Bayesian Approaches to Bilinear Inverse Problems Involving Spatial Evidence: Color Constancy and Blind Image Deconvolution

by

Barun Singh

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Engineer of Computer Science

and

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

© Massachusetts Institute of Technology 2006. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
26, 2006

Certified by .....
William T. Freeman
ing
isor

Accepted by .....
nith
Chairman, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
NOV 0 2 2006
LIBRARIES

BARKER

Bayesian Approaches to Bilinear Inverse Problems Involving Spatial Evidence: Color Constancy and Blind Image Deconvolution

by

Barun Singh

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 2006, in partial fulfillment of the
requirements for the degrees of
Engineer of Computer Science
and
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis examines two distinct but related problems in low-level computer vision: color constancy and blind-image deconvolution. The goal of the former is to separate the effect of global illumination from other properties of an observed image, in order to reproduce the effect of observing a scene under purely white light. For the latter, we consider the specific instance of deblurring, in which we seek to separate the effect of blur caused by camera motion from all other image properties in order to produce a sharp image from a blurry one. Both problems share the common characteristic of being bilinear inverse problems, meaning we wish to invert the effects of two variables confounded by a bilinear relationship, and of being underconstrained, meaning there are more unknown parameters than known values. We examine both problems in a Bayesian framework, utilizing real-world statistics to perform our estimation. We also examine the role of spatial evidence as a source of information in solving both problems. The resulting blind image deconvolution algorithm produces state-of-the art results. The color constancy algorithm produces slightly improved results over the standard Bayesian approach when spatial information is used. We discuss the properties of and distinctions between the two problems and the solution strategies employed.

Thesis Supervisor: William T. Freeman
Title: Professor of Computer Science and Engineering

Acknowledgments

Thanks go first and foremost to my advisor, Bill Freeman. Not only were his insights and assistance critical in helping me conduct the research presented in this thesis, he was also particularly understanding of the fact that life can take myriad twists and turns that set you on new and exciting paths. He allowed me to make the most of my MIT graduate experience and I doubt most advisors would have been so supportive.

Thank you to all of my colleagues who have answered so many questions and helped me learn much more about computer science through casual conversations than I could have through any class. In particular, Marshall Tappen has been my officemate for the last four years and a great person to ask questions, discuss ideas with, and debate.

The deblurring research that constitutes about half of this thesis was conducted jointly with Rob Fergus, without whose tireless efforts that work likely would never have seen fruition. It was great to be able to work with him on the project.

The time I have spent in the AI/CSAIL lab has been made particularly enjoyable by all of the great people I get to interact with on a day-to-day basis, not just within my research group but also the other vision and graphics groups. More generally, I have had the fortunate opportunity to become friends with some spectacular people throughout this Institute and to all of you, I say thank you for being part of my life.

I will always owe my family, my mother and father in particular, an enormous debt of gratitude for all of the sacrifices they have made for me, for teaching me, and for guiding me so that I would have the opportunities I have today.

Words aren't anywhere near adequate to express my deep appreciation for my wife, Sara. Her intense love and support has seen me through the most difficult times in my life, and has made every day we're together seem like a blessing. Thank you, thank you, thank you.

This thesis is dedicated to my brother, Bhuwan. He lived his life filling the hearts of everyone around him with love, and left this earth far before he should have. I always looked up to him, and I can only hope to one day live up to the memory he has left behind.

During the time this research was conducted, I have been supported by the National Science Foundation, Shell Oil, the National Geospatial Intelligence Agency NGA-NEGI, the Nippon Telegraph and Telephone Corporation as part of the NTT/MIT Collaboration Agreement, and the MIT Office of the Dean for Graduate Students Ike Colbert.

Contents

Chapter 1

Introduction

We often take our ability to process visual stimuli for granted. In reality, however, images are very complex objects, and processing the information contained within them is a difficult task. The image formation process involves the characteristics of and interactions among the items in a scene, the light sources shining upon and within the scene, and the imaging device. Humans are able, for the most part, to separate out many or all of these effects into their component parts, an important step in being able to extract useful information from visual data. Examples of this include our ability to recognize objects within an image and ascertain their three dimensional nature, deduce the true shape and characteristics of objects in blurry images, differentiate between shadows and changes in material properties, determine the illumination-independent color of items in a scene, and so on.

Though it encompasses a number of sub-fields, most of computer vision can be thought of as ultimately seeking to replicate the abilities of the human visual system. It is often helpful to break down this general category of research into two sub-categories: low-level and high-level computer vision. Low-level vision problems are those that relate to the process of light being shone from a source, reflecting off a set of surfaces and being recorded by an imaging device, and how to understand the characteristics of that process. These often involve operations at the pixel or sub-pixel level. In contrast, a high-level vision problem involves understanding concepts about the scene being imaged, or objects within that scene. One can think of low-level vision as involving image processing, while high-level vision deals with image understanding.

Much of the challenge of solving a high-level vision problems is being able to define it in mathematical terms. For example, identifying how many cars are present in a given scene is intuitively obvious for people to understand, but defining some

mathematical notion of what a car looks like for use by a computer algorithm is very challenging. Low-level vision problems have a benefit over high-level problems in that they are usually very straightforward to describe mathematically. However, they are often very difficult to solve because nearly all of them are underconstrained. That is, there are more unknown parameters than there are known values. This thesis examines a particular class of underconstrained low-level problems known as bilinear inverse problems.

An inverse problem is one in which we observe some effect, and we wish to understand the causes that led to that effect. In computer vision, the effect is generally the observation of a particular image, and the causes are some set of properties of the scene and image capture device. A bilinear problem is one in which the unknowns are related to the known value by a bilinear function, i.e. a function with two unknowns that is linear with respect to each variable if the other is held constant.

The simplest example of an underconstrained bilinear inverse problem is

$$y = ab, \tag{1.1}$$

where we observe y and we wish to determine the values of the unknown variables a and b. Clearly, without any additional information, there is no way for us to obtain an answer, as there are two unknown variables and only one data point. We approach this sort of problem using Bayesian methods. Bayesian methods are built upon a probabilistic framework that allows us to incorporate statistical information about the world into our solution. In the types of problems we are concerned with, the unknown variables will represent properties of natural images, and as we will see, the statistics that govern these properties can provide very strong cues to solving low-level vision tasks.

This thesis examines in particular how statistics regarding the spatial configuration of natural images might help in low-level computer vision tasks. To do so, we consider two specific problems: color constancy and blind image deconvolution.

1.1 Color Constancy

Color is important in our understanding of the visual world and provides an effective cue for object detection and recognition. In general, however, the observed color of an object differs from its "true" color due to contextual factors such as lighting and orientation. An example of this can be seen in Figure 1 which shows the difference

between an outdoor scene as it would look under white light and as it actually appears at dusk. Color constancy refers to the ability to perceive the color of an object as approximately constant regardless of any such contextual factors. We are able to reliably use color for a variety of vision tasks largely because of our ability to perform color constancy well.

(a) Outdoor scene under white light

(b) Outdoor scene at dusk

Figure 1-1: Example of an outdoor scene under two different lighting conditions. In performing color constancy, we wish to determine what a given image would look like under some canonical illuminant (white light). In this particular example, we would want to obtain the image on the left given the image on the right.

The color constancy problem, even in its simplest formulation, is inherently underconstrained. The image formation process consists of an illumination reflecting off of a surface, then being passed through a set of sensors. The input to the sensors is the product of the illumination spectrum and the surfaces reflectance spectrum. Different combinations of illumination and reflectance spectra can produce the same spectrum impinging on the sensor, and various sensor inputs can result in the same sensor responses. Due to these ambiguities, it is generally impossible to uniquely separate the effect of the illumination and the surface reflectances in an image. The goal of computational color constancy, therefore, is to optimally (under some metric) estimate either the reflectance and illumination spectra, or the effect of these spectra, at each point in the image.

1.2 Blind Image Deconvolution

Blind image deconvolution is used to restore images that have been corrupted by some stationary linear process. The corrupted observation is the result of a convolution between the original image and some unknown point spread function (PSF). Image deconvolution is referred to as blind if the PSF is unknown, and non-blind if it is known.

Corruption caused by an unknown PSF is common in a variety of imaging applications, including astronomical and medical imaging. For natural images, the most common form of PSF is a blur kernel that is caused by camera shake during the image capture process (in which case the corrupted image is a blurry version of the original sharp image). An example is shown below in Figure 1-2. Camera shake is a significant problem for the average consumer-level photographer, particularly in situations where longer exposure times are needed to capture the scene because of low ambient light. In this thesis, blind image deconvolution will be synonymous with deblurring images that have been corrupted by camera shake.

(a) Actual (sharp) scene

(b) Camera motion

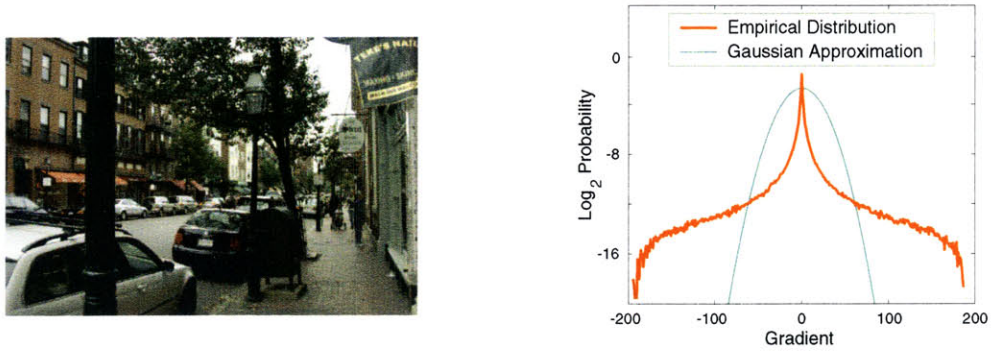
(c) Captured (blurry) image

Figure 1-2: Example of a blurry image caused by camera shake. Figure 1-2(a) shows what the image would have looked like if the camera remained still. The camera motion is shown in 1-2(b), where the magnitude of the grayscale value is inversely proportional to the length of time that the camera spent in a particular location. The resulting blurry image is shown in 1-2(c).

1.3 Spatial Characteristics of Natural Images

A great deal of recent work in computer vision has shown that images of real-world scenes exhibit particular spatial characteristics that can be helpful in performing a wide variety of tasks. Although natural images do not obey well-defined distributions in terms of their absolute color values, there is a great deal of regularity in the distribution of their gradients and their responses to a variety of linear filters. In particular, these distributions are heavy-tailed; that is, they have most of their mass close to zero, but they also have some probability mass at large values as well. A Gaussian distribution, in comparison, would have much less probability mass at these large values. This property describes our intuitive understanding of natural images: they consist primarily of regions that are mostly "flat" or have slight gradients due to shading, interrupted occasionally by abrupt changes due to albedo, occlusions, or edges.

Heavy-tailed natural image statistics have been very useful in producing state-of-the-art results for many computer vision tasks, including denoising [40], superresolution [43], estimating intrinsic images [46], inpainting [27], and separating reflections [26]. Encouraged by these results, this thesis explores if these types of statistics can be utilized to achieve better results when applied to the problems of color constancy and blind image deconvolution.



(a) A typical real-world image

(b) Distribution of image gradients for the scene at left

Figure 1-3: Example of heavy-tailed distribution in the gradients of a real-world image. The figure at right is shown on a logarithmic scale to emphasize the heavy-tailed nature of the distribution and its non-Gaussian nature.

1.4 Outline of Thesis

The remainder of this thesis is divided into four sections. The first of these gives an overview of Bayesian methods, introducing concepts and notation that are necessary in order to understand the remaining chapters.

The third chapter in this thesis discusses the color constancy problem in depth. This begins with an outline of the standard problem formulation and prior work. We then introduce the concept of spatio-spectral basis functions as a means of describing spatial statistics. We present the results of color constancy when using these spatio-spectral basis functions, and find that for this particular problem, the use of spatial information helps to an extent in synthetic examples, and less so in real-world images.

Chapter four deals with the blind image deconvolution problem. We again formulate the problem and discuss previous work. This is followed by an overview of algorithmic details involved in solving the problem, as well as a discussion of important factors that must be considered and dealt with in order to achieve optimal performance. We present results for the application of deblurring images corrupted

by camera shake, showing that the algorithm provides state-of-the-art performance for this application.

The final chapter discusses and compares the results of the color constancy and blind image deconvolution algorithms. In particular, we offer explanations as to why the use of spatial information assisted us in the case of deblurring but less so in the case of color constancy. We also compare the approaches utilized in solving both problems and describe inherent differences between the two problems that dictated the algorithmic choices made.

Chapter 2

Bayesian Methods

The work in this thesis relies on Bayesian methods. The primary characteristic of these methods is that they frame problems probabilistically and thus are able to produce solutions that incorporate a wide variety of statistical information, from the general characteristics of real-world processes to the assumed noise models for the particular problem at hand.

The probabilistic framework utilized by all Bayesian methods is based on three fundamental probability distributions — the prior, the likelihood and the posterior. Consider a problem of the form

$$y = f(\theta) + w \tag{2.1}$$

where y is a set of observed data, $\theta = \{\theta_{i\in[1,n]}\}$ is a set of n unknown variables that describe the inner state of the system we are observing, $f(\cdot)$ is some known function, and w is a stochastic noise term that corrupts our observation. Our goal is to determine the inner state of the system θ given a set of observations y.

We model the system through its posterior probability, $P(\theta|y)$. The posterior is a mathematical representation of the probability that the internal state θ takes on a specific set of values given that we make a particular observation y. It can be calculated using Bayes' rule as

$$P(\theta|y) = \frac{1}{Z}P(y|\theta)P(\theta), \tag{2.2}$$

where $P(y|\theta)$ is known as the likelihood, $P(\theta)$ is the prior probability, and $\frac{1}{Z}$ is a normalization constant that is independent of the parameters to be estimated.

The likelihood, given by $P(y|\theta)$, governs the relationship between the parameters

being estimated and the data. This includes both the functional relationship that maps the internal state to the observation in a noise-free environment, $f(\cdot)$, as well as the effect of the stochastic noise process.

The prior, given by $P(\theta)$, describes the probability that a certain set of values for θ is the correct estimate without taking into account what we have observed. Any constraints we have on the state variables (such as positivity or threshold requirements) are described by the prior. This term also encodes the statistics we would expect θ to observe in general (if we know nothing of these statistics, the prior will be uniform within whatever range of values θ is allowed to take).

The posterior combines the information encoded within both the prior and the likelihood, and is the most descriptive model of the system available to us. Given the posterior probability, there are a variety of methods to obtain what may be considered an "optimal" estimate. These methods all rely on the notion of minimizing some cost function dependent upon the posterior probability, but differ in the type of cost function used.

2.0.1 Probability Distributions

Throughout this thesis, we will utilize several forms of probability distributions to describe the statistics of various image and scene properties. For the purpose of clarity, these distributions are described below:

,

Gaussian A Gaussian distribution, also known as a normal distribution, over a random vector X is given by

$$\mathtt{G}(X|\mu,\Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right), \quad (2.3)$$

where $\mu = E[X]$ and $\Sigma = E[(X-\mu)(X-\mu)^T]$ are the mean and covariance of X, respectively. Note that all of the terms multiplying the exponential can be combined into one normalizing constant, and we will generally ignore this constant and simply use the relation that

$$\mathtt{G}(X|\mu,\Sigma) \propto \exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right). \quad (2.4)$$

In the case where X is a single random variable, μ and Σ are both scalars.

Truncated Gaussian

We shall refer in this thesis to a truncated Gaussian distribution as a distribution over a random vector X that obeys the following:

$$\mathtt{G}^{(\mathtt{Trunc})}(X|a,b,\boldsymbol{A},B) \propto \begin{cases} \mathtt{G}(X|a,b) & \text{if } 0 \leq \boldsymbol{A}X + B \leq 1 \\ 0 & \text{otherwise} \end{cases}, \quad (2.5)$$

for some matrix $\boldsymbol{A}$ and vector B. Unlike the standard Gaussian distribution, none of the parameters of the truncated Gaussian distribution correspond directly to the mean or variance of the distribution.

Rectified Gaussian

A rectified Gaussian distribution shall refer in this thesis to a specific instance of the truncated Gaussian distribution in which we are dealing with a single random variable x, $\boldsymbol{A} = 1$, and $B = 0$. This results in a distribution similar to the standard Gaussian distribution over a random variable except that it enforces positivity:

$$\mathtt{G}^{(\mathtt{R})}(x|a,b) \propto \begin{cases} \exp\left(-\frac{b}{2}(x-a)^2\right) & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

Exponential

The exponential distribution is defined over a random variable x as

$$\mathtt{E}(x|\lambda) = \begin{cases} \lambda \exp(-\lambda x) & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (2.7)$$

where λ is known as the inverse scale parameter.

Gamma

The gamma distribution is defined over a random variable x as

$$\mathtt{Gamma}(x|a,b) = \begin{cases} \frac{a^b}{\Gamma(b)} x^{(b-1)} \exp(-ax) & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (2.8)$$

where $a > 0$ is the shape parameter and $b > 0$ is the scale parameter.

Mixture Models

A mixture model does not refer to a specific form for a probability distribution. Rather, it refers to a method of combining many distributions of the same functional form, known as mixture components, into a single distribution over the random vector X:

$$\mathtt{Mixture}(X) = \prod_{i=1}^{N} \pi_i P(X|\theta_i), \quad (2.9)$$

where N is the number of mixture components being combined, π_i is a weight assigned to component i, and θi is the set of parameter values associated with component i, and

$$\sum_{i-1}^{N} \pi_i = 1. \tag{2.10}$$

$P(X|\theta_i)$ may have any functional form so long as it is a valid probability distribution and all mixture components have the same functional form (with different parameter values for each). In this thesis we will be using mixtures of Gaussians and mixtures of exponentials.

2.1 Estimation by Minimizing the Bayesian Expected Loss

The most common Bayesian estimation techniques operate by minimizing the Bayesian expected loss:

$$L(\theta'|y) = \int_{\theta} C(\theta, \theta') P(\theta|y) d\theta, \tag{2.11}$$

$$\theta_{opt} = \min_{\theta'} L(\theta'|y), \tag{2.12}$$

where $C(\theta, \theta')$ is a cost function that defines a penalty associated with a parameter estimate of θ' when the true value is θ, and θ_{opt} is the optimal estimate.

Maximum a posteriori (MAP) estimation is one specific instance within this family of approaches in which

$$C_{MAP}(\theta, \theta') = -\delta(\theta - \theta'), \tag{2.13}$$

$$\theta_{MAP} = \max_{\theta} P(\theta|y). \tag{2.14}$$

That is, MAP estimation produces the set of parameter values that, as its name would suggest, maximize the posterior probability. A very closely related technique is maximum likelihood (ML) estimation, which is equivalent MAP estimation under the assumption that the prior $P(\theta)$ is uniform. In this case, the prior has no effect on the shape of the posterior, so that the set of parameters that maximizes the posterior are the same parameters that maximize the likelihood.

Another commonly used estimation technique is minimum mean squared error

(MMSE) estimation, defined by the cost function:

$$C_{MMSE}(\theta, \theta') = (\theta - \theta')^2. \tag{2.15}$$

The value that minimizes the minimum mean squared error is the expected value of the probability distribution, $\theta_{MMSE} = E[\theta]_{P(\theta|y)}$.

Which of these techniques is best to use depends upon the problem being considered. An advantage to the MAP solution is that it does not require any additional computation beyond knowing the posterior probability. However, there may not be a unique solution when using the MAP approach. This is often true in the case of underconstrained problems in which there are more unknowns than known values. For these problems, there may be an infinite number of solutions that explain the data equally well, and many of these solutions may have the same prior probability as well — hence, it is possible that there is not a unique set of parameter values for which the posterior probability is maximized.

