to the illumination slope gives a 1D image that depends on reflectances only. An illustration of the invariant image formation process is shown in Fig. 2.7.

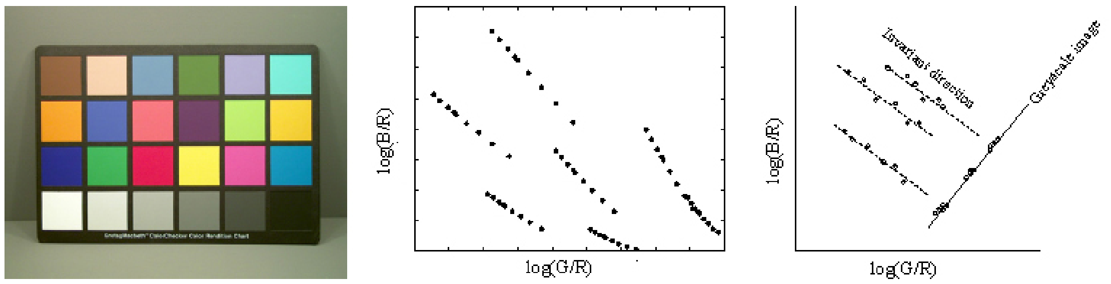


Figure 2.7: *Left: a Macbeth colour chart, middle: Log-chromaticities of 6 patches under 14 different Planckian lights; the variation due to the illumination lies on a line. Right: a schematic projection of the chromaticities along the invariant direction that results in an invariant 1D image.*

We note that, to find the invariant direction (the projecting direction), a camera calibration step is required. The direction is found by calculating the chromaticities of known surfaces across different lights e.g., by imaging a reference chart in a light booth.

While the result of the invariant image process, shown in Fig. 2.8, is indeed shadow-free and can be used "as is" in some applications [FH01], do the assumptions of Dirac sensors and Planckian lights hold for real images?

While Planck's law models incandescent and natural lights well, fluorescent lights are generally characterized by highly localized emission spikes and are therefore not accurately described by their correlated colour temperature. It has however been shown in [MO01] and [Lu06] that most typical illuminants -including daylights and artificial lights- fall very close to the planckian locus, i.e., the Planckian assumption leads to an approximation of the light but that approximation is still good enough to obtain invariant

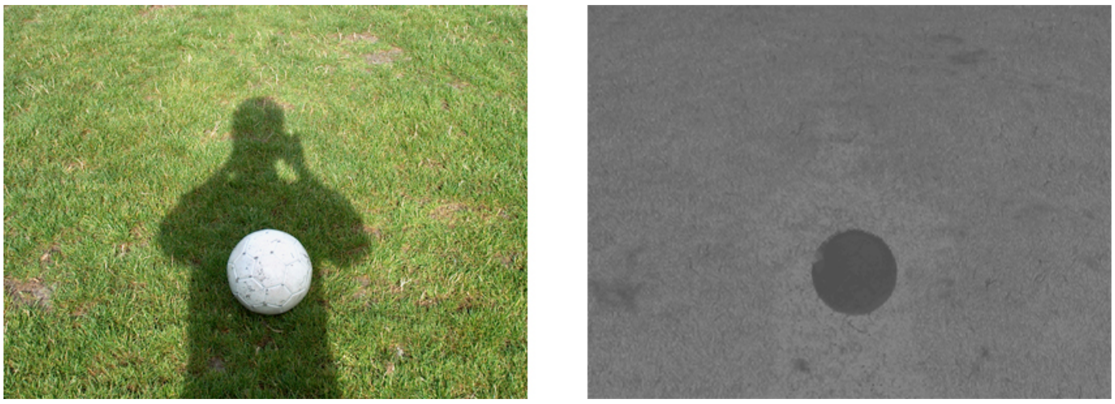


Figure 2.8: *Original image and the 1D invariant image obtained after projection.*

images.

Regarding the use of Dirac-type sensors: as shown in Fig. 2.6, actual camera sensors are not delta functions. However, a comparison of various camera sensors done in [Lu06] showed that, in practice, most sensors are narrow-band enough so that the log-chromaticities will form straight lines. If a given camera has really broad sensors, they can be sharpened -made to behave as if they were narrow band- using the method presented in [FDF94].

Entropy minimization

To remove the need for calibration, and thus permit the obtention of invariant images using an unknown camera, an extension to the invariant image framework has been proposed in [FDL04]. The key insight is that if, in an ideal case, chromaticities of reflectances under different illuminants are represented by straight lines, then projecting them on the correct direction -the perpendicular to the invariant lines- will minimize the entropy of the 1D projection. To this effect, one can form the 2D log chromaticities of the image and then project this data along all possible projection angles. The direction that minimizes the entropy of the 1D projection is then considered to be the correct one (Fig. 2.9 illustrates this idea).

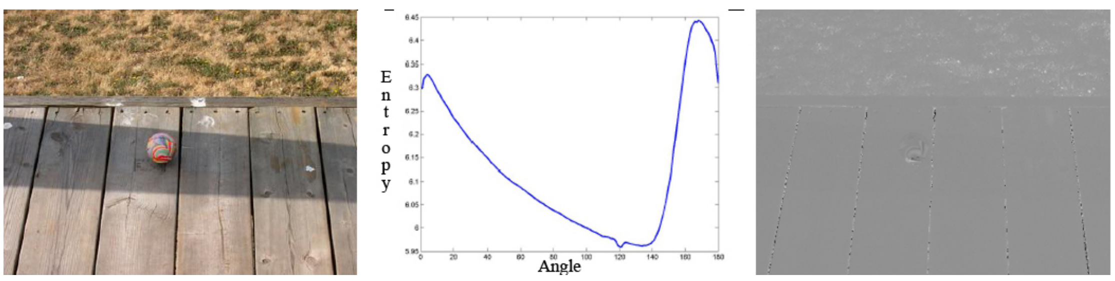


Figure 2.9: *the original image; the plot representing the entropy of the 1D projection at every angle; the 1D invariant image resulting from the projection in the min-entropy direction*

It has been shown in [FDL04] that this simple method was adequate to recover good invariant images without requiring camera calibration. This work was extended in [Lu06] where quadratic entropy [Ren87] is used and the Fast Gauss Transform applied to speed up the computations.

2D Invariant Images

In [DFH03], Drew et al. proposed that the 1D, grayscale, invariant image could be coloured using the original 2D chromaticities of the image.

Let I_{inv} be the 1D invariant image; a 2D colour illumination invariant image I_{inv2} can be obtained by taking into account both the projection direction and its perpendicular, thereby allowing a 2D representation.

This 2D representation is however "flat" since all the lighting information from the image has been removed. One cannot add lighting back on a per-pixel basis, as it would amount to undoing the invariant properties. A simple scheme proposed in [DFH03] is to use the brightest 1% of the pixels in the image (which are assumed to be non-shadow pixels) and use this offset to match the correct chromaticities of these brightest pixels with the estimated, illumination free, 2D chromaticities. Figure 2.10 shows the resulting images of this algorithm. As one can see, they are more stable -less noisy- than their 1D counterparts.

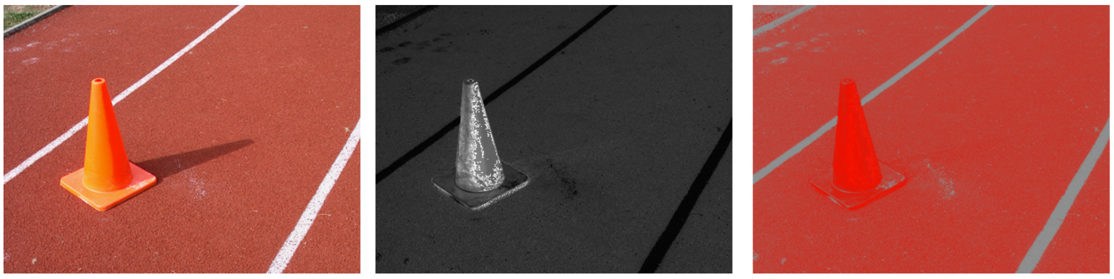


Figure 2.10: *Original image, 1D invariant image, 2D chromaticity invariant image*

From invariant images to shadow edges

The 2D chromaticity invariant image can be compared with the original image to locate the changes in illumination. The idea, proposed in [FDL04], is in essence simple. The original image contains edges that are due to both reflectance and illumination transitions. The invariant image however contains only information relevant to reflectance changes. It follows that by comparing those two edge maps one should be able to recover information pertaining to illumination changes only.

The shadow edge detection algorithm proceeds by first segmenting both the original and 2D invariant image with the Mean-Shift algorithm [CM02] using a small kernel: in effect smoothing the image. This will result in suppressing features such as noise and high frequency textures so that fewer spurious edges are detected. Edge maps are then obtained by using the Canny edge detector [Can86] on both mean-shifted images. A comparison of those edges yields an edge map that contains only illumination-related edges. Let I be the mean-shifted image, I_{inv2} be the mean-shifted 2D chromaticity invariant image and $C_{x,y}$ be the Canny operator along the x or y direction. The images gradients can then be written as:

$$\|\nabla_{x,y} I_{\mathrm{inv2}}(x,y)\| = \max_i (C_{x,y}[I_{\mathrm{inv2_i}}(x,y)]) \tag{2.32}$$

$$\|\nabla_{x,y} I_k(x,y)\| = C_{x,y}[I_k(x,y)] \tag{2.33}$$

With $i = \{1, 2\}$: the 2D chromaticities and $k = \{1, 2, 3\}$: the 3D RGB values.

Let $q_s(x, y)$ be a Boolean variable: taking the value 1 if the pixel located at (x,y) belongs to a shadow edge and 0 otherwise. An edge is classified as a shadow edge if its magnitude is high in the original image and low in the invariant one. A second criterion is the orientation; if both the original image and the invariant one have a strong edge magnitude at a given location, but differ in their orientations, the edge will be deemed to be a shadow edge. The edge detection is therefore defined as:

$$q_s(x, y) = \begin{cases} 1 & \text{if } \|\nabla_{I_i}\| > \tau_1 \text{ and } \|\nabla_{I_{\text{inv2}}}\| < \tau_2 & (2.34a) \\ 1 & \text{if } |\frac{\|\nabla_x I_i\|}{\|\nabla_y I_i\|} - \frac{\|\nabla_x I_{\text{inv2}}\|}{\|\nabla_y I_{\text{inv2}}\|}| > \tau_3 & (2.34b) \\ 0 & \text{otherwise} & (2.34c) \end{cases}$$

This process is illustrated in Fig. 2.11. One can notice that despite the simplicity of the algorithm, it performs well enough to detect most shadow edges. However, one can also observe that the shadow edges are approximate and incomplete, which leads to problems when attempting to remove the shadows.

2.3 Removing Shadows

When more than one illuminant is present in a scene, one often needs to remove the unwanted illumination in order to use conventional vision techniques. The problem of shadows is perhaps the most studied because they are the major sources of unwanted illumination changes, and, the stark contrast they provide within an image makes them all the more problematic.

Shadow removal in itself has been significantly less studied than shadow detection. The main reason is that all video-based methods effectively have "built-in" shadow removal. In essence, once a pixel is deemed to belong to a shadow, it can be replaced by its non-shadow version obtained in another frame. In the field of computer graphics, how-

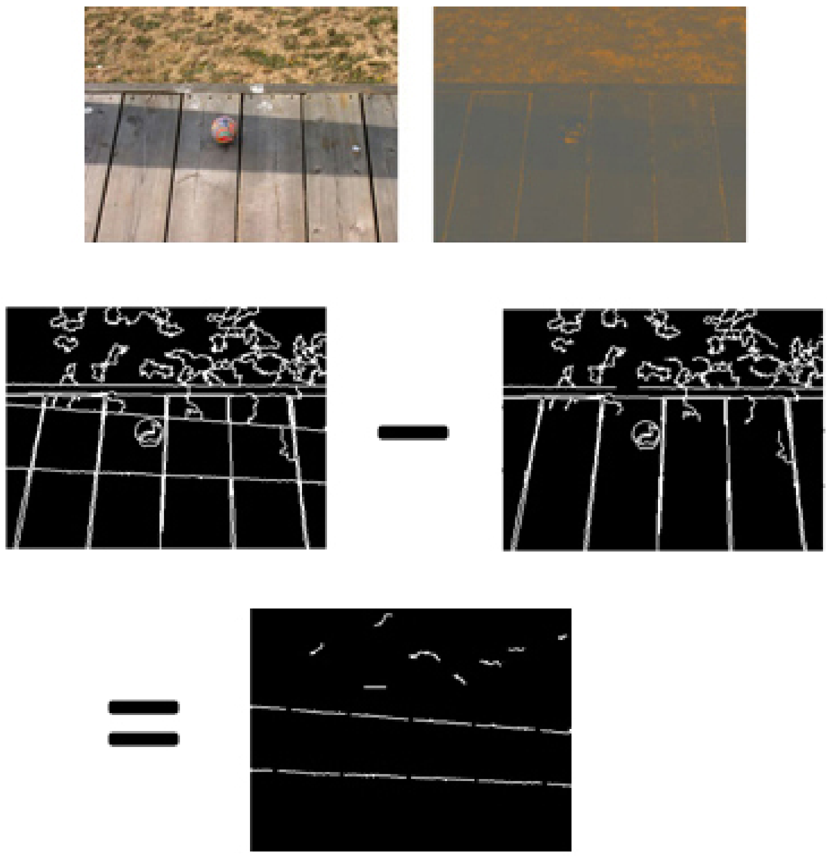


Figure 2.11: *Edge detection algorithm. Top row: the original image and the 2D invariant image. Middle row: the detected edges of both the original (left) and invariant (right) images. Bottom row: the edges deemed to be shadow edges*

ever, shadow manipulation -including detection, removal, synthesis and compositing- has been of interest for some time. In this context, shadow free images can be obtained by, for example, Reinhard et al. method for transferring colour and lighting from a source to a target image [RAGS01]; Oh et al. removal of cast shadows on uniformly textured areas using a texture-illuminance decoupling filter [OCDD01] or Perez et al. Poisson image editing framework [PGB03] where regions from images having different lighting conditions can be copied onto the background of another image and rendered under a uniform lighting field. There are many more computer graphics inspired methods, however, most of them require user input and often necessitate several images to be used: a source image and a target image where the compositing will take place.

In this thesis, we will take as a starting point the method of Finlayson et al. [FHD02] that removes shadows using a 2-dimensional integration framework. Shadow removal pre-supposes that shadows have been detected beforehand and that a mask is available, either as shadow edges or as a labelling that marks the membership of a pixel as either shadow or non-shadow.

Shadow Removal with 2D Integration

Let I denote the log of an image. Its gradient, ∇I is

$$\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}) \tag{2.35}$$

Denote the location of the shadow edges, e.g., obtained using the method set forth in the previous section, by S. The derivatives belonging to S can then be thresholded using a function $T(\nabla I)$ such that

$$T(\nabla I) = \begin{cases} 0, & \text{if } |\nabla I| \in S \quad (2.36a) \\ \nabla I, & \text{otherwise} \quad (2.36b) \end{cases}$$

How can I be recovered from $T(\nabla I)$? This is not an easy question to answer since a gradient image is composed of two numbers per pixel but the reintegrated image only has a single number per pixel. Besides, a 2D function can be reintegrated only if the gradient field is conservative [FC88], i.e., $\frac{\partial I}{\partial x \partial y} = \frac{\partial I}{\partial y \partial x}$ for all points in the image (a condition that, in essence, tells us that the 2 derivatives do not provide independent information). Thresholding the edges implies that this condition is usually not met and one therefore has to reintegrate in a least squares sense [Sta79], [FC88]. Effectively, this results in solving a Poisson equation of the form

$$\nabla^2 I = \mathrm{div}(T(\nabla I)) \tag{2.37}$$

Where ∇^2 is the Laplacian operator $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ and $\text{div}(T(\nabla I)) = \frac{\partial(T(\nabla I))_x}{\partial x} + \frac{\partial(T(\nabla I))_y}{\partial y}$

Provided that the gradient field is integrable, Blake showed in [Bla85] that the log of lightness can be recovered from the log of the intensity by inverting the Laplacian of the image if correct boundary conditions are used. In practice, the conditions one assumes are either Dirichelet: the boundary of the image is known, Neumann: the *derivatives* at the image boundary are known constraints or homogeneous Neumann: the directional derivatives at the boundary are known to be zero. Homogeneous Neumann conditions are used in [FDL04], for they were shown to yield higher quality images than either Dirichelet or Neumann.

Subject to these constraints we can invert the Poisson equation in (2.37) using standard techniques (e.g., by using Fourier or Multigrid [HE81] methods). The derivatives of the reintegrated image found using this method are as close as possible to the thresholded derivatives of the original image in a least square sense. While minimized, the derivatives errors can however be spread over the entire image, resulting in global artifacts.

Other 2D methods for enforcing integrability have recently been introduced and have been shown to perform better than simply solving a Poisson equation. Petrovic et al. [PCF$^+$01] use belief propagation in graphical networks to enforce integrability in photometric stereo and shape from shading problems. While their method results in fewer artifacts than Poisson solving, integration errors are still, necessarily, propagated throughout the image. A different approach to integrability has been developed by Agrawal et al. in [ACR05] where they exploit the information contained in the curl -the non integrable part of the gradient field- in order to minimize integration errors. Their method, however, requires boundary edges to be all of the same type -shadow or non shadow in our case-, a condition not necessarily met. Their method is also deterministic: if the integration incurs errors, they cannot be prevented.

Figure 2.12 illustrates the shadow-free images that are obtained with the Poisson reintegration method. The shadow boundaries themselves are inpainted using a simple diffusion-based process. This is necessary for better visual results, as image derivatives at the shadow boundary are set to 0 they would look constant and would stand out in the reintegrated image.

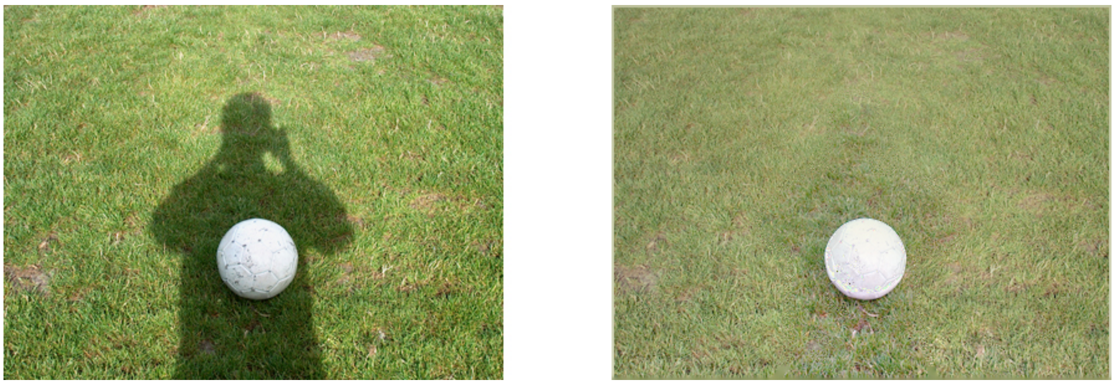


Figure 2.12: *2-D Shadow removal. The original image (left), and the shadow free image resulting from Poisson integration (right).*

A problem to note, that we will review in detail in Chapter 5, is that a hidden assumption is made in this, and other 2D integration algorithms. Since shadow edges are thresholded, reintegrating the whole image will, independently of the way integrability is enforced, result in assuming that there are no coincident reflectance/illumination changes. Such occurrences are, however, not uncommon and will significantly decrease the quality of the recovered shadow-free images.

Chapter 3

Chromagenic Illuminant Estimation

In this chapter we look at a new algorithm for illuminant estimation. We begin by reviewing the concept of chromagenic colour constancy, first introduced by Finlayson et al. in [FM05] and [FHM05b] and look at parameters known to affect its performance [FHM05a].

We show that the basic formulation of the chromagenic algorithm has inherent weaknesses: a need for perfectly registered images and occasional large errors in illuminant estimation. Our first contribution is to analyze the algorithm performance with respect to the reflectances present in a scene and demonstrate that fairly bright and desaturated reflectances (e.g., achromatic and pastel colours) provide significantly better illuminant estimation. Put in another way: if the scene contains bright or achromatic colours we do not expect large errors.

This analysis leads to the bright-chromagenic algorithm. We show that it not only remedies the large error problem but that we can also remove the image registration constraint. Experiments performed on a variety of synthetic and real data show that the newly designed bright-chromagenic algorithm significantly -in a strict statistical sense- outperforms current illuminant estimation methods, including those having a substantially higher complexity.

3.1 The Chromagenic Algorithm

Chromagenic colour constancy is performed using two images of the same scene: a normal image and one where a coloured filter is placed in front of the camera. The relationship between these two images is then used to estimate the scene illuminant.

The idea of using coloured filters to improve vision tasks is not new. In optometry, chromagen lenses are used to subjectively improve the quality of colour vision for colour-blind observers [Hod98] and special coloured filters are also used to improve the reading speed of some dyslexic patients [Wil03]. The idea of using two images of a scene to facilitate some tasks is also common in computer vision. In stereo, two pictures of a scene are taken in order to recover the three dimensional position of objects in the scene. In photometric vision, polarizing filters can be used to remove specular highlights [LL97]. In the case of colour constancy, pairs of images taken with and without flash can be used to estimate the original scene illuminant [DXW01, PSA$^+$04, LD06].

The chromagenic illuminant estimation algorithm proceeds as follows: Let $S(\lambda)$ be the descriptor of surface reflectances, $E(\lambda)$ the scene illuminant SPD, $Q_k(\lambda)$ the camera sensitivities (we consider here trichromatic cameras, so $k = \{R, G, B\}$) and $F(\lambda)$ be the transmittance of the colour filter placed in front of the camera.

The sensor responses of the unfiltered, $\underline{\rho}$, and filtered, $\underline{\rho}^F$, image can be written as:

$$\rho_k = \int_\omega E(\lambda)S(\lambda)Q_k(\lambda)d\lambda \tag{3.1}$$

$$\rho_k^F = \int_\omega E(\lambda)S(\lambda)F(\lambda)Q_k(\lambda)d\lambda \tag{3.2}$$

thus, for each scene we recover six responses per pixel that form the input to the illuminant estimation problem. For the purposes of this work, we set out to recover $\underline{\rho}_E$: the RGB of a white surface under the scene illuminant E.

Let us first consider the equations of filtered and unfiltered image formation (3.1) and (3.2). We can approximate the filtered image by posing a second illuminant, $E^F(\lambda)$

so that it is equivalent to putting the filter $F(\lambda)$ in front of the light source $E(\lambda)$, i.e., $E^F(\lambda) = F(\lambda)E(\lambda)$. We can therefore think of $\underline{\rho}$ and $\underline{\rho}^F$ as sensor responses of a single surface under two different illuminants. It has been shown in [MW86] and [DI93] that when the same surfaces are viewed under two lights, the corresponding RGBs can, to a good approximation, be related by a linear transform and so we use a 3×3 matrix to relate the RGBs captured with and without the coloured filter. We can then write:

$$\underline{\rho}^F = T_E^F \underline{\rho} \tag{3.3}$$

where T_E^F is a 3×3 linear transform that depends on both the chromagenic filter and the scene illuminant. Equation (3.3) implies that, given the chromagenic filter and sensor responses under a known illuminant, we can predict the filtered responses. In the problem of illuminant estimation, however, we know both the filtered and unfiltered responses but not the illuminant. Moreover, the task of finding the illuminant corresponds to finding T_E^F. If we know all possible illuminants *a priori* we can, for a given filter, determine the transforms T_E^F for every illuminant. We can then estimate which of these pre-computed transforms best fits the pair of filtered-unfiltered responses and thus, determine the illuminant.

Before outlining the actual algorithm, it is worth pointing out two cases where, depending on the filter or the sensor sensitivities, chromagenic colour constancy is not possible: if the filter has a neutral density or if the camera sensors behave like Dirac delta functions.

If the chosen filter has a neutral density, i.e., its transmittance does not vary across the spectrum, the relationship between filtered and unfiltered RGBs will be a constant scaling (the same for all lights). If we write:

$$F(\lambda) = \alpha, \ \forall\lambda \tag{3.4}$$

where α is a constant value, then

$$\underline{\rho}^F = \alpha \underline{\rho}\ ,\ \forall S, E \tag{3.5}$$

It follows that the 6D responses will in fact span only three dimensions and thus solving for colour constancy is impossible.

If we suppose Dirac-type sensors where the non-null response of the k^{th} sensor is at the wavelength λ_k, we can rewrite equations (3.1) and (3.2) as:

$$\rho_k = E(\lambda_k)S(\lambda_k)Q_k(\lambda_k) \tag{3.6}$$

$$\rho_k^F = E(\lambda_k)S(\lambda_k)F(\lambda_k)Q_k(\lambda_k) \tag{3.7}$$

It follows that $\rho_k^F = F(\lambda_k)\rho_k$ and that the responses are, again, three dimensional and their relation depends neither on the reflectances nor on the scene illuminant. Additionally, while not as limiting as the neutral density case, using a rank-deficient filter -e.g., a pure red filter- will deliver poor constancy since significant information is lost -in this case the relationship between the blue pixels.

Barring the cases outlined above, the transforms can be pre-computed by choosing a set of n typical scene illuminants: $E_i(\lambda),\ i = 1, \ldots, n$ and a set of m surface reflectances: $S_j(\lambda),\ j = 1, \ldots, m$ representative of the real world. For each illuminant i, we create a $3 \times m$ matrix $\mathcal{Q}_i$ whose j^{th} column contains the sensor response of the j^{th} surface illuminated by the i^{th} illuminant. We also create $\mathcal{Q}_i^F$, which contains the equivalent filtered responses. For each illuminant, we can then define the transform matrix as:

$$\mathcal{T}_i = \mathcal{Q}_i^F \mathcal{Q}_i^+ \tag{3.8}$$

where + denotes the Moore-Penrose pseudo-inverse operator, $\mathcal{Q}^+ = (\mathcal{Q}^{\mathrm{T}}\mathcal{Q})^{-1}\mathcal{Q}^{\mathrm{T}}$. $\mathcal{T}_i$ can then be described as the transform that best maps, in a least square sense, unfiltered to filtered responses under illuminant i. Because it is a least squares fit, $\mathcal{T}_i$ will not, in practice, map the responses without errors. What matters, however, is that the error committed when mapping responses under illuminant i is the smallest when the corresponding transform $\mathcal{T}_i$ is used.

Once the n transforms have been pre-computed, the illuminant estimation proceeds as follows: let $\mathcal{Q}$ and $\mathcal{Q}^F$ denote the $3 \times m$ matrices of unfiltered and filtered RGBs of arbitrary reflectances under an unknown light. For each plausible illuminant we calculate the fitting error, e_i, as:

$$e_i = \|\mathcal{T}_i \mathcal{Q} - \mathcal{Q}^F\|, \quad i = 1, \ldots, n \tag{3.9}$$

under the assumption that $E_i(\lambda)$ is the actual scene illuminant. We then choose the transform that minimizes the error and surmise that it corresponds to the scene illuminant. Our estimated illuminant is $E_{\text{est}}(\lambda)$ where

$$\text{est} = \arg\min_i (e_i) \quad i = 1, \cdots, n \tag{3.10}$$

It was shown in [FHM05b] that if both reflectances and illuminants can be *exactly* described by three basis functions each, i.e., they are three dimensional, then the chromagenic algorithm delivers perfect illuminant estimation. In natural scenes, however, these dimensions are generally higher [PJ89,JMW64] and so there are estimation errors.

Indeed, while in general the chromagenic algorithm can deliver good colour constancy, it has two major weaknesses: the first one is that, though good on *average*, the performance can, on occasion, be (very) poor. The second problem comes from equation (3.9) where we see that the fitting error for each candidate illuminant is evaluated on a per-pixel basis. This implies that, for the algorithm to deliver an optimal performance, the two images have to be perfectly registered, a demanding requirement when images are taken outside of the lab. Registration methods can be of some help but since we are looking for an exact registration at pixel-level, they may not be sufficient.

3.2 The Choice of Filter and Sensors

Two important aspects of the chromagenic algorithm are the filter choice and the camera sensors, for they can greatly influence the performance of the algorithm and are the only "controllable" parameters in the image formation process.

Looking at equation (3.9), we see that in order to maximize the performance of the chromagenic algorithms, two properties have to be satisfied:

1. The transforms $\mathcal{T}_i$ should be as different as possible from each other (maximizing the discriminative power of the illuminant).

2. A transform $\mathcal{T}_i$ should map unfiltered responses under illuminant i to filtered responses with as small an error as possible (minimizing the uncertainty).

3.2.1 Optimal Filters

In general, given a filter set -in our case the filter set is a selection of 53 different Kodak Wratten filters [Kod69]- one wants, for chromagenic purposes, to select one that is non-cutoff and whose transmittance varies across the visible spectrum. Among the then possible filters, it was shown in [FHM05b] that, in practice, they all deliver a good level of performance. Finlayson et al. in [FHM05a] have however reported that designing a filter specifically for chromagenic processing gave significantly better results than simply picking an existing Wratten filter.

The filter design can be analyzed either in closed-form or by sampling. The closed-form method formulates the two desirable properties for $\mathcal{T}_i$ in an optimization problem. Specifically, this is done by maximizing the inter-transform variance (property 1) while minimizing the fitting error (property 2). They have shown that the optimal filters are solution to an eigenvector problem. The closed-form approach however does not guarantee to minimize illuminant estimation error.

Optimal filters can otherwise be obtained by sampling the space of possible filters. The performance of the filters is defined to be in relation to the average illuminant estimation error obtained on a representative set of images. The average has to be used because the performance of a given filter will be image and illuminant dependent (e.g., if both the surfaces and the illuminant are red, a reddish filter will yield better results than a blueish one). Let us assume that the filter transmittance $F(\lambda)$ is a linear combination of three basis functions that can be found by performing a principal component analysis on the transmittance of all Wratten filters. To find all filters one then needs to investigate all possible weight combinations for the filter basis: a computationally intensive problem. In practice though, the exhaustive search for the weights can be restricted to a finite set of uniformly sampled points on the unit sphere (5000 such points were used in [FHM05a]).

3.2.2 Optimal Sensors

Arguably, one might also design sensors for chromagenic illuminant estimation. But, pragmatically, when choosing sensors one has to consider aspects of image quality such as colour rendering and image noise that strongly depend on the sensors. So, though optimal sensors for chromagenic illuminant estimation were discussed in [FHM05a], we will work with conventional camera sensors.

3.3 The Bright-Chromagenic Algorithm

We saw in the previous section that the average performance of the chromagenic algorithm can be enhanced by choosing specific filters and sensor sensitivities. These improvements do not, however, address the main limitations of the chromagenic algorithm: potential large estimation errors and the need for perfectly registered images.

We propose here a modification of the chromagenic algorithm that has three outcomes: it improves the average illuminant estimation performance, it reduces the maximal errors observed when the estimation is erroneous and, more importantly, it allows the algorithm to be used on unregistered images.

Let us first look back at equations (3.1) and (3.2) where we see that the sensor responses depend on the scene illuminant, the chromagenic filter, the sensor sensitivities and the surface reflectances. It follows that the linear transforms $\mathcal{T}_i$ also depend -to some degree- on those factors. Among them, the illuminant is what we aim to recover so it has to remain a variable; achievable improvements due to choosing both the filter and the sensors sensitivities have been explored in [FHM05a]. We therefore set out to explore the only unknown that remain in the equation: the scene reflectances.

Building a model based on reflectances can be difficult for a couple of reasons. The main problem is that we have, in general, no control as to which reflectances are present in a scene. This uncertainty is the reason why simple estimation methods such as gray-world and Max-RGB are unreliable (if every scene contained a white patch, Max-RGB would be very accurate). Another issue is linked to what is the input of most illuminant estimation algorithms: RGBs. Indeed, both the training and testing steps of the chromagenic algorithm are RGBs, which are composed of all of the image formation process parameters.

3.3.1 Reflectances Analysis

For the chromagenic algorithm to work well, the transforms $\mathcal{T}_i$ that map RGBs to their filtered counterparts should depend strongly on the illuminant. Here, we want to quantify the variance of the transforms when the illuminant changes and compare it to the variance observed when the illuminant is fixed but the reflectances vary.

To perform this assessment, we follow the methodology of [FHM05a] and [FHM05b] in our choice of parameters. The illuminants belong to a set of 87 measured illuminants Spectral Power Distributions (SPD) that include all common light sources. These SPDs are sampled every 10nm, from 380nm to 780nm. More details about the set can be found here [BCF02] and the set itself can be found at [Bar02]. For surface reflectances, we use a synthesized set of 1995 Munsell surface reflectances [Uni89]. The reflectances are also sampled every 10nm from 380nm to 780nm, more details about that set can be found in [PJ89].

Concerning the choice of camera sensitivities and filter, we could use the results of [FHM05a] to maximize the performance but since they (for now) just exist on paper and that we want to have a framework as unified as possible over all our experiments, we decide to use the sensors of a Sony DXC-930 camera (as in [BCF02] and [HF06]) and a Wratten 81B filter (a yellowish filter). Both the filter and the sensor sensitivities are shown in Fig. 3.1.

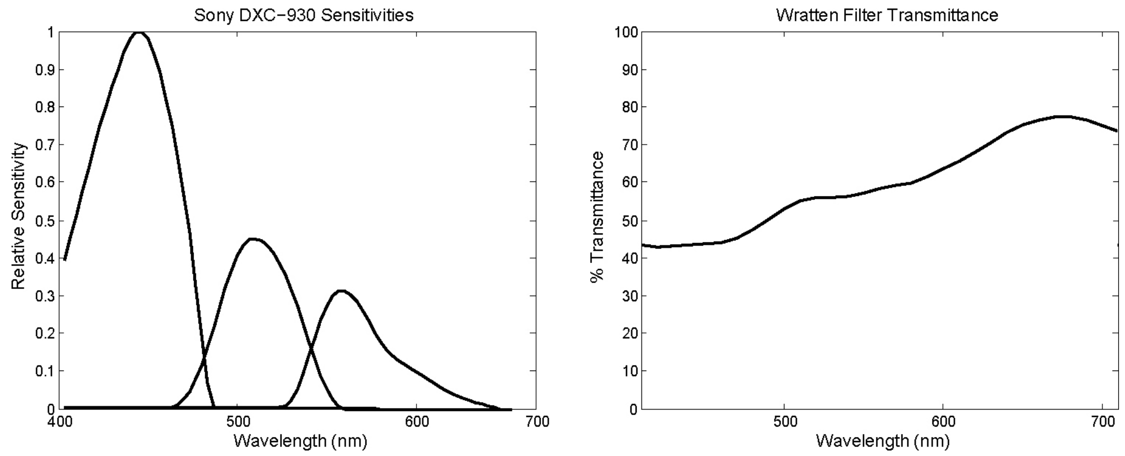


Figure 3.1: *Left: The Sony DXC-930 sensitivities. Right: The transmittance of the 81B Wratten filter used in the experiments.*

We start by creating the transforms $\mathcal{T}_i$ according to equation (3.8) by imaging all the 1995 reflectances under the 87 illuminants, we thus have 87 distinct transforms. To assess the variability of the $\mathcal{T}_i$, i.e., how differently they map the reflectances depending on the illuminant, we calculate the inter-transform variance σ_E^2:

$$\sigma_E^2 = \frac{1}{87}\sum_{i=1}^{87}(\underline{t}_i - \underline{\mu}_T)^{\mathrm{T}}(\underline{t}_i - \underline{\mu}_T) \tag{3.11}$$

where $\underline{t}_i$ is the 9×1 vector representation of the 3×3 transform $\mathcal{T}_i$ and $\underline{\mu}_T$ is the average of all $\underline{t}_i$. Equation (3.11) quantifies the transforms variation across illuminants: the larger the variance, the better the algorithm will perform. However, we must compare (3.11) with the variation in transforms due to choosing different reflectance sets.

Calculating the equivalent variance generated by different reflectances is somewhat more complicated. Let us denote by S the entire reflectance set; we randomly partition S into N subsets of equal size, denoted s_j such that:

$$\bigcup_{j=1}^{N} s_j = \mathbf{S} \tag{3.12}$$

and

$$s_j \cap s_k = \varnothing,\ \forall j,k \in [1,N],\ j \neq k \tag{3.13}$$

We denote by $\mathcal{T}_i^{s_j}$ the transform obtained with equation (3.8) when the subset s_j is imaged under illuminant i. The inter-transform variance for reflectances σ_S^2 is calculated as:

$$\sigma_S^2 = \frac{1}{87}\sum_{i=1}^{87}\sigma_{S_i}^2 \tag{3.14}$$

where:

$$\sigma_{S_i}^2 = \frac{1}{N}\sum_{j=1}^{N}(\underline{t}_i^{s_j} - \underline{\mu}_T^{S_i})^{\mathrm{T}}(\underline{t}_i^{s_j} - \underline{\mu}_T^{S_i}) \tag{3.15}$$

In this formulation, $\underline{\mu}_T^{S_i}$ is the mean of all $\underline{t}_i^{s_j}$, i.e., the mean of all subset-induced transforms under illuminant i.