Conversely, the minimum mean squared error estimate is in general unique but it is also computationally very expensive, requiring an integration over the entire posterior probability distribution for every point that is evaluated. An additional downside to it is that it can be very sensitive to small amounts of probability mass located at extreme points in the parameter space.

Other techniques that minimize the Bayesian expected loss have also been proposed, including maximum local mass (MLM) estimation, for which the cost functions is given by

$$C_{MLM}(\theta, \theta') = -\exp(-|K_L^{-\frac{1}{2}}(\theta - \theta')|^2), \tag{2.16}$$

and K_L is set ahead of time. The MLM solution, like the MMSE estimate, requires an integration over the entire posterior probability distribution if computed exactly. However, good approximations exist which allow much more efficient computation that takes advantage of the fact that the MLM cost function only has non-negligible values in a very localized region [6]. The MLM estimate still requires more computation than the MAP estimate, but it has the advantage that by taking into account the value of the posterior probability distribution in the region surrounding a certain estimate, it is more likely to provide a unique estimate without being affected by minor probability mass at the edges of the parameter space.

Figure 2-1 shows a toy example of a one-dimensional posterior probability that

illustrates the differences between the MAP, MMSE, and MLM approaches.

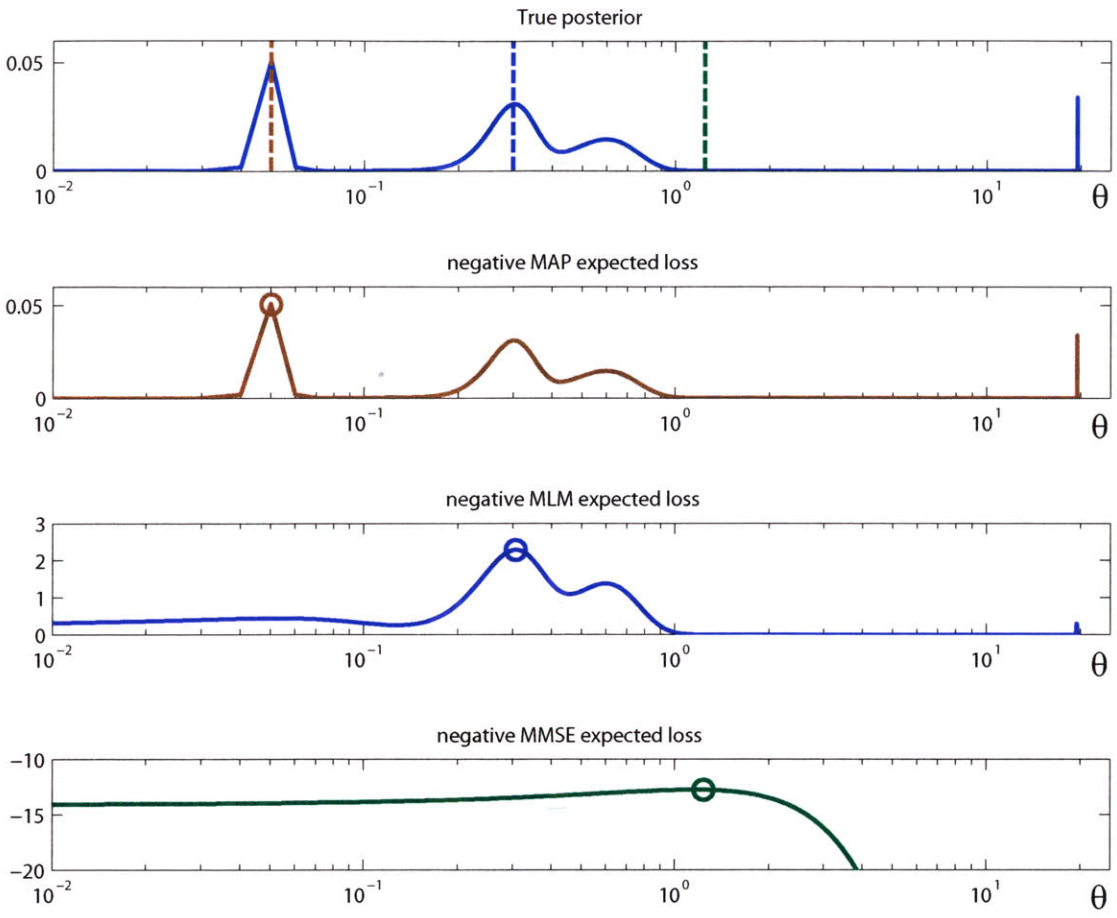


Figure 2-1: Example illustrating the effect of using different cost functions when minimizing the Bayesian expected loss. The top blue curve is a simple toy posterior probability distribution, as a function of a single scalar parameter θ. The three curves below illustrate the negative of the cost function being minimized when utilizing the maximum a posteriori (MAP), maximum local mass (MLM) and minimum mean square error (MMSE) approaches. The optimal solution is shown as a circle on these graphs, and the dashed lines on the top curve show the "optimal" estimate for θ when using these methods. The x-axis for all plots is shown with a logarithmic scale in order to make the differences between the three methods more readily visible.

2.1.1 Implementation and Local Minima

In the discussion above, we have assumed that it is possible to find the true minimum of a given cost function. In reality, though, this can be very difficult. The form of the posterior probability is unknown, so finding the optimal solution for a given problem requires searching the entire space of θ in order to evaluate the cost function at every

point. In toy examples where θ is one dimensional, such as the one illustrated in Figure 2-1, this may be possible. In real-world applications, however, θ can often have millions of dimensions. Consider, for example, the fact that an average consumer-level digital camera produces an image with 4 million or more pixels. In a computer vision application, each of these pixels may have a unique θ_i parameter associated with it (if we deal with color images, there may actually be three parameters for every pixel value associated with the red, green and blue color channels). In such a case, it is simply impossible to search the entire space of all pixels for the specific set of values that produces the maximum posterior probability, much less to evaluate a cost function at every point.

The construction of efficient algorithms for searching high-dimensional parameter spaces is a widely studied field of research. All existing approaches are, to some extent, still susceptible to local minima (or maxima). These local extrema are often characterized by strong nearby gradients that prevent search routines from being able to find more optimal solutions far away in the search space.

In real-world computer vision problems, the true posterior probability is generally plagued by many local maxima scattered throughout the search space. Initializations can play an important role in both the MAP and MLM approaches, and dimensionality reduction methods, if they can be applied, may help. The MMSE approach, because it requires a global integration, is usually intractable in these applications. In the color constancy problem described in the next chapter, the problem is simply reformulated in a manner that greatly reduces the dimensionality of the search space (though problems of local minima can and generally do still exist even in this lower dimensional space).

2.2 Variational Bayesian Approach

An alternate way of looking at the maximum a posteriori approach described above is that it attempts to approximate the true posterior with a delta function, and the location of this delta function is the MAP estimate. Variational Bayesian methods [1, 33] build upon this idea and instead seek to fit more descriptive functional forms to the true posterior, and apply the methods of the previous method to this functional form. This results in an approach that is sensitive to probability mass rather than probability value.

Let us denote by $Q(\theta)$ an approximation to the true posterior probability distribution. The variational approach seeks to find the optimal approximation $Q_{opt}(\theta)$

that minimizes the Kullback-Leibler (K-L) divergence, D_{KL}, between the two distributions:

$$Q_{opt}(\theta) = \min_{Q(\theta)} D_{KL}(Q||P) \tag{2.17}$$

$$= \min_{Q(\theta)} \int_{\theta} Q(\theta) \log \left[\frac{Q(\theta)}{P(\theta|y)} \right] d\theta. \tag{2.18}$$

The K-L divergence is a measure of the "distance" between two probability distributions, and is non-negative with $D_{KL}(Q||P) = 0$ if and only if $P = Q$. Use of the K-L divergence has foundations in information theory, and $D_{KL}(Q||P)$ can be described as the difference between the cross-entropy of P and Q and the entropy of P. In the context of information coding the K-L divergence tells us the number of extra bits that must be transmitted to identify a particular value of a signal if the code being used corresponds to a probability distribution Q instead of the true distribution P.

In general, it is often desirable to obtain a single estimate for θ as a solution. Because we know the analytical form of $Q(\theta)$, marginalizing over the distribution becomes tractable. This allows us to calculate the mean of the optimal approximation, which is often taken to be the optimal estimate:

$$\theta_{opt} = \mathrm{E}[Q(\theta)]. \tag{2.19}$$

However, this need not be the case. The solution presented in Equation 2.19 is the MMSE estimate over the approximate distribution $Q(\theta)$. We may alternatively choose to apply any of the other methods described in Section 2.1 to produce a single point estimate simply by replacing $P(\theta|y)$ in Equation 2.11 with $Q(\theta)$. In this thesis we will be utilizing MMSE estimation in conjunction with the variational approach, so the optimal solution will be as defined above.

Figure 2-2 shows the result of using the variational approach on the same posterior probability used earlier in Figure 2-1. In this example, a Gaussian distribution is chosen for $Q(\theta)$.

2.2.1 Probability Mass Versus Probability Value

One of the primary advantages to the variational approach is that by fitting a distribution to the posterior it is able to describe where the probability mass is located, and the mean of the variational approximation provides the center location of this mass. This differs from the maximum a posteriori approach, which (in the ideal case)

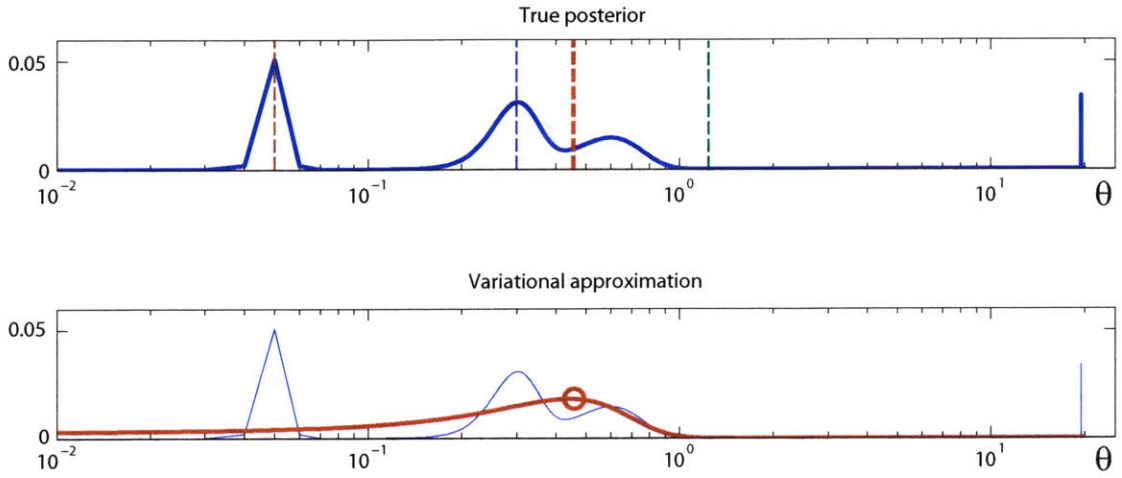


Figure 2-2: Example illustrating the effect of approximating the posterior probability with a Gaussian when using the variational approach. The top curve shows the same toy posterior probability illustrated in Figure 2-1. The light dashed lines are also the same as in Figure 2-1, illustrating the "optimal" estimate when using the MAP, MLM, and MMSE approaches. The thick dashed line shows the optimal estimate under the variational approach. This is identical to the mean of the approximate Gaussian distribution, which is shown in red on the bottom graph. Note that the Gaussian does not have the standard "bell" shape only because it is plotted along a logarithmic axis. The Gaussian is centered along the area of greatest probability mass of the posterior.

provides the single highest probability value location of the posterior. In general, describing the probability mass can oftentimes be more useful than describing the single point of highest probability.

The maximum local mass estimate is also more sensitive to probability mass than the MAP approach. However, the MLM technique is only sensitive to the probability mass in the local vicinity of a given value for the parameters, whereas the variational approach is sensitive to the probability mass of the posterior distribution as a whole. An alternate way of viewing the MLM approach is that it attempts to find the set of parameter values whose local vicinity is best described by a Gaussian with a pre-selected variance K_L (as defined in equation 2.16). This is similar to the variational approach in the case where $Q(\theta)$ is chosen to be Gaussian, except that the variational approach treats the variances as a free parameter to be estimated rather than a fixed value. Thus, the variational approach provides increased flexibility over the maximum local mass approach in order to find a more descriptive solution.

To illustrate how the MLM and variational approaches are related, suppose we run the variational Bayes algorithm using a Gaussian for $Q(\theta)$ and a solution of (θ_{var}, K_{var}) as the mean and covariance of this Gaussian. If we then run the MLM algorithm with $K_L = K_{var}$, we would get a solution of $\theta_{MLM} = \theta_{var}$. That is, the optimal estimate under the maximum local mass estimate is identical to the mean of the variational approximation if the MLM variance is set to the variance of the variational approximation. In actuality, though, when running the MLM algorithm, we would not know what K_L should be set to.

2.2.2 Implementation

As with the other techniques discussed in Section 2.1, there are limitations to what can be done with the variational approach in order to make it feasible to implement.

We wish to produce a set of analytic update equations that can be used to optimize the settings for the approximate posterior distribution $Q(\theta)$ (determining the values of these parameters through a search would be impractical even for small problems). In order to do this, we start with the assumption that the priors for each of our parameters are independent:

$$P(\theta) = \prod_i P(\theta_i). \tag{2.20}$$

We also model our approximate posterior distribution as being separable according

to the parameters:

$$Q(\theta) = \prod_i Q(\theta_i). \tag{2.21}$$

In this case, Equation 2.22 can also be rewritten as:

$$Q_{opt}(\theta) = \min_{Q(\theta)} \int_\theta Q(\theta) \log\left(\frac{Q(\theta)}{P(\theta|y)}\right) d\theta, \tag{2.22}$$

$$= \min_{Q(\theta)} \int_\theta Q(\theta) \log\left(\frac{Q(\theta)}{P(y|\theta)P(\theta)}\right) d\theta, \tag{2.23}$$

$$= \min_{Q(\theta)} \int_\theta \left[\prod_i Q(\theta_i)\right] \log\left(\prod_i \frac{Q(\theta_i)}{P(y|\theta)P(\theta_i)}\right) d\theta, \tag{2.24}$$

$$= \min_{Q(\theta)} \int_\theta \left[\prod_i Q(\theta_i)\right] \left[\sum_i \log\left(\frac{Q(\theta_i)}{P(\theta_i)}\right) - \log P(y|\theta)\right] d\theta. \tag{2.25}$$

We can then marginalize over all but one of the parameters to get

$$Q_{opt}(\theta_i) = \min_{Q(\theta_i)} \int_{\theta_i} Q(\theta_i) \left[\log Q(\theta_i) - \log P(\theta_i) - \langle \log P(y|\theta)\rangle_{Q(\theta|\theta_i)}\right] d\theta_i, \tag{2.26}$$

where $\langle\cdot\rangle$ represents the marginalization operation, i.e.,

$$\langle \log P(y|\theta)\rangle_{Q(\theta|\theta_i)} = \int_{\theta_{j,j\neq i}} Q(\theta|\theta_i) P(y|\theta) d\theta_{j,j\neq i}. \tag{2.27}$$

Solving equation 2.26 yields

$$Q_{opt}(\theta_i) = \frac{1}{Z_i} P(\theta_i) \exp\left(\langle \log P(y|\theta)\rangle_{Q(\theta|\theta_i)}\right), \tag{2.28}$$

$$Z_i = \int P(\theta_i) \exp\left(\langle \log P(y|\theta)\rangle_{Q(\theta|\theta_i)}\right) d\theta_i, \tag{2.29}$$

where Z_i acts as a normalizing term [33].

Equation 2.28 does not have a closed form solution. However, for particular forms of the prior probability, likelihood, and approximate posterior, iterative update equations may be derived that are practical to implement on real-world problems. These probability distributions must belong to the exponential family, as defined by

$$P(x) \propto \exp\left(\sum_i \alpha_i f_i(x)\right). \tag{2.30}$$

There are only particular combinations of distributions in this family that will lead to iterative analytic solutions. Luckily, many of these combinations are practical for

use in real-world problems, and in Chapter 4 we will describe how the variational approach can be used for blind image deconvolution.

Chapter 3

Color Constancy

3.1 Standard Problem Formulation

Maloney [31] categorizes color constancy algorithms into two general forms: those that presuppose a "Flat World" environment and those that allow for a "Shape World" environment. The former class of algorithms do not account for the effects of geometry, assuming that the true scene is flat. In this case, each point in the scene is fully described by a reflectance spectrum, a function only of wavelength. Any effects of geometry are folded into this reflectance spectrum. By contrast, the "Shape World" class of algorithms attempt to provide three dimensional models of the world, describing each point in a scene using a bidirectional reflectance density function which is a function of wavelength and geometry. Such algorithms generally assume that something is already known about the geometry of the scene, and the models they utilize are not able to describe the complexities of real world scenes. In this work, we refer to color constancy in the context of the "Flat World" environment.

In order to frame the color constancy problem, it is first necessary to clearly articulate the process by which an image is formed. We assume that there is one uniform illuminant in the scene. All effects other than this uniform illumination are considered to be due to the surface reflectances within the scene.

We assume image sensors that respond only within the visible range of 400nm to 700nm, and so are only concerned with the components of reflectance and illumination spectra that fall within that range. In order to make computations possible, we consider discretizations of these spectra, sampled at 10nm intervals to yield vectors of length 31.

We assume that the light spectrum leaving each surface is the term-by-term prod-

uct of the illumination and local reflectance spectrum. Let λ index the discrete wavelength samples. For a surface reflectance $S(\lambda)$ and illumination $E(\lambda)$, the response at position x of a photoreceptor with a spectral response of $R_k(\lambda)$ is

$$y_k^x = \sum_\lambda R_k(\lambda)E(\lambda)S^x(\lambda), \tag{3.1}$$

where we have assumed zero sensor noise.

The illumination and surface reflectance spectra can be written as linear combinations of a set of illumination basis functions $E_i(\lambda)$ and reflectance basis functions $S_j(\lambda)$, with coefficients e_i and s_j^x, respectively, at position x:

$$E(\lambda) = \sum_{i=1}^{L} E_i(\lambda)e_i, \tag{3.2}$$

$$S^x(\lambda) = \sum_{j=1}^{L} S_j(\lambda)s_j^x, \tag{3.3}$$

where $L = 31$ is defined as the number of elements in the λ vector (i.e., the number of wavelength samples). Doing so allows us to write the rendering equation as

$$y_k^x = \sum_\lambda R_k(\lambda) \sum_{i=1}^{L} E_i(\lambda)e_i \sum_{j=1}^{L} S_j(\lambda)s_j^x. \tag{3.4}$$

Summing over λ, we get a bilinear form,

$$y_k^x = \sum_{i=1}^{L}\sum_{j=1}^{L} e_i G_{ij,k} s_j^x, \tag{3.5}$$

where $G_{ij,k} = \sum_\lambda R_k(\lambda)E_i(\lambda)S_j(\lambda)$.

Given training data, principal components analysis (PCA) can be used to calculate the reflectance and illumination basis functions. An n-dimensional reconstruction of the training data using these bases is optimal in the sense that it provides the minimum mean-squared error between the actual data and its reconstruction as compared to any other n-dimensional reconstruction.