An important aspect of this calculation is how to partition S. On one hand, we want enough subsets for the test to be meaningful, but we also know that the chromagenic algorithm is three dimensional. While most daylight illuminants are at least 3D, this is not necessarily the case with reflectances, so, a combination of reflectances is required. To balance these imperatives, we choose to use subsets of 16 reflectances -a plausible number for many real scenes. This, however, yields C_{16}^{1995} possibilities to select our subsets, a daunting number. Instead of seeking every possible partition, we partition the 1995 reflectances in subsets of 16 reflectances 1000 different times, thus averaging the results over 1000 observations of σ_S^2.

The results of this experiment are: $\sigma_E^2 = 0.0306$ and $\sigma_S^2 = 0.0753$. The variation of σ_S^2 with respect to the number of reflectances used is shown in Fig. 3.2.

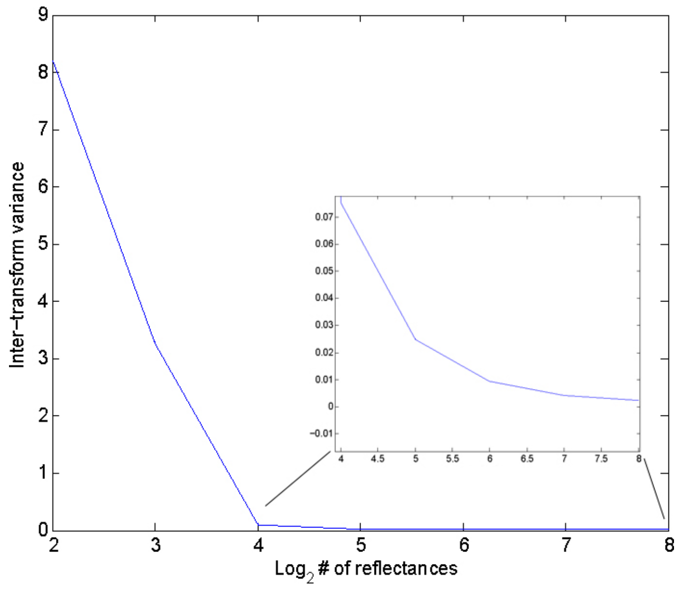


Figure 3.2: *Inter-transform variance using a synthetic test with multiple reflectances from the Munsell set under a given illuminant. The values shown are the average over 1000 tests.*

Based on these results we can conclude that the linear transforms used in the chromagenic algorithm vary significantly with the reflectances used in training. It follows that there is also a significant variability in testing. Thus, a subset of reflectances that are better suited to illuminant estimation should also exist, and the performance will increase the more of these reflectances are present in the scene.

3.3.2 Modelling In and Outliers

We know that the estimation accuracy will partly depend on the set of tested reflectances and we would like to model that dependency. Because there is a multitude of combinations of illuminants and reflectances, we will analyze the performance of the algorithm on single reflectance scenes (where good estimates can still be obtained [FHM05b]).

The transforms are calculated as before, creating 87 of them. The test set for this experiment consists of all possible single reflectance scenes under 287 illuminants, therefore creating $\sim$570,000 pairs of filtered and unfiltered RGBs. This larger illuminant set used in testing covers the same gamut as the 87 training lights; the chromagenic algorithm will select one of the 87 lights as the scene illuminant.

Figure 3.3 shows the angular errors for all the 570,000 scenes. The angular errors range from 0 to 42 degrees, with a mean of 9.3 and a median of 5. For this particular dataset, our experiments indicate that an angular error of 3 degrees or less is necessary for acceptable colour cast removal. From these results, we see that we need to reduce both the overall and, especially, the maximum errors -an angular error of 42 degrees is equivalent to mistaking green for yellow.

To understand what is happening, we look at the RGBs that comprise the top and bottom 20% of performance. We plot the brightness and saturation of these RGBs in Fig. 3.4a for the highest errors and Fig. 3.4b for the lowest ones. It is clear that low errors correlate with fairly desaturated RGBs (pastel tones and achromatic) whereas high errors correlate with dark and saturated RGBs. More interesting perhaps is the fact that bright achromatic RGBs are not at all present amongst the RGBs linked to high errors. This finding is corroborated by the result of Fig. 3.5 where we plot elliptical summaries for the high and low error sets. The ellipses, each of which accounts for 90% of its respective data, are mostly disjoint. Assuming a uniform distribution of colours in an image, we propose that it is easy to find RGBs and their filtered counterparts that belong to this preferred set. We simply look for a small percentage of the brightest

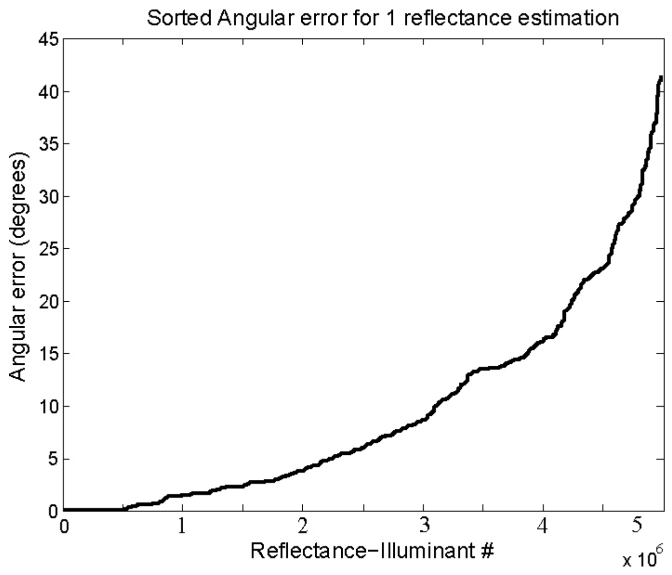


Figure 3.3: *Sorted errors for the single reflectance test. The mean error value is 9.3 degrees and the median 5 degrees.*

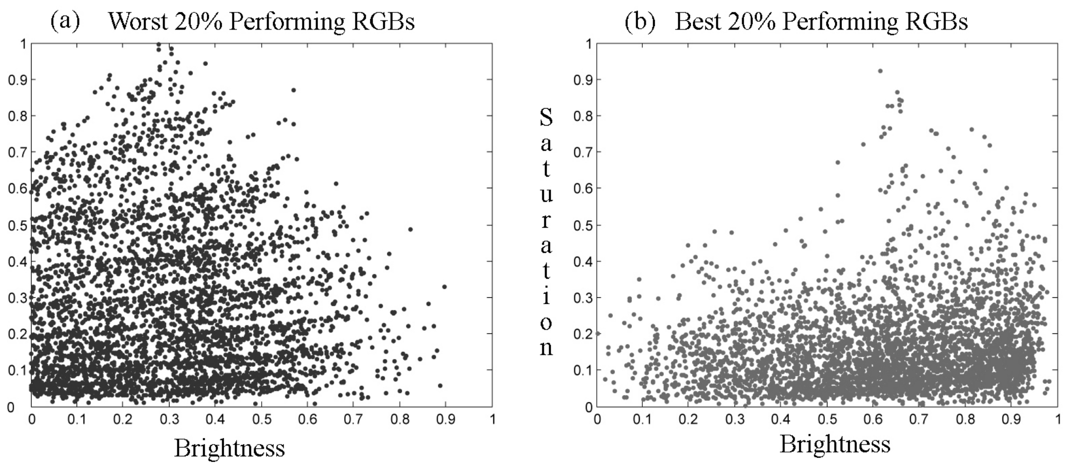


Figure 3.4: *(a) Brightness-Saturation scatter plot of the 20% worst performing RGBs. (b) Brightness-Saturation scatter plot of the 20% best performing*

image regions. We propose that the basic chromagenic algorithm should be modified so that only bright image pixels are considered.

The **Bright-Chromagenic** algorithm is defined as:

Preprocessing: For a database of m lights $E_i(\lambda)$ and n surfaces $S_j(\lambda)$ calculate $\mathcal{T}_i \approx$

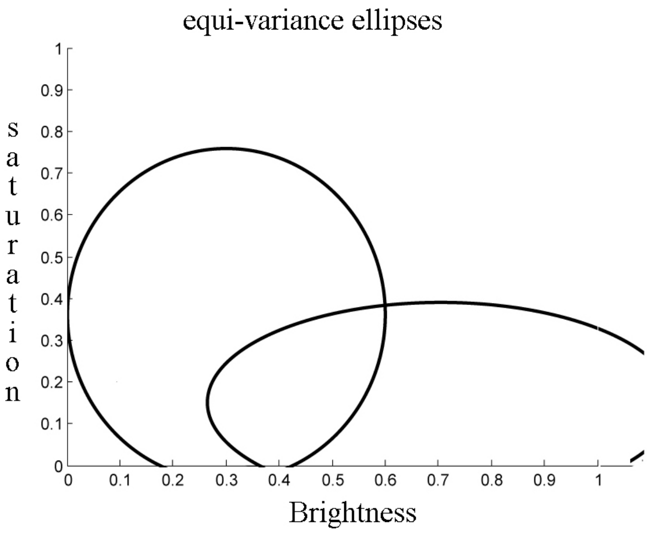


Figure 3.5: *Equi-variance ellipses of both sets, each containing 90% of their respective data, showing they are mostly disjoint.*

$\mathcal{Q}_i^F \mathcal{Q}_i^+$ where $\mathcal{Q}_i$ and $\mathcal{Q}_i^F$ represent the matrices of unfiltered and filtered sensor responses to the n surfaces under the ith light and + denotes a *pseudo-type-inverse*

Operation: Given P surfaces in an image we have $3 \times P$ matrices Q and Q^F. From these matrices we choose the $r\%$ brightest pixels giving the matrices $\mathcal{Q}$ and $\mathcal{Q}^F$, where the brightest pixels are defined to be the ones with the largest $R^2+G^2+B^2$ value. Then the estimate of the scene illuminant is $\underline{\rho}_{est}$ where

$$est = \arg\min_i(err_i) \quad (i = 1, 2, \cdots, m)$$

and

$$err_i = ||\mathcal{T}_i Q - \mathcal{Q}^F||$$

This formulation is robust since it does not make assumptions about which reflectances might or might not be present in the scene, i.e., if there are no bright reflectances in the image, the bright-chromagenic algorithm will still have an equivalent performance to the original chromagenic algorithm.

Because we "exclude" -select them only if no other are available- the worst performing RGBs, we expect the bright chromagenic algorithm to significantly reduce the worst errors.

We also make the following observation: if we assume that scenes admit a diversity of reflectances, then it follows that -if the filter does not vary too drastically across the spectrum- the brightest unfiltered RGBs will most likely be mapped on to the brightest filtered RGBs. If we are relatively conservative with the number of bright pixels we use to estimate the illuminant (we typically use the top 1-3% of the brightest pixels), the bright-chromagenic algorithm will then be able to estimate illuminants even when the images are not registered. Both these properties are verified in our experiments.

Because we are proposing to look only at bright image responses, the transform matrices might be calculated using a least-squares estimator where bright values are weighted more strongly. This is what we mean by a *pseudo-type-inverse*. However, in our subsequent experiments, we have not found any strong benefit from building transforms using only the bright image RGBs. So, for the experiments presented in the next section we use the conventional (unweighted) Moore-Penrose inverse.

3.4 Experiments

In this section, we analyze the performance of the bright-chromagenic algorithm, and compare it to various other illuminant estimation methods on four datasets of increasing difficulty, ranging from perfect synthetic data to real images taken with a prosumer digital camera whose sensitivities are unknown.

We evaluate algorithms according to the framework of Hordley and Finlayson [HF06] where it was shown that, if one wants to summarize the performance of an illuminant estimation algorithm over a dataset, one should use the median angular error instead of the mean or Root Mean Square error. The use of a median statistic also permits to assess if the difference of performance between two algorithms is statistically significant at chosen confidence level. That significance is given by using the Wilcoxon sign test [HT01] at a 95% confidence level.

To simplify the writing, we will use the following notations: S_M is the set of 1995 synthetised Munsell reflectances of [Uni89], E_{87} and E_{287} are the sets of respectively 87 and 287 illuminants from [Bar02].

3.4.1 Synthetic Reflectances and Lights

The test on synthetic images is run according to the testing protocol proposed by Barnard et al. in [BCF02]:

Training: The linear transforms are created by imaging the whole of S_M under E_{87}, thus generating 87 transforms. We use the Sony-DXC camera sensitivities and the 81B Wratten filter, whose transmittance is shown in Fig. 3.1.

Testing: We generate 1000 images containing n different reflectances, $n = \{1, 2, 4, 8, 16, 32\}$, randomly taken from S_M. We then illuminate these images with one light taken at random from E_{287}; sample filtered and unfiltered images are shown in Fig. 3.6. We estimate the illuminant of each image using both the bright-chromagenic and the original chromagenic algorithms. For images where $n > 4$, the bright-chromagenic version estimates the illuminant based on the four brightest reflectances only.

The results are displayed in Table 3.1 where the last column indicates the ranking of the considered algorithms, taking in to account the results of Wilcoxon's sign test. An algorithm is ranked better than another if its median is lower *and* if the difference is statistically significant at the 95% level. If the sign test is inconclusive, the algorithms will be ranked equally.

The results show that the bright-chromagenic algorithm performs significantly bet-

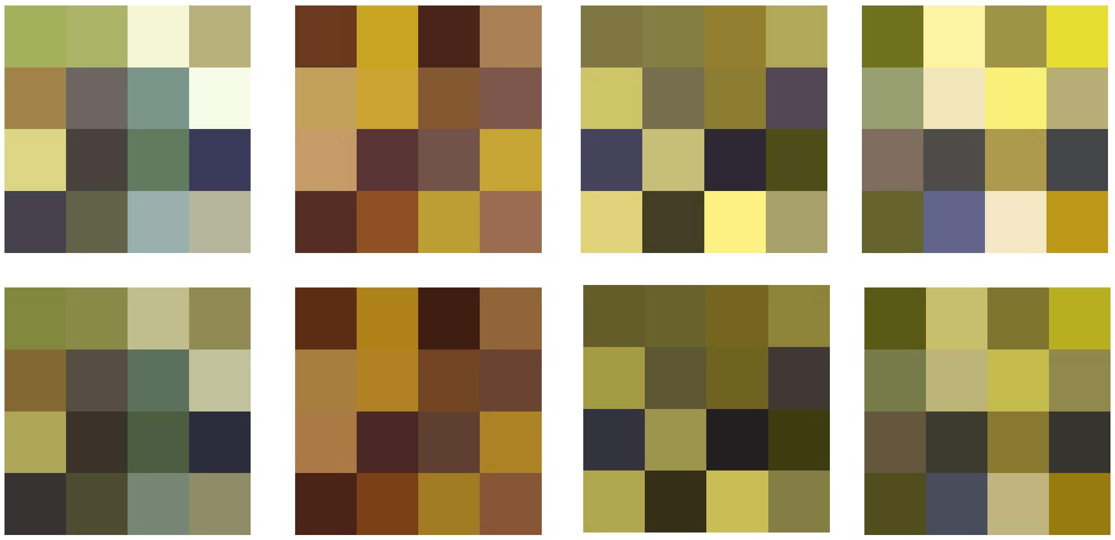


Figure 3.6: *Sample pairs of synthetic images with 16 reflectances under random illuminants from E_{287}*

♯ surfaces	1	2	4	8	16	32	rank
Chromagenic	6	5.2	4.5	3.5	3	2.2	2
Max RGB	9.7	7.9	6.1	4	2.9	2.6	6
Grey World	9.1	7.3	5.8	4.9	4.8	4.8	8
Database GW	9.5	6.7	4.8	3.4	2.8	2.5	4
LP Gamut Mapping	9.6	6.7	4.8	3.3	2.7	2.4	4
Neural Network	8.8	7.1	5	4	2.9	2.6	6
Colour by Corr.	6.9	5	3.5	3.1	2.4	2.3	2
Bright-Chromagenic	6	5.2	4.1	2.8	2.1	0.9	**1**

Table 3.1: *Average median angular error for 1000 tests at each complexity level. The last column is the rank, based on the 32 surfaces test, according to the Wilcoxon sign test with a confidence level of 95%*

ter than all other methods. We also see that the more complex methods form a group that is, in turn, significantly better than the simpler scene assumptions algorithms.

An additional result, shown in Fig. 3.7, is the reduction in maximal error achieved by the bright chromagenic algorithm. This experiment validates our selection of the bright RGBs to reduce the high max errors observed with the original chromagenic algorithm.

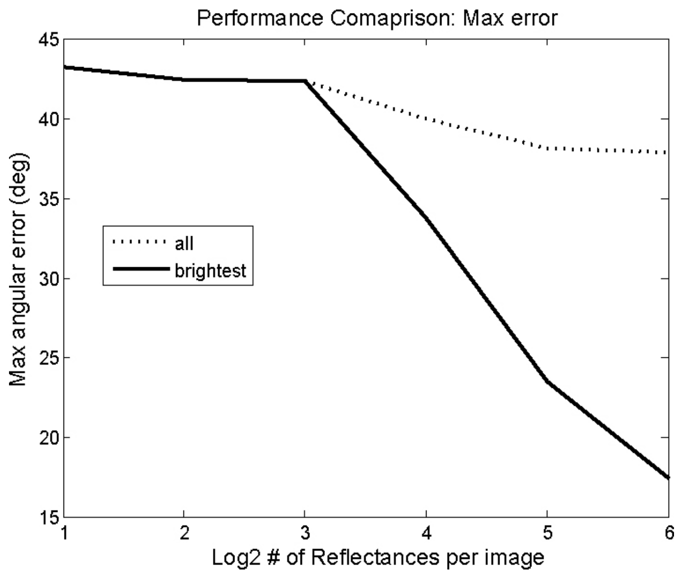


Figure 3.7: *Comparison of the max angular error between the original and the bright chromagenic algorithm, one can see the significant reduction achieved by selecting only the brightest RGBs.*

3.4.2 Real Reflectances and Synthetic Lights

In this second test, we use spectral outdoor reflectance images measured -as opposed to synthesized- by Nascimento et al. [NFF02]. Figure 3.8 shows the eight images that will be used (the images can be obtained at [Uni02]).

Note these images measure only reflectances. To generate RGB images, we use the scene lights and camera sensors as in the previous experiment.

Training: We leave here the training step unchanged from the previous experiment, i.e., the transforms are created on a different reflectance set than the one tested.

Testing: We start by creating images using the eight reflectance images illuminated with E_{287} to to generate 2,296 "half-synthetic" images. We then proceed to test our algorithm on each of those images, selecting the top 3% of the brightest pixels to estimate the illuminant.

Results from this dataset, Table 3.2 ,exhibit the same trend than in the all synthetic experiment, i.e., the bright-chromagenic significantly outperforms other methods.

Figure 3.8: *The eight outdoor reflectance images measured by Nascimento et al. [NFF02] that compose the second test.*

Algorithm	Median	Chrom.	Max-RGB	GW	Bright-Chrom
Chromagenic	6.7	=	+	+	-
Max RGB	8.7	-	=	+	-
Grey World	13	-	-	=	-
Bright Chromagenic	3.5	+	+	+	=

Table 3.2: *Summary of the results on the Nascimento images. This table displays the mean angular error values over the 2,296 images. A '+' in a row indicate that an algorithm performs significantly better (at 95% confidence level) than the one in the corresponding column, '-' and '=' are for when an algorithm performs worse or if there is no significant difference.*

3.4.3 SFU dataset

The next set we evaluate our algorithm on is the non-specular Simon Fraser University (SFU) dataset, which is described in detail in [BCF02].

The data set consists of 31 colourful objects captured under 11 illuminants. Figures 3.9 and 3.10 show some objects under one light and one object under all the available lights respectively. In the second case it is apparent that the images are not registered (in fact, the objects were rotated in between two pictures when creating the dataset).

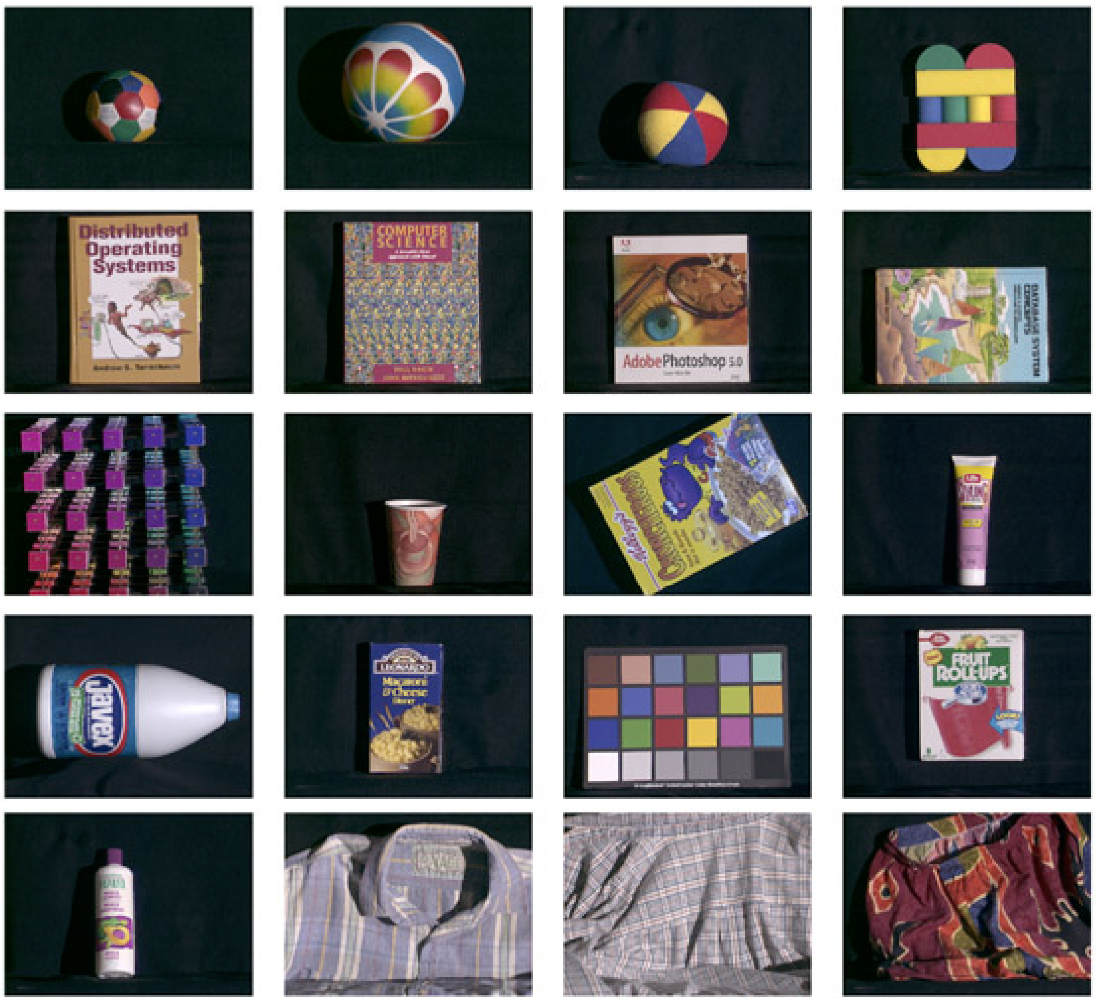


Figure 3.9: *Some objects of the SFU dataset under one illuminant.*

This experiment differs from the previous ones in the sense that here we are directly provided with the RGBs of the images instead of reflectances. This, plus the non-registration of the image will provide a difficult test for the bright-chromagenic algorithm. The SFU dataset has been used in several illuminant estimation comparisons because ground truth is provided in addition to the images themselves. That is, both the SPD of the 11 illuminants (they are actually a subset of E_{87}) and the camera sensitivities are given (the camera used to take the images is the Sony-DXC 930 whose sensitivities are shown in Fig. 3.1 and that we used in the previous tests).

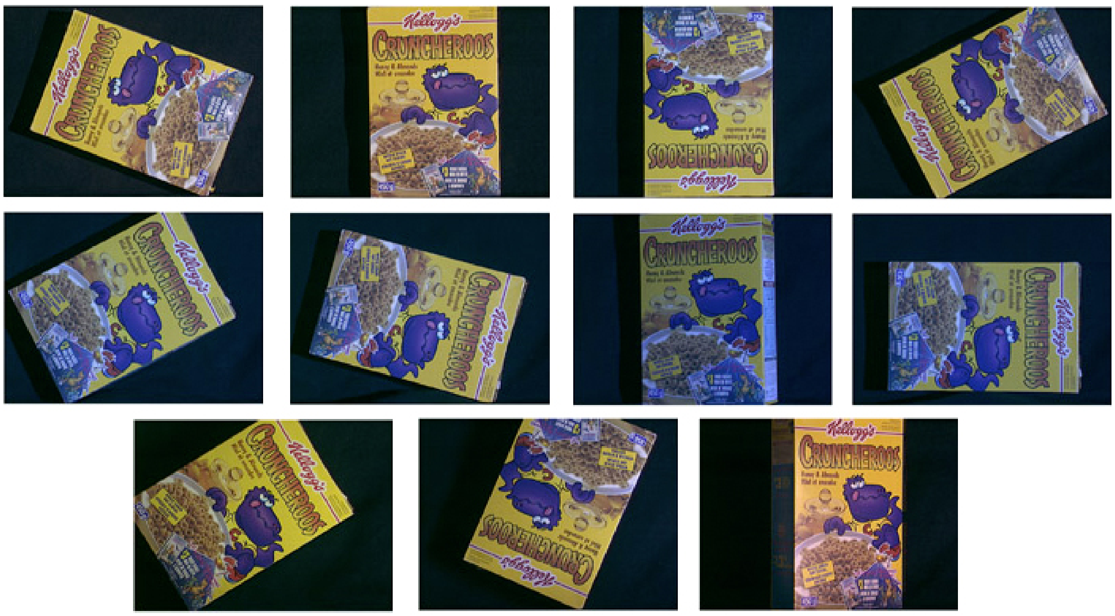


Figure 3.10: *One object from the SFU dataset under the 11 illuminants. Note that the lights in the second row are the same as the ones of the first row with a bluish filter placed in front of them. One can also see that the images are not registered.*

To perform chromagenic illuminant estimation, we require pairs of images taken with and without a coloured filter. However, if we only have image RGBs at our disposal we cannot retroactively model the filtered responses. As it turns out, 8 out of the 11 illuminants present in the set come in pairs: the original lamp lights and those lights filtered with a blue filter (the lights and their filtered counterparts are shown in the first 2 rows of figure 3.10). Since the actual illuminant SPDs are known, we can derive the filter that was used -we do not actually know what it is- by dividing the spectra of the lights by the filtered ones. The eight -two pairs of four- lights that are considered and the derived filter are shown in Fig. 3.11.

Training: The transforms $\mathcal{T}_i$ are obtained by imaging the synthetic reflectances S_M under the illuminants of E_{87}. As filter, we use the one derived from the eight illuminants shown in Fig. 3.11; the camera sensitivities are the same as in previous experiments.

Testing: To test the algorithm, we estimate the illuminant of all the possible pairs of images (124 pairs in total) using the top 3% of the brightest pixels in both filtered and

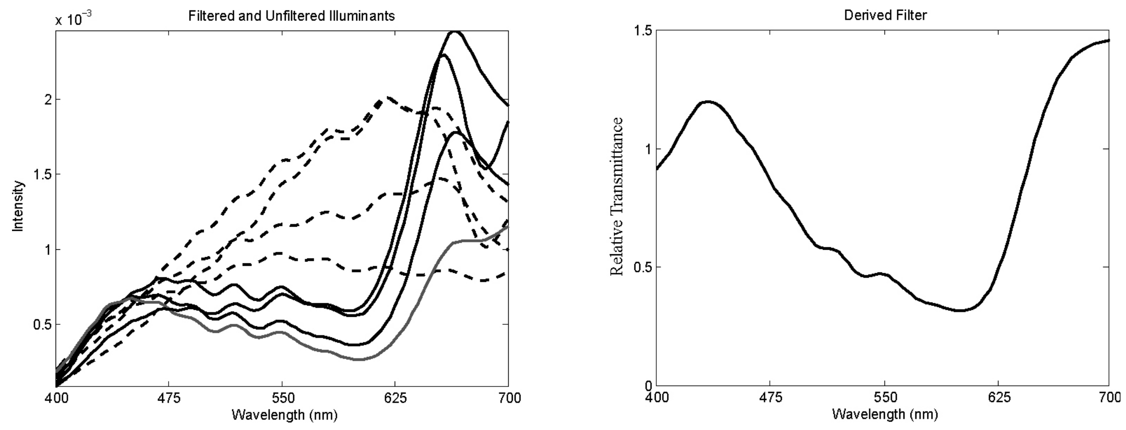


Figure 3.11: *Left: The 8 light sources considered in this experiment. The dashed lines are spectra of the light sources, while the continuous ones are from the filtered sources. Right: The Filter derived from the light source data. The maximum transmittance higher than 1 is due to the camera auto-exposure function. While this is not the data of the physical filter, it is what the camera "sees" and therefore what we use in training the transforms.*

unfiltered images independently. These pixels typically belong to one or two of the surfaces in the scene (we do not need a white reflectance *per se*, we simply take the brightest ones available). Since the images are not registered, we are able to test our hypothesis that, in general, the brightest pixels in both images come from the same surfaces, and that the bright-chromagenic algorithm does away with the need for registration.

The angular errors reported in the first two columns of Table 3.3 show that, despite its simplicity, the bright chromagenic algorithm outperforms in terms of both mean and median angular error all other algorithms at the 95% confidence level. The original chromagenic algorithm is not shown here since its registration requirement is not fulfilled.

Perhaps the most remarkable aspect of the bright-chromagenic algorithm is that, despite modelling the transforms on synthetic data with a filter derived from measurements, it is still able to estimate accurately the illuminants of real, significantly non-registered images.

Algorithm	Mean	Median	B-Chr	RGB	GW	DB	NN	GM	CbyC
Bright-Chromagenic	4.8	3.4	=	+	+	+	+	+	+
Max RGB	6.4	4.1	-	=	+	+	+	=	-
Grey World	11.9	9.3	-	-	=	-	=	-	-
Database GW	10	7	-	-	+	=	=	-	-
Neural Network	8.9	7.8	-	-	=	=	=	-	-
LP Gamut Mapping	5.5	3.8	-	=	+	+	+	-	=
Colour by Corr.	6	3.6	-	+	+	+	+	=	=

Table 3.3: *Summary of the results on the Simon Fraser dataset. The table shows mean and median angular error as well as the results of the Wilcoxon sign test at the 95% level.*

3.4.4 Real Images

The last, experiment is designed to evaluate the performance of the bright-chromagenic algorithm *in situ*. Whereas the previous datasets were obtained in “controlled conditions” (purely synthetic data, partly synthetic data and controlled lighting environment), we use here a set of real-world images taken with a digital camera whose specifications are unknown.

Chromagenic Photography:

For the illuminant estimation to be meaningful, we must take a couple of precautions when capturing the images. The camera we used is a Nikon D70: a prosumer Single Lens Reflex camera. The camera was setup to capture linear RAW, unprocessed, images[1]. To prevent the camera from using a different white balance between filtered and unfiltered images, all images were captured with the white balance set to “daylight”.

Additional technical aspects to consider are that we want to have image pairs that are as registered as possible, and also wish to avoid over or under-exposed regions, which make the relationship between filtered and unfiltered pixels meaningless. To that effect, the images were captured using a tripod and a remote shutter release (to minimize registration errors) and the settings of the camera aperture and shutter speed were set to

[1]In fact, even with RAW settings, the camera and associated software will process the image somewhat; it is however as unprocessed as one can have with a general purpose digital camera.

"manual" mode where we aimed to capture the entire dynamic range of the image.

For the filter, we used an actual 81B-type Wratten filter. The captured images were then exported, using Nikon capture, as 16bits/channel linear tiff images.

All images were captured by Dazlong Luang at the University of East Anglia and we gratefully acknowledge his help and assistance.

The dataset consists of 86 pair of images taken under a variety of indoor and outdoor illuminants. In every scene, we placed a Macbeth colour checker that is used to accurately determine the color of the prevailing light, thereby providing a ground truth to assess the accuracy of illuminant estimation algorithms.

From the dataset, we then create separate training and testing sets. The training set consists of the 24 Macbeth patches present in all the images. The testing set is created by blacking the colour checker from the images. Images from the original (with the colour chart) and the testing set are shown in Fig. 3.12.

We note that, despite the precautions taken, the registration between images is not perfect and some image regions can be over-exposed. Additionally, multiple illumination is sometimes present in images, which can lead to errors when estimating the prevailing illuminant.

Preliminary assessment:

Before testing the performance of the algorithm on the images themselves, we want to find out whether reasonable estimates can be obtained with simple reflectances. To do so, we perform two simple tests: in the first one, we calculate the $\mathcal{T}_i$ on half of the Macbeth charts and test the algorithm on the 43 other charts (the partition is chosen randomly). The chromagenic algorithm is tested using the entire chart while the bright chromagenic is tested on the 5 brightest reflectances. In the second test, we calculate the $\mathcal{T}_i$ based on 16 of the 24 reflectances of the checker (we chose to use the 6 grayscale and 10 coloured ones selected at random among 18). We estimate the illuminant based on 4 of the remaining -untrained- reflectances. In the case of the original chromagenic

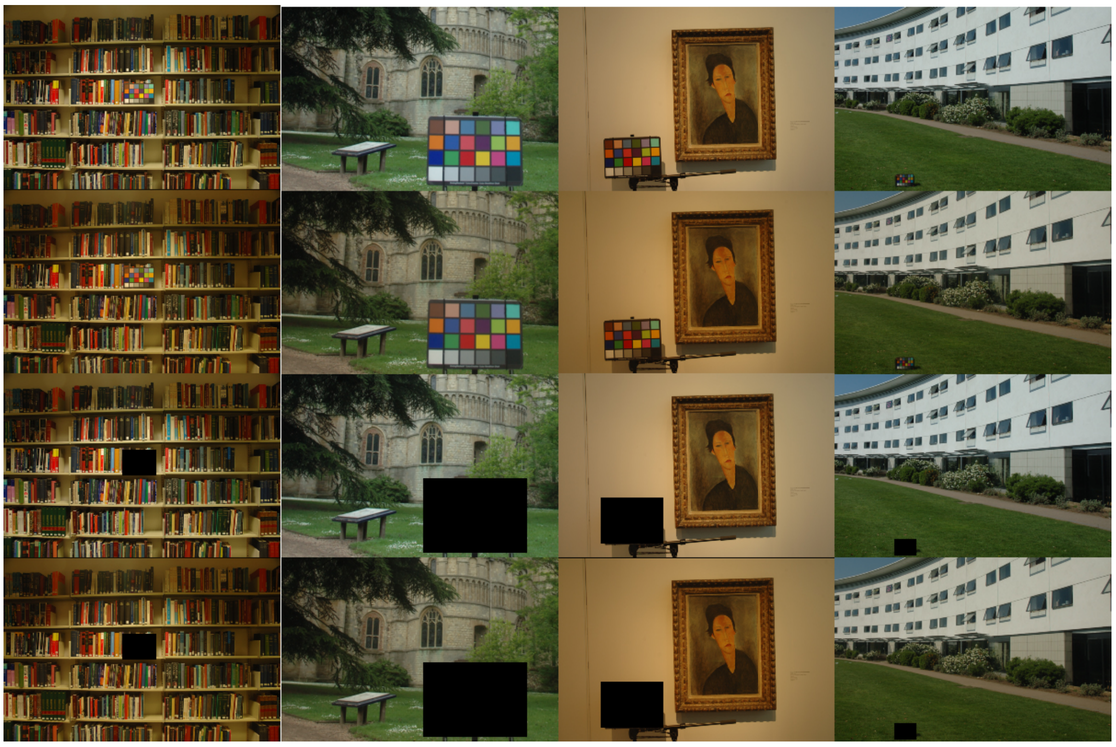


Figure 3.12: *Real images taken with a Nikon D70 camera. The first row is unfiltered images. The second row consists of the corresponding filtered images. The last two rows are the equivalent images in the testing set, where the colour chart has been removed.*

algorithm, those 4 are taken at random among the 8 untrained ones. For the bright chromagenic algorithm, we use the 4 brightest reflectances among them. As these tests have inherent randomness, each of them is repeated 1000 times and their average is mentioned here. The first test reports median angular errors of 2.53 and 2.4 (for the original and bright-chromagenic respectively, the difference is significant). The second tests yields 4.49 and 4.05 respectively (significant difference as well). These results are broadly in line with the ones obtained on synthetic data in our first experiment, which allows us to put the results obtained on our real data in perspective with the other tests previously performed.