The number of basis functions used in the linear models of (3.2) and (3.3) are equal to the number of wavelength samples, so that the model fully describes the entire illumination and reflectance spectra exactly. In general, however, we wish to approximate the true illuminant $E(\lambda)$ and surface spectra $S^x(\lambda)$ using lower dimen-

sional linear models. Thus, we can define

$$\tilde{E}(\lambda) = \sum_{i=1}^{d_E} F_i(\lambda) e_i, \tag{3.6}$$

$$\tilde{S^x}(\lambda) = \sum_{j=1}^{d_S} S_j(\lambda) s_j^x, \tag{3.7}$$

where d_E is the dimensionality of the illuminant approximation $\tilde{E}(\lambda)$, d_S is the dimensionality of the surface approximation $\tilde{S^x}(\lambda)$, and $0 < (d_S, d_E) < L$. We now decompose (3.5) as:

$$\begin{aligned} y_k^x &= \sum_{i=1}^{d_E}\sum_{j=1}^{d_S} e_i G_{ij,k} s_j^x + \sum_{i=1}^{d_E}\sum_{j=d_S+1}^{L} e_i G_{ij,k} s_j^x + \sum_{i=d_E+1}^{L}\sum_{j=1}^{d_S} e_i G_{ij,k} s_j^x + \sum_{i=d_E+1}^{L}\sum_{j=d_S+1}^{L} e_i G_{ij,k} s_j^x \\ &= \sum_{i=1}^{d_E}\sum_{j=1}^{d_S} e_i G_{ij,k} s_j^x + w_k^x, \end{aligned} \tag{3.8}$$

where w_k^x is the aspect of the sensor response that is not described by the lower dimensional approximations to the true illumination and reflectance. We refer to this as the "model noise" for the remainder of this thesis. Mathematically, this model noise can be treated in the same manner as observation noise produced by the camera. An outline of all the symbol notation introduced above is given in Table 3.1.

While lower dimensional linear models allow for fewer parameters to be computed, they also produce larger amounts of model noise. Many researchers have examined the question of what dimensionality is required for these linear models in order to provide a "good" description of the data. For surfaces, it has been shown [30, 31, 35] that between 5 and 8 (depending on which data set was used) basis functions are needed to get a representation that accounts for greater that 99.5% or so of the variance of all surface reflectance spectra. Using as few as 3 basis functions can account for greater than 98% of the variance in these spectra. For natural illuminants, a 3 dimensional linear model accounts for 0.9997 of the variance according to [38]. Artificial lights, however, are not found to be very well fit by linear models since their spectra are not smooth.

Symbol	Symbol Definition
N	number of pixels in an image
λ	wavelength
L	number of wavelength samples (length of λ vector)
$E(\lambda)$	spectral radiance of illuminant
$E_i(\lambda)$	ith basis function of illumination
e_i	scalar weight of ith illumination basis function
e	row vector of all e_i
d_E	dimensionality of illumination representation
$\tilde{E}(\lambda)$	approximation to illuminant from d_E dimensional linear model
$S^x(\lambda)$	surface reflectance at point x
$S_j(\lambda)$	jth basis function of surface reflectance
s_j^x	scalar weight of jth surface reflectance basis function at position x
s^x	column vector of s_j^x for all j
s	matrix whose columns are given by s^x
d_S	dimensionality of surface reflectance representation
$\tilde{S}^x(\lambda)$	approximation to surface reflectance at position x from d_S dimensional linear model
$R_k(\lambda)$	spectral sensitivity of kth sensor
$R(\lambda)$	matrix of $R_k(\lambda)$ for all k
y_k^x	scalar response of kth sensor at x
y^x	vector of y_k^x for all k
y	matrix of y^x vectors for all x
$G_{ij,k}$	3-d tensor relating illumination and reflectance weights to sensor responses
w_k^x	model noise contained in y_k^x

Table 3.1: Symbol Notation for Standard Color Model

3.2 Related Work

Color constancy has a long history in the image processing community, and the many existing approaches may be distinguished from one another not only by the differences between their solution strategies, but also by their formulation of the problem and the assumptions that they make. In this section we describe the most significant approaches found in the literature. Some of these methods (gray world, subspace, and Bayesian) model the image formation process explicitly and attempt to determine the full illumination and reflectance spectra at every point in the image. Others (gamut mapping, retinex, neural networks, color by correlation) attempt to find a single mapping from the input RGB image to the color corrected image without ever dealing explicitly with spectra. In this thesis we shall refer to these as inverse-optics and transformational approaches, respectively.

One approach to color constancy is to simply reformulate the problem so that it is not underconstrained. The subspace method does this by assuming the reflectance spectra can be adequately modeled using only two coefficients rather than three [32, 45]. Given a trichromatic visual system and a three-dimensional illumination, an N pixel image contains $3N$ data points and, under the subspace assumption, only $2N + 3$ unknowns. In theory, we should be able to solve for all data points exactly. This method fails in real images, however, because of the model noise (we will revisit the subspace method in Section 3.4.2).

The gray world algorithm [8] is an inverse-optics approach that assumes the average reflectance of an image scene is equal to some known spectrum or, as an alternate formulation, that the average sensor response is equal to some constant value. In either case, the additional constraint allows for a straightforward solution. However, because the gray world assumption is not very accurate one, solutions obtained through this method are not very accurate.

Retinex is a transformational algorithm that relies on the assumption that illumination causes only slowly varying color changes in an image, and attempts to construct a color-corrected image by removing this effect [25, 24, 19]. The original implementation of Retinex takes random walks across an image starting at randomly chosen pixels. The sensor responses at every pixel along a walk are scaled by the value of the starting pixel, and all ratios that are close to unity are set equal to 1. The output of the algorithm is the average of all scaled and thresholded values at each pixel. With the exception of the thresholding effect, the remaining computations amount to a scaling of the image by the brightest sensor value within the image. Other variants

of the Retinex algorithm have been presented which essentially differ from the original in that they scale the image by the mean (rather than the maximum) sensor value.

An alternate approach to color constancy is to take advantage of physical constraints in the imaging system. Gamut mapping approaches [14, 13] utilize the property that the gamut of all possible RGB sensor responses under a given illumination can be represented by a convex hull. They model the effect of an illumination change as a linear transformation in RGB space, so that the convex hull of RGB values under one illuminant is related by a linear mapping to the convex hull of RGB values under another illumination. Thus, the illumination of a scene is estimated by finding the linear transformation that best maps the convex hull of the observed sensor responses to the known convex hull of all real world objects under white light.

The neural network approach [15, 16] examines the color constancy problem as if it were a classification problem. The image is transformed from the 3-d RGB color space into a 2-d RG-chromaticity space (in which the R and G channels are scaled by the sum of all three channels so that all values are between 0 and 1). A discrete histogram of the RG-chromaticity values are fed into a neural network which is trained to output the RG-chromaticity values of the illuminant. While this approach has produced some promising results, all of the assumptions it makes are implicit within the neural network and it is therefore difficult to understand exactly what it is doing.

The Bayesian approach to color constancy was introduced by Brainard and Freeman [6] as a means of incorporating both image statistics and physical constraints in an explicit manner. They began from the inverse-optics formulation of the problem and utilized truncated Gaussian distributions as priors on illumination and surface reflectance coefficients. In order to make estimation feasible, they performed estimation over the illumination coefficients using the maximum local mass technique, and solved for the surface reflectance coefficients at every iteration given the illuminant at that iteration. Nearly all of the other color constancy algorithms can be seen as a special instance of the Bayesian approach. Results using the general Bayesian formulation, however, have only been shown for synthetic images. Difficulties in modeling natural image statistics and implementation issues have prevented this approach from being used more extensively on real-world images.

Color by correlation is a transformational algorithm that borrows from the Bayesian approach. It uses a correlation matrix to estimate the scene illuminant given an observed image. This correlation matrix is equivalent to a discrete nonparametric likelihood function that describes how likely a set of surfaces are given a particular

illumination. Recent work more clearly reformulates the color by correlation method in a Bayesian framework and has been shown to achieve very good results [39]. This work explicitly models the imaging process as a linear transformation in RGB space with diagonal transformation matrix and incorporates a prior over the color-corrected image as well as a prior over illuminations that enforces physical realizability.

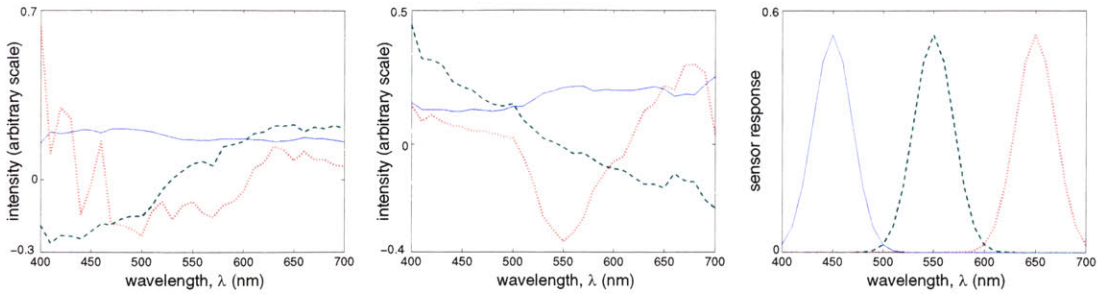
Many of the methods described above incorporate some form of information or statistic about the overall set of values in the reflectance image or the color-corrected image. None of them, however, incorporate any information about local image statistics of the form described in Section 1.3. We wish to examine if natural image statistics that describe this local structure can assist in color constancy. To do so, we will build on the Bayesian color constancy algorithm of [6] as this method can most easily be extended to use additional statistical information.

3.3 Hyperspectral Data

To perform simulations, we use a hyperspectral data set of 28 natural scenes collected at Bristol University by Brelstaff, et. al. as described in [7]. Each of these images is 256×256 pixels in size and contains sensor responses in 31 spectral bands, ranging from 400nm to 700nm in 10 nm intervals. Each scene also contains a Kodak graycard at a known location with a constant reflectance spectrum of known intensity. The scene illuminant is approximated as the spectrum recorded at the location of the graycard divided by its constant reflectance. The reflectance is then found by factoring out the illumination from the given hyperspectral signal (the reflectance images had to be thresholded to be less than or equal to one in some places). Our dataset therefore consists of 256x256x28 reflectance spectra and 28 illumination spectra.

The first three illumination and reflectance basis functions, obtained by applying principal components analysis (PCA) to this data, are plotted in Figure 3-1(a) and 3-1(b), respectively (PCA is performed on all 28 illumination spectra and approximately half of the available reflectance spectra). We assume, without loss of generality, a Gaussian model for sensor responses centered at 650nm (red), 550nm (green), and 450nm (blue) as shown in Figure 3-1(c). Sample images from the dataset, after being passed through these Gaussian sensors, are shown in Figure 3-2.

Normalized histograms of the coefficients corresponding to the first three reflectance basis functions are shown in Figure 3-3. The best fitting Gaussian distributions are shown as well. It can be seen that all of the coefficients associated with the first basis function are positive. This is as we would expect, since the first basis

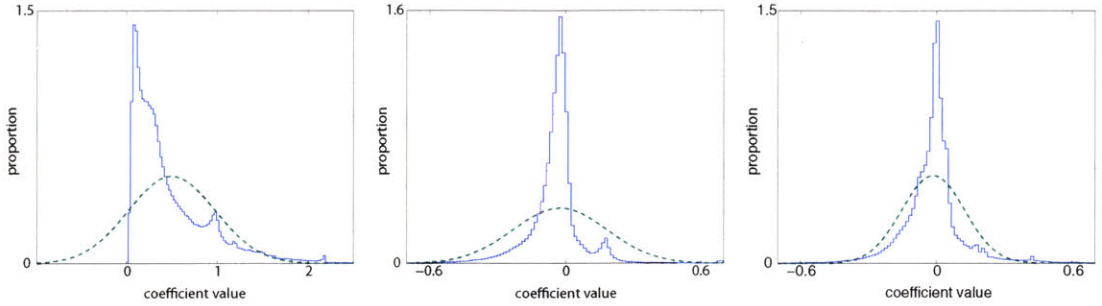


(a) Illumination Basis Functions (b) Reflectance Basis Functions (c) Sensor Responses

Figure 3-1: Linear models of reflectance and illumination obtained from the hyperspectral dataset, and assumed linear sensor responses.

Figure 3-2: Sample images from dataset of 28 hyperspectral images after being passed through the sensors of Figure 3-1(c). The hyperspectral images contain information from 31 spectral bands at each pixel, and the graycards in each of the images is used to find the true illumination of the scene.

function is essentially constant (thus acting as a bias term), and all reflectance spectra must be nonnegative. The distributions for the next two coefficients are centered close to zero, and exhibit much greater kurtosis than the Gaussian approximation (as indicated by the "peakiness" of the histograms).



(a) Basis 1 Coefficients (b) Basis 2 Coefficients (c) Basis 3 Coefficients

Figure 3-3: Histograms, obtained from the full hyperspectral dataset, of the coefficients corresponding to the PCA reflectance basis functions shown in Figure 3-1(b). The best fit Gaussian (dashed) probability distributions are also shown.

3.4 Using Spatial Information: Spatio-Spectral Basis Functions

Although the existing Bayesian approach to color constancy allows a great deal of flexibility, it is limited to considering all image pixels to be independent of one another. In reality, the spatial configuration of image pixels are governed by physical properties of the world and are far from independent. By ignoring the spatial statistics of images, the existing approach does not utilize all of the information available.

The existing approach can be modified to incorporate the relationship between neighboring pixels by using linear models to describe the reflectance of groups of pixels, rather than individual pixels. Without loss of generality, we can group pixels into $m \times m$ blocks and use the same basic formulation as was developed in Section 3.1. In order to do so, it is necessary to convert blocks of pixels into vector format. We do this by rasterizing the pixels within the block. The reflectance of a block of pixels is defined as a vector of length m^2L consisting of the reflectance of each pixel within the block stacked on top of each other in raster order. The same process is used to describe the sensor response of the block as a vector of length m^2K and the illumination as a vector of length m^2L.

The basis functions used to model the reflectances of the blocks of pixels are now referred to as *spatio-spectral* reflectance basis functions, since they implicitly describe both the spectral and spatial (within an $m \times m$ block) characteristics of a block of pixels.

We shall denote a group of pixels by $\boldsymbol{x}$, so that the illumination and reflectance of a block of pixels is given by:

$$\boldsymbol{E}(\gamma) = \sum_{i=1}^{m^2 L} \boldsymbol{E}_i(\gamma)\boldsymbol{e}_i, \tag{3.9}$$

$$\boldsymbol{S}^{\boldsymbol{x}}(\gamma) = \sum_{j=1}^{m^2 L} \boldsymbol{S}_j(\gamma)\boldsymbol{s}_j^{\boldsymbol{x}}, \tag{3.10}$$

where $\boldsymbol{E}_i(\gamma)$ and $\boldsymbol{S}_j(\gamma)$ are the spatio-spectral illumination and reflectance basis functions, $\boldsymbol{e}_i$ and $\boldsymbol{s}_j^{\boldsymbol{x}}$ are the weights associated with these basis functions, $\boldsymbol{E}(\gamma)$ is the illumination of all blocks in the image, and $\boldsymbol{S}^{\boldsymbol{x}}(\gamma)$ is the reflectance of the block of pixels $\boldsymbol{x}$. Note that the elements of the scene are now written as a function of γ rather than λ. This is due to the fact that the spatio-spectral representation contains information about both the frequency and spatial characteristics of the scene, so it cannot be parameterized by the wavelength (λ) alone. Approximating these models with fewer dimensions, we can define

$$\tilde{\boldsymbol{E}}(\gamma) = \sum_{i=1}^{\boldsymbol{d_E}} \boldsymbol{E}_i(\gamma)\boldsymbol{e}_i, \tag{3.11}$$

$$\tilde{\boldsymbol{S}}^{\boldsymbol{x}}(\gamma) = \sum_{j=1}^{\boldsymbol{d_S}} \boldsymbol{S}_j(\gamma)\boldsymbol{s}_j^{\boldsymbol{x}}, \tag{3.12}$$

where $\tilde{\boldsymbol{E}}(\gamma)$ is the approximate illumination of all blocks in an image, constructed using a $\boldsymbol{d_E}$ dimensional model, and $\tilde{\boldsymbol{S}}^{\boldsymbol{x}}(\gamma)$ is the approximated reflectance for the block $\boldsymbol{x}$, constructed using a $\boldsymbol{d_S}$ dimensional model.

We define an $m \times m$ block of pixels as having m^2K sensor outputs, with K sensor outputs per pixel. Thus, we define the sensor responses of the group of pixels as the

block diagonal matrix

$$\boldsymbol{R}(\gamma) = \begin{bmatrix} R(\lambda) & 0 & \dots & 0 \\ 0 & R(\lambda) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & R(\lambda) \end{bmatrix} \tag{3.13}$$

with m^2 blocks along the diagonal, where

$$R(\lambda) = \begin{bmatrix} R_1(\lambda) & R_2(\lambda) & \dots & R_K(\lambda) \end{bmatrix}.$$

We let $\boldsymbol{R}_k(\gamma)$ refer to the kth column of the matrix $\boldsymbol{R}(\gamma)$.

Following a derivation analogous to that presented in Section 2, we can write the sensor output in a bilinear form with noise as:

$$\boldsymbol{y}_k^{\boldsymbol{x}} = \sum_{i=1}^{\boldsymbol{d_E}} \sum_{j=1}^{\boldsymbol{d_S}} \boldsymbol{e}_i \boldsymbol{G}_{ij,k} \boldsymbol{s}_j^{\boldsymbol{x}} + \boldsymbol{w}_k^{\boldsymbol{x}}, \tag{3.14}$$

where $\boldsymbol{y}_k^{\boldsymbol{x}}$ is the sensor response of the block of pixels $\boldsymbol{x}$, $\boldsymbol{G}_{ij,k}$ is defined as

$$\boldsymbol{G}_{ij,k} = \sum_{\gamma} \boldsymbol{R}_k(\gamma) \boldsymbol{E}_i(\gamma) \boldsymbol{S}_j(\gamma), \tag{3.15}$$

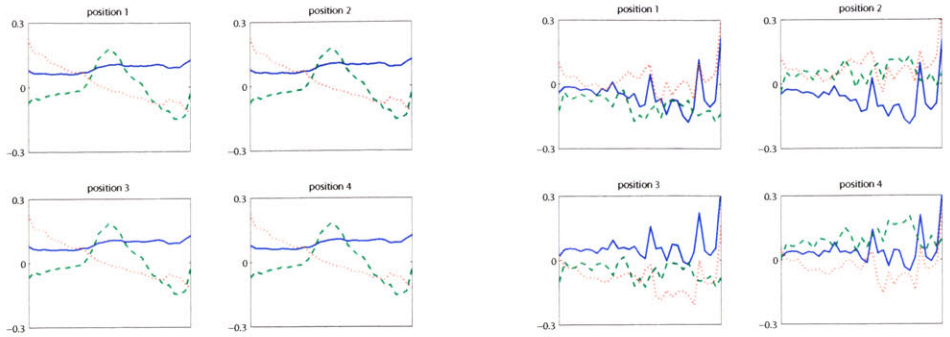
and $\boldsymbol{w}_k^{\boldsymbol{x}}$ is the noise introduced from using a lower-dimensional linear model.

3.4.1 Characteristics of Spatio-Spectral Basis Functions

Figure 3-4 plots the first 12 spatio-spectral reflectance basis functions when considering a 2×2 pixel blocksize. These basis functions were extracted from our hyperspectral data using the calibration gray card in each image of the dataset, as described in Section 3.3. The correlation between neighboring pixels in the 2×2 block is apparent when examining these basis functions. Discarding sign, it can be see that the first three spatio-spectral basis functions are essentially the same as the basis functions for the single pixel shown in Figure 3-1(b), repeated across all four pixels in the 2×2 block. That is, the first three basis functions exhibit virtually no spatial variation. To some extent, Figure 3-1(b) shows a separability between the spatial and spectral characteristics of the reflectance. Ruderman [41] has shown a similar result when examining images in terms of their L,M,and S cone responses.

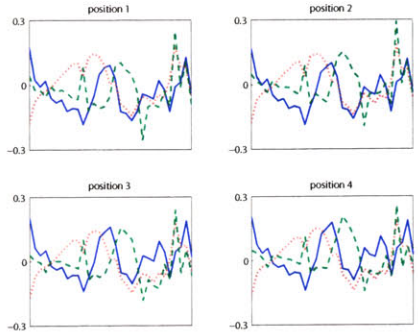
Symbol	Symbol Definition
m	number of pixels along one dimension of a square block being considered within an image
γ	index into spatio-spectral basis functions
K	number of sensor responses available at any pixel
$\boldsymbol{E}(\gamma)$	spatio-spectral radiance of illumination in $m \times m$ block
$\boldsymbol{E}_i(\gamma)$	ith spatio-spectral illumination basis function
$\boldsymbol{e}_i$	scalar weight of ith spatio-spectral illumination basis function
$\boldsymbol{e}$	row vector of all $\boldsymbol{e}_i$
$\boldsymbol{d_E}$	dimensionality of spatio-spectral illumination representation
$\tilde{\boldsymbol{E}}(\gamma)$	approximation to spatio-spectral illuminant from $\boldsymbol{d_E}$ dimensional linear model
$\boldsymbol{S}^{\boldsymbol{x}}(\gamma)$	spatio-surface reflectance for pixel block $\boldsymbol{x}$
$\boldsymbol{S}_j(\gamma)$	jth spatio-spectral surface reflectance basis function
$\boldsymbol{s}_j^{\boldsymbol{x}}$	weight of jth spatio-spectral surface reflectance basis function for pixel block $\boldsymbol{x}$
$\boldsymbol{s}^{\boldsymbol{x}}$	column vector of $\boldsymbol{s}_j^{\boldsymbol{x}}$ for all j
$\boldsymbol{s}$	matrix whose columns are given by $\boldsymbol{s}^{\boldsymbol{x}}$
$\boldsymbol{d_S}$	dimensionality of spatio-spectral surface reflectance representation
$\tilde{\boldsymbol{S}}^{\boldsymbol{x}}(\gamma)$	approximation to spatio-spectral surface reflectance at pixel block $\boldsymbol{x}$ from $\boldsymbol{d_S}$ dimensional linear model
$\boldsymbol{R}_k(\gamma)$	spatio-spectral sensitivity of kth sensor
$\boldsymbol{R}(\gamma)$	matrix of $\boldsymbol{R}_k(\gamma)$ for all k
$\boldsymbol{y}_k^{\boldsymbol{x}}$	scalar response of kth sensor for pixel block $\boldsymbol{x}$
$\boldsymbol{y}^{\boldsymbol{x}}$	vector of $\boldsymbol{y}_k^{\boldsymbol{x}}$ for all k
$\boldsymbol{y}$	matrix of $\boldsymbol{y}^{\boldsymbol{x}}$ vectors for all $\boldsymbol{x}$
$\boldsymbol{G}_{ij,k}$	3-d tensor relating spatio-spectral illumination and reflectance weights to spatio-spectral sensor responses
$\boldsymbol{w}_k^{\boldsymbol{x}}$	model noise contained in $\boldsymbol{y}_k^{\boldsymbol{x}}$

Table 3.2: Symbol Notation for Spatio-Spectral Formulation

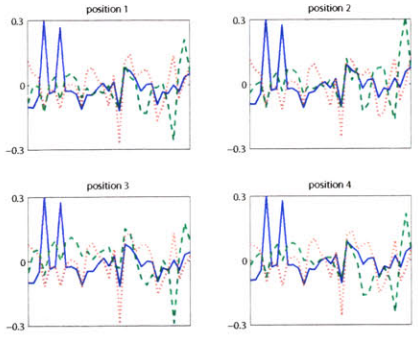


(a) basis functions 1, 2, and 3

(b) basis functions 4, 5, and 6



(c) basis functions 7, 8, and 9



(d) basis functions 10, 11, and 12

Figure 3-4: The first 12 spatio-spectral basis functions for 2×2 pixel blocks. Each basis function is plotted for wavelength values from 400nm to 700nm on the x-axis. The first 3 spatio-spectral basis functions, (a), show only spectral variations and no spatial variation. The next 3 basis functions, (b), correspond to spatial derivatives. The final 6 indicate a more complex relationship of spatial and spectral variations, (c) and (d).

The advantage of incorporating spatial information into the linear models used to describe scenes is that it provides more descriptive power using fewer parameters. The amount of additional information captured through the use of spatio-spectral basis functions can be tested by using reflectance images to find linear reflectance bases for blocks of various sizes (including $m = 1$, which corresponds to the traditional case of examining each pixel independently) and, in each case, calculating the squared reconstruction error as the dimensionality of the reflectance model is varied. To do this, we randomly select approximately half of the data in our hyperspectral dataset of 28 scenes to find the linear model, and calculate the squared reconstruction error using the remaining data.

A comparison of squared reconstruction error when considering block sizes of $m = 1$, 2 and 4 is shown in Figure 3-5. It can be seen that using a block size of $m = 2$ dramatically improves performance. For example, the data shows that the error rate

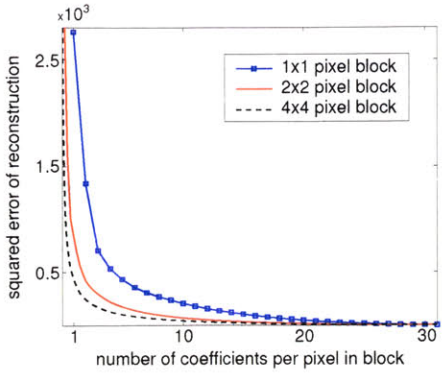


Figure 3-5: The squared reconstruction error using linear models of varying dimensions for block sizes of m =1, 2, 3, and 4. The x-axis is equivalent to the number of basis functions used to describe the $m \times m$ block divided by m^2. Using block sizes of $m = 2$ or larger greatly reduces the number of coefficients needed to describe each pixel.

obtained when describing each pixel with 3 basis functions can be achieved by using only 1.25 basis functions per pixel when describing an image with 2×2 blocks (i.e., 5 basis functions for the block). Increasing the block size beyond $m = 2$ shows even further improvement in the ability to describe the data using fewer parameters, since more of the image correlations can be captured within these larger blocks.

3.4.2 Reducing the Number of Parameters

Suppose we reformulate the color constancy problem using 2×2 blocks instead of single pixels. Each block contains 12 data points. If we describe the reflectance of each block with 12 coefficients (the same number of parameters per pixel as in the standard case), we get an underconstrained problem with $12N$ data points and $12N + 3$ unknowns, where N is now the number of 2×2 blocks.

However, Figure 3-5 shows us that when we describe surfaces using 2×2 blocks, we can achieve the same modeling accuracy as the standard formulation (which describes the reflectance of a single pixel with 3 coefficients) with many fewer than 12 basis functions per 2×2 block. This points to the natural solution of simply modeling the reflectance of a single 2×2 block with 11 coefficients instead of 12, thereby giving us a problem which still models the image much more accurately than our standard model and which has $12N$ data points with only $11N + 3$ unknowns (so it contains more data points than unknown variables if there are more than 12 2×2 blocks in the image).

At first glance, it seems that reformulating the problem to use spatio-spectral basis functions and so that it is no longer underconstrained should allow us to easily solve for both the surface reflectance and illumination. Unfortunately, this is not the case. We attempted to apply both the subspace method and the Bayesian approach to this underdetermined system, and found that both failed due to very high sensitivity to the model error.

The subspace method, introduced in Section 3.2, works by first noting that because the reflectances are of lower dimensionality than the data points, they must, in the standard formulation, lie in an two-dimensional subspace of the three-dimensional RGB space. The analog in our case is that the reflectance of 2×2 blocks lie in an 11-dimensional subspace of the 12-dimensional space of 2×2 RGB image data. In either case, every possible subspace corresponds to a unique illuminant, and if we can determine the subspace we can compute the associated illumination.

All of the existing literature explains that the subspace method fails in the standard case because two-dimensional models for surface reflectance don't model surfaces well enough. If that were the true reason for the subspace method to fail, it should work very well when using the spatio-spectral basis functions because an 11-dimensional model for 2×2 block reflectances is even more accurate than the standard 3-dimensional model for single pixel reflectances. However, this is not the case. We find that the true reason that the subspace method fails in the standard single-pixel formulation is because the energy contained within the second reflectance coefficient is not high enough relative to the model noise to allow us to accurately estimate the subspace. That is, the failure of the subspace method is not due to the total energy contained within the first two reflectance coefficients, but instead it is because that energy is not distributed evenly enough among the two coefficients. The spatio-spectral basis functions exhibit this same problem. The higher-level coefficients in our 11-dimensional approximation contain very little energy, and the small amount of model noise in the system is enough to prevent us from accurately estimating the subspace.

An alternate strategy that was explored in [42] is to utilize the Bayesian approach with our 11-dimensional surface reflectance model. In the Bayesian context, the underconstrained nature of the standard color constancy problem appears in the structure of the likelihood function. If the system is underdetermined by m parameters, the likelihood will have zero curvature across m dimensions, indicating that the data is ambiguous as to which point in that m-dimensional space is the most likely solution. Describing how when describing surfaces in 2×2 blocks with only 11

coefficients results in a likelihood that only has one dimension of zero-curvature (associated with the overall magnitude of the illumination, which cannot be recovered). Applying the Bayesian algorithm in this instance should allow us to easily recover the true reflectance and illumination values. It is shown in [42], however, that while this technique works very well if there is no model noise it fails in the case of real images. The modeling noise in real images is shown to be so small that it cannot even be perceived by the human eye in the final RGB image. Yet this small noise term is enough to destroy the algorithm.

The failure of the Bayesian method to provide good results by simply reducing the dimensionality of the model illustrates the difference between the true characteristics of a problem and how we model it. The color constancy problem is inherently underconstrained. We can try to describe our unknowns using fewer parameters in order to make it mathematically overconstrained, but in doing so we increase the model noise as well as our sensitivity to it. For this reason, for the remainder of this chapter, we describe 2×2 surface reflectance blocks with 12 coefficients.

3.5 Bayesian Color Constancy Using Spatial Information

3.5.1 Choice of Probability Models

Using previous work as a guide [6], truncated Gaussian distributions are used to describe the prior probabilities of reflectance and illumination coefficients. That is, the distribution of coefficients is given by a Gaussian distribution so long as they correspond to physically realizable spectra:

$$P(\boldsymbol{e}) = \begin{cases} \mathtt{G}(u_{\boldsymbol{e}}, \Lambda_{\boldsymbol{e}}) & \text{if } 0 \leq \boldsymbol{E}(\gamma) \text{ for all } \gamma \\ 0 & \text{otherwise} \end{cases}, \tag{3.16}$$

$$P(\boldsymbol{s}^{\boldsymbol{x}}) = \begin{cases} \mathtt{G}(u_{\boldsymbol{s}}, \Lambda_{\boldsymbol{s}}) & \text{if } 0 \leq \boldsymbol{S}^{\boldsymbol{x}}(\gamma) \leq 1 \text{ for all } \gamma \\ 0 & \text{otherwise} \end{cases}, \tag{3.17}$$

where $(u_{\boldsymbol{e}}, \Lambda_{\boldsymbol{e}})$ and $(u_{\boldsymbol{s}}, \Lambda_{\boldsymbol{s}})$ are the means and covariances of the illumination and spatio-spectral reflectance coefficients, respectively. In the current formulation, all blocks are modeled as independent of one another – that is, we only consider spatial statistics of pixels within the same block. This allows us to write the total prior

probability as

$$P(\boldsymbol{e}, \boldsymbol{s}) = P(\boldsymbol{e})P(\boldsymbol{s}) \tag{3.18}$$

$$= P(\boldsymbol{e}) \prod_{\boldsymbol{x}} P(\boldsymbol{s}^{\boldsymbol{x}}). \tag{3.19}$$

Recall that the image formation process is described by:

$$\boldsymbol{y}_k^{\boldsymbol{x}} = \sum_{i=1}^{\boldsymbol{d_E}} \sum_{j=1}^{\boldsymbol{d_S}} \boldsymbol{e}_i \boldsymbol{G}_{ij,k} \boldsymbol{s}_j^{\boldsymbol{x}} + \boldsymbol{w}_k^{\boldsymbol{x}}. \tag{3.20}$$

This can also be written as:

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{e})\boldsymbol{s} + \boldsymbol{w}, \tag{3.21}$$

where $\boldsymbol{y}$ is a matrix of image data, $\boldsymbol{s}$ is a matrix of spatio-spectral reflectance coefficients, $\boldsymbol{A}$ is a matrix given by

$$\boldsymbol{A}(\boldsymbol{e})_{k,j} = \sum_{i=1}^{\boldsymbol{d_E}} \boldsymbol{e}_i \boldsymbol{G}_{ij,k}, \tag{3.22}$$

$\boldsymbol{w}$ is a matrix of the "model noise", and where each of the image patches is described within a given column of the $\boldsymbol{y}$, $\boldsymbol{s}$, $\boldsymbol{w}$ matrices. The model noise is described by a Gaussian with mean $u_{\boldsymbol{w}}$ and covariance $\Lambda_{\boldsymbol{w}}$, so that the distribution of the likelihood function is given by

$$P(\boldsymbol{y}|\boldsymbol{e}, \boldsymbol{s}) = \prod_{\boldsymbol{x}} \mathtt{G}(\boldsymbol{A}(\boldsymbol{e})\boldsymbol{s}^{\boldsymbol{x}} + u_{\boldsymbol{w}}, \Lambda_{\boldsymbol{w}}). \tag{3.23}$$

The means and covariances of the priors and observation are all obtained from the hyperspectral dataset described in Section 3.3.

3.5.2 Implementation Issues

We use the maximum a posteriori approach to estimate the illumination and surface reflectance. No closed form solution exists to determine the optimal set of coefficients, so we must carry out a numerical optimization to solve the problem. Ideally, we would like to search the entire space of illumination and spatio-spectral surface reflectance coefficients in this optimization procedure. However, such a search process is infeasible given the very high dimensionality of this space for any real image (since the number of reflectance coefficients scales with the size of the image).

In order to approximate a solution to the problem, we utilize an algorithm similar to the one used in [6]:

1. Pick an initial guess for the illumination coefficients
2. Estimate the surfaces using the illumination estimate from the previous step and the known image data
3. Evaluate the posterior probability of the image given the illumination and reflectance coefficients
4. Numerically search the illumination space (within some small fixed distance from the current illumination estimate) for an illumination that increases the posterior probability (this involves performing the above two steps repeated times while searching the illumination space)
5. Set the estimate of illumination coefficients to the set of values, among those evaluated in the previous step, that corresponds to the largest posterior probability
6. Repeat the previous 4 steps until convergence

In order to be able to implement the above procedure in a reasonable time frame, we randomly pick a present number of $2x2$ blocks from the image rather than running on the entire image. We run the algorithm with multiple initializations in order to combat the effect of local minima. We utilize one of two initial guesses for the illuminant in these runs -- the illumination obtained using the gray world approach as well as the mean illumination according to our prior. Every time we run the algorithm we also pick a different set of image blocks as an input. The final output of the algorithm is the result, among all of the runs, that yields the highest posterior probability.

Intuitively, the above procedure should produce an estimate of the illumination and reflectance coefficients that locally maximizes the posterior probability, if we are able to find the best estimate of the surfaces in step 2 of the procedure. We would like to be able to estimate $\boldsymbol{s}$ given $\boldsymbol{y}$ and $\boldsymbol{A}(\boldsymbol{e})$.

In [6], a maximum-likelihood (ML) estimation is performed at this stage of the algorithm, so that the surface at a particular image patch location, $\boldsymbol{x}$, is estimated as

$$\boldsymbol{s}_{ML}^{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{e})^{-1}(\boldsymbol{y}^{\boldsymbol{x}} - u_{\boldsymbol{w}}). \tag{3.24}$$

Although this method works reasonably in the standard Bayesian color constancy algorithm, it does not work when using the spatio-spectral basis functions. This is

because the ML estimate constrains us to finding solutions that exactly describe the image data under the assumption that the noise is equal to its mean value. Of course, the noise is not equal to its mean value everywhere so our estimates have to describe all aspects of the noise that deviate from the mean. Since the first twelve spatio-spectral basis functions exhibit very little spatial variation, as discussed in the previous section, they are unable to explain spatially varying model noise without producing surface reflectances that are unrealizable. In mathematical terms, the near-separability of the spatial and spectral characteristics of the spatio-spectral basis functions results in the matrix $\boldsymbol{A}(\boldsymbol{e})$ being very badly conditioned, resulting in an unstable inverse.

In order to provide a more stable and accurate estimate of the surfaces given an illumination estimate, we find the maximum a posteriori estimate at step 2 of the procedure outlined above. The MAP estimate incorporates our priors over the spatio-spectral surface coefficients given in Equation 3.17, which prevents the algorithm from producing surface estimates that are physically impossible. As outlined in [5], we can find the MAP estimate in closed form as:

$$\boldsymbol{s}^{\boldsymbol{x}}_{MAP} = Q\boldsymbol{y}^{\boldsymbol{x}} + s_0, \tag{3.25}$$

where

$$Q = \Lambda_{\boldsymbol{S}}(\boldsymbol{A}(\boldsymbol{e}))^T(\boldsymbol{A}(\boldsymbol{e})\Lambda_{\boldsymbol{S}}(\boldsymbol{A}(\boldsymbol{e}))^T + \Lambda_{\boldsymbol{s}})^{-}1, \text{ and} \tag{3.26}$$

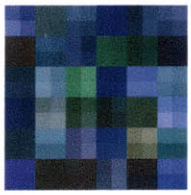
$$s_0 = u_{\boldsymbol{S}} - Q((\boldsymbol{A}(\boldsymbol{e})u_{\boldsymbol{S}} - u_{\boldsymbol{w}}). \tag{3.27}$$

3.6 Results

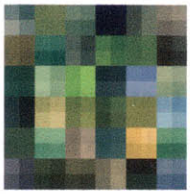
We compared the performance of standard (treating each pixel independently) Bayesian color constancy to our approach that utilizes spatio-spectral basis functions. The metric utilized to measure performance was the average of the root mean-square errors of surface reflectance spectrum estimates at each pixel in the image. Testing was done on both synthetic and real world images. The synthetic images were generated by drawing surfaces and illuminants from the assumed Gaussian prior distributions (surfaces were drawn in 2×2 blocks) and adding noise as described by 3.23. Real-world images were obtained from the hyperspectral data set described in Section 3.3. In both cases ground-truth reflectance data is available.

The synthetic dataset used for testing consisted of twenty-eight images of 72×72 pixels in size (so there are 36 2×2 blocks in each image). An example of one of these

synthetic images, as well as the corresponding surface reflectances (after being passed through the sensors) is given in Figure 3-6. The assumptions implicit in our model can be clearly seen in this image. The pixels within each of the 36 2×2 blocks are correlated with one another, but each of these blocks is independent of the others.