Training: We create 86 linear transforms, using the 24 RGBs of the colour checker present in each image.

Algorithm	Mean	Median	B-Chr	RGB	GW	Chromagenic	ℓ^4
Bright-Chromagenic	7.09	4.15	=	+	+	+	+
Max RGB	7.87	7	-	=	+	-	+
Grey World	11	10.8	-	-	=	-	-
Chromagenic	7.96	5.1	-	+	+	=	+
ℓ^4 Gray [FT04]	10.3	9.6	-	-	+	-	=

Table 3.4: *Mean and median angular errors over the 86 real images. The significances are reported according to the test sign with a confidence level of 95%.*

Testing: We estimate the illuminant for both the original and bright-chromagenic (using the top 3% brightest pixels) algorithms on the 86 pairs of images that have the colour chart clipped out.

The results are shown in Table 3.4 and illustrate that the most accurate illuminant estimation is given by the bright-chromagenic algorithm. The results otherwise exhibit the same behavior as previous experiments.

3.5 Conclusion

A chromagenic illuminant estimation algorithm exploits the relationship between RGBs captured by a conventional camera and those captured through a coloured filter. Different lights induce different relationships and so, the illuminant colour can be estimated by testing pre-computed relations *in situ*. While the chromagenic approach can work well, it occasionally performs poorly. Moreover, typical chromagenic camera embodiments such as a stereo rig or where there are multiple surveillance cameras (a filter can easily be placed over one camera) do not have pixel registration and this is assumed in the chromagenic theory.

In this chapter, a detailed error analysis demonstrated that bright pixels in images lead to smaller chromagenic estimation errors. This led to the bright-chromagenic algorithm, which bases its estimation only on a fixed percentage of the brightest pixels in the filtered and unfiltered images. Importantly, these pixels are chosen independently in each image so there is no need for image registration. Experiments on various sets of synthetic and real data demonstrate that the bright-chromagenic algorithm delivers a better illuminant estimation than all other tested algorithms. The performance is especially promising considering that, if the camera sensitivities are known, the transforms can be pre-computed on synthetic data even without knowing the content of the test scenes, such as demonstrated on the SFU test (third experiment).

It was proposed in [FHM05b] that the average performance of the chromagenic algorithm could be improved by incorporating knowledge about the plausibility of an illuminant given the colours present in an image. This modified algorithm effectively combines Gamut Mapping and chromagenic illuminant estimation. It was subsequently shown that this hybrid algorithm performed better than the original formulation alone. In our work, while we have tested this constraint and found it to improve our bright-chromagenic algorithm as well, we opted not to add it. The rationale is that we are trying to develop a method that has minimal inference and so coupling it with Gamut Mapping would, in a sense, defeat this purpose.

Chapter 4

Detecting Multiple Illuminants

The presence of multiple illuminants in an image is an obstacle for many computer vision algorithms. Most illuminant estimation algorithms can also perform poorly when confronted with this situation, as we have seen in Chapter 3. Scenes where there is more than one principal light source do however occur in many situations, such as the presence of shadows or daytime indoor environments.

Common ways to minimize the problems created by multiple lights are to perform a given task only in regions containing a single light or to remove the non-prevailing illumination, thereby rendering the entire image under a single illuminant.

Before these methods can be implemented though, it is necessary to first detect the parts of the image that are predominantly illuminated by spectrally different lights. In this chapter, we propose two different methods to achieve that aim.

First, we look at Finlayson et al. invariant image method for shadow detection presented in [FDL04] and reviewed in Chapter 2. We note that while the shadow edges are mostly accurate, they are not closed (this will introduce artifact in the reintegration) and they cannot account for shadows of varying strength (such as the ones created when multiple point light source are present). We propose to remedy these shortcomings by simple extensions of their framework.

In a different approach, we propose a novel method to detect multiple illuminants using the chromagenic theory. We constrain the illuminant estimation problem so that we look for a set of linear transforms where each image region is indexed by one of the transforms of the set. Results at a pixel-level show that while noisy, this method gives a good outline of the different regions. By segmenting the image, however, we obtain illumination masks that are highly faithful to the original scene.

4.1 Completing Invariant-Based Shadow Edges

Looking back at the shadow detection method of [FDL04], we see that shadow edges are obtained by comparing edge maps from both the original and the invariant image (this framework is shown in Fig.4.1). Basically, this method will not detect coincidental material and illuminant change because of its assumption that edges present in the invariant image are necessarily material edges only.

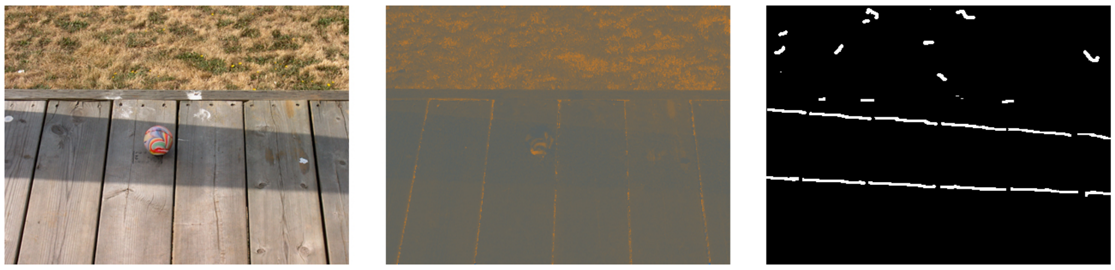


Figure 4.1: *The method to obtain shadow edges according to [FDL04]. While most of the shadow edges are correctly detected, some edges are mistakenly classified. Also, the edge map is fragmented.*

Having open edges is, however, problematic. Indeed, an algorithm basing its decisions on whether a region is in the shadow or not can make a wrong choice when confronted with fragmented edges. Moreover, shadow removal algorithms need to have complete edges when processing difficult scenes; open edges will result in integration errors that can hinder the performance of the removal procedure (this problem is discussed in detail in Chapter 5).

To address this issue, we modify the original formulation by incorporating a post-processing step that will automatically close the edges based on the mean-shift segmentation of the original image. This does not come at an additional cost since the mean-shift segmentation is already used in [FDL04] to detect edges in both the invariant and the original image.

In a different approach, we propose to simplify the framework of shadow detection by removing the step where edges in the invariant image are sought. Instead, we provide a region-based approach that will ensure the edges are closed (since we now look for regions) and that also allows to distinguish relative shadow strengths.

4.1.1 Closing Shadow Edges

To close shadow edges, we will use two different edge maps. The first one, I_S, is obtained using the method of [FDL04] and shown in Fig. 4.1 (right). The second map is the result of segmenting the original image with the meanshift algorithm [CM02] where default parameters are used (except for the minimum region size which we chose to be 0.5% of the image size) in order to preserve all shadow edges; let I_M be that image. The workflow of our method is shown in Fig. 4.2.

By construction, I_S contains mostly shadow edges but they are generally incomplete and the map can be "noisy". I_M on the other hand contains most of the image edges, we will therefore use it to complete the edges of I_S. The first step is to prune the noisy edges of I_S: we identify edges in I_S (the white pixels) and remove all those that are not coincidental to an edge in I_M, Fig. 4.2d. Then, we look at the relative lengths of the edges; that is, for each remaining edge in I_S, we look at the corresponding edge in I_M. If the length of the edge in I_S is less than 10% the length of the one in I_M we remove it as well, Fig. 4.2e. The remaining edges are then completed using I_M as a guide by "stretching" them so that they are equivalent, Fig. 4.2f. In the event where the resulting edge map is still incomplete, we then look at the closest distance using edges from I_M

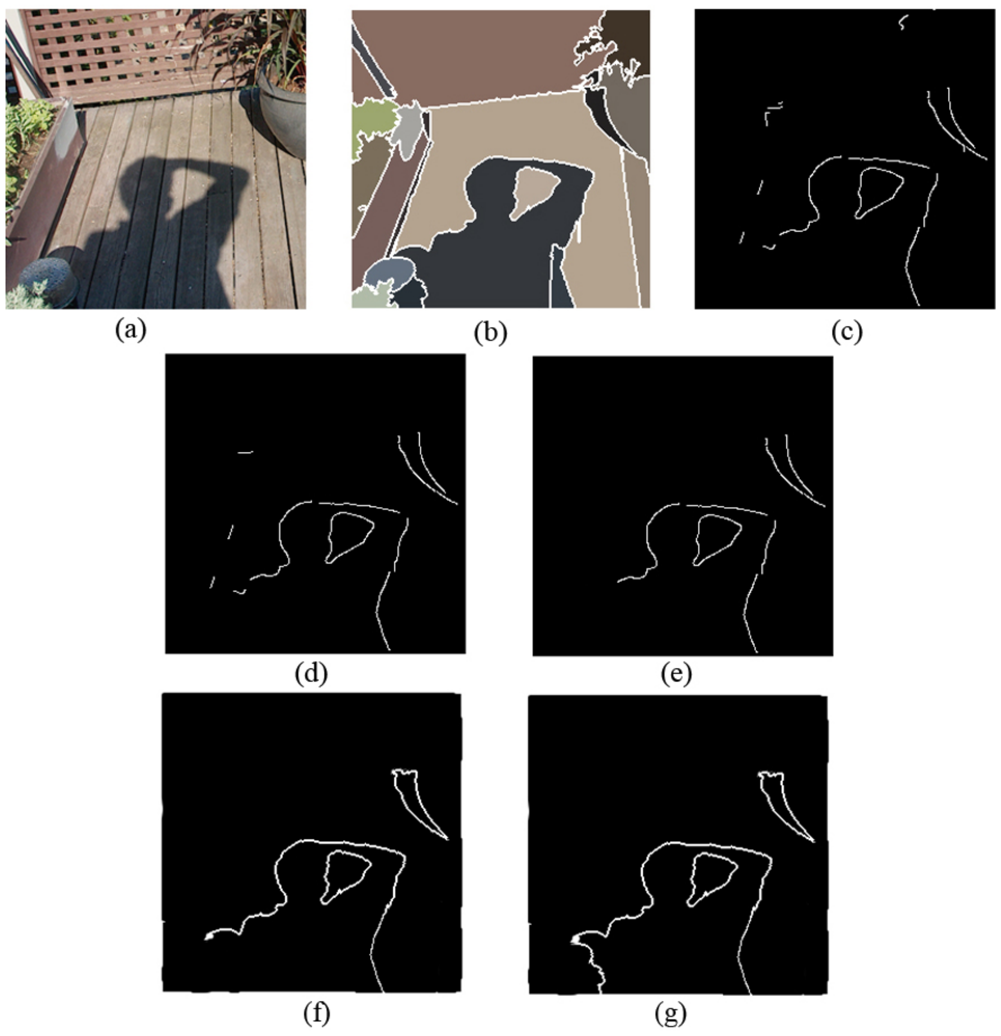


Figure 4.2: *First row: (a) the original image, (b) the meanshift segmentation I_M and (c) the original shadow edge from [FDL04] I_S Second Row: (d) Edges from I_S that correspond to edges in I_M (the others have been deleted); (e) Edges from I_S that are less than 10% in length than the corresponding edge in I_M. Last row: (f) Extending edges from I_S using I_M as a guide. (g) Completing the still open segmentation using the shortest path available in I_M to join the open ends.*

to join the open ends, Fig. 4.2g.

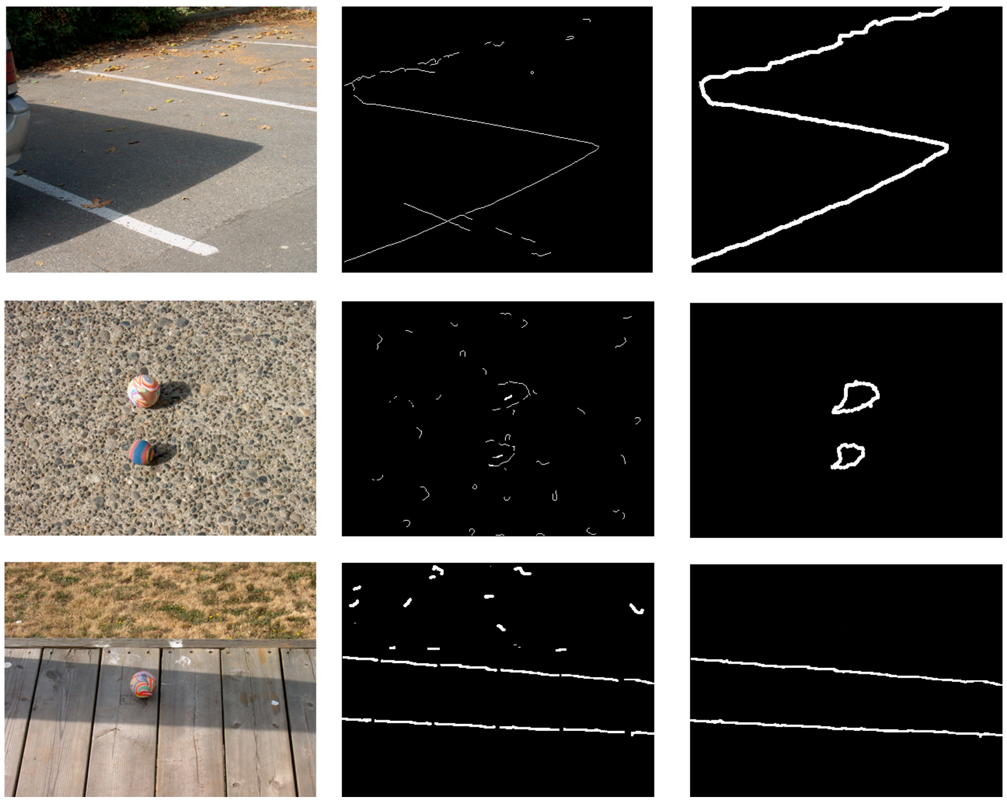


Figure 4.3: *Left and middle column: Images and their shadow edges from [FDL04]. Right column: the complete edges obtained with our method.*

4.1.2 Region-Based Approach

We propose here a method that simplifies the framework of shadow detection, that ensure all shadow edges will be closed, and that provides a shadow mask that takes into account the possibility of multiple intersecting shadows.

This time, we do not use the shadow edge image. Instead, we will use the 2D chromaticity invariant image, I_{inv}, and the meanshift image, I_M, obtained with the same parameters as before. Let us denote by R_i the regions of I_M and by R_{C_i} the regions that are connected to (i.e, have a common edge with) R_i. We look at the regions in I_{inv} (e.g, by overlaying the I_M edge map, see Fig. 4.4 for an illustration). Let us consider a specific region $R_{\hat{\imath}}$. If all its connected regions $R_{C_{\hat{\imath}}}$ have different RGB values from $R_{\hat{\imath}}$ (more than 10% difference), we do nothing because it implies that the difference picked up in meanshift is a material difference. Otherwise, if there is a $\hat{\jmath} \in C_{\hat{\imath}}$ such that $R_{\hat{\jmath}}$ has less than 10% difference with $R_{\hat{\imath}}$, we label the darkest region of the two as a shadow region. Since we label regions instead of edges, we are sidestepping the problem of open edges. Additionally, this method allows us to rank the relative strength of adjacent shadow regions.

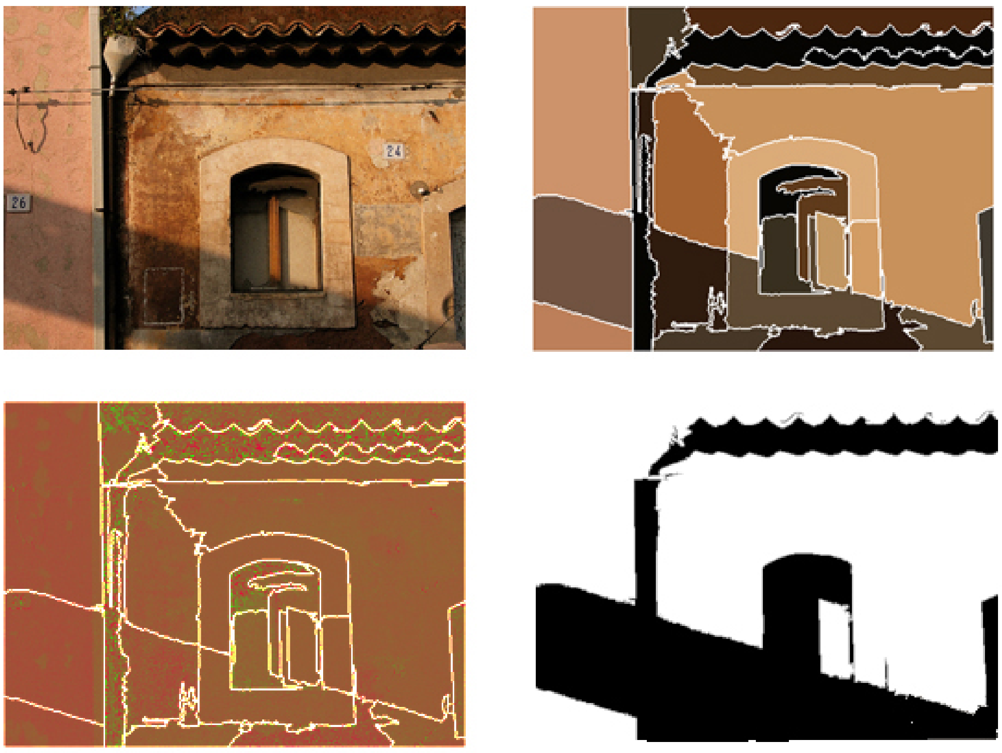


Figure 4.4: *Top row: The original image and the meanshift image, used to compare the relative darkness of regions. Bottom row:the invariant image with the edges from I_M superimposed and the resulting shadow mask.*

Consider the case -illustrated in Fig 4.5 top left- where multiple shadows are cast by different point light sources. When the shadows from two light sources intersect, their intersection will be darker than the original regions, thus creating another region in I_S. The invariant image will, however, not bear an edge and we will therefore face the case where a shadow region will be connected to a "darker shadow" region. We can thus decide to label them as such, therefore enabling us to remove those multiple shadows more accurately.

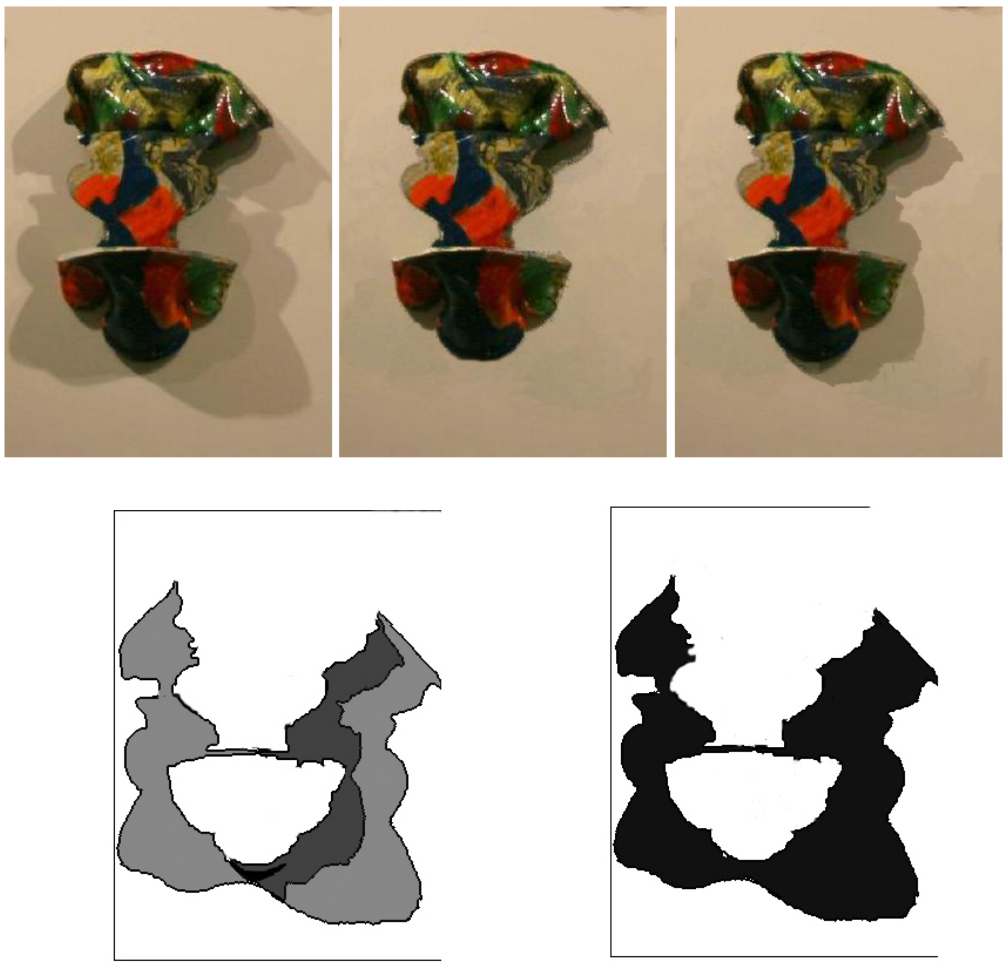


Figure 4.5: *Top row: The original image, the reintegration using a multiple level shadow mask, the reintegration using a binary mask. Bottom row: the multiple level mask proposed here, each shade of gray corresponds to a different relative shadow strength -here: 3 different levels; the binary shadow mask used in [FDL04].*

A comparison of the influence of the new shadow mask and the one proposed in [FF05] on reintegration can be seen in Fig. 4.5, as well as the resulting reintegration from the different masks. Shadow masks obtained with this method are shown in Fig. 4.6.

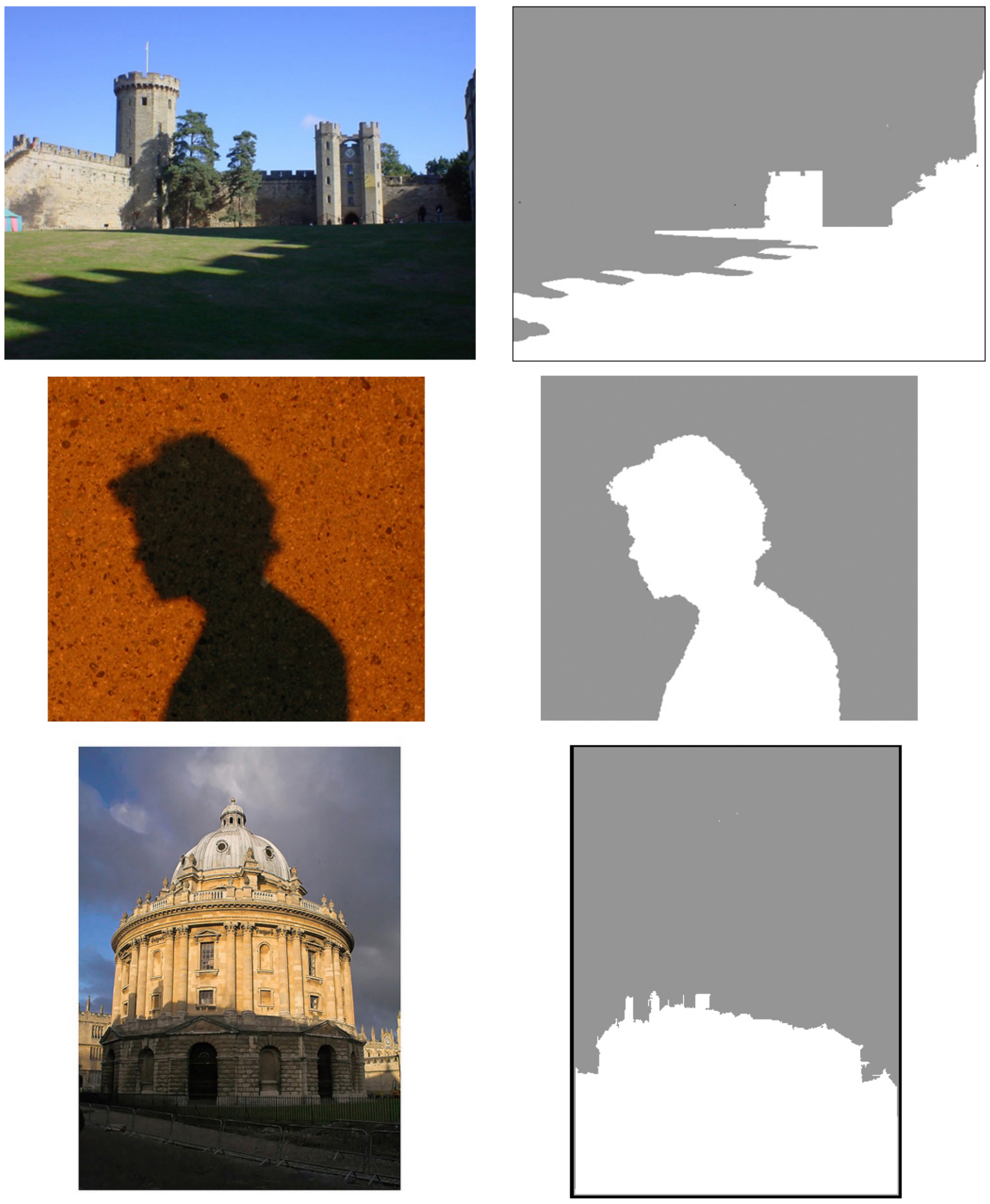


Figure 4.6: *Left column: Original Images. Right column: Shadow regions detected with our method.*

4.2 Chromagenic Illuminant Detection

In this section, we propose a new method to detect shadows and other types of multiple illumination. Our approach, based on the chromagenic method for illuminant estimation outputs very precise binary illumination maps that can even accurately detect occlusion shadows in cases where all the edges surrounding a shadow region are coincidental material/illuminant edges, a non-feasible task for most shadow detection methods.

In Chapter 3, we reviewed the chromagenic algorithm for illuminant estimation. This algorithm can accurately estimate an image illuminant and, importantly, is pixel-based. One can therefore assume that the same methodology can be applied to illuminant detection. To do so, given a pair of filtered and unfiltered images as well as a set of pre-computed linear transforms $\mathcal{T}_i$, one simply applies the chromagenic illuminant estimation algorithm and, for each pixel in the image, records which transform best maps the unfiltered RGB values to their filtered counterparts. The best transform for each pixel indexes the incident illuminant for that pixel.

An example of such processing is shown in Fig. 4.7 where a reflectance image from the Nascimento set [NFF02] is illuminated with two distinct lights from E_{87} (the 87 lights from [BCF02]). We see from (4.7 right) however that the detected illuminants -each pixel has for value the index of its best transform- do not correspond to the input, even though the images are perfectly registered. The problem here is that the chromagenic approach is efficient when multiple surfaces are present in the scene but is also -as we have seen in Chapter 3, Fig. 3.3- fragile, when a single surface is present in the scene.

To address this stability issue, we transform the illuminant estimation algorithm in one of illuminant discrimination -or detection. Importantly, we do not aim to recover the actual scene illuminants but, instead, we look for the transforms that best discriminate the multiple illuminants in the scene irrespectively of the estimation accuracy. The starting point of our approach is to suppose there are m lights present in the image. In practice $m \leq 2$ will be appropriate for most images, $m = 2$ is particulary important for it represents the shadow detection case.

Let E_N be a set N lights for which we carry out the chromagenic preprocessing

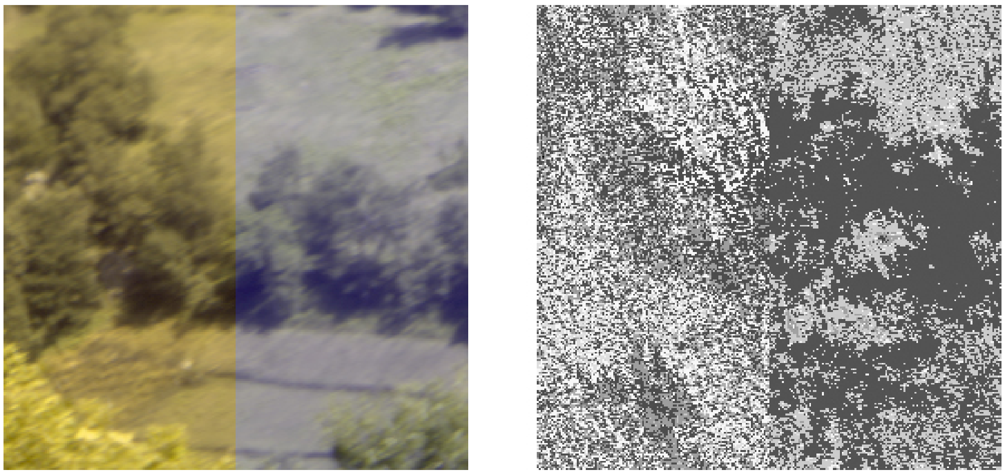


Figure 4.7: *Left: One of the reflectance image from [NFF02], the left and right halves of the image are illuminated by two different lights. Right: The result of illuminant detection using the standard chromagenic algorithm. Each pixel of the image has for value the index of the transform that best maps it to its filtered counterpart.*

step and solve for the N relations, the 3×3 linear transforms $\mathcal{T}_i$, that best map RGBs to filtered counterparts.

Suppose we now select m elements in N, denoting the corresponding subset E_m, with $E_m \subset E_N$. Taking each pixel in turn, we determine which of the m relations best maps its RGB values to the filtered ones. Once each pixel is assigned a single of the m relations, we can assess how well the subset E_m accounts for the data by calculating its error:

$$\text{error}_{E_m} = \sum_{i=1}^{m} \|\mathcal{T}_i \, I_i - I_i^F\| \tag{4.1}$$

Where I_i and I_i^F are the unfiltered and filtered RGB pairs of the image I that are best mapped by $\mathcal{T}_i$.

Equation (4.1) measures how well *one* subset of E_N models the transition from the image to its filtered equivalent. It does not, however, guarantee that this particular subset is optimal -it does not even tell how good this subset is. To minimize detection errors, we therefore have to evaluate equation (4.1) for all possible m-elements subsets of E_N. Let $\mathcal{E}(m)$ be the set of all m-elements subsets of E_N. The $E_{\text{opt}} \in \mathcal{E}(m)$ that best describes the relation between the image and its filtered counterpart is:

$$E_{\text{opt}} = \arg \min_{E_m \in \mathcal{E}(m)} (\text{error}_{E_m}) \tag{4.2}$$

Once we have found subset E_{opt} that minimizes the mapping error among all m-subsets, we create our illuminant map M as:

$$M(x) = \arg \min_{i \in m} \|\mathcal{T}_i I(x) - I^F(x)\| \tag{4.3}$$

i.e., the x^{th} pixel of M takes the value of the index of the transform that best maps the x^{th} pixel of I to its filtered response.

Implemented naively, obtaining M can be computationally laborious. The computational cost is proportional to the cardinality of the set $\mathcal{E}(m)$. If we chose m lights among N, then:

$$\sharp(\mathcal{E}(m)) = \frac{N!}{(N-m)!m!} \tag{4.4}$$

Considering our set of 87 lights, the number of different m-sets is $(3,741)$, $(105,995)$ and $(2.2\ 10^6)$, for $m = 2, 3, 4$ respectively. A brute force search is only really possible for small m, i.e., $m = 2$ or $m = 3$.

4.3 The Two Illuminants Problem

The case where $m = 2$ is the commonest instance of the multiple illumination problem. Indeed, in every day circumstances such as the presence of shadows or the combination of natural and artificial light sources in indoor environments, the number of main, different, illuminants is rarely greater than two.

Limiting ourselves to the case where two illuminants are present also makes the problem more tractable since there are only $\frac{N^2 \pm N}{2}$ relations to test for if we have a set of N lights (the $\pm N$ comes from the decision to include or not pairs of identical illuminants). In our experiments, since we have either 87 -for the synthetic case- or 86 -for the real images- illuminants: the number of possibilities is smaller than $4,000$.

To test our framework for illuminant detection, we start by considering the reflectance images of Nascimento et al [NFF02] and E_{87}: the SFU set of 87 measured lights. We generate images by illuminating each half of the image with a different light. Then, we use our algorithm to classify pixels as '0' or '1' depending on which of the two illuminant best maps them to the filtered values, thus creating a binary mask of illumination. This procedure is illustrated in Fig. 4.8 where the two images are illuminated by sky-light/sunlight and floodlight/sunlight respectively, thereby reproducing two of the most frequent naturally occurring two-illuminant scenes.

The results illustrate that, while generally accurate, the classification is noisy even though the images are perfectly registered. The presence of noise is expected since the classification is still done at pixel-level only; the confusion can be explained by a misclassification due to having a single reflectance for each computation. To improve the detection accuracy, we propose to perform the classification not at pixel-level but at region-level.

The main insight of a region-based approach is that, over an area, the predominance of a class of pixels is correlated with the prevailing illuminant. It follows that we can modify the pixel-level classification to take into account neighboring information.

To do so, let $\mathbf{R}$ be a partition of I into K distinct regions $R_j,\ j = \{1, \ldots, K\}$. We formulate our region-based approach by rewriting equation (4.3), defining the illuminant map for a region R_j, M_{R_j}, to be the result of a function over the filtered and unfiltered corresponding region of the image.

$$M_{R_j} = \mathcal{F}_{R_j}(I_{R_j}, I^F_{R_j}) \tag{4.5}$$

where I_{R_j} represents the pixels of I that belong to region R_j; M_{R_j} will take a single value for the entire region. For each region of the image, the region is labelled using

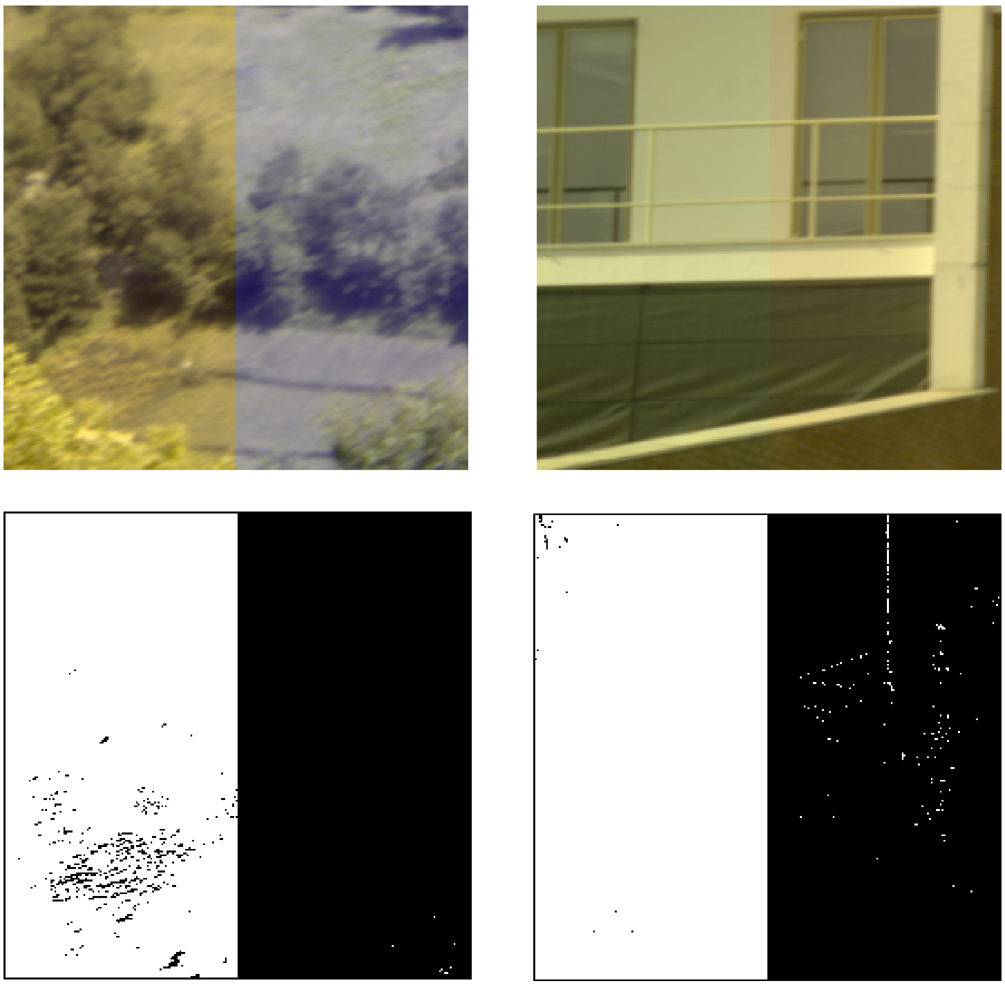


Figure 4.8: *Top row: 2 reflectance images from the Nascimento dataset [NFF02]. The left image created using both skylight and sunlight spectra, The right image using neon light and sunlight spectra. Bottom row: The pixel wise classification of illuminants for the two images.*

the index of the transform that best maps it to its filtered response. We use here the function $\mathcal{F}_{R_j}$ to define what that best mapping is. The function can take any form and we illustrate here two: majority voting and minimization of pixel-based error over the region.