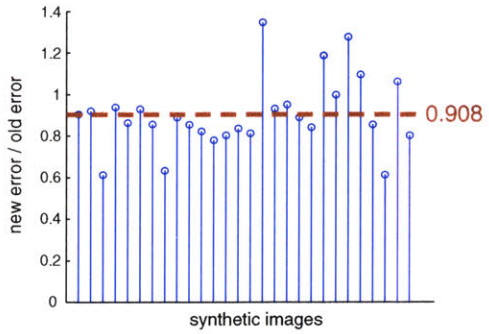
(a) Synthetically Generated Image



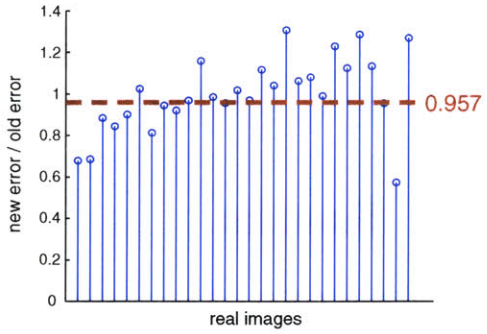
(b) Surfaces of Synthetic Image

Figure 3-6: Synthetically generated data.

Figure 3-7(a) shows the ratio of the error when using the new approach to the error when using the standard approach for images in the synthetic dataset, and 3-7(b) shows this same information for the real-image dataset. When testing on synthetic images, the error decreases by almost 10% on average by using the spatio-spectral method. The results on real images also show improvement, albeit less so, with the spatio-spectral providing a 4.3% reduction in error over the standard approach on our set of 28 real images.



(a) Results on Synthetic Data



(b) Results on Real Data

Figure 3-7: Test results comparing the spatio-spectral Bayesian algorithm with the standard Bayesian algorithm on both synthetic 3-7(a) and real-world 3-7(b) images. A value less than unity indicates that the new approach performed better than the standard approach on a given test image, and a value greater than unity indicates that it performed worse. The dashed lines show the performance on all images in the given test set, calculated as the sum of all errors obtained using the spatio-spectral method divided by the sum of all errors when using the standard method.

As the figure shows, even when using synthetic examples there are instances in which the standard approach produces a more accurate estimate than the new

method. This is likely because the spatio-spectral approach uses a more restrictive probability model that, while it provides a more accurate description of the entire set of test images, may be overly restrictive for particular examples. In addition, the iterative algorithm outlined in Section 3.5.2 is able to find only local minima, and it is possible that when running the algorithms on a given image, the standard algorithm was able to find a better local minima than our algorithm.

The basic hypothesis behind the spatio-spectral Bayesian approach was that incorporating spatial information into our prior models would allow for better color constancy. The reduction in error for both synthetic and real-world data validates this hypothesis. The improvement from using spatial statistics are not as substantial as one might expect, however. Chapter 5 discusses a number of reasons why this might be the case.

Chapter 4

Blind Image Deconvolution [1]

4.1 Related Work

Image deconvolution is a classic image processing problem. The non-blind version of the problem, in which the convolution kernel is known, is widely studied [22] and relatively well-solved (though newer techniques continue to be developed [12]). Blind image deconvolution is also well-studied [23, 47] but is far from satisfactorily solved.

Many of the previous approaches to blind image deconvolution rely on making simple assumptions about the convolution kernel or the true image. One common method is to use a constant-velocity model for the camera motion. Other parametric forms include Gaussian blurs, or models that consist of relatively low-frequency Fourier components. In reality, though, these assumptions do not hold for the camera shake problem, in which the camera motion is often jerky (high-frequency) and cannot be adequately described by a simple parametric form. Low-frequency assumptions are also often made for the input image, such as by modeling the image as a two-dimensional autoregressive process (in conjunction with modeling the blur as a two-dimensional moving average process). Such image models result in image reconstructions that are stripped of their high frequency content, so that edges are not as sharp as they should be and textures are often blurred. Caron et al. uses a power-law distribution, a type of natural image statistic that describes global frequency content but not local structures, to describe image frequency content [9]. This method requires the existence of a reference image that has the same frequency distribution as

[1] Much of the work presented in this chapter is to appear in publication as "Removing Camera Shake from a Single Photograph" by Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman, in ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006), [11]

the image being estimated. Power-law distributions have also been combined with constraints in the wavelet domain [21, 34]; however these methods still do not work for the type of complex blur kernels caused by camera shake.

Some iterative methods have been developed which attempt to solve the blind image deconvolution problem by alternately holding the blur kernel and sharp image fixed at each iteration and, at each iteration, tackling the non-blind problem of inferring one given the other [4, 22]. These methods are very limited, however, and are generally only able to deal with small and simple blur kernels. In Section 4.6 we show that these algorithms are not very useful when dealing with real-world blurry images.

Astronomy and medical imaging have been particularly active application areas for blind image deconvolution, and a number of approaches have been developed for these tasks [18, 37, 44, 48]. These methods take advantage of the unique constraints and statistics that astronomical images exhibit, such as finite support for a sparse set of object being imaged against a uniform background. These additional constraints greatly simplify the problem, and allow for reasonably high quality algorithms. For example, in astronomical images, a blurry image of a single star gives a good estimate of the point spread function. Unfortunately, the constraints and statistics utilized by these algorithms are violated by natural images, and thus are not useful for our problem.

Some of the more successful blind image deconvolution techniques operate by assuming the existence of additional data or modified hardware that helps to solve the problem. For example, [2] and [36] both use a sequence of motion blurred images rather than just a single one. Hardware approaches include using optically stabilized lenses [20], specially designed CMOS sensors [28], and hybrid imaging systems [3]. In this thesis, however, we wish to examine the problem where such additional data and hardware are not available (as is currently the case for the large majority of existing photos and consumer level cameras).

4.2 Problem Formulation

4.2.1 Image Formation Process

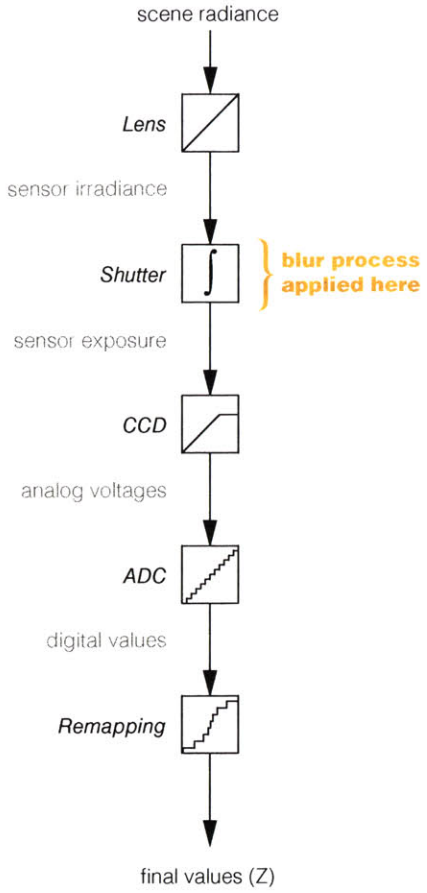
In order to formulate the blind image deconvolution problem, we must first define the process by which digital images are formed. In the previous chapter, we described the image formation process as the process by which an illumination is partially reflected

from a surface and imaged by a set of sensors. There we described the sensors as a set of linear filters, which was indeed the case for the data being used (because we generated the RGB images from hyperspectral data). For photographs produced by real-world digital cameras, however, the pixels values are not linearly related to the spectral intensity of the light coming in through the lens.

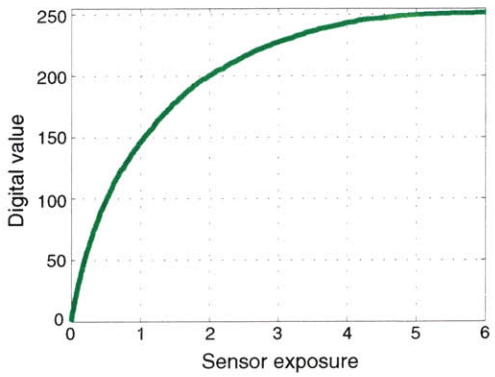
Figure 4-1(a) shows an outline of the process by which the light coming in to the camera lens (known as the scene radiance) is transformed into a digital image [10]. Once the input light energy is passed through the lens, it falls upon the camera's sensors (known as charge couple devices, or CCDs). The shutter acts as a switch controlling how long the CCD sensors are allowed to absorb light energy. Thus, the energy absorbed by the CCDs is integrated over the amount of time the shutter is open. The first nonlinearity within the camera is found within the CCDs themselves, which have a saturation point to the amount of energy they can absorb (but are otherwise linear). The charge stored in the CCD sensors is then converted to a digital value by an analog-to-digital converter (ADC), which is also non-linear. Finally, the output of the ADC is passed through a highly nonlinear remapping function to produce the final image. This last remapping function performs a number of tasks, including: converting the 12-bit output of the ADC to an 8-bit number (the common size used for most image formats), gamma correcting the image to make it look more realistic (given that neither human vision nor display monitors have linear responses), and compressing the image in the appropriate manner (depending upon which image format it is stored in).

Camera shake affects the image capture process before the camera's nonlinearities are applied, at the stage where the CCD sensors are exposed to light energy. If there is no camera shake, each CCD sensor absorbs a constant amount of energy from a particular location in the scene for the entire duration that the shutter is open (excluding changes in the scene itself during that time period). If the camera undergoes motion during that time period, however, a given sensor is exposed to different parts of the scene for varying amounts of time, resulting in a blurry image.

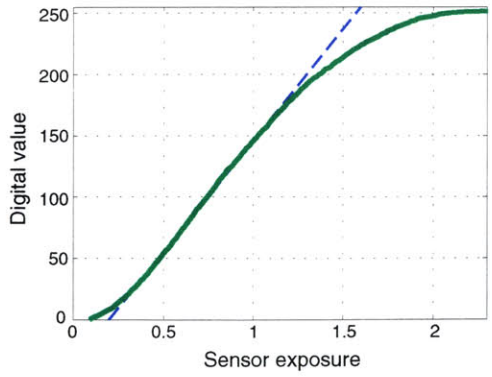
In order to properly remove the effect of the camera motion, we must know both the sensor exposure and the motion path. Unfortunately, we cannot fully recover the sensor exposure, both because we do not have adequate models for all of the nonlinearities and because some nonlinearities are not fully invertible. We can, however, get a good approximate to the sensor exposure by undoing the effect of the gamma



(a) Image formation process.



(b) Camera response for Canon SD550



(c) Camera response for Canon SD550 after inverse gamma correction

Figure 4-1: The camera introduces many nonlinearities into the image formation process after the image has already been affected by camera shake, as shown in 4-1(a). Figure 4-1(b) shows the net effect of all of the nonlinearities we must contend with for a typical camera. Undoing the camera's gamma correction almost linearizes its response with the exception of saturation effects, as shown in 4-1(c).

correction. Gamma correction is a simple power-function nonlinearity,

$$\text{Pixel value} = (\text{CCD sensor value})^{1/\gamma}. \tag{4.1}$$

For the large majority of digital cameras, $\gamma = 2.2$ is very close to the true γ value, so this is the parameter value we will use. Figure 4-1(b) shows the response curve for a Canon SD550 digital camera, calculated using the method outline in [10]. The response curve after removing the effect of gamma correction is given in Figure 4-1(c). These curves are representative of consumer-level digital cameras. As the figures show, removing the gamma correction makes the response nearly linear for a large range of digital values. The portion of the response curve that remains highly non-linear can be modeled reasonably with a saturation effect.

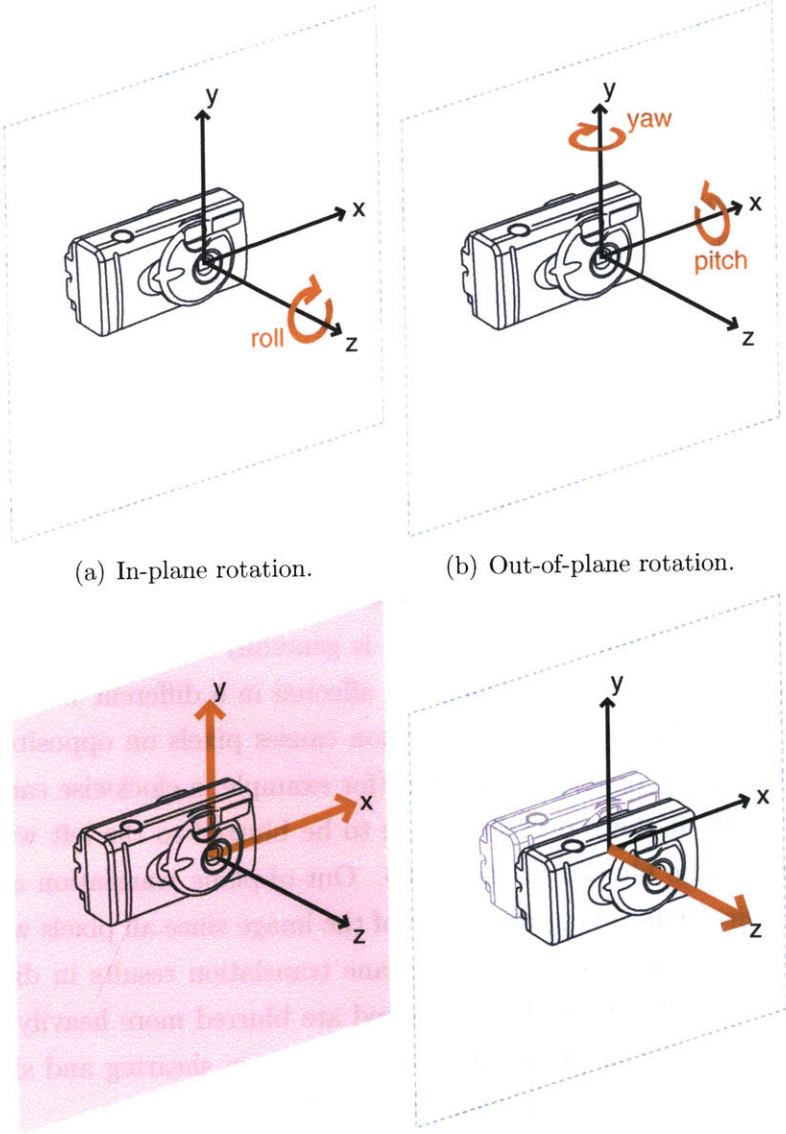
4.2.2 Blur Process

The appropriate model for camera shake depends upon the type of motion we wish to describe. The blur kernel may be due to rotation and/or translation of the camera. Figure 4-2 illustrates the types of movement we must be concerned with.

The exact effect of any camera motion is generally non-linear across the image, so that pixels in one part of the image are affected in a different way than pixels in another part of the image. In-plane rotation causes pixels on opposite sides of the image to be blurred in opposite directions (for example, a clockwise camera rotation would make pixels at the top of the image to be blurred to the left while pixels on the bottom would be blurred to the right). Out-of-plane translation also results in opposing blur directions on opposite sides of the image since all pixels will be blurred either radially inwards or outwards. In-plane translation results in distortions due to parallax, so that objects in the foreground are blurred more heavily than objects in the background. Out-of-plane rotation can result in shearing and skewing of the blurred image, depending on lens distortion.

Under certain conditions, some of the nonlinear image transformations caused by camera motion can be reasonably modeled by a stationary blur kernel (i.e., a blur kernel that is the same for all pixels in the image). In particular, the effect of out-of-plane rotation can be nearly linear. In addition, if the distance between the camera plane and the objects in the scene are close to the same for all objects in the scene, the parallax effect becomes negligible.

Mathematically, the effect of a stationary blur kernel is given by a linear convo-



(a) In-plane rotation.

(b) Out-of-plane rotation.

(c) In-plane translation

(d) Out-of-plane translation

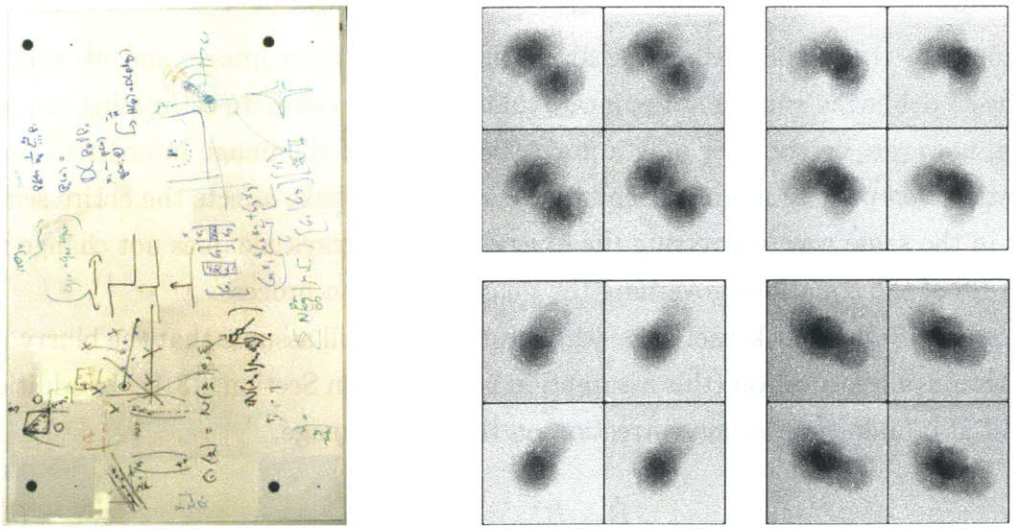
Figure 4-2: Types of camera movement.

lution:

$$\text{Blurry image} = \text{Sharp image} \otimes \text{Blur kernel}. \tag{4.2}$$

Note that if the camera experiences no motion, the blur kernel would simply be a centered delta function, so that the convolution operation would have no effect.

We conducted an experiment to determine if linear convolution is an adequate model for realistic camera shake. Eight people were asked to photograph the whiteboard shown in Figure 4-3(a), which had equally sized black dots placed on each of the four corners. The camera was held approximately 5 meters from the whiteboard and its shutter speed was set to 1 second (the long exposure time resulted in all of the images being blurred due to camera shake). The camera lens had a focal length of 50mm mounted on a 0.6x DSLR sensor. If the effect of the camera shake could be described by a linear convolution, we would expect that to observe that all four black dots would be blurred in the same way. Figure 4-3(b) shows images of all the black dots extracted from a random sampling of four images taken by different people. As shown in the figure, in each case the blur profiles for all four dots are very close to equal, providing evidence that the blur equation given by 4.2 is a good model for the true blur process.



(a) Whiteboard used in experiment

(b) Blur patterns

Figure 4-3: An experiment in which participants were asked to photograph a whiteboard (at left) with a camera set to a one-second exposure time shows that convolution with a stationary blur kernel is a good approximation to blur due to camera shake. The four sets of blur patterns in 4-3(b) are extracted from photographs taken by four distinct people. They show that the dots from each corner of the whiteboard exhibit nearly identical blur patterns.

4.2.3 Image Model

Given the discussion above, we now formulate the blind image deconvolution problem as follows: given an input blurry image, B, we wish to estimate the latent image L and blur kernel K where the blurry image was formed according to the process

$$B = (\text{sat}(K \otimes L + N))^{1/\gamma}, \tag{4.3}$$

where N is some additive sensor noise, B, L, and K must fall within the range $[0, 1]$, $\gamma = 2.2$, and the saturation function $\text{sat}(\cdot)$ is given by

$$\text{sat}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } 0 < x < 1 \\ 1 & \text{if } 1 \leq x \end{cases}. \tag{4.4}$$

We divide the blind image deconvolution task into two distinct steps. We first estimate the blur kernel and then recover the latent image given that estimate.

4.3 Blur Kernel Estimation

The input to the overall deblurring algorithm is a full color image, and we wish to produce a deblurred version of the image that is also full color. In estimating the blur kernel, however, we consider only a grayscale version of the image in order to make the estimation process more efficient. Because camera shake affects the entire sensor array in the same way, converting the blurry image to grayscale does not change the structure of the equations governing the image formation process.

To simplify the blur kernel estimation problem, we will assume that the blurry image contains no saturation (this assumption is revisited in Section 4.3.4). In addition, we will also undo the gamma correction on the blurry image,

$$\tilde{B} = B^{\gamma}_{\text{no saturation}}, \tag{4.5}$$

resulting in a linear image formation process:

$$\tilde{B} = K \otimes L + N. \tag{4.6}$$

We wish to utilize Bayesian methods in order to estimate the unknown blur kernel. These methods require us to specify a prior on the statistics of the latent image and

blur kernel. The natural image statistics that provide an informative prior about the latent image relate to the gradients of the latent image rather than its pixel values, as described in Section 1.3. We must therefore reformulate the problem in the image gradient domain. Fortunately, this is easy to do because convolution is a linear operation, which allows us to replace 4.6 with

$$\nabla \tilde{B} = K \otimes \nabla L + \nabla N, \tag{4.7}$$

where ∇ represents the gradient operation.

4.3.1 Choice of Probability Models

For tractability, we assume that the gradients across the image are all independent of one another. Let $l_{ij} = \nabla L(i,j)$ index a single pixel of the latent gradient image. We model the heavy-tailed natural image prior by a mixture of C zero-mean Gaussian distributions with variance v_c and weight π_c for the c-th Gaussian distribution, yielding a prior distribution of

$$P(\nabla L) = \prod_{i,j} P(l_{ij}) \tag{4.8}$$

$$= \prod_{i,j} \sum_{c=1}^{C} \pi_c \mathtt{G}(l_{ij}|0, v_c). \tag{4.9}$$

We based our image prior off of single typical natural image (the street scene shown earlier in Figure 1-3(a)). Figure4-4(a) shows the empirical distribution of gradients obtained from that image along with the mixture of Gaussians latent image prior that we utilize.

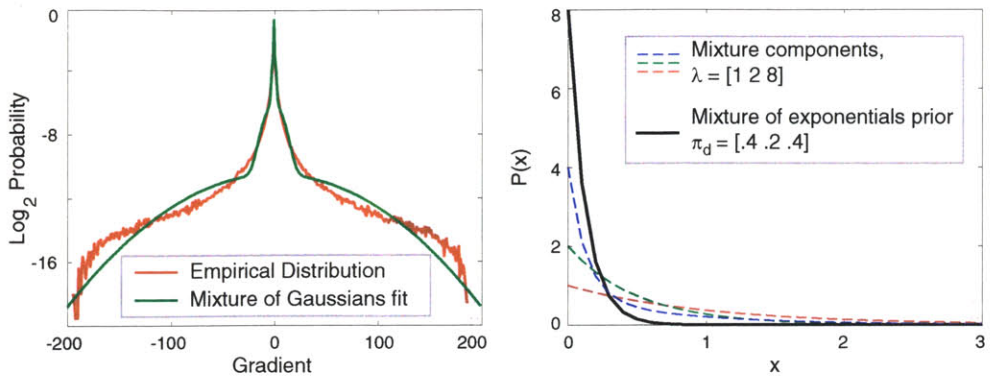
We also assume independence among all pixels in the blur kernel. Let $k_{mn} = K(m,n)$ index a single pixel of the blur kernel. We model each pixel in the blur kernel by a mixture of D exponential distributions with scale factor λ_d and weight κ_d for the d-th exponential distribution, resulting in

$$P(K) = \prod_{m,n} P(k_{mn}) \tag{4.10}$$

$$= \prod_{m,n} \sum_{d=1}^{D} \kappa_d \mathtt{E}(k_{mn}|\lambda_d). \tag{4.11}$$

This prior on the blur enforces positivity and encourages sparsity. Positivity is a

requirement because a negative value would correspond to negative light intensity. Sparsity is also very important because the blur kernel should be zero at all locations other than the path of the camera. Figure 4-4(b) shows the blur prior.



(a) Latent image prior (b) Blur prior

Figure 4-4: Priors used to model latent image gradients and blur kernel values.

We model the sensor noise N as a zero-mean Gaussian and assume it applies to each pixel independently. Since the gradient is a linear operation, we therefore also model the gradient of the sensor noise ∇N as a zero-mean Gaussian with some variance σ^2. Let $b_{ij} = \nabla\tilde{B}(i,j)$ index a single pixel in the saturation-free inverse-gamma corrected gradient blurry image. We can then write the likelihood term as

$$P(\nabla\tilde{B}|K,\nabla L) = \prod_{i,j} P(b_{ij}|K,\nabla L) \tag{4.12}$$

$$= \prod_{i,j} \mathtt{G}(b_{ij}|(K\otimes\nabla L)_{ij},\sigma^2). \tag{4.13}$$

The noise level for a given image depends on a number of factors, including what type of camera is used, the ISO settings, what time of day and where the image was taken, etc. Because all of these things can vary from one image to another, we treat the sensor noise variance σ^2 as an unknown random variable. Using [33] as a guide, we model the inverse variance by a gamma distribution with hyper-parameters α,β,

$$P(\sigma^{-2}|\alpha,\beta) = \mathtt{Gamma}(\sigma^{-2}|\alpha,\beta). \tag{4.14}$$

The posterior probability is now given by

$$P(K, \nabla L|\nabla \tilde{B}) \propto P(\nabla \tilde{B}|K, \nabla L)P(\nabla L)P(K)P(\sigma^{-2}) \quad (4.15)$$

$$= \prod_{i,j} \mathtt{N}(b_{ij}|(K \otimes \nabla L)_{ij}, \sigma^2) \prod_{i,j} \sum_{c=1}^{C} \pi_c \mathtt{N}(l_{ij}|0, v_c) \quad (4.16)$$

$$\prod_{m,n} \sum_{d=1}^{D} \pi_d \mathtt{E}(k_{mn}|\lambda_d)\ \mathtt{Gamma}(\sigma^{-2}|\alpha, \beta).$$

4.3.2 Variational Bayes Estimation

We utilize the variational Bayesian approach described in Section 2.2. The analytical forms of the probability distributions we have chosen for our model allow us to derive iterative update equations.

Update Equations

We provide a partial derivation of the variational Bayes update equations below. This is a simplified version of the derivation presented in [33], and we refer the reader to that text for further details.

If we write out the convolution of Equation 4.7 in terms of summations, the image formation process is given by

$$b_{ij} = \sum_m \sum_n k_{mn} l_{i-m,j-n} + n_{ij}, \quad (4.17)$$

where $n_{ij} = \nabla N(i, j)$ indexes a single pixel of the gradient noise and the remaining terms are as defined in the previous section.

The full posterior probability, given by 4.15, is a function of three unknown variables: the blur kernel K, the latent image ∇L, and the inverse noise variance σ^{-2}. The variational approximation to the posterior assumes independence among these three variables, allowing us to write the variational approximation as the product of marginal distributions

$$Q(K, \nabla L, \sigma^{-2}) = Q(K)Q(\nabla L)Q(\sigma^{-2}). \quad (4.18)$$

Following Section 2.2.2, the KL divergence between the posterior and the variational

approximation is given by

$$D_{KL}(Q||P) = \left\langle \log\left(\frac{Q(K)}{P(K)}\right)\right\rangle_Q + \left\langle \log\left(\frac{Q(\nabla L)}{P(\nabla L)}\right)\right\rangle_Q + \left\langle \log\left(\frac{Q(\sigma^{-2})}{P(\nabla\tilde{B}|\nabla L, K)}\right)\right\rangle_Q . \tag{4.19}$$

A final simplification that must be made is to apply Jensen's inequality to Equations 4.9 and 4.11 to get:

$$-\log P(l_{ij}) \leq \sum_c w_{ijc} \log\left(\pi_c \mathtt{N}(l_{ij}|0, v_c)\right), \tag{4.20}$$

$$-\log P(k_{mn}) \leq \sum_d u_{mnd} \log\left(\pi_d \mathtt{E}(k_{mn}|\lambda_d)\right), \tag{4.21}$$

where we have introduced two new sets of parameters that must satisfy the constraints

$$\sum_c w_{ijc} = 1 \qquad \forall ij, \tag{4.22}$$

$$\sum_d u_{mnd} = 1 \qquad \forall mn. \tag{4.23}$$

Because of the form of the priors we have chosen, the optimal marginal distributions are given by

$$Q(k_{mn}) = \mathtt{G}^{(\mathtt{R})}(k_{mn}|k'_{mn}, k''_{mn}), \tag{4.24}$$

$$Q(l_{ij}) = \mathtt{G}(l_{ij}|l'_{ij}, l''_{ij}), \tag{4.25}$$

$$Q(\sigma^{-2}) = \mathtt{Gamma}(\sigma^{-2}|\tilde{\alpha}, \tilde{\beta}). \tag{4.26}$$

The update equations are given below. For reference, Table 4.1 outlines of all the symbol notation used in these equations and in the preceding derivations.

$$\hat{u}_{mnd} = \pi_d \lambda_d \exp(-\lambda_d E[k_{mn}]) \tag{4.27}$$

$$u_{mnd} = \hat{u}_{mnd} / \sum_d \hat{u}_{mnd} \tag{4.28}$$

$$\hat{w}_{ijc} = \frac{\pi_c}{\sqrt{v_c}} \exp\left(-\frac{E[l_{ij}^2]}{2v_c}\right) \tag{4.29}$$

$$w_{ijc} = \hat{w}_{ijc} / \sum_c \hat{w}_{ijc} \tag{4.30}$$

$$k'_{mn} = \sum_{ij} E[\sigma^{-2}] \left\langle l^2_{i-m,j-n} \right\rangle_{Q(l)} \tag{4.31}$$

$$k''_{mn} = \sum_{ij} E[\sigma^{-2}] \left\langle \left(b_{ij} - \sum_{m',n' \neq i-m,j-n} k_{m'n'} l_{i-m',j-n'} \right) l_{i-m,j-n} \right\rangle_{Q(k,l)} - \sum_d \frac{u_{mnd}}{\lambda_d} \tag{4.32}$$

$$l'_{ij} = \sum_c \frac{w_{ijc}}{v_c} + E[\sigma^{-2}] \sum_{mn} \left\langle k^2_{m,n} \right\rangle_{Q(k)} \tag{4.33}$$

$$l''_{ij} = \sum_{mn} E[\sigma^{-2}] \left\langle \left(b_{i+m,j+n} - \sum_{m'n' \neq m,n} k_{m'n'} l_{i+m-m',j+n-n'} \right) k_{m,n} \right\rangle_{Q(k,l)} \tag{4.34}$$

$$\tilde{\alpha} = \alpha + \frac{1}{2} \sum_{ij} (\nabla \tilde{B} - (K \otimes \nabla L))^2_{ij} \tag{4.35}$$

$$\tilde{\beta} = \beta + IJ/2 \tag{4.36}$$

$$E[l_{ij}] = l''_{ij}/l'_{ij} \tag{4.37}$$

$$E[l^2_{ij}] = (l''_{ij}/l'_{ij})^2 + 1/l'_{ij} \tag{4.38}$$

$$E[k_{mn}] = k'_{mn} + \sqrt{\frac{2}{\pi k''_{mn}}} \frac{1}{\operatorname{erfcx}\left(k'_{mn}\sqrt{k''_{mn}/2}\right)} \tag{4.39}$$

$$E[k^2_{mn}] = (k'_{mn})^2 + (k''_{mn})^{-1} + \sqrt{\frac{1}{\pi k''_{mn}}} \frac{k'_{mn}}{\operatorname{erfcx}\left(k'_{mn}\sqrt{k''_{mn}/2}\right)} \tag{4.40}$$

where the erfcx function is the complementary error function, defined as

$$\operatorname{erfcx}(x) = \exp(x^2) \frac{2}{\pi} \int_x^{\inf} \exp(-t^2) dt. \tag{4.41}$$

Variational Inference Algorithm

Suppose we are given an initialization for K and ∇L. We set the means of the variational distributions $Q(K)$ and $Q(\nabla L)$ equal to these values, and set the variance of the distributions to large values, due to our uncertainty in the initializations.

4.3.3 Coarse-to-Fine Estimation

Despite using the variational approach, the `Inference` algorithm described above is still susceptible to many local minima. The issue of finding a good initialization for

Symbol	Symbol Definition
B	blurry image
L	latent sharp image
K	blur kernel
N	sensor noise
$\tilde{B}$	gamma-corrected blurry image assuming no saturation
$\nabla\tilde{B}$	gradients of $\tilde{B}$
∇L	gradients of L
l_{ij}	value of ∇L at pixel (i,j)
b_{ij}	value of $\nabla\tilde{B}$ at pixel (i,j)
k_{mn}	value of K at location (m,n)
C	number of mixture components in mixture of Gaussians prior $P(\nabla L)$
π_c	weight of mixture component c in $P(\nabla L)$
v_c	variance of mixture component c in $P(\nabla L)$
D	number of mixture components in mixture of exponentials prior $P(K)$
κ_d	weight of mixture component d in $P(K)$
λ_d	scale factor of mixture component d in $P(K)$
σ^2	variance of sensor noise
α, β	shape and scale parameters for gamma prior over inverse noise variance σ^{-2}
k'_{mn}, k''_{mn}	parameters for rectified Gaussian component of variational distribution that approximates the posterior over blur kernel values
l'_{ij}, l''_{ij}	mean and variance of Gaussian component of variational distribution that approximates the posterior over latent gradient image values
$\tilde{\alpha}, \tilde{\beta}$	shape and scale parameters for gamma component of variational distribution that approximates the posterior over the inverse noise variance
w_{ijc}	contribution of mixture component c from the mixture model prior $P(l_{ij})$ to the variational approximation to the posterior over l_{ij}
u_{mnd}	contribution of mixture component d from the mixture model prior $P(k_{md})$ to the variational approximation to the posterior over k_{md}

Table 4.1: Symbol Notation for Deblurring Probability Models

Algorithm 1 `Inference` (simplified from Miskin and MacKay [33])

Require: Observed blurry gradients $\nabla\tilde{B}$; initial blur kernel K; initial latent gradients ∇L; kernel prior parameters $\theta_K = \{\kappa_d, \lambda_d\}$; latent gradient prior parameters $\theta_L = \{\pi_c, v_c\}$.

% Initialize $Q(K)$, $Q(\nabla L)$ and $Q(\sigma^{-2})$
For all m, n, $\mathrm{E}[k_{mn}] = K(m,n)$, $\mathrm{var}[k_{mn}] = 10^4$, $\mathrm{E}[k_{mn}^2] = \mathrm{var}[k_{mn}] + \mathrm{E}[k_{mn}]^2$.
For all i, j, $\mathrm{E}[l_{ij}] = \nabla L(i,j)$, $\mathrm{var}[l_{ij}] = 10^4$, $\mathrm{E}[l_{ij}^2] = \mathrm{var}[l_{ij}] + \mathrm{E}[l_{ij}]^2$.
$\mathrm{E}[\sigma^{-2}] = 1$; *% Set initial noise level*
$\psi = \{\mathrm{E}[\sigma^{-2}], \mathrm{E}[k_{mn}], \mathrm{E}[k_{mn}^2], \mathrm{E}[l_{ij}], \mathrm{E}[l_{ij}^2]\}$ *% Initial distribution*

repeat
  $[\psi^*, D_{KL}^*] = \mathtt{Update}(\psi, \nabla L, \theta_K, \theta_L)$
  $\nabla\psi = \psi^* - \psi$; $\alpha = 1$ *% Get update direction; init. step size*
  $[\alpha^*, D_{KL}] = \mathtt{Line\ search}(\psi, \nabla\psi, \alpha, \mathtt{Update})$ *% Find α to minimize D_{KL}*
  $\psi \leftarrow \psi + \alpha^* \cdot \nabla\psi$ *% Update distribution*
until Convergence: $|D_{KL} - D_{KL}^* < 5 \times 10^{-3}$

$K^{new} = \mathrm{E}[k]$, $\nabla L^{new} = \mathrm{E}[l]$. *% Max marginals*
Output: K^{new} and ∇L^{new}.

$[\psi^*, D_{KL}] =$ **function** $\mathtt{Update}(\psi, \nabla L, \theta_K, \theta_L)$
% Sub-routine to compute optimal update and corresponding D_{KL}
  Run update equations 4.27 through 4.40.
  $\psi^* = \{\mathrm{E}[\sigma^{-2}], \mathrm{E}[k_{mn}], \mathrm{E}[k_{mn}^2], \mathrm{E}[l_{ij}], \mathrm{E}[l_{ij}^2]\}$ *% Collect updates*
$D_{KL}^* = D_{KL}$ at ψ^* *% See Equation 4.19*
Return: ψ^*, D_{KL}^*

the blur kernel and sharp image at the start of the algorithm is an important one. If our initialization is not close enough to the correct answer, the algorithm is likely to converge at the wrong solution. In order to combat this effect, we apply a multi-scale approach to the inference.

We begin by downsampling the image until the corresponding blur kernel is only 3×3 pixels in size. We estimate the amount of downsampling needed to do this based on a user-specified maximum blur kernel size. The user also initializes the value of the blur kernel at this scale (user input will is described in more detail in Section 4.3.4). We initialize the sharp image by running the inference algorithm while holding the blur kernel fixed at the user-specified value.

After running the inference at the coarsest scale, we upsample the result to the next higher-resolution scale, and use this upsampled result as the initialization for

inference at the finer scale. This process is repeated until we converge upon our final result at the finest scale.

At the lowest resolution, the search space is considerably less complex than our initial high-resolution image. Therefore, this approach is not very sensitive to our initialization of the 3×3 blur kernel. As we go to higher levels of the pyramid, the search space becomes more complex. However, our initializations are close enough to the correct solution that this is not a problem.

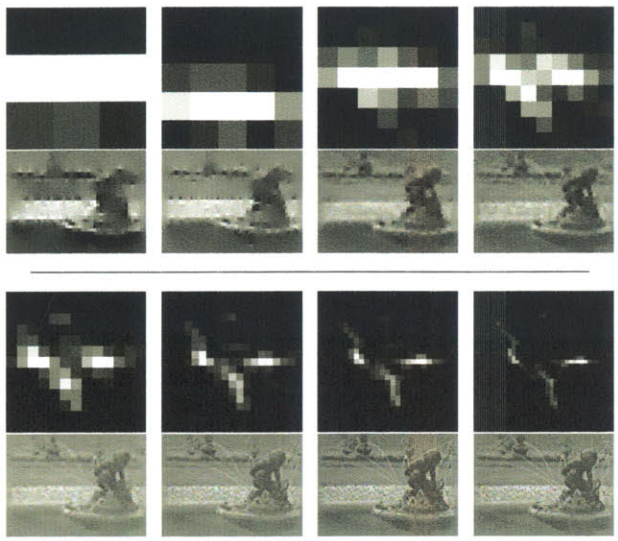


Figure 4-5: The multi-scale inference scheme operating on a blurry image of a fountain. The inferred blur kernel at each scale level is shown along with the corresponding estimated image patch (the top row shows the first four scale levels, from left to right, and the bottom row shows then next four scale levels). Note that the inference algorithm outputs the gradient latent image — the images shown here were reconstructed from the gradients using Poisson reconstruction [46] purely for illustrative purposes. Section 4.4 describes how we remove the blur after we have estimated the blur kernel. A full-color image of this fountain scene along with our deblurred result is shown later in Figure 4-11.