In the majority voting case, the image is first processed according to equation (4.1). Then, within each region, the number of pixels that belong to each class is counted. The entire region is assigned the label of the majority of pixels:

$$\mathcal{F}_{R_j}(I_{R_j}, I^F_{R_j}) = \arg\max_i(\sharp(\mathcal{T}_i I_{R_j} - I^F_{R_j})) \tag{4.6}$$

That is, the function takes for value the index i which maps the largest number of pixels in R_j with the least error (compared to all other transforms).

The region-based labelling can also be done using a $\min$ error metric. In that case, the whole region is evaluated using both illuminants in turn. The region is then labelled according to which one of the two transforms has minimal error, i.e.,

$$\mathcal{F}_{R_j}(I_{R_j}, I^F_{R_j}) = \arg\min_i \|\mathcal{T}_i I_{R_j} - I^F_{R_j}\| \tag{4.7}$$

The outcome of both methods is illustrated in Fig. 4.9 where the image is partitioned in 8×8 blocks (the outcome of both region-based labelling is identical, so a single map is shown). On most of our experiments, we have not found any significative difference between either method, so in the following we keep our framework coherent and use, for region classification, the method of error minimization.

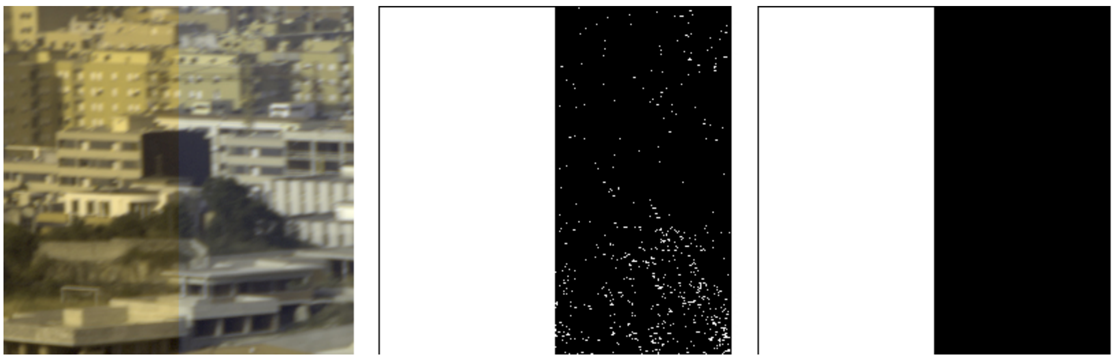


Figure 4.9: *Left: The original image with two illuminants. Middle: the result of pixel-wise classification. Right: the illuminant mask processed on* 8×8 *regions with the methods of either majority voting and error minimization. The results of both region-methods is identical; in general no significant difference is observed between the two methods.*

Using a region-based labelling has an additional advantage when illuminant detection is performed on real images: image registration (or lack thereof). Consider the pair of images shown in Fig. 4.10; the images appear to be registered but, at a pixel-level, it is actually not the case. Consequentially, the detection will be noisier than in the synthetic case (Fig. 4.10: right). A region-based approach is therefore more adapted for a cleaner classification.

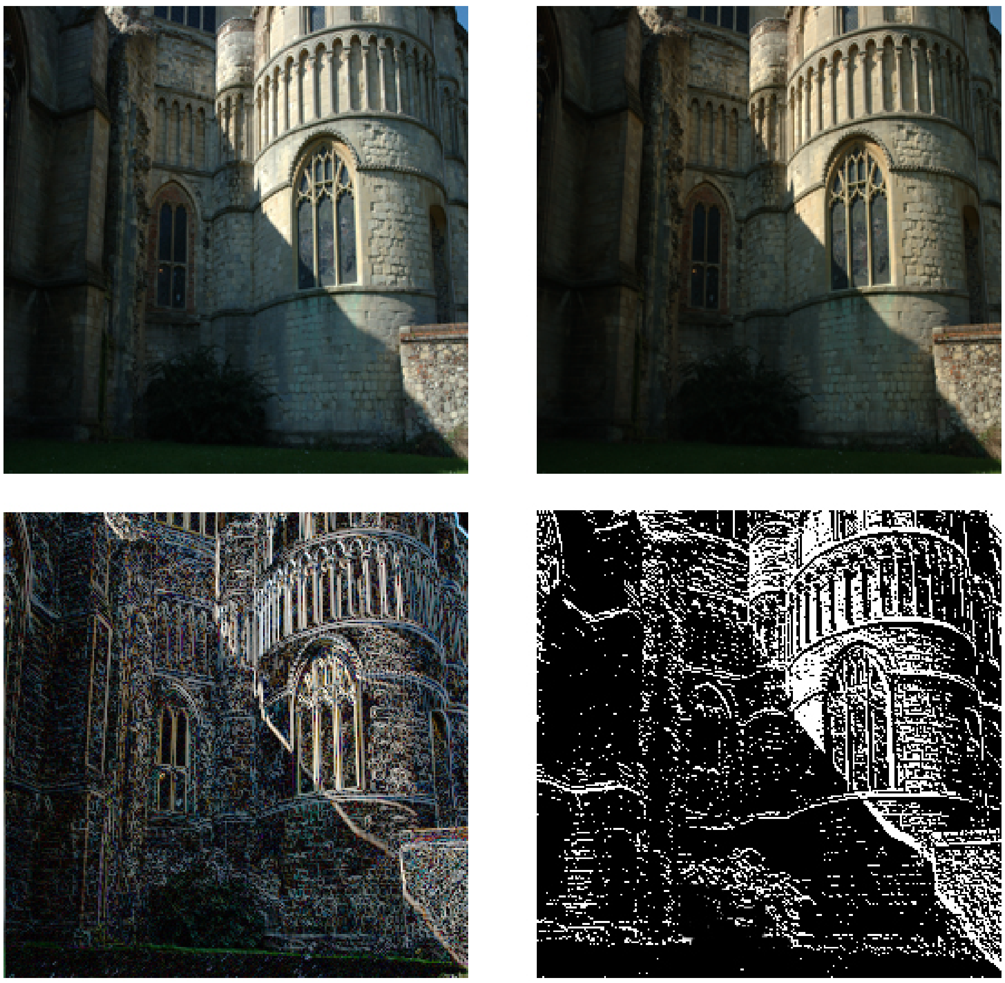


Figure 4.10: *Top Row left and right: the original and filtered image, while they appear to be registered, this is not the case at pixel level. Bottom row: Registration differences (left) and the result of pixel-based classification (right). The detection is mostly accurate but noisy.*

Since real images have noisier masks, the regions' shape and size noticeably influence the final results. This dependency is illustrated in Fig. 4.11 where we process the pixel-level mask using both a 8×8 window and regions obtained with a segmentation algorithm (in this case, the meanshift algorithm [CM02]). While in both case we significantly oversegment the image, the meanshift segmentation is better for our purposes as it preserves the image edges and results in more accurate masks. We point out that any segmentation method that preserves most of the edge structure of the image would be suited for our type of processing.

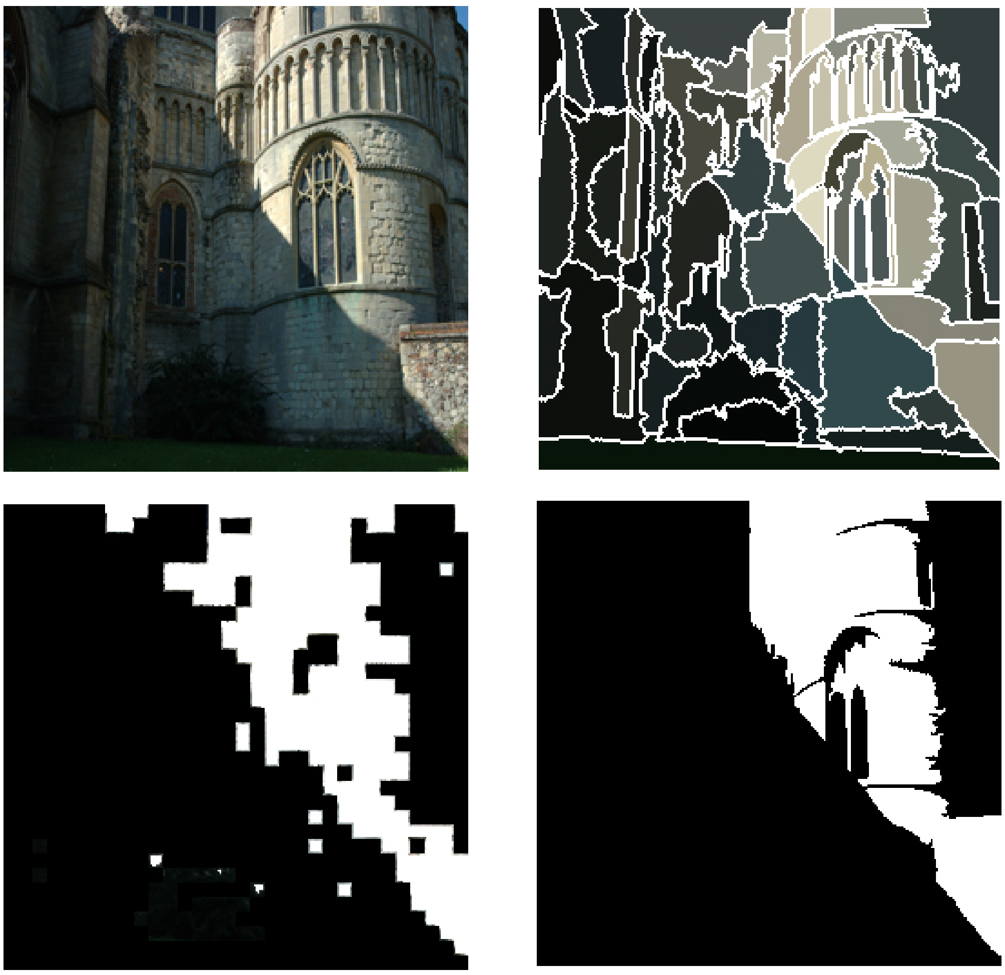


Figure 4.11: *Top row: The original image and the meanshift segmentation. Bottom row: the results from partitioning the image with 8×8 blocks and with the meanshift algorithm. Since it preserves the edges, the segmentation results yields more accurate masks.*

To visualize the results, the image can be decomposed according to the binary mask. Doing so allows us to actually see what parts of the image are detected as being differently illuminated. The results, Fig. 4.12, show that the image is effectively segmented in shadow and non-shadow regions.

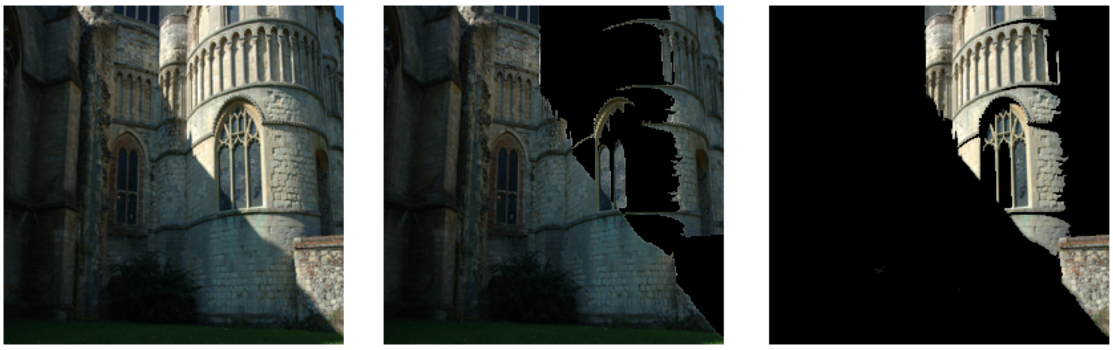


Figure 4.12: *The original image and its segmentation according to the binary mask obtained with the chromagenic illuminant detection. Both the shadow and lit parts of the image are accurately segmented.*

Results for various situations of indoor and outdoor lighting are shown in Fig. 4.13. For every image we used meanshift to segment the images using its standard parameters and a minimum region size of 0.5% the size of the image. Despite some minor accuracies, the illuminations are well separated. One of the main strengths of this method is its ability to detect occlusion shadows even when all of the region edges are coincidental material and shadow edges, a major improvement over gradient or region comparison methods that generally assume the reflectances on both sides of a shadow edge are identical.

4.4 $\mathbf{m \neq 2}$

When two illuminants are present in an image and the detection algorithm is constrained to discriminate them, we saw that this discrimination was accurate. It is however difficult to infer the number of illuminants present in an image *a priori*. First, we want to analyze the algorithm's behavior when a single illuminant is present but we try to find two.

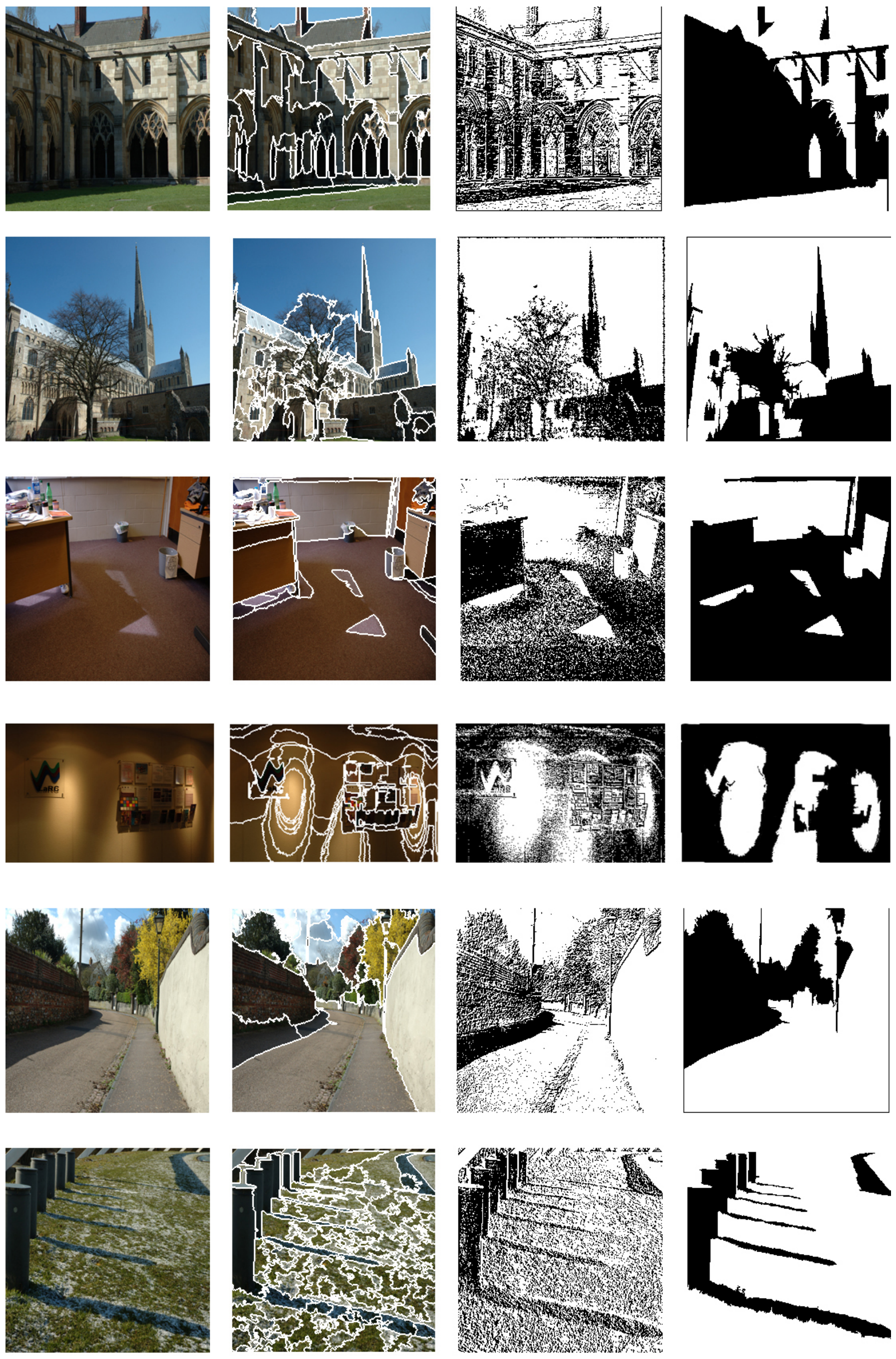


Figure 4.13: *Columns from left to right: the original images, the meanshift segmentation, the pixel-based classification, and the region-based classification .*

In essence, we run the exact same test as in the previous section, but on images that contain only a single illuminant. The results (Fig. 4.14) show that, while the algorithm is looking for the best pair of illuminants, the returned map is almost unitary. From these results, we infer that we can assume the maximum number of lights in the image to be higher than in reality and still obtain accurate results.

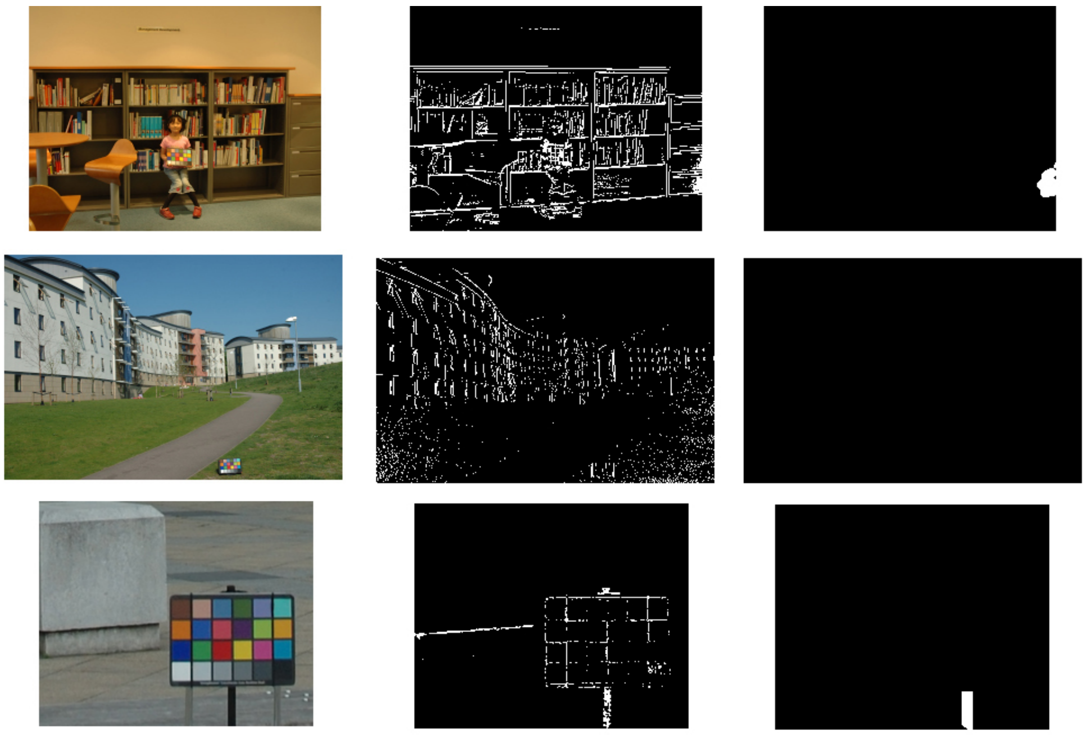


Figure 4.14: *columns from left to right, 1) the original images containing a single illuminant; 2) the pixel-based pair classification: the map is almost unitary but for some minor noise; 3) the results using region processing: again, almost all regions are labelled using a single illuminant.*

For completeness, we now address the case where more than two illuminants are present. While theoretically possible, the number of different mappings for $m = 3$ and 4 is $105,995$ and $2.2\ 10^6$ respectively (for our set of 87 transforms). Also, in natural images, there are few cases where more than three main, spectrally different, lights are present. The case of three illuminants can usually be put down to the presence of two distinct lights plus the mixture of those lights (such as the penumbra in shadows where the transition is not immediate).

We illustrate the three lights case on both reflectances images (where the lights are distinct) and on real images (where the distinction is blurred). For real images, we look at the difference between assuming the presence of either two or three illuminants. The results (Fig. 4.15) show that the detection, in the synthetic case, is as accurate as when two illuminants are sought. On real images, we see that the transitory regions are picked up as different illuminants but the improvement does not necessarily justify the increase in complexity.

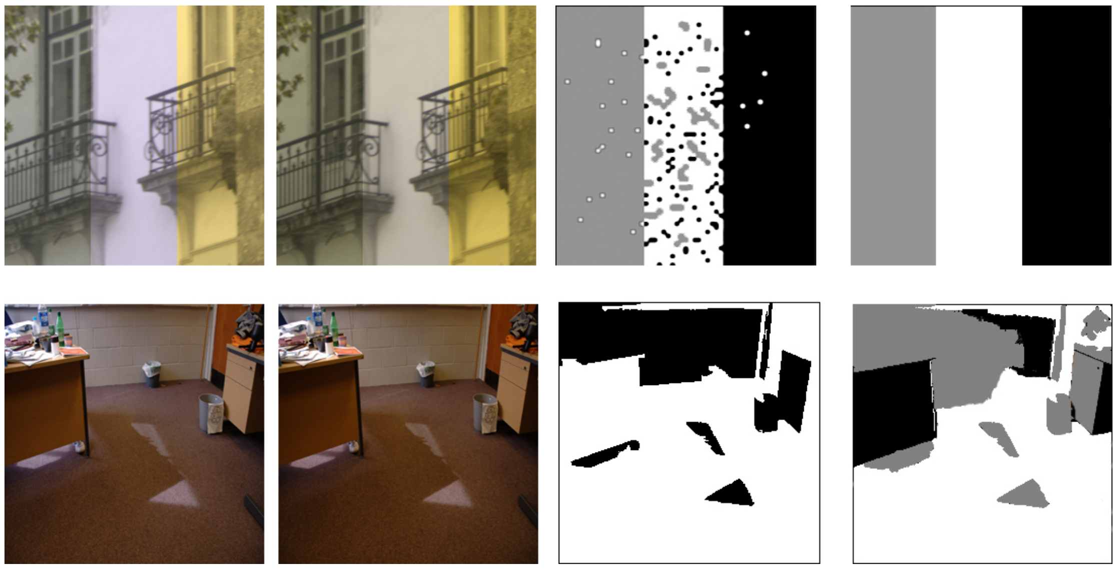


Figure 4.15: *Detection assuming three illuminants. Top row: synthetic images and results from both pixel and region classification. Bottom row: indoor scene with two main illuminants. Assuming two illuminants sperate them (3rd image). Assuming three illuminants enables to refine the classification depending on the relative proportion of the two illuminants in the scene.*

We point out that, for illumination detection, the number of transforms $\mathcal{T}_i$ can be reduced. Indeed, since we are interested in finding out illumination difference irrespectively of the accuracy of illuminant estimation, we can limit the number of transforms used so that they still cover the possible gamut of lights but with a coarser sampling.

This can be achieved, for example, by looking at the images in the training set and manually selecting a single transform per illuminant class (i.e., keeping only a single sunlit image to represent the whole class of sunlight illuminants). Alternatively, we obtained similar results to the ones presented throughout this section by plotting the mapped 2D chromaticities of a white patch with all the transforms and selecting a smaller number (15 out of 87) using the k-means algorithm to form clusters.

4.5 Conclusion

In this chapter we have shown that the shadow edge detection method presented in [FDL04] could be improved upon to complete the edges and to incorporate the possibility of spatially varying shadows. Both those improvements translate in significantly better shadow removal possibilities.

We also have introduced a new method to detect multiple illuminants in images based on the chromagenic theory. By forcing the lights to be examined pair-wise without preoccupying ourselves about the accuracy of the illuminant estimation and processing the results on a region instead of a pixel basis, we obtained very accurate results for a variety of illuminations. We went on to show that this method does not require prior knowledge of the number of lights in the image and were also able, when more than two illuminants were assumed, to detect shadow penumbra.

Chapter 5

Removing Shadows

In this chapter, we investigate the problem of shadow removal. To do so, we assume throughout this chapter that shadows have been detected (e.g., using one of the methods from Chapter 4).

We start by presenting a 1-dimensional method to remove shadows and analyze the behavior of both our 1D and the 2D (proposed in [FDL04] and reviewed in Chapter 2) reintegration methods. We go on to provide a robust framework to remove shadows using random Hamiltonian paths and develop an algorithm to generate paths, over a special class of grid graphs, in linear time. Results on real images show that our 1D robust method provides significantly better results than a simple 2D reintegration.

Finally, when the conditions of shadow formation are known (i.e., we know which type of shadow is present in the image), we provide a simple method to remove shadows with minimal inference by simply adding a constant to the logarithm of the colour image. Results show that, with known conditions, this method can provide results that are as accurate as the 1D method.

5.1 1D Shadow Removal

To process an image I in one dimension, we will use a path p that visits every pixel once and once only. This ensures that the integration is well posed: we recover one brightness value from one derivative per pixel. From [FHLD06] we know that, if the shadow boundaries are given, a shadow-free image can be obtain by setting the image derivatives to zero at the shadow boundaries and then reintegrating the image in log space. Throughout this chapter, we will tacitly assume that the image is first transformed in log space, where all the calculations are done. The resulting shadow-free image is then exponentiated prior to display.

For the sake of simplicity, let da denote the derivatives of I along p in either direction, $a = x$ or $a = y$. I is integrated, in standard calculus notation:

$$I + c = \int_p \frac{dI}{da} da \tag{5.1}$$

with an unknown integration constant c, using the appropriate derivative direction (dx or dy) as the path is swept out. Starting the reintegration at a non-shadow pixel, e.g., one of the brightest pixels, which we denote p_1, we can uniquely determine c and obtain a correct shadow-free image I'. Let p_i be the i^{th} pixel visited along p; the path-based integration becomes:

$$I'_{p_1} = I_{p_1} \tag{5.2}$$

$$I'_{p_i} = I'_{p_{i-1}} + T(\nabla I)_{p_i} \tag{5.3}$$

Where $T(\nabla I)$ is defined as

$$T(\nabla I) = \begin{cases} 0, & \text{for } |\nabla I| \in S & (5.4a) \\ \dfrac{dI}{dx}, & \text{if } \overrightarrow{p_{i-1}p_i} \text{ in the } x \text{ direction} & (5.4b) \\ \dfrac{dI}{dy}, & \text{if } \overrightarrow{p_{i-1}p_i} \text{ in the } y \text{ direction} & (5.4c) \end{cases}$$

Where S represents the location of the shadow edges.

From a complexity point of view, the integration problem is reduced to a series of sums with no boundary conditions to consider.

Let us now consider how we can generate paths that visit all pixels once and once only. We interpret I as a grid graph of size $n \times m$ where each pixel is a node and where edges are assigned on a 4-neighborhood basis: left, right, up and down. Finding a path p that visits all pixels once and once only then amounts to finding a Hamiltonian path over the graph.

The general problem of finding Hamiltonian paths in a general graph is NP-complete [GJ90], i.e., its complexity varies exponentially with the size of the input. Even for the sub-class spanned by general grid-graphs, this complexity doesn't change [IPS82]. However, since the simple 1D approach works on complete rectangular grid graphs (a sub-sub-class), there are certain available easy paths; for example, raster and fractal type paths such as Peano curves [Man82] among others, illustrated in Fig. 5.1. Using such paths and thresholding the image gradients at the shadow edge locations enables the reintegration of shadow-free images. However, in practical situations we will not always correctly estimate where a shadow edge is, and incorrect estimates will lead to errors in reintegration. Moreover, regular paths tend to propagate regular errors. As an example: using a raster path often results in banding errors in the reintegrated image (see for example in Fig. 5.2). While integrating the same image multiple times and averaging the results can help, it serves only to diminish, not always by much, the reintegration errors.

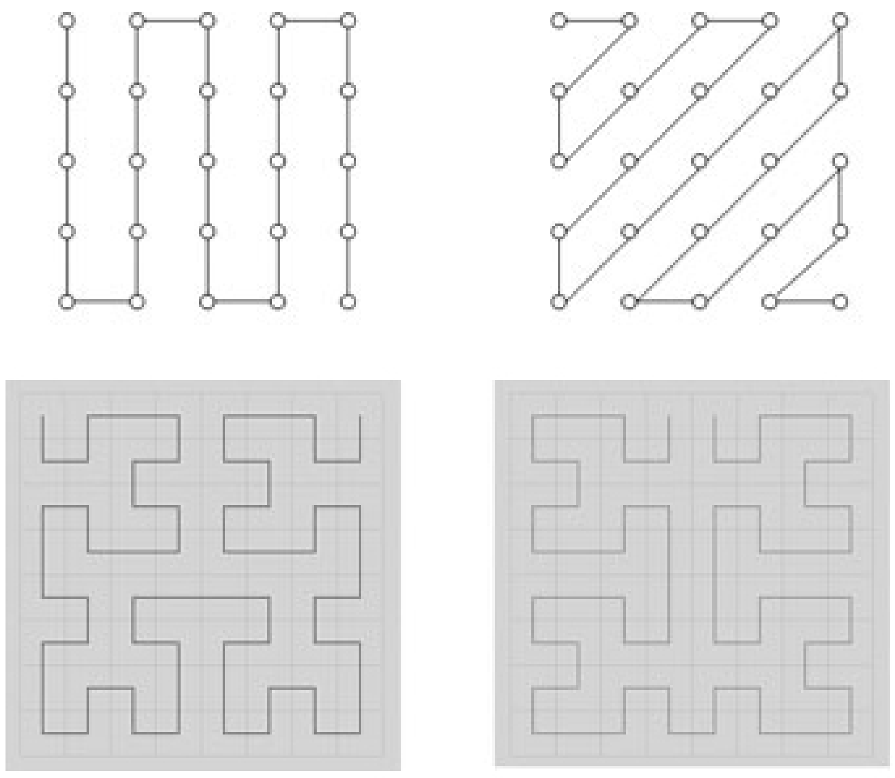


Figure 5.1: *Examples of Hamiltonian paths on grid graphs. The 2 top images are raster paths and the 2 bottom ones are example of Peano curves obtained with different generators.*

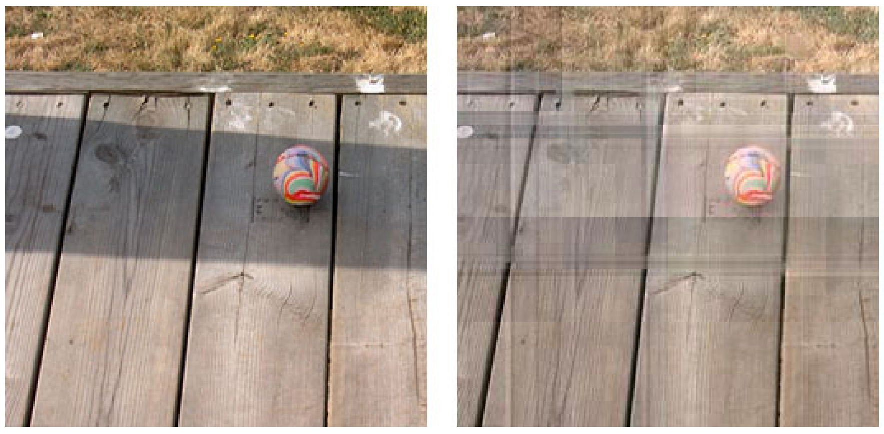


Figure 5.2: *The original image (left) and its shadow-free 1D reintegration using the average reintegration of the two raster paths and Peano curves (right). One can notice that even though the image is shadow-free, many regular artifacts are created during the reintegration.*

5.2 Analysis of Reintegration

Let us first consider the case where we have perfect shadow masks. In this case, for both the 1D or 2D situations, we set derivatives under the shadow edges to zero and then reintegrate. This ‘setting to zero’ results in an image that is non integrable in two dimensions: there does not exist an image with the x and y derivatives we have after the masking step. To solve this non-integrability problem we seek a least-squares solution and solve for the image whose x and y derivatives are as close as possible, in the least squares sense, to the derivatives we seek to integrate by inverting the associated Poisson equation. Unfortunately, the best least-squares solution tends to return an image where the non integrability caused by the shadow edge can result in errors propagated across large sections of the image. 2D reintegration is a global procedure. In the 1D case, if we use Hamiltonian paths, then the problem is well-posed, there are no integrability problems and the boundary conditions are easily determined.

A more subtle problem is the issue of coincident shadow and material edge boundaries. In masking out the shadow edge, there is an implicit assumption that both sides of the shadow correspond to the same surface. This, though not frequently, can be violated in images. Again, the issue manifests itself as potentially quite large errors during reintegration.

We illustrate these ideas in Fig. 5.3. This figure can be used as ground truth since the desired result is exactly known -the shadow free image we start with. The first image shown is an artificial image composed of small gradients and a couple of step edges (large gradients). The superimposed shadow region, shown as the ellipse in black overlay, contains a step edge while most of the shadow boundaries are laid over small gradients. The image on the bottom left shows the reintegration using the Poisson method when a perfect shadow mask is used. Clearly, the result is shadow free but the image colour is incorrectly estimated. In the bottom right, we look at the 1D reintegration using a raster type path. We again see a failure of reintegration. Moreover, the shape of

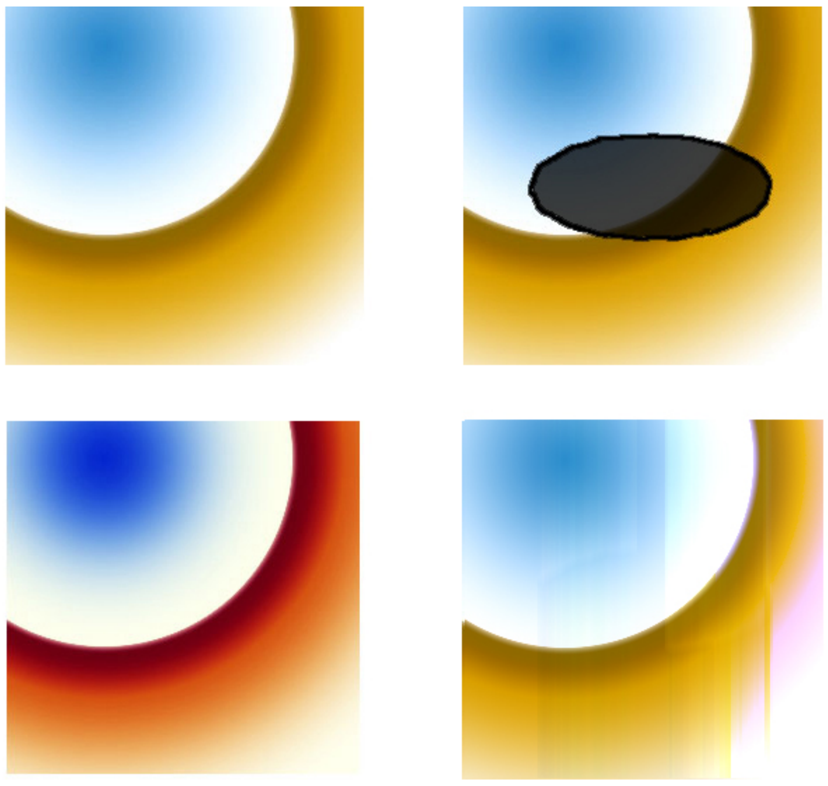


Figure 5.3: *The original image (top left) with the artificial shadow region in black overlay (top right); the 2D reintegration (bottom left) where the image colour is altered, and the 1D reintegration with a raster path (bottom right) where the regular propagation errors can be seen.*

the integrating path is clearly seen. The main problem here is the assumption that there are no coincidental material and shadow changes. This assumption is violated and so we propagate incorrect brightnesses across the edge.

It is worth making a couple of additional remarks about Fig. 5.3: first, this is the performance for a perfect shadow mask. Second, the left two thirds of the 1D reintegrated image (bottom right) are almost acceptable. Here the raster reintegration is working fine until we reach a coincident shadow and material edge. This gives us a hint on how to proceed. If we can manage the reintegration and *statistically* choose to avoid problem edges, then we can use the 1D path approach to reintegrate the images. In contrast, there seems to be no easy solution to the 2D reintegration problem: the *whole* of the 2D reintegration is affected by error.