4.3.4 Inference Algorithm Details

The formulation thus far assumes that we will be using the entire blurry input image when performing inference. One might think that using the entire image would provide the most information and therefore be the best option. In practice, however, we find that the inference algorithm performs better if the user instead provides it with a patch from the blurry image that is rich in structure and contains a number of edges. Doing so helps the algorithm avoid areas of the image that are saturated or

are flat (and thus contain no useful information). It also reduces the dimensionality of the search space, which helps both in reducing the number of local minima and allows the algorithm to run much faster.

In Section 4.3 we made the assumption that the blurry image contains no saturated regions in order to allow us to treat the image formation process as linear. In reality, many images will contain at least some saturation. In some cases, the algorithm may not have to deal with this because the saturation may fall outside the particular patch of the image supplied to the algorithm by the user. For some images, however, the user may not be able to select a region of the image that is both large enough to contain enough image structure for the algorithm to work well and avoids saturated regions entirely. In this case, we simply apply a binary mask and mask out all pixels in the blurry image that are saturated, and only run the inference algorithm on the non-saturated regions. Any pixels that are at their maximum whiteness value are assumed to be saturated.

There are two additional inputs that must be supplied to the inference algorithm so that it might be initialized correctly. First, the user must select one of two initial estimates for the 3×3 pixel downsampled version of the blur kernel: either a horizontal bar or a vertical bar. As mentioned in Section 4.3.3, the algorithm is in general robust to this setting, but it is useful in some cases. Second, the user must provide the algorithm with a maximum size of the blur kernel. This parameter is required by the algorithm so that it can determine how many scales of the image it must consider in the multi-scale approach. The size of blur due to camera shake can vary from a few pixels to hundreds of pixels. If we were to set the maximum blur kernel size to a fixed value, either it would be too small to handle the blurs in some images, or it would be too large to adequately resolve the small blurs in others, or both. Setting this parameter manually is therefore necessary, and is also simple to do by examining any blur artifact in the image.

4.4 Blur Removal

Recovering the deblurred image once we have an estimate for the blur kernel is an instance of non-blind image deconvolution and there are a number of methods that have been developed to tackle this problem. We experimented with a number of methods found in the literature (see [17], [34] and [48]) but found that although these algorithms worked well on synthetic images, they were unable to handle real-world blurry photos even when provided with a good estimate of the blur kernel, and

produced a number of artifacts.

Bayesian approaches may also be used for the non-blind deconvolution problem. One could simply run the variational inference algorithm again on the entire blurry gradients image data while holding the blur kernel fixed at its estimated value to recover the deblurred image gradients. The deblurred image can then be recovered from its gradients using Poisson image reconstruction [46]. Alternatively, another Bayesian approach would be to find the MAP estimate of the sharp image given the blurry image and the estimated blur kernel, and using a field of experts prior as described in [40].

Unfortunately, neither of these methods are without problems. The variational approach is very slow, and results in reconstructions that contain a number of artifacts. The MAP approach is faster, but also results in undesirable artifacts (in part, we believe, because of the existence of multiple local optima).

The best algorithm to balance the need for producing visually plausible results and running in a reasonably efficient manner was the Richardson-Lucy (RL) algorithm [37, 29]. The RL algorithm produced results comparable with any of the other methods tested but took minutes to run on large images instead of hours or days as other methods did.

Richardson-Lucy deconvolution is an iterative procedure that converges to a maximum likelihood estimate of the deblurred image under the assumption of Poisson distributed image noise. Let $\boldsymbol{b}$ be a column vector obtained by concatenating all of the columns of $\nabla\tilde{B}$ (the saturation-free inverse-gamma corrected gradient blurry input image). We can then rewrite the convolution operation of Equation 4.7 as a matrix multiplication, where the blur kernel K is replaced by a matrix $\boldsymbol{K}$ that expresses the same blur. We can then rewrite Equations 4.7 and 4.17 as

$$\boldsymbol{b} = \boldsymbol{K}\boldsymbol{l} + \boldsymbol{n}, \tag{4.42}$$

$$\boldsymbol{b}_i = \sum_j \boldsymbol{K}_{ij}\boldsymbol{l}_j + \boldsymbol{n}_i, \tag{4.43}$$

where $\boldsymbol{l}$ and $\boldsymbol{n}$ represent vectorized versions of the gradient latent image ∇L and the noise ∇N. Given this formulation, the Richardson-Lucy update equation at iteration $(r+1)$ is given by

$$\tilde{\boldsymbol{l}}_j^{(r+1)} = \tilde{\boldsymbol{l}}_j^{(r)} \sum_i \frac{\boldsymbol{b}_i}{\tilde{\boldsymbol{b}}_i}\boldsymbol{K}_{ij}, \tag{4.44}$$

where

$$\tilde{\boldsymbol{b}}_i = \sum_j \tilde{\boldsymbol{l}}_j^{(r)} \boldsymbol{K}_{ij}, \tag{4.45}$$

and we initialize $\tilde{\boldsymbol{l}}^{(0)}$ to some starting image guess.

The Richardson-Lucy update equation is guaranteed to produce non-negative output values that increase the likelihood at every iteration. We must set the number of iterations ahead of time – a larger number of iterations increases the processing time and also brings us closer to the maximum likelihood estimate. One advantage of the RL algorithm is that because of the form of the update equation, the estimate moves very rapidly towards the true maximum likelihood solution in the first few iterations, and later iterations result in successively smaller changes. Maximum likelihood estimates in general, because they do not contain any prior information about what images should look like, tend to explain the observed image data very well but produce unnatural artifacts in the estimate. Qualitatively, we find that for deblurring real-world images, the initial RL iterations remove the majority of the blur caused by camera motion while the later iterations, which perform less substantial refinements to the estimate, result in the undesirable image artifacts common to maximum likelihood methods. We found through empirical testing that a value of ten RL iterations was the optimal setting to reliably deblur the image without introducing undesirable image artifacts in real-world images.

We utilize Matlab's built in `deconvlucy` function to perform the RL deblurring. This implementation of the algorithm performs updates very quickly through use of the fast Fourier transform. Some minor preprocessing needs to be done before applying the Richardson-Lucy algorithm. First we threshold the blur kernel so that all values that fall below 15% of the maximum are set to zero. This is done in order to reduce noise present in our kernel estimate. Second, we taper the discontinuities present at the edges of the input blurry image using Matlab's `edgetaper` function. This is necessary in order to reduce ringing in the output estimate.

4.5 Complete Deblurring Algorithm

The overall deblurring algorithm consists of four parts: preprocessing, inference, blur removal, and postprocessing. In the preprocessing step, the user-selected region of the image is converted to grayscale and gradients are computed. The algorithm then

determined the number of scales needed based on the maximum blur size. Inference is performed by iterating over scales as described in Section 4.3.3. The inference procedure presented in Algorithm 1 is performed at each scale to produce an estimate that is upsampled and used as an initialization for the next scale. The blur is removed according to the procedure described in Section 4.4 using the input blurry image and the estimated blur kernel produced at the largest scale of the inference. The resulting latent image is then gamma corrected and histogram matched to the original blurry image to preserve the overall color tone. Below we present the pseudo code for the overall deblurring algorithm, `Image Deblur`.

4.6 Results

We tested the deblurring algorithm on a number of real images blurred by camera shake. Because the ground truth sharp image is not available, the quality of the results must be judged by the subjective appearance of the deblurred result. The images on the following pages demonstrate the performance of our algorithm when dealing with varying degrees of blur and saturation. With the exception of the fountain and cafe scenes, all blurry images were obtained from personal photo collections and taken using consumer-level digital cameras (the fountain and cafe scene were taken with a high-end digital SLR camera). For all of the results, the deblurred result is shown along with the inferred blur kernel.

The running time of the algorithm is dependent on the size of the user-selected input patch. For a 128×128 pixel patch, the algorithm implemented in Matlab takes approximately 10 minutes to run. All convolution operations are done using FFTs, so the run time is $O(N \log N)$ where N is the number of pixels.

Figure 4-6 shows an image degraded by a small and simple camera blur that is sharpened by our algorithm. The user-selected region in the image contains text that is rich in structure, which helps the algorithm to find an accurate blur estimate. Figure 4-7 also contains a relatively small camera motion but includes a moving subject. Because the user-selected patch consists primarily of the moving subject, the inferred blur kernel takes into account the motion of the subject as well as the camera.

Figures 4-8 through 4-11 contain progressively larger and more complex blur kernels that the algorithm is able to correctly infer. These images also demonstrate the types of artifacts introduced in the blur removal stage. The artifacts are generally more noticeable in images with more significant blurs. The most common failure

Algorithm 2 `Image Deblur`

Require: Blurry image B; selected sub-window P; maximum blur size ϕ; overall blur direction o (= 0 for horiz., = 1 for vert.); parameters for prior on ∇L: $\theta_L = \{\pi_c^s, v_c^s\}$; parameters for prior on K: $\theta_K = \{\pi_d, \lambda_d\}$.

Convert P to grayscale.
Inverse gamma correct P (default $\gamma = 2.2$).
$\nabla P_x = P \otimes [1, -1]$. *% Compute gradients in x*
$\nabla P_y = P \otimes [1, -1]^T$. *% Compute gradients in y*
$\nabla P = [\nabla P_x, \nabla P_y]$. *% Concatenate gradients*
$S = \lceil -2 \log_2 (3/\phi) \rceil$. *% # of scales, starting with* 3×3 *kernel*

% Loop over scales, starting at coarsest
for $s = 1$ to S **do**
  ∇P^s =`imresize(`∇P,$(\frac{1}{\sqrt{2}})^{S-s}$,`'bilinear')`. *% Rescale gradients*
  % Initial kernel and gradients
  if (s==1) **then**
    $K^s = [0, 0, 0; 1, 1, 1; 0, 0, 0]/3$. If $(o == 1)$, $K^s = (K^s)^T$.
    $[K^s, \nabla Lp^s]$ = `Inference(`∇P^s,K^s,∇P^s,θ_K^s,θ_L^s`)`, keeping K^s fixed.
  else
    % Upsample estimates from previous scale
    ∇L_p^s = `imresize(`∇L_p^{s-1},$\sqrt{2}$,`'bilinear')`.
    K^s = `imresize(`K^{s-1},$\sqrt{2}$,`'bilinear')`.
  end if
  $[K^s, \nabla Lp^s]$ = `Inference(`∇P^s,K^s,∇Lp^s,θ_K^s,θ_L^s`)`. *% Run inference*
end for

Set elements of K^S that are $< max(K^S)/15$ to zero. *% Threshold kernel*
B = `edgetaper(`B,K^S`)`. *% Reduce edge ringing*
L = `deconvlucy(`B,K^S,10`)`. *% Run RL for* 10 *iterations*
Gamma correct L (default $\gamma = 2.2$).
Histogram match L to B using `histeq`.
Output: L, K^S.

mode of our algorithm is not in the inference step, but rather in the non-blind deconvolution step. Figure 4-10 provides a good example of how the algorithm can fail. The blur kernel that is inferred is very close to accurate (as evidenced by the closeup of a specularity shown in Figure 4-13), yet the deblurred output shows significant artifacts.

Figure 4-12 shows a closeup of Figure 4-9 and another similar image with a large blur pattern. It demonstrates how image structures not apparent in the original blurry image such as specularities and text can be clearly discerned in the deblurred

output. It also shows that the deblurring process can make color noise artifacts visible in the output image. This noise is not as visible in the blurry image because the blur process has smoothed it out to a large extent. The presence of such noise is not a failure of our algorithm; rather, it is the case that because of the low-light setting of the scene being imaged, an image of the same scene in which there was no camera motion (which is what we seek to recover) would contain color noise artifacts. Such noise can be removed (to a large extent) from the deblurred output through the use of a median filter or other standard noise removal technique.

In most images, the blur kernel is usually revealed in the image somewhere in the form of a point light source that has been transformed by the blur. Figure 4-13 illustrates these patterns for four distinct images, and gives us an idea of how accurate the results from our inference procedure are. Because the types of patterns shown in this figure are easily noticed in the blurry image, they can be very useful in helping the user select the maximum size of the blur kernel (a necessary input for our algorithm).

Figures 4-14 and 4-15 show the performance of our algorithm on scenes with very significant saturation. In both cases, the user-selected region contains relatively little, but not zero, saturation relative to the image as a whole. Our results show some artifacts visible near the saturated regions, but the unsaturated regions are recovered reasonably well.

Finally, as a baseline comparison, we ran Matlab's built in blind deconvolution routine, `deconvblind`, on the fountain and cafe scenes. This routine implements an iterative deconvolution as described in [4] and [22]. It uses the Richardson-Lucy algorithm to iteratively estimate the blur kernel while holding the image fixed, and vice versa, until convergence. The results of this technique, shown in Figure 4-16 are very poor compared to the results of our algorithm, shown in Figures 4-11 and 4-15.

(a) Original blurry image

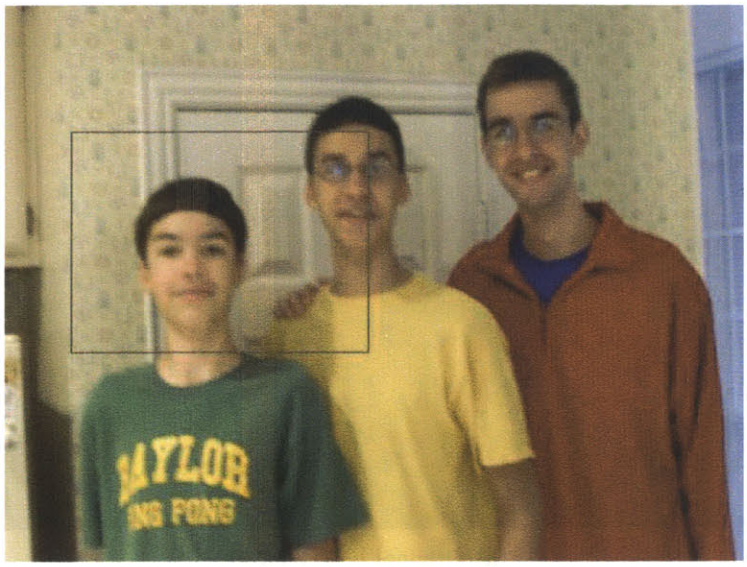
(b) Deblurred result (inferred blur kernel shown at top right)

Figure 4-6: Deblurring a scene with a small blur. The gray rectangle on the input image indicates the patch selected by the user. Note the crisp text on the deblurred output.

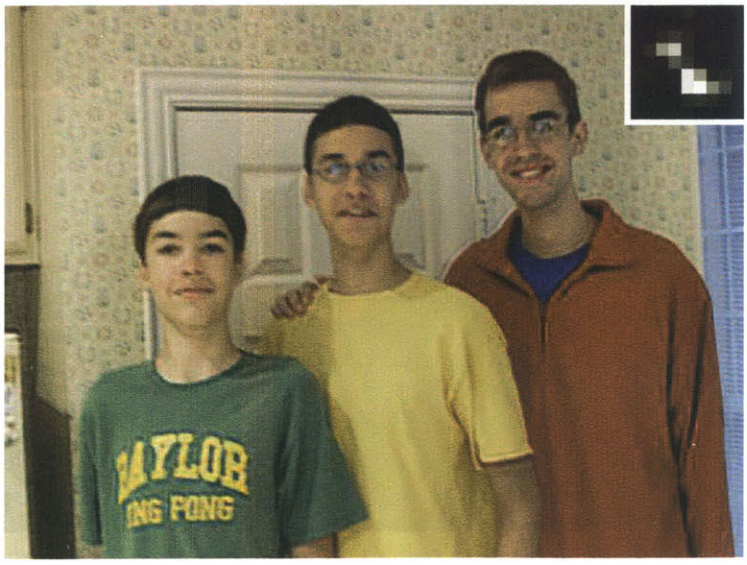
(a) Original blurry image

(b) Deblurred result (inferred blur kernel shown at top left)

Figure 4-7: Deblurring a scene with complex motions. The extent of camera shake is relatively small, but the child in the image is translating and rotating his arm. The region of the image selected by the user, illustrated by the gray rectangle on the top image, consists primarily of the child's face. The blur kernel inferred by our algorithm therefore describes the combined effect of the camera motion and the child's overall translation. The child's face and shirt appear sharp in the resulting deblurred image. The child's arm remains blurred in the output since its motion is not captured by the blur kernel. The scene background is also blurry since it was not translating like the child was.

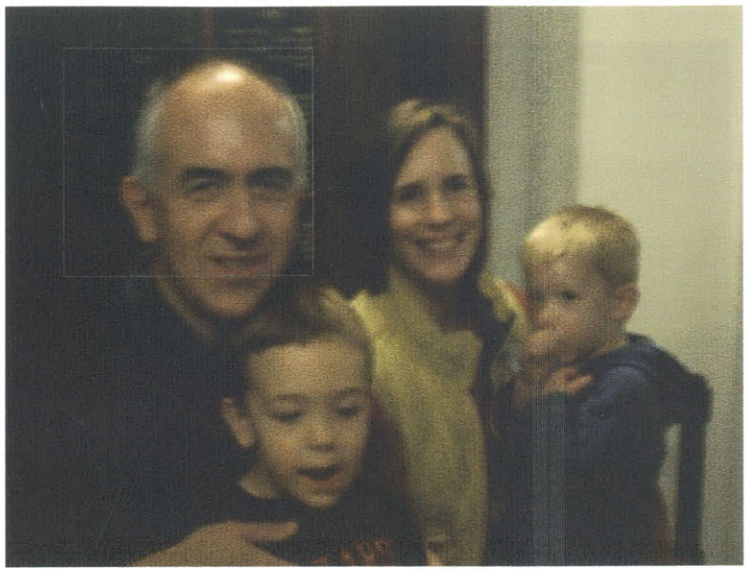


(a) Original blurry image

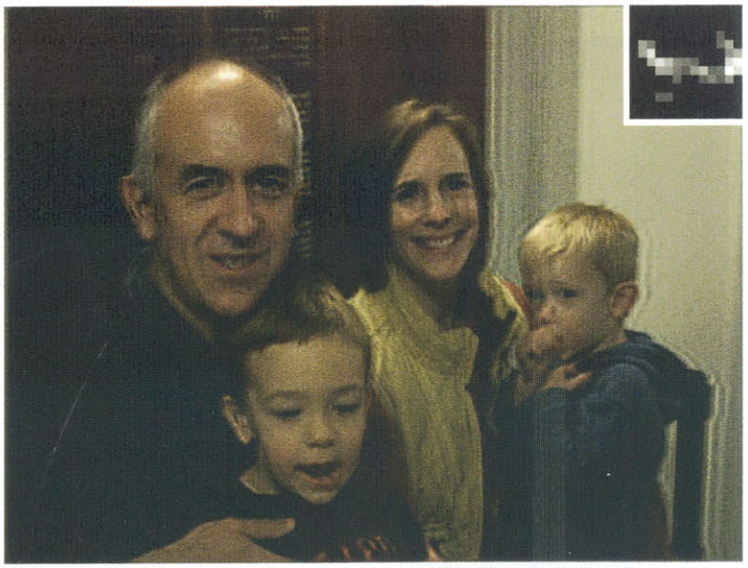


(b) Deblurred result (inferred blur kernel shown at top right)

Figure 4-8: Deblurring a scene with moderate blur. Note that the fine detail of the wallpaper is visible in the deblurred output.