5.3 Making 1D reintegration work

Let us begin by considering in detail the sources of error we might encounter during path based reintegration. The creation and propagation of artifacts is illustrated in Fig. 5.4. The x-axis of the graphs represents the path progression, while the y-axis represents pixel intensity. Fig. 5.4a displays the ideal case where the non-shadow parts of the image are preserved and the shadow is effectively removed. If the shadow mask is not closed, a path can enter the shadow region through a detected shadow edge but then exit it through a "hole" in the edge map (this is the main reason we closed the edge maps in Chapter 4). Such a case is shown in Fig. 5.4b where the resulting high error is clear.

Lastly, errors are also caused when a material edge coincides with a shadow edge or, equally, when noise is present. Here the problem is that one assumes that both sides of the shadow edge are "continuous", i.e., they would be the same given identical lighting conditions, even though they might not be. While this assumption generally holds, it is, and must be, sometimes violated in actual images. Fig. 5.4c shows the case where noise is present at the exit of the shadow region; the thresholded gradient does not take the noise into account and errors result. The profile in Fig. 5.4c is also indicative of a coincident shadow/material boundary.

While these errors are cause for concern, we draw the reader's attention back to the 1D reintegration shown in Fig. 5.3. There it is clear we did well at the start but then encountered an error (about one third of the way through) that was propagated throughout the rest of the image. The problem occurred, in this case, where we had a coincident shadow and "material" boundary. Let us imagine that, instead of taking a raster path, we reintegrate along a path which, by design, enters and exits the shadow region only once and that the entry and exit are located where no error is present. The

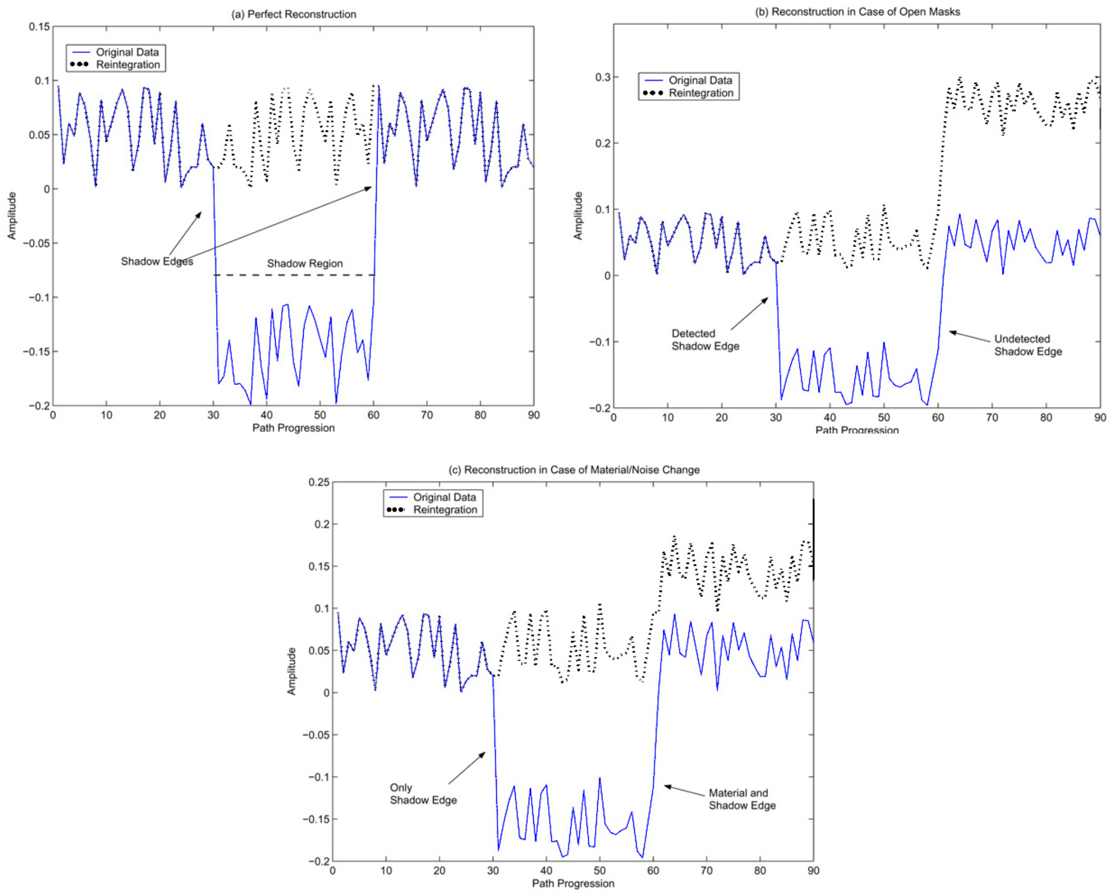


Figure 5.4: *Graph representation of the shadow regions of an image. (a) a perfect (supposed) reintegration. (b) Reintegration with errors due to an imperfect shadow mask. The entry in the shadow region is well detected but the exit is not, thus creating a large error. (c) The noise/material edges case. The shadow edges are well detected, but the assumption of similarity is not enforced. Note how an error created at a point p_1 in time is still propagated throughout all the subsequent pixels visited $p_i, i > 1$.*

result is shown in Fig. 5.5(left). We see that we have a perfect reintegration (note the black line is the shadow mask which is not reintegrated here because we crossed the shadow edge once). To obtain Fig. 5.5(right), we removed the remaining shadow edge by inpainting. This simple example shows how we can reintegrate in one dimension without error as long as we choose entry and exit points where there is no error. But, how can we achieve this reintegration for real images where we do not know where the sources of error are?

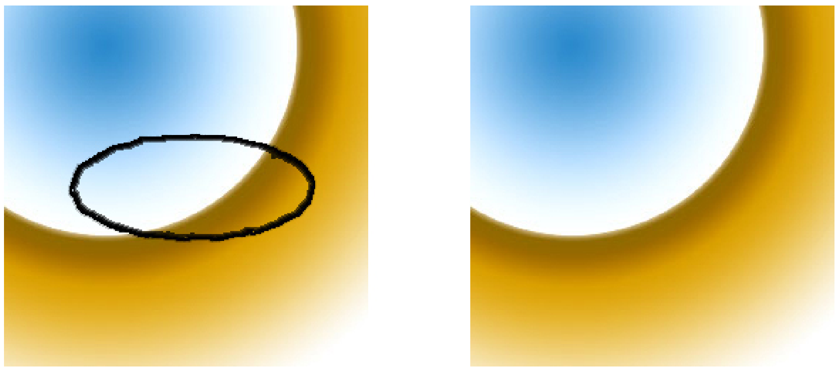


Figure 5.5: *Robust 1D reintegration. The left figure is the direct output of the reintegration. Most of the shadow mask is still present since the path crosses it only once. The right figure is the result after inpainting the shadow mask.*

The solution here is to consider what happens *statistically* during reintegration. Let p_e denote the probability of encountering either noise or a coincidental luminance/material edge (i.e., the probability of committing an error in the reintegration). If we enter and exit a shadow region N times, such as is done using one of the paths of Fig. 5.1, then the probability of at least one error being propagated is $1-(1-p_e)^N$, which tends to 1 when N is large. If, however, by design we only enter and exit the shadow region once, the probability of error propagation is p_e (for $N > 1, p_e \ll 1-(1-p_e)^N$). So, if we enter and exit a shadow once it is unlikely that we will commit an error. But, this said, we might still be unlucky.

Now, suppose that we reintegrate an image several times, and each time use a different path that enters and exits the shadow at a different place. If α is the number of paths, the probability of all the paths being corrupted becomes p_e^α (which is almost always close to zero). If p_e is 0.5 (which is much higher than can possibly occur) and $\alpha = 8$ then the chance that all 8 paths are corrupted is $1/256$.

The statistical argument set forth above notwithstanding, some errors will exist. Empirically, we find that the trajectory of the path can show up in the reintegrated image if there is an error. We therefore wish to generate non regular paths, in order to limit the conspicuity of errors (see [Wan95] for discussion of how the human visual system is sensitive to regular patterns).

In summary, in order to improve 1D reintegration we propose three modifications to the basic 1D algorithm (the closing of shadow edges has been addressed in the previous chapter):

- Constraining the path structure so that there is a single crossing of the shadow edge (illustrated in Fig. 5.6). This minimizes the likelihood of encountering either noise or a material change at the location of the crossing.
- Randomizing the path structure in order to minimize the visibility of artifacts.
- Inpainting the uncrossed parts of the shadow edges

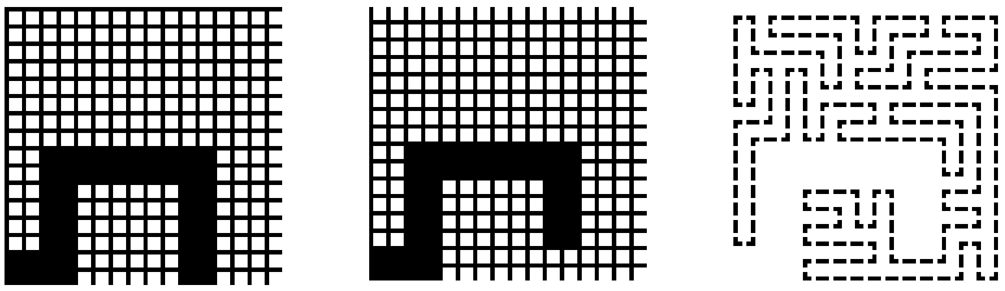


Figure 5.6: *A shadow edge (in black) where we allow one opening (left & middle) and a random Hamiltonian cycle over the image visiting every valid pixel. It can be noticed that the simple paths previously shown are not usable for such a graph.*

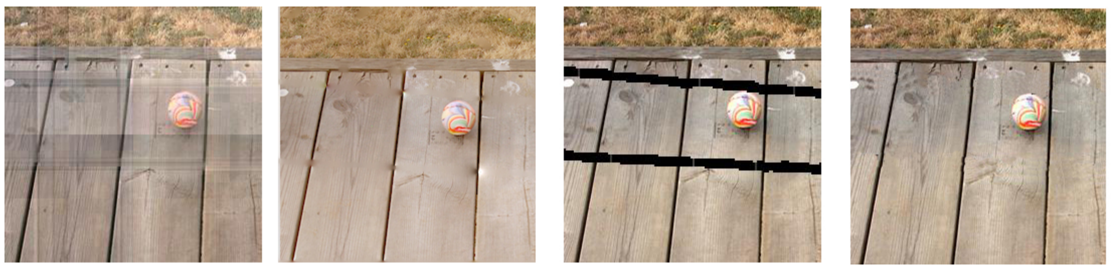


Figure 5.7: *From left to right: Results of shadow removal with the original 1D formulation, the 2D integration method, the robust 1D method outlined above without and with inpainting.*

Figure 5.7 exhibits all type of shadow removal discussed so far: 1D with regular paths, the 1D method outlined above and the 2D method. In the robust 1D method, since a single opening in the shadow mask is allowed, so the image is not fully reintegrated. The boundary can however be removed by way of inpainting.

We first look at all the necessary steps to achieve a simple yet robust framework for shadow removal. We develop a method to obtain random Hamiltonian paths in a linear time; a coding friendly method is presented in Appendix A. We then show that more stable results can be achieved through averaging the outcome of several paths and we propose a metric to assess the reintegration quality.

5.4 Random Hamiltonian Paths

In our earlier discussion, we proposed that paths should be random in nature and that they should be Hamiltonian, i.e. they should enter and exit shadow regions only once. In [IPS82] it has been shown that, for a general grid graph, finding a Hamiltonian path is an NP-complete problem. There exist, however, special classes of grid graphs for which such paths can be found in a polynomial time [CST02, IPS82, Joh02, MS93, Adh01, UL97, Sab91]. Our contribution here is twofold as we propose both the class for which we can find a Hamiltonian cycle as well as an easy to implement algorithm to obtain the cycle in a linear time; the complexity of the algorithm is $O(N)$ where N is the number of pixels in the image.

We propose that this process can be achieved in 3 steps: initialization of the path, adding constraints (the once-in-once-out entry to shadows), and construction, where the path is returned.

5.4.1 Initialization

The necessary and sufficient condition for our algorithm to output a Hamiltonian path in a linear time is that the graph has to be complete when downsampled by a factor of two.

Downsampling a graph by a factor of two consists of dividing its height and width by two. Doing this, one effectively merges four vertices together while keeping the existing edges between these groups of four vertices.

Complete is used to describe a graph where all the downsampled vertices were composed of vertices of a single type: valid or non-valid; a vertex will be deemed non-valid if it belongs to the shadow edge. Both properties are illustrated in Fig. 5.8.

Since the shadow edges can assume any shape, the completeness property is generally not met. In practice, we enforce this property using a majority voting procedure such as the one illustrated in Fig. 5.9. This method ensures that a complete graph is obtained while minimizing the modifications of the shadow mask.

Let I be an $n \times m$ image where the shadow mask has been found and whose graph representation is complete. A Hamiltonian cycle over the valid pixels of I can be found in a linear time.

In the graph representation of I, all valid pixels are represented as vertices and all non valid ones (the shadow mask pixels) are represented as holes in the graph. Let G be the graph representation of I and G_R the $\frac{n}{2} \times \frac{m}{2}$ graph obtained by downsampling G by a factor 2.

5.4.2 Constraining Openings

Since the shadow mask is closed and composed of only non-valid pixels, G and G_R will have at least two distinct components. It follows that, in order to compute a path, the graph must first be connected; we will here connect G_R. To do so, let us consider

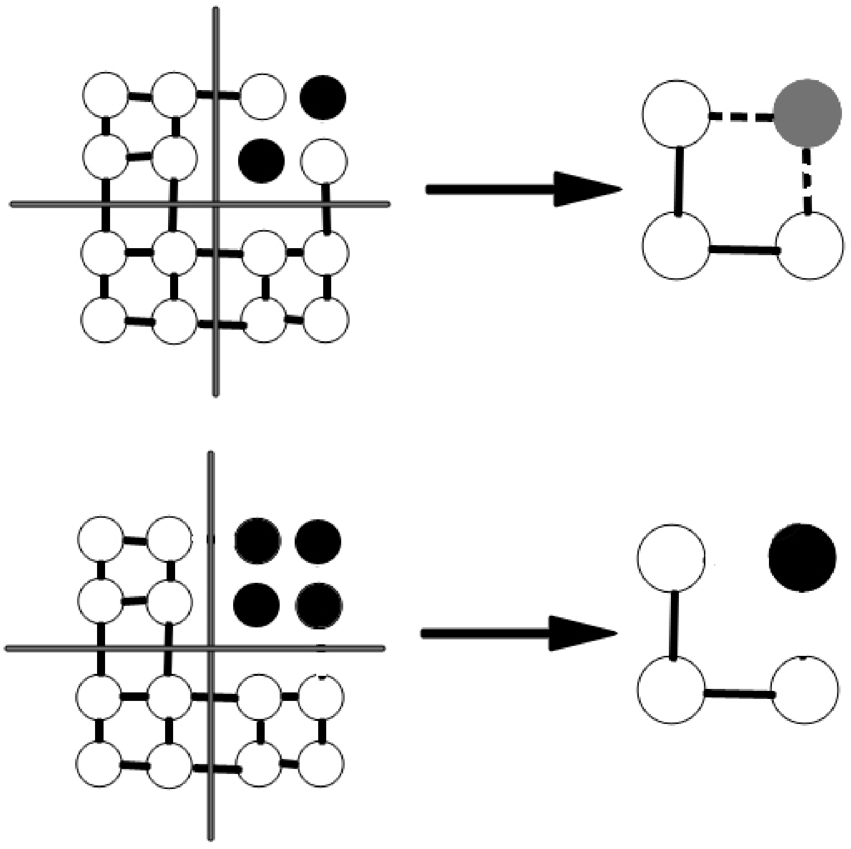


Figure 5.8: *An incomplete grid graph (top left) and its downsampled representation (top right). The gray node represents a partial node since it is obtained from nodes having different types. The bottom graphs illustrate a complete grid graph and its downsampled representation.*

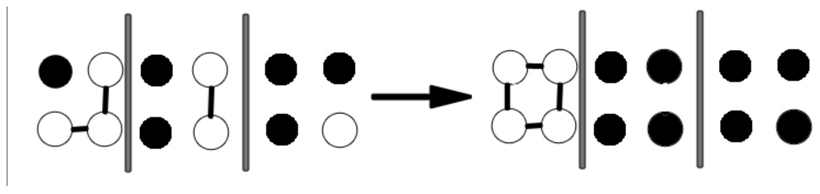


Figure 5.9: *A graph containing partial nodes and its complete version. Partial nodes are transformed according to majority voting.*

the case of a two components graph, i.e., the image contains one shadow and one non-shadow region. The graph is connected by first labelling every node on one side of the edge with s and every node on the other side with t. An s vertex is selected at random and the shortest path from it to any t vertex is computed. The edges and vertices visited by this $s - t$ path are "validated" and thus the graph is then connected. An illustration of this procedure is shown in Fig. 5.10. The shortest path maximizes the chances to enforce continuity since the "travel" through the shadow edge is minimized. If G_R admits $k \geq 2$ distinct components, the connecting procedure is repeated $k - 1$ times.

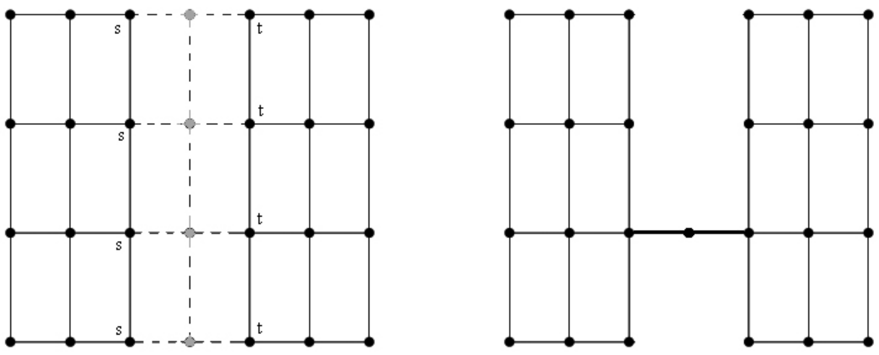


Figure 5.10: *Left: The graph at the start of the st procedure. The dashed edges and gray vertices represent the possibilities of going from an s to a t vertex. On the right: the outcome. An s vertex has randomly been chosen and the shortest way to a t vertex is computed with all edges having a unitary weight.*

5.4.3 Construction

Once G_R is connected, we proceed by finding its minimum spanning tree, T_R, using the method of Krager et al. [KKT95] that has a linear time complexity. Since at first all the edges in G_R have unitary weight and the path structure will closely resemble the one of T_R, we weigh the edges of G_R with random weights prior to computing T_R. This ensures the resulting path will have a suitable random structure.

An Hamiltonian cycle over G is then found by: forming a cycle over T_R, upsampling T_R by a factor of 2 to obtain T, a partial version of G and finally, completing T so that

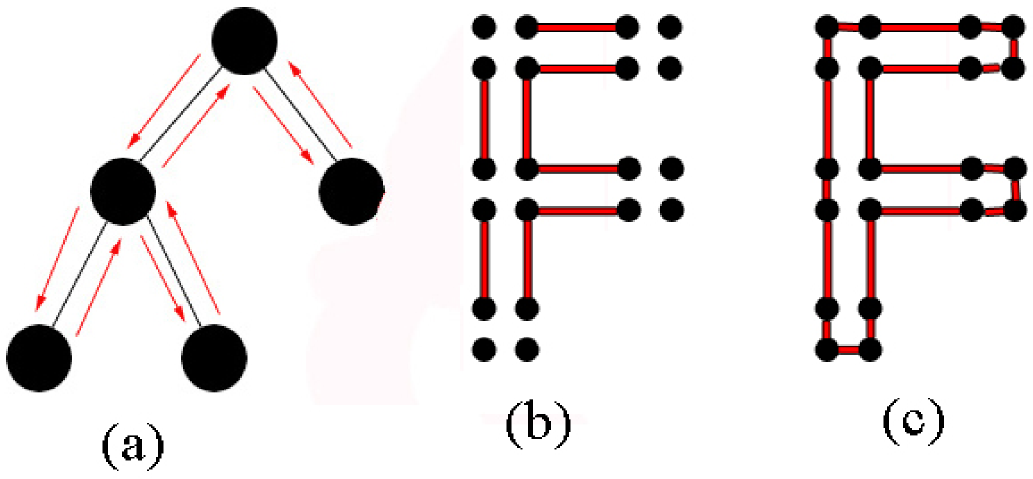


Figure 5.11: *(a) The spanning tree T_R and its walk around; (b) T, the upsampled version of T_R with the existing edges being upsampled as well. (c) The Hamiltonian cycle found by our algorithm on the original graph.*

it forms a Hamiltonian cycle over G.

The cycle over the spanning tree is formed by walking around T_R. Starting at the root, T_R is explored, in a depth first search manner, until all its edges have been visited twice. This yields the order in which the vertices will be visited in the final path. This procedure is illustrated in Fig. 5.11.

Upsampling a graph by a factor of two is the dual operation of downsampling. We first double the width and height of the graph, thus quadrupling the number of vertices. Existing edges are then reinstated. To preserve the graph structure, the edges are doubled so that every group of four vertices has the same connectivity to its neighbors as in the downsampled version. The process of upsampling can be seen in Fig. 5.11a and 5.11b.

Because of the down/upsampling procedures, some information (here edges) has been lost and thus T is incomplete, as shown in Fig. 5.11b. We therefore need to complete T in such a way that it forms an Hamiltonian cycle. To admit such a cycle, T needs to fulfill two properties: every vertex has degree two and the cycle must span G (there should be no sub-cycles).

Using the geometrical structure of grid graphs, we know that each vertex of G has a degree of *at most* four. This limitation allow us to prove by exhaustion that an Hamiltonian cycle over G can always be found. In T, depending on the structure of the four-vertices groups, there are five distinct patterns (rotations notwithstanding) that can be used to complete the edges while ensuring that each vertex has degree two and that no sub-cycles are created. The complete set of patterns is shown in Fig. 5.12.

While this algorithm comprises several steps, its complexity is nonetheless linear. Down and upsampling are both linear operations and the minimum spanning tree can also be found in a linear time [KKT95]. The completion step is linear since for every 4-vertices group there are at most five comparisons in order to determine which completion pattern to apply. Therefore, our algorithm ensures that we will find a random Hamiltonian cycle over G with a linear time complexity. Our C++ implementation of the algorithm, shown in Appendix A, takes 0.5 seconds to calculate a path on a 1024$\times$1024 image.

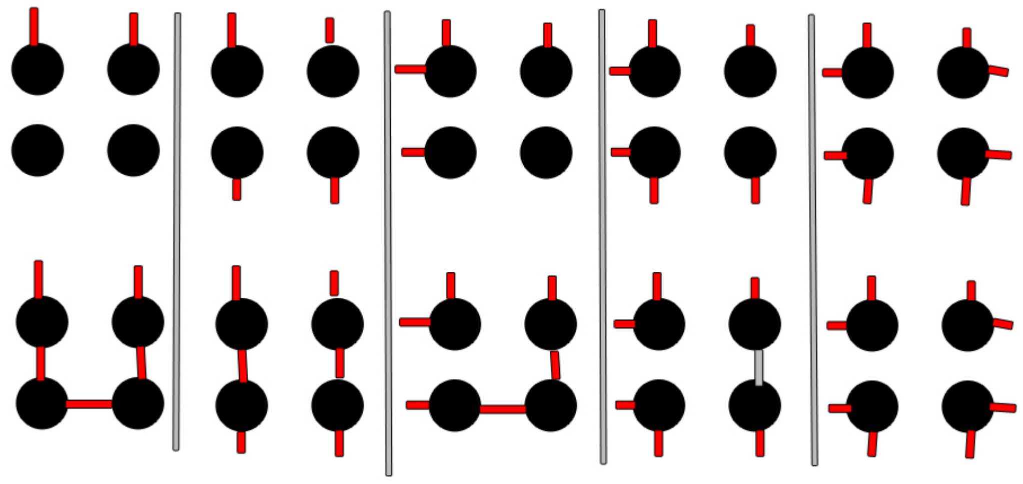


Figure 5.12: *The 5 different cases to complete the missing edges in order to ensure that no node has a degree other than two and that all nodes are connected. The "outside edges" are created during the upsampling procedure and the number of edges to add therefore depends on the degree of the equivalent node in T. Note that these are the only possible cases that can occur. It follows that it is always possible to obtain a Hamiltonian cycle with this method.*

5.5 Finalizing the Integration

Let p_e be the probability of committing an error because of noise or a material change while crossing the shadow edge when the shadow edge is entered and exited once. General Hamiltonian paths proposed in [FF04] cross the shadow edge N times. They therefore commit an error with a probability of $(1 - (1 - p_e)^N)$. Using the random Hamiltonian paths previously described, we can reduce this probability to p_e.

While this approach minimizes the chances of erroneous reintegration, it does not by itself guarantee an artifact free integration. To ensure that minimal errors will be committed, we proceed as follows.

We start by generating n different random paths and integrate along them to obtain n shadow-free images $I_1, \ldots, I_n$. If p_e is the probability of error for one image, then the probability that all images are significantly corrupted (contain very visible artifacts) is p_e^n. In practice, using $n = 10 - 15$ always yielded at least one almost error free image.

From those n images, we therefore have to choose which ones are free from artifacts. To do so, we first separate all reintegrated images in three regions; this is illustrated in Fig. 5.13. The first region, R_1, is composed of the pixels between the start of the integration and the first entry into a shadow region. By definition, all of $I_1, \ldots, I_n$ display a perfect reintegration in this region (i.e., their values are the same as the ones of I, the original image). The second region, R_2 comprises pixels that are located within the shadow regions of I. Since there is no indication as to what values those pixels *should* have, they contain no useful information to distinguish the reintegrated images. The last part, R_3 is composed of the pixels that are in shadowless parts of I and that are located, path-wise, after exiting at least one shadow region. Here, we have a reference value as to what those pixels should be (i.e., the values of I). We can therefore devise a reference metric, Δ_{RGB} to find out how much has an image been wrongly modified during reintegration. Since errors are propagated, we weight the distance according to the size of R_3.

Let M be the number of pixels in R_3 and I' be a reintegrated image. We calculate Δ_{RGB} between I and I' as follows:

$$\Delta_{RGB} = \frac{\sqrt{(R_I - R_{I'})^2 + (G_I - G_{I'})^2 + (B_I - B_{I'})^2}}{M} \tag{5.5}$$

Among the n images, we are then able to select the ones that are the closest to the original image in the non-shadow regions, thereby ensuring that a good reintegration has been performed. In practice, we select the four images that have the lowest Δ_{RGB} and average them to further minimize the visibility of artifacts that might remain.

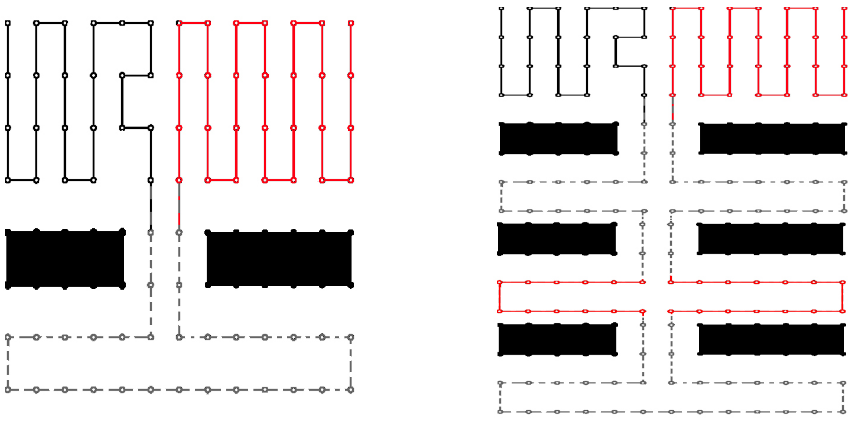


Figure 5.13: *The solid black line is the path prior to its entry in the shadow region (R_1) (symbolized here with thick black edges), the reintegration along that part of the path is perfect and not considered. The dashed line is the portion of the path that lies within a shadow region (R_2) The solid red line is the part of the path that we can compare to the original image (R_3) and measure the errors that may have been committed.*

5.5.1 Addressing the Shadow Edges

The strength of the path-based approach is that we can ignore the shadow mask region (which is the major source of reintegration errors). Depending on the desired application, we may however want to consider that part of the image as well. As we know the location of the shadow edges and since they are a very small part of the image, applications such as recognition and indexing can proceed by discounting all information coming from that part of the image. In applications where a photographic-like output is desirable, one needs to render the shadow edges.

To do so generally requires the use of an inpainting technique. A widely used approach is the diffusion-based method of [BSCB00] that "regrows" information based on the boundaries of the region to inpaint and on the physics diffusion equation. The approach we chose here however is the texel-based method of Criminisi et al. [CPT04]. Our main reason to use this method is the quality of the results -see Fig. 5.14 for an example. It does, however, replace the target region with information already present in the original image; it is therefore closer to a synthetic approach such as cloning. Depending on the intended application for the shadow-free images, this inpainting step can either be omitted or replaced by another inpainting technique, without modification to the shadow detection-removal framework we proposed.

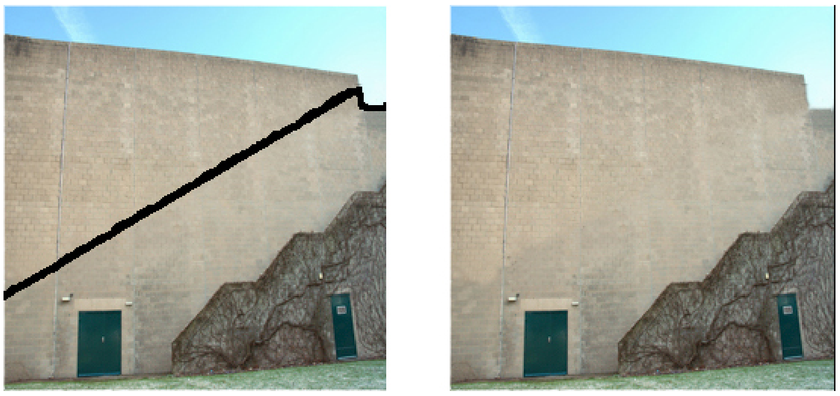


Figure 5.14: *An example of the inpainting procedure. The left image is the outcome of the reintegration procedure. The shadow mask, in black, is not visited by the path and is "missing information". We use inpainting to fill in the mask with information already present in the image.*

5.6 1D Reintegration Results

Let us begin by returning to the reintegrations illustrated in Fig. 5.3 and Fig. 5.7. We seek an objective measure for quantifying how well the original image function has been recovered. Let ∇I be the gradient of the non-shadow pixels of the original image I and, ∇I_{1D} and ∇I_{2D} be the gradients of the images reintegrated with the robust 1D and 2D methods respectively. We compute the error between those gradients, d_{1D} and d_{2D}, using the following

$$d_{1D} = \frac{|\nabla I - \nabla I_{1D}|}{\|\nabla I\|} \quad , \quad d_{2D} = \frac{|\nabla I - \nabla I_{2D}|}{\|\nabla I\|} \tag{5.6}$$

Since the 2D method solves the integration in the least squares sense, it is expected that the recovered derivatives will be globally close to the originals. The path-based method on the other hand recovers derivatives that are much closer to the original ones on a pixel by pixel comparison due to the locality of the procedure. Averaging over a number of images yields $d_{1D} = 3\%$ and $d_{2D} = 9\%$.

This result might seem counterintuitive since we have less error than a least-squares solution. However, the higher error in the 2D reintegration results from trying to recover an image which has zero derivatives over the entire shadow edge. Since this is generally not the case for the entire shadow boundary, errors are then created. The 1D method we develop here works better because it ignores the details (and derivatives) under the shadow edges.

5.6.1 General Images

Let us now consider reintegration results on real images. Fig. 5.15 shows results obtained over a variety of images; the 1D results are obtained by reintegrating with ten different paths and then averaging the output of the three best ones, with the method mentioned in the previous section. The first three images have been taken with a Nikon D70 camera with raw settings and then exported in 16 bits linear tiff. The fourth image has been taken with an HP Photosmart camera also exporting 16 bits linear tiff images. For both the detection and reintegration, no calibration nor pre-processing has been performed on the camera or the images themselves. On the results of Fig. 5.15, the first line shows the original shadow image. Note that among the images, the shadows vary in their location, size, shape and the background against which it is cast. The second line displays the results obtained with the 2D integration method proposed in [FDL04] where homogeneous Neumann boundary conditions were used. Finally, the third row depicts the shadow-free images obtained with our method.

Based on those results, one observes that the images obtained with our method are much more realistic. In many cases, one cannot guess that a shadow was ever there. Furthermore, the non-shadow part of the original image is exactly preserved, thus keeping the overall colorfulness and dynamic range of the scene.

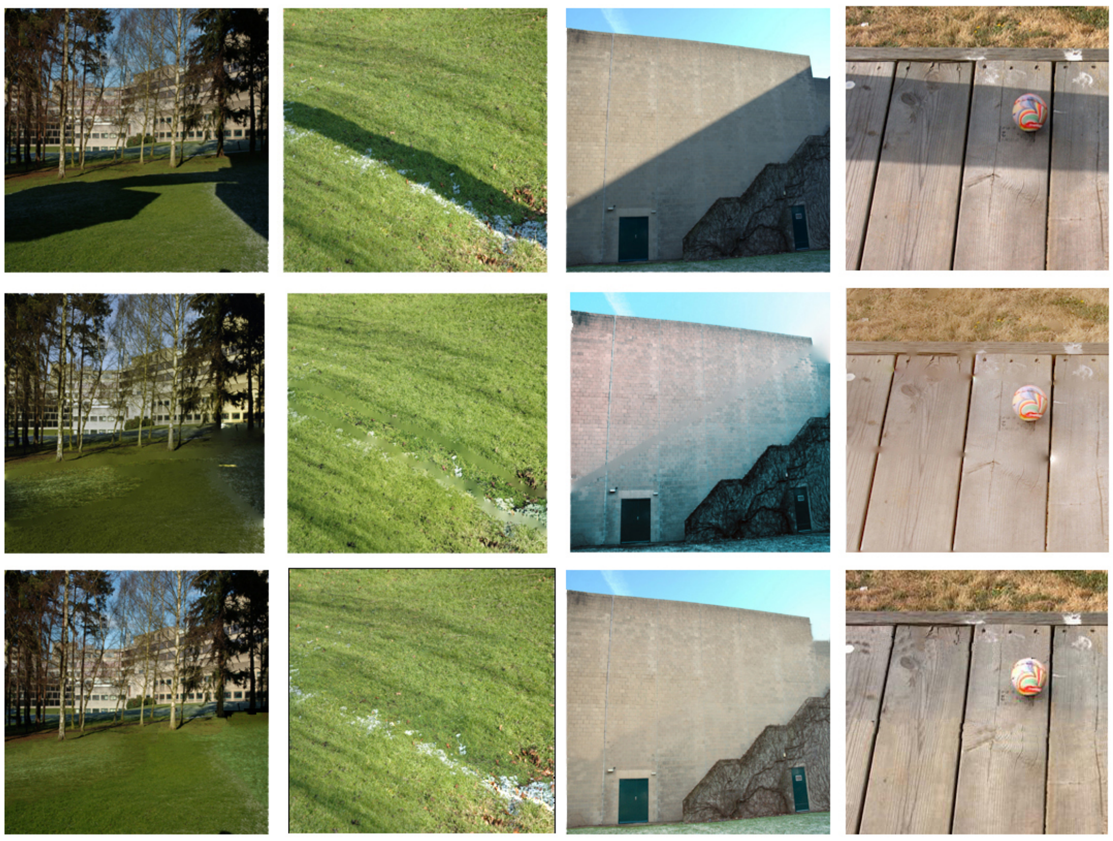


Figure 5.15: *Typical results from shadow removal. The second line are results obtained with the 2D method; the third line are results obtained with our path-based algorithm.*

Results on more complex scenes are shown in Fig. 5.16. These images are jpgs downloaded from the internet, the only precaution we have taken is to ensure they were not too altered ("photoshopped") so that the removal can still look realistic. The shadow masks are obtained using the method explained in Chapter 4.1.2 and the masks are the ones shown in Fig. 4.6.

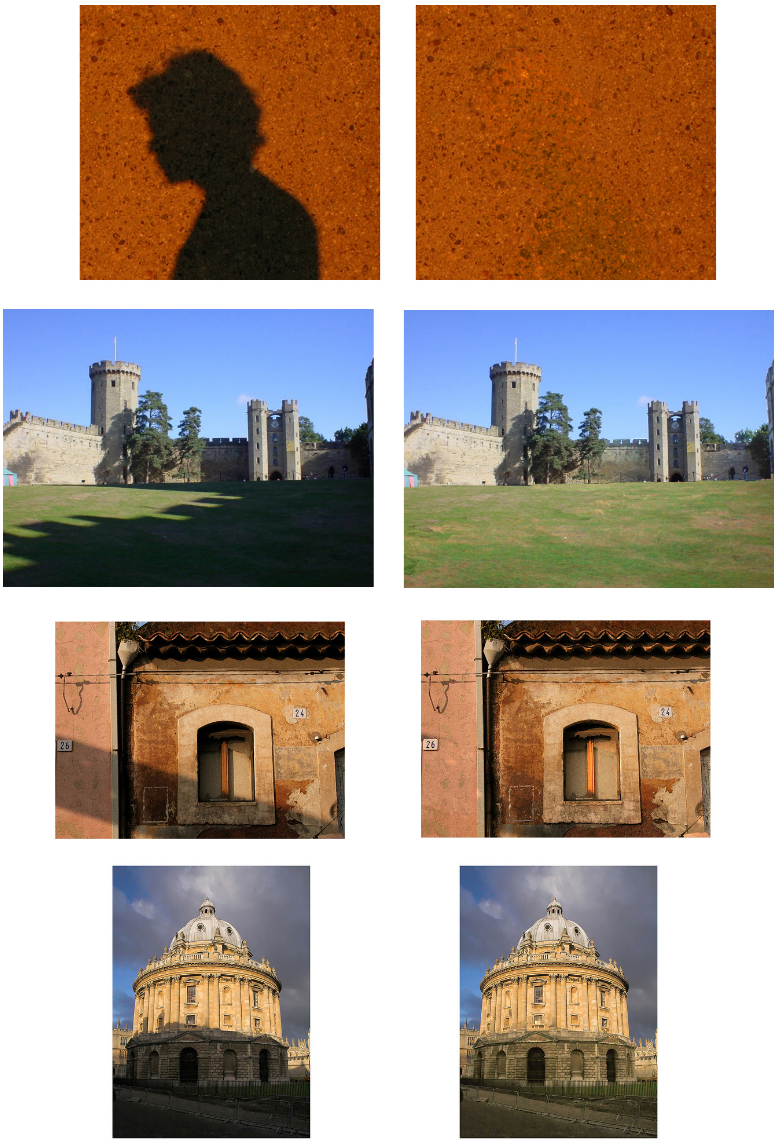


Figure 5.16: *Results of the 1D integration on jpg images where the shadows have been detected with the region method from Chapter 4.*

5.6.2 Specific Cases

While the proposed approach to shadow detection and removal yields high quality results on general natural scenes, we are also interested in more specific cases where the approach sometimes does not deliver as good results. The cases considered in this section are noisy images, gradually varying shadows, and blurred (soft) shadows.

Noise: When a picture is taken under difficult conditions, a digital camera will have a tendency towards generating coloured noise in the darker regions. As a consequence, the probability of committing an error will be higher and artifacts may occur. This phenomenon is illustrated in Fig 5.17 -first and second row-, where one can observe the amount of noise present in the original image as well as its influence on reintegration -in both the 2D and 1D cases. We can see that despite the noise, the shadows are greatly attenuated and few artifacts occur. The more noise, the greater the artifacts though.

Gradually Varying Shadows (GVS): The case of GVS -an example can be seen in Fig 5.17 third row-, both the detection and reintegration methods do not pick up the small shading gradients between adjacent pixels. As a result, the shadow region will be reintegrated uniformly and thus exact removal will not occur. The method does not, however, fail as a strong shadow attenuation will occur, significantly improving the image for further processing.

Soft Shadows: Soft shadows are typically created when an area light source is obscured by an object. The shadows strength and the blur of their edges can range from the almost unnoticeable to the very visible. There are two main issues when dealing with soft shadow removal: firstly, detecting them. Due to the general absence of strong edges as well as relatively small shading gradients, shadows are not always picked up by the segmentation algorithm (in our case the meanshift segmenter). Secondly, soft shadows usually also are GVS. The combination of those two factors make soft shadows both difficult to detect and to remove adequately. The results for our method are illustrated in Fig 5.17 -last row- where one can see that while the shadow(s) is significantly attenuated, lighting aberrations can occur.

When dealing with shadow removal what you detect is what you get. If a shadow region in the image goes undetected, it cannot be removed thereafter. Such undetected regions happen mostly in the case of soft shadows or in high dynamic range images. In HDR images, clipping in highlights and shadows often occur -an example of such clipping is in Fig 5.17, the shadow under the car. If a shadow region is clipped, it does not follow the laws of physics anymore and can therefore niether be detected nor removed.

5.7 Simple Shadow Removal

Finally, we propose a simple method that results in virtually error and shadow-free images in a very short time. Our approach is based on the insight that, when shadows have a constant strength within a region, shadow regions differ from their shadow-free counterparts by a constant scaling factor. The scaling factor can be found by minimizing the image differences across the shadow edge and by constraining this minimization to known shadow formation behavior.

5.7.1 Finding the Constant

Looking back at Fig. 5.6, one can see that once the shadow boundary is crossed no further modification of the image occurs. For a given opening and path, let P_1 be the last pixel visited by the path before crossing the shadow boundary and S_1 be the first pixel visited after the shadow boundary. What the 1D procedure does is to set $S_1 = P_1$ (the derivatives between those points are set to 0) and then reintegrates the shadow region using the original derivatives. This is therefore equivalent to adding a constant value $c = P_1 - S_1$ to the shadow region. While this is mathematically exact, it is however not possible to assess the correctness of c with respect to the problem -namely,

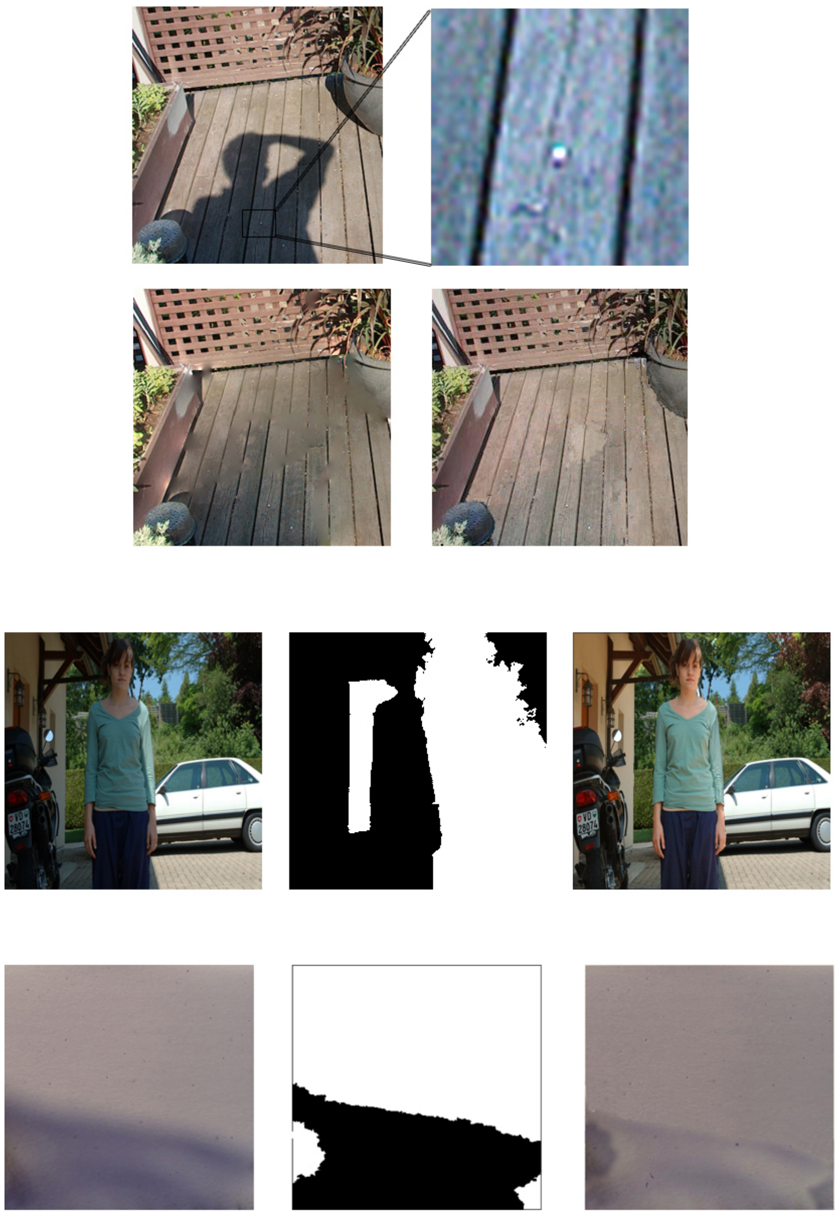


Figure 5.17: *1st row: Original image and illustration of coloured noise present in the shadow region. 2nd row: shadow removal with the 2D and 1D method on a noisy image. 3rd row: GVS image; the detected shadow mask -note the shadow under the car is undetected because of clipping; attenuated shadows obtained with our method. 4th row: soft shadow image; the detected shadow mask; shadow removed with our method. While the main shadow components are attenuated, lighting aberrations can occur.*

does it remove shadows?

Let us now consider what happens at the exit of the shadow region. Denote the last pixel visited in the shadow region by S_2 and the first pixel visited after exiting by P_2. By construction, after adding c, the value of S_2 becomes $S_2 + c$. Since the derivatives are also set to 0 when exiting the shadow region, P_2 is replaced by $S_2 + c$. Error due to noise, or a different relation between $\{P_1, S_1\}$ and $\{P_2, S_2\}$ can thus be assessed by

$$\text{error} = P_2 - (S_2 + c) = P_2 - (S_2 + P_1 - S_1) \tag{5.7}$$

A low error value is, however, not sufficient to validate the constant. A simple, and yet not uncommon, example of failure is the presence of sky at the shadow boundary -see Figure 5.19 -top left corner. Sky being a very smooth region, the associated error will be low, even though the constant will not be correct. The issue here is that there is a single point of failure, i.e., the constant is determined at a single location. A standard method is to find the constant c that minimizes errors in a least square sense. Let **P** be the array of pixels just outside the shadow edge and **S** be the array of pixels just inside the shadow edge, such as represented in Fig. 5.18 bottom left. Let us also assume that **P** and **S** have been sampled such that their lengths are equal. We then have

$$c = \min_a \|P - S + a\|^2 \tag{5.8}$$

In doing so, one however assumes that a (large) majority of the shadow boundary has no coincident material edges, which is a similar assumption to the 2D integration method previously presented. When this assumption is violated, significant errors can occur, as illustrated in Fig. 5.19.

To find an appropriate constant, we have to look at intrinsic properties of shadow to non-shadow transitions [RR82]. First, if there is a shadow boundary between two pixels that have near-equal reflectance, then in RGB space:

$$K_{\text{non-shadow}} > K_{\text{shadow}};\ K = \{R, G, B\} \tag{5.9}$$

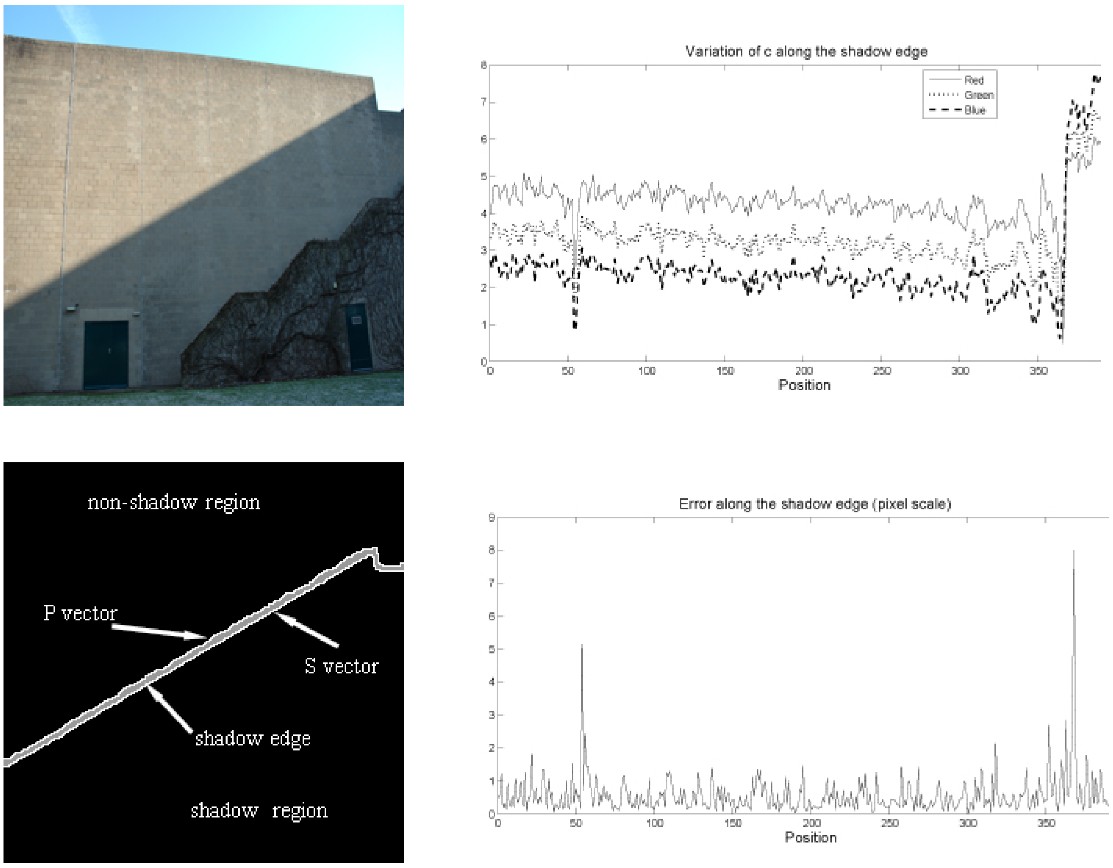


Figure 5.18: *Top left, the original image; top right, the evolution of the constant along the shadow edge. Note the change in c when the transition is between sky and shadows. Bottom left, illustration of the various parts of the shadow mask used in computing the constant. Bottom right, the errors induced by c along the shadow edge. While they are high when there is a region transition, the error is not significantly higher for a shadow/non-shadow constant than for a sky/shadow constant.*

Secondly, going back to the sky example we know that outdoor shadows are caused by an object occluding sunlight. We can then further constrain c to

$$R_c > G_c > B_c \tag{5.10}$$

Where R_c, G_c, B_c are the red, green and blue values of c and the $>$ relations are obtained by taking into account the spectra of sun and skylight as well as generic camera sensitivities [WS82]. If one wants to remove shadows that occur in a different environ-

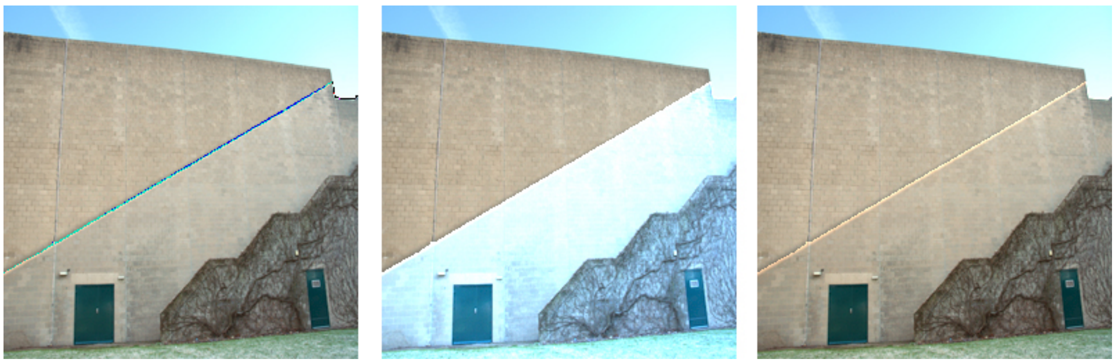


Figure 5.19: *Shadow free images using different methods to compute the constants.Left: the result of using a global Min Square Error constant. Note the slight shift in blue despite the sky-shadow transition being only a small part of the overall shadow edge. Middle: result when the minimal error (pixel-wise) occurs in the sky-shadow transition. Right: the result of computing a constrained constant. Both the colour balance and luminosity are very accurate.*

ment (from a light source point of view), then additional constraints have to be added to accurately determine c. While the above constraints are simple, they are necessary to correctly evaluate c.

We now have all the elements to find c. We first use Equations (5.9) and (5.10) to weed out implausible values. Then, taking noise into account, we select the constant at locations where the error, equation (5.8), is minimum. Finally, in order to avoid the single point of failure problem, we average c over the 1% of locations where the error is minimum.

When the image admits more than one shadow region, we repeat the procedure to find a specific constant per region. This will lead to better results than using only a single value of c for all shadow regions. The reason is that, in removing shadows, it is assumed that the lighting field is uniform within the shadow region. While this assumption usually holds, shadow regions located in various parts of the image may well have significant lighting differences. It is therefore worthwhile to treat different regions separately.

Finally, we have to consider what happens to the shadow boundary. The main issue in this case is that the transition between shadow and non-shadow regions is rarely immediate (i.e., the shadow edges are thicker than 1 pixel). Accordingly, this prevents us from using the same constant on the shadow edges. We have tried interpolating the constant across the boundary (for example, linearly going from 0 to c), but the results were unsatisfactory. We therefore decided to inpaint the boundary, using the method of Criminisi et al. [CPT04].

5.7.2 Simple Removal Results

Some results obtained with the constant method can be seen in Fig. 5.20. Despite the complexity of some of the scenes, the shadows are correctly removed or attenuated. The luminance levels on both sides of the (former) shadow are almost identical and the color balance is adequate. One of the main advantages of this method, though, is its speed. Indeed, given the shadow edges, the problem is reduced to finding a constant under 2 simple constraints. Such a task can easily be done in real time (even in $\mathrm{MATLAB}^{\mathrm{tm}}$).

This method is simple but knowledge of the environment in which the image is taken can be necessary to correctly constrain the constant and weed out potential errors.

5.8 Conclusion

We have established a framework for robust reintegration of shadow-free images. We have addressed the different problems of both existing 1D and 2D methods and proposed solutions to their shortcomings.

We have shown that a 1D approach was more suited to the task of shadow removal and that its results were more accurate than its 2D counterpart while still less computationally expensive. We further devised solutions for existing 1D reintegration using the insights that shadow regions had to be closed and that the number of crossings through the shadow edges should be limited. Additionally, we proposed a fast method to de-

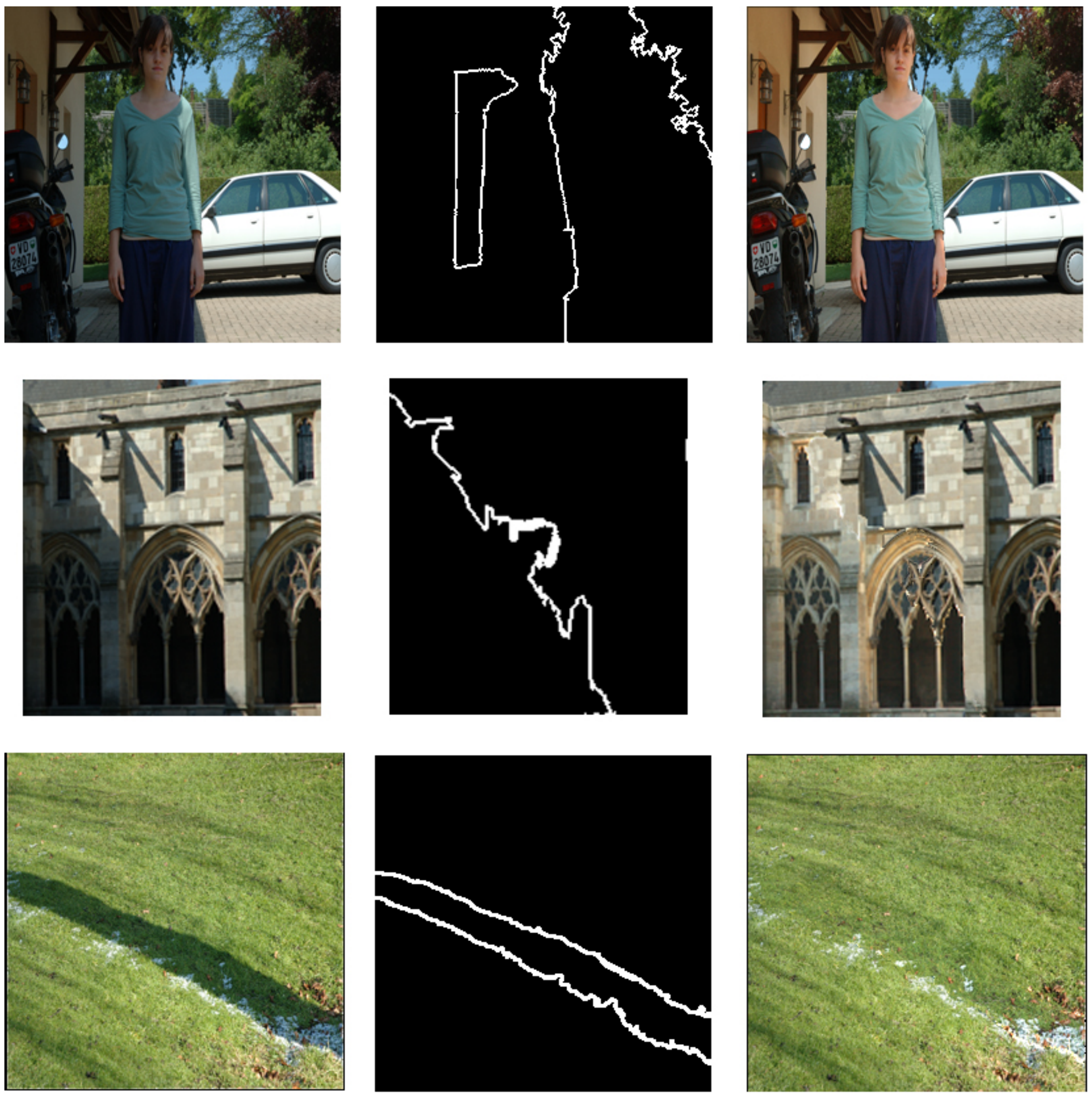


Figure 5.20: *Left column: original images, middle column: the shadow mask that was used. Right column: the results of adding a constrained constant to the shadow region. Even though the first two images are fairly complex scenes, the shadows are either completely removed or strongly attenuated and the rendering is realistic.*

rive random Hamiltonian paths in grid graphs. We finally proposed that non-visited shadow edge pixels do not have to be reintegrated and can simply be inpainted once the reintegration is complete. Experiments indicate that the new method removes shadows without visible artifacts.

We also proposed that, given some additional knowledge about the image (such as how shadows are cast), the shadow removal problem can be reduced to finding a constant at the “smoothest” locations of the shadow edge under simple constraints. The results show that this method outputs high quality images where the shadows are either removed or strongly attenuated.

Chapter 6

Applications of Random Hamiltonian Paths

In this chapter, we propose that the random Hamiltonian paths presented in Chapter 5 can be used with success in various path-based algorithms. Specifically, we look at the problem of path-based image segmentation where we show that using a small number of paths and a measure of region density, one can obtain good quality segmentations with only first order image statistics.

Additionally, we show the random Hamiltonian paths can be incorporated in the sieve framework [BHLA96] so that, depending on the number of paths used, one effectively obtains an algorithm that lies between the 1D and 2D version of the sieve algorithm. We then apply our 1.5D sieve to the problems of image denoising and texture classification. With a reasonable number of paths, the 1.5D sieve outputs comparable results, performance-wise, than the 2D algorithm but with the advantage of a much simpler implementation as well as possibilities for parallelization (Matlab code for the 1.5D algorithm is provided in Appendix B).

6.1 Path-Based Image Segmentation

The 2-pass raster segmenter is simple, fast and is often quoted in the literature. Unfortunately, it tends to oversegment images even in the presence of small amounts of noise. We present here a generalization of this approach where we discover regions by taking multiple random paths through an image. This approach fares better but still over segments an image. Yet, an analysis of region density shows that the underlying image structure can be discovered from the path-based segmentation. Indeed, the discovered edges are broadly comparable to those discovered by the widely used mean shift algorithm.

6.1.1 Introduction

From Land's Retinex [LM71] to scale-space processing [BHLA96], path-based methods have often been used with success in image processing and computer vision. Those paths can usually be divided in three categories: short random walks (as in [Lan77] and [MS00]), partially complete (in the sense that they almost cover the entire image), such as Frankle MacCann spiral path for retinex [FM83] or complete as raster paths for segmentation [BB82]. Most examples of complete paths are instances of the more general class of Hamiltonian paths, whose definition is "A path in a graph such that every vertex is visited once and once only" [Bol79] (or, in terms of images, we visit each pixel once and visit all pixels in the image). We here take the results presented in Chapter 5 for the generation of random Hamiltonian paths.

The problem of image segmentation has been studied for a long time and has spawned a wide variety of approaches ([MM00], [CM02] and [NB93] among others). The best performing algorithms currently make use of a combination of colour, texture and scale features and usually have many parameters that can be adjusted for optimum segmentation. As a result, many of these algorithms are either difficult to implement and/or computationally expensive to use. One of our goals here is to develop a segmentation algorithm that is broadly comparable to antecedent methods but, due to its simple path-based framework, is simpler and easier to implement.

We first introduce the 2-pass raster segmentation commonly cited in the literature. We then show how to use multiple Hamiltonian paths and simple first order image statistics on colour channels calculated along a path to group similar pixels.

After a single Hamiltonian path through the image there are many line like segmented structures (as oppose to desired regions). We group these linear structures and discover arbitrarily shaped regions by repeating our segmentation along different paths where now we group together the linear structures. After a small number of path segmentations we can discover arbitrarily shaped regions. We provide a detailed discussion of the convergence of our method.

Section 6.1.2 presents the standard two pass path based raster segmentation. In section 6.1.3 we look at how path based segmentation works in experiments and this allows us to elaborate on the basic algorithm. Results on real images are presented in section 6.1.4 for our path based approach and for the widely used mean shift algorithm. For the images tested both algorithms provided broadly similar performance, with the former being delivered much more quickly.

6.1.2 Background

Raster Segmentation (Sequential Labelling)

Sequential labelling is a technique used in computer vision for efficient segmentation of images [BB82]. Two orthogonal raster paths (such as the ones shown in Fig. 6.1) are used sequentially to connect pixels belonging to a same region. This method, first based on binary images, where determining the connectivity of pixels is straightforward [Gra71] was extended to encompass grayscale and colour images in [NB93].

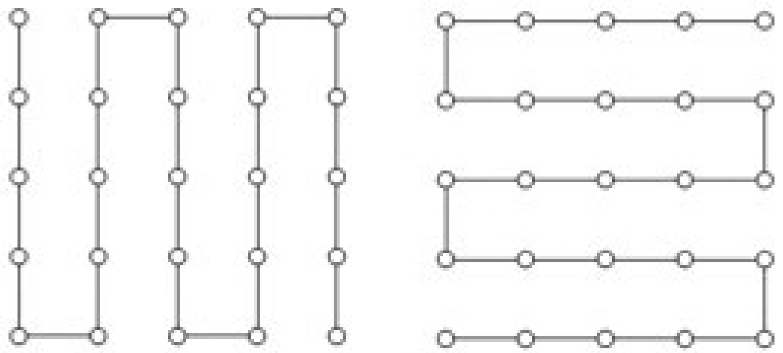


Figure 6.1: *The 2 orthogonal raster paths used in the original sequential labelling method.*

The sequential algorithm proceed as follows: the image is examined according to the paths shown in Fig. 6.1. If neighboring pixels are connected, they are then assigned the same label. When a pixel can be connected to more than one of its neighbors, the labels are considered to be equivalent (and are therefore merged).

To determine whether neighboring pixels are similar we will use Nayar and Bolle's reflectance ratio criterion [NB93].

$$\frac{I_a - I_b}{I_a + I_b} \leq \theta \tag{6.1}$$

This reflectance ratio, taken for image pixels a and b has the advantage that, for grey scale, it is independent of intensity. And, if computed on R, G, and B separately the triplet of ratios is independent of illumination [NB93]. And, so, supports segmentations which are independent of the lighting conditions.

6.1.3 Segmenting Images

To segment images, we use the same framework as the sequential labelling of [NB93]. However, instead of using two orthogonal raster paths, we recursively apply different random Hamiltonian paths.

Since the random Hamiltonian paths algorithm can generate a large number of random paths, we propose that can segment images with more accuracy than the 2-pass algorithm: we can use multiple paths to discover region connectivity. In the 2-pass approach, to get large regions one needs to be "optimistic" about the underlying image structure and so use a fairly large threshold to determine pixel (and hence region) similarity. With multiple paths we can be "pessimistic" and use a smaller threshold since we are secure in the knowledge that we can joint pixels in multiple different ways. Using a large number of paths results in a area-like processing of the image, despite it not being explicitly defined in the segmentation algorithm.

Finally, we note that while the paths can be efficiently computed, they can also be pre-computed for a certain image size. Thus the algorithm cost is the number of pixels multiplied by the number of paths. Typically, the latter is small and so the algorithm is very fast.

Let us now consider how images are segmented. Before proceeding further we are interested in the plausibility of our approach. If we take an image with 2 regions that are hard to segment can we automatically find the segmentation?

Convergence of the Algorithm

Let us create an image that consists of a double spiral. The two spirals are one pixel wide, while the image itself is of size 256×256, as illustrated in Fig. 6.2a. The first step in sequential labelling is to label all pixels in the image as belonging to a different region; here we have 256x256 pixels so we have 65536 different regions. We then recursively use the different pre-computed paths to process the image, using the colour reflectance-ratio merging criterion [NB93], i.e., two labels a and b are equivalent if

$$\max\{\frac{R_a - R_b}{R_a + R_b}, \frac{G_a - G_b}{G_a + G_b}, \frac{B_a - B_b}{B_a + B_b}\} \leq \theta \tag{6.2}$$

Where we define the value of θ to be 0.035. This value has been found through experimentation and is the same value throughout all the results presented in this chapter.

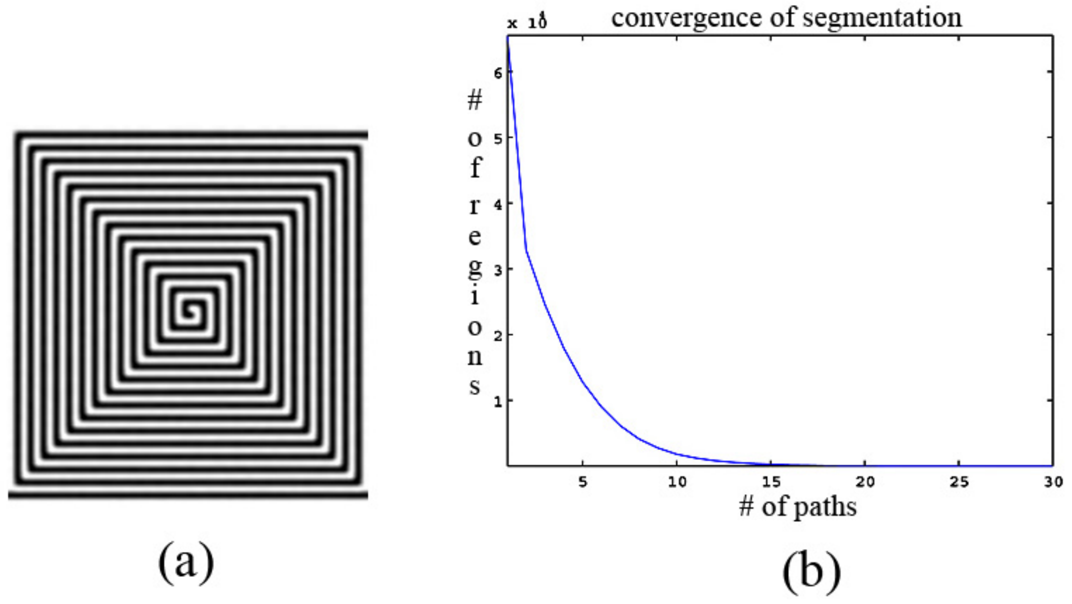


Figure 6.2: *(a): The spiral figure used in the convergence experiment. (b): The curve showing the actual convergence.*

If the structure of the different paths is random enough, and if the set of paths is complete with respect to the image size, then the segmentation should converge towards two distinct labels. Fig. 6.2b displays the number of distinct labels (i.e., regions) after each path. After 29 paths, the algorithm has converged to two distinct regions, each of them containing one spiral. Due to the random nature of the paths, we repeated this experiment 50 times. The mean number of paths of convergence was 26 and the highest number was 31. From this example, it can be inferred that since real images generally have much larger regions, the algorithm should then converge with less paths. For equivalently sized images, we have used 15 different paths, since the improvement in quality beyond them was not significant.

Moreover, it is simple matter to prove convergence in general. Consider an image with distinct regions, where each region can be discriminated from one another using the ratio test. The segmentation fails if after n iterations we have two adjacent pixels that should belong to the same region but are labelled differently. By assumption, these adjacent pixels satisfy the ratio criterion and so if we considered a path that joined these pixels together then these pixels (and their associated regions) would be merged. Since our paths are generated randomly this must happen given enough paths.

Segmentation Experiment on a real image

We are now interested in the detail of our algorithm: how will it perform on a real image?

Top left of Fig. 6.3 shows a simple image with well defined colour regions. Let us now consider what happens when we recursively apply our path-based method using the ratio criterion. Fig. 6.3 also shows the evolution of the segmentation for an image (each shade of grey is another label). These different steps picture how the segmentation converges towards stability, usually after 15 steps or so. While the convergence is fast, as shown in Fig. 6.4, it is really the steps between paths number 10 and 15 that effectively shape the segmentation.

The method chosen to represent the segmentation, however, is one based on regions density. The underlying assumption of this method is that a segmented image is composed of several regions within which the pixels have the same “label”. The region density is obtained by sliding a small $n \times n$ window over the image (in all our experiments, $n = 3$). The number of different regions (or labels) within this window expresses the region density for the center pixel. If we look in a small window and there is a single underlying region then we say this window has density one. If there are two regions then we have density two and so a small edge, up to a region density of nine (the maximal value) where all pixels within the window belong to a different region.

By definition, all pixels within a region have the same value (label). A region density higher than one is therefore indicative of the presence of an edge. Additionally, since

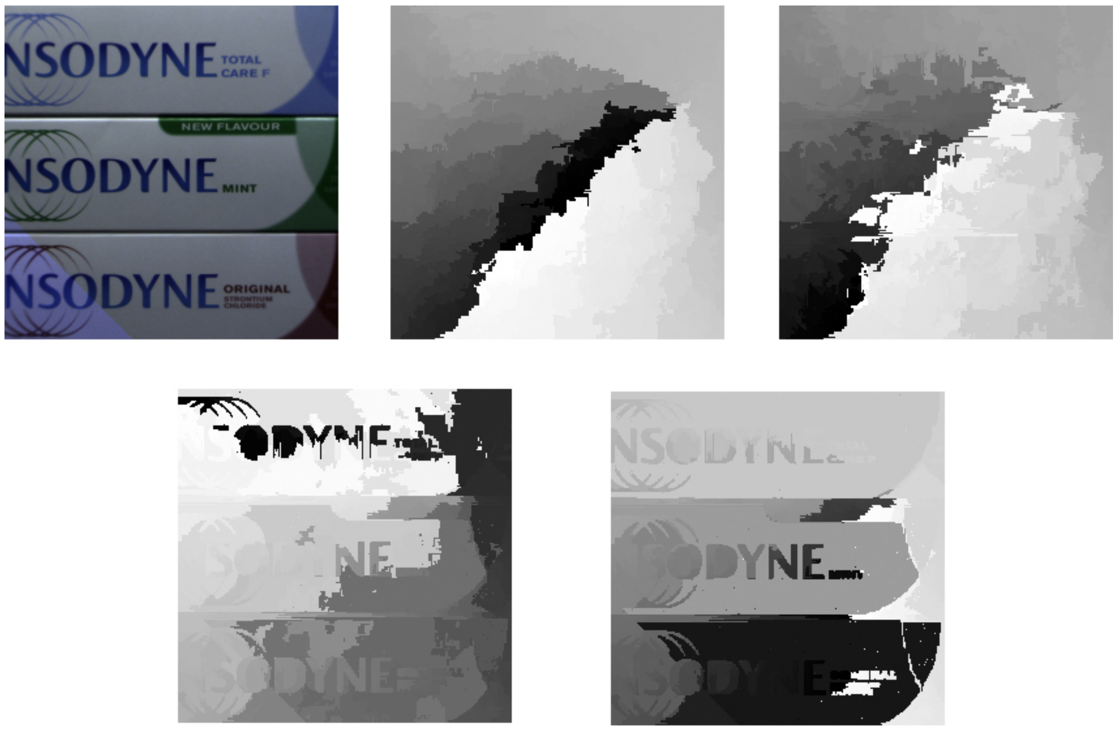


Figure 6.3: *From left to right and top to bottom: The original image and the segmentations after 1, 5, 10 and 15 paths respectively.*

the segmentation is based on colour ratios, we can encounter very high region densities in case of fast-changing reflectances, such as in grass or vegetation regions. However, most of the pixels belonging to such regions will appear solid white on the density map and edges can also therefore be extrapolated. The region density map after various number of paths is shown in Fig. 6.5.