(a) Original blurry image



(b) Deblurred result (inferred blur kernel shown at top right)

Figure 4-9: Deblurring a scene with a large blur. Notice the complex shape of the inferred blur kernel. See Figure 4-12 for a closeup view. Some ringing artifacts are visible in the output, particularly on the right half of the image near the edges of the wall, chair and child's body.

(a) Original blurry image

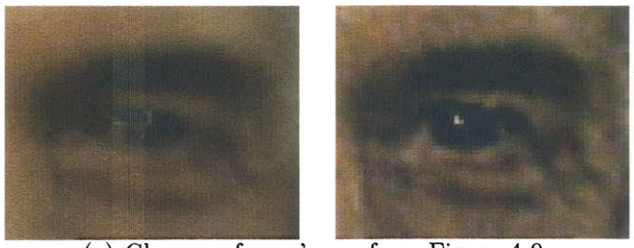
(b) Deblurred result (inferred blur kernel shown at bottom left)

Figure 4-10: Deblurring a scene with very a large blur (failure case). The algorithm is able to infer the blur kernel quite well, as evidenced by the closeup view in Figure 4-13. The output, however, still contains many artifacts and is not properly deblurred.

(a) Original blurry image

(b) Deblurred result (inferred blur kernel shown at bottom left)

Figure 4-11: Deblurring a scene with a very large and very complex blur. Note the tri-lobed nature of the inferred blur kernel. Other blind image deconvolution methods would not be able to estimate such a complex blur pattern (closeup view in Figure 4-13 shows that our kernel estimate is reasonably accurate). The output is significantly sharpened, although it does contain some ringing artifacts.



(a) Closeup of man's eye from Figure 4-9

(b) Closeup from another image of the family in Figure 4-9

Figure 4-12: Closeup of deblurring scenes with large blurs. In 4-9(a) the algorithm is able to deblur a specularity in the man's eye. In 4-9(b) the text on the child's shirt is legible in the deblurred result. Color noise artifacts due to low light exposure are visible in both deblurred results.

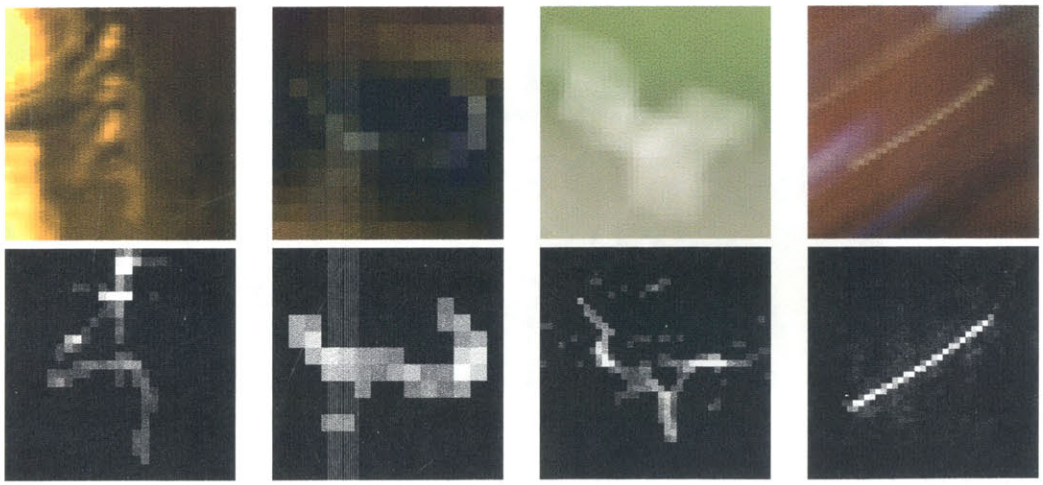
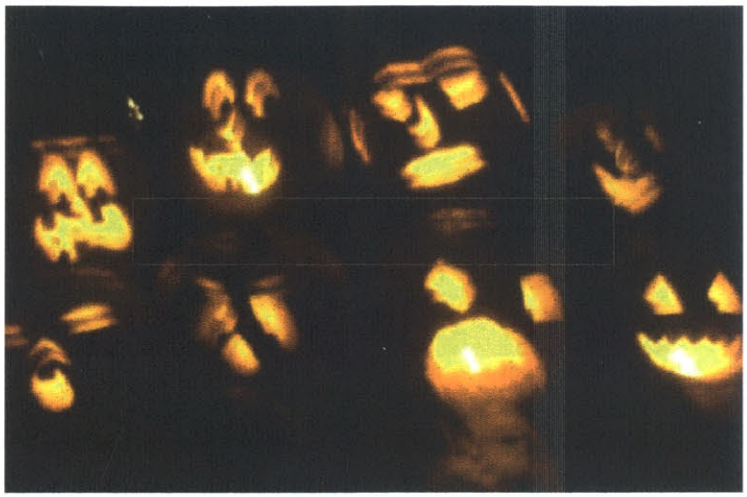


Figure 4-13: Each column shows on top a patch extracted from an image where the true blur pattern has been revealed, and shows on bottom the inferred blur kernel for the image using our method. The patches shown come from (going left to right) the cafe, family, fountain and fruit scenes of Figures 4-15(a), 4-9(a), 4-11(a) and 4-10(a), respectively. In the cafe image, two lights give a dual image of the kernel. In the fountain scene, a white square is transformed by the blur. The true kernel in the other two images is revealed by specularities that have been blurred by the camera motion.



(a) Original blurry image

(b) Deblurred result (inferred blur kernel shown at top left)

Figure 4-14: Deblurring a scene with significant saturation. The user-selected patch (the long thin region between the two rows of pumpkins, outlined by a gray rectangle in 4-14(a)) has limited saturation. As a result, our algorithm is able to capture the highly discontinuous double-exposure form of the blur kernel. The deblurred result undoes this double exposure reasonably well.

(a) Original blurry image

(b) Deblurred result (inferred blur kernel shown at top left)

Figure 4-15: Deblurring a scene with heavy saturation and a very complex blur. Note all of the saturated light sources located throughout the input image. Our algorithm is able to recover an accurate estimate of the blur kernel, as evidenced by the closeup view in Figure 4-13. The deblurred output is much sharper in non-saturated regions of the image (see, for example, the lettering on the man's jacket and the stripes on the woman's shirt on the left side of the image), but contains significant artifacts near all of the saturated image regions.

(a) Deblurred fountain scene from Figure 4-11(a) using Matlab's `deconvblind`

(b) Deblurred cafe scene from Figure 4-15(a) using Matlab's `deconvblind`

Figure 4-16: Baseline experiments, using Matlab's blind deconvolution algorithm `deconvblind` on two of the blurry images previously shown. The algorithm was initialized in both cases with a Gaussian blur kernel, similar in size to the blur artifacts. The results shown here are significantly worse than the results obtained when using our method.

Chapter 5

Discussion

In the preceding chapters we have presented algorithms for solving two specific computer vision problems: color constancy and blind image deconvolution. Both problems share some basic similarities, in that they are both underconstrained bilinear inverse problems. Together they encompass many of the types of problems encountered in low-level computer vision.

There are, however, significant distinctions between the color constancy and blind image deconvolution problems. As a result, we utilized different solution strategies for the two problems, and found that spatial incorporating spatial image statistics into our algorithms was not equally helpful in both cases.

5.1 Estimation Techniques

In Chapter 2 we described two general categories of Bayesian methods: estimation by minimizing the Bayesian expected loss, which encompasses MAP estimation, and the variational Bayesian approach. In that chapter, we suggested that the variational approach is likely to provide a more accurate estimate because it allows for tractable marginalization and thus describes probability mass rather than probability values.

Our hypothesis that using MMSE estimation with the variational method would outperform MAP estimation was confirmed in the case of the blind image deconvolution problem. MAP estimation was originally attempted to perform deblurring, keeping all other relevant parts of the deblurring algorithm the same as what was finally used, but this failed. If the noise variance was set to a high value, the MAP solution tended to move towards a two-tone image in which nearly all gradients were zero. If the noise variance was set to a low value, MAP estimation would produce

a delta-function blur kernel. Varying the noise parameter would produce different results within this spectrum, but none of them was able to reliably capture the true blur kernel. The variational approach with MMSE estimation, on the other hand was able to produce reliable and accurate estimates of the blur kernel for a variety of images.

In the color constancy problem, we utilized MAP estimation in our final algorithm. It is possible that a variational approach would yield improved results in this case as well, but certain properties of the color constancy problem made this much more difficult to implement than in the deblurring problem. Most significantly, the realizability constraints for surface reflectance and illumination spectra required very complex prior probability distributions. The variational method has been formulated to handle positivity constraints on the random variables being estimated (e.g., when using mixtures of rectified Gaussians or exponentials as priors). However, no such formulation of the variational approach currently exists that can account for general inequality constraints on linear transformations of random vectors (e.g., the truncated Gaussian priors we utilized for color constancy). When using MAP estimation we could incorporate these complex constraints within our numerical search routine by restricting our search space. An analogous thing cannot be done using the variational approach because it relies on analytical parameter updates.

5.2 Prior Probability Models

One of the things that allowed our deblurring algorithm to be successful was the fact that we were able to accurately model the statistics of natural images. As shown in Figure 4-4, our mixture of Gaussians prior provided a very accurate approximation to the empirical distribution. In our color constancy algorithm, on the other hand, we modeled the distributions over surface reflectance and illumination coefficients with Gaussians, which provide relatively poor approximations to the empirical distributions (the approximation is even worse than the standard case shown in Figure 3-3).

Again, the modeling choices in the color constancy algorithm were dictated largely by implementation issues. As described in Section 3.5.2, the dimensionality of our search space required us to search only over illumination coefficients and compute the associated MAP estimates for the surface coefficients given the illumination at every iteration of the search. In order to make this approach feasible, it was necessary to have a closed-form solution when estimating the surface coefficients within a given

iteration. More complex priors over our surface coefficients would likely have produced a more accurate estimate, but would have also required the algorithm to perform search routines nested within search routines, increasing the computation time by orders of magnitude.

5.3 Use of Spatial Information

In the case of blind image deconvolution, the prior over natural image gradients was a critical element for the deblurring algorithm to be successful. Previous approaches could only work with very specific (and unrealistic in the case of camera shake) blurs, or required additional image or video information to solve the problem. Because we were able to place strong realistic priors on our sharp images, the prior over blur kernels in our approach was reasonably weak, enforcing only positivity and sparsity. This allowed us to estimate very complex blur kernels that corresponded to realistic camera motions.

In the case of color constancy, the formulation of the problem motivated us to incorporate the spatial statistics of natural images through the use of spatio-spectral basis functions. The extra information contained within these basis functions over the standard spectral basis functions is evident through the decreased error when reconstructing reflectance images (see Figure 3-5). Using these basis functions for color constancy did result in improved performance over the standard Bayesian method but not by as much as one might expect based on the amount of extra information contained within the spatio-spectral basis functions (we achieved a 4.3% reduction in error for real images).

One factor that likely affected the quality of the color constancy results is the increased complexity of the spatio-spectral model. While offering increased descriptive power, this model also exhibits an increased sensitivity to spatially-varying noise, which may affect the quality of the results.

More significantly, though, the discrepancy between how much local image statistics affected the deblurring and color constancy problems illustrates an important difference between the two problems. When an image is blurred, its local structure is affected. That is, the value of a given pixel in the blurry image is dependant upon the values of a particular local region in the original sharp image. Therefore, even if two pixels have the same value in the original image, their values in the blurry image are likely to be different, since the local regions around the sharp image pixels will probably be different even if the pixels are the same. The result of this is that

local statistics that were present in the sharp image will not be present in the blurry image. If we know what those local properties are, that gives us useful information in recovering the sharp image from the blurry one.

In the case of illumination falling upon a scene, the effect is largely global, not local. Consider the an image of a scene under white light. If a different illuminant falls upon the same scene, the resulting image will still contain, for the most part, the same general local structure as the original white-light image. The image as a whole will change its color – for example, of a blue light shines on the scene, the entire will look like it has been tinted blur. But if you consider a particular pair of pixels, the pixel that is darker or more blue under the white light will usually remain darker or more blue under the new illuminant. Of course, this is not true absolutely, which is why we get some improvement in color constancy when using spatial statistics.

5.4 Conclusions

This thesis has examined the use of spatial image statistics within the context of Bayesian estimation theory and applied it to the problems of color constancy and blind image deconvolution. In the former we have shown that incorporating spatial statistics into the solution does yield improved results, but these improvements are not very substantial. In the case of blind image deconvolution, on the other hand, we have shown how utilizing natural image statistics with variational Bayesian estimation theory can produce state-of-the-art results for deblurring images corrupted by camera shake.

Bibliography

[1] H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In *15th conference on Uncertainty in Artificial Intelligence*, pages 21–30, 1999.

[2] B. Bascle, Andrew Blake, and Andrew Zisserman. Motion Deblurring and Superresolution from an Image Sequence. In *ECCV (2)*, pages 573–582, 1996.

[3] Moshe Ben-Ezra and Shree K. Nayar. Motion-Based Motion Deblurring. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):689–698, 2004.

[4] D. Biggs and M. Andrews. Acceleration of iterative image restoration algorithms. *Applied Optics*, 36(8):1766–1775, 1997.

[5] David H. Brainard. An ideal observer for appearance: reconstruction from samples. Technical report, UCSB Vision Labs, Department of Psychology, UC Santa Barbara, Santa Barbara, CA, 1995.

[6] David H. Brainard and William T. Freeman. Bayesian color constancy. *Journal of the Optical Sociatey of America A*, 14(7):1393–1411, 1997.

[7] Gavin J. Brelstaff, C. Alejandro Parraga, Tom Troscianko, and D. Carr. Hyperspectral camera system: acquisition and analysis. In *Proc. of the Intl. Society for Optical Engineering*, volume 2587, pages 150–159, 1995.

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

[9] J.N. Caron, N.M. Namazi, and C.J. Rollins. Noniterative blind data restoration by use of an extracted filter function. *Applied Optics*, 41(32):68–84, November 2002.

[10] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. *Computer Graphics*, 31(Annual Conference Series):369-378, 1997.

[11] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006)*, To Appear, 2006.

[12] Mário A. T. Figueiredo and Robert D. Nowak. An em algorithm for wavelet-based image restoration. *IEEE Trans. on Image Processing*, 12(8):906–916, 2003.

[13] Graham D. Finlayson. Color constancy in diagonal chromaticity space. In *Intl. Conf. on Computer Vision*, pages 218–223, June 1995.

[14] David A. Forsyth. A novel algorithm for color constancy. *Intl. Journal of Computer Vision*, 5(1):5–36, 1990.

[15] Brian Funt, Vlad Cardei, and Kobus Barnard. Learning color constancy. In *Proc. of Fourth Color Imaging Conf*, pages 58–60, 1996.

[16] Brian Funt, Vlad Cardei, and Kobus Barnard. Neural network color constancy and specularly reflecting surfaces. In *Proc. AIC Colour*, Kyoto, Japan, May 1997.

[17] D. Geman and G Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(3):367–383, 1992.

[18] S. Gull. Bayesian inductive inference and maximum entropy. In J. Skilling, editor, *Maximum Entropy and Bayesian Methods*, pages 54 71. Kluwer, 1998.

[19] Berthold K.P. Horn. Determining lightness from an image. *Computer Graphics and Image Processing*, 3(1):277–299, December 1974.

[20] Canon Inc. What is optical image stabilizer? online publication: http://www.canon.com/bctv/faq/optis.html, 2006.

[21] A. Jalobeanu, L. Blanc-Fraud, and J. Zerubia. Estimation of blur and noise parameters in remote sensing. In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, Orlando, FLA, USA, May 2002.

[22] Peter A. Jansson. *Deconvolution of Images and Spectra.* Academic Press, 1997.

[23] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, May 1996.

[24] Edwin H. Land. Recent advances in Retinex theory and some implications for cortical computations: color vision and the natural image. *Proc. of the Natl. Academy of Science*, 80:5163–5169, August 1983.

[25] Edwin H. Land and John J. McCann. Lightness and Retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971.

[26] Anat Levin and Yair Weiss. User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior. In *Intl. Conf. on Computer Vision*, volume 1, pages 602–613, 2004.

[27] Anat Levin, Assaf Zomet, and Yair Weiss. Learning How to Inpaint from Global Image Statistics. In *Intl. Conf. on Computer Vision*, pages 305–312, 2003.

[28] X. Liu and A. Gamal. Simultaneous image formation and motion blur restoration via multiple capture. In *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, volume 3, pages 1841–1844, 2001.

[29] L. Lucy. Bayesian-based iterative method of image restoration. *Journal of Astronomy*, 79:745–754, 1974.

[30] Laurence T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *Journal of the Optical Sociatey of America A*, 3(10):1673–1683, October 1986.

[31] Laurence T. Maloney. *Color vision: from genes to perception*, chapter 19, pages 387–422. Cambridge U. Press, Cambridge, U.K., 1999. chapter title: "Physics-based approaches to modeling surface color perception".

[32] Laurence T. Maloney and Brian A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Sociatey of America A*, 3(1):29–33, January 1986.

[33] James Miskin and David J. C. MacKay. Ensemble Learning for Blind Image Separation and Deconvolution. In M. Girolani, editor, *Adv. in Independent Component Analysis*. Springer-Verlag, 2000.

[34] R. Neelamani, H. Choi, and R.G. Baraniuk. Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Trans. on Signal Processing*, 52:418–433, Feburary 2004.

[35] Jussi P.S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of Munsell colors. *Journal of the Optical Sociatey of America A*, 6(2):318–322, February 1989.

[36] Alex Rav-Acha and Shmuel Peleg. Two motion-blurred images are better than one. *Pattern Recognition Lctters*, pages 311–317, 2005.

[37] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Sociatey of America A*, 62:55–59, 1972.

[38] Javier Romero, Antonio Garca-Beltran, and Javier Hernandez-Andres. Linear bases for representation of natural and artificial illuminants. *Journal of the Optical Sociatey of America A*, 14(5):1007–1014, May 1997.

[39] Charles Rosenberg, Thomas Minka, and Alok Ladsariya. Bayesian color constancy with non-gaussian models. In *Neural Info. Processing Systems*, 2003.

[40] Stefan Roth and Michael J. Black. Fields of Experts: A Framework for Learning Image Priors. In *Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, 2005.

[41] Daniel L. Ruderman, Thomas W. Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *Journal of the Optical Sociatey of America A*, 15(8):2036–2045, August 1998.

[42] Barun Singh, William T. Freeman, and David H. Brainard. Exploiting spatial and spectral image regularities for color constancy. In *Third International Workshop on Statistical and Computational Theories of Vision*, October 2003.

[43] Marshall F. Tappen, Bryan C. Russell, and William T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Third International Workshop on Statistical and Computational Theories of Vision*, 2003.

[44] F. Tsumuraya, N. Miura, and N. Baba. Iterative blind deconvolution method using Lucy's algorithm. *Astron. Astrophys.*, 282(2):699–708, Feb 1994.

[45] Brian A. Wandell. The synthesis and analysis of color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:2–13, 1987.

[46] Yair Weiss. Deriving intrinsic images from image sequences. In *Intl. Conf. on Computer Vision*, volume 2, pages 68–75, July 2001.

[47] Y. Yitzhaky, R. Milberg, S. Yohaev, and N.S. Kopeika. Comparison of direct blind deconvolution methods for motion-blurred images. *Applied Optics*, 38(20):4325–32, July 1999.

[48] C. Zarowin. Robust, noniterative, and computationally efficient modification of van Cittert deconvolution optical figuring. *Journal of the Optical Sociatey of America A*, 11(10):2571–83, October 1994.