We can use this approach because, in effect, noise is not a significant factor in our edge maps. An illustration can be seen in Fig. 6.6, where the region density of the original image is shown on the left and the right image is the edges obtained with our method. The original image contains significant noise, but the use of several random paths in effect denoised the image while preserving edge information.

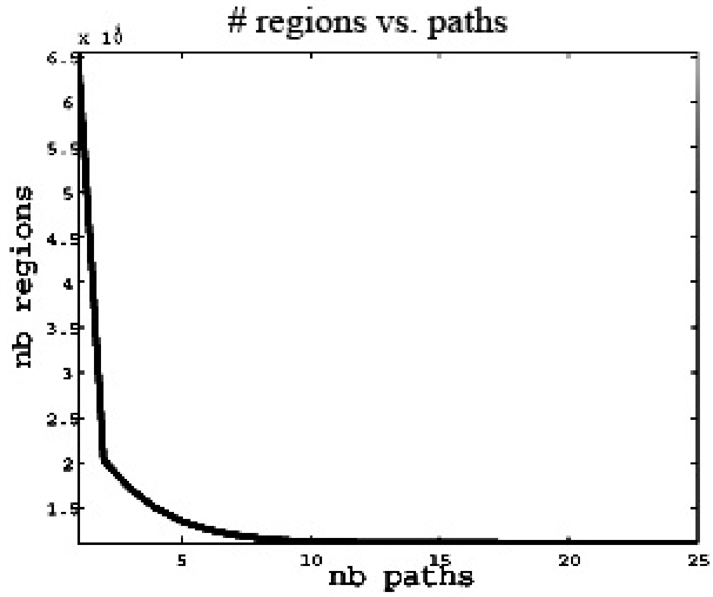


Figure 6.4: *The speed of convergence for the image shown in Fig. 6.3.*

6.1.4 Results

We first compare our results with the ones obtained using the 2-pass raster scan method. From the convergence curves previously shown, we see that the main reduction in the number of regions occurs within the first step. We might therefore expect both methods to deliver similar number of regions and, using the density approach described above, to exhibit similar edge representations. The results are displayed in Fig. 6.7. While the strong edges of the image are present in both results, we also see that the edge density map for the raster segmentation is much noisier. And, this shows that it has not merged regions as effectively as our multiple path approach.

Fig. 6.8 show results obtained with a variety of images. The first row contains the original images and the second one edges obtained with the 2-pass raster approach. The third row are results obtained with our method; the fourth row consists in segmentations obtained with the meanshift algorithm [CM02] where standard parameters were used.

From these results, two main aspects can be observed. The first one is that, as

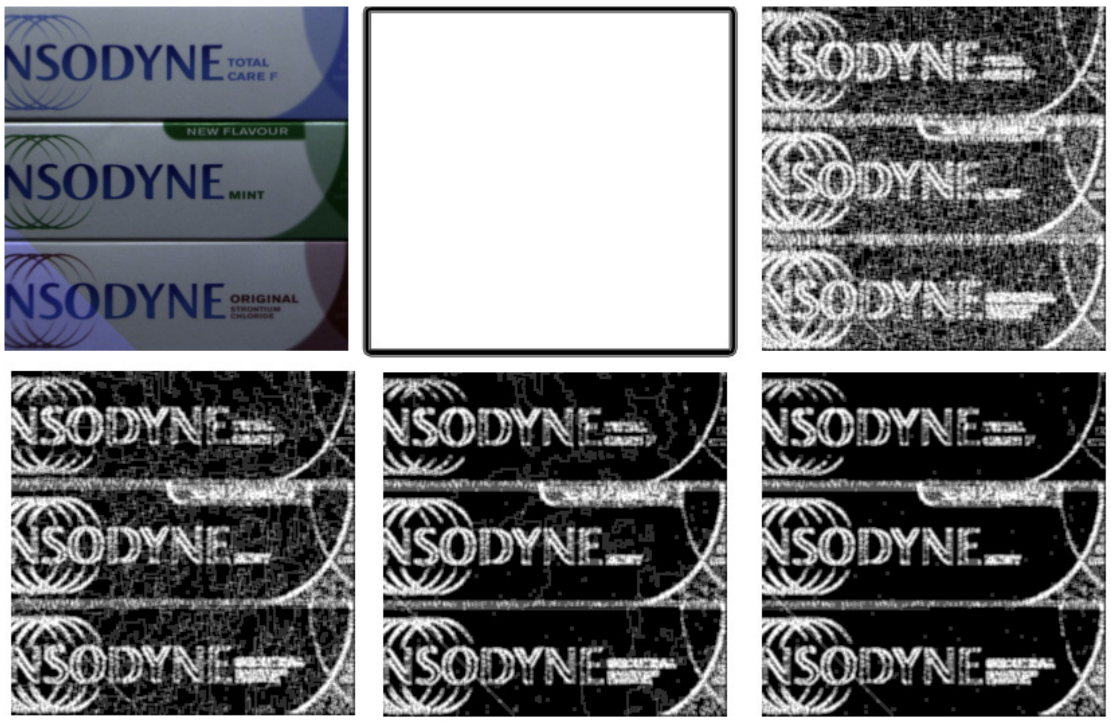


Figure 6.5: *From left to right and top to bottom: The original image and the segmentations after 1 (all white since the first step is to label all pixels differently), 2, 5, 10 and 15 paths respectively.*

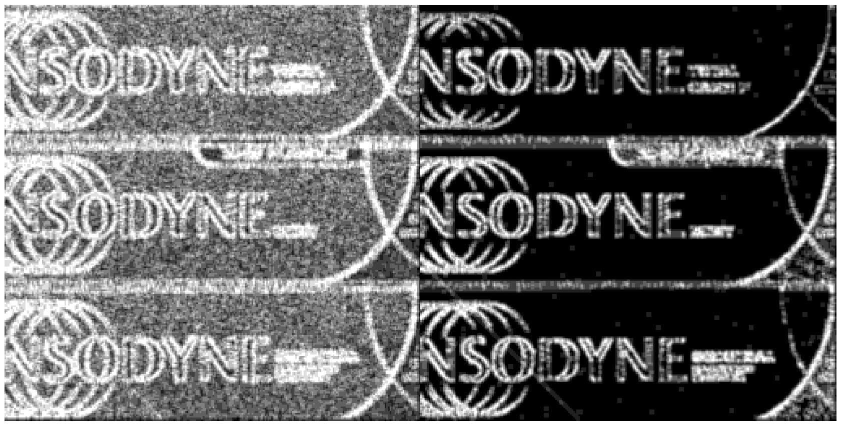


Figure 6.6: *Left: region density of the original image. Right: region density of the segmented image.*

previously thought, the results from our algorithm are an improvement over the original sequential labelling formulation problem. The second one, comparing our results to meanshift, is that while our algorithm is intrinsically much simpler, the results are

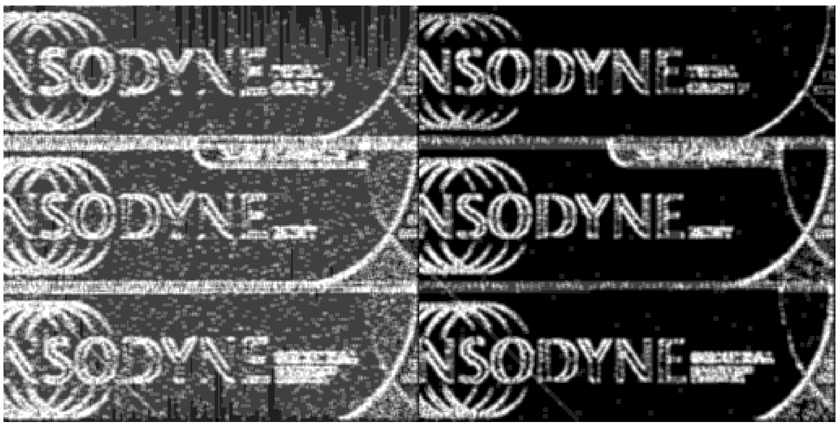


Figure 6.7: *Left: region density of the raster segmented image. Right: region density of the segmented image.*

broadly comparable. Both the 2-pass approach and our method also contain large vegetation regions compared to the meanshift algorithm. Since those regions are rapidly changing reflectance-wise, their underlying region density will be high. Filtering the region density map with a simple point-based high pass filter allow us to extract edges for both black (low density) and white (high density) regions. The resulting edges therefore oversegment some parts of the image, while undersegmenting others. A drawback, however, is the presence of noisy regions, explained by the fact that we only use local colour information to merge different labels/regions.

6.1.5 Conclusion

Up to this point, only first order statistics have been used in our segmentation framework. The obtained edges are, while accurate, sometimes either too thick or too noisy compared to the size of segmented regions. In [NB93], Nayar and Bolle discarded noisy or small regions in order to focus only on "valid" regions. Here, we however would like to obtain a full segmentation of the image. To improve current segmentations, one will have to look at higher order statistics, such as the rate of changes, in order to accurately detect and segment textures without adding too much complexity.

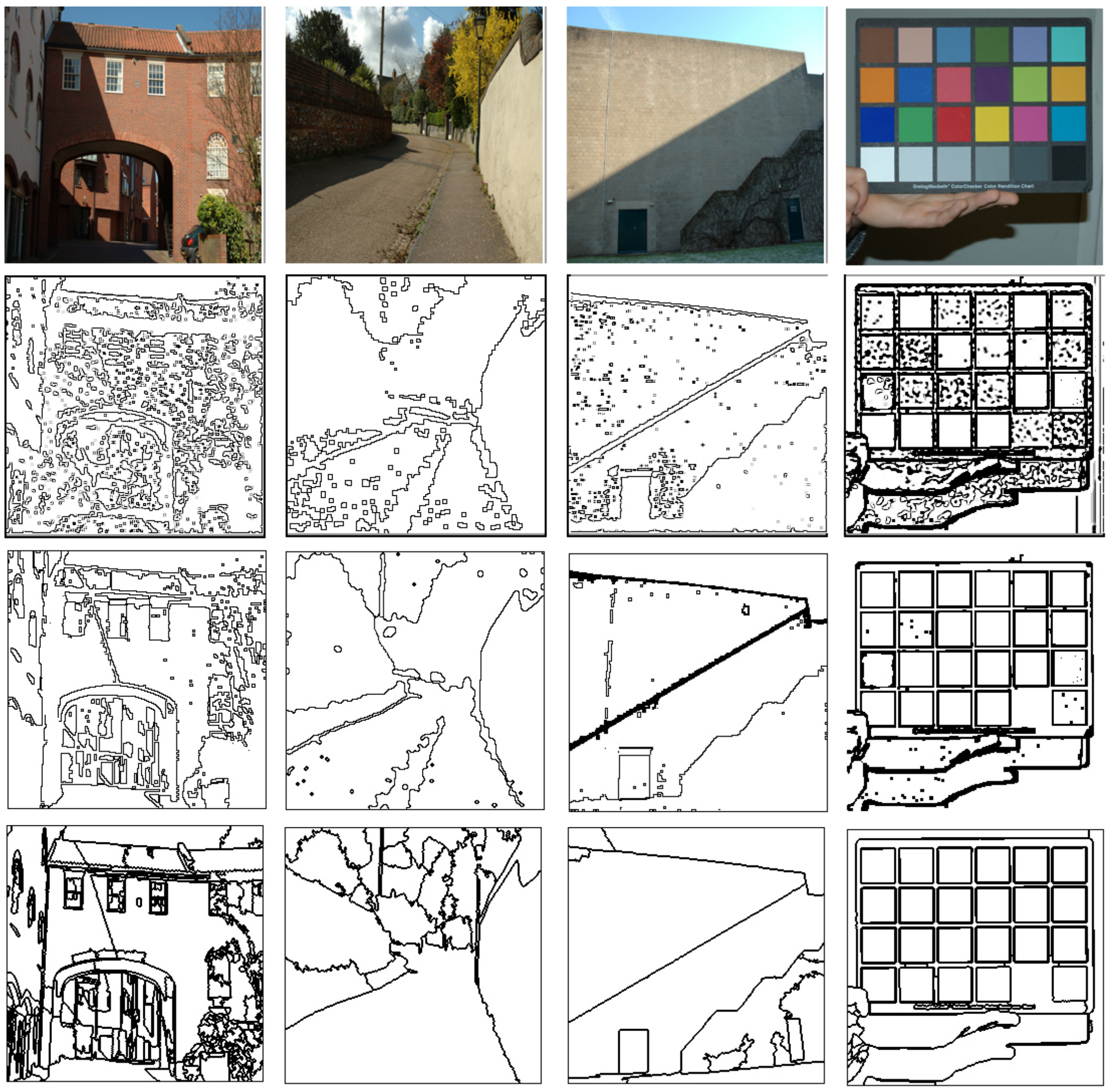


Figure 6.8: *1st row: Original images, 2nd row: edges obtained with the sequential labelling method, 3rd row: edges obtained with our method, 4th row: edges obtained with the mean-shift algorithm.*

6.2 The 1.5D Sieve

In this section we present a Hamiltonian path-based version of the sieve algorithm [JPR95, BHLA96]. Our method is simple to implement and can be made to behave like either the 1D or 2D sieve algorithm, depending on the number of paths used. Experiments indicate its performance for noise removal and texture analysis to be in line with the original sieve formulation.

6.2.1 Background

The idea behind the sieve algorithm is to use morphological filters to remove a signal extrema at different scales. The filters used are successive $\max$ and $\min$ operators that are applied on windows of different sizes (the size of the window corresponds to the scale). The sieve output is causal, edges vanish as the scale increases and no new image structure is introduced.

In the field of image processing, sieves can be used in either 1D, where the image is processed along a path [JPR95], or 2D, where the processing is done area-wise [BHLA96]. Sieves have been successfully applied, applications including image denoising [RAJ97] and texture analysis [Sou06] where it was shown that the algorithm of choice (1D or 2D) depended on the type of texture analyzed, namely on whether the texture set is rotated or not. The 2D version of the algorithm is more robust and generally performs better, but is challenging to implement.

We propose to use random Hamiltonian cycles to develop an algorithm that is equivalent to the 1D sieve when a single cycle is used and that is equivalent to the 2D sieve when all potential cycles over an image are used. In general, we use a small number of different random cycles (say between 10 and 20 depending on the size of the image and the desired scale) and therefore find ourselves somewhere between the 1D and the 2D case. To generate the cycles, we use the method presented in Chapter 5 that has a linear ($O(N)$) complexity, where N is the number of pixels in the image.

6.2.2 The Algorithm

The 1D sieve output of a signal X at a scale S is calculated in two passes. In each pass, maxima and minima of length $\leq S$ are respectively detected. The passes are:

$$\begin{aligned} Y_{\text{tmp}} &= \max_f(\min_b(X)) \qquad (6.3) \\ Y &= \min_f(\max_b(Y_{\text{tmp}})) \qquad (6.4) \end{aligned}$$

where f and b mean that for each pixel, the operation is done on a respectively forward or backward centered window of size $S + 1$. The proceeding of the 1D sieve algorithm is shown in Fig. 6.9.

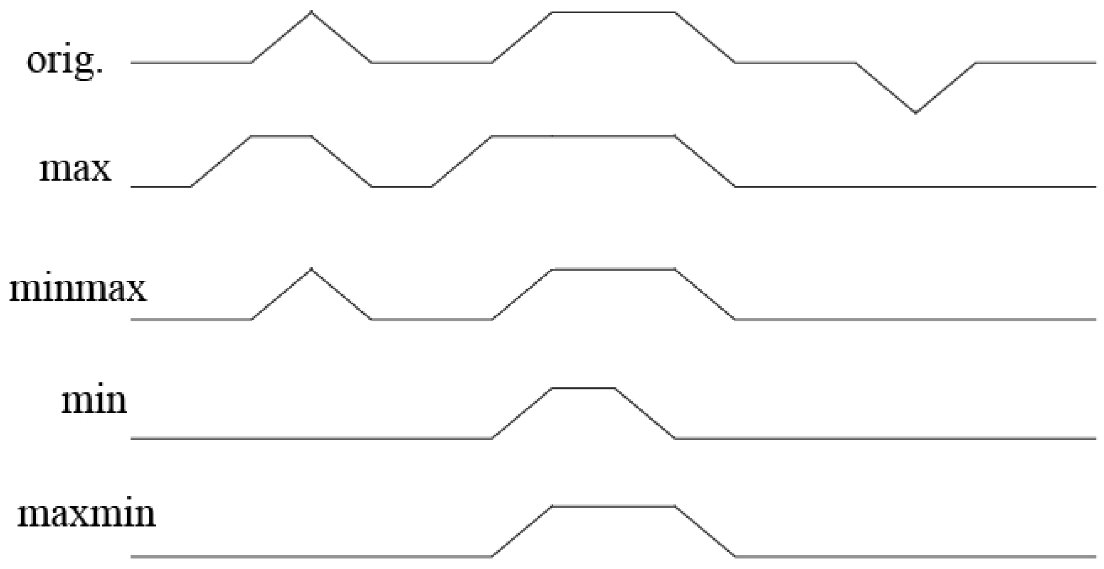


Figure 6.9: *A 1D signal (first row) and the result of the sieve algorithm at scale one after each of the passes described in equations (6.3) and (6.4)(third and fifth row)*

The 2D sieve use a more complex graph connectivity-based approach where, for a 2D signal, the extrema are detected if their connected area is $\leq S$. This process involves complicated "bookkeeping" because extrema can have complicated shapes -e.g. a "Q"-shaped region with 100 pixels is an extremum when $S = 100$. The 2D algorithm is illustrated in Fig. 6.10.

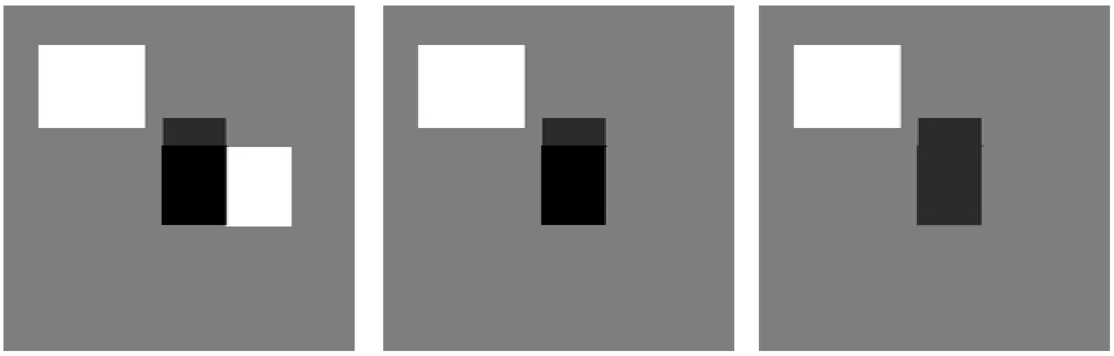


Figure 6.10: *A synthetic image (left) and the output of the 2D sieve algorithm after the maxmin (middle) and minmax (right) steps.*

In our approach, we first create N random Hamiltonian cycles over the image. We then create our 1.5D sieve in 4 steps:

$$
\begin{aligned}
Y^i_{\text{tmp}} &= \max_f(\min_b(X)) && (6.5)\\
Y_{tmp} &= \min_i(Y^i_{tmp}) && (6.6)\\
Y^i &= \min_f(\max_b(Y_{\text{tmp}})) && (6.7)\\
Y &= \max_i(Y^i) && (6.8)
\end{aligned}
$$

where Y^i and Y^i_{tmp} are sieve outputs for path i and where $\min_i$ and $\max_i$ represent selecting the minimum (or maximum) value for each image pixel over all the paths. The addition of the two extra steps, equations (6.6) and (6.8) is important because it is conservative. These steps indicate that if a region is deemed to be an extremum by all paths but one, then it will not be considered an extremum for the algorithm. This conservative estimation is shown in Fig. 6.11. We note that the algorithm can then be parallelized, since the steps described by equations (6.5) and (6.7) can be done separately for each path, combining their outputs at the end.

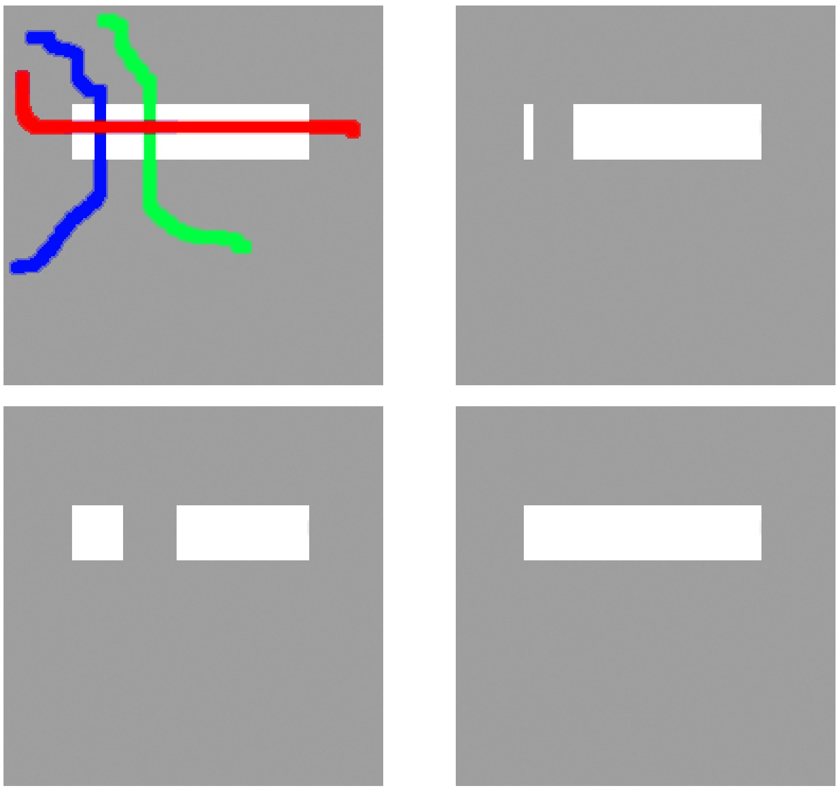


Figure 6.11: *Proceeding of the 1.5D sieve algorithm. Each path is used in a 1D sieve framework, but the outcome of all the paths is then assessed conservatively. The top-left image shows a synthetic example of an image region traversed by three paths. The first two admit an extremum at a scale of the height of the white rectangle and their processing results in the top-right and bottom-left images. The third path however does not admit an extremum at that scale. Its result, bottom-right, is therefore also the output of the 1.5D sieve algorithm.*

6.2.3 Equivalence of the 1.5D to the 1D and 2D sieve

In the 1.5D algorithm, by definition, when one path is used its behavior is equivalent to the 1D sieve.

For the 2D case, we have to look at the behavior at the limit. Let I be the image and I_2 be the result of upsampling I along both rows and columns. From Chapter 5, we know that every partition R of I is transformed in an equivalent region R_2 in I_2 and that R_2 admits an Hamiltonian cycle. If we define $\overline{R}$ to be the complement of R, it then follows that $\overline{R_2}$ also admits an Hamiltonian cycle.

Graph theory tells us that, on grid graphs, two disjoint, adjacent Hamiltonian cycles can be merged in a single one. Thus we deduce that there exists at least a cycle over I_2 such that the whole R_2 will be explored sequentially, allowing one to determine its precise scale (area) with our algorithm. The method to create Hamiltonian paths also guarantees that all possible cycles can be found. It therefore follows that if we generate all possible cycles on I_2, we then will be able to exactly find the scale of every region, thus making our algorithm the equivalent of the 2D sieve.

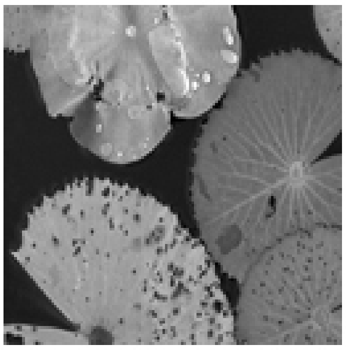
In practice, however, the number of possible paths, $N_{\max}$, is extremely large and we therefore choose N paths with $1 < N \ll N_{\max}$, which makes the algorithm behave in between the 1D and 2D sieve. In fact, the possible number of paths our algorithm can generate is the same as the number of possible spanning trees on the downsampled graphs. In [Wu77], it has been shown that a square graph with M vertices admitted $e^{1.16M}$ different spanning trees, close enough to infinity considering the size of typical images (Matlab actually considers it to be infinite for a 128×128 image).

6.2.4 Experiments

In this section, we evaluate the performance of the 1.5D sieve algorithm. Fig. 6.12 shows the output of the 1D, 2D and 1.5D sieves for different scales.

We now look at the robustness of the algorithm in the presence of noise. To do so, we use the framework of Harvey et al. [RAJ97] where the robustness of scale space algorithms was evaluated in the presence of Gaussian and Impulse noise. In this experiment, we compare the robustness of the 1.5D sieve with the original 2D algorithm from [BHLA96].

The images used in this test, shown in Fig. 6.13 are composed of a grayscale disc of different amplitudes cast on a 100×100 uniform square of amplitude 112. To these images is added either uncorrelated Gaussian noise ($\mu = 0,\ \sigma = 24$) or, alternatively, Impulse noise, where pixels are replaced with a random value in the range [0,255] with a noise density of 0.2.



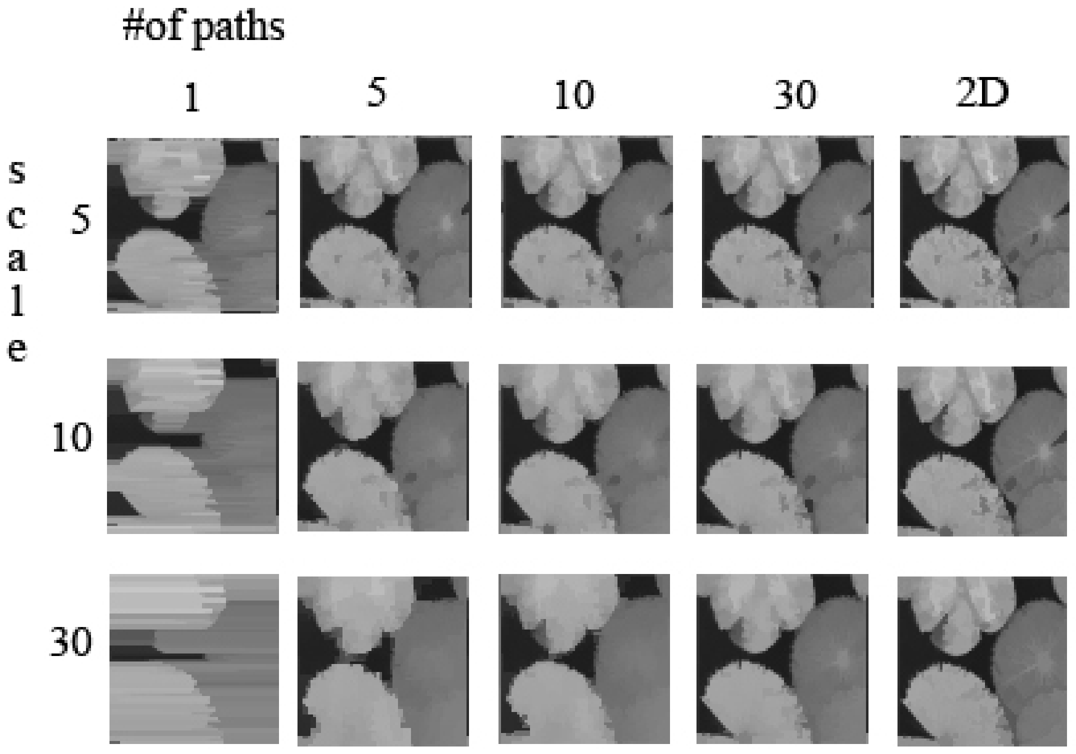


Figure 6.12: *The results of sieving an image with the 1D (first column), 1.5D (with 5, 10 and 30 paths) and 2D (last column) sieves at various scales. The size of the image is* 64×64*; for the 1.5D sieve, we see that the more paths used, the closer the results are to the 2D algorithm*

The experiment then consists in finding out the position and the scale of the central disc. To do so, one sieves the image at all possible scales. The scale at which the largest area is detected is assumed to be the scale of the disc (which is true in the noiseless case). The disc center is then assumed to be the center of mass of the detected area.

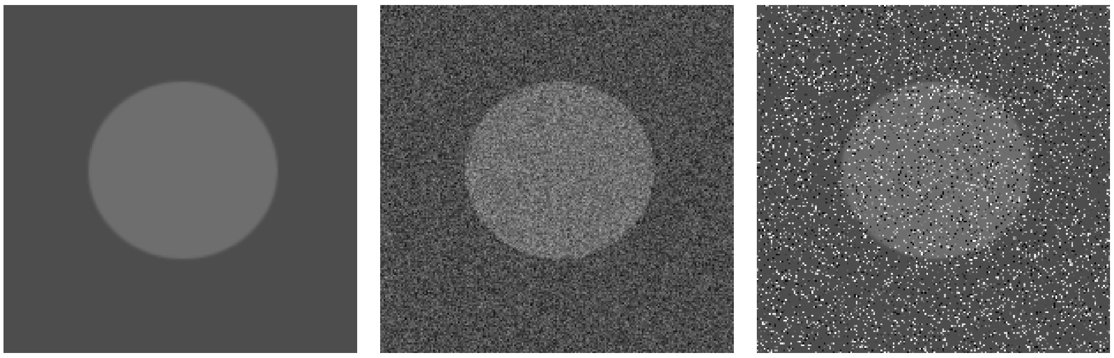


Figure 6.13: *The target image used in the robustness experiment (left); corrupted with Gaussian noise (middle) and Impulsive noise (right)*

Gaussian Noise	2D	1.5D
σ_x	0.280	0.270
σ_y	0.243	0.22
σ_s	55	48

Impulsive Noise	2D	1.5D
σ_x	0.0425	0.046
σ_y	0.0416	0.0472
σ_s	3.91	4.01

Table 6.1: *Standard deviations of estimates in Gaussian (G) and Impulsive (I) noise for the 2D sieve and the 1.5D sieve (using 30 paths).*

Repeating this experiment over 150 instances of noise allow us to calculate the standard deviation of the estimated circle position, σ_x and σ_y as well as the scale, σ_s. The smaller this standard deviation, the more robust the algorithm in the presence of noise. Table 6.1 shows the results for both the 2D sieve algorithm and the 1.5D sieve using 30 paths. We see that the difference between the 2 algorithms is small and thus, the 1.5D algorithm is almost as robust as the 2D sieve.

Finally, we want to assess how our modified sieve algorithm fares in texture classification. A comprehensive review of state of the art algorithms done in [Sou06] showed that, for the Outex_TC_00000 and Outex_TC_00010 test suites [Out, OMP$^+$02], the best performing algorithms were the 1D and 2D sieve respectively. The test suite TC_00000 is rotationally invariant and uses leave-out half cross validation (in all the Outex test suites, the training and testing sets are provided). The TC_00010 on the other hand is used to test rotation invariance. The training set consists of 480 texture from 24 classes imaged at a fixed orientation of 0^o. The testing set consist in 3840 images of the same 24 classes but the textures are at rotations of 5^o, 10^o, 15^o, 30^o, 45^o, 60^o, 75^o and 90^o.

It follows that, for the 1D sieve, one path will not suffice to discriminate between textures. For this test, the 1D sieve is actually composed of six different raster-type paths oriented at every 15^o. Examples of the texture sets can be seen in Fig. 6.14.

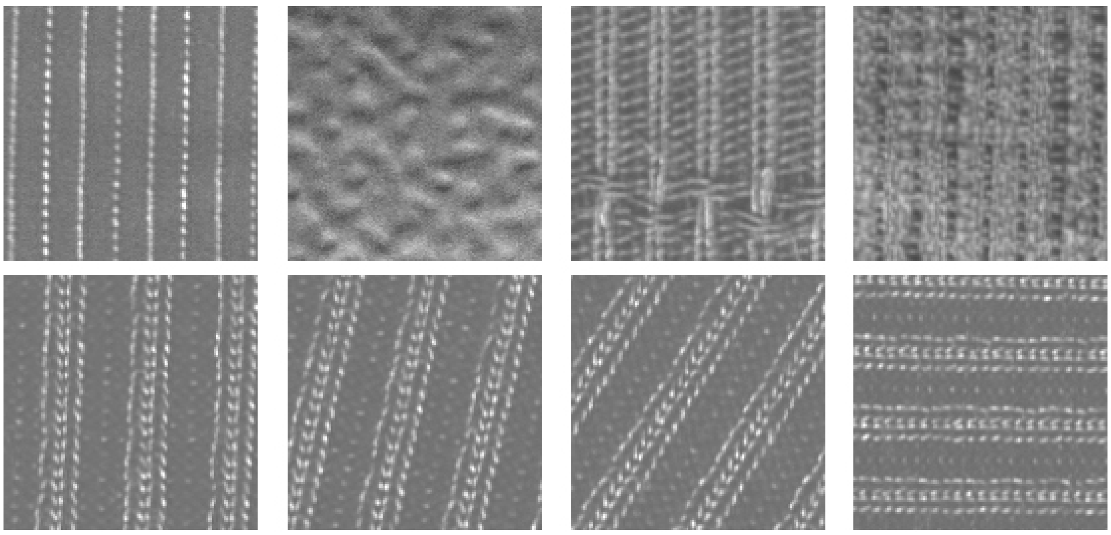


Figure 6.14: *Example of textures from the Outex TC_00000 suite (first row) and a texture under various orientation in the Outex TC_00010 suite (second row).*

The texture analysis is done using scale granularity. For each algorithm: 1D, 1.5D and 2D sieve, the texture images are sieved at scales 1, 2, 5, 13 and 30. From these six images (the original one and the five sieved images), we generate granularity images by taking the differences of two consecutive scales: i.e., original and scale one, scale one and scale two, scale two and scale five, etc. On these five granularity images, the first three moments (mean, standard deviation and skewness) are calculated, thus resulting in a 15-dimensional feature vector. In the 1D case, since there are six independent paths, the feature vector will have 90 entries. A feature vector is created for each image in the training set. Then, for each test image, its feature vector is calculated and its Euclidian

Outex_000000	Success Rate	vector length
1D	**0.998**	90
1.5D	0.99	15
2D	0.95	15
Outex_000010	Success Rate	vector length
1D	0.718	90
1.5D	0.902	15
2D	**0.943**	15

Table 6.2: *Recognition rates for both the rotationally invariant and variant Outex sets with 1D, 1.5D and 2D sieves.*

distance to the training data is used to determine to which class the test image belongs.

Since the textures are fairly strongly oriented, we use “masks” to guide the paths. Specifically, we create gradient images that will be the input of the random Hamiltonian paths algorithm. As a result, the paths will loosely adopt a given orientation. In these experiments, we will use 24 paths based on 12 rotations of the mask at regular angles (we create 2 paths per angle). Some orientations of the masks are shown in Fig. 6.15.

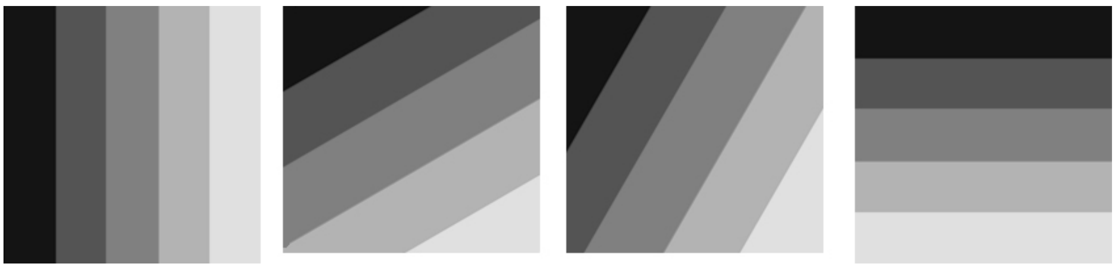


Figure 6.15: *Various orientations of masks used in the creation of random Hamiltonian paths.*

Using these paths, we form the output of our 1.5D sieve algorithm and thus have a 15-dimensional feature vector (the same size than the 2D sieve) that we use to classify textures. The Outex sets have defined training and testing procedures that we follow. The results, Table 6.2, illustrate that the 1.5D sieve algorithm performs “in between” the 1D and 2D methods and display a generally good performance.

6.2.5 Conclusion

We have shown that multiple random Hamiltonian paths could be used in the sieve framework. Using multiple paths and adopting a conservative approach, a 1.5D sieve algorithm (named as such because it behaves as either the 1D or 2D sieve in its limiting case) was developed.

We showed that this simpler version of the algorithm was as robust as the 2D sieve in the presence of noise and that a similar performance to state of the art algorithms could be achieved in texture classification.

We note that the 2D sieve algorithm is intrinsically rotationally invariant, which is not the case of either the 1D or 1.5D algorithms, invariance that explains its better performance in the rotation test. We point out that it is possible to create a path-based 1D sieve that would be rotation invariant, creating a minimum spanning tree using the gradients of the image as an input. One then creates a different path for each test image, thus having a rotationally invariant metric but loosing its generality (since it would mostly be an ad-hoc construction for the test set).

Chapter 7

Conclusion and Future Work

In this thesis, we have investigated the problem of interaction between light and Lambertian surfaces; our research has focused on estimating, detecting and removing illuminants.

Starting from the chromagenic theory, we have performed a detailed error analysis of the original algorithm and shown that bright, achromatic RGBs yielded more accurate illuminant estimations than darker more saturated ones. This lead us to propose the bright-chromagenic algorithm for illuminant estimation. Through extensive testing on both synthetic and real data, we have shown that the bright chromagenic algorithm remedies the weaknesses of the original chromagenic formulation, notably by removing the need for registered images, and that it performed significantly better than current state of the art illuminant estimation algorithms.

We have further extended the chromagenic framework in Chapter 4 to show that one can change the scope of the algorithm from illuminant estimation to illuminant detection. Instead of finding a single best fit for the entire image, pixels (or regions) are labelled with one illuminant from a subset of two or three. Constraining the problem in that way allows us to obtain accurate illuminant masks with good accuracy without requiring the illuminants to be estimated.

Chapter 5 focused on the removal of shadows in images. We provided a framework for robust shadow removal based on 1-dimensional reintegration methods and showed that is provides better results than 2D integration methods. We also proposed that, if the conditions of shadow formations are known, shadows could be attenuated or removed by simply adding them a constant found by constrained minimization.

To perform 1D shadow removal we have, in Chapter 5, developed an algorithm that produces random Hamiltonian paths on graphs that are complete by downsampling in a linear time. We use a maze-based implementation that produces such paths in a short time (0.5 seconds for a 1024×1024 image). In Chapter 6, we showed that Hamiltonian paths could be used in different applications, where they could provide simpler frameworks or improved results.

We are now at the end of this thesis, and may ask ourselves where this work can be taken. There are a number of ways the work presented here can be extended and we would like to show some possibilities.

Concerning illuminant estimation: one could manufacture optimal filter and sensors, which would increase the algorithm accuracy even further. The main problem of the chromagenic theory, however, is that it is based on having two images of each scene at our disposal. A possible improvement is to obtain these images at the same time, either through a "chromagenic camera" (with 2 CCDs) or, through a special filter that would produce an image composed in parts of a filtered and unfiltered response (for instance a filter whose left half is neutral and right half is yellow). While there are optical conditions to consider, our own eyes actually possess such a filter: the macular pigment, which covers only part of the retina. Replicating this would allow chromagenic illuminant estimation to be performed on a single image.

Illuminant detection can be improved upon as well. For now, we are able to detect different illuminants in an image but one could extend that further. Consider an image, which, we surmise, has two different illuminants. Once these illuminants are found, we can look into each of the two classes and suppose that each of them also contains two illuminants. In this event, the differences are smaller but they still exist and are most likely caused by different aspects such as mutual illuminations, specularities, etc. Another possibility is to interpret the illuminant maps not as binary or ternary as we have done here but as continuous, i.e., creating an alpha-mask of illumination. The success of these extended scheme, however, appears to be dependant in either having perfectly registered images, or being able to distinguish between registration errors and illumination changes.

Finally, as we have shown in Chapter 6, multiple random Hamiltonian paths can be used in various applications. One can therefore explore the possibilities offered by these paths in both path-based applications and 2-dimensional frameworks. Given the method used to created the paths, one can also make them “adaptive”, since their structure follows the spanning tree one, spanning tree that can be obtained using adequate weighting schemes.

Appendix A

A Maze Implementation of Hamiltonian Paths

In this appendix, we show a fast implementation of the random Hamiltonian paths using a maze framework.

To illustrate this analogy, consider Fig. A.1, where a schematic image is represented as a graph, over which a spanning tree is created. Reforming the schematic image based on the spanning tree, one sees that this image can effectively be considered as a maze. In fact, it is known [Maz02] that creating a spanning tree over a graph yields a *perfect maze:* a maze such that there is a single path between two given cells.

With this in mind, we create random Hamiltonian paths by first creating a perfect maze over the downsampled image. The maze is then upsampled and, starting from an arbitrary location, one "walks" around the upsampled maze keeping its hand (the left one in this case) on the walls. Doing so, one will walk the entire maze, visiting all cells in turn before ending up at the starting point; the order of visited pixels is therefore an Hamiltonian cycle over the original sized image. Fig. A.2 shows the framework of the algorithm, using screen shots from our program. We would like to acknowledge the help of Fabien Ezber for his deep knowledge of efficient programming and memory management.

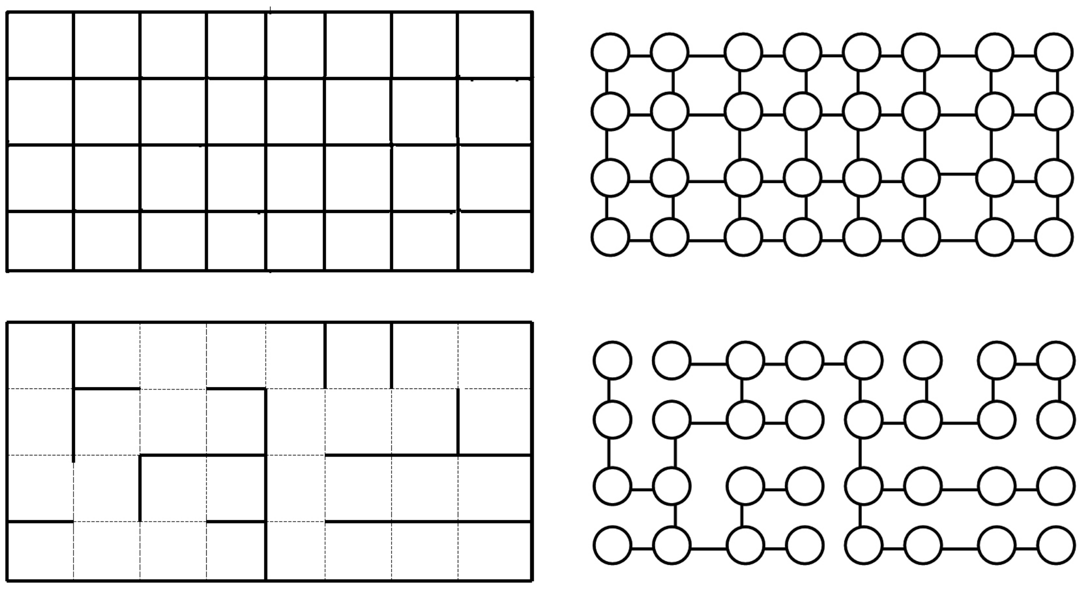


Figure A.1: *The relationship between image, graph, spanning tree and maze.*

The performance of our implementation is shown in Fig. A.3 where the time taken to obtain an Hamiltonian cycle is plotted versus the size of the graph (the total number of pixels in the image).

We note that storing a path takes an equivalent amount of space as the spanning tree, i.e., one quarter of the image size. In applications that require a large number of paths be stored that might, however, be too memory consuming. If one accepts that the paths can have a slightly less random structure, one can upsample the maze by a larger factor than two. In that case, the amount of space needed for storing the relevant information can be reduced to suit one's needs. An illustration of this possibility is shown in Fig. A.4, where one can see that an Hamiltonian path (instead of a cycle) is obtained.

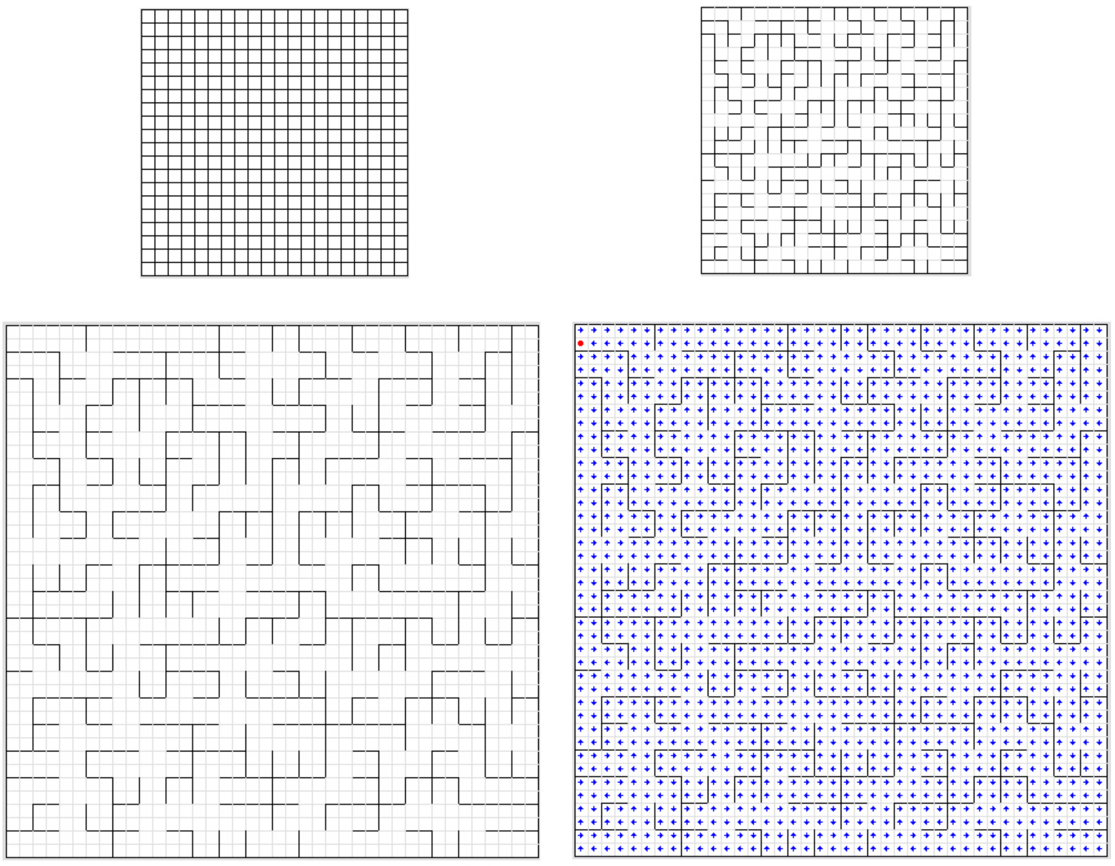


Figure A.2: *Screen shots of our algorithm used to generate random Hamiltonian paths. We first create a perfect maze and upsample it. Finally, the maze is walked through using the "keeping a hand on the wall" technique. The blue arrows indicate the direction taken at each pixel. The starting point is at the top-left corner and the ending point is symbolized by the red dot.*

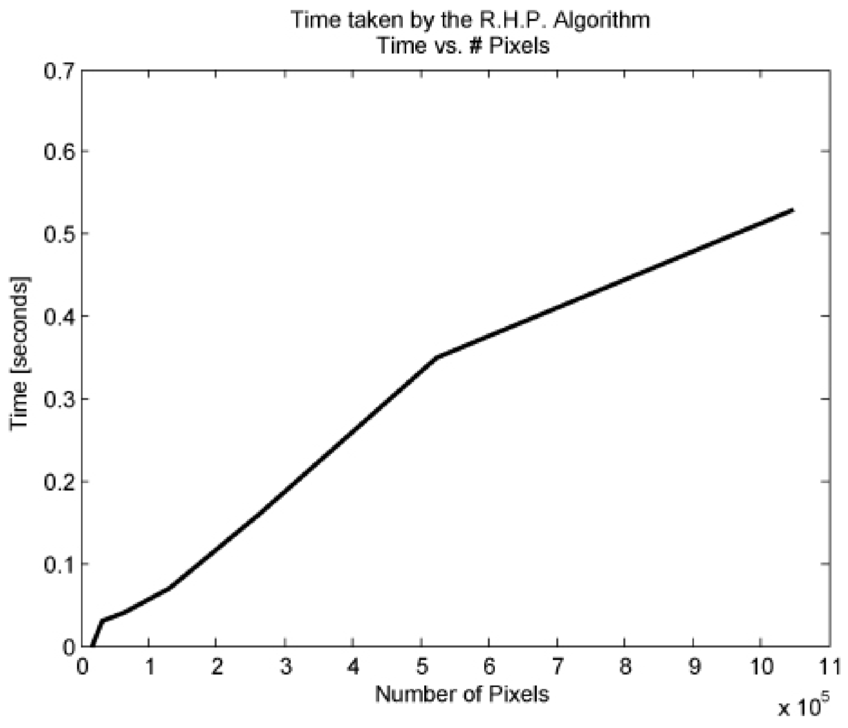


Figure A.3: *The performance of our algorithm in seconds versus the number of pixels in the image. A* 1024×1024 *path can be obtained in 0.53 seconds.*

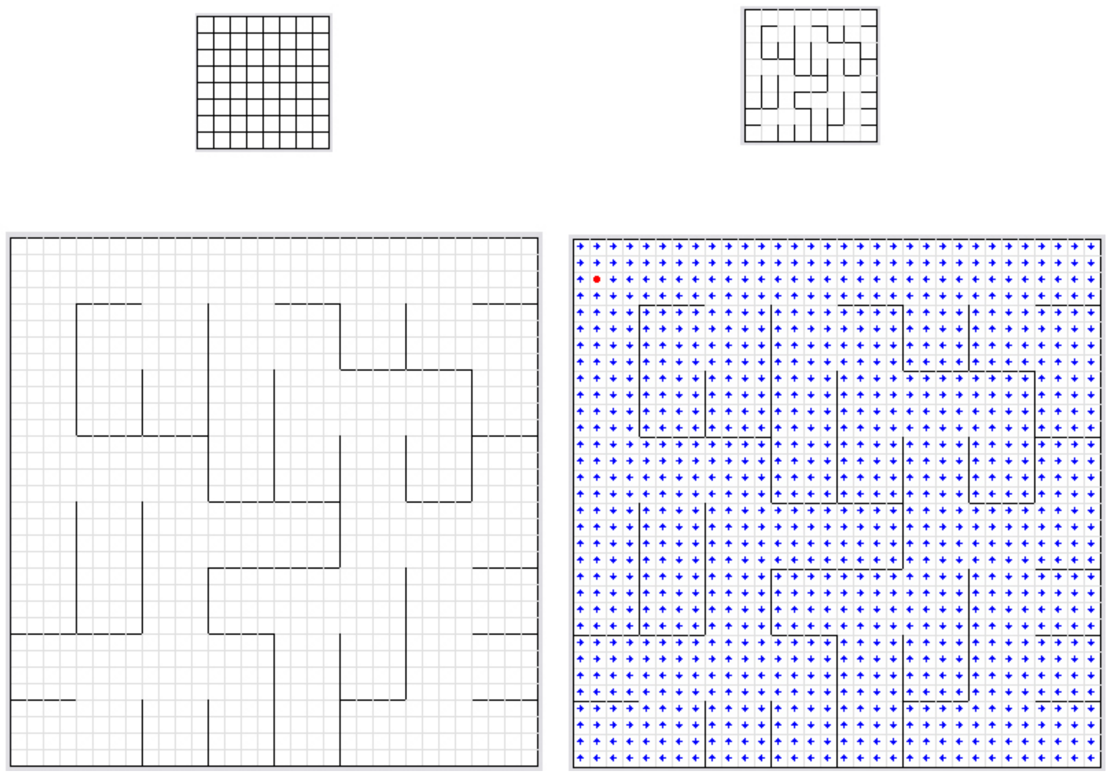


Figure A.4: *A graph and a path obtained by upsampling the maze by a factor of 4. In that case, the algorithm does not output a cycle but a path.*

Appendix B

1.5D Sieve Code

In this appendix, we show the Matlab code used to process the image according to our 1.5D sieve algorithm. We do not here make optimality claims about the code. Instead, we want to illustrate that contrarily to the 2D sieve algorithm, the 1.5D version can be easily coded.

```
function S_out=sieve_filter_split(path,I,scale_max);

%path is a matrix containing all the desired paths
[nb_paths, path_length]=size(path);
[r,c]=size(I);
first_pass=zeros(nb_paths,r*c); %%%the max-min direction%%%
%%all paths processed independently => can be parallelized
for i=1:nb_paths,
    %processing the image along the ith path
    I=I(path(i,:));
    M1=[I;I(2:path_length) I(path_length)]; %scale 1
    m1=max(M1); %scale 1
    M2=[m1;m1(1) m1(1:path_length-1)]; %scale 1
    m2=min(M2); %scale 1
    for scale_level=2:scale_max,
        vect_scale=m2(path_length)*ones(scale_level+1,1);
        %processing the window
        M1=fliplr(toeplitz(vect_scale,fliplr(m2)));
        m1=max(M1);
        vect_scale=m1(1)*ones(scale_level+1,1);
        M2=toeplitz(vect_scale,m1);
        m2=min(M2);
    end
    %reordering the results from different paths
    results_tmp=[path(i,:); m2]';
    results_tmp=sortrows(results_tmp,1);
    first_pass(i,:)=results_tmp(:,2)';
end
%equation (6.6): being conservative
first_pass_outcome=min(first_pass);
%preparing the min-max direction
second_pass=zeros(nb_paths,path_length);
%%all paths processed independently => can be parallelized
for i=1:nb_paths,
    %we start with the results from max-min
    m4=first_pass_outcome(path(i,:));
    M3=[m4;m4(2:path_length) m4(path_length)]; %scale 1
    m3=min(M3); %scale 1
    M4=[m3;m3(1) m3(1:path_length-1)]; %scale 1
    m4=max(M4); %scale 1
    for scale_level=2:scale_max,
        vect_scale=m4(path_length)*ones(scale_level+1,1);
        M3=fliplr(toeplitz(vect_scale,fliplr(m4)));
        m3=min(M3);
        vect_scale=m3(1)*ones(scale_level+1,1);
        M4=toeplitz(vect_scale,m3);
        m4=max(M4);
    end
    results_tmp=[path(i,:); m4]';
    results_tmp=sortrows(results_tmp,1);
    second_pass(i,:)=results_tmp(:,2)';
end
% equation (6.8), being conservative
second_pass_outcome=max(second_pass);
I_sieve=reshape(second_pass_outcome,r,c);
%outcome of the sieve at the chosen scale
S_out=I_sieve;
```

Bibliography

[ACR05] A. Agrawal, R. Chellapa, and R. Raskar. An algebraic approach to surface reconstruction from gradient fields. In *Proc. of the International Conference on computer Vision (ICCV)*, pages 174–181, 2005.

[Adh01] G.S. Adhar. Optimal parallel algorithms for cut vertices, bridges, and hamiltonian path in bounded interval tolerance graphs. In *Proc. of the Eighth International Conference on Parallel and Distributed Systems. (ICPADS)*, pages 91–98, 2001.

[AR86] L. Arend and A. Reeves. Simultaneous color constancy. *Journal of The Optical Scociety of America part A*, 3:1743–1751, 1986.

[Bar99] K. Barnard. Practical color constancy. Ph.D. Thesis, Simon Fraser University, Vancouver, Canada, 1999.

[Bar02] K. Barnard. Data for computer vision and computational colour vision [online]. http://www.cs.sfu.ca/∼color/data, 2002.

[BB82] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice Hall, 1982.

[BBS97] D.H. Brainard, W.A. Brunt, and J.M. Speigle. Color constancy in the nearly natural image: 1 asymmetric matches. *Journal of The Optical Scociety of America part A*, 14:2091–2110, 1997.

[BCF02] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms- part i: methodology and experiments with synthetized data. *IEEE Trans. on Image Processing*, 11:972–984, 2002.

[BF97] D.H. Brainard and W.T. Freeman. Bayesian color constancy. *Journal of The Optical Scociety of America part A*, 14:1393–1411, 1997.

[BHLA96] J.A. Bangham, R. Harvey, P.D. Ling, and R.V. Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5:283–299, 1996.

[Bla85] A. Blake. Boundary conditions for lightness computation in mondrian world. *Computer Vision, Graphics and Image Processing*, 32:314–327, 1985.

[BMCF02] K. Barnard, L. Martin, A. Coath, and B. Funt. A comparison of computational color constancy algorithms- part ii: experiments with image data. *IEEE Trans. on Image Processing*, 11:985–996, 2002.

[BMF00] K. Barnard, L. Martin, and B. Funt. Colour by correlation in a three dimensional colour space. In *Proc. of the sixth European Conference on computer Vision*, pages 375–389, 2000.

[Bol79] B. Bollobas. *Graph Theory.* Springer Verlag, 1979.

[Bra98] D.H. Brainard. Color constancy in the nearly natural image: 2 achromatic loci. *Journal of The Optical Scociety of America part A*, 15:307–325, 1998.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[Buc80] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310:1–26, 1980.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[CFB02] V.C. Cardei, B. Funt, and K. Barnard. Estimating the scene illuminant chromaticity using a neural network. *Journal of The Optical Scociety of America part A*, 19:2374–2386, 2002.

[CGC$^+$03] Y-Y Chuang, D.B. Goldman, B. Curless, D.H. Salesin, and R. Szeliski. Shadow matting and compositing. In *ACM SIGGRAPH 2003*, pages 494–500, 2003.

[CM02] D. Comanicu and P. Meer. Mean shift: A robust approach towards feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24:603–619, 2002.

[Coh64] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psycon. Science*, 1:369–370, 1964.

[CPT04] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. on Image Processing*, 13:1200–1212, 2004.

[CST02] S.D. Chen, H. Shen, and R. Topor. An efficient algorithm for constructing hamiltonian paths in meshes. *Parallel Computing*, 28:1293–1305, 2002.

[DFH03] M.S. Drew, G.D. Finlayson, and S.D. Hordley. Recovery of chromaticity image free from shadows via illumination invariance. In *ICCV Workshop on Color and Photometric Methods*, pages 32–39, 2003.

[DI93] M. D'Zmura and G. Iverson. Color constancy i: Basic theory of two-stage linear recovery of spectral descriptors for lights and surfaces. *Journal of the Optical Society of America*, 10:2148–2165, 1993.

[DI94] M.M. D'Zmura and G. Iverson. Probabilistic color constancy. Geometric Representations of Perceptual Phenomena: Papers in Honor of Tarow Indow's 70th birthday, 1994.

[DXW01] J.M. DiCarlo, F. Xiao, and B.A. Wandell. Illuminating illumination. In *Proc. of the ninth Color Imaging Conference*, pages 27–34, 2001.

[D'Z92] M. D'Zmura. Color constancy: surface color from changing illumination. *Journal of the Optical Society of America*, 9:490–493, 1992.

[FBM98] B. Funt, K. Barnard, and L. Martin. Is machine color constancy good enough? In *Proc. of the Fifth Color Imaging Conference*, pages 455–459, 1998.

[FC88] R. Frankot and R. Chellapa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. on Pattern analysis and machine intelligence*, 10:439–451, 1988.

[FDF94] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: Sensor transformations for improved color constancy. *Journal of the Optical Society of America*, 11:1553–1563, 1994.

[FDH91] B.V. Funt, M.S. Drew, and J. Ho. Color constancy from mutual reflection. *International Journal of Computer Vision*, 6:5–24, 1991.

[FDL04] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 582–595, 2004.

[FF94] G.D. Finlayson and B.V. Funt. Color constancy with shadows. *Perception*, 23:89–90, 1994.

[FF04] G.D. Finlayson and C. Fredembach. Fast re-integration of shadow free images. In *Proc. of the 12th IS & T color imaging conference*, pages 117–122, 2004.

[FF05] C. Fredembach and G.D. Finlayson. Hamiltonian path based shadow removal. In *Proc. of the 16th British Machine Vision Conference (BMVC)*, pages 970–980, 2005.

[FH99] G.D. Finlayson and S.D. Hordley. Selection for gamut mapping colour constancy. *Image and Vision Computing*, 17:597–604, 1999.

[FH01] G.D. Finlayson and S.D. Hordley. Color constancy at a pixel. *Journal of the Optical Society of America*, 18:253–264, 2001.

[FHD02] G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images. In *Proc. of the 7th European Conference on Computer Vision (ECCV)*, pages 823–836, 2002.

[FHH01] G.D. Finlayson, S.D. Hordley, and P.M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1209–1221, 2001.

[FHLD06] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEEPAMI*, 28:59–68, 2006.

[FHM05a] G. Finlayson, S. Hordley, and P. Morovic. Chromagenic filter design. In *Proceedings of the 10th AIC*, pages 1079–1083, 2005.

[FHM05b] G. Finlayson, S. Hordley, and P. Morovic. Colour constancy using the chromagenic constraint. In *Computer Vision and Pattern Recognition (CVPR) 2005*, pages 1079–1086, 2005.

[Fin96] G.D. Finlayson. Color in perspective. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:1034–1038, 1996.

[FM83] J. Frankle and J. McCann. Method and apparatus for lightness imaging. US Patent No. 4,384,336, 1983.

[FM05] G. Finlayson and P. Morovic. Human visual processing: Beyond 3 sensors. In *Proc. of the IEE international conference on visual information engineering*, pages 1–7, 2005.

[For90] D.A. Forsyth. A novel algorithm for colour constancy. *Intl Journal of Computer Vision*, 5:5–36, 1990.

[FS99] G.D. Finlayson and G. Schaefer. Single surface colour constancy. In *Proc. of the Seventh Color Imaging Conference*, pages 106–113, 1999.

[FT04] G.D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Proc. of the Twelfth Color Imaging Conference*, pages 37–41, 2004.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[GJT88] R. Gershon, A.D. Jepson, and T.K. Tsotsos. From [r,g,b] to surface reflectance: computing color constant descriptors in images. *Perception*, pages 755–758, 1988.

[Gra71] S.B. Gray. Local properties of binary images in two dimension,. *IEEE Trans. on Computers*, 20:551–560, 1971.

[HE81] W. Hackbusch and U. Trottenberg Editors. *Multigrid Methods*. Springer-Verlag, 1981.

[HF06] S.D. Hordley and G.D. Finlayson. Reevaluation of color constancy algorithm performance. *Journal of the Optical Society of America A*, 24:1008–1020, 2006.

[HHD99] T. Horprasert, D. Hardwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. of ICCV Frame-Rate Worshop*, 1999.

[HHFD97] P.M: Hubel, J. Holm, G.D. Finlayson, and M.S. Drew. Matrix calculations for digital photography. In *Proc. the Fith color Imaging Conference*, pages 105–111, 1997.

[Hod98] N. Burnett Hodd. Putting chromagen to the test. *Optometry today*, 38:39–42, 1998.

[HT01] R.V. Hogg and E.A. Tanis. *probability and Statistical Inference*. Prentice Hall, 2001.

[IPS82] A. Itai, C.H. Papadimitrou, and J.L. Szwarcfiter. Hamiltonian paths in grid graphs. *SIAM Journal of Computing*, 11:676–686, 1982.

[JD03] H. Jiang and M. Drew. Tracking objects with shadows. In *CME03: International Conference on Multimedia and Expo,*, pages 100–105, 2003.

[JMW64] D.B. Judd, D.L. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *Advances in Neural Information Processing*, 54:1031–1040, 1964.

[Joh02] K. Johansson. Non-intersecting paths, random tilings and random matrices. *Probab. Theory Related Fields*, 123:225–280, 2002.

[JPR95] J.A.Bangham, P.D.Ling, and R.Harvey. Scale space from nonlinear filters. In *Proc. of the fifth International Conference on Computer Vision*, pages 163–168, 1995.

[JR99] M.J. Jones and J. Rheg. Statistical color models with applications to skin detection. In *Proc. of Computer Vision and Pattern Recognition*, pages 274–280, 1999.

[JW94] C. Jiang and M.O. Ward. Shadow segmentation and classification in a constrained environment. *CVGIP: Image Understanding*, 59:213–225, 1994.

[KKT95] Krager, Klein, and Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42:321–328, 1995.

[Kod69] Kodak. *Kodak Wratten Filters 4th Edition*. Kodak Limited London, 1969.

[KSK88] G.J. Klinker, S.A. Shafer, and T. Kanade. Color image analysis with an intrinsic reflection model. In *Proc. of the second International Conference on Computer Vision*, pages 292–296, 1988.

[KSK90] G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding,. *International Journal of Computer Vision*, 4:7–38, 1990.

[Lan77] E.H. Land. The retinex theory of color vision. *Scientific American*, pages 108–129, 1977.

[LB05] M.D. Levine and J. Bhattacharyya. Removing shadows. *Pattern Recognition Letters*, 26:251–265, 2005.

[LD06] C. Lu and M.S. Drew. Practical scene illuminant estimation via flash/no-flash pairs. In *Proc. of the fourteenth Color Imaging Conference*, pages 1–1, 2006.

[LDB06] A. Leone, C. Distante, and F. Buccolieri. A shadow elimination approach in video-surveillance context. *Pattern Recognition Letters*, 27:345–355, 2006.

[LL97] S. Lin and S. Lee. Detection of specularity using stereo in color and polarization space. *computer Vision and Image Understanding*, 65:336–346, 1997.

[LM71] E.H. Land and J.J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971.

[Lu06] C. Lu. Removing shadows from color images. Ph.D. Thesis, Simon Fraser University, Vancouver, Canada, 2006.

[Man82] B. B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman, CA, 1982.

[MAU94] Y. Moses, Y. Adini, and S. Ullman. Face recognition, the problem of compensating for changes in illumination direction. In *Europ. Conf. on Comp. Vision (ECCV)*, pages 286–296, 1994.

[Maz02] Mazeworks. How to build a maze. http://www.mazeworks.com/mazegen/mazetut/index.htm, 2002.

[McC04] J.J. McCann. Capturing a black cat in shade: past and present of Retinex color appearance models. *Journal of Electronic Imaging*, 13(1):36–47, January 2004.

[MM00] W.Y. Ma and B.S. Manjunath. Edgeflow: a framework for boundary detection and image segmentation,. *IEEE Trans. on Image Processing*, 9:1375–1388, 2000.

[MMT76] J.J. McCann, S.P. McKee, and T.H. Taylor. Quantitative studies in retinex theory. a comparison between theoretical predictions and observer responses to the "color mondrian" experiments. *Vision Research*, 16:445–458, 1976.

[MNIS02] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi. Shadow elimination for robust video surveillance. In *Proc. of Worshop on Motion and Video Computing*, pages 15–21, 2002.

[MNIS04] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakaushi. Illumanation normalization with time-dependent intrinsic images for video survelliance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1336–1347, 2004.

[MO01] J.A. Marchant and C.M. Onyango. A color invariant for daylight changes: relaxing the constraints on illuminants. *Journal of The Optical Scociety of America part A*, 18:2704–2706, 2001.

[MS93] P.D. MacKenzie and Q.F. Stout. Optimal parallel construction of hamiltonian cycles and spanning trees in random graphs. In *Proc. of the 5th ACM Symposium on Parallel Algorithms and Architectures*, pages 224–229, 1993.

[MS00] M. Meila and J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 873–879, 2000.

[MW86] L.T. Maloney and B.A. Wandell. Color constancy: A method for recovering surface spectral reflectance. *Journal of The Optical Scociety of America part A*, 3:29–33, 1986.

[MW92] D.H. Marimont and B.A. Wandell. Linear models of surface and illuminant spectra. *Journal of The Optical Scociety of America part A*, 9:1905–1913, 1992.

[MZ93] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41:3397–3415, 1993.

[NB93] S.K. Nayar and R.M. Bolle. Computing reflectance ratios from an image. *Pattern Recognition*, 26:1529–1542, 1993.

[NFF02] S.M.C. Nascimento, F. Ferreira, and D.H. Foster. Statistics of spatial cone-excitation ratios in natural scenes. *Journal of the Optical Society of America A*, 19:1484–1490, 2002.

[OCDD01] B.M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *SIGGRAPH2001*, pages 433–442, 2001.

[OMP$^+$02] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex- new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 701–706, 2002.

[Out] Outex. http://www.outex.oulu.fi.

[PCF$^+$01] N. Petrovic, I. Cohen, B.J. Frey, R. Koetter, and T.S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *Proc. of the IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, pages 743–748, 2001.

[PGB03] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. In *SIGGRAPH2003*, pages 313–318, 2003.

[PJ89] J. Parkkinen and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of The Optical Scociety of America part A*, 6:318–322, 1989.

[PSA$^+$04] G. Petschnigg, R. Szeliski, M. Agrawala, M.F. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. on Graphics*, 23:664–672, 2004.

[PTMC03] A. Prati, M.M. Trivedi, I. Mikic, and R. Cucchiara. Detecting moving shadows: algorithms an devaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:918–923, 2003.

[RAGS01] E. Reinhard, M. Ashikhmnin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21:34–41, 2001.

[RAJ97] R.Harvey, A.Bosson, and J.A.Bangham. The robustness of some scale-spaces. In *Proc. of the British Machine Vision Conference*, pages 11–20, 1997.

[Ren87] A. Renyi. *A Diary on Information Theory*. Wiley-Interscience, 1987.

[RR82] J.M. Rubin and W.A. Richards. Color vision and image intensities: when are changes material? *Biological Cybernetics*, 45:215–226, 1982.

[Sab91] K.K. Sabelfeld. *Monte Carlo Methods : in Boundary Value Problems (Springer Series in Computational Physics)*. Springer-Verlag, 1991.

[Sap85] G. Sapiro. Color and illuminant voting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7:210–218, 1985.

[Sap98] G. Sapiro. Bilinear voting. In *Proc. of the International Conference on Computer Vision*, pages 178–183, 1998.

[SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[SF90] J.P Shu and H. Freeman. Cloud shadow removal from aerial photographs. *Pattern Recognition*, 23:647–656, 1990.

[SMO99] J. Stauder, R. Melch, and J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Trans. on Multimedia*, 1:65–77, 1999.

[Sou06] P. Southam. Texture analysis with the sieve. Ph.D. Thesis, University of East Anglia, Norwich, U.K., 2006.

[Sta79] I. Stakgold. *Green's Functions and Boundary Value Problems.* Wiley-Interscience, 1979.

[TFA03] M.F. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 1343–1350, 2003.

[TNI03] R.T. Tan, K. Nishino, and K. Ikeuch. Illumination chromaticity estimation using inverse-intensity chromaticity space. In *Proc. of the conference on Computer Vision and Pattern Recognition*, pages 417–421, 2003.

[TP91] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Proc. of Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[TW89] S. Tominaga and B.A. Wandell. Standard surface-reflectance model and illumination estimation. *Journal of The Optical Scociety of America part A*, 6:576–584, 1989.

[UL97] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *38th Annual Symposium on Foundations of Computing Sciences*, 1997.

[Uni89] Joensuu University. Musell colors matt [online]. http://spectral.joensuu.fi/databases/download/munsell_atof.htm, 1989.

[Uni02] Manchester University. Hyperspectral images [online]. http://personalpages.manchester.ac.uk/staff/david.foster/, 2002.

[vdWG05] J. van de Weijer and T. Gevers. Color constancy based on the grey-edge hypothesis. In *Proc. of the International Conference on Image Processing, Genoa*, pages 722–725, 2005.

[Wan95] B. A. Wandell. *Foundations of Vision.* Sinauer Associates, Inc., 1995.

[Wei01] Y. Weiss. Deriving intrinsic images from images sequences. In *International comference in Computer Vision (ICCV)*, pages 68–75, 2001.

[WHR91] C. Wang, L. Huang, and A. Rosenfeld. Detecting clouds and cloud shadows on aerial photographs. *Pattern Recognition Letters*, 12:55–64, 1991.

[Wil03] Arnold Wilkins. *Reading through colour: How coloured filters can reduce reading difficulty, eye strain, and headaches.* John Wiley, 2003.

[WS82] G. Wyszecki and W. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* Wiley, 1982.

[WT05] T.-P. Wu and C.-K. Tang. A bayesian approach for shadow extraction from a single image. In *Proc. of the tenth International Conference on Computer Vision*, pages 480–487, 2005.

[Wu77] F.Y. Wu. Number of spanning trees on a lattice,. *J. Phys. A: Math. Gen.*, 10:L113–L115, 1977.

[YKE02] J.J. Yoon, C. Koch, and T.J. Ellis. Shadowflash: an approach for shadow removal in an active illumination environment. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 626–645, 2002.